

Instituto Tecnológico de Aeronáutica

CT-213: Inteligência Artificial para Robótica Móvel

Lab 8: Redes Neurais Convolucionais

Bruno Benjamim Bertucci

1 Introdução

A área de visão computacional foi beneficiada significativamente com a popularização das redes neurais. Em particular, as redes neurais convolucionais, que contêm camadas que realizam operações de convolução, mostraram-se muito eficientes para visão computacional. Um exemplo típico dessa área é o reconhecimento de caracteres escritos à mão. Para treinar uma rede neural direcionada a essa tarefa, é necessário utilizar, por exemplo, um banco de dados do MNIST (*Modified National Institute of Standards and Technology*), que contém uma variedade de imagens de números escritos à mão.

2 Implementação da rede LeNet-5

Para avaliar a capacidade de uma rede neural na tarefa de reconhecimento de escrita, foi implementada, usando a plataforma Keras, uma rede neural popular, cuja arquitetura é direcionada a essa tarefa, denominada LeNet-5. Essa rede é construída da seguinte forma: Primeiramente, ela recebe como entrada uma imagem de 32x32 pixels, com uma camada de cor, sendo cada entrada uma imagem de número do banco de dados do MNIST. Essa imagem passa por uma camada de convolução 2D com 6 filtros, um *kernel* de 5x5, *stride* de 1 em ambas as direções, e função de ativação do tipo tangente hiperbólico.

Para uma camada de convolução 2D, a dimensão da saída é dada pela Equação 1, onde a dimensão da imagem de entrada é dada por (n_H, n_W) , p é um parâmetro que indica o número de células nulas adicionais que devem ser criadas nas bordas da imagem, com o intuito de evitar a perda de informações das bordas, f é a dimensão do *kernel* do filtro, e s é um parâmetro que indica o deslocamento da janela de convolução.

Para cada filtro, uma matriz com essas dimensões é criada, portanto, o resultado final é um conjunto de matrizes. Sendo assim, a saída dessa primeira camada de convolução é constituída de 6 matrizes 28x28.

$$\text{outputShape} = \left(\left\lfloor \frac{n_H + 2p - f}{s} + 1 \right\rfloor, \left\lfloor \frac{n_W + 2p - f}{s} + 1 \right\rfloor \right) \quad (1)$$

A próxima camada realiza um *Average pooling* na saída da camada anterior, com tamanho do *pool* sendo 2 e com *stride* de 2 em ambas as direções, de tal forma que a saída dessa camada é constituída de 6 matrizes de 14x14, ou seja, as matrizes foram reduzidas a metade do tamanho original em cada direção. Essa saída é, então, direcionada a um conjunto semelhante de camadas de convolução e de *Average pooling*, porém com 16 filtros. Portanto, desse segundo conjunto de camadas, a saída é constituída de 16 matrizes 5x5.

Em seguida, os dados passam por uma última camada de convolução 2D, semelhante às anteriores, porém com 120 filtros, o que, junto com o *kernel* de 5x5, produz como saída 120 matrizes 1x1.

Nessa etapa, é necessário adaptar essa saída para as próximas camadas, que são do tipo densas, ou seja, completamente conectadas. Por isso, insere-se uma camada intermediária denominada *Flatten*, que transforma as 120 matrizes 1x1 em um vetor de 120 elementos. Esse vetor é, então, levado a uma primeira camada densa, com 84 neurônios e usando tangente hiperbólico

como função de ativação. Finalmente, tem-se a camada dos *outputs*, contendo 10 neurônios e com função de ativação *softmax*, produzindo, então, o resultado final da rede neural.

3 Treinamento da rede LeNet-5

Para realizar o treinamento da implementação da rede neural LeNet-5, foi utilizado um banco de dados do *MNIST*, junto com os números corretos apresentados em cada imagem. Em seguida, esse conjunto foi dividido da seguinte forma: 80% dele foi destinado ao treinamento da rede neural, e os 20% restantes, para o teste do desempenho dela ao final de cada época de treinamento. Em seguida, foi criada uma instância da rede neural, compilada usando entropia cruzada como *Loss function* e com otimizador *Adam*.

O treinamento foi executado, sendo geradas *batches* de tamanho 128 tanto para o conjunto de treinamento quanto o conjunto de teste. Ao todo, foram executadas 10 épocas de treinamento. Ao final dele, os dados do modelo foram salvos em um arquivo *.json*, para uso posterior. Os resultados desse treinamento encontram-se na Figura 1, onde a curva vermelha refere-se ao conjunto de treinamento, e a azul, ao conjunto de teste. As curvas destacadas foram suavizadas na plataforma *Tensorboard*, de forma que as demais curvas, que têm cores correspondentes às das curvas suavizadas, porém com maior transparência, representam as curvas originais, sem aplicação de suavização.

