

---

---

# Lista 2

# Aprendizado Profundo

Edgard Henrique, André Hugo, Gustavo Costa

---

---

## Questão 1

- Objetivo Principal: Desenvolver uma rede neural convolucional (CNN) e uma rede neural de múltiplas camadas (MLP) para classificar imagens do dataset MNIST e compará-las.
  - Ferramentas utilizadas: Python, Tensorflow/Keras, numpy, matplotlib, scikit learn...
-

---

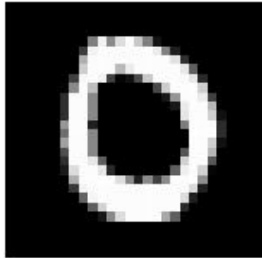
# Conhecendo o Dataset

- Dataset MNIST:
    - Contém 70.000 imagens tamanho 28x28 (60.000 de treino, 10.000 de teste).
    - 10 classes: Números de 0 a 9 escritos manualmente.
-

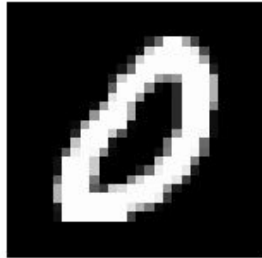
---

# Conhecendo o Dataset

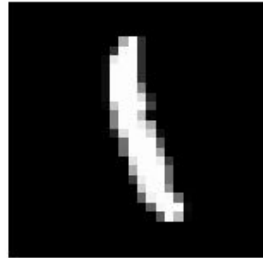
Label: 0



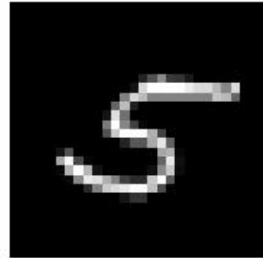
Label: 0



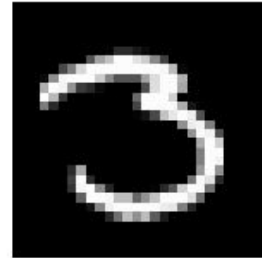
Label: 1



Label: 5



Label: 3



Label: 6



---

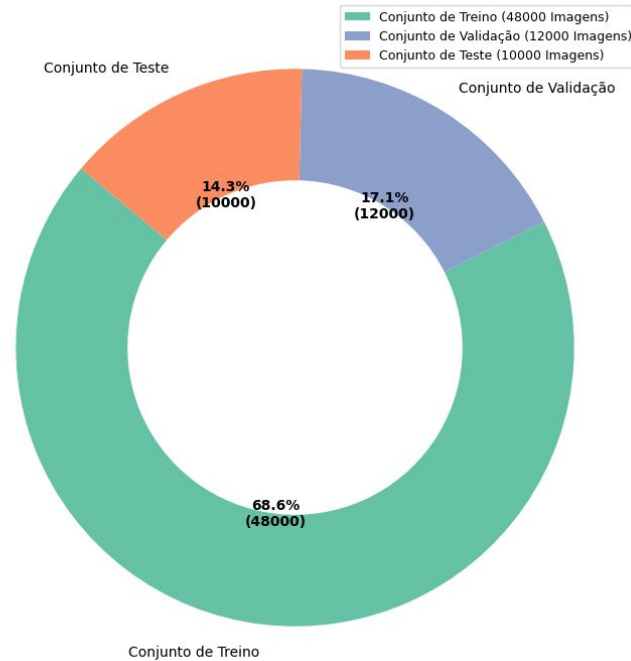
# Separação dos dados

- Contém 70.000 imagens (60.000 de treino, 10.000 de teste), porém 20% das imagens do conjunto de treino foram utilizadas para validação, o que resultou em um conjunto de dados composto por: 48.000 imagens de treino, 12.000 imagens de validação e 10.000 imagens de teste.
-

---

# Separação dos dados

Proporções de Treino/Teste/Validação



---

---

# Solução MLP

---

---

# Arquitetura da Rede MLP

- Camada de entrada tensor Flatten tamanho 28x28;
  - 1 Camada densa escondida com 512 neurônios e função de ativação Relu;
  - 1 Camada densa escondida com 256 neurônios e função de ativação Relu;
  - 1 Camada de saída densa com 10 neurônios (cada um referenciando uma classe) e função de ativação softmax;
-



