The importance of stochasticity in DropOut

Advanced Topics in Machine Learning

Surya Teja Chavali
Tuesday, 3rd May 2016

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Introduction

- Dropout, a common regularization mechanism in neural nets, can be viewed as a bagging strategy - training multiple models on small subsets of the data, and combining them by taking the (geometric)mean of their predictions.
- The major difference is that the ensemble of models shares parameters, and most models are not trained .
- We seek to study the boosting, bagging(model averaging) and regularization aspects of DropOut

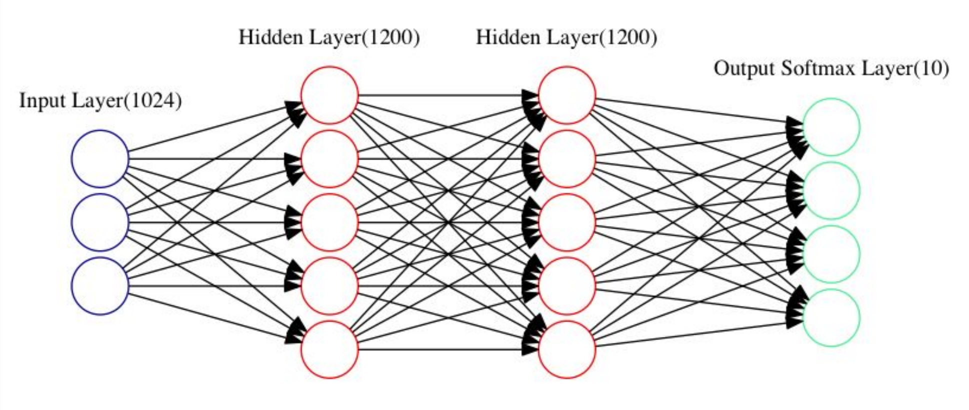
Motivation

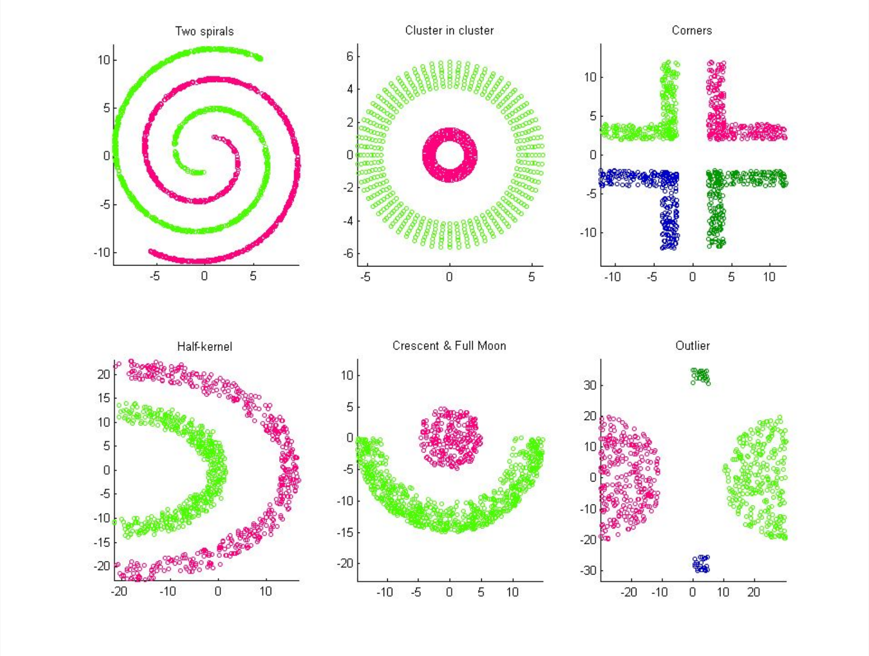
We seek to answer the following questions:

● DropOut seemingly samples several sparse models at random, and averages them. Can we re-weight a (possibly smaller) number of sparse models and average them? ● It also seems that weights which have abnormally small or large magnitudes are probably over-fitting.	● Drop weights in the band of $[\mu - p\sigma, \mu + p\sigma]$ while training the net, where μ is the mean weight, σ is the standard deviation of the weights, and p ($0<p\leq 1$) is a user-specified parameter. ● Multiply all weights by 0.5 during test time. ● What if we divide each weight by (#times dropped + 1) at test time? ● What about absolute values of weights? ● What if we did just the opposite?
Is the uniform distribution the best distribution to sample the weights to drop out?	Try other distributions. Specifically, a Gaussian?

