Derek Lee Deep Learning Fall 2020

Professor Curro Quiz #3

1. Describe the main method and key takeaways from "probes." What was the main hang-up that prevented the authors from completing more significant experimentation?

The authors attempt to better understand deep neural networks by using 'probes' at various layers. These probes are linear classifiers, applied to the features of each layer to predict classes. The authors found that there was a monotonic increase in the linear separability of features in each layer as the depth increases. The authors argue that "the first layers of a convolution network for image recognition contains filters that are relatively 'general'… The last layers are specific to the dataset being used".

Each layer has less information than the previous layer, but it has a better representation of the information. The probes are trained alongside the model, but such that the training of the probes does not affect the model. This is accomplished by blocking the backpropagation from the probes to the model.

The main hang-up during the experiment was that some hidden layers have "an exceedingly large quantity of features". The memory requirements for a single probe is ridiculously large ("upwards of a few gigabytes"). The authors proposed several strategies for dealing with this problem. The first strategy involved using a random subset of the features. The second strategy involved projecting the features to a lower-dimensional space, although the projection matrix could still potentially take a large amount of memory. Finally, when dealing with images, 2D pooling can be used on each channel, which will reduce the number of features to the number of channels.

2. Describe the main method and key takeaways from "confidence penalty."

The authors propose a 'confidence penalty' on the output distribution of a neural network to improve generalization. The negative entropy of the output distribution is added to the negative log-likelihood. This is ideal for reinforcement learning, encouraging exploration. In

supervised learning, a quick convergence is desired, while overfitting should be avoided. One implementation involved annealing, Another implementation involved a weak confidence penalty at the beginning and a strong penalty near convergence. This can be achieved by penalizing output distributions when they fall below a certain entropy threshold.

A confidence penalty can act as a strong regularizer in supervised learning, while also avoiding the need to modify hyper-parameters. Using the confidence penalty can "improve a wide range of state-of-the-art models.

3. Is there a difference at inference time between batch-norm and batch-renorm?

In batch-norm, the model uses means and variances to normalize the activations of a neuron in a deep network. During training time, the model uses statistics from each individual minibatch. During inference time, the model uses a moving average of minibatch means and variances. This changes the activations in the network during inference time. With large minibatch sizes and i.i.d. samples from the training distribution, the changes are small, and can help generalize. However, with small minibatch sizes, the estimates of the mean and variances are less accurate. These problems are compounded with depth.

In batch-renorm, the moving average of minibatch means and variances is used during both training and inference. The model utilizes correction terms to account for the fact that the minibatch statistics differ from the population statistics. This allows layers to observe the 'correct' activations that would be present during inference. This ensures "all layers are trained on internal representations that will be actually used during inference."

Batch-renorm runs at the same speed as batch-norm during training and inference. It significantly improves performance on small or non-i.i.d. minibatches. However, it has more hyperparameters: the update rate for the moving averages, and the schedules for correction limits.