MEDIATED EXPERTS FOR DEEP CONVOLUTIONAL NETWORKS

Sebastian Agethen *Winston H. Hsu*

National Taiwan University
Graduate Institute of Networking and Multimedia

ABSTRACT

We present a new supervised architecture termed Mediated Mixture-of-Experts (MMoE) that allows us to improve classification accuracy of Deep Convolutional Networks (DCN). Our architecture achieves this with the help of expert networks: A network is trained on a disjoint subset of a given dataset and then run in parallel to other experts during deployment. A *mediator* is employed if experts contradict each other. This allows our framework to naturally support incremental learning, as adding new classes requires (re-)training of the new expert only. We also propose two measures to control computational complexity: An early-stopping mechanism halts experts that have low confidence in their prediction. The system allows to trade-off accuracy and complexity without further retraining. We also suggest to share low-level convolutional layers between experts in an effort to avoid computation of a near-duplicate feature set. We evaluate our system on a popular dataset and report improved accuracy compared to a single model of same configuration.

Index Terms— Deep Learning, Incremental Learning, Mixture of Experts

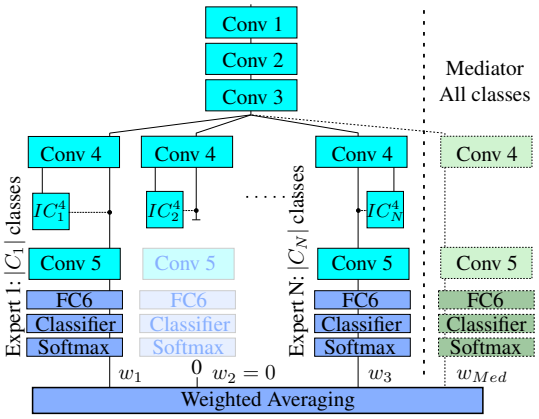


Fig. 1. N experts are trained on their corresponding *superclasses* C_i. At inference time, *confidence* is measured in *confidence modules* IC_i^j. In this example, expert two is stopped at layer `Conv 4` due to low confidence and its activations are set to zero. A *mediator* with full classifier improves accuracy of the system should more than one expert remain.

1. INTRODUCTION

Deep learning methods in general, and *Deep Convolutional Neural Networks* (DCN) in particular, have seen a surge in popularity among researchers over the past decade or so. While the application of early DCN was limited to simple tasks such as hand-written digit recognition [5], modern state-of-the-art methods can meet or exceed human-level performance on far more complex tasks, including generic image classification [2, 4, 7]. Two factors that have contributed to this development are the availability of large-scale image corpora like *ImageNet* [6], and the wide-spread availability of GPU-based computing hardware which renders computationally expensive model training procedures feasible.

Experts systems are known to improve the classification accuracy of neural networks even further and have been studied extensively. Previous work on *Mixture of Experts* (MoE) [3] is however flawed in two aspects: First, computational complexity in MoE is a multiple of that of a traditional model. To this end, we propose early-stopping of experts. Our mechanism evaluates each expert's *confidence* and is influenced by a hyper-parameter, allowing a trade-off between complexity and classification accuracy without the need for retraining. Second, as recently pointed out in [8], training becomes more difficult as datasets are getting larger. Intuitively, we expect that an expert has extensive knowledge on a limited area of expertise, i.e., a subset of the training data, and thereby naturally avoids the issue. Previous MoE systems are trained on full datasets, however, as difficulties arise when a decision is needed on which expert to trust. To remedy this issue, we introduce a *mediator*, see Figure 1. The mediator has a full classifier, allowing it to arbitrate conflicting predictions when multiple experts are confident.

In this paper we propose a new architecture which we name *Mediated Mixture of Experts* (MMoE) that aims to increase the classification accuracy of a general DCN. We briefly discuss related research in Section 2 before formally defining our method in Section 3. We follow by summarizing

This work is supported in part by grants MOST 104-2622-8-002-002 and MediaTek.

the outcome of our experimental evaluation in Section 4 and give some concluding remarks in Section 5.

