Feature preprocessing in HEP at the example of a SUSY classification problem

Christoph Smaczny

Supervisors: Adam Elwood, Dirk Krücker

September 5, 2018

Abstract

In this work, it is studied how the performance of a neural network on a SUSY classification problem depends on feature scaling, especially whether results can be improved by scaling some features by the same factor, and preprocessing of angles given to the network. Moreover, it is investigated, if a fixed size network is able to effectively gain information from additional features which are only available for part of the data set.

Contents

1 Introduction **2**
- 1.1 Supersymmetry . . . 2
- 1.2 Neural Networks . . . 2
- 1.3 General Setup . . . 3

2 Dependence on the Standardization **4**
- 2.1 Motivation . . . 4
- 2.2 Results . . . 4

3 ϕ Angles relative to ϕ_{MET} **6**

4 Changing the number of Jets **7**

5 Conclusions **9**

1 Introduction

1.1 Supersymmetry

The Standard Model (SM) of particle physics allows to make very accurate predictions of how the elementary partcles, which our world is made of, interact. Nevertheless, there are phenomena, for which no explanation could be found within the SM, for example the hierarchy problem. Explanations for some of these phenomena are given by an extension of the SM called supersymmetry (SUSY) [1]. In SUSY, every SM particle gets a partner, which is a boson if the SM particle is a fermion and vice versa. The superpartners of fermions get an "s" in front of their name, for example the stop is the superpartner of the top. The superpartners have different masses than the SM particles. If they were comparably light, they would already have been found experimentally. The lightest supersymmetric particles are the stop and the neutralino, which is also called Lightest Supersymmetric Particle (LSP).

The supersymmetric process considered in this work is the decay of a stop and the decay of an antistop produced as a pair in a proton-proton collision into an (anti-)top and the LSP (figure 1). As background, only top-antitop production from gluons is considered (figure 2).

For the signal two scenarios for the masses of the relevant supersymmetric particles are looked at:

- Compressed: $m_{stop} = 600$ GeV, $m_{LSP} = 400$ GeV,
- Uncompressed: $m_{stop} = 900$ GeV, $m_{LSP} = 100$ GeV.

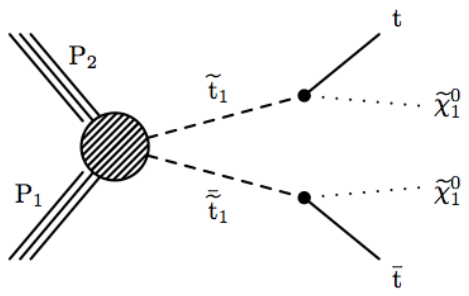

Figure 1: Signal

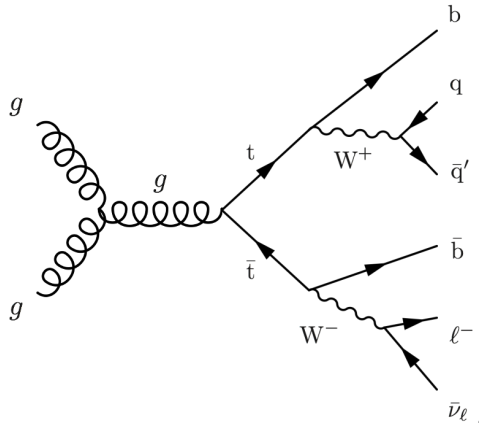

Figure 2: Background

1.2 Neural Networks

(Artificial) Neural Networks are a class of machine learning techniques inspired by the way neurons in the brain interact to solve difficult problems. In the following, only Feed Forward Neural Networks will be used. They are made of a fixed number of layers (an input layer, a number of so called hidden layers and an output layer) with a certain number of neurons

each. Every neuron in a hidden layer and the output layer is connected to every neuron in the previous layer.

The output of the neural network is given as the activation of the neurons in the output layer, which is just one neuron in the case of a classification problem. The activation of a neuron is computed by calculating the weighted sum of activations of the neurons in the previous layer and applying a nonlinear activation function on the result. Good weights are learned by iteratively modifying them to minimize the difference between the correct output and the output given by the network. This can be done efficiently by backpropagating the error through the network and modify the weights based on how much they contributed to the error [2].

