FYS3150 Machine Learning - Project 3
Classifying Pulsar Candidates With Machine Learning Methods

Nils Johannes Mikkelsen
(Dated: December 16, 2018)

All material for project 3 may be found at:
https://github.com/njmikkelsen/machinelearning2018/tree/master/Project3

I. INTRODUCTION

II. METHOD

The data set used in this project can be downloaded from the following Kaggle post:

https://www.kaggle.com/pavanraj159/predicting-a-pulsar-star

The data set contains just under 18,000 samples of pulsar condidates from the High Time Resolution Universe Survey, all post-verified as either a known pulsar or a known non-pulsar. The data set originates from the 2012 article [4] by S.D. Bathes et. al. in which an MLP classifier was successfully used to *"blindly detect 85% of pulsars"*.

A. Preparation

The data set contains of 8 real-valued predictors and 1 target label. The predictors include the mean value, standard deviation, skewness and excess kurtosis of the pulsar candidates' IPP and DM-SNR curve. The target label is a binary label ± 1 that identifies the candidate as a pulsar (+1) or a non-pulsar (-1). A histogram for each predictor has been included in appendix **??**, the pulsars and non-pulsars have been separated to highlight the difference.

Despite some minor overlap in the predictor space, there are clear differences between the pulsars and non-pulsars, and consequently an appropriate neural network or SVM should not have much trouble separating the two. With this in mind, this project deals not with *whether* the two classes are separable, in fact Bathes et. al. proved as much in their original article, but instead to what extent are neural networks and SVMs applicable.

The machine learning in this project is implemented using `scikit-learn` [5], in particular the classes `neural_network.MLPClassifier` and `svm.SVC` and the `train_test_split` functionality. The function randomly divides a data set into a training set and test set.

The neural networks and SVMs are compared using a so-called accuracy score. This score is a measure of the average correct identification of a pulsar or non-pulsar:

$$\text{accuracy} = \frac{1}{N}\sum_{i=1}^{N} I_i, \quad I_i = \begin{cases} 1, & \text{correct label} \\ 0, & \text{incorrect label} \end{cases} \tag{1}$$

where i traverses the test set. The accuracy score is number between 0 and 1.

B. Hyperparameter Analysis: Neural Networks

The MLP network is very robust, converging on a minimum for almost all reasonable choices of hyperparameters. Ideally one would be able to vary all hyperparameters simultaneously, but this is simply not feasible on a standard computer, and in fact, probably not necessary. Instead certain "types" of hyperparameters, e.g. "learning parameters" and "structure parameters", will be studied separately.

There are countless possible variations of even the simplest of networks. Therefore, the following restrictions are introduced in order to reduce the number of combinations of paramaters:

1. All layers in a network have the same number of nodes.
2. All layers in a network use the same activation function.
3. With the exception of section II B 3, the pulsar data set is divided 50-50 into training and test sets.

1. *Network complexity*

The first stage of the analysis takes a look at whether simple or complex networks work best with the pulsar data set. Recall that L is the number of layers and N is the number of nodes per layer. To span a large range of complexity levels, 900 networks structures are considered: The networks use 30 linearly spaced L values between 1 and 50, and 30 linearly spaced N values between 1 and 100.[1] Each network is trained using a logistic, hyperbolic tangent and rectifier activation. Hence, a total of $3 \times 30 \times 30 = 2700$ unique networks are trained.

Having trained a network, the performance of the network is measured according to three metrics: its accuracy score, the number of required training epochs and the time spent training. The accuracy score is interesting for obvious reasons, the two latter metrics are interesting because they reveal the network's training efficiency.

The network designs used in the following sections will be largely based on the results of this complexity analysis.

2. *Network learning scheme*

Having looked at the network designs, this section will study the network's learning phase. The hyperparameters of interest are the initial learning rate γ_0 and the strength of the learning momentum η. Note that equation (**??**) allows for a varying momentum strength, but this will not be explored here. This analysis will look at 17×15 networks using 17 logarithmically spaced γ_0 values between 10^{-4} and 10^{0}, and 15 linearly spaced η values between 0.01 and 0.99. Each network is trained using a logistic, hyperbolic tangent and rectifier activation, making the total number of 765 unique learning schemes per network structure. The number of network

[1] The "linearly spaced" L and N values are created using `np.linspace(...).astype(int)`. The type switch from `float` to `int` may slightly interfer with the linear spacing.

structures considered is chosen based on the results of the complexity analysis.

The network is measured using the same metrics as in the complexity analysis. The results of this analysis will also be used in the following sections.