---

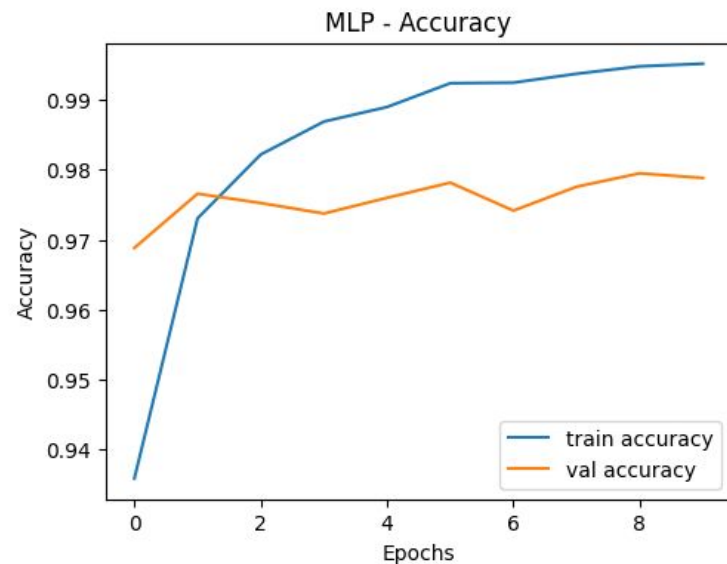
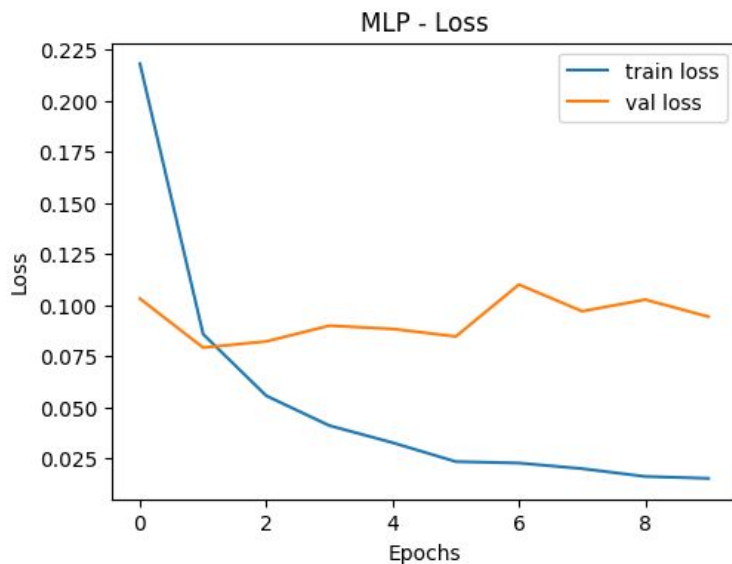
# Treinamento da Rede MLP

- Métrica: Acurácia
  - Métrica de erro: Categorical CrossEntropy;
  - Otimizador: ADAM;
  - Learning Rate: 0.001 (Padrão)
  - Número de épocas: 10
  - Tamanho do Lote: 64
-

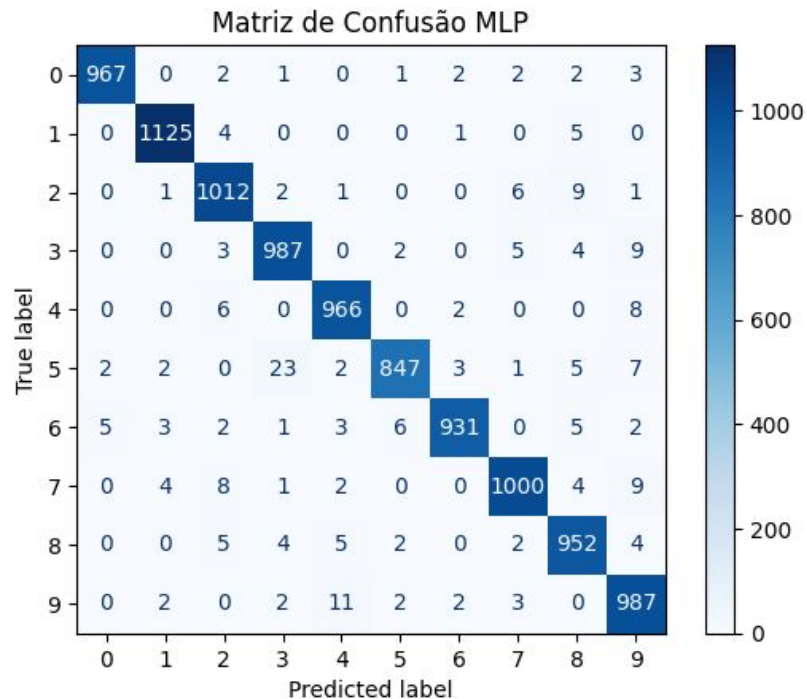
---

# Métricas atingidas pelo modelo MLP

- Acurácia: 0.9959
- Erro: 0.0125
- Acurácia(val.): 0.9788
- Erro(val.): 0.0943



# Matriz de confusão do modelo MLP



---

---

# Solução CNN

---

---

# Arquitetura da Rede CNN

- Entrada tamanho 28x28x1;
  - 1 Camada Convolutacional com 32 neurônios, máscara 3x3 e função de ativação Relu;
  - 1 Camada Convolutacional com 64 neurônios, máscara 3x3 e função de ativação Relu;
  - 1 Camada de Max Pooling com máscara 2x2;
  - 1 Camada para vetorizar os valores (Flatten);
  - 1 Camada densa escondida com 128 neurônios e função de ativação Relu;
  - 1 Camada de saída densa com 10 neurônios (cada um referenciando uma classe) e função de ativação softmax;
-

