CSCI 5722 Assignment 5

Name: Ajay Kumar Kedia
Identity Key: ajke2555

This paper introduces a deep CNN (Convolution Neural Network) which got break recording result in the ImageNet LSVRC 2012. There are bunch of things done to train the model so that it got a faster performance and good accuracy. Let's take these one by one:

1. The neural network was trained on 60 million parameters and 650K neurons which consists of 5 convolution layers, among these convolution layers, some of the layers consist of max pooling layer and 3 fully connected layers.
2. To increase the training model speed, used non-saturating neurons and parallelized GPU implementation for convolution operation in each layers of the training model.
3. To avoid overfitting, "dropout" regularization method and data augmentation are used.
4. Also, stated about the features like max pooling, SoftMax, regularization and normalization have been used to increase the performance.

CNN (Convolution Neural Network):

1. To classify the 1.2 million images into 1000 different categories, a strong assumption about the nature of the image is required like stationary, pixel dependencies. If it's a high-resolution image, then it needs to downsize the image.
2. Less connection and parameters will help in training the model faster in full connected CNN.

Dataset:

1. ImageNet- Around 15 million images from 22K categories and all are high resolution one and all were labelled manually by Amazon Mechanical Turk.
2. ILSVRC- It consists of subset of ImageNet where 1.2 million are the training set, 50K validation set and 150K test image to train and test our model for 1000 categories.
3. Also, all the images were downsized to 256 X 256 Pixels.

Architecture:

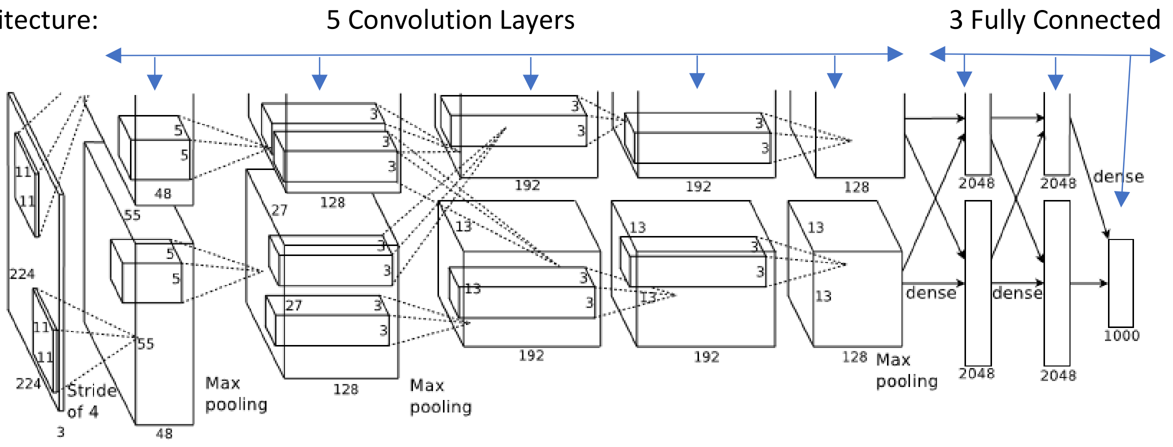


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

As I have already mentioned about the architecture, as it consists of

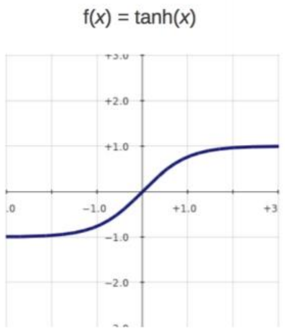
1. 8 Layers
    a. 5 Convolution Layers
    b. 3 Full connected Layers
2. 1000 – way SoftMax function at the output layer
3. Some of the convolution layers are also using the max pooling layers to get those features which are mostly contributing in the model and removed the others which are less than some threshold value.
4. GPU layers communicate at certain layers and consists of 2 GPU, one at top layer in the figure and second at the bottom layer.
5. 650K Neurons, 60M Parameters and 630M Connections.

