JMLR: Workshop and Conference Proceedings 42:57-67, 2015 HEPML 2014

Dissecting the Winning Solution of the HiggsML Challenge

Gábor Melis MEGA@RETES.HU
Fixnum Services

Editor: Cowan, Germain, Guyon, Kégl, Rousseau

Abstract

The recent Higgs Machine Learning Challenge pitted one of the largest crowds seen in machine learning contests against one another. In this paper, we present the winning solution and investigate the effect of extra features, the choice of neural network activation function, regularization and data set size. We demonstrate improved classification accuracy using a very similar network architecture on the permutation invariant MNIST benchmark. Furthermore, we advocate the use of a simple method that lies on the boundary between bagging and cross-validation to both estimate the generalization error and improve accuracy.

Keywords: neural networks, deep learning, cross-validation, bagging, high-energy physics

1. The Challenge

The task in the HiggsML challenge (http://higgsml.lal.in2p3.fr/) was to identify simulated particle collider measurements likely to originate from Higgs boson to tau-tau decay. Fortunately, no physics knowledge was required to compete for the top positions and even less to read this paper.

On first sight, the problem posed by the contest organizers was that of highly unbalanced weighted binary classification. The tau-tau decay events (called "signal") were outnumbered by the "background" events 600-to-1 which reduced the effective sample size greatly. Worse, the score was defined by a non-linear (but convex) function of the true positive and false positive rate. This function, the Approximate Median Significance (or AMS, for short) and the data set were such that roughly one sixth of the examples were classified as positive by typical solutions. Consequently, the majority of test set examples did not feature in the score.

While in theory it was possible to directly propose a set of examples deemed positive to maximize the AMS, in practice this was done in two stages. First, a classifier was trained that predicted the probability of an example being in the positive class. Then a certain number of test examples (those with the highest predicted probabilities) were classified as positive. The decision threshold (often referred to as "cutoff") was usually determined by cross-validation performed on the training set to maximize the AMS.

The problem is that the AMS vs cutoff curve tends to be very jagged. To find the location of the expected peak one has to make assumptions about overall shape of the curve which is difficult, because features may provide discriminative power in rather localized regions of the confidence domain. That is, bumps in the AMS curve can be explained by locally discriminative information just as much as by noise. For these reasons, instead of

reporting AMS values at a single cutoff, plots of the most interesting region of the AMS curve are included.

After the competition ended, the previously hidden labels for the test were made public and the labeled training and test sets were released as the opendata data set (ATLAS collaboration, 2014) upon which the post-mortem analysis presented in this paper is based.

2. The Winning Model

The final model was an ensemble of 70 neural networks whose predicted probabilities were combined by simple arithmetic averaging. The neural networks only differed in their initializations and training sets. Every neural network had three fully connected hidden layers with 600 neurons each. The output layer consisted of 2 softmax units representing the two classes. With the 35 input features that is 1,102,200 parameters per network. The hidden layers had Local Winner-Take-All activations (Srivastava et al., 2013) with block size 3. Equivalently, they can be described as channel-out activations with selection function argmax (Wang and JáJá, 2013). In a nutshell, these activation functions zero out their inputs except for the maximum in every block of 3:

$$f(x_i) = \begin{cases} x_i, & \text{if } x_i = \text{argmax}_{j \in \{\lfloor i/3 \rfloor \cdots \lfloor i/3 \rfloor + 2\}} x_j \\ 0, & \text{otherwise} \end{cases}$$

All of the above choices of the model itself, its architecture and parameters were made after careful local cross-validation. In the following sections, we describe the most important characteristics of the winning solution. The source code is available at `http://github.com/melisgl/higgsml`.

3. CV Bagging

It was independently discovered by multiple contestants that cross-validation did not provide a reliable estimate of the AMS. The natural solution was to use repeated cross-validation where the results of several cross-validation runs performed with different splits of the data are averaged. Repeated cross-validation requires training $C \times K$ models where C is the number of repetitions and K is the number of folds. When each model is a bag of B constituent models, then the number of models to train becomes $C \times K \times B$. This can easily be prohibitively expensive.

