STEVENS INSTITUTE OF TECHNOLOGY

Dropout in Deep Learning

Rensheng Wang,
https://sit.instructure.com/courses/34886

What is Dropout in Neural Networks?

- Dropout refers to ignoring units (i.e. neurons) during the training phase of certain set of neurons which is chosen at random.
- By ignoring, it means these units are not considered during a particular forward or backward pass.
- More technically, At each training stage, individual nodes are either dropped out of the net with probability 1-p or kept with probability p, so that a reduced network is left; incoming and outgoing edges to a dropped-out node are also removed.

Why Dropout?

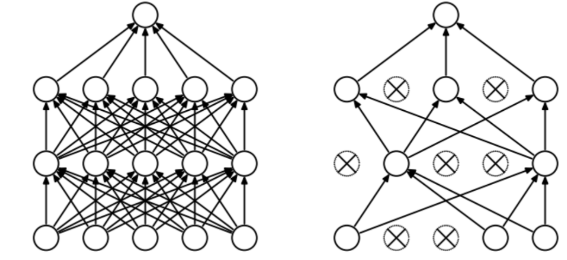
- Why do we need dropout at all? Why do we need to literally shut-down parts of a neural networks?
- : The answer is: "to prevent over-fitting".
- A fully connected layer occupies most of the parameters, and hence, neurons develop co-dependency amongst each other during training which curbs the individual power of each neuron leading to over-fitting of training data.

Regularization vs. Dropout

- Recall that in machine learning, regularization is way to prevent over-fitting. Regularization reduces over-fitting by adding a penalty to the loss function. By adding this penalty, the model is trained such that it does not learn interdependent set of features weights.
- Dropout is an approach to regularization in neural networks which helps reducing interdependent learning amongst the neurons.

Dropout

- **Training Phase**: For each hidden layer, for each training sample, for each iteration, ignore (zero out) a random fraction, p, of nodes (and corresponding activations).



(a) Standard Neural Net (b) After applying dropout.

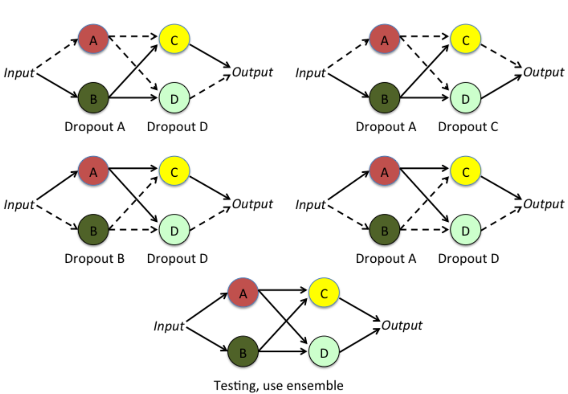
- **Testing Phase**: Use all activations, but reduce them by a factor p (to account for the missing activations during training).

Dropout

- Dropout forces a neural network to learn more robust features that are useful in conjunction with many different random subsets of the other neurons.
- Dropout roughly doubles the number of iterations required to converge. However, training time for each epoch is less.
- With H hidden units, each of which can be dropped, we have 2^H possible models. In testing phase, the entire network is considered and each activation is reduced by a factor p.

Why Dropouts Prevent Overfitting?

- Consider the case where 2 hidden layers with neurons A and B in one, and C and D in second.
- We want to train AC, AD, BC and BD all to learn the relation between input and output. Therefore, we have 4 models learning the same relation.
- For a 2 layer model with 100 neurons in each layer, this results in a scenario where average over billion possible models. As a result, the tendency to overfit is significantly reduced.



L2 Parameter Regularization

- λ is the regularized hyperparameter. As λ increases, the bias increases (and the model becomes less flexible) with the following extreme cases
- $\lambda = 0$, no regularization.
- $\lambda \to \infty$, model becomes very simple where all weights are essentially zero. In the case of regression, we would end-up with the intercept only equal to average of the target variable.