EPJ Web of Conferences **158**, 06008 (2017) DOI: 10.1051/epjconf/201715806008
QFTHEP 2017

Deep Learning Neural Networks and Bayesian Neural Networks in Data Analysis

Andrey Chernoded[1], Lev Dudko[1], Igor Myagkov[1], and Petr Volkov[1]

[1]Skobeltsyn Institute of Nuclear Physics M.V. Lomonosov Moscow State University, Moscow 119991, Russian Federation

Abstract. Most of the modern analyses in high energy physics use signal-versus-background classification techniques of machine learning methods and neural networks in particular. Deep learning neural network is the most promising modern technique to separate signal and background and now days can be widely and successfully implemented as a part of physical analysis. In this article we compare Deep learning and Bayesian neural networks application as a classifiers in an instance of top quark analysis.

1 Introduction

Neural networks (NN) are especially useful in the top quark analyses owing to large background contributions to the same detector signature of the signal events. The efficient technique to distinguish signal events is critical to achieve reliable physics results. Different machine learning techniques are used to separate and suppress background processes such as multijet QCD events (QCD), W boson plus jets production (Wjets) and other Standard model (SM) processes. In the searches for the possible deviations from SM different machine learning techniques are used to distinguish Beyond of SM (BSM) effects from the huge SM background. For example in the searches for the anomalous Wtb interactions and flavour changing neutral currents (FCNC) [1] Bayesian neural networks (BNN) technique [2, 3] was applied. This technique is one of the most advanced approach compared to conventional Artificial neural networks. BNN allowed authors to successfully conduct a research on single top quark production properties, the analysis scheme is represented in Fig. 1. In the upcoming analysis the Deep neural networks can be used as a successors of BNNs.

In the article [4] it was shown that DNNs are capable to significantly improve precision of the analysis. In this article the suppression of QCD and SM events will be demonstrated using various techniques and packages, such as Markov Chain Sampling and Flexible Bayesian Modeling [5], Tensorflow [6] and Keras [7]. Significant improvement of the sensitivity has been demonstrated in comparison with usual NN methods.

2 Neural network introduction

A brief description of neural networks is presented in this section. Neural network is a set of neurons and their connections. Neurons have an input weights and activation functions. Set

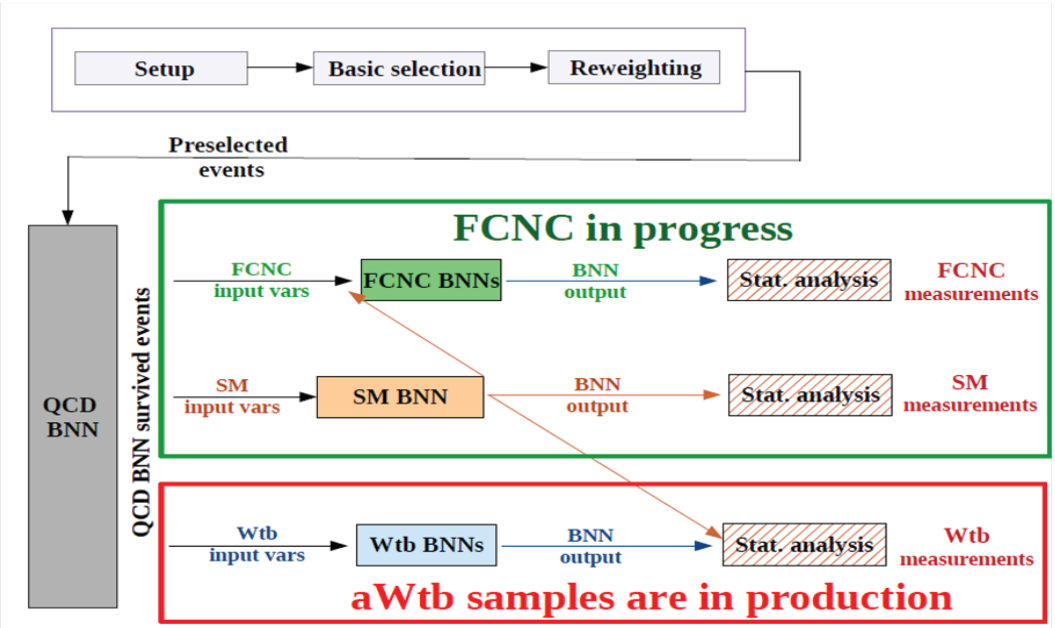

Figure 1. Analysis scheme, Bayesian neural network uses on four analysiss stages.

of neurons and their connections are a neural network architecture which is a very important part of conception. Usually neural networks are schematically described with the help of perceptron. Neural network can fit some function by means of training with simulated data. During the training algorithm minimizes a loss function with the help of changing of weights of neurons. For the productive work of the network, it is necessary to choose efficient architecture and loss function. There are not strict and universal recipe how to find optimal architecture and parameters to train NN, but with some experience one can achieve very good results.

