

Comparação RN e CNN

Desafio 3

Carlos Henrique Vieira Marques Veeck

chvmv@cin.ufpe.br

Professor: Cleber Zanchettin

Introdução

Para realizar a comparação entre o RN e o CNN, selecionei o Dataset da biblioteca Keras **“MNIST digits classification dataset”**. Selecionei esse dataset pela sua acessibilidade, sendo apenas necessário uma linha de código para acessar seu conteúdo.

Esse dataset nos disponibiliza 60.000 imagens 28x28, em grayscale, dos 10 dígitos para treino. Além disso, um set de validação com 10.000 imagens com as mesmas especificações está incluso.

```
import matplotlib.pyplot as plt
plt.figure(figsize=(40,40))

num_images = 20
for i in range(num_images):
    row = x_train[i]
    label = y_train[i]

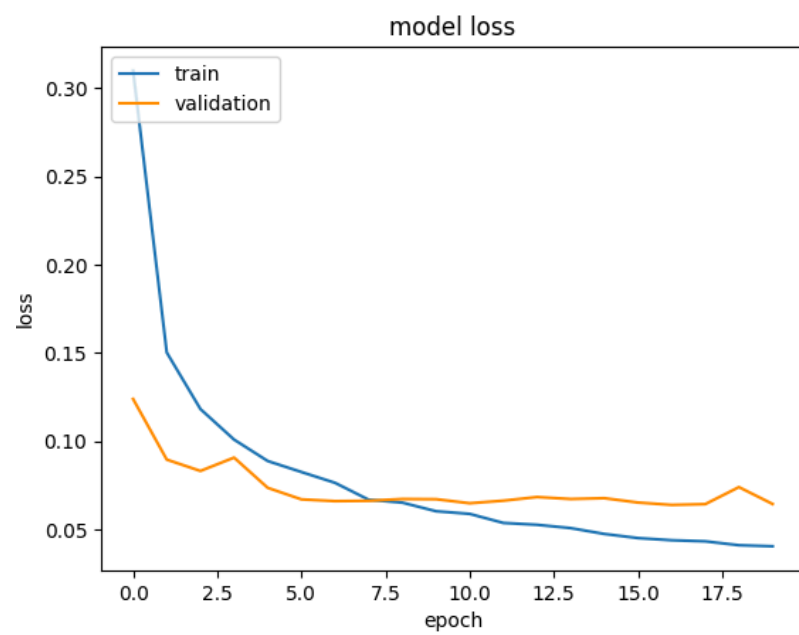
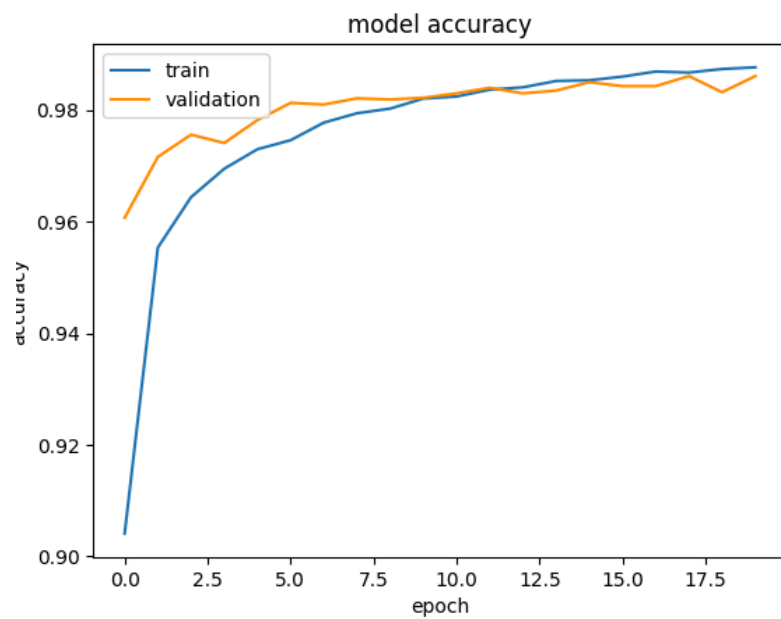
    image = row.reshape(28,28)
    plt.subplot(1, num_images, i+1)
    plt.axis('off')
    plt.imshow(image, cmap='gray')
```