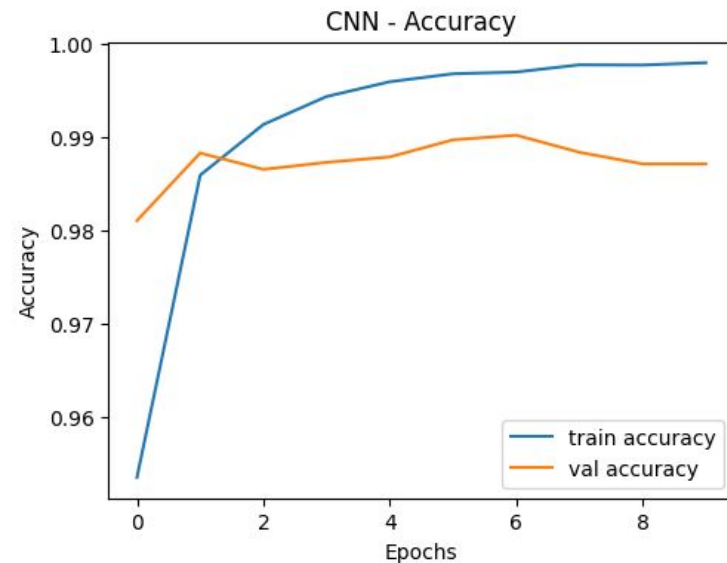
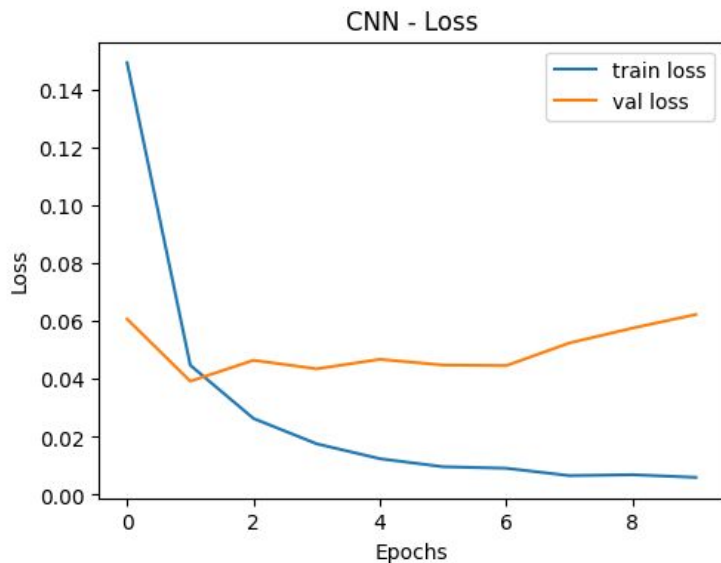
---

# Treinamento da Rede CNN

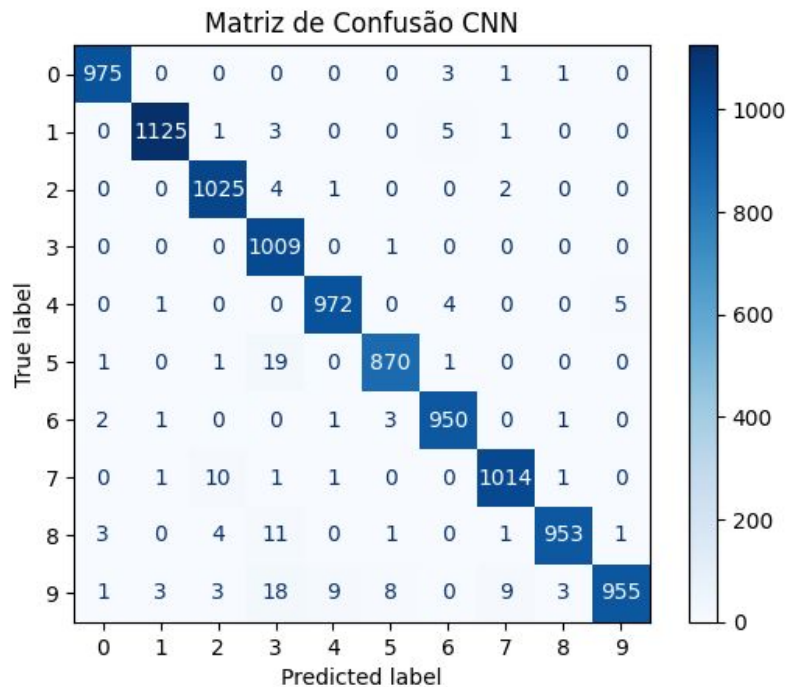
- Métrica: Acurácia
  - Métrica de erro: Categorical CrossEntropy;
  - Otimizador: ADAM;
  - Learning Rate: 0.001 (Padrão)
  - Número de épocas: 10
  - Tamanho do Lote: 64
-

