

**UNIVERSIDAD NACIONAL JORGE BASADRE GROHMANN – TACNA**

**Facultad de Ingeniería**

**Escuela Profesional de Ingeniería en Informática y Sistemas**

**ANÁLISIS PREDICTIVO DEL RENDIMIENTO ACADÉMICO  
EN LOS ALUMNOS DE LA ESCUELA PROFESIONAL DE  
INGENIERÍA EN INFORMÁTICA Y SISTEMAS DE LA  
UNJBG, UTILIZANDO REDES NEURONALES,  
SEMESTRE 2017-I**

**TESIS**

Presentada por:

Bach. Silvana Beatriz Cabana Yupanqui

Para optar el Título Profesional de:

**INGENIERO EN INFORMÁTICA Y SISTEMAS**

**TACNA - PERÚ**

**2018**

**UNIVERSIDAD NACIONAL JORGE BASADRE GROHMANN - TACNA**  
**FACULTAD DE INGENIERÍA**  
**JURADO CALIFICADOR Y CALIFICACIÓN DE LA SUSTENTACIÓN DE TESIS**

**TESIS N°:** \_\_\_\_\_

**TÍTULO PROFESIONAL DE:**  
**Ingeniero en Informática y Sistemas**

La Secretaría Académica Administrativa de la Facultad de Ingeniería, por resolución de Facultad N° 04611-2017-FAIN/UNJBG, designó jurado para la sustentación oral de la Tesis titulada: ANÁLISIS PREDICTIVO DEL RENDIMIENTO ACADÉMICO EN LOS ALUMNOS DE LA ESCUELA PROFESIONAL DE INGENIERÍA EN INFORMÁTICA Y SISTEMAS DE LA UNJBG, UTILIZANDO REDES NEURONALES, SEMESTRE 2017-I.

**El mismo que está conformado por:**

Presidente: MSc. Edgardo Teófilo Valdez Cortijo

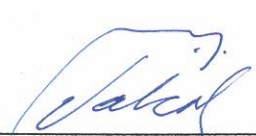
Secretario: MSc. Edgar Aurelio Taya Acosta

Vocal: Mgtr. Gianfranco Alexey Málaga Tejada

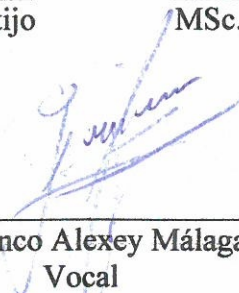
Para calificar la sustentación de la tesis en acto público el día 21 de diciembre del 2017, presentado por la Bachiller Silvana Beatriz Cabana Yupanqui, de la Escuela Profesional de Ingeniería en Informática y Sistemas.

El jurado Calificador en forma secreta e individual emitió su opinión sobre el tema de la tesis expuesta y procedió a obtener el promedio que arrojó el calificativo de aprobado con la nota de Dieciséis (16).

Para ratificar lo detallado firman:

  
\_\_\_\_\_  
MSc. Edgardo Teófilo Valdez Cortijo  
Presidente

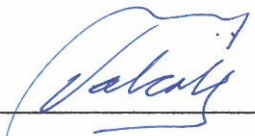
  
\_\_\_\_\_  
MSc. Edgar Aurelio Taya Acosta  
Secretario


  
\_\_\_\_\_  
Mgtr. Gianfranco Alexey Málaga Tejada  
Vocal

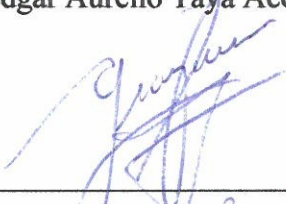
**UNIVERSIDAD NACIONAL JORGE BASADRE GROHMANN**  
**Facultad de Ingeniería**  
**ESCUELA PROFESIONAL DE INGENIERÍA EN INFORMÁTICA Y SISTEMAS**


**ANÁLISIS PREDICTIVO DEL RENDIMIENTO ACADÉMICO EN LOS  
ALUMNOS DE LA ESCUELA PROFESIONAL DE INGENIERÍA EN  
INFORMÁTICA Y SISTEMAS DE LA UNJBG, UTILIZANDO  
REDES NEURONALES, SEMESTRE 2017-I**

TESIS SUSTENTADA Y APROBADA EL 21 DE DICIEMBRE DEL 2017  
ESTANDO EL JURADO CALIFICADOR INTEGRADO POR:

Presidente :   
MSc. Edgardo Teófilo Valdez Cortijo

Secretario :   
MSc. Edgar Aurelio Taya Acosta

Vocal :   
Mgtr. Gianfranco Alexey Málaga Tejada

Asesor :   
Dr. Edwin Antonio Hinojosa Ramos

## **DEDICATORIA**

Dedicado a mis padres y mis hermanos  
que han estado conmigo superando toda adversidad.

## **AGRADECIMIENTO**

A mi asesor el Dr. Edwin Hinojosa Ramos por haberme acompañado en esta experiencia universitaria.

## CONTENIDO

DEDICATORIA	iv
AGRADECIMIENTO	v
CONTENIDO	vi
ÍNDICE DE TABLAS	ix
ÍNDICE DE FIGURAS	x
ÍNDICE DE ANEXOS	xi
RESUMEN	xii
INTRODUCCIÓN	1
CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA	2
1.1. Descripción del problema	2
1.1.1. Antecedentes del problema	2
1.1.2. Problemática de la investigación	3
1.2. Formulación del problema	4
1.2.1. Problema general	4
1.2.2. Problemas específicos	4
1.3. Justificación e importancia	5
1.4. Alcances y limitaciones	5
1.5. Objetivos	6

1.5.1.	Objetivo general	6
1.5.2.	Objetivos específicos	7
1.6.	Hipótesis	7
1.6.1.	Hipótesis global	7
1.6.2.	Hipótesis específicas	7
<b>CAPÍTULO II: MARCO TEÓRICO</b>		<b>9</b>
2.1.	Antecedentes del estudio	9
2.2.	Bases teóricas	13
2.2.1.	Rendimiento académico	13
2.2.2.	Predicción	14
2.2.3	Inteligencia artificial	15
2.2.4.	Redes neuronales	15
2.2.5.	Perceptrón multicapa	17
2.2.6.	Retropropagación	19
2.3.	Definición conceptual de términos	20
<b>CAPÍTULO III: MARCO METODOLÓGICO</b>		<b>23</b>
3.1.	Tipo y diseño de la investigación	23
3.2.	Población y muestra	24
3.3.	Variables	24
3.3.1.	Identificación de variables	24
3.3.2.	Definición de las variables	24

3.3.3.	Operacionalización de variables	25
3.3.4.	Clasificación de las variables	26
3.4.	Técnicas e instrumentos para la recolección de datos	27
3.5.	Procesamiento y análisis de datos	27
CAPÍTULO IV: RESULTADOS Y DISCUSIONES		29
4.1.	Resultados	29
4.1.1.	Construcción de red neuronal	29
4.2.	Validación	50
4.3.	Análisis de sensibilidad	55
4.4.	Predicción	57
4.2	Discusiones	58
CONCLUSIONES		60
RECOMENDACIONES		62
REFERENCIAS BIBLIOGRÁFICAS		62

## ÍNDICE DE TABLAS

Tabla 1: Áreas generales para exámenes de admisión fase I y II	39
Tabla 2: Áreas generales para examen de admisión CEPU	40
Tabla 3: Cantidad de datos de exámenes CEPU	40
Tabla 4: Cantidad de datos de exámenes Fase I y II	40
Tabla 5: Normalización de las variables de entrada de la red neuronal	42
Tabla 6: Normalización de las variables de salida de la red neuronal	42
Tabla 7: Data set normalizada para la red neuronal	43
Tabla 8: Errores en validación de red neuronal con topología 8:3:4:1	53
Tabla 9: Análisis de sensibilidad de las entradas de la red neuronal	56
Tabla 10: Predicción del rendimiento académico en ingresantes CEPU-I 2018	57

## ÍNDICE DE FIGURAS

Figura 1: Red neuronal de 3 capas	18
Figura 2: Perceptrón simple	29
Figura 3: Perceptrón multicapa	30
Figura 4: Función de activación sigmoide	31
Figura 5: Propagación hacia adelante	34
Figura 6: Retropropagación hacia atrás	36
Figura 7: Entradas y salidas de implementación de la red neuronal	44
Figura 8: Entrenamiento: error en validación y nro de épocas	47
Figura 9: Entrenamiento: iteraciones (épocas) vs error	48
Figura 10: Entrenamiento: salidas calculadas vs salidas reales	49
Figura 11: Validación: salidas de la red neuronal	50
Figura 12: Validación: medidas de error (suma, promedio, mínimo y máximo)	52
Figura 13: Topología 8:3:4:1	54
Figura 14: Validación: salida de la red neuronal vs valor real.	55
Figura 15: Análisis de sensibilidad en la red neuronal de topología 8:3:3:4	56

## ÍNDICE DE ANEXOS

Anexo 1: Matriz de consistencia	66
Anexo 2: Código main.cpp	67
Anexo 3: Entradas red neuronal en fase de entrenamiento	75
Anexo 4: Red neuronal en fase de validación	78

## **RESUMEN**

El propósito de la tesis denominada: “Análisis Predictivo del Rendimiento Académico en los Alumnos de la Escuela Profesional de Ingeniería en Informática y Sistemas de la UNJBG, utilizando Redes Neuronales, Semestre 2017-I” se centró en aprovechar la capacidad predictiva de las redes de neuronales artificiales para analizar el rendimiento académico en base a las puntuaciones por asignatura en el exámen de admisión a la Universidad.

El diseño de la investigación es no experimental descriptivo, se realiza la construcción de una red neuronal acorde a las especificaciones de las entradas y salidas de nuestro conjunto de datos que comprende de las respuestas correctas que tuvo en el Examen de Admisión y el promedio de notas del alumno en el primer ciclo de su carrera universitaria, basándonos en el aprendizaje supervisado con el algoritmo de Repropagación en perceptrón multicapa.

En los resultados de la investigación observamos un entrenamiento satisfactorio de la red neuronal artificial con el algoritmo supervisado de retropropagación donde se obtuvo error menor al 5 %, un error cuadrático medio de

6,2 % en la fase de validación de la red y en el análisis de sensibilidad de la red neuronal se tuvo que las áreas de Razonamiento Matemático, Aritmética y Algebra y Razonamiento Verbal influyen positivamente en un 0,59; 0,13 y 0,09 respectivamente al rendimiento académico en los alumnos del primer semestre de la Escuela Profesional de Ingeniería en Informática y Sistemas.

## **INTRODUCCIÓN**

El presente trabajo de investigación se ha estructurado de la siguiente manera: en el Capítulo I se describe la problemática con la que planteamos los objetivos e hipótesis; en el Capítulo II se integran las bases teóricas, definición de términos que se toman en cuenta para el desarrollo de la investigación; en el Capítulo III desarrolla el marco metodológico de la investigación; en el Capítulo IV se muestran los resultados obtenidos a partir de la validación de las hipótesis, se discuten los resultados obtenidos teniendo como soporte y análisis los antecedentes del presente estudio. Finalmente damos a conocer las conclusiones a las que se arribó en la presente investigación, luego se presentan las recomendaciones y la bibliografía que se consideró para el presente trabajo.

# **CAPÍTULO I**

## **PLANTEAMIENTO DEL PROBLEMA**

### **1.1. Descripción del problema**

#### **1.1.1. Antecedentes del problema**

La mejora de la calidad educativa es un motivo de preocupación para la universidad. Las investigaciones de indicadores de calidad relacionadas con los estudiantes hacen necesario el desarrollar herramientas para predecir en qué medida son capaces de alcanzar un buen rendimiento académico y terminar con éxito sus estudios. Hay muchos factores que pueden influir el desempeño de los estudiantes tales como: características personales, antecedentes socioeconómicos y culturales, motivación, conocimientos previos y aptitudes educativas.

El propósito de este estudio es explorar áreas que se tienen en el examen de admisión para la predicción del rendimiento académico en el primer semestre de los ingresantes a la Escuela Profesional de Ingeniería en Informática y Sistemas.

### **1.1.2. Problemática de la investigación**

El objetivo principal de las universidades es tener profesionales de calidad que pueden tener dificultad por los conceptos estudiados sin profundidad en la Educación Básica Regular. La modalidad de ingreso en la Universidad Nacional Jorge Basadre Grohmann es de exámenes estándar el cual está dividido en 4 canales (Ciencias de la Salud y Biomédicas; Ciencias, Ingenierías y Arquitectura; Ciencias Sociales y Humanidades; Ciencias Actuariales y Empresariales) donde las nuevas instancias de la Ley Universitaria que aboca al ingreso del alumnado sin una valla de nota aprobatoria, lo cual puede dificultar la calidad estudiantil. Nuestra universidad tiene una malla curricular inicial de estudios generales como también de asignaturas específicas de la carrera. Estando nuestra carrera orientada al área de ingeniería, se necesita en el primer semestre un mayor grado de conocimiento en áreas afines a esta.

Una red de neuronas artificiales es una herramienta que nace inspirada en el funcionamiento del cerebro humano que recibe continuamente miles de entradas y emite la respuesta más adecuada cada circunstancia. Por lo tanto, las redes neuronales están formados por procesos interconectados capaces de aprender mediante el ajuste de los parámetros que componen la red (pesos sinápticos).

Una preocupación de la Escuela Profesional de Ingeniería en Informática y Sistemas es poder dar educación de calidad a los estudiantes para que estos absorban todos los temas expuestos en cátedra y puedan tener un buen rendimiento académico en el primer semestre de estudios ya que esta es la base de los siguientes semestres a lo largo de su vida universitaria.

## **1.2. Formulación del problema**

Habiendo definido y delimitado nuestro estudio, podemos resumir nuestro problema mediante la siguiente proposición:

### **1.2.1. Problema general**

¿Se puede realizar un análisis predictivo del rendimiento académico de los alumnos de la Escuela Profesional de Ingeniería en Informática y Sistemas?

### **1.2.2. Problemas específicos**

- a. ¿Es posible entrenar una red neuronal para la predicción del rendimiento académico en los alumnos del primer semestre con un error menor al 5 %?
- b. ¿Es posible validar la red neuronal con error cuadrático medio menor al 10%?

- c. ¿Es posible analizar la predicción del rendimiento académico?

### **1.3. Justificación e importancia**

El disponer de un pronóstico adecuado el rendimiento académico en los alumnos del primer semestre en la carrera profesional ayudará en la toma de decisiones en el avance de su carrera de manera satisfactoria ya que se podrán hacer refuerzos y ayudar a los alumnos con más condiciones a tener un rendimiento académico malo o regular.

El método de las redes neuronales artificiales presenta un gran número de características semejantes a las del cerebro humano, extrayendo las características esenciales del grupo de datos. Las redes neuronales artificiales son metodologías que permiten hacer pronósticos donde hay cierto comportamiento no lineal, donde  $R$  es la dimensión de las entradas, las cuales prometen un buen desempeño en este campo de investigación.

### **1.4. Alcances y limitaciones**

Para esta investigación se tiene como datos las notas tipificadas en las áreas de conocimiento que maneja el temario del prospecto de admisión con respecto al Examen de Admisión de los alumnos ingresantes en el semestre 2016-I y 2017-I.