2. RELATED WORK

A great deal of literature on deep learning and in particular on DCN has been published in recent years. Image classification with deep learning has profited from the availability of large datasets (in terms of both number of classes and images). In this paper, our choice falls on the popular ImageNet [6] dataset. Both a 22K classes and a 1K classes version exist of this dataset, we evaluate our system on the latter.

AlexNet by Krizhevsky *et al*. [4] achieved a significant increase in classification performance over traditional methods on ImageNet, i.e., an top-1 accuracy of 62.5%, showing the potential of deep learning for large datasets such as ImageNet. Their success is arguably founded on three factors: Depth, Rectified Linear Units (ReLU) as non-linearity and data augmentations. We use *CaffeNet*, a variation of this model, as a baseline to evaluate our system.

A hierarchical architecture for incremental learning was presented in [8]. The authors suggested to use a *Branching* layer to determine the *superclass* a specific problem belongs to. Following this, a leaf model provided fine-grained classification. We follow their work insofar that we also propose to use *experts* (leaf models) for fine-grained classification. However, we avoid branching as it introduces a conditional error that cannot be recovered. Branching error also poses a scalability issue, as it grows with the number of superclasses. Finally, the prediction process in their work has large redundancy: Full models are used for both branching prediction and fine-grained prediction.

Mixture of experts (MoE), as first proposed in [3], have been well-known for a while. A great deal of work has been done in this area and an overview can be found in [10]. Generally, a number N of expert networks are trained together on a dataset. The experts learn discriminant features and thereby improve the overall accuracy of the system. While this can be used to improve classification results, the drawback lies in the parallel execution of all N experts, which results in longer runtime. In contrast to our proposed method, experts are trained on the full dataset and have a complete classifier. This is disadvantageous for incremental learning, a scenario in which new classes are added over time, as all experts need to be retrained on the new data.

The work of [9] discusses *generality* and *specialization* of convolutional features, giving insight into how deep learning works. The same phenomenon has also been discussed in [11] in the context of network visualization. In particular, both works show that the first few layers of a DCN produce *general* Gabor-like features, i.e., lines and blobs. It is only in the higher layers that these features become more *class-specific*. Inspired by this fact, instead of N times producing nearly identical general features, we can compute a single set of these features.

Table 1. Branching networks introduce additional error: We trained such a network with two experts on superclasses C_1, C_2 as in Sec. 4.1. Due to branching, the average accuracy on the full dataset $C_1 \cup C_2$ remains below the traditional model. However, experts on their own have encouragingly higher accuracy on their area of expertise than a traditional model, see columns C_1, C_2. Note that the accuracy shown here was achieved with a slim configuration of *CaffeNet*, i.e., the FC layers were reduced from 4096 to 512 neurons.

Name	$C_1 \cup C_2$	Class C_1	Class C_2
Branching	93.2 %	93.35%	93.04%
Expert 1	–	**46.262 %**	–
Expert 2	–	–	**59.202%**
Average	49.151%	–	–
Baseline [4]	**49.35** %	41.52 %	55.96 %

3. PROPOSED METHOD

When tackling a problem, a divide-and-conquer approach can often be helpful: In deep learning, we can train experts on small problems such that the accuracy improves. This has been used in MoE [3]: Experts are trained competitively in the hope that they automatically learn discriminant features. A gating network combines the prediction results.

Each expert's area of expertise can also be designed with help of prior knowledge. We can utilize two methods to gain this knowledge: Spectral Clustering (as done in [8]) and explicit hierarchies (such as the ImageNet dataset provides), see also Sec. 4.1. In both cases the i-th expert is then trained exclusively on the subset of data in that superclass, which we denote as C_i.

