Instituto Tecnológico de Aeronáutica - ITA
Inteligência Artificial para Robótica Móvel - CT-213
Aluno: Vinícius José de Menezes Pereira

Relatório do Laboratório 8 - Redes Neurais Convolucionais

1. Breve Explicação em Alto Nível da Implementação

A implementação da rede neural LeNet-5 foi feita com o auxílio do framework Keras do TensorFlow. Criou-se uma rede neural sequencial para treinar os dados, com camadas criadas segundo a Tabela 1 do notebook. Assim, temos as seguintes camadas:

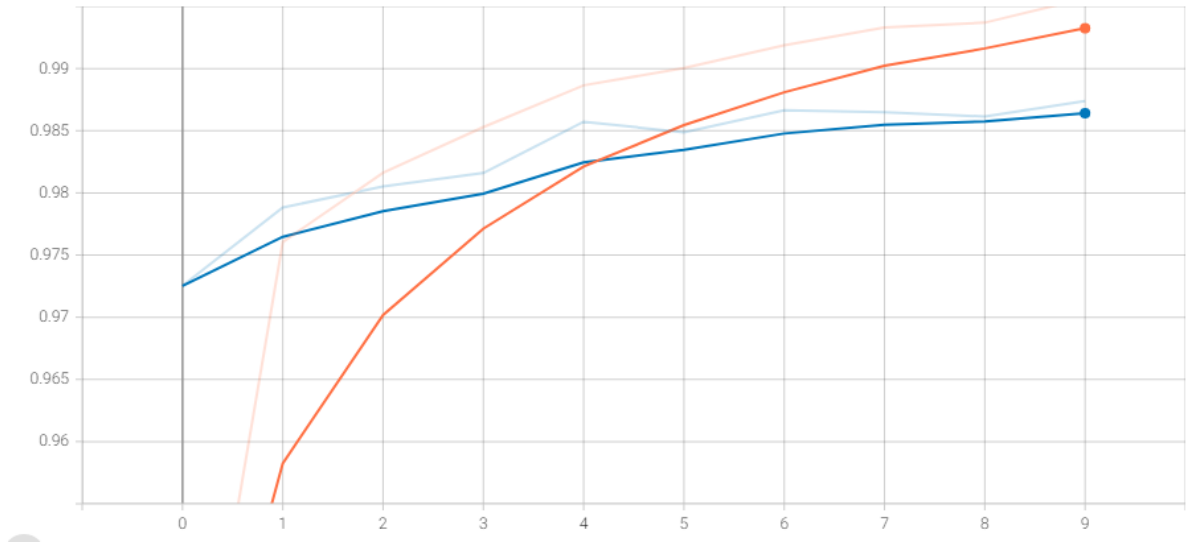
- Convolução com tanh
- Average Pooling
- Convolução com tanh
- Average Polling
- Convolução com tanh e muitos filtros
- Completamente conectada com tanh e 84 neurônios
- Completamente conectada com softmax e 10 neurônios

Sabemos que em CNN's (Convolutional Neural Network) há a primeira etapa de detecção de blobs e arestas seguida da detecção de textura, seguida da detecção de partes de objeto e por fim a determinação da classe do objeto, em geral. Nesse caso, as primeiras 4 camadas da rede neural são responsáveis justamente pelas etapas iniciais da detecção de aresta e textura. A camada de convolução com muitos filtros deve ser responsável pela detecção de objetos e as últimas duas camadas completamente conectadas devem ser responsáveis pela classificação dos objetos. Essas últimas camadas que são completamente conectadas só podem ser chamadas no fim da rede neural, pois demandam uma alta força computacional e não podem ser aplicadas quando os dados não estão devidamente filtrados. Em especial, a função de ativação da última camada é responsável por suavizar os resultados correntemente encontrados pelo treino para que possa se encaixar nos rótulos.

