Trabalho 2 de Arquitetura de computadores

CIFAR-10 Object Recognition in Images

Matheus de Araújo Nogueira

***Resumo*— Esse relatório traz os resultados do segundo trabalho de arquitetura de computadores, no qual a atividade consistia em escolher um problema do site Kaggle e gerar uma solução utilizando redes neurais profundas.**

I. INTRODUÇÃO

O problema escolhido foi o CIFAR-10 (CIFAR-10 - Object Recognition in Images, 2013), que é uma base de imagens pequenas (32x32) e coloridas, com 60000 exemplares distribuídos em 10 classes. Essa base é amplamente utilizada como benchmark. Hoje essa base já possui soluções que passaram de 90% de precisão, (Graham, 2014) possui a maior precisão com 96,53%, superando a precisão humana.

II. METODOLOGIA

O método para solucionar o problema da CIFAR-10, foi utilizar uma arquitetura de rede neural pronta, a AlexNet (Alex Krizhevsky, 2012).
A AlexNet possui 8 camadas, sendo 5 convolucionais e 3 totalmente conectadas.
Escolhi essa arquitetura pois ela foi criada para classificação de imagens e por ser bem utilizada na literatura.

Foi necessários realizar algumas modificações na arquitetura para que seja possível o uso dela na base CIFAR-10. Essas modificações foram no processamento da imagem, na arquitetura original, a rede realizar um corte na imagem de entrada para 227 pixels. Como a base escolhida só possui imagens de 32x32 pixels, não fazia sentido continuar com esse processo, pois isso iria gerar ruído na imagem de entrada.

Outros parâmetros da rede também foram alterados, como o L*earn Rate* para 0,05, a politica de queda da *L*oss para *Step Down* em 75% e a quantidade de épocas para 30.

Utilizamos a técnica de validação cruzada *k-fold*, com 5-folds. Sendo 30 mil imagens para o treino, 10 mil para teste e 10 mil para validação da rede. A porção de validação serviu para monitorar o treinamento da rede.
O computador utilizado para treinar a rede foi a LCAD205, com as seguintes especificações.

Especificações do processador	
Modelo	Intel Xeon E5606 2.13GHz
Quantidade de núcleos	4
Quantidade de threads	4
Cache	8 Megabytes
RAM	12 Gigabytes

Especiações da placa gráfica	
Modelo	GTX 660
VRAM	1.5 Gigabytes
Frequência	0.889 GHz
Quantidade de WARPS	32
CUDA cores	1152
Max de threads por SMP	2048
Max de threads por bloco	1024

Para facilitar a construção e execução da rede, foram utilizados a Caffe (Jia, 2014)e o DIGITS (NVIDIA). A Caffe é um poderoso framework de rede neural profunda, com muitas funcionalidades e amplamente utilizada na literatura. Já o DIGITS é gerenciador de tarefas de rede neural, ele controla o uso dos recursos das placas gráficas durante o processo e auxilia na visualização do aprendizado da rede. Assim, essas duas ferramentas são ótimas para ajudar nos trabalhos de rede neural com placas gráficas.

III. RESULTADOS

Analisando os resultados de todos os *folds*, a média de precisão na primeira saída da rede foi de 64,13% e nas cinco primeiras saídas foi de 95,4%. Esses resultados são melhores do que probabilidade de uma escolha aleatória das classes, mas inferiores aos encontrados na literatura.

Saída da rede em cada *fold*		
	TOP-1	TOP-5
Fold-1	63,69	95,46
Fold-2	64,21	95,37
Fold-3	64,68	95,26
Fold-4	64,25	95,73
Fold-5	63,84	95,67
Media	**64,134**	**95,498**

Tabela 1: Valores das saídas da rede em cada *fold* e a média final.

A figura 1 mostra o comportamento da rede durante o treino com o *fold 3*, o que mostrou melhor resultado na primeira saída da rede. A *loss* se comportou como o esperado, cair com o passar do tempo, mas a precisão dela não foi alta. Isso se deu pelos parâmetros da rede.