One way around that is to build the ensemble from the models trained during cross-validation. This is equivalent to Subagging (Bühlmann, 2003) with the twist that instead of random subsampling, the training/test splits are generated as in k-fold CV. This way, after training $C \times K$ models we have exactly C predictions for every example in the training set which makes it possible to use fewer of models to estimate the generalization error. Note that for example the R adabag library's `bagging.cv` function (Alfaro et al., 2013) also builds an ensemble based on CV splits, but it doesn't do repeated CV which we found to be crucial for both estimating generalization error and improving prediction accuracy.

In the contest, 2-fold stratified CV Bagging was used with 35 repetitions. This was very much an overkill and all experiments in this paper are based on 10 repetitions. Figure 1 illustrates how the peak AMS improves as more models are added to the bag.

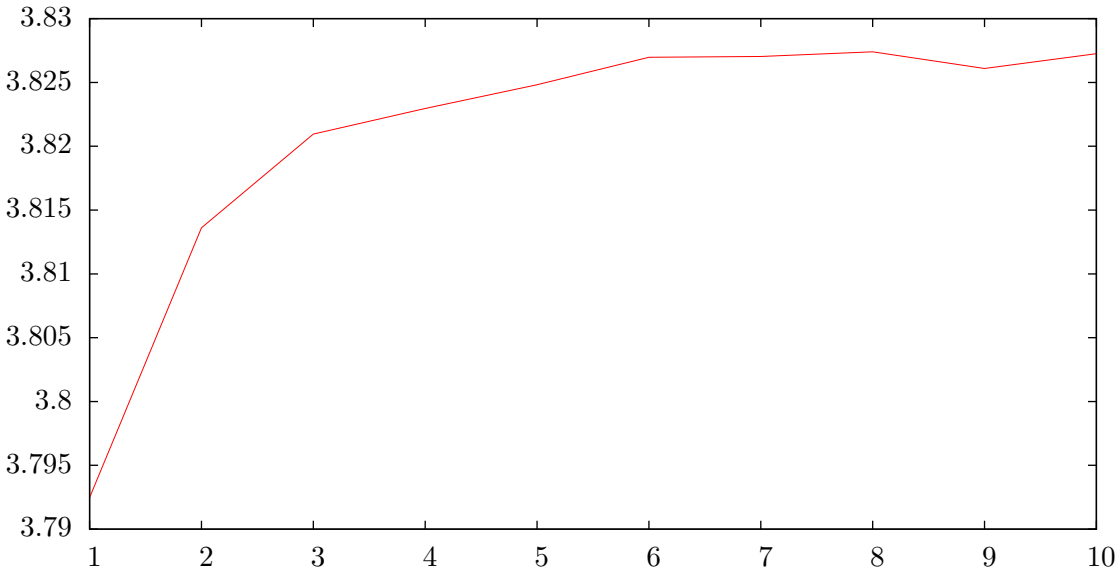

Figure 1: Out-of-sample estimates of peak AMS vs number of CV runs in CV Bagging using the entire opendata data set. Average of 25 runs.

Reliable model evaluation is key in most contests and it was doubly important in HiggsML. Figure 2 shows how the out-of-bag estimation on a training set of 250,000 examples compares to results obtained from the remaining 550,000 examples in the rest of the opendata data set. The difference between the two curves is within one standard deviation estimated by bootstrap. Note that none of the AMS scores reported in this paper are directly comparable to the results from the contest due to the use of different data sets.

3.1. Feature Selection and Preprocessing

The data provided by the contest organizers consisted of 30 features for each training example of which 17 were primary features corresponding to actual measurements and 13 were derived features combining primary features according to some formula (Adam-Bourdarios et al., 2014).

It was observed that the azimuth angle features greatly increased the variance of the predicted probabilities and were dropped. While there was a large positive effect on the score of individual models with the angle features dropped, post-mortem experiments show that the loss of the accuracy was clawed back if enough models were bagged (see Figure 4).

