

LAB02

PMC



20 de marzo de 2023

Diego Viñals Lage

Miguel Ángel Lobo Bartolomé

Pablo Medina de la Iglesia

Javier Garrido Cobo

Contenido

[Práctica 1 2](#_Toc130238434)

[Resuelve la función AND. Prueba diferentes learning rates y umbrales para saber cuál es el óptimo. Sacar por pantalla las salidas de la red neuronal y el valor del error para cada iteración. 2](#_Toc130238435)

[Repetir el ejercicio anterior para la función XOR. 4](#_Toc130238436)

[Práctica 2 6](#_Toc130238437)

[Continúa en la memoria creando varias tablas (como la siguiente) con los resultados del MLP que resuelve la función. Cada fila de la tabla será una de las variaciones que hayas probado para ese modelo. Haz una tabla distinta para los experimentos con cada una de las dos funciones de activación propuestas. 6](#_Toc130238438)

[Práctica 3 9](#_Toc130238439)

[Explica cómo has llevado a cabo la categorización y la normalización de los datos de entrada. 9](#_Toc130238440)

[Prueba distintas arquitecturas (en número de capas y número de neuronas por capa) ¿Cuál es la mejor arquitectura? Dibújala y justifícala con una tabla que recoja los valores de loss y accuracy para el conjunto de entrenamiento y el de validación para las distintas pruebas que has llevado a cabo. 9](#_Toc130238441)

[Bibliografía 13](#_Toc130238442)

# Práctica 1

## Resuelve la función AND. Prueba diferentes learning rates y umbrales para saber cuál es el óptimo. Sacar por pantalla las salidas de la red neuronal y el valor del error para cada iteración.

La función AND es una función linealmente separable, para una serie de entradas, se puede separar las salidas esperadas con una línea.

Gráfico

Descripción generada automáticamente

A la hora de resolver el problema, probamos con diferentes learning rates y con umbrales diferentes obteniendo los siguientes resultados.

Interfaz de usuario gráfica, Texto, Chat o mensaje de texto

Descripción generada automáticamentePara learning rate = 0.1 y umbral = 0.3, obtenemos error 0 en 15 épocas.

Interfaz de usuario gráfica, Texto, Chat o mensaje de texto

Descripción generada automáticamentePara learning rate = 0.1 y umbral = 0.2, obtenemos error 0 en 16 épocas.

Para learning rate = 0.3 y umbral = 0.7, obtenemos error 0 en 6 épocas.

Interfaz de usuario gráfica, Texto, Chat o mensaje de texto

Descripción generada automáticamente

Para learning rate = 0.6 y umbral = 0.9, obtenemos error 0 en 4 épocas.

Interfaz de usuario gráfica, Texto, Chat o mensaje de texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Chat o mensaje de texto

Descripción generada automáticamenteFinalmente, para learning rate = 0.8 y umbral = 1, obtenemos error 0 en 4 épocas. Siendo este nuestro mejor resultado en el set de pruebas, lo almacenamos en el archivo CSV mostrando para cada época la variación en la matriz de pesos, error, valores de entrada y error esperado.

Las pruebas se han realizado con la misma matriz de pesos con valores aleatorios. En cada caso la ecuación del hiperplano quedaría de la siguiente forma:

Ecuación: W**1 \* X1 + W2 \* X2 – O**

**W1 y W2:** valores de los pesos

**X1 y X2:** variables de entrada

**O:** Umbral

Esta ecuación define una línea recta en un plano 2D que separa las entradas en dos clases: la clase de entradas que producen una salida positiva del perceptrón y la clase de entradas que producen una salida negativa: 0.54 \* X1 + 0.64 \* X2 – 0.9 = 0

## Repetir el ejercicio anterior para la función XOR.

Al analizar el problema de una puerta XOR, nos podemos dar cuenta de que no es un problema linealmente separable. A la hora de realizar pruebas, los valores de learning rate y umbral nunca llegarán a ser los idóneos para obtener una solución al problema.

Gráfico

Descripción generada automáticamente

Con unos valores de learning rate 0.6 y umbral 0.9, obtenemos los siguientes resultados:

Tabla

Descripción generada automáticamente

Como límite para encontrar soluciones a los problemas planteados se establece mil épocas. A la hora de analizar los resultados, podemos observar que el problema no tendrá solución ya que los pesos y errores para la penúltima época obtenida y la última época obtenida son iguales. Al no sufrir modificaciones en la matriz de pesos esta nunca será capaz de entrenar para encontrar una solución.

Se podrían seguir probando distintos valores de learning rate y umbral, pero el problema nunca tendría solución con un perceptrón lineal, se necesitaría implementar un MLP.

# Práctica 2

## Continúa en la memoria creando varias tablas (como la siguiente) con los resultados del MLP que resuelve la función. Cada fila de la tabla será una de las variaciones que hayas probado para ese modelo. Haz una tabla distinta para los experimentos con cada una de las dos funciones de activación propuestas.

Para resolver esta cuestión hemos creado tres modelos distintos, para cada función de activación usada, se han probado los distintos modelos. El modelo 1, contiene una capa oculta con 2 neuronas. El modelo 2, tiene una capa oculta con 3 neuronas. El modelo 3, tiene una capa oculta con 4 neuronas.

El rango elegido para las neuronas lo obtenemos de la siguiente expresión:

Texto, Carta

Descripción generada automáticamente

* Donde M es el número de ejemplos del conjunto de entrenamiento, en este caso 4, para cada combinación posible de las entradas. Conjunto entrenamiento: [0,0] | [0,1] | [1,0] | [1,1] |
* Donde N es el número de neuronas de la capa de entrada, 2 neuronas con entradas 0 y 1.
* n por tanto sería el número de neuronas en la capa oculta.

Resolviendo la expresión, el rango de neuronas estaría entre 1 y 4.

Interfaz de usuario gráfica, Aplicación, Sitio web

Descripción generada automáticamenteCon la función de activación sigmoide, obtenemos los siguientes resultados para cada uno de los modelos.

Podemos observar el número de neuronas utilizado para cada modelo junto con:

* Número de épocas utilizado.
* Matriz de los pesos una vez finalizado el entrenamiento.
* El valor de la función de pérdida que se utiliza en el proceso de entrenamiento para ajustar los pesos del modelo y minimizar el error cuadrático medio.
* El número de aciertos a la hora de resolver el problema ,0.5 🡪 2/4 aciertos.

Interfaz de usuario gráfica, Aplicación, Sitio web

Descripción generada automáticamenteAhora realizamos las mismas pruebas con la función de activación ReLu, obteniendo los siguientes resultados.

Podemos observar el número de neuronas utilizado para cada modelo junto con:

* Número de épocas utilizado.
* Matriz de los pesos una vez finalizado el entrenamiento.
* El valor de la función de pérdida que se utiliza en el proceso de entrenamiento para ajustar los pesos del modelo y minimizar el error cuadrático medio.
* El número de aciertos a la hora de resolver el problema ,1 🡪 4/4 aciertos.

Una vez probados los modelos con distintas funciones de activación, para una matriz de pesos aleatoria, el modelo 3, con 4 neuronas en la capa oculta y 900 épocas, es capaz de resolver la puerta XOR, con una tasa de aciertos de 4/4.

Con 4 neuronas el error para ambas funciones de activación es menor respecto al resto de modelos, esto se debe a que cada neurona se aprende cada uno de los posibles elementos en el conjunto de entrenamiento. Cuantas más épocas se utilicen, menor será el error cometido, acercándose cada vez más a 0.

# Práctica 3

## Explica cómo has llevado a cabo la categorización y la normalización de los datos de entrada.

Para la categorización de los datos de entrada se ha utilizado la función to\_categorical de Keras. Esta función toma como argumento el array de datos categóricos y devuelve una matriz donde cada columna corresponde a una categoría y cada fila representa una observación. En cada fila, se coloca un 1 en la columna correspondiente a la categoría y ceros en el resto de las columnas.

Para la normalización de los datos, se ha utilizado la función StandardScaler de Scikit-learn. Esta función estandariza los datos restando la media y dividiendo por la desviación estándar, de manera que los datos tengan una media igual a cero y una desviación estándar igual a uno. Esto permite que todas las variables tengan una escala similar y evita que una variable con valores muy grandes tenga un peso excesivo en la predicción.

## Prueba distintas arquitecturas (en número de capas y número de neuronas por capa) ¿Cuál es la mejor arquitectura? Dibújala y justifícala con una tabla que recoja los valores de loss y accuracy para el conjunto de entrenamiento y el de validación para las distintas pruebas que has llevado a cabo.

Para las diferentes arquitecturas probadas, la arquitectura que tiene mejores valores para loss y accuaracy es la siguiente:

* Capa de entrada, con los posibles inputs del dataset, 20 neuronas.
* Capa oculta con 32 neuronas, función ReLu.
* Capa oculta con 32 neuronas, función ReLu.
* Capa de salida, con 1 neurona, función Sigmoide.

Presentamos los distintos modelos probados con datos de entrenamiento y datos de validación.

Captura de pantalla de computadora

Descripción generada automáticamente con confianza media

Gráfico

Descripción generada automáticamente con confianza media

# Bibliografía

https://becominghuman.ai/multi-layer-perceptron-mlp-models-on-real-world-banking-data-f6dd3d7e998f

https://es.linkedin.com/pulse/artificial-neural-nets-con-keras-framework-jose-luis-sanchez-del-coso

https://anderfernandez.com/blog/como-programar-una-red-neuronal-desde-0-en-python/

Tejedor, Á. J. (s.f.). *IAII\_Canvas*. Obtenido de https://ufv-es.instructure.com/courses/26539/pages/sobre-la-asignatura