# Métricas atingidas pelo modelo CNN

- Acurácia: 0.9984
- Erro: 0.0054
- Acurácia(val.): 0.9872
- Erro(val.): 0.0623



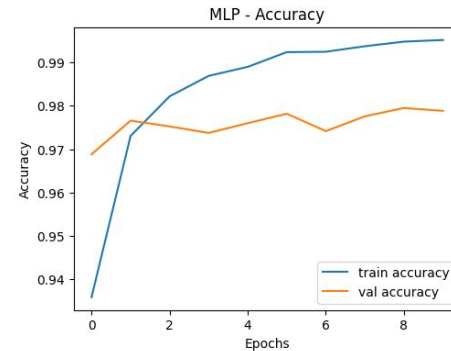
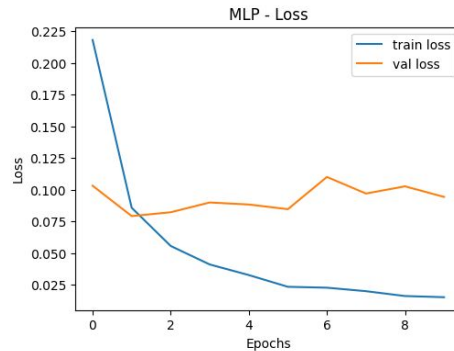
# Matriz de confusão do modelo CNN



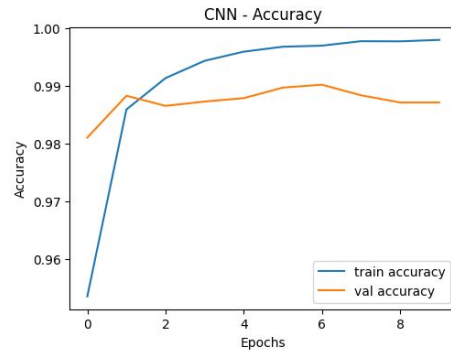
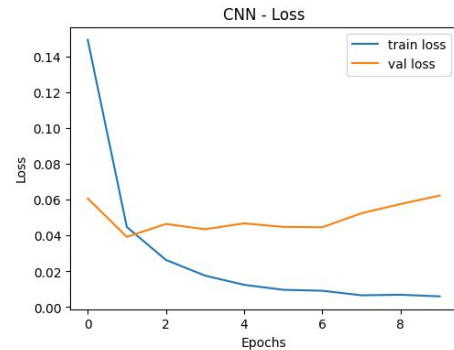


# Comparação de Métricas

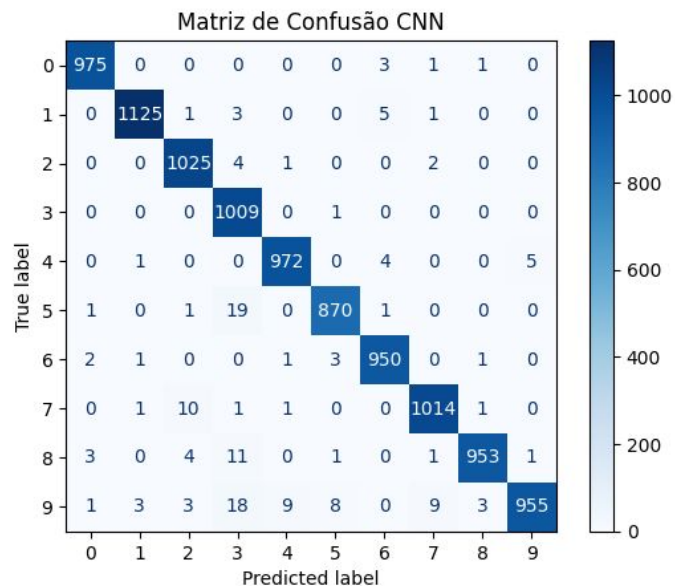
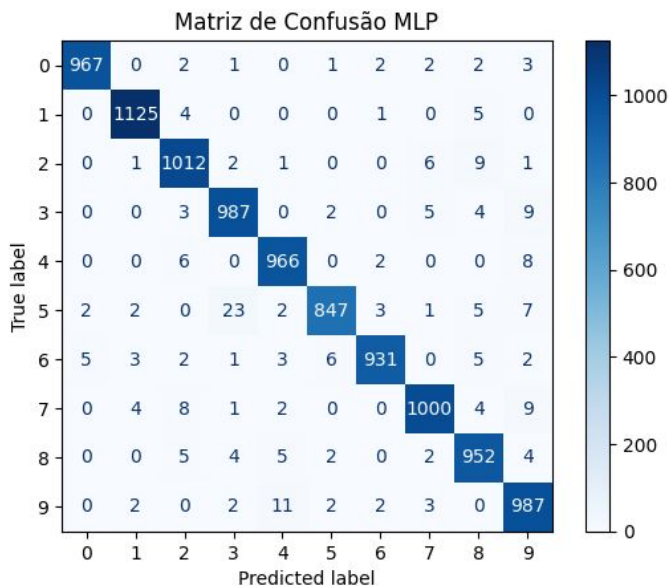
- Acurácia MLP: 0.9959
- Erro MLP: 0.0125
- Acurácia(val.) MLP: 0.9788
- Erro(val.) MLP: 0.0943



- Acurácia CNN: 0.9984
- Erro CNN: 0.0054
- Acurácia(val.) CNN: 0.9872
- Erro(val.) CNN: 0.0623



