28th International Scientific Conference of Young Scientists and Specialists (AYSS-2024)

A new algorithm for optimising the parameters of a high-performance neural network

V.Kiseeva

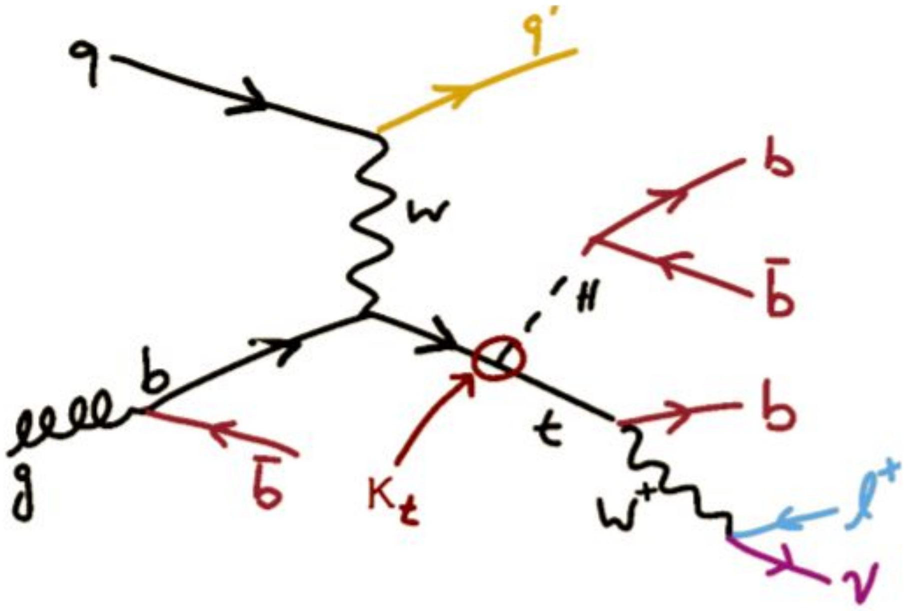
JINR team: I.Boyko, A.Didenko, O.Dolovova, N.Huseynov, V.Lybushkin, I.Yeletskikh

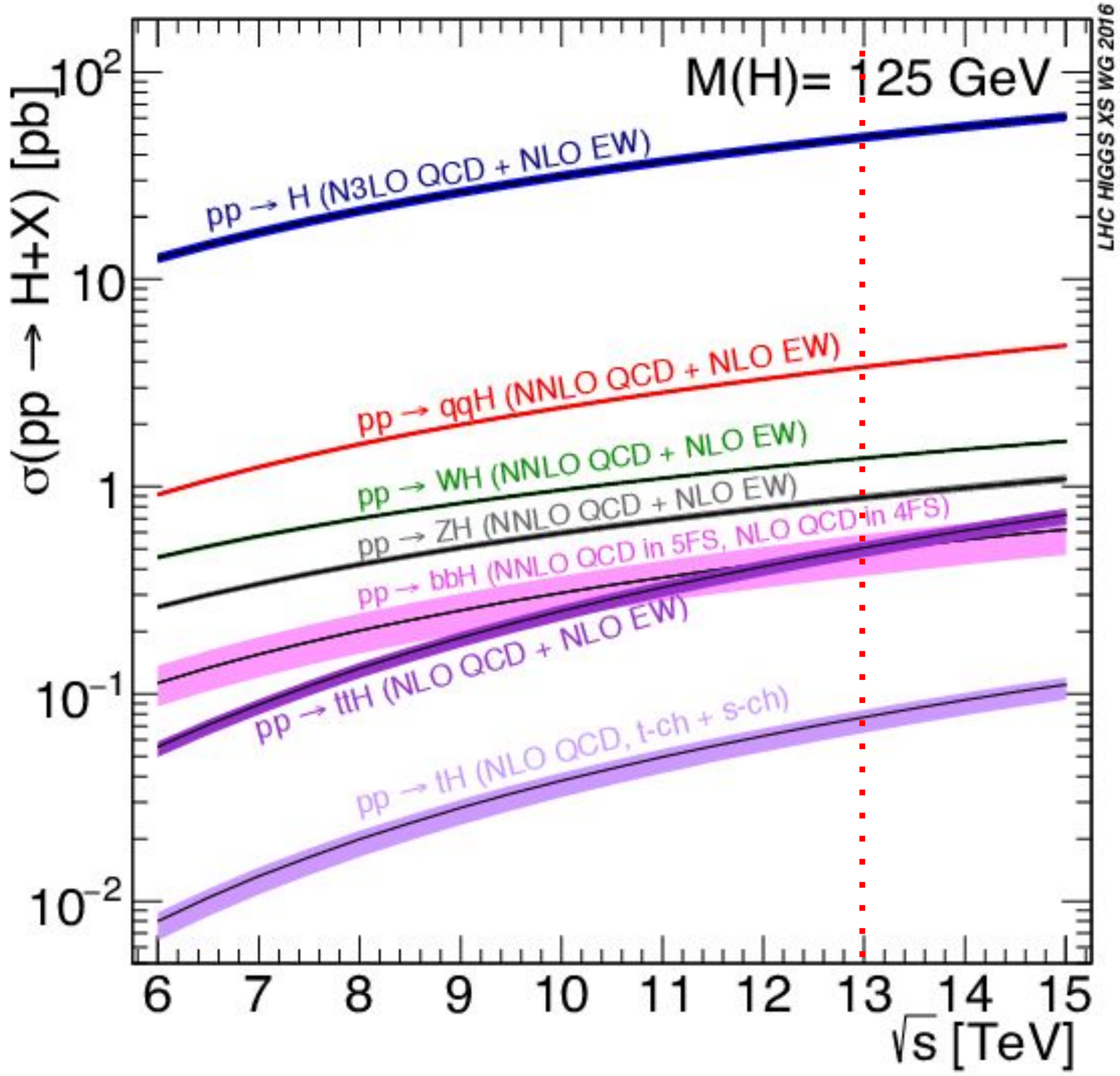
29.10.2024

Motivation: tH signal

- The number of signal events is extremely small compared to the number of background events