Another type of Neural Networks are Recurrent Neural Networks, which posses an internal state and allow to get an output which depends on several consecutive inputs to the network. They are for example used to process time series data and data with a variable number of features.

Neural networks can suffer from overfitting, which means that the performance on the training set improves while the performance on the test set gets worse, so they start to learn specifities of the training set which don't generalize to the whole data set. This can be reduced by using regularization techniques. Two common ones are L2 regularization and Dropout. In L2 regularization, a term is added to the loss function penalizing large weights. When Dropout is used, a random fraction of the neurons is ignored in each step.

1.3 General Setup

Most of the time, a network with two hidden layers of 20 neurons each (notation: [20, 20]) is used. A set of 100 000 events is used of which 70 percent are used for training and 30 percent as a test set.

If not stated otherwise, the features given to the network are:

- H_T: The scalar sum of the transverse momenta.
- MET: The transverse missing energy.
- m_T and m_{T2}^W: Transverse masses.
- n_{jet} and n_{bjet}: The number of jets and of b-tagged jets.
- 1 Lepton (p_t, η, ϕ)
- 3 Jets (p_t, η, ϕ, m)

Training of the network is stopped after at most 100 steps for each of which 128 events are used. To make training faster, early stopping is used. After two consecutive steps without an improvement on the test set, training is stopped.

The network is trained using binary cross entropy as a loss function and for early stopping, but in the plots the Asimov estimate of the statistical significance will be shown [3].

For all plots shown in this work, the network has been run many times for different random initial weights of the neural network and a different subset of size 100 000 of a larger

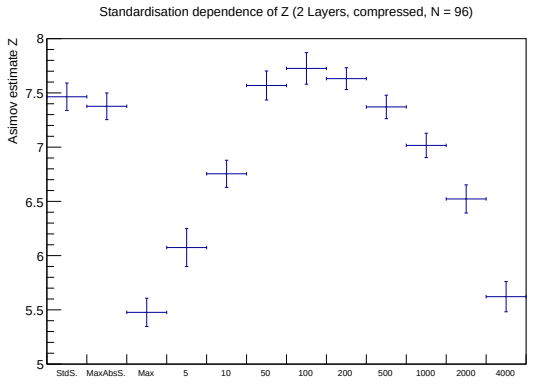

Figure 3: Dependence of the Asimov estimate of the significance on the standardization for common scalers between 5 and 4000.

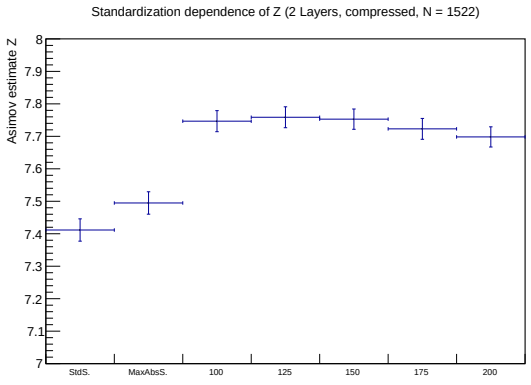

Figure 4: Dependence of the Asimov estimate of the significance on the standardization for common scaler between 100 and 200.

set containing 2 000 000 events. The number of runs for a point is given as N in the plots. The error bars give the statistical error of the mean.

2 Dependence on the Standardization

2.1 Motivation

Features can have very different ranges of values and different units. To make training easier, they are typically scaled before given to the neural network. Two common ways to do this are:

- StandardScaler: Subtracts the mean over the data set from each feature and scales them to have unit variance.
- MaxAbsScaler: Divides each feature by its absolute maximum in the data set, such that they lie between -1 and 1.

Both methods handle each feature individually. In the following, it shall be investigated, whether it is better to scale values with same units by the same factor, as their relative size can be meaningful.

All features with energy units (H_T, MET, m_T, m_{T2}^W and p_t and m for the lepton and the jets) will be scaled by the same factor.

2.2 Results

It can be seen in figure 3, that there is an optimal common scaler between 100 and 200. Figure 4 shows, that the optimal common scaler performs better than MaxAbsScaler and StandardScaler. Networks with three layers give a similar result (figure 5).

