Improving Neural Networks by Preventing Co-adaptation of Feature Detectors

1 Citation

Hinton, Geoffrey E., et al. "Improving neural networks by preventing co-adaptation of feature detectors." arXiv preprint arXiv:1207.0580 (2012).

`https://arxiv.org/pdf/1207.0580.pdf`

2 Abstract

Randomly drop half of hidden units on each input to force neurons to work on their own rather than depending on other neurons. This is dropout.

3 Paper

Dropout is a kind of model averaging where the models share weights for the non-dropped hidden units. We also set an upper bound on the L2 norm of the weight vector for each hidden unit (this works better than an L2 penalty). At test time, we halve the weights because they are all turned on (instead of half turned on).

For MNIST, dropout reduces the number of test errors from 160 to 130 (or 110 if we also constrain weight L2).

Dropout also works for deep belief networks. We verify this with a deep belief net based Hidden Markov Model for speech recognition on the TIMIT dataset.

Dropout also works for CIFAR-10 and ImageNet image classification models. We use a convolutional neural net with some conv layers (with pooling and ReLU), and some fully-connected (or locally-connected) layers, and a softmax.

We also evaluated on the Reuters document classification dataset.

The 50% dropout rate works well for hidden units. If you try dropout on input units, you should use something smaller.

Dropout is simpler to implement and more compute efficient than Bayesian Model Averaging. It's like an extreme form of bagging where models share their non-dropped weights.

4 Experiments on MNIST

We use 4 nets, each with a few fully connected layers. We use dropout on hidden and visible units with rates 50% and 20%, respectively. We do L2 weight constraint and train with SGD with momentum, where both the learning rate and momentum are cut as we train for a long time.

Training a deep belief network (i.e. stacked Restricted Boltzmann Machines trained with contrastive divergence) with dropout also works well.

Visualizing features, we see that they are more interpretable when dropout is used, indicating that it prevents coadaptation.

5 Experiments on TIMIT

We preprocess TIMIT with the Kaldi tool. Using dropout on the deep belief net improved accuracy. This deep belief network is then fine-tuned in a supervised manner on TIMIT, and dropout helps there too.

6 Experiments on Reuters

We consider 63 classes (i.e. the ones that have sufficiently many examples). Dropout improves accuracy and eliminates the need for early stopping.

7 Tiny Images and CIFAR-10

CIFAR-10 is a subset of the Tiny Images dataset. Tiny Images was collected by doing a Google Images search for the class and using the images that show up as examples. CIFAR-10 removes mislabled images.

8 ImageNet

1M images in 1000 classes. Sometimes there are multiple classes in an image, so that's why top-5 error rate is the primary metric of interest here.

9 Convolutional Neural Networks

We use convnets as our models for CIFAR-10 and ImageNet. The conv layers have fewer parameters than the hidden layers thanks to the convolution of small filters. We also use max-pooling with small stride. The human vision system has "complex cells" that do pooling. We also use local response normalization (LRN), which is where each filter competes for activation with its nearest neighbors (the filters are given some arbitrary order beforehand) - this lateral inhibition is also found in neurons in the brain. We use the ReLU activation function (becayse they don't saturate) and minimize cross entropy loss. We train on centered RGB pixels. Weights are initialized from a zero-mean Gaussian. We use a high variance to ensure that every neuron has at least one positive input. We train with SGD with momentum. Learning rate is cut after training for a while.

10 Models for CIFAR-10

We use 3 conv layers, with pooling and LRN. Then we have a locally-connected layer (we do dropout here). We end with a softmax.

11 Models for ImageNet

We train on random crops and flips (data augmentation). Our network has 5 conv layers, 2 fully-connected layers, and softmax. We use pooling and LRN. Some conv layers are not connected to all the input channels of the previous layers. We do dropout on the two fully-connected layers. At test time, we consider 5 crops and their flips (so 10 images total).