Applications and Opportunities of Machine Learning in HEP

UCJF Seminar 2023

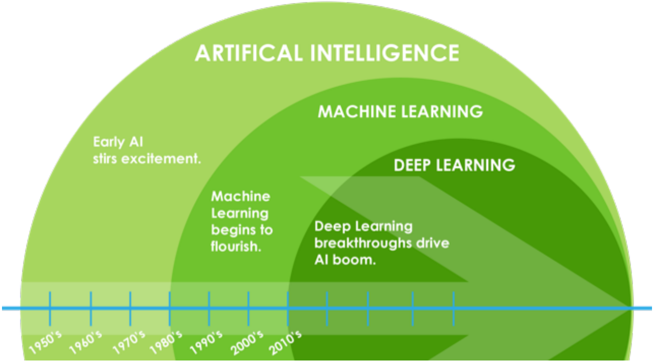
Jiri Franc

April 19, 2023

Czech Technical University in Prague
Faculty of Nuclear Sciences and Physical Engineering

Outline

- Short summary of MVA and ML history in HEP (personal view).
- Current status of MVA and ML in HEP (based on papers and conferences).
- Software used for MVA and ML in HEP (based on conferences and workshops).
- Challenges and future of ML in HEP.
- Some of most used ML methods in HEP.



Difference Between AI, ML and DL [source].

Quick History summary

Machine Learning: Giving computers the ability to learn without explicitly programming them.

1763 Bayes's theorem

1805 Least squares method

1950 Turing's machine

1957 k-means algorithm used in Bells laboratory

1958 Perceptron - an algorithm for pattern recognition

1965 Multilayer perceptrons

1967 Nearest Neighbor

1977 EM algorithm

1995 Support vector machines
Random forest

1999 Gradient boosting decision trees (XGboost 2014)

2009 Big bang: deep-learning neural networks were trained with GPU's

Quick History summary

Last decade applications and models driven by BigTech:

- **DeepMind**: AlphaGo - 2015, AlphaFold - 2020, AlphaCode - 2022
- **Google**: Google Brain - 2011, TensorFlow - 2015, AutoML - 2018,TPU (Tensor Processing Unit) - 2018, BERT (Bidirectional Encoder Representations from Transformers) - 2018
- **Meta**: DeepFace - 2014, Torch - 2015, PyTorch - 2018, DALL-E - 2021, Segment Anything Model (SAM) - 2023
- **OpenAI**: GPT (Generative Pre-trained Transformer) - 2018, GPT-4 (ChatGPT) - 2022

We have Chatbots passing the Turing Test, Self driving cars, Image generators, Github copilot, etc.

History of application of MVA and ML in HEP experiments

- Linear Decision Boundaries and Naive Bayesian classifiers in τ particle identification and studies:
  - MARK III at SLAC (1980s),
  - LEP collaborations ALEPH and OPAL (1990s).
- Artificial Neural Networks in jet identification and tracking at CDF and D0 (1992).
- Boosted Decision trees (BDT) - MiniBooNE, an experiment at Fermilab searching for neutrino oscillations (2005).
- TMVA - Toolkit for Multivariate Data Analysis (2007).
  The "era of hard cuts" was gradually ending.

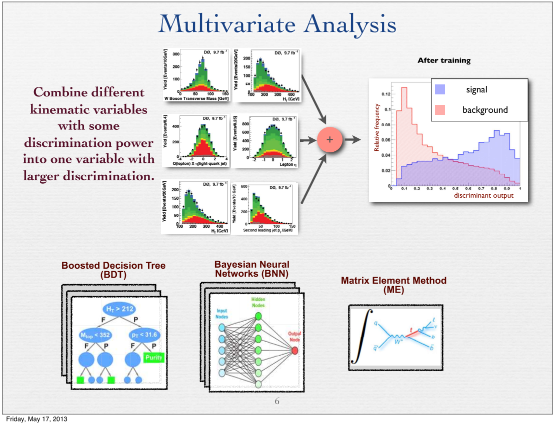
TMVA Toolkit for Multivariate Data Analysis with ROOT [source].

- Personal view: maybe the last period when ML in HEP form state of the art.

History of application of MVA and ML in HEP experiments

- Combination of BDT, BNN and ME - Observation of Single Top-Quark Production (2009).
- 49 input variables -> 3 discriminants -> one final discriminant

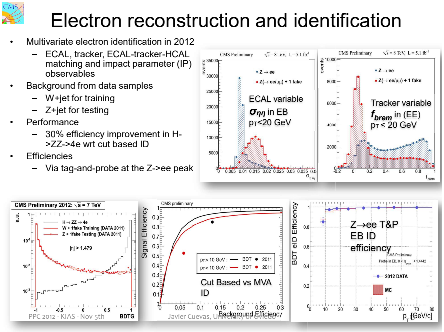
This approach was reused in 2013 measurement again:

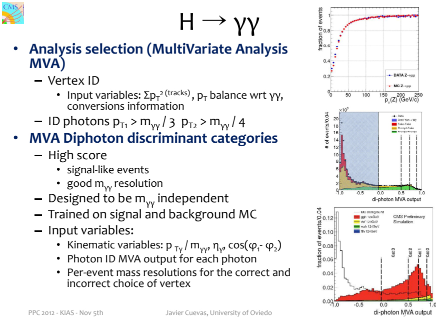


[D0 ADM meeting 17.5.2013]

History of application of MVA and ML in HEP experiments

- Observation of Higgs Boson by **CMS** and ATLAS collaborations (2012).





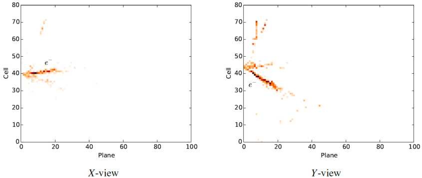
Example of MVA usage by Cuevas in "CMS SM Higgs searches".

