Zixuan He

2425131518

April 1, 2020

# Homework 4 Report

1. Finetune a model pretrained on ImageNet (e.g, ResNet50). Frameworks nowadays provide easy access to those, refer to documentations online.
Finetuned 'Xception' model with freezing weights with the rest Dense layers trainable.
Without Frozen, the training time per epoch is around 2h.
With Frozen, the training time per epoch is around 50min.

2. Construct a model of your own and start training from scratch.
Mobilenetv2 from scratch since it is a mature model and has been proven to work well. Its parameters are 14M, which could be easier to train.
The training time per epoch is around 1h.

3. Compare these two models and record the results. What is the advantage of using a finetuned model? What's the difference between the learning rates when you apply these two learning strategies (i.e finetuning vs from scratch)?
From the training progress, the finetuned model can be easily over-fitted, since the finetuned part has already trained to extract efficient features for the output. The model reaches maximum valid accuracy within a single epoch and overfits afterwards.
Training scheme
Adam as optimizer with learning rate of 0.001
# If layers are not frozen, the model behaves better.

For training from scratch, I had implemented mobilenetv2 from noise. The valid accuracy starts growing from a small number and climbing towards summit. In the process I perform several interruption and adjust the hyper-parameters.
1~10 epoches, SGD optimizer with learning rate of 0.005 with decay of 1e-6.
1~12 epoches, Adam optimizer with learning rate of 0.001.

Bonus:
Make outfit compatibility prediction based on pairwise predictions (i.e, average over n(n-1)/2 pairwise scores and then set a threshold for outfit compatibility)
The model structure is mobilenetv2 Siamese network. It takes 2 input image at the same time.
# I try to use transfer learning after my first try, while trying to modify hyper-parameters a little bit. However, the history was covered.

output format:
Undecided 0.4731 217320763_1 217320763_2 217320763_3 217320763_4
Positive 0.6238 225331161_1 225331161_2 225331161_3
Negative 0.4338 225333683_1 225333683_2 225333683_3
Conclusion compatibility_level file_info[… … …]

The threshold is set to be 0.05, which is to say
Conclusion pairwise_prediction
Negative [0,0.45]
Undecided (0.45,0.55]
Positive (0.55,1]

Extra Bonus:

• Perform learning rate scheduling

The only change for my learning rate is on my own model, where for the first 10 epoches I set the decay to be 1e-6. The other time I used Adam with learning rate = 1.

• Perform data augmentation to increase robustness

There are not extra space and time for augmentation, otherwise an epoch would cost too much time.

• Perform hard-negative mining for pairwise compatibility prediction

Sorry, but I don't quite get the idea. I think it is for object detection and I don't know how to label the hard-negative while training. I will be pleased if you could offer the Keras code for it. There are not much Keras code on google.