CS-433: Project 1
Higgs boson machine learning challenge

Daniel Tavares Agostinho, Thomas Castiglione, Jeremy Di Dio

***Abstract*—In this project, we present several machine learning models that aim to solve the Higgs boson machine learning challenge by correctly predicting whether an event corresponds to the decay of a Higgs boson or not. Our best model is a neural network we built from scratch that achieves an accuracy of 0.839 on the testing set.**

I. Introduction

Discovered experimentally in 2012 at CERN near Geneva, the Higgs boson is an elementary particle of the Standard Model of particle physics produced by the quantum excitation of the Higgs field.

In 2014, The Higgs Boson Machine Learning Challenge [1] was organized. This challenge aimed to use the power of the "crowd" to optimize the analysis of the Higgs boson. In this project, we will present our methodology to answer this challenge which, in other words, consists in building a binary classifier to predict whether an event corresponds to the decay of a Higgs boson or not.

II. Methodology

A. Data processing

First and foremost, we analyzed and processed the provided data to make them the most suitable for our chosen models.

1) Dataset splitting: After analyzing the distribution of each feature, we observed that, depending on the value of one particular feature, named PRI_jet_num, other features would take undefined values. This particular feature takes value in $\{0, 1, 2, 3\}$.

As a result of this observation, we chose to separate our datasets based on the value of this feature, i.e., from a single dataset, we obtained four separate datasets, one for each possible value of jet number. Then, for each new dataset, we eliminated all undefined features. Table I shows the number of significant features remaining, out of the 29 primary features, for each dataset.

It is important to note that we chose to combine dataset 2 and 3 as they had the exact same significant features.

2) Remaining undefined values: To deal with the remaining undefined values which appear punctually across the datasets, we simply chose to replace them by the mean of the corresponding feature.

Number of Jet	Number of features
0	18
1	22
2	29
3	29

Table I
Number of significant features per jet value

3) Data transformation: As a mean to reduce the weight of outliers, we chose to apply an $arcsinh$ transformation [2] on each dataset. This function has the convenience to be definite for 0 and negative values (which were present across the datasets) and is almost parallel to the log function for values greater than 2.

Thus, for each sample x we replaced it by x' as follow:

$$x' = arcsinh(x) \tag{1}$$

4) Standardization: Finally, we standardized our datasets. This process helped to makes all variables contribute equally to the computation of the weights. We replaced each sample x' by x'' as follows:

$$x'' = \frac{x' - mean(x')}{std(x')} \tag{2}$$

B. Models and Methods

We were asked to implement six different models to perform this binary classification :

A Linear regression using gradient descent
B Linear regression using stochastic gradient descent
C Least squares regression using normal equations
D Ridge regression using normal equations
E Logistic regression using gradient descent
F Regularized logistic regression using gradient descent

In those models two loss functions were used :

1) The Mean Square Error (MSE), used and minimized by Linear regression models. Also used by the least squares and Ridge regressions that minimize it with normal equations.
2) The logistic loss, used and minimized by Logistic regression models.

C. Hyperparameter tuning

To find the best performing model, we did a grid search over the hyperparameters. We first splitted the dataset into training/validation sets and we computed the accuracy of the predictions over the validation set. We used even numbers in the range $[2, 12]$ for the degree of the polynomial expansion. $\gamma = \{10^{-3}, 10^{-2}, 10^{-1}, 2*10^{-1}, 3*10^{-1}\}$ for the step size of the gradient descent models. $\lambda = \{10^{-3}, 10^{-2}, 10^{-1}, 2*10^{-1}, 3*10^{-1}\}$ for the regularization term in the ridge regression. $\lambda = \{10^{-3}, 10^{-1}, 3*10^{-1}\}$ for the regularization term in the regularized logistic regression. For gradient descent models, the weights are initialized with uniform distribution over $[0, 1[$.

