UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA

FACULTAD DE INGENIERÍA

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

PREDICCIÓN DE DESEMPEÑO ACADÉMICO POR MEDIO DE REDES NEURONALES

POR:

JESÚS RAMÓN CARMONA JÁQUEZ

TESIS PRESENTADA COMO REQUISITO PARA OBTENER EL GRADO DE

MAESTRO EN CIENCIAS BÁSICAS

CHIHUAHUA, CHIH., MÉXICO **MAYO 2022**

UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA

Predicción de desempeño académico por medio de redes neuronales. Tesis presentada por *Jesús Ramón Carmona Jáquez* como requisito parcial para obtener el grado de *Maestro en Ciencias Básicas*, ha sido aprobada y aceptada por:

M. I. Javier González Cantú
Director de la Facultad de Ingeniería

Dr. Alejandro Villalobos Aragón
Secretario de Investigación y posgrado

M.S.I. Karina Rocío Requena Yañez
Coordinadora Académica

Dr. José Luis Herrera Aguilar
Director de tesis.

Mayo 2022.

Comité:
Dr. José Luis Herrera Aguilar
Dr. Octavio Hinojosa de la Garza
Dr. Cornelio Álvarez Herrera
Dr. Antonio Daniel Rivera López.

UNIVERSIDAD AUTÓNOMA DE
CHIHUAHUA

30 de mayo de 2022.

ING. JESÚS RAMÓN CARMONA JÁQUEZ
Presente.

En atención a su solicitud relativa al trabajo de tesis para obtener el grado de Maestro en Ciencias Básicas, nos es grato transcribirle el tema aprobado por esta Dirección, propuesto y dirigido por el director **Dr. José Luis Herrera Aguilar** para que lo desarrolle como tesis, con el título **"PREDICCIÓN DE DESEMPEÑO ACADÉMICO POR REDES NEURONALES"**.

Índice de Contenido

FACULTAD DE INGENIERÍA
Circuito No.1, Campus Universitario 2
Chihuahua, Chih., México. C.P. 31125
Tel. (614) 442-95-00
www.fing.uach.mx

UNIVERSIDAD AUTÓNOMA DE
CHIHUAHUA

4.6. Modelo 05
4.7. Modelo 06
4.8. Modelo 07
4.9. Modelo 08
4.10 Resultados adicionales

5. Conclusiones
   5.1. Conclusiones

Referencias

Solicitamos a Usted tomar nota de que el título del trabajo se imprima en lugar visible de los ejemplares de las tesis.

A T E N T A M E N T E
"Naturam subiecit aliis"

EL DIRECTOR

M.I. JAVIER GONZÁLEZ CANTÚ

SECRETARIO DE INVESTIGACIÓN Y POSGRADO

DR. ALEJANDRO VILLALOBOS ARAGÓN

FACULTAD DE INGENIERÍA
Circuito No.1, Campus Universitario 2
Chihuahua, Chih., México. C.P. 31125
Tel. (614) 442-95-00
www.fing.uach.mx

Dedicatoria

Este trabajo es la culminación de una meta, pocas cosas han estado tan metidas en mi cabeza como el hambre de terminar una tesis tan interesante como esta creo que es. Un trabajo así va dirigido a quien guste de leerlo, ya que creo fervientemente que el conocimiento debe de ser compartido lo más posible con todos a nuestro alrededor. Por otro lado, esta tesis solo ha sido posible gracias al doctor Herrera, ya que si no hubiese sido por su invitación a la maestría, no habría podido ser parte de este proyecto.

Ahora bien, quienes me conocen saben que esta tesis, y en general todo lo que puedo lograr hacer va dirigido a quienes más me han apoyado. Mis padres desde siempre me han ayudado a seguir mis metas, y eso mismo ha sido de los más grandes detonantes en todos mis logros.

Por último, he dejado lo mejor para el final, Abigail, este trabajo, todo el esfuerzo, todo lo que hemos podido lograr juntos hasta ahora me ha dado la fuerza para concluir este proyecto, y esto solo es un escalón más de todo lo que viene.

Resumen

El aprendizaje computacional es un área de la inteligencia artificial bastante utilizada en la actualidad, su funcionamiento básico permite a un sistema "aprender" por medio de datos en lugar de utilizar programación explícita. Su influencia va más allá de estudios relacionados a problemas en computación e internet, impactando en áreas como la educación, ciencia de materiales, biología computacional, salud, finanzas, entre otras. El implementar redes neuronales a bases de datos provenientes de instituciones educativas es un área que se ha desarrollado desde hace algunos años. En este trabajo se aplicaron técnicas de aprendizaje computacional y redes neuronales con el propósito de entrenar un modelo de red neuronal que sea capaz de predecir el estatus académico de los estudiantes de la Universidad Autónoma de Chihuahua en base a descriptores académicos y socio económicos. A través de 9 modelos distintos, especializados en diferentes puntos de la vida académica de los estudiantes, logramos alcanzar precisiones superiores al 90 %, lo que se traduce en una potencial mejoría en los programas de seguimiento interno de la universidad.

Índice general

Índice de figuras

CAPÍTULO: 1

Introducción

1.1. Introducción

El aprendizaje computacional es un área de la inteligencia artificial bastante utilizada en la actualidad, su funcionamiento básico permite a un sistema "aprender" por medio de datos en lugar de utilizar programación explícita. Su influencia va más allá de estudios relacionados a problemas en computación e internet, impactando en áreas como la educación, ciencia de materiales, biología computacional, salud, finanzas, entre otras (Jordan y Mitchell, 2015) (Das, Dey, Pal, y Roy, 2015). A pesar de su reciente crecimiento y extensión a tantas áreas, el aprendizaje computacional no es algo nuevo, la idea de inteligencia artificial data de alrededor de 1950. En 1959 Arthur Lee Samuels utilizó por primera vez el termino de aprendizaje computacional (Samuel A.L., 1959). Con el pasar del tiempo se han refinado y mejorado constantemente los métodos utilizados en el desarrollo del área de inteligencia artificial. En años recientes gracias al desarrollo de mejores y más baratas herramientas computacionales, se ha generado un renovado interés en la inteligencia artificial y el aprendizaje computacional. El surgimiento de grandes bases de datos en distintas áreas, la capacidad de distribuir tareas entre computadoras, la creación de procesadores más poderosos y económicos son solo algunos de los detonantes que han impulsado el desarrollo de manera tan significativa en el área durante los últimos años.

El implementar redes neuronales a bases de datos provenientes de instituciones educativas es un área que se ha desarrollado desde hace algunos años (Treasure-Jones, Sarigianni, Maier, Santos, y Dewey, 2019). La inteligencia artificial se ha tornado en una herramienta efectiva a la hora de predecir el desempeño académico de los estudiantes de educación superior (Aydogdu, 2020). Estas herramientas ven su efectividad aumentar gracias a la existencia de masivas bases de datos que son sencillas de guardar y procesar (Aydogdu, 2020). La minería de datos es un término nacido de la creciente necesidad de obtener información que sean considerada de utilidad para problemas específicos de una fuente de datos de gran tamaño. Esta técnica consiste en la extracción de dicha información útil a través de algoritmos de clasificación, regresión, etc.(Alsalman, Khamees Abu Halemah, Alnagi, y Salameh, 2019).

En lo que respecta a la educación superior, el creciente número de estudiantes que desertan sus estudios se ha tornado en un problema a nivel global. Habiendo muchos factores en esta problemática, como el contexto socio-económico, calidad de la educación media superior, entre otros (Bassi, Dada, Hamidu, y Elijah, 2019)(Francis y Babu, 2019). Dado que el desempeño acádemico de los estudiantes es una indicador de gran importancia para las instituciones educativas por factores como la reputación o la asignación de apoyos gubernamentales (Francis y Babu, 2019), el predecir estos desempeños con el fin de mejorarlos es una área de oportunidad para la inteligencia artificial y minería de datos (Alsalman y cols., 2019). La predicción del desempeño académico puede tornarse en un desafío debido a la gran cantidad de estudiantes que hay en las instituciones(Francis y Babu, 2019).
Dicho esto, los repositorios de centros de educación y universidades se vuelven uno de los principales medios para la obtención de bases de datos.(Waheed y cols., 2020). En la literatura el uso de redes neuronales para la predicción del desempeño académico ha mostrado resultados bastante prometedores (Aydogdu, 2020; Alsalman y cols., 2019; Francis y Babu, 2019; Bassi y cols., 2019). Utilizando distintos parámetros tales como el contexto socio-económico (Bassi y cols., 2019), calificaciones, género, etc.(Aydogdu, 2020) distintas investigaciones han dejado un camino a seguir en orden de mejorar como se predice el desempeño académico.

Con un camino tan claro a nuestras espaldas sobre como se puede predecir el desempeño académico de los estudiantes es momento de empezar a trazar nuestro aporte, el objetivo de esta investigación es utilizar técnicas de aprendizaje computacional y redes neuronales sobre una base de datos real de estudiantes de

la Universidad Autónoma de Chihuahua, con el fin de crear y entrenar un modelo de red neuronal capaz de predecir el estatus académico que tendrá un estudiante basándonos en su desempeño en semestres anteriores y en su educación media superior. Para alcanzar esta meta será necesario limpiar los datos con los que se trabajara, además de necesitar de una amplia investigación para elegir el tipo de arquitectura que se utilizará en el modelo final. En las siguientes secciones se profundizará en las matemáticas sobre las que se sustentan los modelos a utilizar, se establecerán los parámetros de nuestro modelo y se presentarán los resultados para discutirlos.

CAPÍTULO: 2

Marco teórico

2.1. Fundamentación Teórica

El área de la inteligencia artificial denominada aprendizaje computacional se compone de una serie de metodologías que permiten generar predicciones a partir de un conjunto de datos. El aprendizaje computacional le permite a un sistema el "aprender" a través de los datos en lugar de utilizar programación explícita, usando una variedad de algoritmos que de manera iterativa mejoren el como describen y predicen el comportamiento de los datos. Este conjunto de datos, también llamado base de datos, se introduce en un algoritmo, haciendo una partición entre los datos que se utilizan para "entrenar" el sistema y los que se utilizan para probar su desempeño.

Los algoritmos se dividen entre algoritmos de aprendizaje supervisado y no supervisado, al mismo tiempo que entre ellos existen los algoritmos de clasificación y de regresión, centrando nuestro interés particular en los de clasificación. Los algoritmos de clasificación nos permiten dividir cosas entre grupos que comparten características similares entre si. Entre los algoritmos de clasificación más prominentes en la actualidad se encuentran las redes neuronales artificiales. Las redes neuronales artificiales están diseñadas para imitar la arquitectura del cerebro humano y sus conexiones entre neuronas, esto a través de la conexión de múltiples *perceptrones*. Por supuesto, si se busca entender que es una red

neuronal el primer paso surge de manera natural, si una red se compone de múltiples perceptrones entonces es necesario entender primero que es un perceptrón y como funciona.

2.2. El Perceptrón

El modelo del perceptrón data del año 1962 (Rosenblatt, 1962). En su forma más simplificada corresponde a un modelo que clasifica dos clases de datos ζ_1 y ζ_2, en donde se utiliza de entrada un vector x de dimensión n de la forma $(x_1, x_2, ..., x_n)$,donde cada x_i representa una característica del dato en cuestión, mediante una transformación no lineal se asocia el vector de entrada con los pesos del perceptrón representados por $w_1, w_2, \ ... \ , w_m$, a los cuales se les añade un valor de umbral b, introduciendo el resultado en una función de activación f (Figura 2.1), dando la forma:

$$v = \sum_{i=1}^{m} w_i x_i + b. \tag{2.1}$$

El objetivo del perceptrón es clasificar correctamente el vector de entrada x en una de las clases ζ_1 o ζ_2, cuyas regiones se encuentran separadas por el hiperplano definido por:

$$\sum_{i=1}^{m} w_i x_i + b = 0.$$

De modo que si la ecuación (2.1) es mayor que 0 entonces la entrada x pertenece a la clase ζ_1 y si la ecuación (2.1) es menor que 0 la entrada x pertenece a la clase ζ_2. Para realizar la clasificación es necesario ajustar el valor de los pesos en cada paso de tiempo, los pesos se actualizarán mediante funciones de minimización de error, donde el error se mide como el número de elementos mal clasificados. El error se minimiza ajustando los pesos en cada paso de tiempo discreto, buscando que el error decrezca utilizando el algoritmo de convergencia del perceptrón.

Para que el perceptrón funcione correctamente el conjunto de datos debe de ser separable de manera lineal en dos clases ζ_1 y ζ_2 (Figura Apartado 2.2.1). Que un conjunto de datos sea linealmente separable significa que las clases tienen la suficiente distancia entre ellas como para que un hiper plano las divida (ecuación

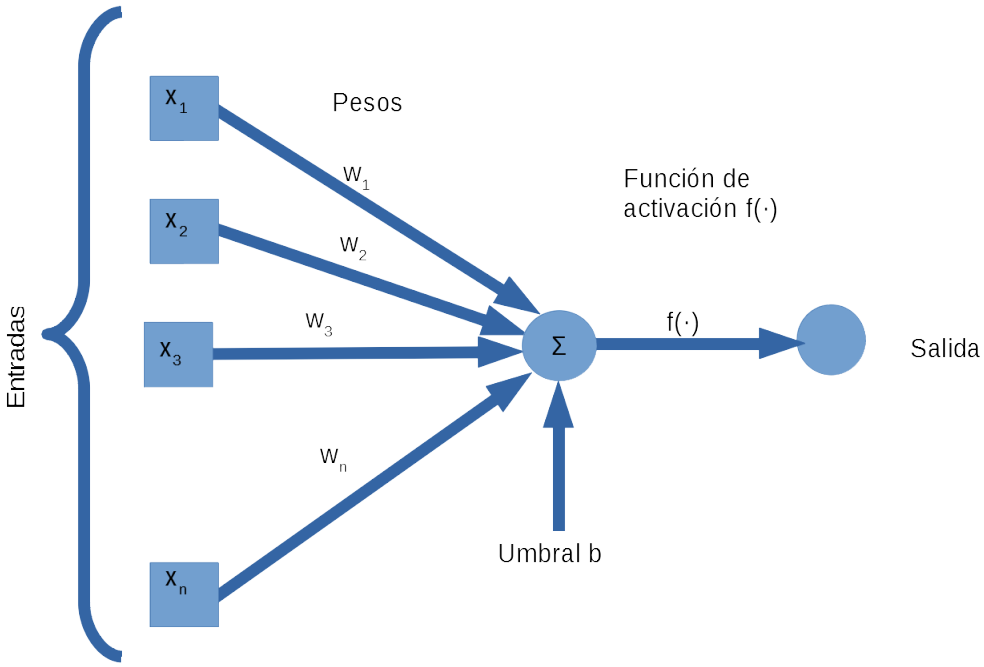


Figura 2.1: Representación gráfica del perceptrón.

(2.2)), es decir, si existe un vector de pesos w tal que:

$$\begin{cases} w^T x > 0, & x \in \zeta_1 \\ w^T x < 0, & x \in \zeta_2. \end{cases}$$

Si dicha condición no se cumple entonces el problema está fuera de la capacidad de computo de un perceptrón.

2.2.1. Teorema de convergencia del perceptrón

A fin de explicar como funciona el algoritmo de convergencia del perceptrón es necesario establecer el teorema sobre el cual se sustenta, brindando su consecuente demostración en las siguientes secciones.

Teorema de convergencia del perceptrón 2.2.1 *Sean los conjuntos de vectores de entrenamiento $\aleph_1$ y $\aleph_2$ linealmente separables. Sean los vectores de entrada*

$$x = (x(1), x(2), ..., x(n)), \tag{2.2}$$

donde $x \in \aleph_1 \cup \aleph_2$. El perceptrón converge después de n_o pasos de modo que el vector de pesos $w(n_o)$ está dado por:

$$w(n_o) = w(n_o + 1) = w(n_o + 2) = ... \tag{2.3}$$

es un vector de solución para $n_0 \leq n_{\text{máx}}$, esto quiere decir que para cualquier paso subsecuente la actualización del valor de vector de pesos debe ser 0 o casi 0 ya que la solución fue encontrada.

A fin de probar que es posible minimizar el error y que el mismo converge a un único valor para cualquier perceptrón independientemente del valor del vector de pesos w es necesario establecer lo siguiente.

