Fine-tuning methods review for neural networks.

R Neychev

December 21, 2017

With given some specific dataset for

There are a lot of pre-trained neural networks available for almost all popular frameworks (mostly, these are neural networks designed for computer vision problem):

- Caffe: Model Zoo
- Torch: LoadCaffe
- MxNet: MxNet Model Gallery
- Keras: Keras Application
- TensorFlow:
  - VGG16
  - Inception V3
  - ResNet

  All links are actual at the moment, for better experience search for pre-trained models on the web. Source blog page: [1]

Fine tuning is strongly connected with transfer learning problem. There are valuable results presented in [2]. In accordance to the source above, the main point is the size of the new dataset, and its similarity to the original dataset. Deep neural networks early layers has a quality of extracting more generic features [3], and more dataset-specific in later layers. There are, actually, four main scenarios:

1. New dataset is smaller in size and similar in content compared to original dataset: If the data is small, it is not a good idea to fine-tune the DCNN due to overfitting concerns. Since the data is similar to the original data, we expect higher-level features in the DCNN to be relevant to this dataset as well. Hence, the best idea might be to train a linear classifier on the CNN-features.

2. New dataset is relatively large in size and similar in content compared to the original dataset: Since we have more data, we can have more confidence that we would not over fit if we were to try to fine-tune through the full network.

3. New dataset is smaller in size but very different in content compared to the original dataset: Since the data is small, it is likely best to only train a linear classifier. Since the dataset is very different, it might not be best to train the classifier from the top of the network, which contains more dataset-specific features. Instead, it might work better to train a classifier from activations somewhere earlier in the network.

4. New dataset is relatively large in size and very different in content compared to the original dataset: Since the dataset is very large, we may expect that we can afford to train a DCNN from scratch. However, in practice it is very often still beneficial to initialize with weights from a pre-trained model. In this case, we would have enough data and confidence to fine-tune through the entire network.

Besides classical methods of fine tuning such as using all except several last ones neural network's layers as feature generator or freezing first layers' weights and tuning only last ones, new approaches also exist. An extra network, working in parallel with the source one in proposed in the article [4]. Tunning second network with applying the source one's outputs (from all or only certain layers) cause valuable increase of quality in some tasks.

Very novel approach is proposed by our colleagues from Skoltech University: using prior for image recovery/hyperresolution and other problems looks very promising and a bit unreal at the moment. The source paper is referred below as [5]

Fine tunning is also used in other areas of Deep Learning tasks: Natural Language Processing [6], Reinforcement Learning [7], Sequence (music) generation [8].

References

[1] Keras fine tunning guide (Link), 10 2016.

[2] Transfer learning and fine-tunning deep convolutional neural networks (Link), 8 2016.

[3] Thomas Käster Thomas Martinetz Lars Hertel, Erhardt Barth. Deep convolutional neural networks as generic feature extractors. 2015.

[4] Kyle Shaffer et. al. Aryk Anderson. Beyond fine tuning: A modular approach to learning on small data. In *ICLR*, 2017.

[5] Victor Lempitsky Dmitry Ulyanov, Andrea Vedaldi. Deep image prior (Link).

[6] Rui Yan et. al. Lili Mou, Zhao Meng. How transferable are neural networks in nlp applications? (Link). 2016.

[7] Ronald S. Fearing Sergey Levine Anusha Nagabandi, Gregory Kahn. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning (Link). 2017.

[8] Dzmitry Bahdanau et al. Natasha Jaques, Shixiang Gu. Sequence tutor: Conservative fine-tuning of sequence generation models with kl-control (Link). 2016.