for LHC Run2 expected: ~100 tH; ~200k total background



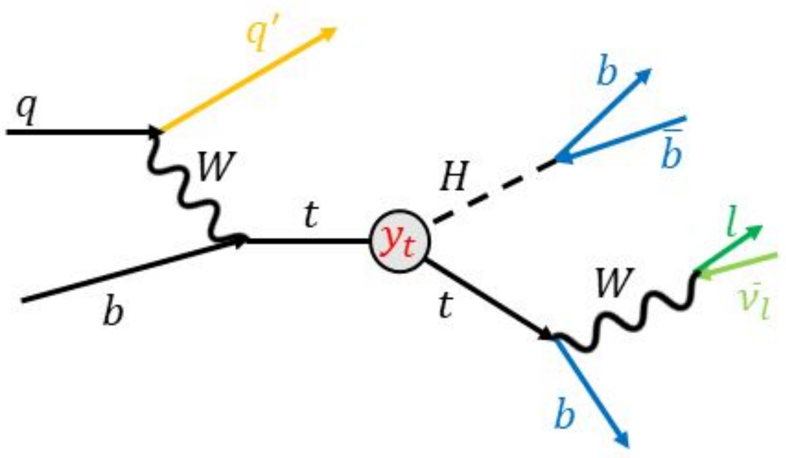


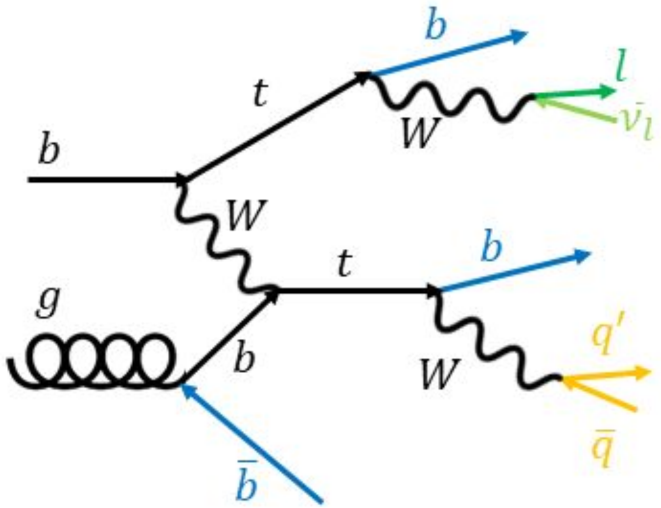
Signature of tH with decay $H \rightarrow b\bar{b}$:
(≥3 b-jets) + (1 light jet) + (1 tight lepton) + (missing transverse momentum)

[1] Higgs Physics, C. Grojean [arXiv:1708.00794]

Motivation: signal vs background

- The kinematics of signal and background processes are very similar





signal: tH process

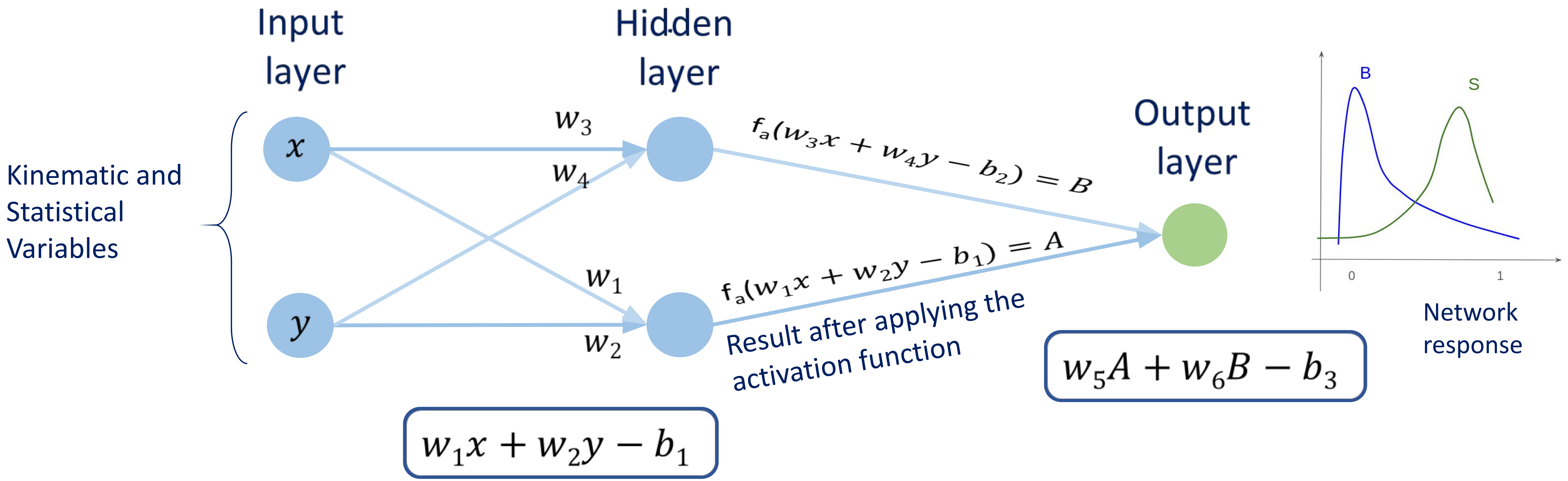
main background: ttb production channel

DATA		
tH	MiniNtuplethbqSM_300K.root	Entries :8300
tt	MiniNtiple_tt_SM_3M.root	Entries : 13994

Neural network

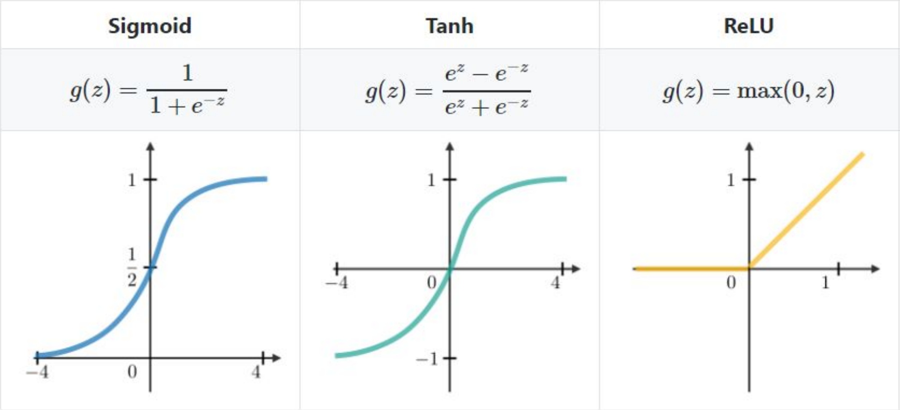
A **neural network** is a function with large number of internal parameters.

