

Transfer Learning

Farzin Shams
farzinshams95@gmail.com

INTRODUÇÃO

Nesta tarefa, temos como objetivo aplicar a técnica de transfer learning para classificar o dataset *Cifar-10*, que contém 60000 imagens de 10 classes diferentes. Nesta técnica, usa-se uma rede neural previamente treinada num dataset para servir de ponto de partida para a otimização desta mesma rede em outro dataset. Isto é feito separando-se as camadas da rede em dois conjuntos: um onde os pesos devem ser otimizados, e outro em que devem permanecer fixos. A ideia por trás deste método é aproveitar uma parte do conhecimento e capacidade de extração de características da rede previamente treinada e aplicá-los em outro problema parecido, diminuindo o ônus do treinamento a ser feito subsequentemente.

Para examinar a influência do uso de diferentes números de parâmetros a serem treinados, testamos três versões da mesma rede, em ordem crescente de parâmetros treináveis. Espera-se que, à medida que a quantidade de parâmetros treináveis aumenta, a acurácia na predição final no conjunto de teste também aumenta, ao preço do aumento do custo computacional.

PROCEDIMENTO EXPERIMENTAL

A rede inicial a ser usada é denominada SqueezeNet, e contém 727,626 parâmetros treináveis. Como ela foi treinada no dataset *ImageNet*, que contém 1000 classes, deve-se modificar as camadas de saída das redes de forma que a saída da função softmax seja de tamanho 10, correspondente às 10 classes do dataset *Cifar-10*. Todos os modelos tiveram como função de custo a entropia cruzada, que é a comumente usada em problemas de classificação multinomial, e o otimizador *adam*. Eles foram treinados com batch size de 32, num número de épocas o suficiente para estabilizar a acurácia no conjunto de validação. O conjunto de treinamento tem 50000 amostras, 20% dos quais (10000) foi usado para validação. Os 10000 restantes pertencem ao conjunto de teste. A arquitetura final usada foi igual àquela para qual a acurácia no conjunto de validação foi maximizada.

A primeira rede a ser *fine-tuned* é dada pela Fig. 1, em que as únicas três modificações são nas últimas três camadas, de forma a modificar o modelo para classificar 10 classes, como mencionado anteriormente. As camadas modificadas são: *NEW CONV 10*, *GLOBAL AVG POOL*, *SOFTMAX*, e

o total de parâmetros ajustáveis são 5,130, equivalente a aproximadamente 0.7% do total.

A Fig. 2 mostra a curva do *log loss* do conjunto de treinamento e validação em função do número de épocas. Devido ao uso de batches, o otimizador é modificado a cada batch, fazendo com que o modelo requiera menos épocas para estabilizar. O erro final no conjunto de teste foi igual a 1,733, e a acurácia igual a 38,4%.

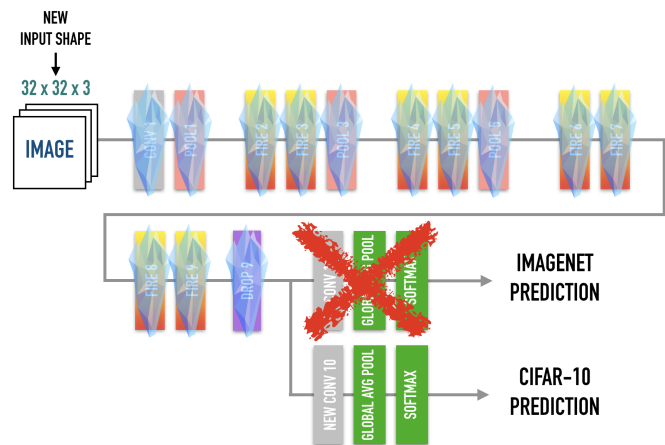


Figura 1. Modelo 1

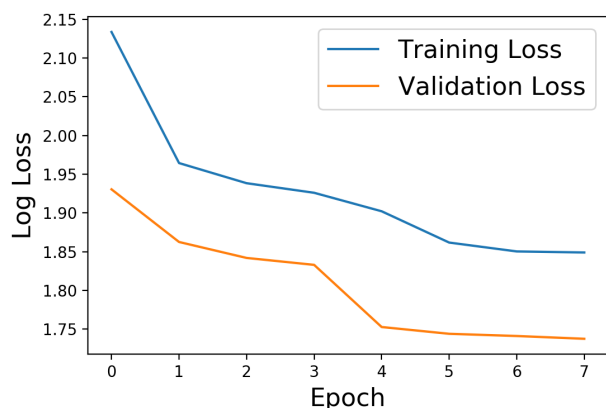


Figura 2. Modelo 1

A segunda rede a ser *fine-tuned* é dada pela Fig. 3, em que modificou-se, além das três camadas finais, mais três camadas antes destas: a *fire 8*, *fire 9* e *drop 9*, aumentando o número

de parâmetros treináveis para 391,306, equivalente a 54% do total. A Fig. 4 mostra a curva do *log loss* deste modelo. O erro final o conjunto de teste foi igual a 1,485, e a acurácia igual a 49,4%.

