



UNIVERSIDAD ANDINA DEL CUSCO

FACULTAD DE INGENIERÍA Y ARQUITECTURA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



Tesis

Estudio comparativo de redes neuronales artificiales aplicadas a la
identificación de violencia escolar en las instituciones educativas

Presentado por:

Bach. Luis Ernesto Colchado Soncco

Para optar al Título Profesional de

Ingeniero de Sistemas

Asesor: Mgt. Edwin Carrasco Poblete

Cusco – Peru

2018



DEDICATORIA

Dedico este trabajo a mis padres Lucho y Lourdes, así como a mi hermano Gabriel, que son las personas que están más al tanto de mí apoyándome, aconsejándome y dándome la confianza necesaria para levantarme cuando me equivoco. También lo dedico a mi tío padrino Alberto por su gran generosidad, así como a mi mamá Juana por el cariño y apoyo, como también a mi querida mamá Mina que lamentablemente ya no está con nosotros. Dedico esta investigación también a mis tíos todos ellos muy peculiares, divertidos y muy inteligentes, primos y sobrinos que siempre de alguna forma están pendientes de mi bienestar, de la misma forma a mis amigos que son las personas con las que puedo compartir buenos momentos y a mis compañeros de universidad por las experiencias vividas en esta época estudiantil.



AGRADECIMIENTOS

Agradezco a mis padres y hermano, pues en el proceso de desarrollo de esta tesis fueron sus palabras de motivación las que me dieron las principales energías para culminar esta investigación.

También agradezco a mi asesor Mgt. Edwin Carrasco por la ayuda y los consejos brindados en el proceso de este trabajo investigativo, así como a la Mgt. Pilar Vanessa Hidalgo por sus conocimientos compartidos conmigo, dichos conocimientos me ayudaron mucho en la parte teórica de la investigación, de igual forma agradezco a mi dictaminante Mgt. Mónica Marca por las observaciones realizadas, pues esto permitió mejorar la calidad de esta investigación.

Finalmente a toda mi familia, docentes de la universidad, compañeros y amigos, creo que toda experiencia te da conocimiento. Todas las actividades y conversaciones que tuve con ellos ayudó directa o indirectamente a la conclusión de esta investigación.



RESUMEN

Un problema social que afecta a los estudiantes de las instituciones educativas es la violencia escolar tanto en sus hogares por parte de sus padres o apoderados, como en las mismas instituciones por parte de docentes o compañeros. A pesar de que el gobierno de Perú inicio proyectos para hacer frente y de esa forma disminuir los casos de violencia escolar, aun estos están en aumento, pues no existe hasta el momento una forma eficiente de identificarlos.

Por otra parte, las redes neuronales artificiales son modelos computacionales que emulan en una parte muy simple el funcionamiento de las neuronas biológicas, estos pueden aprender de problemas y situaciones ya resueltas, adquiriendo de esa forma la capacidad de realizar clasificación y predicción, entonces se podría pretender utilizar una red neuronal artificial con el fin de predecir los casos de violencia escolar en las instituciones educativas, pero existe una limitante, la cual es que estos modelos computacionales son muy diversos distinguiéndose en los tipos de redes, en los métodos de aprendizaje y además las diferentes funciones de activación para las neuronas artificiales. La presente investigación tuvo como objeto determinar a través de un estudio comparativo la red neuronal artificial que tenga un mayor nivel de eficiencia en la identificación de violencia escolar en las instituciones educativas, para cumplir con este propósito primero se recolectó información de los centros educativos que fue utilizada como datos de entrada en las redes neuronales artificiales. Las seis redes neuronales artificiales consideradas en esta investigación están divididas en tres tipos de redes y métodos de aprendizaje y estas son: la red PMS que es un tipo de red Perceptrón Multicapa, que utiliza el método de aprendizaje de Retropropagación y función de activación Sigmoidal. La red PMH comparte el mismo tipo de red y método de aprendizaje que PMS, pero con la única diferencia que esta red utiliza función Tangente Hiperbólica en lugar de la función Sigmoidal. Las redes BRG y BRC son de tipo Redes Neuronales de Base Radial, ambas con el método de aprendizaje Híbrido, la única diferencia es que la primera utiliza la función de activación Gaussiana y la segunda la función de activación Inversa Cuadrática. Las redes RRS y RRH son de tipo Redes Neuronales Recurrentes ambas utilizan el método de aprendizaje Recurrente en Tiempo Real, de igual forma que en los anteriores casos estas redes se distinguen en la función de activación que utilizan, RRS utiliza la función Sigmoidal y RRH la función Tangente Hiperbólica. Estas redes fueron desarrolladas con el lenguaje de programación C++, para luego realizar la experimentación de cada una de ellas, con la finalidad de determinar la que alcanzó un mayor nivel de eficiencia.



ABSTRACT

A social problem that affects the students of educational institutions is school violence both in their homes by their parents or guardians, and in the same institutions by teachers or peers. Despite the fact that the government of Peru started projects to deal with and thus reduce the number of cases of school violence, these are still increasing, as there is no efficient way to identify them.

On the other hand, artificial neural networks are computational models that emulate in a very simple part the functioning of biological neurons, they can learn from problems and situations already resolved, acquiring in this way the ability to perform classification and prediction, then it could be try to use an artificial neural network in order to predict the cases of school violence in educational institutions, but there is a limitation, which is that these computational models are very different, distinguishing between types of networks, methods of learning and also the different activation functions for artificial neurons. The objective of the present investigation was to determine, through a comparative study, the artificial neural network that has a higher level of efficiency in the identification of school violence in educational institutions. In order to fulfill this purpose, information was first collected from the educational centers that were used as input data in artificial neural networks. The six artificial neural networks considered in this research are divided into three types of networks and learning methods and these are: the PMS network which is a type of Multilayer Perceptron network, which uses the Backpropagation learning method and Sigmoid activation function. The PMH network shares the same type of network and learning method as PMS, but with the only difference that this network uses Hyperbolic Tangent function instead of the Sigmoid function. The BRG and BRC networks are of the Radial Base Neural Networks type, both with the Hybrid learning method, the only difference being that the former uses the Gaussian activation function and the latter the Quadratic Inverse activation function. The RRS and RRH networks are of the Recurrent Neural Networks type, both use the Recurrent Real-Time Learning method, in the same way that in the previous cases these networks are distinguished in the activation function they use, RRS uses the Sigmoid function and RRH the Hyperbolic Tangent function. These networks were developed with the C++ programming language, to then perform the experimentation of each of them, in order to determine which reached a higher level of efficiency.



INTRODUCCIÓN

Las redes neuronales artificiales son modelos computacionales que simulan de mínima forma el funcionamiento de las neuronas biológicas, y que en base a un proceso de aprendizaje de situaciones estos adquieren la capacidad de clasificación o predicción de nuevas situaciones de una misma naturaleza. Estos modelos se diferencian en el tipo de red, método de aprendizaje y las funciones de activación que utilizan, además de otros indicadores que pueden variar su comportamiento y desempeño.

La violencia escolar en los estudiantes de instituciones educativas es un problema social que en los últimos años ha ido en aumento, en el Perú el gobierno ha iniciado muchos proyectos y campañas para disminuir este problema en los estudiantes, pero lamentablemente aun estos persisten.

Entonces, en base a lo antes mencionado se puede utilizar las redes neuronales artificiales como una herramienta de detección para poder identificar los casos de violencia escolar en los estudiantes de instituciones educativas, pero para conocer la red neuronal artificial que tenga un mayor nivel de eficiencia en la identificación de violencia escolar se pretende realizar un estudio comparativo, entonces la pregunta que nos plantemos es si: ¿El estudio comparativo permitirá determinar la red neuronal artificial que tenga un mayor nivel de eficiencia en la identificación de violencia escolar en las instituciones educativas? En base a esta interrogante nuestro objetivo es determinar la red neuronal artificial que tenga un mayor nivel de eficiencia en la identificación de violencia escolar a través del estudio comparativo.

Para lograr dicho objetivo primero se recolectó la información de casos de violencia escolar de estudiantes en las instituciones educativas de la provincia de Urubamba, esta se prepara, y luego en base a la información preparada se diseña una base de datos para el procesamiento de las redes neuronales artificiales.

También es necesario desarrollar y ajustar las redes neuronales artificiales que sean capaces de realizar la predicción de violencia escolar, tomando en cuenta la teoría de sus tipos de red, sus secuencias de aprendizaje y sus otros indicadores.

Posteriormente se realiza la experimentación de cada red neuronal artificial para conocer los niveles de eficiencia que alcanzan, permitiéndonos finalmente determinar la que tenga aun mayor nivel de eficiencia en la identificación de violencia escolar en las instituciones educativas.



El contenido de la presente tesis está establecido de la siguiente manera:

En el capítulo I se presentan los aspectos generales de la investigación, como es la descripción y el planteamiento del problema principal, además de los objetivos establecidos, así como el planteamiento de la hipótesis.

En el capítulo II se desarrollan las bases teóricas de los temas considerados en esta investigación, además de los antecedentes que fueron una base para el desarrollo de esta investigación.

En el capítulo III se realiza el desarrollo de la metodología de la investigación.

En el capítulo IV se lleva a cabo el desarrollo del estudio comparativo de las redes neuronales artificiales, dicho estudio comparativo consta de tres etapas que están profundizadas en este capítulo.

En el capítulo V se da a conocer los resultados de la investigación específicamente del estudio comparativo de las redes neuronales artificiales. En base a estos resultados se determina la red neuronal artificial que tiene un mayor nivel de eficiencia en la identificación de violencia escolar en las instituciones educativas.

En el capítulo VI se desarrolla la discusión de esta investigación, contrastándose con los objetivos planteados, la hipótesis y los antecedentes.

En esta investigación también se realiza un glosario donde se puede observar las definiciones de los términos más importantes para este trabajo.

Finalmente, se establece las conclusiones de la investigación, así como las recomendaciones para futuras investigaciones de esta misma línea investigativa.



CONTENIDO

DEDICATORIA i

AGRADECIMIENTOSii

RESUMEN iii

ABSTRACT.....iv

INTRODUCCIÓN v

CAPÍTULO I: ASPECTOS GENERALES 1

 1.1. Descripción de situación actual 1

 1.2. Formulación de problema2

 1.2.1. Problema General.....2

 1.2.2. Problemas Específicos2

 1.3. Objetivos de Investigación.....2

 1.3.1. Objetivo general2

 1.3.2. Objetivos específicos3

 1.4. Hipótesis3

 1.5. Variables e indicadores3

 1.5.1. Variable Dependiente.....3

 1.5.2. Indicadores de variable dependiente3

 1.6. Justificación4

 1.6.1. Conveniencia.....4

 1.6.2. Relevancia social4

 1.6.3. Implicancia practica5

 1.6.4. Valor teórico5

 1.6.5. Relevancia Aplicativa5

 1.7. Matriz de consistencia.....6

CAPITULO II: MARCO TEÓRICO 7

 2.1. Bases teóricas.....7

 2.1.1. Redes neuronales artificiales.....7

 2.1.2. Error cuadrático Medio (ECM).....36

 2.1.3. Validación Cruzada (Cross validation)37

 2.2. Antecedentes39

 2.2.1. Antecedentes internacionales.....39

 2.2.2. Antecedentes nacionales46



CAPITULO III: METODOLOGÍA47

3.1. Tipo de investigación47

3.2. Diseño de la investigación47

3.2.1. Etapa 1: El diseño de una base de datos para las redes neuronales artificiales .47

3.2.2. Etapa 2: El desarrollo y ajuste de redes neuronales artificiales aplicadas a la identificación de violencia escolar48

3.2.3. Etapa 3: Experimentación de las redes neuronales artificiales48

3.3. Población y muestra49

3.3.1. Población:49

3.3.2. Muestra:49

3.4. Técnicas de recolección de datos53

3.5. Técnicas de procesamiento de datos53

3.6. Aspectos Administrativos53

3.6.1. Recursos53

3.6.2. Presupuesto53

CAPITULO IV: ESTUDIO COMPARATIVO DE REDES NEURONALES ARTIFICIALES55

4.1. Etapa 1: Diseño de una base de datos para las Redes neuronales artificiales55

4.1.1. Identificación de entidades y atributos55

4.1.2. Modelo conceptual57

4.2. Etapa 2: Desarrollo y ajuste de redes neuronales artificiales aplicadas a la identificación de violencia escolar58

4.2.1. Diseño de clases generales para los algoritmos de redes neuronales artificiales 59

4.2.2. Desarrollo de los algoritmos de redes neuronales artificiales64

4.3. Etapa 3: Experimentación de las redes neuronales artificiales68

4.3.1. Experimentación de la Red neuronal artificial PMS71

4.3.2. Experimentación de la red neuronal artificial PMH76

4.3.3. Experimentación de la red neuronal artificial BRG81

4.3.4. Experimentación de la red neuronal artificial BRC86

4.3.5. Experimentación de la red neuronal artificial RRS91

4.3.6. Experimentación de la red neuronal artificial RRH96

CAPÍTULO V: RESULTADOS DE LA INVESTIGACIÓN 102

5.1. Resultado del Diseño de la Base de Datos para las Redes Neuronales Artificiales 102



5.2. Resultados del desarrollo y ajuste de redes neuronales artificiales aplicadas a la identificación de violencia escolar..... 103

5.2.1. Tipo 1: Perceptrón Multicapa con método de aprendizaje de Retropropagación (PM) 103

5.2.2. Tipo 2: Red Neuronal de Base Radial con método de aprendizaje Hibrido (BR) 109

5.2.3. Tipo 3: Red Neuronal Recurrente con método de aprendizaje Recurrente en Tiempo Real (RR)..... 115

5.3. Resultado de las mejores experimentaciones de las redes neuronales artificiales. 120

5.4. Resultado del estudio comparativo de redes neuronales artificiales..... 121

CAPÍTULO VI: DISCUSIÓN 122

GLOSARIO 125

CONCLUSIONES 130

RECOMENDACIONES..... 131

BIBLIOGRAFÍA 132

ANEXOS 134

Anexo A: Cuestionario utilizado para la recolección de datos 134

Anexo B: Prototipo Web..... 135



CONTENIDO DE TABLAS

Tabla 1: Matriz de consistencia	6
Tabla 2: Tabla de contingencia para atributos binarios	26
Tabla 3: Instituciones educativas de donde se obtendrá la información.....	49
Tabla 4: Muestra	50
Tabla 5: Regla para transformar la información recolectada.....	51
Tabla 6: Información de conjunto de datos Estudiante.....	51
Tabla 7: Atributos del conjunto de datos Estudiante	52
Tabla 8: Estadística del conjunto de datos	52
Tabla 9: Presupuesto de la Investigación.....	53
Tabla 10: Redes neuronales artificiales consideradas.....	58
Tabla 11: Procesos de Prueba y Error.....	68
Tabla 12: Tabla del proceso de experimentación de las redes neuronales artificiales.....	69
Tabla 13: Tabla comparativa de los escenarios realizados con PMS	76
Tabla 14: Tabla comparativa de los escenarios realizados con PMH.....	81
Tabla 15: Tabla comparativa de los escenarios realizados con BRG	86
Tabla 16: Tabla comparativa de los escenarios realizados con BRC	91
Tabla 17: Tabla comparativa de los escenarios realizados con RRS.....	96
Tabla 18: Tabla comparativa de los escenarios realizados con RRH	101
Tabla 19: Resultados de las experimentaciones de las Redes Neuronales artificiales en el estudio comparativo	120



CONTENIDO DE ILUSTRACIONES

Ilustración 1. Número de casos de maltrato reportados en el Siseve..... 1

Ilustración 2. Neurona Biológica 7

Ilustración 3. Neurona Artificial 8

Ilustración 4. Estructura básica de una red neuronal 9

Ilustración 5. Un perceptrón 12

Ilustración 6: Estructura de capas de una Red Neuronal 13

Ilustración 7. Red neuronal de dos capas, tres neuronas en la primera capa y dos en la segunda 16

Ilustración 8. Error cometido por un patrón y retropropagado hacia las neuronas anteriores 17

Ilustración 9: Errores calculados en cada neurona..... 18

Ilustración 10: Umbrales en cada neurona..... 19

Ilustración 11: Estructura de una red de base radial 21

Ilustración 12: Evolución de clases formadas por el método K-medias 23

Ilustración 13: Neuronas con conexión recurrente 30

Ilustración 14: Neuronas con conexión recurrente 30

Ilustración 15: Red recurrente en tiempo real..... 33

Ilustración 16: Validación cruzada 38

Ilustración 17: Tiempos de ejecución en segundos de las redes neuronales para los diferentes datasets del estudio de Comparación de Redes Neuronales aplicadas a la predicción de Series de Tiempo 39

Ilustración 18: Niveles de efectividad alcanzados por las redes neuronales para los diferentes datasets del estudio de Comparación de Redes Neuronales aplicadas a la predicción de Series de Tiempo 39

Ilustración 19: Resultados observados de la evaluación entre una MLP y una RBFNN en el estudio titulado “Comparison of artificial neural network models for hydrologic predictions at multiple gauging stations in an agricultural watershed” 41

Ilustración 20: Flujo observado y el predicho por los algoritmo en base al tiempo de entrenamiento y validación en el estudio titulado Comparison of artificial neural network models for hydrologic predictions at multiple gauging stations in an agricultural watershed 42

Ilustración 21: Varianza media relativa de los modelos recurrentes planteados 44



Ilustración 22: Resultados del estudio de Aplicación de Redes Neuronales Artificiales a la Modelización y Previsión de Caudales Medios Mensuales del Río Huancané46

Ilustración 23: Diseño de la investigación49

Ilustración 24: Modelo conceptual de la base de datos para los algoritmos de RNA.....57

Ilustración 25: Diagrama de las clases identificadas63

Ilustración 26: Resultados del escenario PMSI72

Ilustración 27: Resultados del escenario PMSII73

Ilustración 28: Resultados del escenario PMSIII.....74

Ilustración 29: Resultados del escenario PMSIV75

Ilustración 30: Resultados del escenario PMHI.....77

Ilustración 31: Resultados del escenario PMHII78

Ilustración 32: Resultados del escenario PMHIII79

Ilustración 33: Resultados del escenario PMHIV80

Ilustración 34: Resultados del escenario BRGI82

Ilustración 35: Resultados del escenario BRGII.....83

Ilustración 36: Resultados del escenario BRGIII.....84

Ilustración 37: Resultados del escenario BRGIV85

Ilustración 38: Resultados del escenario BRCI87

Ilustración 39: Resultados del escenario BRCII88

Ilustración 40: Resultados del escenario BRCIII.....89

Ilustración 41: Resultados del escenario BRCIV.....90

Ilustración 42: Resultados del escenario RRSI.....92

Ilustración 43: Resultados del escenario RRSII.....93

Ilustración 44: Resultados del escenario RRSIII94

Ilustración 45: Resultados del escenario RRSIV95

Ilustración 46: Resultados del escenario RRHI97

Ilustración 47: Resultados del escenario RRHII.....98

Ilustración 48: Resultados del escenario RRHIII.....99

Ilustración 49: Resultados del escenario RRHIV100

Ilustración 50: Diagrama de base de datos para Redes neuronales artificiales..... 102

Ilustración 51: Resultados de RSS vs Resultados esperados 121

CAPÍTULO I: ASPECTOS GENERALES

1.1. Descripción de situación actual

En el Perú el Ministerio de Educación lanzó una iniciativa y estableció una plataforma virtual SISEVE, encargada de recoger información de Directores, docentes o terceras personas que reporten casos de violencia escolar en estudiantes, sea en el círculo familiar o escolar. El reporte solo incluye el registro del caso y el denunciante debe presentar la documentación, fotografías o videos que permitan validar la denuncia. En esta plataforma no existe un mecanismo de identificación de violencia escolar en base a la información entregada por el denunciante.

Los casos registrados de violencia escolar en nuestro país en las instituciones educativas públicas y privadas desde el 15 de Septiembre de 2013 al 30 de Abril del 2017 es de solo 11 643 casos ya registrados en la plataforma SISEVE (ver ilustración 1).

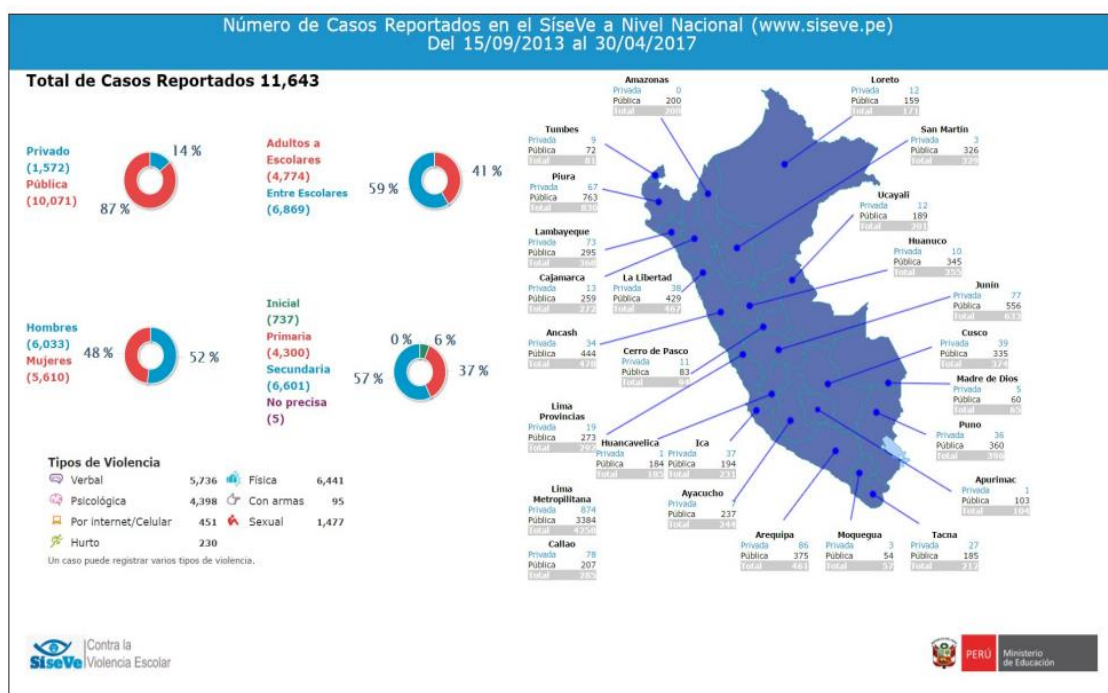


Ilustración 1. Número de casos de maltrato reportados en el Siseve

Fuente: Ministerio de Educación del Perú. (2017). *Estadísticas de violencia escolar en las instituciones educativas de Perú*. Obtenido de MINEDU: <http://www.siseve.pe/Seccion/Estadisticas>

Ante este problema de efectividad en la identificación de casos de violencia escolar, se plantea desarrollar algoritmos de redes neuronales artificiales que sean capaces de detectar en probabilidades la existencia de violencia escolar en los estudiantes en base a indicadores de convivencia reales, pero existe una limitante que a pesar de la

diversidad de estudios que hasta el momento se realizaron sobre redes neuronales artificiales en la predicción de aspectos sociales, aun ninguno se aplica al diagnóstico de violencia escolar, esto sumado a que existen muchos tipos de redes neuronales artificiales y cada uno posee características diferentes y por ende sus niveles de desempeño son distintos. Por estos motivos no se puede afirmar qué red neuronal artificial alcanzaría mejores resultados en la identificación de violencia escolar a priori, razón por la cual se considera realizar un estudio comparativo de Redes Neuronales Artificiales, de esta forma la problemática planteada se resume en la siguiente pregunta:

¿El estudio comparativo permitirá determinar la red neuronal artificial que tenga un mayor nivel de eficiencia en la identificación de violencia escolar en las instituciones educativas?

1.2. Formulación de problema

1.2.1. Problema General

¿El estudio comparativo permitirá determinar la red neuronal artificial que tenga un mayor nivel de eficiencia en la identificación de violencia escolar en las instituciones educativas?

1.2.2. Problemas Específicos

- ¿De qué manera se puede procesar la información recolectada de casos de violencia escolar para ser datos de entrada en las redes neuronales artificiales?
- ¿De qué manera obtener redes neuronales artificiales aplicadas a la identificación de violencia escolar?
- ¿De qué manera obtener el mayor nivel de eficiencia de cada red neuronal artificial en la identificación de violencia escolar?

1.3. Objetivos de Investigación

1.3.1. Objetivo general

Determinar a través de un estudio comparativo la red neuronal artificial que tenga un mayor nivel de eficiencia en la identificación de violencia escolar en las instituciones educativas.



1.3.2. Objetivos específicos

- Diseñar una base de datos de la información recolectada para las redes neuronales artificiales.
- Desarrollar y ajustar redes neuronales artificiales aplicadas a la identificación de violencia escolar.
- Experimentar cada red neuronal artificial aplicada a la identificación de violencia escolar.

1.4. Hipótesis

El estudio comparativo permite determinar la red neuronal artificial que tiene un mayor nivel de eficiencia en la identificación de violencia escolar en las instituciones educativas.

1.5. Variables e indicadores

1.5.1. Variable Dependiente

- Nivel de eficiencia en la identificación de violencia escolar

Se refiere al porcentaje alcanzando por una red neuronal artificial en la identificación de violencia escolar, el criterio de evaluación de esta variable dependiente es el error cuadrático medio (ECM). Donde:

Nivel de eficiencia en la identificación de violencia escolar = $(1 - \text{ECM}) * 100$

Limitante:

- El error cuadrático medio debe ser menor a 1

$$\text{ECM} < 1$$

1.5.2. Indicadores de variable dependiente

- Tipo de red neuronal
- Método de aprendizaje
- Función de activación
- Numero de neuronas



- Numero de ciclos¹
- Instantes de tiempo²
- Razón de aprendizaje

1.6. Justificación

1.6.1. Conveniencia

Si bien es cierto que se cuenta con la plataforma SISEVE para el registro de casos de violencia escolar en la Instituciones Educativas, el número de casos registrados hasta el último estudio estadístico es de tan solo 11 643 casos registrados en cinco años, como muestra la *ilustración 1* y además que los casos denunciados deben ser evidenciados con fotografías, videos o documentos que demuestren la veracidad de la denuncia convirtiéndose en un proceso burocrático y poco efectivo porque demanda tiempo y recursos de los denunciantes y como consecuencia no todos terminan de registrar la denuncia de forma correcta.

Por lo tanto, realizar un estudio comparativo con la finalidad de lograr obtener un modelo computacional óptimo para la identificación en probabilidades de violencia escolar, que automatice el proceso de registro de denuncia, utilizando tan solo los indicadores de convivencia que son datos de entrada a dicho modelo computacional podría formar parte de una herramienta práctica en la labor de las organizaciones encargadas de brindar asesoría y ayuda a las personas que padecen este problema social.

1.6.2. Relevancia social

Con el desarrollo de la presente investigación se pretende ofrecer una herramienta para las entidades encargadas de hacer frente a la violencia escolar en estudiantes de instituciones educativas, y de este modo colaborar en la identificación de nuevos casos de este tipo.

¹ Solo para tipos de redes Perceptrón Multicapa con método de aprendizaje de Retropropagación y para Redes Neuronales de Base Radial con aprendizaje Híbrido, se denomina número de ciclos al número de iteraciones que se repiten en el proceso de aprendizaje, donde el proceso de todos los patrones de entrenamiento completan una iteración.

² Sola para tipos de Redes Neuronales Recurrentes con aprendizaje de retropropagación en tiempo real, cada instante de tiempo en el aprendizaje es el proceso de un patrón de entrenamiento.

1.6.3. Implicancia practica

Se busca brindar a las entidades públicas o privadas encargadas de atender los casos de violencia escolar una herramienta ágil que identifique en probabilidades los potenciales casos de existencia de violencia en los estudiantes de Instituciones Educativas de nuestro país, con el propósito de sustentar las intervenciones y la toma de decisiones para de esta forma erradicar o minimizar este problema social.

1.6.4. Valor teórico

Las redes neuronales artificiales son modelos computacionales que pueden simular al menos la lógica del funcionamiento de las redes neuronales biológicas, en la presente investigación se consideran seis redes neuronales artificiales, donde cada una es una combinación de un tipo de red, un método de aprendizaje y una función de activación. Estas redes serán sometidas a un estudio comparativo para encontrar la combinación que alcance un mayor nivel de eficiencia en la identificación de violencia escolar en estudiantes de instituciones educativas.

En esta misma perspectiva investigativa se pretende también nutrir las bases teóricas científicas sobre el estudio y la aplicación de las redes neuronales artificiales en los problemas de las diferentes áreas de estudio donde se requiera un mecanismo de detección.

1.6.5. Relevancia Aplicativa

La red neuronal artificial que obtenga un mayor nivel de eficiencia en el estudio comparativo, podría ser incluida como motor principal en la implementación de un software que diagnostique en probabilidades la existencia de violencia escolar en los estudiantes de instituciones educativas.

1.7. Matriz de consistencia

Tabla 1: Matriz de consistencia

Problema	Objetivo	Hipótesis	Variables
<p>Problema General ¿El estudio comparativo permitirá determinar la red neuronal artificial que tenga un mayor nivel de eficiencia en la identificación de violencia escolar en las instituciones educativas?</p> <p>Problemas Específicos</p> <ul style="list-style-type: none"> • ¿De qué manera se puede procesar la información recolectada de casos de violencia escolar para ser datos de entrada en las redes neuronales artificiales? • ¿De qué manera obtener redes neuronales artificiales aplicadas a la identificación de violencia escolar? • ¿De qué manera obtener el mayor nivel de eficiencia de cada red neuronal artificial en la identificación de violencia escolar? 	<p>Objetivo General Determinar a través de un estudio comparativo la red neuronal artificial que tenga un mayor nivel de eficiencia en la identificación de violencia escolar en las instituciones educativas.</p> <p>Objetivos Específicos</p> <ul style="list-style-type: none"> • Diseñar una base de datos de la información recolectada para las redes neuronales artificiales. • Desarrollar y ajustar redes neuronales artificiales aplicadas a la identificación de violencia escolar. • Experimentar cada red neuronal artificial aplicada a la identificación de violencia escolar. 	<p>Hipótesis General El estudio comparativo permite determinar la red neuronal artificial que tiene un mayor nivel de eficiencia en la identificación de violencia escolar en las instituciones educativas.</p>	<p>Variable dependiente Nivel de eficiencia en la identificación de violencia escolar</p> <p>Criterio</p> <ul style="list-style-type: none"> • Error cuadrático medio (ECM) <p>Indicadores de la variable dependiente</p> <ul style="list-style-type: none"> • Tipo de red neuronal • Método de aprendizaje • Función de activación • Numero de neuronas • Razón de aprendizaje • Numero de ciclos. • Instantes de tiempo.

Nota. Matriz de consistencia de la presente tesis. Fuente: Propia

CAPITULO II: MARCO TEÓRICO

2.1. Bases teóricas

2.1.1. Redes neuronales artificiales

2.1.1.1. Fundamentos biológicos

Las redes neuronales artificiales son modelos computacionales que emulan el funcionamiento del proceso biológico de información que se da en nuestro cerebro, donde este contiene una enorme cantidad de neuronas biológicas interconectadas.

En la *ilustración 2* se puede observar una neurona biológica que es una célula especializada en procesar información, está compuesta por un cuerpo celular que es considerado como “Soma” y dos tipos de ramificaciones el “Axón” y las “dendritas”. Las neuronas reciben señales o impulsos de otras neuronas de la red a través de las dendritas, así como también transmiten señales a través del axón. (Cruz, 2010)

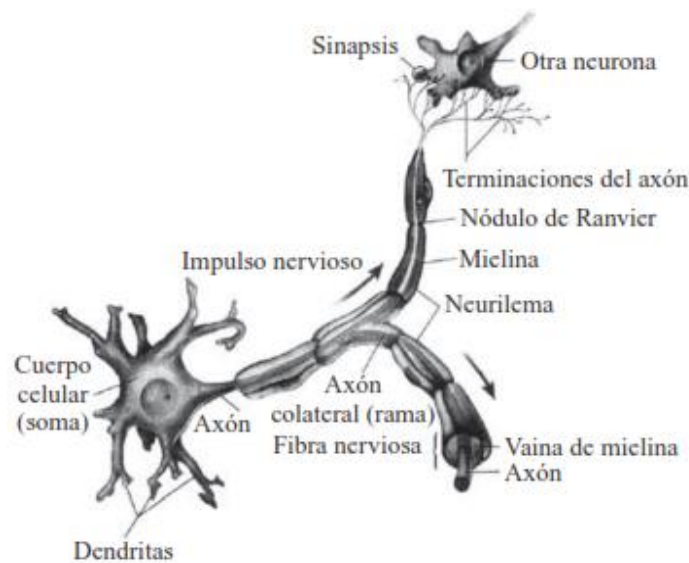


Ilustración 2. Neurona Biológica

Fuente: Cruz, P. P. (2010). *Inteligencia Artificial con aplicaciones a la ingeniería*. Mexico: Alfaomega.

Otro elemento importante es la sinapsis que representa la unión entre el axón de una neurona con la dendrita de otra, en cada sinapsis se da una transmisión electroquímica hasta alcanzar el soma de esta última neurona, una vez ahí se da una sumatoria con todas las demás sinapsis que llegaron con otra tipo de señal, posteriormente el cuerpo celular realiza un tipo de función de activación que genera una respuesta que es propagada por medio de su Axón hacia otra neurona que este interconectada a ella.

2.1.1.2. Neuronas artificiales

Las neuronas artificiales son elementos que poseen un nivel de activación, este nivel es variable y depende del cálculo de la función de activación o también llamada función de transición de estado y las señales recibidas, donde además cada señal tiene asociada un peso sináptico, este valor cambia al transcurrir el tiempo de vida de la neurona.

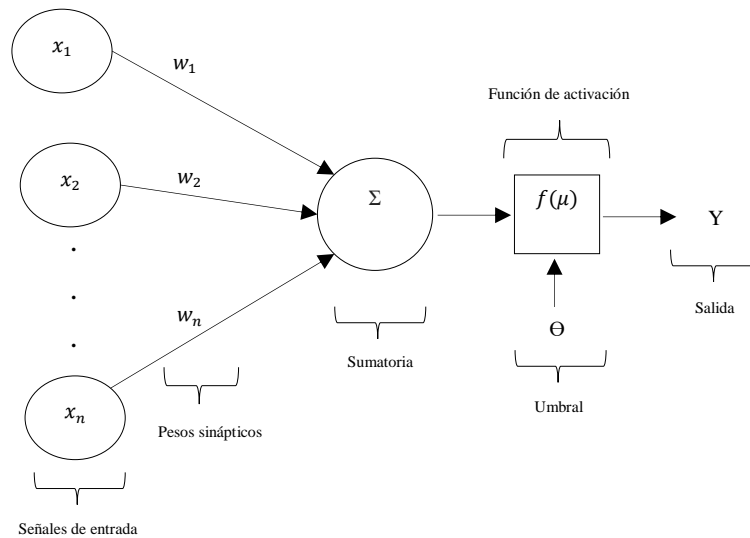


Ilustración 3. Neurona Artificial

Fuente: José Hernández, José Ramírez, Cesar Ferri. (2004).
Introducción a la minería de datos. Madrid: PEARSON EDUCACIÓN.

En la *ilustración 3* se observa una neurona artificial, donde el valor de la salida de la neurona viene dado por:

$$y = f \left(\sum_k x_i w_i + \theta \right) \quad (1)$$

Donde:

y : Salida de la neurona

x_i : Señales de entrada

w_i : Pesos de las conexiones

f : Funcion de activación no lineal simple

2.1.1.3. Estructura básica de una red neuronal artificial

El valor de la salida de una neurona artificial es enviado a las otras neuronas que están interconectadas con ella, muchas neuronas interconectadas forman lo que se denomina una red neuronal artificial.

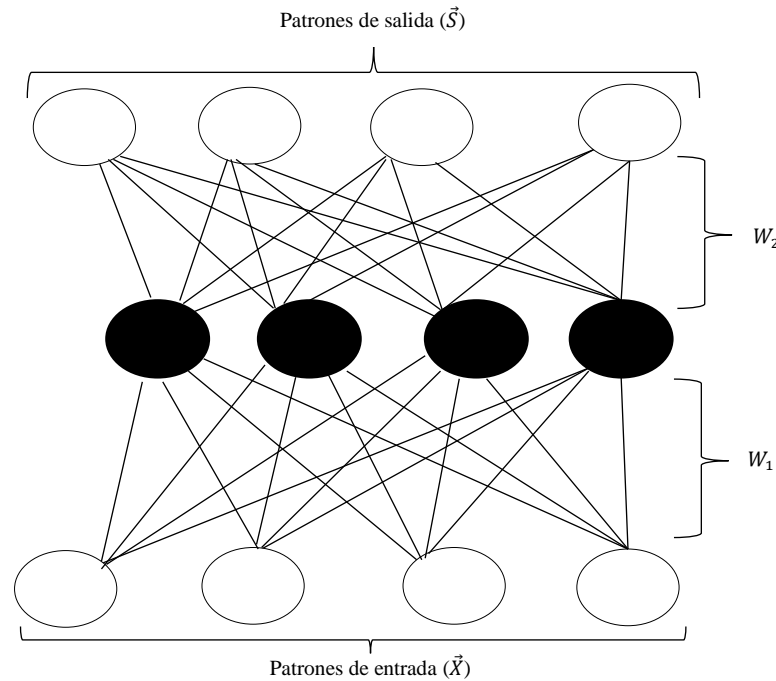


Ilustración 4. Estructura básica de una red neuronal

Fuente: Adaptado de Pedro Isasi Viñuela, I. G. (2004). *Redes Neuronales Artificiales un enfoque practico*. Madrid : Pearson Educación.

“Así pues, una red neuronal artificial básica podría definirse como un grafo cuyos nodos están constituidos por unidades de proceso idénticas, y que propagan información a través de los arcos. En este grafo se distinguen tres tipos de nodos: los de entrada, los de salida y los intermedios”. (Pedro Isasi Viñuela, 2004)

Los llamados nodos de entrada son las neuronas encargadas de recibir los patrones que viene a ser vectores numéricos que posteriormente son procesados y propagados hacia los nodos de los siguientes niveles.

El esquema de funcionamiento de la *ilustración 4* puede describirse mediante la ecuación:

$$\vec{S} = f(f(\vec{X}.W_1).W_2) \quad (2)$$

Donde:

\vec{S} : Es el vector de salida

W_1 Y W_2 : Son las matrices con los valores de los pesos de las conexiones de la primera y segunda capa respectivamente.

\vec{X} : Es el vector de entrada a la red

f : Es la función de activación

2.1.1.4. *Función de activación*

Las neuronas tienen la característica de poseer una función de activación (F) que es procesada basándose en la entrada efectiva de la neurona, de lo ya explicado en el apartado sabemos que la entrada efectiva (μ) se trata de la sumatoria del producto del peso de cada sinapsis con el valor que este ingresa a la neurona, el resultado de la función es la salida simple de la neurona.

$$OUT = f(\mu) \quad (3)$$

a) *Función Sigmoidal*

La función sigmoide se utiliza en los casos donde el valor de salida es cercano a uno de los valores asintóticos. Esto produce que, en la mayoría de los casos, el valor de salida está comprendido en la zona alta o baja de la Sigmoide. Lo importante de esta función es que su derivada siempre es positiva o cercana a cero para los valores grandes positivos o negativos; además toma su valor máximo cuando $\mu = 0$.

$$F(\mu) = \frac{1}{1 + e^{-\mu}} \quad (4)$$

b) *Función Tangente Hiperbólica*

La función Tangente Hiperbólica se utiliza en casos donde los valores de la señal a clasificar sean positivos y negativos, es una de las funciones más utilizadas en aprendizaje supervisado.

$$F(\mu) = \frac{1 - e^{-\mu}}{1 + e^{-\mu}} \quad (5)$$

c) *Función Gaussiana*

Los mapeos ocultos algunas veces pueden realizarse con un solo nivel de neuronas mediante el uso de activación tipo Gauss.

$$F(\mu) = e^{-\mu/2} \quad (6)$$

d) *Función inversa cuadrática*

$$F(\mu) = \frac{1}{1 + \mu^2} \quad (7)$$

2.1.1.5. *Aprendizaje*

El aprendizaje es una de las partes más relevantes de las redes neuronales artificiales, donde se lleva a la red a un proceso de adaptación debido a las variaciones de los pesos de sus sinapsis. La red debe aprender en base a un conjunto de datos de entrada, pero no solo se encarga de aprender, clasificar e identificar dicho conjunto, sino que también adquiere la capacidad de generalización que permite si en el caso se presente un nuevo conjunto de datos diferentes pero con las mismas características de los que ya aprendió pueda también reconocerlos de manera eficiente.

a) *Aprendizaje supervisado*

Dentro del proceso de aprendizaje supervisado todas las sinapsis que son las conexiones que unen a cada una de las neuronas de una red neuronal determinada tienen un peso establecido al principio, muchas veces de forma aleatoria, cuando ya se procesa toda la red neuronal esta nos devuelve un resultado, entonces este resultado es comparado con el resultado que debería haber producido, esta diferencia influye en la modificación de los pesos de las conexiones. Este proceso se repite hasta conseguir un punto estable, es decir: hasta que se consiga una mínima diferencia respecto a los resultados que se esperan o que las conexiones lleguen a estabilizarse.

b) Aprendizaje no supervisado

“El aprendizaje no supervisado consiste en que la red descubra por si sola las características, regularidades, correlaciones o categorías en los datos de entrada, y se obtenga de forma codificada a la salida. Por lo tanto, se puede decir que estas unidades y conexiones muestran cierto grado de auto-organización”. [(Pedro Isasi Viñuela, 2004)]

En base a esto se puede también afirmar que en este tipo de aprendizaje no es necesario que la red neuronal cuente con un supervisor externo que realice esta tarea.

