APPLICATION OF NEURAL NETWORKS OPTIMIZED BY GENETIC ALGORITHMS TO HIGGS BOSON SEARCH

František Hakl[*]*, Marek Hlaváček*[**]*, Roman Kalous*[**]
[*]Institute of Computer Science AS CR, Pod Vodárenskou věží 2, 182 07 Prague 8, Czech Republic
[**]Mathematics department, Faculty of Nuclear Science and Physical Engineering, Czech Technical University, Prague, Czech Republic

Abstract

This paper describes an application of a neural network approach to SM (standard model) and MSSM (minimal supersymetry standard model) Higgs search in the associated production $t\bar{t}H$ with $H \to b\bar{b}$. This decay channel is considered as a discovery channel for Higgs scenarios for Higgs boson masses in the range 80 - 130 GeV. A neural network model with a special type of data flow is used to separate $t\bar{t}jj$ background from $H \to b\bar{b}$ events. This neural network combines a classical neural network and linear decision tree. Parameters of these neural networks are randomly generated and a population of predefined size of those networks is trained as an initial generation for a subsequent genetic algorithm optimization. A genetic algorithm tunes parameters of further neural network individuals derived from the previous neural networks by GA operations of crossover and mutation. The goal of this GA process is optimization of the final neural network performance.

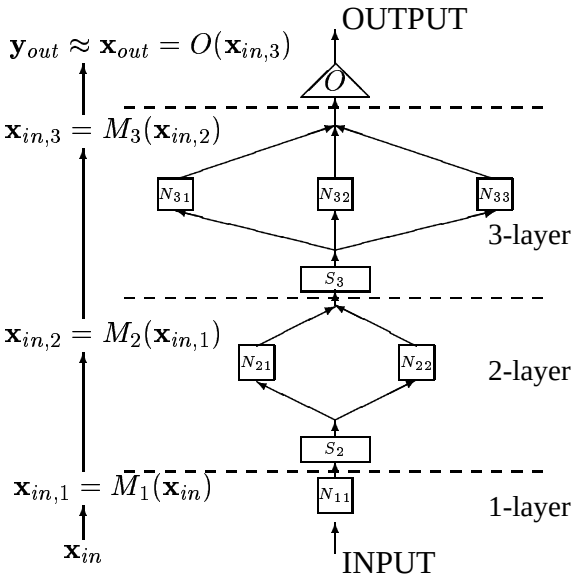
Our results show that NN approach is applicable to the problem of Higgs boson detection. Neural network filters can be used to emphasize the difference of the M_{bb} distribution for events accepted by filter (with better $\frac{signal}{background}$ rate) and the M_{bb} distribution for original events (with original $\frac{signal}{background}$ rate) with no loss of significance. This improvement of the shape of the M_{bb} distribution can be used as a criterion for existence of Higgs boson decay in considered discovery channel.

1 Introduction

This work is devoted to the application of neural networks to high energy physics. There is a broad consensus in physics community that the Large Hadron Collider (LHC) at CERN, should have the capability to confirm the presence of Higgs boson. Using simulated output from the LHC based on the simulation package PYTHIA, we have available two sets of events, one with Higgs boson decay (we denote this as signal) and a second without Higgs boson decay (we denote this as background). So we can see the problem of Higgs boson search as a classical problem of pattern recognition with the exception that the quality of separation is measured as a difference between two distribution curves corresponding to separated sets.

Neural nets are widely used in pattern recognition problems and as functions approximation tools. There are many types of artificial neural networks which differ in architecture, in the type of implemented transfer functions and strategy of learning. In view of their universal approximation property, we decided to use a special kind of neural nets, namely neural network with switching units [1], [2], [3] to solve the pattern recognition problem postulated above. To reach better performance of these neural networks, we tune the topology and parameters of such networks via a genetic algorithm optimization.

[*]This work is supported by grant of Ministry of Trade and Industry of the Czech Republic, Project No. RP-4210/69/97.



Switching net maps input $\mathbf{x}_{in} \in R^d$ into output

$$\mathbf{x}_{out} = M(\mathbf{x}_{in}) \in R^b,$$

where the mapping M is a composition of mappings performed by each layer

$$M = S \odot M_{nl} \odot M_{nl-1} \odot \cdots \odot M_2 \odot M_1,$$

n_l is number of layers, $M_1 = N_{11}$, and

$$\mathbf{x}_{in,k} = M_k \odot \cdots \odot M_1(\mathbf{x}_{in})$$

$$M_k = N_{kS_k(\mathbf{x}_{in,k-1})}, \quad \text{for} \quad k > 1.$$

Fig. 2: A topology of simplified building block.

