



Instituto Tecnológico de Aeronáutica - ITA  
CT-213 - Inteligência Artificial aplicada à Robótica Móvel  
Aluno: Ulisses Lopes da Silva

## Relatório do Laboratório 8 - Redes Neurais Convolucionais

---

### 1 Breve Explicação em Alto Nível da Implementação

No presente laboratório, foi desenvolvida uma rede neural convolucional baseada na arquitetura clássica LeNet-5, utilizando a biblioteca Keras com backend em TensorFlow. O objetivo consistiu na implementação de um modelo capaz de realizar classificação de imagens, semelhante à realizada com o conhecido dataset MNIST, mantendo, contudo, a estrutura original da rede, projetada para entrada de imagens no formato 32x32 com um canal (tons de cinza).

A estrutura da rede segue o padrão da LeNet-5: uma camada convolucional de entrada é seguida por duas camadas de convolução intercaladas com camadas de pooling (neste caso, de média, ao invés de max pooling), finalizando com duas camadas densas, sendo a última dotada de ativação softmax para classificação em múltiplas classes. As funções de ativação escolhidas nas camadas convolucionais e na penúltima camada densa foram tangente hiperbólica (tanh), de acordo com a proposta original do LeNet. De modo geral, a implementação seguiu coerentemente a seção 6 do arquivo Jupyter, que continha dicas para a sintaxe do código.

O treinamento foi conduzido com o uso da ferramenta ImageDataGenerator, que facilita o carregamento e a organização dos dados em lotes. Para a etapa de otimização, foi empregado o algoritmo Adam, enquanto a função de perda utilizada foi a entropia cruzada categórica, apropriada para tarefas de classificação com múltiplas classes. Durante o treinamento, os registros de desempenho foram acompanhados em tempo real por meio da ferramenta TensorBoard. O *dataset* foi previamente dividido em partes para treinamento e validação, de modo a treinar com melhor eficiência a rede neural, além de utilizar a técnica de *cross-validation*.

De forma geral, a implementação apresentou resultados compatíveis com o esperado para o tipo de problema proposto. Como esperado, a rede acertou boa parte das figuras, com alguns erros em figuras mais difíceis de identificar. Todavia, apesar das *mispredictions*, o desempenho foi satisfatório.

## 2 Figuras Comprovando Funcionamento do Código

### 2.1 Evolução do Treinamento no TensorBoard

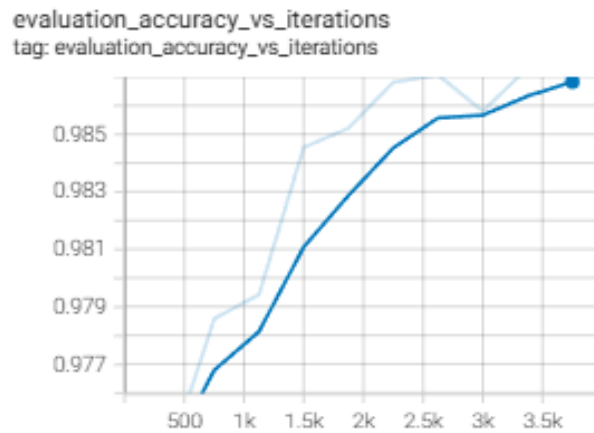


Fig. 1: Evolução da acurácia de validação ao longo das iterações.

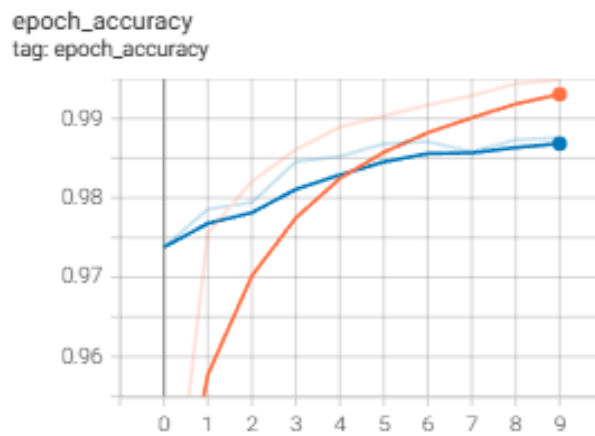


Fig. 2: Acurácia por época durante o treinamento e validação.