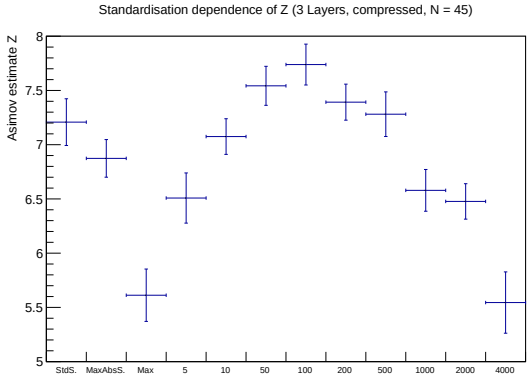

Figure 5: Dependence of the Asimov estimate of the significance on the standardization for common scalers between 5 and 4000 for a three layer network.

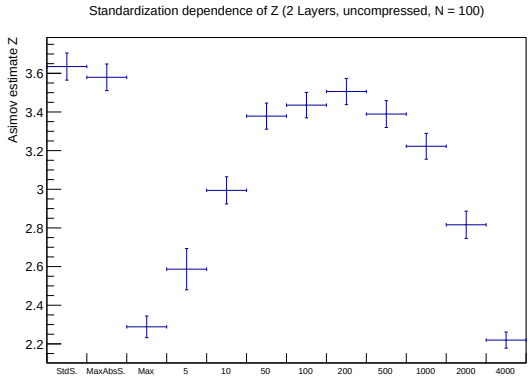

Figure 6: Dependence of the Asimov estimate of the significance on the standardization for common scalers between 5 and 4000 for the uncompressed case.

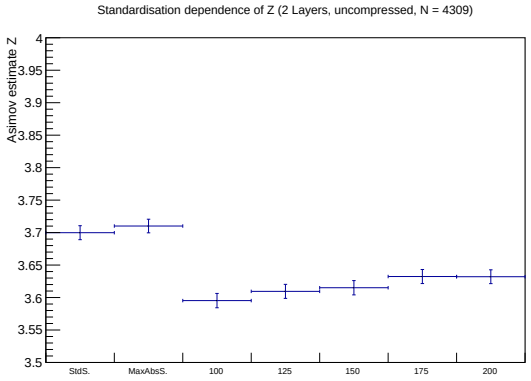

Figure 7: Dependence of the Asimov estimate of the significance on the standardization for common scaler between 100 and 200 for the uncompressed case.

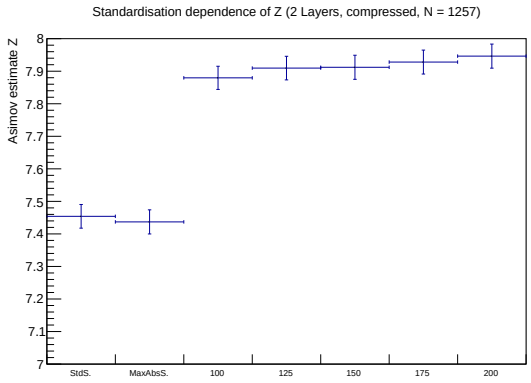

Figure 8: Dependence of the Asimov estimate of the significance on the standardization for common scalers between 100 and 200 for the compressed case without early stopping.

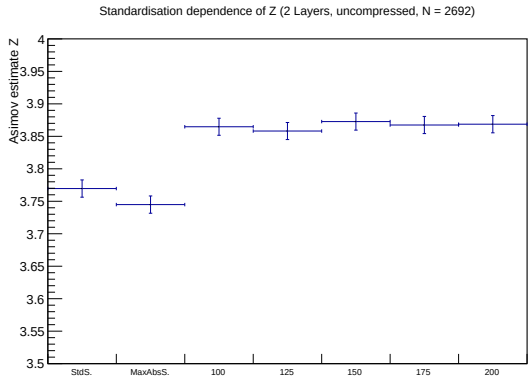

Figure 9: Dependence of the Asimov estimate of the significance on the standardization for common scaler between 100 and 200 for the uncompressed case without early stopping.

For the uncompressed case, the general shape of the curve is similar as well (figure 6), but the StandardScaler and the MaxAbsScaler lead to better results than scaling by any common factor (figure 7).