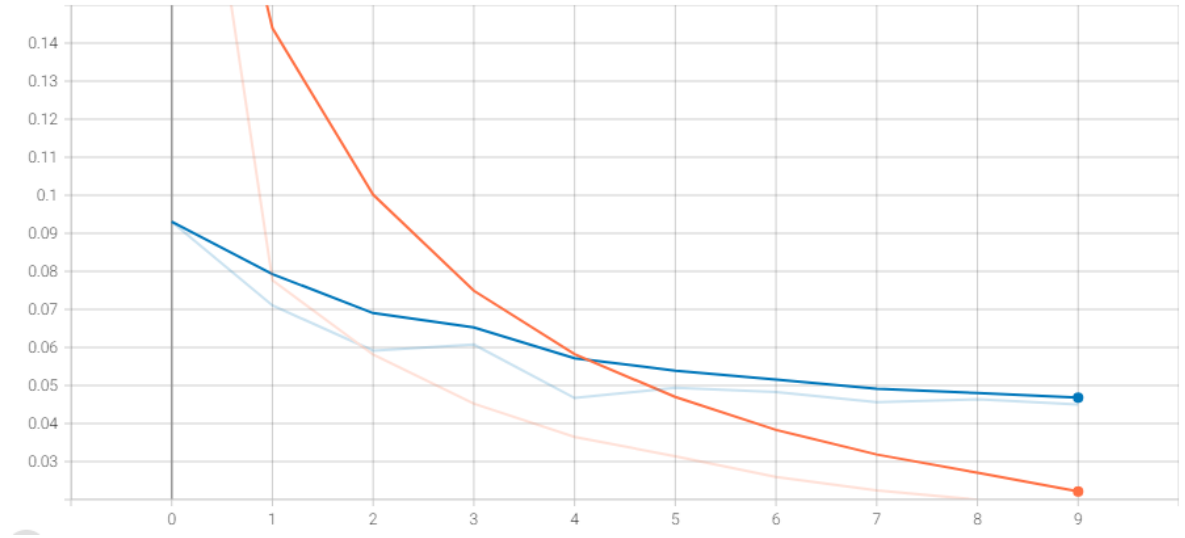
2. Figuras Comprovando Funcionamento do Código