Loss function is the basic element of learning algorithms. In mathematical optimization, statistics, econometrics, decision theory, machine learning and computational neuroscience, a loss function or cost function is a function that maps an event or values of one or more variables onto a real number intuitively representing some "cost" associated with the event. An optimization task tries to minimize a loss function.

Machine learning uses concepts such as algorithm optimizers, training and testing datasets, and criteria to estimate efficiency. They will be described in more details below.

3 Bayesian neural network

Bayesian neural network is an ensemble of neural networks. This approach is more stable and avoids the problem of overfitting one specific network. Programming package for BNN training is Flexible Bayesian Modeling and Markov Chain Sampling package [5]. This software supports Bayesian regression and classification models based on neural networks and Gaussian processes, Bayesian density estimation and clustering using mixture models and Dirichlet diffusion trees. It also supports a variety of Markov chain sampling methods which may

be applied to distributions specified by simple formulas, including simple Bayesian models defined by formulas for the prior and likelihood [2]. In addition "bnn-hep" interfacing package is useful for creating training sets and submitting training results. It allows to preprocess input data and combine it into complex variables required by the analysis Fig. 2. The hybrid Monte Carlo algorithm [3], which is also known as Hamiltonian Monte Carlo, is used during network training. It uses Markov Chain Monte Carlo method for obtaining a sequence of random samples from a probability distribution for which the direct sampling is difficult. This sequence can be used to approximate the distribution (i.e., to generate a histogram), or to compute an integral (such as an expected value).