Se predijo el rendimiento académico con el modelo propuesto de los alumnos ingresantes a la Escuela Profesional de Ingeniería en Informática y Sistemas por el examen de admisión CEPU-I 2018.

Se utilizó datos de los ingresantes a la Escuela Profesional de Ingeniería en Informática y Sistemas de la UNJBG para el diseño de la red neuronal.

No se tomó en cuenta a los alumnos ingresantes por la modalidad de examen de Admisión Extraordinario.

Se cuenta con datos de los exámenes de admisión 2016 (CEPU-II, FASE II), 2017 y CEPU – I 2018 para la predicción.

## **1.5. Objetivos**

### **1.5.1. Objetivo general**

Realizar un análisis predictivo del rendimiento académico de los alumnos de la Escuela Profesional de Ingeniería en Informática y Sistemas.

### **1.5.2. Objetivos específicos**

- a. Entrenar una red neuronal con un error menor al 5 % para la predicción del rendimiento académico.
- b. Validar la red neuronal con un error menor al 10 % en la predicción del rendimiento académico.
- c. Analizar la predicción del rendimiento académico.

## **1.6. Hipótesis**

### **1.6.1. Hipótesis global**

$H_0$ : No es posible realizar un análisis predictivo del rendimiento académico.

$H_1$ : Es posible realizar un análisis predictivo del rendimiento académico.

### **1.6.2. Hipótesis específicas**

#### **Hipótesis específica 1**

$H_0$ : No es posible entrenar una red neuronal con un error menor al 5% para la predicción del rendimiento académico.

$H_1$ : Es posible entrenar una red neuronal con un error menor al 5% para la predicción del rendimiento académico.

### **Hipótesis específica 2**

H<sub>0</sub>: No es posible validar la red neuronal con un error menor al 10% en la predicción del rendimiento académico.

H<sub>1</sub>: Es posible validar la red neuronal con un error menor al 10% en la predicción del rendimiento académico.

### **Hipótesis específica 3**

H<sub>0</sub>: No es posible analizar la predicción del rendimiento académico.

H<sub>1</sub>: Es posible analizar la predicción del rendimiento académico.

## **CAPÍTULO II**

### **MARCO TEÓRICO**

#### **2.1. Antecedentes del estudio**

A nivel internacional se han encontrado varias investigaciones que buscan predecir mediante redes neuronales el rendimiento académico.

La investigación “Redes Neuronales para predecir el rendimiento académico de los alumnos ingresantes a la carrera de bioquímica de la FACENA-UNNE (Argentina) en función de sus conocimientos matemáticos previos” de los autores Barreto, López, Ramirez Arballo, Porcel, & Mata (2012) desarrollan modelos RNA para analizar la incidencia de los conocimientos matemáticos previos de los alumnos ingresando a la carrera de bioquímica, donde se analizaron los datos del estado académico de los alumnos y de sus conocimientos matemáticos previos, siendo obtenidos a partir de una evaluación diagnóstica al inicio del curso de nivelación de los ingresantes. Se diseñaron modelos de RNA de tipo perceptrón multicapa (PM) y función de base radial (FBR). Siendo variable independiente el número de ítems correctos en cada contenido del

diagnóstico. Y el rendimiento académico una variable dependiente la cual se midió en cada modelo con una variable dicotómica que toma el valor de 0 o 1, si el alumno aprobó los exámenes parciales respecto a los cursos de álgebra y geometría analítica. Los modelos PM y FBR permitieron obtener porcentajes de clasificación correcta total del 81,3 % y 78,8 % respectivamente.

Alcázar, Silva, & Fernández (2014) enfocan un “Sistema inteligente para perfilar la deserción en estudiantes universitarios de carreras técnicas” donde se ejecuta el análisis de asociación entre las variables mediante red neuronal artificial multicapa en la información histórica sociodemográfica y académica de 396 estudiantes del primer semestre. La red neuronal que modela la deserción se construyó a través del ajuste de los pesos de las variables mediante el algoritmo de aprendizaje retropropagación. El 68 % de la información disponible sirvió para construir la red y el 32 % para probar el acierto de su funcionamiento. El peso final de cada variable que contribuyó a explicar la deserción se repartió entre las materias reprobadas en el primer semestre, materias aprobadas en el primer semestre, nota de colegio categorizada, nota de admisión categorizada, edad, numero de materias tomadas en el primer semestre, tipo de colegio, sexo, donde el modelo generado permitió identificar a los estudiantes en riesgo de desertar y apoya la toma de decisiones para prevenir la deserción estudiantil en el primer año de su facultad.

Los autores Cataldi, Salgueiro, & Lage (2009) enfocan la “Predicción del rendimiento de los estudiantes y diagnóstico usando redes neuronales” donde se toman los datos de las evaluaciones parciales y a partir del análisis de los mismos se busca efectuar un diagnóstico con base en los errores cometidos. Para efectuar la predicción de los próximos errores se usa red neuronal y sobre esta base se le sugiere al estudiante una secuencia de ejercicios y problemas a fin de mejorar su producción y por lo tanto sus calificaciones.”

Gonzales (2008) propone la “Detección de alumnos en riesgo y medición de la eficiencia de centros escolares mediante redes neuronales” donde utilizan redes neuronales para intentar solucionar empíricamente tres problemas habituales en el campo de la economía de la educación: Detección de alumnos de alto riesgo de fracaso escolar, evaluación de la eficiencia relativa del centro educativos y evaluación de la eficiencia.

Existen investigaciones nacionales respecto al uso de redes neuronales artificiales para la predicción Acosta & Pizarro (2011) autores de “Predicción del rendimiento académico en la educación superior usando minería de datos y su comparación con técnicas estadísticas” Se aplica las técnicas de predicción a 7

cursos de la especialidad de ingeniería química de la universidad nacional de ingeniería, usando los datos de los períodos académicos del 1993-1 al 2010-2. La aplicación de las técnicas de redes neuronales de retropropagación y de regresión logística para la predicción de la aprobación o no de un curso, arrojan promedios de porcentajes de aciertos similares, de 70,45 % y 70,39 % para los modelos, y de 72,83 % y 74,04 % para los pronósticos, respectivamente. La aplicación de las técnicas de redes neuronales de retropropagación y de regresión múltiple para la predicción de la "nota" de un curso, arrojan promedios de raíz de errores medios cuadráticos similares, de 0,1495 y 0,1430 para los modelos, y de 0,1397 y 0,1380 para los pronósticos, respectivamente.

A nivel regional encontramos investigaciones como la de Chávez & Mendoza (2006) donde buscaron una correlación entre el rendimiento académico de los alumnos del primer año de la facultad de ingeniería y su rendimiento académico en la educación superior básica regular, determinando que se llega a establecer que efectivamente el rendimiento académico de los estudiantes de secundaria al ingresar a la universidad, está relacionado con el rendimiento académico del primer año de universidad, con un coeficiente de correlación de 0,76.

El autor Yanarico Espinoza (2013) enfocó el “Desarrollo de un software utilizando redes neuronales artificiales para la simulación del ciclo hidrológico de la laguna Aricota” en el cual planteó desarrollar un software que permita realizar el análisis de las variables hidrobiológica, el entrenamiento de la red neuronal artificial de incremento de volumen y nivel de agua de la laguna de Aricota y la simulación del ciclo hidrobiológico de la laguna de Aricota.

## **2.2. Bases teóricas**

### **2.2.1. Rendimiento académico**

En tanto que Novárez (1986) sostiene que el rendimiento académico es el quantum obtenido por el individuo en determinada actividad académica. El concepto de rendimiento está ligado al de aptitud, y sería el resultado de ésta, de factores volitivos, afectivos y emocionales, además de la ejercitación.

Chadwick (1979) define el rendimiento académico como la expresión de capacidades y de características psicológicas del estudiante desarrolladas y actualizadas a través del proceso de enseñanza-aprendizaje que le posibilita obtener un nivel de funcionamiento y logros académicos a lo largo de un período o semestre, que se sintetiza en un calificativo final (cuantitativo en la mayoría de los casos) evaluador del nivel alcanzado.

Por lo que se ve al rendimiento académico como un indicador del nivel de aprendizaje alcanzado por el alumno, por ello por lo que el rendimiento académico se convierte en una "tabla imaginaria de medida" Reyes Tejada (2003), para el aprendizaje logrado en el aula, que constituye el objetivo central de la educación. Sin embargo, en el rendimiento académico, intervienen muchas otras variables externas al sujeto, como la calidad del docente. Es pertinente dejar establecido que aprovechamiento del estudio no es sinónimo de rendimiento académico. En tanto que el aprovechamiento está referido, más bien, al resultado del proceso enseñanza-aprendizaje, de cuyos niveles de eficiencia son responsables tanto el que enseña como el que aprende.

### **2.2.2. Predicción**

Para Sierra Romero (2003), pronosticar es el arte de especificar información significativa acerca del futuro. Se menciona que los pronósticos jamás son perfectos debido a que, básicamente, se utilizan métodos que generan pronósticos sobre la base de la información previa. Los pronósticos serán menos confiables mientras mayor sea el horizonte que se va pronosticar.

### **2.2.3 Inteligencia artificial**

Según Rich & Knight (1994) “El estudio de cómo lograr que las computadoras realicen tareas que, por el momento, los humanos hacen mejor.”

La Inteligencia Artificial como ciencia y tecnología ha ido acumulando conocimientos sobre como emular diversas capacidades del ser humano para exhibir comportamiento vigente y se han desarrollado sistemas cada vez más perfeccionados que producen parcialmente dichas capacidades.

La Inteligencia Artificial trata de abarcar y estudiar muchas de las capacidades del hombre para poder ofrecerlas a la máquina y también al propio hombre en el entendimiento de los principios de su inteligencia. Ejemplos: agentes, lógica, redes neuronales, sistemas expertos, vida artificial, tecnología del habla y síntesis, visión artificial, robótica entre otros.

### **2.2.4. Redes neuronales**

Según Pino, Gómez, & de Abajo (2001): “Se trata de reproducir el esquema neuronal, tanto en sus unidades de proceso(neuronas), como en sus conexiones y modo de establecerlas”.

Una de las ramas del campo científico de la inteligencia artificial son las Redes Neuronales Artificiales (RNAs), entendiendo como tales aquellas redes en las que existen elementos procesadores de información cuyas interacciones locales depende del comportamiento en conjunto del sistema.

Tablada & Torres (2009) define las redes neuronales como “Modelos matemáticos cuya inspiración surgió en las neuronas biológicas y la estructura paralela masiva del cerebro, y poseen la capacidad de adquirir, almacenar y utilizar conocimiento a partir de los datos”.

Las neuronas se distribuyen en la red formando capas de un número determinado de elementos básico. Es decir, existe una capa de entrada que recibe directamente la información proveniente de las fuentes externas de la red, capaz ocultas que son internas a la red y no tienen contacto directo con el exterior (desde cero niveles hasta un número elevado), pudiendo estar interconectadas de distintas maneras, lo que termina junto a su número, las distintas topologías y una capa de salida que transfiere la formación de la red hacia el exterior.

La topología de las redes neuronales es la forma de organización de las neuronas en la red formando capas o agrupaciones de neuronas entre la entrada y

salida de la red neuronal. Por lo tanto, los parámetros fundamentales de la red serán: el número de capas, el número neuronas por capa, el grado de conectividad y el número de conexiones entre neuronas.

### **2.2.5. Perceptrón multicapa**

Rosenblatt (1958) define un perceptrón como una máquina que aprende, utilizando ejemplos, asigna los vectores de entrada (muestras) a diferentes clases, utilizando una función lineal de las entradas. Minsky & Papert (1969) describen el perceptrón como un gradiente de descenso estocástico, algoritmo que intenta separar linealmente un conjunto de datos de entrenamiento n-dimensionales. Se utiliza la palabra perceptrón en el primer sentido como una máquina, siguiendo Rosenblatt, y se refiere explícitamente al perceptrón como un algoritmo de aprendizaje.

Perceptrón utiliza una regla de corrección de errores para cambiar los pesos de cada unidad que hacen las respuestas erróneas a estímulos que presentan a la red, libre de bucles que tiene sus unidades dispuestas en capas, con cada unidad aportaciones solo para unidades en la capa siguiente de la secuencia. La primera capa comprende unidades de entrada; se puede entonces ser varias capas de unidades ocultas entrenables que llevan una representación interna, y por último

está la capa de unidades de salida, también con pesos sinápticos entrenables. Arbib (2003)

En la figura 1 se representa una red multicapa donde se definen tres tipos de capas: de entrada, oculta y de salida.

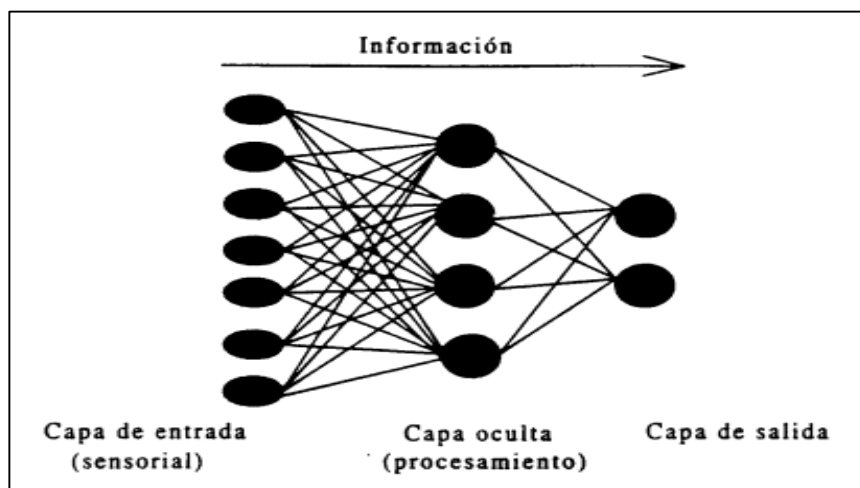


Figura 1. Red neuronal de 3 capas

Fuente: “Apuntes y transparencias de la asignatura de ingeniería neurosensorial”.

Intensificación de bioingeniería. ETSIT, UPM, curso 2000- 2001.

Este tipo de redes se caracterizan por su facilidad de implementación. Su aprendizaje se basa en la retropropagación: se parte de unos pesos iniciales en las conexiones interneuronales. Para un conjunto de entradas se obtiene una cierta

salida. Basándose en que se conoce la salida que deberíamos haber obtenido (patrón catalogado – aprendizaje supervisado), calculamos el error. A partir de este error se modifican los pesos siguiendo el sentido inverso al de evolución de la red (se parte de la salida hasta llegar a la entrada).

De la misma manera se opera con el resto de entradas de entrenamiento. Se puede observar que el error irá disminuyendo a medida que se aplique el algoritmo.

#### **2.2.6. Retropropagación**

El algoritmo de Retropropagación está basado en la regla delta generalizada, es seleccionado para entrenar la red multicapa de alimentación directa. La entrada se propaga hacia delante a través la capa media a la capa de salida, y luego la salida resultante se propaga hacia atrás, desde la capa de salida a través de la zona central hacia la capa de entrada, para calcular el error. Posteriormente, los derivados de error de peso y los cambios de peso se actualizan.

El entrenamiento de la red consiste en presentar repetidamente los patrones en establecidos en la red hasta que el error generado en la inicial aguda de la

pendiente de la curva y su sedimentación lenta indique que la red no tiene ninguna dificultad en el aprendizaje de los patrones de entrada.