2.1. Evolução do Treinamento no TensorBoard

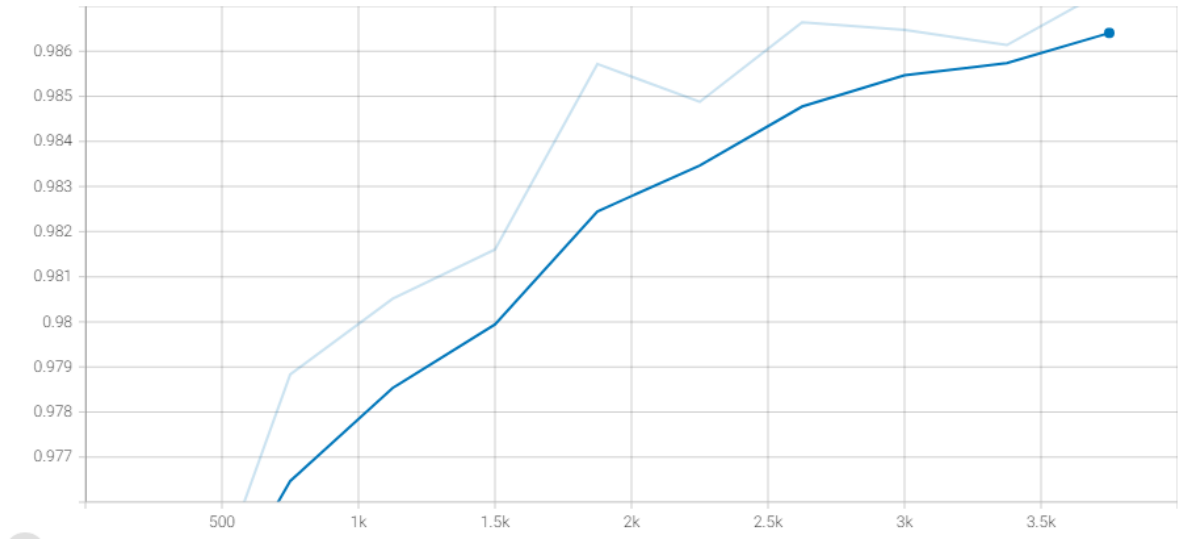
epoch_accuracy

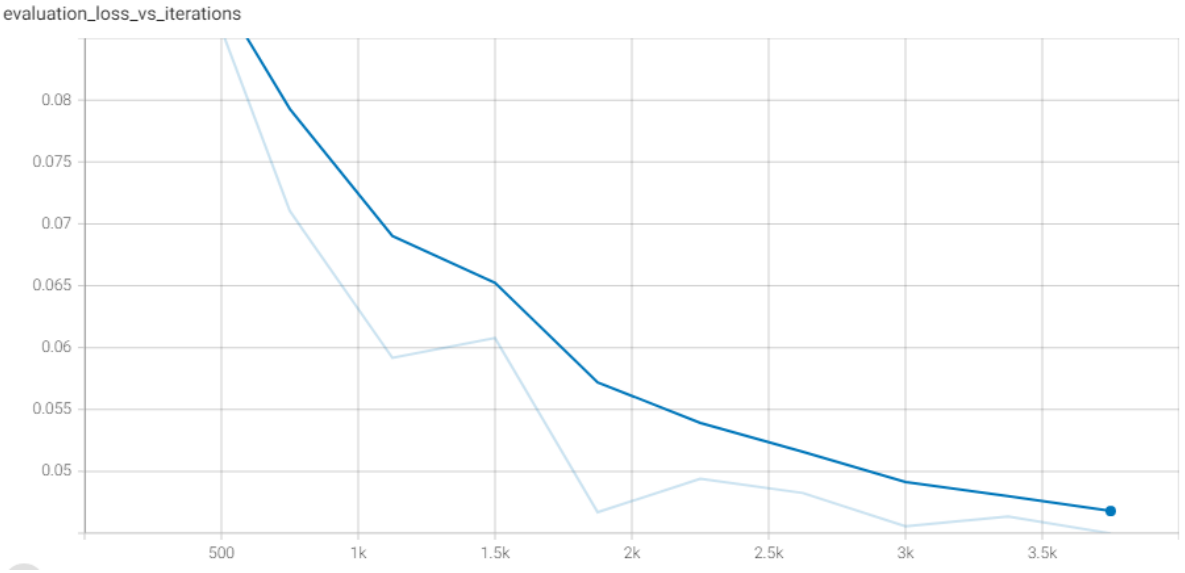


epoch_loss

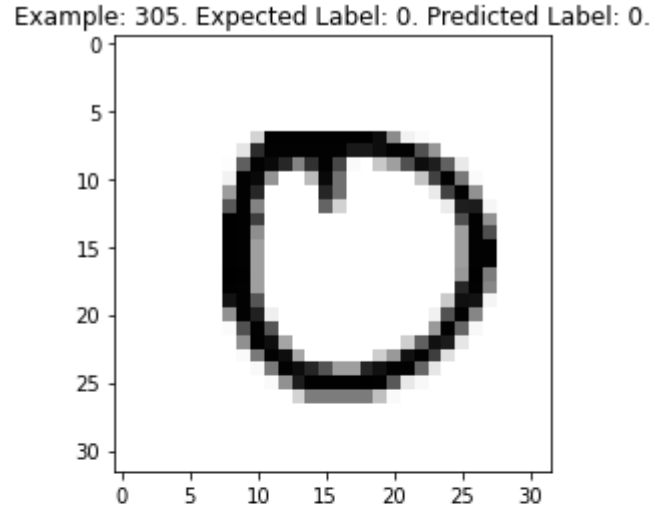


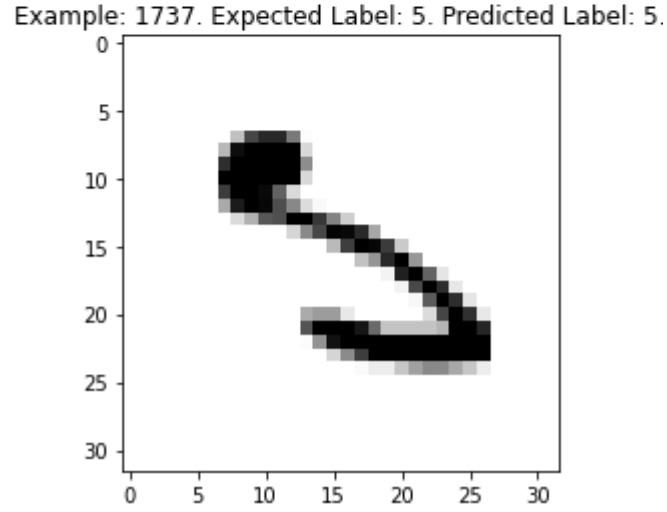
evaluation_accuracy_vs_iterations



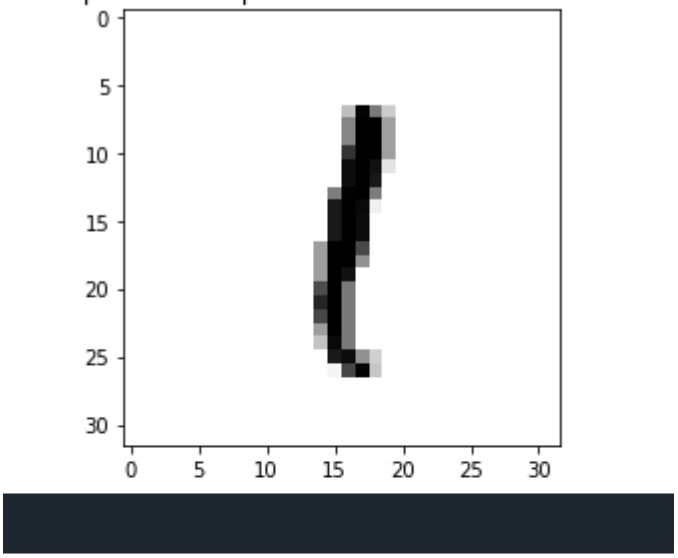


2.2. Avaliação da LeNet-5