Features with long tailed distributions were log transformed to reduce the variance of the gradient. As the last step, all primary, derived, log transformed features were normalized to zero mean and standard deviation one.

Missing values were represented by zero (not subject to normalization). No missing value indicators were added, because the presence of all features except one could be deduced from

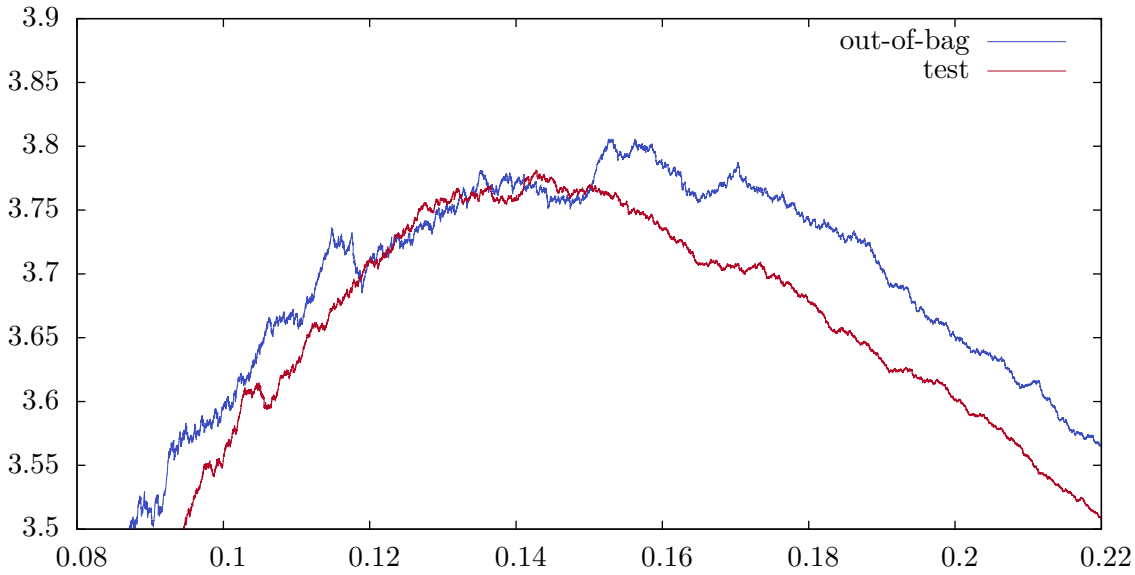

Figure 2: Out-of-bag estimation of the AMS vs cutoff curve of the winning solution compared to the AMS curve of the same ensemble on the test set.

the feature `pri-jet-num`. Not even this single feature, `der-mass-mmc`, profited from being accompanied by a missing value indicator.

3.2. Feature Generation

In addition to the primary and derived features provided by the contest organizers, a number of extra derived features were hand-crafted although the increase in score provided by them was low compared to the noise in the cross-validation score. First, four features based on the dropped azimuthal angles were added:

$$min(\phi_{tau} - \phi_{lep}, \phi_{tau} - \phi_{met}, \phi_{lep} - \phi_{met}), \quad \text{(radian 1)}$$

$$min(\phi_{tau} - \phi_{met}, \phi_{lep} - \phi_{met}), \quad \text{(radian 2)}$$

$$min(\phi_{tau} - \phi_{lep}, \phi_{tau} - \phi_{met}), \quad \text{(radian 3)}$$

$$min(\phi_{lep} - \phi_{met}). \quad \text{(radian 4)}$$

In the above equations the difference between two radian values is brought back to the $[-\pi, \pi[$ range. While these features, being composed of differences of angles, are rotation invariant, that is unlikely to be their only contribution, since simply normalizing all angles with respect to any one of the tau, the lepton or the missing transverse energy did not improve the results.

