# Improving neural networks by preventing co-adaptation of feature detectors

**Authors:** G.E.Hinton, N.Srivastava, A.Krizhevsky, I.Sutskever and R.R. Salakhutdinov
**Year:** 2012
**Journal:** Computing Research Repository

**Report:**

1. **Abstract:**

   “When a large feedforward neural network is trained on a small training set, it typically performs poorly on held-out test data. This “overfitting” is greatly reduced by randomly omitting half of the feature detectors on each training case. This prevents complex co-adaptations in which a feature detector is only helpful in the context of several other specific feature detectors. Instead, each neuron learns to detect a feature that is generally helpful for producing the correct answer given the combinatorially large variety of internal contexts in which it must operate. Random “dropout” gives big improvements on many benchmark tasks and sets new records for speech and object recognition.”

2. **Introduction:**
   - The huge change in the input training and predictive output will results in the varying of the weights of many hidden networks. This is very arduous.
   - And even the weights will be working fine on training data but not on test data,as they are being worked on them.
   - So, instead of varying weight vectors, the authors introduced a way of dropping of certain feature detectors randomly to increase the accuracy and reduce the error.
   - Moreover, it decreases the computational expense and the complexity of the neural networks architecture.

3. **Content and Approach:**
   - The article focuses on decreasing the feature detectors to decrease the complexity and which also surprisingly lessened the errors.
   - One of the general method to decrease the error is to train the model on different networks and create the average of the test data set.
   - Intuitively, the training and testing both cost high in computing and time. And moreover all the hidden networks of all these neural networks will have the same weights of the features.
   - The authors used the basic stochastic gradient descent procedure for training the dropout neural networks in the training case with mini-batches.

- The dropout is a process of leaving some of the features detectors, while training the data.
- This process is in fact simpler to use instead of using the Bayesian model averaging for calculating weights of each model of the training data.
- Furthermore they have not used the L2 norm for finding the errors on the whole data. Instead, they have created the threshold of the error which limits the value to a certain level.
- They will then subtract the values from the mean, so that the data will centre itself. This process will give the symmetry of the data.
- This helps in restricting the error to maximize to uncertain bounds, irrespective of the proposed weight updates' variance.
- During the testing , they have used the mean network of all the hidden units with the outgoing weights.
- Many of the experiments have been done to test the following algorithm on different datasets and data types.
- As the extreme probabilities of dropouts will result in worse results, a probability of 0.5 is taken throughout the experiments.
- A pre-training is done for the neural networks by the team. The learning rate is taken as 0.01. The bias of the each unit is taken as the log(p/(1-p)), where p is the mean activation of the unit in the dataset.
- The pretrained RBM(restricted boltzmann constant) was used to initialize the weights of the neural networks. That network is then fine tuned with dropout backpropagation.
- The process also results in the decreasing the probability of overfitting, as the unnecessary and infrequent features have been deleting from the dataset.

## 4. Results:

- The tests on MNIST had reduced the errors form 160 to 130 while implementing the feed-forward network with 50% dropout and to 110 by dropping out further 20% of the pixels of the images.
- The speech recognition data type TIMIT had also been resulted with better outputs .
- With the application of Hidden Markov Models, in the neural networks, with drop out it resulted in 19.7% recognition. This result had been the best till date for the information given without speaker identity.
- CIFAR-10, which had been benchmark dataset for object recognition gave 16.6% of error rate with drop outs which is less than the 18.5% without drop out.
- ImageNet and Reuters datasets also resulted in the significant changes of the error results with the application of dropout.

- The dropouts in all the hidden layers had given better results than the dropouts in one hidden layer.

5. **Discussion and Thoughts:**
    - Various terms had been defined in the article to support their approach. The terms from convolutional neural networks to neuron nonlinearities have been explained in detail.Moreover, the approach seems to be more flexible and simple.
    - It is also evident that the top features only define the most of the data as they are decreasing the significant amount (50%) of features from the data.
    - The results also shows that the below 50% features instead of helping in the prediction, they are disturbing the accuracy due to their variance and uncommon features.
    - The approach seems to be so structured in the article, but the depiction had been so clumsy.
    - The advancement also seems to reduce the overfitting of neural network in the field of machine learning.
    - The experiments on various datasets resulted in building the confidence of the usage of dropouts in various applications.

6. **Conclusion:**
    - Instead of creating something to decrease the error rate, the article concentrated on reducing the feature detectors of the data.
    - This made the procedure so simple and abstract, that it decreased the both computational expense and the complexity.
    - Moreover, it also resulted in significant decrease of the error percentage in all types of data and various datasets.
    - This process seems to open the gates for the learning of more complex data with low error rates, by decreasing the features randomly.