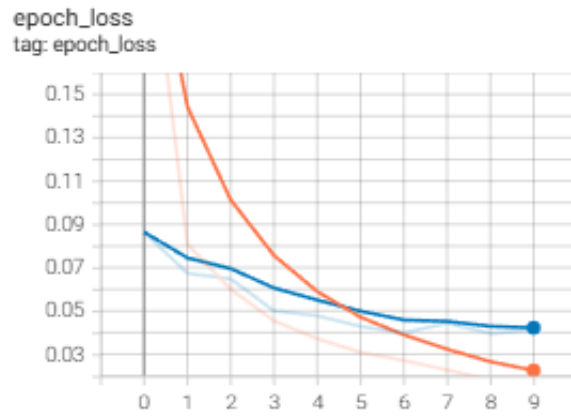


Fig. 3: Perda por época para os dados de treinamento e validação.

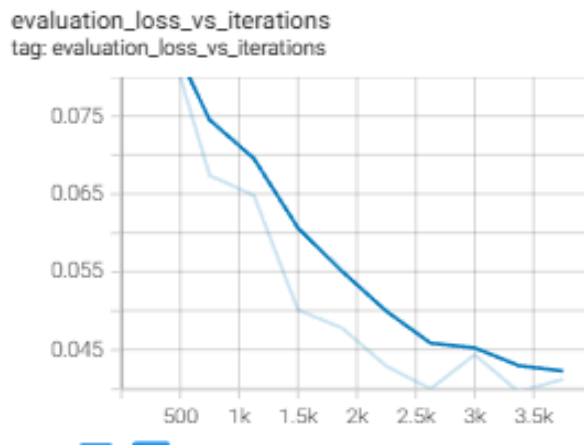


Fig. 4: Evolução da função de perda de validação ao longo das iterações.

## 2.2 Avaliação da LeNet-5

Example: 193. Expected Label: 9. Predicted Label: 8.

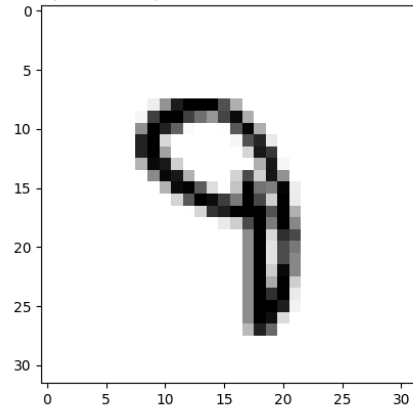


Fig. 5: Exemplo 193: Rótulo esperado 9, rótulo previsto 8 (erro)

Example: 247. Expected Label: 4. Predicted Label: 6.

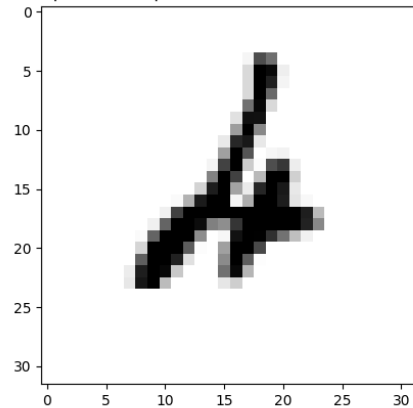


Fig. 6: Exemplo 247: Rótulo esperado 4, rótulo previsto 6 (erro).

Example: 321. Expected Label: 2. Predicted Label: 7.

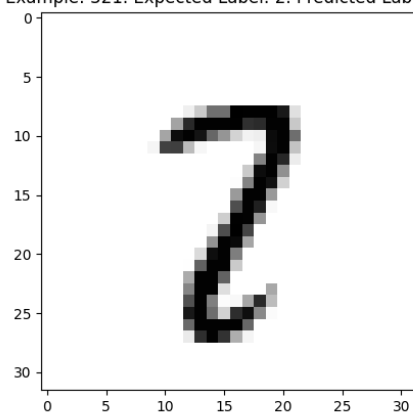


Fig. 7: Exemplo 321: Rótulo esperado 2, rótulo previsto 7 (erro).

Example: 340. Expected Label: 5. Predicted Label: 3.

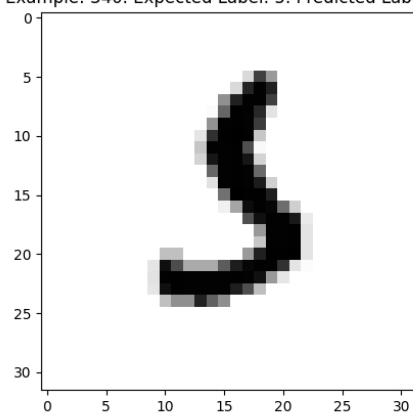


Fig. 8: Exemplo 340: Rótulo esperado 5, rótulo previsto 3 (erro).

