Universidad Nacional Autónoma de México

Facultad de Ciencias

Redes Neuronales
Grupo 7111

Implementación del perceptrón multicapa para un problema de clasificación de pequeñas empresas en los Estados Unidos

---

Alumnos:
Valencia Cruz Jonathan Josué
Miguel Ángel Liera
Cruz Ramírez Nicolás

Profesor: Sergio Hernández López
Ayudantes: José Alejandro Rojas López
Julio César Misael Monroy González

1 Introducción

La Agencia Federal de Pequeños Negocios en EU (SBA por sus siglas en inglés) es una agencia gubernamental encargada de brindar ayuda a las pequeñas empresas que desean iniciar, construir o expandir sus negocios y actividades. Entre los servicios que ofrece esta agencia se encuentran programas de asesorías, capital, contratación gubernamental, entre otros. Su fundación se remonta a 1953 y desde entonces ha ayudado a empresas a progresivamente mejorar sus negocios e incrementar su nivel de operaciones, algunos de los casos de éxito más conocidos son Fedex y Appel. En la actualidad la SBA cuenta con oficinas en cada uno de los estados de los Estados Unidos, y continúa promoviendo beneficios sociales tales como la creación de empleos que ofrecen las empresas que inician sus operaciones. Entre los programas que tiene la SBA existe uno llamado "Garantía de préstamo", dicho programa busca incentivar a los bancos de aprobar préstamos a pequeñas empresas, ya que la SBA se compromete a cubrir una parte de la deuda de la empresa en caso de incumplimiento. Esto es parecido al funcionamiento de un seguro, donde la SBA absorberá parte de la deuda en caso insolvencia de las pequeñas empresas. Sin embargo, aun cuando la SBA garantiza un porcentaje del préstamo, los bancos siguen expuestos a la probabilidad de incumplimiento y pérdidas económicas. Para los bancos en EU es importante analizar con métodos confiables la información de las empresas a fin de tomar la decisión de otorgarles el préstamo o no. En este proyecto nos daremos a la tarea de implementar un modelo de red neuronal para clasificación de empresas que solicitan préstamos bajo el programa de "Garantía de préstamo".
La razón del uso de una red neuronal para la resolución del problema está en que este se soluciona a partir de una respuesta sí o no, es decir, a partir de los diferentes datos obtenidos de una empresa se debe tomar una decisión binaria, caso que puede ser complicado de resolver pues a primera instancia nos podemos preguntar, ¿Qué datos están relacionados en la predicción de que una empresa cumpla con el préstamo solicitado?, ¿De qué manera contribuyen?, ¿Qué datos son innecesarios?, la respuesta a éstas preguntas y más se presentarán en las siguientes secciones del presente escrito.

2 Planteamiento del problema

Los bancos que reciben solicitudes de créditos de empresas pequeñas deben tomar la decisión de otorgar el crédito o rechazar la solicitud. Esta decisión es de suma importancia, ya que están expuestas al riesgo de incumplimiento por parte del prestatario; para mantener una cartera de créditos sana recurren a diversos métodos para mitigar el riesgo. En la actualidad se usan diversas herramientas para ayudar a la toma de decisiones, una de las alternativas es la implementación de modelos de Aprendizaje de Máquina o Aprendizaje Profundo, estos permiten para estimar el comportamiento futuro de sus clientes de acuerdo a sus características. En este trabajo se busca obtener el mejor modelo para realizar esta estimación del comportamiento de la empresas (cumplimiento o incumplimiento de su deuda) en función de sus características.

3 Objetivo

Implementar una red neuronal artificial de tipo perceptrón multicapa para realizar la clasificación de pequeñas empresas que solicitan créditos en los Estados Unidos.

4 Marco teórico

Como se sabe, un perceptrón multicapa se trata de un modelo de una red neuronal artificial, es decir, imita el funcionamiento del cerebro humano para resolver problemas en computación. Más específicamente de clasificación y regresión. Son especialmente útiles para resolver problemas descomponiendo estos en partes computables por cada una de las neuronas, de manera que la red colabora en conjunto para computar una pequeña porción del problema.

