Maximizing Feature Extraction in DenseNet for Transfer Learning

Introduction

Deep learning relies on large datasets for its success in image classification. One technique for handling smaller datasets is transfer learning, where a network is pre-trained on a large dataset and then adapted to the target dataset [1]. Transfer learning could also be useful for less curated online datasets, such as eBird, where many classes have fewer examples than the common classes [2].

The idea behind transfer learning is that earlier layers learn general features that are combined and specialized in the later layers specifically to the current dataset [1]. To adapt to a different dataset, the latter layers are replaced and re-trained. As opposed to only learning the minimal features for the task we would like to force the original network, by increasing the data complexity, to learn many different independent features in the earlier layers. By learning diverse features, it should improve target dataset performance.

We propose that by randomly mapping the existing class labels $k \in \{1, \dots, k\}$ into two new classes $\{0, 1\}$ (and periodically reshuffling), the network will learn many general features of these objects so that it can map unrelated classes to the same label. [3] found that a network learned the data patterns even when randomizing the individual example labels, which is possibly harder than our task. Although the final layer supervision of previous layers in a generic DenseNet probably makes the early layer features less useful for transfer learning (which we also want to investigate), we suspect that DenseNet's lack of redundancy is well suited for learning many independent features when forced, like above.

Prior Work

Yosinski et al. [1] classically studied the transferability of a layer's features as a function of layer depth in a network trained on ImageNet and found that generality decreased with depth. Kruithof et al. [4] later used the CIFAR-10 dataset to highlight its effectiveness on smaller data-sets. We want to understand how to optimize the features extracted for transfer learning.

Approach

We will start by replicating transfer learning of Yosinski et al. [1] with ImageNet using DenseNet. Then we'll try to improve performance by randomly re-assigning the classes to two new classes as described. If successful we'll apply it to the eBird dataset for the classes with fewer examples. Finally, if time permits we'll investigate the relationship between transferability and layer depth, learning rate in generic DenseNet.

Evaluation

- To evaluate our work we'll measure the accuracy on the ImageNet and eBird classes learned using transfer learning and compare it to the state of the art. Both eBird and ImageNet are publicly available datasets.
- We have an NVIDIA Quadro M4000 (8GB) and may use Amazon EC2 GPUs depending upon requirement.

References

[1] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?,” in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 3320–3328.

[2] G. V. Horn and P. Perona, “The Devil is in the Tails: Fine-grained Classification in the Wild,” *SciRate*, Sep. 2017.

[3] D. Arpit *et al.*, “A Closer Look at Memorization in Deep Networks,” *ArXiv170605394 Cs Stat*, Jun. 2017.

[4] M. C. Kruithof, H. Bouma, N. M. Fischer, and K. Schutte, “Object recognition using deep convolutional neural networks with complete transfer and partial frozen layers,” presented at the Optics and Photonics for Counterterrorism, Crime Fighting, and Defence XII, 2016, vol. 9995, p. 99950K.