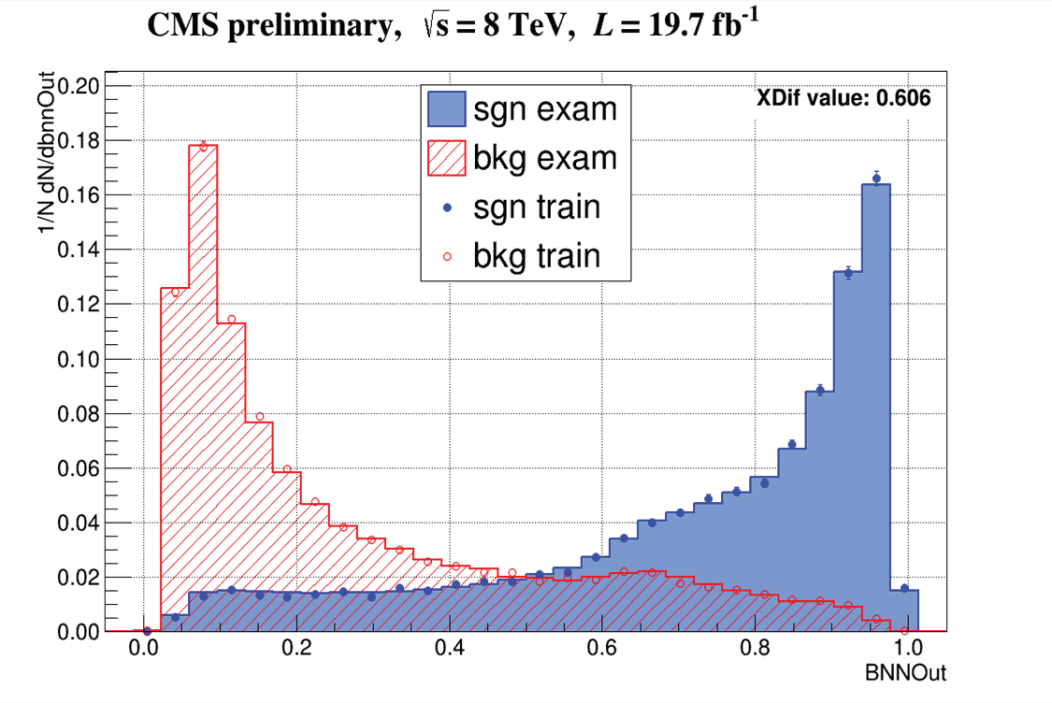

Figure 2. Example of applying Bayesian neural network

4 Deep Learning neural networks

Deep learning neural networks are distinguished from the conventional Artificial neural networks by their depth; that is, the number of node layers through which data is passed in a multistep process of pattern recognition is usually more than 2. Traditional machine learning relies on shallow nets, composed of one input and one output layer, and at most one hidden layer in between. More than three layers (including input and output) qualifies as "deep" learning. In deep-learning networks, each layer of nodes trains on a distinct set of features based on the previous layer's output.

TensorFlow [6] is an open source software package for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges

represent the multidimensional data arrays (tensors) communicated between them. The flexible architecture allows to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API. TensorFlow was originally developed by researchers and engineers working on the Google Brain Team within Google's Machine Intelligence research organization for the purposes of conducting machine learning and deep neural networks research, but the system is general enough to be applicable in a wide variety of other domains as well.

Keras [7] is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation. Keras allows to create networks with different architecture, loss function, activation function, algorithm optimizers and other features.

Activation functions such as Sigmoid function is a mathematical function having a characteristic "S"-shaped curve or sigmoid curve. Often sigmoid function refers to the special case of the logistic function shown in the first figure and defined by the formula

$$S(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1}.$$

In the context of artificial neural networks, the rectifier is an activation function defined as:

$$f(x) = x^+ = \max(0, x),$$

where x is the input to a neuron.

In mathematics, the softmax function, or normalized exponential function, is a generalization of the logistic function that 'squashes' a K-dimensional vector $\mathbf{z}$ of arbitrary real values to a K-dimensional vector $\sigma(\mathbf{z})$ of real values in the range [0, 1] that add up to 1. The function is given by $\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$ for j = 1, ..., K.

Algorithm optimizers such as Stochastic Gradient Descent (SGD) is a simple and efficient approach to discriminative learning of linear classifiers under convex loss functions such as (linear) Support Vector Machines and Logistic Regression. The advantages of Stochastic Gradient Descent is efficiency. The disadvantages of Stochastic Gradient Descent include: SGD requires a number of hyperparameters such as the regularization parameter and the number of iterations. SGD is sensitive to feature scaling. Adam is an optimization algorithm that can be used instead of the classical SGD procedure to update network weights iteratively based on training data. Keras allows for a number of regulization methods. Regularization is a key component in preventing overfitting. Also, some techniques of regularization can be used to reduce model capacity while maintaining accuracy, for example, to drive some of the parameters to zero. This might be desirable for reducing model size or driving down cost of evaluation in mobile environment where processor power is constrained.

Dropout is a technique where randomly selected neurons are ignored during the training. They are "dropped-out" randomly. This means that their contribution to the activation of downstream neurons is temporally removed on the forward pass and any weight updates are not applied to the neuron on the backward pass [8].

5 Neural network efficiency estimate criteria

The Receiver Operating Characteristic (ROC) curve and the Mean-Square Error (MSE) criteria were taken to estimate the performance of the network.

A ROC curve is a way to compare diagnostic tests. It is a plot of the true positive rate against the false positive rate. A ROC curve demonstrates a relationship between sensitivity and specificity. For example, a decrease in sensitivity results in an increase in specificity. Test accuracy: if the graph is closer to the left and to the top borders, the the test is more accurate. Otherwise, if the graph is closer to the diagonal, the test is less accurate. A perfect test would go straight from zero up the top-left corner and then straight across the horizontal. The likelihood ratio: given by the derivative at any particular cut point. Area under ROC curve is usually used as a most simple estimation criteria.