End of pre-Higgs boson discovery era - "do it your self".

History of application of MVA and ML in HEP experiments

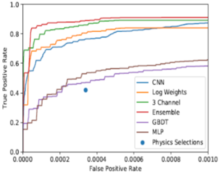
Application of Deep Neural Networks (ten years later than Google)

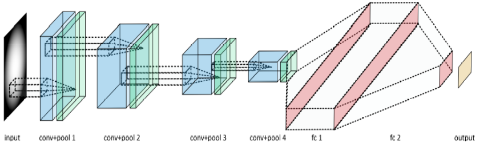
- Identification of neutrino interactions at NOvA by CNN (2016)



(b) ν_e CC interaction.

- Identification of objects or particular particle types on ATLAS/CMS (2017)





- Identification of events in ProtoDUNE by CNN and TensorFlow (2018)

Last years in HEP

DL Methods used in ML by tasks:

- **Monte Carlo simulations:** Generative Adversarial Networks (GAN), Variational AutoEncoders (VAE), Continuous-Time Dynamic Encoder (CTDE).
- **Event classification, Jet identification, etc:** Convolutional Neural Networks (CNNs), Vision transformers (ViT), Graph Neural Networks (GNNs).
- **Signal separation** from tabular data: Boosted Decision Trees (BDT), Deep neural networks, Ensamble methods.
- Triggering systems, Anomaly detection, FPGA, etc.

ML methods are in all areas and are accepted as a standard tool.

Collaborating with other communities

When HEP community found out that state-of-the-art in ML is somewhere else then they starts with explore new research directions and applications of ML, novel algorithms and challenges:

Academic Engagement:
Many conferences connecting HEP and ML, inviting ML experts to HEP workshops, new university lectures, research teams etc.

- Review of Machine Learning for Particle Physics: https://iml-wg.github.io/HEPML-LivingReview/
- book Deep Learning for Particle Physicists: https://lewtun.github.io/dl4phys/intro.html
- book Artificial Intellligence for HEP.
- Inter-Experimental LHC Machine Learning Working Group (since 2016)
- HEPML Resources: https://github.com/iml-wg/HEP-ML-Resources
- Conferences: IML Machine Learning Workshop, PyHEP Workshop, ACAT

Machine Learning Challenges and Collaborative Benchmark Datasets:
Challenges such as the Higgs Boson Challenge (2014) and Track ML Challenge (2018) organized on Kaggle. ML community meets HEP problems and can come with unique solutions.

Collaborating with other Software and Tools

There are following ML analysis approaches in HEP.

- HEP-developed ML toolkits, such as the TMVA in ROOT.
- Externally developed software and frameworks (outsourcing)
- Interfaces from ROOT to Python such as PyROOT, Scikit-HEP

There are many programming languages, but two are crucial

- Particle physics has been reliant on C and C++ over the past decades.
- ML community uses many languages, but Python-based ecosystem dominates.

You can try and compare these approaches at SWAN (Service for Web based ANalysis) - platform to perform interactive data analysis in the cloud.
https://swan.web.cern.ch/content/basic-examples

Question:
What aspects of ML development should the HEP community focus on in the next years?
Aspects to consider: data formats, community size, programming language, and interfaces.

Status of MVA and ML in HEP couple of year ago

Layer	HEP (Circa 2013)	HEP (Circa 2018)	Industry
Data Layer	ROOT Files	ROOT Files	DB / HDFS etc.
Loading Layer	Ad hoc ROOT ETL logic	Numpy / HDF5 Converters / Loaders	Numpy / HDF5 Converters / Loaders
Training Layer	TMVA	Keras, TensorFlow, PyTorch, XGBoost, scikit-learn, ...	Keras, TensorFlow, PyTorch, XGBoost, scikit-learn, ...
Serving Layer	Deployment Target (TMVA)	Deployment Target (lwtnn, TensorFlow, TMVA wrappers)	Deployment Target (TensorFlow Serving, SageMaker, etc.)

source: Luke de Oliveira talk at IML2

Experiment Management: Make people training ML models more productive.
Universal Serving Layer: Make people using ML models more productive.

ML models

To use ML we need to implement a **mathematical model** which depends on our **task** and our **data set**.

The most used task and regarding models in HEP are:

- Regression, Classification (tabular): LM, GLM, BDT, XgBoost, NN, Ensamble.
- Classification (visual): CNN, ViT.
- Dimensionality reduction: SVD (PCA).
- Detector Simulation: VAE, GAN.

Tasks can be categorized by type:

- Supervized (needs Learning, Testing, Validation, Prediction, Inference): Classification, Regression.
- Unsupervized: (data without known structure, labels , etc.): Clustering, Dimensionality reduction.

ML models - Classification

As mentioned: Classification is the most common task with ML applications in HEP analysis.

- Binary classification: most methods - LR, SVM, BNN, NN, CNN
- Multi-class classification: few methods - transformation to or extension from binary.

Binary classification is the most common ML problem and if you want to learn ML, you should start here!

That means not only understand how different methods works, but primarily understand:

- How to prepare and validate training-testing-validation samples.
- What is ROC curve, FPR - Background efficiency, TPR - Signal efficiency.
- Simple cross-validation, k-fold cross-validation, Bootstrap.
- Overtraining - Overfitting.

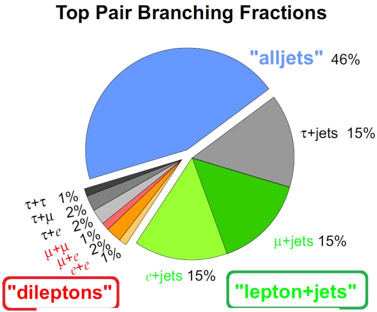
Tabular Data - BDT Approach

D∅ experiment - Top Quark measurement and analysis

Classical approach: create tabular data set from measured events.

