"ImageNet Classification with Deep Convolutional Neural Networks" Summary

The goal of the paper is to train a deep convolutional neural network to classify 1.2 million images into 1000 different categories. The dataset used is the ImageNet since ImageNet images are variable resolution, and the model presented in this paper requires fixed size images, they scaled every image to 256x256 pixels. The network presented in the paper works with 224x224 images, which are generated by randomly sampling patches of that size from each 256x256 image. The model has 5 convolutional layers and 3 fully-connected layers. They chose the ReLU function which made the model train faster. Local Response Normalization is used which mimics a form of lateral inhibition found on real neurons. This is applied after ReLU in the 1st and 2nd convolutional layers. It improves top-1 and top-5 error rates by 1.4% and 1.2%. Then they use max pooling but with overlapping windows containing filter size of 3 and stride of 2. To reduce overfitting, Data Augmentation and Dropout is used. In data augmentation, image translations and horizontal reflections is generated and the intensities of RGB channels is altered. During learning, Stochastic Gradient Descent, batch size = 128, momentum = 0.9, weight decay = 0.0005 is used. Bias in 2nd, 4th, and 5th convolutional layers initialized as 1 which accelerated learning as the ReLU was fed with positive inputs from the start. Bias in remaining layers initialized as zeros. The learning rate is equal for all layers. It is adjusted manually (divided by 10 when validation error stopped decreasing) and is initialized at 0.01 and reduced 3 times during training.

The training is done on two GPUs for parallelism, and the setup is quite interesting. The GPUs used each have 3GB memory. The interesting bit is that while layer 4 (conv) operates on input which comes from the layer 3 activations from *both* GPUs, other conv layers in the network do not have this cross GPU communication going on, and only work with activations from the GPU local half of the network. Selecting how many layers should have cross-GPU comms going on is a problem for cross-validation, and apparently this scheme seems to have worked well for them. As the authors mention in the paper, the two halves of the network consistently specialized in features, with one focusing on frequency/orientation, and the other in colored blob detection. This model was trained during 90 epochs which took 5-6 days on two NVIDIA GTX 580 3GB GPUs. Here, convolutional kernels showed specialization, most of top-5 labels were reasonable and image similarity based on the feature activations induced at the last fully connected layer. This work was the first of its kind to have trained deep convolutional networks on GPUs to achieve impressive results on the ImageNet dataset for object detection.

There is a big fluctuation of loss for the last epoch of training because of the relatively small batch size of 128 within the million images presented in the datset.