Example: 445. Expected Label: 6. Predicted Label: 0.

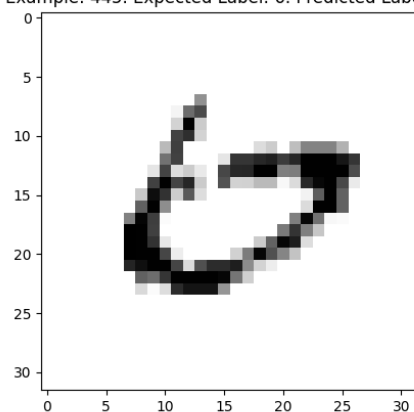


Fig. 9: Exemplo 445: Rótulo esperado 6, rótulo previsto 0 (**erro**).

Example: 1021. Expected Label: 7. Predicted Label: 7.

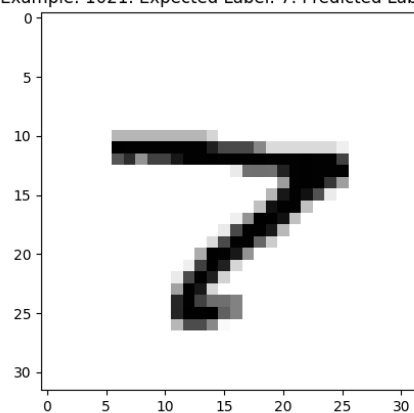


Fig. 10: Exemplo 1021: Rótulo esperado 7, rótulo previsto 7 (**acerto**).

Example: 2168. Expected Label: 8. Predicted Label: 8.

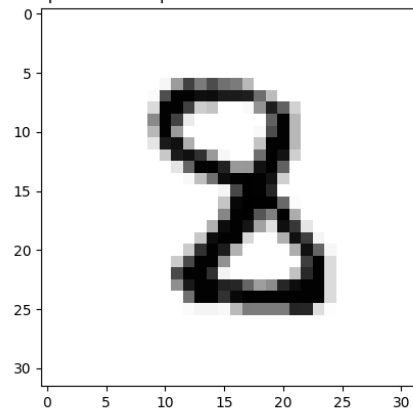


Fig. 11: Exemplo 2168: Rótulo esperado 8, rótulo previsto 8 ([acerto](#)).

Example: 5071. Expected Label: 7. Predicted Label: 7.

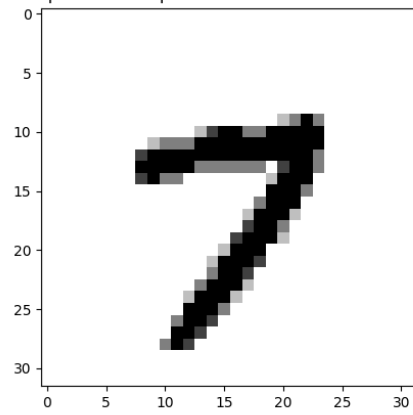


Fig. 12: Exemplo 5071: Rótulo esperado 7, rótulo previsto 7 ([acerto](#)).

Example: 5703. Expected Label: 3. Predicted Label: 3.

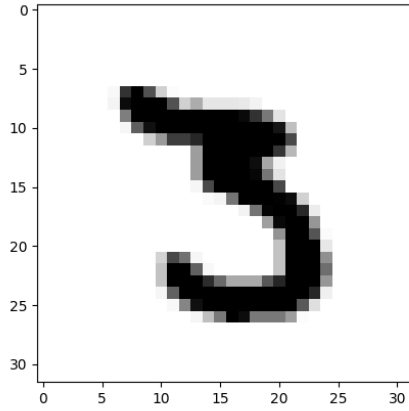


Fig. 13: Exemplo 5703: Rótulo esperado 3, rótulo previsto 3 (acerto).

Example: 8694. Expected Label: 3. Predicted Label: 3.

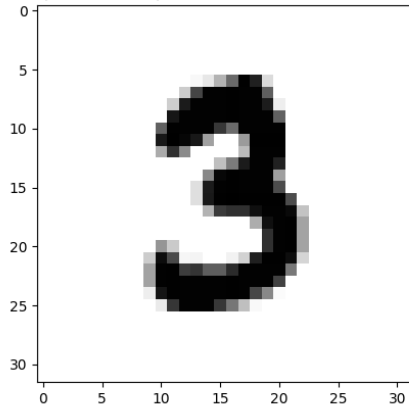


Fig. 14: Exemplo 8694: Rótulo esperado 3, rótulo previsto 3 (acerto).

