

Colinear Convolution Layer

Pang Liang

October 9, 2014

1 Cifar10 Database

The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. (See <http://www.cs.toronto.edu/~kriz/cifar.html>)

Fig 1 are the classes in the dataset, as well as 10 random images from each.

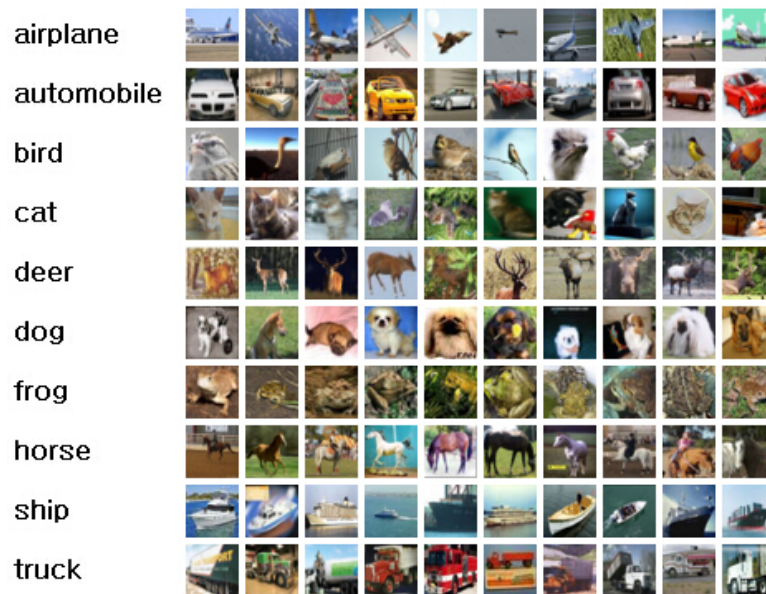


Figure 1: Example of cifar10 database.

2 Caffe Framework of Deep Learning

Caffe is a deep learning framework developed with cleanliness, readability, and speed in mind. (See <http://caffe.berkeleyvision.org>)

It implement many kind of layers list below:

- Vision Layers

- Convolution ✓
- Pooling ✓
- Local Response Normalization (LRN) ✓
- Loss Layers
 - Softmax ✓
 - Sum-of-Squares / Euclidean
 - Hinge / Margin
 - Sigmoid Cross-Entropy
- Activation / Neuron Layers
 - ReLU / Rectified-Linear and Leaky-ReLU ✓
 - Sigmoid
 - TanH / Hyperbolic Tangent
- Common Layers
 - Inner Product ✓
 - Splitting
 - Flattening
 - Concatenation
 - Slicing

3 Fast Version of NN Structure for Cifar10

The Cifar10 model Fig 2 is a CNN that composes layers of convolution, pooling, rectified linear unit (ReLU) nonlinearities, and local contrast normalization with a linear classifier on top of it all.

The inputs of the model are not the raw images or raw images normalized into $[0, 1]$, but the images preprocess by the image mean operation. In other word, the pixel at same position average over all images, so the input pixel value range is $[-127, 127]$. In the end of the network we use Softmax loss as the classification loss.

4 Colinear Convolution Layer in Formula

4.1 Feed Forward

$$\begin{aligned}
 a_{in} &= e_i F_n e_i^T + S_n e_i^T + b_n \\
 &= \sum_x \sum_y e_{ix} f_{xy} e_{iy} + \sum_x s_{xn} e_{ix} + b_n
 \end{aligned} \tag{1}$$

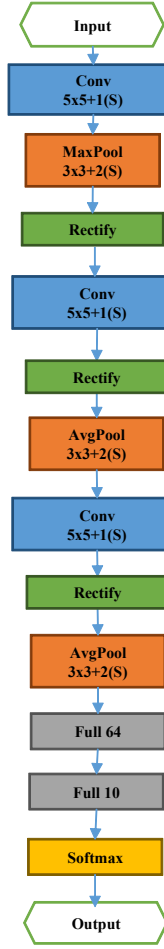


Figure 2: Cifar10 quick model.

4.2 Back Propagation

$$\sum_n \frac{\partial E}{\partial a_{in}} = \epsilon_{in} \quad (2)$$

- Error Propagation

$$\frac{\partial E}{\partial e_{ix}} = \sum_n \frac{\partial E}{\partial a_{in}} \cdot \frac{\partial a_{in}}{\partial e_{ix}} = \sum_n \epsilon_{in} ((\sum_y (f_{xyn} + f_{yxn}) e_{iy}) + s_{xn}) \quad (3)$$

- Weight Update

$$\frac{\partial E}{\partial f_{xyn}} = \sum_i \frac{\partial E}{\partial a_{in}} \cdot \frac{\partial a_{in}}{\partial f_{xyn}} = \sum_i \epsilon_{in} e_{ix} e_{iy} \quad (4)$$

$$\frac{\partial E}{\partial s_{xn}} = \sum_i \frac{\partial E}{\partial a_{in}} \cdot \frac{\partial a_{in}}{\partial s_{xn}} = \sum_i \epsilon_{in} e_{ix} \quad (5)$$

$$\frac{\partial E}{\partial b_n} = \sum_i \frac{\partial E}{\partial a_{in}} \cdot \frac{\partial a_{in}}{\partial b_n} = \sum_i \epsilon_{in} \quad (6)$$

See details in Fig 3.