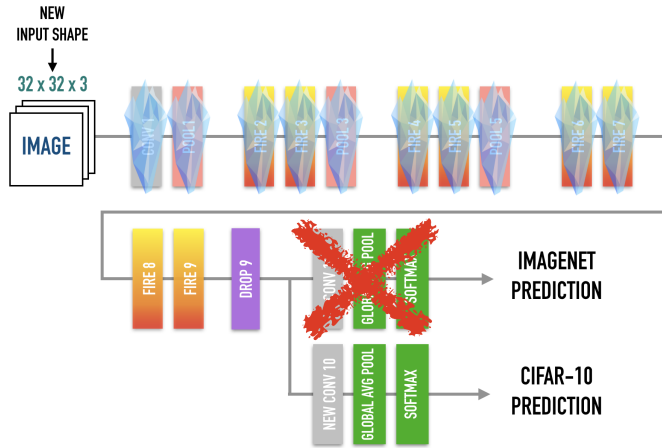


Figura 3. Modelo 2

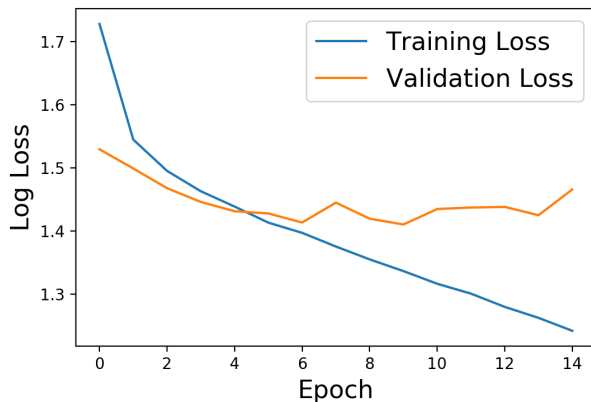


Figura 4. Curva log loss, modelo 2

Finalmente, o último modelo é dado pela Fig. 5, em que todos os 727,626 parâmetros foram treinados. A Fig. 6 mostra a curva do *log loss* deste modelo. O erro final o conjunto de teste foi igual a 1,085, e a acurácia igual a 67,7%.

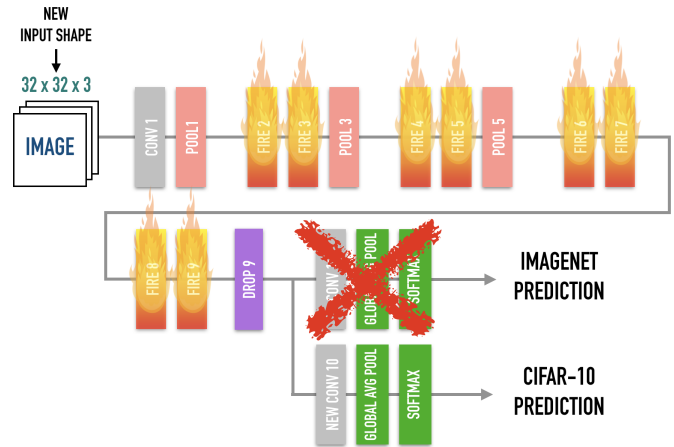


Figura 5. Modelo 3

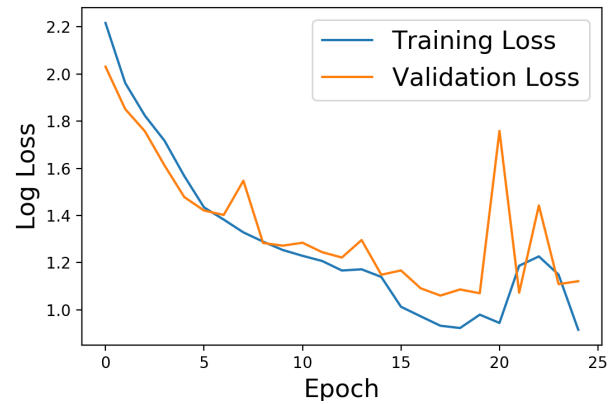


Figura 6. Curva log loss, modelo 3

Para todas as três estruturas, o modelo final escolhido para prever o conjunto de teste é aquele para o qual a acurácia no conjunto de validação foi maximizada. Este modelo foi obtido pelo *callback* to keras denominado ModelCheckpoint onde, a cada batch, o modelo que maximiza tal acurácia é salvo.

CONCLUSÃO

Como esperado, pudemos observar que, quanto maior o número de parâmetros ajustáveis, melhor a qualidade do modelo final. Porém, considerando a facilidade de treinar modelos com poucos parâmetros ajustáveis, a técnica de transfer learning mostra-se uma alternativa interessante para escolher um modelo onde o *trade-off* entre acurácia final e custo computacional compense para o problema a ser resolvido.