3. Overfitting the training data

Gridsearch of train-to-test split ratio.

4. •

C. Hyperparameter Analysis:

Support Vector Machines

1. Violation

2. Linear kernel

3. Polynomial kernel

4. Radial Gaussian kernel

5. Hyperbolic tangent kernel

III. RESULTS

A. Neural Networks

The results from the complexity analysis are shown in figure 1.

The results from the learning scheme analysis are shown in figure 2.

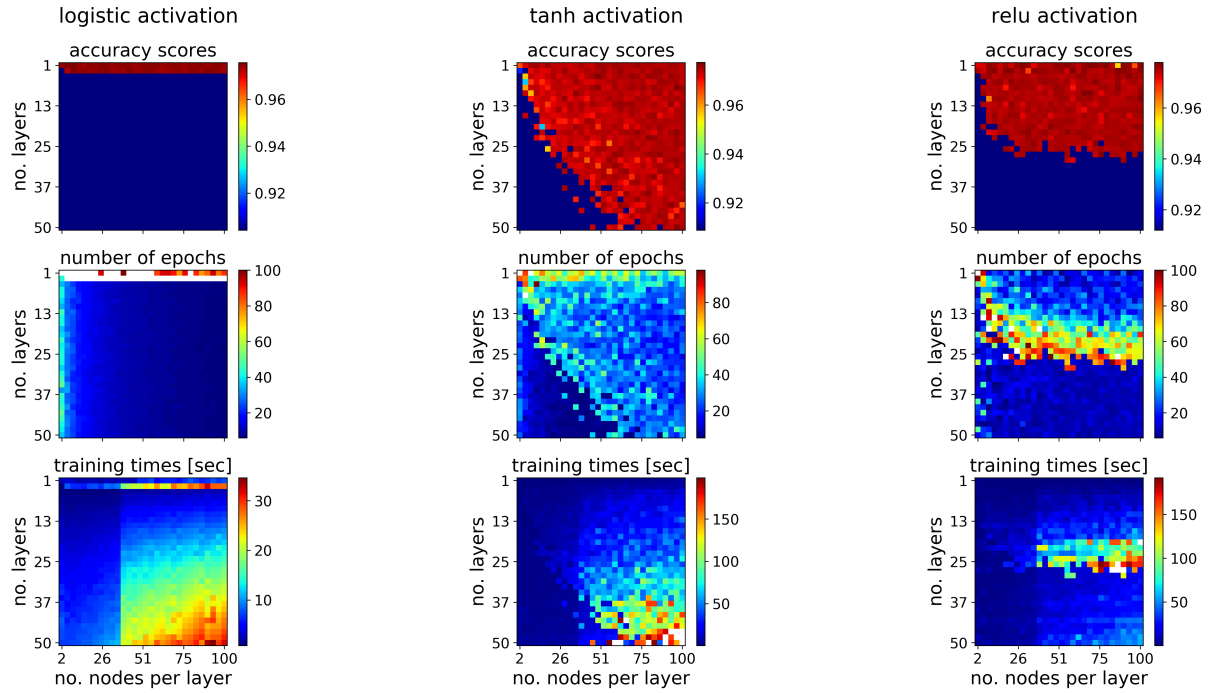

Figure 1: Hyperparameter analysis of the complexity of neural networks. In general, network complexity increases along the diagonal from the upper left to the lower right. An upper threshold of 150 epochs and 200 seconds are used to focus the colormap, white pixels have passed the threshold.

