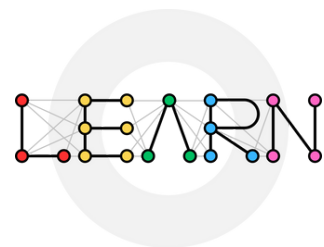


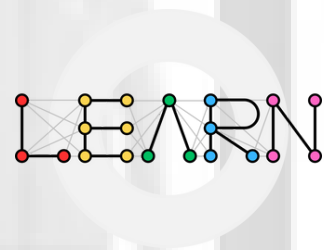
# COMPARAÇÃO

## MLP vs. CNN



# ÍNDICE

- Dataset utilizado e suas variações
- Arquiteturas de MLP
- Arquiteturas de CNN
- Resultados
- Comparações



# FASHION MNIST

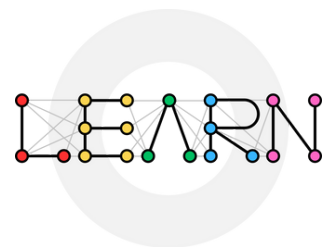
## Categorias:

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

## Quantidade:

- 60.000 - Treino
- 10.000 - Teste
- 7.000 / Categoria





# OS DADOS

## **DATASET A:**

Normal, sem nenhuma transformação.

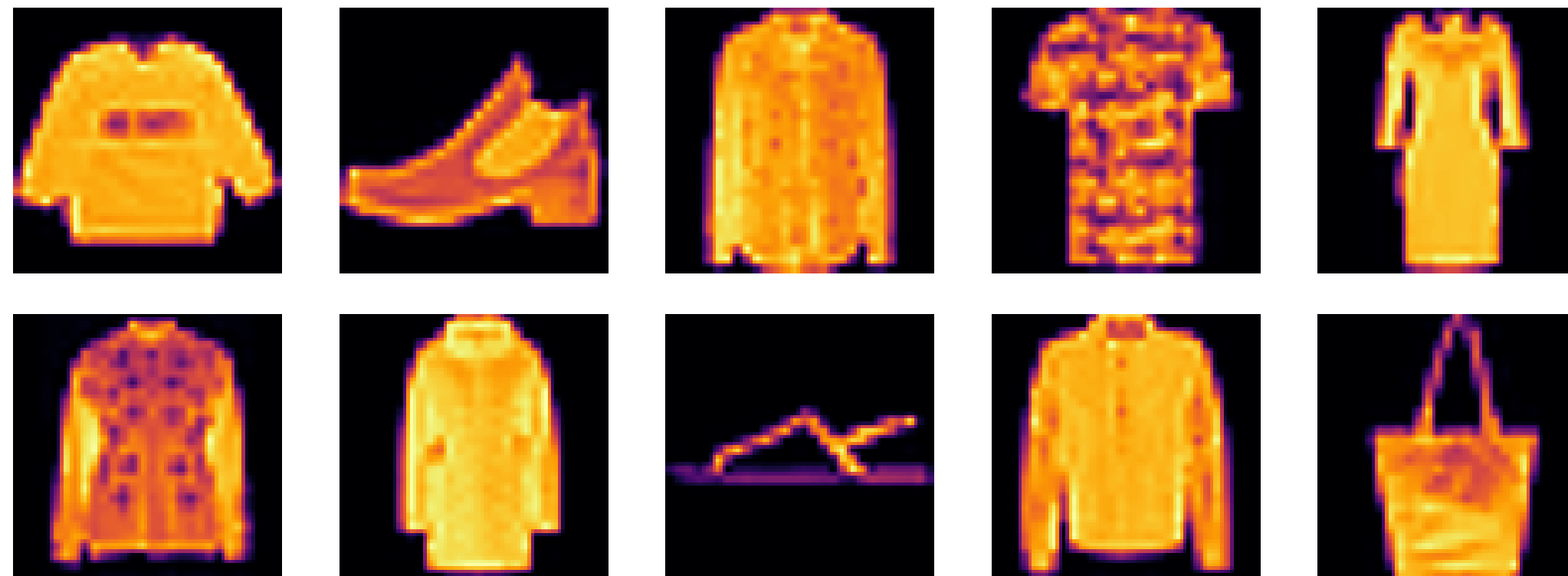
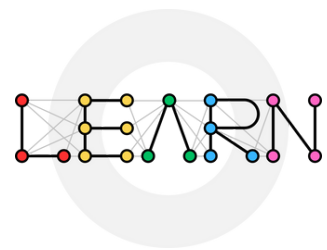