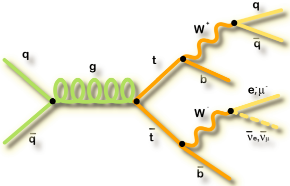
- The goal is to distinguish between signal and background.
- We have first cut based selection.
- We compute many features.
- We train ML method with these features and splited data set.

Strong interaction: Top pair production

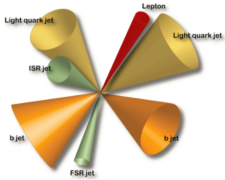


Top Quark at Tevatron:

- **Mass:** $m_t = 174.34 \pm 0.64 GeV$
- **Lifetime:** $t \approx 5 \times 10^{-25} s << \Gamma_{QCD}$
- **Production:**
  $\approx 85\%$ by $q\bar{q}$ annihilation
  $\approx 15\%$ by gg fusion
- **Top decay:** $\mathrm{BR}(t \to W + b) \approx 100\%$

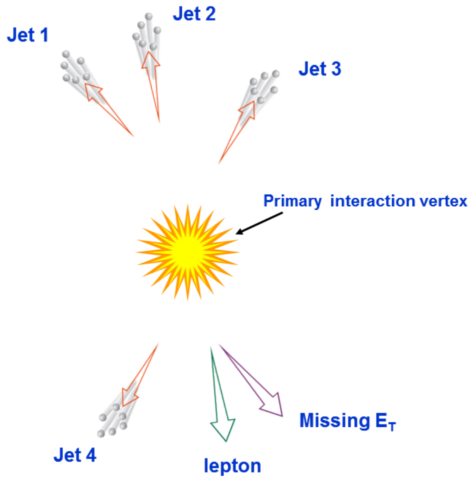


Situation in detector (+ missing transverse energy)

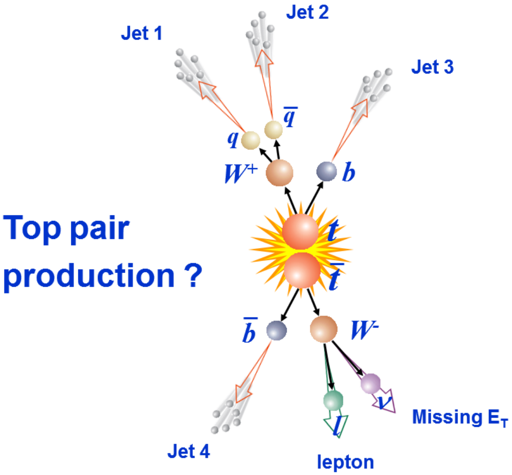


Samples are classified according to W-decay: $\ell +$ jets and $\ell\ell$ channels are under concern and **full dataset 9.7fb^{-1}** is used.

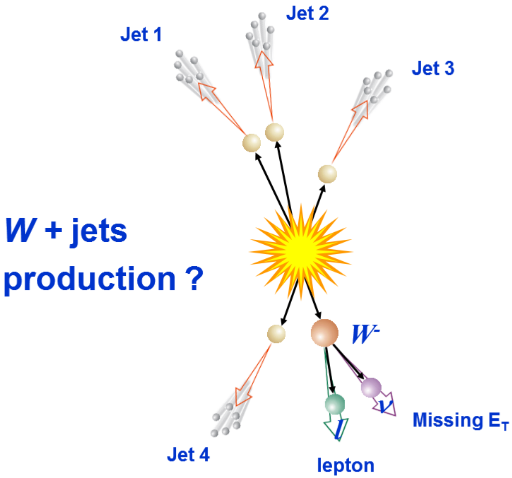
Primary interaction vertex in Top pair production



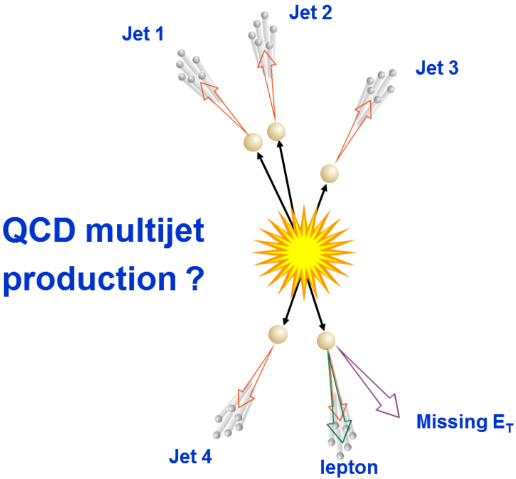
Primary interaction vertex in Top pair production



Primary interaction vertex in Top pair production

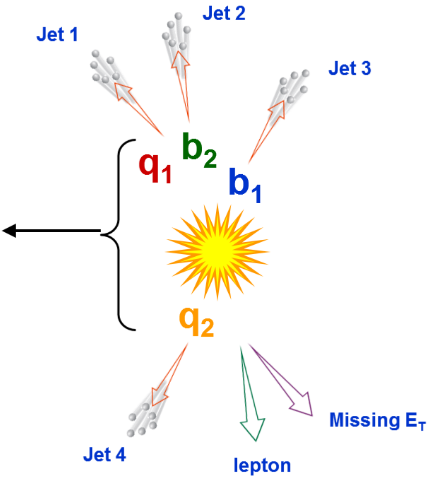


Primary interaction vertex in Top pair production



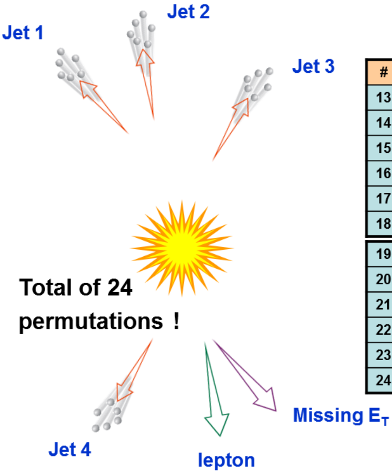
Primary interaction vertex in Top pair production

