research paper :alexnet

this paper is about classifing the image net dataset using CNNs. after the release of this model it was for the first time that we achieved test data set top-1 and top-5 error rates were 37.5%, 17.0% better then the previous rates over the dataset containg 1.2 million high resolution images and 1000 different classes. top 5 error rates means fraction of the test images for which the correct label is not among the five labels considered the most probable by the model.

Dataset:

model was trained on image net dataset of over 15 million labeled high-resolution images belonging to roughly 22,000 categories. But the images were of the variable sizes so they downsampled it to the 256 X 256.first they rescaled the image such that the shorter side was of length 256 and then they cropped the central part from the resulting image. for the sake of preprocessing the data they only subtracted the mean activity over the training set from each pixel. in order to train on this dataset it required 2 GTX 580 3GB GPUs and took 5 to 6 days to train on it.

Features:

1. earlier models used tanh or sigmoid activation functions but it was trained on relu activation function because in terms of training time with gradient descent, these saturating nonlinearities are much slower than the non-saturating nonlinearity.

2. due the large dataset it was not possible to train the model on one gpu . so 2 gpus were used in parallel to train the model. in this kernels were halved on each gpu and also the gpu communicate with each other only at certain layers.

3.ReLUs have the desirable property that they do not require input normalization to prevent them from saturating. If at least some training examples produce a positive input to a ReLU, learning will happen in that neuron. but still a unique type of local normalization was used.

bixy=aixy/((k+α(summation of (ajxy)^2 from j=max(0,n/2) to j=min(N-1,i+n/2)))^β)

it was applied after the relu non linearity. In it sum runs over n “adjacent” kernel maps at the same spatial position. The constants k, n, α, and β are hyper-parameters whose

values are determined using a validation set they used k = 2, n = 5, α = 10−4

, and β = 0.75.

4. instead of simply doing the local pooling they used the overlapping pooling with stride =2,filter size=3. it was found by experimentation that the overlapping pooling find it slightly more difficult to overfit.

Architecture:

the net contains eight layers with weights; the first five are convolutional and the remaining three are fully connected. The output of the last fully-connected layer is fed to a 1000-way softmax which produces a distribution over the 1000 class labels.The kernels of the second, fourth, and fifth convolutional layers are connected only to those kernel

maps in the previous layer which reside on the same GPU (see Figure 2). The kernels of the third convolutional layer are connected to all kernel maps in the second layer. The neurons in the fully connected layers are connected to all neurons in the previous layer.

input image size 227 X 227 X3

size of kernels	no.of kernels
11X 11X 3	96
5 X 5X 48	256
3 X 3 X 256	384
3 X 3 X 192	384
3 X 3 X 192	256

fully connected layers have 4096 neurons.

preventing overfitting:

Data augmentation:

The first form of data augmentation consists of generating image translations and horizontal reflections. We do this by extracting random 224 × 224 patches (and their horizontal reflections) from the 256×256 images and training our network on these extracted patches. At test time, the network makes a prediction by extracting

five 224 × 224 patches (the four corner patches and the center patch) as well as their horizontal reflections (hence ten patches in all), and averaging the predictions made by the network's softmax layer on the ten patches this is called test time augmentation (TTA).

The second form of data augmentation consists of altering the intensities of the RGB channels in training images. Specifically, we perform PCA on the set of RGB pixel values throughout the ImageNet training set. To each training image, we add multiples of the found principal componentswith magnitudes proportional to the corresponding eigenvalues times a random variable drawn from a

Gaussian with mean zero and standard deviation 0.1. Therefore to each RGB image pixel (Ixy =[IRxy, IGxy, IBxy]^T) we add the following quantity:

([p1, p2, p3][α1λ1, α2λ2, α3λ3]^T)

where pi and λi are ith eigenvector and eigenvalue of the 3 × 3 covariance matrix of RGB pixelvalues, respectively, and αi is the aforementioned random variable

Dropout

dropout consists of setting to zero 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 neuralnetwork samples a different architecture,but all these architectures share weights. At test time, we use all the neurons but multiply their outputs by 0.5. due to dropout neurons are forces to get more robust features from image.

Learning:

They trained models using stochastic gradient descent with a batch size of 128 examples, momentum of 0.9, and weight decay of 0.0005 . they initialized the weights in each layer from a zero-mean Gaussian distribution with standard deviation 0.01. they initialized the neuron biases in the second, fourth, and fifth convolutional layers, as well as in the fully-connected hidden layers, with the constant 1. This initialization accelerates

the early stages of learning by providing the ReLUs with positive inputs. they initialized the neuron biases in the remaining layers with the constant 0.