Methodology and Activation Function:

1. ReLU function train our model several times faster than the tanh activation function.
    a. In tanh, during training time, saturating non-linearity with gradient descent is slower than the non-saturating non-linearity i.e. f(x) = max (0, x)
    b. Also, faster learning had influence the large model trained on large dataset.

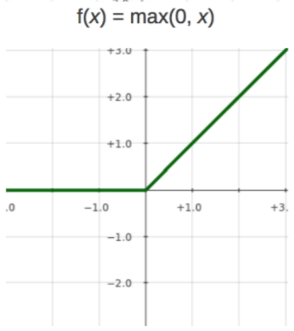
ReLU Saturating Nonlinearity:

a. F(x) = tanh(x) or f(x) = $(I + e^{-x})^{-I}$
b. Very slow to train using gradient descent.



ReLU Non-Saturating Nonlinearity:

a. F(x) = max (0, x)
b. Quick to train

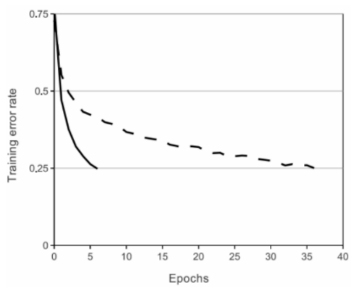


ReLU Nonlinearity:

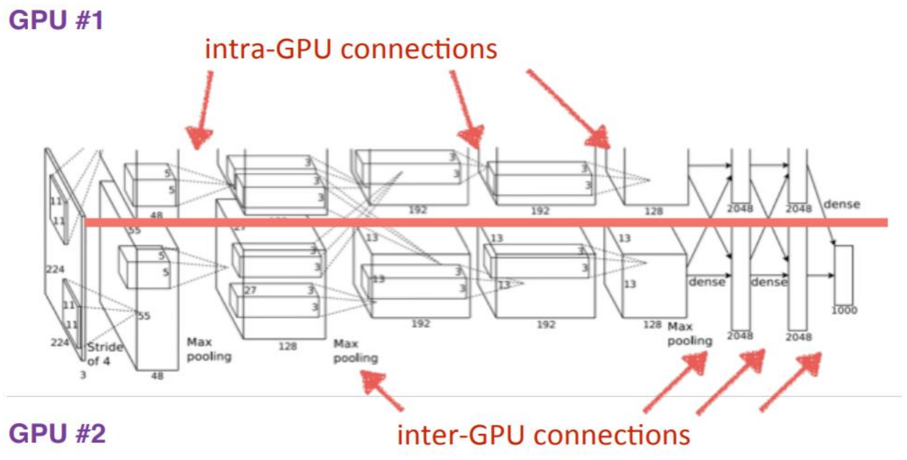
4 Layer ReLU Converges 6 times faster than with equivalent network with tanh neurons.

Solid Line: ReLU Activation function (Converging faster)

Dashed Line: tanh Activation function (Converging slow)



2. This model is trained on multiple GPUs



3. Response Normalization Locally:

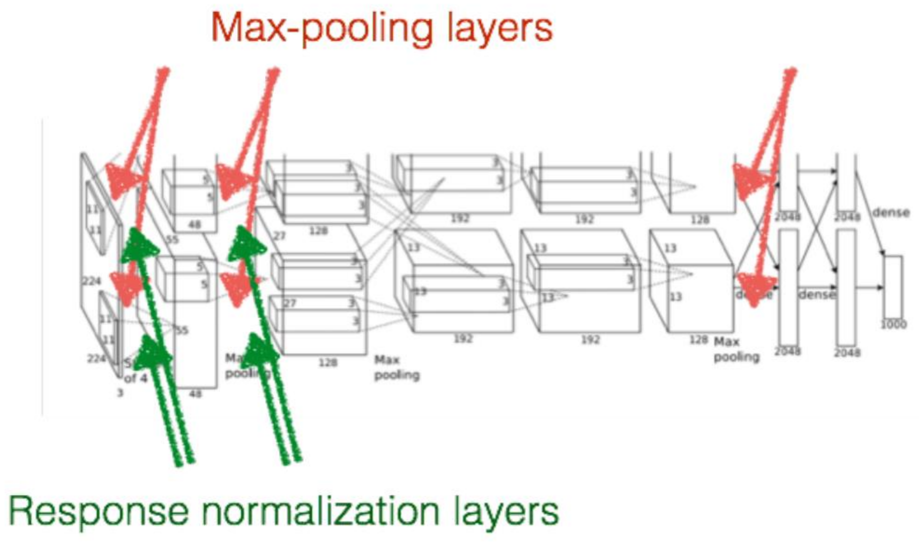
$$b^i_{x,y} = a^i_{x,y} \Big/ \left(k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (a^j_{x,y})^2 \right)^{\beta}$$