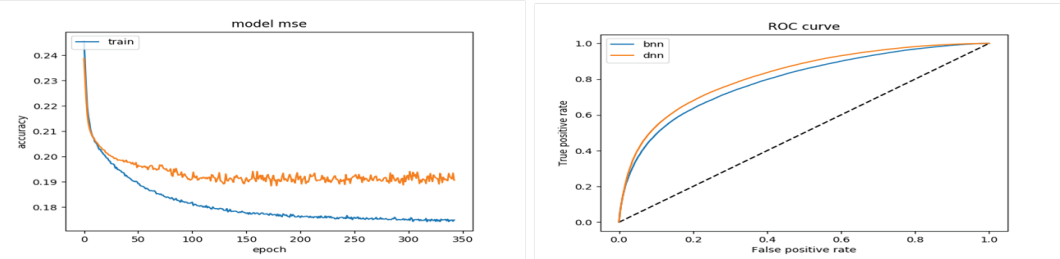

Figure 3. Estimate criteria: MSE and ROC curve

If $\hat{Y}$ is a vector of n predictions, and Y is the vector of observed values corresponding to the inputs of the function which generates the predictions, then the MSE of the predictor can be estimated as

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2$$

6 QCD Network

One of the main backgrounds for the single top is the process of QCD-multijet production. It can be suppressed using classifier which is capable of distinguishing QCD-multijet from other Standard model processes. In QCD-multijet production process muons originate either from leptonic decay of heavy hadrons or from charged hadrons. As a result, these muon candidates are usually surrounded by hadronic activity. This feature is exploited to define a QCD control region in which exactly one muon with passes inverted criterion of isolation from hadronic activity. In order to better reproduce kinematics of the signal region, the jets falling inside a cone of size 0.5 around the selected muon are not considered in the analysis. The surviving jets are subject to the same selection as in the signal region[1]. The neural network is then trained using events from the QCD control region. The training set is composed of 120'000 events with QCD-multijet defined as the background and various Standard Model processes associated with single top quark production as the signal. Bayesian network is formed from an ensemble of 20 networks. Deep network consists of 5 hidden layers with 50 neurons and uses regularization and dropout.

Figures 4-6 show different behavior of separation power and ROC curves in three scenarios. In the first scenario, which results are shown on Fig.4, neural networks were given high-level variables, which had been evaluated prior to training step. Such variables should contain most important information required for particular classification task, their set is a subject to choice of analyst and are selected with respect to underlying physical process. In the second

scenario, shown on Fig.5 only the most basic variables are provided as a neural network input. Thus a network is encouraged to find most suitable features from the most essential vectors of data. In the third scenario, shown on Fig.6, the networks are given a complete list of variables which incorporates both of the previous.

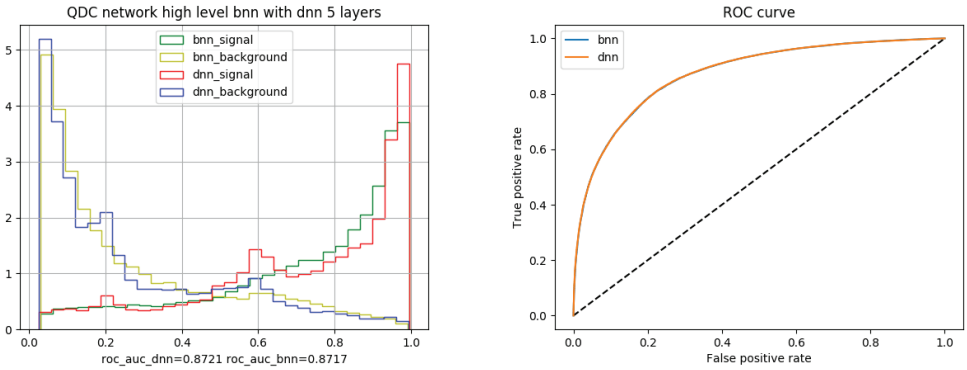

Figure 4. Comparison of QCD networks trained on high level variables.

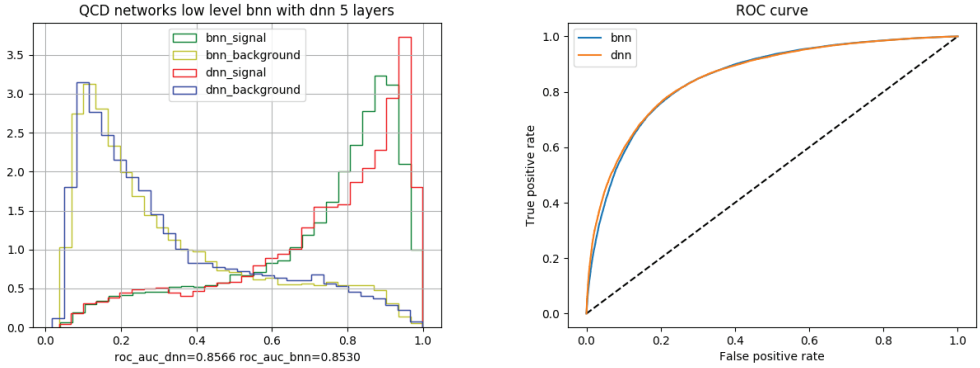

Figure 5. Comparison of QCD networks trained on low level variables.

