# MC886 - Relatório do Assignment 04
# Transferência de Aprendizado: Deep Learning

Pedro Stramantinoli P. Cagliume Gomes
175955
p175955@dac.unicamp.br

Ruy Castilho Barrichelo
177012
r177012@dac.unicamp.br

## I. Introdução

Ao passo que os algoritmos e práticas do aprendizado de máquina em redes profundas se desenvolveram, mostrou-se necessário, também, o progresso de técnicas que as auxiliassem, principalmente nos pontos que deixam a desejar.

Isto é, tanto a necessidade de uma grande quantidade de dados para se treinar uma rede profunda efetivamente, quanto a quantidade de tempo e de recursos computacionais que devem ser alocados para que o processo funcione adequadamente.

Para isso, um dos métodos criados foi a transferência de aprendizado, na qual são aproveitados modelos pré-treinados para agilizar o treinamento da nova rede desejada, em geral a partir de uma técnica chamada *fine-tuning*, em que as últimas camadas da rede são treinadas novamente, no entanto, com o *dataset* de interesse.

Em geral, essa estratégia apresenta melhores resultados quanto mais próximos são os *datasets* das tarefas utilizadas. Neste caso, o foco de estudo deste trabalho é a análise da transferência de aprendizado entre o *dataset* da *ImageNet*[3] e o *dataset* CIFAR-10[1], no contexto da arquitetura *SqueezeNet*.

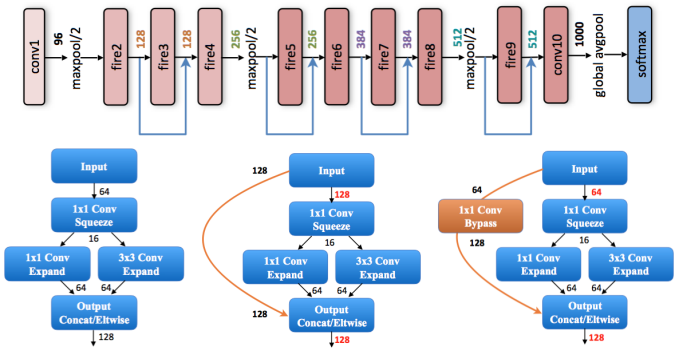


Figura 1. Arquitetura da *SqueezeNet*

## II. Tratamento dos Dados

O *dataset* CIFAR-10 consiste em 60000 imagens de 32 x 32 pixels de dimensão, com 3 canais RGB. Ou seja, contém matrizes de dimensões 32 x 32 x 3, com valores inicialmente entre 0 e 255.

Primeiramente, foi realizado o *scaling* e a normalização dos dados, com o intuito de reter os valores entre 0 e 1 e, posteriormente, centralizá-los em torno de zero.

Já em relação aos rótulos das classificações, foi realizado um *encoding*, de forma a transformar o que era uma matriz coluna - que continha apenas o identificador da classe correspondente - em uma matriz que continha informações sobre as 10 classes existentes no CIFAR-10.

Por fim, o conjunto de dados, retirada a fração destinada ao teste final, foi dividido em conjuntos de treino e validação, para que os resultados possam ser verificados a cada treinamento realizado.

## III. Experimentação e Resultados

Os experimentos foram divididos em 3 segmentos, de acordo com a quantidade de camadas a permanecerem congeladas durante o *fine-tuning*, com exceção, é claro, das últimas quatro camadas, que são reinicializadas para o novo treinamento.

A primeira parte consiste em manter todas congeladas; a segunda, em descongelar os dois últimos módulos *Fire* e, o último, em manter todas descongeladas.

Em relação a execução dos algoritmos, optou-se por variar o *batch_size* entre 250 e 500, enquanto o número de *epochs* foi variado entre 50, 150 e 300, ambos com o objetivo de analisar o impacto desses parâmetros nos resultados.

Ainda, vale ressaltar que o código foi executado via *Google Colab*[6], com aceleração de *hardware* via *GPU*.

### A. Todas camadas congeladas

Para este primeiro caso, os resultados obtidos foram, como previa o próprio enunciado, inferiores ao ideal, em torno de 40% de acurácia. Os valores de *Loss* e *Accuracy* para os diferentes *batch_size* e *epochs* podem ser vistos na tabela I.

Tabela I
Todas camadas congeladas

| *Batch size* de 250 | | | |
|---|---|---|---|
| *Epochs* | 50 | 150 | 300 |
| *Loss* | 0.0745 | 0.0742 | 0.0742 |
| Acurácia | 0.4086 | 0.4087 | 0.4063 |
| *Batch size* de 500 | | | |
| *Epochs* | 50 | 150 | 300 |
| *Loss* | 0.0747 | 0.0743 | 0.0742 |
| Acurácia | 0.4067 | 0.4078 | 0.4101 |

Como comparação, os resultados de *Loss* no treino mantiveram-se em torno de 0.078 e a acurácia, 0.345.

### B. Módulos Fire e Camadas de Classificação descongeladas

Em seguida, com a decisão de descongelar os dois últimos módulos *Fire*, pode-se notar que os valores resultantes foram similares aos anteriores, ao aumentar absolutamente a acurácia em cerca de 10% e permanecer ao redor de 50% . Novamente, os resultados estão explicitados na tabela II.

