Review on ALEX Net

AlexNet marked its own style in deep learning. This model presented in this paper used ILSVRC dataset images which are of different resolutions It has training set of 1.2 million images, 50,000Validation images and 150,000 testing images.

The model requires fixed size image but the Imagenet dataset has high resolution images. Hence, the images were scaled to 256×256 pixels. This was done by Scaling a possibly rectangular image so that the shorter side is 256 pixels. Then the middle 256×256 patch was taken as the input image.

It Overall architecture contains five convolutional and three fully connected layers. The output of the last fully connected layer is sent to a 1000-way SoftMax layer which corresponds to 1000 class labels in the ImageNet dataset. In this model 1.2 million training parameters are too big to fit into the NVIDIA GTX 580 GPU with 3GB of memory. Therefore, the author spread the network across two GPUs.

Alex Net uses Rectified Linear Units (ReLUs) activations which are non-saturating nonlinearity. ReLU avoid vanishing gradients for positive values. It is computationally efficient and converges performance is higher than sigmoid and tanh.

In the paper the author uses Overlapping max pooling of size 3x3 with stride 2. This helps the model to reduces the top-1 and top-5 error rate by 0.4% and 0.3%. Local Response Normalization (LRN) is used to help with generalization. It is used for lateral inhibition (inspired from real neurons), So that locally maximum pixels values are used as excitation for next layers. This normalization layer used after ReLU nonlinearity function in certain layers. LRN reduces the top-1 and top-5 error rates by 1.4% and 1.2%.

The model used a regularization technique called **Dropout** which will randomly set the output of each hidden neuron to zero with the probability of p=0.5. Those dropped out neurons do not contribute to forward and backward passes. Dropout was used in the first two fully connected layers. It helped the model to prevent from overfitting. But it will double the number of iterations to converge.

This model uses two forms of data augmentation. First, it consists of generating image translations and horizontal reflections. This scheme helped increase the size of the data by a scale of 2048 without which the network will suffer from overfitting.

Second form is to alter the intensity of RGB channels by performing PCA on the set of RGB pixel values throughout the training set. Then, use the eigenvalues and eigenvectors to manipulate the pixel intensities. Eigenvalues are selected once for entire pixels of a particular image.

The model was trained with Stochastic Gradient Descent of Batch size 129, Momentum0.9, Weight Decay=0.0005. The bias was initialized as 1 for 2nd, 4th, 5th conv layers and fully connected layers. And zero is initialized for remaining layers.

The results were top-1 and top-5 test set error rates of **37.5%** and **17.0%**. This model *stood first in the* ILSVRC *context in 2012.*

1. **What assumptions do CNNs make that make them appropriate for the recognition of images?**

Ans: CNN's assumes "Locality of Pixel Dependencies" and "Stationary of Statistics". Locality of pixel dependency means close pixels are very likely to be dependent on each other. Stationary of statistics means the data's mean and variance do not change over the change of time.

2. **What are the new features introduced in AlexNet that had not been used before?**

   Ans: An important feature of the AlexNet is the use of ReLU (Rectified Linear Unit) Nonlinearity. The main advantage of this non-linear function is unlike Sigmoid and tanh functions it doesn't saturate.

3. **How many parameters are there in AlexNet that need to be learned, and how many images are there in the training set? Why is overfitting is a problem, and what is done in AlexNet to help prevent overfitting.**

   Ans: i. Any CNN will have two parameters to calculate: Weights and Biases. AlexNet has over 60 million parameters in its model. There are 1.2 million images in the training set.

   ii. Overfitting is a problem because our model will poorly generalize on new data. To prevent overfitting the author artificially enlarged the dataset using Data Augmentation technique. This is done is two ways:

   a. First, it consists of generating image translations and horizontal reflections. This scheme helped increase the size of the data by a scale of 2048.
   b. Second form is to alter the intensity of RGB channels by performing PCA on the set of RGB pixel values throughout the training set. Then, use the eigenvalues and eigenvectors to manipulate the pixel intensities. Eigenvalues are selected once for entire pixels of a particular image.

   iii. Overfitting is also controlled by using a Dropout method in AlexNet. This method used in the first two fully connected layers. In this method our model will randomly set the output of each hidden neuron to zero with the probability of p=0.5. Those dropped out neurons do not contribute to forward and backward passes.

4. **What preprocessing of the images was done in AlexNet, and why?**

   Ans: The AlexNet model requires fixed size image but the image dataset has high resolution. Hence, the images were preprocessed and scaled to 256×256 pixels which were suitable for the model.

5. **What is a ReLU activation function, and for what reason(s) is it preferred over the sigmoid or tanh activation functions? Are there any other activation functions that researchers have considered? If so, what are they and what are the advantages that they may have?**

   Ans: I. ReLU activation function converges better than Sigmoid and hyperbolic Tangent functions. This is because it is a Non-saturating nonlinearity function which is much faster than saturating non-linear functions. Hence Complex CNNs with ReLU will train faster than other activation functions. But ReLU can be used only within hidden layers of CNN models. Hence, we have to use softmax function for the output layers.

   ii. The researchers have considered many other Activation functions in various Papers. Example: Jarrett.K proposed a nonlinearity f(x)=|tanh(x)|. This worked particularly well while using contrast normalizations and local average pooling on Caltech-101 dataset.

6. **Explain, intuitively, what local response normalization is and why it is used?**

   Ans: ReLU neurons have unbounded activations function, with such activation function the output layer is not constrained within a bounded range. But our output grows as per training input data. To prevent, we generally use Local response normalization.

   But here, LRN is used for lateral inhibition (inspired from real neurons), So that locally maximum pixels values are used as excitation for next layers. This normalization layer used after ReLU nonlinearity function in certain layers.

7. **Explain dropout and how and why it is used in AlexNet?**

   Ans: Dropout is a regularization technique. In this technique, the output of hidden layer neurons is randomly set to zero with the probability of p=0.5. These neurons are not considered during the forward and backward passes. Dropout was used in the first two fully connected layers in Alex net. It helped the model to **prevent from overfitting**. But it will double the number of iterations.

8. **The paper says that "We trained our models using stochastic gradient descent with a batch size of 128 examples, momentum of 0.9, and weight decay of 0.0005." What does this mean? What learning rate was used, and how was the CNN initialized?**

   Ans: Batch size, Momentum, weight decay are the Hyper Parameters for Gradient Decent. The learning Rate was initialized with 0.1 and was divided by 10 when ever the validation error rate stopped improving with the current learning rate. Learning rates were equal for all layers.

CNN weights were initialized from zero mean gaussian distribution with a standard deviation of 0.01. The bias was initialized as 1 in the 2nd, 4th and 5th Convolution layers and also in the three fully connected hidden layers. The remaining layers are initialized with 0.