7 SM network

Distinguishing between single top production process and its background on per-event level is next crucial step of a single top analysis. This classification task is accomplished using either kind of classifier; Bayesian and Deep neural networks will be considered in this section.

After preforming basic initial selection and suppressing QCD-multijet background, as described in previous section, events which pass such filtering are then passed as input to Standard model neural network. The purpose of this classifier is to assign a value of likeliness

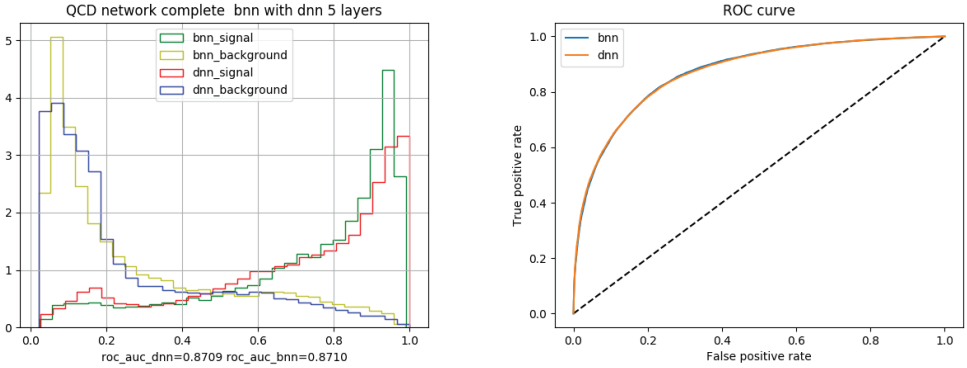

Figure 6. Comparison of QCD networks trained on complete set of variables.

of the event of being produced in t-channel single top production process. Amongst main undesired backgrounds to this process are tW-channel single top process, paired top quark production process, W+jets, diboson and Drell-Yan[1].

As well as a QCD-multijet neural network, Standard Model network takes a set of basic and complex kinematic variables as its input. Training set is composed of 190'000 events for train and test sets. In first approach using Bayesian neural network, the ensemble is composed of 50 networks. Deep network is formed using 5 hidden layers with 100 neurons in each with applied dropout regularization technique.

Figures 7-9 show different behavior of separation power and ROC curves in three scenarios, same as in section 6.

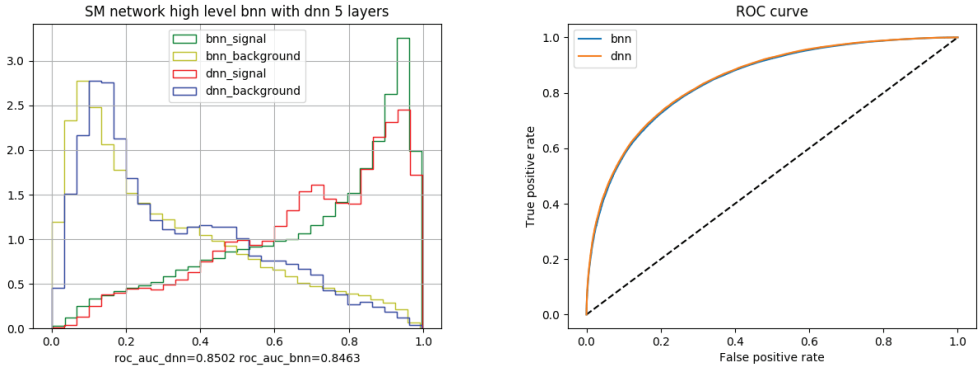

Figure 7. Comparison of SM networks trained on high level variables.

