

Universidad Iberoamericana Puebla

*Agentes inteligentes*



*Fashion MNIST*

Estudiante:

Fernando Jose Figueroa Olivo 3103133

Jueves 19 de Septiembre, 2024

# Arquitectura 1

## Parámetros:

Epocas	Learning rate	Optimizer	Numero de capas
15	0.01	Adam	3

## Descripción:

La arquitectura utilizada es una red neuronal feedforward de tipo Sequential. La red consta de tres capas principales:

### 1. Capa de entrada:

- La capa de entrada recibe imágenes de tamaño 28x28 píxeles y las aplana utilizando la capa Flatten, lo que transforma la imagen en un vector unidimensional de 784 valores ( $28 \times 28 = 784$ ).

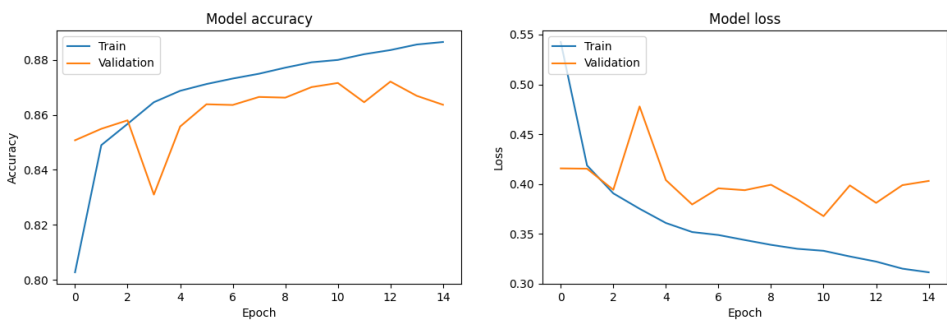
### 2. Capas ocultas:

- Se utilizaron dos capas densas (fully connected), con la función de activación ReLU. Estas capas ayudan a capturar las características de las imágenes a diferentes niveles de abstracción:

### 3, Capa de salida:

- La capa de salida es una capa completamente conectada con 10 neuronas, que corresponde a las 10 clases del problema de clasificación. Utiliza la función de activación Softmax, que convierte las salidas de la red en una probabilidad para cada clase.

## Gráficas comparativas:



Test accuracy: **0.84710** – Test loss: **0.45237**

# Arquitectura 2

## Parámetros:

Epocas	Learning rate	Optimizer	Numero de capas
45	0.01	Adam	6

## Descripción:

La arquitectura utilizada es una red neuronal feedforward de tipo Sequential. La red consta de cinco capas principales:

### 1. Capa de entrada:

- La capa de entrada recibe imágenes de tamaño 28x28 píxeles y las aplana utilizando la capa Flatten. Esto transforma cada imagen en un vector unidimensional de 784 valores ( $28 \times 28 = 784$ ), que luego es procesado por las siguientes capas densas.

### 2. Capas ocultas:

#### Primera capa oculta:

- Número de neuronas: 256
- Activación: ReLU

#### Segunda capa oculta:

- Número de neuronas: 128
- Activación: ReLU

#### Tercera capa oculta:

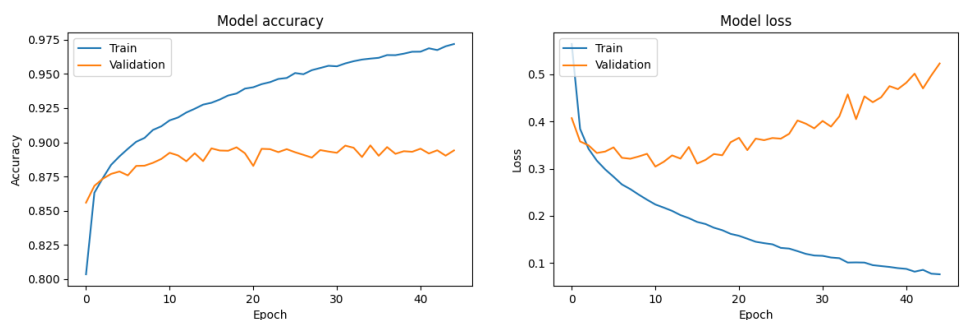
- Número de neuronas: 64
- Activación: ReLU

#### Cuarta capa oculta:

- Número de neuronas: 32
- Activación: ReLU.

3. Capa de salida: es una capa completamente conectada con 10 neuronas, que corresponden a las 10 clases del problema de clasificación. Utiliza la función de activación Softmax, que convierte las salidas de la red en probabilidades, permitiendo así que la clase con la mayor probabilidad sea seleccionada como la predicción final.

