

MC886 - Transfer Learning

Naomi Takemoto
RA 184849
naomitkm1@gmail.com

Thiago Dong Chen
RA 187560
thiagodchen@gmail.com

I. INTRODUÇÃO

A. Objetivos

Neste trabalho são estudadas diferentes maneiras de se aplicar a técnica de *transfer learning*, sendo analisados os resultados em termos de acurácia e queda do erro para que fosse possível verificar a eficácia da técnica. Para tanto, foi utilizada a rede *SqueezeNet* pré treinada com dados do *dataset ImageNet*, o objetivo era adaptar essa rede de classificação para que pudesse trabalhar com imagens do *dataset CIFAR-10*, que conta com imagens de tamanho 32x32 pixels em 3 canais de cor(RGB).

II. TEORIA

A técnica de *transfer learning* é usada com muita frequência para treinamento de redes convolucionais, um dos principais motivos é a falta de dados anotados para se treinar uma rede convolucional do zero (com pesos inicializados de forma aleatória). As duas principais maneiras de se implementar este técnica consiste em:

- Usar as camadas convolucionais como extratoras de *features*.
- Realizar o *fine tuning* da rede, isto é "congelar" as camadas convolucionais mais superficiais e treinar as camadas mais profundas junto com as camadas *fully connected*.

[1] *Transfer learning* em redes convolucionais se vale do fato de que os pesos pré treinados dos filtros de convolução aprenderam a extrair características relevantes das imagens, portanto se aplicada a um *dataset* semelhante, é possível aproveitar essa capacidade de extração de propriedades.

III. METODOLOGIA

Foram realizados 4 experimentos com a rede.

- 1) Utilização da rede como extratora de *features*.
- 2) Treinamento em apenas os dois últimos *fireblocks*.
- 3) Treinamento em todas as camadas.
- 4) Treinamento com inicialização randômica dos pesos.

O *dataset* foi separado em 3 conjuntos:

- 1) teste, contando com 10000 imagens, 1000 de cada classe.
- 2) treinamento com 45000 imagens.
- 3) validação com 5000.

Depois de normalizar as imagens para o intervalo (0-1), foi realizado um treinamento para cada uma das configurações de rede, conforme descrito acima. Posteriormente foi escolhido o melhor modelo com base na acurácia normalizada obtida no

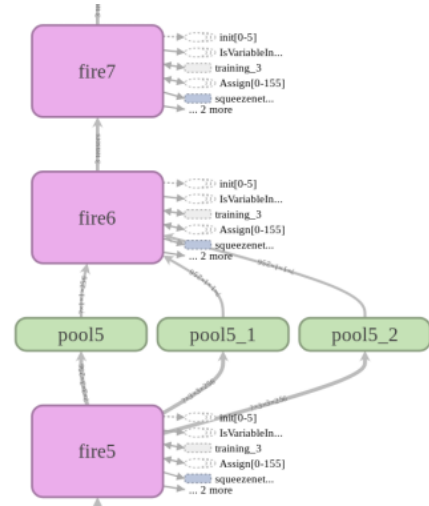


Figura 1. Trecho do grafo que representa a rede *SqueezeNet* depois do *fine tuning*

conjunto de validação. Posteriormente, foi utilizado o conjunto de testes para examinar a qualidade do melhor modelo.

IV. EXPERIMENTOS

Tabela I
ACURÁCIA NO CONJUNTO DE TREINAMENTO PARA 50 ÉPOCAS

Modelo	Acurácia Treinamento (%)	Acurácia validação (%)	Tempo de Treinamento (s)
1	31,93	17,48	540
2	56,42	32,02	731
3	94,84	59,39	1351
4	10,05	10,00	1951

A. Acompanhando o treinamento com o TensorBoard

O *TensorBoard* é uma ferramenta utilizada para visualização do grafos criados em *TensorFlow* que permite também o acompanhamento em tempo real do treinamento a partir de métricas como acurácia e erro, facilitando na verificação do ponto de *earlystop*, ponto em que o erro de validação deixa de diminuir e começa a aumentar.

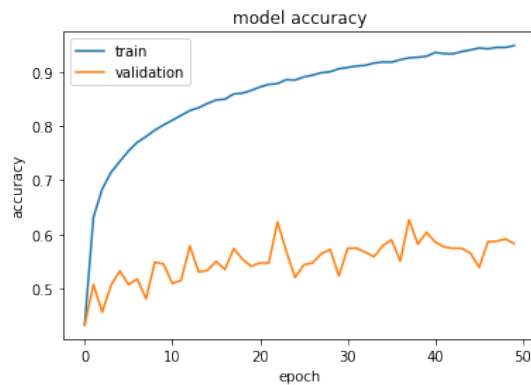


Figura 2. Acurácia do melhor modelo em função do número de épocas

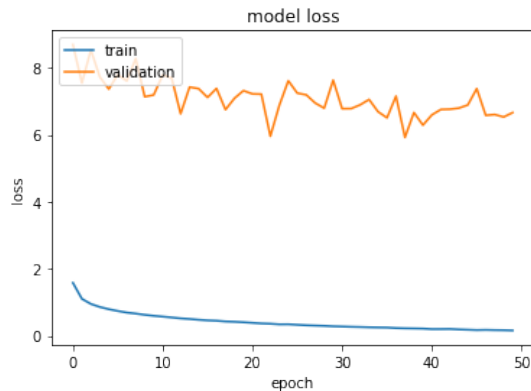


Figura 3. Custo do melhor modelo em função do número de épocas

B. Escolha do Melhor Modelo

A decisão da escolha do melhor modelo (com inicialização de pesos pré-treinados e treinamento de todas as *layers*) foi baseada na acurácia de validação I. No conjunto de testes, essa métrica atingiu o valor de 58,65%.

Comparando-se os resultados nos diferentes conjuntos, observa-se uma queda significativa de acurácia, tanto em validação quanto em teste, comparada à acurácia com o conjunto de treinamento, indicando a ocorrência de *overfitting*. Nesse caso, o treinamento acarretou em memorização ao invés de aprendizado, já que o modelo não é capaz de generalizar bem, apesar de fazer previsões melhores do que atribuição aleatória de classes.

Nas figuras 2 e 3 é possível perceber a evolução da acurácia e do erro levando-se em conta os conjuntos de treinamento e validação. Observa-se que conforme aumenta-se o número de épocas o desempenho (tanto em erro como acurácia) aumentam e tendem a se estabilizar, mas no conjunto de validação essas curvas possuem um comportamento mais instável.

V. CONCLUSÃO

O *transfer learning* é uma técnica poderosa para treinar o modelo sem necessitar o inicializar o modelo randomicamente, isto permite que o modelo convirja mais rapidamente, considerando que os pesos iniciais consegue extrair *features*. Além

disso, treinar o modelo com pesos inicializados aleatórios em um conjunto de dados pequenos dificulta a convergência em um ponto ótimo, como mostra no modelo 4, em que o modelo possui apenas 10% de acurácia, ver Tabela I. Nos modelos 1, 2 e 3 comparamos o eficiência de congelar os pesos das camadas iniciais, e concluímos que congelar as camadas pode aumentar a velocidade de treinamento, entretanto a acurácia diminui em comparação ao modelo 4, que apenas utiliza os pesos como inicialização.

REFERÊNCIAS

- [1] Transfer Learning. Disponível em : <<http://cs231n.github.io/transfer-learning/>>. Acesso em: 23. nov. 2018.