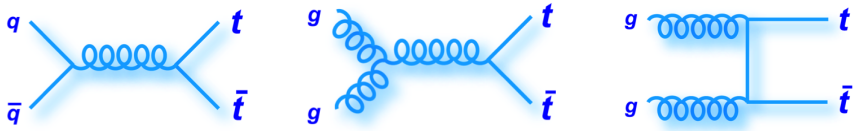
Unfolding technique for the ttbar $e-\mu$ analysis via Machine Learning (deep neural network)

Unfolding the leading jet pt distribution; supervised by Thorsten Kuhl & Yichen Li at DESY Zeuthen

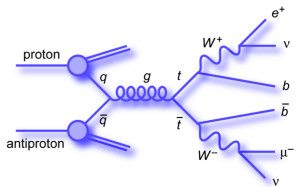
Luiza Adelina Ciucu

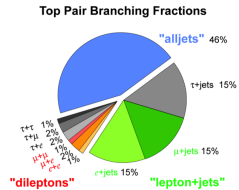
Zeuthen, 02.09.2019

Measuring the top-quark properties is key to test the validity of the SM. The LHC is a top-quark factory.



- The top quark is the heaviest elementary particle $\rightarrow$ largest Yukawa coupling.

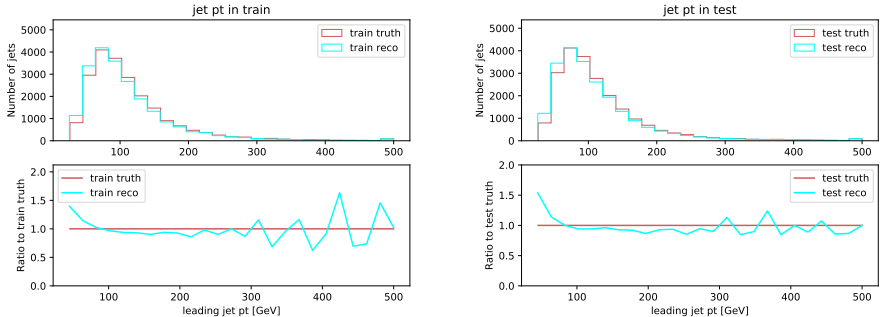




- Our analysis is top-quark pair-production in $e - \mu$ (2% of ttbar).

Theoretical simulation vs. experimental analysis

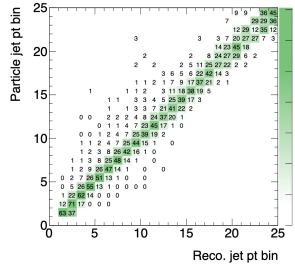
- Theoretical simulation: generated (truth) are known → observed (reco).
- Experimental analysis: observed (reco) are known → truth (nature)



- The shapes of the distributions of the truth and reco observables for the training of the unfolding are very similar, but not identical (especially in the first bins).

The 2D histogram of the migration matrix

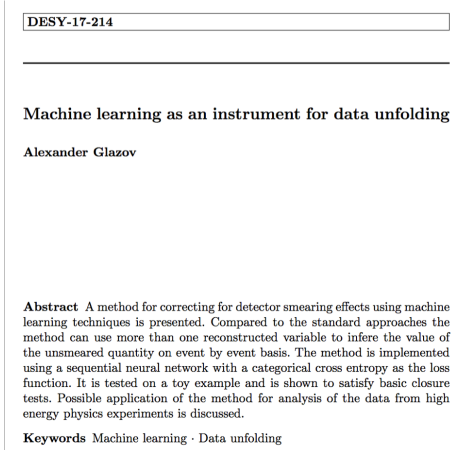
- The migration matrix between truth and reco is used in the traditional unfolding.
- Unfolding = inferring the generated (truth) from the observed (reconstructed).



- Plot from the same data, from the report of Yichen Li. The number in a cell represents the probability of an event in the truth bin i to be reconstructed in a reco bin j.
- The more diagonalizable matrix, the easier unfolding will be.
- Events are migrating from one truth bin to other bins for reco, and vice-versa.