2.1.1.6. Perceptrón multicapa (MLP)

a) Perceptrón simple

Es uno de los primeros modelos computacionales investigado por primera vez por Rosenblatt en 1962. (Rosenblatt, 1962)

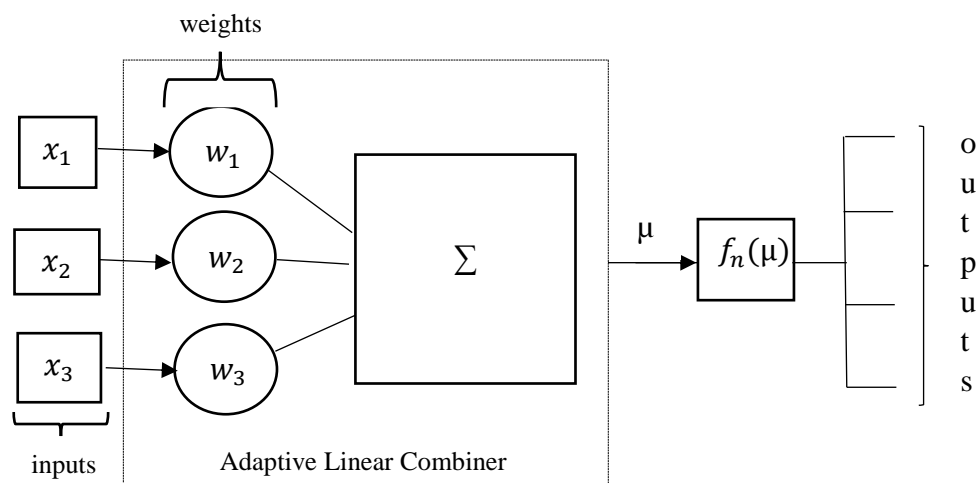


Ilustración 5. Un perceptrón

Fuente: Adaptado de Graupe, D. (2007). *Principles of Artificial Neural Networks*. Chicago - Estados Unidos: World Scientific.

Como se aprecia en la *ilustración 5* un perceptrón es un modelo neuronal de varias entradas (x), una célula y salidas ponderadas, en donde sus entradas ponderadas se propagan por medio de las conexiones cuyos pesos (w) son ajustables, la entrada efectiva (μ) es la sumatoria del producto de todas las entradas con los pesos, luego esta entrada es parámetro de una función lineal, donde el resultado de dicha función es la salida final del perceptrón.

b) Perceptrón multicapa

El perceptrón simple explicado anteriormente fue desarrollado para solucionar problemas únicamente lineales, frente a esta limitación, Minsky y Papert desarrollaron en 1969 (Minsky & Papert, 1969). La combinación de varios perceptrones simples con el fin de conseguir un modelo computacional basado en el perceptrón simple, pero que sea capaz de solucionar problemas no lineales, es así como presentaron un modelo denominado perceptrón multicapa.

“... Hoy en día es posible mostrar que muchos conjuntos de datos que no son linealmente separables pueden ser modelados mediante el empleo del perceptrón multicapa, es decir una red neuronal en forma de cascada, que tiene una o más capas ocultas...”. (José Hernández, José Ramírez, Cesar Ferri, 2004)

c) Arquitectura de un perceptrón multicapa

El perceptrón multicapa es una red que cuenta con una capa de entrada y una capa de salida ambas con una o muchas neuronas, pero además este tipo de red puede tener varias capas ocultas, a su vez cada capa oculta puede contener n neuronas.

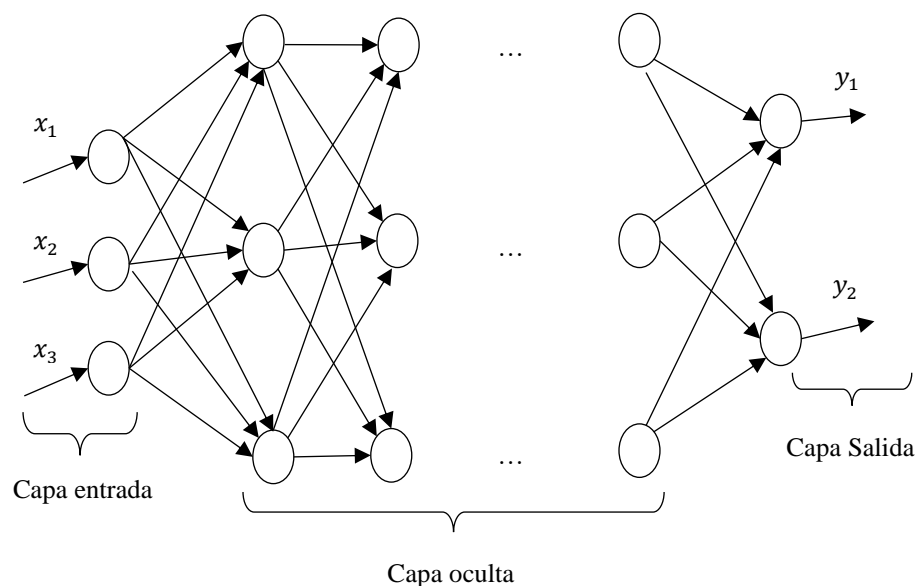


Ilustración 6: Arquitectura del Perceptrón Multicapa

Fuente: Pedro Isasi Viñuela, I. G. (2004). *Redes Neuronales Artificiales un enfoque practico*. Madrid : Pearson Educación.

Como se observa en la *ilustración 6* las conexiones siempre están dirigidas hacia adelante, por esto reciben el nombre también de redes “feedforward”. Todas las conexiones entre neuronas llevan asociado un peso de conexión que se comprende como un número real. (Pedro Isasi Viñuela, 2004)



También es importante señalar que no existe una técnica cien por ciento certera que permita establecer el número de capas ocultas o de neuronas, es responsabilidad del investigador establecerlos, esto puede depender de la naturaleza del problema que se pretende solucionar.

A pesar de eso, varias fuentes bibliográficas coinciden en que la mejor forma de establecerlos es por medio de la experimentación, probando el aprendizaje de la red varias veces con diferentes números de capas ocultas, así como de neuronas, hasta lograr la combinación que de un mejor resultado.

“...Por lo tanto, si pensamos en el aprendizaje, hay que actualizar dos conjuntos de pesos: aquellos entre la capa oculta o intermedia y la de la salida, y aquellos entre la capa de entrada y la intermedia...”. (José Hernández, José Ramírez, Cesar Ferri, 2004)

Sin embargo, hace años atrás no había ninguna herramienta poderosa capaz de establecer los pesos de una red multicapa por medio de un aprendizaje supervisado, que evalué y busque los pesos adecuados para que el resultado final de la red sea el más cercano al resultado que se espera. Una solución a este problema es el algoritmo de propagación hacia atrás, que será detallado a continuación.

d) Retropropagación (backpropagation)

La propagación hacia atrás (BP) es un algoritmo que fue propuesto en 1986 por Rumelhart, Hinton y Williams para establecer pesos y por lo tanto para el entrenamiento de perceptrones multicapa.

El algoritmo de propagación hacia atrás comienza, al contar con el cálculo de la capa de salida, que es el único en el que los resultados deseados se pueden comparar, ya que esto en la capa oculta no se puede realizar. Entonces se tiene que hallar el error que es la diferencia del resultado real y el resultado esperado o deseado, de ahí ir hacia atrás multiplicando los pesos de las conexiones de las neuronas de la capa oculta para hallar una tasa de error que permitirá hallar un nuevo peso a cada conexión, que se espera mejore la capacidad de la red en general. (Pedro Isasi Viñuela, 2004)

El objetivo de este método de aprendizaje es acercar el resultado de la propagación hacia delante de la red con el resultado deseado, para ello lo primero que se debe realizar es calcular el error cometido y propagarlo hacia atrás.

$$e(p) = \frac{1}{2} \sum_{i=1}^{ns} (d_i - y_i)^2 \quad (8)$$

Donde $e(p)$ es el error cometido por el patrón de entrada “p”, donde d_i es el vector de las salidas deseadas, y_i es el vector de salidas de la red. Además ns representa el número de salidas de la red, es importante mencionar que cada salida de la red debe estar asociada a un resultado deseado.

En consecuencia el error total de todos los patrones evaluados es igual a:

$$E = \frac{1}{P} \sum_{p=1}^P e(p) \quad (9)$$

Aunque estrictamente hablando, el aprendizaje de la red debe realizarse para minimizar el error total, el procedimiento más utilizado está basado en métodos de gradiente estocástico, los cuales consisten en una sucesiva minimización del error para cada patrón, en lugar de minimizar el error total E . (Pedro Isasi Viñuela, 2004)

El deceso de gradiente se aplica a la actualización de los pesos en base a la tasa de error calculado, por tanto el nuevo peso w para el patrón p es igual a:

$$w(p) = w(p - 1) - \alpha \frac{\partial e(p)}{\partial w} \quad (10)$$

La actualización del peso se da en base al error calculado, pero como las neuronas de la capa oculta no tienen un valor de error directo, y estas no cuenta con salidas deseadas es necesario propagar el error de las neuronas de la capa de salida hacia atrás, por esta razón este algoritmo se denomina “retropropagación de error”.

A continuación se explicara con un ejemplo el cálculo del error, la retropropagación de dicho error y la actualización de los pesos. (Cruz, 2010)

Si suponemos tener una red de dos entradas y una salida, y en la capa oculta dos capas la primera de tres neuronas y la segunda de dos neuronas como se mostrará en la *ilustración 7*.

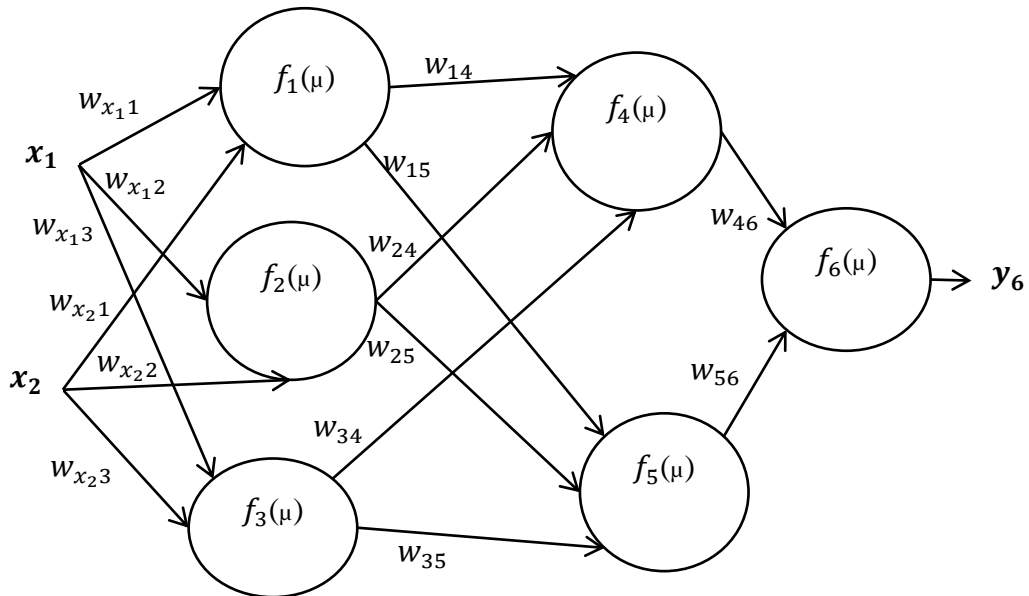


Ilustración 7. Red neuronal de dos capas, tres neuronas en la primera capa y dos en la segunda

Fuente: Cruz, P. P. (2010). *Inteligencia Artificial con aplicaciones a la ingeniería*. Mexico: Alfaomega.

En la anterior ilustración cada una de las neuronas tiene una función de activación asociada, como también cada conexión entre las neuronas tiene un peso (w).

Para empezar el proceso de aprendizaje primero se propaga las salidas de las neuronas hacia adelante, en este sentido la salida de la primera neurona se determina de la siguiente manera:

$$y_1 = f_1(w_{(x_11)} x_1 + w_{(x_21)} x_2) \tag{11}$$

La salida calculada será propagada hacia adelante, al igual que las neuronas de esa primera capa, por lo tanto siguiendo esta lógica la salida de la quinta neurona es igual a:

$$y_5 = f_5(w_{15} Y_1 + w_{25} Y_2 + w_{35} Y_3) \tag{12}$$

Finalmente la propagación hacia delante termina cuando se obtiene la salida de la última neurona, con ello obtenemos la salida de la red.

$$y_6 = f_6(w_{46} Y_4 + w_{56} Y_5) \tag{13}$$

Con el valor de la salida de la red neuronal y con la salida deseada asociada a esta, se puede hallar el error cometido para este patrón.

$$\delta_6 = d - y_6 * f'_6(w_{46} Y_4 + w_{56} Y_5) \tag{14}$$

Donde y_6 es el resultado de la red neuronal, d es el resultado deseado y f' es la derivada de la función de activación. Una vez calculado el error cometido para este patrón, se realiza la propagación hacia las neuronas anteriores de la red, de la siguiente forma:

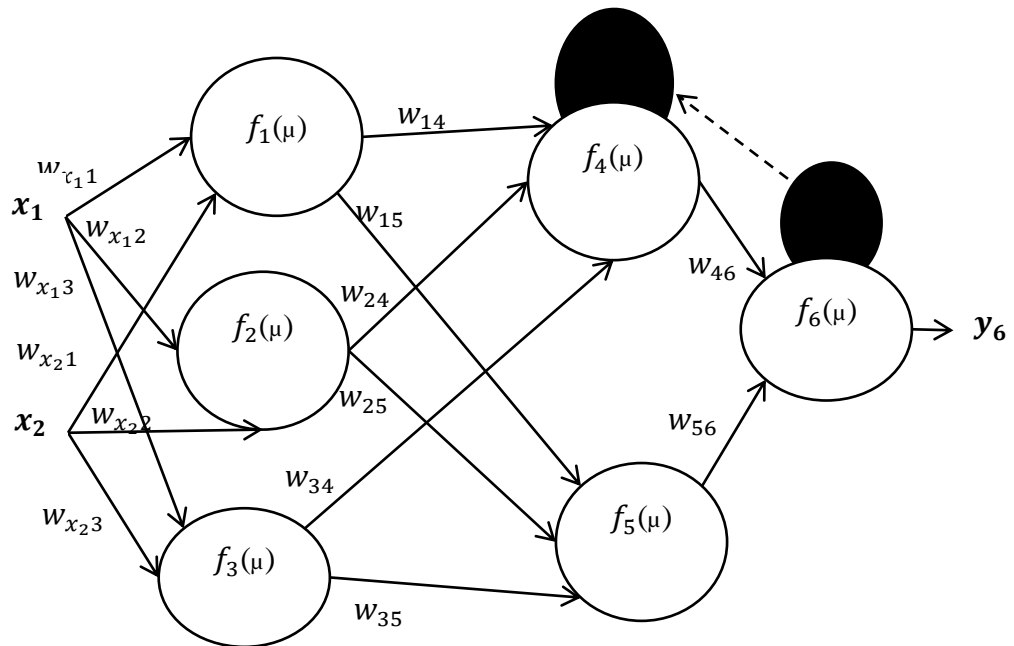


Ilustración 8. Error cometido por un patrón y retropropagado hacia las neuronas anteriores

Fuente: Cruz, P. P. (2010). *Inteligencia Artificial con aplicaciones a la ingeniería*. Mexico: Alfaomega.

Como se observa en la *ilustración 8* el error de las neuronas anteriores depende del error de las posteriores, entonces el error en la cuarta neurona está dado por:

$$\delta_4 = w_{46} \delta_6 \tag{15}$$

De la misma forma el error en la primera neurona estaría dado por:

$$\delta_1 = w_{14} \delta_4 + w_{15} \delta_5 \tag{16}$$

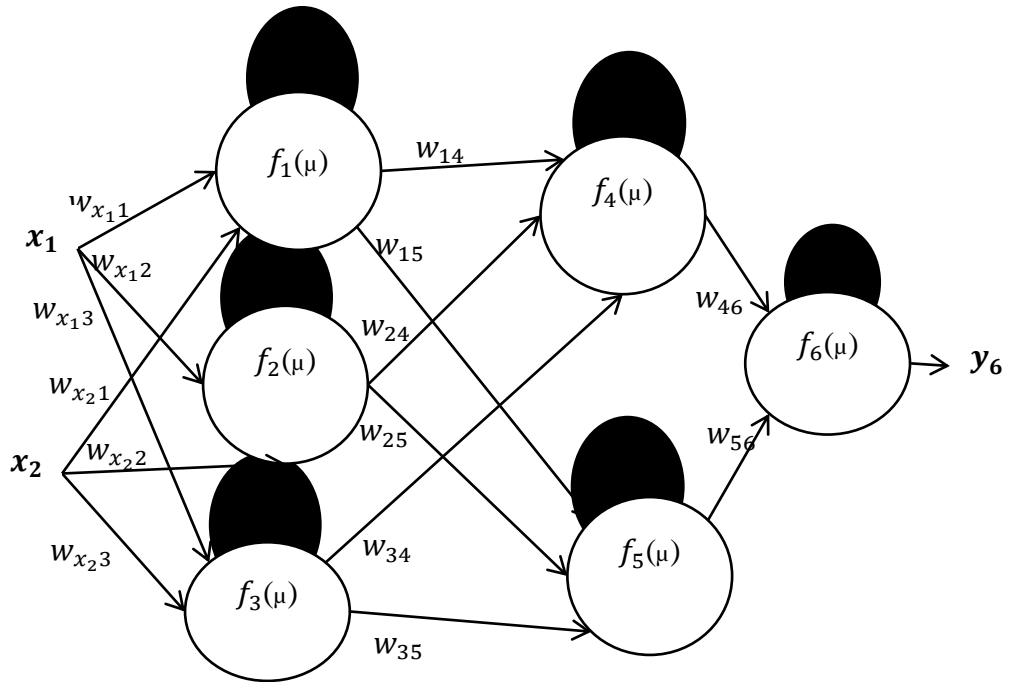


Ilustración 9: Errores calculados en cada neurona

Fuente: Cruz, P. P. (2010). *Inteligencia Artificial con aplicaciones a la ingeniería*. Mexico: Alfaomega.

En la *ilustración 9* se puede observar que cada neurona tendrá un error calculado, en base a estos errores se pueden modificar los pesos por medio del valor actual del peso y la de un coeficiente que indica la velocidad de aprendizaje (α), esta operación permitirá tener un nuevo peso por medio de un tipo de amortiguamiento respecto al anterior. Si por ejemplo se quiere actualizar el peso w_{x_11} se debe realizar la siguiente operación:

$$w_{x_11} = w_{x_11}(p - 1) + \alpha * \delta_1 * x_1 \tag{17}$$

Siguiendo la misma lógica para actualizar el peso de la conexión de la cuarta neurona a la sexta, sería:

$$w_{46} = w_{46}(p - 1) + \alpha * \delta_6 * y_4 \tag{18}$$

Como se explicó anteriormente las neuronas artificiales también cuentan con umbrales o bias (θ), estas también tienen que ser actualizadas.

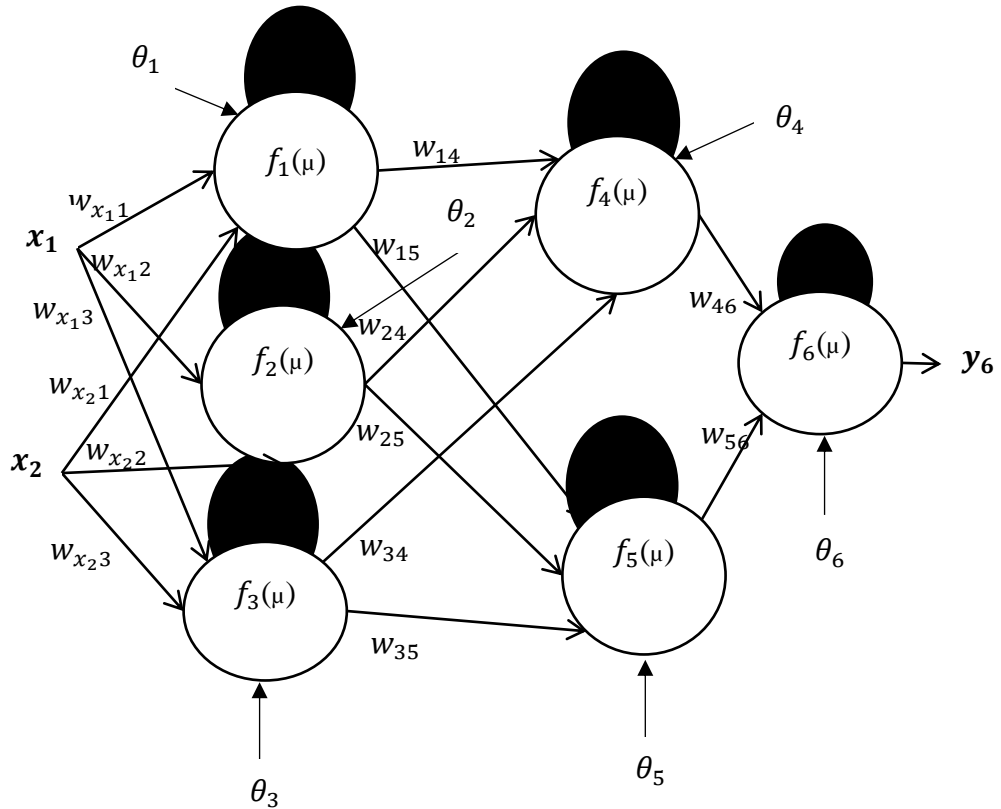


Ilustración 10: Umbrales en cada neurona

Fuente: Cruz, P. P. (2010). *Inteligencia Artificial con aplicaciones a la ingeniería*. Mexico: Alfaomega.

En la *ilustración 10* se puede ver los umbrales asociados a cada neurona y de la misma forma que en los pesos utilizamos el error calculado en cada neurona para la actualización de sus umbrales. Por ejemplo el cálculo del nuevo valor del umbral de la cuarta neurona estaría dado por:

$$\theta_4 = \theta_4(p - 1) + \alpha \delta_4 \tag{19}$$

En la misma lógica se actualiza los valores de los umbrales de las demás neuronas. Se realiza la actualización de los pesos y umbrales para cada patrón de entrada, al realizar dicha actualización con todos los patrones de entrada se completa un ciclo

de aprendizaje, en cada ciclo de aprendizaje se evalúa el error de aprendizaje utilizando la *ecuación 9*. El algoritmo acaba generalmente cuando se calcula el mínimo error de aprendizaje.

2.1.1.7. *Redes Neuronales de base radial*

Las redes de base radial al igual que el perceptrón multicapa también son redes que poseen varias capas, pero se caracteriza principalmente por tener una sola capa oculta, además las funciones de activación de las neuronas de la capa oculta deben utilizar funciones de base radial, como la función gaussiana o la función invertida cuadrática. Uno de los objetivos principales por los que se investigó este tipo de redes neuronales artificiales era construir una red de neuronas que requiriese un menor tiempo de aprendizaje, y de este modo, disponer de una red de neuronas que pudieran ser apropiadas para aplicaciones de tiempo real. (Pedro Isasi Viñuela, 2004)

a) Arquitectura de redes de base radial

La arquitectura de una red de tipo base radial está formada por tres capas diferentes:

- **Capa de entrada:** Esta capa se encarga de recibir las señales del exterior, y las propagan hacia la siguiente capa que es la capa oculta o intermedia.
- **Capa Oculta:** Las neuronas que pertenecen a esta capa reciben las señales de las neuronas de entrada por medio de conexiones, como ocurre en el perceptrón multicapa, pero con la diferencia que no existen pesos asociados, además estas neuronas se activan por medio de una función de base radial que define una relación no lineal entre la capa de entrada y la de salida.
- **Capa de salida:** En esta capa es la salida de la red, pero además realiza una combinación lineal de las neuronas de la capa oculta.

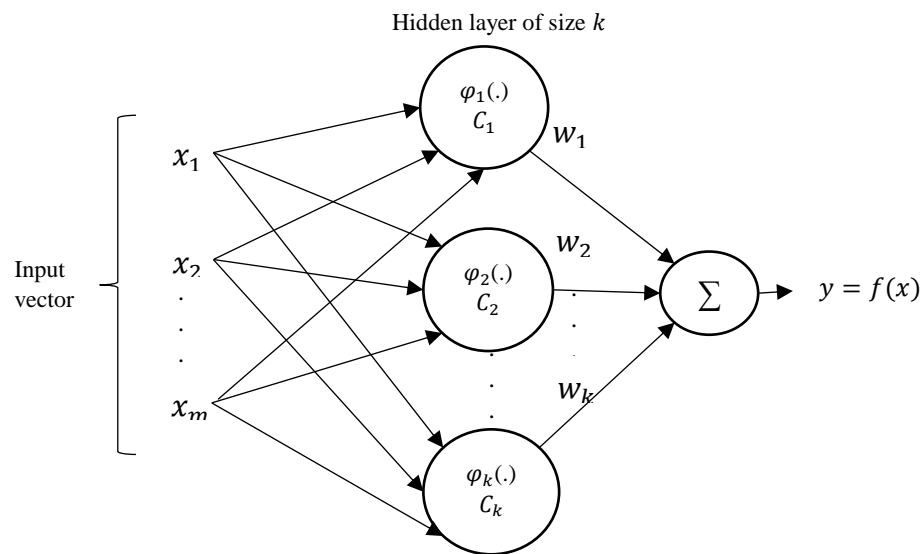


Ilustración 11: Estructura de una red de base radial

Fuente: Haykin, S. (2009). *Neural Networks and Learning Machines*. New Jersey: Pearson Education.

En la *ilustración 11* se observa una arquitectura típica de una red de base radial donde: X es el vector de entrada y k es el número de neuronas en la capa oculta, como anteriormente se mencionó las conexiones de la capa de entrada hacia la capa oculta no tienen un valor de peso asociado w , como si ocurre en las conexiones de la capa oculta hacia la de salida. La función de salida está definida por:

$$f(x) = \sum_{j=1}^k w_j \varphi_j(x, X_j) \tag{20}$$

Las funciones φ , también conocidas como funciones de base radial, determinan las activaciones de las neuronas ocultas de la red en función al vector de entrada X y viene dado por:

$$\varphi_k = \left(\frac{\|X(n) - C_i\|}{d_i} \right) \text{ para } i = 1, 2, 3, \dots, k \tag{21}$$

Donde C_i son vectores que representan los centros de la función de base radial; d_i son los números reales que representan la desviación de la función; y $\|X(n) - C_i\|$ es la distancia euclidiana del vector de entrada al centro, definida como:

$$||X(n) - C_i|| = \left(\sum_{l=1}^m (x_l - C_{il})^2 \right)^{1/2} \quad (22)$$

b) *Método de aprendizaje híbrido*

El método híbrido realiza el aprendizaje de las redes neuronales de base radial en dos fases distintas, una de ellas es la fase no supervisada donde se determinan los centros y amplitudes de las neuronas de la capa oculta, y la otra fase es la supervisada donde se determinan los pesos y umbrales de la capa de salida.

Fase no supervisada

En la Fase no supervisada se debe tomar en consideración que el número de clases es igual al número de neuronas, estas clases representan zonas diferentes en el espacio de los patrones de entrada, es por esto que es necesario establecer los valores de los centros y las desviaciones de las neuronas.

Un método muy utilizado en esta fase es el algoritmo de k-medias (k-means), que será explicado a continuación:

- K-medias (k-means)

K-medias es un algoritmo de clasificación, que divide el espacio de patrones de entrada en clases o regiones. El representante de cada una de las clases es C_i que viene a ser el centro de la neurona oculta i .

Los centros son establecidos con el fin de minimizar las distancias euclidianas entre los patrones de entrada y el centro más cercano. (Pedro Isasi Viñuela, 2004)

$$J = \sum_{i=1}^k \sum_{n=1}^N \text{Min} ||X(n) - C_i|| \quad (23)$$

Donde:

N : Numero de patrones

$X(n)$: Patrón de entrada n

Min : Función de pertenencia, que vale 1 si el centro C_i es el más cercano al patrón $X(n)$, caso contrario vale 0.

Los pasos del algoritmo de K-medias son los siguientes: (Pedro Isasi Viñuela, 2004)

Paso 1: Se inicializan los centros de las k clases

Paso 2: Se asignan n patrones a cada clase i .

El patrón $X(n)$ pertenece a la clase i si $\|X(n) - C_i\| < \|X(n) - C_s\|$ para $s \neq i$ con $s = 1, 2, 3, \dots, k$.

Cada clase tendrá a asociado los patrones de entrada más cercanos a su centro.

Paso 3: Se calculan la nueva posición de los centros de las clases como la media de todos los patrones que pertenecen a su clase.

Paso 4: Se repiten los pasos 2 y 3 hasta que las nuevas posiciones de los centros no se modifiquen respecto a su posición anterior.

$\|C_i \text{ Nuevo} - C_i \text{ Anterior}\| < \varepsilon$, para $i = 1, 2, 3, \dots, k$

Siendo ε un número positivo cercano a 0.

Como tras una nueva iteración los patrones de entrada son reasignados a una clase, es necesario volver a calcular los centros de cada clase en base a sus dichos patrones asignados. A medida que los centros de las clases se van situando a los patrones asignados, se minimizara el desplazamiento.

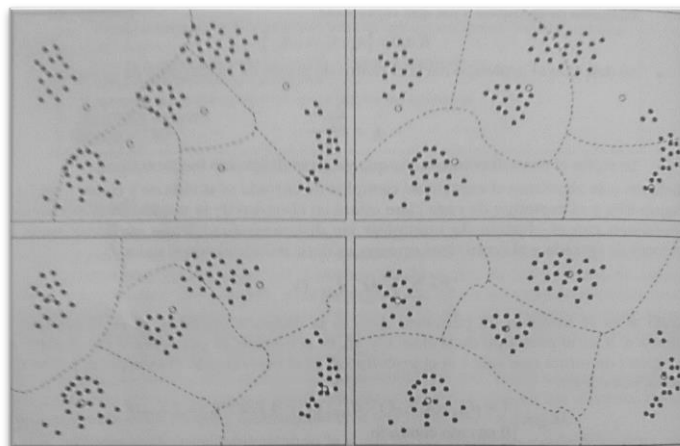


Ilustración 12: Evolución de clases formadas por el método K-medias

Fuente: José Hernández, José Ramírez, Cesar Ferri. (2004). *Introducción a la minería de datos*. Madrid: PEARSON EDUCACIÓN.

En la *ilustración 12* se puede observar la evolución en cada iteración, a medida que los patrones se van situando se producen menos cambios en las agrupaciones, y por ende el desplazamiento es menor. Así poco a poco los desplazamientos van desapareciendo.

El algoritmo de *k*-medias tiene una extensión denominada *K*-modas (*k*-modes), que fue desarrollada como una opción para casos en donde los patrones de entrada son datos categóricos, a continuación será profundizada.

- **K-modas (k-modes)**

K-modas (Huang, 2003) es una extensión para datos categóricos de *k*-medias, este algoritmo utiliza una medida de similitud entre objetos de tipo categóricos. *K*-modas calcula la moda estadística en lugar de la media como lo hace *K*-medias en el proceso de formación de centros y agrupación de los datos. Además se realiza un método donde se actualiza las modas en base a la frecuencia de los datos.

La medida de similitud entre dos objetos categóricos se da de la siguiente forma:

Si suponemos que *A*, *B* son dos objetos categóricos y cada uno de ellos tiene *m* atributos categóricos. La distancia de dichos objetos esta definida por:

$$d(A, B) = \sum_{j=1}^m \delta(a_j, b_j) \quad (24)$$

Donde:

$$\delta(a_j, b_j) = \begin{cases} 0 & (a_j = b_j) \\ 1 & (a_j \neq b_j) \end{cases} \quad (25)$$

Entonces, como se mencionó anteriormente se utiliza esta medida para calcular la distancia o diferencia entre dos objetos de tipo categóricos, con el fin de clasificar los objetos categóricos y actualizar los centros en base a la frecuencia de sus atributos, por lo tanto los pasos del algoritmo serian:

Paso 1: Se inicializa los centros de las *k* clases

Paso 2: Se asignan *n* patrones a cada clase *i*.

El patrón $X(n)$ pertenece a la clase i si $d(X(n), C_i) < d(X(n), C_s)$ para $s \neq i$ con $s = 1, 2, 3, \dots, k$.

Paso 3: Se calculan la nueva posición de los centros de las clases como la moda de todos los patrones que pertenecen a su clase.

Paso 4: Se repiten los pasos 2 y 3 hasta que las nuevas posiciones de los centros no se modifiquen respecto a su posición anterior.

$$\|C_i \text{ Nuevo} - C_i \text{ Anterior}\| < \varepsilon, \text{ para } i = 1, 2, 3, \dots, k$$

Siendo ε un número positivo cercano a 0.

- Determinación de amplitudes

Ya establecidos los centros de las funciones de base radial, las amplitudes de estas funciones se deben determinar de manera que cada neurona oculta se active de acuerdo al patrón de entrada. Para esta tarea se debe usar heurística.

Una opción bastante efectiva es determinar la amplitud de la función de base radial como la media geométrica de la distancia del centro a sus dos vecinos más cercanos. (Pedro Isasi Viñuela, 2004)

$$d_i = \sqrt{\|C_i - C_p\| \|C_i - C_s\|} \quad (26)$$

Donde:

C_p Y C_s : Son los centros más cercanos a C_i

- Índice Jaccard

El coeficiente de similitud Jaccard permite medir la similitud de dos conjuntos de datos.

La medición de similitud jaccard entre i y j donde ambos son conjuntos de datos, es el resultado de la división del número de características comunes en ambos, que viene a ser la intersección de los mismos, dividido entre su unión. Como se muestra a continuación:

$$sim(i, j) = \frac{|i \cap j|}{|i \cup j|} \quad (27)$$

Si en caso lo que se desea es hallar la distancia o la disimilitud entre los dos objetos, esta se podría hallar como la inversa del coeficiente anteriormente calculado, o igual a la unión menos la intersección todo esto dividido entre la unión de ambos conjuntos.

$$d(i, j) = \frac{|i \cup j| - |i \cap j|}{|i \cup j|} \quad (28)$$

Por lo tanto la ecuación anteriormente mostrada es la diferencia simétrica entre dos conjuntos de datos.

Similitud de atributos binarios asimétricos:

El conjunto i y j son conjuntos de n atributos binarios cada uno.

Tabla 2: **Tabla de contingencia para atributos binarios**

		Object j		
		1	0	sum
Object i	1	q	r	q + r
	0	s	t	s + t
	Sum	q + s	r + t	p

Nota. Tabla de contingencia entre dos objetos binarios “i” y “j”. Fuente: Han, J., Kamber, M., & Pei, J. (2012). *Data Mining Concepts and Techniques*. Illinois: ELSEVIER.

Donde los valores de los elementos de la *Tabla 2* son:

p : es el número total de atributos.

q : Es el número total de atributos que sean iguales a “1” en ambos conjuntos.

t : Es el número total de atributos que sean iguales a “0” en ambos conjuntos.

r : Es el número total de atributos donde el valor es “1” para el conjunto i y “0” en el conjunto j .

s : Es el número total de atributos donde el valor es “0” para el conjunto i y “1” en el conjunto j .

Por lo tanto la disimilitud entre i y j es:

$$d(i, j) = \frac{r + s}{q + r + s + t} \quad (29)$$

Para atributos binarios, los dos estados donde el atributo coincide con el del otro conjunto, es decir siguiendo la *tabla 2*, (q) y (t), ambos no son igualmente importantes, se considera más significativo la coincidencia positiva (q) cuando los atributos de los conjuntos coinciden en el valor “1”. (Han, Kamber, & Pei, 2012)

Por lo tanto la *ecuación 29* sería resumida de la siguiente forma:

$$d(i, j) = \frac{r + s}{q + r + s} \quad (30)$$

En base a la *ecuación 28* podemos afirmar que el coeficiente de similitud es igual a:

$$sim(i, j) = \frac{q + t}{q + r + s + t} \quad (31)$$

Y quitando el número de coincidencias menos significativas (t), la ecuación quedaría de la siguiente manera:

$$sim(i, j) = \frac{q}{q + r + s} \quad (32)$$

Fase Supervisada

En esta fase se calculan los pesos y umbrales, el proceso de aprendizaje está guiado por la minimización de una función de error computada en la salida de la red con el fin de minimizar las diferencias entre las salidas de la red y las salidas deseadas.

$$E = \frac{1}{P} \sum_{p=1}^P e(p) \quad (33)$$

Donde:

N : Número de patrones

$e(p)$: Error cometido por la red en el patrón $X(p)$

Además $e(p)$ está definido por:

$$e(p) = \frac{1}{2} \sum_{j=1}^s (d_j(p) - y_j(p))^2 \quad (34)$$

Donde:

$y_k(p)$: Es el vector de salida de la RNA

$d_k(p)$: Es el vector de salida deseada

s : Es el número de neuronas en la capa de salida

La reducción del error se puede realizar con el método de mínimos cuadrados.

- Mínimos cuadrados

Las salidas de las redes de base radial depende linealmente de los pesos y umbrales, por lo que un método bastante eficiente es el algoritmo de mínimos cuadrados, este algoritmo determina los pesos y umbrales mediante un proceso iterativo.

La secuencia de pasos de este método que fue recuperado del libro Redes Neuronales Artificiales un Enfoque Práctico (Pedro Isasi Viñuela, 2004), será mostrado a continuación:

Para la modificación de pesos y umbrales se utiliza las siguientes formulas:

$$w_{ij}(p) = w_{ij}(p-1) - \alpha \frac{\partial e(p)}{\partial w_{ij}} \quad (35)$$

$$\theta_j(p) = \theta_j(p-1) - \alpha \frac{\partial e(p)}{\partial \theta_j} \quad (36)$$

Para:

$$i = 1, 2, 3, \dots, k \text{ y para } j = 1, 2, 3, \dots, s$$

Donde:

$e(p)$: Es el error cometido por la red en el patrón $X(p)$

α : Es la Razón de aprendizaje

k : Es el número de neuronas en la capa oculta

s : Es el número de neuronas en la capa de Salida

Además $e(p)$ está definido por:

$$\frac{\partial e(p)}{\partial w_{ij}} = -(d_j(p) - y_j(p)) \frac{\partial y_j(p)}{\partial w_{ij}} \quad (37)$$

Derivando la salida $y_j(p)$ se tiene:

$$\frac{\partial y_j(p)}{\partial w_{ij}} = \varphi_i(p) \quad (38)$$

Donde:

$\varphi_i(p)$: Es la activación de la neurona oculta i para e patrón de entrada $X(p)$.

Derivando respecto a los umbrales, el resultado es el siguiente:

$$\frac{\partial y_j(p)}{\partial \theta_j} = 1 \quad (39)$$

Por lo tanto, los pesos y umbrales se calculan de la siguiente forma:

$$w_{ij}(p) = w_{ij}(p - 1) + \alpha(d_j(p) - y_j(p)) \varphi_i(p) \quad (40)$$

$$\theta_j(p) = \theta_j(p - 1) + \alpha(d_j(p) - y_j(p)) \quad (41)$$

Al calcular los pesos y los umbrales mediante esta ley de aprendizaje la convergencia es bastante rápida, consiguiendo una solución en un número pequeño de iteraciones o ciclos de aprendizaje.

2.1.1.8. *Redes Neuronales Recurrentes*

Las redes recurrentes se caracterizan porque se crean bucles en las neuronas de la red mediante el uso de las llamadas conexiones recurrentes, pudiendo aparecer en la red conexiones de una neurona a ella misma, conexiones entre neuronas de una misma capa o conexiones de las neuronas de una capa a la capa anterior. (Ver *ilustración 13*). (Pedro Isasi Viñuela, 2004)

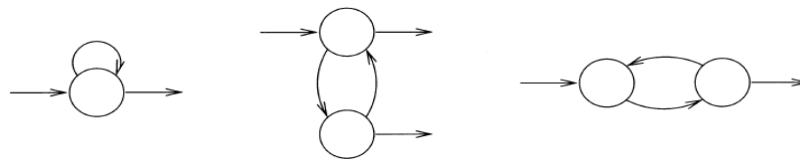


Ilustración 13: Neuronas con conexión recurrente

Fuente: Pedro Isasi Viñuela, I. G. (2004). *Redes Neuronales Artificiales un enfoque practico*. Madrid : Pearson Educación.

“Este tipo de redes son dinámicas por naturaleza o de naturaleza estática pero realimentadas de sus salidas hacia las entradas. Son herramientas poderosas para simular e identificar sistemas dinámicos no lineales”. (Graupe, 2007)

Una estructura recurrente puede ser introducida en las redes neuronales mediante la retroalimentación de la salida de la red a su entrada después de que una época de aprendizaje haya terminado.

