An empirical analysis of DropOut in Piecewise Linear Networks

Advanced Topics in Machine Learning

Surya Teja Chavali
Thursday, 14th April, 2016

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 and Key Contributions

The key contribution of this paper is to analyze DropOut in a nuanced manner in the context of ReLU's. It answers the following questions:

- How close does DropOut come to approximating the GM of the models?
- How good a proxy is GM for AM - in terms of classification accuracy?
- Does DropOut work by exploiting co-ordination amongst the models, or by model averaging, or by encouraging each unit to work well in a variety of tasks?
- In DropOut, we do not update the current model using gradients from previous models. Can we use that information as well to perform **"DropOut Boosting"**?

Earlier Work

DropOut has been studied and extended by several researchers in the past:

- Wang and Manning, 2013; Wager et al., 2013 study DropOut in Linear Models.
- Baldi and Sadowski, 2013 study DropOut in Linear and Sigmoidal Models - characterize the approximation to the Geometric Mean.
- Cartwright and Field, 1978 show that the difference between AM and GM is bounded.
- Goodfellow et al., 2013a show that, in MLPs with no non-linearity applied to each unit, with SoftMax output, DropOut is equivalent to GM-based model averaging.
- Srivastava (2013); Goodfellow et al. (2013a) previously investigated the fidelity of the weight scaling approximation in the context of rectifier networks and maxout networks, respectively, through the use of a Monte Carlo approximation to the true model average.

Methodology

The following investigations were carried out to answer the questions raised:

How good is DropOut's approximation to the GM?	Compare against the true average, in networks small enough for exact computation.
How does using GM instead of AM affect accuracy?	Compare accuracy in AM-averaged and GM-averaged nets.
Why exactly does DropOut work?	Train a set of independent models on resamplings of training data, with a fixed random DropOut mask, and recombine the ensembles in many ways. Compare performance against an identical DropOut-based net.

DropOut Boosting -1

- Can DropOut be seen as a means to make the model robust to input perturbations, or is the model averaging perspective more important?
- Propose an algorithm that injects exactly the same noise as dropout - DropOut Boosting.
- Objective Function:
  $log(p_{ensemble}(y|v;\theta))$

$$p_{\text{ensemble}}(y \mid v; \theta) = \frac{1}{Z}\tilde{p}(y \mid v; \theta)$$

$$Z = \sum_{y'} \tilde{p}(y' \mid v; \theta)$$

$$\tilde{p}(y \mid v; \theta) = \sqrt[2^{|\mathcal{M}|}]{\Pi_{\mu \in \mathcal{M}} p(y \mid v; \theta_\mu)}.$$

DropOut Boosting - 2

- Learning Rule:

$$\Delta\theta_\mu \propto \frac{1}{2^{|\mathcal{M}|}} \left(\nabla_{\theta_\mu} \log p(y \mid v; \theta_\mu, \mu) + \sum_{y'} p_{\text{ensemble}}(y' \mid v) \nabla_{\theta_\mu} \log p(y' \mid v; \theta_\mu, \mu) \right)$$

- Weight sampling approximation to estimate the second term.
- Alternatively, maximize the log likelihood by averaging together the gradient for multiple values of μ, and optionally using a different μ for the term in the left and the term on the right.
- Empirically, ‘boosting’ - i.e., using the same μ on both sides did best.