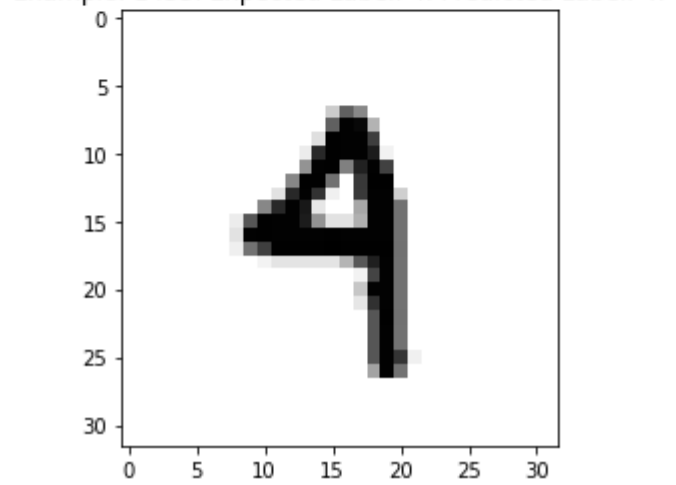




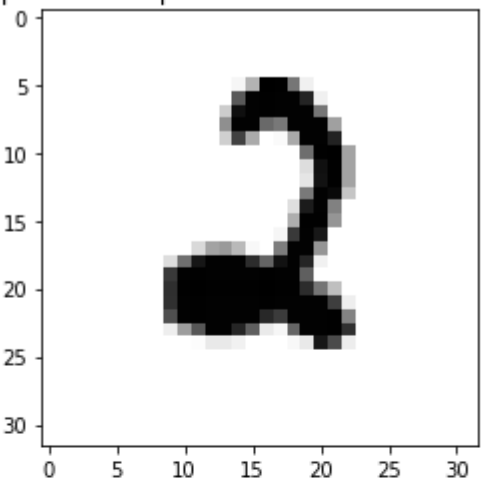
Example: 4021. Expected Label: 1. Predicted Label: 1.



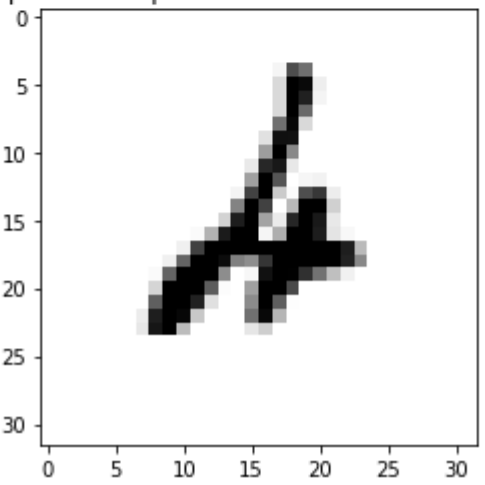
Example: 1453. Expected Label: 4. Predicted Label: 4.



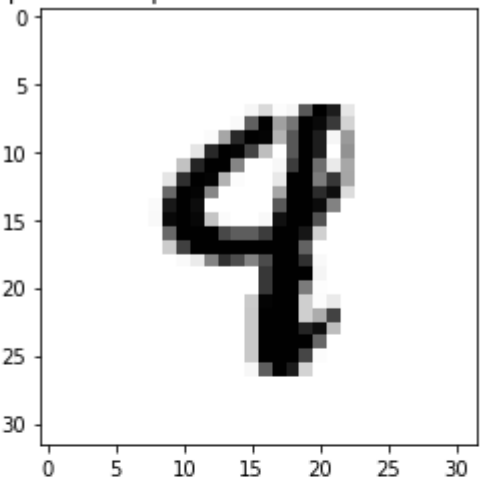
Example: 7281. Expected Label: 2. Predicted Label: 2.