Five mass based features were added. The log normalized invariant mass (Adam-Bourdarios et al., 2014) of the tau and the leading jet:

$$\ln(1 + m_{inv}(tau, jet)), \quad \text{(mass 1)}$$

the log normalized invariant mass of the tau and the subleading jet:

$$\ln(1 + m_{inv}(tau, jet_2)), \quad \text{(mass 2)}$$

the log normalized invariant mass of the tau and the lepton:

$$\ln(1 + m_{inv}(tau, lep)), \quad \text{(mass 3)}$$

the log normalized transverse mass (Adam-Bourdarios et al., 2014) between the tau and the leading jet:

$$\ln(1 + m_{tr}(tau, jet)), \quad \text{(mass 4)}$$

and the log normalized transverse mass between the tau and the subleading jet:

$$\ln(1 + m_{tr}(tau, jet_2)). \quad \text{(mass 5)}$$

The last feature added was the modulus ($\ln(p_T \times \eta)$) of the pseudo particle whose momentum vector is the sum of the momentum vectors of the tau, the lepton and the jets. Like all other features the above were also normalized to zero mean and standard deviation one. In Figure 3, the background vs signal histograms paint a rough sketch of the information content of these extra features when viewed in isolation. Of course, a more interesting question is how much they help when added to the standard features. In Figure 4, we can see that dropping the azimuthal features basically had no effect on the score of the ensemble and adding the extra features did provide a small but consistent boost across a wide range of cutoffs.

Late in the contest, a new set of derived features was published by team C.A.K.E. (George et al., 2014) that provided a boost of about 0.01 to 0.02 for many contestants. Ultimately, these features were not used in the final solution due to concerns about reproducability of results which later turned out to be unfounded with the publication of the code.

4. Regularization

Dropout (Hinton et al., 2012; Srivastava et al., 2014) was applied to hidden layers. Dropping out inputs didn't work, because the inputs had very little redundancy. Thus, weights of connections between the input and the first hidden layer were given L_1 and L_2 penalties of 5E-6 and 5E-5, respectively. Furthermore, as a mild sparsity constraint each neuron in the first hidden layer was connected to only up to 10 randomly selected input units.

While adding L_1 and L_2 penalties was beneficial when training on only the 250,000 examples available during the contest, that may not be necessarily be the case for the larger opendata data set. In fact, one can reasonably expect to need less regularization. Figure 5 shows that simply removing L_1 and L_2 penalties in this case improves the AMS significantly.

5. Training

Weights were initialized to small random numbers from a normal distribution with standard deviation 0.01, then the model was trained by backpropagation (Rumelhart et al., 1988)