3.1. Simple Branching model

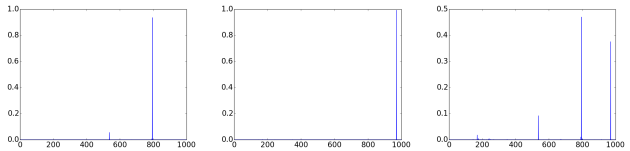
We briefly discuss a *branching network* as in [8], which predicts the superclass of an image. The branching network is a DCN such as *AlexNet* [4] with the number of outputs in the classifier reduced to N, the number of superclasses. In the *branching decision*, the expert corresponding to the highest activation is then selected to obtain a fine-grained prediction.

We trained such a system with two slim[1] experts, as seen in Table 1. Even though we used two experts only, classification accuracy suffered from large branching errors: Should the wrong expert be chosen, the error cannot be recovered. Nevertheless, we also see that single experts show superior performance on their area of expertise, confirming that we are on the right track. The model we discuss in the following is an attempt at reducing branching errors.

3.2. Branched experts with early stopping

We propose the following framework as depicted in Fig. 1: The input is passed through a number of convolutional layers

[1] By reducing the number of neurons of both layers `FC6` and `FC7` to 512, in the hope to keep parameter complexity low.



(a) (b) (c)

Fig. 2. Mediation process: Experts (a) and (b) both show strong opinions that contradict each other. Mediator (c) is able to solve the conflict (here in favor of expert (a)).

to generate *general* features, as defined in [9]. These layers are shared between all experts and could encompass the first two or three layers in the case of *CaffeNet*. Higher layer features have a higher degree of specialization, and therefore need to be finetuned for each expert. Parallel execution of these specialized layers is costly, however, in particular for large N. To this end we propose a *confidence* module, which determines the confidence of whether the expert is able to solve the problem. In particular, an expert i is deemed to have low confidence (in layer j) if for its given score s_i^j and a threshold T:

$$\max_{k \neq i} \left(s_k^j \right) - s_i^j \geq T, \quad 1 \leq i, k \leq N \tag{1}$$

In the following, we implement the confidence module simply as a fully connected layer with N outputs, trained on the complete dataset with the original labels replaced by a superclass label $l_{C_i} \in [0, N-1]$. Note the placement of the confidence module: While lower layers have too general features to compute a reliable score, early stopping in higher layers diminishes the resource saving effect. We suggest an intermediate position such as after `Conv4`.

Let $\mathbf{u}_{C_i}^j$ be the activation of the confidence module in layer j of the expert on C_i. The appropriate score is then simply the i-th component of the activation vector:

$$s_i^j = \mathbf{u}_{C_i,i}^j$$

To support our decision for a simple threshold, we run a simple branching network (as described above) on a test set, and measure the magnitude of the left-hand side of Equation (1). We then note whether the branching decision is correct or not. As is to be expected, bad branching decisions occur when this magnitude is small, while correct decisions often have higher confidence, see Table 2.

3.3. Mediator

Given that the inequality in (1) holds, an expert is deemed to have low confidence and stopped. Its corresponding activations in the final layer are set to zero. So far, the system described above performs slightly worse than a single model with same configuration. This is due to those instances, where more than one expert is active: An expert trained on a different superclass could "mistake" the input for a particular fine-grained class of his own domain, leading to conflicting opinions. In these cases, we found it helpful to add a *mediator*, that is, a slim model trained on all 1000 classes to arbitrate opinions, see Figure 2. The resulting architecture then outperforms the single model baseline.

Table 2. Mean and standard deviation of confidence scores in cases of correct and incorrect branching decisions. Incorrect decisions very often imply a low confidence value, allowing us to decide whether additional experts need to be used to solve the task.