La red neuronal multicapa es la red neuronal que más se aplica, que se ha utilizado en la mayoría de las investigaciones hasta el momento. Un algoritmo de retropropagación se puede utilizar para entrenar a estas redes de alimentación directa multicapa con un diferenciable al transferir funciones para llevar a cabo la función de aproximación, la asociación de patrón y clasificación de patrones. Retropropagación, este término se refiere al proceso de por el cual los derivados de error en la red, con respecto a los pesos de la red y sesgos, se pueden calcular.

La formación de las redes neuronales artificiales por propagación hacia atrás implica tres etapas, según Fausett (1994) son: Primera etapa, la alimentación hacia adelante del patrón de entrenamiento de entrada; segunda etapa, el cálculo y la propagación hacia atrás del error asociado; tercera etapa, el ajuste de los pesos.

### **2.3. Definición conceptual de términos**

#### **Rendimiento académico:**

Forma de medición de lo aprendido en algún determinado curso o semestre. Hace referencia a la evaluación del conocimiento adquirido, el rendimiento

académico es una medida de las capacidades del alumno, que expresa lo que éste ha aprendido a lo largo del proceso formativo. También supone la capacidad del alumno para responder a los estímulos educativos. En este sentido, el rendimiento académico está vinculado a la aptitud.

**Predicción:**

Es especificar información significativa acerca del futuro. Se menciona que los pronósticos jamás son perfectos debido a que, básicamente, se utilizan métodos que generan pronósticos sobre la base de la información previa.

**Red neuronal:**

Se trata de reproducir el esquema neuronal, tanto en sus unidades de proceso(neuronas), como en sus conexiones y modo de establecerlas.

**Perceptrón:**

Máquina que aprende, utilizando ejemplos, asigna los vectores de entrada (muestras) a diferentes clases, utilizando una función lineal de las entradas.

**Retropropagación:**

Modelo de conexiones hacia delante (feedforward) y hacia atrás (backforward) que utiliza pares de vectores en las entradas y salidas con valor verdadero para entrenar a la red.

## **CAPÍTULO III**

### **MARCO METODOLÓGICO**

#### **3.1. Tipo y diseño de la investigación**

El enfoque de nuestra investigación es cuantitativo donde usa la recolección de datos para probar hipótesis, con base en la medición numérica y el análisis para establecer patrones de comportamiento y probar teorías.

En este caso nuestra investigación de diseño no experimental transaccional descriptiva es decir que esta investigación se realiza sin manipular deliberadamente variables. Se basa fundamentalmente en la observación de fenómenos tal y como se dan en su contexto natural para después analizarlos.

Su objetivo, según Hernández Sampieri, Fernández Collado, & Baptista Lucio (2014), es indagar la incidencia de las modalidades o niveles de una o más variables en una población. Es un estudio puramente descriptivo y cuando establecen hipótesis, éstas son también descriptivas de pronóstico de valores.

### **3.2. Población y muestra**

#### **Población**

Ingresantes 2016-I y 2017-I de la Escuela Profesional de Ingeniería en Informática y Sistemas.

#### **Muestra**

Se trabajará con toda la población ya que se tienen los datos almacenados y son manipulables computacionalmente.

### **3.3. Variables**

#### **3.3.1. Identificación de variables**

- Red neuronal
- Rendimiento académico

#### **3.3.2. Definición de las variables**

##### **Redes neuronales artificiales**

- Por la función que cumplen en la hipótesis: Variable independiente
- Por su naturaleza: Desactiva

- Por el método de estudio: Cualitativa
- Por la posesión de la característica: Discreta
- Por los valores que adquiere: Politómica

### **Rendimiento académico**

- Por la función que cumplen en la hipótesis: Variable dependiente
- Por su naturaleza: Activa
- Por el método de estudio: Cuantitativa
- Por la posesión de la característica: Continua
- Por los valores que adquiere: Politómica

### **3.3.3. Operacionalización de variables**

**Variable independiente:** Redes neuronales

**Definición conceptual:** Las redes neuronales son un paradigma de aprendizaje y procesamiento automático inspirado en la forma en cómo funciona el sistema nervioso.

**Definición operacional:** Para que la red se ajuste de manera adecuada a los datos, se tiene que realizar varias configuraciones como el número de neuronas por

capa, numero de capas ocultas, funciones de transferencias y los pesos sinápticos de cada neurona.

**Variable dependiente:** Rendimiento académico

**Definición conceptual:** El rendimiento académico es la nota promedio que tiene un alumno en un semestre.

**Definición operacional:** Para realizar la predicción es necesario que el conjunto de datos cumpla un modelo estadístico que se ajuste a estos de manera adecuada y posteriormente realizar la predicción

**Indicadores:**

- Error cuadrático medio (error de entrenamiento de la red)
- Promedio

#### **3.3.4. Clasificación de las variables**

Variable independiente: Red neuronal

Variable dependiente: Rendimiento académico

### **3.4. Técnicas e instrumentos para la recolección de datos**

Se recopiló datos para establecer las bases de parámetros para el entrenamiento de la red neuronal artificial.

**Método de pronóstico:** Método de diseño de pronóstico con redes neuronales

**Técnicas:** Perceptrón multicapa con algoritmo de entrenamiento supervisado

**Herramienta:** Hoja de cálculo de Microsoft Excel y DevC++

**Fuente:** Los datos serán obtenidos de la Oficina de Admisión y la Dirección Académica de Actividades y Servicios Académicos (DASA) de la Universidad Jorge Basadre Grohmann. Se obtendrán datos reales para el propósito de la investigación.

### **3.5. Procesamiento y análisis de datos**

Para la implementación de las redes neuronales del pronóstico del rendimiento, la información y los valores no tienen datos nulos, se aplicará un test de validación de resultado (con las notas promedio existentes que se utilizará para

el entrenamiento supervisado) y se hallará el error cuadrático medio en validación y pronóstico de los futuros alumnos del primer semestre 2018-I.

## CAPÍTULO IV

### RESULTADOS Y DISCUSIONES

#### 4.1. Resultados

##### 4.1.1. Construcción de red neuronal

##### 4.1.1.1. Perceptrón

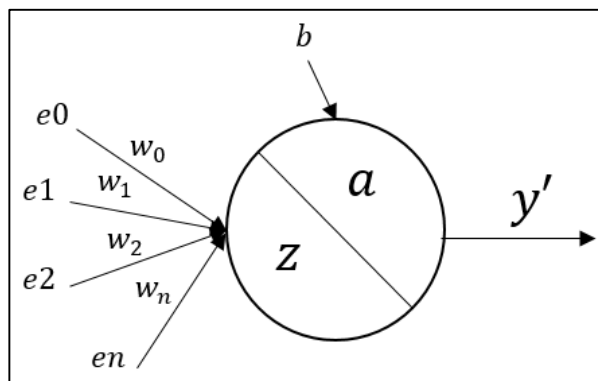


Figura 2. Perceptrón simple

Fuente: Elaboración propia

$$a = \sum_{i=0}^n (w_i * e_i) + b \quad (1)$$

### 4.1.1.2. Perceptrón multicapa

La función que cumplen dichas capas intermedias (capas ocultas) es tratar de realizar una proyección en la que resulten separables linealmente los patrones de entrada de manera que la unidad de salida pueda tener salida con un mínimo error.

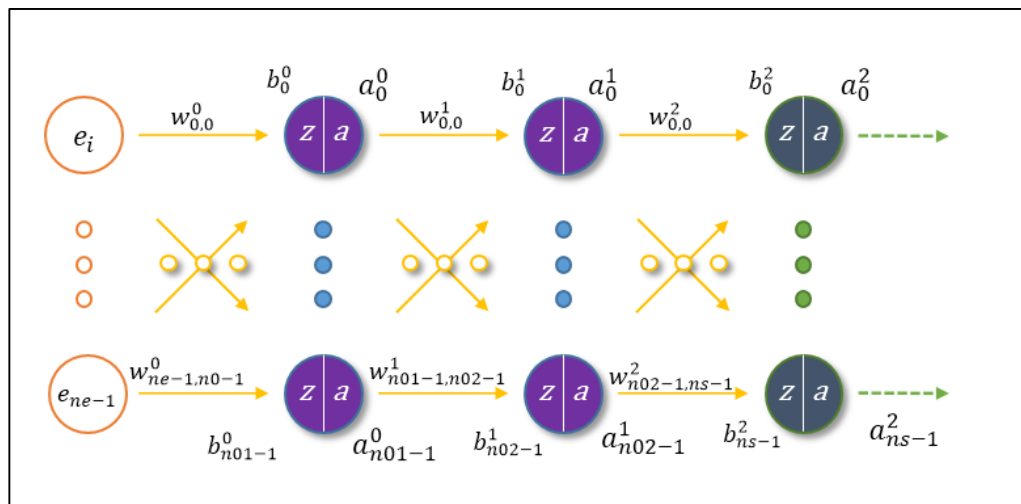


Figura 3. Perceptrón multicapa

Fuente: Elaboración propia

Donde:

⋮ Abreviación de un número de neuronas.

... Abreviación de un número de conexiones

**ne** Número de entradas

**n01** Número de neuronas en la capa oculta 1

**n02** Número de neuronas en la capa oculta 2

**ns** Número de salidas.

El Perceptron multicapa tiene dos componentes que se ayudan entre sí, el forward o hacia adelante que busca con los pesos inicialmente aleatorios que son asignados respectivamente a cada entrada puedan ser calculadas en cada neurona donde cada neurona contiene una función de activación  $Z$ .

Donde la función de actización:  $act(z) = \frac{1}{1+e^{-z}}$  en nuestra investigación

será la función sigmoide.

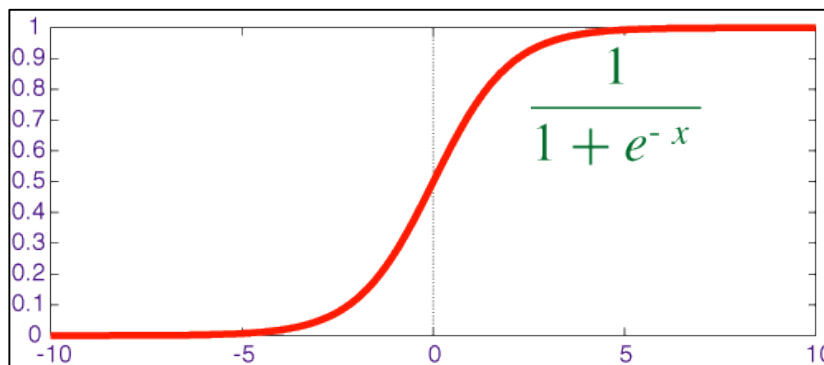


Figura 4. Función de activación sigmoide

Fuente: Elaboración propia

El cual halla una salida calculada y será comparada con la salida real para así encontrar el error de cada función.

## Función de activación

Esta investigación se utilizó la función de sigmoïdal ya que es la más conveniente para nuestro dataset.

$$act(z) = \frac{1}{1 + e^{-z}} \quad (2)$$

Derivando:

$$\begin{aligned} act(z) &= \frac{1}{1+e^{-z}} = (1 + e^{-z})^{-1} \\ \frac{\partial act(z)}{\partial z} &= \frac{\partial (1+e^{-z})^{-1}}{\partial z} = -2 \cdot e^{-z} \cdot (1 + e^{-z})^{-2} \\ \frac{\partial act(z)}{\partial z} &= e^{-z} \cdot (1 + e^{-z})^{-2} \\ \frac{\partial act(z)}{\partial z} &= \frac{e^{-z}}{1+e^{-z}} \cdot \frac{1}{1+e^{-z}} \text{ donde } \frac{1}{1+e^{-z}} \text{ es } act(z) \\ \frac{\partial act(z)}{\partial z} &= \frac{1+e^{-z}-1}{1+e^{-z}} \cdot act(z) \\ \frac{\partial act(z)}{\partial z} &= \left( \frac{1+e^{-z}}{1+e^{-z}} - \frac{1}{1+e^{-z}} \right) \cdot act(z) \\ \frac{\partial act(z)}{\partial z} &= (1 - act(z)) \cdot act(z) \quad (3) \end{aligned}$$

### Estimación de error

$$Error = \frac{1}{2} \sum_{i=0}^{ns} (t_i - a_i^2)^2 \quad (4)$$

Derivando:

$$\frac{\partial E}{\partial a_j^2} = \frac{1}{2} \cdot 2 \cdot -1 \cdot (a_j^2 - t_j)$$

$$\frac{\partial E}{\partial a_j^2} = (a_j^2 - t_j) \quad (5)$$

#### 4.1.1.3. Retropropagación

La retropropagación se basa en una regla de ajuste, que trata de determinar las conexiones sinápticas de manera que las salidas de red coincidan con las salidas deseadas o por lo menos, sean lo más próximas posibles. Utiliza el método de descenso de gradiente para el ajuste de los pesos de la red neuronal.

### FeedForward (hacia adelante)

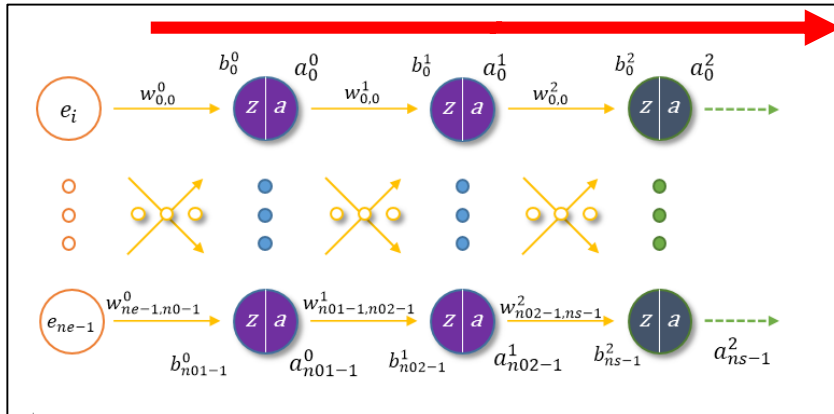


Figura 5. Propagación hacia adelante

Fuente: Elaboración propia

Es el cálculo de las entradas con los pesos inicialmente aleatorios en cada neurona de las capas ocultas hasta llegar a la capa de salida.

Se calcula los pesos con los valores  $n$  entradas donde  $a$  es el valor de la neurona.

$$j = \text{capa oculta } 1$$

$$a_j^0 = \text{act}\left(\sum_{i=0}^{ne} (w_{i,j}^0 * e_i) + b_j^0\right) \quad (6)$$

$j = \text{capa oculta 2}$

$$a_j^1 = \text{act}\left(\sum_{i=0}^{n01} (w_{i,j}^1 * a_i^0) + b_j^1\right) \quad (7)$$

$j = \text{capa salida}$

$$a_j^2 = \text{act}\left(\sum_{i=0}^{n02} (w_{i,j}^2 * a_i^1) + b_j^2\right) \quad (8)$$

### **Función de activación**

$$\text{act}(z) = \frac{1}{1 + e^{-z}} \quad (9)$$

$$\frac{\partial \text{act}(z)}{\partial z} = (1 - \text{act}(z)) \cdot (\text{act}(z)) \quad (10)$$

### **Error**

$$\text{Error} = \frac{1}{2} \sum_{i=0}^{ns} (t_i - a_i^2)^2 \quad (11)$$

$$\frac{\partial E}{\partial a_j^2} = (a_j^2 - t_j) \quad (12)$$

## Backforward (hacia atrás)

Dicho ajuste se realiza comenzando por la capa de salida, y se va propagando este error por las capas anteriores, hasta llegar a la capa de las unidades de entrada.