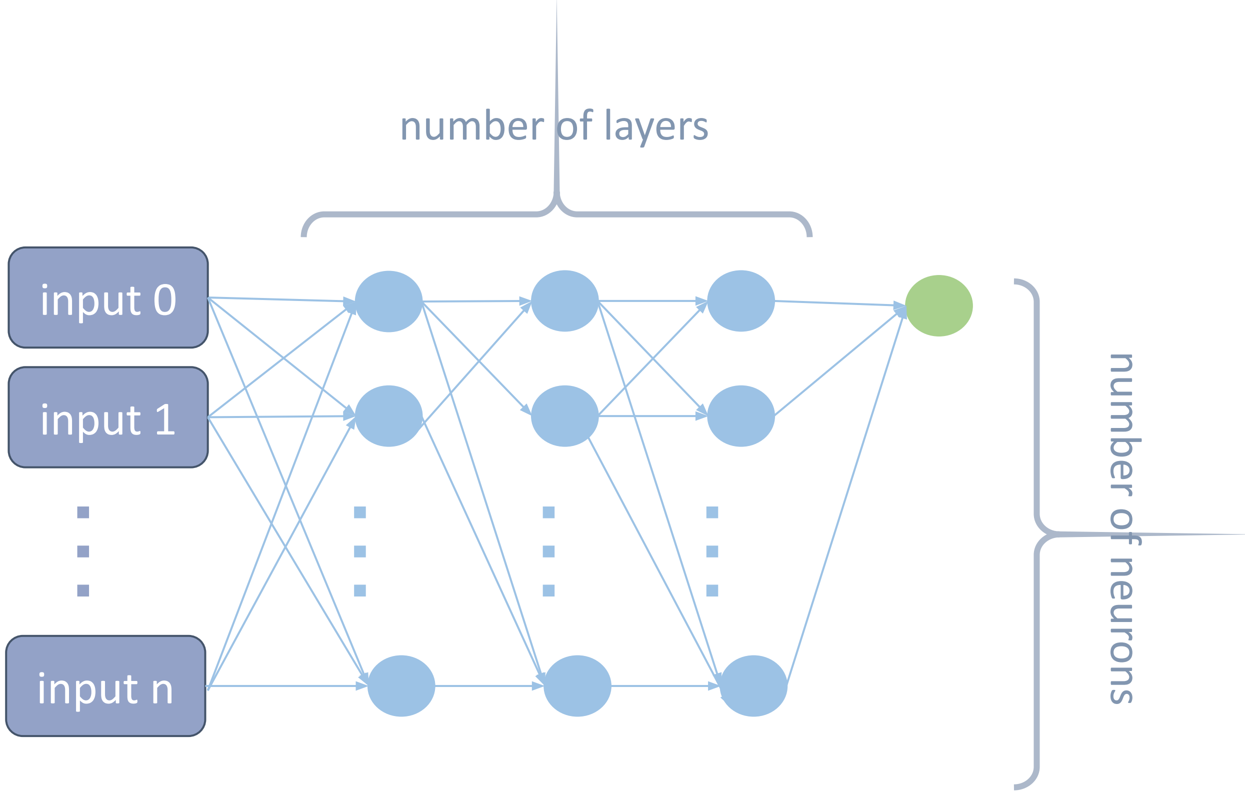
Internal parameter: w1…w6 and b1…b6



Neural network. Parameters neural network structure (hyperparameters)

- Number of layers
- Number of neurons in a layer
- Activation functions:





- Input variables
- Number of training iterations
- The size of the batch of parameters trained at a time
- Training algorithms: Adam, SGD, RMSprop, …

What we want to improve

Problem of classical methods. Why not Keras?

Library such as Keras and other cant hold functions which cannot calculate the derivative directly, due to the implemented algorithm

Implementing Algorithms with Arbitrary Optimization Functions

- The goal is to allow user to choose any function for optimization
- Cross-entropy is now optimized during network training.
- ROC-AUC is an example of optimization metric with clear physical meaning.

Cross-entropy

$$H(P^*|P) = -\sum_i \underbrace{P^*(i)}_{\text{TRUE CLASS DISTRIBUTION}} \log \underbrace{P(i)}_{\text{PREDICTED CLASS DISTRIBUTION}}$$

ROC-AUC

common sense AUC = ∫TPRd(FPR)

physical sense AUC = ∫Background_Rejection d(Signal_Efficiency)

Neural network. Training algorithms

Training algorithms - algorithms for searching parameter values. The algorithm minimizes the measure of difference between the “true” value of the target variable and the value predicted by the neural network.