Figura 1 - Saída da função `plot_10_imagens(treino_loader_a)`.



# OS DADOS

## DATASET B:

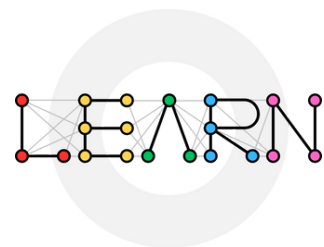
Transformação de transladação fixa  
aplicada numa matriz de 56 x 56.



Figura 2.1 - Saída da função  
`plot_10_imagens(treino_loader_b)` primeira chamada.



Figura 2.2 - Saída da função  
`plot_10_imagens(treino_loader_b)` segunda chamada.



# OS DADOS

## DATASET C:

Transformação de transladação aleatória a cada chamada aplicada numa matriz de 56 x 56.

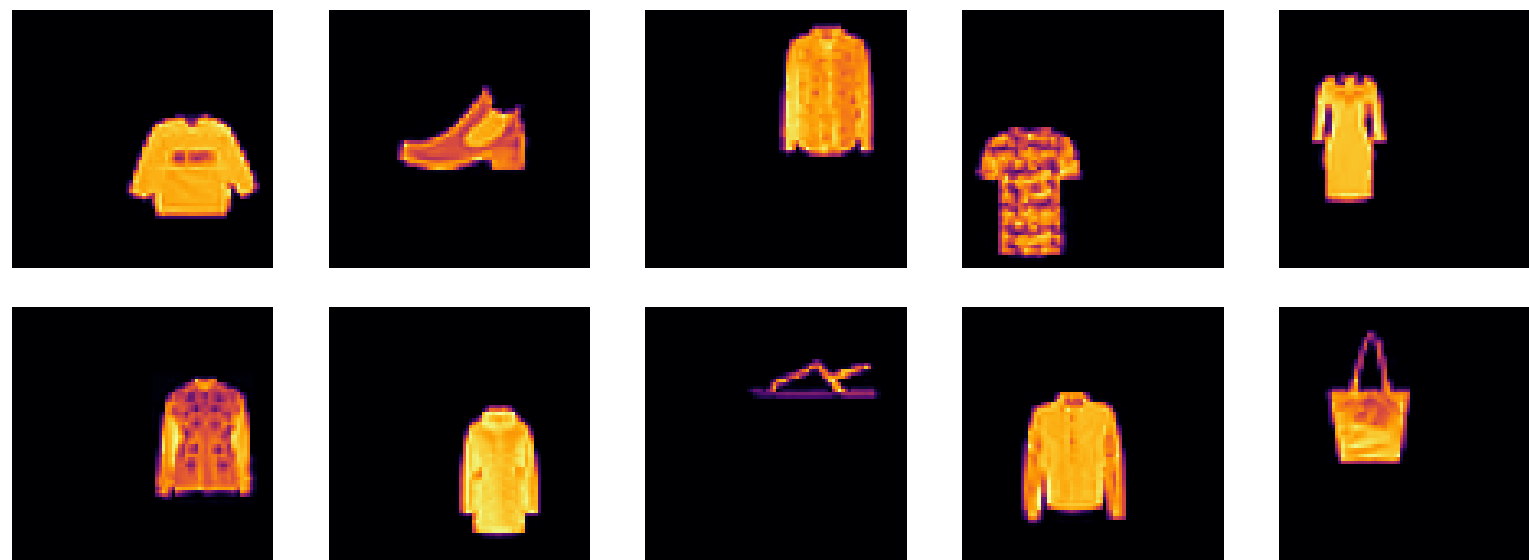


Figura 3.1 - Saída da função `plot_10_imagens(treino_loader_c)` primeira chamada.