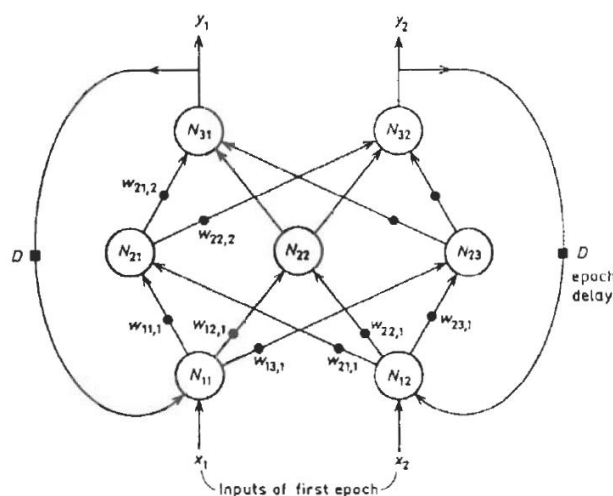


Ilustración 14: Neuronas con conexión recurrente

Fuente: Graupe, D. (2007). *Principles of Artificial Neural Networks*. Chicago - Estados Unidos: World Scientific.

En la *ilustración 14* los bucles se separan entre los Time-steps (épocas; que por lo general corresponden a iteraciones individuales). Al final de cada época las salidas son devueltas a la entrada. (Graupe, 2007)

Por ello en las redes neuronales recurrentes se debe incluir la variable tiempo y la activación o estado de una neurona.

$$A_i(t + 1) = F_i\left(\sum_j W_{ji}A_j(t)\right) \quad (42)$$

Donde el índice j varía en el conjunto de todas las neuronas conectadas a la neurona i . La presencia de la variable tiempo en las activaciones de las neuronas, hace que este tipo de redes tengan un comportamiento dinámico o temporal. (Pedro Isasi Viñuela, 2004)

a) Redes totalmente recurrentes

En este tipo de redes neuronales no existe restricción de conectividad, sus neuronas reciben como entradas la activación de otras neuronas así como su propia activación. Por lo tanto la activación de la neurona i estaría en función a:

$$A_i(t) = F_i\left(\sum_{j \in E \cup O} W_{ij}A_j(t - 1)\right) \quad (43)$$

Donde:

F_i : Es la función de activación

E: Conjunto de Neuronas de entrada

O: Resto de neurona de la red

W_{ij} : El peso de la conexión de la neurona j a la neurona i

En este tipo de red los parámetros o pesos no se consideran fijos, sino que suelen someterse a un proceso de adaptación. Esto aumenta considerablemente el número de parámetros ajustables en la red y así su capacidad de representación, pero a la vez complica su proceso de aprendizaje.

b) *Aprendizaje en redes recurrentes*

Existen dos modos de aprendizaje en las redes recurrentes, que se describirán a continuación (Haykin, 2009):

Entrenamiento “Epochwise”

Para una época determinada la red recurrente utiliza un patrón de entrada, el cual empieza a ejecutarse desde un estado inicial, una vez que alcanza un estado estable este se detiene. Nuevamente la red se reinicia con nuevos valores iniciales y vuelve a procesar un nuevo patrón dando inicio a una nueva época.

Los valores iniciales de cada época no necesariamente deben ser iguales.

Entrenamiento continuo

Este tipo de aprendizaje es importante para casos en los que no existe un reinicio de valores de inicio por época, pues se requiere que la capacitación de la red sea continua y no pare. Es decir el funcionamiento continuo no permite detener el aprendizaje y modificar los valores iniciales.

c) *Algoritmo de aprendizaje Recurrente en Tiempo Real*

Este algoritmo permite que la etapa de aprendizaje sea en tiempo real, es utilizado en las redes recurrentes en tiempo real.

Esta red recurrente en tiempo real está compuesta por N neuronas y M neuronas de entrada que reciben las señales del exterior. De las N neuronas, algunas se consideran neuronas de salida, es decir, neuronas para las que se tiene una salida deseada. Se trata de una red con $N * N$ conexiones recurrentes, pues las N neuronas se conectan con las demás e incluso con ellas mismas. Además esta red tiene $M * N$ conexiones hacia adelante, pues cada neurona de entrada se conecta con el resto de las neuronas de la red como se observar en la *ilustración 15*. Finalmente en base a esta ilustración se muestra el proceso de aprendizaje de este método que fue recuperado del libro *Redes Neuronales Artificiales un Enfoque Práctico*. (Pedro Isasi Viñuela, 2004)

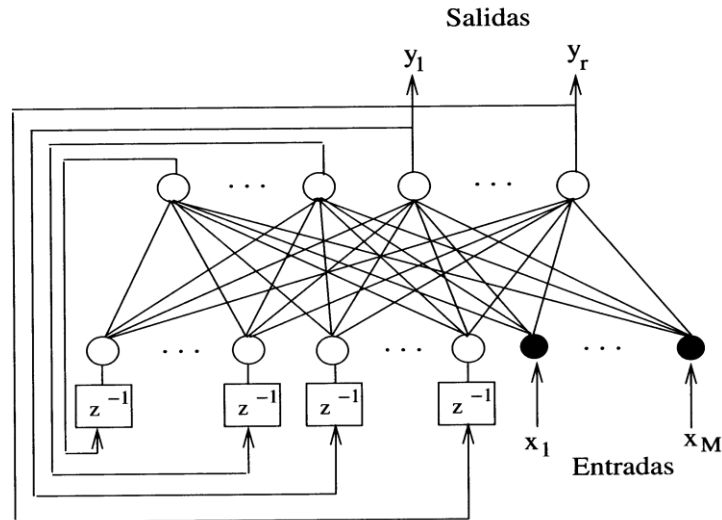


Ilustración 15: Red recurrente en tiempo real

Fuente: Pedro Isasi Viñuela, I. G. (2004). *Redes Neuronales Artificiales un enfoque practico*. Madrid : Pearson Educación.

De la ilustración, sea:

A : Es el conjunto de índices que representan las neuronas de entrada

B : Es el conjunto de índices para el resto de las neuronas de la red

Además el cardinal de A es M y el cardinal de B es N .

$x(n) = (x_i(n))_{i \in A}$: Es el vector de entrada a la red en el instante de tiempo n .

$y(n) = (y_i(n))_{i \in B}$: Es el vector formado por las activaciones de las N neuronas de la red en el instante de tiempo n .

$u(n)$: La concatenación de los vectores A Y B .

De manera que la coordenada i del vector $u(n)$ está definida por:

$$u_i(n) = \begin{cases} x_i(n) & \text{si } i \in A \\ y_i(n) & \text{si } i \in B \end{cases} \quad (44)$$

Las activaciones de las neuronas de la red vienen a ser:

$$y_i(n) = f(v_i(n-1)) \text{ para } i \in B \quad (45)$$

Donde:

f : Es la Función de activación

$v_i(n - 1)$: Es el nivel total de activación que recibe la neurona i

El nivel total de activación que recibe la neurona i está definida por:

$$v_i(n - 1) = \sum_{j \in A \cup B} W_{ij} u_j(n - 1) \quad (46)$$

Que viene a ser la suma de los productos de las entradas a la neurona por sus correspondientes conexiones. Es importante mencionar que el índice en el peso w , es decir ij se refiere a la conexión que va de la neurona j a la neurona i .

El error cometido por la red en un instante n es:

$$E(n) = \frac{1}{2} \sum_{i \in S} e_i^2(n) \quad (47)$$

Donde:

$e_i(n)$: Es el error para la neurona de salida i

El error para la neurona de salida i está definido por:

$$e_i(n) = d_i(n) - Y_i(n) \quad (48)$$

La modificación de pesos se realiza por medio de la siguiente formula:

$$W_{kl}(n) = W_{kl} - \alpha \frac{\partial E(n)}{\partial W_{kl}} \quad (49)$$

para $k \in B, l \in A$

Donde:

W_{kl} : Representa el peso de la conexión de la neurona l a la neurona k

Derivando la ecuación 47 se obtiene:

$$\frac{\partial E(n)}{\partial W_{kl}} = \sum_{i \in S} e_i(n) \frac{\partial y_i(n)}{\partial W_{kl}} \quad (50)$$

Aplicando la regla de la cadena a la *ecuación 45*, se define lo siguiente:

$$\frac{\partial y_i(n)}{\partial W_{kl}} = f'(v_i(n-1)) \frac{\partial v_i(n-1)}{\partial w_{kl}} \quad (51)$$

Y al derivar la expresión de $v_i(n-1)$ se obtiene:

$$\frac{\partial v_i(n-1)}{\partial w_{kl}} = \sum_{j \in A \cup B} \frac{\partial}{\partial w_{kl}} (w_{kl} u_j(n-1)) \quad (52)$$

En base a eso se puede afirmar que:

$$\frac{\partial v_i(n-1)}{\partial w_{kl}} = \sum_{j \in A \cup B} w_{ij} \frac{\partial u_j(n-1)}{\partial w_{kl}} + \delta_{ki} u_l(n-1) \quad (53)$$

Donde:

δ_{ki} : Es la función delta de Kronecker, vale 1 cuando $i = k$ y cero en cualquier otro caso

Por lo tanto:

$$\frac{\partial y_i(n)}{\partial w_{kl}} = f'(v_i(n-1)) \sum_{j \in A \cup B} w_{ij} \frac{\partial u_j(n-1)}{\partial w_{kl}} + \delta_{ki} u_l(n-1) \quad (54)$$

La derivada de $u_j(n-1)$ respecto al peso de la conexión es:

$$\frac{\partial u_j(n-1)}{\partial w_{kl}} = \begin{cases} 0 & \text{si } j \in A \\ \frac{\partial y_j(n-1)}{\partial w_{kl}} & \text{si } j \in B \end{cases} \quad (55)$$

Entonces:

$$\frac{\partial y_i(n-1)}{\partial w_{kl}} = f' \left(v_i(n-1) \sum_{j \in B} w_{ij} \frac{\partial y_j(n-1)}{\partial w_{kl}} + \delta_{ki} u_l(n-1) \right) \text{denotado por } P_{kl}^i \quad (56)$$

Donde P_{kl}^i está definido por:

$$P_{kl}^i = f'(v_i(n-1)) \sum_{j \in B} w_{ij} P_{kl}^i(n-1) + \delta_{ki} u_l(n-1) \quad (57)$$

para $k \in B, l \in A \cup B$ e $i \in B$

Por lo tanto la derivada del error está definida por:

$$\frac{\partial E(n)}{\partial w_{kl}} = - \sum_{i \in S} e_i(n) P_{kl}^i(n) \quad (58)$$

Donde:

P_{kl}^i : Es la salida en el instante de tiempo n

Por lo tanto con este algoritmo de aprendizaje los pesos se adaptan a cada instante de tiempo n usando las salidas calculadas anteriormente por la red. Este algoritmo no necesita almacenar en memoria el estado de la red durante un intervalo de tiempo sino que puede aplicarse en tiempo real.

2.1.2. Error cuadrático Medio (ECM)

La variable de la presente tesis es el nivel de eficiencia en la identificación de violencia escolar, que está en base al criterio de evaluación que es el Error cuadrático Medio cometido por una Red Neuronal Artificial en la identificación de violencia escolar. Este Error es igual a la diferencia entre la predicción de un modelo y el resultado real, que en esta tesis vendrían a ser el resultado de una red neuronal artificial aplicada a la identificación de violencia escolar y la situación real del estudiante respecto a la existencia de violencia escolar respectivamente, esta diferencia luego es elevada al

cuadrado. Se eleva al cuadrado para dar un mayor peso a los errores, y asegurar que todos sean positivos para la suma al momento de calcular la media.

Supongamos que y es una predicción de un modelo y que y' es el resultado real. Entonces: (Han, Kamber, & Pei, 2012)

$$(y - y')^2 \quad (59)$$

Si se trata de n predicciones se calcula la media de los errores cometidos en cada una de ellas por el modelo, de la siguiente forma:

$$ECM = \frac{1}{n} \sum_{i=1}^n (y_i - y'_i)^2 \quad (60)$$

2.1.3. Validación Cruzada (Cross validation)

Un problema muy común en el entrenamiento y la validación de modelos clasificadores o predictivos es el sobreajuste, esto ocurre cuando un modelo funciona de forma óptima con un conjunto de datos pero no es igual de óptimo cuando se le presenta un nuevo conjunto, es decir el modelo no es “generalizable”. Entonces nos preguntamos: ¿Cómo dividir el conjunto de datos para entrenamiento y validación? Y ¿Qué divisiones podrían funcionar mejor en el entrenamiento?

Por lo tanto la experimentación de los modelos se vuelve dependiente al modo en el que se hace la partición de los datos.

“Un mecanismo que permite reducir la dependencia del resultado del experimento en el modo en el cual se realiza dicha partición, es utilizar validación cruzada (cross validation)...”. (José Hernández, José Ramírez, Cesar Ferri, 2004)

La validación cruzada es una técnica que permite dividir el conjunto de datos en k subconjuntos de similar tamaño, se toma los $k - 1$ subconjuntos para el entrenamiento y el subconjunto que sobra se utiliza para la validación del modelo. En la etapa de validación se calcula los resultados del subconjunto utilizado. Este proceso se repite k veces, utilizando siempre un subconjunto diferente para validación.

Supongamos que dividimos el conjunto de datos en seis subconjuntos, entonces $k = 6$. Por lo tanto son cinco subconjuntos para entrenamiento y uno para validación. (Ver *ilustración 16*)

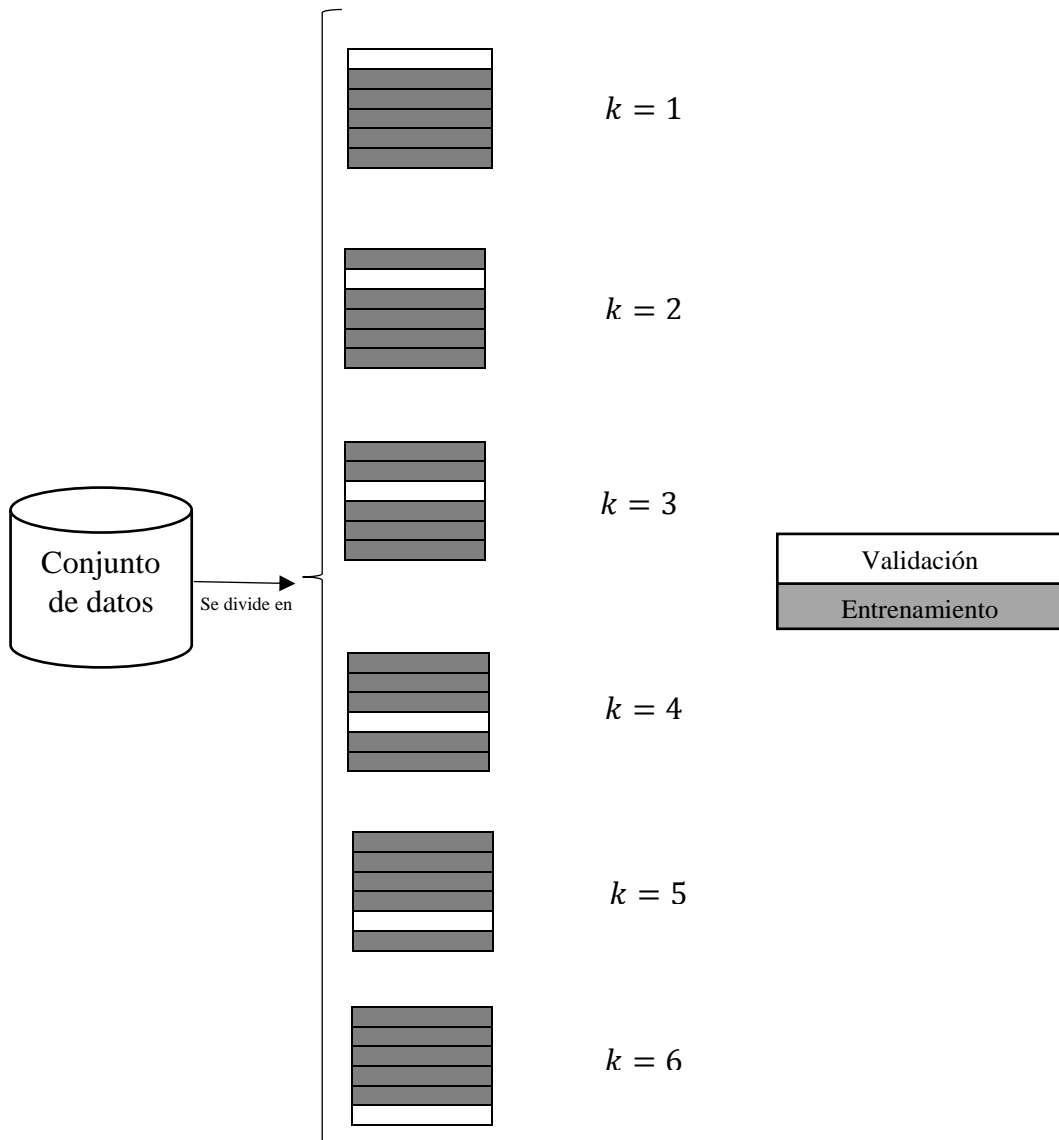


Ilustración 16: Validación cruzada

Fuente: Propia

2.2. Antecedentes

2.2.1. Antecedentes internacionales

a) Antecedente N° 1: Comparación de Redes Neuronales aplicadas a la predicción de Series de Tiempo. (Darwin Mercado Polo, 2015)

Este estudio considera una red neuronal artificial de tipo perceptrón multicapa (MLP) con un algoritmo de aprendizaje resilient backpropagation y una red neuronal de base radial (RBF) con el algoritmo de aprendizaje de los k-emanes y el método de la matriz pseudoinversa como los dos tipos de redes neuronales a comparar.

Dataset (Id)	Tiempo (segundos)	
	MLP	RBF
1	6,1	3,6
2	7,2	3,5
3	9,3	4,1
4	7,5	3,8
5	6,4	3,7
6	9,2	4,5
7	6,2	3,6
8	7,3	3,8
9	7,1	3,6
10	6,5	3,4

Ilustración 17: Tiempos de ejecución en segundos de las redes neuronales para los diferentes datasets del estudio de Comparación de Redes Neuronales aplicadas a la predicción de Series de Tiempo

Fuente: D. Polo, L. Pedraza, E. Martínez, "Comparison of Neural Network applied to prediction of times Series", Prospect, Vol 13, N° 2, 88-95, 2015.

Dataset (Id)	Porcentaje Efectividad (%)	
	MLP	RBF
1	65,4	66,5
2	66,5	70,1
3	74,5	72,3
4	75,4	77,2
5	66,5	68,5
6	74,7	75,5
7	76,8	75,2
8	75,8	76,4
9	75,6	77,9
10	78,5	76,4

Ilustración 18: Niveles de efectividad alcanzados por las redes neuronales para los diferentes datasets del estudio de Comparación de Redes Neuronales aplicadas a la predicción de Series de Tiempo

Fuente: D. Polo, L. Pedraza, E. Martínez, "Comparison of Neural Network applied to prediction of times Series", Prospect, Vol 13, N° 2, 88-95, 2015.



En la *ilustración 17* se puede observar los tiempos de ejecución de las redes neuronales. En la *ilustración 18* se muestra los niveles de efectividad alcanzados por las redes neuronales artificiales en los diferentes conjuntos de datos en la predicción de series de tiempo.

Los desarrolladores de esta investigación llegaron a la conclusión de que la red de base radial es más eficiente en cuanto tiempo de ejecución, mientras que en calidad de respuesta los dos tipos se aproximan mucho, y que entre las redes de base radial y las de tipo perceptrón multicapa existe un 95% de certeza al afirmar que no son totalmente diferentes, pero en cuanto a las series de tiempo que es donde se aplicó su investigación recomiendan el uso de una red neuronal de base radial.

b) Antecedente N° 2: Comparison of artificial neural network models for hydrologic predictions at multiple gauging stations in an agricultural watershed. (Mutlu, I. Chaubey, & Bajwa, 2008)

Este artículo de investigación se da a conocer la evaluación de dos modelos de redes neuronales artificiales aplicadas a la predicción hidrológica en múltiples calibraciones estacionales en una cuenca agrícola. Los tipos de redes neuronales artificiales que evaluaron son: un perceptrón multicapa (MLP) y una red neuronal de base radial (RBFNN). Los resultados de las pruebas en cuatro escenarios diferentes y con diferentes calibraciones estacionales para cada modelo fueron:

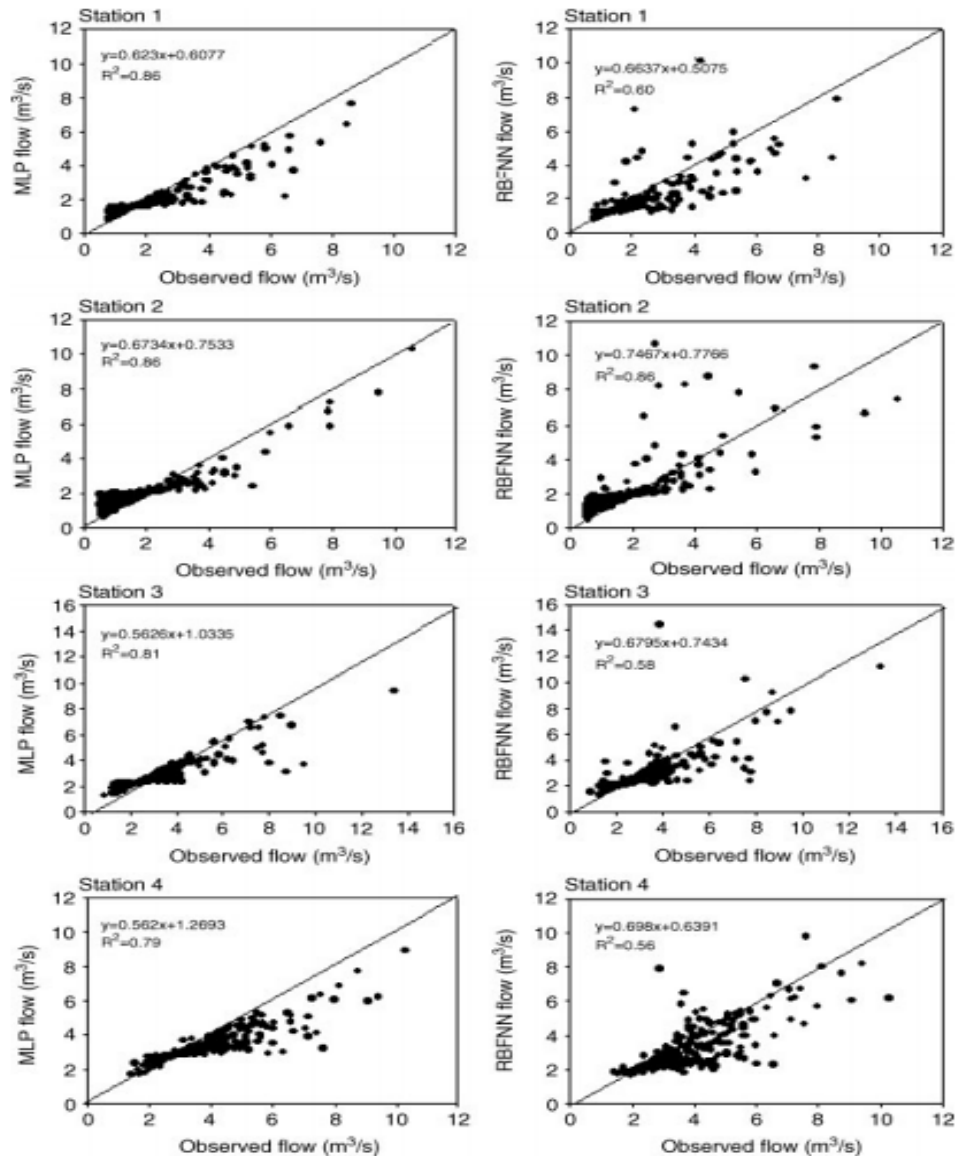


Ilustración 19: Resultados observados de la evaluación entre una MLP y una RBFNN en el estudio titulado “Comparison of artificial neural network models for hydrologic predictions at multiple gauging stations in an agricultural watershed”

Fuente: Mutlu, E., I. Chaubey, H., & Bajwa, H. y. (2008). Comparison of artificial neural network models for hydrologic predictions at multiple gauging stations in an agricultural watershed. *Hydrological processes*, 22, 5097-5106.

En base a estos resultados observados en la *ilustración 19* los desarrolladores de esta investigación determinaron que el modelo MLP frente al modelo RBFNN obtuvo mejores resultados en la predicción hidrológica en base a los R^2 ajustados calculados en cada una de las cuatro estaciones.

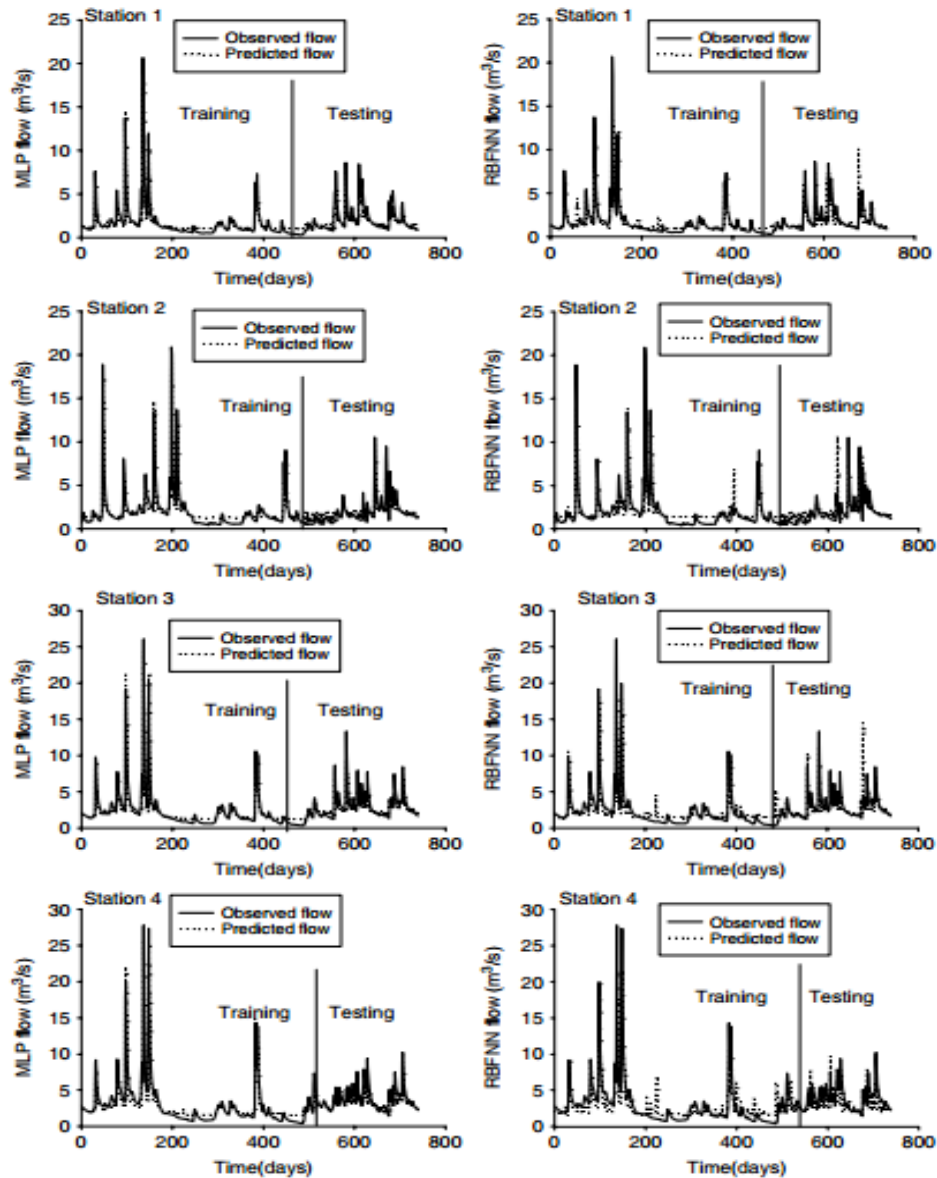


Ilustración 20: Flujo observado y el predicho por los algoritmo en base al tiempo de entrenamiento y validación en el estudio titulado Comparison of artificial neural network models for hydrologic predictions at multiple gauging stations in an agricultural watershed
Fuente: Mutlu, E., I. Chaubey, H., & Bajwa, H. y. (2008). Comparison of artificial neural network models for hydrologic predictions at multiple gauging stations in an agricultural watershed. *Hydrological processes*, 22, 5097-5106.

En este estudio también se analizó el tiempo de entrenamiento y validación de ambos modelos, como se observa en la *ilustración 20*. Los investigadores de este estudio concluyeron que el modelo MLP alcanzó tiempos de ejecución en un rango de 1-2 segundos, mientras que el modelo RBFNN un rango de 5-10 segundos. Por lo tanto ellos afirman que un modelo MLP es más eficiente en la predicción hidrológica.

c) Antecedente N° 3: Predicción al consumo de éxtasis a partir de Redes Neuronales Artificiales. (Palmer Pol, Montaña Moreno, & Calafat Far, 2000)

El objetivo de este estudio fue mostrar como una red neuronal artificial (RNA) puede ser útil para predecir el consumo de éxtasis (MDMA). Tratando de discriminar entre los que consumen éxtasis y los que no lo hacen. Para el entrenamiento de la RNA utilizaron el algoritmo de retro propagación de error (back propagation) en la etapa de entrenamiento, los datos utilizaron se trataba de cuestionarios de 25 preguntas cada uno, con una muestra de de 148 consumidores de éxtasis y 148 no consumidores de éxtasis. En los resultados de la validación, su modelo de red neuronal artificial alcanzo un 96.66% de nivel de eficiencia, es decir que la RNA con las respuestas de las 25 preguntas de un sujeto, podría predecir si este sujeto es consumidor o no consumidor de éxtasis con un margen de error muy pequeño. En esta investigación también se demostró que los pesos del modelo de la red neuronal artificial pueden dar información acerca del grado de influencia de las variables de entrada sobre las de salida, a si conocer cuáles son los indicadores o datos de entrada más desencadenantes.

d) Antecedente N° 4: Redes neuronales artificiales aplicadas al análisis de supervivencia: un estudio comparativo con el modelo de regresión de Cox en su aspecto predictivo. (Pol Plamer & Montaña Moreno, 2002)

Este estudio se comparó el rendimiento de predicción entre un modelo de Redes Neuronales Artificiales y un modelo de riesgos proporcionales de Cox en el contexto de análisis de supervivencia. Básicamente se comprobó:

- Si el modelo de redes neuronales jerárquicas es más preciso que el modelo de Cox.
- Si el modelo de redes secuenciales supone una mejora respecto al modelo de redes neuronales jerárquicas.

Las conclusiones de este estudio fueron:

- Que las redes neuronales jerárquicas tienen un mejor rendimiento en resolución que el modelo de Cox.
- Que las redes secuenciales no suponen una mejora respecto a las redes neuronales jerárquicas.

e) *Antecedente N° 5: Probability density estimation using artificial neural network. (Aristidis Likas, 2001)*

En esta investigación se propuso el uso de Redes Neuronales Artificiales para la estimación de densidad de probabilidad (pdf), se evaluó el desempeño de un modelo de red neuronal artificial tipo perceptrón multicapa con función de activación sigmooidal en las neuronas de la capa oculta (original), además de un modelo modificando las fases de su pre procesamiento donde los parámetros de la red se inicializaron mediante un aprendizaje supervisado (MLP), ambos modelos fueron comparados con el método de mezclas gaussianas.

Se comprobó en este estudio que el modelo *MLP* presentado tiene una capacidad superior frente al modelo *original* y además al usualmente usado modelo de mezclas gaussianas.

f) *Antecedente N° 6: Redes neuronas recurrentes para el reconocimiento de patrones temporales. (Pérez Segovia, 1992)*

El propósito de esta tesis fue el estudio de redes neuronales recurrentes aplicadas al reconocimiento de patrones temporales. A través de una revisión de conocimientos sobre estos temas esta investigación pretende aumentar la capacidad de las redes recurrentes y en base a la introducción de nuevas ideas en el diseño de sus arquitecturas permitió crear una base teórica, así como práctica para el diseño de redes recurrentes que resuelvan problemas en el dominio temporal, hasta ese momento sin solución.

	ENTR		TEST	
	ARV	ERROR	ARV	ERROR
R1	0.26781	0.00615	0.23033	0.00255
R2	0.09364	0.00227	0.07984	0.00088
R3	0.08165	0.001986	0.07703	0.00085
R4 (tanh)	0.12721	0.003094	0.09576	0.00106

Ilustración 21: Varianza media relativa de los modelos recurrentes planteados

Fuente: Recuperado de Pérez Segovia, F. J. (Febrero de 1992). *Redes de neuronas recurrentes para el reconocimiento de patrones temporales (Tesis doctoral)*. Madrid, España: Universidad Politécnica de Madrid.

En la *ilustración 21* se puede observar los resultados de la evaluación realizada en esta tesis, en el entrenamiento y el test de las redes recurrentes. El investigador de esta tesis afirma que en los modelos R2 y R3 los resultados no son demasiados



diferentes y que tienen como diferencia 100 conexiones adicionales recurrentes, también afirma que las pruebas de testeo dan mejores resultados que las realizadas con el conjunto de entrenamiento, y esto es debido a que la señal sea mucho más uniforme en el conjunto de test, con formas bastante bien codificadas dentro de la red.

Esta tesis llega a la conclusión que a través de este estudio se han podido extraer una serie de importantes afirmación, que son las siguientes:

- Una neurona obedeciendo a una ecuación usual de activación puede encontrarse en tres estados diferentes dependiendo del valor de su realimentación, los cuales son: oscilaciones, estabilidad asintótica hacia el 0 y estabilidad asintótica hacia un punto distinto de 0.
- La creación de oscilaciones en las neuronas aisladas depende grandemente del algoritmo de aprendizaje.
- La creación de estados estables distintos de 0 creados en neuronas aisladas permiten que estas sean sensibles al número y frecuencia de los pulsos que tengan como entrada.
- El diseño jerarquizado de las redes recurrentes permite que estas resuelvan correctamente problemas que sin un diseño similar sería imposible, como el caso del tratamiento simultaneo de entradas binarias y analógicas o la predicción de series temporales.
- En problemas tratados con redes multicapa donde existe un carácter temporal, es preferible introducir recurrencias ya que no solo aportan un mayor número de pesos (y por tanto mayor potencia a la red), sino que las recurrencias permiten recoger más adecuadamente la influencia de la variable tiempo.

2.2.2. Antecedentes nacionales

a) Antecedente N° 7: Aplicación de Redes Neuronales Artificiales a la Modelización y Previsión de Caudales Medios Mensuales del Río Huancané. (Vilca Laqui, 2010)

En este estudio se desarrollaron modelos basados en redes neuronales artificiales con el algoritmo de aprendizaje de retro propagación aplicados a la modelización y previsión de los caudales medios mensuales del río Huancané, en base a datos medios mensuales de caudales, precipitación y evaporación esta información se utilizó en las etapas de aprendizajes y validación. Estos modelos fueron compararon frente a un modelo autor regresivo periódico de primer orden (PAR).

Se desarrollaron cuatro modelos de estos el modelo número tres fue el más eficiente, este fue comparado con el modelo PAR.

Modelos	Estadísticas				
	ECM	EAM	EPM	EPmax	R ²
PAR (1)	961.03	15.90	86.14	1,858.49	0.59
RNA Modelo 3	386.79	9.37	60.20	1,220.15	0.72

Ilustración 22: Resultados del estudio de Aplicación de Redes Neuronales Artificiales a la Modelización y Previsión de Caudales Medios Mensuales del Río Huancané

Fuente: Recuperado de Vilca Laqui, W. F. (2010). Aplicación de Redes Neuronales Artificiales a la Modelización y Previsión de Caudales Medios Mensuales del Río Huancané. *Peruana Geo-Atmosférica RPGA*, 2, 30-44.

Como se puede observar en la *ilustración 22* el modelo de redes neuronales artificiales número tres consiguió un menor error promedio máximo (EPmax), por lo tanto es más eficiente que el modelo PAR.

CAPITULO III: METODOLOGÍA

3.1. Tipo de investigación

La presente investigación tiene un enfoque Cuantitativo y en cuanto al nivel es Descriptivo. Según Sampieri:

“Las investigaciones descriptivas únicamente pretenden medir o recoger información de manera independiente o conjunta sobre los conceptos o las variables a las que se refieren, esto es, su objetivo no es indicar cómo se relacionan éstas”. (Sampieri, 2014)

En esta investigación se determinó a través de un estudio comparativo la red neuronal artificial que tiene un mayor nivel de eficiencia en la identificación de violencia escolar en las instituciones educativas. En el estudio comparativo se modificaron los indicadores para medir el nivel de eficiencia alcanzado por cada red neuronal artificial, pero no se analizó cuál es la relación entre sus indicadores y tampoco cuanto influyó cada indicador en dicho nivel de eficiencia.

3.2. Diseño de la investigación

Según Sampieri:

“...los diseños experimentales se utilizan cuando el investigador pretende establecer el posible efecto de una causa que se manipula...” (Sampieri, 2014)

Por lo tanto se puede afirmar que la presente investigación tiene un diseño experimental, pues se manipularon los indicadores de la variable dependiente, la cual es el nivel de eficiencia de las redes neuronales artificiales en la identificación de violencia escolar, de manera que se analizó el efecto que esta acción ocasiono en dicha variable y en base a su criterio de evaluación que es el error cuadrático medio (ECM) cometido por las redes neuronales artificiales en las experimentaciones realizadas. Las experimentaciones forman parte del estudio comparativo que fue el medio para determinar la red neuronal artificial que alcanzó un mayor nivel de eficiencia en la identificación de violencia escolar.

El estudio comparativo incluye las siguientes etapas:

3.2.1. Etapa 1: El diseño de una base de datos para las redes neuronales artificiales

Se diseñó una base de datos que incluyó la información recolectada de las instituciones educativas en una entidad, pues esta información es necesaria para el procesamiento de las redes neuronales artificiales, además la base de datos diseñada también cuenta

con otras entidades más que permiten el registro de los resultados del procesamiento de los algoritmos.

3.2.2. Etapa 2: El desarrollo y ajuste de redes neuronales artificiales aplicadas a la identificación de violencia escolar

Las seis redes neuronales artificiales que se tomaron en cuenta en la investigación están divididas en tres tipos de redes y métodos de aprendizaje, es decir cada dos comparten el mismo tipo y método de aprendizaje, con una diferencia en su función de activación, por lo tanto para cada tipo se realizó un algoritmo que permitió la selección de la función de activación.

a) Tipo 1: Perceptrón multicapa con método de aprendizaje de Retropropagación

- Con función de activación Sigmoidal (PMS)
- Con función de activación Tangente Hiperbólica (PMH)

b) Tipo 2: Red Neuronal de base radial con método de aprendizaje Híbrido

- Con función Gaussiana (BRG)
- Con función Inversa Cuadrática (BRC)

c) Tipo 3: Red Neuronal Recurrente con método de aprendizaje Recurrente en Tiempo Real

- Con función Sigmoidal (RRS)
- Con función Tangente Hiperbólica (RRH)

3.2.3. Etapa 3: Experimentación de las redes neuronales artificiales

En esta etapa se realizó la experimentación de las redes neuronales artificiales con diferentes valores en sus indicadores, para obtener los mejores niveles de eficiencia alcanzados por cada uno de ellos.

Con el desarrollo de estas tres etapas se logró determinar la red neuronal artificial que alcanzó un mayor nivel de eficiencia en la identificación de violencia escolar. En la *ilustración 23* se resume el diseño de la presente investigación:

Diseño de Base de
datos para los
algoritmos RNA

Desarrollo y ajuste de
RNA aplicadas a la
identificación de
violencia escolar

Experimentación de las
RNA

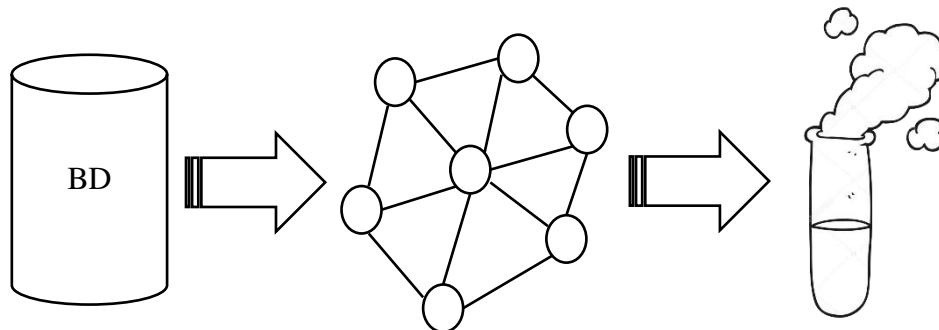


Ilustración 23: Diseño de la investigación

Fuente: Propia

3.3. Población y muestra

3.3.1. Población:

Como población se toma en cuenta cuatro centros educativos, que están detallados en la siguiente tabla:

Tabla 3: Instituciones educativas de donde se obtendrá la información

Numero de Institución Educativa	Nombre de Institución Educativa	Número de estudiantes
50584	Institución Educativa Ollantaytambo	177
50598	Institución Educativa Wiñayninchis	90
50721	Institución Educativa Chicon	120
50572	Institución Educativa San Luis de Gonzaga 711	310

Nota. Instituciones educativas de donde se recolecto la información de los estudiantes. Fuente: Propia

3.3.2. Muestra:

La muestra está constituida por 120 datos de casos de violencia escolar en estudiantes, donde también se conoce la situación real de cada caso respecto a la existencia de este problema social. El muestreo es no probabilístico por conveniencia, pues:

“Permite seleccionar aquellos casos accesibles que acepten ser incluidos. Esto, fundamentado en la conveniente accesibilidad y proximidad de los sujetos para el investigador”. (Otzen & Manterola, 2017)

En la presente investigación se da por la accesibilidad a la información.