# Comparação da Matriz de Confusão



---

## Questão 2

- Objetivo Principal: Desenvolver uma rede neural convolucional (CNN) para classificar imagens do dataset CIFAR-10.
  - Dataset CIFAR-10:
    - Contém 60.000 imagens (50.000 de treino, 10.000 de teste).
    - 10 classes: avião, automóvel, pássaro, gato, veado, cachorro, sapo, cavalo, navio, caminhão.
-

---

# Pré-Processamento

- As imagens do CIFAR-10 são originalmente armazenadas como vetores unidimensionais de tamanho 3072 (pois  $32 \times 32 \text{ pixels} \times 3 \text{ canais de cor} = 3072$ ).
    - Imagens são convertidas para formato (3, 32, 32) e normalizadas para valores entre 0 e 1.
    - As imagens e os rótulos são convertidos para tensores PyTorch e preparados para a entrada no modelo
-

---

# Arquitetura CNN

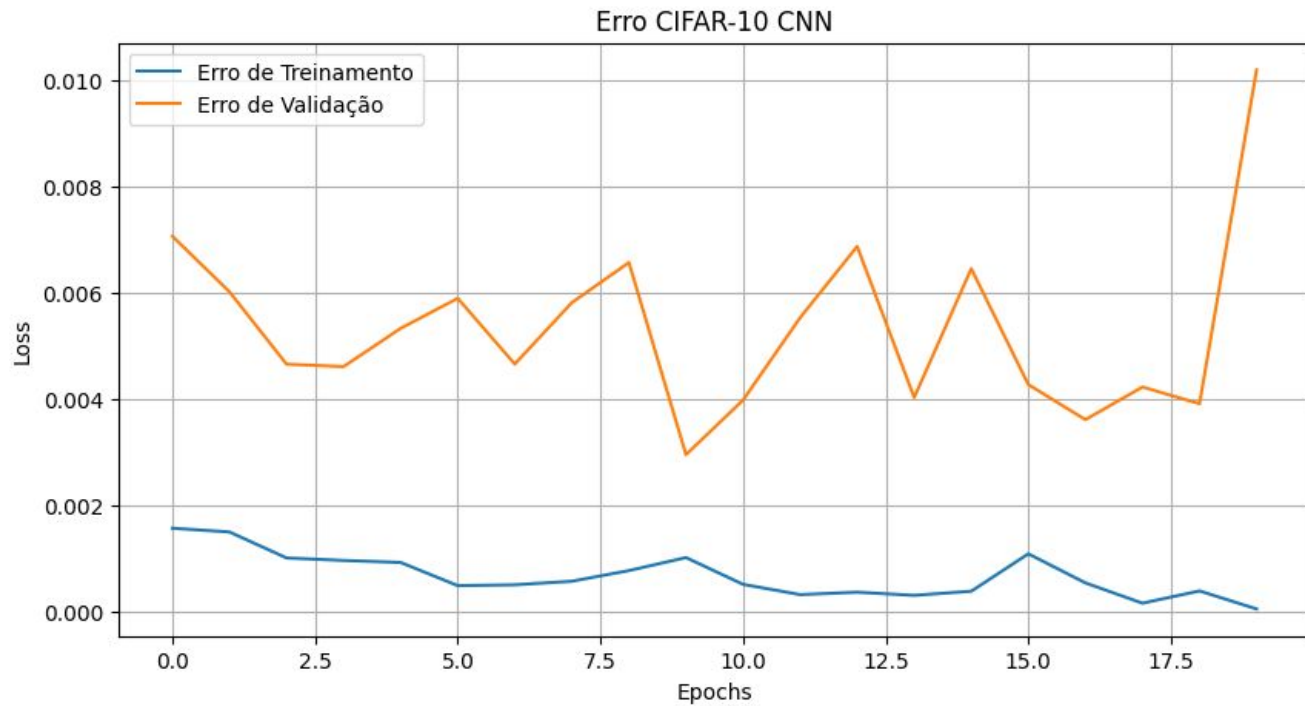
- **4 camadas convolucionais**
    - **Conv1 e Conv2:** Convoluções com 64 filtros, kernel 3x3.
    - **Conv3 e Conv4:** Convoluções com 128 filtros, kernel 3x3.
    - **Ativação:** Relu
  - **Pooling:** Max Pooling 2x2 após a 2ª e 4ª camada convolucional para reduzir dimensionalidade.
  - **Dropout:** Após **Conv2** e **Conv4**: Dropout de 25% para prevenir overfitting.
  - **Camadas Fully Connected (FC)**
    - **FC1:** 1024 neurônios, dropout de 50%, Relu
    - **FC2:** 512 neurônios, Relu
    - **FC3 (Saída):** 10 neurônios (um para cada classe do CIFAR-10)
-