Branching	Mean	Std.-dev.
Correct	6.1	3.44
False	1.8	1.67

Subsequently, the softmax probabilities of each expert are weighted and averaged: The slim 1000-classes mediator is only added when more than one expert is executed and weighted with w_{Med}. We set $w_{Med} = 0.6$ if executed or $w_{Med} = 0$ otherwise. Experts are weighted according to their confidence scores normalized to $[0, 1]$ and subsequently scaled by $(1 - w_{Med})$.

Our framework can also be used in the context of incremental learning. To add a new superclass C_{N+1}, we simply train a single expert on the new data. Following this, the confidence modules of other experts need to be finetuned to accommodate for the new superclass. We stress that this is significantly faster than training a full network, as the confidence module only encompasses a single layer with N outputs, rendering it very shallow and slim. In our experiments, 3 epochs were found to be sufficient. In order to update the mediator network, we can simply add extra neurons to the output and finetune the network, see the Flat Increment technique in [8].

4. EVALUATION

The evaluation is performed on the ImageNet 1K dataset. We begin with a description of how superclasses are formed.

4.1. Superclass construction

Superclasses can be defined in an automated or manual fashion: In the former case *Spectral Clustering*, as was done in [8], is a suitable choice. Alternatively, we can traverse the hierarchy available with ImageNet and join several leaf classes that are conceptually related. This provides a convenient advantage in an incremental learning scenario: We simply train one or more new experts for new data. Contrary, when using Spectral Clustering, it is necessary to retrain all of the existing experts as well.

Due to space limitations, we limit ourselves to manually defined superclasses and $N = 2$. We choose C_1 as "artifact" (with synset: n00021939) and C_2 as all remaining classes, resulting in a split of 517 vs. 483 leaf classes with roughly

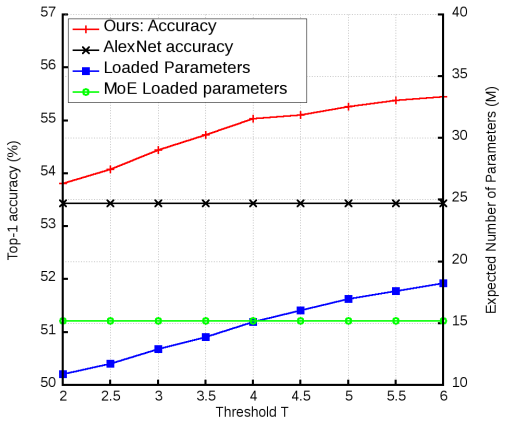


Fig. 3. Top-1 accuracy of our proposed system in relation to threshold T, here on the slim seven layer configuration with the first three layers shared. The expected number of parameters to be loaded equals that of traditional MoE systems for $T = 4$.

Table 3. Top-1 accuracy, and stopping probabilities p_1, p_2 of both experts for $T = 4$ under three different configurations.

Name	Acc.	p_1	p_2
MMoE (slim, 7L)	**0.5618**	0.2213	0.1512
MMoE (slim, 8L)	**0.5296**	0.1805	0.1545
MMoE (default)	**0.5855**	0.2225	0.17144
MMoE (slim, 7L, shared)	0.5361	0.2122	0.1801
Unmediated (slim, 7L, shared)	0.5052	0.2122	0.1801
Baseline (slim, 7L)	0.5344	–	–
Baseline (slim, 8L)	0.4948	–	–
Baseline (CaffeNet) [1]	0.5584 (0.574) [2]	–	–

660K and 620K images each. In other words, this simulates adding "artifact" data to an existing model. In the following, we refer to this as the *Hierarchical set*.

4.2. Results on Hierarchical set

Network configuration. We evaluate our system on three configurations: First, a slim configuration of *CaffeNet*, where both FC layers are reduced to 512 neurons. In our second configuration we drop layer `FC7` altogether, while reducing `FC6` to 512 neurons. Finally, our third configuration uses the default *CaffeNet* configuration with both fully connected layers containing 4096 neurons. All configurations were finetuned from the *CaffeNet* model available with Caffe, see [1].