Datasets

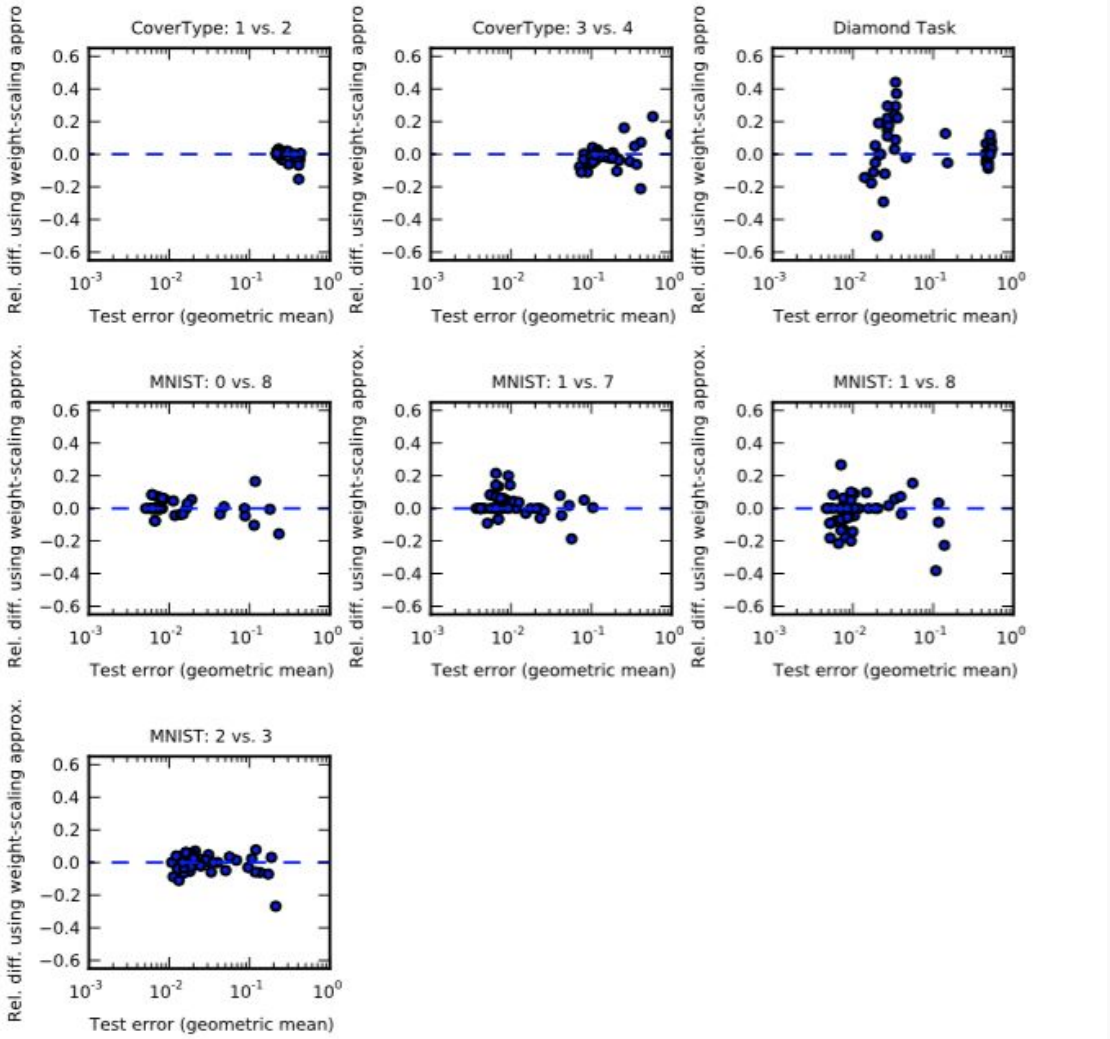
- Four binary tasks from MNIST: 1 vs. 7, 1 vs. 8, 0 vs. 8, and 2 vs. 3
- Two binary sub-tasks from the CoverType dataset of the UCI Machine Learning Repository- classes 1 and 2 (Spruce-Fir vs. Lodgepole Pine) and classes 3 and 4 (Ponderosa Pine vs. Cottonwood/Willow).
- The final task is a synthetic task in two dimensions: inputs lie in $(-1, 1) \times (-1, 1) \subset R2$, and the domain is divided into two regions of equal area: the diamond with corners (1, 0), (0, 1), (−1, 0), (0, −1) and the union of the outlying triangles. In order to keep the synthetic task moderately challenging, the training set size was restricted to 100 points sampled uniformly at random. An additional 500 points were sampled for a validation set and another 1000 as a test set.

Experiments

- Used rectifier networks with 2 hidden layers and 10 hidden units per layer, and a single logistic sigmoid output unit.
- To keep computation tractable, applied DropOut only to hidden layers.
- This also simplified computation, as $p>0.5$ is considered for input layer dropping(Hinton uses $p=0.8$)
- Hyperparameters chosen by random search over learning rate, momentum and mini-batch size.
- Early stopping on the validation set, terminating when a lower validation error had not been observed for 100 epochs.
- When training with dropout, the figure of merit for early stopping was the validation error using the weight-scaled predictions.
- Randomly sampled 50 sets of hyperparameters and trained 50 networks with dropout.
- Computed, for each point in the test set for each task, the activities of the network corresponding to each of the 2^{20} possible dropout masks.