El vector de entrada de dimensión $(m + 1)$ se denotará como:

$$x(n) = (+1, x_1(n), x_2(n), ..., x_m(n)) \tag{2.4}$$

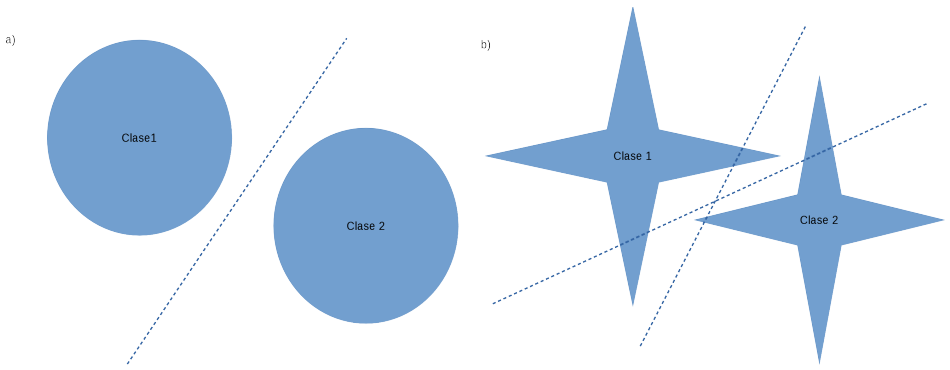
con n representando el *paso* o número de iteración en que se encuentra el algoritmo(Este vector corresponde a cualesquiera de los $x(n)$ en (2.2)). Dado que el

vector de pesos tiene la misma dimensión que el vector de entrada, este se puede definir como:

$$w(n) = (b, w_1(n), w_2(n), ..., w_m(n)) \tag{2.5}$$

Ahora haremos el producto interno de las ecuaciones (2.4) y (2.5) ya definidas es posible calcular el producto interno entre ambas como:

$$\begin{aligned} v(n) &= \sum_{i=0}^{m} w_i(n)x_i(n) \\ &= w^T(n)x(n). \end{aligned}$$



(a) Conjuntos linealmente separables. (b) Conjuntos que no son linealmente separables.

El algoritmo de ajuste de los pesos se realiza de la siguiente manera:

1. Si el n-ésimo paso de tiempo, el elemento $x(n)$ está clasificado de manera correcta por el vector de pesos $w(n)$, entonces no es necesario hacer correcciones y
$$w(n+1) = w(n)$$

2. Si el elemento no fue clasificado correctamente el vector se actualiza con la regla:

$$\begin{cases} w(n+1) = w(n) - \eta(n)x(n), & \text{si } w^T x(n) > 0 \text{ y } x(n) \in \zeta_2. \\ w(n+1) = w(n) + \eta(n)x(n), & \text{si } w^T x(n) \leq 0 \text{ y } x(n) \in \zeta_1. \end{cases}$$

Donde η es el parámetro de aprendizaje. El parámetro η puede o no depender del número de iteración. Si $\eta(n) = \eta > 0$, donde η es una constante independiente de la iteración n, entonces el valor de los pesos del perceptrón se incrementa o decrementa de manera fija.

2.2.2. Demostración del teorema de convergencia

Para los lectores que no estén interesados en la demostración del teorema, en resumen el teorema dice que para cualquier perceptrón en proceso de entrenamiento existe un número de pasos, después de los cuales estos pesos ya no cambian, llegando a una solución. La calidad de esa solución se mide de acuerdo al problema particular, pero lo importante de esta demostración es que existe una solución calculable para cualquier perceptrón.

La siguiente demostración está basada en la demostración que aparece en el libro (Haykin, 1990). Sea la condición inicial $w(0) = 0$ y un vector de entrada $x(n)$ que pertenece a la clase uno, donde $\aleph_1$ representará a los vectores etiquetados en la clase uno y $\aleph_2$ los de la clase dos, pero que $w^T(n)x(n) \leq 0$ para todo $n = 1, 2, ...,$, es decir que el vector de entrada $x(n)$ está clasificado erróneamente. Considerando la constante $\eta(n) = 1$ entonces la actualización de pesos se realiza como:

$$w(n+1) = w(n) + x(n), \text{ para } x(n) \in \zeta_1. \tag{2.6}$$

Con la condición inicial $w(0) = 0$, la ecuación se resuelve para $w(n+1)$

$$\begin{cases} w(n) & = x(1) + x(2) + ... + x(n-2) + x(n-1) \\ w(n+1) & = x(1) + x(2) + ... + x(n-1) + x(n) \end{cases} \tag{2.7}$$

Suponemos las dos clases ζ_1 y ζ_2 como linealmente separables, entonces existe una solución w_o para la que $w_o^T x(n) > 0$ para cualquier vector x que pertenezca al conjunto de vectores $\aleph_1$. Para dicha solución se define el número positivo α

$$\alpha = \min_{x(n) \in \aleph_1} w_o^T x(n).$$

Con α definido se multiplican ambos lados de la ecuación (2.7) por la solución del vector de pesos w_o^T, obteniendo:

$$w_o^T w(n+1) = w_o^T x(1) + w_o^T x(2) + ... + w_o^T x(n) \ ,$$

y de acuerdo a la definición de α

$$w_o^T w(n+1) \geq n\alpha \ . \tag{2.8}$$

Lo siguiente es aplicar la desigualdad de Cauchy - Schwarz, la cual dice que para dos vectores w_o y $w(n+1)$ se cumple:

$$||w_o||^2||w(n+1)||^2 \geq \left[w_o^T w(n+1)\right]^2,$$

donde $||\cdot||$ representa la norma del vector y el producto interno $w_o^T w(n+1)$ es un escalar. Usando la ecuación (2.8) es posible deducir que $\left[w_o^T w(n+1)\right]^2 \geq n^2\alpha^2$. En este caso se optó por recurrir a Cauchy - Schwarz con el fin de lograr despejar $w(n+1)$, ya que no existe tal cosa como división entre un vector, esta desigualdad permite continuar el despeje de $w(n+1)$ en función de α. Entonces se llega a:

$$\begin{aligned} ||w_o||^2||w(n+1)||^2 \geq \left[w_o^T w(n+1)\right]^2 &\geq n^2\alpha^2 \\ ||w_o||^2||w(n+1)||^2 &\geq n^2\alpha^2 \end{aligned}$$

que puede despejarse para $||w(n+1)||$ como:

$$||w(n+1)||^2 \geq \frac{n^2\alpha^2}{||w_o||^2}.$$

Terminando de despejar el valor siguiente del peso en función de alpha retomamos la ecuación (2.6) con k como índice.

$$w(k+1) = w(k) + x(k), \text{ para } k = 1, ..., n \text{ y } x(k) \in \aleph_1$$

Calculando la norma al cuadrado de ambos lados se obtiene:

$$||w(k+1)||^2 = ||w(k)||^2 + ||x(k)||^2 + 2w^T(k)x(k)$$

Como se definió al principio de la demostración $w^T(k)x(k) \leq 0$, ya que el elemento se considera mal clasificado, la ecuación anterior se puede simplificar a

$$||w(k+1)||^2 \leq ||w(k)||^2 + ||x(k)||^2,$$

ya que todas las normas euclidianas son mayores a 0, eliminar el único elemento negativo permite dar sentido a la desigualdad. Prosiguiendo con el despeje de la norma de $w(k+1)$ se llega a :

$$||w(k+1)||^2 - ||w(k)||^2 \leq ||x(k)||^2, k = 1, ...n.$$

Sumando sobre todos los valores de k y respetando la condición inicial $w(0) = 0$, se llega a la desigualdad:

$$||w(k+1)||^2 \leq \sum_{k=1}^{n} ||x(k)||^2 \leq n\beta \tag{2.9}$$

donde β está dado por:

$$\beta = \max_{x(k)\in\aleph_1} ||x(k)||^2,$$

en este caso β entra a la desigualdad de ese modo ya que su definición de máximo nos asegura que al menos en un elemento $||x(k)||$ es menor que $\max_{x(k)\in\aleph_1} ||x(k)||^2$. Por la ecuación (2.7) se obtiene que el vector de pesos crece a lo mucho linealmente con el número de iteraciones n. n no puede ser más grande que un valor $n_{\max}$ cuya solución es

$$\frac{n_{\max}^2 \alpha^2}{||w_o||^2} = n_{\max}\beta.$$

Resolviendo para $n_{\max}$,

$$n_{\max} = \frac{\beta||w_o||^2}{\alpha^2}.$$

Todo esto para $\eta(n) = 1$, para toda n y $w(0) = 0$, suponiendo que el vector de solución w_o existe, prueba que la regla de actualización de pesos del perceptrón debe terminar en a lo mucho $n_{\max}$ iteraciones, o dicho de otro modo, nos asegura que despues de máximo $n_{\max}$ pasos se converge a una solución w_o.

2.3. Redes neuronales

Una red neuronal esta definida como múltiples capas de perceptrones (neuronas) que realizan a través de diversas operaciones una regresión en base a un conjunto de datos. Dichos modelos de regresión se basan en combinaciones lineales de funciones base $\phi_j(x)$ de la forma:

$$y(x, w) = f\left(\sum_{j=1}^{M} w_j \phi_j(x)\right) \ ,$$

donde $f(\cdot)$ es una función llamada *función de activación*. El objetivo de una red neuronal cualesquiera es lograr que las M funciones base $\phi_j(x)$, que dependen de parámetros, se ajusten a través de los coeficientes $\{w_j\}$ durante el *entrenamiento*.

Dado un conjunto de parámetros ordenados como un vector $\{t_n\}$, se minimiza la función de error

$$Error(w) = \frac{1}{2}\sum_{n=1}^{N} ||y(x_n, w) - t_n||^2,$$

es decir, encontrar un vector w con los valores de los pesos tal que $Error(w)$ sea lo más pequeño posible. Se utilizan valores iniciales del vector de pesos $w^{(0)}$ aleatorio tal que se realice una sucesión de pasos de la forma

$$w^{(i+1)} = w^{(i)} + \Delta w^{(i)}$$

donde i marca el número de la iteración en la que se encuentra el proceso. Dependiendo del algoritmo de optimización el incremento de los pesos $\Delta w^{(i)}$ se calcula diferente, y en base del cambio en valor de la función de error $\nabla Error(w)$ se repetirá el proceso que lleva a la actualización de los pesos hasta que $\nabla Error(w) = 0$ o casi 0. Una vez que se han puesto sobre la mesa las definiciones básicas de lo que se compone una red neuronal, es posible definir en lo particular el algoritmo de interés a utilizar.

2.3.1. Perceptrón multicapa

El perceptrón multicapa se basa en el perceptrón de una sola neurona de pesos ajustables, pero la diferencia radica en que cuenta con múltiples neuronas. Las principales características del perceptrón multicapa son:

- Cada neurona cuenta con una función de activación no lineal y diferenciable.
- La red cuenta con una o más capas *ocultas* entre las entradas y salidas.
- La red cuenta con bastantes conexiones entre las neuronas que la componen.

A pesar del potencial que estas características agregan al algoritmo, la presencia de no linealidad y la conectividad entre neuronas complican el análisis teórico del perceptrón multicapa, al mismo tiempo que las capas ocultas complican la vizualización del proceso de aprendizaje.

Cuando en una red todas las neuronas de una capa están conectadas con todas las neuronas de la capa siguiente, se dice que la red es *Completamente conectada* (Figura 2.3).

El método más utilizado para entrenar perceptrones multicapa es el algoritmo de propagación hacia atrás (*Back-propagation*)(McClelland, Rumelhart, Group, y cols., 1986), el cual consta de dos fases:

1. La fase *hacia adelante* donde la entrada de la red se distribuye hacia adelante capa por capa hasta alcanzar la salida.

2. La fase *hacia atrás* donde se evalua el error comparándolo con la salida deseada de la red.

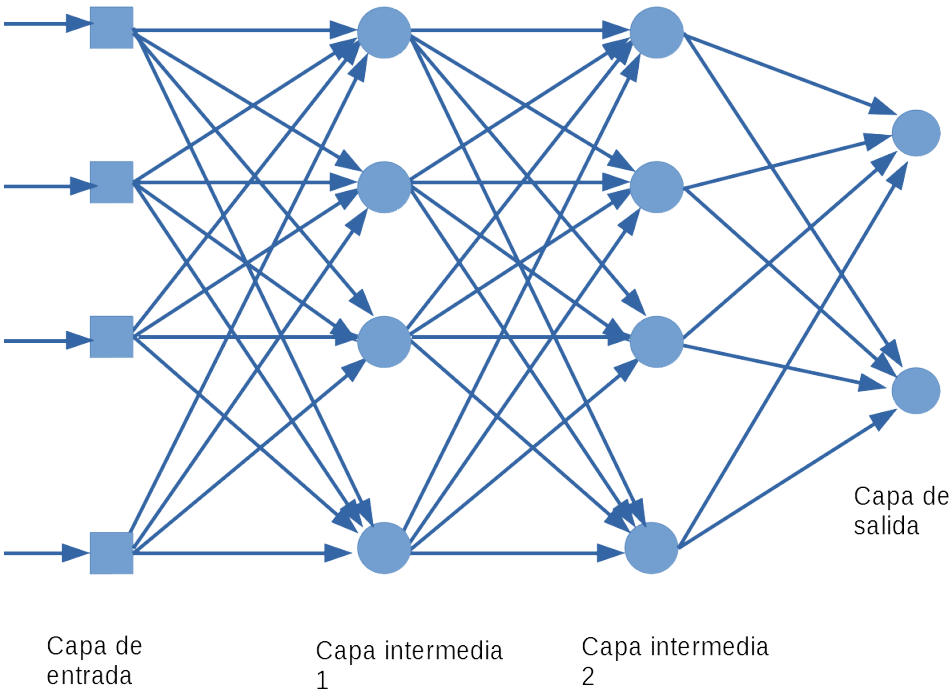


Figura 2.3: Red neuronal completamente conectada.

Estas dos fases se identifican entre si por los tipos de señales que emiten, siendo estas:

1. **Señales de función:** Es una señal de entrada que entra a la neurona en su primera capa, propagándose hacia adelante hasta generar una señal de salida. Se le llama "señal de función" debido a que cumple con una función en la salida de la red y también porque al pasar por cada neurona esta señal se somete a las funciones de activación de las mismas.

2. **Señales de error:** Señales que se originan en la salida de la red y se distribuyen hacia atrás capa por capa. Su nombre proviene de que su cálculo se realiza sobre cada neurona con una función dependiente del error.

Las neuronas de la última capa constituyen la capa de salida de la red mientras que las neuronas de la primer capa constituyen la capa de entrada. Entonces, todas las capas entre la capa de entrada y la de salida se denominan capas ocultas debido a que no interactúan de manera directa con la entrada y salida de la red. Cada capa oculta o de salida de una red de perceptrón multicapa cumple con dos funciones:

1. El cálculo de la señal de función como salida de cada neurona, expresado como una función continua no lineal asociada a los pesos de la neurona.

2. El cálculo de un gradiente estimado del error, necesario para actualizar los pesos a través del error.

Las neuronas en las capas ocultas cumplen la función de detectar características en la red, ya que con cada paso de tiempo que transcurre en el entrenamiento las mismas características comienzan a acentuarse más, simplificando la separación de las clases.

Continuando con el cálculo de errores retomaremos el hablar del algoritmo de propagación hacia atrás, pero para poder entenderlo hay que hablar de los dos principales métodos de aprendizaje que se utilizan en el entrenamiento de redes multi capa.

2.3.2. Aprendizaje por lotes y aprendizaje en línea.

Sea un perceptrón multicapa con al menos una capa oculta. Se define el conjunto de entrenamiento τ como:

$$\tau = \{x(n), d(n)\}_{n=1}^{M}.$$

Sea $y(n)$ la señal de función producida por una neurona j de la capa de salida de la red después de una entrada $x(n)$.La señal de error producida por la salida de la neurona j está dada por:

$$e_j(n) = d_j(n) - y_j(n) \tag{2.10}$$

donde $d_j(n)$ es la salida esperada de la neurona dada la entrada $x(n)$. Definimos ahora la *energía de error instantáneo* por la ecuación:

$$\varepsilon_j(n) = \frac{1}{2}e_j^2(n).$$

Sumando para todas las neuronas de la capa de salida el resultado es:

$$\varepsilon(n) = \sum_{j \in C} \varepsilon_j(n) \tag{2.11}$$

$$= \frac{1}{2}\sum_{j \in C} e_j^2(n), \tag{2.12}$$

en donde C es el conjunto de todas las neuronas de la capa de salida. Sabiendo que la muestra de entrenamiento consta de N elementos, se puede promediar el error para obtener la *energía promedio del error por muestra de entrenamiento* como:

$$\varepsilon_{av}(n) = \frac{1}{N}\sum_{n=1}^{N} \varepsilon(n)$$

$$= \frac{1}{2N}\sum_{n=1}^{N}\sum_{j \in C} e_j^2(n).$$

Este error depende también de los pesos, ya que el ajuste de los mismos tiene como objetivo el minimizar el error. Dependiendo de como se decida entrenar el perceptrón multi capa existen dos métodos de dividir los conjuntos de entrenamiento, siendo estas el aprendizaje por lotes y el aprendizaje en línea.

2.3.3. Aprendizaje por lotes.

En este método, los ajustes a los pesos de la red se realizan *después* de que todos los N ejemplos de la muestra de entrenamiento τ, lo cual se conoce como una *época* de entrenamiento. Este método provee de una manera de asegurar la convergencia a un mínimo local del error.

2.3.4. Aprendizaje en línea.

En este método, los ajustes de los pesos se realizan ejemplo por ejemplo. Considerando la época de N elementos de entrenamiento $\{x(1), d(1)\}$, $\{x(2), d(2)\}$,... ,$\{x(N), d(N)\}$. El primer ejemplo $\{x(1), d(1)\}$ se introduce a la red, se ajustan los pesos y se introduce el segundo ejemplo $\{x(2), d(2)\}$, ajustando nuevamente los pesos, repitiendo hasta llegar al elemento $\{x(N), d(N)\}$. Este proceso cuenta con la desventaja de no ser posible de paralelizar,dependiendo de cada paso individual para la actualización. Los elementos de entrenamiento se presentan a la red de manera aleatoria.