2 Description of neural network with switching units

Neural network with switching is a combination of a classical neural network architecture and a decision tree. This network is actually an oriented acyclic graph (see Fig. 1) whose nodes are structures called building blocks.This acyclic graph will be referenced as the outer graph.

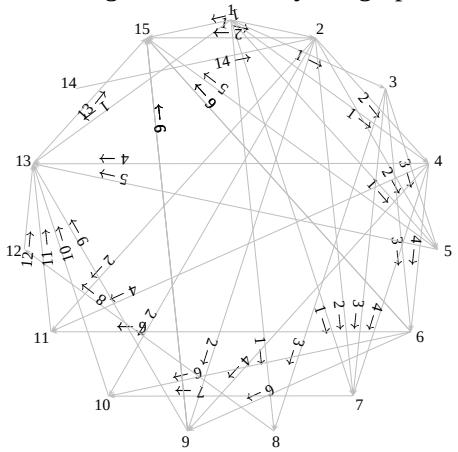


Fig. 1: Schema of connection between building blocks.

Each building block is a neural network consisting of two types of nodes. These nodes are connected in such a way that they again form an acyclic graph but with the restriction that the outputs dimension of the building block is the same for all building blocks in the outer graph. The first type of node, which we refer to as a functional unit, makes a predefined mapping from the input space to the output space of this node. Hence such node can be described by a tuple of integers (input vector dimension and output vector dimension) and by its transfer function. The definition of this transfer function includes parameters of this functional unit (weight vectors, threshold etc. in current neural networks terminology).

This transfer function can differ for each functional unit. For example, let now we describe transfer the function currently implemented. Let $\vec{\boldsymbol{x}}_i \in \Re^n$, $i \in \{1, \ldots, p\}$ are input patterns into the functional unit. Let its corresponding desired outputs be denoted as $\vec{\boldsymbol{y}}_i \in \Re^m$, $i \in \{1, \ldots, p\}$. The desired output of functional unit can generally be different from the desired output of the whole network, but for the sake of clarity we assume that desired output of each functional unit is the same as the desired output of the whole network. A simple case of a transfer function is a linear mapping which minimizes the norm of the vector $\boldsymbol{AW} - \boldsymbol{Y}$, where $\boldsymbol{W} \in \Re^{m \times n}$ is matrix of weight parameters, $\boldsymbol{A} \in \Re^{n \times p}$ is a matrix which rows are vectors $\vec{\boldsymbol{x}}_i$ and $\boldsymbol{Y} \in \Re^{m \times p}$ is a matrix which rows are formed by vectors $\vec{\boldsymbol{y}}_i$.

The second type of nodes, switching units, collect all outputs from parent functional units, concatenate them together to form one vector, and search for a predefined number of clusters in the set of such input vectors. We use the Jancey cluster algorithm which is a non-deterministic procedure described in the following schema:

Let d be the number of desired clusters (which is equal to the number of switching unit children).

$\vec{z}_1, \ldots, \vec{z}_p$ are output vectors of switching unit parents, $\|\vec{x}\|_E$ is Euclidean norm of the vector $\vec{x}$.

1. for a randomly chosen sequence $1 \leq j_1 < j_2 < \cdots < j_d \leq p$ set centers

$$\vec{c}_q^{new} = \vec{c}_q^{old} = \vec{z}_{j_q} \quad \text{and sets} \quad \bar{S}_q^{new} = \bar{S}_q^{old} = \{\vec{z}_{j_q}\}, q \in \{1, \ldots, d\}$$