#	J1	J2	J3	J4
1	q_1	q_2	b_1	b_2
2	q_1	q_2	b_2	b_1
3	q_1	b_1	q_2	b_2
4	q_1	b_1	b_2	q_2
5	q_1	b_2	q_2	b_1
6	q_1	b_2	b_1	q_2



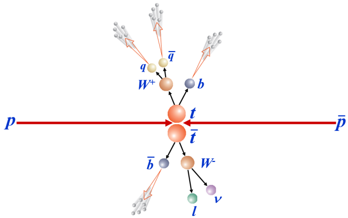
Primary interaction vertex in Top pair production

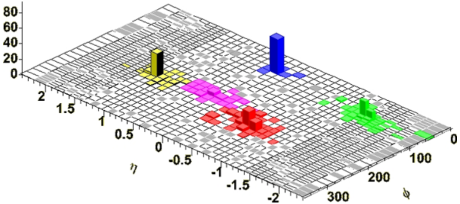
#	J1	J2	J3	J4
1	q_1	q_2	b_1	b_2
2	q_1	q_2	b_2	b_1
3	q_1	b_1	q_2	b_2
4	q_1	b_1	b_2	q_2
5	q_1	b_2	q_2	b_1
6	q_1	b_2	b_1	q_2
7	q_2	q_1	b_1	b_2
8	q_2	q_1	b_2	b_1
9	q_2	b_1	q_1	b_2
10	q_2	b_1	b_2	q_1
11	q_2	b_2	q_1	b_1
12	q_2	b_2	b_1	q_1



#	J1	J2	J3	J4
13	b_1	q_1	q_2	b_2
14	b_1	q_1	b_2	q_2
15	b_1	q_2	q_1	b_2
16	b_1	q_2	b_2	q_1
17	b_1	b_2	q_1	q_2
18	b_1	b_2	q_2	q_1
19	b_2	q_1	q_2	b_1
20	b_2	q_1	b_1	q_2
21	b_2	q_2	q_1	b_1
22	b_2	q_2	b_1	q_1
23	b_2	b_1	q_1	q_2
24	b_2	b_1	q_2	q_1

Selection of $t\bar{t}$ Candidates





Main selection cuts in $\ell +$ jets channel:

variable	kinematic range
$p_T(l)$	$p_T(l) > 20$ GeV
$\eta(e)$	$\lvert\eta(e)\rvert < 1.1$
$\eta(\mu)$	$\lvert\eta(\mu)\rvert < 2.0$
$\not{E}_T$	$\not{E}_T > 20$ GeV
jet $\eta(jet)$	$\lvert\eta(jet)\rvert < 2.5$
jet $p_T(jet)$	$p_T(jet) > 20$ GeV

+ additional cuts

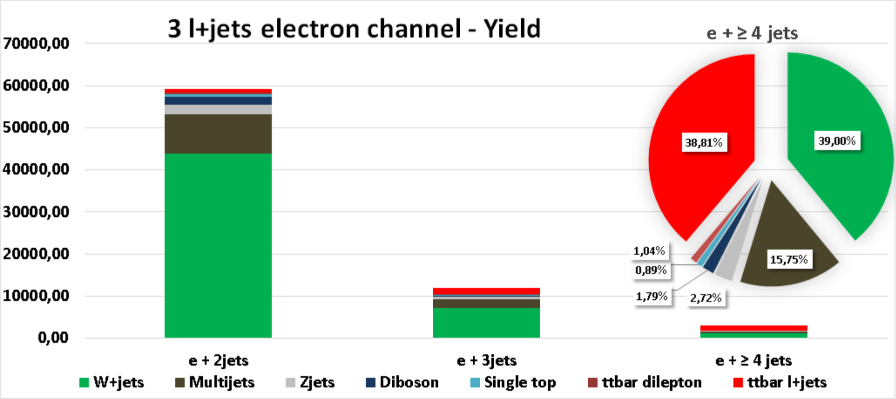
The measurements in both decay channels employ the b-tagging discriminant output distribution as provided by the b-ID MVA.

Data sample: Full Data Set ($9.7 fb^{-1}$) with selection: Phys.Rev.D 90,092006 (2014)

The main goal: Measurement of the inclusive $t\bar{t}$ cross section using MVA and b-ID methods in $\ell +$ jets and $\ell\ell$ channels and compute pole mass.

$\ell +$ jets Yield table:

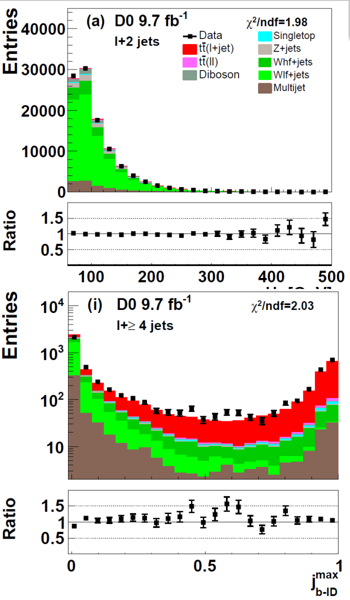
6 analysis channels in $\ell +$ jets :
to the lepton type (electron, muon) and the number of jets (2, 3, $\geq$ 4)



Signal rate in MC(Signal from Alpgen+Pythia):

	2 jets	3 jets	$\geq$ 4 jets
e	1.17%	12.24%	38.81%
μ	0.88%	11.01%	39.01%

Does the MC describe the data?