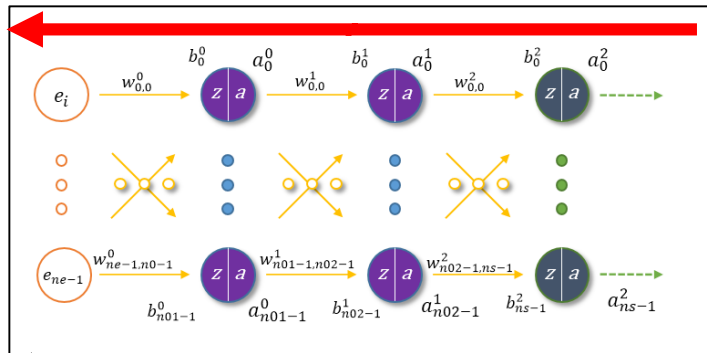


Figura 6. Retropropagación hacia atrás

Fuente: Elaboración propia

## Capa salida

En la capa de salida se tiene las siguientes derivadas respecto al bias que acompaña a la neurona de salida y sus respectivos pesos.

$$\frac{\partial E}{\partial b_i^2} = \frac{\partial E}{\partial a_i^2} \cdot \frac{\partial a_i^2}{\partial z_i^2} \cdot \frac{\partial z_i^2}{\partial b_i^2}$$

$$\frac{\partial E}{\partial b_i^2} = \partial error(e_i, a_i^2) \cdot \partial act(a_i^2) \cdot 1$$

$$\frac{\partial E}{\partial w_{ij}^2} = \frac{\partial E}{\partial a_j^2} \cdot \frac{\partial a_j^2}{\partial z_j^2} \cdot \frac{\partial z_j^2}{\partial w_{ij}^2}$$

$$\frac{\partial E}{\partial w_{ij}^2} = \frac{\partial E}{\partial a_j^2} \cdot a_i^1 \quad (13)$$

## Capa oculta 2

Se debe obtener también el cálculo del bias y los pesos de cada neurona de la capa 2.

$$\frac{\partial E}{\partial b_i^1} = \frac{\partial E}{\partial a_i^1} \cdot \frac{\partial a_i^1}{\partial z_i^1} \cdot \frac{\partial z_i^1}{\partial b_i^1}$$

$$\frac{\partial E}{\partial b_i^1} = \left( \sum_{j=0}^{ns} \frac{\partial E}{\partial a_j^2} \cdot \frac{\partial a_j^2}{\partial z_j^2} \cdot \frac{\partial z_j^2}{\partial a_i^1} \right) \cdot \frac{\partial a_i^1}{\partial z_i^1} \cdot \frac{\partial z_i^1}{\partial b_i^1}$$

$$\frac{\partial E}{\partial b_i^1} = \left( \sum_{j=0}^{ns} \frac{\partial E}{\partial b_j^2} \cdot w_{ij}^2 \right) \cdot \partial act(a_i^1) \cdot 1$$

$$\frac{\partial E}{\partial w_{ij}^1} = \frac{\partial E}{\partial a_j^1} \cdot \frac{\partial a_j^1}{\partial z_j^1} \cdot \frac{\partial z_j^1}{\partial w_{ij}^1}$$

$$\frac{\partial E}{\partial w_{ij}^1} = \frac{\partial E}{\partial b_j^1} \cdot a_i^0 \quad (14)$$

### Capa oculta 1

Se debe obtener también el cálculo del bias y los pesos de cada neurona de la capa 1.

$$\frac{\partial E}{\partial b_i^0} = \frac{\partial E}{\partial a_i^0} \cdot \frac{\partial a_i^0}{\partial z_i^0} \cdot \frac{\partial z_i^0}{\partial b_i^0}$$

$$\frac{\partial E}{\partial b_i^0} = \left( \sum_{j=0}^{n_0-2} \frac{\partial E}{\partial a_j^1} \cdot \frac{\partial a_j^1}{\partial z_j^1} \cdot \frac{\partial z_j^1}{\partial a_i^0} \right) \cdot \frac{\partial a_i^0}{\partial z_i^0} \cdot \frac{\partial z_i^0}{\partial b_i^0}$$

$$\frac{\partial E}{\partial b_i^0} = \left( \sum_{j=0}^{n_0-2} \frac{\partial E}{\partial b_j^1} \cdot w_{ij}^1 \right) \cdot \partial act(a_i^0) \cdot 1$$

$$\frac{\partial E}{\partial w_{ij}^0} = \frac{\partial E}{\partial a_j^0} \cdot \frac{\partial a_j^0}{\partial z_j^0} \cdot \frac{\partial z_j^0}{\partial w_{ij}^0}$$

$$\frac{\partial E}{\partial w_{ij}^0} = \frac{\partial E}{\partial b_j^0} \cdot e_i \tag{15}$$

#### 4.1.1.4. Análisis de sensibilidad

El análisis de sensibilidad está basado en la medición del efecto que se observa en una salida  $a_{2,j}$  debido al cambio que se produce en una entrada  $e_i$ .

Cuanto mayor efecto se observe sobre la salida, mayor sensibilidad se puede deducir que presenta respecto a la entrada.

$i = \text{entrada } j = \text{salida}$

$$\frac{\partial a_{2,j}}{\partial e_i} = \sum_{k=0}^{\text{oculta1}} \underbrace{\frac{\partial a_{2,j}}{\partial a_{0,k}}}_{\partial \text{sigmoide}(a_{1,l})} \cdot \underbrace{\frac{\partial a_{0,k}}{\partial z_{0,k}}}_{\frac{\partial z_{1,l}}{w_{1,k,l}}} \cdot \underbrace{\frac{\partial z_{1,l}}{\partial e_i}}_{\frac{\partial z_{1,l}}{w_{1,k,l}}}$$

$$\frac{\partial a_{2,j}}{\partial e_i} = \sum_{j=0}^{\text{salidas}} \underbrace{\frac{\partial a_{2,j}}{\partial z_{2,j}}}_{\partial \text{sigmoide}(a_{2,l})} \cdot \underbrace{\frac{\partial a_{2,j}}{\partial a_{1,l}}}_{w_{2,l,j}} \quad (16)$$

#### 4.1.1.5. Entradas red neuronal

Se ha propuesto como variables de entrada las respuestas de cada ingresante a la Escuela Profesional de Ingeniería en Informática y Sistemas tipificadas en 8 áreas generales del examen de admisión en la UNJBG observadas en la tabla 1.

Tabla 1

*Áreas generales para exámenes de admisión fase I y II*

ÁREA	Nro de Preguntas	Inicio - Fin
<b>Razonamiento Verbal</b>	12	1-12
<b>Razonamiento Matemático</b>	12	13-24
<b>Aritmética y Álgebra</b>	8	25-32
<b>Geometría y Trigonometría</b>	8	33-40
<b>Lenguaje</b>	2	41-42
<b>Lógica</b>	3	43-45
<b>Física</b>	8	46-53
<b>Química</b>	7	54-60
<b>Total</b>	<b>60</b>	<b>60</b>

Fuente: Oficina de admisión UNJBG

Tabla 2

*Áreas generales para examen de admisión CEPU*

<b>Área</b>	<b>Nro de Preguntas</b>	<b>Inicio - Fin</b>
<b>Razonamiento Verbal</b>	16	1-16
<b>Aritmética y Algebra</b>	16	17-32
<b>Física</b>	14	33-46
<b>Química</b>	14	47-60
<b>Razonamiento Matemático</b>	22	1-22
<b>Lógica</b>	10	23-32
<b>Geometría Y Trigonometría</b>	18	33-50
<b>Lenguaje</b>	10	51-60
<b>Total</b>	<b>120</b>	<b>1-60</b>

Fuente: Oficina CEPU – UNJBG

Tabla 3

*Cantidad de datos de exámenes CEPU*

<b>Tipo de Examen</b>	<b>Cantidad de Datos</b>
CEPU VERANO	7
CEPU OTOÑO 2017	10
CEPU INVIERNO	7
CEPU VERANO	10
<b>TOTAL</b>	<b>34</b>

Fuente: Elaboración propia

Tabla 4

*Cantidad de datos de exámenes fase I y II*

<b>Tipo de Examen</b>	<b>Cantidad de Datos</b>
FASE 2 2016	17
FASE 2 2017	11
FASE 1 2017	7
<b>TOTAL</b>	<b>35</b>

Fuente: Elaboración propia

#### **4.1.1.6. Salidas de la red neuronal**

La salida de nuestra red neuronal será el promedio de sus notas en el primer semestre académico de su carrera universitaria el cual fue otorgado por DASA.

#### **4.1.1.7 Conjunto de datos**

Se dispone de 69 registros de ingresantes a la escuela profesional de ingeniería en informática y sistemas cuya nota promedio en su primer semestre fue mayor a 7,00.

#### **4.1.8. Normalización de datos**

Al ser nuestra variable de entrada el número de respuestas correctas que hizo cada ingresante a la carrera tipificadas por las áreas del examen de admisión. La variable de salida será el rendimiento académico normalizado de cada ingresante en el primer semestre de su carrera.

Donde los datos de entrada son:

$$entrada_{\text{área}} = \frac{\text{número de respuestas correctas por área}}{\text{número de preguntas del área}} \quad (17)$$

$$salida = \frac{\text{promedio alumno}}{20} \quad (18)$$

En la tabla 5 se toma en consideración los datos normalizados respecto a cada área. En la tabla se pone a disposición los datos reales de 69 ingresantes que se utilizó como conjunto de datos para el entrenamiento y validación de la red neuronal. Cabe mencionar que no se utilizó los nombres de ingresantes en esta investigación ya que son datos sensibles y no son materia de observación.

Tabla 5

*Normalización de las variables de entrada de la red neuronal*

<b>Ingresante</b>	<b>RV</b>	<b>RM</b>	<b>AA</b>	<b>GT</b>	<b>LENG</b>	<b>LOG</b>	<b>FIS</b>	<b>QUIM</b>
<b>Alumno 1</b>	{0 - 1}	{0 - 1}	{0 - 1}	{0 - 1}	{0 - 1}	{0 - 1}	{0 - 1}	{0 - 1}
<b>Alumno 2</b>	{0 - 1}	{0 - 1}	{0 - 1}	{0 - 1}	{0 - 1}	{0 - 1}	{0 - 1}	{0 - 1}
<b>Alumno 69</b>	{0 - 1}	{0 - 1}	{0 - 1}	{0 - 1}	{0 - 1}	{0 - 1}	{0 - 1}	{0 - 1}

Fuente: Elaboración propia

Tabla 6

*Normalización de las variables de salida de la red neuronal*

<b>Ingresante</b>	<b>Rendimiento académico primer semestre</b>
<b>Alumno 1</b>	{0 - 1}
<b>...</b>	{0 - 1}
<b>Alumno 69</b>	{0 - 1}

Fuente: Elaboración propia

Tabla 7

*Data set normalizada para la red neuronal*

NRO	RV	RM	A y A	G y T	LENG	LOG	FISICA	QUIMICA	PROMEDIO
1	0,875	0,954545	0,8125	0,833333	0,6	0,6	0,428571	0,5	0,82857143
2	0,666667	0,75	0,5	0,75	1	0,666667	0,125	0,285714	0,77857143
3	0,875	0,681818	0,5	0,777778	0,7	0,8	0,785714	0,357143	0,73571429
4	0,916667	0,916667	0,5	0,5	1	0,666667	0,375	0,571429	0,72142857
5	1	0,666667	0,5	0,625	1	0,666667	0,625	0,428571	0,71428571
6	0,916667	0,5	0,875	0,625	1	1	0,125	0,142857	0,7
7	0,4375	0,5	0,75	0,611111	0,4	0,5	0,5	0,642857	0,68571429
8	0,8125	0,863636	0,5	0,5	0,4	0,8	0,428571	0,857143	0,68571429
9	0,75	0,5	0,75	0,25	0,5	0	0,25	0	0,68571429
10	0,8125	0,772727	0,625	0,388889	0,3	0,7	0,642857	0,571429	0,67142857
11	0,666667	0,583333	0,875	0,5	1	1	0,375	0,142857	0,67142857
12	0,5	0,416667	0,75	0,25	1	1	0,375	0,285714	0,67142857
13	0,875	0,545455	0,4375	0,777778	0,4	1	0,785714	0,428571	0,67142857
14	0,75	0,727273	0,5	0,666667	0,5	0,6	0,857143	0,5	0,67142857
15	0,5625	0,818182	0,6875	0,277778	0,1	0,4	0,285714	0,785714	0,66875
16	0,333333	0,416667	0,375	0,5	1	1	0,5	0,285714	0,66428571
17	0,875	0,590909	0,5625	0,611111	0,3	0,5	1	0,642857	0,66428571
18	0,583333	0,333333	0,625	0,625	1	0,666667	0,125	0,428571	0,65
19	0,666667	0,5	0,75	0	1	0,666667	0,25	0,142857	0,65
20	0,875	0,727273	0,875	0,5	0,1	0,9	0,714286	0,571429	0,64375
21	0,416667	0,416667	1	0,5	1	0,666667	0,125	0,285714	0,64285714
22	0,9375	0,590909	0,3125	0,444444	0,6	0,9	0,142857	0	0,6
23	0,875	0,681818	0,3125	0,722222	0,7	0,5	0,714286	0,5	0,6
24	0,375	0,727273	0,6875	0,111111	0,3	0,4	0,5	0,714286	0,59375
25	0,666667	0,5	0,75	0,5	0,5	0,666667	0,25	0,571429	0,59285714
26	0,75	0,772727	0,5	0,388889	0,1	0,3	0,5	0,857143	0,5875
27	0,583333	0,333333	0,875	0,5	1	1	0,5	0,571429	0,58333333
28	0,75	0,166667	0,5	0,5	1	1	0	0,428571	0,58125
29	0,875	0,909091	0,8125	0,666667	0,9	0,9	0,428571	0,357143	0,58125
30	0,5	0,583333	0,875	0,75	0,5	1	0,375	0,714286	0,57777778
31	0,875	0,727273	0,3125	0	0,7	0,6	0,285714	0,214286	0,575
32	0,8125	0,727273	0,4375	0,333333	0,2	0,3	0,5	0,642857	0,575
33	0,625	0,727273	0,4375	0,388889	0,3	0,5	0,642857	0,642857	0,575
34	0,25	0,75	0,5	0,625	0,5	0,666667	0,25	0,142857	0,56111111
35	0,5625	0,772727	0,5	0,166667	0,2	0,4	0,571429	0,714286	0,55625
36	0,5	0,5	0,875	0,125	0,5	1	0	0,285714	0,53888889
37	0,666667	0,666667	0	0,75	0	1	0,375	0,714286	0,53333333
38	0,916667	0,916667	0,125	0,5	1	0,666667	0,25	0,142857	0,52777778
39	1	0,416667	0,5	0,375	1	0,666667	0,125	0	0,525
40	0,333333	0,333333	0,625	0,625	1	1	0,125	0,428571	0,52222222
41	0,5	0,636364	0,5625	0,277778	0,4	0,2	0,642857	0,571429	0,52222222
42	0,6875	0,772727	0,75	0,333333	0,1	0,4	0,571429	0,571429	0,5125
43	0,833333	0,25	0,875	0,25	1	0	0,125	0,142857	0,51111111

Fuente: Oficina de admisión y DASA

#### 4.1.9. Implementación

El código fuente de la implementación de la red neuronal utilizando el algoritmo de retropropagación se presenta en el Anexo 2.