2. let $r_1, \cdots, r_p$ be a random permutation of the $1, \cdots, p$,
3. FOR ALL $k = r_1, \cdots, r_p$
   DO
   find an index q such that $\vec{z}_k \in \bar{S}_q^{old}$, $\rho = \min_{h \in \{1,\ldots,d\}} \left\{ \left\| \vec{c}_h^{old} - \vec{z}_k \right\|_E \right\}$
   $i = \min \left\{ v \quad \middle| \quad \left\| \vec{c}_v^{old} - \vec{z}_k \right\|_E = \rho \right\}$
   and redefine:
   $c_q^{old} = \vec{c}_q^{old} - \frac{\vec{z}_k - \vec{c}_q^{old}}{|S_q^{old}|}$, $c_i^{old} = \vec{c}_i^{old} + \frac{\vec{z}_k - \vec{c}_i^{old}}{|S_i^{old}|}$
   $\bar{S}_q^{old} = \bar{S}_q^{old} \setminus \{\vec{z}_k\}$, $\bar{S}_i^{old} = \bar{S}_i^{old} \cup \{\vec{z}_k\}$,
   END
4. IF $(\exists q)(\bar{S}_q^{new} \neq \bar{S}_q^{old})$
   THEN for all such q let $\vec{c}_q^{new} = \vec{c}_q^{old}$, $\bar{S}_q^{new} = \bar{S}_q^{old}$ and GOTO 2
5. STOP

After clustering each cluster is associated with a corresponding child functional unit and the parameters of this functional unit are adjusted with regard to patterns in the corresponding cluster only. In fact, division of input patterns into two or more disjoint sets, and consecutive learning over these subsets of patterns, put a separation hypersurface into the input space. The type of these hypersurfaces is defined by the type of transfer functions of switching unit parents.

So each building block is learned, the output from each building block is propagated to all children, and the output of the top building block is considered as final output from the neural network.

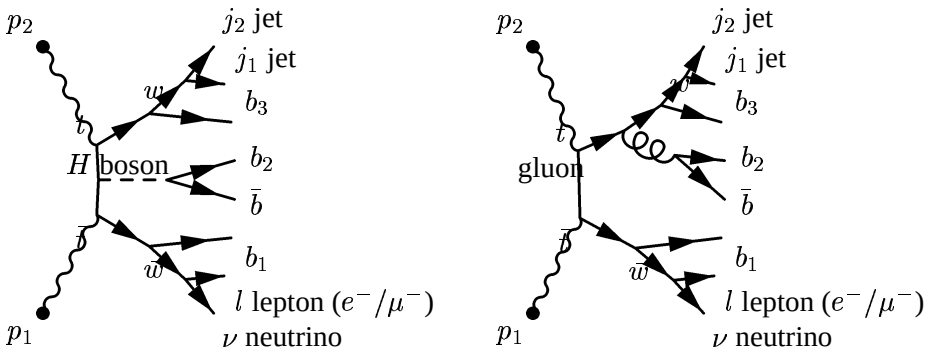
3 Tuning of neural nets via GA procedure

The computational power of a neural network depends in general on

1. structure and connection state (topology of the net)
2. learning method.

The second one is more or less question of amount of learning data and quality of the learning method. The main goal of each learning method is optimization of some fitness function defined on the set of all possible neural net parameters. We view this problem as a global optimization of the fitness function. There are many gradient methods to find a local solution (minimum or maximum). But these methods does not allow to find a global extreme point of the fitness function due to the non-continuous nature of some parameters (types of transfer function, number of inputs, graph connections, etc.). We need to use some nongradient global optimization method to do this. But many deterministic techniques of global optimization, like divide and conquer, or analytic extreme, aren't efficient on acyclic graph problems. Therefore we decided use genetic algorithms to set up the parameters and topology of neural net. Theoretical analysis of GA implies that no best solution is reached, but the average fitness increases at all in new generations. This is done due a special property of GA which is known as a implicit parallelism (see [5]). Briefly, the implicit parallelism theorem implies that an exponentially large sets of parameter space are searched for domains with above-average fitness in polynomial time.

Concerning the implementation, a queue (FIFO) of randomly generated networks is constructed (this queue has user predefined length). Then every network is trained and tested, and its fitness evaluated. We use the following fitness functions (denoting S_a by accepted signal, e.g. all signals correctly classified, B_a accepted background, S_r rejected signal, e.g. all misclassified signal, B_r rejected background) at this moment:



(a) Signal (b) Background

Fig. 4: Feynman diagram of decay trees.

1. maximal enrichment factor, e.g. $\max \frac{S_a}{S_r}$ under condition that the amount of accepted signal will be statistically significant
2. minimal loss of signal under condition on amount of rejected background, e.g. $\max S_a$ and B_r =predefined value
3. maximal rejection of signal under condition on amount of accepted signal, e.g. $\max B_r$ and S_a =predefined value
4. maximization of quality factor, e.g.
$$\max \frac{S_a}{\sqrt{S_a + B_a}}.$$
5. negative value of total mean square error over all events

The values of fitness of the networks in the queue forms the basis for the probability according to which the parents of newborn networks are chosen. After parents are selected, crossover is considered. Crossover is an interchange of corresponding blocks. Note that blocks have the same number of inputs and the same number of outputs, so they are changeable. Another genetic operator considered is mutation which should be used very rarely. We have some mutation operation (edge removing, edge adding, activation function change, etc.) for disposal. The new structures is grown, then trained, tested and evaluated. And so on, until some criteria are reached, for example generation count, acceptable level of specified fitness, etc.