En resumen este método es sencillo de implementar y da soluciones efectivas pero es lento en comparación al método por lotes.

2.3.5. Algoritmo de propagación hacia atrás.

Ahora que se han establecido los métodos de aprendizaje es momento de definir como se hace la actualización de los pesos. Este proceso se realiza de la siguiente manera; considerando una neurona j a la que se le introdujeron una serie señales de función. La salida de la neurona está dada por:

$$v_j(n) = \sum_{i=0}^{m} w_{ji}(n) y_i(n), \tag{2.13}$$

donde m es el número de entradas que recibe la neurona j y el valor w_{j0} es el valor de umbral. La señal de función de la salida j de la neurona en el paso n es:

$$y_j(n) = \phi_j(v_j(n)) \tag{2.14}$$

En este algoritmo la correción de los pesos se realiza mediante un incremento $\Delta w_{ji}(n)$ que se calcula sobre la derivada parcial del error $\dfrac{\partial \varepsilon(n)}{\partial w_{ji}(n)}$. De acuerdo

a la regla de la cadena de cálculo, esto se simplifica a:

$$\frac{\partial \varepsilon(n)}{\partial w_{ji}(n)} = \frac{\partial \varepsilon(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)} \tag{2.15}$$

La derivada parcial $\frac{\partial \varepsilon(n)}{\partial w_{ji}(n)}$ representa el *factor de sensibilidad*, que determina la dirección hacia la que se actualiza el peso, es decir si se actualiza hacia arriba o hacia abajo.

Diferenciando la ecuación (2.12) con respecto a $e_j(n)$, se obtiene:

$$\frac{\partial \varepsilon(n)}{\partial e_j(n)} = e_j(n). \tag{2.16}$$

Luego, diferenciando la ecuación (2.10) con respecto a $y_j(n)$ llegamos a:

$$\frac{\partial e_j(n)}{\partial y_j(n)} = -1.$$

Lo siguiente es diferenciar (2.14) respecto a $v_j(n)$,:

$$\frac{\partial y_j(n)}{\partial v_j(n)} = \phi'(v_j(n)) \tag{2.17}$$

Por último, la derivada parcial de (2.13) respecto a $w_{ji}(n)$ lleva a:

$$\frac{\partial v_j(n)}{\partial w_{ji}(n)} = y_i(n) \tag{2.18}$$

El uso de todas las ecuaciones desde (2.16) hasta (2.18) en (2.15) conlleva:

$$\frac{\partial \varepsilon(n)}{\partial w_{ji}(n)} = -e_j(n)\phi'(v_j(n))y_i(n). \tag{2.19}$$

Entonces la corrección del error $\Delta w_{ji}(n)$ que se aplica a $w_{ji}(n)$ se define por la regla:

$$\Delta w_{ji}(n) = -\eta \frac{\partial \varepsilon(n)}{\partial w_{ji}(n)}, \tag{2.20}$$

donde η es el parámetro de aprendizaje del algoritmo.

En resumen, el algoritmo de propagación hacia atrás consta de fases:

1. **Inicialización:** Los pesos y umbrales se establecen de manera aleatoria en caso de que no exista información previa.
2. **Presentación del conjunto de entrenamiento:** Se somete la red a una época de entrenamiento.
3. **Computo hacia adelante:** La salida de la red se compara a los valores esperados para calcular el error.
4. **Computo hacia atrás:** Se calcula el error delta para actualizar los pesos (Ecuación (2.20)).
5. **Iteración o paso de tiempo:** Repetición de los pasos hacia adelante y atrás para las épocas programadas, donde en cada una el conjunto de entrenamiento se introduce de manera aleatorio.

2.3.6. Funciones de activación

En las secciones anteriores se ha hablado de funciones de activación que se aplican a las salidas de las neuronas con el propósito mejorar la calidad de la salida de las neuronas. Algunos ejemplos de estas funciones de activación son las siguientes:

- **Sigmoide:** También llamada función logística. Se aplica de manera independiente a cada elemento del vector y lo reduce al intervalo$(0, 1)$ mediante la función:
$$f(s_i) = \frac{1}{1 + e^{-s_i}}$$
- **ReLU (Rectified Linear Unit):** Denominada en ocasiones como función rampa o rectificador. Esta función suele obtener menos error que la función logística en algunos problemas. Su función matemática es la siguiente:
$$f(x) = max(0, x) \tag{2.21}$$
- **Softmax:** Esta función reduce los vectores de salida al rango $(0, 1)$, pero no se aplica de manera independiente a cada elemento de vector de salida, sino que se calcula por la función:
$$f(s_i) = \frac{e^{s_i}}{\sum_j^C e^{s_j}}$$

donde s_j son los resultados para cada clase de la clasificación.

2.4. Optimizadores

En los apartados de la sección anterior hablamos constantemente de minimizar el error, por lo que fundamentos matemáticos sobre porque es valido dicho procedimiento es necesario si se quiere continuar usando esos sistemas. En concreto daremos sustento a los dos métodos utilizados en el desarrollo de este proyecto; minimización por mínimos cuadrados y minimización por entropía.

2.4.1. Minimización por mínimos cuadrados.

Mínimos cuadrados es una técnica optimización en la que, por medio de datos, se busca encontrar una función que se aproxime a los datos de la mejor manera posible. Esta técnica en relación a aprendizaje computacional y redes neuronales se aplica minimizando el valor instantáneo de la función de costo:

$$\varepsilon(\hat{w}) = \frac{1}{2}e^2(n)$$

donde $e(n)$ es la señal de error en el paso de tiempo n y $\hat{w}$ es el vector de parámetros estimados. Derivando con respecto a al vector $\hat{w}$ se llega a:

$$\frac{\partial \varepsilon(\hat{w})}{\partial \hat{w}} = e(n)\frac{\partial e(n)}{\partial w}.$$

La señal del error está expresada como:

$$e(n) = d(n) - x^T(n)\hat{w}(n)$$

Entonces,

$$\frac{\partial e(n)}{\partial \hat{w}(n)} = -x(n)$$

y

$$\frac{\partial \varepsilon(\hat{w})}{\partial \hat{w}(n)} = -x(n)e(n).$$

Usando este último resultado, el algoritmo de minimización por mínimos cuadrados qued definido como:

$$\hat{w}(n+1) = \hat{w}(n) + \eta x(n) e(n).$$

El parámetro de aprendizaje η dictará el ritmo al que el algoritmo de mínimos cuadrados converge a una solución. En resumidas cuentas el algoritmo de mínimos cuadrados repite el cálculo del peso $\hat{w}(n+1)$ buscando que se cumpla:

$$\hat{w}(n+1) - \hat{w}(n) = 0$$

2.4.2. Entropía

El método de entropía cruzada es un modelo estocástico de optimización, el cual consta de un proceso iterativo dividido en dos fases:

1. Generación de datos muestra aleatorios de acuerdo al problema.
2. Actualizar los parámetros, o pesos, de modo que la muestra siguiente está mejor clasificada.

Este método tiene como principal característica rápidez y precisión para la actualización de reglas de aprendizaje(De Boer, Kroese, Mannor, y Rubinstein, 2005). Si nos adentramos más, la entropía es una cantidad definida para distribuciones de probabilidad que está relacionada con la medida de la cantidad de información que contiene una variable aleatoria. Puede decirse que la entropía es la información que una variable aleatoria contiene de si misma (Thomas M. Cover, 2006). Dado que nuestro interés principal es distinguir dos o más grupos(clases), entonces es necesario utilizar el término de *entropía relativa*, la cual es una forma de medir la distancia entre dos distribuciones de probabilidad. La entropía puede considerarse como una manera de medir la incertidumbre de una variable aleatoria (Thomas M. Cover, 2006).

Definición: La entropía $H(X)$ de una variable aleatoria discreta X está definida como:

$$H(X) = -\sum_{x \in X} p(x) \log p(x).$$

En donde $p(x)$ representa la probabilidad de que un elemento pertenezca a una de las clases a evaluar. La convención más común es utilizar logaritmo de base

2, por lo que la entropía se expresa en bits. Como la entropía está en función de la distribución X, esta no depende de los valores que toma la variable aleatoria X, solo de sus probabilidades.

El valor esperado E de una variable aleatoria $g(X)$ se escribe como:

$$E_p g(X) = \sum_{x \in X} g(x)p(x) \ ,$$

el valor de la entropía de X puede interpretarse como el valor esperado de la variable $log\frac{1}{p(X)}$, donde X se escribe acorde a su función de probabilidad.

$$H(X) = E_p \log \frac{1}{p(X)}$$

2.4.3. Entropía relativa

La entropía relativa $D(p||q)$ es una manera de medir la distancia entre dos distribuciones, también sirve para medir la ineficiencia de suponer que la distribución es q, cuando en realidad era p.

La entropía relativa o *distancia de Kullback-Leiber* entre dos funciones de probabilidad $p(x)$ y $q(x)$ está definida como:

$$\begin{cases} D(p||q) & = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)} \\ & = E_p \log \frac{p(X)}{q(X)}. \end{cases}$$

Por conveniencia se considerará que $0 \log \frac{0}{0} = 0$, $0 \log \frac{0}{q} = 0$ y $p \log \frac{p}{0} = \infty$. La entropía siempre es positiva o cero, cero solo en el caso $p = q$. Aunque en realidad no es una distancia como tal, es útil para visualizarla como la "distancia" entre distribuciones(Thomas M. Cover, 2006).

2.4.4. Optimizador Adam

Si bien el uso de entropía en la optimización de los pesos que llevan a la correcta clasificación es crucial, esta teoría no es totalmente aplicable a los métodos de aprendizaje computacional de manera directa, por lo que para realizar esta conexión con la teoría y la práctica es necesario acceder a herramientas y

métodos acordes a esta misma práctica. Para esto el lenguaje de programación Python y la librería Tensorflow hacen uso de optimizadores computacionales tales como *Adam*(Adaptive moment estimation, nombre en su idioma original), el cual es un método de optimización que requiere poca memoria y gradientes de primer orden. El método Adam calcula los parámetros de entrenamiento de manera individual a través de gradientes en dos momentos(Kingma y Ba, 2014; Sashank, Satyen, y Sanjiv, 2018).

2.5. Georreferenciación

La georreferenciación es una técnica que permite posicionar puntos con ciertas características en un sitema de coordenadas, estos sistemas pueden ser sistemas de coordenadas geográficas o proyectadas. Para este caso particular se recurrió a coordenadas geográficas. Esta técnica es fundamental en el análisis de datos geoespaciales, lo que conlleva a una correcta ubicación de información en un mapa. Los dos sistemas de coordenadas de más importancia están descritos de la siguiente manera.

- **Sistemas de coordenadas geográficas.** Describe los datos en término de las coordenadas de latitud-longitud asociadas a un plano geodésico especifico, usualmente la Tierra. El más común y extendido es el World Geodetic System 84 (WGS84), aunque en proyectos europeos se promueve el uso del European Terrestrial Reference System 89 (ETRS89).
- **Sistemas de coordenadas proyectadas.** Son coordenadas referidas a un plano en el que se ha proyectado parte de la superficie terrestre. Como es imposible una proyección no distorsionada de la superficie elíptica de la Tierra, estos sistemas se restringen a regiones pequeñas para minimizar el error. Uno de los sistemas más populares es el sistema de coordenadas universal transversal de Mercator (UTM).

CAPÍTULO: 3

El modelo

El objetivo de este proyecto es desarrollar y entrenar redes neuronales capaces de predecir el estatus académico de los estudiantes. Como ya fue tratado en la sección 2.3.1, el modelo de red neuronal de perceptrón multicapa cuenta con una gran versatilidad, por lo que este tipo de modelo de red neuronal fue el elegido en el desarrollo del modelo capaz de predecir el estatus académico de los estudiantes. Este modelo se programó en el lenguaje de programación *python*, haciendo uso de la librería más popular en la implementación de inteligencia artificial de acuerdo a la mayoría de los artículos revisados en el desarrollo de este proyecto, la cual es *Tensorflow*. Una de las partes más importantes en el desarrollo de una red neuronal es el proceso de entrenar la red para que esta clasifique correctamente los datos, el proceso de entrenamiento se compone de una fase de entrenamiento y otra de prueba, para los cuales se requiere el uso los datos que se busca clasificar. Al saber de antemano a que clase pertenencen los datos, el entrenamiento se denomina aprendizaje supervisado. Dado que no es prudente utilizar todos los datos para entrenar la red, ya que esto genera un sobre entrenamiento que a su vez reduce la flexibilidad de la red para clasificar datos fuera del conjunto de entrenamiento, de manera predeterminada se acostumbra una proporción de 80 % de datos de entrenamiento contra 20 % de datos de prueba. Donde los datos de prueba fueron previamente limpiados y codificados.

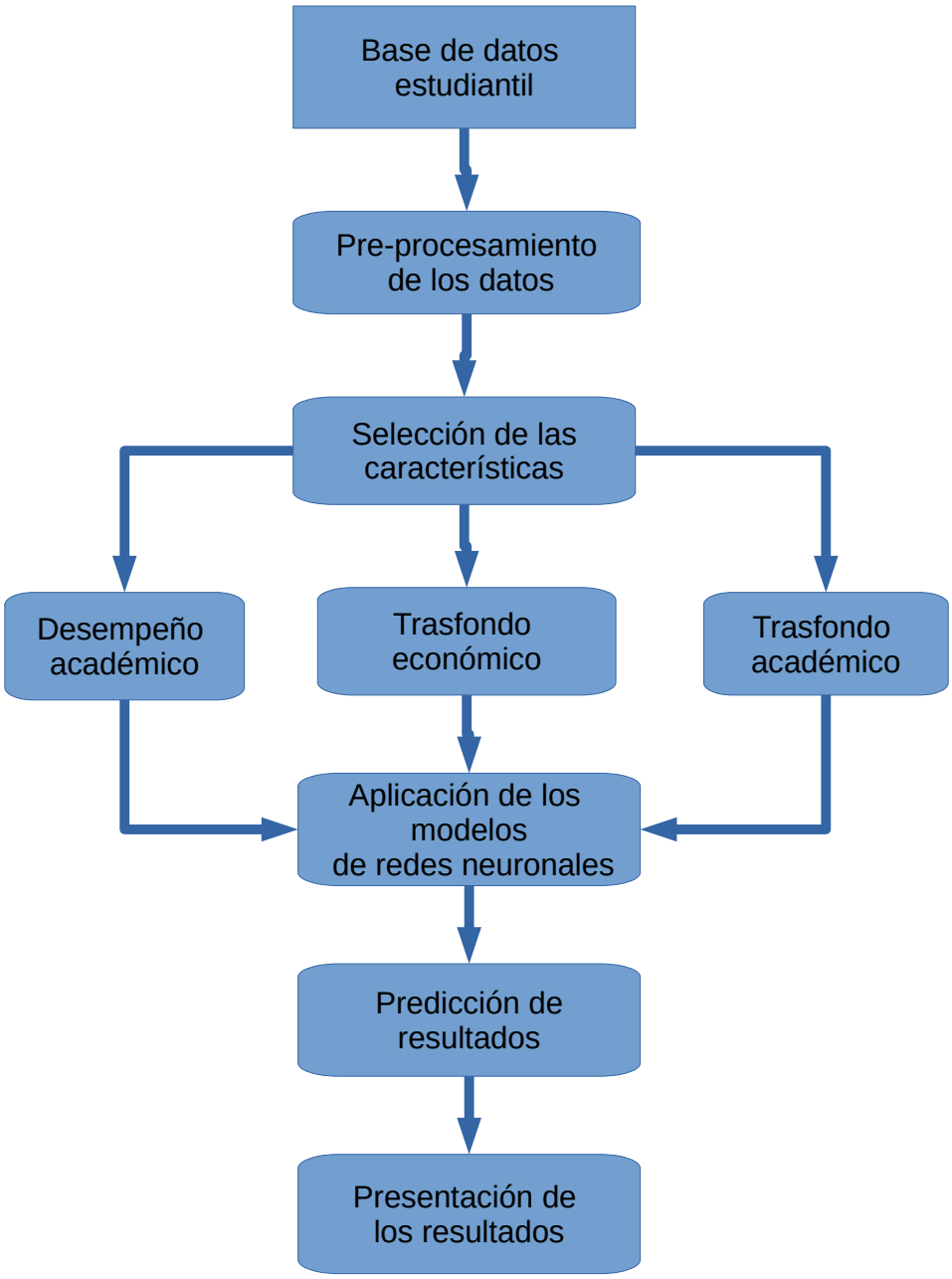


Figura 3.1: Diagrama de flujo que muestra de manera simplificada los pasos que se siguieron a la hora de desarrollar nuestros modelos de red neuronal. Si bien puede parecer sencillo a simple vista, cada paso requiere un extremo cuidado en el manejo de los datos y en la selección de los parámetros que definirán el entrenamiento.