Training algorithms

Stochastic

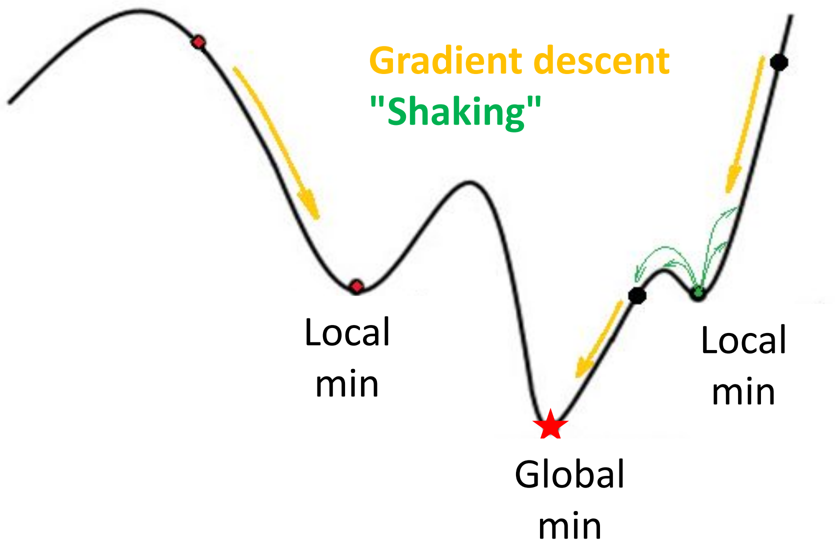
- Random search

Deterministic

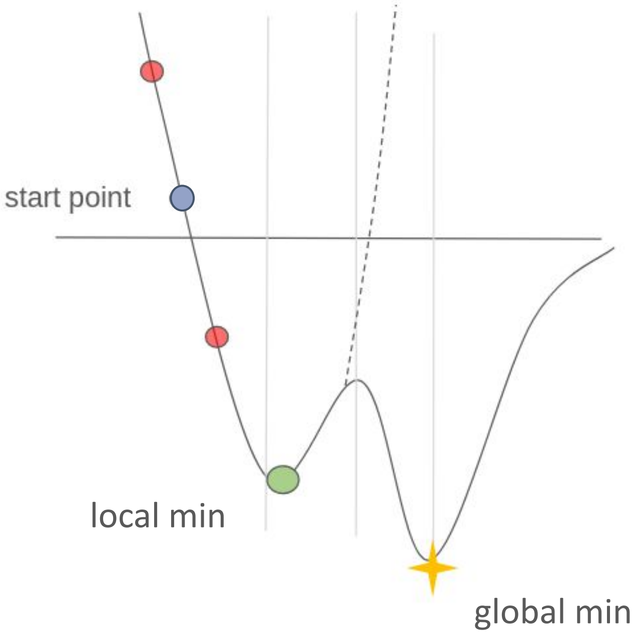
- Gradient Descent

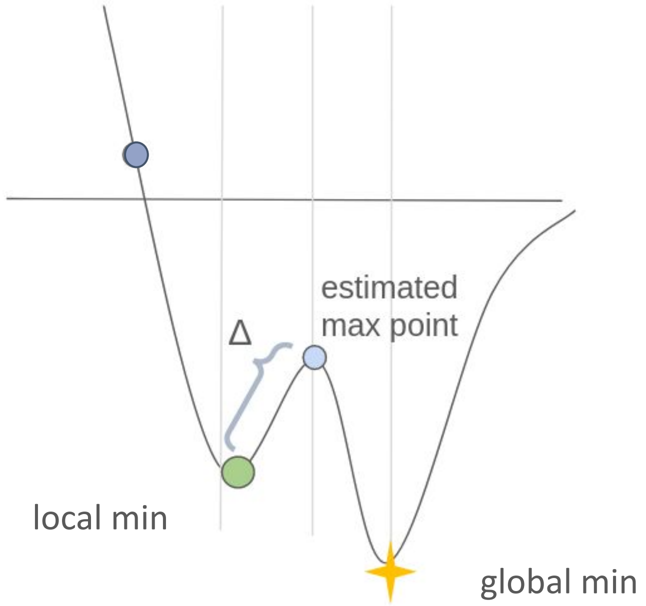
Combined

- Stochastic gradient descent



Practically used algorithms try to combine the advantages of deterministic and stochastic methods. In most problems, a sufficiently **deep local minimum** is a satisfactory solution.





number of shaking

random NN parameters (weights of synapses) → search of local min → shaking → optimized NN parameters (weights of synapses)

search of local min

- determine left and right derivative
- estimate min point by parabola method:
  $f = ax^2 + bx + c$
  $f' = 2ax + b$
  $f'' = 2a$

shaking

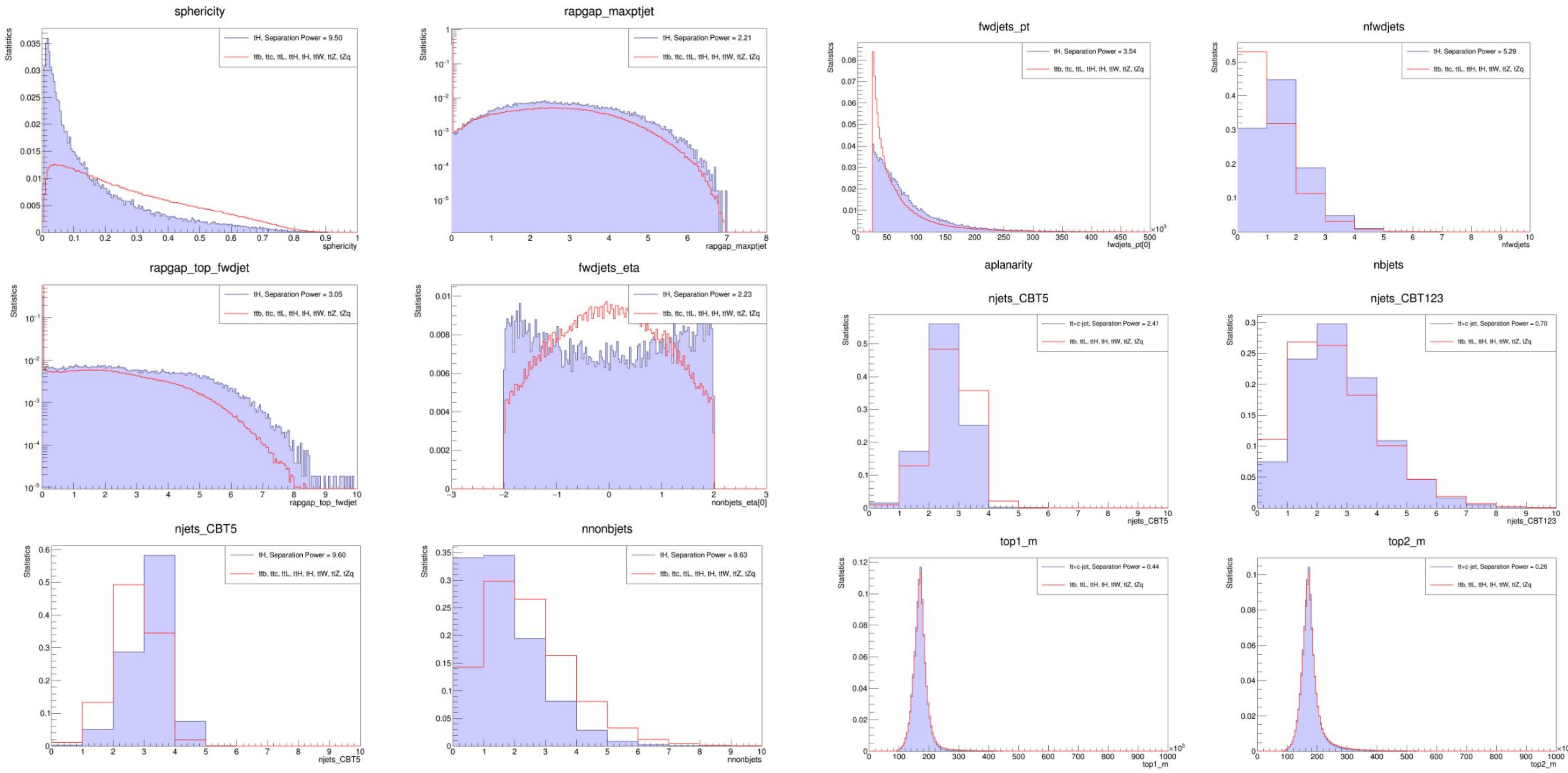
- find estimated max point
- random jump by gaussian distance
  step=Δ*gauss_point