4 Application of neural network with switching units and genetic algorithms to Higgs boson search

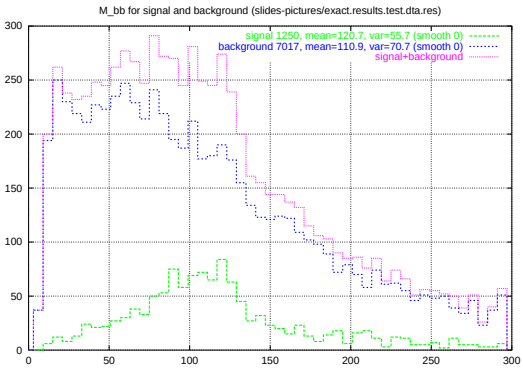


Fig. 3: Histogram of $M_{b\bar{b}}$ for signal and background.

As we already mentioned the objective of our work is a search for $H \rightarrow b\bar{b}$ decay. There is a certain probability that Higgs bosons are produced in considered collision (see Fig. 4, a)). In the case that the mass of the Higgs boson is $m \leq 200 GeV/c^2$ a decay $H \rightarrow b\bar{b}$ dominate. The main background (see Fig. 4, b)) involves a gluon instead of the Higgs boson, producing the same final state.

The products of this type of collision are 2 jets from one W decay, 4 b-jets, and one lepton (electron or muon) plus missing energy from an unobserved neutrino. Each visible particle (2 jets, 4 b-jets and lepton) is described by three values

P_T – transverse momentum $[GeV/c^2]$, η – pseudorapidity and ϕ – direction angles, corresponding to particle vectors with negligible mass. The neutrino is described by two missing energy values E_x^{miss}, E_y^{miss}. A fundamental variable which can be used in the Higgs boson search is a effective mass M_{bb} of two b-'s which can arise either form Higgs boson decay or from gluon decay after $p\bar{p}$ collision. There are lot of events with gluon decay (background) and much fewer events with Higgs decay (signal). Each of these two classes of events have different distribution of the effective mass M_{bb}. The Higgs boson decay is theoretically a Gaussian distribution with mean 120 GeV/c^2 and $\sigma = 15GeV/c^2$, whereas the gluon decay is much broader. The difference between those two distributions can be exploited to decide if Higgs boson decay is present in the data or not. In addition, other physical reasons reject all events in which at least one of the following conditions has been satisfied: at least one jets has $P_T < 15GeV$, at least one jet has pseudorapidity out of the range $(-2.5, 2.5)$, the lepton is electron and $P_T^{lep} < 20GeV$, the lepton is muon and $P_T^{lep} < 6GeV$. All events passing those restrictions form the histogram in Fig. 3. Really data do not provide information about the presence of Higgs decay in the event, hence for real data we have available the total distribution of M_{bb} (see Fig. 3, upper curve) only. For simulated data, we can plot two histograms of M_{bb}, one for background only and the second one for pure signal (see Fig. 3, two lower curves). Neural networks assume we know the distributions of signal and separated background. So the main idea of how to exploit neural networks to confirm Higgs decay presence is based on filtering events in such a way that percentage of signal will be increased after filtering and at the same time significance $\frac{S_a}{\sqrt{S_a+B_a}}$will stay on the same level.