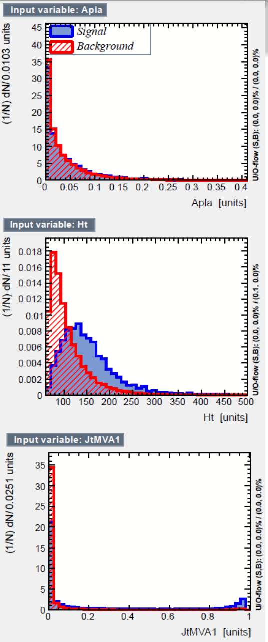


- **Task:** Check the MC and data agreement for all used kinematical and topological variables.
- **Tools:** Control plots and statistical hypothesis testing.
- **Weighted homogeneity tests:** new tests - developed and written by our group

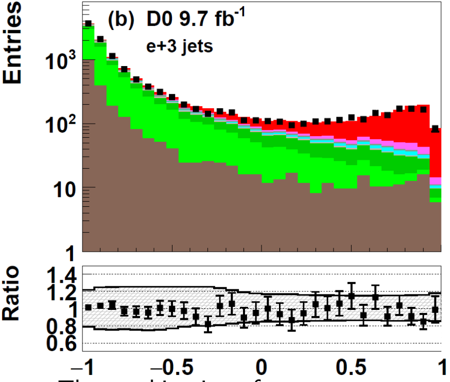
The data are compared to the sum of predicted contributions from signal and background processes, using the theoretical value of $\sigma_{t\bar{t}} = 7.48$ pb and $m_t = 172.5$ GeV.

$\ell +$ jets variables selection

- **Analyzed:** 46 kinematical and topological variables (e.g. H_T, Aplanarity, Sphericity, $M_T(jets)$, lepton p_T, ...) + $j^{\text{lead}}_{\text{b-ID mva}}$.
- **Task:** Select variables with good MC vs. Data agreement and good separation power between signal and background.
- **Tools:** Statistical hypothesis testing (KS test, AD test, χ^2 test, LR test, ...) and TMVA ranking.
- **Selection:** 25 types of good modeled variables with best separation power.
- *b*-**tagging:** MVA *b*-ID output distribution $j^{max}_{b-\text{ID}}$ has been included

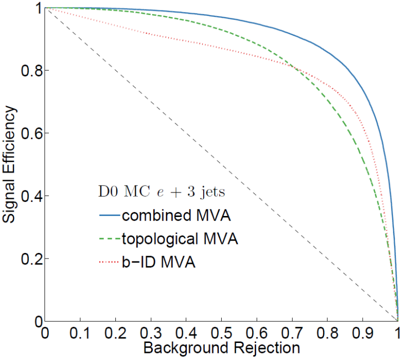


$\ell +$ jets discrimination by TMVA

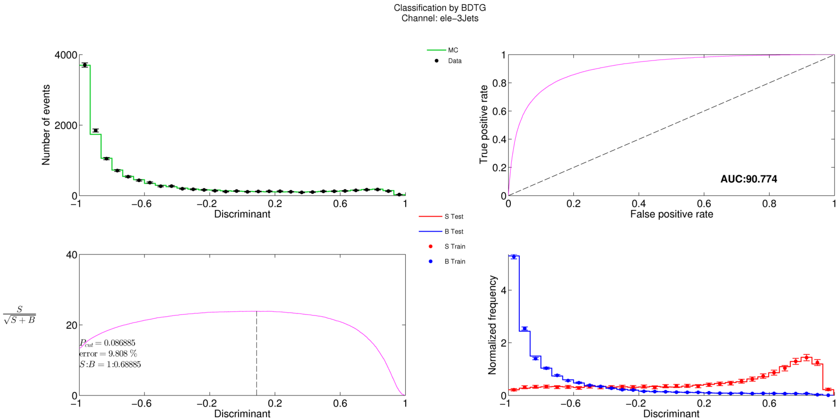


- The combination of pure topological and MVA b-ID method improved the separation by 10%.

- Different MVA methods has been tried.
- TMVA BDTG with gradient boost trained on 25 types of variables + $j^{\mathrm{lead}}_{\text{b-ID mva}}$.
- The area under the ROC curve is around 0.9 for all 6 analysis channels when using TMVA discr.



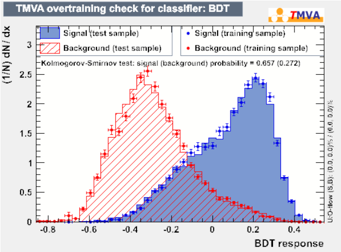
Discrimination by BDTG - optimal cutting:

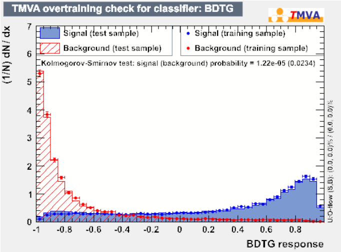


Signal rate in MC - Improvement:

	Before discrimination	After discrimination with optimal cut
Electron 3jb	12.24%	59.24%

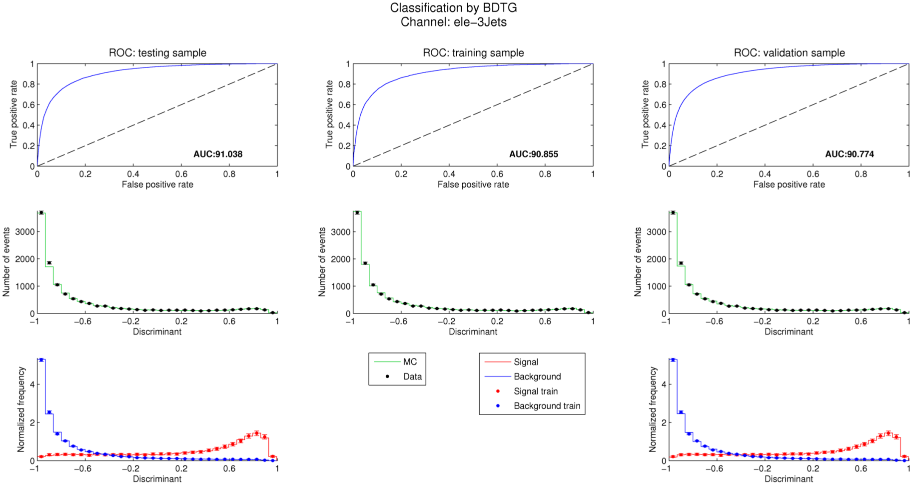
Discrimination by BDTG - output from TMVA:





Each individual background contribution was used in the training and verified that there is no bias due to overtraining of the method.

