Convolutional neural network for centrality in fixed target experiments

Denis Uzhva

17 oct. 2019

Saint Petersburg University

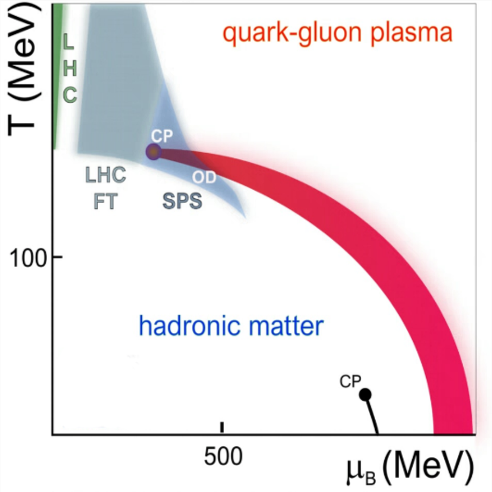
Laboratory of Ultra Hight Energy Physics

PWG1/MPD Meeting

Table of contents

Introduction

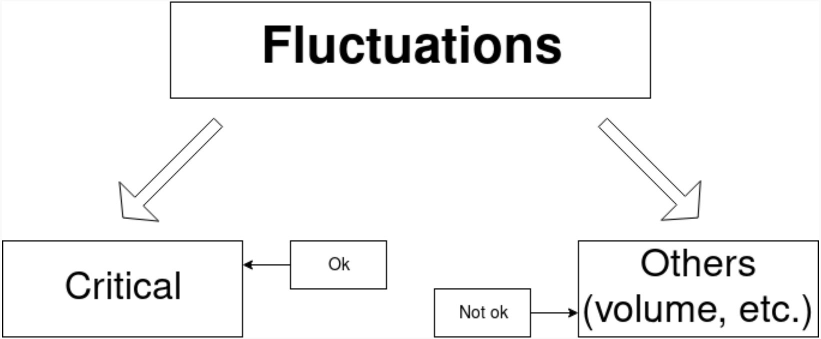
The critical point of QGP to hadronic matter transition



Quark matter phase diagram

Fluctuations of centrality

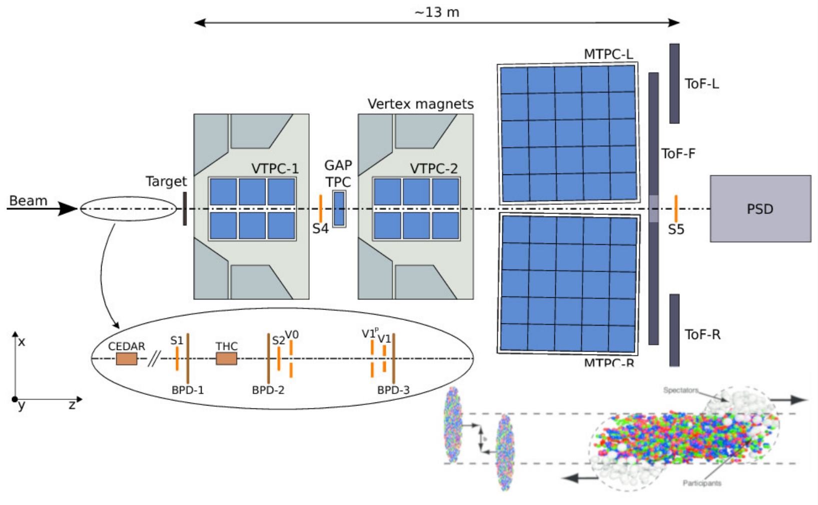
The critical point can be found (if it exists) by analysis of the fluctuations of centrality