Experimental setup

- Used MNIST dataset as well as six artificially generated datasets.
- Vectorized MNIST to a 1024 dimensional vector, and normalizing all pixel values by dividing them by 255.
- In each experiment, we considered three activation functions: ReLU, tanh and sigmoid. The value of p was increased from 0.05 to 1 in steps of 0.05. The 'rethink' phase of the net was applied epochally. Loss and error were plotted.



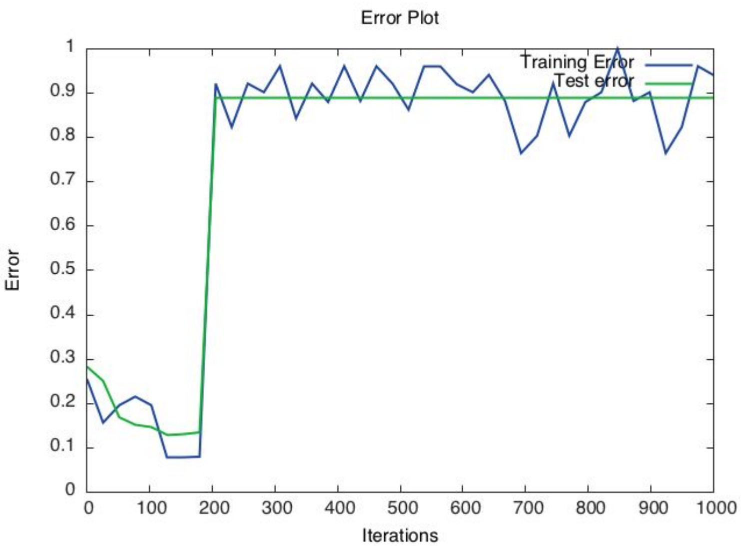


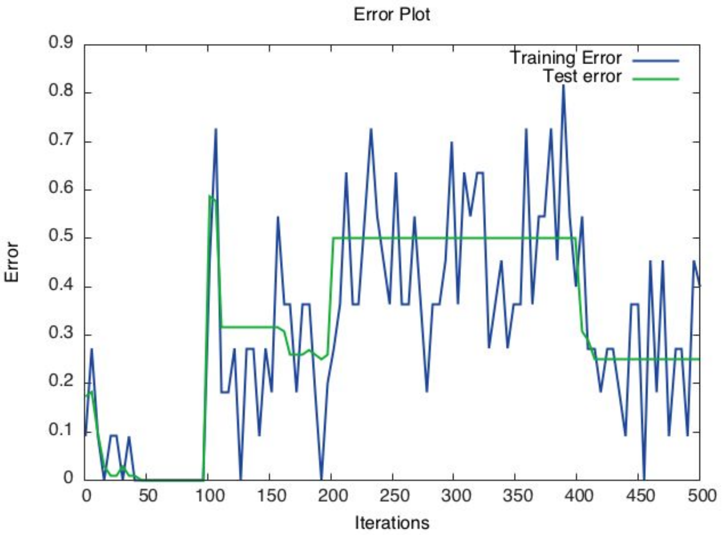
The deadly seven

- Drop weights in the band $[\mu-p\sigma, \mu+p\sigma]$ during training. Halve the weights while testing.
- Drop weights outside the band $[\mu-p\sigma, \mu+p\sigma]$ during training. Halve the weights while testing.
- Drop weights in the band $[\mu-p\sigma, \mu+p\sigma]$ during training. Divide weights by (#of times dropped) + 1 while testing.
- Drop weights outside the band $[\mu-p\sigma, \mu+p\sigma]$ during training. Divide weights by (#of times dropped) + 1 while testing.
- Drop weights whose absolute values are in the band $[\mu-p\sigma, \mu+p\sigma]$ during training.
- Drop weights whose absolute values are outside the band $[\mu-p\sigma, \mu+p\sigma]$ during training.
- Use a Gaussian to sample px(#of weights) numbers, and drop the weights in a small band of each sampled number during training.