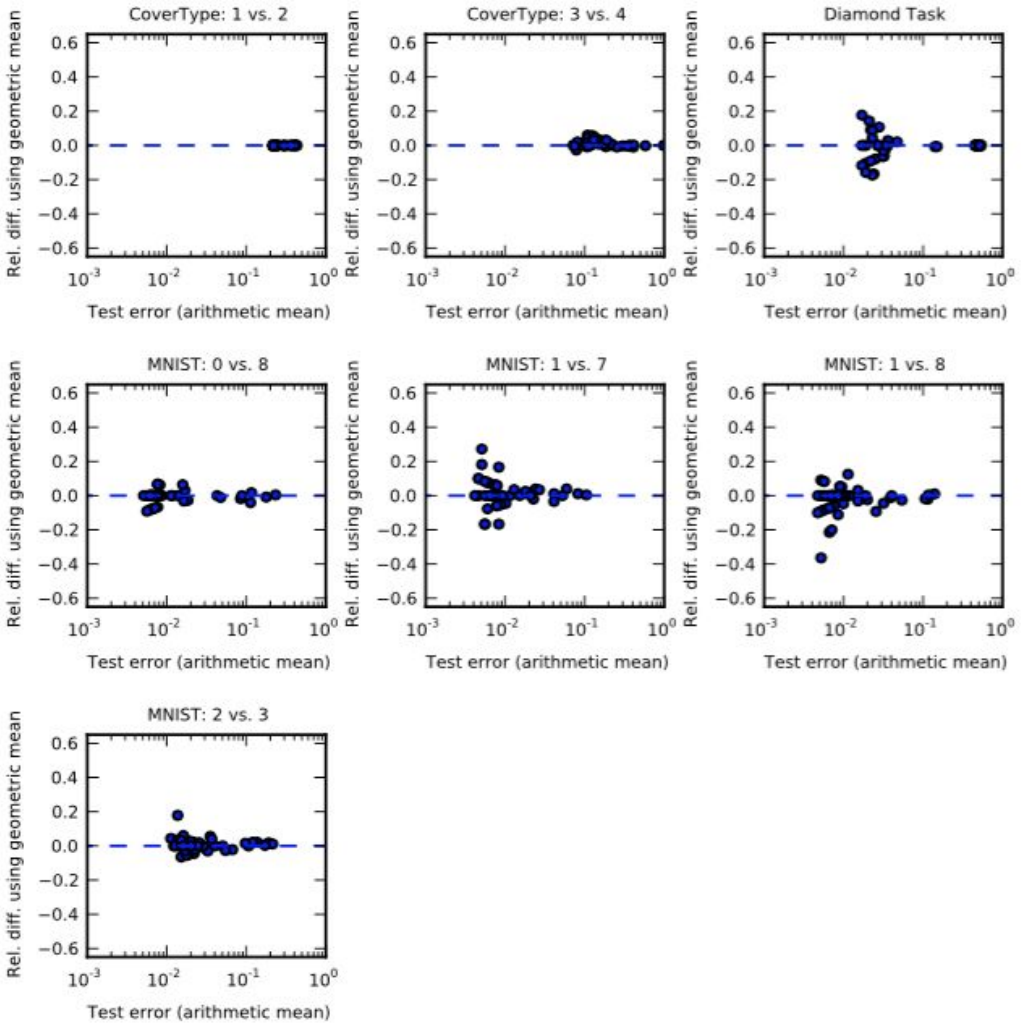
Results: How close are we to the GM?

- Compare true GM to Weight Averaging
- They're pretty close most of the time.



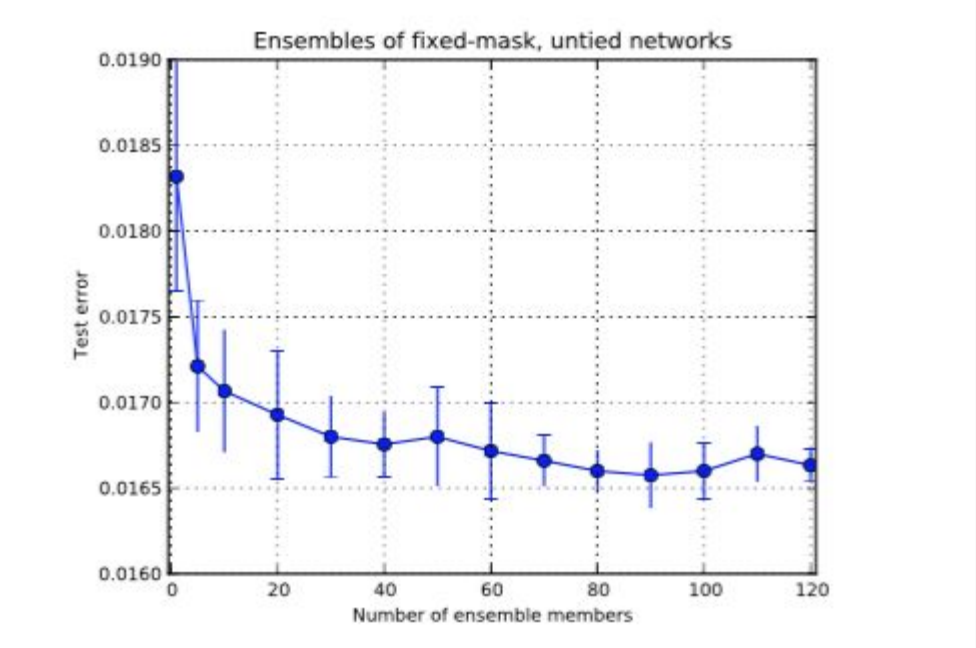
Results:
AM vs. GM

- Compare classification accuracy of GM approximation and true AM.
- They're pretty close too!



