Contextual Dropout

Finding subnets for subtasks

Sam Fok

samfok@stanford.edu

Abstract

The feedforward networks widely used in classification are static and have no means for leveraging information about their context to improve their performance. Attentional models seek to improve upon the standard feedforward network using recurrence and modifying the network's computation. This project explores a simple method of modifying the feedforward network using dropout to define a subnetwork within an existin network. Given a trained network and the task of classifying a subset of the original training classes, are there better subnetworks to than the full network?

1. Introduction

Natural images rarely consist of single objects, and objects can often be decomposed and composed into further more objects. For example, a cat is composed of a head, a body, four legs, and a tail. A cat's head has two cat eyes, two cat ears, a mouth etc.... Whether explicit or implicit, any cat classifier must answer the underlying question of: What makes a cat a cat? Outside of cats, the classification problem faces a problem of scale: How many classes of objects exist? Feedfoward artificial networks with convolutional layers are the current state-of-the-art image classifiers [5]. These networks are inspired by the hierarchical layering and localized structure of the mammalian visual cortex. While state-of-the-art classifiers are beginning to surpass human performance on such challenges as ImageNet, how can networks be scaled to handle more complicated tasks that involve attributes of objects and relations between objects [2]? Are we going to train a classifier network with a category for every single object class and combination of objects? Such a universal classifier seems implausible. Another drawback of the feedforward classifiers is that it considers all classes simulataneously. More concretely, how often are cats, beer bottles, and jet fighters present in the same scene?

There are still features of the biological visual cortex that have yet to be used effectively in artificial neural networks. Namely, the visual processing system of the cortex is not only feedforward but also contains top-down feedback pathways that modulate lower levels of the processing hierarchy [1] [4]. The goal of this project is to implement an efficient means for a top-down control mechanism to alter a network's processing in different contexts. Namely, I use dropout to improve a network's performance in the context of classifying only a subset of the classes that a network was originally trained on. Classifying this subset of classes defines a subtask. The subset of neurons that remain after dropout defines a subnetwork. For N potential dropout units, dropout only requires N bits to define a subnet, making dropout a compact means for modifying the behavior of an existing network. This project seeks to find the best subnetwork for a given subtask.

2. Background and Related Work

Models such as in [4] use recurrent networks to build internal models of the image during classification. In contrast, the method proposed in this project implements an explicit means of modifying the network based on context.

Dropout was originally proposed as a method to mitigate overfitting [6]. In class, dropout was described as a kind of regularization by randomly traversing the space of subnetworks during training. For a network of N neurons, dropout explores a space of 2^N subnetworks. However, these 2^N networks do not evolve independently as they share many of their parameters. Since its creation, dropout has been characterized as a form of adaptive regularization [7].

Transfer learning provides another method of repurposing a network for a subtask without retraining the entire network.Transfer learning retrains a subset of layers using a data set different from the training dataset and works because of the statistical similarities between natural scenes. However, even when only replacing and retraining a fraction of a many-layered networks, it still seems implausible that one would maintain separate final layer sets for every conceivable set of classes.

Layer	Specifications
Conv	3×3×32, stride 1
Pool	2×2, stride 2
Affine	128 fully connected units
Dropout	p=.5
Affine	10 fully connected units

Table 1. Model parameters

3. Approach

In this project, I search for the best dropout subnetwork for a given subtask and compare it to the base network and with the network retrained with transfer learning. The base network's structure is as follows:

Conv - ReLU - Pool - Affine - ReLU - Dropout - Affine - Softmax

The base network parameters are listed in Table 1. The base network was trained on the CIFAR-10 dataset [3] until training and validation accuracies were 0.803 and 0.717. Figure 1 visualizes the raw pixel data and the features extracted by the above network after training. The features were extracted from the output of the second ReLU layer. Although the network manages to increase the degree of separation of the features relative to the raw image data, the linear classifier at the end is still not able to generalize over the full task. The confusion matrix for the base network is shown in Figure 5, left. Based on the confusion matrix, I define the subtask used for this project as classifying birds, cats, and deer, which correspond to class indices 2, 3, and 4, respectively. These are classes for which the base network has poor accuracy.

As positioned in the network structure, the dropout layer zeros out features output from the ReLU layer before they are used as input to the final linear classifier. For the subtask, this can be interpreted as dropping features which are not beneficial or harmful for classifying the subset of classes in the subtask. To find units to drop, I use the mean gradient of the loss with respect to the each of the features on the training set and select from the units with mean positive gradient (see Figure 2). Units with mean positive gradient were sorted in descending order of gradient magnitude and cumulatively dropped one-by-one to find the optimal set of units using the validation data set (see Figure 3.