3.1. La base de datos

La base de datos es una parte crucial en el desarrollo de redes neuronales, ya que los datos deben de contener información acerca del problema a resolver. La base de datos que se utilizó consta de datos académicos de estudiantes de la universidad de las generaciones 2012 al 2014. Los datos se utilizaron respetando los reglamentos de privacidad, ya que no incluyen ni nombre ni identificadores reales, utilizando un identificador único (índice), manteniendo seguros los datos. La base contiene múltiples entradas de cada estudiante, correspondiendo cada duplicado a una materia distinta cursada por el estudiante. El primer punto a tratar en la limpieza de la base de datos fue obtener los datos con los que no se contaba explícitamente pero que resultaban de interés, entre estos datos se encuentran los promedios semestrales e índices de aprobación, además de los medidores socio-económicos, para los cuales se utilizó geo referenciación sobre las direcciones contenidas en la base de datos. Las direcciones se ubicaron mediante coordenadas latitud y longitud, después se utilizaron las bases de datos del CONEVAL (Consejo Nacional de la Evaluación de la Política de Desarrollo Social) para asociar las direcciones a un indicador socio económico. El indicador seleccionado fue el *grado de marginación*, el cual se relacionó por municipio de residencia a cada dato.

Si bien la base de datos contiene mucha información, y queda bastante completa al agregar las columnas adicionales, no todos los datos de la red son necesarios para hacer funcionar el modelo de red neuronal, principalmente porque se quiere evitar un sobre entrenamiento.Una manera de visualizar mejor la estructura de nuestra base de datos puede ser a través de tablas donde se describe cada atributo tanto en como se ve, que tipo de valor es y una breve descripción de ese valor (Tablas 3.1 y 3.2).

3.2. El modelo de red

Luego de la limpieza de los datos comenzó la construcción del modelo de clasificación. Para resolver distintos problemas se recurrió a generar diferentes modelos, debido a que no es lo mismo clasificar a un estudiante en su primer semestre que en el último, llegando a la decisión de múltiples modelos. En total se realizaron 9 modelos de clasificación, de los cuales tocaremos a profundidad

Atributo	Descripción	Tipo de dato
Idinscripcion	Valor de identificación único para cada estudiante.	Número entero de 6 a 7 dígitos.
Ingreso	Ciclo educativo y modalidad a la que ingresó el alumno.	Cadena de texto.
Genero	Género biológico del estudiante.	Cadena de texto.
Fecha de Nacimiento	Fecha de nacimiento del estudiante.	Fecha.
Estado	Estado de residencia del estudiante.	Cadena de texto.
Municipio	Municipio de residencia del estudiante.	Cadena de texto.
Colonia	Colonia de residencia del estudiante	Cadena de texto.
Calle	Calle de residencia del estudiante.	Cadena de texto.
NumeroExterior	Numero de la residencia.	Cadena de texto.
Lugar de trabajo	Lugar donde trabaja el estudiante en caso de.	Cadena de texto.
Etnia	Codificación del estudiante perteneciente o no a una etnia.	Cadena de texto descriptiva del tipo de etnia.
Discapacidad	Código de identificación del tipo de discapacidad.	Cadena de texto.
Promediomediosup	Promedio académico del estudiante en la educación media superior.	Valor flotante entre 0 y 10.
CENEVAL	Puntaje del estudiante en el examen de admisión.	Valor flotante de 4 dígitos.
Estatus	Estatus académico del estudiante.	Cadena de texto.

Cuadro 3.1: Atributos que componen la base de datos de estudiantes de la Universidad Autónoma de Chihuahua. En la segunda columna se describe brevemente a los atributos y en la tercer columna se describe el tipo de dato que se usa. Los atributos de Estado, Municipio, Colonia, Calle y NumeroExterior no se utilizaron directamente en el entrenamiento de la red, con ellos se construyó un campo de dirección que se utilizó para realizar la georreferencia de las coordenadas geográficas donde se ubica el estudiante, para poder asociar un grado de marginación (GM ver tabla 3.2). El atributo estatus engloba los diferentes tipos de baja que puede tener un alumno, si es pasante, reingreso o titulado, para el modelo se co-

Atributo	Descripción	Tipo de dato
DescProgramaeducativo	Programa educativo al que pertenece el estudiante.	Cadena de texto.
Promedio semestral*	Valor numérico que representa el promedio del estudiante en el semestre denotado, cualquiera desde 1ro a 8vo.	Valor flotante entre 0 y 10.
Índicedeaprobación semestral	Valor numérico que representa el indice de materias aprobadas contra las tomadas durante el semestre especificado.	Valor flotante entre 0 y 1.
GM	Valor del grado de marginación, obtenidos a través de la base de datos del CONEVAL.	Cadena de texto que puede ir desde "Muy bajo" a "Muy alto".

Cuadro 3.2: Segunda parte de los atributos que componen la base de datos de estudiantes de la Universidad Autónoma de Chihuahua. Los campos de promedio e índice de aprobación
El promedio se calculó para cada estudiante en cada semestre cursado tomando los valores de la calificación obtenidas en las materias de cada ciclo en cuestión.

cuatro de ellos debido a la gran similitud entre los mismos.

La forma de modelo elegida para este proyecto fue la de *perceptrón multicapa*. La arquitectura implementada consta con una capa de entrada con neuronas por cada descriptor, 3 capas ocultas con 126 neuronas y una función de activación *relu* y una capa de salida de 2 neuronas, representando estas si el dato pertenece a la clase *reingreso* o a la clase *baja*. La clase *reingreso* incluye a los alumnos de reingreso, pero también a los que aparecen como pasante y titulado, ya que ellos ya concluyeron de manera exitosa sus estudios y no están en riesgo de causar baja. Cualquier tipo de baja, ya sea temporal o por cambio de carrera se considera para la clase de *baja*. Los modelos se sometieron a 500 épocas de entrenamiento.

El primer modelo, llamado *Modelo 0* contó con 3 capas ocultas completamente conectadas en conjunto a las capas de entrada y de salida, las capas ocultas contaron con 126 neuronas activadas por la función ReLU (ecuación (2.21)). La función ReLU tiene la ventaja de que no activa todas las neuronas al mismo tiempo, anulando cualquier salida menor a 0. Este comportamiento conlleva a una buena eficiencia computacional en comparación con otras funciones como la sigmoide, que tiene problemas cuando las entradas son valores mayores a 1 o menores a 0. En el modelo se contó con 8 variables de entrada y 500 épocas de entrenamiento con una capa de salida de dos neuronas para clasificar entre las categorías de baja y reingreso, siendo entrenada utilizando el optimizador por entropía(sección 2.4.2). El porque de las 500 épocas de entrenamiento fue elegir un número lo suficientemente grande de actualizaciones de los pesos para que los resultados fueran aceptables, pero no tanto como para que fuera muy tardado en tiempo de cómputo, las gráficas de entrenamiento mostraron que en ese número de épocas ya se está bastante cerca al punto en que lo pesos ya no se actualizan. Este primer modelo se desarolló con el fin de probar si las características del estudiante antes de formar parte de la institución eran influyentes en la conclusión de los estudios del estudiante.

Para el segundo modelo,el *Modelo 01*, se mantuvo la misma arquitectura de 3 capas ocultas completamente conectadas, donde el número de neuronas fue de 126 por capa, y la función de activación utilizada es la ReLU. En este caso la estructura de la red sufrió cambios mínimos salvo la adición de dos neuronas a la capa de entrada, agregadas para que la red tuviera de entrada dos variables más, teniendo 10 en total. Se mantuvieron las 500 épocas de entrenamiento con resultados similares. El objetivo de este modelo fue evaluar el impacto del primer semestre del alumno en su continuidad en el programa educativo o la baja del

mismo.

El *Modelo 02* recibió ajustes en la capa de entrada, incrementando a 13 entradas que representan 13 características de cada dato de estudio. Se utilizaron el optimizador por entropía, 3 capas ocultas completamente conectadas de 126 neuronas cada una activadas por la función ReLU, dos neuronas en la capa de salida y 500 épocas de entrenamiento. Este modelo se realizó para predecir el estatus académico utilizando el conocimiento del desempeño de los estudiantes después de su segundo semestre de clases en la institución. A partir de este modelo, hasta llegar al *Modelo 08* se agregaron 2 variables en cada modelo, variables que describen el desempeño del alumno en cada semestre hasta llegar al octavo semestre a fin de continuar evaluando la influencia de las calificaciones obtenidas en cada semestre. Se decidió no tomar en cuenta el noveno semestre debido a que no todos los programas educativos llegan 9 semestres.

CAPÍTULO: 4

Resultados

Durante el desarrollo de los modelos la elección de los descriptores y la evaluación del desempeño de las redes se realizó con el único objetivo de mejorar los valores de precisión, para cada modelo tomando en consideración puntos como la disminución de datos y la evolución de los alumnos en sus programas educativos. Por esta razón fue necesario someter la base de datos original a distintos cambios a modo de no introducir ruido en los modelos de semestres superiores, tomando en consideración los datos que no existen en los distintos semestres, es decir, los estudiantes que se dieron de baja. Este proceso de filtrar los alumnos que terminan en estado de baja permitió observar como la cantidad de alumnos por semestre se reduce de una manera casi exponencial, lo que llevo la creación de la siguiente gráfica (Figura 4.1).

Esta información nos abre la posibilidad de estudiar la cantidad de bajas de los estudiantes a fin de ubicar puntos específicos en los programas educativos donde sea más probable la deserción de los estudiantes. Retomando los modelos de red neuronal, una vez que se realizó el proceso de entrenamiento sobre las redes, fue necesario evaluarlo, y para esto fue necesario mostrar la evolución del entrenamiento. Luego de realizar el entrenamiento de los distintos modelos podemos ver a través de las gráficas que entre más avanza el estudiante en su programa educativo es más sencillo el predecir su estatus próximo, esto si observamos como con cada modelo las curvas se acercan más al 100 % de precisión. La forma de las curvas de entrenamiento y pérdida permite ver que el entrenamiento si llegó

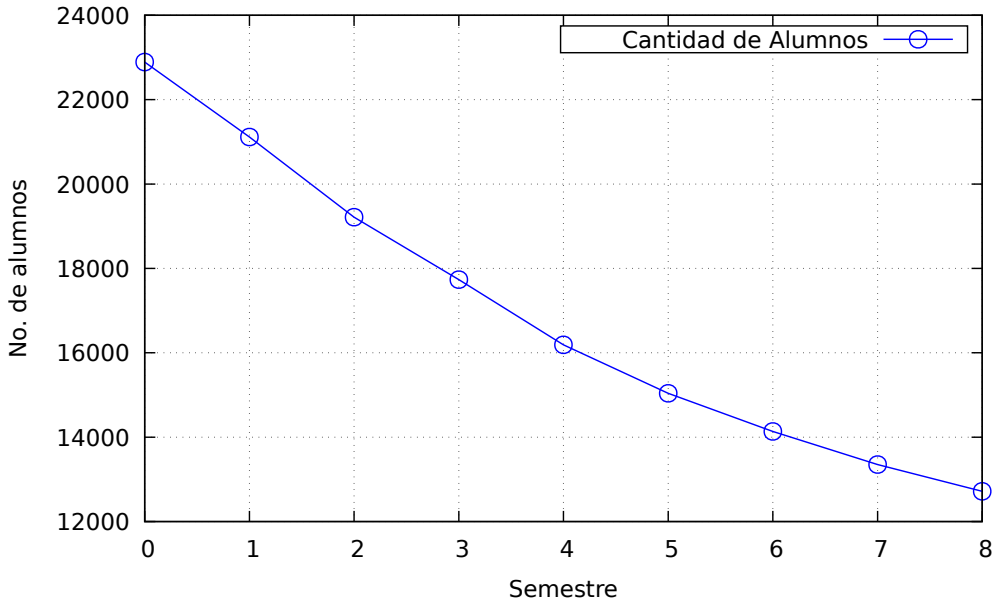


Figura 4.1: Gráfica de número de alumnos contemplados en la red por cada semestre de las 8 generaciones contempladas.

a un punto relativamente bueno en el que el crecimiento muestra una tendencia hacia arriba. A pesar de la pérdida de datos en cantidad para los modelos de los semestres más altos, el proceso de entrenamiento se mantuvo estable y con buenos resultados(Figuras 4.2 - 4.19).

Otra de las maneras existentes de medir los resultados obtenidos del entrenamiento de las redes neuronales es utilizando las matrices de confusión, en las cuales se puede apreciar como la red clasifica los datos, dividiéndolos entre las distintas formas de clasificar, teniendo las clases clasificadas correctamente y las que fueron clasificadas incorrectamente. En este proyecto las matrices de confusión cuentan con 4 divisiones, donde tenemos los elementos clasificados en "*reingreso*" y "*baja*" de manera exitosa y aquellos que fueron clasificados en la categoría equivocada. Al haber contado en cada uno de los modelos con una presición superior al 70 %, las matrices de confusión cuentan con una mayor concentración de elementos en su diagonal, ya que ahí es donde se encuentran los elementos clasificados correctamente.

4.0.1. Modelo 00

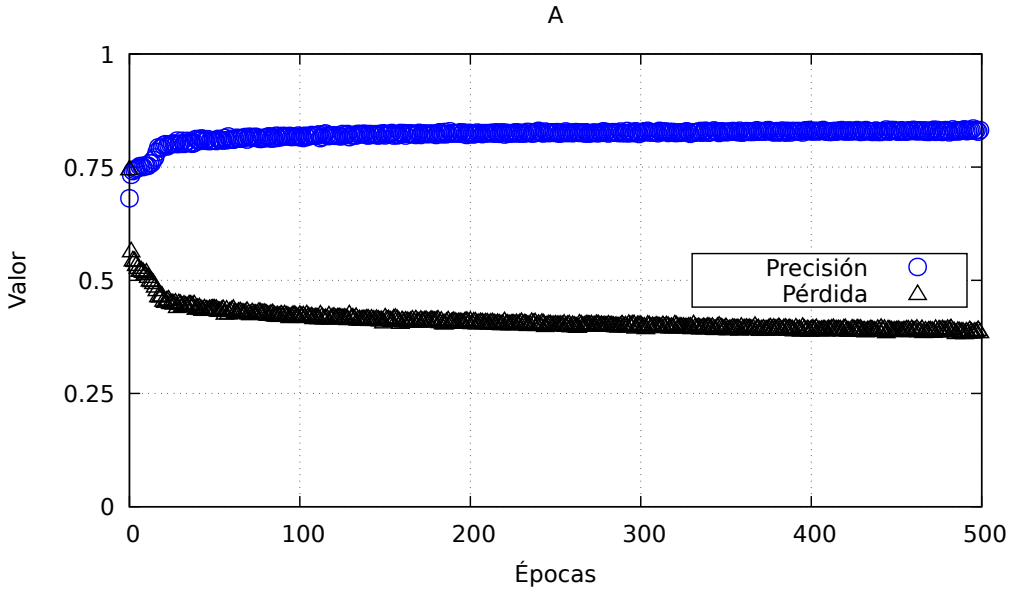


Figura 4.2: Resultados de entrenamiento y pérdida del modelo 00. Representado por círculos podemos ver el incremento del valor de precisión conforme avanza el proceso de entrenamiento hasta llegar a un punto en el que el valor recibe cambios insignificantes por paso de tiempo. En triángulos tenemos la representación gráfica de la pérdida en el entrenamiento, las cuales muestran curvas en descenso debido a la minimización pertinente en el entrenamiento de las redes. El valor promedio de la precisión después del entrenamiento fue del 74 %.