## Gráficas comparativas:



Test accuracy: **0.88719** – Test loss: **0.53084**

# Arquitectura 3

## Parámetros:

Epocas	Learning rate	Optimizer	Numero de capas
20	0.001	Adam	5

## Descripción:

La arquitectura utilizada es una red neuronal convolucional (CNN) de tipo Sequential. La red consta de varias capas especializadas diseñadas para capturar las características espaciales de las imágenes mediante el uso de convoluciones y pooling.

### 1. Capa de entrada:

- La capa de entrada recibe imágenes de tamaño 28x28 píxeles en escala de grises (canal único), y esta información se pasa a las capas convolucionales para el procesamiento.

### 2. Capas convolucionales y de pooling:

#### Primera capa convolucional:

- Filtros: 32 filtros con un tamaño de 3x3.
- Activación: ReLU.

#### Primera capa de MaxPooling:

- Tamaño del pool: 2x2.

#### Segunda capa convolucional:

- Filtros: 64 filtros con un tamaño de 3x3.
- Activación: ReLU.

#### Segunda capa de MaxPooling:

- Tamaño del pool: 2x2.

### 3. Capa de aplanado (Flatten):

- Después de las capas convolucionales y de pooling, la red utiliza una capa Flatten para convertir la salida bidimensional en un vector unidimensional, el cual puede ser procesado por las capas densas.

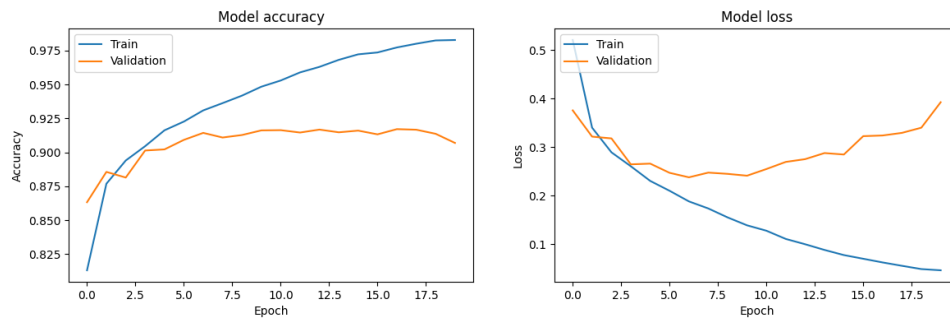
### 4. Capas densas:

- Una vez aplanados los datos, se pasa por una capa densa con 128 neuronas y activación ReLU. Esta capa ayuda a capturar las características finales antes de pasar a la clasificación.

### 5. Capa de salida:

- La capa de salida es una capa completamente conectada con 10 neuronas, que corresponde a las 10 clases del problema de clasificación. Utiliza la función de activación Softmax.

## Gráficas comparativas:



Test accuracy: **0.90380** – Test loss: **0.42776**

## **Resumen**

---

1. Número de capas y neuronas: Afectan la capacidad del modelo para aprender características complejas. Más capas y más neuronas permiten capturar patrones más profundos, sin embargo no representa necesariamente una ventaja, ya que también aumentan el riesgo de sobreajuste y los requisitos de hardware y software.

2. Tasa de aprendizaje: Controla la velocidad con la que el modelo ajusta los pesos.

3. Optimizador: Define cómo se actualizan los pesos. Adam converge rápidamente y se adapta bien a muchos problemas, mientras que SGD es más estable pero puede requerir más tiempo de ajuste.

4. Número de épocas: Más épocas permiten que el modelo aprenda mejor, pero demasiadas pueden causar sobreajuste, mientras que pocas pueden llevar al subajuste.

5. Tamaño del lote: Afecta la estabilidad y la velocidad del entrenamiento. Tamaños pequeños permiten actualizaciones más rápidas, mientras que tamaños grandes proporcionan estabilidad, pero requieren más memoria y recursos.

## **Discusión:**

---

No existe una “mejor” arquitectura única para el Fashion MNIST, ya que el rendimiento óptimo depende de varios factores, como el poder computacional disponible, el tiempo de entrenamiento y la capacidad de generalización que busques.

Sin embargo, una de las arquitecturas más efectivas y ampliamente utilizadas para Fashion MNIST es una Convolutional Neural Network (CNN). Este tipo de red neuronal es adecuada para problemas de clasificación de imágenes debido a su capacidad para capturar relaciones espaciales en los datos.