Se ha utilizado dos capas ocultas en la red neuronal donde:

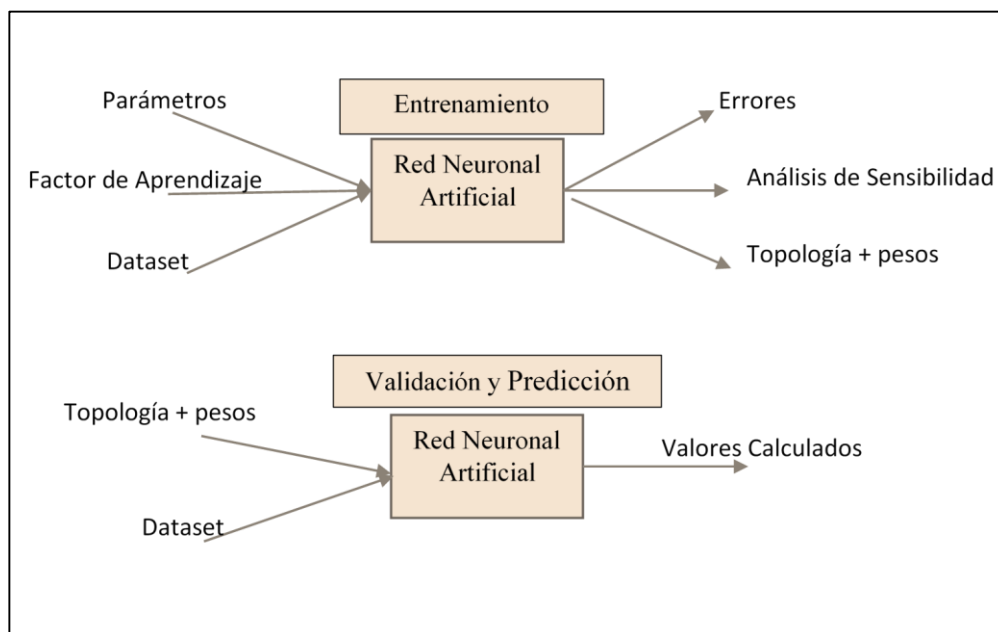


Figura 7. Entradas y salidas de implementación de la red neuronal

Fuente: Elaboración propia

#### **4.1.10. Selección de topología**

No hay procedimiento sencillo ni general para determinar la topología adecuada de la red neuronal perceptron multicapa.

Normalmente se comienza por redes de tamaño reducido, dado que su entrenamiento es más rápido y se va incrementando paulatinamente el número de neuronas hasta alcanzar una topología que presente el comportamiento deseado, es decir la topología que tenga el mínimo error en la fase validación. (Palma Méndez, Marin Morales, Guijarro Berdiñas, Fontela Romero, & Sánchez Meroño, 2008).

#### **4.1.11. Entrenamiento**

Esta investigación es no experimental, es decir busca observar los diferentes aspectos que abocan a la predicción más exacta del rendimiento académico para los alumnos ingresantes a la Escuela Profesional de Ingeniería en Informática y Sistemas utilizando redes neuronales.

En la construcción se planteó la jerarquización de la red neuronal con respecto a las entradas y salidas. En la implementación se esquematizó como la red neuronal ha sido implementada en base a las ecuaciones descritas en la construcción de la red neuronal.

Se ha tomado aleatoriamente el 75 % (51 registros) del dataset propuesto para el entrenamiento de la red neuronal donde se testeó hasta obtener el mínimo error en entrenamiento (5%) con diferentes topologías. Con el algoritmo de retropropagación y el código propuesto (Anexo 2). El siguiente gráfico muestra cuantas épocas necesito las topologías para poder llegar al 5% de error en entrenamiento.

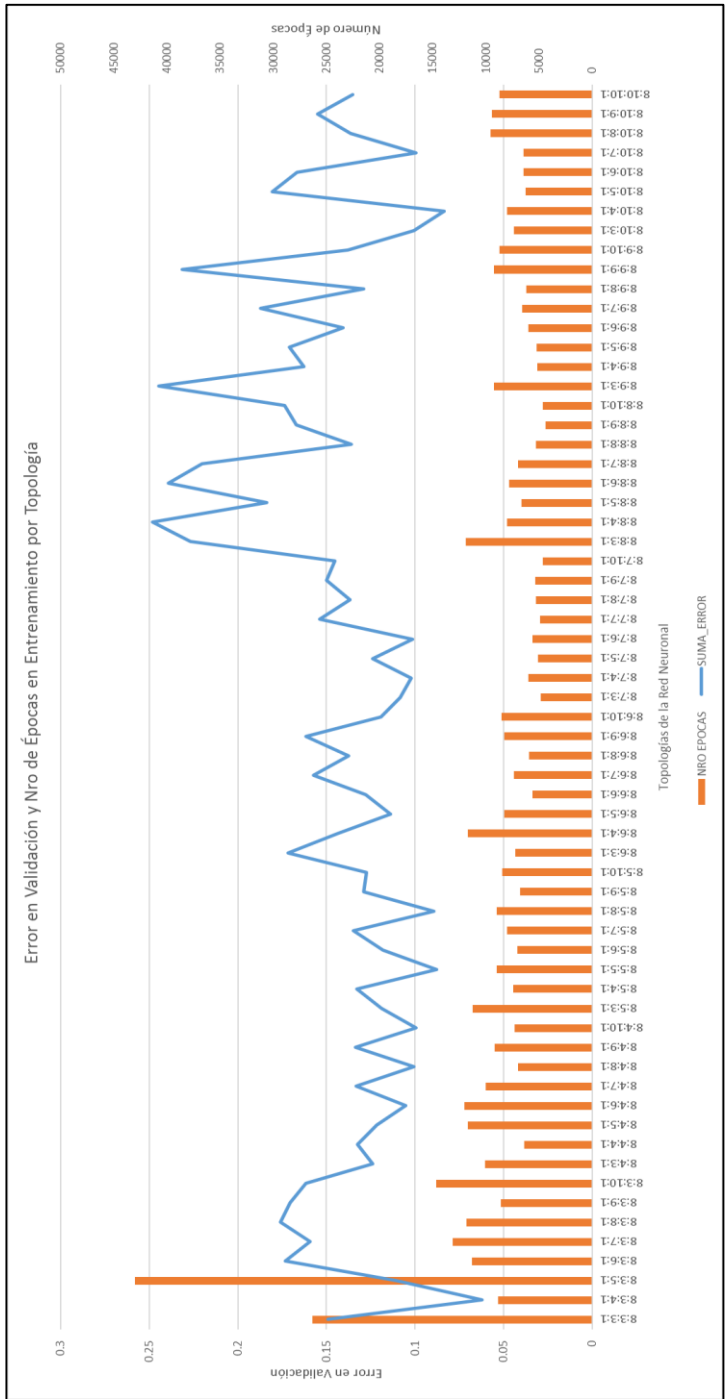


Figura 8. Entrenamiento: error en validación y nro de épocas  
Fuente: Elaboración propia

En la siguiente figura observamos los errores que se obtuvo en el entrenamiento en la red neuronal de topología 8:3:4:1 donde utilizó 8841 iteraciones con un factor de aprendizaje del 0,5 para tener menos del 5% de error en entrenamiento.

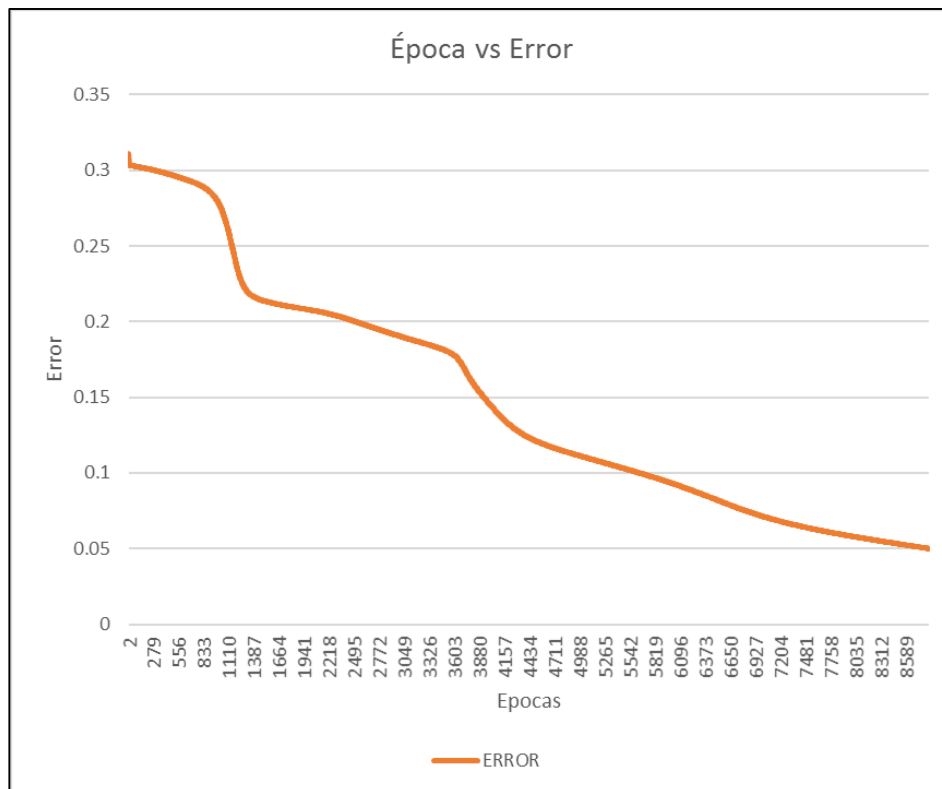


Figura 9. Entrenamiento: iteraciones (épocas) vs error

Fuente: Elaboración propia

En la figura siguiente observamos cómo se comporta el valor calculado en la topología 8:3:4:1 con 5% de error con el dataset de entrenamiento para 51 alumnos.

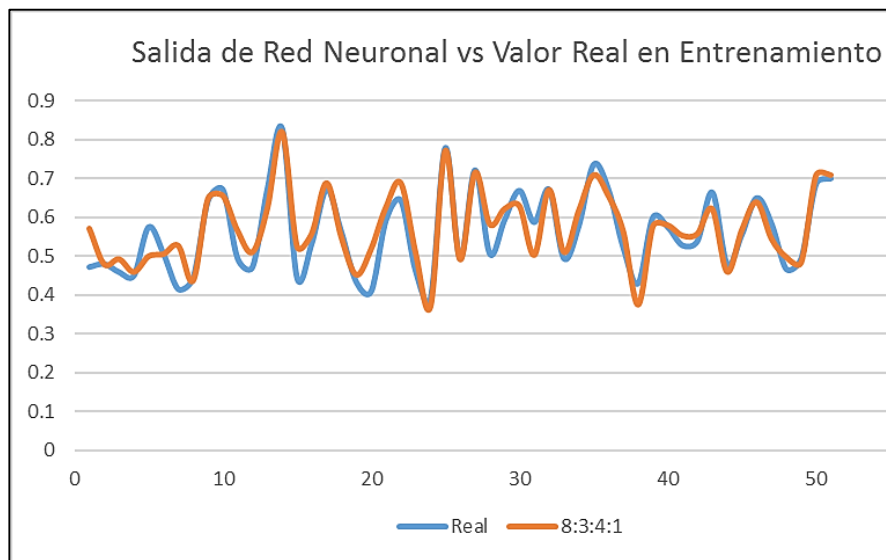


Figura 10. Entrenamiento: salidas calculadas vs salidas reales

Fuente: Elaboración propia

También se guardaron los pesos de cada topología en un archivo para utilizarlo en la validación de los datos.

## 4.2. Validación

Teniendo como datos para la validación el 25 % del dataset restante se ejecutó el código que está en el Anexo 2 que predice con los pesos obtenidos en el entrenamiento de la red neuronal de cada topología.

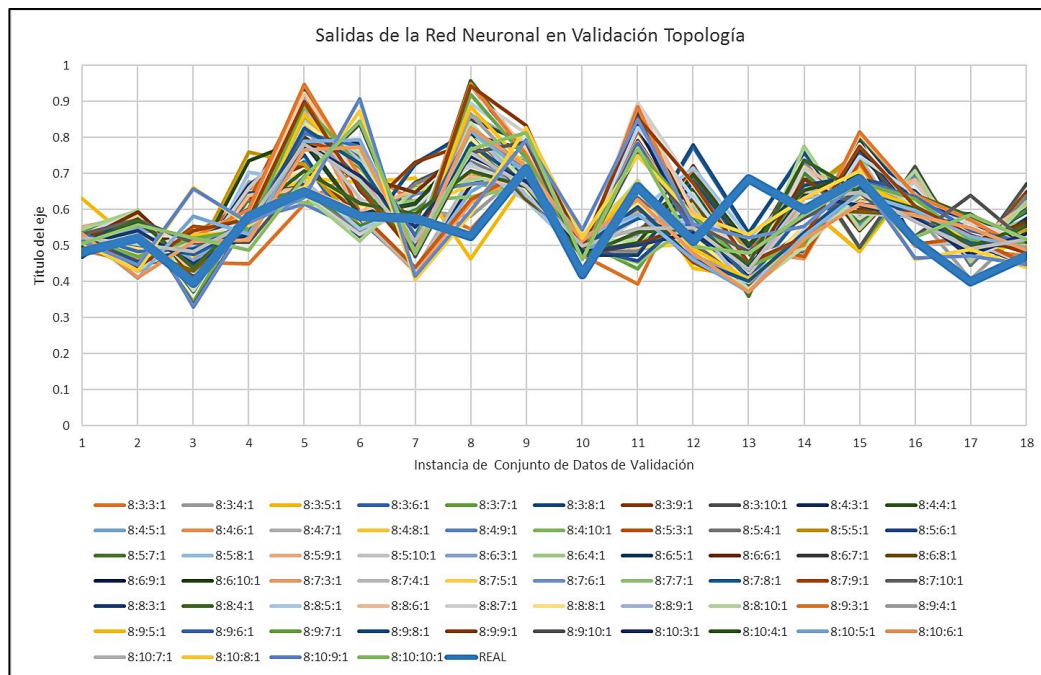


Figura 11. Validación: Salidas de la red neuronal

Fuente: Elaboración propia

En la figura anterior se observa la tendencia general del valor real (línea azul) con los valores calculados en la fase de validación de las diferentes topologías del 8:3:3:1 al 8:10:10:1 los cuales están calculados con los pesos generados en el entrenamiento de cada topología.

Compararemos también con los datos reales de cada promedio para la obtención de la suma de errores de mínimos cuadrados.

En la siguiente figura se observa como varia el promedio del error, mínimo error, máximo error y el error medio cuadrático en las diferentes topologías.