En el modelo 00 contamos con la primer iteración de la arquitectura de red trabajando con los datos reales. En este modelo se alimento a la red con todos los datos de estudiantes de la universidad antes de concluir su primer semestre, es decir, sin promedios semestrales o indices de aprobación. El objetivo de este modelo en particular es el evaluar a un estudiante mediante datos socio económicos y su desempeño en la educación media superior. Siendo el ingreso uno de los puntos más importantes para los estudiantes, el poder identificar a los estudiantes en situación de riesgo antes de la primer evaluación puede ser un punto crítico a la hora de apoyarlo en prevenir una posible baja. Durante el proceso de entrenamiento, representada por círculos de color azul tenemos la precisión de la red, la cual se incrementó gradualmente hasta llegar a un

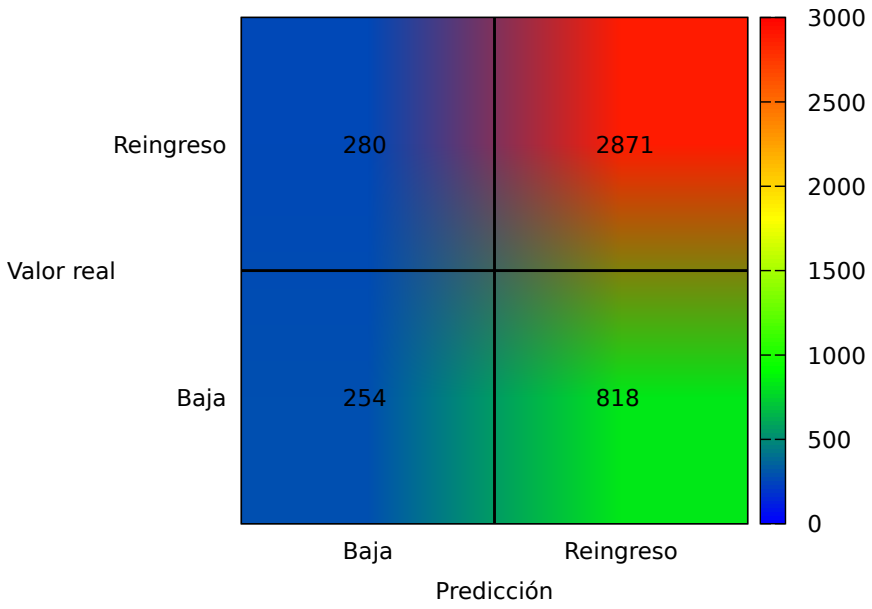


Figura 4.3: Matriz de confusión del modelo 00, en la diagonal descendente de izquierda a derecha podemos ver los elementos clasificados correctamente mientras que la otra diagonal tenemos, en la casilla superior derecha tenemos bajas clasificadas incorrectamente como reingreso y en la casilla inferior izquierda tenemos reingresos clasificados incorrectamente como bajas.

punto límite alrededor del 75 % de precisión de la red, mientras que en triángulos de color negro se encuentra representada la pérdida en el entrenamiento, la cual si se redujo de manera considerable a lo largo de todo el proceso (Figura 4.2). Para evaluar de mejor manera el desempeño del modelo se recurrió al uso de la matriz de confusión, la cual muestra en un diagrama sencillo de ver la cantidad de elementos de prueba que se clasificaron de manera correcta o incorrecta, con el agregado de etiquetar las categorías tanto bien como mal clasificadas. En el caso particular de los modelos desarrollados en este trabajo tenemos dos categorías que son *reingreso* y *baja*, por lo que la matriz de confusión cuenta con 4 casillas: reingreso(esquina inferior derecha), baja(esquina superior izquierda), falso reingreso(esquina superior derecha) y falsa baja(esquina inferioir izquierda) (Figura 4.3). En nuestros resultados es posible ver como el total de elementos clasificados incorrectamente cuenta con una especie de *balance*, no estando la mayor parte de los datos cargados hacia una sola de las categorías. Este comportamiento le da a la red neuronal una capacidad de robustez a la hora de evaluar datos nuevos, caso que está sujeto a realizar esas pruebas con datos nuevos. El valor promedio del resultado de precisión al final del proceso de entrenamiento se encuentra en 0.7461 (74 %), lo cual se traduce a una correcta predicción de 7 de cada 10 estudiantes, que a su vez representa una aplicación de los programas de tutorías más informada en al menos esos mismos 7 estudiantes.

4.0.2. Modelo 01

En el modelo 01 contamos con la introducción del promedio semestral y el indice de aprobación correspondientes al primer ciclo escolar concluido por los alumnos. En lo que respecta a objetivos, el modelo 01 mantiene el mismo objetivo, solo que este modelo cuenta con dos variables más para cumplirlo. En las gráficas de entrenamiento y perdida podemos ver curvas con una tendencia un poco más hacia arriba que el modelo 00, alcanzando un 78 %. El entrenamiento mantiene la representación en triángulos de la pérdida y de círculos azules para el entrenamiento (Figura 4.4). En este caso la matriz de confusión muestra un comportamiento similar al modelo 00, sin embargo en este caso hay una cantidad considerable de datos en la categoría de baja en contraste a la categoría reingreso, lo cual des balancea los resultados de la matriz de confusión, cargando una mayor cantidad de datos hacia la categoría de *baja* (Figura 4.5). Dada la similitud entre los modelos desarrollados, para modelos posteriores se simplificara un poco la descripción para dar énfasis a los resultados númericos. El modelo 01 cuenta con

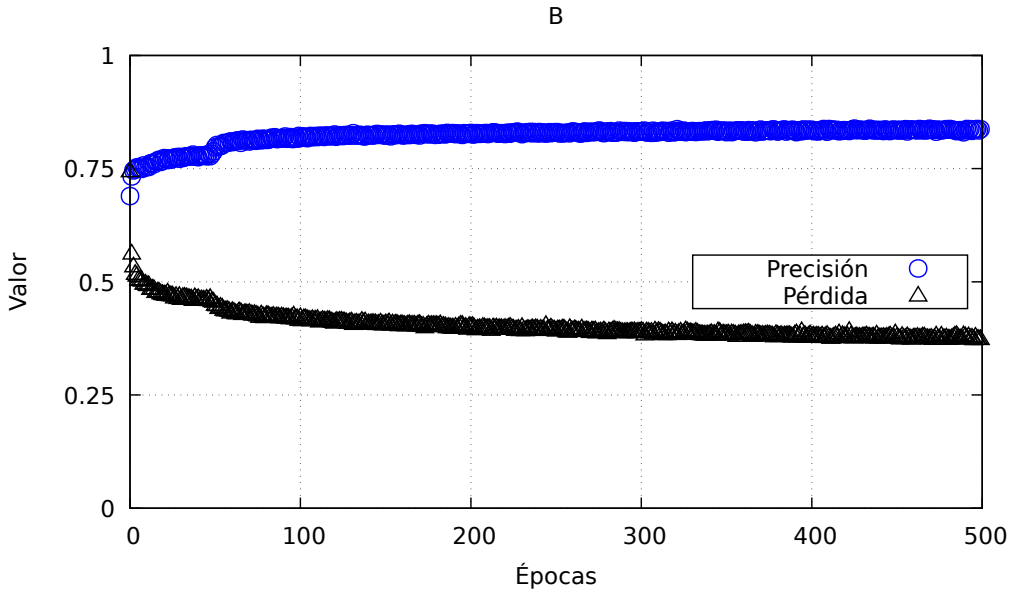


Figura 4.4: Resultados de entrenamiento y pérdida del modelo 01. Representado por círculos podemos ver el incremento del valor de precisión conforme avanza el proceso de entrenamiento hasta llegar a un punto en el que el valor recibe cambios insignificantes por paso de tiempo. En triángulos tenemos la representación gráfica de la pérdida en el entrenamiento, las cuales muestran curvas en descenso debido a la minimización pertinente en el entrenamiento de las redes.Los resultados finales del modelo se encuentran alrededor del 78 %.

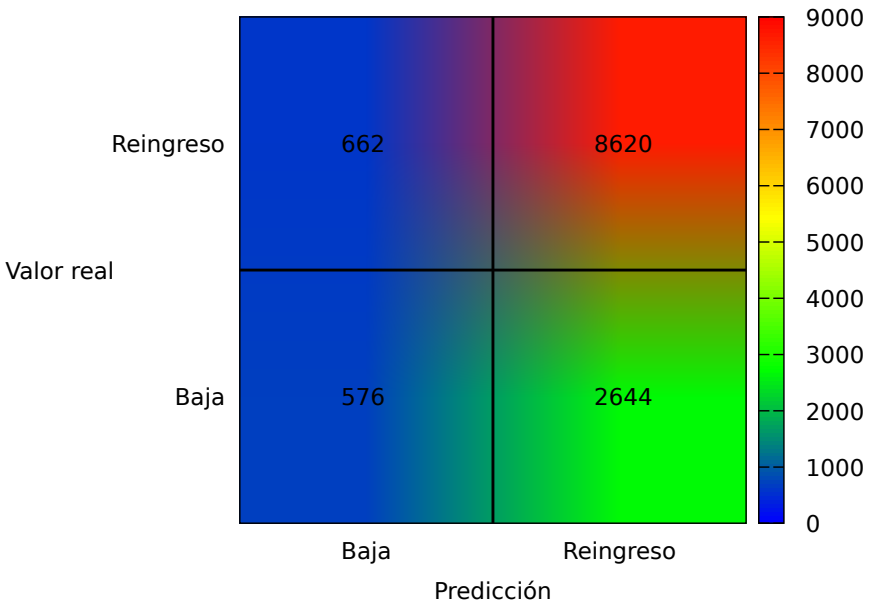


Figura 4.5: Matriz de confusión del modelo 01, en la diagonal descendente de izquierda a derecha podemos ver los elementos clasificados correctamente mientras que la otra diagonal tenemos, en la casilla superior derecha tenemos bajas clasificadas incorrectamente como reingreso y en la casilla inferior izquierda tenemos reingresos clasificados incorrectamente como bajas. En este caso se cuenta con una mayor cantidad de datos en categoría de baja, por lo que la cantidad de elementos por categoría de la matriz se des balancea.

un valor promedio de resultado final de 0.7817, alrededor de un 78 %, si bien es solo un poco mayor al resultado del modelo 00, este resultado sigue siendo muy bueno para la búsqueda de una mejor implementación de los programas de tutoría.

4.0.3. Modelo 02

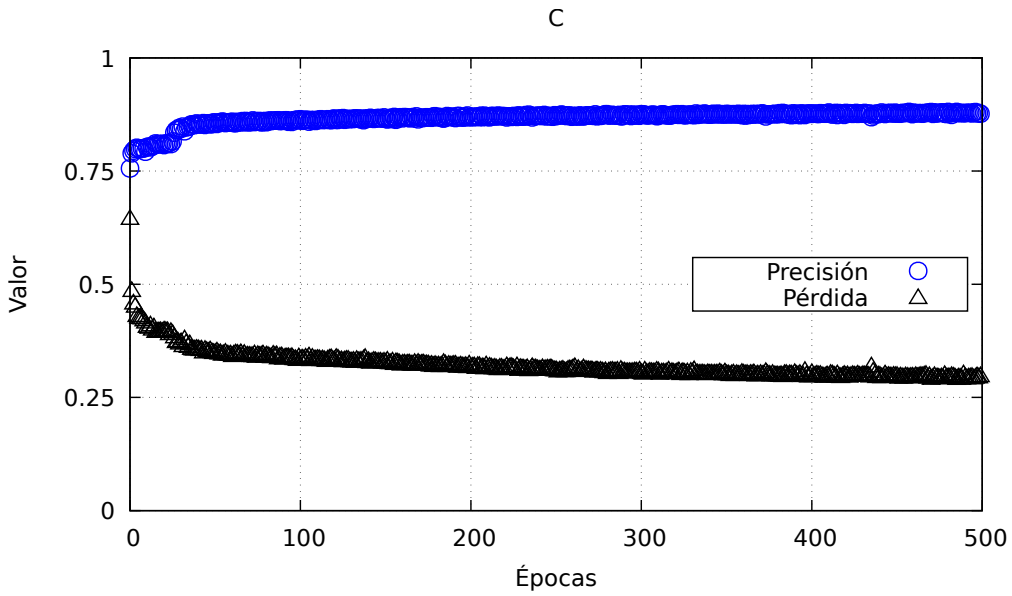


Figura 4.6: Resultados de entrenamiento y pérdida del modelo 02. Representado por círculos podemos ver el incremento del valor de precisión conforme avanza el proceso de entrenamiento hasta llegar a un punto en el que el valor recibe cambios insignificantes por paso de tiempo. En triángulos tenemos la representación gráfica de la pérdida en el entrenamiento, las cuales muestran curvas en descenso debido a la minimización pertinente en el entrenamiento de las redes.Los resultados van en crecimiento hasta arrojar finalmente un valor alrededor del 82 %.

A modo de no ser tan redundante, a partir de este caso la descripción de los modelos se hará de manera más resumida para dar paso principalmente a los resultados. La gráfica de entrenamiento cuenta con una curva más plana, la cual indica que el entrenamiento alcanza muy rápido su punto de estancamiento tanto

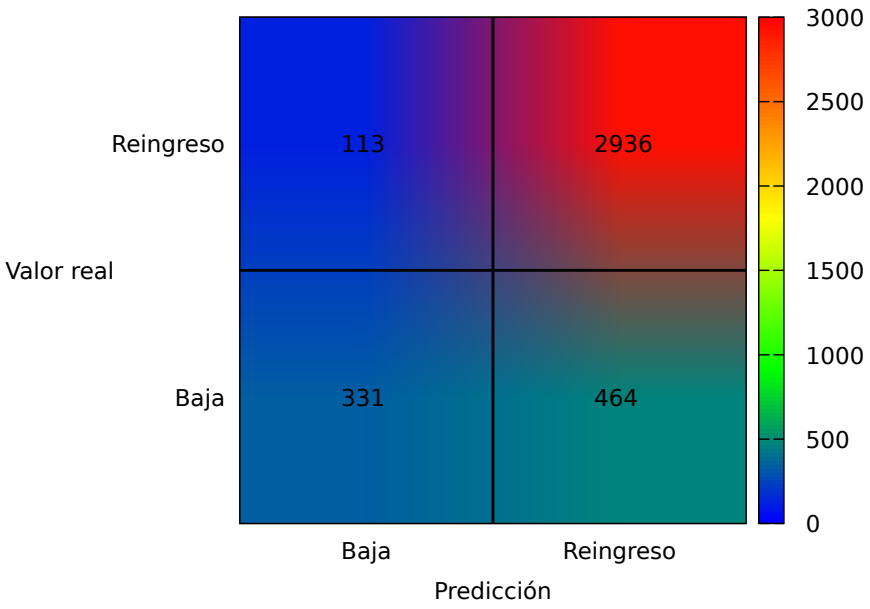


Figura 4.7: Matriz de confusión del modelo 02, en la diagonal descendente de izquierda a derecha podemos ver los elementos clasificados correctamente mientras que la otra diagonal tenemos, en la casilla superior derecha tenemos bajas clasificadas incorrectamente como reingreso y en la casilla inferior izquierda tenemos reingresos clasificados incorrectamente como bajas. En este caso se cuenta con una mayor cantidad de datos en categoría de reingresos.

como para el entrenamiento como la pérdida, ese comportamiento puede ser un indicador de que la cantidad de épocas es muy elevada para este modelo, sin embargo la curva aunque se ve plana si lleva una pequeña pendiente hacia arriba, que mantiene a consideración si es relevante el crecimiento(Figura 4.6). La matriz de confusión del modelo 02 muestra muchos de los elementos en la categoría *reingreso* contra la categoría de *baja*, esto podría llevar a una generalización del modelo hacia la categoría de baja, sin embargo la matriz de confusión muestra que a pesar de este caso, el modelo distingue relativamente bien entre las categorías y no muestra tendencia a clasificar todo hacia una sola categoría(Figura 4.7). El modelo 02 es una versión del modelo de red desarrollado que cuenta con otro par extra de datos de entrada, siendo estos datos los promedios e indices de aprobación correspondientes al segundo semestre cursado por los estudiantes que componen la red. La adición de esas dos variables a las ya existentes en el modelo 01 lleva a un mejor desempeño de la red, alcanzando un valor de precisión de 0.8249, lo cual es muy bueno, el pasar la barrera del 0.8 vuelve los resultados bastante competitivos contra la literatura actual. En la práctica estos resultados de arriba del 80 % de precisión vuelve la confiabilidad de la predicción aún más útil para los sistemas de tutorías, teniendo como punto de discusión el que este tipo de resultados sean dependientes de que el alumno curse al menos dos semestres en la universidad.

4.0.4. Modelo 03

El entrenamiento del modelo 03 lleva una curva de crecimiento más plana que la de los modelos anteriores, esto conlleva un crecimiento en el valor de precisión lento, sin embargo, debido a la homogeneidad entre todos los modelos en cuanto al número de épocas el valor de precisión que se obtuvo al final es bastante bueno, siendo este un 0.8708 (87.08 %)(Figura 4.8). En la matriz de confusión del modelo 03 tenemos una distribución más homogénea que en el modelo 02(Figura 4.9). Al estar más balanceada la cantidad de datos de la categoría *baja* con la categoría *reingreso*, el modelo distribuye bastante bien los datos, no cayendo en problemas de sobre entrenamiento ni mandando la mayoría de los datos a una sola categoría que por su magnitud abarque mayor porcentaje. Retomando la definición del modelo, el modelo 03 cuenta con la adición de promedios e indice de aprobación correspondientes al tercer semestre de los estudiantes, si vemos la gráfica de número de estudiantes (Figura 4.1) podemos observar que la cantidad de estudiantes es casi al mitad de los que ingresaron inicialmente.