Now, early stopping is removed, so the network always trains for 100 steps and the best network obtained at some point during training is used in the end.

In the compressed case, this leads to even better results for the common scaling factors while the results for the StandardScaler and the MaxAbsScaler stay about the same (figure 8). In the uncompressed case, the StandardScaler and MaxAbsScaler are slightly better than before, but the common scaling factors are now better than both (figure 9).

When using early stopping after 5 steps without an improvement, results lie in between (figure 10) and for 10 steps they are already close to the results without early stopping (figure 11).

So scaling by the same factor is in general better, but too early stopping is a bigger problem when scaling by a common factor and can lead to worse results than when scaling feature wise.

3 ϕ Angles relative to ϕ_{MET}

In the first part of this work, ϕs are given to the network relative to the detector, but for a perfectly symmetric detector, only the relative angles should matter. It is now tried to give ϕ relative to ϕ_{MET} instead. To make the differences more visible, m_T, m_{T2}^W are removed from the features as they depend on the angles. Note that a lower systematic error is assumed, so plots from now on should not be compared to plots in the first part.

The results are shown in figure 12. The worst results are obtained, when the angles are given relative to the detector and ϕ_{MET} is not given to the network (red points), so when the network has neither explicit nor implicit information about the angle between the ϕs and

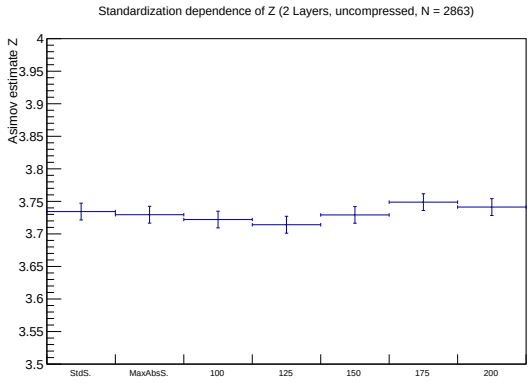

Figure 10: Dependence of the Asimov estimate of the significance on the standardization for common scalers between 100 and 200 for the uncompressed case with early stopping after 5 steps.

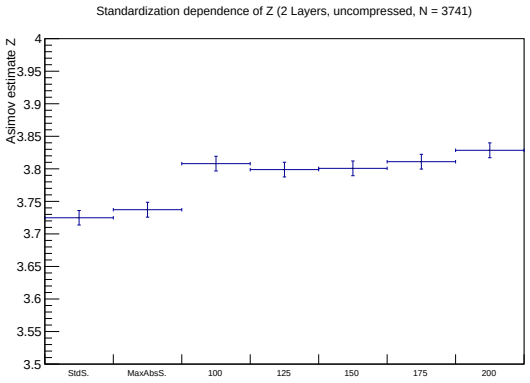

Figure 11: Dependence of the Asimov estimate of the significance on the standardization for common scaler between 100 and 200 for the uncompressed case with early stopping after 10 steps.

ϕ_{MET}.

The results get better, when ϕ_{MET} is given to the network (green points), so the network can compute the relative angles by itself. The results get even better, when the angles are given relative to ϕ_{MET} and then it makes no difference whether ϕ_{MET} is given to the network (blue points) or not (yellow points).

The best results are obtained, when the absolute values of the angles between the ϕs and ϕ_{MET} are given (black line).

Figure 13 shows the results for a very simple neural network having just one hidden layer of five neurons. The difference between the black and the blue and yellow points got a lot bigger, so only the absolute value of the difference is relevant and for simpler networks it is more difficult to compute the absolute value by themselves.

4 Changing the number of Jets

Until this point, the network was only given the 3 lead jets. As there is a preselection of at least 4 jets on the data set and about half of all events have more than 4 jets, one could expect, that using more jets leads to better results.

For this, one network of fixed size is used for all events. Empty entries are filled with zeros. Figure 14 shows the result of doing this for networks taking between 0 and 8 jets as input. The best result is obtained for 4 jets, so the network is not able to use more jets effectively.

In figure 15, the same is shown without early stopping. The resulting significances are generally better and have a slightly different shape, but they still decreasing for a high number of jets.