Dropout is equivalent to modifying the final affine layer's weight matrix. That is, for feature vector x, the final affine layer computes

$$y = Wx + b$$

where y is the class scores, W is the weight matrix, and b are the biases. The dropout units cause the final affine layer to compute

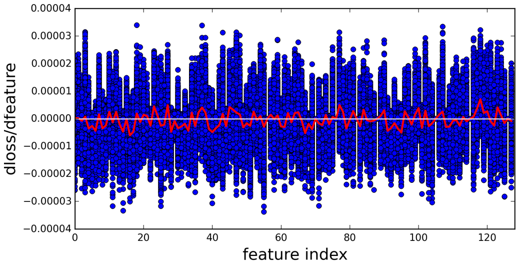

Figure 2. Gradient of the subtask loss with respect to the features extracted by the network. Each data point is a training image. No features contribute positively to the loss for all images, but the mean gradient can be used to identify units to drop. Note that some features, vertical white bands in the data cloud, are unused by the base network's final affine layer as their gradients are exactly 0 for all images.

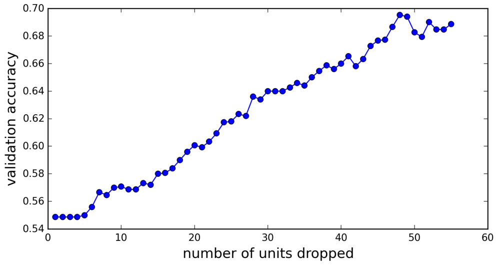

Figure 3. Gradient-based dropout: Subtask validation accuracy as units are are dropped in descending order of their mean gradient. Only units with positive mean gradient are considered for dropout. Dropping the 48 units with the highest mean loss gradient produced the subnetwork with best validation accuracy on the subtask.

$$y = WDx + b$$

where D is a diagonal matrix of 1s or 0s describing which units were dropped. D can be folded into the final affine weight matrix to produce W_D so that the dropout subnetwork effectively computes $y = W_D x + b$ in its final affine layer.

4. Experiment

The performance of the base network and dropout network were compared on the subtask. Performance on the subtask is listed in Table 2. For control, I conducted a random search over the space of dropout units and used transfer learning to retrain the last affine layer in the classifier. The performance of subnets with randomly dropped units is shown in Figure 4. The best subnet found using the mean gradient to drop units outperforms all of the subnets found by randomly dropping units. However, transfer learning

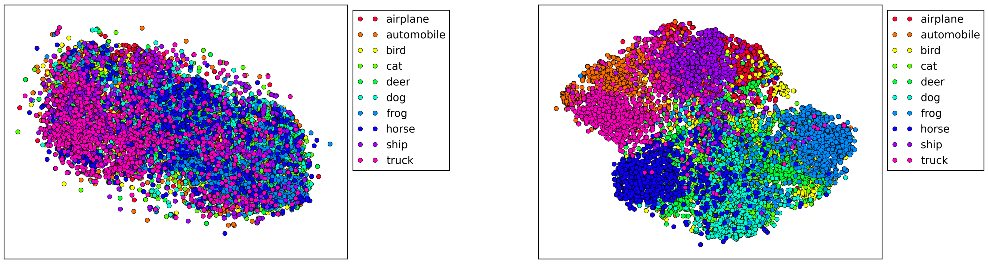

Figure 1. t-SNE visualizations of CIFAR-10 raw pixel data (left) and features extracted from the trained network (right). Although the network seems to increase separation between the classes, there is still overlap between the features after training.

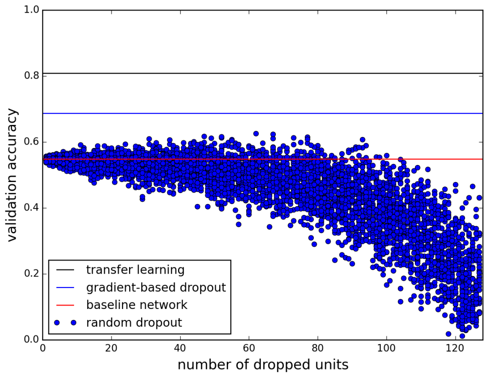

Figure 4. Performance on the subtask. Transfer learning performs best, but gradient-based dropout is able to capture part of the performance gains. Random dropout does surprisingly well and outperforms the base network in many instances.

performs the best on the subtask. That transfer learning performs the best is not so surprising as we've previously seen that the effect of dropout can be folded into the affine layer's weight matrix. Images that were misclassified by the base network but correctly classified by the dropout network and transfer learning network are shown in Figure 6.

To compare the effect of dropout and transfer learning on the base network, the weight matrix of the final affine layer for each configuration is visualized in Figure 8. The matrix singular values are shown in Figure 7. As expected, transfer learning produces a low rank weight matrix whose rank reflects the number of classes in the subtask. Likewise, the optimal dropout subnet effective weight matrix is sparser than the base network's weight matrix.

5. Conclusions

The results of this project indicate that dropout can indeed be used to modify a network and improve performance on a subtask. However, there are a number of open questions and directions the project could take in the future. Namely, what generates the top-down control signal? A full model of attention model could be comprised of a feedforward network to classify and a feedback network to modify the feedforward network's layer depending on the current estimate of the class. More broadly, even if the network were able to reconfigure itself based on a model of the current input, there remains the question of how a network could learn to create entirely new classes or combine existing classes.