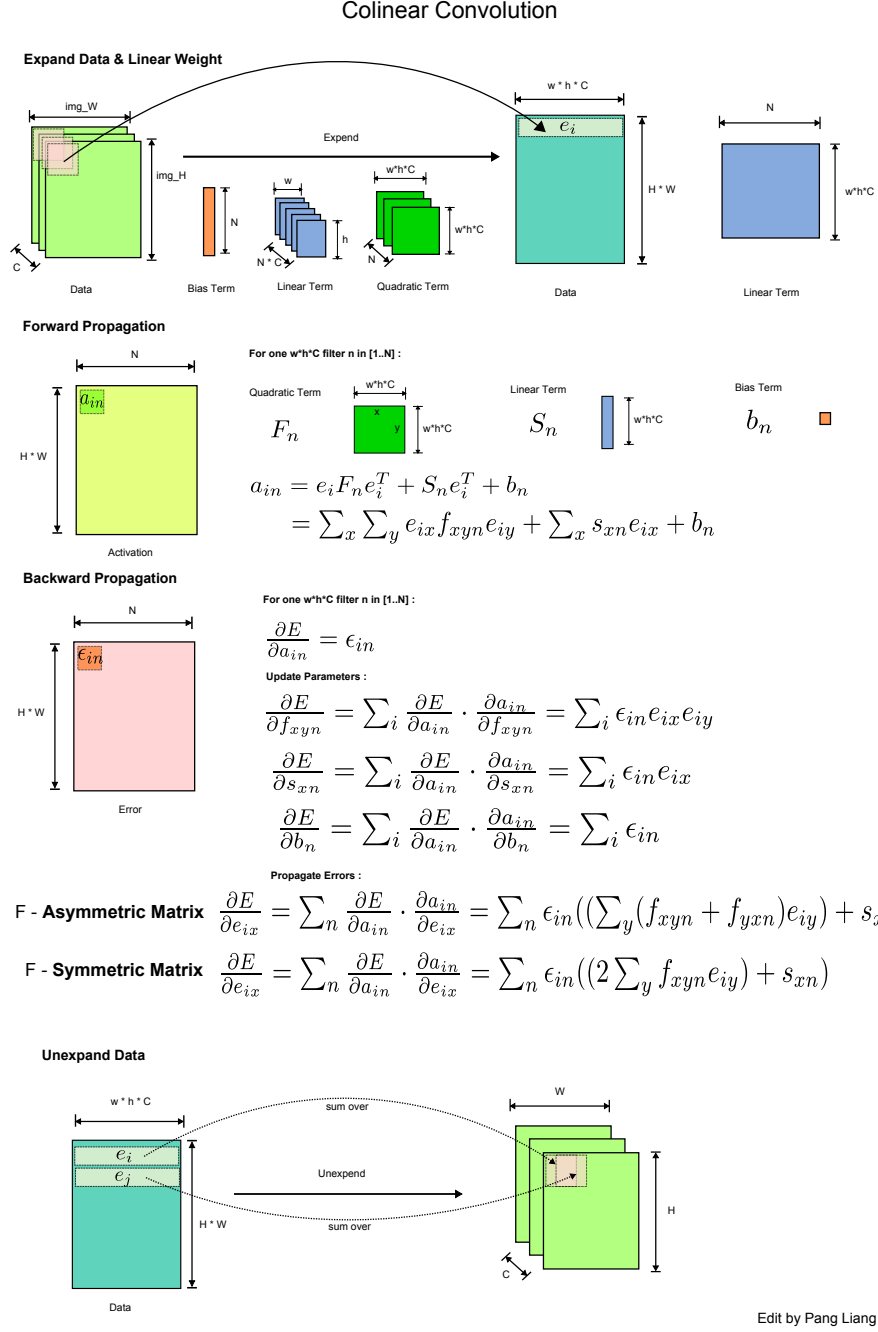


Figure 3: Colinear Convolution Layer.

5 Experiments

We replace the first Convolution layer to our Colinear Convolution layer. Weights in quadratic term are initialed in range of $[0, 0.000001]$, for the sake of the input range $[-127.127]$ and form of $e*f*e$. We want the activations of the layer not too large, so the weights in quadratic term are very small. For the same reason, the weight decay is set to 1000.

In Fig 4 and Fig 5, the number 16 or 32 denotes the first convolution layer output channel number. The lines entitled with Base mean that the result produce by Convolution layer, and others without produce by Colinear Convolution layer. The experiments show that the original configuration of the network is better than what we proposed. But there're some points we should optimize, which will show in next section.

