Dropout: A Simple Way to Prevent Neural Networks from Overfitting

Summary written by Sahana Srihari
March 16, 2020

Summary: The paper addresses one solution to the problem of overfitting in Deep Neural Networks called Dropout regularization. The method finds a way to efficiently average an exponential combination of different network architectures. This is achieved through the proposed method of randomly setting some units in the network to zero conditioned on a random probability during the training phase and during the testing the probability *p* of the units is a used as a multiplier. The paper presents motivation and extensive experimental proof on the proposed methods.

Related work: The author's work is heavily influenced by the work of Denoising Auto Encoders(DAEs) [5][6] for noise-free output amidst noisy input. The deterministic part of the stochastic dropout regularisation is drawn from[4] which explore noise distributions. Marginalising the output dropout noise showed results of increased speed, as seen in [7]. [1] showed how autoencoders marginalized for noise whereas [3] [2] highlighted the minimization of loss achieved through dropping units in a network.

Approach: Given a neural network with parameters - L hidden layers, $z^{(l)}$ inputs, $y^{(l)}$ output, $W^{(l)}$ weights and $b^{(l)}$ biases at layer l & $r_j^{(l)}$ -vector from Bernoulli distribution for assigning probabilities to units. During feed forward operation, the probabilities are sampled from this and multiplied with the inputs to produce thinned outputs $\tilde{y}^{(l)}$.The $\tilde{y}^{(l)}$ are inputs to the successive layers and during learning & back propagation the weights are scaled by this probability $\implies$ W$^{(l)}_{test}$ = $pW^{(l)}$. The main difference is that training is done on a mini-batch of thinned networks and controlled with a hyperparameter *p*.

Datasets, Experiments and Results:

(i) MNSIT: Images(28x28) for identifying handwritten digits. Dropout + maxout gave low generalization error(0.94%) compared to standard neural nets(1.6%). The best performance was attained through using Deep Boltzman Machine(DBM) + dropout finetuning - 0.79%.

(ii) Street View House Numbers(SVHN): Images(32x32) of house numbers.Dropout + CNNs(all 3 convolution & 2 FC layers) extremely low error(2.47%) was achieved compared to 3.02% & 3.95% for dropout only in FC layers and no dropout respectively.

(iii) CIFAR-10 and CIFAR-100: Images(32x32) drawn from 10 & 100 categories. Dropout + maxout constraints(11.68%) performs better than Bayesian Hyperparamter optimization(14.98%) for CIFAR-10 and a great drop in error of 6.28% is seen for CIFAR-100.

(iv) ImageNet: Large database of images from 1000 categories. Dropout + CNN achieved top-1 and top-5 error rate of 37.5% and 17% in comparison to standard vision features($\sim$ 47% & $\sim$ 28%) with an impressive test error of 16.4%.

(v) TIMIT: Speech dataset containing 680 speakers of 10 American dialects, phone error rate(PER) used as a metric. Dropout on 6-layer net gives PER of 21.8% over standard(23.4%).4-layer net + RBMs(22.7%) in comparison With dropout reduces to 19.7% & an 8-layer reduces error to 19.7%.

(vi)Reuters-RCV1: Text dataset containing 800,000 newswire articles.Using dropout + neural net decreased the error from 31.05% to 29.62%.

(vii) Alternative Splicing data set: Genetic dataset to predict alternative splicing based on RNA features with Code Quality as a metric. Dropout(567 bits) achieves close performance to Bayesian Networks(623) which highlights its strong regularization effects.

Effects of tuning parameter *p* were tested out in two constraints as well as considering the effect of datasize. The paper also presents an adaptation of dropout to Restricted Boltzman Machine which proved picking efficient features due to reduction in co-adaptation & sparsity. Information on how dropout being an adaptive regularizer through linear and logistic regression is also provided.

Strengths: It solves the problem of overfitting in a relatively simple manner.Stress on the GPU is reduced as few neurons are learnt at a given time for every epoch.It is generalized technique that boosts performance in multiple domains.It allows for good reconstruction and meaningful features due to restriction of co-adaption of neurons.It automatically gives sparse representation in the hidden layers.Although dropout roughly doubles the number of iterations needed to converge,it reduces the time taken for each epoch.

Weaknesses: The major weakness of dropout is its long training times, without convergence it gives horrible results. The process of tuning newer and larger architectures from dropout can be quite tedious. For small datasets dropout lowers the performance. For each new architecture, the gradients computed are not the final gradients and a significant amount of noise in introduced.

Reflections: With the advent of batch normalization, dropout has been used lesser in newer papers. Dropout was a very intuitive way of improving the training of certain architectures and engendered variations of its kind and other regularization techniques. It definitely revitalized the deep learning industry and paved the way for faster, larger and more complex neural networks to be used today.

References

[1] M. Chen, Z. Xu, K. Weinberger, and F. Sha. Marginalized denoising autoencoders for domain adaptation. *arXiv preprint arXiv:1206.4683*, 2012.

[2] O. Dekel, O. Shamir, and L. Xiao. Learning to classify with missing and corrupted features. *Machine learning*, 81(2):149–178, 2010.

[3] A. Globerson and S. Roweis. Nightmare at test time: robust learning by feature deletion. In *Proceedings of the 23rd international conference on Machine learning*, pages 353–360, 2006.

[4] L. Maaten, M. Chen, S. Tyree, and K. Weinberger. Learning with marginalized corrupted features. In *International Conference on Machine Learning*, pages 410–418, 2013.

[5] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

[6] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(Dec):3371–3408, 2010.

[7] S. Wang and C. Manning. Fast dropout training. in proceedings of the 30th international conference on machine learning. 2013.