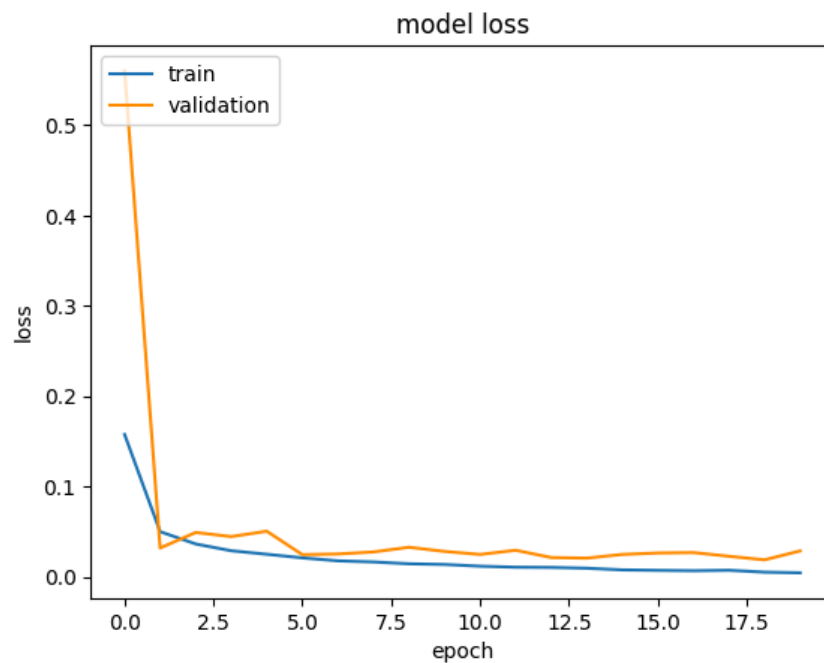
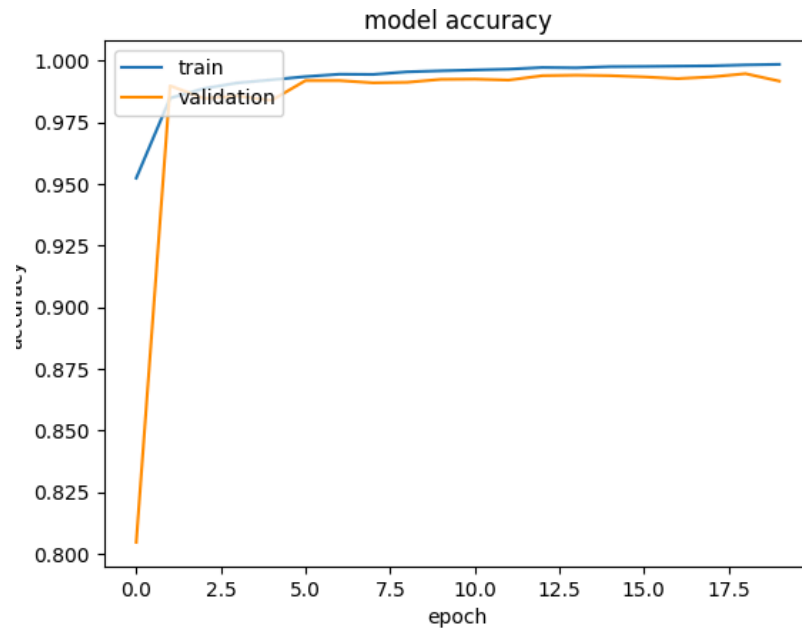
Comparação

Após construir e testar ambos os modelos requisitados, utilizei a biblioteca **matplotlib** para traçar gráficos de precisão e perda para facilitar a comparação entre ambos. A seguir, os resultados

RN:



CNN:



Podemos observar ao observar os gráficos dos dois modelos que o CNN apresentou resultados melhores que a RN convencional. Para a classificação de imagens, o modelo de CNN consegue desempenhar melhor. Muito disso se deve às **convolution layers** que adicionamos ao modelo, que permitem obter informações locais e mais específicas sobre

os pixels. A autonomia do deep learning nesse caso também se provou uma excelente adição ao modelo, em contraste com a abordagem de machine learning.

A única desvantagem que destaco da abordagem DL foi seu tempo de treinamento. Devido a problemas no meu COLAB, não consegui conectar com a GPU disponibilizada, fazendo com que as épocas levassem cerca de 2 a 3 minutos para serem processadas corretamente pela CPU.

Resultado final do treinamento:

RN:

```
loss: 0.0405 - accuracy: 0.9877 - val_loss: 0.0644 - val_accuracy: 0.9861
```

CNN:

```
loss: 0.0045 - accuracy: 0.9985 - val_loss: 0.0288 - val_accuracy: 0.9917
```