Types of the fluctuations

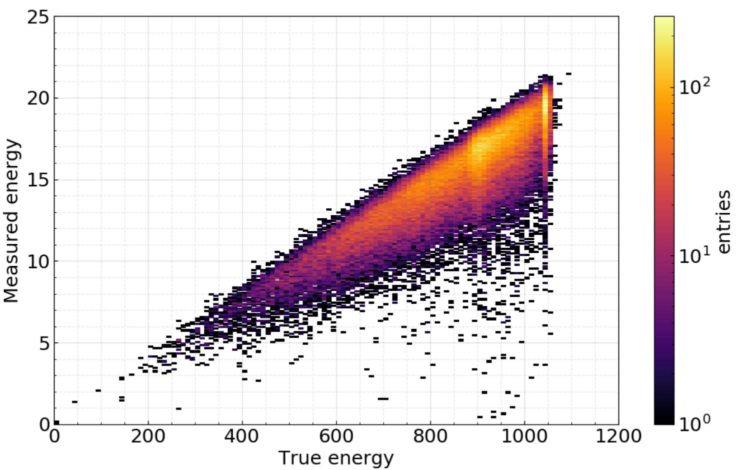
The scheme of NA61/SHINE

The centrality is measured by using only forward energy from the Projectile Spectator Detector (PSD)



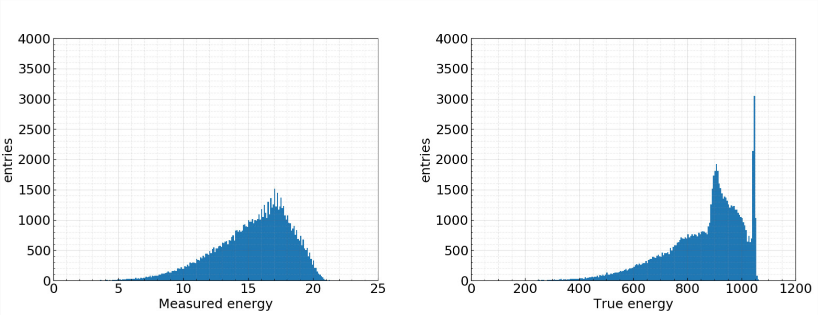
Energy cloud

SHIELD MC + GEANT4 model of PSD (Li7 + Be9). We have a dataset of 80000 minimum bias events



Histogram of the events

The reality behind measurements

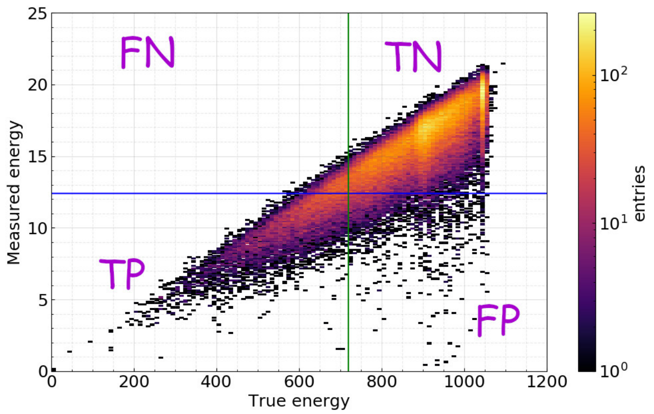


What we measure vs. what we want to measure

The problems are based on energy leakage, sandwich structure, electronics resolution and existence of matter between the PSD and the target

Cut-based analysis

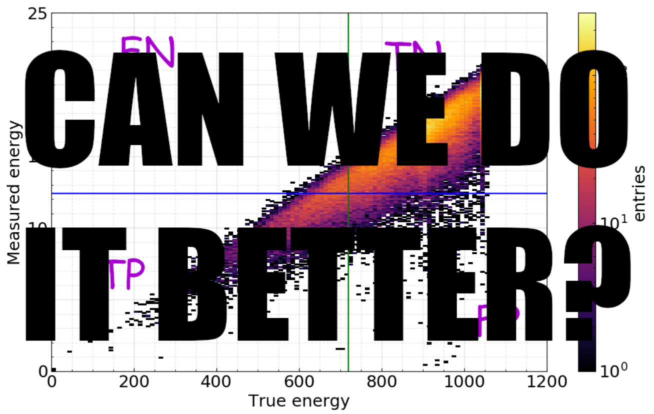
Let's choose 15.8% most central events (both by E_{true} and E_{meas}). The accuracy ϵ is calcutated as $\epsilon = TP + TN/(TP + TN + FP + FN)$



$\epsilon = 93.1\%$

Cut-based analysis

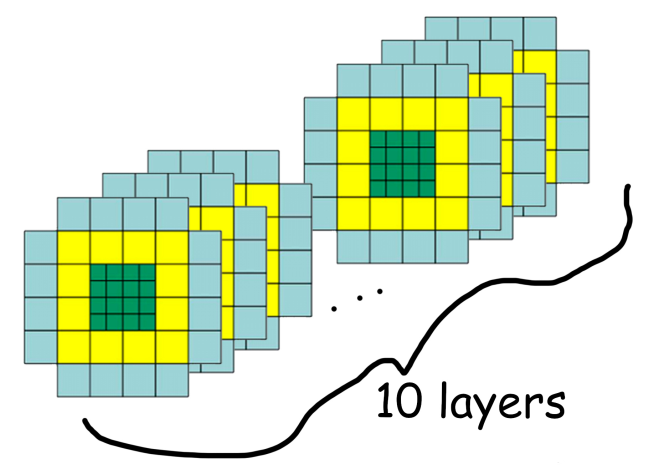
Let's choose 15.8% most central events (both by E_{true} and E_{meas}). The accuracy ϵ is calcutated as $\epsilon = TP + TN/(TP + TN + FP + FN)$



$$\epsilon = 93.1\%$$

NA61/SHINE's PSD data as pictures

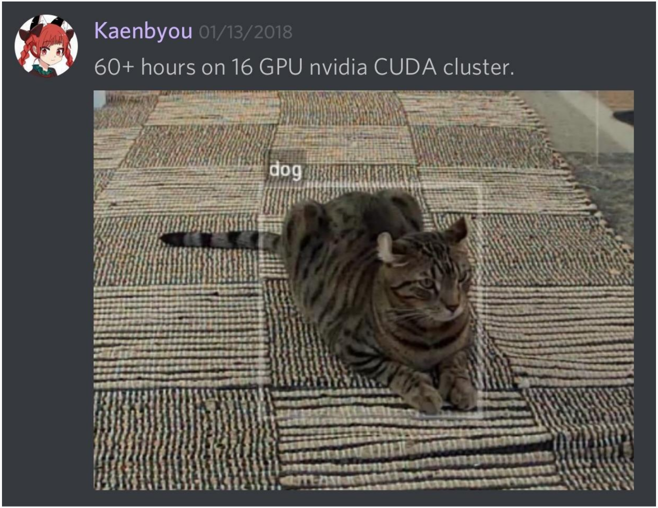
In fact, data from the PSD can be considered as 3D pics, so that we can try to use convolutional neural networks for analysis



Machine learning in HEP

What is it all about...

A modern and multipurpose method of solving various problems



The tasks for ML

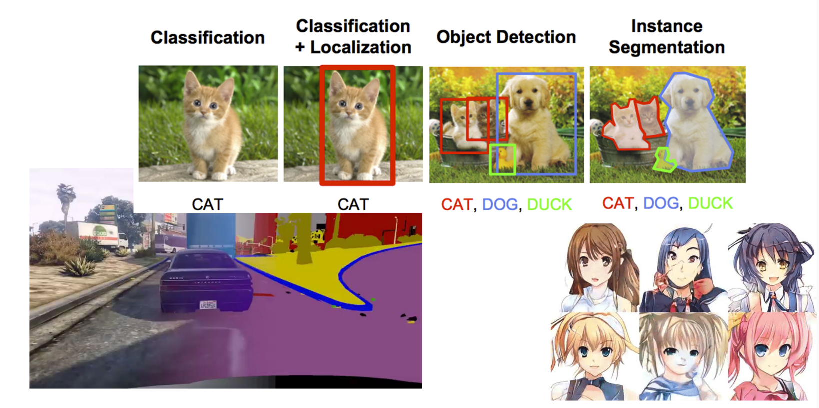
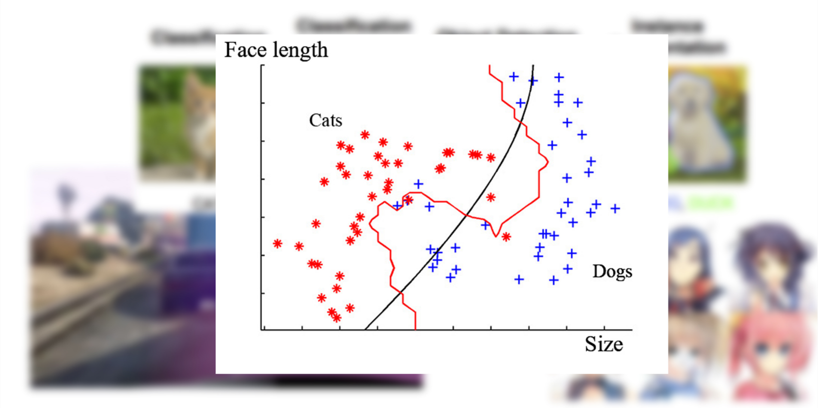


