Cluster Optimization for Specialist Networks

10 November 2015

Sébastien Arnold

Abstract

With the recent advances in deep neural networks, several experiments involved the generalist-specialist paradigm for classification. However, until now no formal study compared the performance of different clustering algorithms for class assignment. In this paper we perform such a study, suggest slight modifications to the clustering procedures, and propose a novel algorithm designed to optimize the performance of of the specialist-generalist classification system. Our experiments on the CIFAR-10 and CIFAR-100 datasets allow us to investigate situations for varying number of classes similar similar data.

Introduction

Designing an efficient classification system using deep neural networks is a complicated one, which often use a multitude of models arranged in ensembles. [Dieleman Planktons, VGG Imagenet] Those ensembles often lead to state-of-the-art results on a wide range of different tasks such as image classification [Latest Imagenet], speech recognition [Deep speach Baidu], and machine translation. [???] Those ensembles are trained independently and in parallel, and different techniques can be used to merge their predictions.

An more structured alternative to ensembling is the use of the specialist-generalist framework. As described by [Bochereau & al. (1990)], a natural analogy can be rises from the medical field; a patient first consults a general practitioner which provides an initial diagnosis which is then refined by one or several specialists. In the case of classification, the practitioners are replaced by neural networks and the final prediction is a combination of the specialists, and may or may not include the generalist's output.

In recent years, generalist and specialists have been studied under different circumstances. In particular [Hinton & al. (2014)] used specialists to create an efficient image classifier for a large private dataset. The final predictions of the

specialists were then used to train a reduced classifier that achieved performance similar to the whole ensemble. [Kahou & al. (2015)] describe a multimodal approach for emotion recognition in videos, based on specialists. Maybe closer to our work, [Warde-Farley & al. 2015] added "auxiliary heads" (acting as specialists) to their baseline network, using the precomputed features for both classification and clustering. They also underlined one of the main advantages of using specialists; a relatively low (and parallelizable) additional computational cost for increased performance.

Clustering Algorithms

In order to assign classes to the specialists networks, we compare several clustering algorithms on the confusion matrix of the outputs of the generalist. This confusion matrix is computed on a held-out partition of the dataset. Following previous works, we started by considering two baseline clustering algorithms, namely Lloyd's K-Means algorithm [(Find good reference)] and Spectral clustering, according to the formulation of [Ng & al. (2002)]. In addition to those baseline algorithms, we evaluate the performance of two novel procedures specifically designed to improve the generalist-specialist paradigm. Those algorithms are described in the following paragraphs, and pseudo code is given in [Figures TODO].

We also experiment with different was of building a confusion matrix. Besides the usual way (denoted here as *standard*) we tried three alternatives:

TODO: Find better word for example

- *soft sum*: for each prediction, we use the raw model output instead of the one-hot multi-class output,
- *soft sum pred*: just like *soft sum*, but only add the example to the confusion matrix, if the class was correctly predicted,
- *soft sum not pred*: similarly to *soft sum pred*, but only if the example was incorrectly predicted.

As discussed in later sections, the influence of the confusion matrix is minimal. Nonetheless we include them for completeness purposes.

Both of our clustering algorithms further modify the confusion matrix A by computing $CM = \mathbf{A}^\top + \mathbf{A}$, which symmetrizes the matrix. We define the entries of the matrix to be the *animosity score* between two classes; given classes a and b, their animosity score is found at $CM_{a,b}$. We then initialize each cluster with non-overlapping pairs of classes yielding maximal animosity score. We then greedily select the next classes to be added to the clusters, according to the following rules:

- In the case of *greedy single* clustering, a single class maximizing the overall animosity score is added to the cluster yielding the largest averaged sum of animosity towards this class. This partitions the classes in clusters, building on the intuition that classes that are hard to distinguish should be put together.
- In the case of *greedy pair* clustering, we follow the same strategy as in *greedy single* clustering but act on pair of classes instead of single classes. In this case we allow the clusters to share elements, and thus specialists can ahve overlapping jdugements.

This process is repeated until all classes have been assigned to at least one cluster.

Experiments

We investigate the performance of the aforementioned algorithms on the CIFAR-10 and CIFAR-100 datasets (Krizhevsky & al, 2009). Both datasets contain similar images, partitioned in 45'000 train, 5'000 validation, and 10'000 test images. They contain 10 and 100 classes respectively. For both experiments we train the generalist network on the train set only, and use the validation set for clustering purposes. As we are interested in the clustering performance we did not augment nor pre-process the images. Note that when trained on the horizontally flipped training and validation set our baseline algorithm reaches 10.18% and 32.22% misclassification error, which is competitive with the current state-of-the-art presented in [Springenberg & al, 2015].

TODO: Asterisk here, saying that the code is available online.

Following [Courbariaux & al, 2015], the baseline network is based on the conclusions of [Simonyan & al, 2015] and uses three pairs of batch-normalized convolutional layers, each followed by a max-pooling layer, and two fully-connected layers. The same model is used for specialists, whose weights are initialized with the trained weights of the generalist. One major departure from the work of [Hinton dark knowledge] is that our specialists are predicting over the same classes as the generalist, ie given a cluster we do not merge all classes outside of the cluster into a unique one. With regards to the generalist, the specialist is only biased towards a subset of the classes, since it has been fine-tuned to perform well on those ones.

CIFAR-10

TODO: Write this section with updated results.

For CIFAR-10 experiments, we considered up to five clusters, and all of the possible combinations of confusion matrix and clustering algorithms.

Table of Results

Point out interesting results don't mention ineficacity of small training set. Say that for small amount of classes, specialists seem useless, and other techniques (algorithmic ?) should be considered.

Results	standard	soft sum	soft sum pred	soft sum not pred
spectral	(0.7342, 2)	(0.4117, 3)	(0.4541, 4)	(0.4143, 2)
greedy singles	(0.2787, 3)	(0.2774, 2)	(0.3869, 4)	(0.2727, 2)
kmeans	(0.8037, 2)	(0.8037, 2)	(0.8034, 2)	(0.804, 2)
greedy pairs	(0.8584, 3)	(0.8483, 3)	(0.8473, 3)	(0.8611, 3)

CIFAR-100

TODO: Write this section with updated results.

Results	standard	soft sum	soft sum pred	soft sum not pred
spectral	(0.5828, 2)	(0.0, None)	(0.0, None)	(0.0, None)
greedy singles	(0.3834, 2)	(0.0, None)	(0.3803, 2)	(0.3551, 2)
kmeans	(0.5908, 2)	(0.0, None)	(0.0, None)	(0.0, None)
greedy pairs	(0.6141, 6)	(0.0, None)	(0.6111, 6)	(0.607, 6)

Table of results

Discuss results, (cm useless, pairs better) nothing about poor performance vs baseline, spec works better than cif10,

Conclusion and Futur Work

Larger dataset (as in Hinton Dark Knowledge and self informed or imagenet) would lead to better results, because we can afford to take images without hurting the train score.

Interesting how some specialists are only biased, not exclusively trained. Funny that this and overlapping works better than exclusive.

Add that normalizing would only help if the train distribution is different than test distribution

Acknowledgments

We would like to thank Greg Ver Steeg, Gabriel Pereyra, and Oriol Vinyals for their comments and advices. We also thank Nervana Systems for providing GPUs as well as their help with neon, their deep learning framework.

References