

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

MAESTRÍA EN CIENCIA DE DATOS

**CLASIFICACIÓN DE IMÁGENES USANDO EL
DATASET FASHION MNIST**

POR:

ACT. DAMIÁN ATILANO MARTÍNEZ ALVARADO

1. Introducción

La clasificación de imágenes es una tarea fundamental en el campo de la visión por computadora, con aplicaciones en diversas áreas como el reconocimiento de objetos, la detección de rostros y la identificación de patrones. En este estudio, utilizamos el dataset *Fashion MNIST* para construir y evaluar un modelo de red neuronal convolucional (*CNN*) para la clasificación de artículos de moda.

2. Descripción de los datos

El dataset *Fashion MNIST* consta de 70,000 imágenes en escala de grises de 28x28 píxeles, divididas en 60,000 imágenes para el conjunto de entrenamiento y 10,000 imágenes para el conjunto de prueba.

1. T-shirt/top
2. Trouser
3. Pullover
4. Dress
5. Coat
6. Sandal
7. Shirt
8. Sneaker
9. Bag
10. Ankle boot

El dataset está balanceado, con un número igual de imágenes para cada categoría.

3. Metodología

3.1. Carga y Preprocesamiento de los Datos

Primero, cargamos el dataset utilizando *TensorFlow* y realizamos una normalización de los datos para escalar los valores de los píxeles al rango $[0,1]$. Esto es importante porque los modelos de aprendizaje profundo tienden a funcionar mejor con datos normalizados. Los valores originales de los píxeles están en el rango $[0, 255]$, por lo que al dividir por 255, se escalan a un rango $[0, 1]$. Además, añadimos una dimensión extra para que las imágenes sean compatibles con las capas convolucionales. En imágenes en escala de grises, cada píxel tiene un único valor que representa su intensidad. Sin embargo, las capas convolucionales (*Conv2D*) en *Keras* esperan que las imágenes tengan tres dimensiones: altura, anchura y número de canales (por ejemplo, 3 para imágenes RGB, 1 para imágenes en escala de grises).

3.2. Visualización de datos

Para entender mejor la naturaleza de los datos, visualizamos algunas de las imágenes del conjunto de entrenamiento. La Figura 1 muestra nueve ejemplos de imágenes de entrenamiento junto con sus respectivas etiquetas. Podemos observar la variabilidad en las imágenes, lo que representa un desafío para la clasificación.



Figura 1: Ejemplos de imágenes del conjunto de entrenamiento con sus etiquetas correspondientes.

3.3. Aumento de Datos

Para aumentar la variabilidad del conjunto de entrenamiento y prevenir el sobreajuste, utilizamos técnicas de aumento de datos, como rotaciones, desplazamientos, zoom y volteo horizontal. Estas técnicas permiten generar nuevas muestras a partir de las existentes, aumentando así la diversidad del conjunto de datos. El aumento de datos es el proceso de crear nuevas muestras de datos a partir de las existentes mediante la aplicación de transformaciones aleatorias. Esto ayuda a aumentar el tamaño del conjunto de datos de entrenamiento, mejora la capacidad de generalización del modelo y reduce el riesgo de sobreajuste.

3.4. Construcción del modelo

Construimos un modelo de red neuronal convolucional con dos capas convolucionales seguidas de capas de *pooling* y *dropout* para la regularización.

Las capas convolucionales (*Conv2D*) detectan características locales en las imágenes (como bordes, texturas y patrones). Los filtros se entrenan para resaltar diferentes aspectos visuales en las imágenes. La capa de *pooling* (*MaxPooling2D*) reduce la dimensionalidad de las imágenes, disminuyendo la cantidad de parámetros y el costo computacional. Dropout es una técnica de regularización que previene el sobreajuste apagando aleatoriamente un porcentaje de neuronas durante el entrenamiento. Las capas densas (*Dense*) conectan todas las neuronas de la capa anterior a cada neurona en la capa actual. Se utilizan para combinar las características extraídas y realizar la clasificación final. Las funciones de activación (*ReLU* y *Softmax*) introducen no linealidad en el modelo y convierten las salidas en probabilidades, útiles para problemas de clasificación multiclase.