En la siguiente tabla se muestra la información de la muestra:

Tabla 4: **Muestra**

Institución Educativa	Número de Estudiantes
Institución Educativa Ollantaytambo	30
Institución Educativa Wiñayninchis	53
Institución Educativa Chicon	27
Institución Educativa San Luis de Gonzaga 711	15
Total	120

Nota. Número de registros de estudiantes recolectados para la muestra.

Fuente: Propia

3.3.2.1. *Preparación de la muestra*

Luego de obtener los datos es necesario realizar la preparación de los mismos para que de esta forma puedan ser utilizados como datos de entrada en las redes neuronales artificiales, para lograr esta tarea se realizaron una serie de pasos, que serán detallados a continuación:

1. *Descripción de la información recolectada*

Los datos que fueron obtenidos de las instituciones educativas tiene el siguiente conjunto de datos:

a) *Estudiante*

Este conjunto de datos tiene los siguientes atributos, divididos de la siguiente manera:

- Información personal del estudiante.- Estos atributos son de identificación.

Los atributos son:

- Nombre: Nombre del estudiante.
- Edad: Edad del estudiante:

- Los indicadores de convivencia y la situación real del estudiante.- Son los 11 indicadores de la situación de los estudiantes en sus hogares y en las instituciones educativas, además de la situación real del estudiante respecto

a la existencia de violencia escolar estos atributos son preguntas cerradas, es decir solo hay la posibilidad de responder “sí” o “no”. Los atributos son:

- ¿Vive con sus padres?
- ¿Vive solo con mama?
- ¿Vive solo con papa?
- ¿Vive con otros?
- ¿Asiste descuidadamente?
- ¿Asiste a la institución sin tomar sus alimentos?
- ¿Manifiesta desinterés por el estudio?
- ¿Agrede a sus compañeros?
- ¿Reacciona pasivamente ante los maltratos?
- ¿Utiliza tonos y gestos inapropiados para hacerse escuchar?
- ¿No manifiesta maltrato a su integridad?
- ¿Existe violencia escolar en el estudiante?

2. Transformación de la información recolectada

Se realizó la transformación de los atributos del conjunto de datos obtenido para que puedan ser procesados por las redes neuronales artificiales, para lograr esto se utilizó una hoja de cálculo electrónica y solo se remplazaron los valores “sí” y “no” siguiendo la siguiente regla:

Tabla 5: **Regla para transformar la información recolectada**

Valores	Se remplazaron con
Si	1
No	0

Nota. La tabla muestra los valores que deben ser remplazados en la transformación de la data. Fuente: propia

3. Descripción detallada del conjunto de datos

En la siguiente tabla se describe el juego de datos de forma detallada, como el nombre, tipo del archivo, así como el número de registros que contiene.

Tabla 6: **Información de conjunto de datos Estudiante**

Nro	Conjunto de datos	Nombre del archivo	Número de registros	Numero de atributos
1	Student	student.csv	120	14

Nota. En esta tabla se detalla la información del conjunto de datos preparado. Fuente: Propia

En la siguiente tabla se detalla los tipos y la descripción del conjunto de datos:

Tabla 7: Atributos del conjunto de datos Estudiante

Nro	Atributo	Tipo	Descripción
1	Nombre	String	Nombre del estudiante
2	Edad	Int	Edad del estudiante
3	Indicador 1	Boolean	¿Vive con sus padres?
4	Indicador 2	Boolean	¿Vive solo con mama?
5	Indicador 3	Boolean	¿Vive solo con papa?
6	Indicador 4	Boolean	¿Vive con otros?
7	Indicador 5	Boolean	¿Asiste descuidadamente?
8	Indicador 6	Boolean	¿Asiste a la institución sin tomar sus alimentos?
9	Indicador 7	Boolean	¿Manifiesta desinterés por el estudio?
10	Indicador 8	Boolean	¿Agrede a sus compañeros?
11	Indicador 9	Boolean	¿Reacciona pasivamente ante los maltratos?
12	Indicador 10	Boolean	¿Utiliza tonos y gestos inapropiados?
13	Indicador 11	Boolean	¿No manifiesta maltrato a su integridad?
14	Situación real	Boolean	¿Existe violencia escolar en el estudiante?

Nota. Esta tabla detalla los atributos del conjunto de datos Estudiante, donde los indicadores son utilizados por algunas instituciones educativas de la provincia de Urubamba para el monitoreo de un posible indicio de violencia escolar en los estudiantes, es factible mencionar que la situación real serán los resultados esperados en el proceso de aprendizaje de las redes neuronales artificiales. Fuente: Propia

4. Estadística del conjunto de datos

De los atributos observados en el *Tabla 7*; tan solo los atributos de nombre y edad no son necesarios para el procesamiento de los algoritmos de Redes neuronales artificiales, de los demás atributos que si son necesarios se realizó un estudio estadístico y en la siguiente tabla se detalla dicho estudio:

Tabla 8: Estadística del conjunto de datos

Nro	Variable	Tipo	Count	Top	Freq	NA	NA%
1	Indicador 1	boolean	120	1	69	0	0
2	Indicador 2	boolean	120	0	95	0	0
3	Indicador 3	boolean	120	0	106	0	0
4	Indicador 4	boolean	120	0	104	0	0
5	Indicador 5	boolean	120	0	71	0	0
6	Indicador 6	boolean	120	0	73	0	0
7	Indicador 7	boolean	120	0	68	0	0
8	Indicador 8	boolean	120	0	85	0	0
9	Indicador 9	boolean	120	0	82	0	0
10	Indicador 10	boolean	120	0	74	0	0
11	Indicador 11	boolean	120	0	62	0	0
12	Situación real	boolean	120	0	71	0	0

Nota. En esta tabla se puede observar un análisis estadístico del conjunto de datos Estudiante donde la columna "count" es la cantidad de datos, "top" es el valor que se repite frecuentemente, "Freq" es el número de veces que se repite el valor top y NA/NA% indican datos vacíos. Fuente: propia

3.4. Técnicas de recolección de datos

Para la recolección de los datos se usó un cuestionario con doce preguntas de convivencia del estudiante en la institución educativa (*Ver ANEXO A*), incluyendo la situación real del estudiante respecto a la existencia de violencia escolar en la institución o en su hogar. Esta información fue registrada por los directores de las instituciones educativas.

3.5. Técnicas de procesamiento de datos

La información recolectada será procesada en los algoritmos de redes neuronales artificiales desarrollados y para los resultados obtenidos del procesamiento de los algoritmos se utilizara gráficos de líneas para analizar la evolución de los errores (ECM) de aprendizaje y validación en cada ciclo de su procesamiento, además de matrices de datos para comparar cada escenario experimentado en el Capítulo IV.

3.6. Aspectos Administrativos

3.6.1. Recursos

- *Humanos*

Investigador: Bach. Luis Ernesto Colchado Soncco

Asesor: Mgt. Edwin Carrasco Poblete

Muestra de estudio: 120 estudiantes de cuatro instituciones educativas de la provincia de Urubamba.

3.6.2. Presupuesto

Tabla 9: Presupuesto de la Investigación

BIENES			
Cantidad	Descripción	Precio unitario (S/.)	Total (S/.)
2 Unidades	Lapiceros	1.50	3.00
1 Unidad	Millar de papel	9.80	9.80
1 Unidad	Tinta para impresora Canon	45.00	45.00
1 Unidad	Resaltador	3.00	3.00
1 Unidades	Laptop Xiaomi Procesador Intel CORE i5 6ta generación	2850.00	2850.00
1 Unidades	Memoria USB Kingston 8Gb	17.00	17.00
		Total	2,927.8



SERVICIOS			
Cantidad	Descripción	Precio unitario (S/.)	Total (S/.)
6 meses	Internet	129.00	516.00
		Total	516.00
SOFTWARE			
Cantidad	Descripción	Precio unitario (S/.)	Total (S/.)
1 licencia gratuita	Entorno de desarrollo Eclipse C/C++	0.0	0.0
1 licencia gratuita	Gestor de Base de datos SQLITE	0.0	0.0
		Total	0.0
SISTEMA DE CONTROL DE VERSIONES			
Cantidad	Descripción	Precio unitario (S/.)	Total (S/.)
1 cuenta gratuita	Bitbucket	0.0	0.0
		Total	0.0
TOTAL			3,443.8

Nota. Presupuesto utilizado para llevar a cabo la presente investigación. Fuente: Propia

CAPITULO IV: ESTUDIO COMPARATIVO DE REDES NEURONALES ARTIFICIALES

El objetivo del presente capítulo fue realizar un estudio comparativo de las seis redes neuronales artificiales detalladas en el diseño de la investigación en el capítulo anterior, donde también se da a conocer las tres etapas que se debieron realizar para llevarlo a cabo.

4.1. Etapa 1: Diseño de una base de datos para las Redes neuronales artificiales

Las redes neuronales artificiales necesitan de datos de entrada tanto en la etapa de aprendizaje como en la validación, en la primera para que aprenda de los patrones ingresados y en la segunda para que pueda ser validada con nuevos patrones, que no fueron procesados en la primera etapa de aprendizaje. Por otra parte el procesamiento de cada red neuronal proporciona los errores cuadráticos medios cometidos en las dos etapas antes mencionadas.

Por lo tanto se diseñó una base de datos que pueda almacenar la información necesaria para el procesamiento de las redes neuronales artificiales, así como para el registro de los resultados obtenidos de dicho procesamiento.

A continuación se explica las fases del diseño de la base de datos:

4.1.1. Identificación de entidades y atributos

Para el diseño de la base de datos primero se debió identificar y definir las entidades con sus respectivos atributos que son necesarios para el procesamiento de las redes neuronales artificiales aplicadas a la identificación de violencia escolar. Las entidades que fueron identificadas son:

- Estudiante.- Esta entidad almacena el conjunto de datos Estudiante, por lo tanto los atributos de esta entidad están en base a lo detallado en la *Tabla 7*.
- Resultados de Perceptrón Multicapa.- Esta entidad almacena los resultados de las experimentaciones de las redes neuronales artificiales de tipo Perceptrón Multicapa con método de aprendizaje de Retropropagación, esta entidad tiene los siguientes atributos:
 - Ciclo de validación cruzada.- Indica el ciclo de la validación cruzada.
 - Ciclo de entrenamiento.- Indica el ciclo de aprendizaje.



- Error en el aprendizaje.- Almacena el error cuadrático medio en el entrenamiento.
 - Error en la validación.- Almacena el error cuadrático medio en la validación
 - Función de activación.- Almacena la función de activación seleccionada en el algoritmo.
- Resultados de Red Neuronal de Base Radial.- Esta entidad almacena los resultados de las experimentaciones de las redes neuronales artificiales de tipo Red Neuronal de Base Radial con método de aprendizaje Híbrido, esta entidad tiene los siguientes atributos:
 - Ciclo de validación cruzada.- Indica el ciclo de la validación cruzada.
 - Ciclo de aprendizaje.- Indica el ciclo de aprendizaje.
 - Error en el aprendizaje.- Almacena el error cuadrático medio en el entrenamiento.
 - Error en la validación.- Almacena el error cuadrático medio en la validación.
 - Función de activación.- Almacena la función de activación seleccionada en el algoritmo.
- Resultados de Red Neuronal Recurrente.- Esta entidad almacena los resultados de las experimentaciones de las redes neuronales artificiales de tipo Red Neuronal Recurrente con método de aprendizaje Recurrente en Tiempo Real, esta entidad tiene los siguientes atributos:
 - Ciclo de validación cruzada.- Indica el ciclo de la validación cruzada.
 - Ciclo de aprendizaje.- Indica el ciclo de aprendizaje.
 - Error en el aprendizaje.- Almacena el error cuadrático medio en el entrenamiento.
 - Error en la validación.- Almacena el error cuadrático medio en la validación.
 - Función de activación.- Almacena la función de activación seleccionada en el algoritmo.

4.1.2. Modelo conceptual

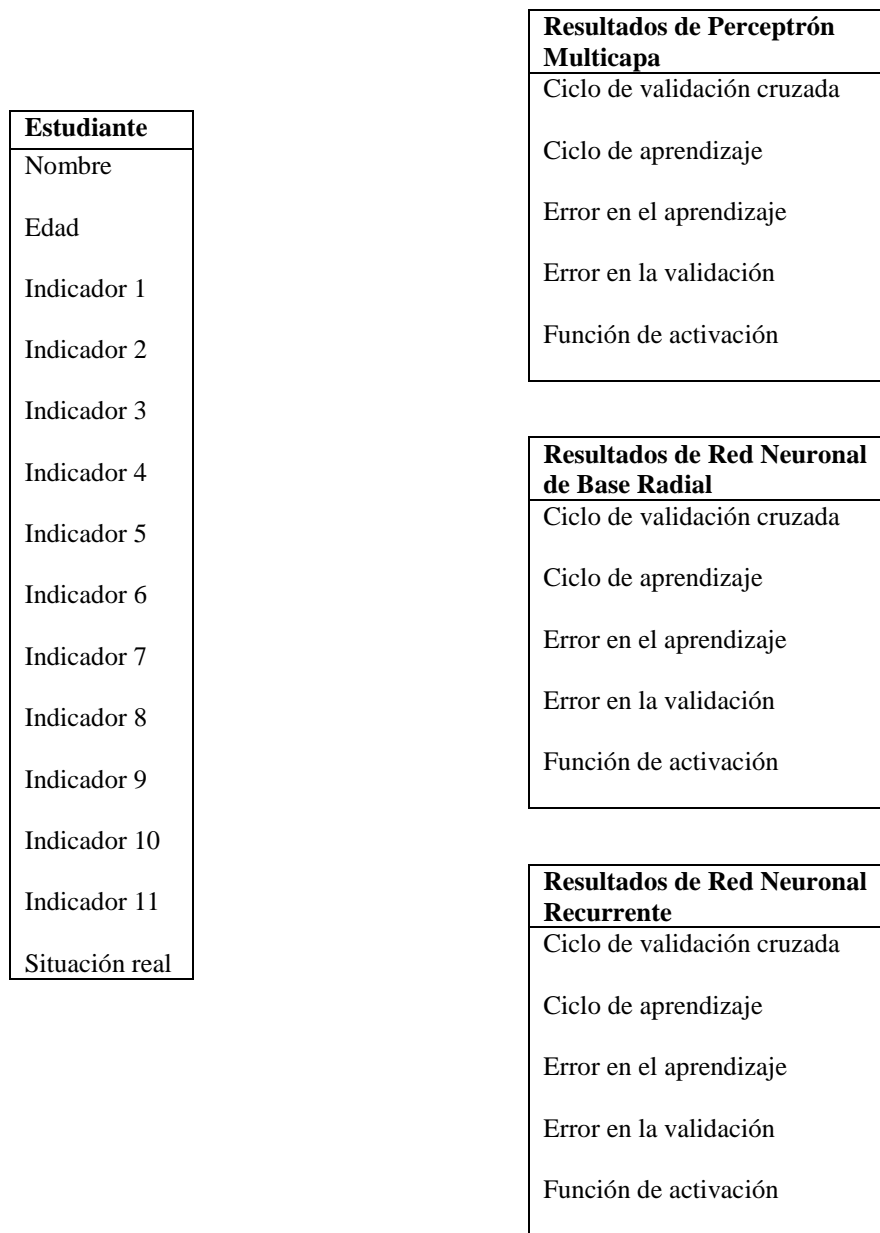


Ilustración 24: Modelo conceptual de la base de datos para los algoritmos de RNA

Fuente: propia

En la *ilustración 24* se muestra el modelo conceptual de la base de datos utilizada para la el procesamiento de las redes neuronales artificiales.

4.2. Etapa 2: Desarrollo y ajuste de redes neuronales artificiales aplicadas a la identificación de violencia escolar

En esta sección se detalla el proceso realizado para el desarrollo y ajuste de redes neuronales artificiales aplicadas a la identificación de violencia escolar.

Las redes neuronales artificiales propuestas en la investigación se dividen en tres tipos de redes que son el Perceptrón Multicapa (PM), las Redes Neuronales de Base Radial (BR) y las Redes Neuronales Recurrentes (RR), en cada uno de estos tipos se aplicó un diferente método de aprendizaje, estos son: El método de Retropropagación (Backpropagation) para redes de tipo PM, el método Híbrido para BR y el método de aprendizaje Recurrente en Tiempo Real (Real-Time Recurrent Learning) para RR. Otro factor que se consideró fueron las funciones de activación de las neuronas, como: La función de activación Sigmoidal, Tangente hiperbólica, Inversa cuadrática y Gaussiana.

En la siguiente tabla se muestra las redes neuronales artificiales que serán desarrolladas en la presente sección:

Tabla 10: **Redes neuronales artificiales consideradas**

Tipo	Red neuronal artificial	Tipo de red	Método de Aprendizaje	Función de Activación
Tipo 1 (PM)	PMS	Perceptrón Multicapa	Retropropagación	Función Sigmoidal
	PMH			Función Hiperbólica
Tipo 2 (BR)	BRG	Redes Neuronales de Base Radial	Híbrido	Función Gaussiana
	BRC			Función Inversa Cuadrática
Tipo 3 (RR)	RRS	Redes Neuronales Recurrentes	Recurrente en Tiempo Real	Función Sigmoidal
	RRH			Función Hiperbólica

Nota. Redes neuronales artificiales que fueron consideradas en la presente investigación. Fuente: Propia

Como se detalla en la *tabla 10* las redes neuronales artificiales se dividen en tres tipos de redes y métodos de aprendizaje, además que en cada uno se tiene una función de activación distinta. Las redes de tipo Perceptrón Multicapa (PM) de acuerdo a la teoría



comúnmente utilizan el método de aprendizaje de Retropropagación y usualmente se utilizan funciones de activación de tipo Sigmoidal o Tangente Hiperbólica. En cambio las Redes Neuronales de Base Radial (BR) requieren un método de aprendizaje distinto, porque este tipo de red necesita primeramente el cálculo de los centros y amplitudes de las neuronas de la capa oculta y posteriormente el ajuste de los pesos sinápticos para la capa de salida, es por ello que se utiliza en este caso el método de aprendizaje Híbrido, que incluye las secuencias necesarias para cumplir con el aprendizaje de este tipo de redes neuronales, además estas redes utilizan solo funciones de base radial como pueden ser la función Gaussiana o la Inversa Cuadrática. Para el aprendizaje de las redes de tipo Recurrente considerando su comportamiento dinámico y temporal se puede utilizar el método de aprendizaje Recurrente en Tiempo Real, en este caso comúnmente también se consideran las funciones de activación Sigmoidal o Tangente Hiperbólica, pues estos no utilizan funciones de tipo base radial.

En base a la distribución antes mencionada se desarrolló para cada dos redes neuronales artificiales un solo algoritmo de red neuronal artificial, es decir un algoritmo para cada tipo, pues solo presentan una única diferencia en la función de activación utilizada, la cual debe ser seleccionada cuando se ejecuta el algoritmo.

4.2.1. Diseño de clases generales para los algoritmos de redes neuronales artificiales

4.2.1.1. Identificación de Clases

Lo primero que se realizó en esta sección fue la identificación y el desarrollo de cinco clases generales, que vienen a ser las principales entidades que forman parte de la arquitectura y el procesamiento de las redes neuronales artificiales, las cuales los algoritmos desarrollados incluyen independientemente de su tipo de red y método de aprendizaje.

A continuación se explica las clases identificadas, algunos de sus atributos y métodos más importantes.

- a) Estudiante.- Esta clase representa los patrones de entrada, es decir la información de cada estudiante, los atributos que tiene son:
- Atributos
 - Código.- Identificador del estudiante.
 - Nombre.- Nombre completo del estudiante.
 - Edad.- Edad del estudiante.



- Indicadores.- Son los 11 indicadores de convivencia del estudiante.
 - Situación real.- Es la situación real del estudiante respecto a la existencia de violencia escolar.
 - Indicador de agrupación.- Solo en caso de los algoritmos de tipo base radial es el número de la agrupación a la que pertenece el patrón.
- b) Base de datos.- Esta clase permite la conexión con la base de datos y la interacción de los algoritmos de redes neuronales artificiales con la misma, los atributos y métodos de esta clase son:
- Atributos
 - Texto de consulta.- consulta SQL.
 - Mensaje de error.- Mensaje de error.
 - Arreglo de estudiantes.- Es la data recibida de la consulta realizada para conseguir los datos de los estudiantes.
 - Métodos
 - Conexión a la base de datos.- Permite abrir la conexión con la base de datos.
 - Conseguir datos de estudiantes de la base de datos.- Permite recibir la data de la entidad estudiante de la base de datos.
 - Insertar resultados a la base de datos.- Permite insertar los resultados del procesamiento de los algoritmos en la base de datos.
- c) Neurona.- Esta clase es la entidad que representa las neuronas artificiales de una red, sus atributos y métodos son:
- Atributos
 - Capa.- Indica la capa a la que pertenece la neurona.
 - Tipo de función.- Tipo de función de activación seleccionada para la neurona.
 - Entrada efectiva.- La sumatoria del producto de las entradas de una neurona con sus respectivos valores de pesos sinápticos.
 - Bias.- El umbral de la neurona.
 - Resultado de la función de activación: Resultado del procesamiento de la función de activación en la neurona.
 - Resultado de la derivada de la función de activación.- Resultado de la derivada de la función de activación, este atributo solo es necesario en



los tipos perceptrón multicapa y red recurrente por sus aprendizajes de retropropagación y retropropagación en tiempo real respectivamente.

- Coeficiente Jaccard.- Solo para los algoritmos de tipo base radial, es el valor de disimilitud entre un patrón de entrada y el centro de la neurona de la agrupación a la que pertenece.
- Amplitud.- Es el valor de distancia entre el centro de la neurona de la agrupación a la que pertenece un patrón entre sus dos centros más cercanos.

- Métodos

- Conseguir resultado del procesamiento de función de activación.- Solicita a la clase función de activación el resultado del procesamiento de la entrada efectiva en base a la función seleccionada.
- Conseguir resultado del procesamiento de la derivada de la función de activación.- Solicita a la clase función de activación el resultado del procesamiento de la entrada efectiva en base a la derivada de la función seleccionada.
- Actualizar sus bias.- Este método permite la actualización de los umbrales de las neuronas artificiales, en el proceso de aprendizaje de la red.

d) Conexión.- Esta clase representa las conexiones entre las neuronas (sinapsis). Los atributos y métodos más importantes son:

- Atributos

- Valor del peso sináptico.- Es el valor numérico de la conexión entre una neurona y otra, este valor cambia durante el proceso de aprendizaje de la red.

- Métodos

- Actualizar valor del peso sináptico.- Actualiza el valor de la conexión, en función al error calculado y a la razón de aprendizaje (este último que influye en la velocidad de aprendizaje de la red neuronal).

e) Función de activación.- Esta clase representa las funciones de activación en las neuronas. Sus atributos y métodos son los siguientes:

- Atributos

- Tipo de función.- Es el tipo de función de activación.



- Entrada efectiva.- Es la entrada efectiva que se debe procesar en la función de activación, solo en tipos de algoritmo Perceptrón multicapa y Red recurrente.
- Coeficiente Jaccard.- Es el valor de disimilitud entre dos objetos binarios, solo en casos de algoritmos de tipo Red base radial.
- Amplitud.- Es el valor de amplitud entre dos centros de neuronas, también solo en casos de algoritmos de tipo Red de base radial.
- Métodos
 - Procesar la función de activación.- Procesa la función de activación con los valores establecidos; si es un algoritmo de tipo perceptrón multicapa o red recurrente procesa la entrada efectiva. En caso de base radial la división entre el coeficiente jaccard y la amplitud del centro de la neurona.
 - Procesar la derivada de la función de activación.- Procesa la derivada de la función de activación.

4.2.1.2. Diagrama de las clases identificadas

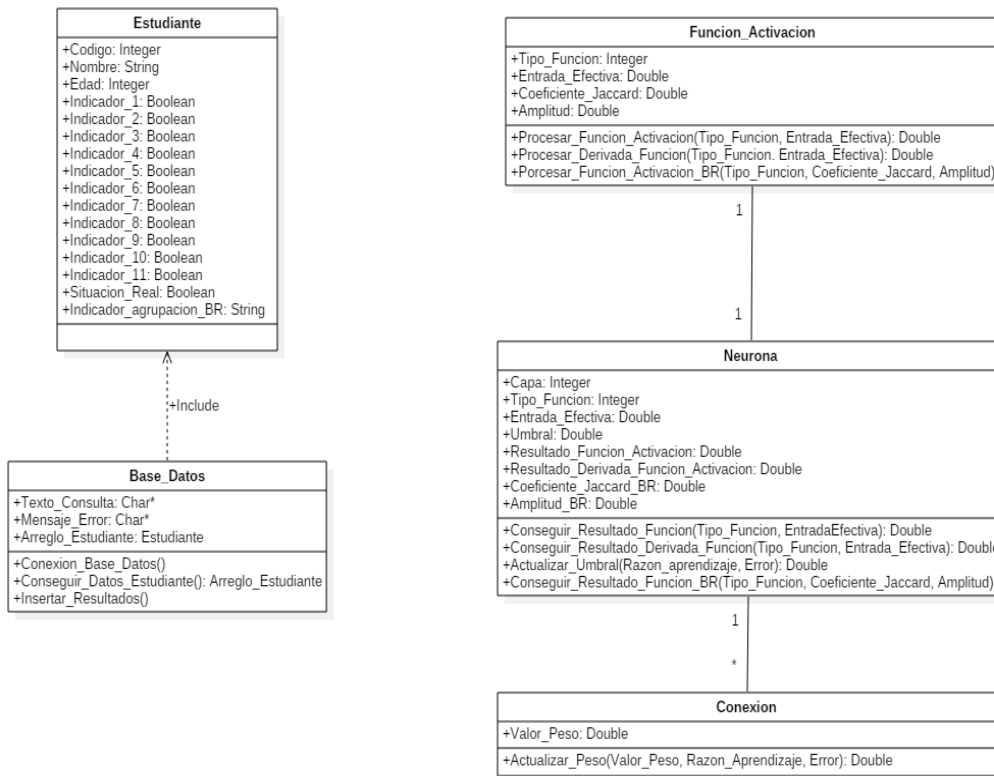


Ilustración 25: Diagrama de las clases identificadas

Fuente: propia

En la *ilustración 25* se puede observar las clases, sus atributos y métodos de cada una de ellas y la relación entre ellas. La base de datos incluye el conjunto de datos Estudiante que fue detallado en el capítulo anterior. La clase neurona tiene una función de activación, además las neuronas deben tener al menos una y en la mayoría de casos pueden tener muchas conexiones (sinapsis), esto depende del tipo de red neuronal artificial.

4.2.2. Desarrollo de los algoritmos de redes neuronales artificiales

a) *Tipo 1: Perceptrón Multicapa con método de aprendizaje de Retropropagación (PM)*

Las redes neuronales artificiales PMS y PMH son redes de tipo perceptrón multicapa y utilizan el método de aprendizaje de Retropropagación, la única diferencia entre estos dos, es que para PMS se utiliza la función de activación Sigmoidal y para PMH la función Tangente Hiperbólica. Para el desarrollo del algoritmo de este tipo de red neuronal artificial se consideraron las siguientes fases:

- Fase I: Construcción de la arquitectura de la red
 - Seleccionar el tipo de función de activación sigmoideal (PMS) o Tangente hiperbólica (PMH).
 - Establecer el número de capas ocultas.
 - Establecer el número de neuronas en cada capa oculta.
- Fase II: Etapa de aprendizaje y validación
 - Establecer el número de ciclos de aprendizaje.
 - Establecer razón de aprendizaje
 - Se divide en seis bloques los patrones de entrada de acuerdo a la teoría de validación cruzada.
 - Se realiza el proceso de aprendizaje con el método de Retropropagación.
 - Se realiza la validación del algoritmo.

b) *Tipo 2: Red Neuronal de Base Radial con método de aprendizaje Híbrido (BR)*

En este caso las redes neuronales artificiales BRG y BRC son redes de tipo base radial y utilizan el método de aprendizaje híbrido. La diferencia entre estos dos es la función de activación, pues el BRG utiliza la función Gaussiana, mientras BRC la función Inversa Cuadrática.

En este tipo de red y método de aprendizaje surgieron dos situaciones que serán explicadas a continuación:

I. *Situación I*

Como se mencionó en el capítulo II en la sección de redes de base radial, el método de aprendizaje híbrido tiene dos fases no supervisada y supervisada,

la primera situación surgió en la fase no supervisada, pues los datos de entrada a las redes neuronales artificiales en la presente investigación son de tipo categóricos, en ese sentido utilizar K-medias no es óptimo para la naturaleza de estos datos, pues según Zhexue Huang:

“El algoritmo K-medias es bien conocido por su eficiencia en la agrupación de grandes conjuntos de datos. Sin embargo, trabajar solo en datos numéricos limita el uso de este algoritmo en áreas tales como minería de datos donde se encuentra con frecuencia grandes conjuntos de datos categóricos...” (Huang, 2003)

Para solucionar este incidente se utilizó una extensión del algoritmo K-medias denominado K-modas en su lugar, este también fue profundizado en el marco teórico de esta investigación.

II. Situación II

Los datos de entrada son categóricos, y son de tipo binarios, por lo que una medida de distancia euclidiana no podría ser la correcta para la naturaleza de estos datos, es por esta razón que fue remplazado por el Índice Jaccard que es un coeficiente de similitud para objetos binarios.

Por lo tanto la diferencia euclidiana de la *ecuación 22*, que es donde se calcula la distancia entre el vector de entrada y el centro de la neurona fue remplazada por la ecuación de disimilitud Jaccard que está definida en la *ecuación 30*. En base a esto la *ecuación 22* sería modificada de la siguiente forma:

$$||X(n) - C_i|| = d(X(n), C_i) \quad (61)$$

Donde:

$d(X(n), C_i)$: Es la disimilitud Jaccard

$X(n)$: Es el vector de entrada del patrón n

C_i : Centro de la neurona i de la capa oculta de la red

En las redes de base radial también se debe calcular la distancia euclidiana de dos objetos para el cálculo de las amplitudes de los centros de las neuronas de la capa oculta definida en la *ecuación 26*. En esta ecuación también se reemplazó esta diferencia euclidiana por la ecuación de disimilitud Jaccard. Por lo tanto la ecuación fue modificada de la siguiente manera:

$$d_i = \sqrt{d(C_i, C_p) d(C_i, C_s)} \quad (62)$$

Donde:

C_i : Es el centro de la neurona i de la capa oculta de la red

C_p Y C_s : Son los centros de las neuronas más cercanas a la neurona i

$d(C_i, C_p)$ Y $d(C_i, C_s)$: Son las funciones de disimilitud Jaccard

En cuanto la fase supervisada se utilizó mínimos cuadrados que es un método que normalmente se usa en el método de aprendizaje híbrido y sin mayores inconvenientes hallados en su proceso.

Por lo tanto para el desarrollo del algoritmo de este tipo de red neuronal artificial se consideraron las siguientes fases:

- Fase I: Construcción de la arquitectura de la red
 - Seleccionar el tipo de función de activación Gaussiana (*BRG*) o Inversa Cuadrática (*BRC*).
 - Establecer el número de neuronas para la capa oculta, cabe recordar que las redes de base radial solo poseen una sola capa como se explicó en el marco teórico, además cada neurona tiene un centro, y es igual a una clase de agrupación.
- Fase II: Etapa de aprendizaje y validación
 - Establecer el número de ciclos de aprendizaje.
 - Establecer razón de aprendizaje
 - Se divide en seis bloques los patrones de entrada de acuerdo a la teoría de validación cruzada.
 - Se realiza el proceso de aprendizaje no supervisado con k-modas.



- Se realiza el proceso de aprendizaje supervisado con mínimos cuadrados.
- Se realiza la validación del algoritmo.

c) Tipo 3: Red Neuronal Recurrente con método de aprendizaje Recurrente en Tiempo Real (RR)

Las redes neuronales artificiales RRS y RRH son redes de tipo recurrente y utilizan el método de aprendizaje de retropropagación en tiempo real, solo se diferencian en la función de activación utilizada. RRS utiliza la función Sigmoidal y RRH la función Tangente hiperbólica.

Para el desarrollo del algoritmo de este tipo de red neuronal artificial se consideraron las siguientes fases:

- Fase I: Construcción de la arquitectura de la red
 - Seleccionar el tipo de función de activación sigmoidal (RRS) o hiperbólica (RRH).
 - Establecer el número de neuronas recurrentes.
 - Se suman al número de neuronas recurrentes establecidas las neuronas de entrada que son 11 y una neurona de salida que por defecto ya están establecidas en el algoritmo.
- Fase II: Etapa de aprendizaje y validación
 - Establecer el número de tiempos de aprendizaje.
 - Establecer la razón de aprendizaje
 - Se divide en seis bloques los patrones de entrada de acuerdo a la teoría de validación cruzada.
 - Realizar el proceso de aprendizaje en base al método Recurrente en Tiempo Real.
 - Realizar la validación del algoritmo

4.3. Etapa 3: Experimentación de las redes neuronales artificiales

En esta etapa se realizó la experimentación de las redes neuronas artificiales, con el propósito de conocer sus mayores niveles de eficiencia alcanzados en la identificación de violencia escolar. Para la experimentación de cada una de las redes neuronales artificiales se utilizó uno de los algoritmos desarrollados en la fase anterior, esto de acuerdo al tipo al que pertenecen.

Primeramente se realizaron varios procesos de prueba y error, estos procesos sirvieron para conocer la razón de aprendizaje, así como el número de neuronas con los que las redes neuronales artificiales lograban alcanzar mejores resultados. En la siguiente tabla se muestra un resumen de dichos procesos:

Tabla 11: Procesos de Prueba y Error

Tipo	Red neuronal Artificial	Razón de aprendizaje	Numero de neuronas y/o Numero de capas	Observaciones
Tipo 1	PMS Y PMH	Se inicializó en 0,00001 y se incrementó en 0,00001 por cada ciclo	De 1 a 22 capas y de 1 a 22 neuronas en cada capa	No se obtenían resultados aceptables, la red neuronal no se estabilizaba
			Se utilizó una sola capa con números de neuronas mayores iguales a 100 neuronas	En este proceso se descubrió que con un número muy alto de neuronas y con una sola capa oculta se lograba obtener resultados óptimos, se encontraron los mejores resultados utilizando un rango de 300 a 600 neuronas
Tipo 2	BRG Y BRC	Se inicializó en 0,0001 y se incrementó en 0,0001 por cada ciclo	De 1 a 20 neuronas	Se pudo observar que en este rango se lograba obtener resultados aceptables, pero que un rango de 11 a 14 neuronas se encontraba los mejores resultados
Tipo 3	RRS	0,001 y no se incrementó	De 2 a 20 neuronas	Se pudo observar que se lograban resultados aceptables pero fue un rango de 9 a 12 neuronas los que ofrecían mejores resultados
	RRH	0,001 y no se incrementó	De 2 a 20 neuronas	La red neuronal artificial de este tipo no se lograba estabilizar con esta razón de aprendizaje
		0,000000001 y no se incrementa	De 2 a 20 neuronas	La red neuronal artificial logro estabilizarse pero los resultados no eran demasiados óptimos, de todas maneras se encontró mejores resultados utilizando un rango de 9 a 12 neuronas

Nota. Procesos de prueba y error, con estos procesos se pudo conocer los indicadores con los que las redes neuronales artificiales lograban alcanzar mejores resultados. Fuente: Propia



Gracias a los procesos de prueba y error detallados en la *tabla 11* se logró encontrar los rangos de número de neuronas que alcanzaban mejores resultados en cada red neuronal artificial. En base a ello se estableció cuatro escenarios para la experimentación de cada red neuronal artificial, cada escenario con un diferente número de neuronas. En la siguiente tabla se puede apreciar de mejor manera la distribución de las experimentaciones de los escenarios de las redes neuronales artificiales (RNA):

Tabla 12: Tabla del proceso de experimentación de las redes neuronales artificiales

Tipo	Números de ciclos / Instantes de tiempo	Razón de aprendizaje	RNA	Función de activación	Escenario	Numero de neuronas	Bloques de validación y Puntos Óptimos								
							B1	B2	B3	B4	B5	B6			
Tipo 1: Perceptrón Multicapa	1000	Inicia en 0,00001 y se incrementa en 0,00001 por cada ciclo	PMS	Sigmoidal	PMSI	300	PO1	PO2	PO3	PO4	PO5	PO6			
					PMSII	400	B1	B2	B3	B4	B5	B6			
					PMSIII	500	PO1	PO2	PO3	PO4	PO5	PO6			
					PMSIV	600	B1	B2	B3	B4	B5	B6			
					PO1	PO2	PO3	PO4	PO5	PO6					
					PMH	Tangente Hiperbólica	PMHI	300	B1	B2	B3	B4	B5	B6	
					PMHII		400	PO1	PO2	PO3	PO4	PO5	PO6		
					PMHIII		500	B1	B2	B3	B4	B5	B6		
			PMHIV	600	PO1		PO2	PO3	PO4	PO5	PO6				
			Tipo 2: Red Neuronal de Base Radial	1000	Inicia en 0,0001 y se incrementa en 0,0001 por cada ciclo	BRG	Gaussiana	BRGI	11	B1	B2	B3	B4	B5	B6
								BRGII	12	PO1	PO2	PO3	PO4	PO5	PO6
								BRGIII	13	B1	B2	B3	B4	B5	B6
BRGIV	14	PO1						PO2	PO3	PO4	PO5	PO6			
BRC	Inversa cuadrática	BRCI						11	B1	B2	B3	B4	B5	B6	
BRCII		12						PO1	PO2	PO3	PO4	PO5	PO6		
BRCIII		13						B1	B2	B3	B4	B5	B6		
PO1		PO2						PO3	PO4	PO5	PO6				



					BRCIV	14	B1	B2	B3	B4	B5	B6
							PO1	PO2	PO3	PO4	PO5	PO6
Tipo 3: Red Neuronal Recurrente	100000	0,001 y no se incrementa	RRS	Sigmoideal	RRSI	9	B1	B2	B3	B4	B5	B6
						PO1	PO2	PO3	PO4	PO5	PO6	
					RRSII	10	B1	B2	B3	B4	B5	B6
						PO1	PO2	PO3	PO4	PO5	PO6	
					RRSIII	11	B1	B2	B3	B4	B5	B6
						PO1	PO2	PO3	PO4	PO5	PO6	
		RRSIV	12	B1	B2	B3	B4	B5	B6			
			PO1	PO2	PO3	PO4	PO5	PO6				
		0,000000001 y no se incrementa	RRH	Tangente Hiperbólica	RRHI	9	B1	B2	B3	B4	B5	B6
					PO1	PO2	PO3	PO4	PO5	PO6		
	RRHII				10	B1	B2	B3	B4	B5	B6	
					PO1	PO2	PO3	PO4	PO5	PO6		
RRHIII	11				B1	B2	B3	B4	B5	B6		
	PO1				PO2	PO3	PO4	PO5	PO6			
	RRHIV	12	B1	B2	B3	B4	B5	B6				
				PO1	PO2	PO3	PO4	PO5	PO6			

Nota. Esta tabla detalla las experimentaciones y los escenarios de cada red neuronal artificial que se debían realizar en esta sección. Fuente: Propia

Como se observa en la *tabla 12* se detalla el número de neuronas utilizadas en cada uno de los escenarios de cada experimentación. En cada escenario se obtuvo los resultados en seis distintos bloques de validación, que se obtuvieron en base a la validación cruzada explicada en el marco teórico, y en cada bloque de validación se utilizó un gráfico de líneas que representa la evolución de los errores de aprendizaje y validación a lo largo de los ciclos de aprendizaje o tiempos de aprendizaje, y en cada bloque (B) también se calculó el punto óptimo (PO) que es el menor error cuadrático medio cometido por la red neuronal artificial utilizando el bloque de datos actual. Luego se seleccionó para cada escenario el bloque que alcanzó un menor error cuadrático medio en su punto óptimo y finalmente para cada experimentación se escogió el mejor escenario.



4.3.1. Experimentación de la Red neuronal artificial PMS

Los parámetros que se utilizaron en la experimentación de PMS son los siguientes:

- Tipo de red: Perceptrón multicapa
- Método de aprendizaje: Retropropagación
- Función de activación: Sigmoidal
- Numero de ciclos máximo: 1000 ciclos
- Razón de aprendizaje: se inicializo en 0,00001 y se incrementa en 0,00001 por cada ciclo

Para esta experimentación se realizó cuatro escenarios con un número de neuronas diferente en cada uno de ellos, y en todos ellos una sola capa oculta.

Algo que se repitió en los resultados de los cuatro escenarios experimentados con PMS fue que en los segundos bloques de validación se encontraron los mejores puntos óptimos con menores errores cuadráticos medios en validación entre los ciclos de aprendizaje 232 al 540.

a) *Escenario PMSI*

En la *ilustración 26* se observa los resultados de la experimentación de la red neuronal artificial PMS con 300 neuronas en una sola capa oculta, donde en el segundo bloque se obtuvo el mejor punto óptimo que tuvo el menor error cuadrático medio, con un valor de 0,114821 en el ciclo 311.