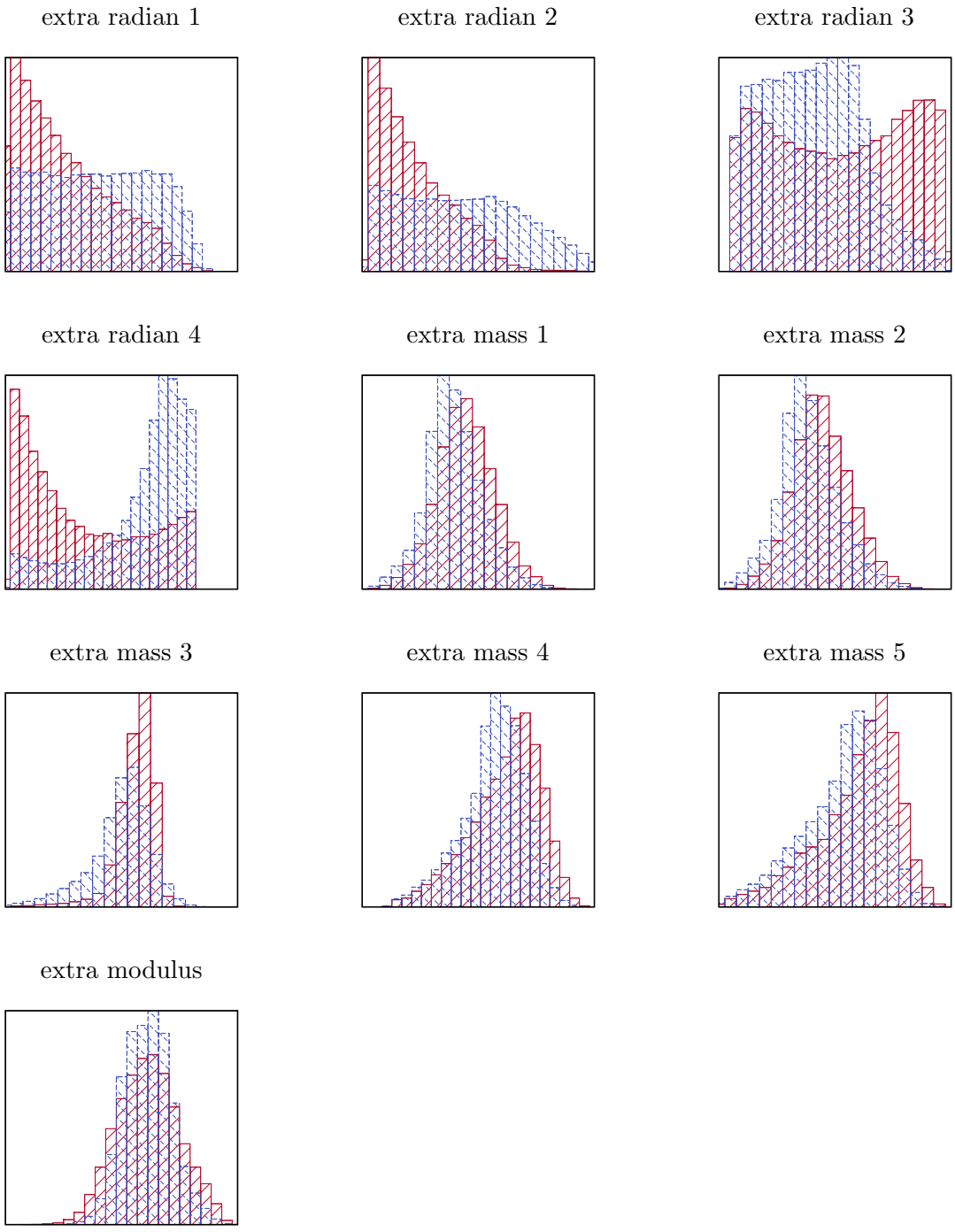

Figure 3: Histograms of weights of background and signal examples across the domains of the features.

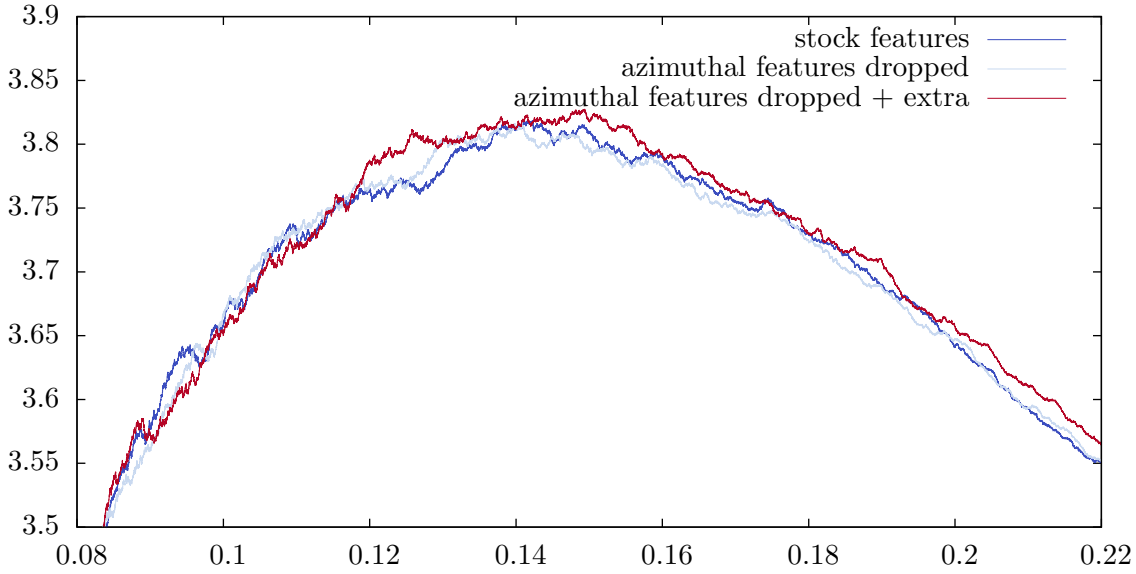

Figure 4: AMS vs cutoff curves of the same model trained with different features. Models were trained with CV bagging on the entire opendata data set.

