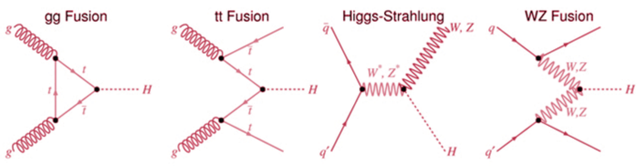
Classification of Higgs Boson Tau-Tau decays using GPU accelerated Neural Networks

Mohit Shridhar (mohits@stanford.edu, mohit@u.nus.edu)

Abstract

In particle physics, Higgs Boson to tau-tau decay signals are notoriously difficult to identify due to the presence of severe background noise generated by other decaying particles. Our approach uses a bag of dropout neural networks to classify events as signals or background noise.

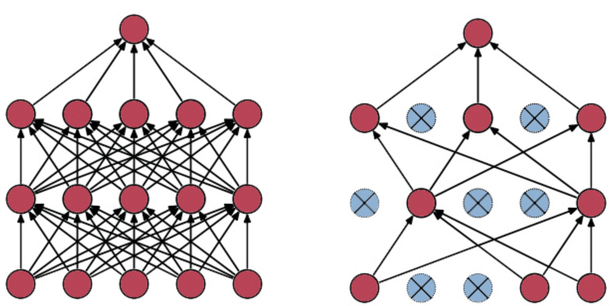


Optimization

The accuracy of the model was determined by the Approximate Median Significance score. Here *s* and *b* are the expected number of signal and background events respectively.

$$\mathrm{AMS}_c = \sqrt{2\left((s+b+b_{\mathrm{reg}})\ln\left(1+\frac{s}{b+b_{\mathrm{reg}}}\right)-s\right)},$$

Dropout Neural Network



Standard Neural Net

Dropout Neural Net

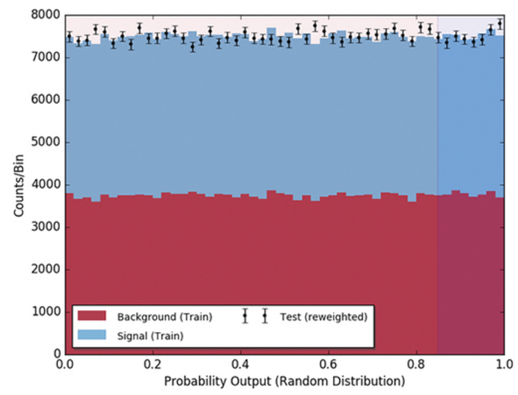
$$\begin{aligned} z_i^{(l+1)} &= \mathbf{w}_i^{(l+1)}\mathbf{y}^l + b_i^{(l+1)}, \\ y_i^{(l+1)} &= f(z_i^{(l+1)}), \end{aligned}$$

Feed Forward Mechanism

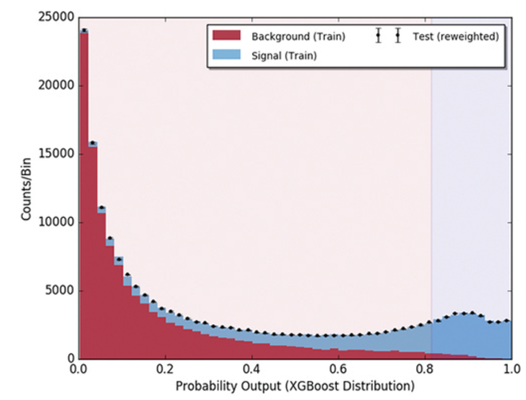
$$\begin{aligned} r_j^{(l)} &\sim \mathrm{Bernoulli}(p), \\ \widetilde{\mathbf{y}}^{(l)} &= \mathbf{r}^{(l)} * \mathbf{y}^{(l)}, \\ z_i^{(l+1)} &= \mathbf{w}_i^{(l+1)}\widetilde{\mathbf{y}}^l + b_i^{(l+1)}, \\ y_i^{(l+1)} &= f(z_i^{(l+1)}). \end{aligned}$$

Comparing Learning Models

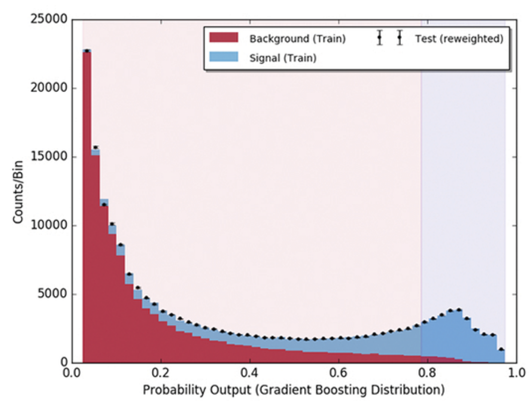
Random Distribution



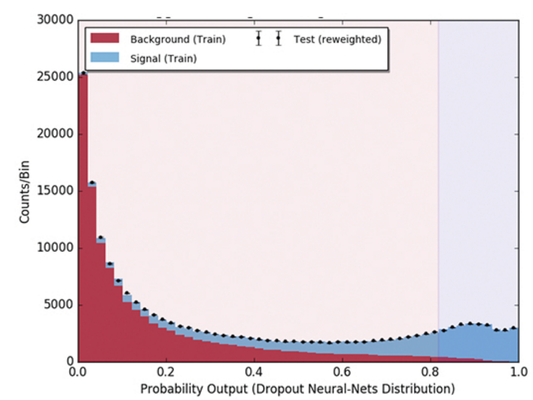
XGBoost



Gradient Boosting



Dropout Neural Network



The training dataset was composed of 30-feature input vectors with binary labels classifying the sample as signal or background event. 4 features were removed through manual-selection to reduce overfitting. The input vector was normalized to have a mean: 0 & stddev: 1. The final learning model was an ensemble of 15 dropout neural networks with 3 hidden layers consisting of 450 neurons each: **26x450x450x450x2**. The outputs were two softmax neurons, which individually predicted the probability of a sample being a signal or an event. An input was classified as a signal if the output of the softmax signal neuron was greater than 0.5575. Results from each network (from the bag) were simply amalgamated by averaging the output probabilities. The model was trained with 2-fold stratified cross-validation with random shuffling to prevent overfitting. Additionally, the cost function was set to reduce the mean of the cross-entropy loss. Each neuron in the hidden layer had a 0.5 probability of being dropped.

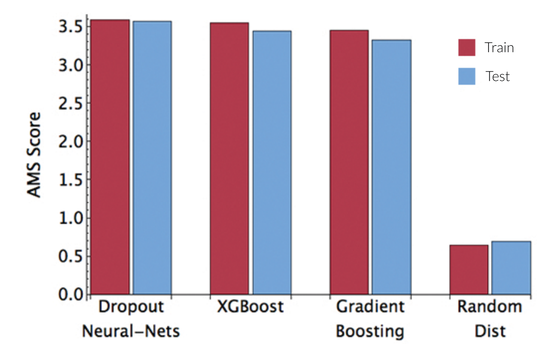
Results

Preliminary results indicate that dropout neural nets perform marginally better than XGBoost and Gradient Boosting in terms of accuracy. However the neural nets took nearly 240x longer to train than XGBoost. The AMS scores roughly correspond to LB scores of 90% accuracy. Testing the neural net on a different dataset showed that it was still suffering from over-fitting issues.

The network was implemented using Google's TensorFlow library on an Amazon EC2 GPU instance. The system was trained with 4 GPUs composed of 1,536 CUDA cores each and 4GB of memory.

Source Code & Project:
github.com/MohitShridhar/higgsml

Benchmarking



Overfitting