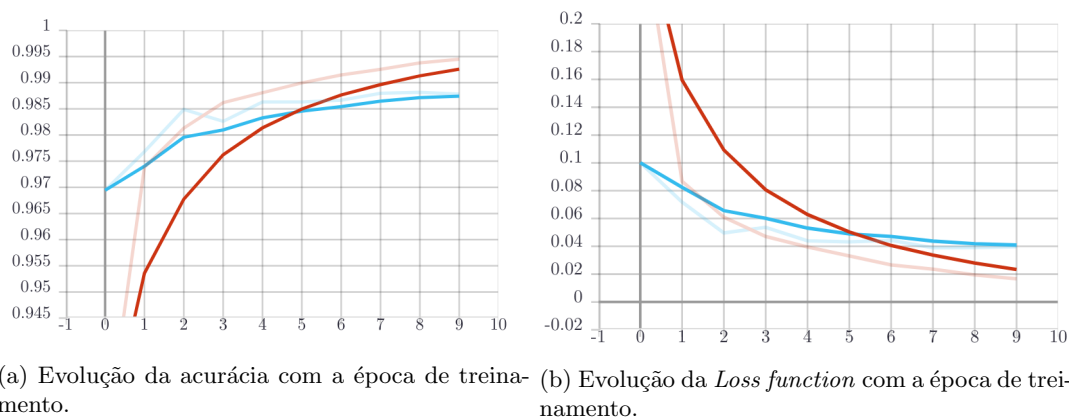
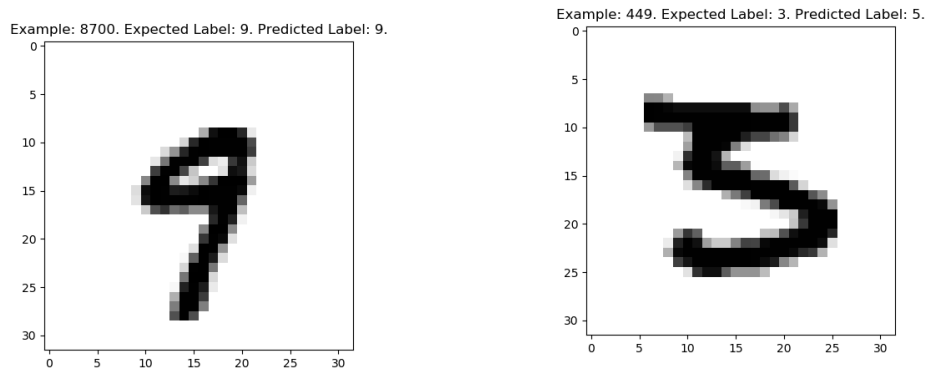


Figura 1: Treinamento da LeNet-5.

Observa-se que o treinamento rendeu uma acurácia alta, junto com um custo baixo, para o conjunto de treinamento e o de teste. É esperado que, após certo número de épocas de treinamento, o conjunto de treinamento tenha um melhor desempenho do que o de teste, o que pode ser verificado nas figuras. Não foi observado aumento no custo da avaliação feita com o conjunto de teste com o passar das épocas, o que sugere que não houve o fenômeno de *overfitting* nesse treinamento.

4 Avaliação do treinamento da rede

Após a realização do treinamento da rede neural, ela foi aplicada para reconhecer os dígitos manuscritos de um *dataset* completo do *MNIST*, a fim de avaliar o desempenho dela após o treinamento. Foram gerados exemplos de acertos e erros realizados nesse teste. A Figura 2 mostra um exemplo de acerto e um exemplo de erro.



(a) Exemplo de número reconhecido corretamente. (b) Exemplo de número reconhecido incorretamente.

Figura 2: Exemplos do teste de avaliação da LeNet-5.

Pode-se observar que a rede neural foi capaz de identificar o número 9 ilustrado, o que é um bom indicativo da sua habilidade, uma vez que essa ilustração poderia facilmente ser confundida por um número 4, mesmo por um humano. Por sua vez, no caso do número 3 ilustrado na figura, a rede interpretou como sendo um número 5, porém, observa-se que, para essa ilustração específica, um humano poderia cometer o mesmo erro, portanto, a ocorrência de erros como esse não desqualifica a habilidade da rede LeNet-5 para o reconhecimento de texto escrito. Em geral, os testes realizados com a implementação da LeNet-5 mostram que essa rede neural tem um desempenho bom para a tarefa de reconhecimento de dígitos manuscritos.