**Summary**

One of the most recurrent tasks in machine learning is to create a model using the given training data which would be able to predict the output correctly given some unseen data. While doing so we may encounter a problem called “***overfitting*”** which may be caused due to a significantly large number of parameters as compared to the training data for our neural network. Overfitting happens when the machine learns noise instead of mapping a relation between training data and output. This network would work very well on the training data set but fails on unseen data increasing generalization error. The authors have addressed this problem using a regularization technique called ***dropout****,* which randomly drops units (hidden layer nodes or input nodes) from the network. Dropping a unit means taking that particular node and removing it from the network along with its input and output links.

The authors have stated that model combination is bound to increase the performance of the machine most of the times provided these models are trained on different data or have different architecture. Getting a large amount of data may not be possible at times and computing parameters for different architectures can be cumbersome. Moreover, training many models needs a lot of computation. Dropout technique addresses this problem by dropping the units **temporarily** and **randomly** giving rise to up to 2n (n represents the number of units) different models where hidden layer units and input units have a probability of not getting dropped set. We train our model using this **thinned** network (network with units which were retained) i.e. we use this thinned network for both forward and backward propagation. However, during testing, all the possible thinned networks are combined by scaling the weights of retained units with p.

Dropout works well because we are forcing nodes to be independent and not be dependent on other nodes for correcting their weights making the network more generally useful. Dropout model is different from our conventional model in the sense that it has additional masking function applied to the output of each unit. If the unit is being dropped the masking value is zero and hence the output of the node becomes zero removing the unit from the network. The authors have used Bernoulli for this purpose. The paper describes a very effective regularization technique during training with Backpropagation which works very well along with dropout called *max-norm regularization*. In this technique, the Euclidean distance of weight vectors is restricted to be equal to or less than a fixed constant. One advantage of this technique is we can use a large learning rate without worrying about the weights taking up large values and the network not converging. Apart from max-norm regularization, dropout can be used with other methods which are known to improve gradient descent. Another way of training the dropout net is by *pretraining* where the weights calculated are multiplied by 1/p but this technique is susceptible to high learning rate.

The paper also concludes that using dropout decreases the error rate of the popular speech recognition task on the TIMIT dataset. Although a major drawback of using dropout is that training can take a lot of time. Dropout can be further improved by not keeping a fixed retention probability for all units and rather assigning a different retention probability for each hidden layer (Zhe Li, Boqing Gong, TianbaoYang, 2016).

REFERENCES:

**Zhe Li, Boqing Gong, TianbaoYang** (2016). Improved Dropout for Shallow and Deep Learning. [arXiv:1602.02220](https://arxiv.org/abs/1602.02220) [cs.LG]