Response-normalized activity

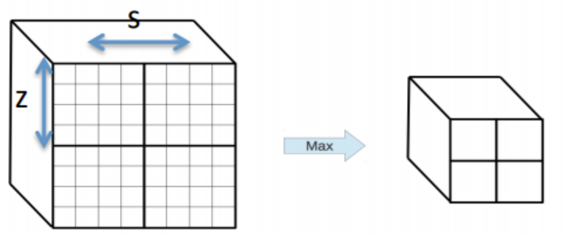
Activity of a neuron computed by applying kernel I at position (x,y) and then applying the ReLU nonlinearity

a. Mimics lateral inhibition on real neurons but no need to input normalization with ReLUs.
b. Normalization applied after ReLU activation function on 1st and 2nd Convolution Layers
c. Compared top-1 and top-5 error rates and it's decreased to 1.4% and 1.2% respectively which is being net trained with one GPU and half neurons.

4. Overlapping Pooling



a. Neighborhood Z = 3 and Stride S = 2, → < Z Overlapping Pooling



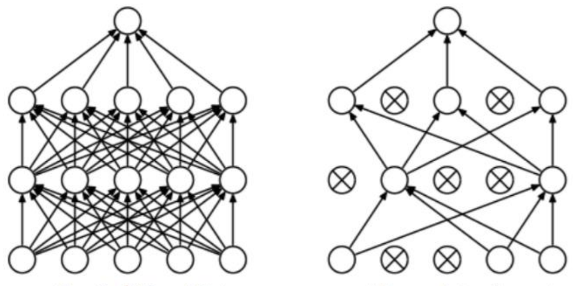
b. After response normalization at 1st and 2nd convolution layers and at the 5th convolution layers

c. Top-1 and Top-5 error rates decreased by 0.4% and 0.3% respectively, compared to non-overlapping S=2, Z=2

5. Avoid Overfitting

a. Dropout

- Employed in first 2 fully connected layers where p = 0.5



b. Data Augmentation

- RGB intensity on image alteration
- Horizontal reflection and image translation
- Add multiple principle components

$$[\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3][\alpha_1\lambda_1, \alpha_2\lambda_2, \alpha_3\lambda_3]^T$$

$$\alpha_i \sim N(0,\ 0.1)$$

6. Details of Learning:

a. Trained the model using stochastic gradient descent with momentum 0.9, batch size of 128 examples and weight decay of 0.0005.

b. Weight decay helped in reducing the training error.

c. Weight initialized from gaussian distribution of mean = 0 and SD = 0.01

d. Weight is biased for 2nd, 4th and 5th layers as these are initialized as 1 where as rest of the layers are initialized as 0.

e. Learning rate is equal in all layers and it's adjusted manually i.e. divided it by 10 as validation error stops decreasing.

f. Learning rate = 0.01 and reduced it 3 times during training the model.