---

# Treinamento e Validação

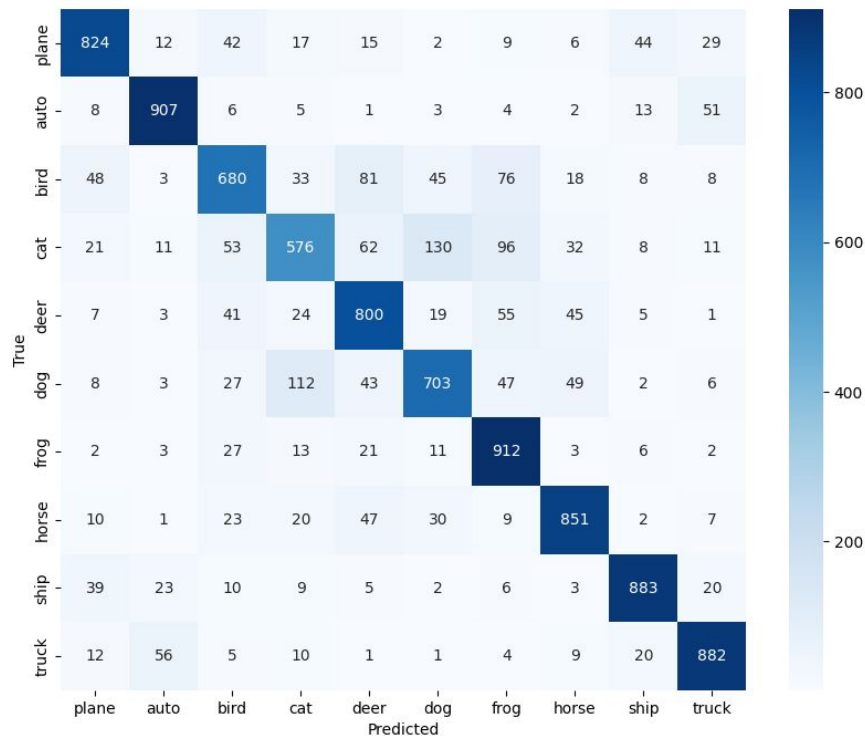
- **Função de Custo:**
    - Cross-Entropy Loss usada como função de perda.
  - **Otimização:**
    - Otimizador Adam, taxa de aprendizado: 0.001.
  - **Estratégia de Treinamento:**
    - Treinamento por 20 épocas com batch size de 64.
-

# Treinamento



# Resultados

- Acurácia Treino: 92%
- Acurácia Teste: 80%
- Matriz de confusão:





---

## Questão 3 - Objetivo Principal

- construir e treinar uma Rede Neural Recorrente (RNN), como LSTM ou GRU, para realizar análise de sentimentos em um conjunto de resenhas de filmes da base de dados IMDb. A tarefa consiste em classificar as resenhas como positivas (1) ou negativas (0) com base no texto.
-

---

## Questão 3 - preparação do dados

```
'A wonderful little production. <br /><br />The filming technique is very unassuming- very old-time-BBC fashion and gives a comforting, and sometimes discomforting, sense of realism to the entire piece. <br /><br />The actors are extremely well chosen- Michael Sheen not only "has got all the polari" but he has all the voices down pat too! You can truly see the seamless editing guided by the references to Williams\' diary entries, not only is it well worth the watching but it is a terrifically written and performed piece. A masterful production about one of the great master\'s of comedy and his life. <br /><br />The realism really comes home with the little things: the fantasy of the guard which, rather than use the traditional \'dream\' techniques remains solid then disappears. It plays on our knowledge and our senses, particularly with the scenes concerning Orton and Halliwell and the sets (particularly of their flat with Halliwell\'s murals decorating every surface) are terribly well done...'
```

- Remoção de tags HTML para limpar o texto.
  - Conversão para letras minúsculas para padronizar o texto.
  - Remoção de pontuações, números e caracteres especiais
  - Remoção de caracteres únicos (como palavras de uma letra só).
  - Remoção de espaços extras que possam ter sido introduzidos.
  - Remoção opcional de stopwords, se ativado no código.
  - (As stopwords são palavras que aparecem com frequência no idioma, mas não fornecem informações úteis para a tarefa de análise de sentimentos. Palavras como "the", "is", "and")
-