Image processing

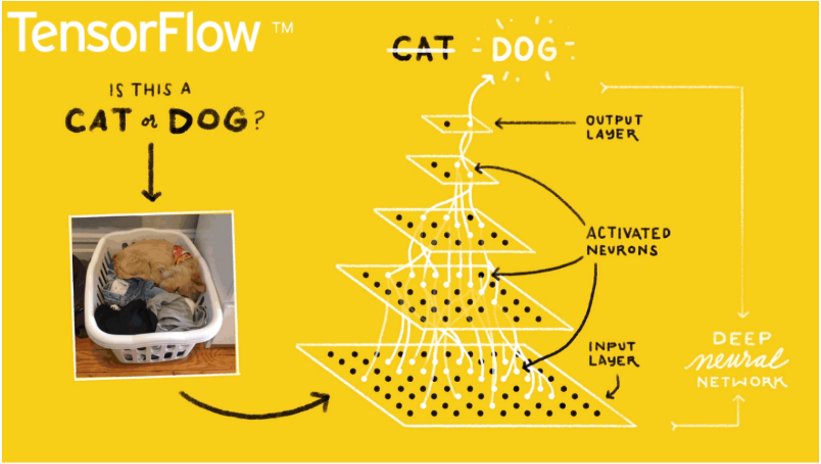
The tasks for ML



Curves separate two classes

Convolutional Neural Networks

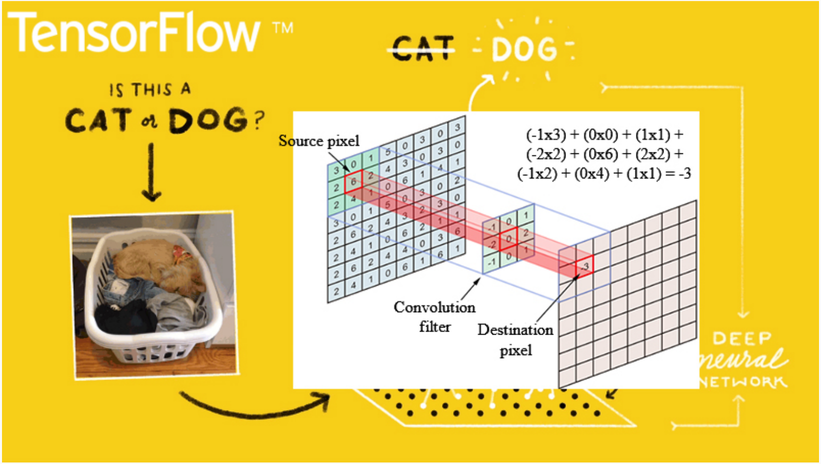
The concept of CNN is motivated by the way a real eye works



Cat-Dog classification with CNN (source: https://sourcedexter.com/quickly-setup-tensorflow-image-recognition/)

Convolutional Neural Networks

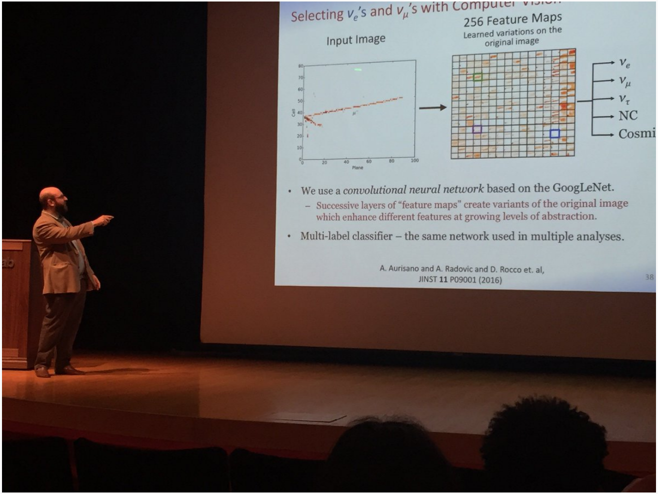
A concept of CNN is motivated by the way a real eye works



Convolution explained

Machine learning... in HEP?