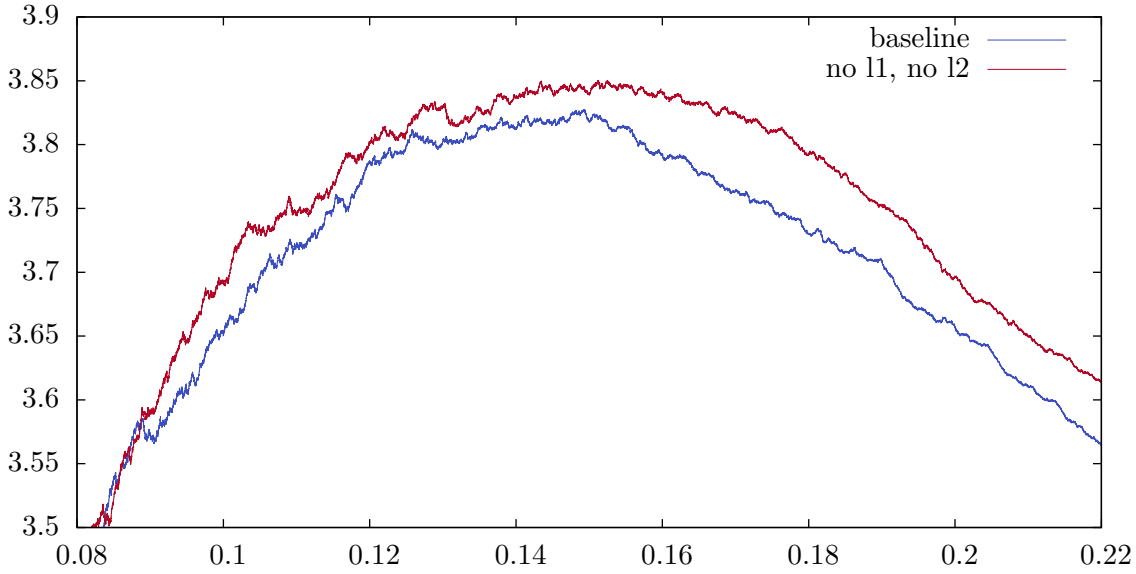

Figure 5: AMS vs cutoff curve of the winning solution and the same model without L_1 and L_2 regularization. Models were trained with CV bagging on the entire opendata data set.

with stochastic gradient descent on a weighted cross-entropy objective where the weight of each example was given in the data set. Models were trained for 200 epochs with batch size 96 and an initial learning rate of 1. The learning rate was decayed by a factor of 0.97 each epoch. Momentum was gradually increased from 0.5 to 0.99 over the course of the first 100 epochs following a linear schedule. Training one network on 250,000 examples took about 10 minutes on a GTX Titan GPU and 30 minutes on an Amazon EC2 g2.2xlarge instance.

During training the weights of the examples were adjusted so that the total weight of the signal and background examples were equal. This was by no means necessary, but it made optimization easier.

6. The Importance of the Public Leaderboard Scores

During the contest, participants trained models on the training set of 250,000 examples and submitted predictions for the unlabeled test set of 550,000 examples. Up to five submissions were allowed per day. For every submision made the only feedback was the public leaderboard AMS score calculated on an unknown subset of 100,000 examples. While local cross-validation was obviously very important, there was also information in the leaderboard scores and the question was just how much.

The variance of the bootstrap estimate of the AMS is inversely proportional to the number examples in the data set. A quick back-of-the-envelope calculation gives weight

$$\frac{\sqrt{100000}}{\sqrt{250000}+\sqrt{100000}} = 0.387$$

to the leaderboard score, but that would be a gross overestimation. Depending on just how smooth we believe the noise-free AMS curve to be, there is much more information available from the training set where we can see the *entire* AMS vs cutoff curve instead of just a single value at a particular cutoff. For this reason, the public leaderboard score was only taken into account as a sanity check with weight 0.1.