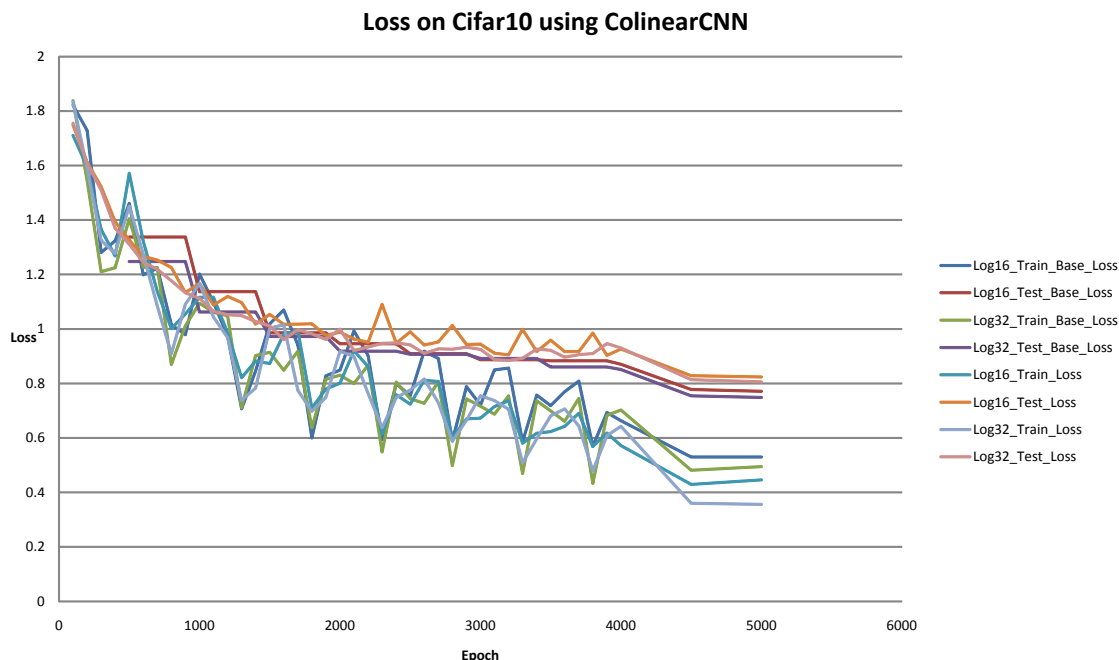


Figure 4: Train loss and Test loss of four net structures.

6 L1 Regularize on CCNN

7 Symmetric Strategy on CCNN

8 Mask the Parameters

Maybe this idea can be compared with dropout connections.

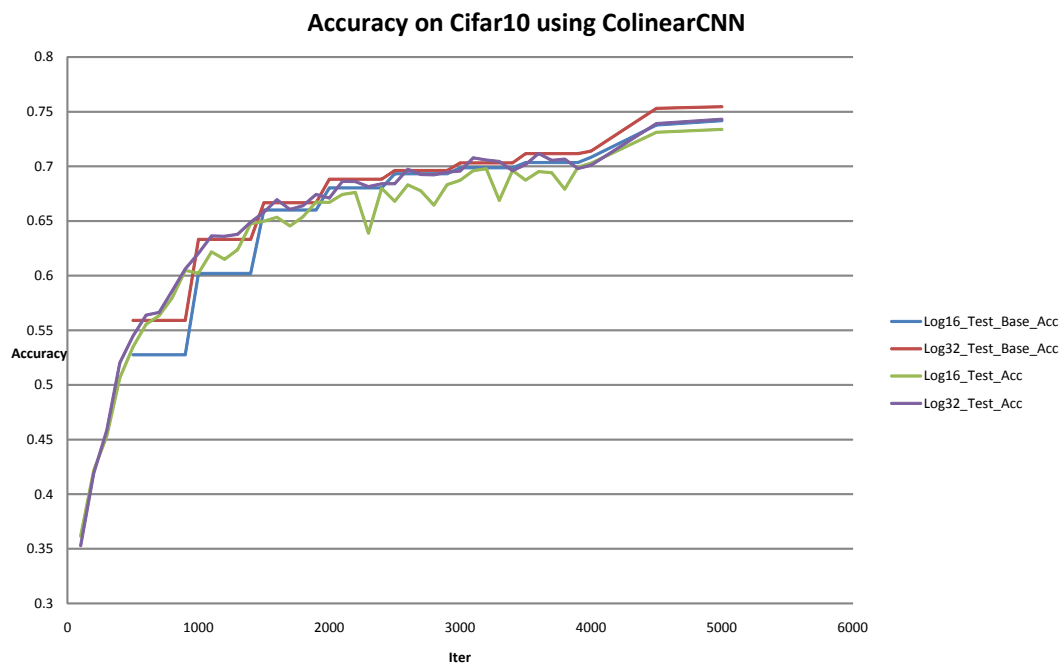


Figure 5: Test Accuracy of four net structures.

9 Restriction and Regularization of CCNN

1. Change the Asymmetric to Symmetric, in order to decrease the parameter numbers.
2. Change the L2 regularize to L1 regularize, for the 2 order relation is weaker than the 1 order relation and we need not that much parameters to model it.
3. Using mask matrix to erase the relations cross the channels.