A new approach: use machine learning to learn how to unfold observed (reco) to true (generated).

- arXiv:1712:01814



DESY-17-214

Machine learning as an instrument for data unfolding

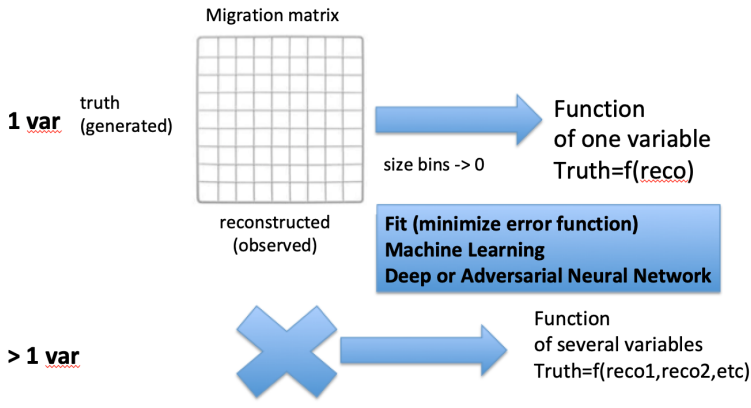
Alexander Glazov

Abstract A method for correcting for detector smearing effects using machine learning techniques is presented. Compared to the standard approaches the method can use more than one reconstructed variable to infere the value of the unsmeared quantity on event by event basis. The method is implemented using a sequential neural network with a categorical cross entropy as the loss function. It is tested on a toy example and is shown to satisfy basic closure tests. Possible application of the method for analysis of the data from high energy physics experiments is discussed.

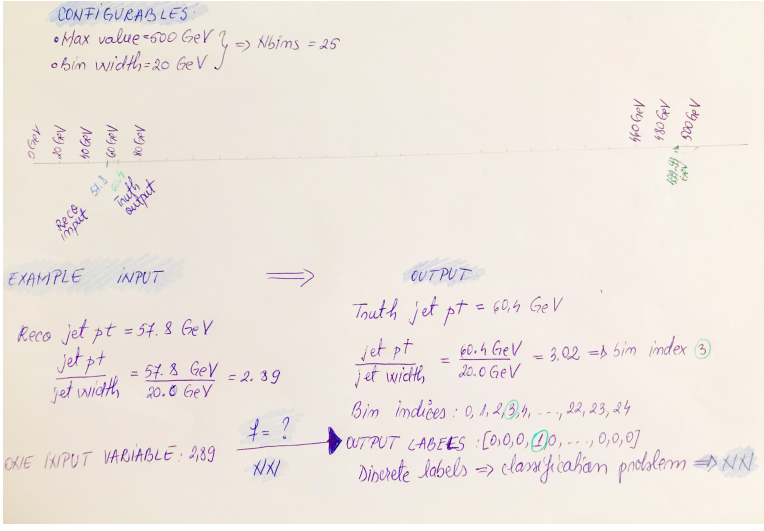
Keywords Machine learning · Data unfolding

Two unfolding methods: traditional and ML.

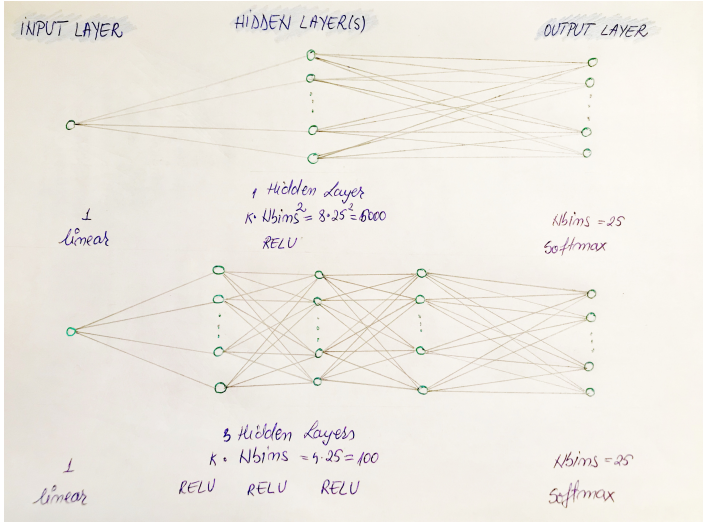
- Traditional: binned migration matrix with only one variable.
- Machine learning: a continuous function of several variables.