Neural network. Input variables

$$SP = \frac{1}{2}\left(\sum_{i=0}^{nbins}\frac{(s_i-b_i)^2}{s_i+b_i}\right)\times 100$$

	Name	Separation (SM) [%]	for SM	
1	lead_lep_charge	3.32		Charge of the leading lepton
2	N_b	10.71		Number of jets generated by b-quarks
3	n_nonb	2.79	excluded	Number of jets generated by quarks other than the b-quark
4	HT_alljets	1.34		Algebraic Sum of all transverse momenta
5	delta_eta_tH	8.41		Difference between the pseudorapidities of the top quark and the Higgs boson
6	sphresity_alljets	8.85		A measure of the uniformity of jet distribution in space
7	sphresity_lnu4maxje	8.83		A measure of the uniformity of jet distribution in space
8	sphresity_allobjects	8.62		A measure of the uniformity of jet distribution in space
9	aplanarity_allobjects	8.23	excluded	A measure of the deviation of jets from one common plane
10	higgs_m	5.94		Recovered mass of the Higgs boson
11	mass_tH	6.73		Invariant mass of the t-quark and Higgs boson
12	aplanarity_alljets	8.54	excluded	A measure of the deviation of jets from one common plane
13	aplanarity_lnu4maxjet	7.94	excluded	A measure of the deviation of jets from one common plane
14	delta_eta_FWD_t	7.35		Pseudo-rapidity difference between the t-quark and the front jet
15	min_chi	3.84		Quality of Higgs and Top Mass determinations
16	mass_H_CenJet	5.05		Invariant mass of the Higgs boson and the central light jet
17	mass_H_FWD	5.37		Invariant mass of the Higgs boson and the front jet
18	FWD_pt	5.37		Transverse momentum of a quark scattered forward
19	fwmlnujet1	5.51	excluded	First Fox-Wolfram moment composed of jets, lepton and neutrino
20	FWD_eta	4.95	excluded	Pseudofastness of a forward scattered quark
21	fwm1	4.67		First Fox-Wolfram moment composed of jets only
22	top_m	3.52		Recovered t-quark mass
23	fwm2	2.50	excluded	Second Fox-Wolfram moment composed only of jets
24	DeltaR_qqW	2.89		Angle between jets of hadronic decay of w boson
25	RapGap_maxptb	1.93		Difference between the pseudo-velocities of the front jet and the b-jet with the highest pt
26	RapGap_closest	1.79		The difference between the pseudo-velocities of the front jet and the b-jet closest to it
27	Central_non_b_maxpt_pt	1.68		Highest transverse impulse among light jets
28	FWD_m	1.63		Invariant mass of the front jet and t quark
29	lead_lep_eta	2.04	excluded	Pseudofastness of the leading lepton
30	jet_b2_e	1.48		Energy of the b-jet second in transverse momentum
31	W_T_m	1.63		Transverse Mass of all jets
32	InvMass_3Jets	12.24		Invariant mass of three jets

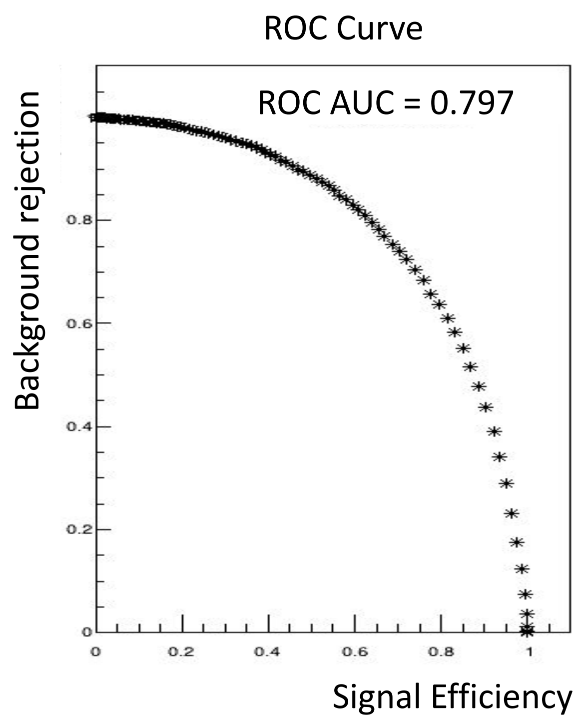
Neural network. Input variables.

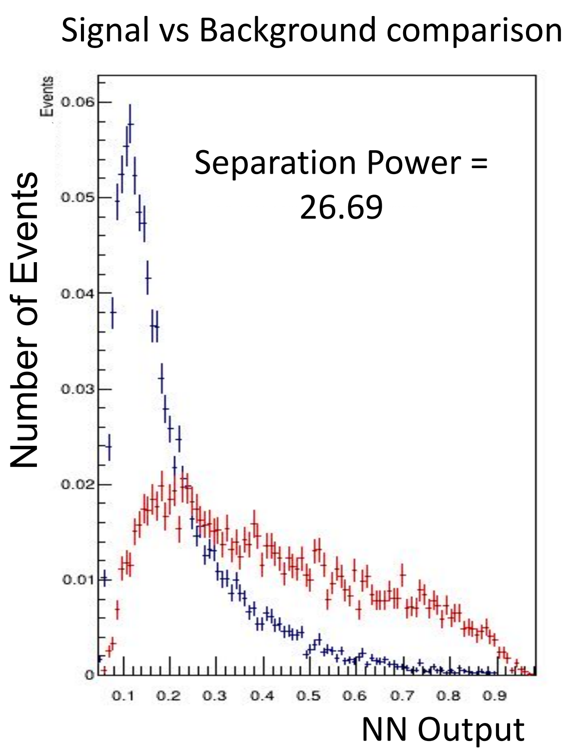


Test of optimization NN parameters.

	config n.1	config n.2
Number of NN training	5	5
Input variables	2	22
Hidden Layers	2	2
Neurons per layer	3,3	3,3
Activation function	Tanh	Tanh
Optimization Function	1-ROCAUC	1-ROCAUC
Number of "shaking"	5	5
Number of batch	not used	not used
Batch size	not used	not used