7. The Effect of Data Set Size

It was theorized both during and after the contest that model selection and prediction accuracy were both severely limited by the amount of data available. Figure 6 shows how the AMS vs cutoff curves evolve as the data set size increases. Note that all models use the same hyperparameters tuned for the training set available during the contest that contained 250,000 examples which roughly corresponds to the 30% curve in the figure.

To keep the comparison fair, the AMS curves were computed from the entire opendata data set. For clarity, 2-fold CV bagging was used with 10 repetitions which means that for an example in the training set of an ensembled model the prediction was the average of the 10 predictions made by the constituent models whose training sets did not include that example, while for an example in the test set it was the average of the predictions of all 20 constituent models. As shown in Figure 1, the number of repetitions is sufficiently high so that the discrepancy between the out-of-bag and test predictions in negligible.

The figure clearly shows quickly diminishing returns. Nevertheless, keeping in mind that using a model tuned for a smaller data set to avoid overfitting is likely to be suboptimal as illustrated by Figure 5, this is clearly not the final word on this topic.

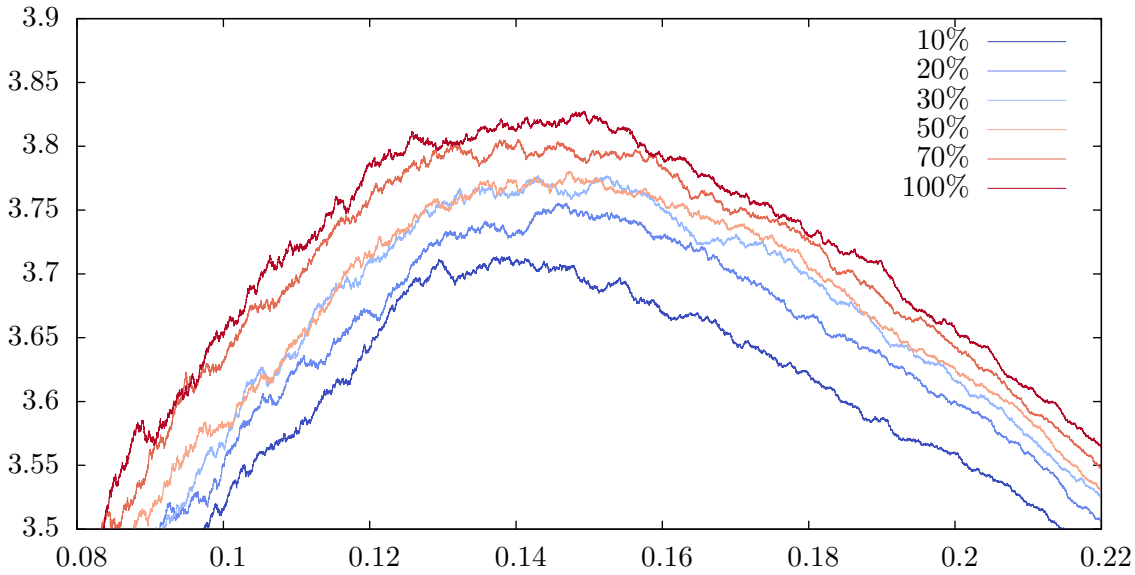

Figure 6: AMS vs cutoff for models trained on various percentages of the 818238 examples in the opendata data set. The cutoff is the proportion of examples labeled as positive.

Baldi et al. (2014) conduct experiments using a similar neural network for the Higgs to $\tau^+\tau^-$ problem with much more data (40 million vs 800 thousand). Crucially, their results cannot be compared directly to ours, because the simulators that produced these artificial data sets are different. Nonetheless, it is thought provoking that the gap between using all features and primary features only is quite a bit smaller in their case (0.21) than in ours (~ 0.5).