Physics problem: index of truth jet pt = f (reco jet pt) = ?



The solution is using these NN architectures.



The NN is implemented in TensorFlow, via Keras, in Python.

- Ran the DNN software for unfolding by DESY Hamburg.
- Ran on lxplus using the ML software docker image via singularity.
- Read the ROOT file via the uproot package.
- Trained the NNs in Python using Keras and TensorFlow.

- Fine tuned the NN hyper parameters for our ROOT data sample.
- One NN training is the first step of the iterative unfolding procedure.

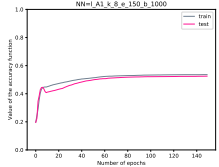
Summary of the physics and NN choices

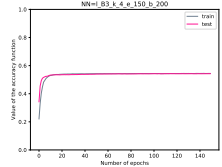
- One file with 43076 events (and leading jets). Half (21538) in training, half in testing.
- jet pt: range=0-500 GeV, bin width=20 GeV, number of bins = 500/20 = 25.
- Chose the NN training hyper-parameters by changing one at a time while keeping the other constant, and choosing those with largest accuracy and smallest loss values.

Choice	Old NN (toy data example)	New NN (my best choice)
number of nodes per input layer	1	1
number of nodes per output layer	Nbins $= 25$	Nbins $= 25$
Number of hidden layers	1	3
k	8	4
Number of nodes per layer	k $\cdot$ Nbins$^2 = 5000$	k $\cdot$ Nbins $= 100$
activation function input	linear	linear
activation function hidden	ReLU	ReLU
activation function output	softmax	softmax
batch size	1000	200
number of epochs	150	150

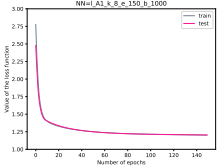
Two figures of merit to optimise the NN hyper-parameters

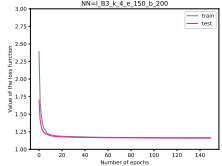
- The accuracy: the larger, the better (left: old NN; right: new NN).





- The loss (error) value: the smaller, the better (left: old NN; right: new NN).

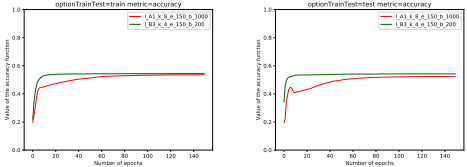




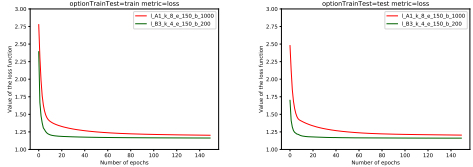
- Overlaid train and test are very similar $\rightarrow$ we did not overtrain the NNs.

Overlaying the two NN architectures: old and new.

- The accuracy: the larger, the better (left: train; right: test).



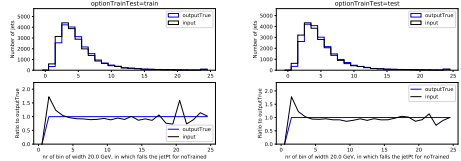
- The loss (error) value: the smaller, the better (left: train; right: test).



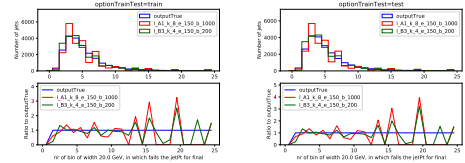
- The new NN architecture outperforms the old one.

The jet pt distribution as the index of the jet pt bin.

- True output (generated, truth) vs input (reconstructed). Used in traditional unfolding.