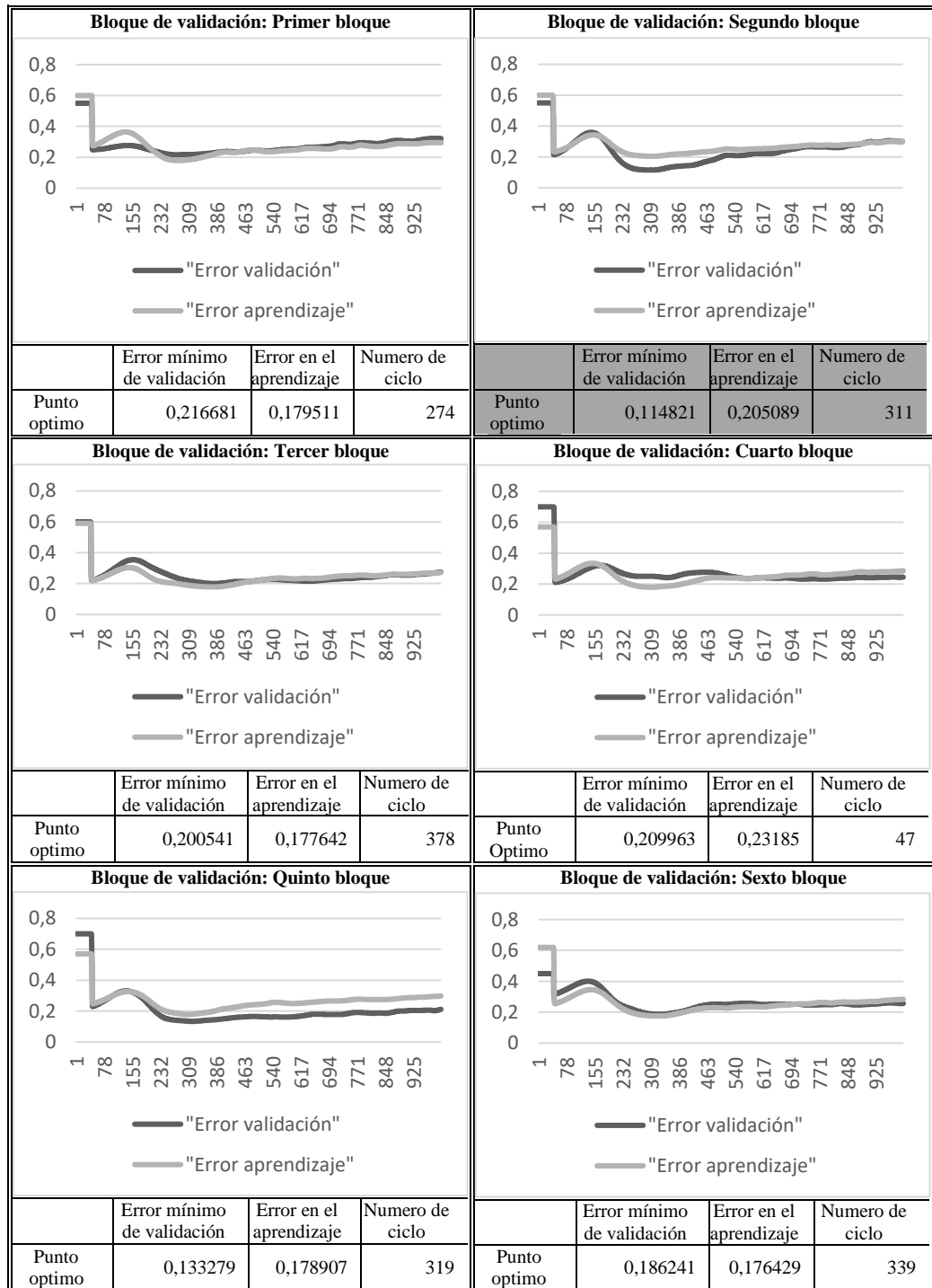


Ilustración 26: Resultados del escenario PMSI

Fuente: Propia

b) Escenario PMSII

En la *ilustración 27* se observa los resultados de la experimentación de la red neuronal artificial PMS con 400 neuronas en una sola capa oculta, donde en el segundo bloque se obtuvo el mejor punto óptimo que tuvo el menor error cuadrático medio, con un valor de 0,10395 en el ciclo 310.

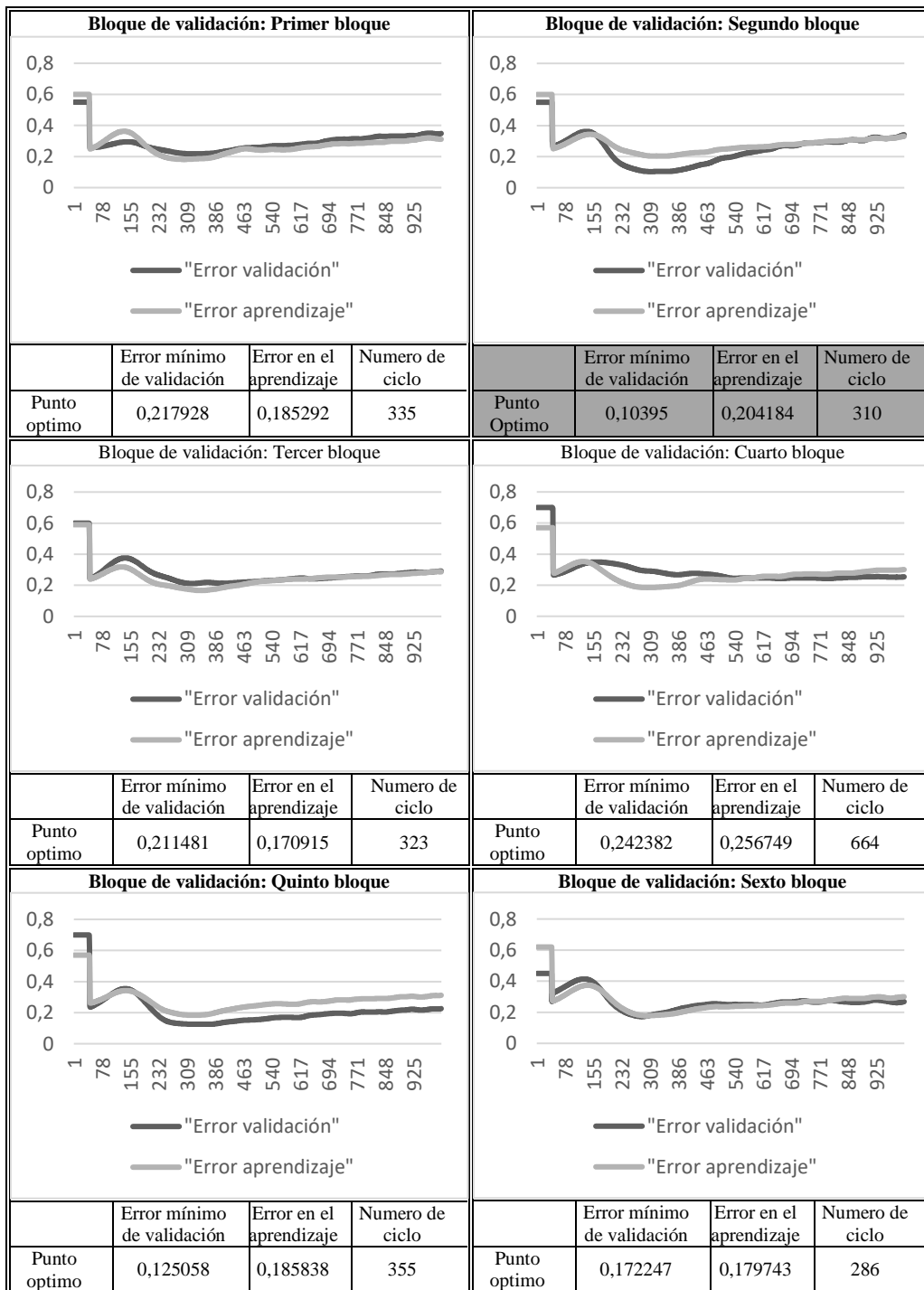


Ilustración 27: Resultados del escenario PMSII

Fuente: propia

c) *Escenario PMSIII*

En la *ilustración 28* se observa los resultados de la experimentación de la red neuronal artificial PMS con 500 neuronas en una sola capa oculta, donde en el segundo bloque se obtuvo el mejor punto óptimo que tuvo el menor error cuadrático medio, con un valor de 0,084697 en el ciclo 278.

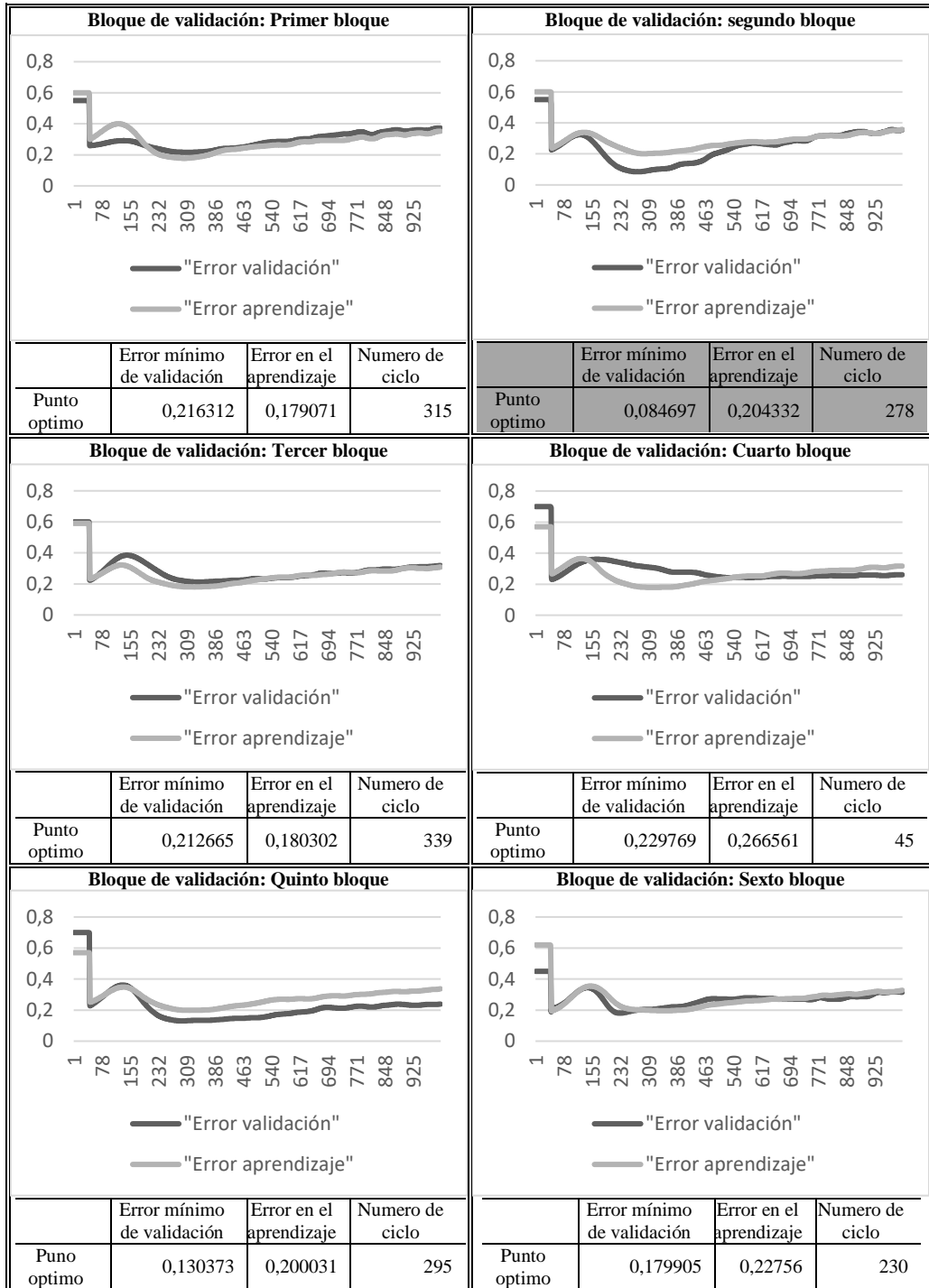


Ilustración 28: Resultados del escenario PMSIII

Fuente: propia

d) Escenario PMSIV

En la *ilustración 29* se observa los resultados de la experimentación de la red neuronal artificial PMS con 600 neuronas en una sola capa oculta, donde en el segundo bloque se obtuvo el mejor punto óptimo que tuvo el menor error cuadrático medio, con un valor de 0,105038 en el ciclo 287.

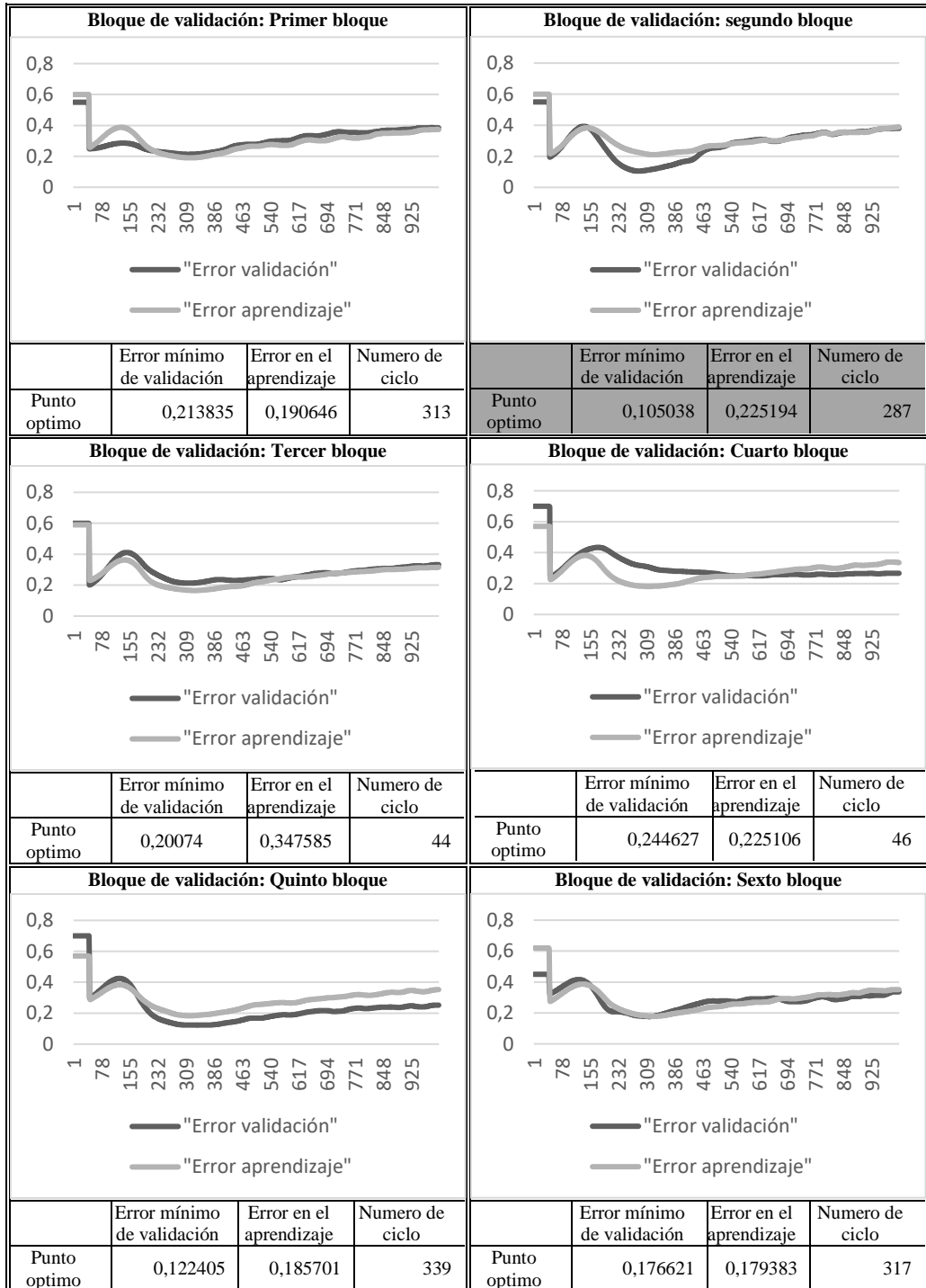


Ilustración 29: Resultados del escenario PMSIV

Fuente: propia

e) **Comparación de los escenarios de la experimentación de la red neuronal artificial PMS**

En cada escenario se seleccionó el bloque de validación que alcanzó el mejor punto óptimo.

Tabla 13: **Tabla comparativa de los escenarios realizados con PMS**

Tipo de red	Método de aprendizaje	Función	Razón de aprendizaje	Escenario	Numero de neuronas	Ciclo	Error mínimo de validación (ECM)
PM	Retropropagación	Sigmoïdal	0,00001	PMSI	300	311	0,114821
				PMSII	400	310	0,10395
				PMSIII	500	278	0,084697
				PMSIV	600	287	0,105038

Nota. Tabla comparativa de los resultados de los cuatro escenarios de la experimentación de la red PMS. Fuente: propia

En la *tabla 13* se observan los cuatro escenarios realizados, en la presente experimentación fue en el escenario PMSIII con 500 neuronas y en el ciclo de aprendizaje 278 donde se encontró el menor error cuadrático medio, con un valor de 0,084697, lo que equivale a un 91,54% de nivel de eficiencia en la identificación de violencia escolar.

4.3.2. Experimentación de la red neuronal artificial PMH

Los parámetros que se utilizaron en la experimentación de esta red son los siguientes:

- Tipo de red: Perceptrón Multicapa
- Método de aprendizaje: Retropropagación
- Función de activación: Tangente hiperbólica
- Numero de ciclos máximo: 1000 ciclos
- Razón de aprendizaje: se inicializo en 0,00001 y se incrementa en 0,00001 por cada ciclo

En los resultados de los cuatro escenarios experimentados en PMH la evolución de los errores tanto de aprendizaje, como de validación variaban en cortos periodos de ciclos de aprendizaje y en todos estos se encontró mejores resultados en los segundos bloques de validación.

a) Escenario PMHI

En la *ilustración 30* se observa los resultados de la experimentación de la red neuronal artificial PMH con 300 neuronas en una sola capa oculta, donde en el segundo bloque se obtuvo el mejor punto óptimo que tuvo el menor error cuadrático medio, con un valor de 0,101357 en el ciclo 259.

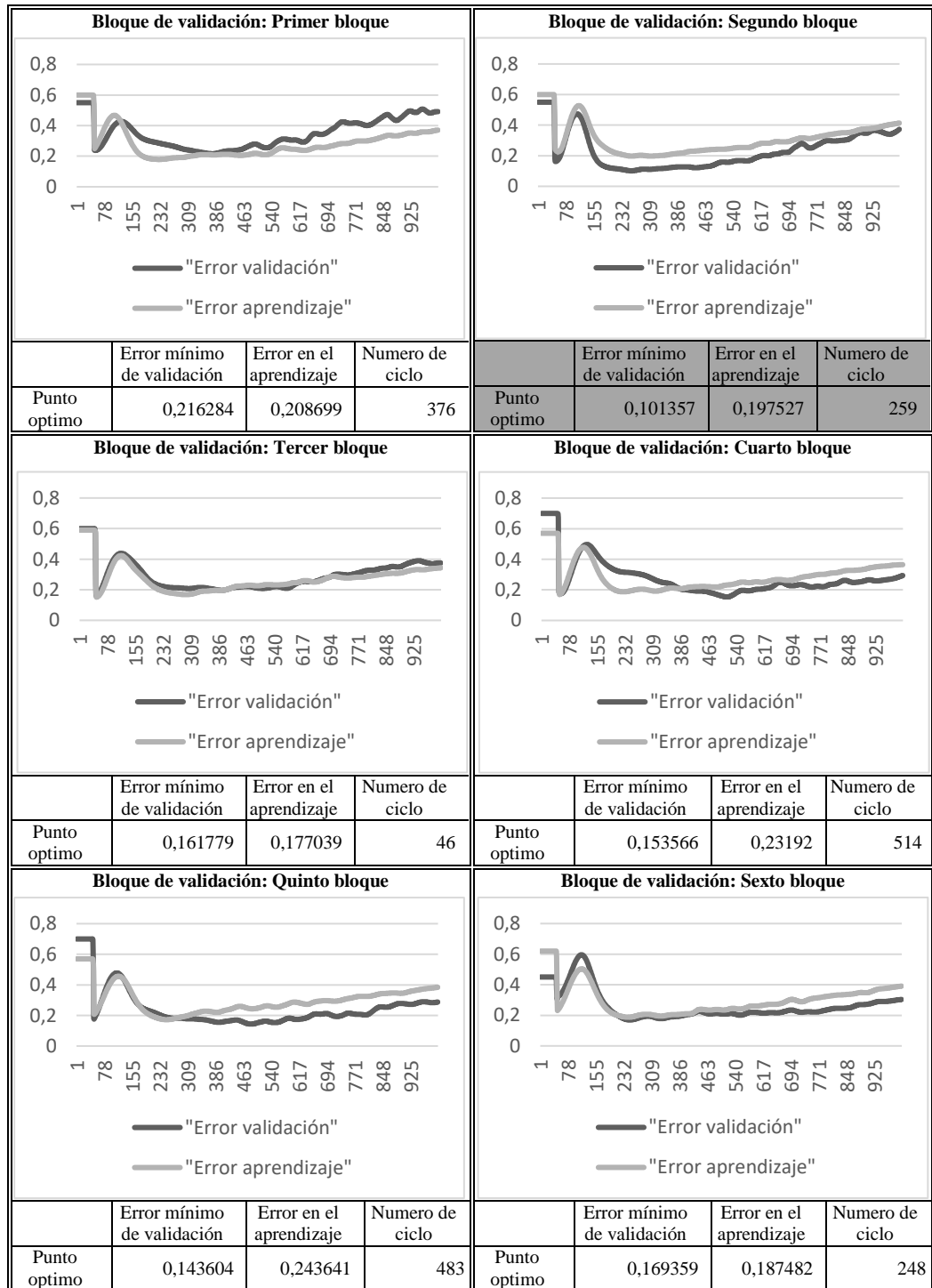


Ilustración 30: Resultados del escenario PMHI

Fuente: propia

b) Escenario PMHII

En la *ilustración 31* se observa los resultados de la experimentación de la red neuronal artificial PMH con 400 neuronas en una sola capa oculta, donde en el segundo bloque se obtuvo el mejor punto óptimo que tuvo el menor error cuadrático medio, con un valor de 0,085021 en el ciclo 325.

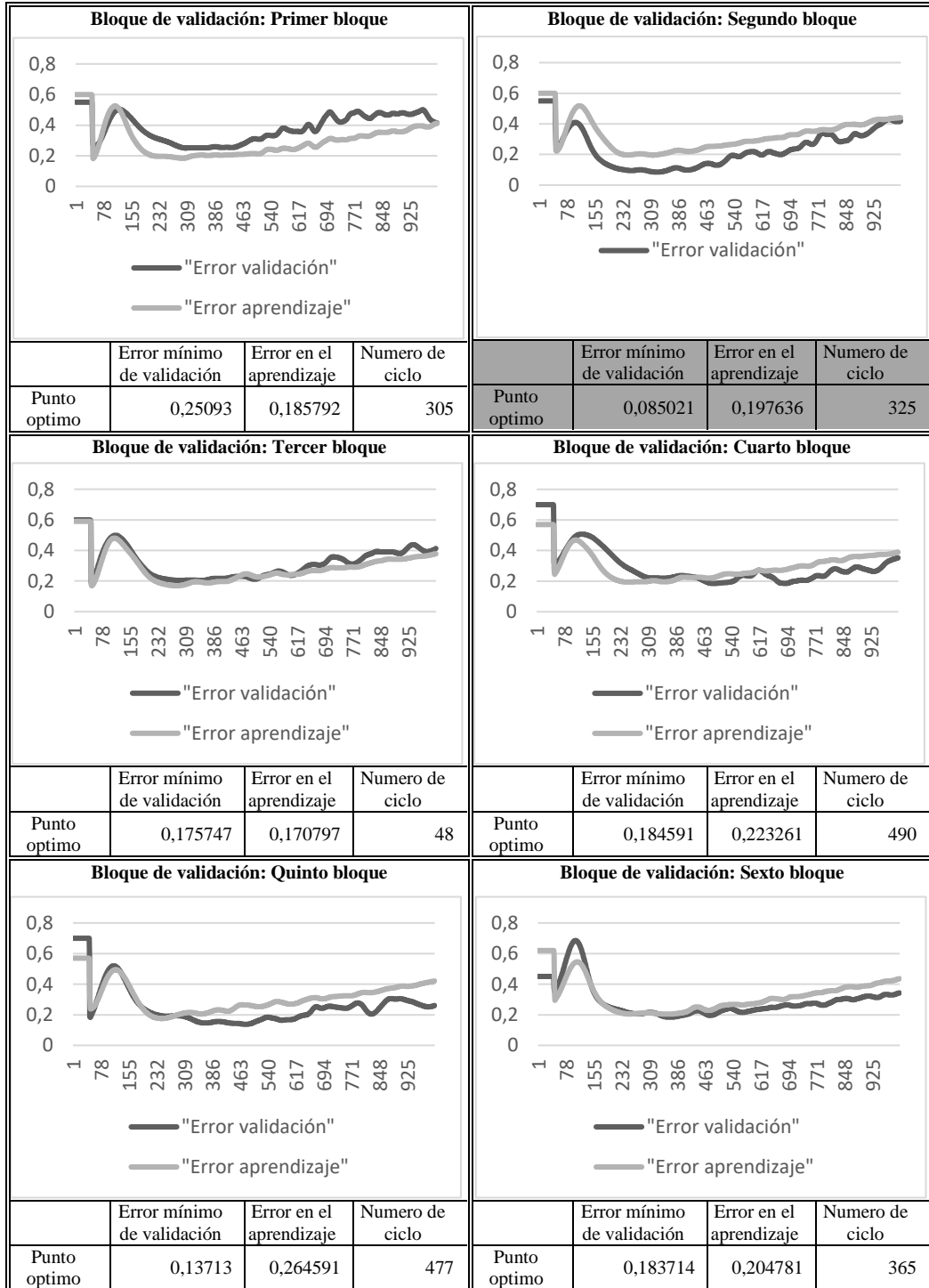


Ilustración 31: Resultados del escenario PMHII

Fuente: propia

c) **Escenario PMHIII**

En la *ilustración 32* se observa los resultados de la experimentación de la red neuronal artificial PMH con 500 neuronas en una sola capa oculta, donde en el segundo bloque se obtuvo el mejor punto óptimo que tuvo el menor error cuadrático medio, con un valor de 0,095131 en el ciclo 382.

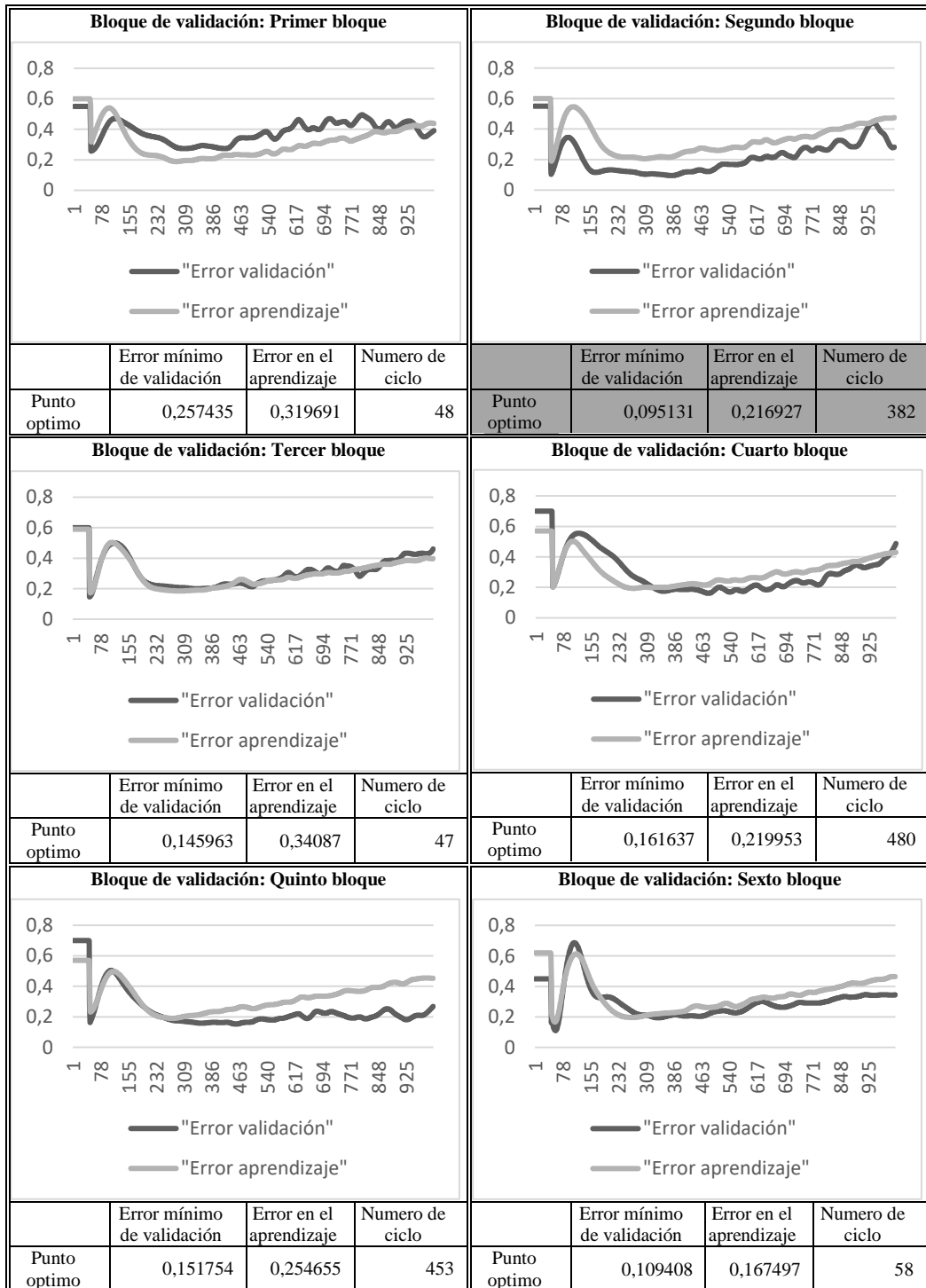


Ilustración 32: Resultados del escenario PMHIII

Fuente: propia

d) Escenario PMHIV

En la *ilustración 33* se observa los resultados de la experimentación de la red neuronal artificial PMH con 600 neuronas en una sola capa oculta, donde en el segundo bloque se obtuvo el mejor punto óptimo que tuvo el menor error cuadrático medio, con un valor de 0,096121 en el ciclo 306.

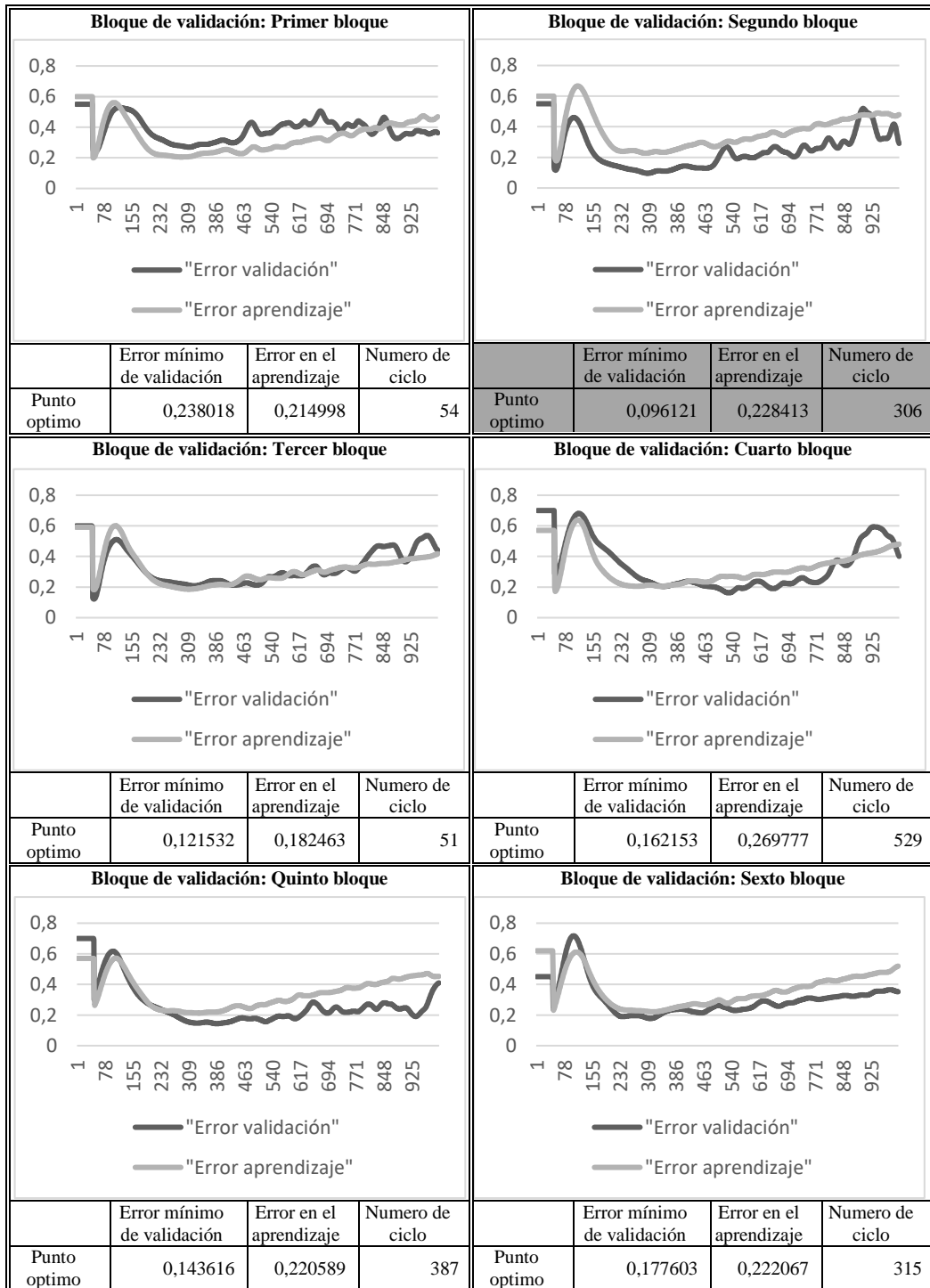


Ilustración 33: Resultados del escenario PMHIV

Fuente: propia

e) **Comparación de los escenarios de la experimentación de la red neuronal artificial PMH**

En cada escenario se seleccionó el bloque de validación que alcanzó el mejor punto óptimo.

Tabla 14: **Tabla comparativa de los escenarios realizados con PMH**

Tipo de red	Método de aprendizaje	Función	Razón de aprendizaje	Escenario	Numero de neuronas	ciclo	Error mínimo de validación (ECM)
PM	Retropropagación	Tangente hiperbólica	0,00001	PMHI	300	259	0,101357
				PMHII	400	325	0,085021
				PMHIII	500	382	0,095131
				PMHIV	600	306	0,096121

Nota. Tabla comparativa de los resultados de los cuatro escenarios de la experimentación de la red PMH. Fuente: propia

En la *tabla 14* se observan los cuatro escenarios realizados, para esta experimentación fue en el escenario PMHII con 400 neuronas y en el ciclo de aprendizaje 325 donde se encontró el menor error cuadrático medio con un valor de 0,085021, lo que equivale a un 91,49% de nivel de eficiencia en la identificación de violencia escolar.

4.3.3. Experimentación de la red neuronal artificial BRG

Los parámetros que se utilizaron en la experimentación de esta red son los siguientes:

- Tipo de red: Red Neuronal de Base Radial
- Método de aprendizaje: Híbrido
- Función de activación: Gaussiana
- Numero de ciclos máximo: 1000 ciclos
- Razón de aprendizaje: se inicializo en 0,0001 y se incrementa en 0,0001 por cada ciclo

En la experimentación de los cuatro escenarios de BRG los segundos bloques de validación fueron los seleccionados, pues entre los ciclos 78 al 309 alcanzaron los mejores puntos óptimos en comparación a los demás bloques.

a) Escenario BRGI

En la *ilustración 34* se observa los resultados de la experimentación de la red neuronal artificial BRG con 11 neuronas en la capa oculta, donde en el segundo bloque se obtuvo el mejor punto óptimo que tuvo el menor error cuadrático medio, con un valor de 0,076314 en el ciclo 148.

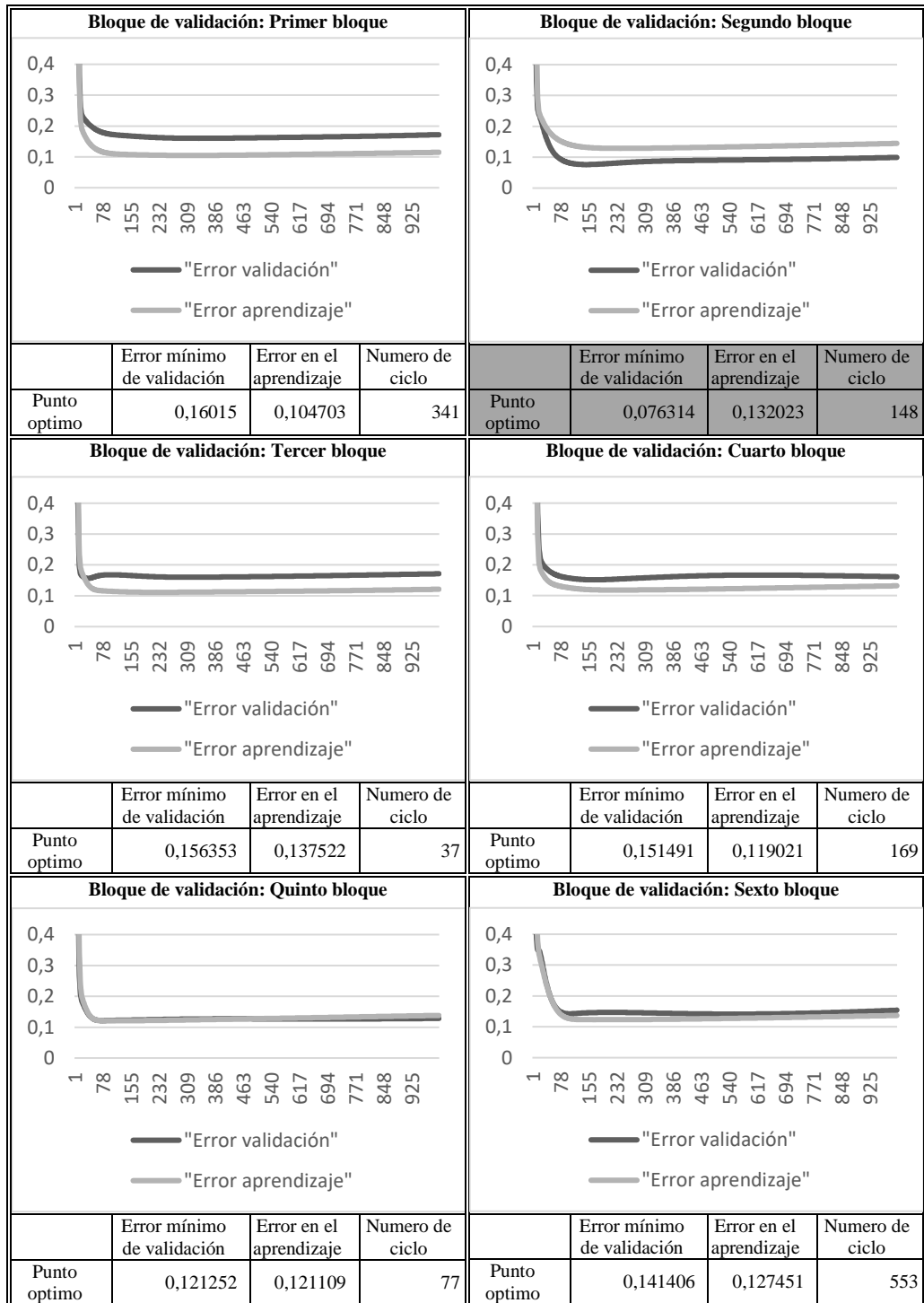


Ilustración 34: Resultados del escenario BRGI

Fuente: propia

b) Escenario BRGII

En la *ilustración 35* se observa los resultados de la experimentación de la red neuronal artificial BRG con 12 neuronas en la capa oculta, donde en el segundo bloque se obtuvo el mejor punto óptimo que tuvo el menor error cuadrático medio, con un valor de 0,052252 en el ciclo 88.