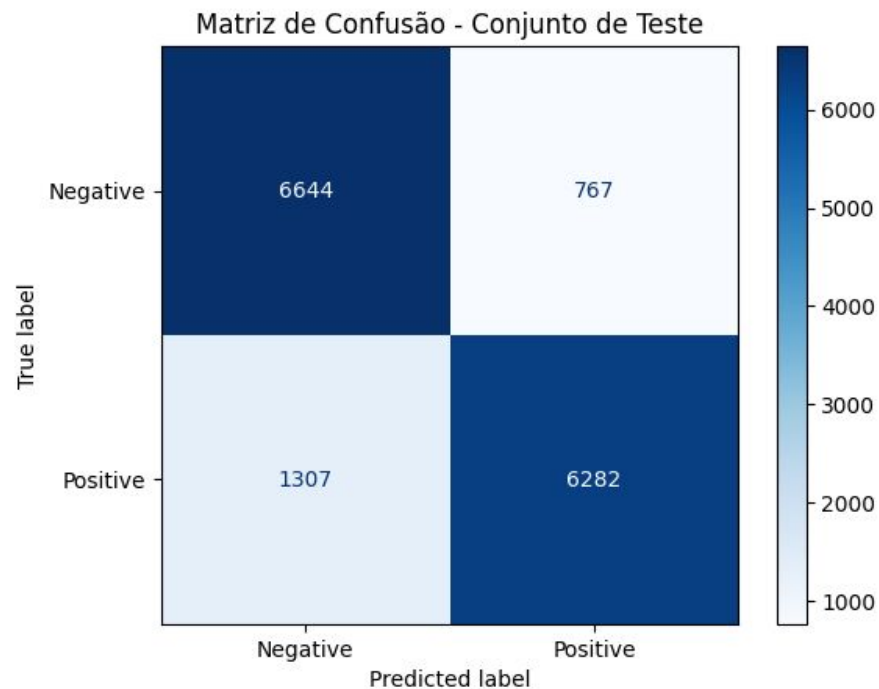
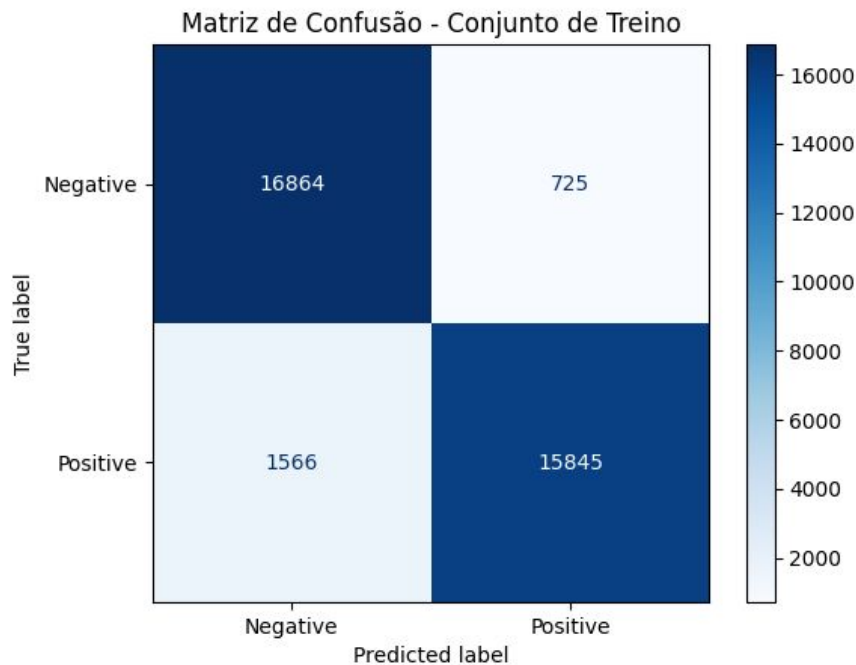
---

## Questão 3 - Criação da rede neural

- **Arquitetura com 2 camadas ocultas**
  - **Primeira Camada Oculta:** LSTM, camada com 128 neurônios, função de ativação Relu e dropout com uma taxa de 30%
  - **Segunda Camada Oculta:** LSTM, camada com 64 neurônios, Relu e dropout
  - **Camada de Saída:** Densa com 1 neurônio: A camada final possui 1 neurônio com ativação sigmoid
  - **dropout\_rate** = 0.3: Taxa de dropout usada para reduzir o overfitting.
  - **taxa de aprendizado de** : 0.001
  - **Função de Perda:** binary\_crossentropy
  - **Early Stopping:** Implementado para parar o treinamento se a perda de validação não melhorar por 2 épocas consecutivas
  - **Número de épocas** = 10
  - **Otimização com Adam e regularização L2.**
-