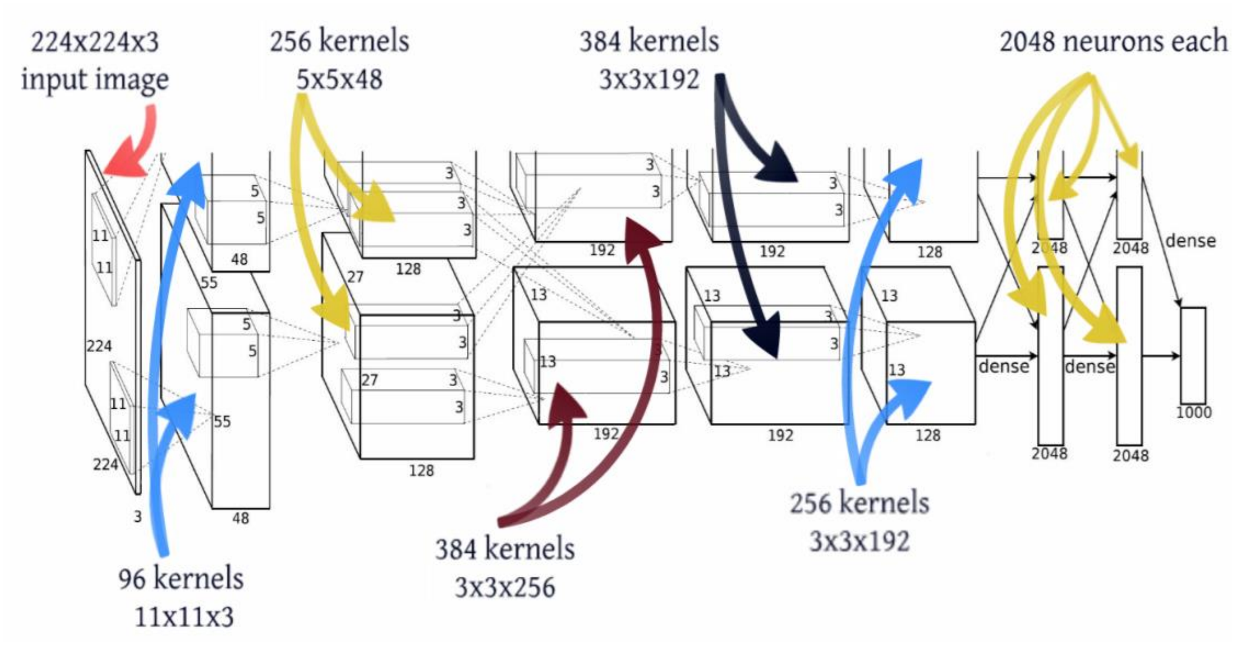
$$v_{i+1} := \overset{\text{momentum(damping parameter)}}{0.9} \cdot v_i - \underbrace{0.0005 \cdot \epsilon \cdot w_i}_{\text{weight decay}} - \overset{\text{Learning rate (initialized at 0.01)}}{\epsilon} \cdot \underbrace{\left\langle \frac{\partial L}{\partial w}\Big|_{w_i} \right\rangle_{D_i}}_{\text{Gradient of Loss w.r.t weight Averaged over batch}}$$

$$w_{i+1} := w_i + v_{i+1}$$

Batch size: 128

g. 5-6 days on training for 90 epochs (specification: 2 NVIDIA GTX 580 3GB GPUs)

7. Overall Architecture



8. Result:
   a. Below table shows the result of ILSVRC 2010 test set.

Model	**Top-1**	**Top-5**
Sparse Coding	47.1%	28.2%
SIFT + FVs	45.7%	25.7%
CNN	**37.5%**	**17.0%**

Table 1: Comparison of results on ILSVRC-2010 test set.

- Baseline sparse coding and fisher vectors.
- CNN achieves Top-1 and Top-5 as error rates of 37.5% and 17.0% respectively

   b. Below table shows the result of ILSVRC 2012 test set.

Model	**Top-1(Validation)**	**Top-5(Validation)**	**Top-5(Test)**
SIFT + FVs	-	-	26.2%
1 CNN	40.7%	18.2%	-
5 CNNs	38.1%	16.4%	16.4%
1 CNN*	39.0%	16.6%	-
7 CNNs*	36.7%	15.4%	15.3%

Table 2: Comparison of error rates on ILSVRC-2012 validation and test sets.