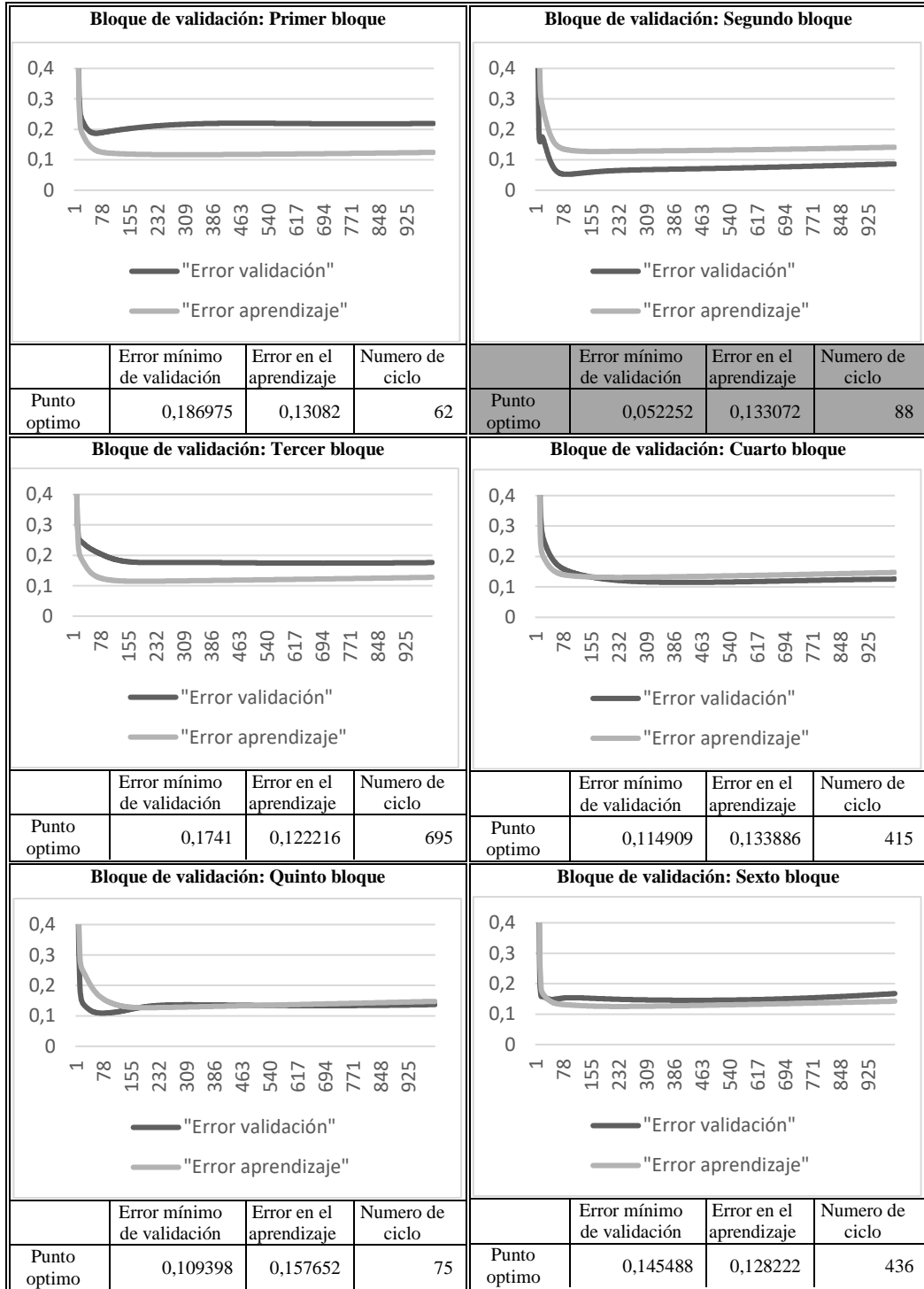


Ilustración 35: Resultados del escenario BRGII

Fuente: propia

c) **Escenario BRGIII**

En la *ilustración 36* se observa los resultados de la experimentación de la red neuronal artificial BRG con 13 neuronas en la capa oculta, donde en el segundo bloque se obtuvo el mejor punto óptimo que tuvo el menor error cuadrático medio, con un valor de 0,078941 en el ciclo 136.

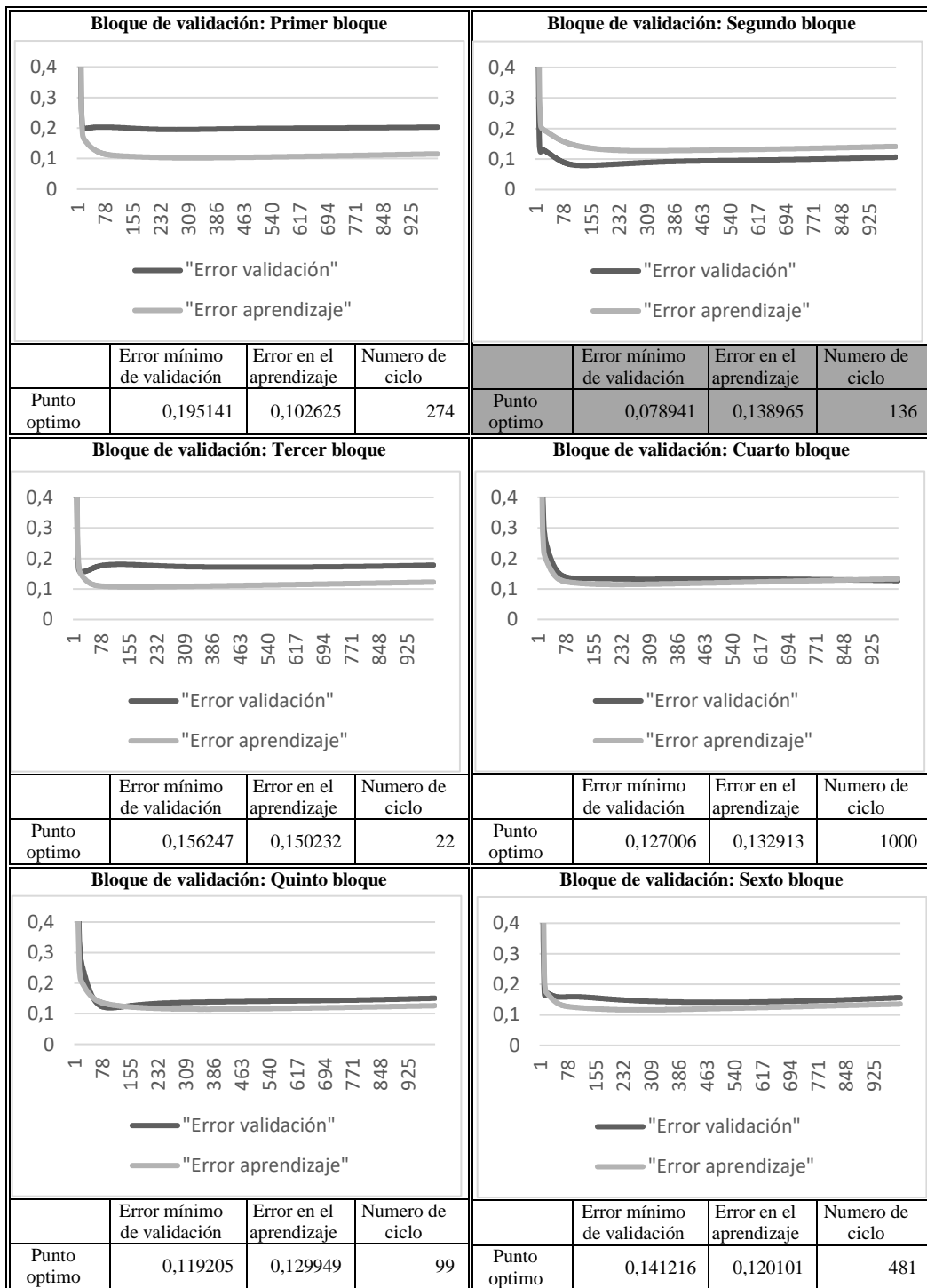


Ilustración 36: Resultados del escenario BRGIII

Fuente: propia

d) Escenario BRGIV

En la *ilustración 37* se observa los resultados de la experimentación de la red neuronal artificial BRG con 14 neuronas en la capa oculta, donde en el segundo bloque se obtuvo el mejor punto óptimo que tuvo el menor error cuadrático medio, con un valor de 0,079409 en el ciclo 234.

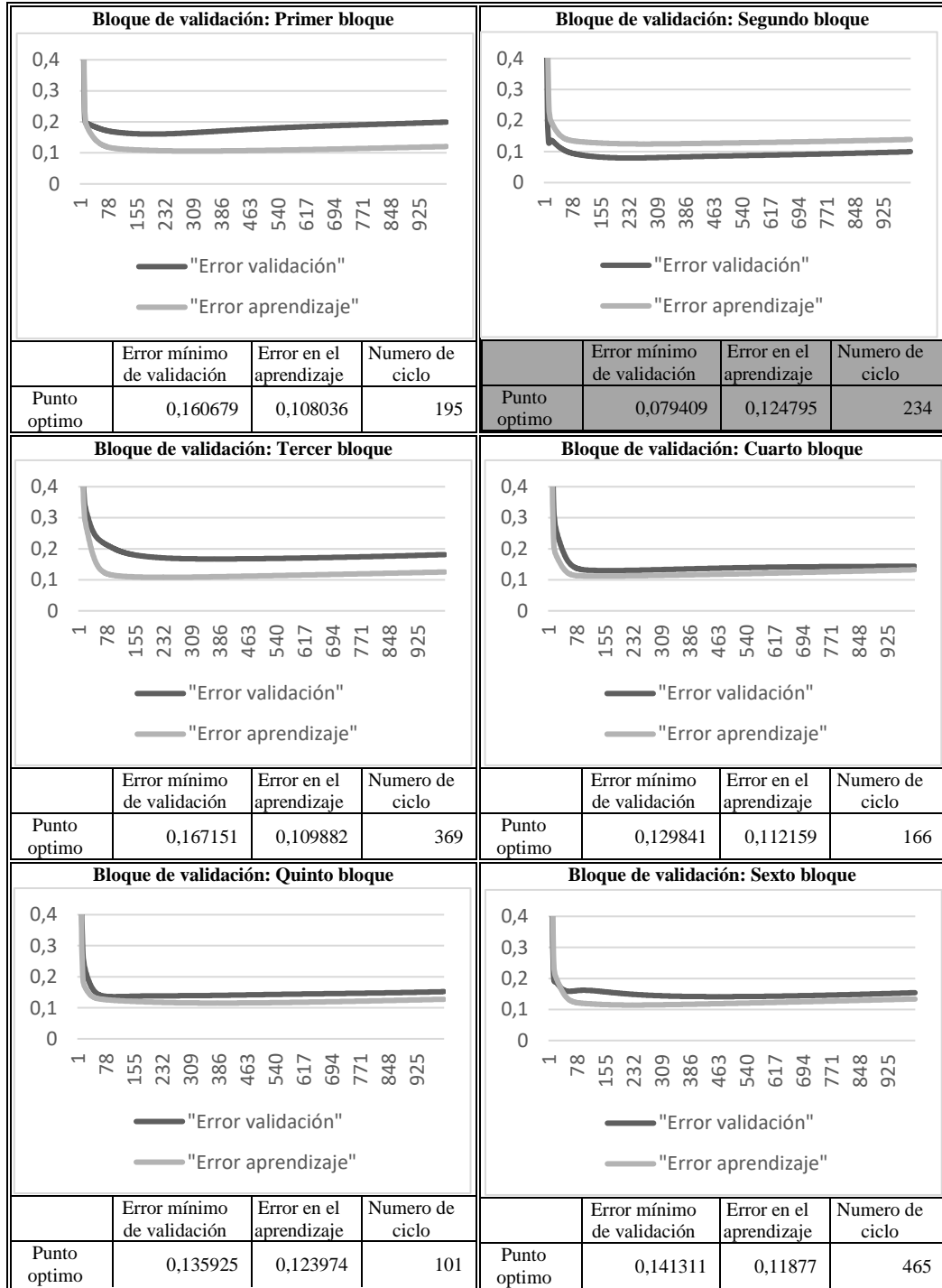


Ilustración 37: Resultados del escenario BRGIV

Fuente: propia

e) **Comparación de los escenarios de la experimentación de la red neuronal artificial BRG**

En cada escenario se seleccionó el bloque de validación que alcanzó el mejor punto óptimo.

Tabla 15: **Tabla comparativa de los escenarios realizados con BRG**

Tipo de red	Método de aprendizaje	Función	Razón de aprendizaje	Escenario	Numero de neuronas	ciclo	Error mínimo de validación (<i>ECM</i>)
BR	Hibrido	Gaussiana	0,0001	BRGI	11	148	0,076314
				BRGII	12	88	0,052252
				BRGIII	13	136	0,078941
				BRGIV	14	234	0,079409

Nota. Tabla comparativa de los resultados de los cuatro escenarios de la experimentación de la red BRG. Fuente: propia

En la *tabla 15* se observan los cuatro escenarios realizados, en esta experimentación fue en el escenario BRGII con 12 neuronas en la capa oculta y en el ciclo de aprendizaje 88 donde se encontró el menor error cuadrático medio con un valor de 0,052252, lo que equivale a un 94,77% de nivel de eficiencia en la identificación de violencia escolar.

4.3.4. Experimentación de la red neuronal artificial BRC

Los parámetros que se utilizaron en la experimentación de esta red son los siguientes:

- Tipo de red: Red Neuronal de Base Radial
- Método de aprendizaje: Hibrido
- Función de activación: Inversa cuadrática
- Numero de ciclos máximo: 1000 ciclos
- Razón de aprendizaje: se inicializo en 0,0001 y se incrementa en 0,0001 por cada ciclo

Los menores errores de validación en los cuatro escenarios realizados con esta red neuronal artificial se hallaron en los segundos bloques de validación, entre los ciclos de aprendizaje del 78 al 309.

a) **Escenario BRCI**

En la *ilustración 38* se observa los resultados de la experimentación de la red neuronal artificial BRC con 11 neuronas en la capa oculta, donde en el segundo bloque se obtuvo el mejor punto óptimo que tuvo el menor error cuadrático medio, con un valor de 0,087667 en el ciclo 164.

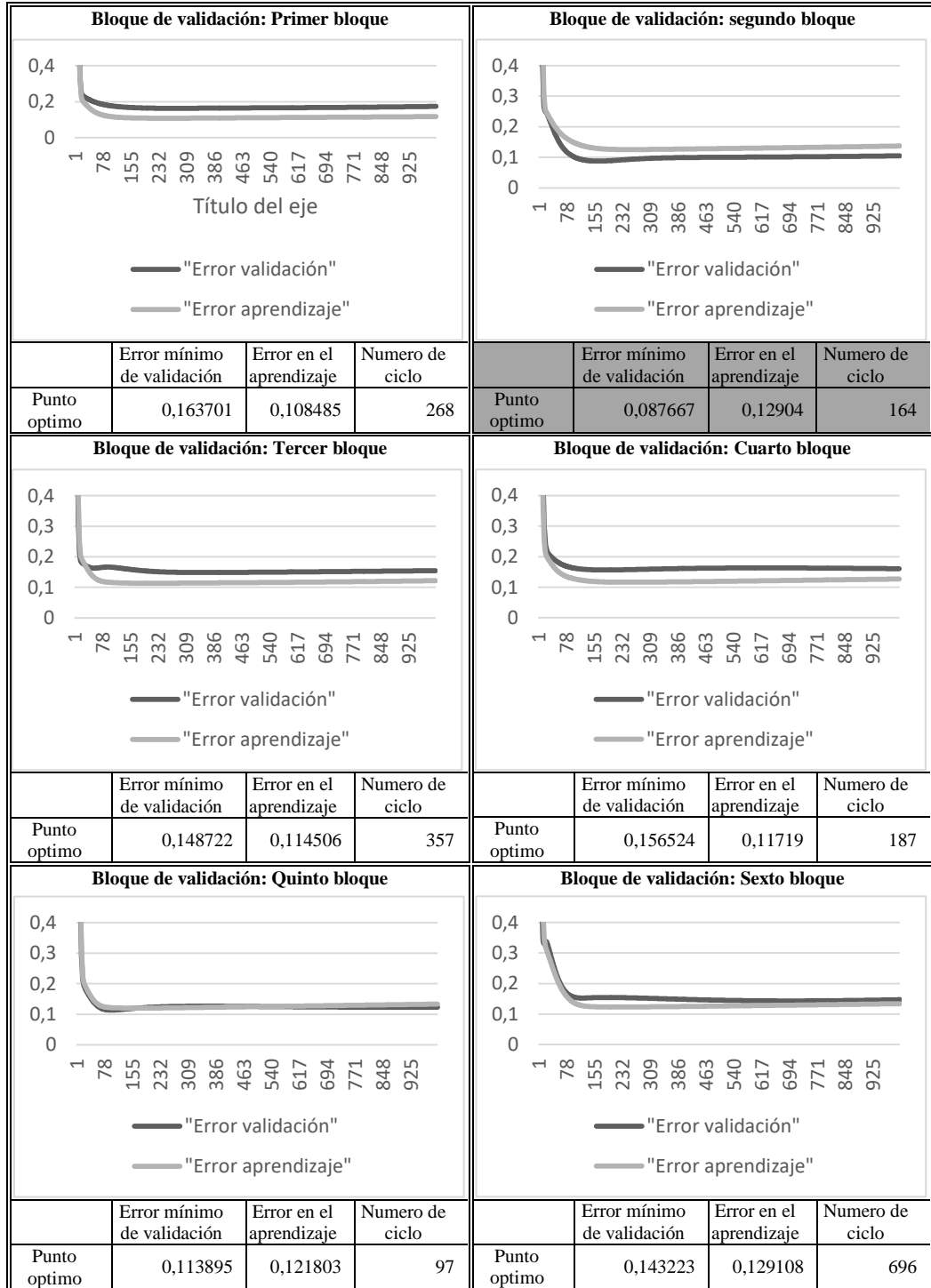


Ilustración 38: Resultados del escenario BRCI

Fuente: propia

b) Escenario BRCII

En la *ilustración 39* se observa los resultados de la experimentación de la red neuronal artificial BRC con 12 neuronas en la capa oculta, donde en el segundo bloque se obtuvo el mejor punto óptimo que tuvo el menor error cuadrático medio, con un valor de 0,069919 en el ciclo 113.

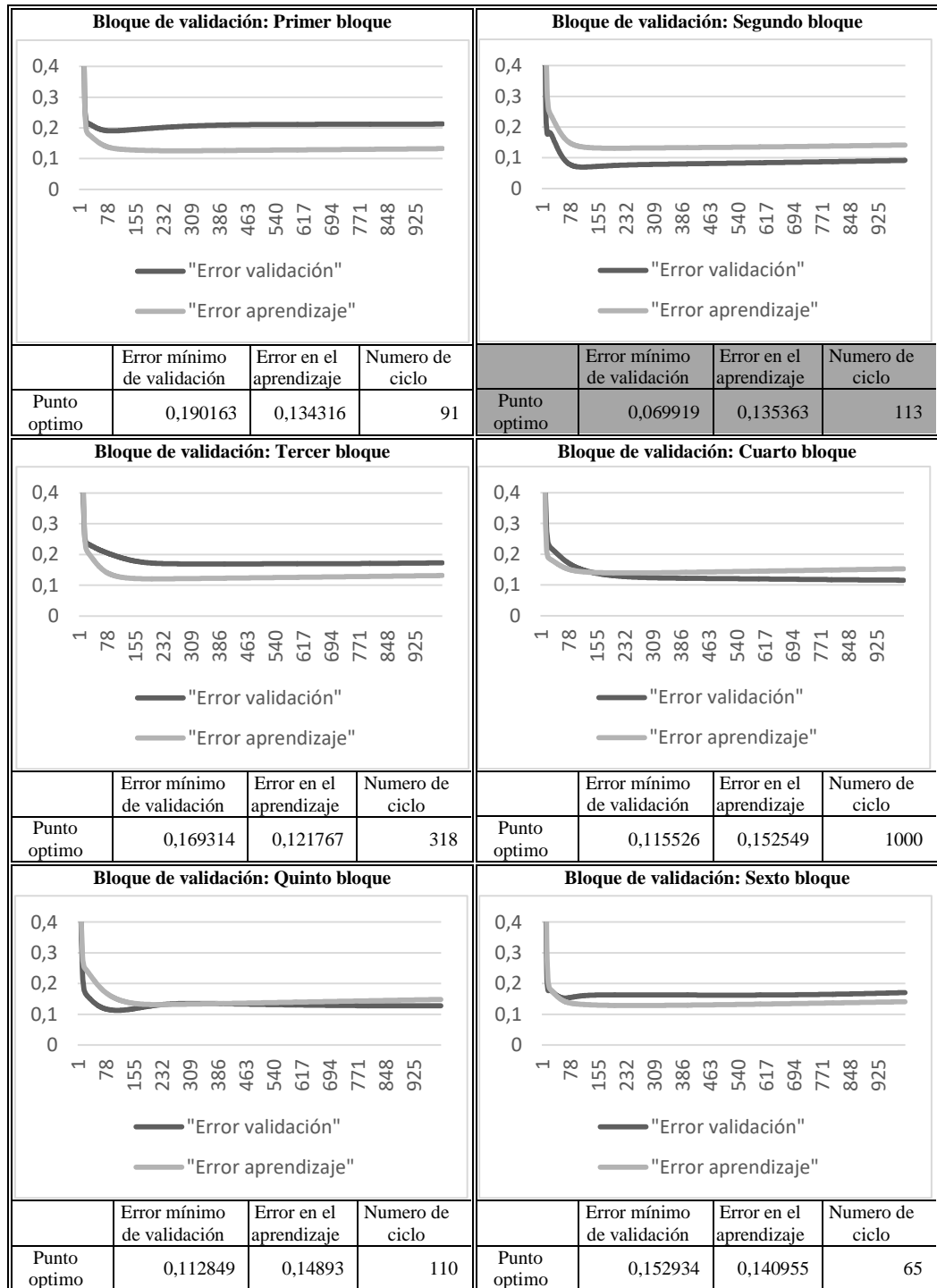


Ilustración 39: Resultados del escenario BRCII

Fuente: propia

c) **Escenario BRCIII**

En la *ilustración 40* se observa los resultados de la experimentación de la red neuronal artificial BRC con 13 neuronas en la capa oculta, donde en el segundo bloque se obtuvo el mejor punto óptimo que tuvo el menor error cuadrático medio, con un valor de 0,095564 en el ciclo 156.

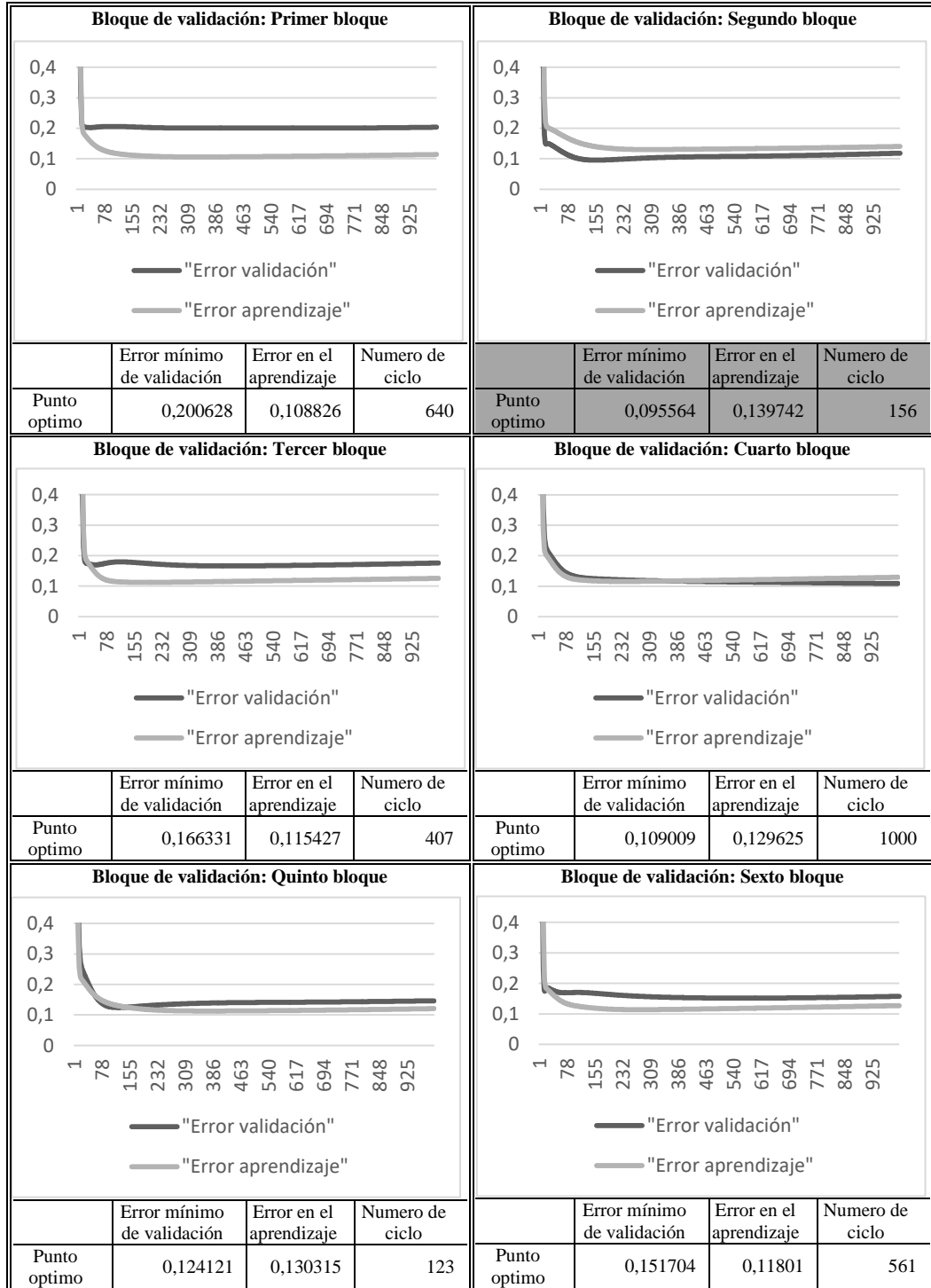


Ilustración 40: Resultados del escenario BRCIII

Fuente: Propia

d) Escenario BRCIV

En la *ilustración 41* se observa los resultados de la experimentación de la red neuronal artificial BRC con 14 neuronas en la capa oculta, donde en el segundo bloque se obtuvo el mejor punto óptimo que tuvo el menor error cuadrático medio, con un valor de 0,096401 en el ciclo 246.

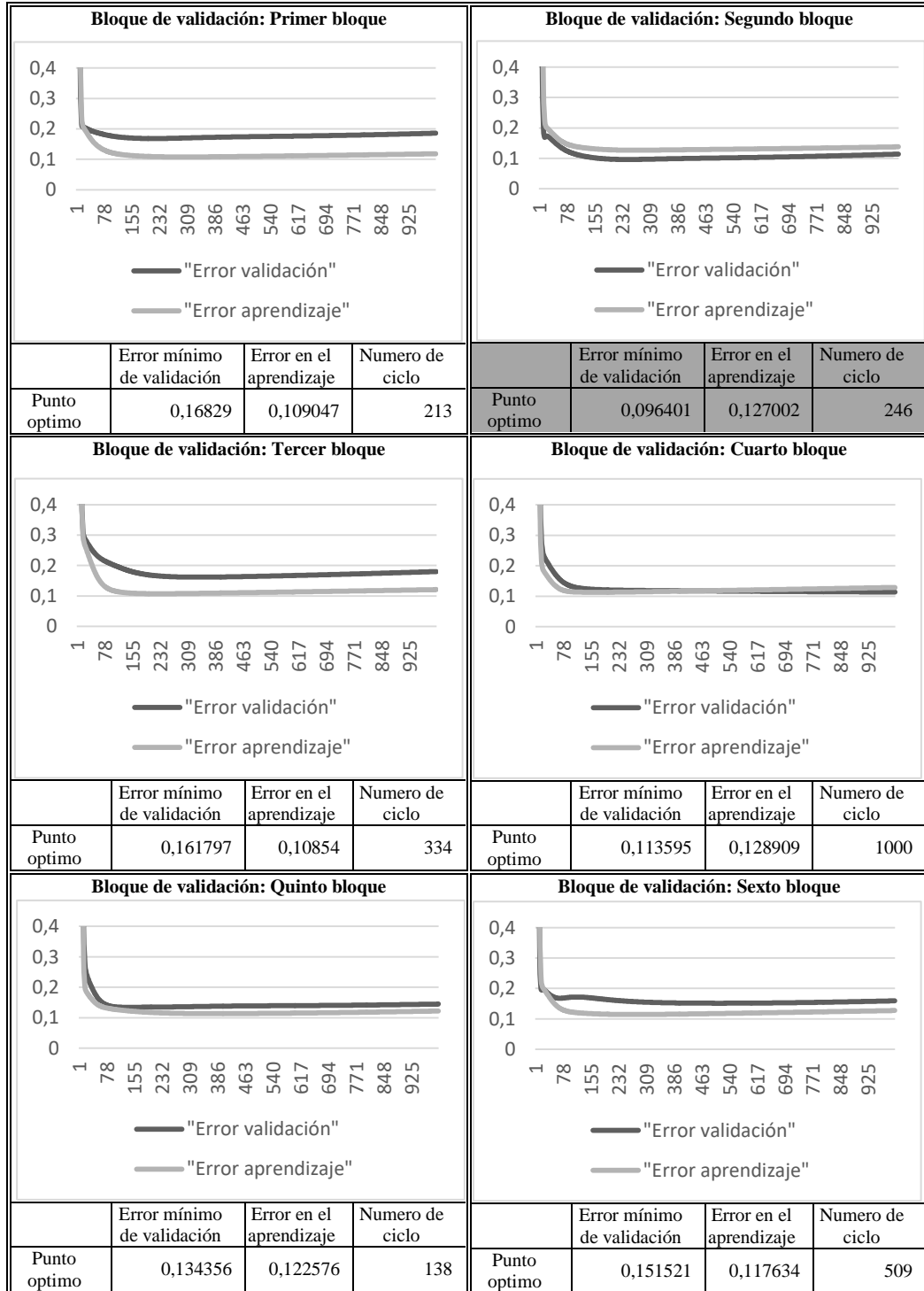


Ilustración 41: Resultados del escenario BRCIV

Fuente: propia

e) **Comparación de los escenarios de la experimentación del algoritmo BRC**

En cada escenario se seleccionó el bloque de validación que alcanzó el mejor punto óptimo.

Tabla 16: **Tabla comparativa de los escenarios realizados con BRC**

Tipo de red	Método de aprendizaje	Función	Razón de aprendizaje	Escenario	Numero de neuronas	ciclo	Error mínimo de validación (ECM)
BR	Híbrido	Inversa cuadrática	0,0001	BRCI	11	164	0,087667
				BRCII	12	113	0,069919
				BRCIII	13	156	0,095564
				BRCIV	14	246	0,096401

Nota. Tabla comparativa de los resultados de los cuatro escenarios de la experimentación de la red BRC. Fuente: propia

En la *tabla 16* se observan los cuatro escenarios realizados, en esta experimentación fue en el escenario BRCII con 12 neuronas en la capa oculta y en el ciclo de aprendizaje 113 donde se encontró el menor error cuadrático medio con un valor de 0,069919, lo que equivale a un 93,01% de nivel de eficiencia en la identificación de violencia escolar.

4.3.5. Experimentación de la red neuronal artificial RRS

Los parámetros que se utilizaron en la experimentación de esta red son los siguientes:

- Tipo de red: Red Neuronal Recurrente
- Método de aprendizaje: Recurrente en Tiempo Real
- Función de activación: Sigmoidal
- Instantes de tiempo máximo: 100000
- Razón de aprendizaje: se estableció en 0,001, no se incrementó en cada tiempo

En la experimentación con RRS se tuvo que realizar 100000 instantes de tiempo, pues su aprendizaje demora más que los otros tipos de redes neuronales artificiales, por lo cual en esta experimentación se consumió un mayor nivel de recursos.

Es importante mencionar que el número de neuronas utilizadas en cada escenario son sumadas a las 11 neuronas de entrada y la única de salida, que por defecto ya están establecidas en el algoritmo de este tipo de red neuronal.

a) Escenario RRSI

En la ilustración 42 se observa los resultados de la experimentación de la red neuronal artificial RRS con 9 neuronas, donde en el segundo bloque se obtuvo el mejor punto óptimo que tuvo el menor error de validación, con un valor de 0,064778 en el instante de tiempo 44500.

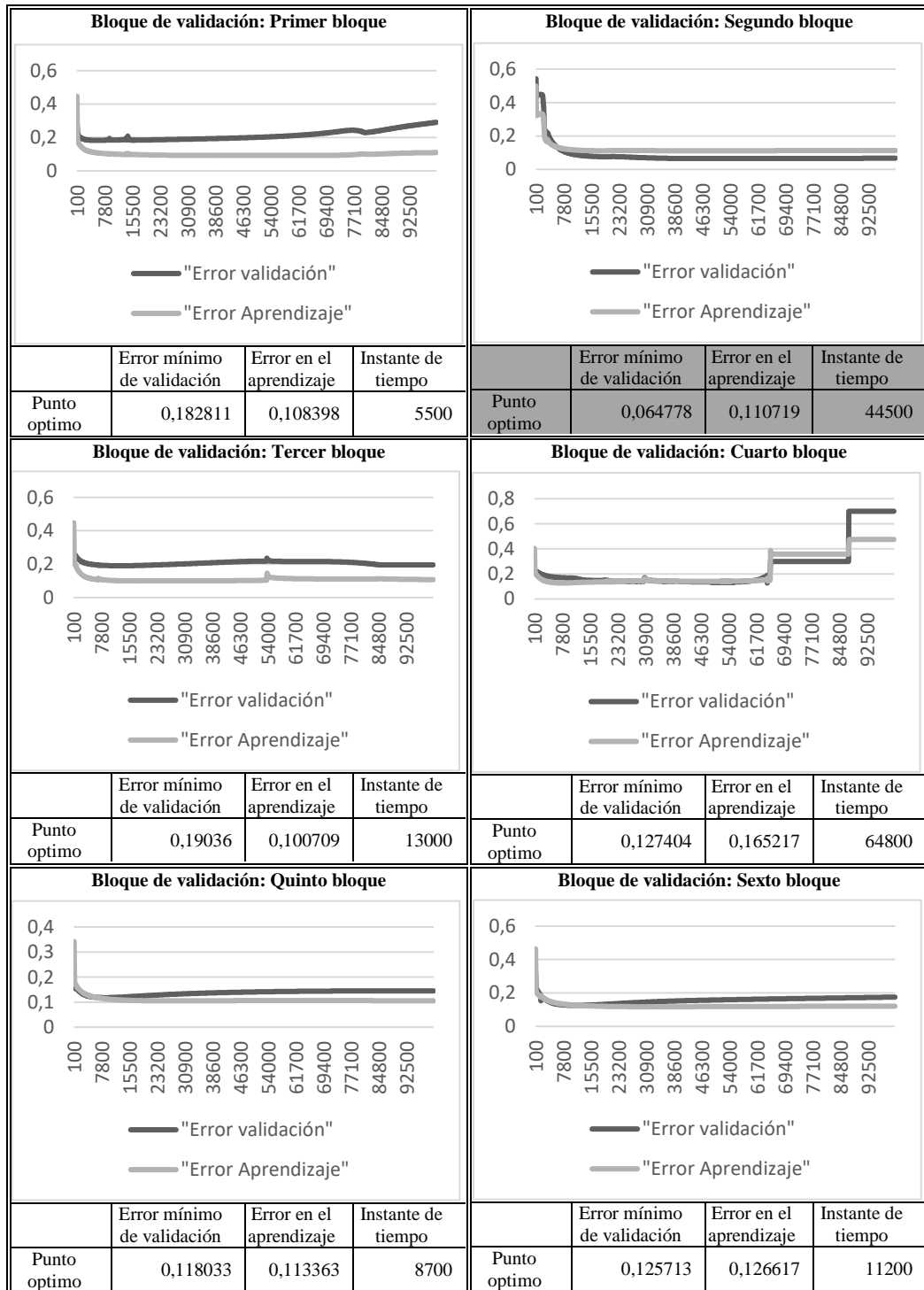


Ilustración 42: Resultados del escenario RRSI

Fuente: propia

b) Escenario RRSII

En la *ilustración 43* se observa los resultados de la experimentación de la red neuronal artificial RRS con 10 neuronas, donde en el segundo bloque se obtuvo el mejor punto óptimo que tuvo el menor error de validación, con un valor de 0,038311 en el instante de tiempo 55600.

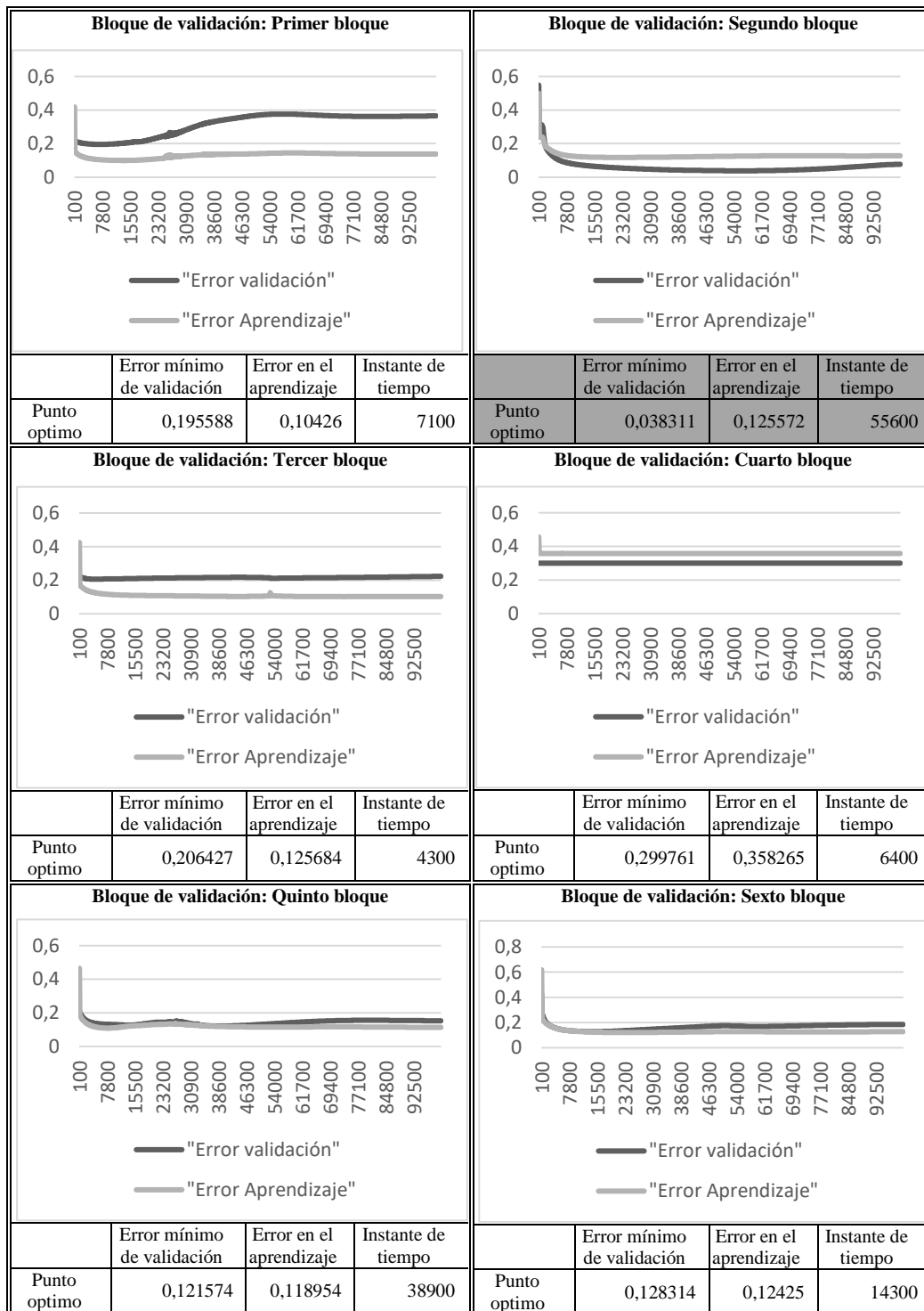


Ilustración 43: Resultados del escenario RRSII

Fuente: propia

c) **Escenario RRSIII**

En la *ilustración 44* se observa los resultados de la experimentación de la red neuronal artificial RRS con 11 neuronas, donde en el segundo bloque se obtuvo el mejor punto óptimo que tuvo el menor error de validación, con un valor de 0,052064 en el instante de tiempo 45700.