8. MNIST experiments

Since the invention of dropout, traditional regularization methods get less emphasis. Table 1 shows that for the permutation invariant MNIST task (LeCun and Cortes, 1998), adding a small L_1 penalty of 1e-6 to the weights of the connections between the input and the first hidden improves the 7-fold cross-validation accuracy by 0.06-0.08 for various networks. The same table also shows that ReLu and Maxout are neck and neck while local winner-take-all (LWTA) enjoys a margin of 0.1% over them.

Note that we report 7-fold stratified cross-validation results, because compared to the data set size the differences are rather small and overfitting the standard test set is a real possibility both here and in other published research. In particular, in our experiments the 7-fold CV results of ReLu (Glorot et al., 2011) and Maxout (Goodfellow et al., 2013) are identical.

Architecture	Activation	L_1 penalty	CV acc.
1200 x 1200 x 1200	ReLu	no	98.91%
1200 x 1200 x 1200	ReLu	yes	98.97%
1200 x 1200	Maxout	no	98.91%
1200 x 1200	Maxout	yes	98.97%
1200 x 1200 x 1200	LWTA	no	99.01%
1200 x 1200 x 1200	LWTA	yes	99.09%

Table 1: The effect of activation function choice and weight penalty in the permutation invariant MNIST task.

9. Conclusion

We have shown a simple but effective solution to the Higgs Machine Learning contest that benefits from, but does not rely on extensive feature engineering. The benefits of some choices, such as the LWTA activation and the L_1 penalty, carry over to permutation invariant MNIST. Finally, CV bagging was described as a practical middle-ground between bagging and repeated cross-validation to both estimate and improve prediction accuracy.

Although the evidence for the effect is convincing, the degree to which more data is beneficial remains an open question, not the least due to the models being tested having been optimized for smaller data sets. High quality simulated data is very expensive to compute (about half an hour for a single example), but it would not be surprising to find that compared to the enormous cost of running the collider it is relatively cheap. Considering that armed with better predictors less data is needed, it is an obvious direction to explore.

References

Claire Adam-Bourdarios, Glen Cowan, Cecile Germain, Isabelle Guyon, Balázs Kégl, and David Rousseau. Learning to discover: the Higgs boson machine learning challenge. 2014. doi: 10.7483/OPENDATA.ATLAS.MQ5J.GHXA. URL http://doi.org/10.7483/OPENDATA.ATLAS.MQ5J.GHXA.

Esteban Alfaro, Matıas Gámez, and Noelia Garcia. Adabag: An R package for classification with boosting and bagging. *Journal of Statistical Software*, 54(2):1–35, 2013.

ATLAS collaboration. Dataset from the atlas higgs boson machine learning challenge 2014, 2014. URL http://doi.org/10.7483/OPENDATA.ATLAS.ZBP2.M5T8.

Pierre Baldi, Peter Sadowski, and Daniel Whiteson. Enhanced higgs to $\tau^+\tau^-$ searches with deep learning. 2014. doi: 10.1103/PhysRevLett.114.111801.

Peter Bühlmann. Bagging, subagging and bragging for improving some prediction algorithms. 2003.

Damien George, Thomas Gillam, and Christopher Lester. C.A.K.E. features. https://bitbucket.org/tpgillam/lesterhome, 2014.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier networks. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics. JMLR W&CP Volume*, volume 15, pages 315–323, 2011.

Ian J Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. *arXiv preprint arXiv:1302.4389*, 2013.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

Yann LeCun and Corinna Cortes. The MNIST database of handwritten digits, 1998.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 5, 1988.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Rupesh K Srivastava, Jonathan Masci, Sohrob Kazerounian, Faustino Gomez, and Jürgen Schmidhuber. Compete to compute. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2310–2318. Curran Associates, Inc., 2013. URL http://papers.nips.cc/paper/5059-compete-to-compute.pdf.

Qi Wang and Joseph JáJá. From maxout to channel-out: Encoding information on sparse pathways. *CoRR*, abs/1312.1909, 2013. URL http://arxiv.org/abs/1312.1909.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320, 2005.