Base network	Best dropout subnet	Transfer network
0.546	0.695	0.809

Table 2. Subtask classification validation accuracy. Both the subnet found with dropout and the transfer learning network outperform the base network.

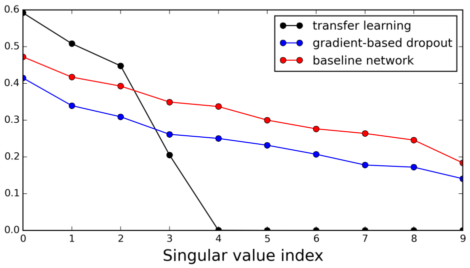

Figure 7. Singular value decompositions of the weight matrices. Note that transfer learning retraining drastically reduces the rank of the affine layer weight matrix, which makes the network classify all objects as one of the subclasses used in the subtask.

References

[1] A. Borji and L. Itti. State-of-the-art in visual attention modeling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1):185–207, Jan 2013.

[2] D. Geman, S. Geman, N. Hallonquist, and L. Younes. Visual turing test for computer vision systems. *Proceedings of the*

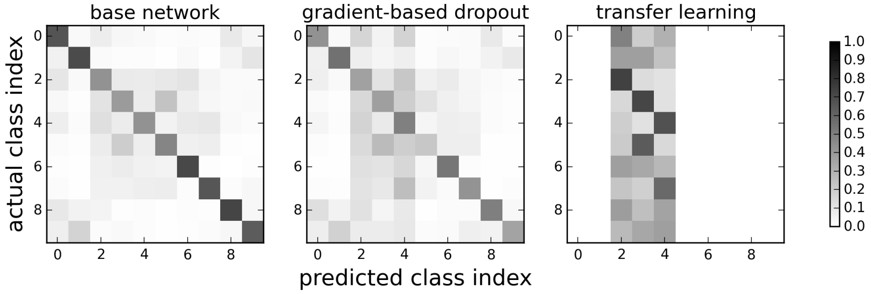

Figure 5. Confusion matrices of the base network (left), gradient-based dropout network (center), and transfer learning network (right). The greyscale is the mean softmax probability for each class. For a network performing well, all of the probability should be centered along the diagonal. Off-diagonal entries indicate confusion between classes.

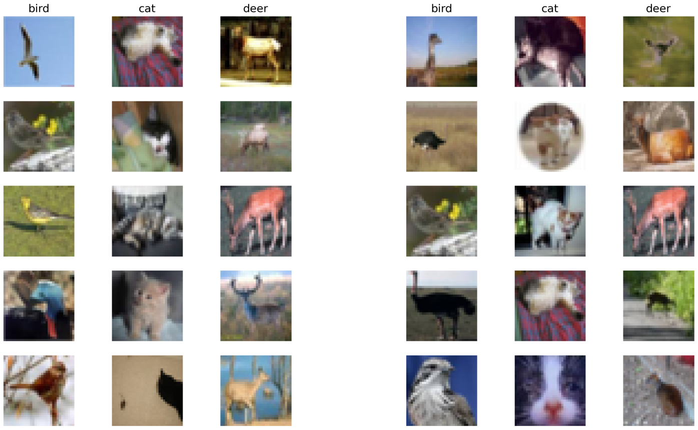

Figure 6. Images in the subtask misclassified by the base network and correctly classified by the transfer learning network (left) and dropout network (right)

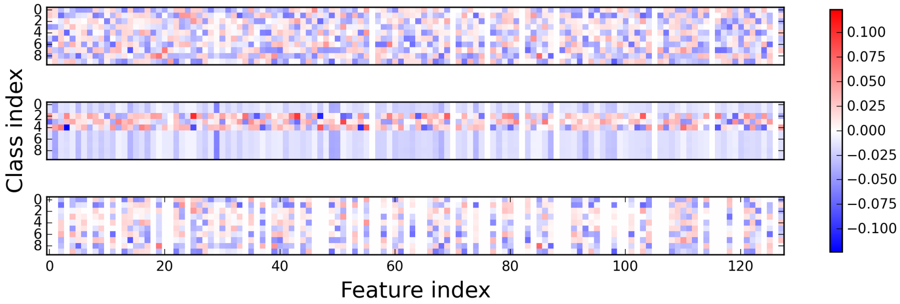

Figure 8. Weights of the last affine layer. (Top) Weights after training on all classes. (Middle) Weights after transfer learning on subtask. (Bottom) Effective weights W_D after applying dropout. Transfer learning produces a low-rank weight matrix because of the few classes used during retraining for the subtask and regularization. In contrast, dropout serves to sparsify the original base network's weight matrix as shown by the bands of white. Note that some features went unsused by the base network's weight matrix as well as the transfer learning weight matrix.

National Academy of Sciences, 2015.

[3] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[4] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu. Recurrent models of visual attention. *CoRR*, abs/1406.6247, 2014.

[5] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014.

[6] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[7] S. Wager, S. Wang, and P. Liang. Dropout training as adaptive regularization. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 351–359. 2013.