Lecture 5:

CNN: Regularization

boris. ginzburg@intel.com

Agenda

- Data augmentation
- Dropout (Hinton et al)
- Stochastic pooling (Zeiler, Fergus)
- Maxout (I.Goodfellow)

Overfitting

Alexnet has 60 mln parameters. Dataset: 1000 classes, 1.5 mln images, 50K validating 150K testing. How to reduce over-fitting?

The easiest and most common method to reduce over-fitting on image data is to artificially enlarge the dataset using label-preserving transformations:

- generating image translations and horizontal reflections
- altering the intensities of the RGB channels in training images
- Elastic deformation (Simard, 2003)

Alexnet: Dropout

Technique proposed by Hinton et al. See Alex' paper: http://www.cs.toronto.edu/~fritz/absps/imagenet.pdf

Dropout was used for training of fully connected layers.

Training:

setting to 0 the output of each hidden neuron with probability 0.5. The neurons which are "dropped out" in this way do not contribute to the forward pass and do not participate in back-propagation.
So every time an input is presented, the neural network samples a different architecture, but all these architectures share weights.

Testing

At test time, we use all the neurons but multiply their outputs by 0.5.

Caffe: implemented as "dropout layer"

Learning rate and dropout

"Optimization proceeds very differently when using dropout than when using ordinary stochastic gradient descent. SGD usually works best with a small learning rate that results in a smoothly decreasing objective function, while dropout works best with a large learning rate, resulting in a constantly fluctuating objective function. Dropout rapidly explores many different directions and rejects the ones that worsen performance, while SGD moves slowly and steadily in the most promising direction."

http://arxiv.org/pdf/1302.4389.pdf

Zeiler & Fergus: Stochastic Pooling

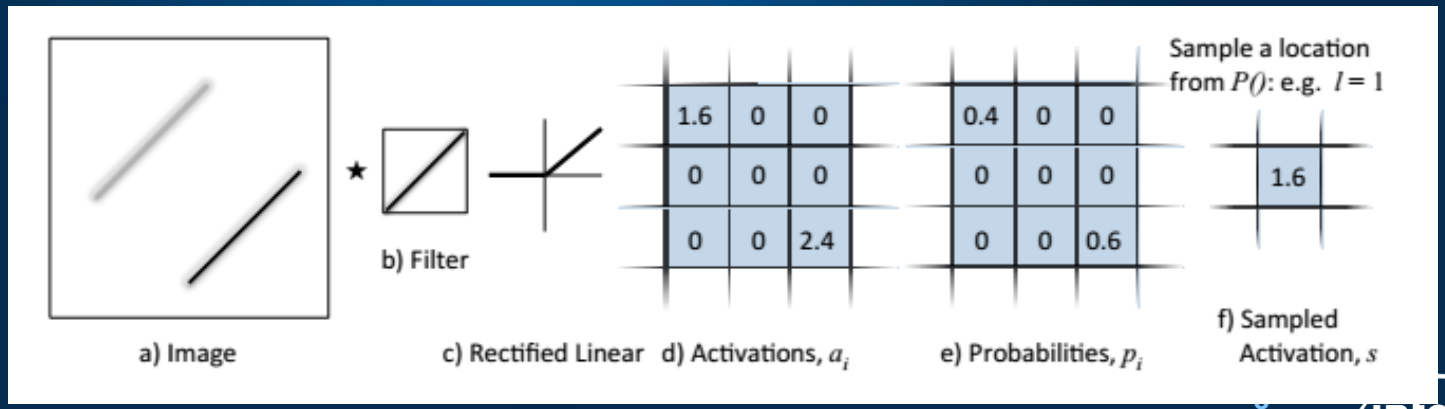
Similar to dropout technique , but used for pooling in convolutional layers: http://arxiv.org/pdf/1302.4389.pdf

Training:

1. Compute probability for each element in pooling region through normalization of activation inside pooling region: $p_i = \frac{a_i}{\sum_{k \in R} a_k}$
2. Pool activation based on Probabilities from step 1.