ML takes care of Big Data

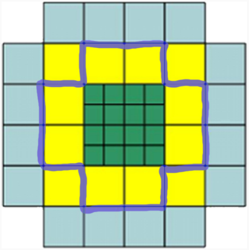


JETP seminar "First Oscillation Results from NOvA", 2018

Results and comparison

The task

Basically, we want to distinguish two classes of centrality: a) 15.8% of most central events, b) others. The dataset of 80000 minimum bias events is obtained with SHIELD MC + GEANT4 model of PSD (Li7 + Be9), 60k are for training, 20k are for validation.



The modules we choose

Only the central "+"-shaped set of PSD modules are of interest, as it is on the experiment

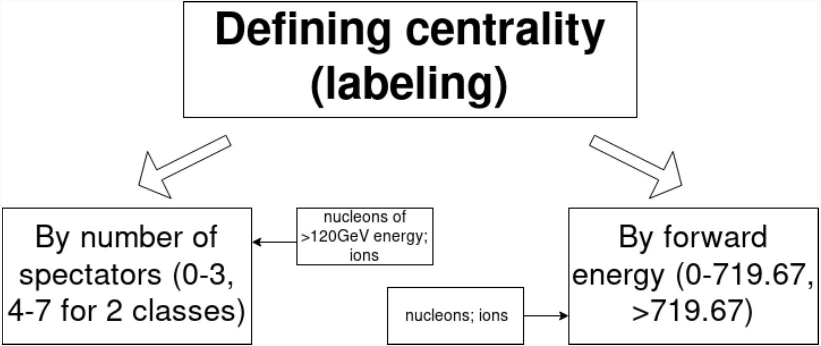
Imperfection of the simulations

- No matter between target and PSD :(

Imperfection of the simulations

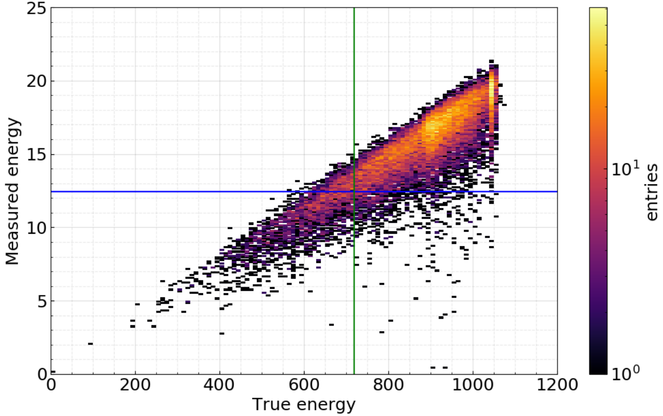
- No matter between target and PSD :(
- The electronics are not simulated :((

Definition of centrality



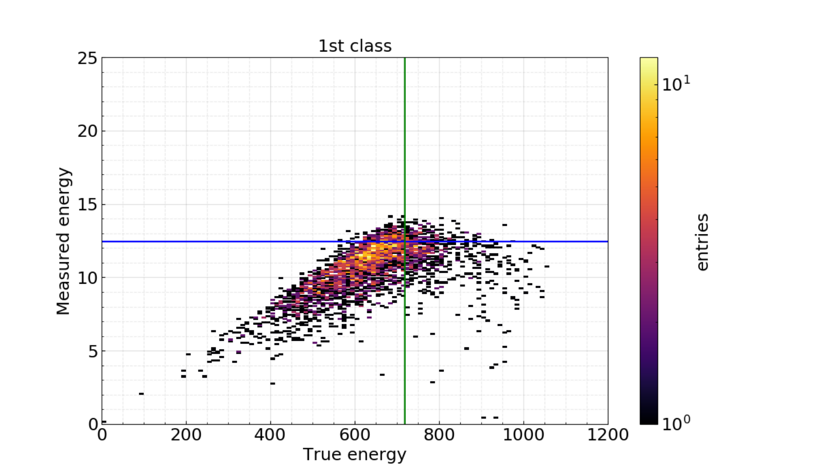
Therefore, 2 CNN models were trained (CNNn and CNNe)