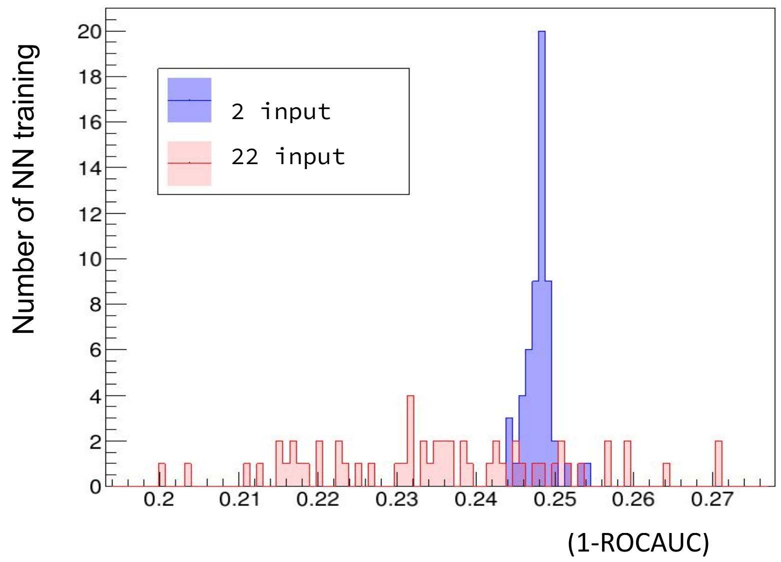
Best results of training





Both configurations was trained 50 times for test.

Problem of converg.
Problem of time



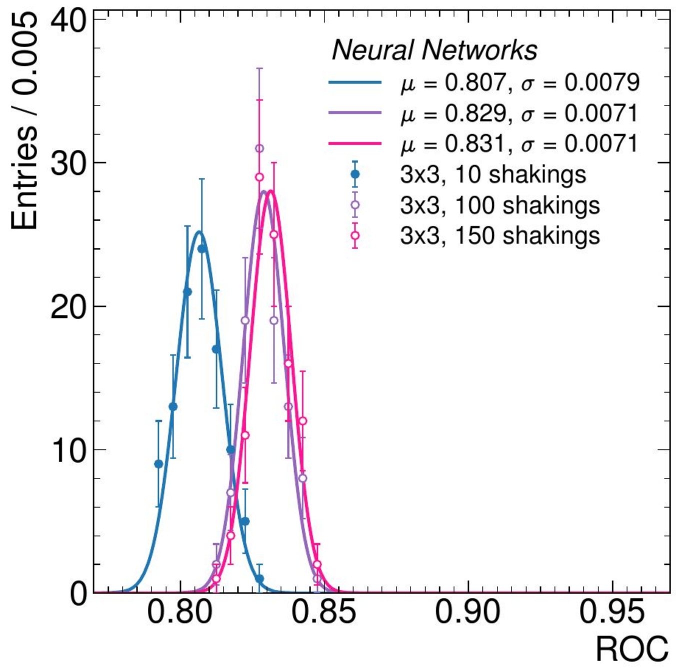
Test of optimization NN parameters

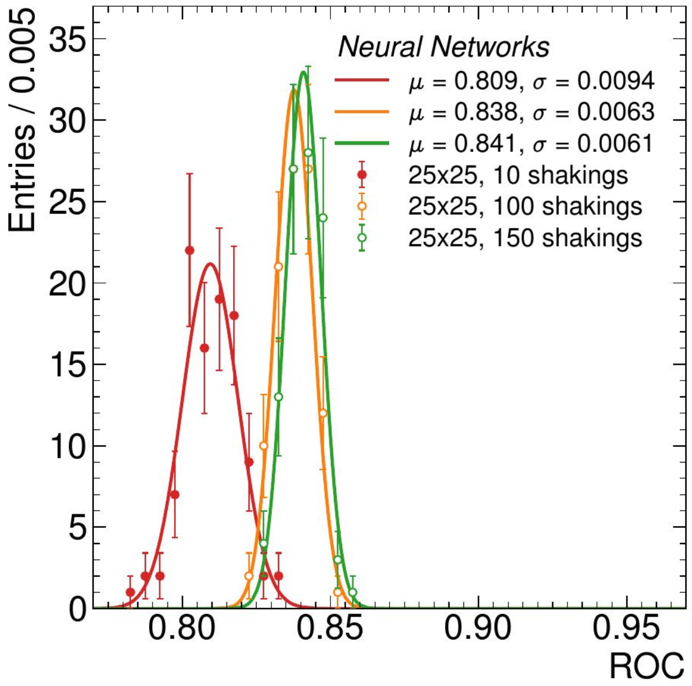
2 hidden layers, 3x3 neurons

Number of configuration	configuration n.1	configuration n.2	configuration n.3
Number of NN training	100	100	100
Input variables	22	22	22
Hidden Layers	2	2	2
Neurons per layer	3, 3	3,3	3,3
Activation function	Tanh	Tanh	Tanh
Optimization Function	1-ROCAUC	1-ROCAUC	1-ROCAUC
Number of shaking	10	100	150

2 hidden layers, 25x25 neurons

Number of configuration	configuration n.1	configuration n.2	configuration n.3
Number of NN training	100	100	100
Input variables	22	22	22
Hidden Layers	2	2	2
Neurons per layer	25, 25	25, 25	25, 25
Activation function	Tanh	Tanh	Tanh
Optimization Function	1-ROCAUC	1-ROCAUC	1-ROCAUC
Number of shaking	10	100	150