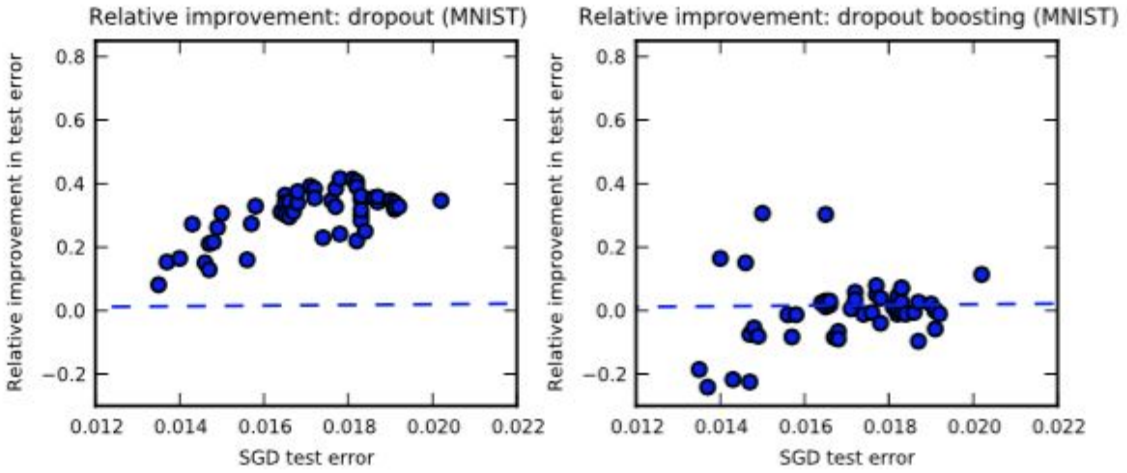
Results: Dropout ensembles versus untied weights

- Use the standard MNIST architecture now.
- Use the best of 50 random hyperparameter setttings
- Error: 1.06%
- Using the same hyperparameters, train 360 models initialized with different random seeds, on different fixed DropOut masks
- Error 1.66%, far above the even the sub-optimally tuned networks trained with dropout.



Results: Bagging vs. Boosting

- Dropout Boosting employs exactly the same noise as regular dropout uses to perform bagging
- Should perform similarly to conventional dropout if learned noise robustness is the important ingredient.
- With the 50 models trained before, used the same hyperparameters to train a set of 50 networks with dropout boosting, and another with plain stochastic gradient de- scent.
- DropOut does much better, but DropOut Boosting fails to beat SGD by a large margin.
- It hence looks like viewing DropOut as a model averaging mechanism is perhaps more appropriate.



Analysis and Conclusions

The answers to the questions posed are as follows:

How good is DropOut's approximation to the GM?	Pretty close. Relative error usually stayed within 20%, and crossed 40% only once.
How does using GM instead of AM affect accuracy?	Not too much. Easy computation is a positive externality. Relative error usually stayed within 20%, and crossed 30% only once.
Why exactly does DropOut work?	Weight sharing taking place in the context of dropout (between members of the implicit ensemble) plays an important role in further regularizing the ensemble.
Bagging vs. Boosting	Employing masking noise with the exact same distribution as is employed by dropout, yields no robustness gains over networks trained with ordinary SGD.

Some thoughts

- Another step in understanding why a heady mix of surprisingly simple ideas like ReLU's and DropOut lead to such remarkable accuracy
- Need to further investigate why parameter-sharing methods are good in the context of boosting/bagging.
- Randomness seems to be a very critical part of DropOut - we tried a deterministic DropOut schema with limited success.
- Any 'blessing of depth' that a large ensemble of learners doesn't have over a deep net with DropOut?
- Why does DropConnect not do significantly better than DropOut?

Reference

An empirical analysis of dropout in piecewise linear networks

David Warde-Farley, Ian J. Goodfellow, Aaron Courville, Yoshua Bengio

De´partement d'informatique et de recherche ope´rationnelle

Universite´ de Montre´al

Montre´al, QC H3C 3J7

Any Questions?


Thank
you!
MOVIECLIPS.COM