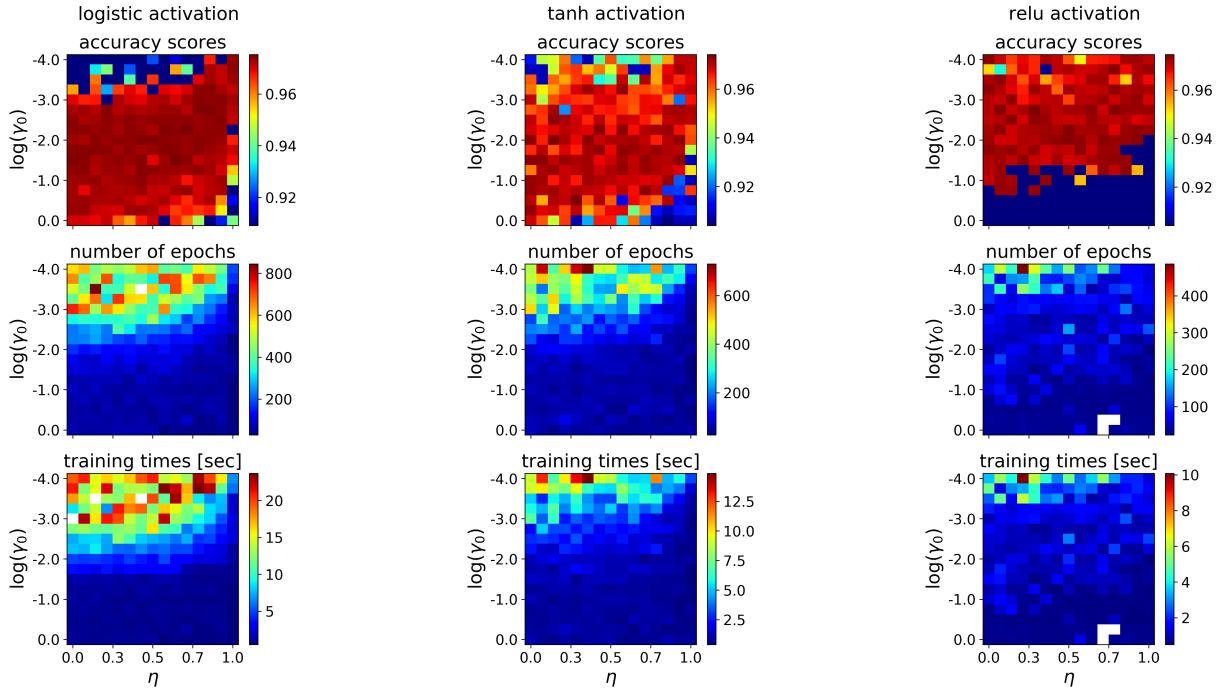

Figure 2: Hyperparameter analysis of the learning scheme of neural networks with 8 nodes in a single layer. As a general rule, the learning rate increases along the diagonal from the upper left to the lower right. An upper threshold of 1000 epochs and 25 seconds are used to focus the colormap, white pixels have passed the threshold.

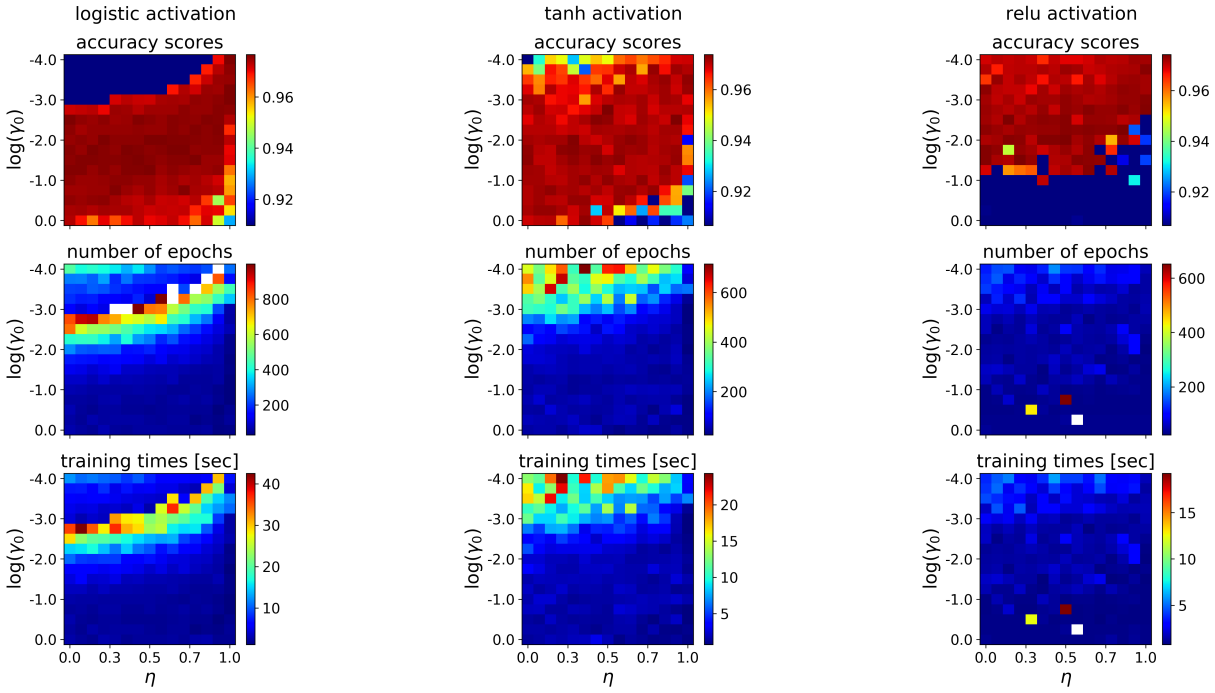

Figure 3: Hyperparameter analysis of the learning scheme of neural networks with 15 nodes in each of 2 layers. As a general rule, the learning rate increases along the diagonal from the upper left to the lower right. An upper threshold of 1000 epochs and 50 seconds are used to focus the colormap, white pixels have passed the threshold.