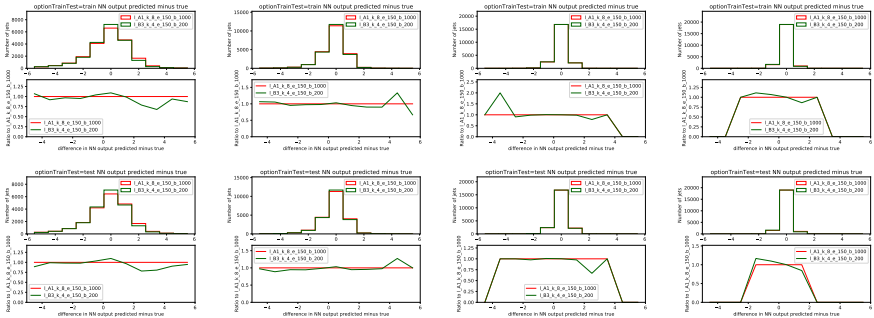


- The true output vs the predicted unfolded output by each NN architecture.
- The new NN architecture is closer to the true output than the old one.



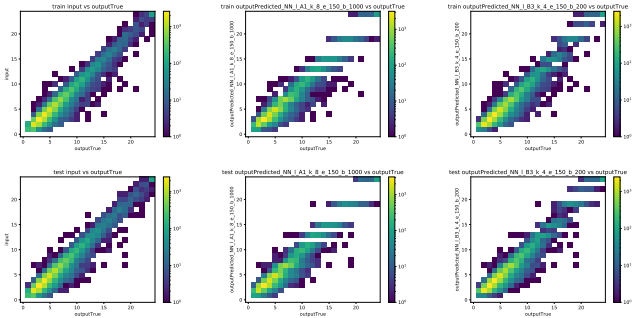
The predicted NN output minus the true output.

- Both outputs are integers, representing bin indices, $\rightarrow$ difference = also integer.
- The greater the count at zero difference, the better.
- Top: train, bottom: test. Left to right, bin widths of 10, 20, 50, 100 GeV.
- The smaller the binWidth, the better is the new NN architecture relative to the old one.



The 2D histogram migration matrix of the index of the jet pt

- The closer to a diagonal matrix, the better.
- Top: train, Bottom: test.
- Left to right: input, old NN output, new NN output vs true output.

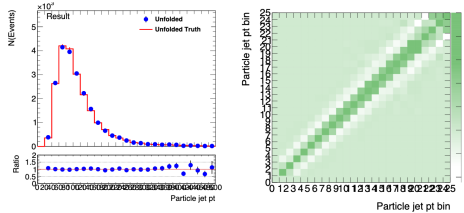


Traditional unfolding results

- Plot taken from Fig. 5 of study by Yichen Li.
- Traditional unfolding: iterative bayesian unfolding with 4 iterations.

5 Unfolding result

The distributions of the unfolded observable and the correlation coefficients between bins are shown in Fig. 6. The truth distribution is shown for comparison.



(a) (b)

Figure 5: The (a) distribution of the unfolded observable and (b) correlation coefficients between bins (in percentage). The truth distribution is shown for comparison.

Summary and Next Steps

- Studied Unfolding of the jet pt distributions in ttbar e-mu analysis.
- Bin of truth jet pt = f (reco jet pt) = ?
- Discrete output values $\rightarrow$ classification problem $\rightarrow$ NN.
- Coded using Tensor Flow via Keras in Python.
- Followed code example with toy data in arXiv (link for arXiv:1712:01814).
- Fine tuned NN hyper-parameters and architecture for our jet data.
- The chosen NN outperforms the one from the toy data.
- The project code and report are in GitLab (link).

- Training recursively several NNs of the found architecture.
  - One NN training is just the first (zeroth) step in the NN unfolding method.
- Use more data (more events and leading jets).
- Add other input variables (e.g. jet η).
- Instead of k-fold=2 (train and test at 50%), use a larger k-fold (e.g. 5).

Backup slides