Histogram analysis (by energy)



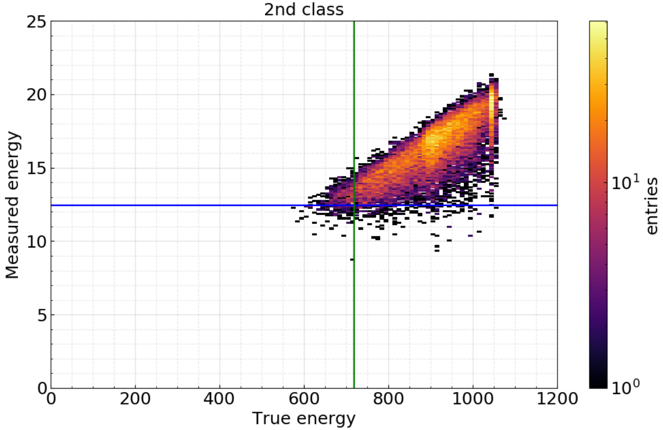
Cut-based: 93.0%

CNN separation (1st class, CNNe)



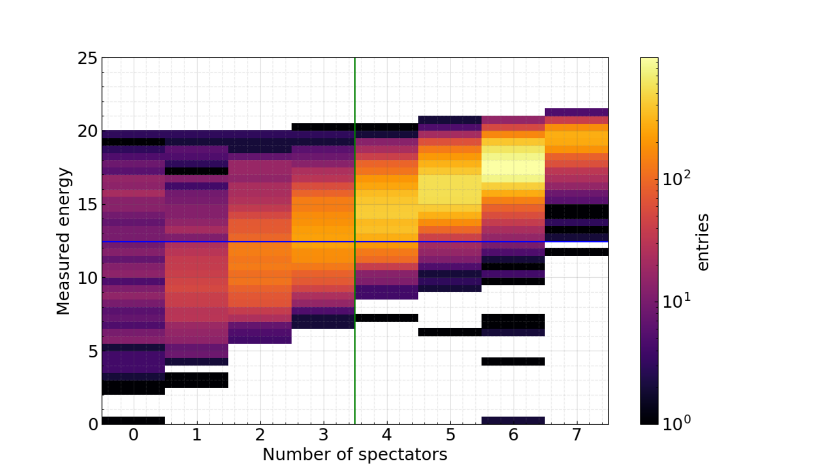
The events the CNN considered to be from the 1st class

CNN separation (2nd class, CNNe)



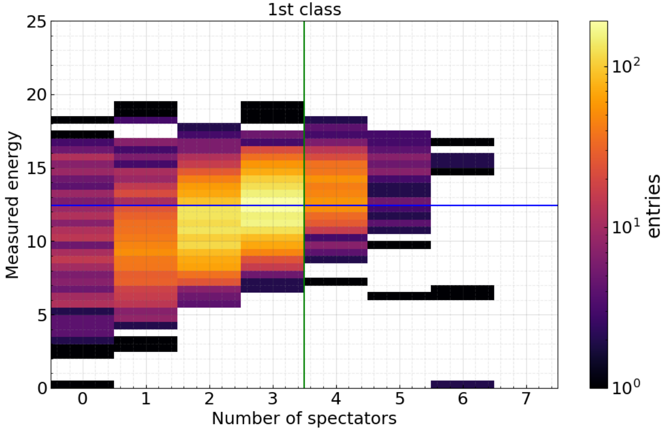
The events the CNN considered to be from the 2nd class

Histogram analysis (by spectators)



