

Relatório do Laboratório 8 - Redes Neurais Convolucionais

1 Breve Explicação em Alto Nível da Implementação

O objetivo desse laboratório foi implementar a rede neural convolucional de classificação LeNet-5 treinada com o dataset MNIST de dígitos manuscritos. A arquitetura da rede implementada é exibida da Tabela 1.

Tabela 1: Arquitetura da LeNet-5 implementada no laboratório.

Camada	Tipo	Número de Filtros	Tamanho da Saída	Tamanho do Kernel	Stride	Activation Function
Entrada	Imagen	1	32x32	-	-	-
1	Conv2D	6	28x28	5x5	1	tanh
2	AveragePooling2D	6	14x14	2x2	2	-
3	Conv2D	16	10x10	5x5	1	tanh
4	AveragePooling2D	16	5x5	2x2	2	-
5	Conv2D	120	1x1	5x5	1	tanh
6	<i>Dense (FC)</i>	-	84	-	-	tanh
7	<i>Dense (FC)</i>	-	10	-	-	softmax

A rede neural foi implementada utilizando-se o framework Keras, do TensorFlow, em Python. O dataset do MNIST foi separado em um *training set* e um *cross-validation set* na proporção de 80% para 20%. Foram utilizadas 10 épocas com tamanho do *batch* de 128 no treinamento e validação. A função de custo utilizada foi *cross entropy* e o otimizador utilizado foi o Adam. Não houve técnicas de regularização ou normalização. O *learning rate* foi mantido constante durante o treinamento.

2 Figuras Comprovando Funcionamento do Código

2.1 Evolução do Treinamento no TensorBoard

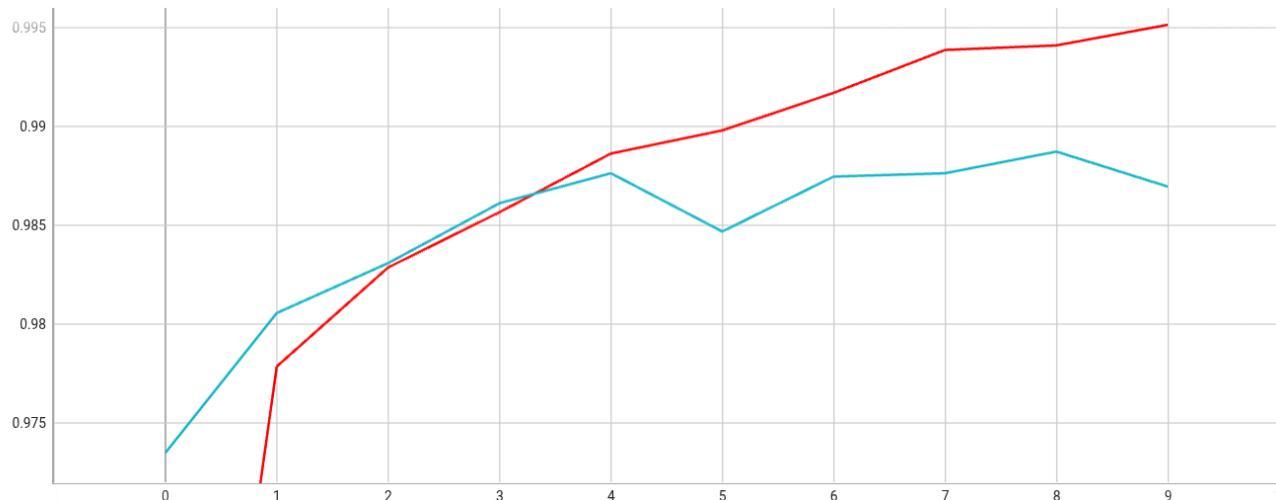


Figura 1: Precisão da rede neural em cada época. O gráfico vermelho representa o desempenho no *training set* e o azul, no *cross-validation set*.