3.5. Entrenamiento del Modelo

Entrenamos el modelo utilizando el conjunto de datos aumentado, durante 50 épocas, y validamos su rendimiento utilizando el conjunto de prueba. Durante el entrenamiento, el modelo ajusta sus pesos para minimizar la función de pérdida utilizando los datos de entrenamiento. El proceso de entrenamiento implica pasar los datos a través del modelo (*forward pass*), calcular la pérdida, y luego ajustar los pesos mediante el algoritmo de optimización (*backpropagation*). El generador de imágenes (*datagen.flow*) genera lotes de datos de entrenamiento aumentados de manera continua. Esto introduce variabilidad en los datos de entrenamiento y ayuda a prevenir el sobreajuste. Un tamaño de lote define cuántas muestras de datos se procesan antes de que los parámetros del modelo se actualicen. Entrenar durante múltiples épocas ayuda al modelo a ajustar sus pesos de manera incremental y aprender mejor las características de los datos.

3.6. Evaluación del modelo y Visualización de la Precisión y Pérdida

Evaluamos el rendimiento del modelo en el conjunto de prueba, obteniendo una precisión del 85.88 %. La evaluación del modelo es el proceso de medir el rendimiento de un modelo en un conjunto de datos independiente del conjunto de entrenamiento. La pérdida es una métrica que cuantifica cuán bien o mal el modelo predice las etiquetas reales. La precisión es la proporción de predicciones correctas entre el total de predicciones realizadas.

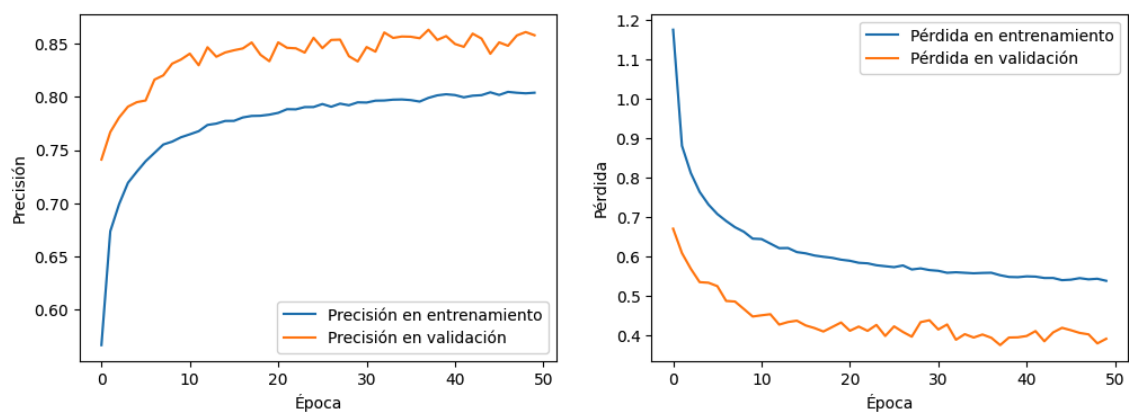


Figura 2: Gráficas de Precisión y Pérdida

Visualizamos la precisión y la pérdida tanto en el conjunto de entrenamiento como en el conjunto de validación a lo largo de las épocas. La Figura 2 muestra estas gráficas. Podemos observar que la precisión en el conjunto de validación se estabiliza alrededor del 85 %, mientras que la pérdida disminuye de manera constante.

3.7. Análisis de Predicciones

Realizamos predicciones sobre el conjunto de prueba y visualizamos algunos ejemplos junto con sus etiquetas verdaderas y predichas. La Figura 3 muestra varios ejemplos de imágenes de prueba, donde se pueden ver las predicciones correctas y las incorrectas (coloreadas en rojo). Esto facilita la interpretación de cómo el modelo está clasificando las imágenes de prueba y permite identificar errores comunes en las predicciones.



Figura 3: Ejemplos de Predicciones

4. Resultados

El modelo entrenado alcanzó una precisión del 85.88 % en el conjunto de prueba. La precisión en el conjunto de entrenamiento se estabilizó alrededor del 80.55 %, lo que indica que el modelo generaliza bien a datos no vistos. Las gráficas de precisión y pérdida muestran una tendencia de mejora constante a lo largo de las épocas, sin signos evidentes de sobreajuste.

5. Conclusiones

En este estudio, desarrollamos un modelo de red neuronal convolucional para clasificar imágenes del dataset Fashion MNIST. Usando técnicas de aumento de datos y regularización, construimos un modelo que funciona bien con datos nuevos, logrando una precisión del 85.88 % en las pruebas. Este trabajo muestra lo efectivas que son las CNNs para clasificar imágenes y la importancia del preprocesamiento de datos y la regularización para mejorar el rendimiento del modelo.