

EXERCÍCIO 3 - APRENDIZAGEM PROFUNDA

BRUNO ANDREGHETTI DANTAS*

Email: brunoandreggetti@gmail.com

Abstract— The purpose of this exercise is to study different parameters of a Convolutional Neural Network and verify their effects on the network's learning process.

Keywords— Convolutional, Neural Network, Deep Learning, Keras.

Resumo— Este exercício visa estudar os diferentes parâmetros de uma Rede Neural Convolutiva e verificar seus efeitos no processo de aprendizagem da rede.

Palavras-chave— Convolutiva, Redes Neurais, Aprendizagem Profunda, Keras.

1 Introdução

Os algoritmos de aprendizagem de máquina para classificação de padrões são diversos. De um modo geral, nos últimos anos tem aumentado cada vez mais o uso de redes neurais convolucionais profundas para classificação de imagens. Para algumas tarefas, como a classificação de algarismos manuscritos no conjunto de dados MNIST, a taxa de erro é inferior a 0.3%, equiparando-se ao desempenho humano. (LeCun et al., 2010)



Figura 1: Exemplos de algarismos do MNIST.

As redes neurais convolucionais são um grande avanço complementar às redes neurais densas. Embora as densas sejam capazes de aproximar resultados com excelente precisão, seu treinamento é computacionalmente custoso e a quantidade de parâmetros a serem ajustados aumenta rapidamente ao adicionar novas camadas à rede. Além disso, as redes densas acabam por exibir um efeito de redundância: diferentes regiões da imagem exigem diferentes conjuntos de pesos a serem treinados para reconhecer o mesmo padrão.

As redes convolucionais resolvem este problema através da aprendizagem de um conjunto de matrizes - chamadas de *kernels* - de convolução. Por serem convolucionadas ao longo da imagem, as matrizes treinadas podem se especializar em detectar padrões diversos como bordas, retas, curvas e padrões de cores, por exemplo. Comumente, tais padrões detectados são então passados

adiante para uma rede densa para que seja feita a classificação da imagem.

Junto à técnica de camadas convolucionais, é comum também a seleção de máximos como representantes de regiões da imagem, de modo a reduzir a quantidade de dados a ser processada, técnica conhecida como camada de *pooling*.

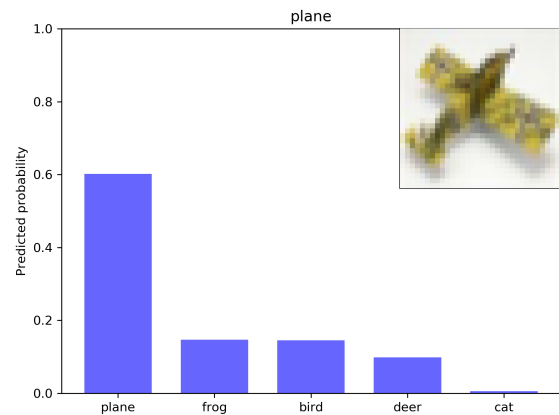


Figura 2: Distribuição de probabilidade dada como saída da rede convolutiva.

Dados topologia e treinamento adequados, a rede neural convolutiva é capaz de classificar padrões dentro do conjunto de classes definidas no treinamento, como é possível observar na Figura 2, onde a rede classifica corretamente o avião com aproximadamente 60% de certeza.

2 Metodologia e Desenvolvimento

Para desenvolvimento do exercício foi utilizada a linguagem Python 3.6 com o módulo Keras utilizando o TensorFlow como *back-end*.

O processamento dos dados foi realizado nas máquinas disponibilizadas aos alunos através da Intel AI DevCloudTM.

2.1 Topologias avaliadas

Foi avaliado o desempenho de 10 topologias - ou modelos, nomenclatura utilizada pelo Keras - di-

ferentes, numeradas de 0 a 9. A topologia 0, chamada de *modelo0*, foi utilizada como base para as outras.

O modelo 0 foi definido com 6 camadas de convolução 2D com cada duas camadas contendo 32, 64 e 128 kernels 3x3 respectivamente, realizando *padding* da imagem. Essas camadas foram intercaladas alternadamente por 3 camadas de *dropout* com taxa de 20% para evitar overfitting da rede e 3 camadas de *pooling* com janela de tamanho 2x2.

Em seguida, a rede passa por duas camadas densas de 1024 e 512 neurônios respectivamente, ambas sucedidas por uma camada de *dropout*, para enfim passar para a última camada de 10 neurônios onde serão calculadas as probabilidades de classificação de cada imagem apresentada utilizando a função de ativação *softmax*. As funções de ativação de todas as outras camadas convolucionais e densas foram do tipo "unidade linear retificada", conhecida como "ReLU".

