Maxout Networks

1 Citation

Goodfellow, Ian J., et al. "Maxout networks." arXiv preprint arXiv:1302.4389 (2013).

https://arxiv.org/pdf/1302.4389v4.pdf

2 Abstract

Maxout units take the maximum from a set of outputs. They are a nice complement to dropout.

3 Introduction

Dropout supposedly does model averaging - we want to verify this. Also, it is typically used as a slight performance boost - we want to build a whole model around it. When training with dropout, you need to take large steps on each gradient update.

4 Review of Dropout

On each training example, sample a binary mask and elementwise multiply it by the hidden units. You can think of this as randomly setting some hidden units to zero. This is kind of like training lots of different neural networks that loosely share parameters. When it comes time to make predictions, we divide the weights by two (roughly simulates a geometric mean over many models).

5 Description of Maxout

A Maxout network is a feedforward net that uses a new kind of activation function. Specifically, the activation function is (for $x \in \mathbf{R}^d$):

$$h_i(x) = \max_{j \in [1,k]} z_{ij} \tag{1}$$

where $z_{ij} = x^T W_{...ij} + b_{ij}$ for learned parameters $W \in \mathbf{R}^{d \times m \times k}$ and $b \in \mathbf{R}^{m \times k}$.

This is quite different from most activation units (e.g. it's not sparse, it does not have curvature, it can saturate).

Basically, you can view a maxout unit as making a piecewise linear approximation to the best convex activation function that fits the data.

6 Maxout is a universal approximator

Like a multilayer perceptron, a maxout network is a universal approximator (assuming each unit can have arbitrarily many affine components). The paper sketches a proof for this.

7 Benchmark Results

For permutation-invariant MNIST, we get state of the art results with two dense maxout layers followed by a softmax with dropout and constraining the norm of weight vectors. For regular MNIST, we used three convolutional maxout layers with spatial max pooling and a densely connected softmax layer. We trained on 50K images and measured the log likelihood at minimum validation error. Then we trained on the full 60K image dataset until we hit that log likelihood.

We preprocessed CIFAR-10 with global contrast normalization and ZCA whitening. We use a similar trick with the log likehood and validation error, but we train from scratch rather than continuing training because the learning rate is low at minimum validation error (we also tried training for the same number of epochs). Our network had three convolutional maxout layers, a fully connected maxout layer, and a dense softmax. We augment the dataset with horizontal flips and translation. We also got state of the art performance on CIFAR-100, and we didn't even tune hyperparameters.

For Street View House Numbers, we used three convolutional maxouts, dense maxout, and softmax. Images were preprocessed with local contrast normalization.

8 Comparison to Rectifiers

If you replace the maxout units with rectifier units, performance drops.

9 Model Averaging

If inputs to each layer are locally linear, dropout does exact model averaging. Most activation functions have curvature, so this assumption fails. Maxout units combined with dropout learn to be locally linear. However, if this was the only advantage of Maxout, you may as well use a ReLU because they are locally linear. This brings us to the next point...

10 Optimization

The difference between Maxout and ReLU + max pooling is that Maxout doesn't have the 0 when taking the max (i.e. there is no $\max(0, \text{otherstuff})$).

This matters because gradient descent takes small steps, but when you use dropout, it takes big steps. The ReLU units saturate often when using dropout, which prevents gradient from flowing. This doesn't happen with Maxout, which is why it works better with dropout.

11 Conclusion

Maxout is a new activation function that works well with dropout because it lets dropout do model averaging and lets the gradient flow more often.