Results

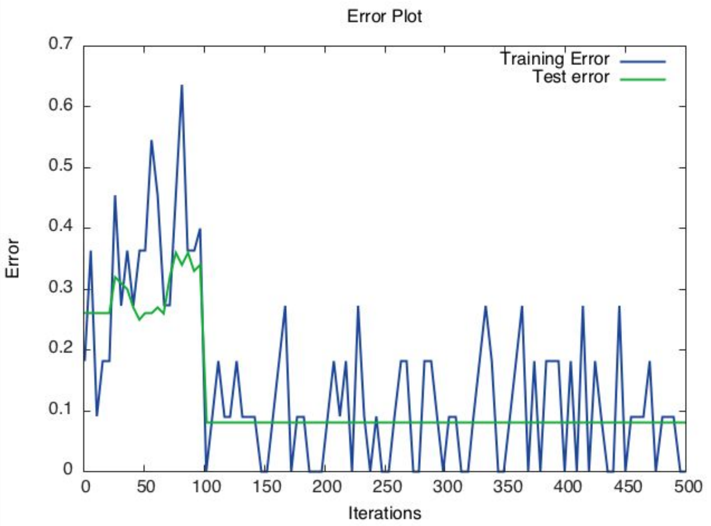
- The plot on the right is representative of what happened with MNIST, with tanh as well as ReLU activations despite varying p.
- Tried the same thing with a large number of epochs; the same result occurred.

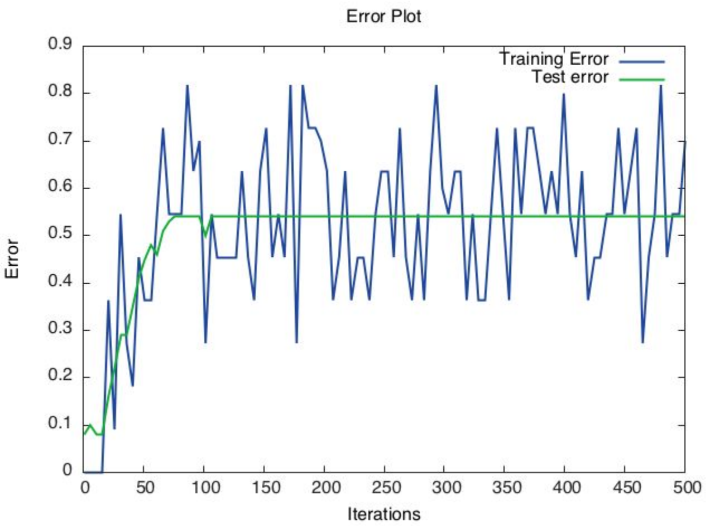




Gaussians?

- In the case of sampling weights from a Gaussian, we observe a peculiar phenomenon.
- After the first epoch, the error sharply increases or decreases, and settles down.
- In other words, we ‘converge’ after the first epoch to more or less the same error rate.
- Can also remain stable at one level throughout.





Key Takeaways

- The result of this study has been resoundingly in favour of randomness in DropOut.
- It seems the randomness of DropOut/DropConnect allows them to theoretically average the entire space of models, while our determinism restricts the space of models to a small(and particularly bad?) subset of the model space, causing us to under-fit miserably.
- More work needed before we conclude that sampling weights to drop from the uniform distribution is the best.
- On the other hand, we feel there is a large scope for interpretation of DropOut in the sense of Pseudo-Ensembles, as well as in the regularization sense.

Demonstration

Any Questions?

Thank you!