Testing: weighted pooling

$$s = \sum_{k \in R} p_k \, a_k$$



Zeiler & Fergus: Stochastic Pooling

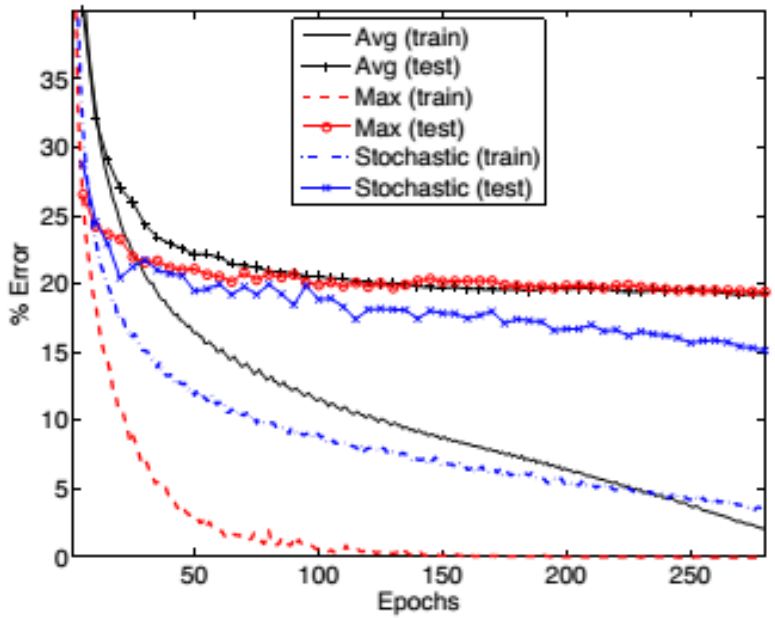


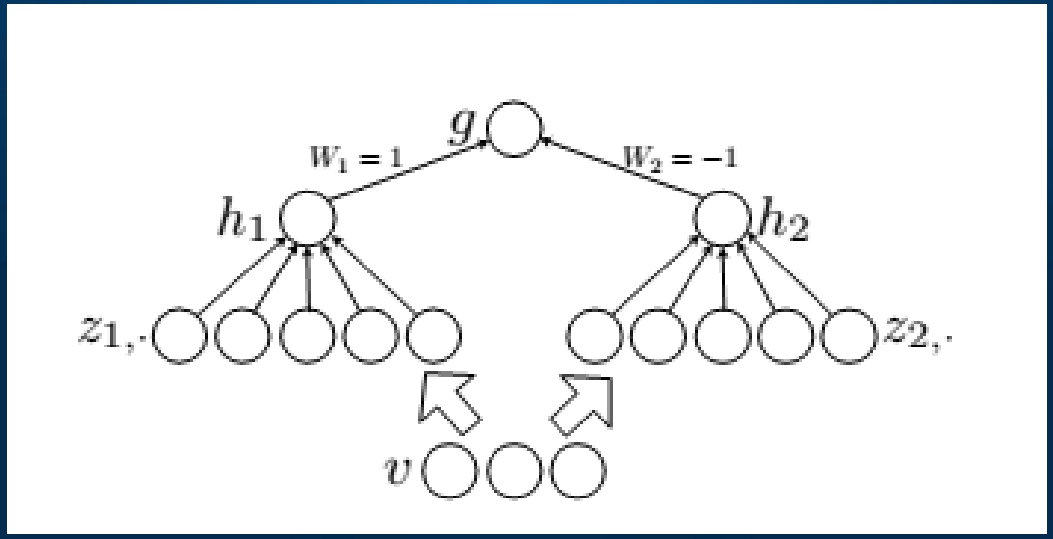
Figure 3: CIFAR-10 train and test error rates throughout training for average, max, and stochastic pooling. Max and average pooling test errors plateau as those methods overfit. With stochastic pooling, training error remains higher while test errors continue to decrease.[1]

Goodfellow: Maxout

http://www-etud.iro.umontreal.ca/~goodfeli/maxout.html

In a convolutional network, a maxout feature map can be constructed by taking the maximum across k affine feature maps (i.e., pool across channels, in addition spatial locations)

$$h_i = \max_{j=1..M} * z_{ij} = \max_{j=1..M} (w_{ij} * v_j + b_{ij})$$



Maxout results

Table 3. Test set misclassification rates for the best methods on the CIFAR-10 dataset.

METHOD	TEST ERROR
STOCHASTIC POOLING (ZEILER & FERGUS, 2013)	15.13%
CNN + SPEARMINT (SNOEK ET AL., 2012)	14.98%
Conv. maxout + dropout	**11.68 %**
CNN + SPEARMINT + DATA AUGMENTATION (SNOEK ET AL., 2012)	9.50 %
Conv. maxout + dropout + data augmentation	**9.38 %**

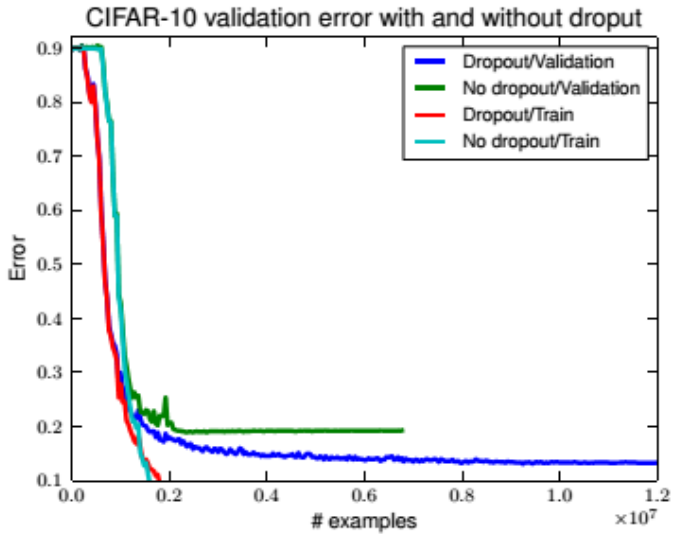


Figure 5. When training maxout, the improvement in validation set error that results from using dropout is dramatic. Here we find a greater than 25% reduction in our validation set error on CIFAR-10.

Exercises

1. CIFAR-10 - experiment with Dropout layer
2. Implement Stochastic pooling and Max-out layers