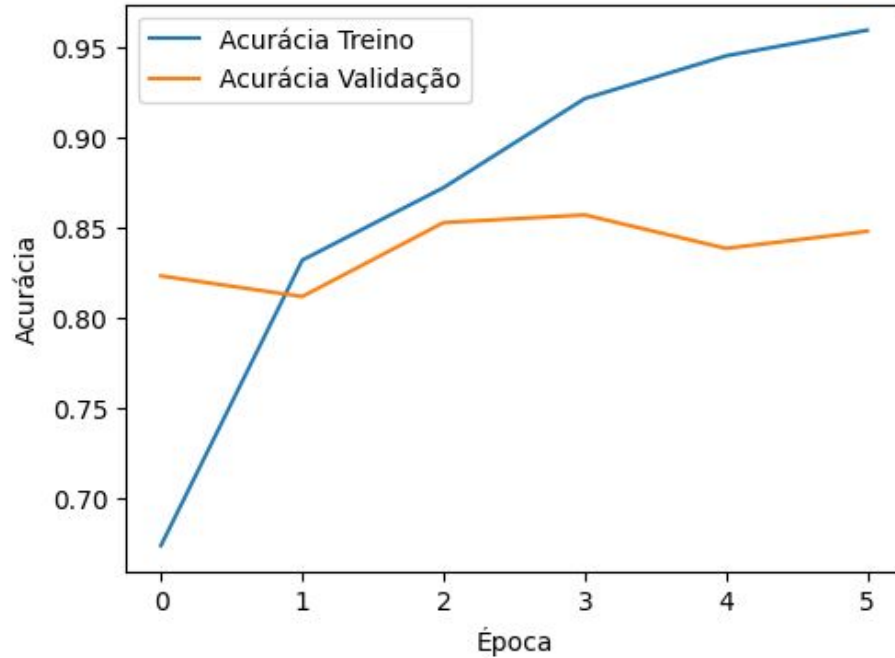
# Questão 3 - Resultados

Acurácia de : 86,17%

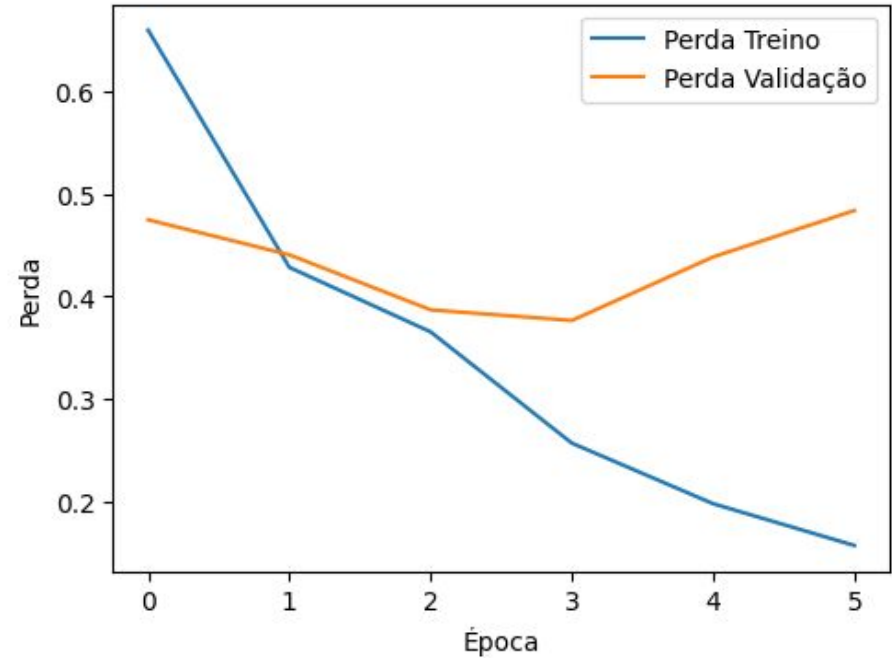


# Questão 3 - Resultados

Acurácia durante o treinamento



Perda durante o treinamento



---

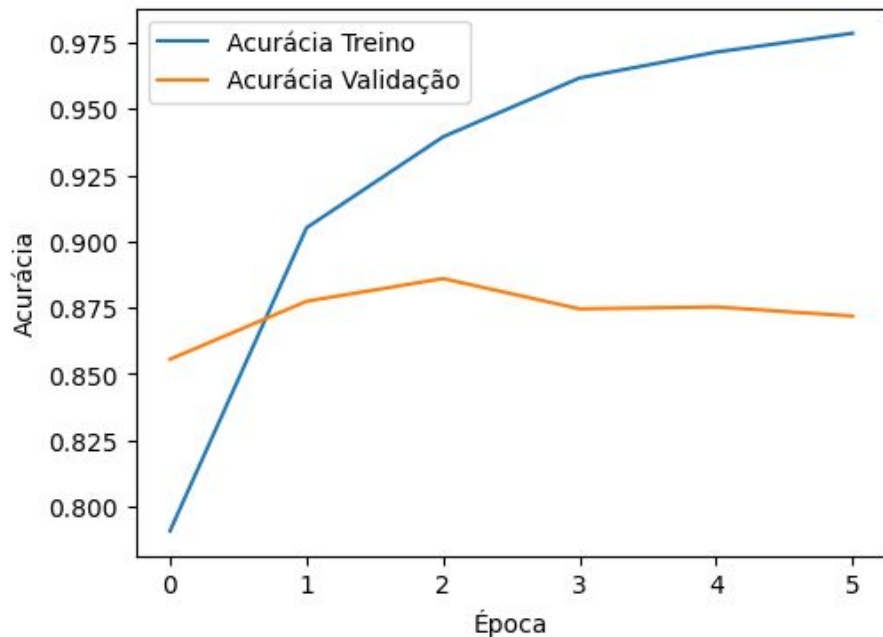
## Questão 3 - Criação da rede neural

- **Arquitetura com 2 camadas ocultas**
  - **Primeira Camada Oculta:** GRU, camada com 128 neurônios, função de ativação Tanh e dropout com uma taxa de 30%
  - **Segunda Camada Oculta:** GRU, camada com 64 neurônios, Tanh e dropout
  - **Camada de Saída:** Densa com 1 neurônio: A camada final possui 1 neurônio com ativação sigmoid
  - **dropout\_rate** = 0.3: Taxa de dropout usada para reduzir o overfitting.
  - **taxa de aprendizado de** : 0.001
  - **Função de Perda:** binary\_crossentropy
  - **Early Stopping:** Implementado para parar o treinamento se a perda de validação não melhorar por 2 épocas consecutivas
  - **Número de épocas** = 10
  - **Otimização com Adam e regularização L2.**
-

# Questão 3 - Resultados

Acurácia de : 88,31%

Acurácia durante o treinamento - GRU com tanh



Perda durante o treinamento - GRU com tanh