### 3 Discussão dos Resultados

#### Análise dos Resultados mostrados no *TensorBoard*

A análise dos gráficos obtidos no *TensorBoard* revela que o modelo treinado com a arquitetura LeNet-5 apresentou um comportamento de convergência estável e eficaz tanto em termos de acurácia quanto de perda.

O gráfico `evaluation_accuracy_vs_iterations` mostra um crescimento progressivo da acurácia durante as iterações de treinamento, indicando que o modelo foi capaz de aprender padrões consistentes ao longo do tempo. A curva atinge valores superiores a 98,5%, o que demonstra um bom desempenho já nas primeiras milhares de iterações.

No gráfico `epoch_accuracy`, a acurácia por época é mostrada tanto para os dados de treino (curva laranja) quanto para os dados de validação (curva azul). Observa-se que ambas as curvas



aumentam de forma consistente e se aproximam bastante, o que é um sinal de que o modelo não sofreu *overfitting* significativo. A acurácia de validação estabiliza próximo de 99%, enquanto a de treino ultrapassa esse valor — comportamento esperado quando o modelo é bem ajustado, mas ainda mantém boa capacidade de generalização.

O gráfico `epoch_loss` complementa essa análise ao exibir a diminuição da função de perda (*loss*) durante o treinamento. A queda rápida no início, seguida por uma desaceleração progressiva, é típica de um treinamento saudável. A diferença entre as curvas de treino e validação permanece pequena ao longo das épocas.

Por fim, o gráfico `evaluation_loss_vs_iterations` mostra a perda calculada durante a avaliação ao longo das iterações. A curva descendente reflete uma melhoria contínua do desempenho do modelo, com valores de perda caindo para abaixo de 0,045, o que está em consonância com os demais gráficos e confirma a boa capacidade de ajuste do modelo.

No geral, os gráficos indicam que o modelo foi treinado de forma eficiente, sem sinais evidentes de *overfitting* ou *underfitting*, apresentando ótima acurácia e baixa perda tanto em treino quanto em validação, evidenciando que a arquitetura, os hiperparâmetros escolhidos e o volume de dados foram adequados para o problema proposto.

## Comentários a respeito da classificação das imagens pela rede

A análise das imagens classificadas incorretamente e corretamente na Seção 2.2 permite avaliar qualitativamente o comportamento da rede em situações ambíguas ou de fácil reconhecimento. Foram selecionados alguns exemplos de erro e acerto da rede neural, os quais seguem abaixo.

As imagens com erro de classificação mostram casos compreensíveis do ponto de vista visual. O **exemplo 193**, por exemplo, representa um dígito “9”, mas sua curvatura superior é sutil e fechada, podendo se confundir com um “8” imperfeito, que foi justamente a predição da rede. O **exemplo 247**, rotulado como “4”, apresenta uma inclinação e traço vertical longos, o que pode ter levado à classificação incorreta como “6”. Situação semelhante ocorre no **exemplo 321**, onde o dígito “2” está bem desenhado, mas pode ter sido confundido com um “7” devido à ausência de uma base inferior mais marcada. O **exemplo 340**, rotulado como “5”, possui formato curvo e conexão superior que lembra o traçado de um “3”. Já no **exemplo 445**, o dígito “6” apresenta um traço aberto e circular que se assemelha ao “0”, erro comum até mesmo entre seres humanos.

Por outro lado, os exemplos corretamente classificados demonstram a robustez da rede em reconhecer padrões bem definidos, mesmo com pequenas variações de escrita. O **exemplo 1021**, rotulado e predito como “7”, possui traços retos característicos. Os exemplos **2168** e **5703**, ambos com rótulo e predição correta como “8” e “3” respectivamente, apresentam estruturas simétricas e centradas. O **exemplo 5071** mostra um “7” com traço superior claro, e o **exemplo 8694** exibe um “3” bem definido e equilibrado.

De maneira geral, observa-se que os erros ocorreram em dígitos com traços ambíguos ou pouco padronizados, enquanto os acertos envolveram grafias mais regulares e bem delimitadas. Esses resultados demonstram que a rede é eficaz na generalização, mas pode ser sensível a variações atípicas no formato dos dígitos manuscritos.