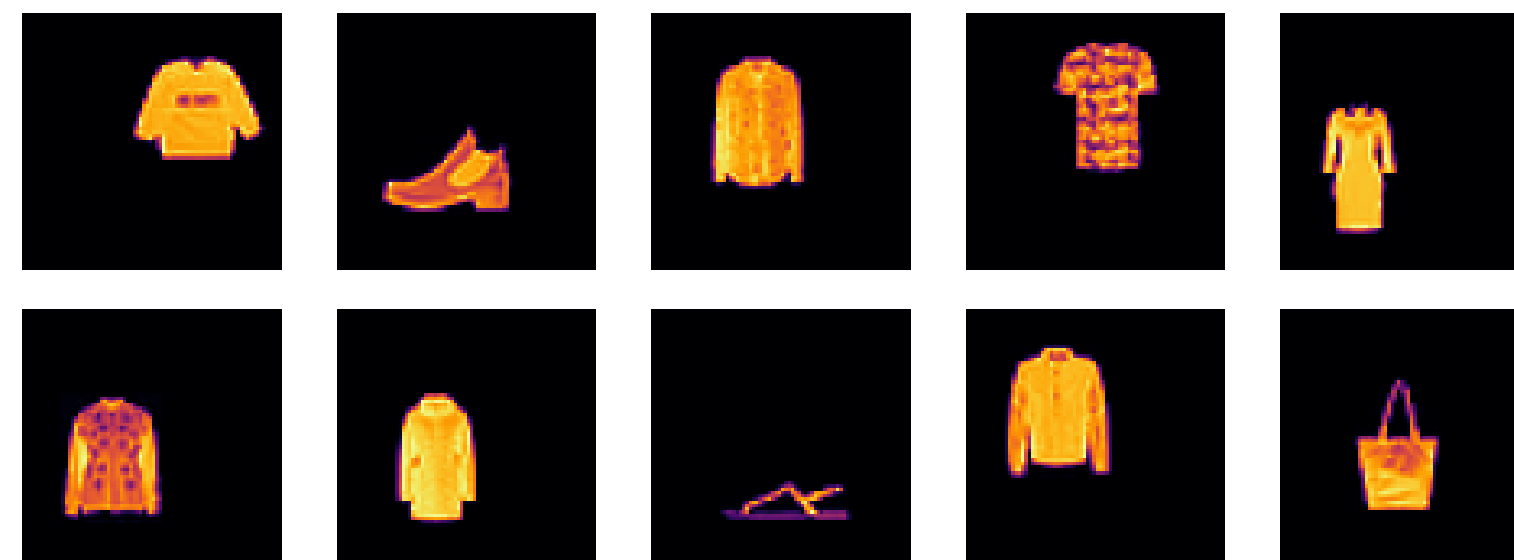
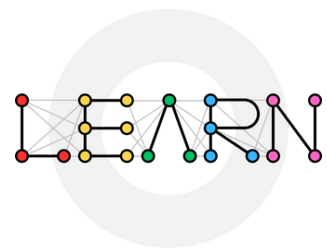
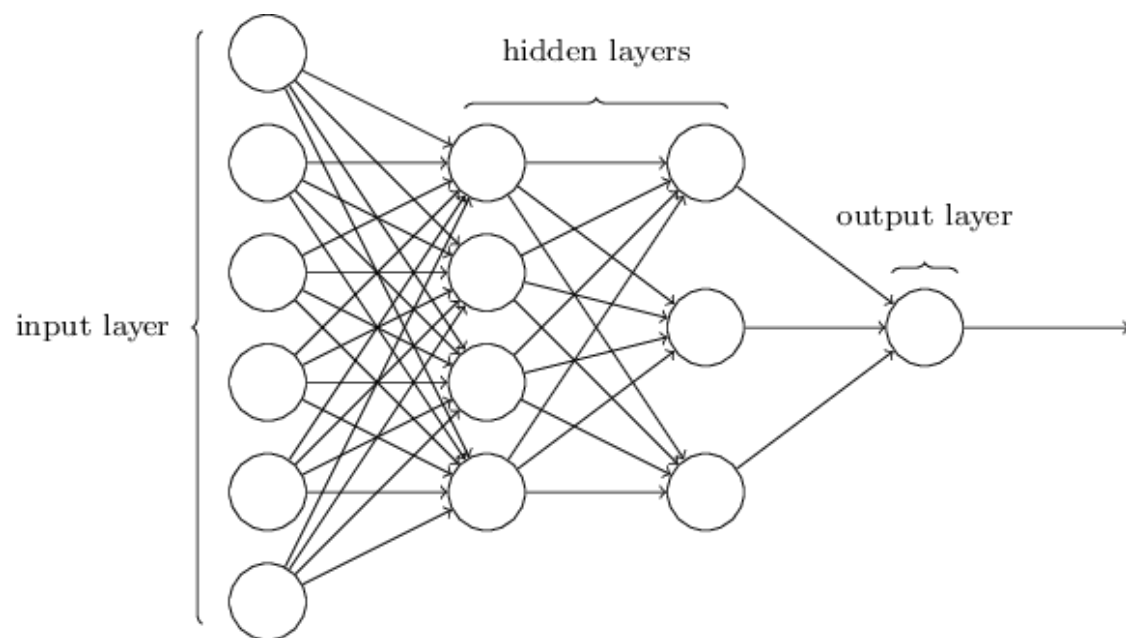