System accuracy. MMoE improves the top-1 classification accuracy in all three configurations. The result is strongly dependent on the value of threshold T, see Figure 3. Classification accuracy peaks at $T = 6$, for which our system outperforms the baseline over 2.8% and 2.7%[2] (in the slim and default configuration respectively).

[2]Using caffe's `test` command, we measured *CaffeNet*'s top-1 accuracy at 55.84% on the validation set, the reported accuracy is 57.4% however.

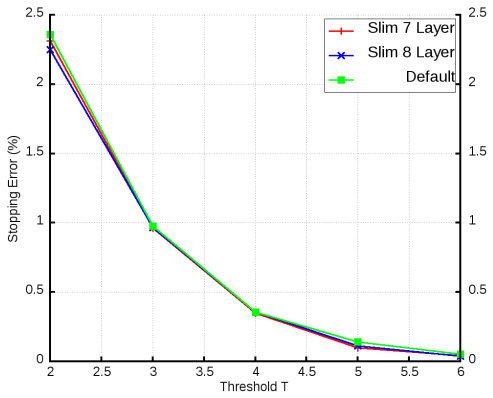


Fig. 4. Probability that the true expert is falsely stopped in relation to threshold T. The error is significantly lower than the branching error in Section 3 (6.8%).

Early stopping. Experts are rarely incorrectly stopped, see Figure 4. For $T \geq 3$, a true expert is falsely stopped in less than 1% of all cases, an improvement over the branching error discussed in Section 3. Table 3 shows the probabilities p_1, p_2 of how often experts C_1 and C_2 are stopped.

Mediator impact. To show the importance of the Mediator, we train a slim configuration lacking mediation. As such, the system only reaches a top-1 accuracy of 50.52%, about 3.1% lower than the same configuration with mediator.

Complexity vs. Threshold. The choice of T also influences how often an expert is run. We show this by evaluating the average number of parameters that need to be loaded, see Figure 3. For $T = 4$, the expected number of parameters is close to the number of parameters when using traditional Mixture-of-Experts, that is, the complexity of N experts alone. The number of computational operations, dominated by the Convolutional layers, can be significantly reduced by layer sharing as discussed earlier.

5. CONCLUSION AND FUTURE WORK

We propose a mediated expert system for deep convolutional networks that enables experts to learn on small partitions of a training set, a case given in an incremental learning scenario. In detail, experts can be stopped early, where the stopping point is controlled by a single hyper-parameter, allowing to adapt to different circumstances in terms of availability of resources. Furthermore, we avoid the branching error that occurs in prior work when training on partitioned datasets.

In order to better underline the benefits of our proposal, two points are left for our future work. First, a more thorough evaluation is necessary that was not possible in the scope of this paper. Second, we believe that the mediator concept can be developed further in order to reduce computational complexity to a higher degree.

6. REFERENCES

[1] Bvlc caffe on github. https://github.com/BVLC/caffe/tree/master/models/bvlc_reference_caffenet. Accessed: 2015-09-23.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *ArXiv e-prints*, Feb. 2015.

[3] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Comput.*, 3(1):79–87, Mar. 1991.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[5] LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in Neural Information Processing Systems*, pages 396–404. Morgan Kaufmann, 1990.

[6] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge, 2014.

[7] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[8] T. Xiao, J. Zhang, K. Yang, Y. Peng, and Z. Zhang. Error-driven incremental learning in deep convolutional neural network for large-scale image classification. In *Proceedings of the ACM International Conference on Multimedia*, MM '14, pages 177–186, New York, NY, USA, 2014. ACM.

[9] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? *CoRR*, abs/1411.1792, 2014.

[10] S. Yuksel, J. Wilson, and P. Gader. Twenty years of mixture of experts. *Neural Networks and Learning Systems, IEEE Transactions on*, 23(8):1177–1193, Aug 2012.

[11] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013.