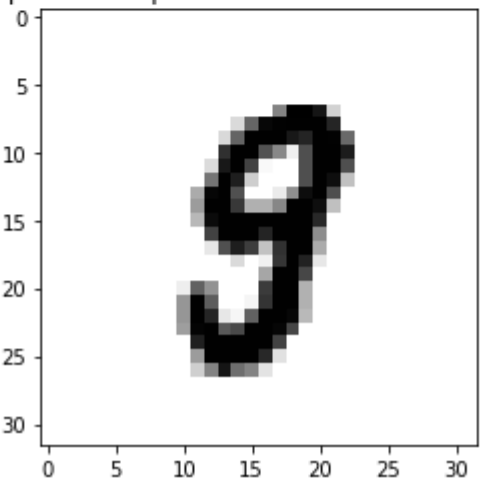
Example: 247. Expected Label: 4. Predicted Label: 6.

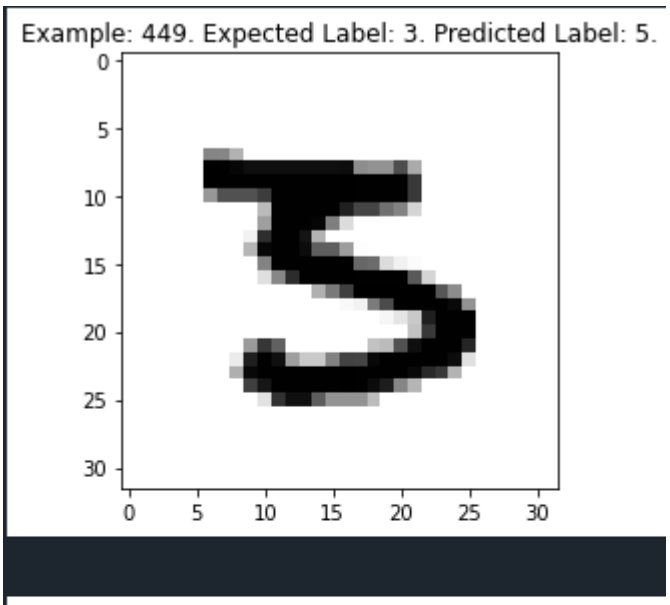


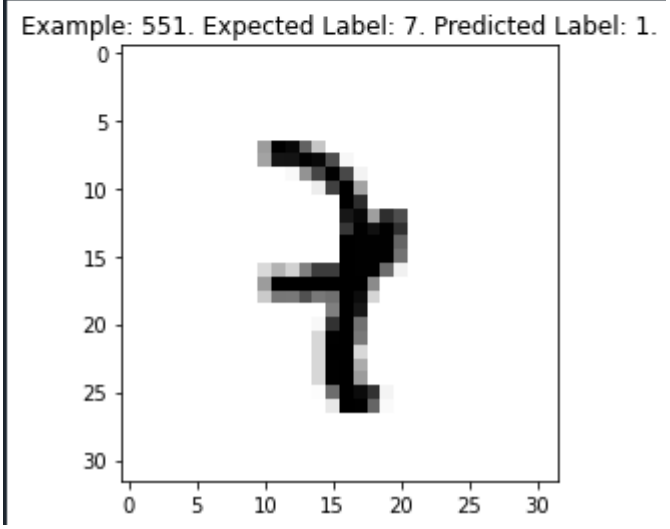
Example: 359. Expected Label: 9. Predicted Label: 8.



Example: 448. Expected Label: 9. Predicted Label: 8.







3. Discussão dos Resultados

Pelos gráficos da Evolução do Treinamento no TensorBoard podemos perceber que a acurácia nos dados de treino cresce conforme passam as iterações das épocas e cresce de forma mais rápida que a acurácia nos dados de validação. Isso é esperado, já que o modelo de treino é feito conforme os dados de treino, e não os de validação. Vemos que a função Loss dos dados de treino decresce mais rapidamente que a dos dados de validação, como também era de se esperar.

Vemos também que foi alcançado um bom resultado na avaliação da rede neural nos testes, obtendo uma precisão de 0.98 no fim das contas e conseguindo classificar alguns exemplos que mesmo humanos teriam dificuldade e ficariam em dúvida, mas, claro, também errando mais em casos mais difíceis.