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 & al. 2015, Simonyan & Zisserman, 2015) Those ensembles often lead to state-of-the-art results on a wide range of different tasks such as image classification (Szegedy & al. 2014), speech recognition (Hannun & al. 2014), and machine translation. (Sutskever & al. 2014) 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 & Bourgine (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

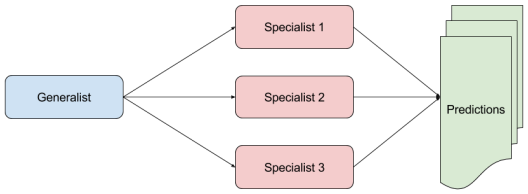

Figure 1: An example of specialist architecture with three specialists

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 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 the Appendix.

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

- *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 prediction output to the confusion matrix, if the class was correctly predicted,
- *soft sum not pred*: similarly to *soft sum pred*, but only if the prediction output 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, 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). [1]

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 & al. (2014) 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

[1] The code for those experiments, is freely available online at github.com/seba-1511/specialists.

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

For CIFAR-10 experiments, we considered up to five clusters, and all of the possible combinations of confusion matrix and clustering algorithms. The results for this experiments are reported in Table 1.

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)

Table 1: Experiment results for CIFAR-10

Interestingly, the choice of confusion matrix has only a limited impact on the overall performance, indicating that the enphasis should be put on the choice of the clustering algorithm. We notice that clustering with greedy pairs consistantly yields better scores. However none of the spcialist experiments are able to improve on the baseline, indicating that specialists might not be as efficient when dealing with a small number of classes.

CIFAR-100

For CIFAR-100 we performed the exact same experiment as for CIFAR-10 but considered using more specialists, the largest experiments involving 28 clusters. The results are shown in Table 2.

Results	standard	soft sum	soft sum pred	soft sum not pred
spectral	(0.5828, 2)	(0.5713, 2)	(0.5755, 2)	(0.5795, 3)
greedy singles	(0.3834, 2)	(0.3733, 2)	(0.3803, 2)	(0.3551, 2)
kmeans	(0.5908, 2)	(0.5618, 2)	(0.5820, 3)	(0.5876, 2)
greedy pairs	(0.6141, 6)	(0.5993, 6)	(0.6111, 6)	(0.607, 6)

Table 2: Experiment results for CIFAR-100

Similarly to CIFAR-10, we observe that greedy pairs clustering outperforms the other clustering techniques, and that the different types of confusion matrix have a limited influence on the final score. We also notice that fewer clusters work better than a lot of them. Finally and unlike the results for CIFAR-10, some of the specialists are able to improve upon the generalist, which confirms our intuition that specialists work better when a large number of output classes is involved.

Our explanation for the improved performance of greedy pairs is the following. Allowing clusters to overlap leads to the assignment of difficult classes to multiple specialists. At inference time, this means that more networks will influence the final prediction which is analogous to building a larger ensemble for difficult classes.

Conclusion and Futur Work

We introduced a novel clustering algorithm for the specialist-generalist framework, which is able to consistantly outperform other techniques on the same task. We also provided a preliminary study of the different factors coming into play when dealing with specialists, and concluded that the choice of confusion matrix from our proposed set only has little impact on the final classification outcome.

Despite our encouraging results with clustering techniques, no one of our specialists-based experiments came close to compete with the generalist model trained on the entire train and validation set. This was a surprising outcome and we suppose that this effect comes from the size of the datasets. In both cases, 5'000 images corresponds to 10% of the original training set and removing that many train examples has a drastic effect on both generalists and specialists. All the more so since we are not using any kind of data augmentation techniques, which could have moderated this downside. An obvious futur step is to validate the presented ideas on a much larger dataset such as Imagenet (Russakovsky & al. 2014) where splitting the train set would not hurt the train score as much.

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

Bochereau, Laurent, and Bourgine, Paul. A Generalist-Specialist Paradigm for Multilayer Neural Networks. Neural Networks, 1990.

Courbariaux, Matthieu, Bengio, Yoshua, and David, Jean-Pierre. BinaryConnect: Training Deep Neural Networks with Binary Weights during Propagations. NIPS, 2015.

Dieleman, Sander, Willett, Kyle W., and Dambre, Joni. Rotation-invarient convolutional neural networks for galaxy morphology prediction. Oxford Journals, 2015.

Hannun, Awni, Case, Carl, Casper, Jared, Catanzaro, Bryan, Diamos, Greg, Elsen, Erich, Prenger, Ryan, Satheesh, Sanjeev, Sengupta, Shubho, Coates, Adam, and Ng, Andrew Y. Deep Speach: Scaling up end-to-end speech recognition. Arxiv Preprint, 2014.

Hinton, Geoffrey E., Vinyals, Oriol, and Dean, Jeff. Distilling th Knowledge in a Neural Network. NIPS 2014 Deep Learning Workshop.

Kahou, Samira Ebrahimi, Bouthiller, Xavier, Lamblin, Pascal, Gulcehre, Caglar, Michalski, Vincent, Konda, Kishore, Jean, Sébastien, Froumenty, Pierre, Dauphin, Yann, Boulanger-Lewandowski, Nicolas, Ferrari, Raul Chandias, Mirza, Mehdi, Warde-Farley, David, Courville, Aaron, Vincent, Pascal, Memisevic, Roland, Pal, Christopher, and Bengio, Yoshua. EmoNets: Multimodal deep learning approaches for emation recofnition in video. Journal on Mutlimodal User Interfaces, 2015.

Krizhevsky, Alex. Learning Multiple Layers of Features from Tiny Images. 2009.

Ng, Andrew Y., Jordan, Micheal I., Weiss, Yair. On spectral clustering: Analysis and an algorithm. NIPS 2002.

Russakovsky, Olga, Deng, Jia, Su, Hao, Krause, Jonathan, Satheesh, Sanjeev, Ma, Sean, huang, Zhiheng, Karpathy, Andrej, Khosla, Aditya, Bernstain, Michael, Berg, Alexander C., and Fei-Fei, Li. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision, 2015.

Simonyan, Karen and Zisserman, Andrew. Very Deep Convolutional Networks for Large-Scale Image Recognition. International Conference on Learning Representations, 2015.

Springenberg, Jost Tobias, Dosovitskiy, Alexey, Brox, Thomas, and Riedmiller, Martin. Striving for Simplicity: The All Convolutional Net. International Conference on Learning Representations Workshop, 2015.

Sutskever, Ilya, Vinyals, Oriol, and Le, Quoc V. Sequence to Sequence Learning with Neural Networks. Arxiv Preprint, 2014.

Szegedy, Christian, Liu, Wei, Jia, Yangqing, Sermanet, Pierre, Reed, Scott, Anguelov, Dragomir, Erhan, Dumitru, Vanhoucke, Vincent, and Rabinovich, Andrew. Going deeper with convolutions. Arxiv Preprint, 2014.

Warde-Farley, David, Rabinovich, Andrew, and Anguelov, Dragomir. Self-Informed Neural Networks Structure Learning. International Conference on Representations Learning, 2015.

Appendix

Greedy Pairs Pseudo Code

1. Given a confusion matrix M and N desired clusters.
2. $M \leftarrow M + M^T$
3. Initialize N clusters with non-overlapping pairs maximizing the entries of M.
4. Until each class has been assigned at least once:
   - Get the next pair maximizing the entry in M
   - Find which cluster minimizes the sum of animosity of both classes.
   - Assign both classes to this cluster
5. Return the clusters

Note: A python implementation of both greedy pairs and greedy single can be found at http://www.github.com/seba-1511/specialists.