Student ID : 20194293

Name : Go, Kyeong Ryeol

[AI502] Dropout: A Simple Way to Prevent Neural Networks from Overfitting

1. Paper Summary

Although deep neural networks are flexible enough to learn any complicated functions, it is vulnerable to overfitting. There are various kinds of regularization techniques to handle this issue. This paper introduces a technique named as "Dropout" which randomly select the units to drop during the training phase. In the test phase, every unit are now active and instead each parameter is multiplied by p to ensure that the expected output stays the same as it was in the training phase.

By doing so, this single neural network is indirectly taking an average of the exponential number of different models whose parameters are shared. This is kind of well relaxed approximation on Bayesian gold standard. Also, as the active units are chosen stochastically, sparsity are achieved and the co-adaptation among the units are broken so that the model gets robust. This was shown by the hidden unit activation plot, the learned feature plot and less generalization error in various domains like image, speech and text. (e.g. MNIST, TIMIT, CIFAR-10/100, SVHN, ImageNet, Reuters-RCV1). Moreover, dropout shows significant improvement when used with max-norm regularization, large decaying learning rates and high momentum. It significantly outperforms many other models such as SVMs with PCA and shows comparable performance to Bayesian neural network with considerably less computation.

Other important side remarks to refer from the paper are follows.

- Dropout can be regarded as an extension of the Denoising Auto-Encoder which drops the values only in the input layers. This implies its validity of use not only for the supervised setting, but for unsupervised setting like the ordinary autoencoders.

- Rather than multiplying the dropout rate sampled from the Bernoulli distribution, it may be better to add the noise from the standard normal distribution. Any case among the two is just fine as the expected values of the activations remain unchanged.

- By using dropout, the training typically takes 2~3 times longer and the parameter updates get noisy, which are the trade-offs occurred to avoid overfitting. One way to handle the inherent stochasticity of the dropout is to marginalize the noise.

2. Discussion

Here, I want to offer two discussion points. To begin with, what variants would be possible during the test phase rather than just multiplying p on each parameter? To better approximate the Bayesian golden standard and take affordable time complexity, [the number of training steps where certain unit are active / the number of training steps] * p can be rather multiplied to each corresponding parameter for prediction. Next, is dropout always a nice choice when used along with the stochastic models? Most of the models referred in the paper are deterministic. Since too much is as bad as too little, the amount of stochasticity must be handled elaborately. Therefore, the validity of using dropout on stochastic models like Variational Auto-encoder must be further checked.