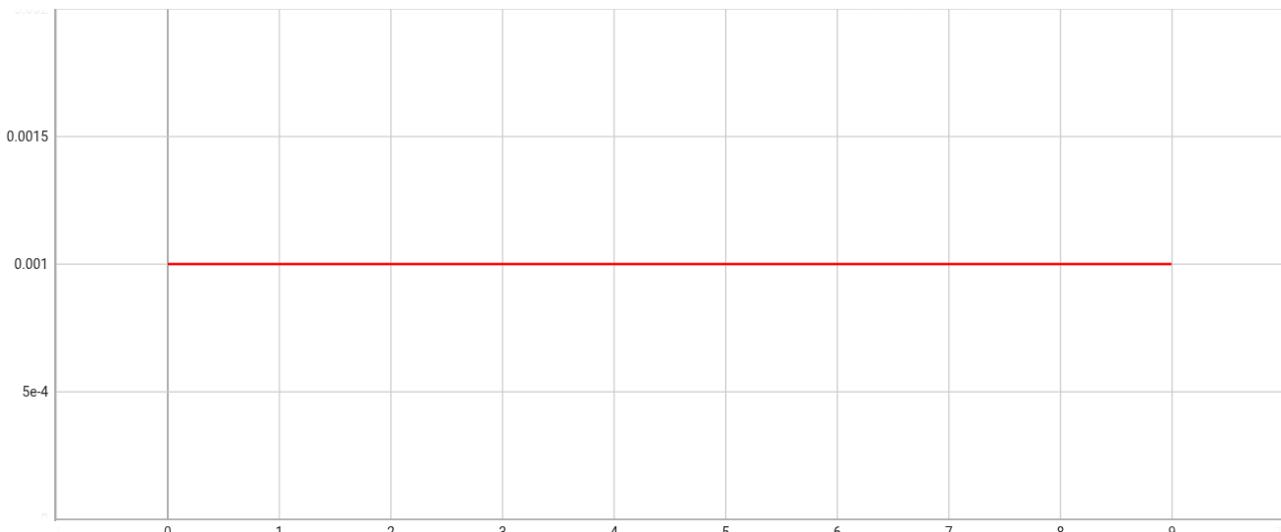


Figura 2: Taxa de aprendizado da rede neural no *training set* em cada época.

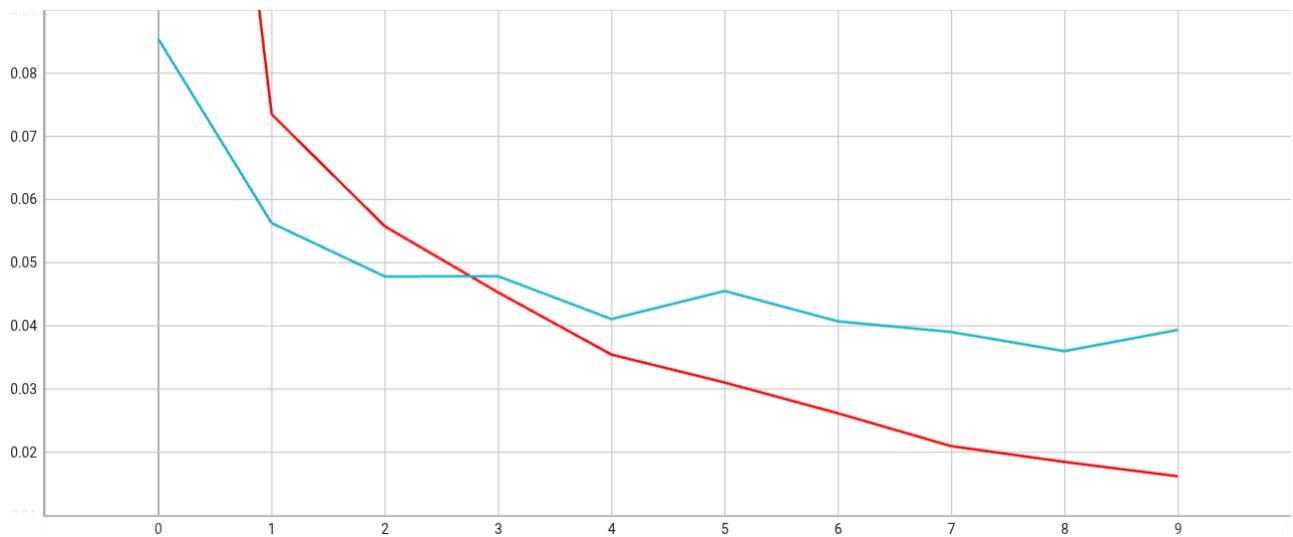


Figura 3: Valores da função de custo da rede neural em cada época. O gráfico vermelho representa o desempenho no *training set* e o azul, no *cross-validation set*.