Tabela II
MÓDULOS *Fire* E CAMADAS DE CLASSIFICAÇÃO DESCONGELADAS

| *Batch size* de 250 | | | |
|---|---|---|---|
| *Epochs* | 50 | 150 | 300 |
| *Loss* | 0.0708 | 0.0876 | 0.0916 |
| Acurácia | 0.5018 | 0.4728 | 0.4654 |
| *Batch size* de 500 | | | |
| *Epochs* | 50 | 150 | 300 |
| *Loss* | 0.0666 | 0.0834 | 0.0884 |
| Acurácia | 0.513 | 0.4851 | 0.4762 |

Comparativamente, os resultados de *Loss* no treino mantiveram-se em torno de 0.014 e a acurácia, 0.89.

### C. Todas camadas descongeladas

Por fim, indagou-se sobre o que aconteceria diante de um *fine-tuning* executado sobre todas as camadas do modelo pré-carregado.

É possível perceber que ouve uma melhora considerável, uma vez que os valores de acurácia saltaram para aproximadamente 75%, resultados que estão disponíveis na tabela III.

Tabela III
TODAS CAMADAS DESCONGELADAS

| *Batch size* de 250 | | | |
|---|---|---|---|
| *Epochs* | 50 | 150 | 300 |
| *Loss* | 0.0392 | 0.0421 | 0.0479 |
| Acurácia | 0.7641 | 0.7548 | 0.7218 |
| *Batch size* de 500 | | | |
| *Epochs* | 50 | 150 | 300 |
| *Loss* | 0.0403 | 0.0414 | 0.0413 |
| Acurácia | 0.7566 | 0.7634 | 0.77 |

Já neste caso, os resultados de *Loss* no treino mantiveram-se em torno de 0.050 e a acurácia, 0.96.

### D. Teste

Agora, com todos os modelos gerados e suas respectivas acurácias calculadas, escolheu-se o de melhor desempenho e, a partir dele, realizar a predição sobre o conjunto de testes para a validação final da qualidade do modelo obtido.

Dessa forma, optou-se por um modelo da terceira etapa, correspondente à execução de 300 *Epochs* e *Batch* de 500. Os resultados associados a ele foram: *Loss* de 0.0426 e acurácia de 0.7612, com uma queda de menos de 1% em relação à validação.

## IV. ANÁLISE E COMPARAÇÕES

Os resultados atingidos com a experimentação ocorrida permitem análises não só sobre a viabilidade da transferência de aprendizado, mas também sobre o impacto dos parâmetros de execução e os diferentes níveis de congelamento das camadas.

Primeiramente, nota-se que a variação em *Batch_size* influenciou muito pouco os valores atingidos, com uma leve vantagem para o maior *batch*, apesar de também ser vantajoso quanto ao tempo de execução, inferior ao menor *batch*.

Já o número de *epochs* apresentou comportamentos diferentes nos modelos. Enquanto o crescimento do número de *epochs* implicou uma redução na acurácia para os 2 primeiros tipos de modelos, resultou em uma elevação, no caso do terceiro, ambos em torno de 2% de variação.

Uma observação interessante é a alta queda nas acurácias dos dois últimos tipos de modelo, entre conjuntos de treino e os de validação. Isso pode ser um indicativo de *overfitting*, no entanto ocorreu similarmente para todas as variações dos parâmetros de execução. Embora a queda no terceiro tipo de modelo também tenha sido alta, foi comparativamente mais reduzida.

Em relação as estratégias de congelamento das camadas, que foram a principal diferença entre os modelos, percebe-se que a transferência de aprendizado não apresentou resultados viáveis para um número reduzido de descongelamentos. Isso mostra que o *fine-tuning* das últimas camadas não foi o suficiente para adequar a rede treinada no *dataset* da *ImageNet* ao do *CIFAR-10*. Porém, ao descongelar todas as camadas e permitir a adaptação de toda a rede ao novo *dataset*, os valores obtidos foram mais aceitáveis, principalmente ao se considerar que, caso não houvesse uma queda brusca nas acurácias entre treino e validação, o modelo seria ainda melhor, o que leva à ideia de que ajustes no modelo e na execução do algoritmo, como os parâmetros variados, possam levar a um resultado que se aproxime mais do desejado.

Por fim, nota-se que a execucação no conjunto de testes gerou, de acordo com o que era esperado, a resultados próximos aos vistos sobre o conjunto de validação, a partir do uso do mesmo modelo.

## V. CONCLUSÃO

Apesar da obtenção de resultados abaixo do que se consideria ideal, este estudo ainda possibilitou não só o entendimento da prática da transferência de aprendizado a partir de um modelo pré-treinado, mas também a maior clareza de como essa técnica pode ser valiosa tanto em termos de tempo e de recursos computacionais - uma vez que foi notável a diferença de tempo consumido para a execução do treino entre redes com e sem transferência, realizada em trabalhos anteriores-, quanto em relação à necessidade de posse de muitos dados - que é um dos pontos mais fundamentais nesse campo de estudo.

## REFERÊNCIAS


[1] The CIFAR-10 dataset. Health News in Twitter. Disponível em: https://www.cs.toronto.edu/ kriz/cifar.html
[2] NumPy v1.15 Manual. Disponível em: https://docs.scipy.org/doc/numpy/
[3] ImageNet Database. Disponível em: http://www.image-net.org/
[4] Scikit-Learn. Stratified Shuffle Split. Disponível em: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedShuffleSplit.html
[5] Keras Documentation. Model. Disponível em: https://keras.io/models/model/
[6] Google Colaboratory. Disponível em: https://colab.research.google.com/