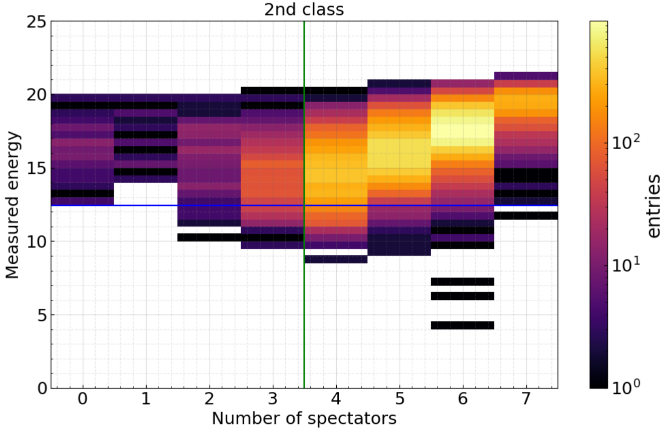
Cut-based: 86.7%

CNN separation (1st class, CNNn)



The events the CNN considered to be from the 1st class

CNN separation (2nd class, CNNn)



The events the CNN considered to be from the 2nd class

Accuracy of the CNN

CNN shows better results in accuracy, especially in the task of N_{spec} classification

	Forward energy	N_{spec}
Cut-based	93.0%	86.7%
CNN	93.7%	92.8%

Average multiplicities and variances

The $\langle N \rangle$ and ω values were calculated for the events from the 1st centrality class. Here centrality $=$ forward energy

	$\langle N \rangle$	ω
Forward energy	19.59	6.07
Cut-based	18.56	7.02
CNNe	18.69	6.82

By forward energy

Average multiplicities and variances

The $\langle N \rangle$ and ω values were calculated for the events from the 1st centrality class; centrality = number of spectators

	$\langle N \rangle$	ω
N_{spec}	15.69	7.58
Cut-based	18.56	7.02
CNNn	16.36	7.35

By number of spectators

Conclusions

Further ideas

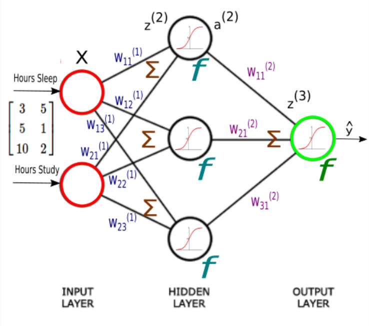
I. Cross-validation on different MC
II. Modifications of the CNN
III. Implementation to the real data

Implementation to other experiments!

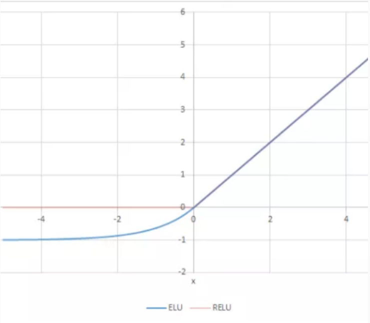
Moreover, such CNN can be used in other experiments like NICA or FAIR, since they have pretty similar calorimeters

A simple neural net

The most popular way to create A.I. today is to develop a clever enough artifical neural network. Here is the example of one.



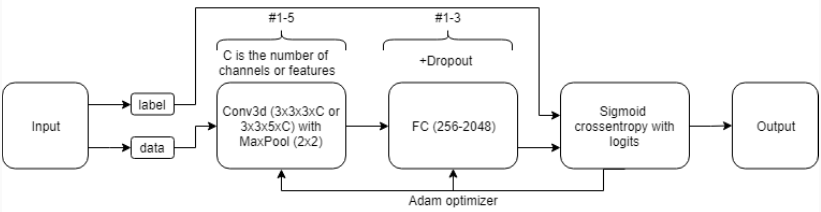
A very simple ANN



ELU and RELU functions

CNN architecture

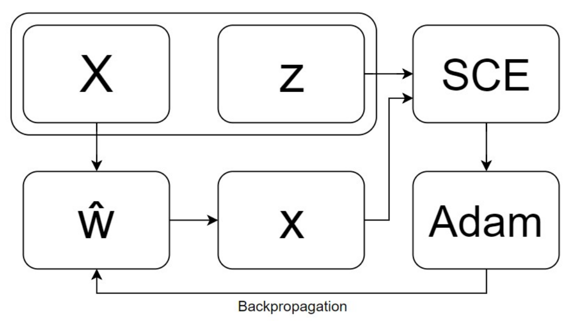
We vary the parameters of the neural network in order to achieve superior accuracy.