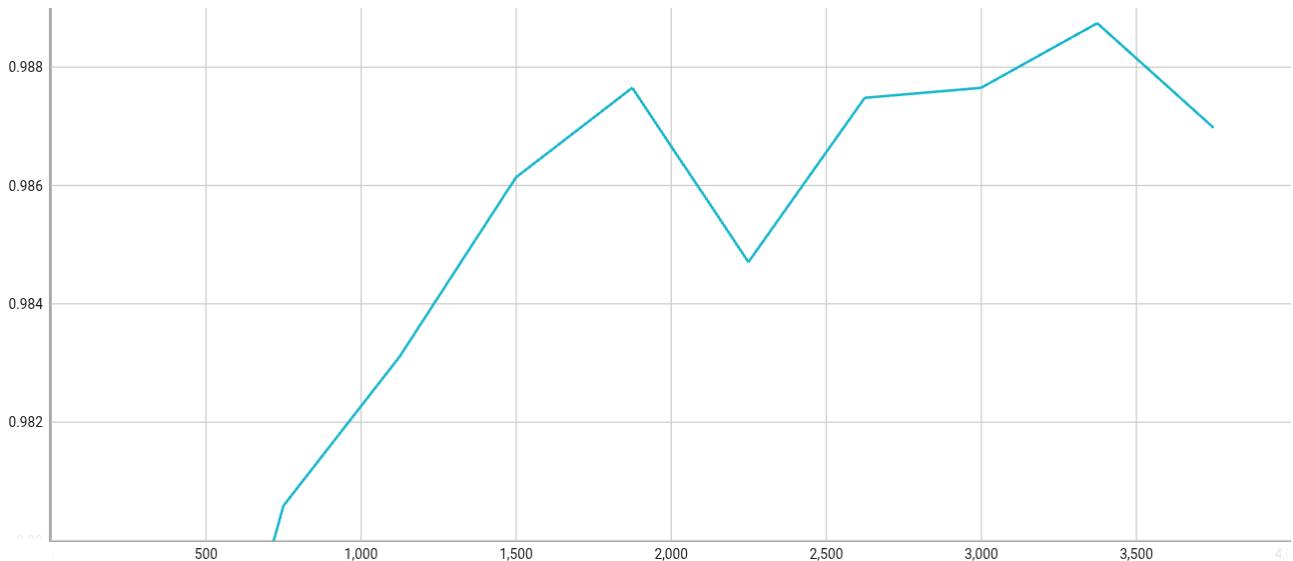


Figura 4: Precisão da rede neural durante a validação em cada iteração.

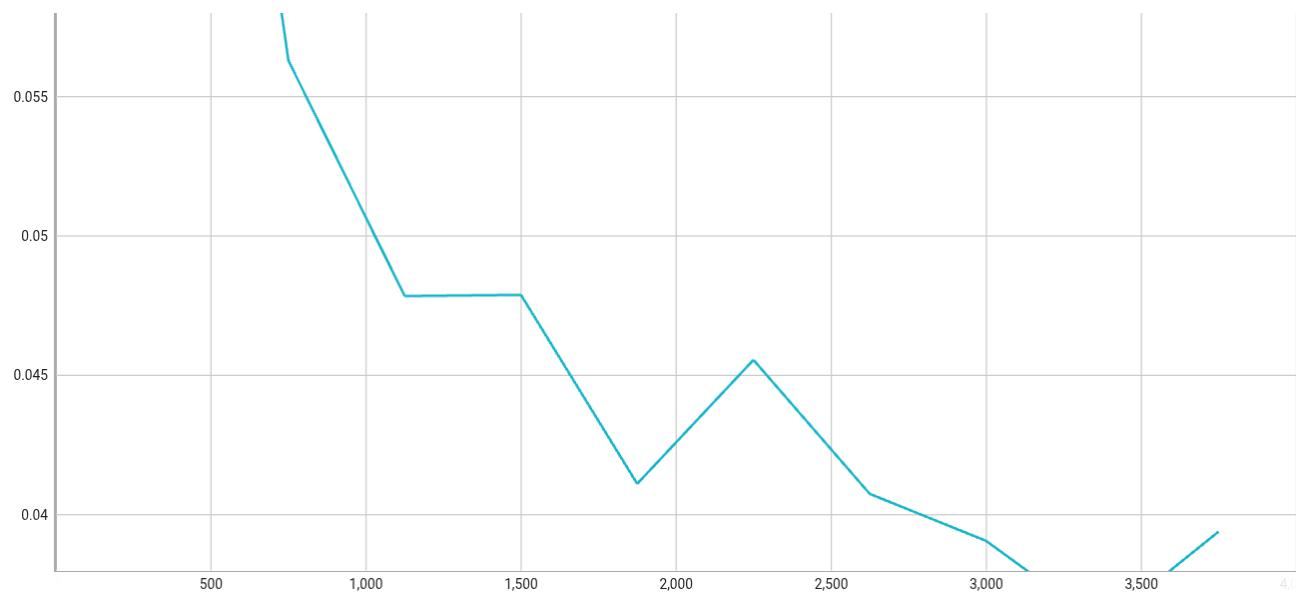


Figura 5: Valores da função de custo da rede neural durante a validação em cada iteração.