D. Neural Network

To improve the results obtained with linear methods, we implemented a feed-forward Neural Network from scratch. Due to the varying number of features in our datasets, we trained 3 different models between which only the input layer sizes differed. The architectures of our final models are shown in Table II. We used an exponential learning rate decay [3] between 0.5 and 1e−5, polynomial expansion of degree 2, and we initialize the weights by sampling them from a uniform distribution over $[-0.1, 0.1]$.

layer	**dimension**	**activation function**
input	36, 44, 58	ReLU
1	64	ReLU
2	32	ReLU
3	16	ReLU
output	2	Softmax

Table II
NEURAL NETWORK ARCHITECTURE. ONLY THE INPUT LAYER SIZE DIFFERS BETWEEN THE 3 MODELS.

To train the Neural Network, we used the Binary Cross Entropy loss:

$$L = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(\hat{y}_i) \tag{3}$$

where y_i and $\hat{y}_i$ are the true and predicted classes respectively.

III. RESULTS

The best performances we reached with linear and non-linear models are shown in Table III.

Model	**Average accuracy**[i]	**Hyperparameters (per dataset split)**
E	0.763	Expansion degrees : 4, 12, 10
Neural Networks	0.835	Epochs: 3000, 1500, 1500

[i] Computed as a sample count weighted average across our validation datasets.

Table III
BEST MODELS

The Neural Networks training losses and accuracies are shown in Figure 1

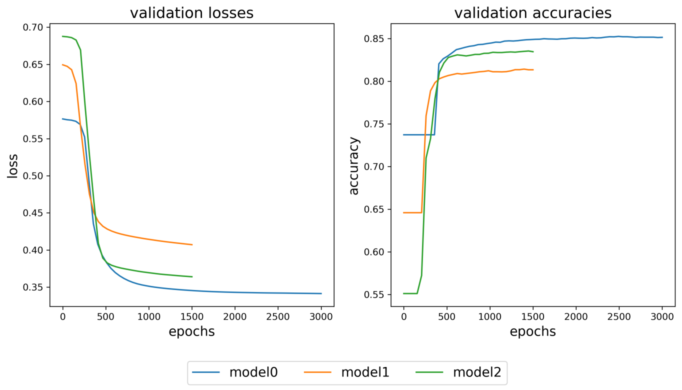

Figure 1. Training and validation losses and accuracies for all 3 Neural Network models. the model index correspond to each dataset ($model0$ to $jet = 0$, $model1$ to $jet = 1$, and $model2$ to $jet = 2, 3$).

From Figure 1 we make two main observations: First, all models face a plateau at the very beginning of each training. This most likely comes from the weights initializations as these plateaus correspond to the labels ratios in each dataset, meaning that the decisions boundaries take time to move into the data point-clouds. Second, the $PRI\_jet\_num = 1$ dataset is harder to classify than the two others. We observe the same behavior with linear models.

For the final test set predictions, we trained our 3 models on the whole training set and submitted our predictions on AIcrowd. Our best performance is an accuracy of 0.839 and an F1-score of 0.756. It is fully reproducible with our *run.py* script.

Model	A	B	C	D	E	F	NN
Accuracy	0.722	0.643	0.737	0.727	0.763	0.754	0.839

Table IV

IV. CONCLUSION

We showed in the project that, a careful analysis of the data can lead to fundamental changes in the classical Machine Learning approach and therefore performances, and that Neural Networks can perform better than linear methods on this dataset. To go further, we would have liked to be able to scale our Neural Networks or try other non-linear methods such as lightGBM [4].

References

[1] C. Adam-Bourdarios, G. Cowan, C. Germain, I. Guyon, B. Kegl, and D. Rousseau, “Learning to discover: the higgs boson machine learning challenge,” 05 2014.

[2] E. C. Norton, “The inverse hyperbolic sine transformation and retransformed marginal effects,” National Bureau of Economic Research, Working Paper 29998, April 2022. [Online]. Available: http://www.nber.org/papers/w29998

[3] Z. Li and S. Arora, “An exponential learning rate schedule for deep learning,” *CoRR*, vol. abs/1910.07454, 2019. [Online]. Available: http://arxiv.org/abs/1910.07454

[4] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, “Lightgbm: A highly efficient gradient boosting decision tree,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf