# PROYECTO PDI1.2

Homero Meneses
Jesús Alejandro Guzmán Cordero
Enrique Ramirez Pérez
Roberto Misael Reyes

June 2023

## 1 Introduction

Las redes neuronales son modelos computacionales inspirados en el funcionamiento del cerebro humano. Estn compuestas por unidades neuronales interconectadas que pueden aprender de los datos y reconocer patrones

Sirven principalmente para: Reconocimiento de imgenes y voz Clasificacin de datos Aprendizaje automtico Procesamiento del lenguaje natural

Funciones de activación: Las funciones de activación se utilizan en las neuronas artificiales para introducir no linealidades en la red neuronal y permitir la modelación de relaciones complejas en los datos. En este proyecto, se utilizan dos funciones de activación: ReLU (Rectified Linear Unit) y Softmax. ReLU se utiliza en la capa oculta de la red neuronal para introducir no linealidades, mientras que Softmax se utiliza en la capa de salida para obtener probabilidades de clasificación.

Descenso de gradiente: El descenso de gradiente es un algoritmo de optimización utilizado para ajustar los pesos y sesgos de una red neuronal durante el entrenamiento. El objetivo es minimizar una función de costo que mide la discrepancia entre las predicciones de la red y las etiquetas verdaderas. El descenso de gradiente calcula las derivadas parciales de la función de costo con respecto a los pesos y sesgos, y actualiza estos parámetros en la dirección que reduce el costo.

Conjunto de datos Fashion MNIST: Fashion MNIST es un conjunto de datos ampliamente utilizado en el campo del aprendizaje automático para la clasificación de imágenes. Consiste en 60,000 imágenes en escala de grises de prendas de vestir clasificadas en 10 categorías diferentes, y un conjunto de prueba de 10,000 imágenes. Cada imagen tiene un tamaño de 28x28 píxeles.

Preprocesamiento de datos: Antes de utilizar los datos para el entrenamiento de la red neuronal, es necesario realizar algunas transformaciones y prepararlos adecuadamente. En este proyecto, se carga el conjunto de datos Fashion MNIST utilizando la función read idx, y se realiza un preprocesamiento que incluye la

normalización de los píxeles en el rango de 0 a 1 y la transformación de las imágenes en vectores unidimensionales.

Entrenamiento de la red neuronal: Durante el entrenamiento, se realizan múltiples pasadas (épocas) sobre el conjunto de datos de entrenamiento. En cada época, se realiza una propagación hacia adelante para calcular las salidas de la red neuronal, se calcula la función de costo (entropía cruzada) y se realiza una retropropagación para actualizar los pesos y sesgos utilizando el descenso de gradiente.

Evaluación del modelo: Una vez que el modelo ha sido entrenado, se evalúa su rendimiento utilizando el conjunto de datos de prueba. Se realiza una propagación hacia adelante en el conjunto de prueba para obtener las predicciones de la red neuronal. Luego, se compara las predicciones con las etiquetas verdaderas para calcular la precisión del modelo.

## 2 Desarrollo

La red neuronal se entrena y evalúa utilizando el conjunto de datos Fashion MNIST, que consiste en imágenes en escala de grises de prendas de vestir clasificadas en 10 categorías diferentes.

Explciación detallada del código:

Importación de bibliotecas: Se importan las bibliotecas necesarias, incluyendo NumPy para el cálculo numérico, Matplotlib para la visualización de imágenes y CV2 (OpenCV) para el procesamiento de imágenes.

Definición de funciones de activación: Se definen dos funciones de activación comunes: ReLU (Rectified Linear Unit) y Softmax. La función ReLU se utiliza en la capa oculta de la red neuronal, mientras que la función Softmax se utiliza en la capa de salida para obtener probabilidades de clasificación.

Definición de la clase NeuralNetwork: La clase NeuralNetwork representa la red neuronal. Se inicializa con los tamaños de entrada, ocultos y de salida. En el método init, se inicializan los pesos y sesgos de las capas oculta y de salida de la red neuronal. El método forward realiza la propagación hacia adelante de la red neuronal, calculando las salidas de cada capa utilizando las funciones de activación. El método backward realiza la retropropagación del error y actualiza los pesos y sesgos utilizando el descenso de gradiente.

Se define la clase ‘NeuralNetwork‘ que representa la red neuronal. En su método ‘init‘, se inicializan los pesos y sesgos de la red con valores aleatorios usando la función ‘np.random.randn()‘ y se multiplican por un factor de escala de 0.01. La red consta de dos capas: una capa oculta y una capa de salida. El método ‘forward‘ realiza el paso hacia adelante de la red, calculando las activaciones de cada capa mediante la multiplicación de matrices y aplicando las funciones de activación. El método ‘backward‘ realiza el paso hacia atrás de la red para actualizar los pesos y sesgos mediante el algoritmo de retropropagación del error.

Funciones para leer los archivos IDX: Se definen dos funciones: read idx y load data. Estas funciones se utilizan para leer los archivos IDX que contienen

las imágenes y las etiquetas del conjunto de datos Fashion MNIST.

Función para entrenar la red neuronal: La función train model se encarga de entrenar la red neuronal utilizando el conjunto de entrenamiento. Se inicializa una instancia de la clase NeuralNetwork con los tamaños adecuados. Dentro del bucle de entrenamiento, se realiza una propagación hacia adelante, se calcula la función de costo (entropía cruzada) y se realiza una retropropagación para actualizar los pesos y sesgos de la red. Se imprime el costo en cada época para monitorear el progreso del entrenamiento.

Función para evaluar el modelo: La función evaluate model se utiliza para evaluar el rendimiento del modelo entrenado utilizando el conjunto de prueba. Se calculan las predicciones para el conjunto de prueba y se compara con las etiquetas verdaderas para calcular la precisión del modelo.

Función para obtener el nombre de la clase: La función get class name se utiliza para obtener el nombre de la clase correspondiente a un índice de clase dado.

Carga y preprocesamiento de datos: Se llaman las funciones load data para cargar y preprocesar los datos del conjunto de entrenamiento y prueba. Los datos se escalan dividiéndolos entre 255 para normalizarlos en el rango de 0 a 1. Las imágenes se aplanan en vectores unidimensionales para que puedan ser utilizadas como entrada para la red neuronal.

Definición de hiperparámetros y entrenamiento del modelo: Se definen los hiperparámetros, incluyendo el tamaño oculto, la tasa de aprendizaje y el número de épocas. Se llama a la función train model para entrenar el modelo utilizando el conjunto de entrenamiento.

Evaluación del modelo y visualización de resultados: Se llama a la función evaluate model para evaluar el rendimiento del modelo en el conjunto de prueba. Se muestra información detallada sobre las primeras 100 imágenes de prueba, incluyendo la imagen original, la imagen mejorada con color, la imagen binarizada y la clase predicha junto con la confianza del modelo en esa predicción

Se rompe el bucle después de mostrar los resultados para las primeras 100 imágenes de prueba

# 3 Conclusion

Para el desarrollo integral de este proyecto además de trabajar con las redes neuronales incluimos el uso de tecnicas de procesamiento digital de imagenes para mejorar la visualización del resultado. Pensando en que en un pryecto real, es necesario que los usuarios tengan un resultado fiable y claro.