Discrimination by BDTG - overtraing check:

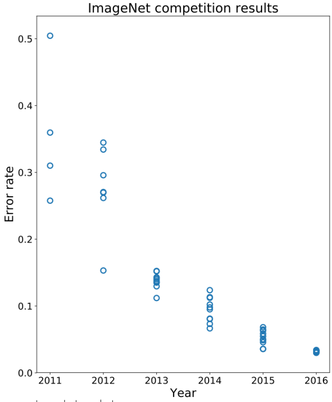


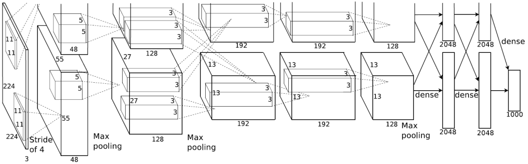
Similar behave in all $\ell +$ jets analysis channels.

Visual Data - CNN/CVN

Short history of CNN

- MLP trained with back-propagation are ideas known since the 1980â€™s.
- LeNet5 and other convolutional networks are used to classify digits since the the 1990â€™s.
- Implementation CNN on a GPU in 2011 and be used in 2012 by winning team in ImageNet Challenge.
(2014 GoogLeNet, 2015 Resnet, 2016 GoogLeNet-v4)



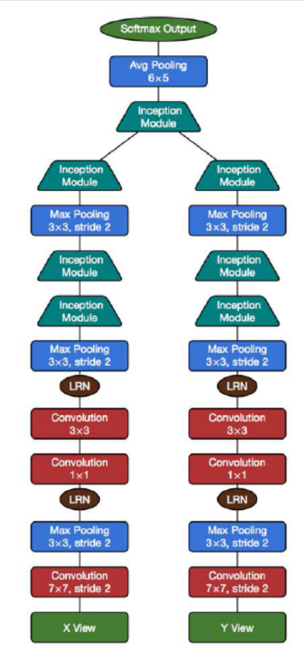


An illustration of the architecture of our CNN used in ImageNet Classification.

Short history of CNN

CNN is about:

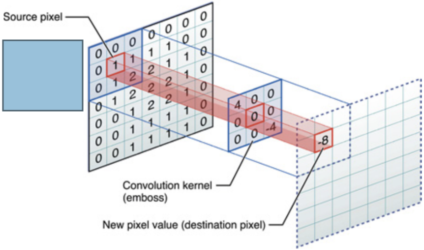
- Various implementation of CNN in image recognition.
- Using relatively little pre-processing compared to other image classification algorithms.
- Content of an input, an output layer, and multiple hidden layers.
- Many different computer vision architectures: AlexNet, ResNet , Inception, EfficientNet, DenseNet
- Convolutional and pooling layers.
- **Transfer learning.**



CNN intro

Image description:

- Grayscale image is a matrix of pixels (picture elements).
- Each pixel stores its brightness (or intensity) ranging from 0 to 255.
- Normalize input pixels: $x_n = \frac{x}{255} - 0.5$
- Dimensions of this matrix are called image resolution (in pixels).
- RGB image has 3 layers (input depth).
- Input and output depth are arbitrary parameters and not equal.



How Convolution Matrix works?

Sharpen

0	0	0	0	0
0	0	-1	0	0
0	-1	5	-1	0
0	0	-1	0	0
0	0	0	0	0

https://docs.gimp.org/en/gimp-filter-convolution-matrix.html

How Convolution Matrix works?

Blur

0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

https://docs.gimp.org/en/gimp-filter-convolution-matrix.html

How Convolution Matrix works?

https://docs.gimp.org/en/gimp-filter-convolution-matrix.html

Edge enhance

0	0	0	0	0
0	0	0	0	0
0	-1	1	0	0
0	0	0	0	0
0	0	0	0	0

How Convolution Matrix works?

Edge detect

0	0	0	0	0
0	0	1	0	0
0	1	-4	1	0
0	0	1	0	0
0	0	0	0	0

https://docs.gimp.org/en/gimp-filter-convolution-matrix.html

How Convolution Matrix works?

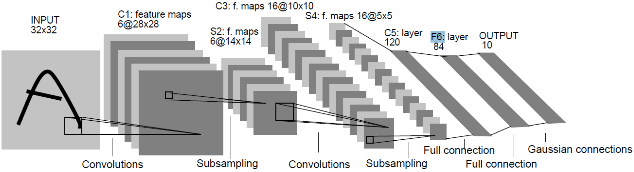
Emboss

0	0	0	0	0
0	-2	-1	0	0
0	-1	1	1	0
0	0	1	2	0
0	0	0	0	0

https://docs.gimp.org/en/gimp-filter-convolution-matrix.html

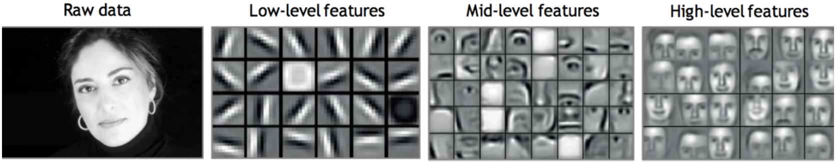
CNN demonstration

Demonstration of LeNet5 CNN applied to document recognition.