As variações aplicadas a essa topologia em cada um dos modelos avaliados nesse exercício estão descritas na Tabela 1. Infelizmente no modelo 3 as alterações não foram efetivas pois não foi observado no momento da definição das topologias que o padrão da camada de *pooling* implementada no Keras é de dar passos do tamanho da janela, logo as redes 0 e 3 são idênticas.

#	Variação da Topologia
0	Modelo utilizado como base.
1	A taxa de <i>dropout</i> foi aumentada para 50% em todas as camadas.
2	As camadas de <i>dropout</i> foram removidas.
3	O passo das camadas de <i>pooling</i> foi definido como 2.
4	Foi retirado o <i>padding</i> das camadas de convolução.
5	Foi retirada a etapa de convolução, restando apenas a rede densa.
6	O kernel das camadas de convolução foi aumentado para 5x5.
7	O número de neurônios foi dobrado em cada uma das camadas densas.
8	A quantidade de kernels em cada camada convolucional foi dobrada.
9	As camadas de <i>pooling</i> foram removidas.

Tabela 1: Variações nos modelos treinados.

2.2 Conjunto de Dados

As redes foram treinadas utilizando o conjunto de dados CIFAR-10. O conjunto contém 60000 imagens no formato 32x32 pixels em RGB separadas em rótulos de 10 classes. As imagens foram divididas em 50000 imagens para treinamento e 10000 para validação e carregadas a partir do Keras. (Krizhevsky and Hinton, 2009)

Embora o objetivo deste exercício não seja tentar atingir precisão comparável ao estado da arte para esse conjunto de dados, é interessante destacar que o resultado mais recente obtido até a escrita deste relatório obteve taxa de acerto de 98.52% para o CIFAR-10. (Cubuk et al., 2018)

2.3 Treinamento

O treinamento foi realizado com um *batch size* de 128 amostras e foi habilitada uma função de pa-

rada antecipada caso fosse observada uma estabilização da taxa de acerto para o conjunto de validação ao longo de 20 épocas. Deste modo, evita-se a memorização dos exemplos pela rede e o desperdício de tempo para o treinamento.

3 Resultados

Após o treinamento de cada rede foram gerados relatórios acerca de seu desempenho e seu histórico de treinamento. Os resultados foram anexados ao relatório em pastas individuais de cada modelo.

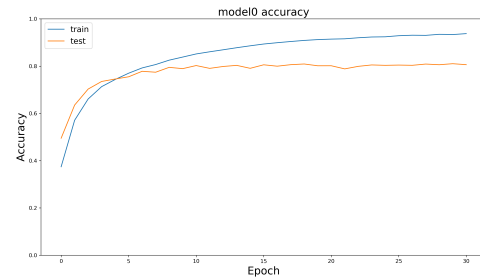


Figura 3: Precisão obtida no reconhecimento dos conjuntos de treino e de validação ao longo das épocas de treinamento.

Na Figura 3 é possível observar um comportamento que se repetiu em todas as redes avaliadas. Após um certo número de épocas, embora a taxa de acerto para o conjunto de treinamento continuasse a aumentar, a taxa para o conjunto de validação se mantinha quase constante até que a função de parada antecipada fosse ativada.

#	Acerto (%)	Nº de Parâmetros	Nº de Épocas	Tempo Médio por Época (s)
0	80,61	2.915.114	31	219
1	72,93	2.915.114	92	224
2	76,50	2.915.114	24	188
3	80,48	2.915.114	34	223
4	77,31	949.034	45	191
5	51,38	3.676.682	70	71
6	75,95	3.424.554	33	250
7	79,35	6.591.786	30	240
8	82,19	5.870.666	30	301
9	66,50	135.035.690	26	655

Tabela 2: Medidas de desempenho das topologias.

É possível observar na Tabela 2 que o desempenho da rede diminuiu em todos os casos de variação dos parâmetros do modelo 0, exceto para o modelo 8, onde é aumentada a quantidade de filtros nas camadas de convolução. É importante também destacar que o tempo de treinamento por época aumentou em aproximadamente 37% para essa rede, o que pode ser um preço alto a se pagar pelos 2% de desempenho extra obtidos.

Para cada rede treinada foram também gerados 200 gráficos de barras como o da Figura 2 para imagens do conjunto de validação, a fim de observar o comportamento da rede diante dos padrões apresentados.