Perceptrón

El perceptrón es la unidad básica de procesamiento, como se mencionó, este tiene n entradas asociadas a un peso. Cada entrada es multiplicada por el peso que se le asigna, de manera que todos estos resultados se sumen, más un sesgo (en la más común de las implementaciones) para que a partir de una función de activación la neurona emita una señal, que regularmente se trata de un número comprendido entre 0 y 1. De esta forma, el perceptrón divide a partir de un hiperplano las entradas que se le dan a este, clasificando como clase [0,1] a aquellas que devuelvan el entero al ser procesadas.
El perceptrón multicapa es una de las arquitecturas de redes neuronales más populares, permite resolver problemas tanto de clasificación o de regresión por lo

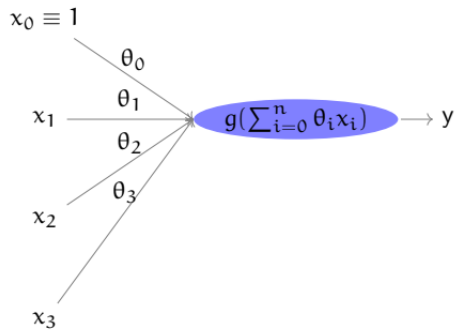


Figure 1: Perceptrón: Donde x_n se trata de las entradas, θ_n los pesos asignados a cada entrada y $g()$ es la función de activación.

Redes Neuronales

Las redes neuronales hacen uso del perceptrón como unidad de ensamble. Una red neuronal está compuesta por un número indeterminado de neuronas organizadas en capas. (El número de capas y neuronas en una red es un tema de vital importancia en la creación de un modelo, ya que no existe una métrica que nos indique a priori cual es la óptima cantidad de neuronas y capas para nuestro problema). Cada perceptrón recibe las salidas

de los perceptrones de la capa anterior al actual, así hasta obtener una salida final que determina una clase resultante para las entradas o una salida asociada en un modelo de regresión. A este tipo de empleo de la red se le llama *feedforward*.

Esta es una de las principales razones por la cual se hará uso de una red neuronal con el objetivo de clasificar si un crédito puede ser otorgado o no a determinada empresa, usando los atributos del dataset como punto de partida para el entrenamiento y generar así una clasificación.

Entrenamiento

Backpropagation se trata de un algoritmo que compara las salidas de la red con la salida esperada de la red con la actual, de manera que trata incrementar las salidas de la capa anterior que potencian la salida que se espera que pertenezca a una clase, decrementando las salidas que potencian una salida ajena a la esperada. De esta forma, se toma como referencia la salida esperada, modificando los pesos a partir de técnicas como descenso por el gradiente, desde la última capa hasta la primera de ellas. Así logrando ajustar la red al modelo

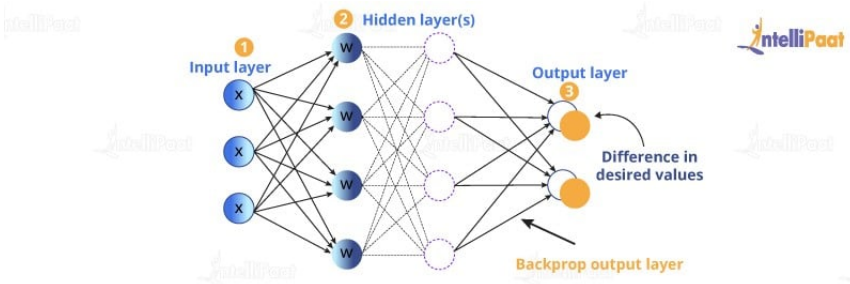


Figure 2: Backpropagation: El algoritmo compara la salida con el resultado deseado, reasigna pesos.

Regularización

Uno de los retos al que nos enfrentamos al implementar una red neuronal es al problema de sobre ajuste, esto ocurre cuando la red se desempeña bien en el conjunto de entrenamiento pero tiene mal desempeño en el conjunto de prueba. Decimos que la red ha "memorizado" las tuplas de entrenamiento y por lo tanto no puede clasificar correctamente tuplas que no ha visto antes. Para combatir este problema se suelen las siguientes técnicas.

Dropout consiste en "desactivar" con cierta probabilidad un conjunto de neuronas en cada iteración de la fase de entrenamiento, estas neuronas al estar inactivas no intervienen en la etapa de forwardpropagation y backpropagation. Al no estar todas la neuronas "prendidas" en cada iteración de entrenamiento, se evita que las neuronas aprendan patrones muy específicos y que sean dependientes de neuronas vecinas.

Normalización del batch consiste en realizar la normalización de los datos no únicamente en la capa de entrada, sino también en capas ocultas de la red.