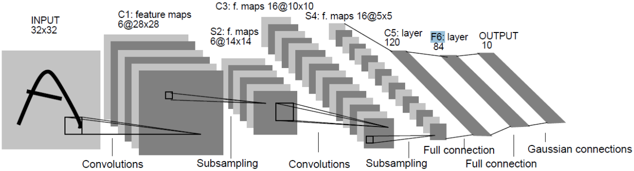
LeCun et al. Gradient Based Learning Applied to Document Recognition.

Illustration of different layers with different inputs.



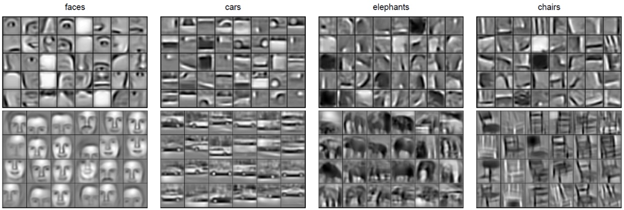
CNN demonstration

Demonstration of LeNet5 CNN applied to document recognition.



LeCun et al. Gradient Based Learning Applied to Document Recognition.

Illustration of different layers with different inputs.

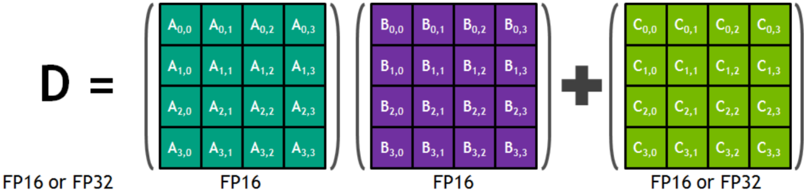


Lee et al. Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations

Why GPU with tensorcores are so powerful with CNN

Tensor Cores are supported for DL training in many frameworks (including Tensorflow, PyTorch, MXNet, and Caffe2).

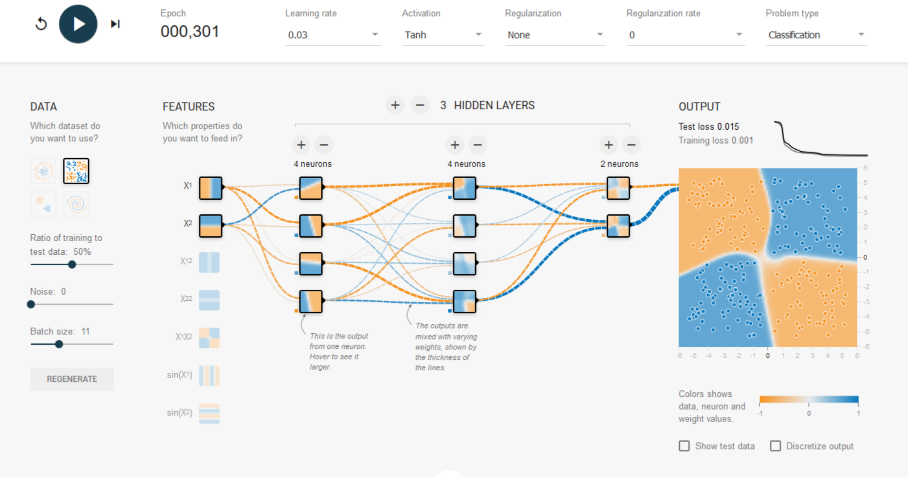
- Tensor Cores are programmable matrix-multiply-and-accumulate units.
- Tensor cores provide a huge boost to convolutions and matrix operations.
- Each Tensor Core provides a 4x4x4 matrix processing array which performs the operation D=A*B+C.



https://devblogs.nvidia.com/programming-tensor-cores-cuda-9/

CNN and TF playground

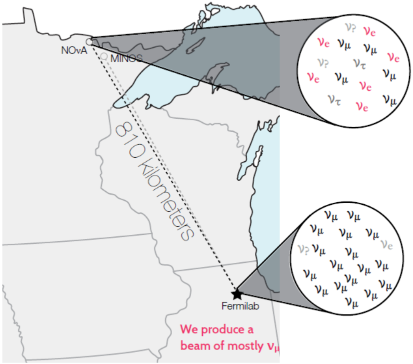
If you want to play with Tensorflow and NN

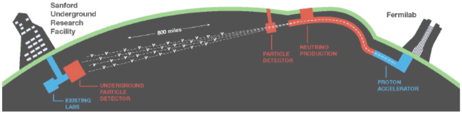


http://playground.tensorflow.org

Neutrino oscillations experiments in FNAL

DEEP UNDERGROUND NEUTRINO EXPERIMENT

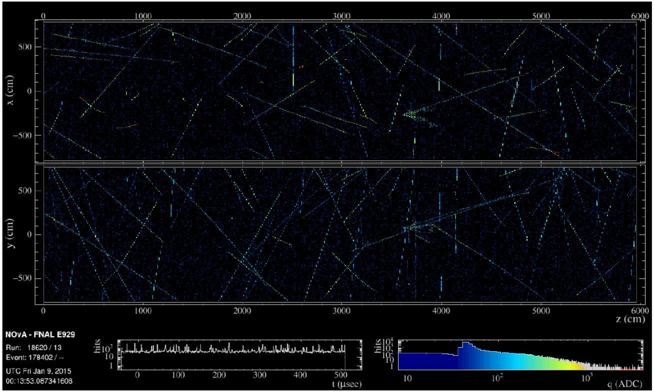




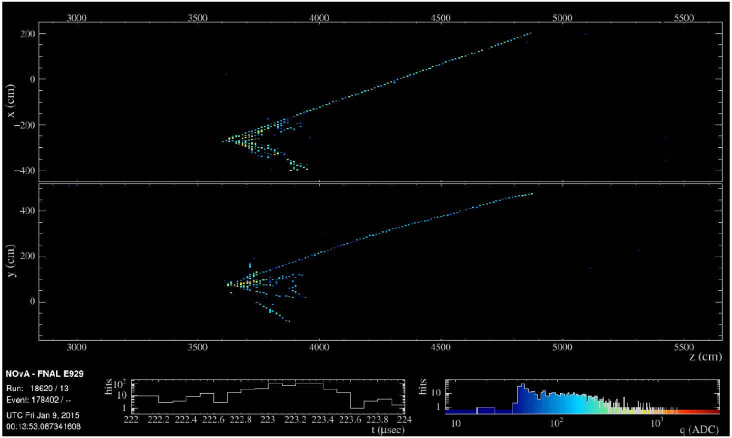
NOvA Experiment

Far detector view.