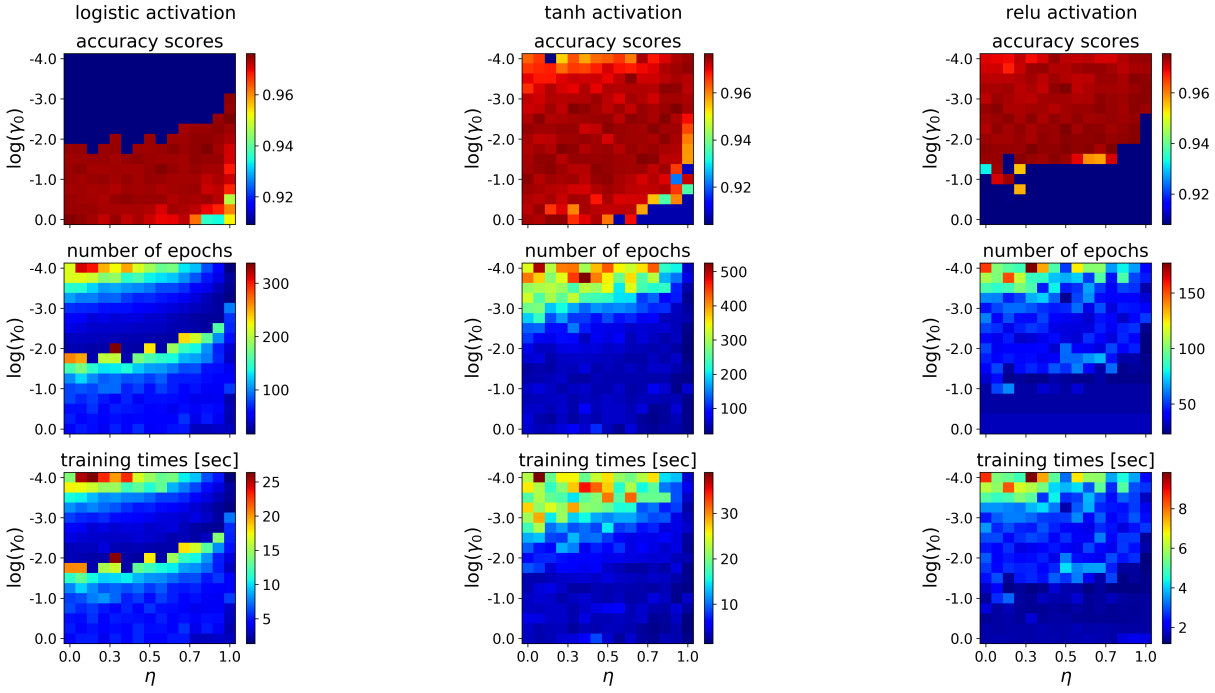

Figure 4: Hyperparameter analysis of the learning scheme of neural networks with 30 nodes in each of 3 layers. As a general rule, the learning rate increases along the diagonal from the upper left to the lower right.

[1] F.Graham Smith, F.R.S., former Director of the Royal Greenwich Observatory (1976-1981). *Pulsars*. Cambridge University Press, 1977.

[2] Jim Condon. Essential radio astronomy, chapter 6: Pulsars. Online resource hosted by National Radio Astronomy Observatory (US), Last modifed 2016. https://www.cv.nrao.edu/~sransom/web/Ch6.html.

[3] Integrated pulse profile of a pulsar. Original work by Lorimer Kramer (Handbook of Pulsar Astronomy), republished by AstroBaki, Last Modified December 6th 2017. https://casper.berkeley.edu/astrobaki/index.php/Dispersion_measure.

[4] S.D. Bathes et. al. The high time resolution universe survey vi: An artificial neural network and timing of 75 pulsars. arXiv:1209.0793v2 [astro-ph.SR], 2012.

[5] sckit-learn: Machine learning in python. Open Source. https://scikit-learn.org/stable/.