5 Metodología

Este proyecto se realizará en las siguientes etapas:

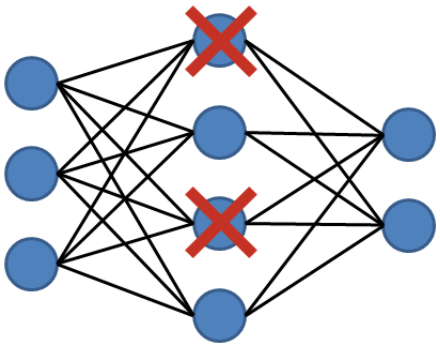


Figure 3: Dropout: Para evitar neuronas con poca importancia en el resultado final, se "apagan" de forma al azar en cada iteración.

- Preparación de los datos.
- Implementación del modelo.
- Validación de métricas de ajuste.
- Selección del modelo.

6 Desarrollo

6.1 Preparación de los datos

En este etapa se realizó la preparación y el preprocesamiento de los datos. Los datos con los que se trabajó se obtuvieron de Kaggle.com, de la página denominada Should This Loan be Approved or Denied?. Esta base consta de 899,164 filas y 27 columnas. Los datos se presentan en formato tabular, las columnas representan caracteristicas de cada préstamos y cada fila representan un préstamo realizado a una pequeña empresa. En la siguiente tabla 4 se presenta a manera de resumen el significado de cada una de las variables independientes del dataset. Entre algunas de las características se cuenta con monto del crédito, año del préstamo, número de empleados de la empresa, monto del préstamo garantizado por la SBA, si el negocio pertenece a un ambiente urbano o rural, etc. En total son 13 variables cuantitativas, 13 variables categóricas y 1 variable objetivo. Conociendo el significado de las variables se procedió a realizar la limpieza de los datos, toda vez que algunas columnas de tipo cuantitativo tenían caracteres, símbolos de moneda, entre otros. Posteriormente se procedió a eliminar aquellas variables que resultaban poco útiles como el identificador de la transacción, y otra variables que contenían la misma información que la variable objetivo y que nos arrojarían resultados erróneos, por ejemplo, fecha en que el crédito cayó en default. Adicional a esto se realizó la imputación de valores NA, en general el dataset contaba con muy pocas columnas con valores faltantes, por lo cual no se modificó a la mayoría de ellas, sin embargo 2 tenían un procentaje de NA mayor de 22% y 28% por lo cual se optó por eliminarlas.
En la tabla 6c se puede observar un resumen con las decisiones tomadas. Posteriormente se

implementó una técnica de reducción de dimensión para las variables cuantitativas y para variables dummy obtenidas de las variables categóricas, sin embargo esto nos presentó la problemática de que el número de componentes para explicar un porcentaje de varianza mayor al 70% de los datos resultaba en más de 20 variables, por lo cual como primera estrategia se optó por aplicar PCA únicamente a las variables cuantitativas lo cual nos resultó en únicamente 3 componentes que alcanzaban a explicar el 96% de la varianza del dataset, y con estas se pasó a la etapa de implementación del modelo de red neuronal.

Variable	Tipo	Descripción
LoanNr_ChkDgt	Cuantitativa	Id de la tabla
ApprovalFY	Cuantitativa	Año de aprobación del préstamo por parte de la SBA
ApprovalDate	Cuantitativa	Fecha de aprobación del préstamo
Term	Cuantitativa	Plazo del préstamo en meses
NoEmp	Cuantitativa	Número de empleados en la empresa
CreateJob	Cuantitativa	Número de empleos creados por la empresa
RetainedJob	Cuantitativa	Número de empleos retenidos por la empresa
BalanceGross	Cuantitativa	Monto pendiente por cubrir
ChgOffPrinGr	Cuantitativa	Monto del préstamo que no fue pagado
GrAppv	Cuantitativa	Monto de préstamo aprobado
SBA_Appv	Cuantitativa	Monto del préstamo garantizado por la SBA
DisbursementGross	Cuantitativa	Monto desembolsado
DisbursementDate	Cuantitativa	Fecha en que se desembolso el crédito
Name	Categórica	Nombre de la empresa que solicitó el prestamo
City	Categórica	Ciudad de donde proviene la empresa
State	Categórica	Estado de donde proviene la empresa
Zip	Categórica	Código de ubicación de la empresa.
Bank	Categórica	Nombre del banco que otorgo el préstamo
BankState	Categórica	Estado del banco que otorgo el préstamo
NAICS	Categórica	Clasificación de negocios de EU por tipo de actividad
NewExist	Categórica	si la empresa tiene 2 años de existencia o no
FranchiseCode	Categórica	Indentificador de si la empresa tiene franquicias o no.
UrbanRural	Categórica	Indicador si el negocio es rural o no.
RevLineCr	Categórica	Si el crédito es revolvente o no. Yes o No
LowDoc	Categórica	Préstamo procesado usando aplicación.
ChgOffDate	Categórica	Fecha cuando el préstamo se declaró en default

Figure 4: Descripción de las columnas del dataset.

Variable	Tipo	Descripción	Decisión	Explicación
LoanNr_ChkDgt	Cuantitativa	Id de la tabla	**Eliminar**	Representa el id de una tabla
ApprovalFY	Cuantitativa	Año de aprobación del préstamo por parte de la SBA		
ApprovalDate	Cuantitativa	Fecha de aprobación del préstamo	**Eliminar**	Se considerará sólo el año, el cual está en ApprovalFY
Term	Cuantitativa	Plazo del préstamo en meses		
NoEmp	Cuantitativa	Número de empleados en la empresa		
CreateJob	Cuantitativa	Número de empleos creados por la empresa		
RetainedJob	Cuantitativa	Número de empleos retenidos por la empresa		
BalanceGross	Cuantitativa	Monto pendiente por cubrir		
ChgOffPrinGr	Cuantitativa	Monto del préstamo que no fue pagado	**Eliminar**	Tiene la misma información que la variable a objetivo
GrAppv	Cuantitativa	Monto de préstamo aprobado		
SBA_Appv	Cuantitativa	Monto del préstamo garantizado por la SBA		
DisbursementGross	Cuantitativa	Monto desembolsado		
DisbursementDate	Cuantitativa	Fecha en que se desembolso el crédito		
Name	Categórica	Nombre de la empresa que solicitó el prestamo	**Eliminar**	No es relevante el nombre de la empresa ya que es tambien un id
City	Categórica	Ciudad de donde proviene la empresa	**Eliminar**	Se agrupará en una categoría más general como State
State	Categórica	Estado de donde proviene la empresa		
Zip	Categórica	Código de ubicación de la empresa.	**Eliminar**	Se agrupará en una categoría más general como State
Bank	Categórica	Nombre del banco que otorgo el préstamo		
BankState	Categórica	Estado del banco que otorgo el préstamo		
NAICS	Categórica	Clasificación de negocios de EU por tipo de actividad	**Eliminar**	Tiene más del 22% de valores NA
NewExist	Categórica	si la empresa tiene 2 años de existencia o no		
FranchiseCode	Categórica	Indentificador de si la empresa tiene franquicias o no.		
UrbanRural	Categórica	Indicador si el negocio es rural o no.		
RevLineCr	Categórica	Si el crédito es revolvente o no. Yes o No	**Eliminar**	Tiene más del 28% de valores NA
LowDoc	Categórica	Préstamo procesado usando aplicación.		
ChgOffDate	Categórica	Fecha cuando el préstamo se declaró en default	**Eliminar**	Tiene la misma información que la variable a objetivo

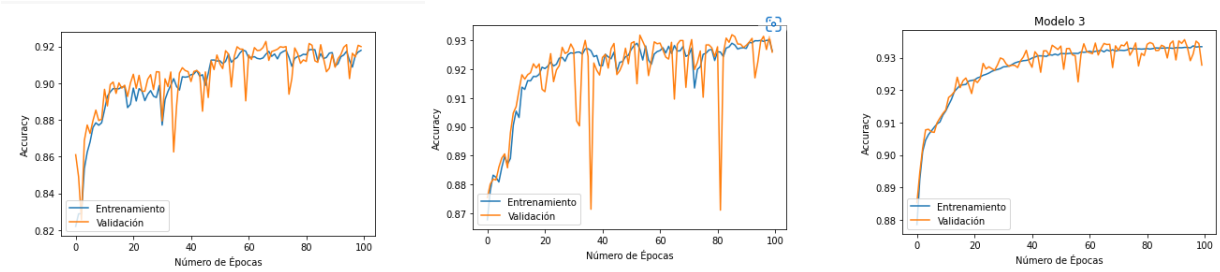
Figure 5: Resumen de la etapa de preprocesamiento de datos.

6.2 Implementación del modelo

Una vez realizada la tarea de preprocesamiento, se procedió a ajustar el modelo perceptrón multicapa. Para un modelo de red neuronal se requiere definir una gran cantidad de parámetros, entre ellos se tiene:

- Número de capas ocultas.
- Número de neuronas en cada capa oculta.
- Funciones de activación de cada capa
- Número de épocas.
- Tamaño del lote.
- Pesos iniciales de la red.
- Tasa de aprendizaje.