- NOvA takes data in 550 ms increments
- Due to location on surface 100,000s cosmic rays/second
- 100s of ν_μ/year
- 10s of ν_e/year



NOvA Experiment

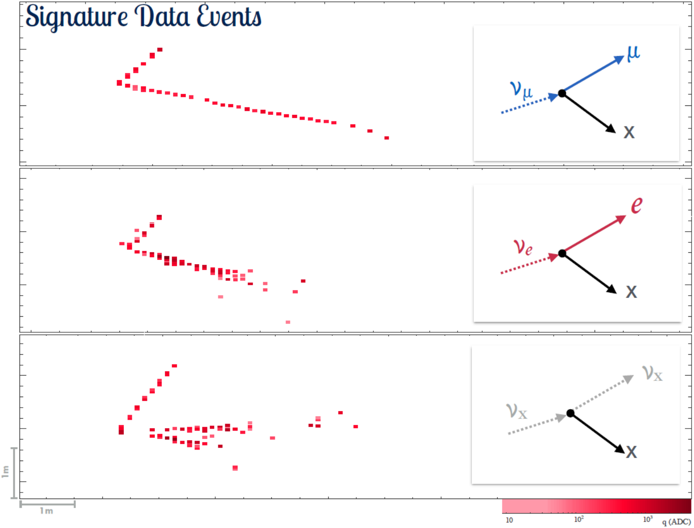


Key challenges with an Event

- **Discriminating between μ and charged pions.**
  Both can produce long tracks, but muons are usually longer and interact less with nuclei
- **Discriminating between e and photons**
  Electrons start showering immediately, but photons travel a short distance before showering (neutral pions decay into photons).

Application of ML in HEP - NOVA

Use images of events to train a CNNs to identify neutrino flavor.



Source: F. Psihas - IML Workshop.

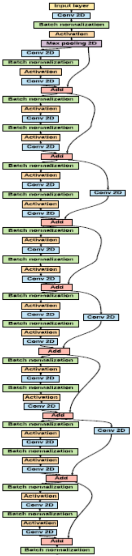
Application of ML in HEP - NOVA

Detector, data and approach description.

- Detector is composed of horizontal and vertical planes, we have two input layers.
- The multiclassifier problem results in more output nods in final layer.
- Resulting pixel maps are sparse.
- First architecture based on GoogLeNet and implemented in Caffe.
- Split the views early and extract parallel features
- Uses minimal reconstruction
- Significantly increased selection efficiency (Equivalent to a 30% increase in detector mass)
- Convolution filters and outputs show interesting features about how the NN is providing discrimination
- The first use of a CNN in a published HEP result.

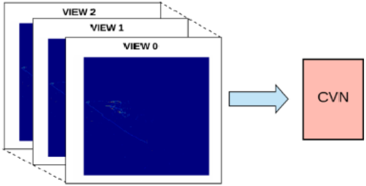
Application of ML in HEP - ProtoDUNE

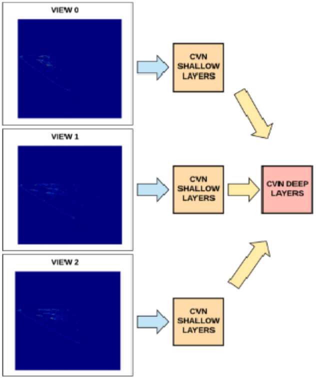
- ProtoDUNE is running in CERN with beamline from beam SPS accelerator complex.
- Input image is a set of three maps of wire hits in LA detector.
- Planes are formed with 500x500 wires.
- DUNE will have 500x3200 (coarse image will be 12.5 times larger than the NOvA CVN inputs).
- Multi output in 7 categories:



Application of ML in HEP - DUNE

- Combination of the three views (500x500x1 pixels each) into a single image with three channels so that the input of the network was a 500x500x3 pixel image.
- Splitting the network input and the first few layers into three branches (one for each view) to let the model learn from each individual view.
- Approach with 34-layer ResNet achieved (accuracies):
  Neutrino/antineutrino 73.5%, Flavour 90.3%, Interaction type 71.5%, Protons: 81.2%, Pions 84.1%, Pizeros: 90.9%, Neutrons: 99.1%.

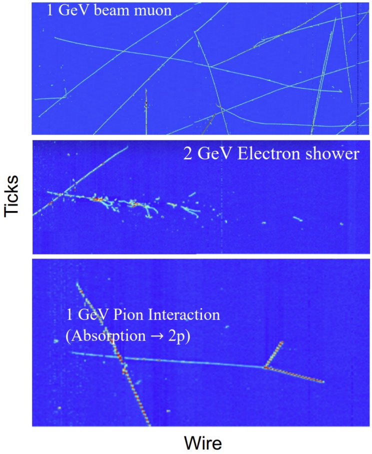




Other ML applications in DUNE

Many deep-learning algorithms developed in DUNE to solve both classification and regression problems:

- CNN based event classifiers.
- CNN for Shower/Track Separation in ProtoDUNE-SP.
- GNNs and Sparse CNNs have shown promise in reconstructing tracks and showers.
- I'm working on Proton decay identification (using CNN and ViT).



Summary and The End of The Talk

This is the end => Thank you for your attention.