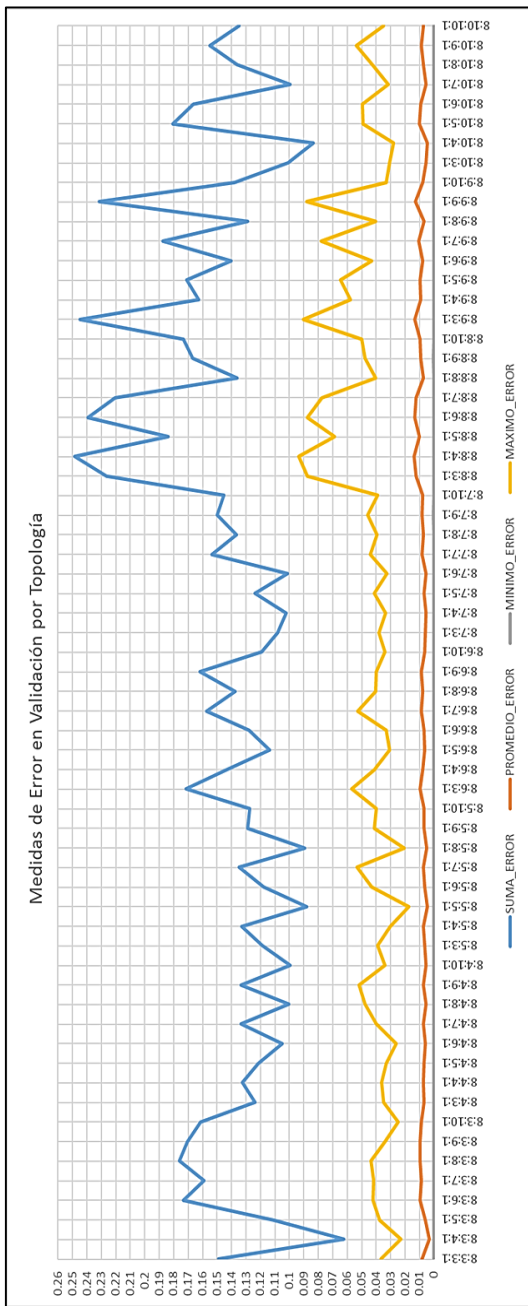


Figura 12. Validación: medidas de error (suma, promedio, mínimo y máximo)

Fuente: Elaboración propia

En la siguiente tabla se muestra la topología que tuvo el menor error cuadrático medio de todas las demás topologías.

Tabla 8

*Errores en validación de red neuronal con topología 8:3:4:1*

<b>INSTANCIA</b>	<b>REAL</b>	<b>8:3:4:1</b>	<b>Error</b>
<b>1</b>	0,48125	0,550944	0,00242863
<b>2</b>	0,522222222	0,583668	0,00188779
<b>3</b>	0,395	0,412834	0,00015903
<b>4</b>	0,58125	0,67849	0,00472781
<b>5</b>	0,65	0,64949	1,3005E-07
<b>6</b>	0,58125	0,62485	0,00095048
<b>7</b>	0,575	0,596768	0,00023692
<b>8</b>	0,525	0,582294	0,0016413
<b>9</b>	0,714285715	0,71242	1,7404E-06
<b>10</b>	0,41875	0,502323	0,00349222
<b>11</b>	0,664285715	0,486638	0,01577936
<b>12</b>	0,511111111	0,516194	1,2918E-05
<b>13</b>	0,685714286	0,471783	0,0228833
<b>14</b>	0,6	0,650269	0,00126349
<b>15</b>	0,685714286	0,68545	3,4923E-08
<b>16</b>	0,5125	0,593131	0,00325068
<b>17</b>	0,4	0,406101	1,8611E-05
<b>18</b>	0,47142857	0,555935	0,00357067
<b>Suma de Errores</b>			Suma de Errores

Fuente: Elaboración propia

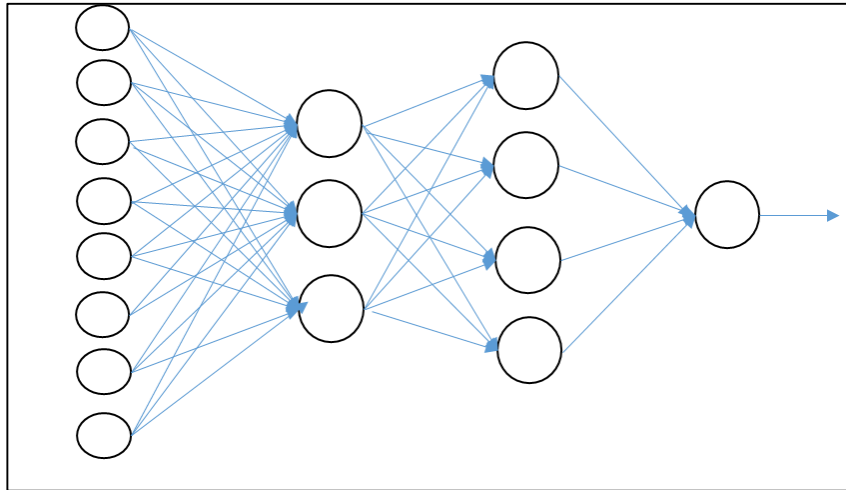


Figura 13. Topología 8:3:4:1

Fuente: Elaboración propia

Donde se puede observar la tendencia que siguen los valores del conjunto de datos de la fase de validación donde el valor real es azul y la calculada es de color naranja.

En el siguiente grafico se observa la tendencia que siguen los valores del conjunto de datos de la fase de validación donde el valor real es azul y la calculada es de color naranja.

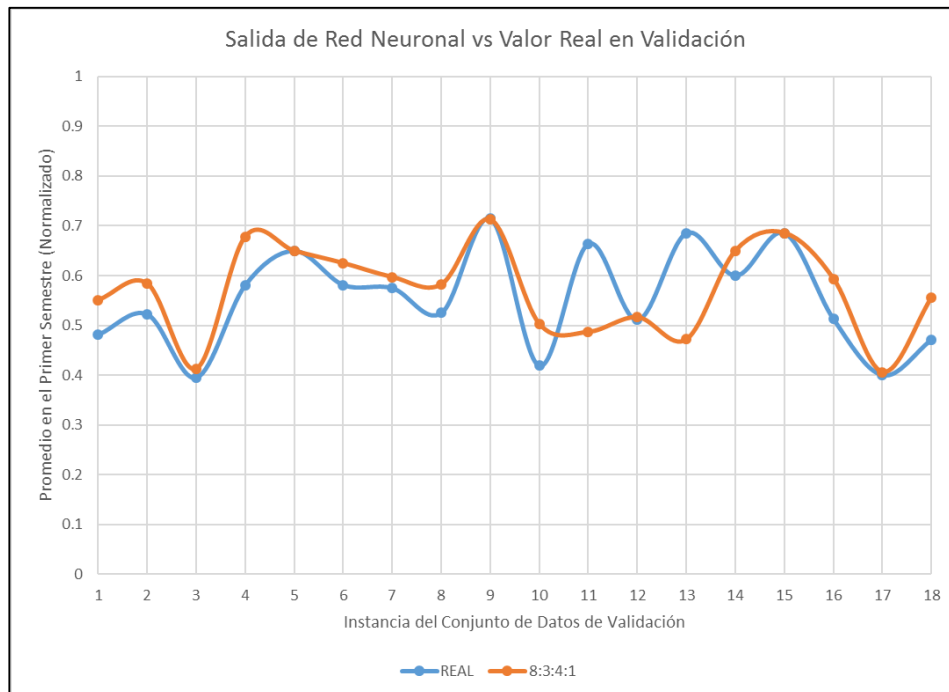


Figura 14. Validación: salida de la red neuronal vs valor real.

Fuente: Elaboración propia.

### 4.3. Análisis de sensibilidad

En la fase de entrenamiento se obtuvo todos los pesos por cada tipo de topología, luego de validar la topología que cumple mejor el objetivo de tener un mínimo error para los datos del conjunto de datos se dispuso en la tabla siguiente el análisis de sensibilidad que tuvo cada peso en modo de entrenamiento que involucrarían el como un área del examen de admisión difiere o es importante para el rendimiento académico en los alumnos ingresantes en el primer semestre de estudios.

Tabla 9

*Análisis de sensibilidad de las entradas de la red neuronal*

Áreas	
<b>Razonamiento verbal</b>	0,096
<b>Aritmética y algebra</b>	0,133
<b>Física</b>	-0,379
<b>Química</b>	-0,410
<b>Razonamiento matemático</b>	0,590
<b>Lógica</b>	-0,226
<b>Geometría y trigonometría</b>	-0,477
<b>Lenguaje</b>	-0,547

Fuente: Elaboración propia

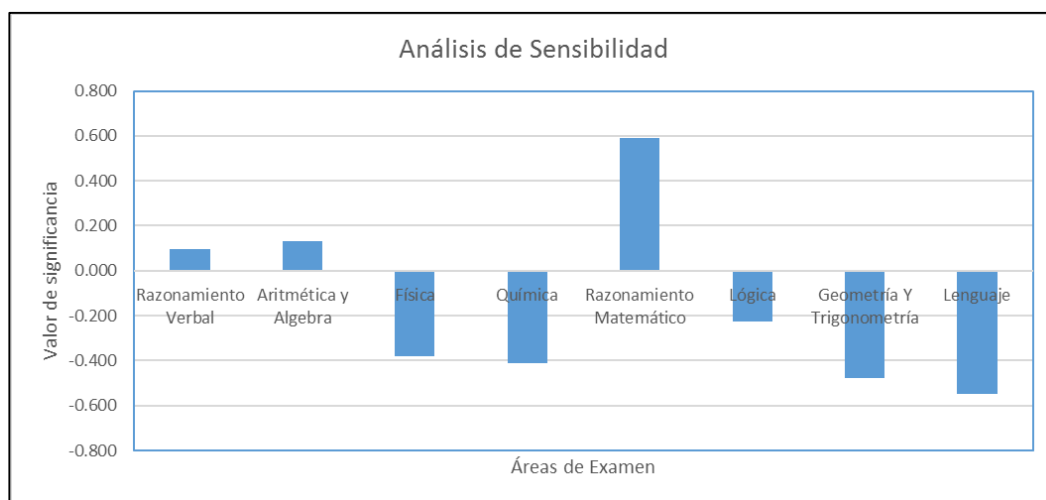


Figura 15. Análisis de sensibilidad en la red neuronal de topología 8:3:3:4

Fuente: Elaboración propia

En la figura anterior, se observa el valor de significancia que tienen las variables de entrada razonamiento verbal, aritmética y algebra y razonamiento matemático en nuestra red neuronal. Es decir, el haber marcado correctamente estas

3 áreas en el examen de admisión influye positivamente un 0,59, 0,13 y 0,09 en el rendimiento académico en el primer semestre del alumno ingresante.

#### 4.4. Predicción

Teniendo de conjunto de datos de 10 registros de los ingresantes CEPU-I 2018 y teniendo validada la red neuronal se utilizó los pesos óptimos y validados por lo que se tiene una predicción de su rendimiento académico para el Semestre 2018-I .

Tabla 10:

*Predicción del rendimiento académico en ingresantes CEPU-I 2018*

<b>INSTANCIA</b>	<b>CALCULADO</b>	<b>CALCULADO [1-20]</b>
<b>Ingresante 1</b>	0,630739	12,61
<b>Ingresante 2</b>	0,4988	9,98
<b>Ingresante 3</b>	0,428414	8,57
<b>Ingresante 4</b>	0,600264	12,01
<b>Ingresante 5</b>	0,688294	13,77
<b>Ingresante 6</b>	0,675693	13,51
<b>Ingresante 7</b>	0,687169	13,74
<b>Ingresante 8</b>	0,462496	9,25
<b>Ingresante 9</b>	0,686482	13,73
<b>Ingresante 10</b>	0,527549	10,55

Fuente: Elaboración propia

## 4.2 Discusiones

Barreto y otros autores (2012) en la investigación “Redes neuronales para predecir el rendimiento académico de los alumnos ingresantes a la carrera de Bioquímica de la FACENA-UNNE en función de sus conocimientos matemáticos previos” obtuvieron una red con topología 16:4:3:1 en perceptrón multicapa, en nuestra investigación se logró construir y entrenar una red neuronal de topología 8:3:4:1 para predicción del rendimiento académico de los alumnos ingresantes a la Escuela Profesional de Ingeniería en informática y Sistemas por lo que se valida que ninguna red neuronal es igual con respecto a la topología ya que cada una tiene características diferentes donde solo se puede obtener una topología ideal probando cada red neuronal y analizando su error mínimo cuadrado, añadiendo también que la salida de la red neuronal de nuestro antecedente es una variable dicotómica y la de nosotros es una salida normalizada de 0 a 1 que al finalizar la podemos convertir en una variable de 1 a 20.

Acosta y Pizarro (2011) en su investigación de “Predicción del rendimiento académico en la Educación Superior usando minería de datos y su comparación con técnicas estadísticas” también en un capítulo de su investigación usa redes neuronales y el aprendizaje supervisado de retropropagación donde tuvo un error medio cuadrático en la validación de la predicción de 13,97%. En nuestra investigación se logró validar la red neuronal con un error medio cuadrático de 6,2%

en la predicción del rendimiento académico de los alumnos ingresantes a la Escuela Profesional de Ingeniería en informática y Sistemas, pero es debido a que Acosta y Pizarro lo utilizan como un clasificador. Se debe tener en cuenta que en las predicciones es posible tener hasta un 20 % de error debido a que las predicciones de variables sociales pueden ser cambiantes y al nosotros usar variables cuantitativas se logra disminuir ese error, pero solo se tiene este de tipo de situaciones cuando se tiene el valor máximo y mínimo de la salida.

## CONCLUSIONES

Fue posible realizar un análisis predictivo, utilizando redes neuronales con el algoritmo de retropropagación en perceptrón multicapa del rendimiento académico de los alumnos ingresantes a la Escuela Profesional de Ingeniería en informática y Sistemas, del cual se tuvo un 5% de error en entrenamiento y 6,2% de error en validación, donde se tuvo que el área que influye al rendimiento académico es Razonamiento Matemático.

Se logró entrenar con un error menor al 5% una red neuronal de topología 8:3:4:1 la cual fue más óptima para predicción del rendimiento académico de los alumnos ingresantes a la Escuela Profesional de Ingeniería en informática y Sistemas.

Se logró validar la red neuronal artificial con un error menor al 10%, teniendo como error medio cuadrado del 6,2% en la fase de validación de la red neuronal que predice el rendimiento académico.

Fue posible analizar la predicción del rendimiento académico donde mediante un análisis de sensibilidad se obtuvo que las áreas de razonamiento matemático, aritmética y álgebra y razonamiento verbal influyen positivamente con un valor del 0,59 0,13 y 0,09 en el rendimiento académico.

## **RECOMENDACIONES**

Se recomienda que se haga un registro histórico de los exámenes de admisión y sus resultados tipificados por área para así poder tener un análisis preciso de los datos en todas las carreras profesionales ofrecidas en este centro de estudios.

Para futuras investigaciones se recomienda también aumentar las variables de entrada de la red neuronal artificial incluyendo aspectos socioeconómicos.

Para futuras investigaciones se podría considerar también las evaluaciones que hacen los alumnos a los docentes que también recopila DASA.