También es necesario definir el tipo de optimización que utilizará, el más común es a través del descenso del gradiente estocástico, sin embargo se utilizó el método Adam, ya que en la práctica nos brindó mejores resultado que SGD. Para la implementación de la



(a) Perceptrón con 5 capas, usando TanH como activación en capa final, usando normalización del batch.

(b) Perceptrón con 5 capas, usando TanH como activación en capa final, sin usar normalización del batch.

(c) Perceptrón con 5 capas, usando Sigmoide como activación en capa final, sin usar normalización del batch.

Figure 6: Gráficas de precisión versus número de épocas para los conjuntos de entrenamiento y prueba

red se uso la paquetería tensorflow.keras, ya que es sumamente intuitiva y permite seleccionar diversas características de cada capa de la red, a diferencia de otras paqueterías como sklearn que implementa el perceptrón multicapa sin suficientes opciones para elegir. Como sabemos no existe una regla general para elegir estos parámetros, por lo cual se procedió a realizar diversas pruebas variando los valores en cada una de ellas. Se comenzó por explorar el número de capas ocultas, primero con 1, 2 , 3, etc, y comenzando con un número fijo de 20 nodos en cada una. Con esto obtuvimos que con 5 capas la precisión del modelo se mantenía por arriba del a 89% en el conjunto de prueba, por lo cual decidimos que se implementaría la red con 5 capas ocultas. Posteriormente se varió el número de nodos en las capas ocultas, con 20, 30, 50, etc, nos encontramos con que con mas de 30 nodos en las capas ocultas no se obtuvo mejora significativa en la precisión del modelo en los datos de prueba, por lo cual se decidió continuar con 30 nodos en las capas ocultas. Se siguió probando ahora con el número de lotes, para más de 300 lotes el algoritmo terminaba rápidamente pero la precisión no mejoraba más allá de 91% mientras que para un mayor número de lotes se obtuvo una precisión del 92%, por esto se siguió probando con 100 lotes en total. Con estas configuraciones se probaron también distintas funciones de activación en el nodo final, y los resultado para este momento fueron bastante parecidos entre tanh y sigmoide, logrando a lo más obtener una precisión del 92%.

6.3 Validación de métricas de ajuste

Posteriormente se procedió a obtener la matriz de confusión y algunas otras métricas para cada uno de los modelos. Como se observó en las gráficas, la precisión de los modelos fue cercada al 92% en el conjunto de entrenamiento y prueba. Por lo cual no hubo mucha diferencia al observar sus matrices de confusión y al obtener su sensibilidad y especificidad. En promedio se obtuvo una sensibilidad del 95% y una especificidad del 80%.

6.4 Selección del modelo

Finalmente al seleccionar el modelo se pueden recurrir a técnicas de remuestreo como bootstrap para cuantificar de mejor manera las métricas del modelo, pero en este caso por el volumen de los datos se optó por no seguir ese camino. En general los modelos que se probaron al final resultaron muy parejos ya que básicamente contaban con el mismo número de capas intermedias y sólo se varió tanto el método de regularización como las función de activación final. Consideramos que para esta etapa cualquiera de estos puede usarse para realizar la clasificación de clientes.

7 Resultados y conclusiones

Se realizaron distintas pruebas usando diferente número de capas, funciones de activación tanto en las capas intermedias como en la capa de salida, así como distintos tamaño de lote y número de épocas. Se observó que los resultados usando 5 capas, en promedio 25 neuronas en cada capa, usando el método de optimización Adam, nos arrojaron los mejores resultados. También se observó que al transformar las variables categóricas a variables dummy y agregarlas como input a la red, los resultados no mejoraron, de hecho disminuyó la precisión del modelo, por lo cual se descartó su uso.

Además de la implementación del uso del modelo de una red neuronal, se hizo usó de un modelo de árbol de clasificación. Uno de los inconveniente más importantes al usar este tipo de clasificador fue que el despliegue del árbol de por si, generó un árbol con una gran cantidad de nodos. Es de esperar que el árbol obtenido sea así de grande, pues la cantidad de parámetros en el dataset de entrenamiento lo ameritan.

8 Bibliografía

1. Ethem Alpaydin, Introduction to Machine Learning', 2nd Edition, The MIT Press, 2010.

2. Data Mining Concepts and Techniques. Jiawei Han, Micheline Kamber & Jian Pei. Morgan Kaufmann. Tercera Edición.

3. Applied Multivariate Statistical Analysis. Richard A. Johnson and Dean W. Wichern. Pearson. Sexta edición.