Assignment 7 - Dropout

Nisim Hurst

Wednesday 28 February 2018

Abstract

The paper was written by Geoffrey Hinton in (Srivastava et al. 2014) in an attempt to propose a new technique for reducing overfitting in neural networks. Combining the output of many slow to use neural networks can be expensive at test time. The idea is to randomly drop units and their connections producing a thinned network. The resulted network has smaller weights but all the units. The domains such as computer vision, speech recognition, document classification and computational biology were tested.

Hypothesis

Sexual reproduction is effective due to its capability to mix genes. Each node learns to cooperate with other random nodes. Also small conspiracies are more effective, less biased and more independent than bigger ones, each allows to find a global minima strategy. Dropout is a technique inspired on these intuitions and that will help neural network models generalize during training.

Evidence and Results

The article presents evidence that dropout can be useful in visible and hidden units and also be extended to supervised learning problems and *Restricted Boltzman Machines* (RBS). The deterministic counterpart of marginalizing the error is also explored. Dropping out 20% of the visible/input units and 50% of the hidden units was found to be optimal.

The results show that dropout is more useful when used in combination with the max-norm regularization, momentum and weight decay. Max-norm was proved to be useful applied to stochastic gradient descent even when no dropout is used. The reason is that it provides a way of start with a high learning rate without loosing model stability.

Several domains were tested by using the following datasets: MNIST (images), TIMIT (speech recognition), CIFAR (images), Google StreetView numbers (images), ImageNet (images), RCV1 (natural language processing).

It achieved error reduction to 1.35%. Other techniques can further reduce the error to 0.95%. It doesn't even need early stopping. Convolutional layers and ReLu architectures were found to be useful in image datasets even though the layers are fully connected and no data augmentation was used.

In speech recognition the improvement wasn't so great but good enough, from 23.4% to 21.8% or 22.7% to 19.7% with pre-training.

In document classification achieved 29.62% versus 31.05% on a bag of words order less features.

Dropout can be compared to Bayesian neural networks by seeing the model as an equally-weighted averaging of many models. However, it does not take into account the prior for the weights. Nonetheless, Bayesian nets are expensive and slow. Input dimensionality reduction can also help to prevent overfitting but it can be circumvented by using dropout.

Appendix B provides more details of all the experiments made.

Contribution

Supervised training using backpropagation

The article describes a model which for applying dropout that consist of modifying gradient descent by sampling a thinned network for each case in a mini-batch:

1. The feed forward operation is modified by adding a probability of maintaining a forward hidden or input unit using a Bernoulli distribution. Thus, each layer is aggregated with a similar sized vector that has the probability of maintaining a unit and the result is multiplied element wise to produce a *thinned network*.
2. Backpropagation also uses the same thinned network.
3. The resulting network is tested without dropout.

Unsupervised pre-training

The procedure stays the same however the weights should be scaled by a factor of $\frac{1}{p}$ to make sure the output will be the same as the expected output. The learning rates also should be smaller to maintain the pre-trained weights even though its stochastic nature (vanishing gradient).

Dropout can be seen as an alternative to L_2 regularization and PCA for reducing overfitting at low costs.

The model using convolutional layers won the ILSVRC-2012 competition.

The article also shows a model for applying dropout the Restricted Boltzman Machines and its properties.

Weaknesses

The article does not explain the relationship between mini-batch size and the dropout proportion for each layer and their distance to the output layer. The article is overly empirical and does not provide theorical mathematical justification for the dropout proportion. However, it hints that dropout in hidden layers are prone to improve accuracy in the model and the results in multifarious domains constitute a solid evidence.

Future Work

On section 7 the article explores the effect of dropout on the quality of features produced. For example it increases sparsity of hidden units activations depending on the probability of retaining units leading to sparse representations. Dropout prevents coadaptation (error amelioration by scion unit collaboration on the same layer). Thus, the unit must preform well alone and allow it to detect special features per layer (like edges or strokes in images).

Dropout rate is another hyperparameter to tune up *manually*.

Future work could use the results of the *hidden units constant test* and *dropout constant test* to explain why there is a *sweet spot* over the amount of data related to the architecture.

References

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." *Journal of Machine Learning Research* 15: 1929–58. doi:10.1214/12-AOS1000.