To prune, or not to prune: exploring the efficacy of pruning for model compression.
Michael H. Zhu, Suyog Gupta**.**

What is the paper about?

Deep neural models show high performance in varying tasks including voice recognition, NLP task and more. Their success comes hand in hand with great amounts of data and computational power, therefore deploying them on edge computing devices with limited resources is not always possible. This paper examines the effect of model pruning as means for model compression. It makes a comparison between two types of compressed models:

1. Large-Sparce:
   In this model the weights are gradually zeroed during training by applying a binary mask over the base model.
2. Small-Dense:
   Use the base structure and reduce its hidden layers size with respect to Large-Sparce.

Methods:

The pruning is applied gradually. First one should choose the layers he wants to prune. Secondly a sparsity-mask is added to each of these layers. At the end of training step t the binary mask blocks the s_t weights with the minimal magnitude. In back propagation gradients flow only through none-masked weights, reducing memory access arithmetic operations. s_t is defined by:

$$s_t = s_f + \left(s_i - s_f\right)\left(1 - \frac{t - t_0}{n\Delta t}\right)^3$$

where s_i, s_f are the initial and final pruning rates and n is number of training steps. In this work Δt between 100 and 1000 training steps had a small affect on the final model quality.

The gradients of this function are very high for small t values and they are decreasing quickly. Setting the pruning rate by this function results in a rapid pruning in the initial stage when the redundant weights are abundant, after that the pruning is done in a moderate pace.

Learning rate should not be extremely high or low. Too low will make it difficult for the subsequent training step to recover from the loss of accuracy and too high will not allow the weights to coverage before the pruning step. The total number of training steps n is largely dependent on the learning rate.

The sparsity rates that were examined ranged between 50%-95%. The pruning worked very well on LSTM layers, dense softmax layers and even in embedding layers.

Results:

The authors show that in some cases the pruning not only reduced the network size, but it also improved accuracy. For instance, for Pen Tree Bank dataset, the Large-sparce model with 95% sparsity got 80.24 perplexity while a medium model with no pruning at all got 83.37.