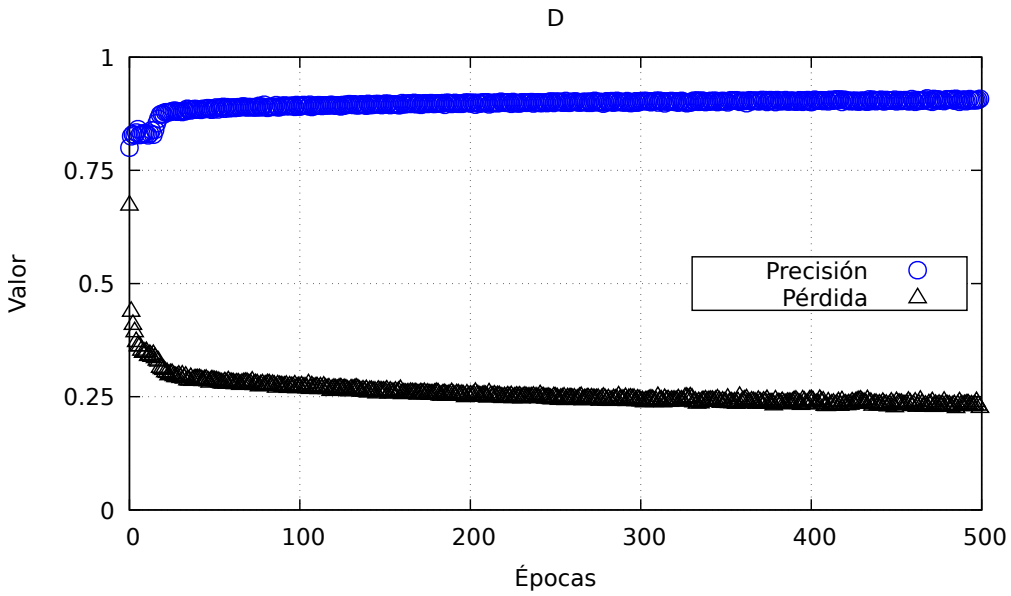


Figura 4.8: Resultados de entrenamiento y pérdida del modelo 03. Representado por círculos podemos ver el incremento del valor de precisión conforme avanza el proceso de entrenamiento hasta llegar a un punto en el que el valor recibe cambios insignificantes por paso de tiempo. En triángulos tenemos la representación gráfica de la pérdida en el entrenamiento, las cuales muestran curvas en descenso debido a la minimización pertinente en el entrenamiento de las redes.Los resultados van en crecimiento hasta arrojar finalmente un valor alrededor del 89 %.

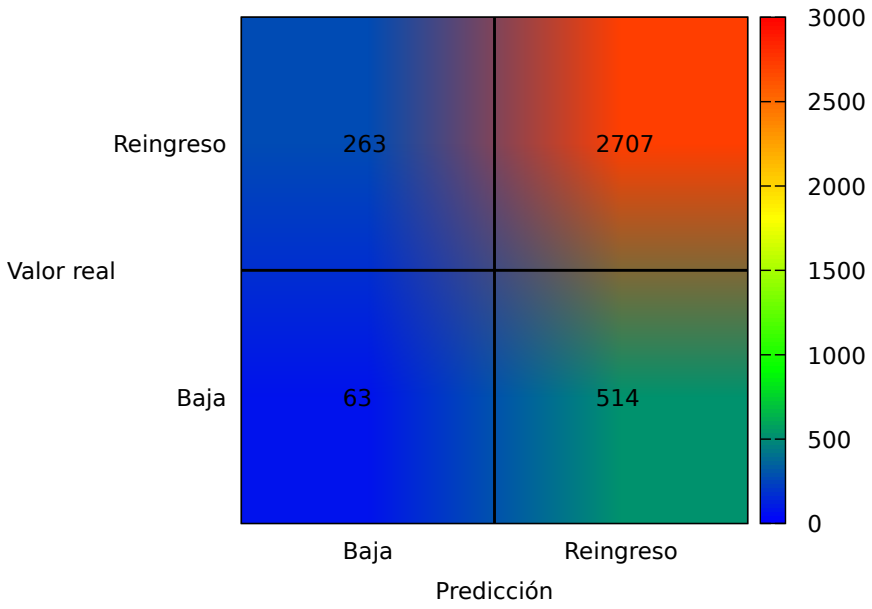


Figura 4.9: Matriz de confusión del modelo 03, en la diagonal descendente de izquierda a derecha podemos ver los elementos clasificados correctamente mientras que la otra diagonal tenemos, en la casilla superior derecha tenemos bajas clasificadas como reingreso y en la casilla inferior izquierda tenemos reingresos clasificados como bajas. En este caso se cuenta con una mayor cantidad de datos en categoría de baja, la diferencia entre las dos clases sigue siendo muy marcada.

A pesar de ello la cantidad de bajas no es tan marcada como en el modelo 02, poniendo en evidencia que este semestre no es un punto de alta cantidad de bajas en los estudiantes. En este semestre se puede ver que los estudiantes están más cargados en la categoría de reingreso por un margen considerable.

4.0.5. Modelo 04

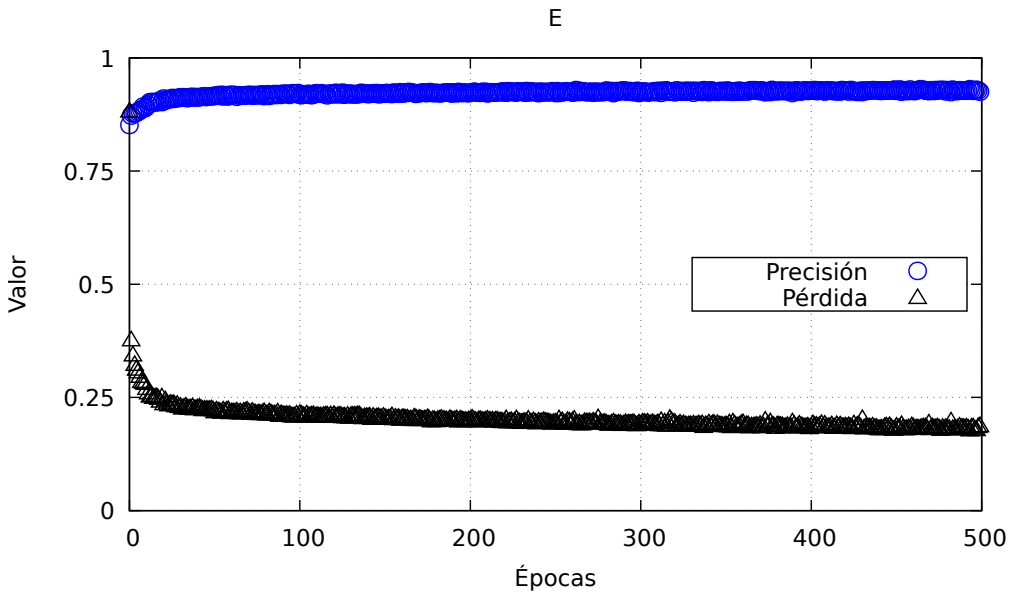


Figura 4.10: Resultados de entrenamiento y pérdida del modelo 04. Representado por círculos podemos ver el incremento del valor de precisión conforme avanza el proceso de entrenamiento hasta llegar a un punto en el que el valor recibe cambios insignificantes por paso de tiempo. En triángulos tenemos la representación gráfica de la pérdida en el entrenamiento, las cuales muestran curvas en descenso debido a la minimización pertinente en el entrenamiento de las redes.Los resultados van en crecimiento hasta arrojar finalmente un valor alrededor del 89 %.

En el modelo 04 tenemos una curva de entrenamiento bastante similar a la que mostró el modelo 03(Figura 4.8) con la diferencia de un aumento en el valor final contra el mismo modelos 03. El modelo 04 tiene una curva muy plana que llega casi al 90 % de precisión(Figura 4.10), esto gracias a que se agregaron, como

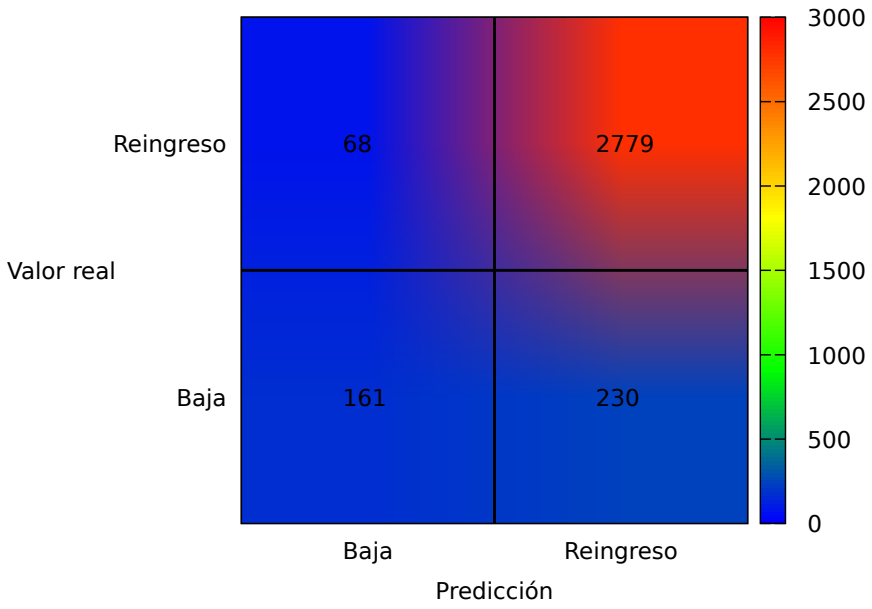


Figura 4.11: Matriz de confusión del modelo 04, en la diagonal descendente de izquierda a derecha podemos ver los elementos clasificados correctamente mientras que la otra diagonal tenemos, en la casilla superior derecha tenemos bajas clasificadas como reingreso y en la casilla inferior izquierda tenemos reingresos clasificados como bajas. En este caso se mantiene una mayor cantidad de datos en categoría de reingreso.

en los casos del modelo 02 y 03, promedio e índice de aprobación correspondiente al cuarto semestre. Entre los estudiantes se suele decir que el cuarto semestre es el parte aguas entre aquellos que concluyen la carrera contra los que no, por lo que este modelo es una oportunidad de evaluar semejante afirmación. En el caso de cantidad de bajas tenemos que a este punto la cantidad de alumnos ya es menor de la mitad de los que se consideraron al inicio (Figura 4.1), sin embargo la pendiente de la curva de alumnos no muestra un declive significativamente mayor entre cuarto y quinto. En el caso de la matriz de confusión, esta mantiene un comportamiento similar a los modelos 02 y 03 (Figuras 4.7 y 4.9), teniendo la cantidad más fuerte en los reingresos correctamente clasificados, si observamos la gráfica de estadística de promedios por semestre (Figura 4.21) es fácil ver como los promedios se alejan del 5 en semestres superiores, lo cual implica que los estudiantes pasen, es decir, sean estudiantes de la categoría *reingreso*.

4.0.6. Modelo 05

En el modelo 05, el valor de precisión se disparó en comparación con la tendencia en modelos anteriores, (Figura 4.20) creciendo hasta un 93.71 %. En cuestión de las curvas de entrenamiento y pérdida, ambas tienen un salto al comienzo del proceso de entrenamiento, luego del cual ambas curvas se aplanan, pero siguen subiendo durante todo el proceso, lento pero seguro (Figura 4.12). Por otro lado, la matriz de confusión muestra de manera muy fuerte el des balance que se ha acentuado en los datos(4.13), ya que mayoritariamente hay muchos reingresos, sin embargo, el modelo no cae en la acción de clasificar todos los datos en una sola categoría, eso permite que en dado caso de meter datos ajenos al proceso de entrenamiento y prueba se puedan obtener resultados competentes.

4.0.7. Modelo 06

El modelo 06 continua con el agregado del promedio e índice de aprobación correspondientes al sexto semestre de estudios del estudiante. El modelo 06 crece muy poco en comparación al modelo 05 en diferencia de como creció el modelo 05 respecto al modelo 04. En lo que respecta a la gráfica de entrenamiento, la curva tiene una forma cada vez más plana, aunque pareciera que sube mucho el valor, el punto máximo es de 0.9504, mientras que la pérdida ya está muy debajo del 0.25 que se ha usado como marco de referencia a lo largo de la

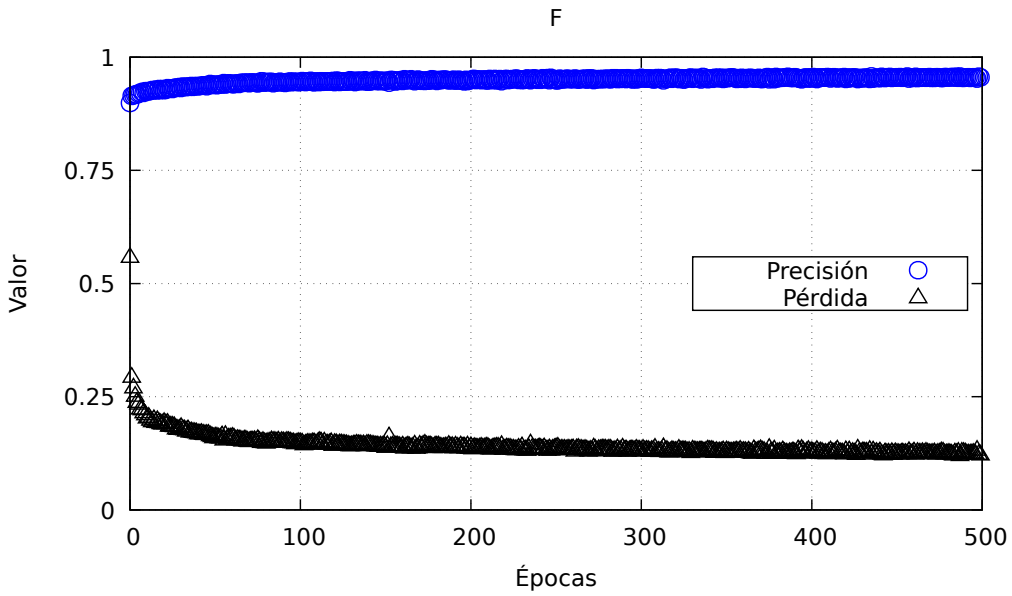


Figura 4.12: Resultados de entrenamiento y pérdida del modelo 05. Representado por círculos podemos ver el incremento del valor de precisión conforme avanza el proceso de entrenamiento hasta llegar a un punto en el que el valor recibe cambios insignificantes por paso de tiempo. En triángulos tenemos la representación gráfica de la pérdida en el entrenamiento, las cuales muestran curvas en descenso debido a la minimización pertinente en el entrenamiento de las redes.Los resultados van en crecimiento hasta arrojar finalmente un valor alrededor del 93 %, este modelo tiene el mayor salto en valor de precisión.

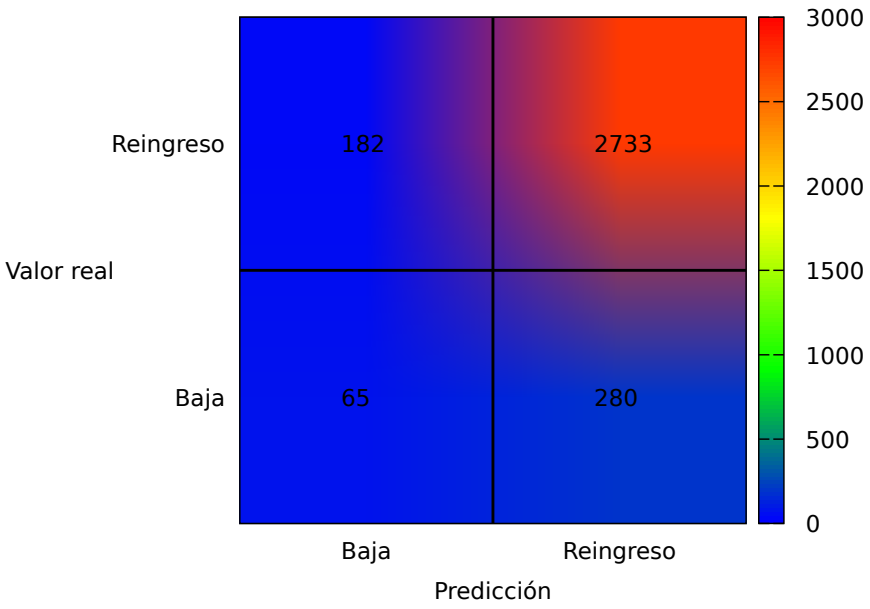
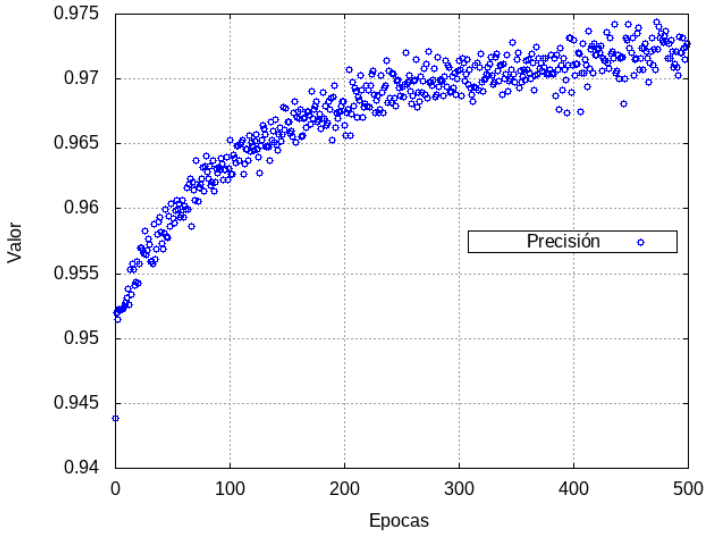
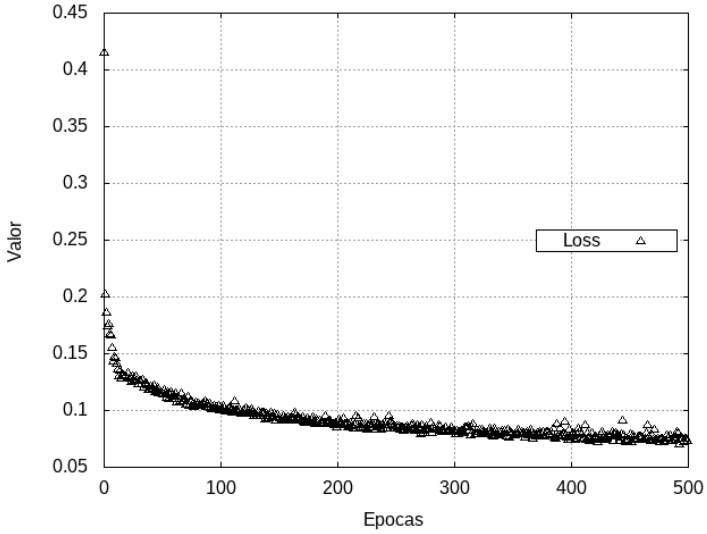


Figura 4.13: Matriz de confusión del modelo 05, en la diagonal descendente de izquierda a derecha podemos ver los elementos clasificados correctamente mientras que la otra diagonal tenemos, en la casilla superior derecha tenemos bajas clasificadas como reingreso y en la casilla inferior izquierda tenemos reingresos clasificados como bajas. La cantidad de datos en categoría de reingreso ya es considerablemente mayor en este modelo.