Figure 16 shows the influence of different data set sizes, network architectures and regularization techniques. It can be seen, that a more complex architecture only leads to slightly

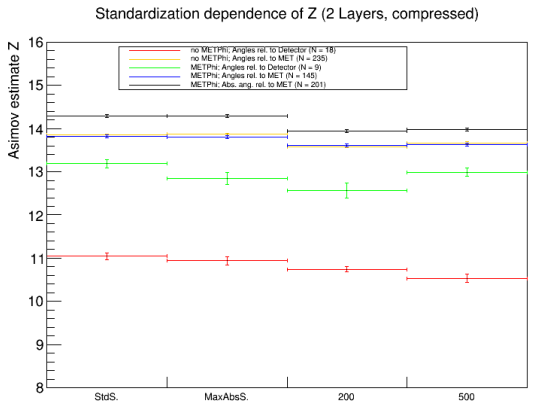

Figure 12: Dependence of the Asimov estimate of the significance on the preprocessing of the ϕ angles.

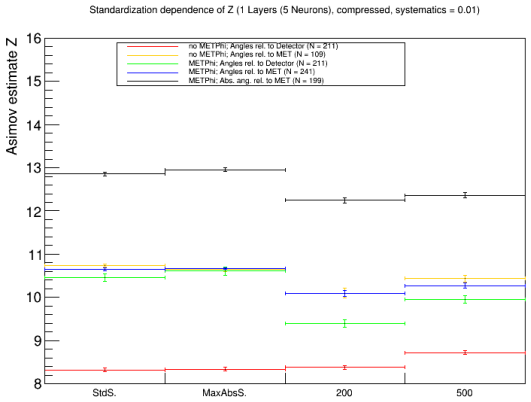

Figure 13: Dependence of the Asimov estimate of the significance on the preprocessing of the ϕ angles for a single layer 5 neuron network.

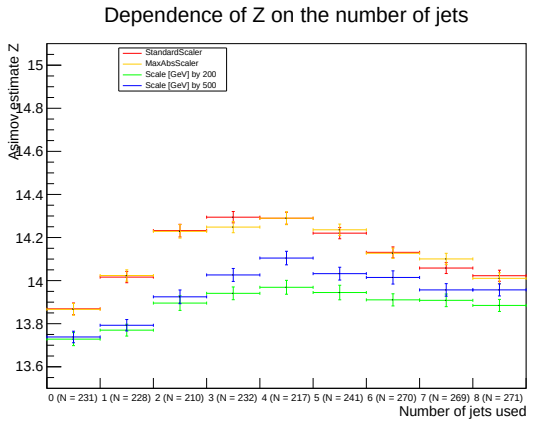

Figure 14: Dependence of the Asimov estimate of the significance on the number of jets used as inputs for the network.

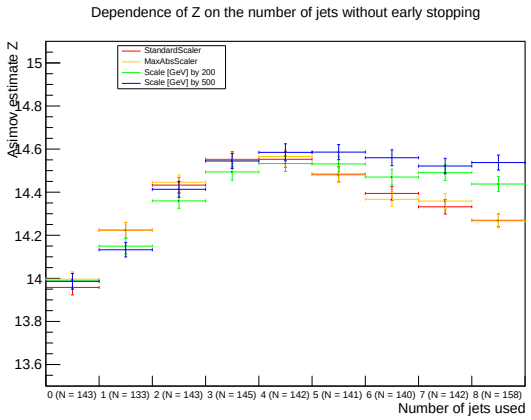

Figure 15: Dependence of the Asimov estimate of the significance on the number of jets used as inputs for the network without early stopping.