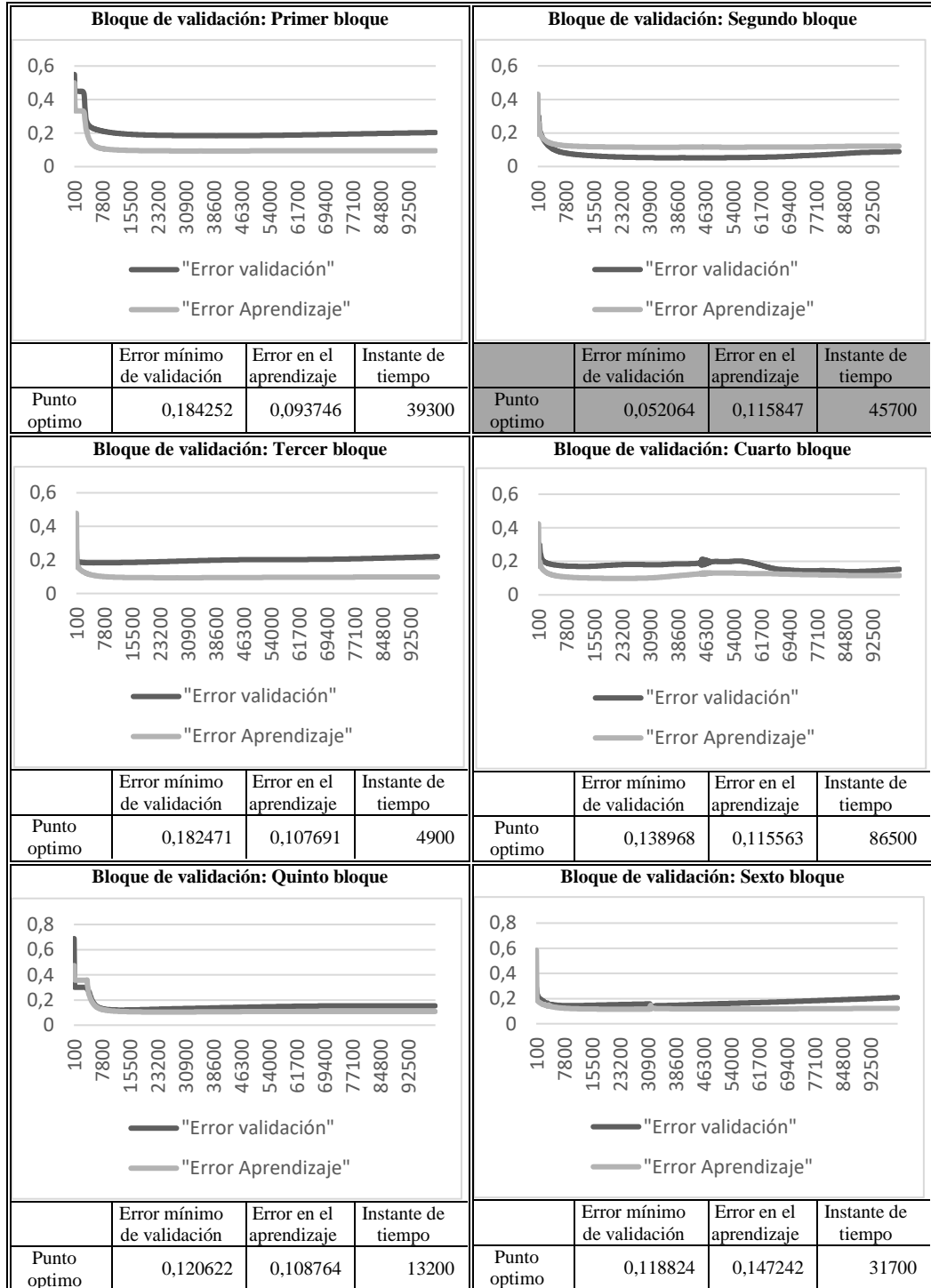


Ilustración 44: Resultados del escenario RRSIII

Fuente: propia

d) Escenario RRSIV

En la *ilustración 45* se observa los resultados de la experimentación de la red neuronal artificial RRS con 12 neuronas, donde en el segundo bloque se obtuvo el mejor punto óptimo que tuvo el menor error de validación, con un valor de 0,067978 en el instante de tiempo 51400.

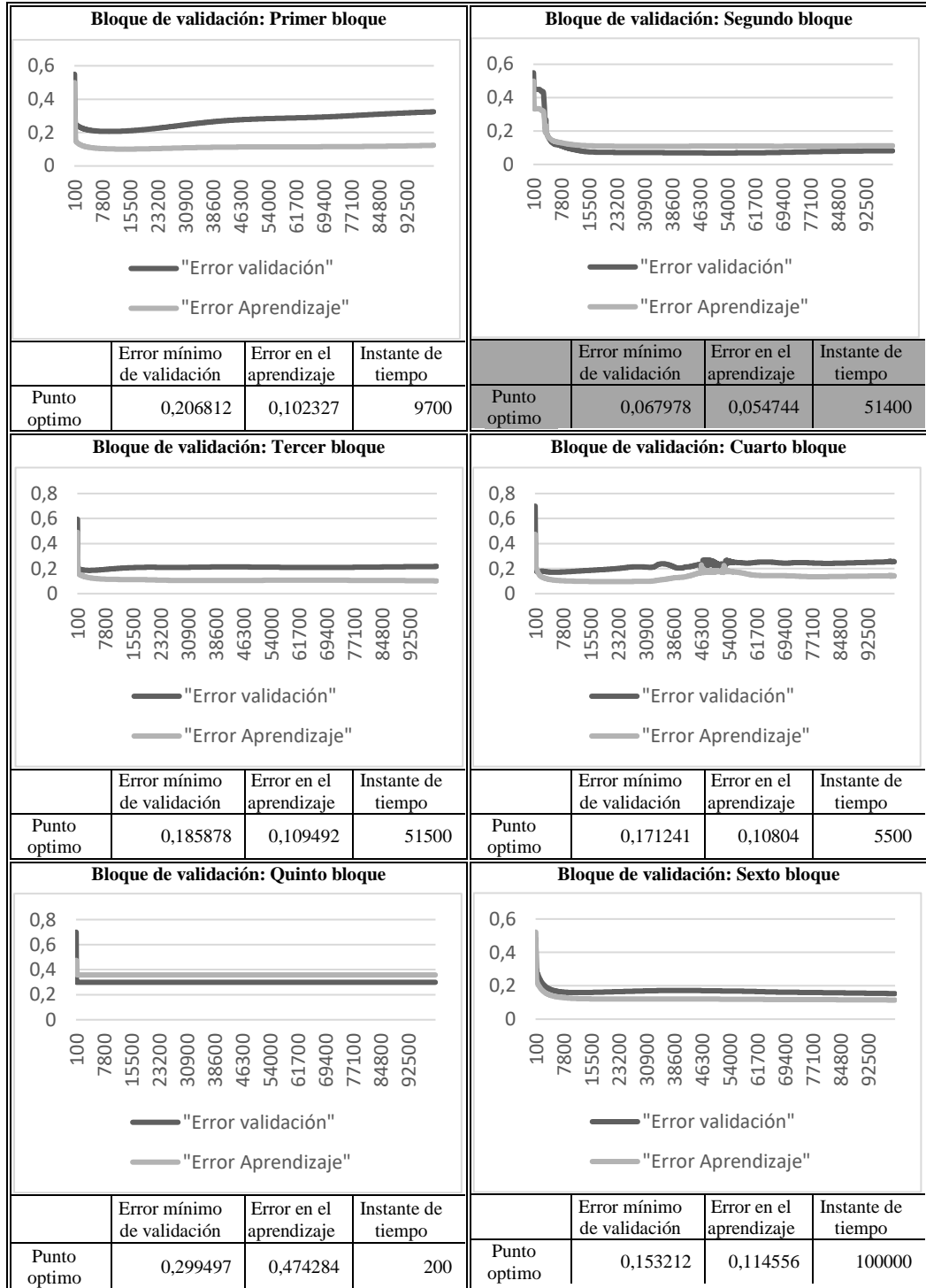


Ilustración 45: Resultados del escenario RRSIV

Fuente: propia

e) **Comparación de los escenarios de la experimentación del algoritmo RRS**

En cada escenario se seleccionó el bloque de validación que alcanzó el mejor punto óptimo.

Tabla 17: **Tabla comparativa de los escenarios realizados con RRS**

Tipo de red	Método de aprendizaje	Función	Razón de aprendizaje	Escenario	Nº de neuronas	Instante de tiempo	Error mínimo de validación (<i>ECM</i>)
RR	Recurrente en Tiempo Real	Sigmoïdal	0,001	RRSI	9	44500	0,064778
				RRSII	10	55600	0,038311
				RRSIII	11	45700	0,052064
				RRSIV	12	51400	0,067978

Nota. Tabla comparativa de los resultados de los cuatro escenarios de la experimentación de la red RRS. Fuente: propia

En la *Tabla 17* se observan los cuatro escenarios realizados, en la presente experimentación fue en el escenario RRSII con 10 neuronas recurrentes y en el tiempo de aprendizaje 55600 donde se encontró el menor error cuadrático medio con un valor de 0,038311, lo que equivale a un 96,17% de nivel de eficiencia en la identificación de violencia escolar.

4.3.6. Experimentación de la red neuronal artificial RRH

Los parámetros que se utilizaron en la experimentación de esta red son los siguientes:

- Tipo de red: Red Neuronal Recurrente
- Método de aprendizaje: Recurrente en Tiempo Real
- Función de activación: Tangente hiperbólica
- Instantes de tiempo máximo: 100000
- Razón de aprendizaje: se estableció en $1 * 10^{-11}$ y no se incremento

En esta red neuronal artificial los resultados no fueron los mejores, pues la red neuronal recurrente RRH no llegaba a estabilizarse, pese a que no solo se realizó pruebas con un número diferente de neuronas, sino que también se experimentó con diferentes valores en la razón de aprendizaje. Por tal razón, en los cuatro bloques de validación donde se encontró los mejores resultados en comparación con los demás bloques, los errores de validación fueron de todas formas demasiados altos.

a) **Escenario RRHI**

En la *ilustración 46* se observa los resultados de la experimentación de la red neuronal artificial RRH con 9 neuronas, donde en el sexto bloque se obtuvo el mejor punto óptimo que tuvo el menor error de validación, con un valor de 0,332174 en el instante de tiempo 600.

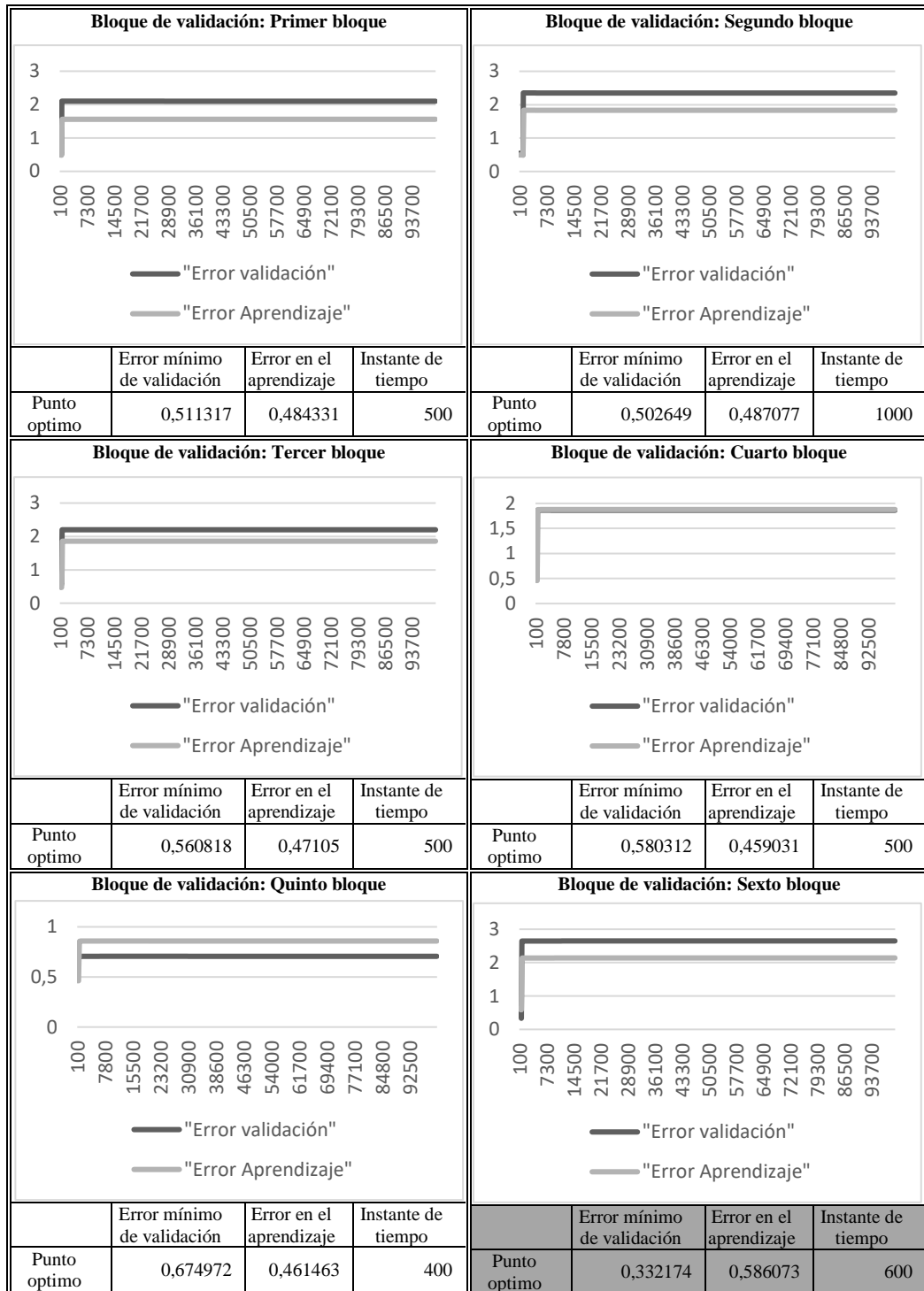


Ilustración 46: Resultados del escenario RRHI

Fuente: propia

b) Escenario RRHII

En la *ilustración 47* se observa los resultados de la experimentación de la red neuronal artificial RRH con 10 neuronas, donde en el tercer bloque se obtuvo el mejor punto óptimo que tuvo el menor error de validación, con un valor de 0,11859 en el instante de tiempo 300.

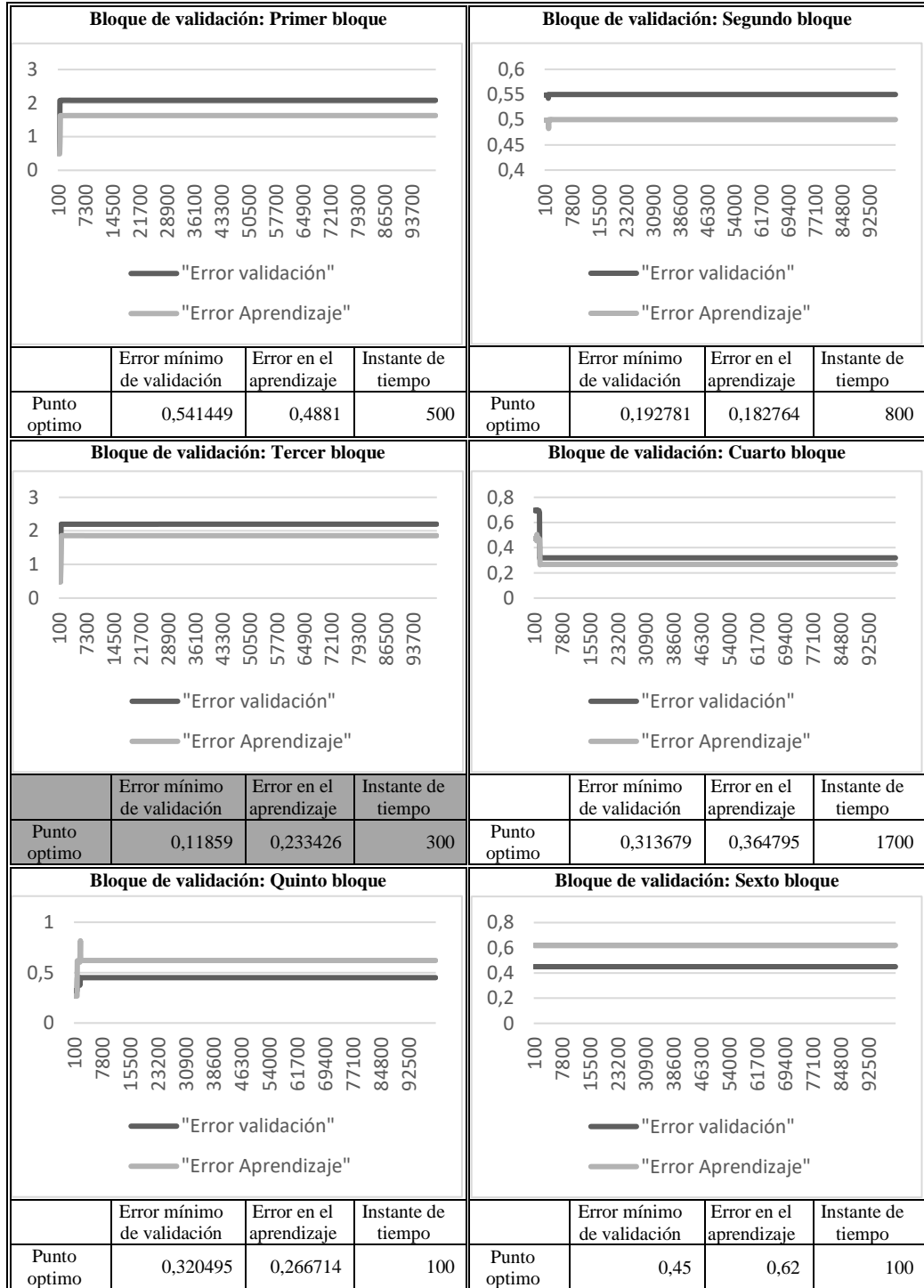


Ilustración 47: Resultados del escenario RRHII

Fuente: propia

c) **Escenario RRHIII**

En la *ilustración 48* se observa los resultados de la experimentación de la red neuronal artificial RRH con 11 neuronas, donde en el cuarto bloque se obtuvo el mejor punto óptimo que tuvo el menor error de validación, con un valor de 0,195154 en el instante de tiempo 1000.

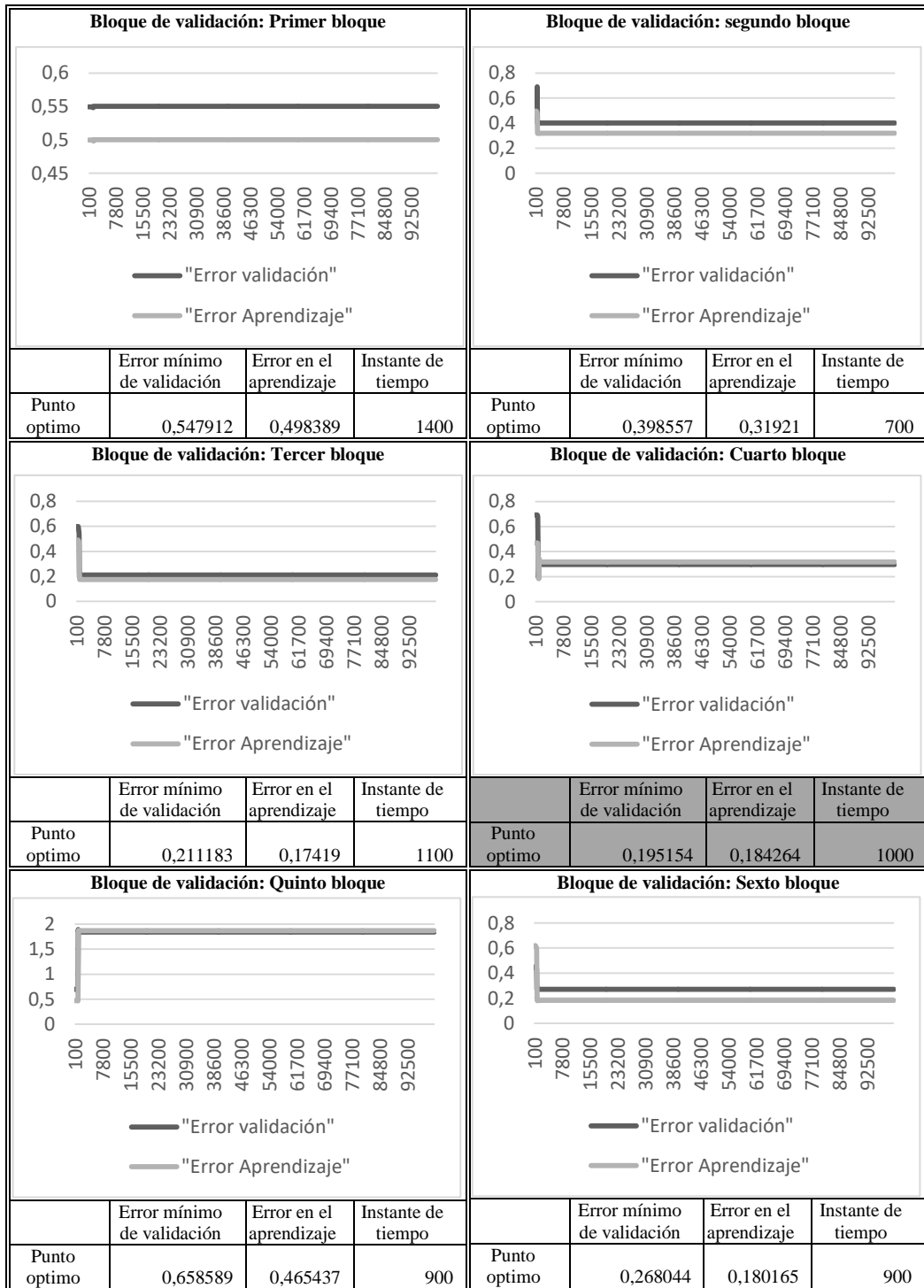


Ilustración 48: Resultados del escenario RRHIII

Fuente: propia

d) Escenario RRHIV

En la *ilustración 49* se observa los resultados de la experimentación de la red neuronal artificial RRH con 12 neuronas, donde en el sexto bloque se obtuvo el mejor punto óptimo que tuvo el menor error de validación, con un valor de 0,436942 en el instante de tiempo 2000.

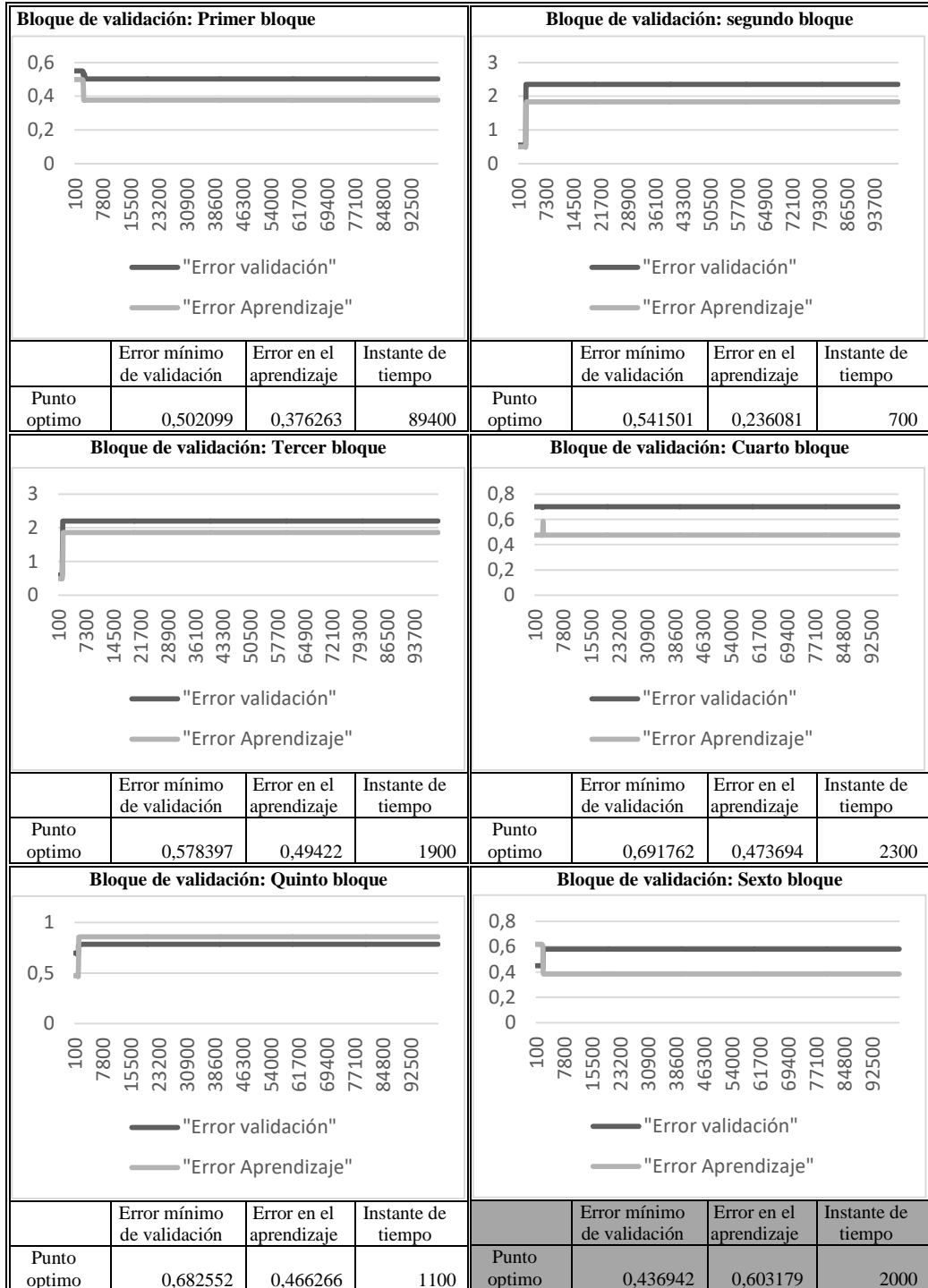


Ilustración 49: Resultados del escenario RRHIV

Fuente: propia

e) **Comparación de los escenarios de la experimentación del algoritmo RRH**

En cada escenario se seleccionó el bloque de validación que alcanzó el mejor punto óptimo.

Tabla 18: **Tabla comparativa de los escenarios realizados con RRH**

Tipo de red	Método de aprendizaje	Función	Razón de aprendizaje	Escenario	Nº de neuronas	Instante de tiempo	Error mínimo de validación (ECM)
RR	Recurrente en Tiempo Real	Tangente hiperbólica	$1 * 10^{-11}$	RRHI	9	600	0,332174
				RRHII	10	300	0,11859
				RRHIII	11	1000	0,195154
				RRHIV	12	2000	0,436942

Nota. Tabla comparativa de los resultados de los cuatro escenarios de la experimentación de la red RRH. Fuente: propia

En la *tabla 18* se observan los cuatro escenarios realizados, en esta experimentación fue en el escenario RRHII con 10 neuronas recurrentes y en el tiempo 300 donde se encontró el menor error cuadrático medio con un valor de 0,11859, lo que equivale a un 88,14% de nivel de eficiencia en la identificación de violencia escolar.

CAPÍTULO V: RESULTADOS DE LA INVESTIGACIÓN

5.1. Resultado del Diseño de la Base de Datos para las Redes Neuronales Artificiales

En la *ilustración 50* se puede observar la base de datos desarrollada y utilizada en la experimentación de las redes neuronales artificiales.



Ilustración 50: Diagrama de base de datos para Redes neuronales artificiales

Fuente: Propia

5.2. Resultados del desarrollo y ajuste de redes neuronales artificiales aplicadas a la identificación de violencia escolar

Los algoritmos desarrollados fueron implementados utilizando el lenguaje de programación C++ en el entorno de desarrollo Eclipse C/C++, pero por motivos de orden y de una mejor comprensión de cada uno de ellos se prefirió mostrar los pseudocódigos en lugar de los códigos desarrollados.

5.2.1. Tipo 1: Perceptrón Multicapa con método de aprendizaje de Retropropagación (PM)

El algoritmo desarrollado que permitió realizar la experimentación de las redes neuronales artificiales PMS y PMH en la identificación de violencia escolar es el siguiente:

Inicio del Algoritmo

```
//Leer número de capas ocultas
Leer(numero_capas_ocultas)

//se le suma la capa de entrada y la de salida
numero_capas = numero_capas_ocultas + 2

//Se lee el número de neuronas para cada capa oculta
para (i = 1; i ≤ numero_capas; i++)
Inicio para
/*Si i = 1 la capa actual es la de entrada, por lo tanto el número de neuronas
*siempre 11 es el número de entradas a la red. En caso de ser i = numero_capas, es
*decir la última capa el número de neuronas es siempre 1 pues la red tiene una sola
*salida.
*/
Leer(numero_neuronas[i])
Fin para

//Se lee la función de activación, puede ser función Sigmoidal (PMS) o
//Hiperbólica (PMH)
Leer(funcion_activacion)

/*
*Se crea una matriz para las neuronas, donde las columnas indican el número de
*capa y las filas el número de la neurona de esa capa; no siempre es una matriz
*cuadrada ya que el número de neuronas en cada capa son pueden ser diferentes, por
*tanto se crea una matriz dinámicamente
*/
neurona[numero_capas]
Para (i = 1; i ≤ numero_capas; i++)
Inicio para
neurona[i][numero_neuronas[i]]
Fin para
```

```
/*
*Se crea un arreglo tridimensional de las conexiones de las neuronas, pues se debe
*representar tres dimensiones la primera representa el número de capa, y como las
*neuronas de una capa están conectadas a todas las de la siguiente capa, entonces la
*segunda dimensión representa el número de neuronas en cada capa y la tercera el
*número de neuronas en la capa siguiente. También se crea una matriz para los
*umbrales de las neuronas donde la columna indica el número de capa y la fila el
*número de la neurona
*/
conexión[numero_capas]
umbral[numero_capas]
Para(i = 1; i ≤ numero_capas; i++)
Inicio para
    conexión[i][numero_neuronas[i]]
    umbral[i][numero_neuronas[i]]
    Para(j = 1; j ≤ numero_neuronas[i]; j++)
    Inicio para
        conexión[i][j][numero_neuronas[i+1]]
    Fin para
Fin para

//Se lee el número de ciclos para el aprendizaje de la red
Leer(numero_ciclos)

/*
*Según la validación cruzada dividimos la data en seis bloques, para ello creamos una
*matriz donde las columnas indican el número de bloque y las filas los patrones que
*pertenece a ese bloque
*/
bloque[6][numero_patrones]

//recorremos los seis ciclos de validación cruzada
contador_cross = 1
Mientras(contador_cross ≤ 6)
Inicio mientras para validación cruzada
    //seleccionamos el bloque para validación y los bloques para aprendizaje
    Para(i = 1; i ≤ 6; i++)
    Inicio para
        Si(i == contador_cross)
        Entonces
            patrones_para_validacion = patrones_para_validacion +
            Contar(bloque[i][numero_patrones])
        Caso Contrario
            patrones_para_aprendizaje = patrones_para_aprendizaje +
            Contar(bloque[i][numero_patrones])
    Fin para
    //se establece la razón de aprendizaje

/*
*Se inicializa los pesos de las conexiones y umbrales de las neuronas con
*números aleatorios
*/
Para(i = 1; i ≤ numero_capas; i++)
Inicio para
    Para(int j = 1; j ≤ numero_neuronas[i]; j++)
    Inicio para
        umbral[i][j] = random
        Para(k=1; k ≤ numero_neuronas[i+1]; k++)
        Inicio para
```



```
        conexion[i][j][k] = random
    Fin para
  Fin para
Fin para

//Se recorre el número de ciclos establecido
contador_ciclo = 1
Mientras(contador_ciclo ≤ numero_ciclos)
  Inicio mientras para ciclos

    //inicializamos el error cuadrático medio de predicción
    ECMP = 0

    //recorremos los patrones de aprendizaje
    contador_patron = 1
    Mientras(contador_patron ≤ patrones_para_aprendizaje)
      Inicio mientras para patrones

        /*
        *Las neuronas de entrada (primera capa) reciben los valores del
        *patrón actual (estudiante) que tiene 11 indicadores de convivencia y
        *una situación real respecto a la existencia de violencia escolar, por
        *lo tanto estudiante es un matriz donde las columnas representan el
        *número de patrón y las filas sus indicadores y situación real
        */
        Para(i = 1; i ≤ 11; i++)
          Inicio para
            neurona[1][i] = estudiante[contador_patron][i]
          Fin para

        //Se realiza la propagación del patrón de entrada actual por la red
        Para(i = 2; i ≤ numero_capas; i++)
          Inicio para
            Para(int j = 1; j ≤ numero_neuronas[i]; j++)
              Inicio para
                /*
                *Se inicializa un auxiliar que acumulara la entrada efectiva
                *de la neurona actual
                */
                auxiliar_mu = 0

                /*
                *Si la capa actual es la segunda entonces esta capa es la
                *primera de las capas ocultas, por lo tanto recibe de la capa de
                *entrada los 11 valores que son los 11 indicadores de los
                *estudiantes
                */
                Si(i == 2)
                  Entonces
                    Para(k = 1; k ≤ 11; k++)
                      Inicio para
                        auxiliar_mu = auxiliar_mu + (neurona[1][k] *
                        conexion[1][k][j])
                      Fin para

                    //Sumamos el umbral de la neurona
                    auxiliar_mu = umbral[i][j]
```



```
/*
*La salida de la neurona actual es el resultado del
*procesamiento de la función de activación respecto a la
*entrada efectiva (auxiliar_mu)
*/
neurona[i][j] = funcion_activacion (auxiliar_mu)

Caso contrario

/*
*Se recorre por las neuronas de las capa anterior ya que la
*salida de ellas son propagadas a la neurona actual
*/
Para(k = 1;k ≤ numero_neuronas[i-1];k++)
Inicio para
    auxiliar_mu= auxiliar_mu + (neurona[i-1][k] *
    conexion[i-1][k][j])
Fin para

//Sumamos el umbral de la neurona
auxiliar_mu = umbral[i][j]

/*La salida de la neurona actual es el resultado del
*procesamiento de la función de activación respecto a la
*entrada efectiva (auxiliar_mu)
*/
neurona[i][j] = funcion_activacion (auxiliar_mu)

Fin para

Fin para
/*
*Se calcula el error cometido por la neurona de salida de la red
*respecto a la situación real del estudiante que viene a ser la fila
*número 12 de matriz estudiante
*/
error_neurona[numero_capas][1] =
estudiante[contador_patron][12] - neurona[numero_capas][1]

//Se retropropaga el error a las neuronas de las capas anteriores de
la red
Para(i = numero_capas - 2; i ≥ 1; i++)
Inicio para

    Para(j = 1; j ≤ numero_neuronas[i] ; i++)
    Inicio para

        suma_error = 0

        /*
        *Se recorre las neuronas de la capa siguiente para
        *conseguir su error asociado y la conexión de entre las
        *dos neuronas
        */
        Para(k=1;k ≤ numero_neuronas[i+1];k++)
        Inicio para
            suma_error = suma_error + neurona[i][j] *
            conexion[i][j][k]
```



```
Fin para

/*
 *Se calcula el error asociado a la neurona actual con la
 *multiplicación de la sumatoria y la derivada de la
 *función de activación
 */
error_neurona[i][j] = suma_error *
Derivada(error_neurona[i][j])

Fin para
Fin para
/*
 *Se actualiza los pesos de las conexiones y los umbrales de las
 *neuronas
 */
Para(i = 2; i ≤ numero_capas; i++)
Inicio para
Para(j = 1; j ≤ numero_neuronas[i]; j++)
Inicio para

//se actualiza el umbral
umbral[i][j] = umbral[i][j] + (razon_aprendizaje *
error_neurona[i][j])

/*
 *Se actualiza los pesos, si la capa es la segunda entonces
 *se actualiza respecto a las salidas de las neuronas de
 *entrada
 */
Si(i == 2)
Entonces
Para(k = 1; k ≤ 11; k++)
Inicio para

/*
 *La actualización se da en función de la razón de
 *aprendizaje, el error de la neurona actual y la
 *salida de la neurona de la capa anterior
 */
conexion[1][k][j] = conexion[1][k][j] +
(razon_aprendizaje * error_neurona[i][j] *
neurona[1][k])

Fin para

Caso contrario
Para(k = 1; k ≤ numero_neuronas[i-1]; k++)
Inicio para

conexion[i-1][k][j] = conexion[i-1][k][j] +
(razon_aprendizaje * error_neurona[i][j] *
neurona[i-1][k])

Fin para

Fin para
Fin para
```



```
        /*
        *Se calcula el error cuadrático medio para este patrón que se
        *acumula en la variable error
        */
        error = error + (estudiante[contador_patron][12] -
            neurona[numero_capas][1])2

        contador_patron = contador_patron + 1

    Fin mientras para patrones

    //Se calcula el error cuadrático medio del aprendizaje
    ECMP_aprendizaje = (1/patrones_para_apredizaje) * error

    /*
    *Se realiza la validación con los patrones del bloque seleccionado para esta
    *etapa
    */
    contador_patron_validacion = 1
    Mientras(contador_patron_validacion ≤ patrones_para_validacion)
    Inicio mientras

        /*
        *Se propaga hacia adelante el patron actual
        *(estudiante[contador_patron_validacion]) de la misma forma que
        *en el aprendizaje
        */

        /*Se calcula el error cuadrático medio de predicción del patrón actual
        *y lo acumulamos en la variable error
        */
        error = error + (estudiante[contador_patron][12] -
            neurona[numero_capas][1])2

        contador_patron_validacion = contador_patron_validacion + 1
    Fin mientras

    //Calculamos el error cuadrático medio de predicción en la validación
    ECMP_validacion = (1/ patrones_para_validacion) * error

    //Registramos en la base de datos los siguientes resultados
    Registramos(contador_cross, contado_ciclo, ECMP_aprendizaje,
    ECMP_validacion,función_activacion)

    contador_ciclo = contador_ciclo + 1

    Fin mientras para ciclos

    contador_cross = contador_cross + 1

    Fin mientras para validación cruzada

Fin del Algoritmo
```

5.2.2. Tipo 2: Red Neuronal de Base Radial con método de aprendizaje Híbrido (BR)

El algoritmo desarrollado que permitió realizar la experimentación de las redes neuronales artificiales BRG y BRC en la identificación de violencia escolar es el siguiente:

Inicio del Algoritmo

```
//Se lee el número de neuronas para la capa oculta, a diferencia de PM el tipo BR --
//solo tiene una capa oculta
Leer(numero_neuronas_capa_oculta)

//Se lee la función de activación, puede ser función Gaussiana (BRG) o
//Inversa cuadrática (BRC)
Leer(funcion_activacion)

/*
 *Se crea una matriz para las neuronas, las columnas indican si es una neurona de la
 *capa de entrada (1), de la capa oculta (2) o de la capa de salida y las filas el número
 *de la neurona en dicha capa, debe ser creado dinámicamente puesto que el número
 *de neuronas en cada capa es diferente
 */
neurona[1][11] //capa de entrada tiene 11
neurona[2][numero_neuronas_capa_oculta] //neuronas de la capa oculta
neurona[3][1] //una sola neurona en la capa de salida

/*
 *Se crea un arreglo de las conexiones de las neuronas entre la capa oculta y la de
 *salida, cabe resaltar que de la capa de entrada a la oculta no se utilizan los pesos
 *asociados a sus conexiones, pues es la parte no supervisada y es donde se utiliza k-
 *modas
 */
conexión[numero_neuronas_capa_oculta]

//Se lee el número de ciclos para el aprendizaje de la red
Leer(numero_ciclos)

/*
 *Según la validación cruzada dividimos la data en seis bloques, para ello creamos
 *una matriz donde las columnas indican el número de bloque y las filas los patrones
 *que pertenecen a ese bloque
 */
bloque[6][numero_patrones]

//recorremos los seis ciclos de validación cruzada
contador_cross = 1
Mientras(contador_cross ≤ 6)
Inicio mientras para validación cruzada
//seleccionamos el bloque para validación y los bloques para aprendizaje
Para(i = 1; i ≤ 6; i++)
Inicio para
Si(i == contador_cross)
Entonces
    patrones_para_validacion = patrones_para_validacion +
    Contar(bloque[i][numero_patrones])
Caso Contrario
```



```
        patrones_para_aprendizaje = patrones_para_aprendizaje +
        Contar(bloque[i][numero_patrones))
Fin para

//se establece la razón de aprendizaje

/*
*Se establece un patrón de entrada de aprendizaje a las modas para cada clase
*(neurona de la capa oculta) aleatoriamente.
*/
Para (i = 1; i ≤ numero_neuronas_capa_oculta; i++)
Inicio para
    Modas[i] = Estudiante[random]
Fin para

/*
*Se repite el siguiente proceso mientras las modas de una clases varíen
*respecto su antigua posición
*/
Mientras(indicador == falso)
Inicio mientras para indicador de variación
    contador_patron = 1
    Mientras(contador_patron ≤ patrones_para_aprendizaje)
    Inicio mientras para asignación a clases
        menor = 12
        contador_diferentes = 0
        Para (i = 1; i ≤ numero_neuronas_capa_oculta; i++)
        Inicio para

            /*
            *Se recorre los 11 indicadores de la clase
            *Estudiante
            */

            Para(j = 1; j ≤ 11; j++)
            Inicio para
                /*
                *Si el indicador del estudiante actual es
                *diferente al indicador de la moda de la
                *clase "i" establecida se suma el contador
                *de diferencias
                */
                Si(Estudiante[contador_patron][j] ≠
                modas[i][j])
                Entonces
                    contador_diferentes++
            Fin para

            /*
            *Si el contador de diferencia en esta clase ha sido la
            *menor entonces se agrupa al patrón en dicha clase
            */
            Si(contador_difentes < menor)
            Entonces
                menor = contador_diferentes
```

```
*El índice de las filas 13 en la matriz
*estudiante es indicador de la clase
*(neurona de la capa oculta) a la que
*pertenece el patrón actual
*/
Estudiante[contador_patron][13] = i

Fin para
Fin mientras para asignación de clases
/*
*Almacenamos las modas de los centros de las neuronas
*actuales en una variable auxiliar.
*/

Para (i = 1; i ≤ numero_neuronas_capa_oculta; i++)
Inicio para
    modas_anterior[i] = modas[i]
Fin para

/*
* Se calcula las nuevas modas de los centros de las neuronas
*de la capa oculta en función a los nuevos patrones
*asignados a su agrupación
*/
Si(modas_anterior == modas)
Entonces
    Indicador = true
Caso contrario
    Indicador = false

Fin mientras para indicador de variación

/*
*Se calcula las amplitudes de las neuronas, en base al coeficiente de
*disimilitud Jaccard
*/
Para(i = 1; i ≤ numero_neuronas_capa_oculta; i++)
Inicio para

    /*
    *Se calcula la amplitud de la clase de la neurona i, donde p y c son las
    *neuronas más cercanas a ella
    */
    amplitud[i] =  $\sqrt{d(modas[p], modas[c])}$ 
Fin para

/*
*Inicializamos los pesos de las conexiones
*/
Para(i=1; i ≤ numero_neuronas_capa_oculta; i++)
Inicio para
    conexion[1][i] = random
Fin para

/*
*Recorremos los ciclos de aprendizaje
*/
```



```
Mientras(contador_ciclo ≤ numero_ciclos)
Inicio mientras para ciclos
    //inicializamos el error cuadrático medio de predicción
    ECMP = 0

    //recorremos los patrones de aprendizaje
    contador_patron = 1
    Mientras(contador_patron ≤ patrones_para_aprendizaje)
    Inicio mientras para patrones
        /*
        *Las neuronas de entrada reciben los valores del patrón
        *actual (estudiante) que tiene 11 indicadores de convivencia
        *y una situación real respecto a la existencia de violencia
        *escolar, por lo tanto, Estudiante es un matriz donde las
        *columnas representan el número de patrón y las filas sus
        *indicadores y situación real
        */
        Para(i = 1; i ≤ 11; i++)
        Inicio para
            neurona[1][i] = Estudiante[contador_patron][i]
        Fin para

        Para(i=1; i ≤ numero_neuronas_capa_oculta; i++)
        Inicio para
            /*
            *Se calcula la diferencia entre el patrón actual y la
            *moda de los centros de la neurona i
            */
            jaccard[i] =
                d(Estudiante[contador_patron], modas[i])

            /*
            *Se calcula la activación de la neurona i de la capa
            *oculta en base a la función seleccionada
            */
            Neurona[2][i] = f(jaccard[i]/amplitud[i])
        Fin para

        /*
        *Se calcula la salida de la red
        */
        suma = 0
        Para(i=1; i ≤ numero_neuronas_capa_oculta; i++)
        Inicio para
            suma = suma + Neurona[2][i] * conexion[1][i]
        Fin para
        neurona[3][1] = suma

        /*
        *Se calcula el error cometido por la salida de la red en
        *relación a la situación real del Estudiante que es el indicador
        *12
        */
        error = Estudiante[contador_patron][12] - neurona[3][1]
        /*
        *La actualización se da en función de la razón de
        *aprendizaje, el error de la salida y la salida de la neurona
        *de la capa anterior
        */
```



```
Para(i=1;i<= numero_neuronas_capa_oculta;i++)
Inicio para
    conexion[1][i] = conexion[1][i] +
    (razon_aprendizaje * error * neurona[2][i])

Fin para

ErrorSum = ErrorSum + error2

    contador_patron++
Fin mientras para patrones
//Se calcula el error cuadrático medio del aprendizaje
ECMP_aprendizaje = (1/patrones_para_aprendizaje) * ErrorSum

/*
*Se realiza la validación con los patrones del bloque seleccionado
*para esta etapa
*/
ErrorSum = 0
Mientras(contador_patron <= patrones_para_validacion)
Inicio mientras para patrones

    Para(i = 1;i <= 11;i++)
    Inicio para
        neurona[1][i] = Estudiante[contador_patron][i]
    Fin para

    Para(i=1;i<= numero_neuronas_capa_oculta;i++)
    Inicio para
        /*
        *Se calcula la diferencia entre el patrón actual y la
        *moda de los centros de la neurona i
        */
        jaccard[i] =
            d(Estudiante[contador_patron], modas[i])

        /*
        *Se calcula la activación de la neurona i de la capa
        *oculta en base a la función seleccionada
        */
        Neurona[2][i] = f(Jaccard[i]/amplitud[i])
    Fin para
    /*
    *Se calcula la salida de la red
    */
    suma = 0
    Para(i=1;i<= numero_neuronas_capa_oculta;i++)
    Inicio para
        suma = suma + Neurona[2][i] * conexion[1][i]
    Fin para
    neurona[3][1] = suma

    /*
    *Se calcula el error cometido por la salida de la red en
    *relación a la situación real del Estudiante que es el
    *indicador 12
    */
    error = Estudiante[contador_patron][12] - neurona[3][1]
```



```
        ErrorSum = ErrorSum + error2

        contador_patron++
    Fin mientras para patrones
    //Se calcula el error cuadrático medio de la validación
    ECMP_validacion = (1/patrones_para_validacion) * ErrorSum

    //Registramos en la base de datos los siguientes resultados
    Registramos(contador_cross, contado_ciclo, ECMP_aprendizaje,
    ECMP_validacion,función_activacion)

    contador_ciclo = contador_ciclo + 1

    Fin mientras para ciclos

    Contador_cross = contador_cross + 1

    Fin mientras para validación cruzada
Fin del Algoritmo
```

5.2.3. Tipo 3: Red Neuronal Recurrente con método de aprendizaje Recurrente en Tiempo Real (RR)

El algoritmo desarrollado que permitió realizar la experimentación de las redes neuronales artificiales RRS y RRH en la identificación de violencia escolar es el siguiente:

Inicio del Algoritmo

```
Leer(numero_neuronas_recurrentes)

/*
*sumamos las 11 neuronas de entrada y una de salida a las neuronas recurrentes
*/
numero_neuronas = numero_neuronas_recurrentes + 12

//Se lee la función de activación, puede ser función Sigmoidal (RRS) o
//Hiperbólica (RRH)
Leer(funcion_activacion)

/*
*Se crea un arreglo para las neuronas, donde el índice indica el número de la neurona
*/
neurona[numero_neuronas]

/*
*Se crea una matriz para las conexiones de cada neurona con todas, incluso con ella
*misma si en caso es una neurona de contexto o salida
*/
conexion[numero_neuronas] [numero_neuronas]

/*
*Cada conexión tiene una salida dinámica del sistema, que es necesario para el
*aprendizaje en tiempo real
*/
salida_sistema[numero_neuronas] [numero_neuronas]

//Se lee el número de ciclos para el aprendizaje de la red
Leer(numero_ciclos)

/*
*Según la validación cruzada dividimos la data en seis bloques, para ello creamos
*una matriz donde las columnas indican el número de bloque y las filas los patrones
*que pertenecen a ese bloque
*/
bloque[6][numero_patrones]

//recorremos los seis ciclos de validación cruzada
contador_cross = 1
Mientras(contador_cross ≤ 6)
Inicio mientras para validación cruzada
```




```
//seleccionamos el bloque para validación y los bloques para aprendizaje
Para(i = 1;i ≤ 6;i++)
Inicio para
  Si(i == contador_cross)
  Entonces
    patrones_para_validacion = patrones_para_validacion +
    Contar(bloque[i][numero_patrones])
  Caso Contrario
    patrones_para_aprendizaje = patrones_para_aprendizaje +
    Contar(bloque[i][numero_patrones])
Fin para

//se establece la razón de aprendizaje

/*
*Se inicializa los pesos de las conexiones y las salidas dinámicas del sistema
*/
Para(i = 1; i ≤ numero_neuronas; i++)
Inicio para
  Para(int j = 1; j ≤ numero_neuronas;j++)
  Inicio para
    conexion[i][j] = random
    salida_sistema[i][j] = 0
  Fin para
Fin para

//Se recorre el número de ciclos establecido
contador_tiempo = 1
Mientras(contador_tiempo ≤ numero_tiempos)
Inicio mientras para tiempo

  //inicializamos el error cuadrático medio de predicción
  ECMP = 0

  //recorremos los patrones de aprendizaje
  contador_patron = 1
  Mientras(contador_patron ≤ patrones_para_aprendizaje)
  Inicio mientras para patrones

    /*
    *Las neuronas de entrada reciben los datos de los patrones y son
    *los 11 primeros objetos del arreglo neurona
    */

    Para(i = 1;i ≤ 11;i++)
    Inicio para
      neurona[i] = estudiante[contador_patron][i]
    Fin para

    /*
    *Se calcula las activaciones de las neuronas en base a la función
    *de activación seleccionada y a la entrada efectiva, todas las
    *neuronas están conectadas unas con otras.
    *Las neuronas de entrada no tienen conexiones a sí mismas, pero
    *las neuronas de contexto o salida si las tienen
    */
    Para(i = 11; i ≤ numero_neuronas; i++)
    Inicio para
      suma = 0
```



```
/*
*Se calcula la entrada efectiva
*/
Para(int j = 1; j ≤ numero_neuronas; j++)
Inicio para
    suma = suma + (neurona[j] * conexion[j][i])
Fin para

/*
*La entrada efectiva se procesa con la función seleccionada, el
*resultado de este proceso es la salida de la neurona i
*/
neurona[i] = f(suma)

/*
*Si la neurona actual es la última es la de salida por lo tanto
*también es la salida de la red, en tal caso se calcula el error
*cometido respecto a la situación real del estudiante
*/
Si(i==numero_neuronas)
Entonces
    error = Estudiante[contador_patron][12] - neurona[i]
Fin para

/*
*Se calcula las salidas dinámicas del sistema para cada conexión
*/
Para(i = 12; i ≤ numero_neuronas; i++)
Inicio para
    Para(j = 1; j ≤ numero_neuronas; j++)
    Inicio para
        Para(k = 12; k ≤ numero_neuronas; k++)
        Inicio para
            Suma_sistema [j][i] = Suma_sistema [j][i] +
            (conexion[k][i] * salida_sistema + neurona[j])
        Fin para
    Fin para
Fin para

/*
*Se actualiza las salidas del sistema para el tiempo actual, en base
*a la derivada de la función de activación seleccionada
*/

Para(i = 12; i ≤ numero_neuronas; i++)
Inicio para
    Para(j = 1; j ≤ numero_neuronas; j++)
    Inicio para
        salida_sistema[j][i] = neurona[i]

/*
*Se actualiza la conexiones en base a la salida dinámica
*del sistema, la razón de aprendizaje y el error cometido
*por la red para el tiempo actual o patrón actual
*/
conexion[j][i] = conexion[j][i] + (razon_aprendizaje, error,
salida_sistema[j][i])
```

```
Fin para
Fin para

error_suma = error_suma +  $error^2$ 

/*
*Se calcula el error cuadrático medio del aprendizaje para el
*tiempo actual
*/
ECMP_aprendizaje = (1/ contador_ tiempo) * error_suma

/*
*Se realiza la validación con los patrones del bloque seleccionado
*para esta etapa
*/
contador_patron_validacion = 1

Mientras(contador_patron_validacion ≤ patrones_para_validacion)
Inicio mientras

Para(i = 1; i ≤ numero_neuronas; i++)
Inicio para
suma = 0
/*
*Se calcula la entrada efectiva
*/
Para(int j = 1; j ≤ numero_neuronas; j++)
Inicio para
suma = suma + (neurona[j] * conexion[j][i])
Fin para

/*
*La entrada efectiva se procesa con la función seleccionada,
*el resultado de este proceso es la salida de la neurona i
*/
neurona[i] = f(suma)

/*
*Si la neurona actual es la última es la de salida por lo tanto
*también es la salida de la red, en tal caso se calcula el error
*cometido respecto a la situación real del estudiante
*/
Si(i==numero_neuronas)
Entonces
error = Estudiante[contador_patron][12] - neurona[i]
Fin para
suma_error = suma_error +  $error^2$ 

Fin mientras

/*
*Calculamos el error cuadrático medio de predicción en la
*validación
*/
ECMP_validacion = (1/ patrones_para_validacion) * suma_error
```



```
//Registramos en la base de datos los siguientes resultados  
Registramos(contador_cross, contado_tiempo, ECMP_aprendizaje,  
ECMP_validacion,función_activacion)
```

```
contador_tiempo = contador_tiempo + 1
```

```
Fin mientras para tiempos
```

```
contador_cross = contador_cross + 1
```

```
Fin mientras para validación cruzada
```

Fin del Algoritmo



5.3. Resultado de las mejores experimentaciones de las redes neuronales artificiales

Tabla 19: Resultados de las experimentaciones de las Redes Neuronales artificiales en el estudio comparativo

Red neuronal artificial	Tipo de red neuronal artificial	Método de aprendizaje	Función de activación	Numero de neuronas	Numero de ciclo / Instante de tiempo	Razón de aprendizaje	Mínimo error cuadrático medio de predicción	Nivel de eficiencia (%)
PMS	Perceptrón Multicapa	Retropropagación	Sigmoidal	500	278	0,00001	0,084697	91,54%
PMH			Tangente hiperbólica	400	325	0,00001	0,085021	91,49%
BRG	Red Neuronal de Base radial	Hibrido	Gaussiana	12	88	0,0001	0,052252	94,77%
BRC			Inversa cuadrática	12	113	0,0001	0,069919	93,01%
RRS	Red Neuronal Recurrente	Recurrente en Tiempo Real	Sigmoidal	10	55600	0,001	0,038311	96,17%
RRH			Tangente hiperbólica	10	300	$1 * 10^{-11}$	0,11859	88,14%

Nota. En esta tabla se detalla los resultados de los mejores escenarios de las experimentaciones de las Redes Neuronales Artificiales. Fuente: propia

5.4. Resultado del estudio comparativo de redes neuronales artificiales

En base a los resultados que se muestran en la tabla comparativa de los mejores escenarios de cada red neuronal artificial (*Ver tabla 19*). Se puede afirmar que la red que alcanzó un mayor nivel de eficiencia fue RRS.

En la *ilustración 51* se puede observar los resultados obtenidos de 20 patrones utilizados para validación de la red neuronal artificial RSS, donde la evolución de los resultados de las redes neuronales artificiales siguen la tendencia de los resultados esperados los cuales son las situaciones reales de los estudiantes de las instituciones educativas, las pequeñas diferencias son las causantes del error cuadrático medio calculado, el cual fue 0,038311 y por ese motivo su nivel de eficiencia alcanzó un 96,17% en la identificación de violencia escolar.

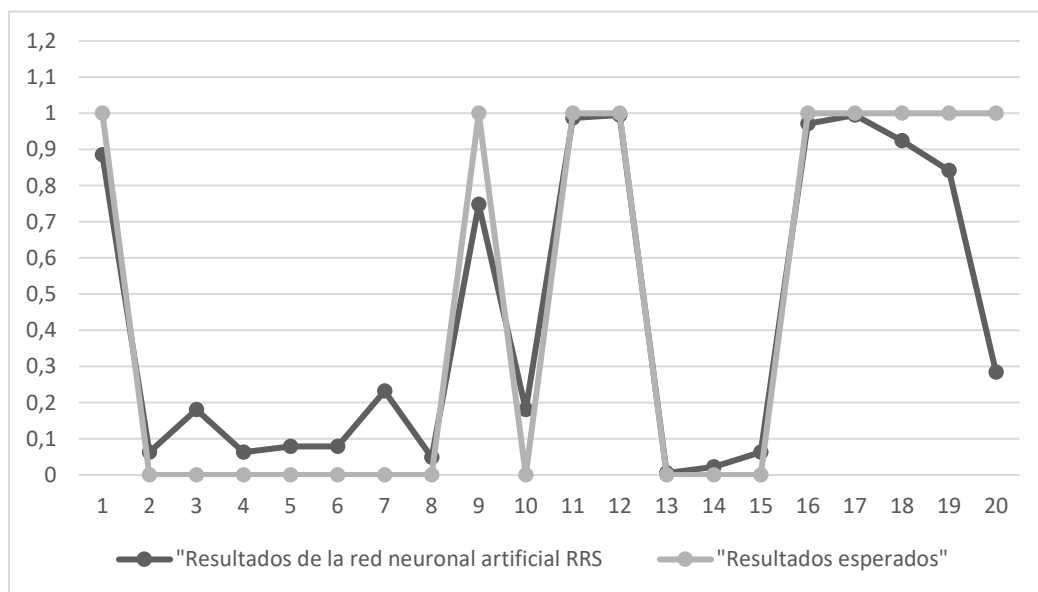


Ilustración 51: Resultados de RSS vs Resultados esperados

Fuente: propia

CAPÍTULO VI: DISCUSIÓN

En la presente tesis se realizó el estudio comparativo de redes neuronales artificiales aplicadas a la identificación de violencia escolar, en base a esto se planteó la hipótesis: “El estudio comparativo permite determinar la red neuronal artificial que tiene un mayor nivel de eficiencia en la identificación de violencia escolar en las instituciones educativas”. Esta hipótesis fue validada por los resultados obtenidos de las experimentaciones de las redes neuronales artificiales observados en el capítulo V, donde el nivel de eficiencia alcanzado por la red neuronal artificial RRS fue el mejor.

Las dos modificaciones explicadas en el capítulo IV, donde se detalla el remplazo del método no supervisado K-Medias por su extensión K-Modas y el cálculo del índice de disimilitud Jaccard en vez del cálculo de las distancias euclidianas, son aceptables en base a los resultados obtenidos de las experimentaciones de las redes de base radial con función Gaussiana (RBG) y con función Inversa cuadrática (RBC) que alcanzaron un 94,77% y 93,01% de nivel de eficiencia respectivamente en la identificación de violencia escolar.

En las experimentaciones de las redes PMS y PMH que comparten el mismo tipo de red y método de aprendizaje, se presentó un inconveniente, pues el algoritmo no llegaba a generalizar los patrones aprendidos en las pruebas realizadas, lo que ocasionaba que el error de entrenamiento, así como el de validación sean demasiados altos, este caso se repetía con pruebas de un rango de 1 a 22 capas ocultas y 1 a 22 neuronas en cada capa, entonces se descubrió que al realizar una prueba con una sola capa oculta y con un número de neuronas mayor e igual a 100 neuronas la red se estabilizaba y de esa forma lograba generalizar los patrones de entrada, consecuentemente se estableció que el número de neuronas optimas está entre 300 a 600 neuronas. En los escenarios realizados en las experimentaciones de PMS y PMH fueron con 500 y 400 neuronas respectivamente que se lograron mayores niveles de eficiencia. Con respecto a la experimentación de las redes de tipo base radial con método de aprendizaje Híbrido BRG y BRC, se realizaron procesos de prueba y error utilizando de 1 a 20 neuronas inicialmente, lográndose obtener en un rango de 11 a 14 neuronas mejores resultados, razón por la cual se consideró este último rango para la experimentación de los cuatro escenarios tanto para BRG como para BRC, y en ambos casos se hallaron mejores resultados con 12 neuronas. En cuanto las redes recurrentes RRS y RRH se estableció un rango de 9 a 12 neuronas para sus cuatro escenarios de experimentación, esto se estableció en base a los



procesos de prueba y error realizados con un rango de 2 a 20 neuronas. Para los cuatro escenarios realizados en cada experimentación tanto de RRS como en RRH se halló que en ambas experimentaciones 10 neuronas recurrentes eran suficientes para lograr alcanzar sus mejores resultados. Entonces se afirma que realizar un proceso de “prueba y error” es la mejor manera de determinar los mejores indicadores para una red neuronal artificial con un alto nivel de eficiencia en la predicción de la problemática que se plantea.

La red neuronal de tipo Recurrente, con método de aprendizaje Recurrente en Tiempo Real y función de activación Sigmoidal (RRS), como se observa en los resultados de la investigación alcanzó un 96,17% de nivel de eficiencia en la identificación de violencia escolar, siendo el mayor nivel de todas las redes evaluadas. De acuerdo a la teoría una red recurrente presenta características importantes como por ejemplo el aumento de pesos asociados a sus conexiones por las llamadas conexiones recurrentes, complicando de esa forma su aprendizaje, pero de igual forma representando mejor los patrones aprendidos, alcanzado así los mejores resultados en su validación.

En el Antecedente “Comparación de Redes Neuronales aplicadas a la predicción de Series de Tiempo” se afirma que existe un 95% de certeza en decir que el nivel de efectividad de una red de tipo perceptrón multicapa frente a una de base radial es similar para la predicción de series de tiempo. En la presente tesis no se calculó esta certeza de igualdad entre los resultados de las experimentaciones de las redes neuronales artificiales, pero si se puede afirmar que los resultados de las redes de tipo base radial frente a los de tipo perceptrón multicapa obtuvieron mayores niveles de eficiencia en la identificación de violencia escolar.

En el antecedente “Comparison of artificial neural network models for hydrologic predictions at multiple gauging stations in an agricultural watershed”; los desarrolladores de esta investigación afirman que la red de tipo perceptrón multicapa es más eficiente en la predicción hidrológica frente a una de base radial, en cambio en el estudio comparativo de las redes neuronales artificiales realizada en la presente tesis se demostró que las redes de base radial son más eficientes en la identificación de violencia escolar frente a las de tipo Perceptrón Multicapa, por alcanzar un menor error cuadrático medio en su etapa de validación. Es importante mencionar que las características de los datos de entrada y los patrones de las redes establecidas en el antecedente son diferentes a los que se tomaron en cuenta en esta tesis y eso puede ser la causa de la variación del desempeño de los modelos.



En el antecedente “Redes neuronas recurrentes para el reconocimiento de patrones temporales” el investigador afirma que los resultados obtenidos son mejores en la etapa de validación que en las etapas de aprendizaje de sus modelos recurrentes, es algo que también se repitió en la presente tesis pues en las redes neuronales artificiales experimentadas se calculó un menor error cuadrático medio en las etapas de validación que en las de aprendizaje.

GLOSARIO

En este glosario se presentan las palabras más importantes de la presente investigación.

Adaptación: Es entendido como la acción y el efecto de adaptar o adaptarse (Real Academia, 2014), el cual es el verbo de acomodar o ajuste de algo respecto a otra cosa. En el contexto de redes neuronales artificiales, se dice que estas tienen la capacidad de aprendizaje en las que ajustan los pesos de sus conexiones (sinapsis), siendo capaces de adaptarse a una situación y lográndose los resultados esperados.

Algoritmo: Conjunto prescrito de reglas o instrucciones bien definidas para la resolución de un problema. (López, y otros, 1998)

Amortiguamiento: Disminución progresiva, en el tiempo, de la intensidad de un fenómeno periódico (Real Academia, 2014). En el proceso de aprendizaje supervisado de la red neuronal artificial se da el amortiguamiento de los errores cometidos en un ciclo de aprendizaje o instante de tiempo determinado.

Amplitudes: Extensión, dilatación (Real Academia, 2014). En el contexto del aprendizaje no supervisado K-medias, son las distancias entre los centros de las clases.

Aprendizaje no supervisado: En este aprendizaje no se necesita de un profesor externo que determine el aprendizaje. La red modificara los valores de los pesos a partir de información interna. Cuando se utiliza el aprendizaje no supervisado la red trata de determinar características de los datos del conjunto de entrenamientos. Por ello a este tipo de modelos también se les conoce como sistemas autoorganizados. (Pedro Isasi Viñuela, 2004)

Aprendizaje Supervisado: Para este tipo de aprendizaje, se dice que hay un profesor externo encargado de determinar si la red se está comportando de forma adecuada, mediante la comparación entre la salida producida y la esperada, y de actuar en consecuencia modificando apropiadamente los valores de los pesos. (Pedro Isasi Viñuela, 2004)

Axón: Fibras fibrosas que transmite los impulsos nerviosos del cuerpo celular al exterior. (López, y otros, 1998)



Base de datos: Conjunto de datos operativos a los que acceden programas de aplicación o los usuarios. Una base de datos es un almacén de información en forma de datos de diferentes tipos. (López, y otros, 1998)

Capa de entrada: En esta capa se encuentran las neuronas que se encargan de recibir las señales del exterior.

Capa de salida: En esta capa se encuentran las neuronas que se encargan de enviar la salida de la red neuronal artificial al exterior.

Capa oculta: Las neuronas de esta capa realizan el procesamiento de los patrones.

Ciclos de aprendizaje: Es el número de iteración en el proceso de aprendizaje en redes neuronales artificiales de tipo Perceptrón Multicapa y Redes de Base Radial.

Dendritas: Prolongaciones cortas de las células que reciben los impulsos nerviosos del exterior. (López, y otros, 1998)

Disimilitud Jaccard: Es un coeficiente que calcula la diferencia entre dos objetos binarios.

Distancia euclidiana: Es la distancia clásica como la longitud de una recta que une dos puntos en el espacio. (José Hernández, José Ramírez, Cesar Ferri, 2004)

Error cuadrático medio: Este error es igual a la sumatoria de las diferencias de las salidas de una red neuronal artificial y los resultados esperados elevada al cuadrado. En esta investigación es el criterio de evaluación para la variable dependiente que es el nivel de eficiencia en la identificación de violencia escolar.

Error de aprendizaje: En esta investigación es el error cuadrático medio cometido por una red neuronal artificial en su proceso de aprendizaje.

Error de validación: En esta investigación es el error cuadrático medio cometido por una red neuronal artificial en su proceso de validación.

Escenarios: En esta investigación en una experimentación de una red neuronal artificial se realizan cuatro escenarios en estos se experimentan un diferente número de neuronas, y en algunos casos se varió la razón de aprendizaje.

Estado estable: En esta investigación consideramos que la red neuronal artificial está en estado estable cuando llega a generalizar los patrones aprendidos.



Experimentación: En esta investigación es el proceso de experimentar diferentes escenarios con una red neuronal artificial.

Función de activación: actualiza el nuevo nivel de activación basándose en la entrada efectiva y la activación anterior de la neurona. (Cruz, 2010)

Instante de tiempo: En Redes de tipo Recurrente es el intervalo en un proceso, en dicho intervalo una red neuronal artificial proceso un patrón del conjunto de datos.

K-medias: Se trata de un método de agrupamiento por vecindad en el que se parte de un número determinado de prototipos y de un conjunto de ejemplos a agrupar, sin etiquetar. Es el método más popular de los métodos de agrupamiento denominados “por partición”. La idea de K-medias es situar a los prototipos o centros en el espacio, de forma que los datos pertenecientes al mismo prototipo tengan características similares. (José Hernández, José Ramírez, Cesar Ferri, 2004)

K modas: Es una versión modificada de K-medias, este realiza una simple tarea haciendo coincidir la medida de desemejanza para variables categóricas, en este caso calcula la moda en lugar de la media. Utiliza un método basado en frecuencia para actualizar las modas en el proceso de agrupamiento, para minimizar la desemejanza entre dichas modas y los patrones de su agrupación. Estas modificaciones permiten al algoritmo agrupar eficientemente grandes categorías conjuntos de datos del mundo real. (Huang, 2003)

Media: Es la medida que mejor representa el valor en torno al cual se mueven todos los valores de una muestra. El valor de la media es el resultado de sumar todos los datos de la muestra y esto dividirlo por el número total de los datos. (López, y otros, 1998)

Moda: Es el dato que dentro de la tabla estadística presenta la mayor frecuencia absoluta. La moda se puede hallar tanto de datos cualitativos como de datos cuantitativos. (López, y otros, 1998)

Métodos de aprendizaje: Nos referimos al algoritmo de aprendizaje utilizado por una red neuronal artificial. Dentro del aprendizaje automático existen muchos algoritmos de aprendizaje de tipo supervisado como el método de Retropropagación y Recurrente en Tiempo Real, y no supervisado que no fueron considerados en esta investigación. También existen de tipo mixto como es el caso de método Híbrido que fue considerado en la presente tesis.



Neurona: Célula de tejido nervioso. Su citoplasma es ramificado, el núcleo es de gran tamaño. Sus prolongaciones más cortas se denominan dendritas y las otras más largas se conocen como axones. Las dendritas reciben los impulsos que llegan del exterior, transmitiéndolos hasta el cuerpo neuronal. Por su parte, los axones transmiten los impulsos nerviosos desde el cuerpo celular hasta el exterior. (López, y otros, 1998)

Neurona Artificial: Es un elemento que posee un estado interno, llamado nivel de activación, y recibe señales que le permiten, en su caso, cambiar de estado. Las neuronas poseen una función que les permite cambiar el nivel de activación a partir de las señales que reciben; a dicha función se la denomina función de transición de estado o función de activación. (Pedro Isasi Viñuela, 2004)

Nivel de eficiencia en la identificación de violencia escolar: En esta investigación es el nivel de eficiencia alcanzado por una red neuronal artificial en la identificación de violencia escolar, esto en base al error cuadrático medio cometido en su etapa de validación.

Pseudocódigo: El pseudocódigo nació como un lenguaje similar al inglés y era un medio de representar básicamente las estructuras de control de programación estructurada que se verán en capítulos posteriores. Se considera un primer borrador, dado que el pseudocódigo tiene que traducirse posteriormente a un lenguaje de programación. El pseudocódigo no puede ser ejecutado por una computadora. La ventaja del pseudocódigo es que en su uso, en la planificación de un programa, el programador se puede concentrar en la lógica y en las estructuras de control y no preocuparse de las reglas de un lenguaje específico. Es también fácil modificar el pseudocódigo si se descubren errores o anomalías en la lógica del programa, mientras que en muchas ocasiones suele ser difícil el cambio en la lógica, una vez que está codificado en un lenguaje de programación. Otra ventaja del pseudocódigo es que puede ser traducido fácilmente a lenguajes estructurados como Pascal, C, FORTRAN 77/90, C++, Java, C#, etc. (Joyanes Aguila, 2008)

Punto óptimo: En la experimentación de las redes neuronales artificiales en esta investigación, llamamos “punto óptimo” al ciclo donde se halló el menor error cuadrático medio de validación, este representa un bloque de validación, luego se compara con los demás puntos óptimos de otros bloques para determinar el mejor bloque, que representara un escenario.



Razón de aprendizaje: Es la razón o tasa del aprendizaje que influye en la magnitud del desplazamiento en la superficie del error en el aprendizaje de las redes neuronales artificiales. (Pedro Isasi Viñuela, 2004)

Redes Neuronales Artificiales: Son un método de aprendizaje cuya finalidad inicial era la de emular los procesadores biológicos de información. Las RNA parten de la presunción de que la capacidad humana de procesar información se debe a la naturaleza biológica de nuestro cerebro. Por tanto, para imitar esta característica debemos estudiar y basarnos en el uso de soportes artificiales similares a los de nuestro cerebro. (José Hernández, José Ramírez, Cesar Ferri, 2004)

Resultados esperados: En esta investigación nos referimos a la situación real de los estudiantes, que nos indica si en realidad cada uno sufre o no de violencia escolar.

Sinapsis: Zona de contacto entre dos neuronas a través de la cual tiene lugar la transmisión del impulso nervioso. (López, y otros, 1998)

Tipos de redes: Nos referimos al tipo de red existen muchos en las redes neuronal artificiales, pero los que consideramos en esta investigación son: Perceptrón Multicapa (PM), Redes Neuronales de Base Radial (BR) y Redes Neuronales Recurrentes (RR)

Validación cruzada: Es un método que consiste en dividir el conjunto de la evidencia en k subconjuntos disjuntos de similar tamaño. El procedimiento se repite k veces, utilizando siempre un subconjunto diferente para estimar el error de la muestra parcial (José Hernández, José Ramírez, Cesar Ferri, 2004). En esta investigación utilizamos este método dividiendo en 6 subconjuntos los patrones de entrada, en cada recorrido hicimos que un subconjunto diferente sea utilizado para la validación de la red neuronal artificial y los demás para su aprendizaje.



CONCLUSIONES

1. El diseño de una base de datos para las redes neuronales artificiales permitió principalmente contar con el conjunto de datos que fueron los patrones de entrada a las redes neuronales artificiales en sus etapas de aprendizaje y validación, la base de datos también permitió el registro de los resultados obtenidos en dichas etapas.
2. Las redes neuronales artificiales cuentan con una serie de variantes que pueden afectar su desempeño, frente a esto la mejor manera de establecer los parámetros correctos para una arquitectura de red óptima son los procesos de prueba y error, y es que hasta el momento en la teoría no se tienen un método cien por ciento confiable. Por ello, el desarrollo de una red neuronal artificial es una tarea complicada y que requiere mucho tiempo, pues se deben realizar varias experimentaciones para encontrar los indicadores con los que se logran mejores resultados.
3. En base a los resultados obtenidos de las experimentaciones de las redes RBG y RBC donde se implementó el método K-Modas para el aprendizaje no supervisado en lugar del comúnmente utilizado K-Medias y el coeficiente de disimilitud Jaccard para calcular la distancia entre vectores binarios en lugar de utilizar la distancia euclidiana, se puede afirmar que estas modificaciones dieron resultados aceptables.
4. En base a la diferencia de resultados entre RRS que es una Red Recurrente que utiliza función Sigmoidal y RRH con función Tangente Hiperbólica, donde el primero fue el que alcanzo un mayor nivel eficiencia y el segundo el que alcanzo el menor nivel de eficiencia en la identificación de violencia escolar entre todas las redes neuronales artificiales evaluadas en el estudio comparativo, se puede afirmar que la modificación de la función de activación influye significativamente en el desempeño de una red neuronal artificial.
5. La red neuronal artificial que alcanzó un mayor nivel de eficiencia en la identificación de violencia escolar fue la Red Neuronal Recurrente, con método de aprendizaje Recurrente en Tiempo Real y función de activación Sigmoidal (RRS).
6. Las redes neuronales de tipo Recurrente al añadir más conexiones y por ende más pesos sinápticos ajustables entre las neuronas de su red complican su aprendizaje, pero a la vez mejoran su capacidad de representación de los patrones aprendidos.
7. Se desarrolló un prototipo web, para observar el funcionamiento de la red neuronal artificial RRS en un entorno real. (*Ver Anexo B*)



RECOMENDACIONES

1. La información fue recolectada de las instituciones educativas de la provincia de Urubamba en función al cuestionario utilizado por las mismas, con el que ellos pretenden saber la situación respecto a la existencia de violencia escolar de sus estudiantes en base a preguntas cerradas. Sería importante conocer la relevancia de cada una de estas preguntas en la tipificación de violencia escolar a través de una investigación en el campo psicológico.
2. La utilización del método k-modas en las dos redes neuronales de base radial dieron buenos resultados, por ello se recomienda considerar la misma modificación en una futura investigación, incluso para solucionar una problemática diferente, experimentado y comprobando los resultados obtenidos.
3. Se recomienda en una siguiente investigación de esta misma línea considerar la mayor cantidad de indicadores en el desarrollo de redes neuronales artificiales y luego experimentar con todas ellos, conociendo de esa forma los niveles de desempeño que estos ocasionan.
4. Las redes neuronales artificiales desarrolladas en esta investigación únicamente alcanzan los desempeños mostrados en la identificación de violencias escolar, razón por la cual si estas serían reutilizadas en otras problemáticas el desempeño no sería el mismo, porque se variarían los patrones de entrada, el número de salidas o la característica de la información recolectada sería distinta.

**BIBLIOGRAFÍA**

- Aristidis Likas. (1 de Abril de 2001). Probability density estimation using artificial neural networks. *Computer Physics Communications*, 135(2), 167-175.
- Cruz, P. P. (2010). *Inteligencia Artificial con aplicaciones a la ingeniería*. Mexico: Alfaomega.
- Darwin Mercado Polo, L. P. (15 de Mayo de 2015). Comparison of Neural Network applied to prediction of Time Series. *Prospect*, 13, 88-95.
- Graupe, D. (2007). *Principles of Artificial Neural Networks*. Chicago - Estados Unidos: World Scientific.
- Han, J., Kamber, M., & Pei, J. (2012). *Data Mining Concepts and Techniques*. Illinois: ELSEVIER.
- Haykin, S. (2009). *Neural Networks and Learning Machines*. New Jersey: Pearson Education.
- Huang, Z. (2003). A Note on K-modes Clustering. Hong Kong: The University of Hong Kon.
- José Hernández, José Ramírez, Cesar Ferri. (2004). *Introducción a la minería de datos*. Madrid: PEARSON EDUCACIÓN.
- López, A., Manso, J., Méndez, M. L., Calvo, A., Saavedra, M., Zamora, M. Á., (1998). *Enciclopedia Estudiantil*. (A. Dorado, Ed.) Madrid, España: CULTURAL S.A.
- Ministerio de Educación del Perú. (2017). *Estadísticas de violencia escolar en las instituciones educativas de Perú*. Obtenido de MINEDU : <http://www.siseve.pe/Seccion/Estadisticas>
- Minsky, M., & Papert, S. (1969). Perceptrons. *Perceptrons*. Oxford, England.
- Mutlu, E., I. Chaubey, H., & Bajwa, H. y. (2008). Comparison of artificial neural network models for hydrologic predictions at multiple gauging stations in an agricultural watershed. *Hydrological processes*, 22, 5097-5106.
- Otzen, T., & Manterola, C. (2017). Técnicas de muestreo sobre una población a estudio. *Int. J. Morphol*, 35(1), 227-232.
- Palmer Pol, A., Montañó Moreno, J., & Calafat Far, A. (2000). Predicción del consumo de éxtasis a partir de redes neuronales artificiales. *Adicciones*, 12, 1.
- Pedro Isasi Viñuela, I. G. (2004). *Redes Neuronales Artificiales un enfoque practico*. Madrid : Pearson Educación.
- Pérez Segovia, F. J. (Febrero de 1992). Redes de neuronas recurrentes para el reconocimiento de patrones temporales (Tesis doctoral). Madrid, España: Universidad Politecnica de Madrid.



- Pol Plamer, A., & Montaña Moreno, J. J. (2002). Redes neuronales artificiales aplicadas al análisis de supervivencia: un estudio comparativo con el modelo de regresión de Cox en su aspecto predictivo. *Psicothema*, 14(3), 630-636.
- Rosenblatt, F. (1962). *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*.
- Sampieri, R. H. (2014). *Metologias de la Investigacion* . México: McGRAW-HILL / INTERAMERICANA EDITORES.
- Sergio Valero Verdú, C. S. (2012). *Aplicación de un modelo de red neuronal no supervisado a la clasificacion de consumidores electricos*. San Vicente (Alicante): Club Universitario .
- Vilca Laqui, W. F. (2010). Aplicación de Redes Neuronales Artificiales a la Modelización y Previsión de Caudales Medios Mensuales del Río Huancané. *Peruana Geo-Atmosférica RPGA* , 2, 30-44.
- Real Academia Española. (2014). *Diccionario de la lengua española* (23.^aed.). Consultado en <http://www.rae.es/>



ANEXOS

Anexo A: Cuestionario utilizado para la recolección de datos

N° de Orden	Nombres del alumno	edad													
			vive con sus padres	vive solo con mama	vive solo con papa	vive con otros	asiste descuidadamente	asiste a la institución sin tomar sus alimentos	manifiesta desinterés por el estudio	agreden a sus compañeros	reacciona pasivamente ante los maltratos	utiliza tonos y gestos inapropiados para hacerse escuchar	no manifiesta maltrato a su integridad	Existe violencia escolar en el estudiante	

Anexo B: Prototipo Web

En el prototipo web desarrollado se implementó la red neuronal artificial RRS y se realizó una prueba en la cual se registraron 20 estudiantes por medio del cuestionario que se observa en la siguiente imagen:

Luego de ingresar la información de los estudiantes a la plataforma web, la red RRS automáticamente obtuvo los resultados que a nivel de porcentajes identifican en probabilidades la existencia de violencia escolar en cada estudiante, como se observa en la siguiente imagen:

Codigo	Nombre del Alumno	Edad	Pregunta 1	Pregunta 2	Pregunta 3	Pregunta 4	Pregunta 5	Pregunta 6	Pregunta 7	Pregunta 8	Pregunta 9	Pregunta 10	Pregunta 11	Resultado RRS	Probabilidad de violencia escolar
1	C. M. Alonzo	7	no	no	no	si	si	si	si	si	no	si	no	0.8868	88.68%
2	M. Q. Nayda	6	si	no	no	no	no	no	no	no	no	no	si	0.0633	6.33%
3	F. T. Sherjhy V.	10	no	si	no	no	no	no	no	no	no	no	si	0.1814	18.14%
4	H. P. Yulissa V.	11	si	no	no	no	no	no	no	no	no	no	si	0.0633	6.33%
5	A. P. Luis A.	9	si	no	no	no	no	no	no	no	si	no	si	0.0797	7.97%
6	A. C. Natan E.	6	si	no	no	no	no	no	no	no	si	no	si	0.0797	7.97%
7	L. C. Magaly	12	no	no	no	si	si	no	si	no	si	no	si	0.0784	7.84%
8	A. H. Jhosep	11	no	no	no	si	no	no	si	no	si	no	si	0.0143	1.43%
9	Z. G. Richar	12	no	no	no	si	si	si	si	no	si	si	no	0.7526	75.26%
10	P. V. Medaty	10	no	si	no	no	no	no	no	no	no	no	si	0.1814	18.14%
11	I. C. Yosmar	13	no	no	si	no	si	no	si	no	no	si	no	0.9635	96.35%
12	C. J. Ramiro	12	no	si	no	no	si	no	si	si	no	si	no	0.9854	98.54%
13	C. Z. Anani	11	no	no	no	si	no	si	no	no	si	no	si	0.0179	1.79%
14	B. Q. Jhonathan S.	12	no	si	no	no	no	si	no	no	no	no	si	0.0725	7.25%
15	C. B. Markus E.	8	si	no	no	no	no	no	no	no	no	no	si	0.0633	6.33%
16	L. B. Yasmina	11	si	no	no	no	si	no	si	no	no	si	no	0.9080	90.80%
17	S. Y. Efrain	12	no	no	si	no	si	no	si	si	no	si	no	0.9848	98.48%
18	Q. V. Franklin H.	9	si	no	no	no	si	no	si	si	no	si	si	0.7749	77.49%
19	S. C. Nancy	13	no	si	no	no	no	si	no	si	no	si	no	0.9401	94.01%
20	S. C. Ubalde	12	si	no	no	no	si	si	no	no	si	no	no	0.5805	58.05%