(a) Gráfica de entrenamiento del modelo 06.



(b) Gráfica de pérdida del modelo 06.

Figura 4.14: Resultados de entrenamiento y pérdida del modelo 06. Representado por círculos podemos ver el incremento del valor de precisión conforme avanza el proceso de entrenamiento hasta llegar a un punto en el que el valor recibe cambios insignificantes por paso de tiempo. En triángulos tenemos la representación gráfica de la pérdida en el entrenamiento, las cuales muestran curvas en descenso debido a la minimización pertinente en el entrenamiento de las redes. Los resultados van en crecimiento hasta arrojar finalmente un valor alrededor del 95 %, este modelo muestra un crecimiento muy lento.

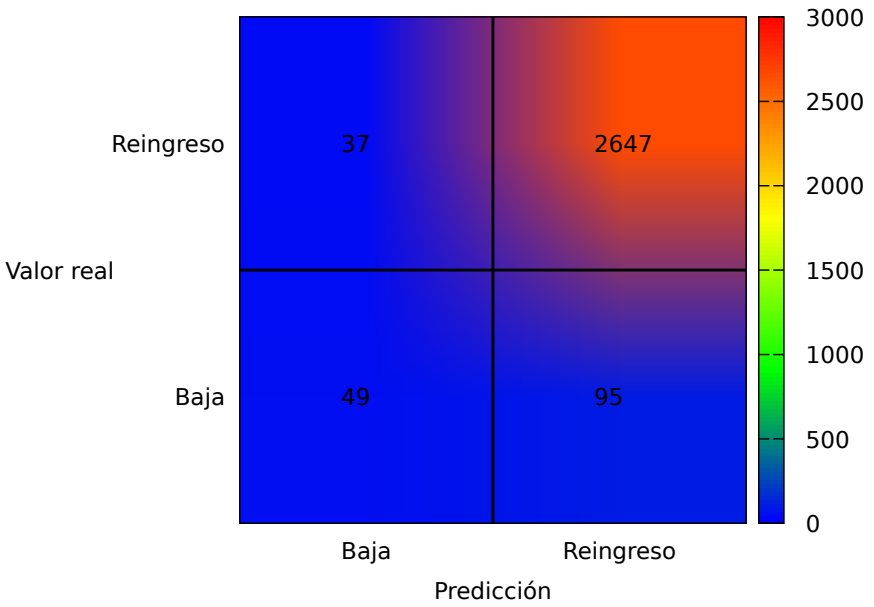


Figura 4.15: Matriz de confusión del modelo 06, en la diagonal descendente de izquierda a derecha podemos ver los elementos clasificados correctamente mientras que la otra diagonal tenemos, en la casilla superior derecha tenemos bajas clasificadas como reingreso y en la casilla inferior izquierda tenemos reingresos clasificados como bajas. La cantidad de datos en categoría de reingreso ya es considerablemente mayor en este modelo, siendo menos de 200 datos en la categoría baja.

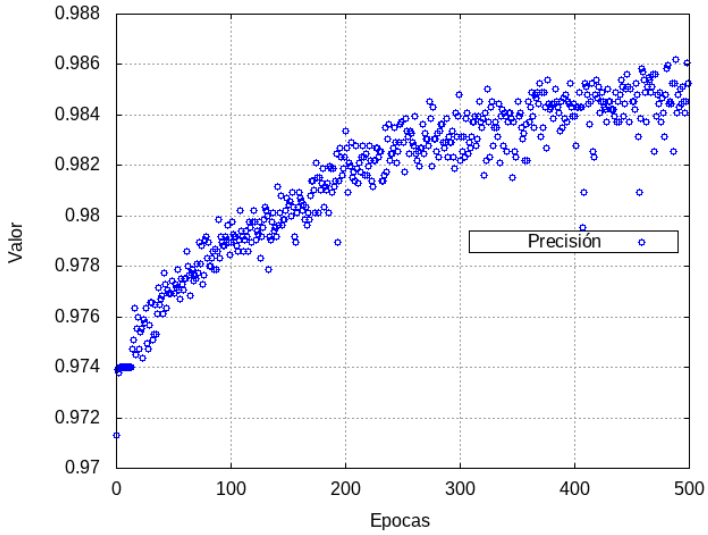
evaluación de los modelos (Figura 4.14). Por otro lado la matriz de confusión se puede ver lo mucho que se ha reducido la categoría de *baja* contra la de *reingreso*, teniendo la clasificación por arriba del 90 % en parte por la masividad de una de las categorías, resaltando que no se esta clasificando todo sobre esa categoría, lo que abre la puerta a pruebas con datos nuevos sin temor al sobre entrenamiento(Figura 4.15). A pesar de que la precisión del modelo es muy alta, el hecho de que esté condicionado a que los estudiantes lleguen al sexto semestre segmenta mucho la utilidad que este modelo agrega a la problemática de predecir a los estudiantes.

4.0.8. Modelo 07

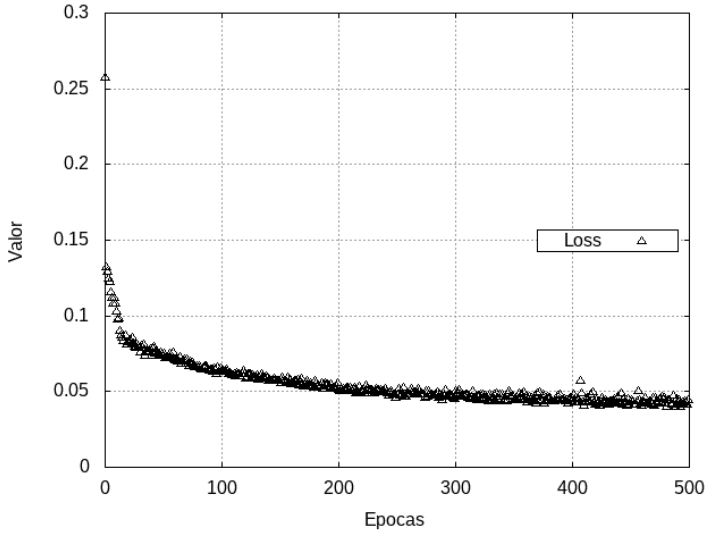
El modelo 07 cuenta con el uso de variables correspondientes a los promedios e índices de aprobación correspondientes a los semestres desde primero hasta séptimo, su precisión se encuentra arriba del 96 %, con una gráfica de entrenamiento y perdida bastante plana en ambos casos, teniendo el entrenamiento estabilizándose alrededor del 0.96, y la pérdida bajando de golpe y estabilizándose alrededor del 0.1(Figura 4.16). Por otro lado, la matriz de confusión de la prueba de este modelo muestra que de los datos(4.17), una increíble mayoría de los datos de la categoría *reingreso* se encuentra clasificado correctamente, sin embargo la categoría *baja* muestra menos éxito en su clasificación, lo cual complica la funcionalidad principal del proyecto, la cual es predecir el estatus siguiente de los estudiantes a modo de que sea posible dirigir los programas de tutoría a los estudiantes que los puedan necesitar.

4.0.9. Modelo 08

El caso del modelo 08 es bastante interesante, los resultados arrojan un porcentaje de precisión bastante alto, siendo 0.9827, sin embargo tenemos que la categoría de *reingreso* se encuentra muy por encima de la categoría de *baja*, siendo aproximadamente 52 datos de *reingreso* por cada dato de *baja*, esto se puede ver en la matriz de confusión(Figura 4.19). La curva de entrenamiento es prácticamente plana al ver su evolución a lo largo de las 500 épocas de entrenamiento, mientrsa la curva de pérdida es la más cercana a 0 en todos los modelos evaluados hasta ahora(Figura 4.18). Evaluando estos datos con los resultados, lo óptimo sería realizar pruebas con el modelo ya entrenado en una base de datos



(a) Gráfica de entrenamiento del modelo 07.



(b) Gráfica de pérdida del modelo 07.

Figura 4.16: Resultados de entrenamiento y pérdida del modelo 07. Representado por círculos podemos ver el incremento del valor de precisión conforme avanza el proceso de entrenamiento hasta llegar a un punto en el que el valor recibe cambios insignificantes por paso de tiempo. En triángulos tenemos la representación gráfica de la pérdida en el entrenamiento, las cuales muestran curvas en descenso debido a la minimización pertinente en el entrenamiento de las redes. Los resultados van en crecimiento hasta arrojar finalmente un valor alrededor del 96 %, este modelo muestra un crecimiento aún más lento y dificil de visualizar.

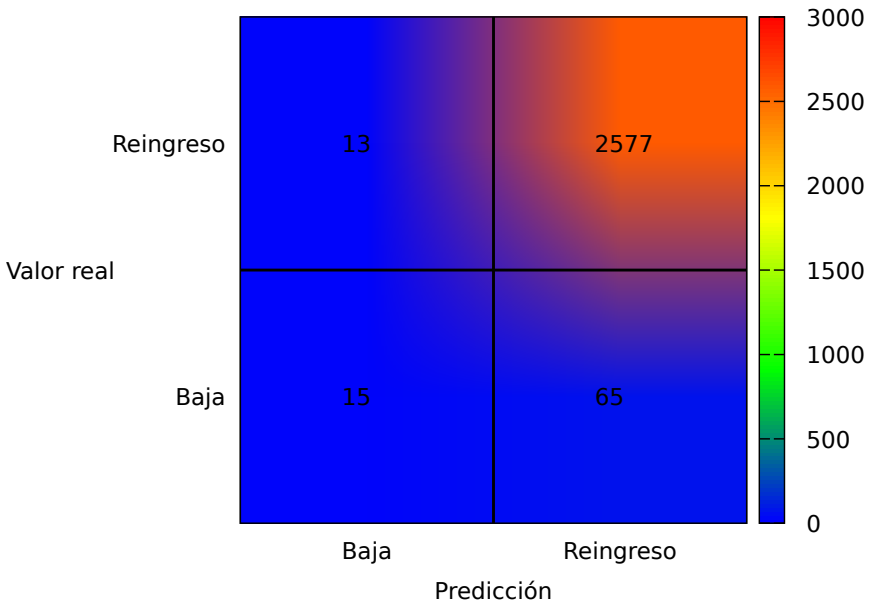
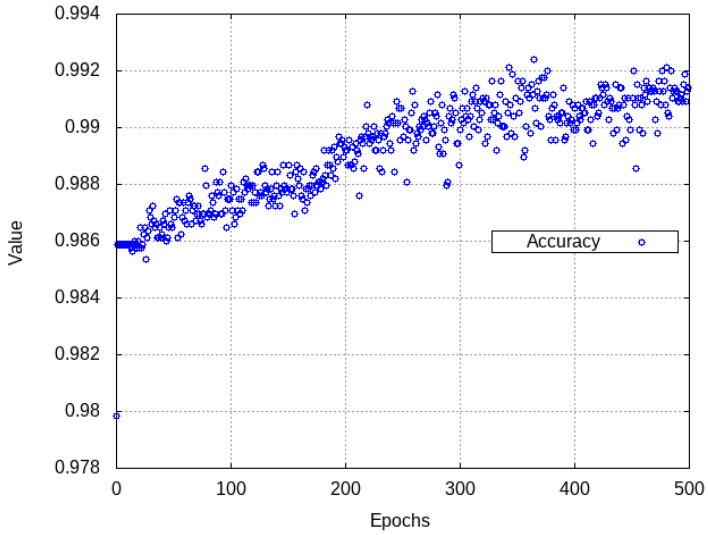
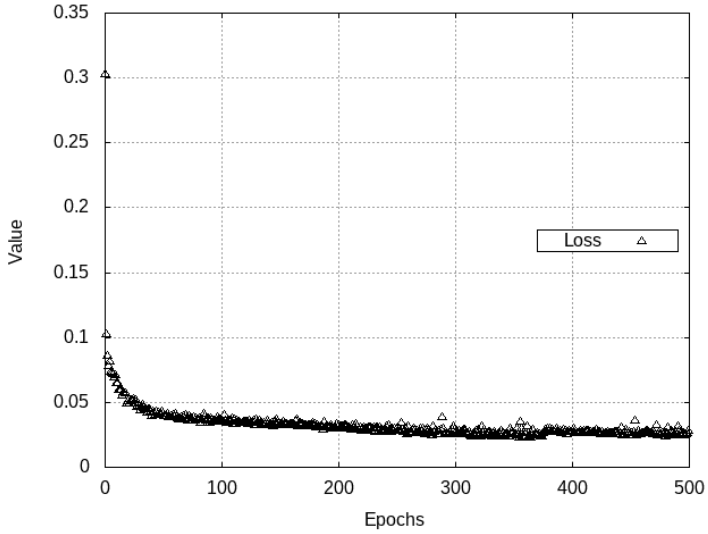


Figura 4.17: Matriz de confusión del modelo 07, en la diagonal descendente de izquierda a derecha podemos ver los elementos clasificados correctamente mientras que la otra diagonal tenemos, en la casilla superior derecha tenemos bajas clasificadas como reingreso y en la casilla inferior izquierda tenemos reingresos clasificados como bajas. La cantidad de datos en categoría de reingreso ya es considerablemente mayor en este modelo, siendo menos de 100 datos en la categoría baja, esto bajando aun más del modelo 06 a este.



(a) Gráfica de entrenamiento del modelo 08.



(b) Gráfica de pérdida del modelo 08.

Figura 4.18: Resultados de entrenamiento y pérdida del modelo 08. Representado por círculos podemos ver el incremento del valor de precisión conforme avanza el proceso de entrenamiento hasta llegar a un punto en el que el valor recibe cambios insignificantes por paso de tiempo. En triángulos tenemos la representación gráfica de la pérdida en el entrenamiento, las cuales muestran curvas en descenso debido a la minimización pertinente en el entrenamiento de las redes. Los resultados van en crecimiento hasta arrojar finalmente un valor alrededor del 98 %, este modelo muestra una gráfica prácticamente plana.

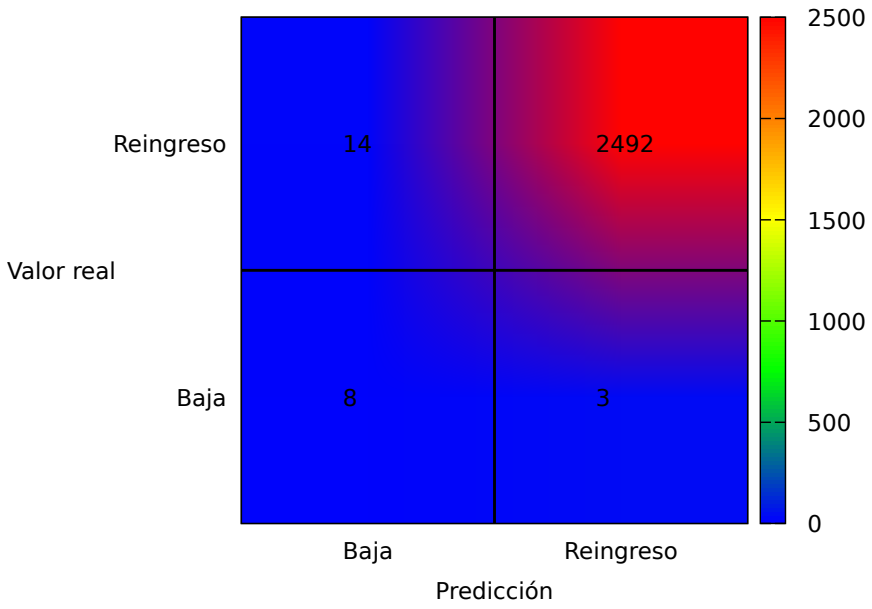


Figura 4.19: Matriz de confusión del modelo 08, en la diagonal descendente de izquierda a derecha podemos ver los elementos clasificados correctamente mientras que la otra diagonal tenemos, en la casilla superior derecha tenemos bajas clasificadas como reingreso y en la casilla inferior izquierda tenemos reingresos clasificados como bajas. La cantidad de datos en categoría de baja ya es muy pequeña, lo que complica la evaluación de los resultados obtenidos.

mejor balanceada para saber si existe o no un caso de sobre entrenamiento que entorpezca la clasificación de los datos de baja.