## REFERENCIAS BIBLIOGRÁFICAS

- Acosta, & Pizarro. (2011). *Predicción del rendimiento académico en la Educación Superior usando minería de datos y su comparación con técnicas estadísticas.*
- Alcázar, Silva, & Fernández. (2014). *Sistema inteligente para perfilar la deserción en estudiantes universitarios de carreras técnicas.*
- Arbib, M. (2003). *The Handbook of brain theory and neural networks.* Massachusetts: MIT Press Cambridge.
- Barreto, S. E., López, M. V., Ramirez Arballo, M. G., Porcel, E. A., & Mata, L. E. (2012). *Redes Neuronales para predecir el rendimiento académico de los alumnos ingresantes a la carrera de Bioquímica de la FACENA-UNNE(Argentina) en función de sus conocimientos matemáticos previos.*
- Cataldi, Z., Salgueiro, F., & Lage, F. J. (2009). *Sistema inteligente para perfilar la deserción en estudiantes universitarios de carreras técnicas.*
- Chadwick, C. (1979). Teoría del aprendizaje y su implicancia en el trabajo en el aula. *Revista de Educación.*

- Chávez, D., & Mendoza, J. (2006). Rendimiento académico de secundaria y su relación en el primer año de educación Universitaria en la UNJBG de Tacna. *Ciencia y Desarrollo UNJBG*, 47-50.
- Fausett, L. (1994). *Fundamentals of neuroal networks, architectures, algorithms and applications*.
- Gonzales, S. (2008). *Detección de alumnos en riesgo y medición de la eficiencia de centros escolares mediante redes neuronales*.
- Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, M. d. (2014). *Metodología de la Investigación* (6 ed.). C.D. México DF, México: McGRAW-HILL.
- Nováez, M. (1986). *Psicología de la actividad*. Mexico: Editorial Iberoamericana.
- Palma Méndez, J., Marin Morales, R., Guijarro Berdiñas, B., Fontela Romero, O., & Sánchez Meroño, N. (2008). *Inteligencia Artificial : Técnica,metodos y aplicaciones*. Murcia: McGraw-Hill.
- Pino, R., Gómez, A., & de Abajo, N. (2001). *Introducción a la ingeniería Artificial: Sistemas Expertos, Redes Neuronales Artificiales y Computación Evolutiva*. Oviedo, España: Servicios de publicaciones Universidad de Oviedo. Recuperado el 2017 de Julio de 18, de <https://docs.google.com/file/d/0By5TADaAJqrLd2R0UmRZU05JQIE/edit?hl=es&forcehl=1>

- Reyes Tejada, Y. N. (2003). *Relación entre el rendimiento académico, la ansiedad ante los exámenes, los rasgos de personalidad, el autoconcepto y la asertividad en estudiantes del primer año de Psicología de la UNMSM*. Lima, Perú.
- Rich, E., & Knight, K. (1994). *Inteligencia artificial* (2 ed.). (P. A. González Calero, & F. Trescastro Bodega, Trads.) McGraw-Hill. Recuperado el 18 de Julio de 2017
- Rosenblatt. (1958). The Perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, Volumen 65(No. 6), 24.
- Sierra Romero, R. (2003). *Mejoras industriales en una industria refresquera*. Puebla, Mexico.
- Tablada, C. J., & Torres, G. A. (2009). Redes neuronales artificiales. *Redes neuronales artificiales*, 24(24.3), 9. Recuperado el 18 de Julio de 2017, de <http://www.famaf.unc.edu.ar/~revm/digital24-3/redes.pdf>
- Yanarico Espinoza, W. (2013). *Desarrollo de un software utilizando redes neuronales artificiales para la simulación del ciclo hidrológico de la laguna Aricota*.

## Anexo 1: Matriz de consistencia

TÍTULO: “ANÁLISIS PREDICTIVO DEL RENDIMIENTO ACADÉMICO EN LOS ALUMNOS DE LA ESCUELA PROFESIONAL DE INGENIERÍA EN INFORMÁTICA Y SISTEMAS DE LA UNJBG, UTILIZANDO REDES NEURONALES, SEMESTRE 2017-I”

PROBLEMA	OBJETIVOS	HIPOTESIS	VARIABLES E INDICADORES
<p><b>PROBLEMA GENERAL:</b> ¿Se puede realizar un análisis predictivo del rendimiento académico de los alumnos de la Escuela Profesional de Ingeniería en Informática y Sistemas?</p> <p><b>PROBLEMAS ESPECIFICOS</b></p> <p>a) ¿Es posible entrenar una red neuronal de predicción del rendimiento académico en los alumnos del primer semestre?</p> <p>b) - ¿Es posible validar la red neuronal con error cuadrático medio menor al 10%</p> <p>c) - ¿Es posible predecir y analizar la predicción del rendimiento académico?</p>	<p><b>OBJETIVO GENERAL:</b> Realizar un análisis predictivo del rendimiento académico de los alumnos de la Escuela Profesional de Ingeniería en Informática y Sistemas.</p> <p><b>OBJETIVOS ESPECIFICOS:</b></p> <p>a) Entrenar una red neuronal con un error menor al 5% para la predicción del rendimiento académico.</p> <p>b) Validar la red neuronal con un error menor al 10% en la predicción del rendimiento académico.</p> <p>c) Predecir y analizar la predicción del rendimiento académico.</p>	<p><b>HIPÓTESIS GENERAL:</b> H0: No es posible realizar un análisis predictivo del rendimiento académico. H1: Es posible realizar un análisis predictivo del rendimiento académico.</p> <p><b>HIPÓTESIS ESPECÍFICAS:</b></p> <p>a) H1: Es posible entrenar una red neuronal con un error menor al 5% para la predicción del rendimiento académico.</p> <p>b) H1: Es posible validar la red neuronal con un error menor al 10% en la predicción del rendimiento académico.</p> <p>c) H1: Es posible predecir y analizar la predicción del rendimiento académico.</p>	<p><b>VARIABLES:</b> <i>Variable Independiente</i> Redes Neuronales Artificiales</p> <p><i>Variable Dependiente</i> Rendimiento académico en alumnos del primer semestre.</p> <p><b>Indicadores</b></p> <ul style="list-style-type: none"> <li>• Porcentaje de clasificación correcta (test de validación)</li> <li>• Error cuadrático medio</li> </ul>
<p><b>TIPO Y DISEÑO</b></p> <p>Enfoque: Cuantitativo Diseño: No experimental Transversal Descriptivo</p>	<p><b>POBLACION Y MUESTRA</b></p> <p>POBLACION: Estudiantes del primer semestre 2016-I y 2017-I E.P. Ingeniería en Informática y Sistemas. MUESTRA: Estudiantes del primer semestre 2016-I y 2017-I E. P. Ingeniería en Informática y Sistemas.</p>		



```

int n_entradas,n_oculta1,n_oculta2,n_salidas,n_instancias;
double
a[3][max_neuronas],w[3][max_neuronas][max_neuronas],b[3][max_neuronas],d_b[3][ma
x_neuronas];
double
entrada[max_instancias][max_entradas],salida[max_instancias][max_salidas];
double fact_aprendizaje,error_epoca,z,error_minimo;
char buffer[50];

//entrenamiento
if(strtol(argv[1],NULL,10)==0)
{
    //parametros
    n_entradas=strtol(argv[2],NULL,10);
    n_oculta1=strtol(argv[3],NULL,10);
    n_oculta2=strtol(argv[4],NULL,10);
    n_salidas=strtol(argv[5],NULL,10);
    error_minimo=strtod(argv[6],NULL);
    fact_aprendizaje=strtod(argv[7],NULL);

    //dataset
    ifstream txt_dataset(argv[8]);
    n_instancias=-1;
    while(!txt_dataset.eof())
    {
        n_instancias++;
        for(int i=0;i<n_entradas;i++)
            txt_dataset>>entrada[n_instancias][i];
        for(int i=0;i<n_salidas;i++)
            txt_dataset>>salida[n_instancias][i];
    }
    txt_dataset.close();

    //pesos
    for(int neurona=0;neurona<n_oculta1;neurona++)
    {
        b[0][neurona]=(double)rand()/RAND_MAX;
        for(int
neurona_atras=0;neurona_atras<n_entradas;neurona_atras++)
            w[0][neurona_atras][neurona]=(double)rand()/RAND_MAX;
    }
    for(int neurona=0;neurona<n_oculta2;neurona++)
    {
        b[1][neurona]=(double)rand()/RAND_MAX;
        for(int
neurona_atras=0;neurona_atras<n_oculta1;neurona_atras++)
            w[1][neurona_atras][neurona]=(double)rand()/RAND_MAX;
    }
    for(int neurona=0;neurona<n_salidas;neurona++)
    {
        b[2][neurona]=(double)rand()/RAND_MAX;
        for(int
neurona_atras=0;neurona_atras<n_oculta2;neurona_atras++)
            w[2][neurona_atras][neurona]=(double)rand()/RAND_MAX;
    }
}

//prediccion-clasificacion
if(strtol(argv[1],NULL,10)==1)
{
    cout<<"entro";
    //parametros
    ifstream txt_nn(argv[2]);
    txt_nn>>n_entradas;
    txt_nn>>n_oculta1;
    txt_nn>>n_oculta2;
}

```

```

txt_nn>>n_salidas;
fact_aprendizaje=0;

//pesos
for(int neurona=0;neurona<n_oculta1;neurona++)
{
    txt_nn>>b[0][neurona];
    for(int
neurona_atras=0;neurona_atras<n_entradas;neurona_atras++)
        txt_nn>>w[0][neurona_atras][neurona];
}
for(int neurona=0;neurona<n_oculta2;neurona++)
{
    txt_nn>>b[1][neurona];
    for(int
neurona_atras=0;neurona_atras<n_oculta1;neurona_atras++)
        txt_nn>>w[1][neurona_atras][neurona];
}
for(int neurona=0;neurona<n_salidas;neurona++)
{
    txt_nn>>b[2][neurona];
    for(int
neurona_atras=0;neurona_atras<n_oculta2;neurona_atras++)
        txt_nn>>w[2][neurona_atras][neurona];
}
txt_nn.close();

//dataset
ifstream txt_dataset(argv[3]);
n_instancias=-1;
while(!txt_dataset.eof())
{
    n_instancias++;
    for(int i=0;i<n_entradas;i++)
        txt_dataset>>entrada[n_instancias][i];
}
txt_dataset.close();
}

//DEBUG
if(FLAGS_DBG==1)
{
    cout<<endl<<"n_entradas : "<<n_entradas;
    cout<<endl<<"n_oculta1 : "<<n_oculta1;
    cout<<endl<<"n_oculta2 : "<<n_oculta2;
    cout<<endl<<"n_salidas : "<<n_salidas;
    cout<<endl<<"error_minimo : "<<error_minimo;
    cout<<endl<<"fact_apren: "<<fact_aprendizaje;
    cout<<endl<<endl<<"Matriz de entradas";
    for(int instancia=0;instancia<n_instancias;instancia++)
    {
        cout<<endl<<"\tInstancia "<<instancia<<" : ";
        for(int i=0;i<n_entradas;i++)
            cout<<entrada[instancia][i]<<",";
        cout<<"\t";
        for(int i=0;i<n_salidas;i++)
            cout<<salida[instancia][i]<<",";
    }

    cout<<endl<<"Pesos y Bias";
    for(int neurona=0;neurona<n_oculta1;neurona++) //capa oculta 1
    {
        cout<<endl<<"b[0] ["<<neurona<<"]="<<b[0][neurona];
        for(int
neurona_atras=0;neurona_atras<n_entradas;neurona_atras++)

```

```

        cout<<endl<<"w[0] ["<<neurona_atras<<"] ["<<neurona<<"] = "<<w[0][
neurona_atras][neurona];
    }

    for(int neurona=0;neurona<n_oculta2;neurona++) //capa oculta 2
    {
        cout<<endl<<"b[1] ["<<neurona<<"] = "<<b[1][neurona];
        for(int
neurona_atras=0;neurona_atras<n_oculta1;neurona_atras++)

            cout<<endl<<"w[1] ["<<neurona_atras<<"] ["<<neurona<<"] = "<<w[1][
neurona_atras][neurona];
    }
    for(int neurona=0;neurona<n_salidas;neurona++) //capa salida
    {
        cout<<endl<<"b[2] ["<<neurona<<"] = "<<b[2][neurona];
        for(int
neurona_atras=0;neurona_atras<n_oculta2;neurona_atras++)
            cout<<endl<<"w[2] ["<<neurona_atras<<"] ["<<neurona<<"] = "<<w[2][
neurona_atras][neurona];
    }
}

error_epoca=1.0;
//epocas
for(int epoca=0;error_epoca>error_minimo;epoca++)
{
    if(FLAG_DBG==1) cout<<endl<<endl<<"Epoca " <<epoca;
    error_epoca=0.0;

    //instancias
    for(int instancia=0;instancia<n_instancias;instancia++)
    {
        //propagacion hacia delante
        for(int neurona=0;neurona<n_oculta1;neurona++)
        {
            z=b[0][neurona];
            for(int
neurona_atras=0;neurona_atras<n_entradas;neurona_atras++)

                z+=w[0][neurona_atras][neurona]*entrada[instancia][neurona_atras];
            a[0][neurona]=sigmoide(z);
        }
        for(int neurona=0;neurona<n_oculta2;neurona++)
        {
            z=b[1][neurona];
            for(int
neurona_atras=0;neurona_atras<n_oculta1;neurona_atras++)
                z+=w[1][neurona_atras][neurona]*a[0][neurona_atras];
            a[1][neurona]=sigmoide(z);
        }
        for(int neurona=0;neurona<n_salidas;neurona++)
        {
            z=b[2][neurona];
            for(int
neurona_atras=0;neurona_atras<n_oculta2;neurona_atras++)
                z+=w[2][neurona_atras][neurona]*a[1][neurona_atras];
            a[2][neurona]=sigmoide(z);
        }

        //prediccion-clasificacion
        if(strtol(argv[1],NULL,10)==1)
        {
            //guardar predicciones-clasificaciones

            sprintf(buffer,"%s_%d_%d.out",argv[3],n_oculta1,n_oculta2);
            if(FLAG_DBG==1) cout<<endl<<"archivo para resultados:
"<<buffer;
            ofstream txt_out(buffer,ofstream::app);

```

```

        for(int neurona=0;neurona<n_salidas;neurona++)
            txt_out<<a[2][neurona]<<" ";
        txt_out<<endl;
        txt_out.close();
    }

    //entrenamiento
    if(strtol(argv[1],NULL,10)==0)
    {
        //calculo del ERROR
        for(int neurona=0;neurona<n_salidas;neurona++)
        {
            error_epoca+=error(salida[instancia][neurona],a[2][neurona]);
            if(FLAG_DBG==1) cout<<endl<<"error
            salida"<<neurona<<"="<<error(salida[instancia][neurona],a[2][n
            eurona]);
        }

        //propagacion hacia atras
        for(int neurona=0;neurona<n_salidas;neurona++)

            d_b[2][neurona]=d_error(salida[instancia][neurona],a[2][neuron
            a])*d_sigmoide(a[2][neurona]);
            for(int neurona=0;neurona<n_oculta2;neurona++)
            {
                z=0.0;
                for(int
                neurona_adelante=0;neurona_adelante<n_salidas;neurona_adelante
                ++)

                z+=w[2][neurona][neurona_adelante]*d_b[2][neurona_adelante];
                d_b[1][neurona]=z*d_sigmoide(a[1][neurona]);
            }
            for(int neurona=0;neurona<n_oculta1;neurona++)
            {
                z=0.0;
                for(int
                neurona_adelante=0;neurona_adelante<n_oculta2;neurona_adelante
                ++)

                z+=w[1][neurona][neurona_adelante]*d_b[1][neurona_adelante];
                d_b[0][neurona]=z*d_sigmoide(a[0][neurona]);
            }

            //nuevos pesos y bias
            for(int
            neurona_adelante=0;neurona_adelante<n_salidas;neurona_adelante++)
            {
                b[2][neurona_adelante]-
                =fact_aprendizaje*d_b[2][neurona_adelante];
                for(int
                neurona=0;neurona<n_oculta2;neurona++)
                w[2][neurona][neurona_adelante]-
                =fact_aprendizaje*d_b[2][neurona_adelante]*a[1][neurona];
            }
            for(int
            neurona_adelante=0;neurona_adelante<n_oculta2;neurona_adelante++)
            {
                b[1][neurona_adelante]-
                =fact_aprendizaje*d_b[1][neurona_adelante];
                for(int neurona=0;neurona<n_oculta1;neurona++)
                w[1][neurona][neurona_adelante]-
                =fact_aprendizaje*d_b[1][neurona_adelante]*a[0][neurona];
            }
            for(int
            neurona_adelante=0;neurona_adelante<n_oculta1;neurona_adelante++)
            {