2.2 Avaliação da LeNet-5

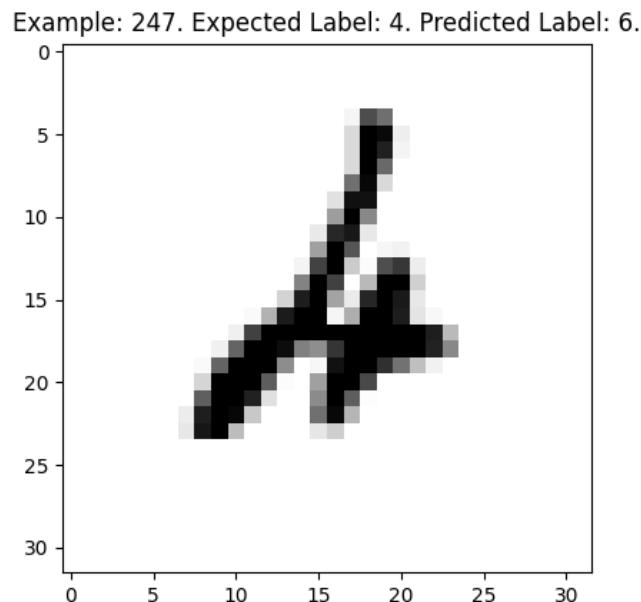


Figura 6: Exemplo de predição errada da rede neural de um número manuscrito. A *label* correta do número era 4, enquanto a rede neural encontrou 6.

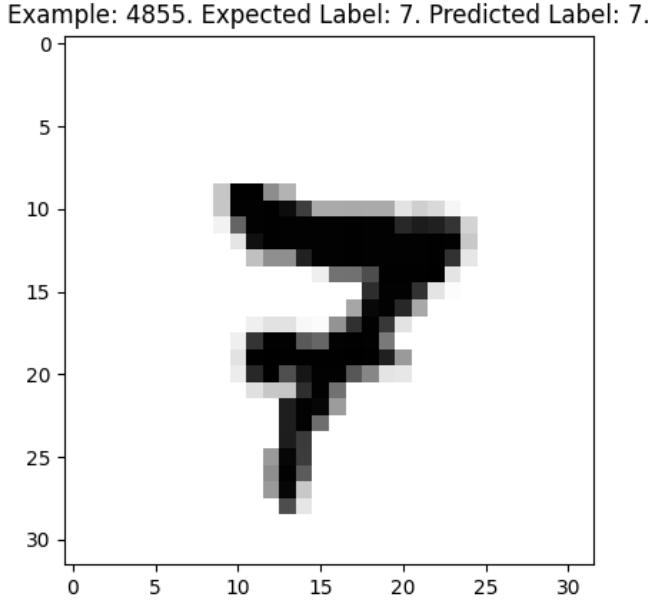


Figura 7: Exemplo de predição correta de rede neural do dígito manuscrito 7.

3 Discussão dos Resultados

Os resultados obtidos durante a implementação e treinamento da LeNet-5 demonstram o comportamento esperado de uma rede neural convolucional clássica aplicada à tarefa de classificação de dígitos manuscritos do dataset MNIST. A precisão no conjunto de treinamento e no conjunto de validação apresentou uma evolução consistente ao longo das épocas, conforme observado nos gráficos gerados pelo TensorBoard. Isso indica que a rede neural foi capaz de aprender características relevantes dos dados, ajustando seus parâmetros de forma eficaz. Os picos e vales presentes em algumas das curvas de precisão ou de função de custo são naturais ao processo de aprendizagem da rede neural.

A seção do dataset destinada à validação da rede neural durante o treinamento foi importante para evitar o *overfitting* em relação aos dados do *training set*, aumentando sua capacidade de generalização para dados com os quais a rede neural não treinou antes. Nota-se que o dígito manuscrito que a rede neural classificou erroneamente era de difícil classificação até mesmo para um humano, pois a escrita está irregular em relação ao comum, ao contrário do que se percebe no exemplo em que a rede neural atribuiu corretamente a classificação.

Em geral, a rede neural satisfez o propósito de detectar dígitos manuscritos, obtendo uma precisão na avaliação final de aproximadamente 98,65%. Outras modificações poderiam ser feitas para terem suas performances testadas e comparadas com a versão atual da rede neural, como alteração dos hiperparâmetros do número de épocas e tamanho do *batch*, além de parâmetros do otimizador como a taxa de aprendizado. Ainda, poderiam ser implementados regularização ou normalização dos dados, além de poderem ser testadas outras versões da arquitetura da LeNet-5.