Figura 3.1 - Saída da função `plot_10_imagens(treino_loader_c)` segunda chamada.

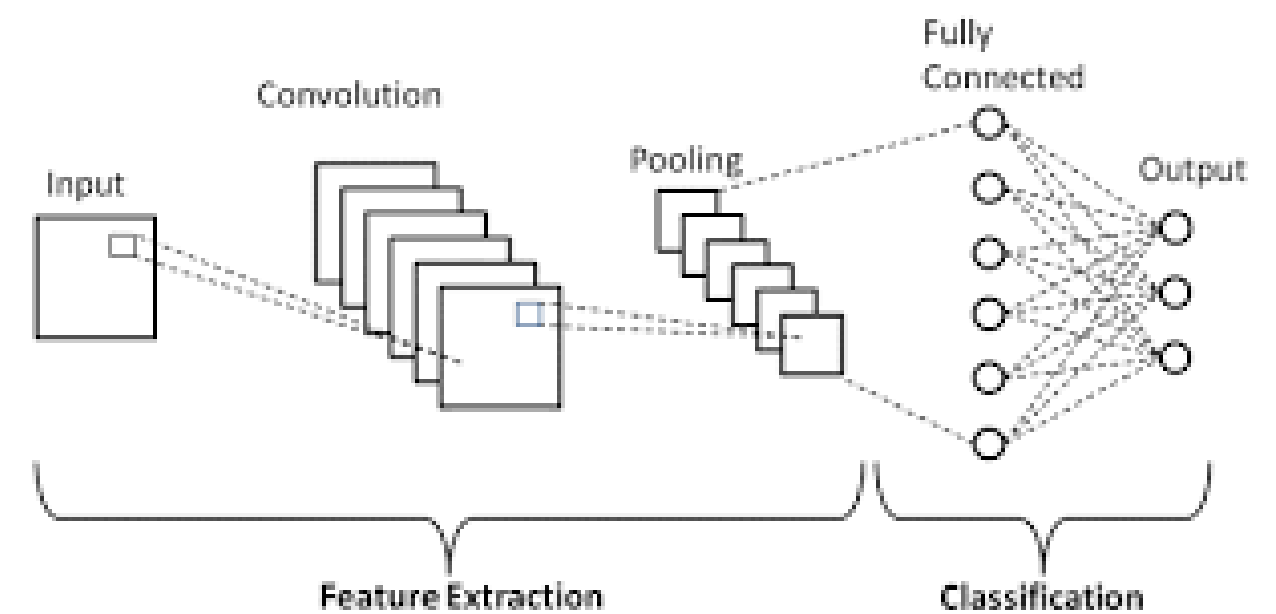


# OS MODELOS

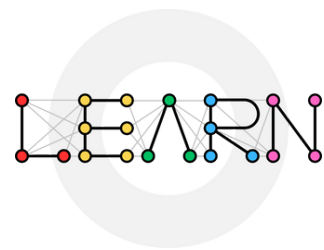
## MLP



## CNN







# MLP

## Multilayer Perceptron

### Modelo X

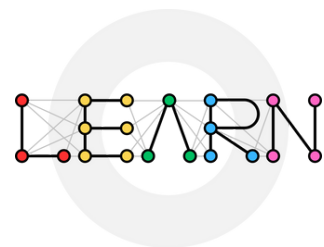
2 Camadas;  
• (tam\_entrada, 128);  
• (128, 64);  
• (64, tamanho\_saida);  
F. Ativação:  
• Relu()  
F. Perda:  
• CrossEntropy()

### Modelo Y

2 Camadas;  
• (tam\_entrada, 256);  
• (256, 128);  
• (128, 64);  
• (64, tamanho\_saida);  
F. Ativação:  
• Relu()  
F. Perda:  
• CrossEntropy()

### Modelo Z

2 Camadas;  
• (tam\_entrada, 512);  
• (512, 256);  
• (256, 128);  
• (128, 64);  
• (64, tamanho\_saida);  
F. Ativação:  
• Relu()  
F. Perda:  
• CrossEntropy()



# CNN

## Convolutional Neural Network

### Modelo 1 - 3C-3K-3M-2F

3 Camadas Convolucionais:

- (1, 32, 3, 1);
- (32, 64, 3, 1);
- (64, 128, 3, 1);

3 Max Pooling:

- (2, 2) - 3 vezes;

Fully Conected:

- (X, 128);
- (128, 10);

F. Ativação:

- Relu()

F. Perda:

- CrossEntropy()

### Modelo 2 - 3C-3K-2M-4F

3 Camadas Convolucionais:

- (1, 32, 3, 1);
- (32, 64, 3, 1);
- (64, 128, 3, 1);

3 Max Pooling:

- (2, 2) - 2 vezes;

Fully Conected:

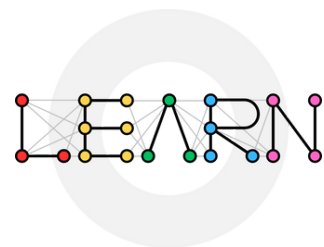
- (X, 512) -> (512, 256);
- (512, 256) -> (256, 128);
- (128, 10);

F. Ativação:

- Relu()

F. Perda:

- CrossEntropy()



# CNN

## Convolutional Neural Network

### Modelo 3 - 3C-53K-3M-2F

3 Camadas Convolucionais:

- (1, 32, 5, 2);
- (32, 64, 3, 1);
- (64, 128, 3, 1);

3 Max Pooling:

- (2, 2) - 3 vezes;

Fully Conected:

- (X, 128);
- (128, 10);

F. Ativação:

- Relu()

F. Perda:

- CrossEntropy()

### Modelo 4 - 3C-53K-2M-4F

3 Camadas Convolucionais:

- (1, 32, 5, 1);
- (32, 64, 3, 1);
- (64, 128, 3, 1);

3 Max Pooling:

- (2, 2) - 2 vezes;

Fully Conected:

- (X, 512) -> (512, 256);
- (512, 256) -> (256, 128);
- (128, 10);

F. Ativação:

- Relu()

F. Perda:

- CrossEntropy()