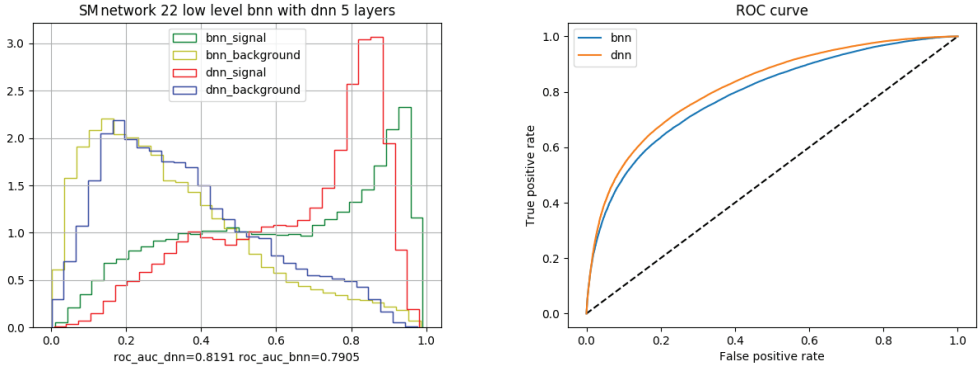

Figure 8. Comparison of SM networks trained on low level variables.

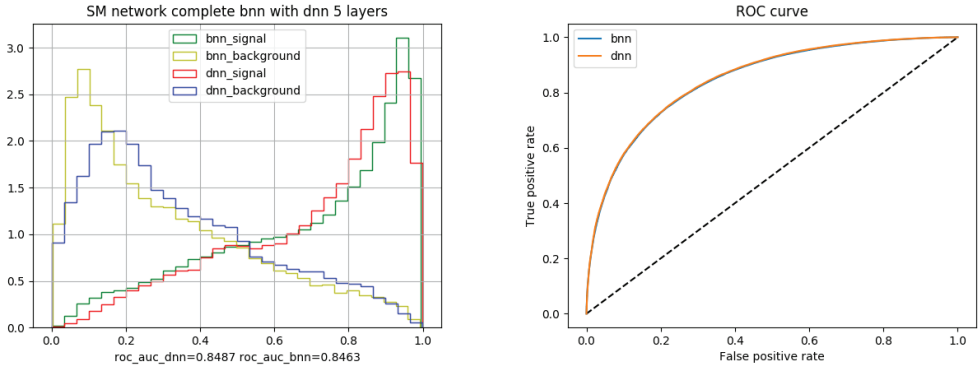

Figure 9. Comparison of SM networks trained on complete set of variables.

8 Conclusion

Neural networks is a powerful technique which allows creating sophisticated classifiers and is widely used in modern physics analysis such as single top quark study considered in this article. In the article authors provide the comprehensive list of packages, regularization and optimization techniques for creating and learning various models of neural networks. Two different techniques for neural network classifiers were implemented, such as Bayesian and deep neural networks. Their separation power and training aspects were considered. The Bayesian neural networks is an improved technique over conventional artificial neural networks, while Deep neural networks allow creating sophisticated multilayer architectures. In combination with modern learning methods and tools DNN is promising technique to significantly improve classifiers sensitivity and overall precision.

References

[1] Search for anomalous Wtb couplings and flavour-changing neutral currents in t-channel single top quark production in pp collisions at sqrt(s)= 7 and 8 TeV. CMS Collaboration. Oct 11, 2016 - 40 pages JHEP 1702 (2017) 028

[2] Neal, R. M. (1994) Bayesian Learning for Neural Networks, Ph.D. Thesis, Dept. of Computer Science, University of Toronto, 195 pages

[3] Neal, R. M. (1992) "Bayesian training of backpropagation networks by the hybrid Monte Carlo method", Technical Report CRG-TR-92-1, Dept. of Computer Science, University of Toronto, 21 pages

[4] Searching for Exotic Particles in High-Energy Physics with Deep Learning - Baldi, Pierre et al. Nature Commun. 5 (2014) 4308 arXiv:1402.4735 [hep-ph]

[5] http://www.cs.toronto.edu/~radford/fbm.software.html

[6] https://www.tensorflow.org/

[7] https://keras.io/

[8] Dropout: A Simple Way to Prevent Neural Networks from Overfitting - Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky,Ilya Sutskever, Ruslan Salakhutdinov. Journal of Machine Learning Research 15 (2014) 1929-1958