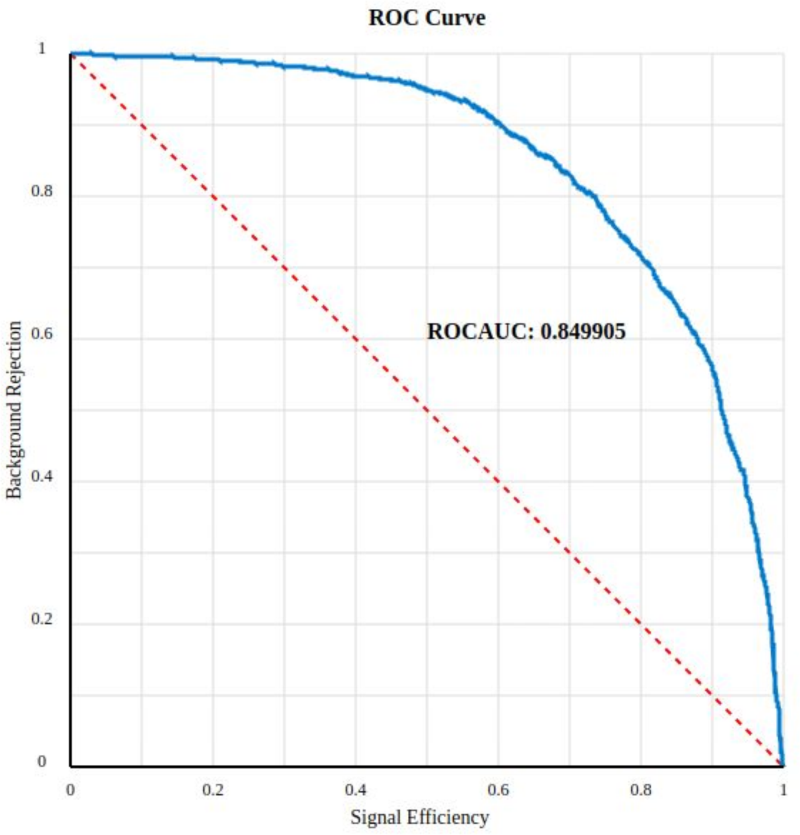


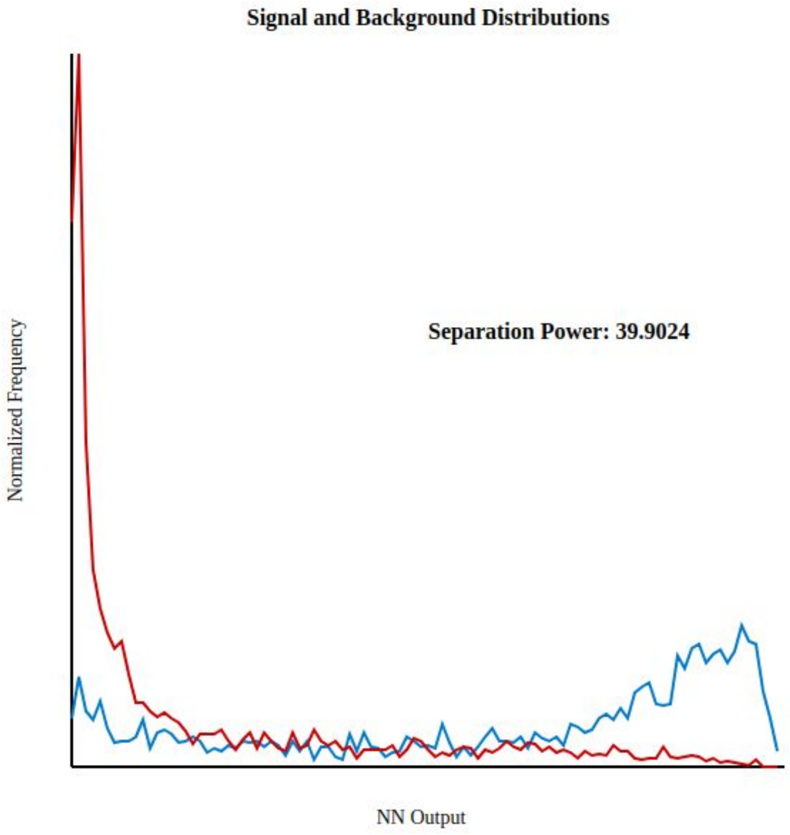


GRID Search space	
Number of (hidden) layers	2
Number of neurons per layers	3, 25
Number of batch	100
Number of shaking (epoches)	10, 100, 150

$$SP = \frac{1}{2}\left(\sum_{i=0}^{nbins}\frac{(s_i-b_i)^2}{s_i+b_i}\right)\times 100$$

The network receives 22 variables as input and outputs 1 variable (network response), which accumulates the differences between the signal and the background contained in all 22 input variables**.**





The optimal ROC-curve integral **0.85**

The optimal Separation Power 39.9%

Conclusion

Results

- New method of NN training has been developed from scratch
- Currently ROC integral is optimized as loss function instead of cross-entropy. Any other physical parameter can be used for optimization
- Current result AUC = 0.85 is comparable with standard NN algorithm. Further improvement is expected after optimization of hyperparameters

Future Plans

- Implement automatic optimization of hyperparameters
- Implement signal significance and separation power as optimization function

Thank you for your attention

Contact: kiseevavi@jinr.ru

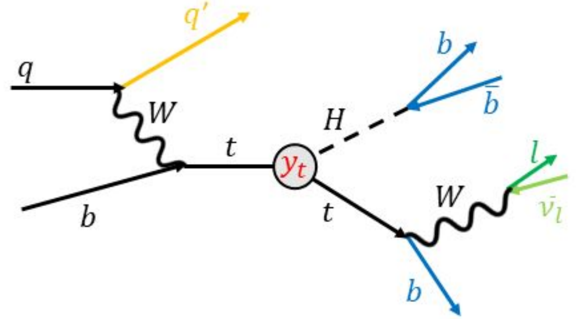
Further optimization of implemented method

Steps to Improve Neural Network Robustness and Effectiveness

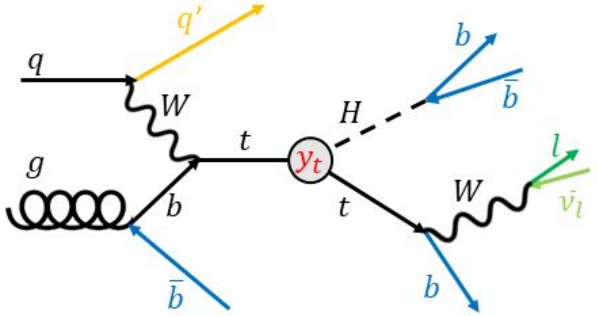
1. Develop a Custom Library (without using of Keras, TensorFlow etc) (done)
   - Create a specialized library to build neural network structures from the ground up.
1. Create a New Training Algorithm (currently in progress)
   - Design an algorithm for training neural networks.
1. Make the Algorithm Optimization-Ready (currently in progress)
   - Ensure the algorithm easily supports optimizing neural network parameters/hyperparameters and allows for any user-defined optimization function.
1. Start with Parameter Optimization
   - Begin by optimizing the neural network parameters (currently in progress).
1. Expand to Hyperparameter Optimization
   - Once performance is good, include the optimization of hyperparameters.