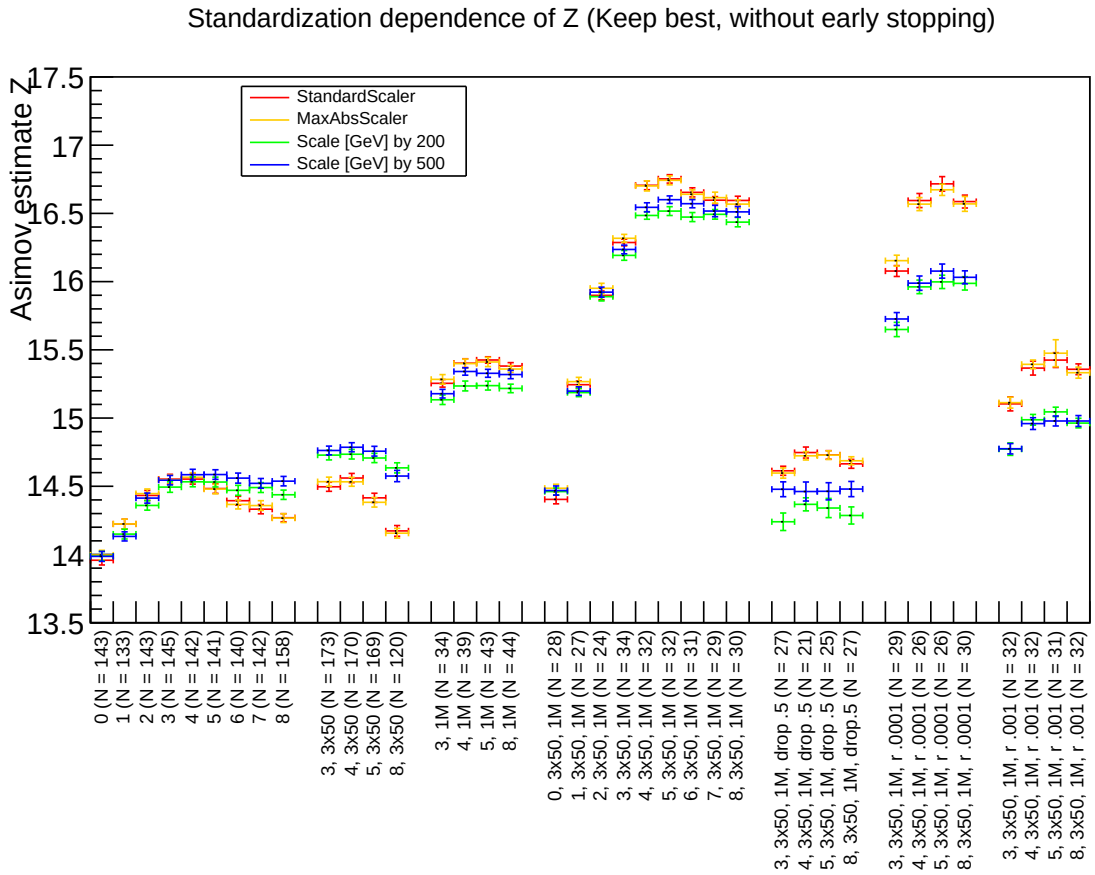

Figure 16: Dependence of the Asimov estimate of the significance on the number of jets used for different data set sizes, network architectures and regurarization techneques (3x50 means architecture [50, 50, 50]; 1M means, that 1 000 000 samples are used, drop stands for dropout and r for L2 regularization).

higher values and to an even more decreasing curve.

Using a larger data set leads to significantly better results and a less decreasing curve. Using both leads to even higher values for the significance. Using dropout and L2 regularization leads to worse results and only slightly changes the shape of the curve.

Training a network to effectively use more than 5 jets turns out to be difficult. Using even more data might solve the problem, but training will take relatively long. Other approaches like one network for each input size or a Recurrent Neural Network should be tried.

5 Conclusions

The dependence of the Asimov estimate of the significance on the standardization has been studied with the result, that scaling features with energy units by a common factor is beneficial, but is more sensitive to too early stopping.

It has been shown, that it is better to give the absolute values of the angles between ϕs

and ϕ_{MET} to the network instead of giving it the signed values or ϕ in detector coordinates.

Moreover, it could be seen, that it is difficult for a fixed size network to effectively use additional jets which are only available in a fraction of the data set.

References

[1] S. P. Martin, *A Supersymmetry primer*, doi:10.1142/9789812839657_0001, arXiv:hep-ph/9709356 [Adv. Ser. Direct. High Energy Phys.18,1(1998)].

[2] Michael Nielsen, http://neuralnetworksanddeeplearning.com/chap1.html

[3] A. Elwood and D. Krücker, *Direct optimisation of the discovery significance when training neural networks to search for new physics in particle colliders*, arXiv:1806.00322v1 [hep-ex] 1 Jun 2018.