- Validation error and test error are not differed by more than 0.1%, Compare Top-5(Validation) and Top-5(test) result for different model.
- Pretrained on ImageNet 2011 fall release and fine-tuned it on ILSVRC-2012 training data using Convolution Neural Network.
- Did experiments on increasing and decreasing the convolution layers in our training data and also worked on max pooling layers with different convolution layers.

9. Qualitative Evaluations:
   a. Two data connected layers learned:
      - frequency and orientation related Kernels
      - Blobs colored parameters
      - Kernel on GPU 1 (top 48 kernels on top layer) is color-agnostic and kernel on GPU 2(bottom 48 kernels on bottom layer) color-specific.
      - Independent on random weight initialization, 96 convolution kernels

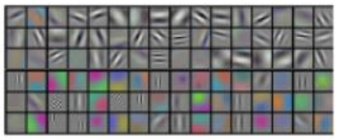
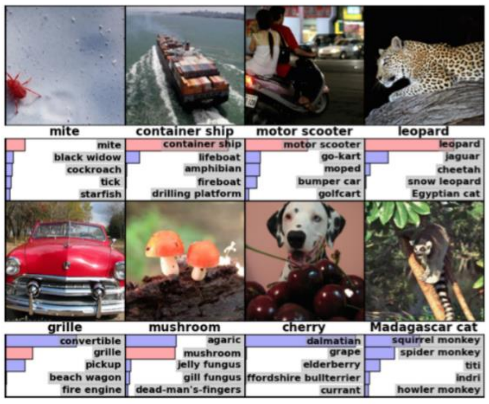
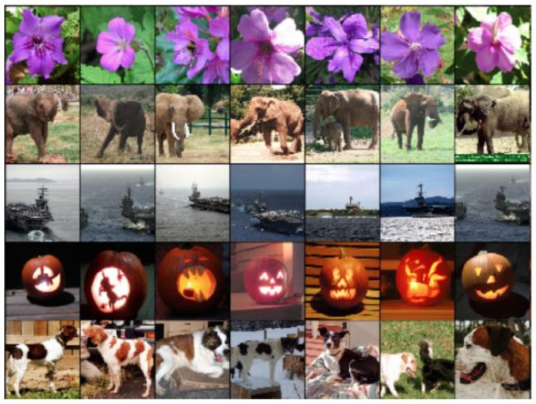


Figure 3: 96 convolutional kernels of size $11\times11\times3$ learned by the first convolutional layer on the $224\times224\times3$ input images. The top 48 kernels were learned on GPU 1 while the bottom 48 kernels were learned on GPU 2. See Section 6.1 for details.

      - 11 x 11 x 3 size kernels
      - At output layer, 1000-softmax, image similarity based on the feature activation function at last full connected layers.



Eight ILSVRC-2010 test Images



Five ILSVRC-2010 test Images

- The correct label is written under each image and the probability assigned to each label is also shown with a red bar (if its happen to be in top5)

10. Weakness:
    Less theoretical explanation and all the decision based on the result got by the publisher of the paper.
11. Future Research Plans and advance Knowledge about the area:
    As we proceed, I have mentioned the information about the different layer of CNN model and which layer gave the good impact which didn't and where it used the max pooling and where wasn't. It's mentioned that writer wanted to try unsupervised pre-training, but he didn't because it would take lots of time to train the model and more computation power which significantly increase the size of the network. Still, have many orders of magnitude to match the infero-temporal pathways of human visual system.

Future Research plans like classification of videos which helps in finding out the temporal sequence missing in the videos. Apart from that, separation of two pathways for spatial and temporal network analogues.

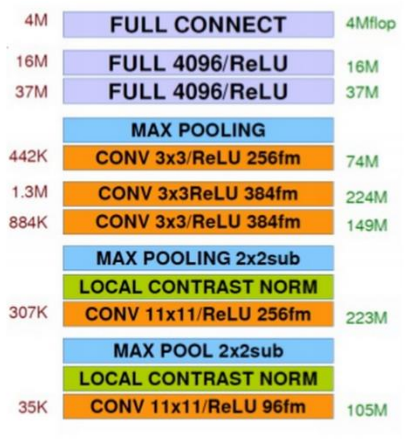


Fig: Different Layers in CNN Model