4.0.10. Resultados adicionales

A modo de complementar los datos de cada modelo, se evaluó el desempeño de cada modelo contra los demás, y a modo de saber que tan bien o mal estaba el funcionamiento, también se evaluó la distribución de los alumnos de la base de datos. La precisión de los modelos se ve en ascenso con cada modelo que avanza, lo que nos lleva a considerar la adición de algún dato o variable extra que permita mejorar el desempeño de los modelos inferiores,el salto principal en los valores de presición se ve del modelo 04 al modelo 05.

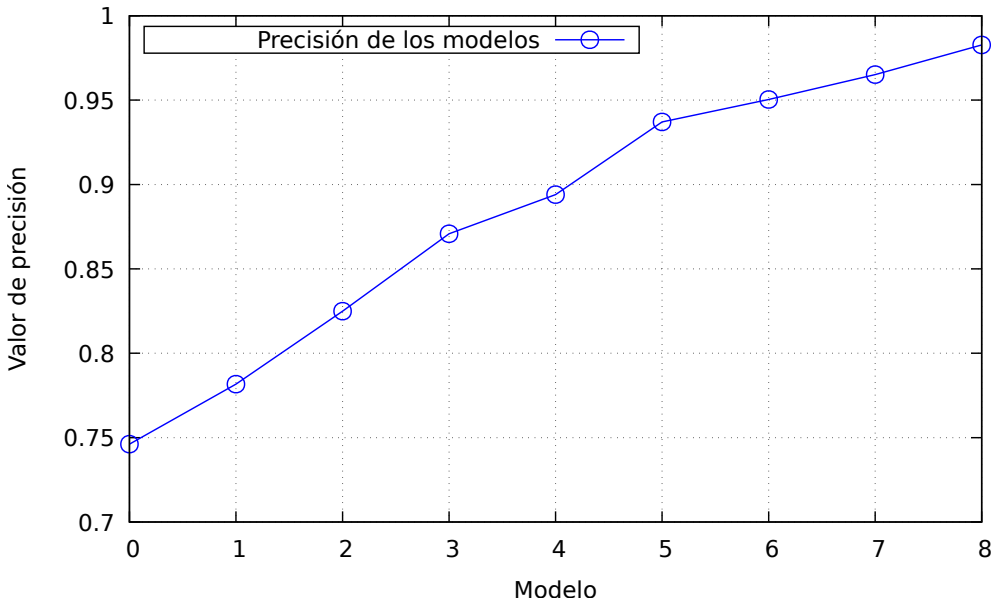


Figura 4.20: Evolución de la precisión, el valor de la precisión crece con cada modelo, más en unos que en otros. Los modelos más simples(como puede ser el modelo 00) comienzan con valores que parecen bajos en comparación con los modelos posteriores, sin embargo estos valores son bastante competitivos considerando factores como numero de variables o tamaño de la base de datos.

A todo esto es posible adjuntar un resumen del desempeño de los modelos a

través de la siguiente tabla:

Modelo	Precisión	Falsos positivos	Falsos negativos
0	74 %	820	250
1	78 %	520	400
2	80 %	470	200
3	87.08 %	330	130
4	89 %	230	110
5	93.71 %	150	38
6	95.04 %	100	37
7	96 %	65	28
8	98.27 %	30	14

Una parte importante de la evaluación de los resultados de los modelos es ver que tan apegados a la realidad están, por lo que se realizó un análisis estadístico de la base de datos, en donde se puede ver que conforme más se avanza en los semestres, el grueso de los datos se aleja cada vez más de los valores reprobatorios(Figura 4.21), lo cual se puede ver en los resultados de los modelos (Figuras 4.2-4.19) donde la categoría *baja* se ve cada vez más reducida.

Habiendo obtenido un máximo de más del 98 % de precisión, es seguro afirmar que el modelo cumple con la hipótesis establecida al comienzo de la tesis 1.1, incluso considerando el primer modelo (que cuenta sólo con la información de antes de que el alumno entrara a la universidad) el cual tiene una precisión de arriba del 70 %, podemos afirmar que es posible el entrenamiento efectivo de una red neuronal capaz de predecir el estatus académico próximo de los estudiantes de la Universidad Autónoma de Chihuahua.

En cuanto a la aportación de las variables, a modo de evaluar esas aportaciones se implemento un análisis de componentes principales (PCA por sus siglas en inglés), el cual consiste en evaluar la varianza que aportan las distintas variables a la descripción del modelo con el fin de decidir que variables pueden ser quitadas del problema. El análisis de PCA que se realizó puede resumirse de manera muy sencilla, debido a que los valores obtenidos del análisis de PCA son muy similares entre sí, y ninguno de los mismos es considerablemente bajo o alto, se puede afirmar que de todas las variables utilizadas, todas aportan de manera similar al problema(Figuras 4.22-4.23).

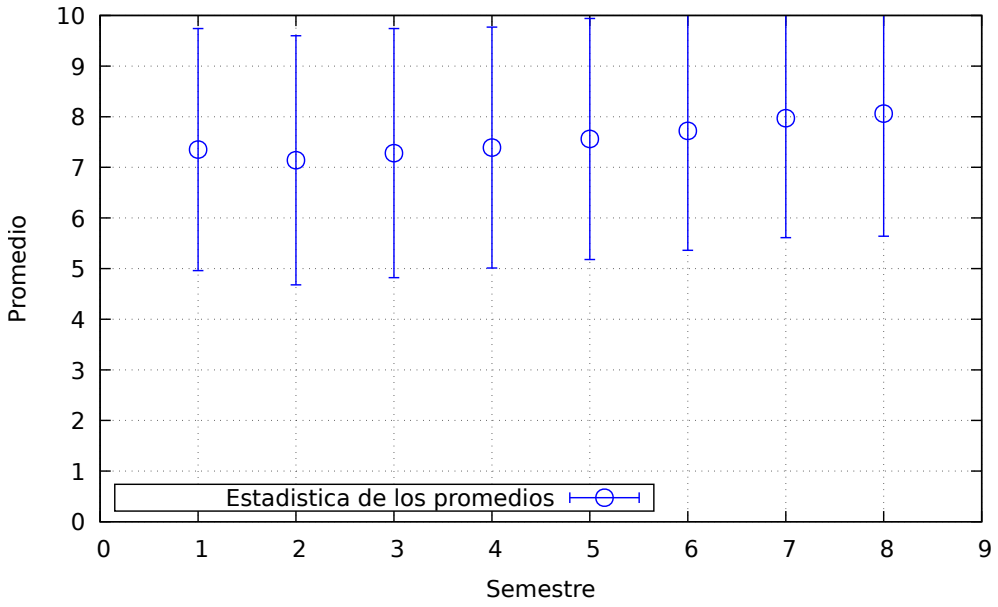


Figura 4.21: Evaluación estadística de los promedios por semestre, en las barras se puede ver que tanto varía la mayor parte de los datos de promedio, dando un mejor contexto a la reducción de la cantidad de datos de *baja* en los modelos superiores, esto ya que el rango de calificaciones se aleja de los valores reprobatorios calculado con la desviación estándar.

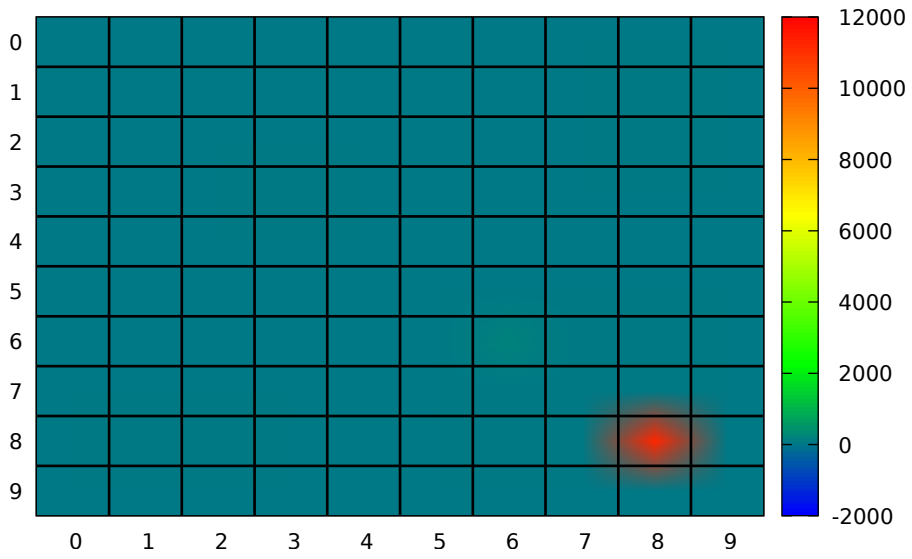


Figura 4.22: Gráfica de covarianza del análisis de componentes principales. El resultado muestra una covarianza muy baja entre todas las variables, con valores muy semejantes entre sí, esta semejanza es evidente en el color prácticamente uniforme entre todas las casillas de la matriz. La uniformidad entre los valores de la matriz indica que la aportación de las variables al modelo está balanceada, y por lo tanto ninguna variable aporta más o menos a los resultados del modelo.

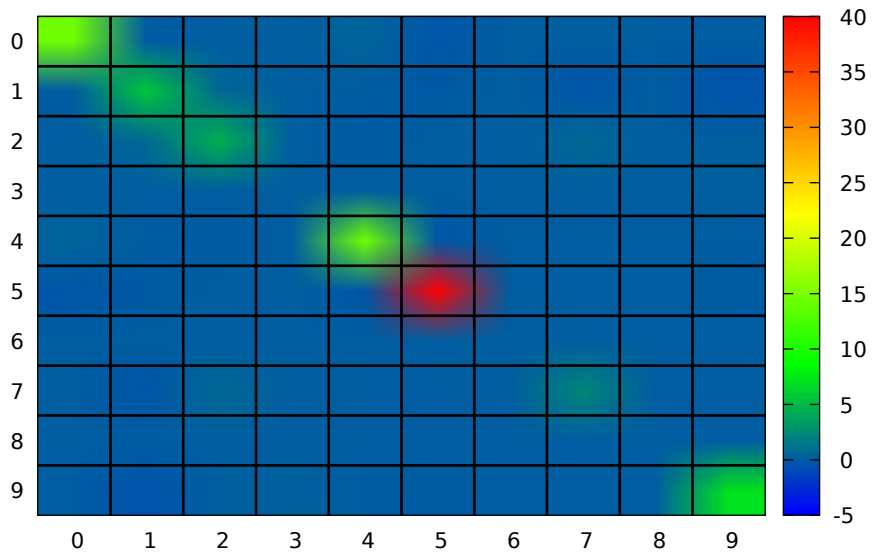


Figura 4.23: Gráfica de precisión del análisis de componentes principales. Los valores obtenidos de este análisis de precisión por medio del PCA da pie a sustentar la afirmación de que todas las variables utilizadas en el modelo tienen aportaciones similares entre si, y no es recomendable prescindir de ninguna de las variables utilizadas si se espera tener mejores resultados. A pesar de que los valores son menos uniformes que en el caso de la covarianza (Figura 4.22) siguen siendo valores muy pequeños que llevan a la conclusión de que las aportaciones se encuentran balanceadas entre si, sin resaltar ni ser irrelevante alguna de ellas.

CAPÍTULO: 5

Conclusiones

5.1. Conclusiones

Llegados a este punto es necesario preguntarnos de manera más específica si el proyecto cumplió con las expectativas colocadas en el mismo. A primera vista podemos decir sin temor a exagerar que los modelos de red neuronal desarrollados en este proyecto cumplen con las hipótesis esperadas de clasificar correctamente a la mayoría de los datos pertenecientes a la base de datos estudiantil de la Universidad Auńonoma de Chihuahua, y aunque esto es cierto la calidad de esta clasificación sigue abierta a mejorar.

Siendo más claro, los modelos diseñados con el fin de predecir a los estudiantes de los primeros semestres, así como a los nuevos ingresos, son los modelos con menor precisión de entre todos los desarrollados, y al mismo tiempo son los que se necesitan mejorar más debido a la importancia de estos primeros semestres en la vida académica de los estudiantes. A pesar de los numerosos esfuerzos puestos en mejorar la predicción en esos semestres en particular, debido a la elección de variables para los modelos, la información relacionable con el desempeño demostró ser un impedimento a la mejoría de estos modelos. Este problema, sin embargo, se convierte en una excelente área de oportunidad, debido a que deja abierta la puerta a la implementación de otros descriptores en el modelo de red que podrían conseguirse en un futuro cuando los estudiantes ingresan a la

universidad.

Un área de oportunidad en el caso de la base de datos es el contenido de información anterior al ingreso del estudiante en la misma, como puede ser información sobre escuelas de procedencia o promedios semestrales durante la educación media superior. También otro tipo de factores socioeconómicos como pueden ser la pertenencia de automovíl propio, el uso de transporte público y las rutas de trasporte público que toman los estudiantes.

A fin de cerrar esta sección quisiera dejar al lector con la siguiente idea, este tipo de trabajos en redes neuronales han demostrado en el tiempo reciente lo poderosas que se pueden volver este tipo de metodologías a fin de ayudar a los estudiantes y a las instituciones educativas a llevar un mejor seguimiento de la vida académica con el fin de regular y mejorar los programas de tutorías a estudiantes y reducir la posibilidad de los mismos de abandonar sus estudios, entonces un desarrollo de bases de datos cada vez más completas se vuelve una prioridad a fin de apoyar el desarrollo de más proyectos como este, que a su vez ayudarán a las instituciones.

Referencias

Alsalman, Y. S., Khamees Abu Halemah, N., Alnagi, E. S., y Salameh, W. (2019). Using Decision Tree and Artificial Neural Network to Predict Students Academic Performance. *2019 10th International Conference on Information and Communication Systems, ICICS 2019*, 104–109. doi: 10.1109/IACS.2019.8809106

Aydogdu, S. (2020). Predicting student final performance using artificial neural networks in online learning environments. *Education and Information Technologies*, *25*(3), 1913–1927. doi: 10.1007/s10639-019-10053-x

Bassi, J. S., Dada, E. G., Hamidu, A. A., y Elijah, M. D. (2019). Students Graduation on Time Prediction Model Using Artificial Neural Network. *IOSR Journal of Computer Engineering (IOSR-JCE)*, *21*(3), 28–35. doi: 10.9790/0661-2103012835

Das, S., Dey, A., Pal, A., y Roy, N. (2015). Applications of Artificial Intelligence in Machine Learning: Review and Prospect. *International Journal of Computer Applications*. doi: 10.5120/20182-2402

De Boer, P. T., Kroese, D. P., Mannor, S., y Rubinstein, R. Y. (2005). A tutorial on the cross-entropy method. *Annals of Operations Research*, *134*(1), 19–67. doi: 10.1007/s10479-005-5724-z

Francis, B. K., y Babu, S. S. (2019). Predicting Academic Performance of Students Using a Hybrid Data Mining Approach. *Journal of Medical Systems*, *43*(6). doi: 10.1007/s10916-019-1295-4

Haykin, S. (1990). *Neural networks and learning machines* (A. Dworkin, Ed.). Hamilton, Ontario, Canada: McMaster University.

Jordan, M. I., y Mitchell, T. M. (2015). *Machine learning: Trends, perspectives,*

and prospects. doi: 10.1126/science.aaa8415

Kingma, D. P., y Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

McClelland, J. L., Rumelhart, D. E., Group, P. R., y cols. (1986). *Parallel distributed processing* (Vol. 2). MIT press Cambridge, MA.

Rosenblatt, F. (1962). A comparison of several perceptron models. *Self-Organizing Systems*, 463–484.

Samuel A.L. (1959). Some Studies in Machine Learning. *IBM Journal of Research and Development*, *44*(1.2).

Sashank, J. R., Satyen, K., y Sanjiv, K. (2018). On the convergence of adam and beyond. En *International conference on learning representations* (Vol. 5, p. 7).

Thomas M. Cover, J. A. T. (2006). *Elements of information theory*. Wiley Interscience.

Treasure-Jones, T., Sarigianni, C., Maier, R., Santos, P., y Dewey, R. (2019). Scaffolded contributions, active meetings and scaled engagement: How technology shapes informal learning practices in healthcare SME networks. *Computers in Human Behavior*, *95*(April 2018), 1–13. Descargado de https://doi.org/10.1016/j.chb.2018.12.039 doi: 10.1016/j.chb.2018.12.039

Waheed, H., Hassan, S. U., Aljohani, N. R., Hardman, J., Alelyani, S., y Nawaz, R. (2020). Predicting academic performance of students from VLE big data using deep learning models. *Computers in Human Behavior*, *104*. doi: 10.1016/j.chb.2019.106189