5 Results and Conclusions

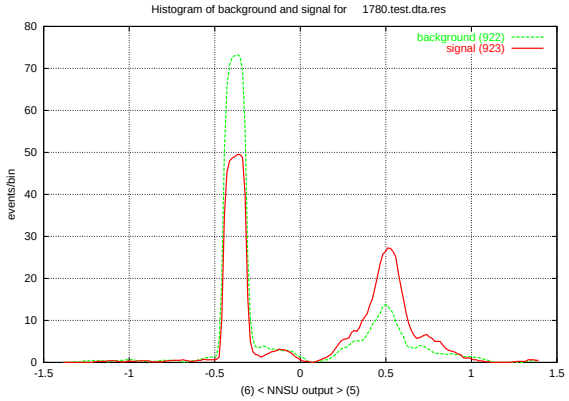
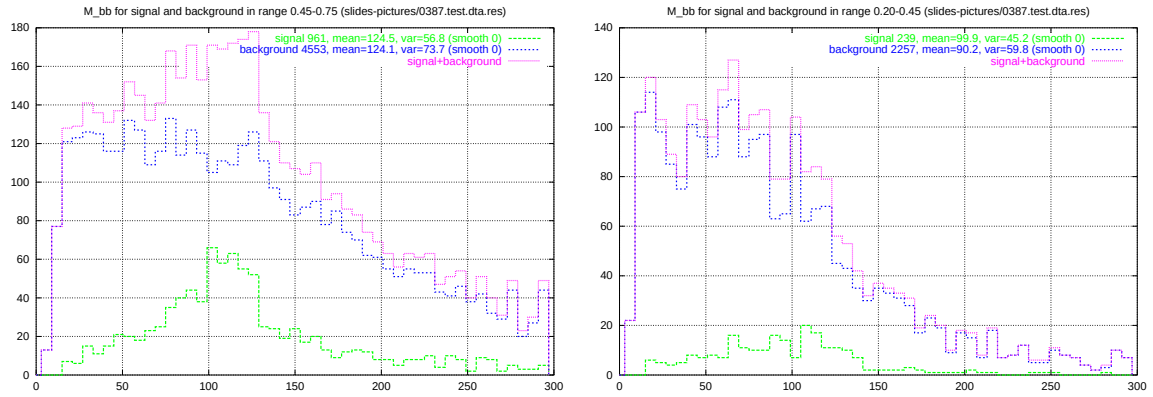


Fig. 5: Histogram of neural network output for signal and background.

Some experiments were performed with signal and background data described in section 4. We use raw data from PYTHIA. Our first results are demonstrated in plot 5. It is evident that neural nets with switching units are able to partly separate signal with Higgs decay and background without Higgs decay. As we can seen on the plot mentioned, it is possible to choose an interval in which signal dominate background, for example $(0.45, 0.75)$. We call such an interval a best signal window. On the other hand we can take interval, in which the signal is suppressed and background will be dominant, for example $(0.20, 0.45)$. We call such an interval a best background window. If Higgs decay is present than we can assume that a plot of M_{bb} over all events mapped by the neural network into the best signal window will differs from based on events mapped into the best background window. In fact, for our simulated data these plots do differ: see Fig. 6 a) and 6 b). Of course these plots should be different from the plot on Fig. 3.

Hence our first experiments convinced us that the chosen approach to separation of Higgs decay seems to be applicable and promise useful detection methods.

Finally we point out that developed separation method based on neural networks with switching units and genetic optimization is universal separation method which can be used for various pattern recognition problem. Perhaps some may find this method too extensive, especially the GA part, but no effective method for neural net topology and parameter tunning is known to this time. In the future, we plan to implement further transfer function for functional nodes, paralleling of transfer function optimization via taking some parallel version of optimization routine (parallel LAPACK, for example)



(a) Signal window: M_{bb} for events mapped into range 0.45-0.75.

(b) Background window: M_{bb} for events mapped into range 0.20-0.45.

Fig. 6: Signal and background windows.

and apply this separation tool to another pattern recognition problems.

References

[1] F. Hakl and M. Jiřina. "Design of a neural net for level-two triggering in Atlas nuclear experiments." *Neural Networks World*, vol. 6(6): pp. 951–973, 1996.

[2] P. Bitzan, J. Šmejkalová, and M. Kučera, "Neural networks with switching units." *Neural Network World*, vol. 4, pp. 515–526, 1995.

[3] M.Jiřina and M.Jiřina jr., "Neural network classifier based on growing hyperspheres" *Neural Network World*, vol. 3, pp. 417–428, 2000.

[4] M. Sapinski, "The $t\bar{t}b\bar{b}$ background to the Higgs searches: Pythia versus CompHEP rates." Tech. rep. no. ATL-PHYS-2000-020, CERN Geneve, 2000.

[5] John H. Holland. "Adaption in Natural and Artificial Systems." *A Bradford Book, The MIT Press*, 1992.

[6] T. Sjöstrand. "PYTHIA 5.7 and JETSET 7.4 Physics and Manual." tech. report *CERN - TH . 7112/93*, dec 1993.