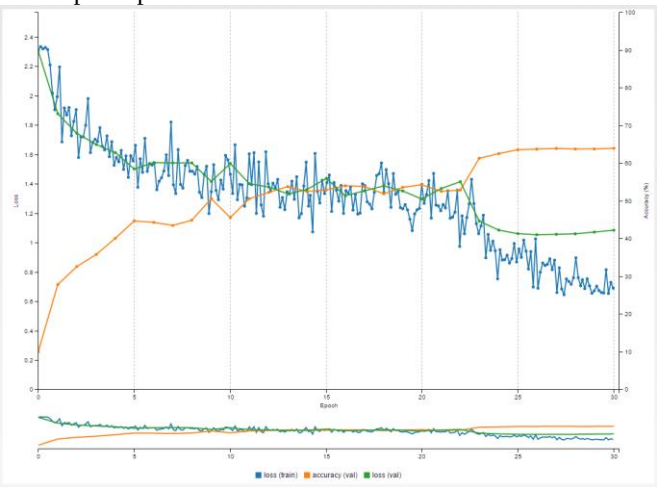

Figura 1: Gráfico com a *loss* e precisão da rede do fold 3 durante o treinamento.

É visível que a rede aprendeu alguma coisa com a base, mas não gera resultados muito precisos. Foram feitos vários outros testes com mais épocas, mas a taxa de acerto do aprendizado continuou a mesma, por isso da escolha de 30 épocas. Esse comportamento também foi visível nos outros *folds.*

O tempo médio de treinamento da rede foi de 764 segundo. A figura 2 mostra o tempo de treinamento dos *folds*.

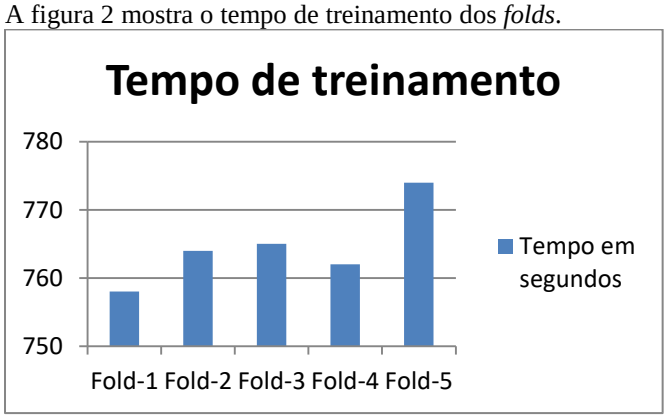

Figura 2: Gráfico com o tempo de treinamento da rede em cada *fold.*

IV. Conclusão

Com esses resultados, a precisão da arquitetura escolhida não obteve bons resultados em relação à literatura, mas já são melhores do que a escolha aleatória de uma das classes. É visível que a rede pode classificar melhor que o humano, pois ela consegue aprender características nas imagens que o cérebro humano não é capaz de associar.

Uma forma de melhorar os resultados seria trabalhar mais nos parâmetros da rede ou até mesmos modificar diretamente a arquitetura, mas essa ultima escolha iria descaracterizar a AlexNet.

V. Bibliografia

(18 de Outubro de 2013). Acesso em 13 de Dezembro de 2016, disponível em CIFAR-10 - Object Recognition in Images: https://www.kaggle.com/c/cifar-10

Alex Krizhevsky, I. S. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems 25 (NIPS 2012)*, p. 9.

Graham, B. (18 de Dezembro de 2014). *Fractional Max-Pooling.* Acesso em 10 de Dezembro de 2016, disponível em https://arxiv.org/abs/1412.6071v4

Jia, Y. a. (20 de Junho de 2014). Caffe: Convolutional Architecture for Fast Feature Embedding. *arXiv preprint arXiv:1408.5093*, p. 4.

Kaggle. (s.d.). Acesso em 10 de Dezembro de 2016, disponível em Kaggle: https://www.kaggle.com/

Krizhevsky, A. (8 de Abril de 2009). *Learning Multiple Layers of Features from Tiny Images.* Acesso em 10 de Dezembro de 2016, disponível em http://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

NVIDIA. (s.d.). *NVIDIA DIGITS | NVIDIA Developer*. Acesso em 10 de Dezembro de 2016, disponível em NVIDIA Developer: https://developer.nvidia.com/digits