Kinematics of signal and backgrounds

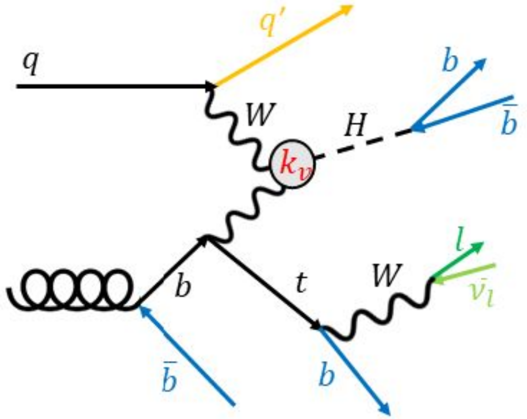
- The kinematics of signal and background processes are very close



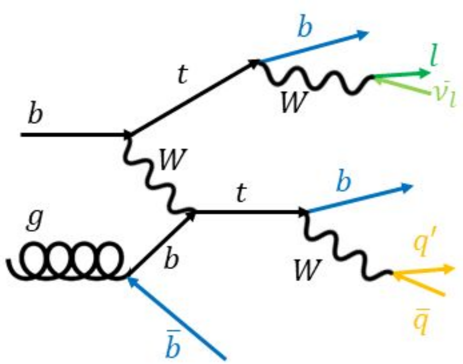
(a) t-channel of the signal tH process



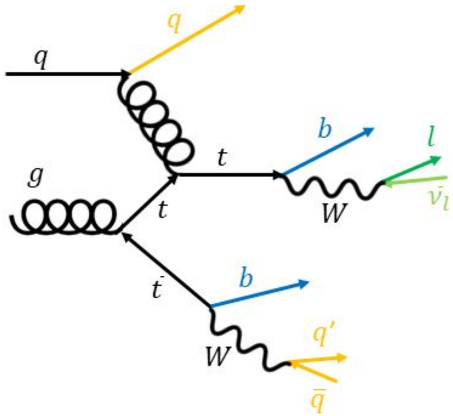
(b) s-channel of the signal tH process



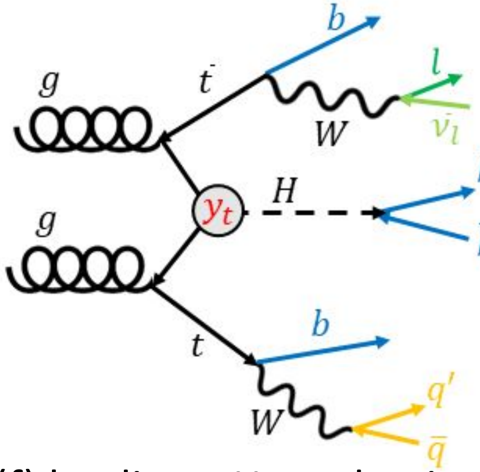
(c) Higgs boson production channel with top quark, where the Higgs boson interacts with the W boson



(d) leading ttb production channel



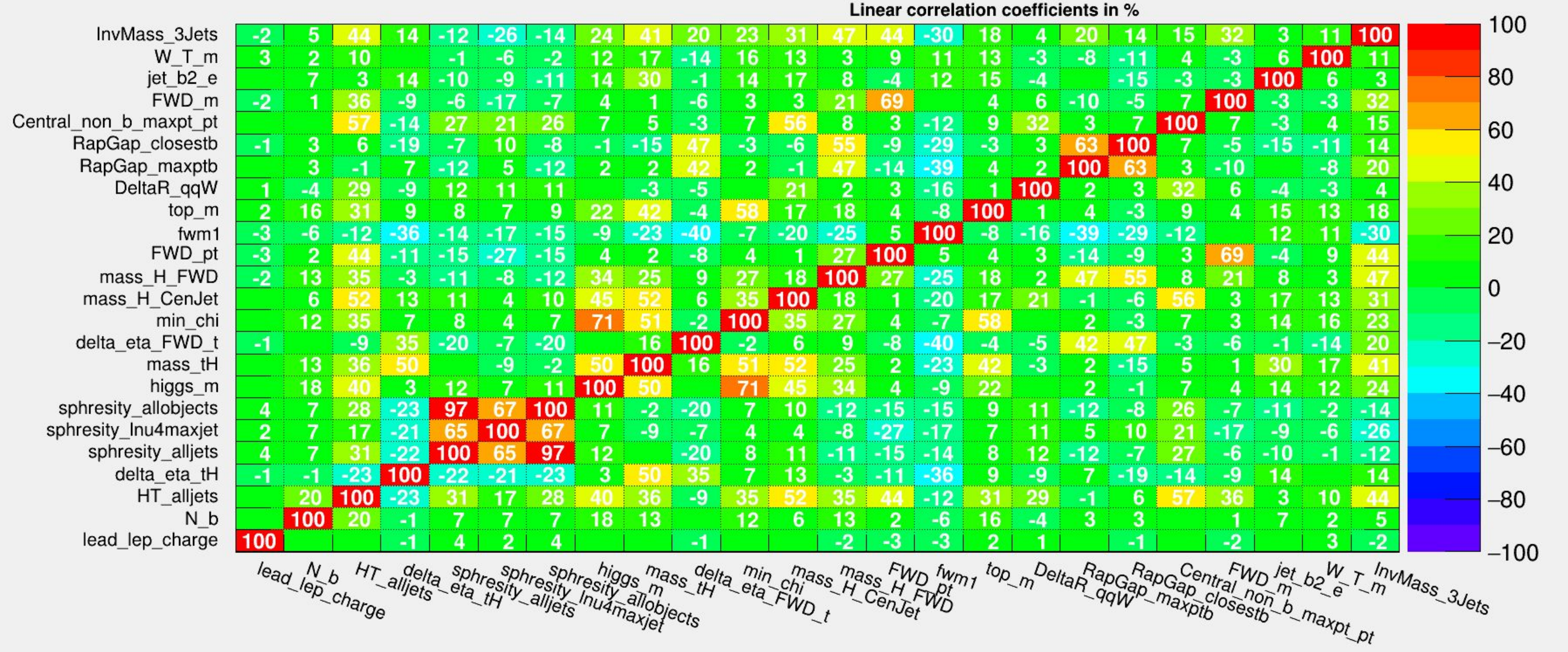
(e) leading ttc and ttL production channels



(f) leading ttH production channel

Back-up. Correlation matrix

24 Variables Most Sensitive to $tHbq_{SM}$



Back-up. ROC-curve integral

The network efficiency curve is the dependence of the cut signal on the number of background events captured.