```

```

                    b[0][neurona_adelante]-
=fact_aprendizaje*d_b[0][neurona_adelante];
                    for(int neurona=0;neurona<n_entradas;neurona++)
                    w[0][neurona][neurona_adelante]-
=fact_aprendizaje*d_b[0][neurona_adelante]*entrada[instancia][neurona];
                    }
                    }

                    //debug
                    if(FLAG_DBG==1)
                    {
                        for(int neurona=0;neurona<n_oculta1;neurona++)
                        {
                            if(FLAG_DBG==1)
                                cout<<endl<<"z[0] ["<<neurona<<"]="<<z;
                            if(FLAG_DBG==1)
                                cout<<endl<<"a[0] ["<<neurona<<"]="<<a[0][neurona]<<endl;
                        }
                        for(int neurona=0;neurona<n_oculta2;neurona++)
                        {
                            if(FLAG_DBG==1)
                                cout<<endl<<"z[1] ["<<neurona<<"]="<<z;
                            if(FLAG_DBG==1)
                                cout<<endl<<"a[1] ["<<neurona<<"]="<<a[1][neurona]<<endl;
                        }
                        for(int neurona=0;neurona<n_salidas;neurona++)
                        {
                            if(FLAG_DBG==1)
                                cout<<endl<<"z[2] ["<<neurona<<"]="<<z;
                            if(FLAG_DBG==1)
                                cout<<endl<<"a[2] ["<<neurona<<"]="<<a[2][neurona]<<endl;
                        }

                        //entrenamiento
                        if(strtol(argv[1],NULL,10)==0)
                        {
                            //calculo del ERROR
                            for(int
                                neurona=0;neurona<n_salidas;neurona++)
                                    cout<<endl<<"error salida
                                <<neurona<<"]="<<error(salida[instancia][neurona],a[2][neurona]);
                                    //propagacion hacia atras
                                    for(int
                                        neurona=0;neurona<n_salidas;neurona++)
                                            cout<<endl<<"d_b[2] ["<<neurona<<"]="<<d_b[2][neurona];
                                            for(int
                                                neurona=0;neurona<n_oculta2;neurona++)
                                                    cout<<endl<<"d_b[1] ["<<neurona<<"]="<<d_b[1][neurona];
                                                    for(int
                                                        neurona=0;neurona<n_oculta1;neurona++)
                                                            cout<<endl<<"d_b[0] ["<<neurona<<"]="<<d_b[0][neurona];
                                                            //nuevos pesos y bias
                                                            for(int
                                                                neurona_adelante=0;neurona_adelante<n_salidas;neurona_adelante++)
                                                                    cout<<endl<<"b[2] ["<<neurona_adelante<<"]="<<b[2][neurona_adelante];
                                                                    for(int
                                                                        neurona_adelante=0;neurona_adelante<n_oculta2;neurona_adelante++)
                                                                            {
                                                                                cout<<endl<<"b[1] ["<<neurona_adelante<<"]="<<b[1][neurona_adelante];
                                                                                for(int
                                                                                    neurona=0;neurona<n_oculta1;neurona++)
                                                                                        cout<<endl<<"w[1] ["<<neurona<<"] ["<<neurona_adelante<<"]="<<w[1][neurona][neurona_adelant
e];

```



```

        txt_nn<<b[2][neurona]<<" ";
        for(int
neurona_atras=0;neurona_atras<n_oculta2;neurona_atras++)
        {
            if(FLAGS_DBG==1)
cout<<endl<<"w[2][]"<<neurona_atras<<"][]"<<neurona<<"]="<<w[2][neurona_atras][neurona];
            txt_nn<<w[2][neurona_atras][neurona]<<" ";
        }
        txt_nn.close();
    }

    //entrenamiento
    if(strtol(argv[1],NULL,10)==0)
    {
        // analisis de sensibilidad
        sprintf(buffer,"%s_%s_%s.as",argv[8],argv[3],argv[4]);
        ofstream txt_as(buffer);
        double aux1;
        double aux2;
        double aux3;
        for(int neu_entrada=0;neu_entrada<n_entradas;neu_entrada++)
            for(int neu_salida=0;neu_salida<n_salidas;neu_salida++)
            {
                txt_as<<"entrada "<<neu_entrada<<" sobre salida
" <<neu_salida<<": ";
                aux1=0.0;
                for(int oculta1=0;oculta1<n_oculta1;oculta1++)
                {
                    aux2=0.0;
                    for(int
oculta2=0;oculta2<n_oculta2;oculta2++)
                    {
                        aux3=0.0;
                        for(int
salida=0;salida<n_salidas;salida++)
                        {
                            aux3+=d_sigmoide(a[2][neu_salida])*w[2][oculta2][neu_salida];
                            aux2+=aux3*d_sigmoide(a[1][oculta2])*w[1][oculta1][oculta2];
                        }
                            aux1+=aux2*d_sigmoide(a[0][oculta1])*w[0][neu_entrada][oculta1];
                        }
                            txt_as<<aux1<<endl;
                    }
                }
            }
        txt_as.close();
    }
}

```

### Anexo 3: Entradas red neuronal en fase de entrenamiento

8 6 4 1 10000 0.5 69  
0.875 0.954545 0.8125 0.833333 0.6 0.6 0.428571 0.5 0.8285714285  
0.666667 0.75 0.5 0.75 1 0.666667 0.125 0.285714 0.7785714285  
0.875 0.681818 0.5 0.777778 0.7 0.8 0.785714 0.357143 0.7357142855  
0.916667 0.916667 0.5 0.5 1 0.666667 0.375 0.571429 0.7214285715  
1 0.666667 0.5 0.625 1 0.666667 0.625 0.428571 0.7142857145  
0.916667 0.5 0.875 0.625 1 1 0.125 0.142857 0.7  
0.4375 0.5 0.75 0.611111 0.4 0.5 0.5 0.642857 0.6857142855  
0.8125 0.863636 0.5 0.5 0.4 0.8 0.428571 0.857143 0.6857142855  
0.75 0.5 0.75 0.25 0.5 0 0.25 0 0.6857142855  
0.8125 0.772727 0.625 0.388889 0.3 0.7 0.642857 0.571429 0.6714285715  
0.666667 0.583333 0.875 0.5 1 1 0.375 0.142857 0.6714285715  
0.5 0.416667 0.75 0.25 1 1 0.375 0.285714 0.6714285715  
0.875 0.545455 0.4375 0.777778 0.4 1 0.785714 0.428571 0.6714285715  
0.75 0.727273 0.5 0.666667 0.5 0.6 0.857143 0.5 0.6714285715  
0.5625 0.818182 0.6875 0.277778 0.1 0.4 0.285714 0.785714 0.66875  
0.333333 0.416667 0.375 0.5 1 1 0.5 0.285714 0.6642857145  
0.875 0.590909 0.5625 0.611111 0.3 0.5 1 0.642857 0.6642857145  
0.583333 0.333333 0.625 0.625 1 0.666667 0.125 0.428571 0.65  
0.666667 0.5 0.75 0 1 0.666667 0.25 0.142857 0.65  
0.875 0.727273 0.875 0.5 0.1 0.9 0.714286 0.571429 0.64375  
0.416667 0.416667 1 0.5 1 0.666667 0.125 0.285714 0.642857143  
0.9375 0.590909 0.3125 0.444444 0.6 0.9 0.142857 0 0.6  
0.875 0.681818 0.3125 0.722222 0.7 0.5 0.714286 0.5 0.6  
0.375 0.727273 0.6875 0.111111 0.3 0.4 0.5 0.714286 0.59375

0.666667 0.5 0.75 0.5 0.5 0.666667 0.25 0.571429 0.592857143  
0.75 0.772727 0.5 0.388889 0.1 0.3 0.5 0.857143 0.5875  
0.583333 0.333333 0.875 0.5 1 1 0.5 0.571429 0.5833333335  
0.75 0.166667 0.5 0.5 1 1 0 0.428571 0.58125  
0.875 0.909091 0.8125 0.666667 0.9 0.9 0.428571 0.357143 0.58125  
0.5 0.583333 0.875 0.75 0.5 1 0.375 0.714286 0.577777778  
0.875 0.727273 0.3125 0 0.7 0.6 0.285714 0.214286 0.575  
0.8125 0.727273 0.4375 0.333333 0.2 0.3 0.5 0.642857 0.575  
0.625 0.727273 0.4375 0.388889 0.3 0.5 0.642857 0.642857 0.575  
0.25 0.75 0.5 0.625 0.5 0.666667 0.25 0.142857 0.561111111  
0.5625 0.772727 0.5 0.166667 0.2 0.4 0.571429 0.714286 0.55625  
0.5 0.5 0.875 0.125 0.5 1 0 0.285714 0.538888889  
0.666667 0.666667 0 0.75 0 1 0.375 0.714286 0.5333333335  
0.916667 0.916667 0.125 0.5 1 0.666667 0.25 0.142857 0.527777778  
1 0.416667 0.5 0.375 1 0.666667 0.125 0 0.525  
0.333333 0.333333 0.625 0.625 1 1 0.125 0.428571 0.522222222  
0.5 0.636364 0.5625 0.277778 0.4 0.2 0.642857 0.571429 0.522222222  
0.6875 0.772727 0.75 0.333333 0.1 0.4 0.571429 0.571429 0.5125  
0.833333 0.25 0.875 0.25 1 0 0.125 0.142857 0.511111111  
0.916667 0.833333 0.625 0.5 1 0.333333 0.25 0.428571 0.50625  
0.333333 0.916667 0.75 0.75 0.5 0.666667 0.5 0.571429 0.50625  
0.875 0.818182 0.4375 0.277778 0.8 0.5 0.142857 0.428571 0.49444444445  
0.6875 0.590909 0.4375 0.277778 0.6 0.9 0.214286 0.428571 0.49444444445  
0.75 0.583333 0.625 0.875 0.5 0.333333 0.125 0.285714 0.49444444445  
0.9375 0.5 0.3125 0.555556 0.8 0.9 0.785714 0.357143 0.49375  
0.875 0.681818 0.375 0.5 0.5 0.9 0.214286 0.0714286 0.48125

0.8125 0.454545 0.4375 0.833333 0.6 1 0.642857 0.285714 0.48  
0.666667 0.916667 0.25 0.5 0.5 0.666667 0.375 0.428571 0.48  
0.4375 0.590909 0.4375 0.611111 0.2 0.6 0.785714 0.357143 0.4722222222  
0.583333 0.416667 0.875 0.375 1 1 0.375 0.285714 0.4722222222  
0.5625 0.681818 0.5625 0.277778 0.3 0.4 0.571429 0.5 0.47142857145  
0.8125 0.727273 0.5625 0.277778 0.6 0.6 0.142857 0.357143 0.46666666665  
0.416667 0.75 0.375 0.625 0.5 0.666667 0.25 0.285714 0.4625  
0.583333 0.416667 0.75 0.375 0.5 1 0.25 0.571429 0.4590909091  
0.6875 0.636364 0.625 0.555556 0.6 1 0.571429 0.285714 0.45  
0.625 0.590909 0.5 0.333333 0.2 0.4 0.714286 0.642857 0.44545454545  
0.75 0.454545 0.3125 0.611111 0.7 0.7 0.785714 0.214286 0.44444444445  
0.666667 0.916667 0.25 0.625 0.5 0.666667 0.5 0.285714 0.43333333335  
0.25 0.833333 0.625 0.625 0 0.333333 0.375 0.142857 0.43  
0.6875 0.727273 0.25 0.5 0.6 0.7 0.285714 0.214286 0.41875  
0.75 1 0.75 1 1 0.666667 0.625 0.142857 0.415  
0.416667 0.666667 0.5 0.375 0.5 0.666667 0.25 0.285714 0.4090909091  
0.625 0.818182 0.625 0.777778 0.4 0.9 0.214286 0.357143 0.4  
0.25 0.5 0.5 0.625 0 0.666667 0.5 0.428571 0.395  
0.416667 0.75 0.5 0.625 0 0.666667 0.5 0 0.39

#### Anexo 4: Red neuronal en fase de validación

##### Entradas Red Neuronal en Fase Validación 8:3:4:1

0.875	0.681818	0.375	0.5	0.5	0.9	0.214286	0.0714286
0.5	0.636364	0.5625	0.277778	0.4	0.2	0.642857	0.571429
0.25	0.5	0.5	0.625	0	0.666667	0.5	0.428571
0.75	0.166667	0.5	0.5	1	1	0	0.428571
0.666667	0.5	0.75	0	1	0.666667	0.25	0.142857
0.875	0.909091	0.8125	0.666667	0.9	0.9	0.428571	0.357143
0.8125	0.727273	0.4375	0.333333	0.2	0.3	0.5	0.642857
1	0.416667	0.5	0.375	1	0.666667	0.125	0
1	0.666667	0.5	0.625	1	0.666667	0.625	0.428571
0.6875	0.727273	0.25	0.5	0.6	0.7	0.285714	0.214286
0.875	0.590909	0.5625	0.611111	0.3	0.5	1	0.642857
0.833333	0.25	0.875	0.25	1	0	0.125	0.1428571
0.75	0.5	0.75	0.25	0.5	0	0.25	0
0.875	0.681818	0.3125	0.722222	0.7	0.5	0.714286	0.5
0.4375	0.5	0.75	0.611111	0.4	0.5	0.5	0.642857
0.6875	0.772727	0.75	0.333333	0.1	0.4	0.571429	0.571429
0.625	0.818182	0.625	0.777778	0.4	0.9	0.214286	0.357143
0.5625	0.681818	0.5625	0.277778	0.3	0.4	0.571429	0.5

##### Pesos de Red Neuronal para la Validación

0.096
0.133
-0.379
-0.410
0.590
-0.226
-0.477
-0.547

##### Salida Red Neuronal en topología 8:3:4:1

1	0.550944
2	0.583668
3	0.412834
4	0.67849
5	0.64949
6	0.62485

7	0.596768
8	0.582294
9	0.71242
10	0.502323
11	0.486638
12	0.516194
13	0.471783
14	0.650269
15	0.68545
16	0.593131
17	0.406101
18	0.555935