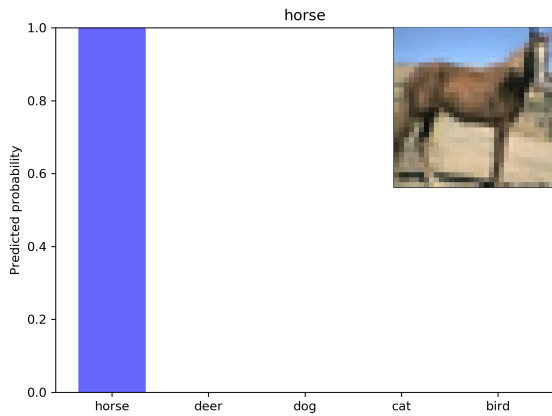


Figura 4: Acerto obtido pela topologia 8.

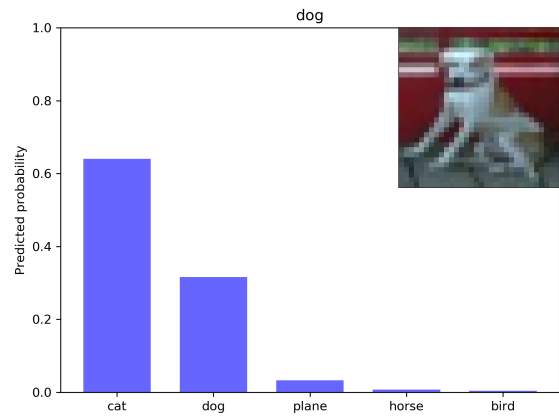


Figura 6: Erro cometido pela topologia 8.

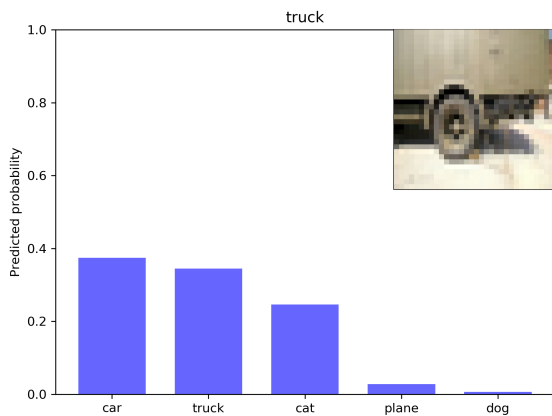


Figura 5: Erro cometido pela topologia 8.

4 Análise e Conclusões

A partir dos gráficos de barras obtidos é possível verificar diversos comportamentos interessantes das redes. O comportamento mais observado é o esperado: a rede tem muita certeza ao reconhecer imagens que um humano facilmente reconheceria, como na Figura 4.

Porém, é interessante analisar os conflitos encontrados pela rede ao classificar certas imagens. Por exemplo, na Figura 5 é possível observar que a rede ficou em dúvida entre as classes carro ou caminhão. É intuitivo imaginar que a oclusão de grande parte do caminhão levou a rede a reconhecer apenas o pneu, impossibilitando a distinção clara entre as duas categorias.

Outro problema recorrente é a falta de resolução nas imagens para distinguir claramente categorias que por natureza são similares, como gatos e cachorros. Na Figura 6 é possível observar que há dúvida entre as duas categorias, ocasionando o erro na classificação.

A baixa resolução das imagens do banco de dados, embora facilite a execução do treinamento uma vez que menos dados precisam ser processados, muitas vezes prejudica a informação contida

nelas, prejudicando consequentemente o desempenho do classificador.

Além disso, com a rede densa implementada pelo modelo 5, foi possível destacar a importância das camadas convolucionais no processo de reconhecimento. Sem elas, a rede obteve desempenho consideravelmente inferior apesar de ter um maior número de parâmetros. Por outro lado, o tempo de treinamento por época foi bastante reduzido e não foram realizados comparativos utilizando técnicas de regularização que pudessem favorecer a rede densa.

A camada convolucional se mostrou uma importante técnica complementar às redes densas, capaz de reduzir o volume de dados apresentados a essas redes e assim extrair das imagens apenas o necessário para que se possa classificar o conteúdo das mesmas, possibilitando o treinamento de classificadores poderosos para uso em aplicações diversas.

Referências

- Cubuk, E. D., Zoph, B., Mané, D., Vasudevan, V. and Le, Q. V. (2018). Autoaugment: Learning augmentation policies from data.
- Krizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images.
- LeCun, Y., Cortes, C. and Burges, C. (2010). Mnist handwritten digit database.