CNN for centrality classification

CNN architecture, but much simpler

In order to understand the concept of training, consider a simplified model



The X and z pair is the input data and labels respectively, $\hat{w}$ is the weight multitensor, x is a prediction, SCE stands for "sigmoid crossentropy", Adam is the optimizer

Sigmoid crossentropy

In binary classification, the loss function can be calculated in this way:

$$L(x, z) = -z \cdot \log \sigma(x) - (1 - z) \cdot \log(1 - \sigma(x)),$$

$$\sigma(x) = 1/(1 + \exp(-x)).$$

x is a prediction ($x = x(\hat{w}, X)$ – function of weights $\hat{w}$ and input data X), z is a label

Adam optimizer

The parameters update iteratively as follows:

$$t := t + 1;$$

$$l_t := l_{t-1} \cdot \sqrt{1 - \beta_2^t}/(1 - \beta_1^t);$$

$$\hat{m}_t := \beta_1 \cdot \hat{m}_{t-1} + (1 - \beta_1) \cdot \hat{g}_{t-1};$$

$$\hat{v}_t := \beta_2 \cdot \hat{v}_{t-1} + (1 - \beta_2) \cdot \hat{g}_{t-1}^2;$$

$$\hat{w}_t := \hat{w}_{t-1} - l_t \cdot \hat{m}_t/(\sqrt{\hat{v}_t} + \epsilon);$$

where t is epoch number, β_1 and β_2 are momenta, l_t is learning rate, $\hat{m}_t$ is "moving average" of gradient, $\hat{v}_t$ is "moving average" of squared gradient, $\hat{w}_t$ is some value (weight) and $\hat{g}_{t-1} = dL(x, z)/d\hat{w}$ at $x = x(\hat{w}_{t-1}, X_{t-1})$ and $z = z_{t-1}$ with respect to all the weights

Data and CNN parameters

- Two classes: 0-3 and 4-7 spectators (15.8% centrality), 99500 events (15687 and 83813 respectively)

Data and CNN parameters

- Two classes: 0-3 and 4-7 spectators (15.8% centrality), 99500 events (15687 and 83813 respectively)
- The best perfomance was obtained with the dropout rate parameter set as 0.1 (only 10% of FC neurons remain unzeroed)

Data and CNN parameters

- Two classes: 0-3 and 4-7 spectators (15.8% centrality), 99500 events (15687 and 83813 respectively)
- The best perfomance was obtained with the dropout rate parameter set as 0.1 (only 10% of FC neurons remain unzeroed)
- 1 conv layer with 128 features (3x3x5)

Data and CNN parameters

- Two classes: 0-3 and 4-7 spectators (15.8% centrality), 99500 events (15687 and 83813 respectively)
- The best perfomance was obtained with the dropout rate parameter set as 0.1 (only 10% of FC neurons remain unzeroed)
- 1 conv layer with 128 features (3x3x5)
- 1 max pool (2x2)

Data and CNN parameters

- Two classes: 0-3 and 4-7 spectators (15.8% centrality), 99500 events (15687 and 83813 respectively)
- The best perfomance was obtained with the dropout rate parameter set as 0.1 (only 10% of FC neurons remain unzeroed)
- 1 conv layer with 128 features (3x3x5)
- 1 max pool (2x2)
- 1 FC layer with 1024 neurons

Data and CNN parameters

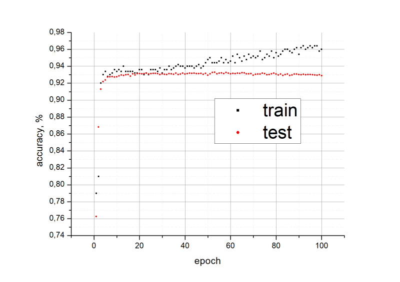
- Two classes: 0-3 and 4-7 spectators (15.8% centrality), 99500 events (15687 and 83813 respectively)
- The best perfomance was obtained with the dropout rate parameter set as 0.1 (only 10% of FC neurons remain unzeroed)
- 1 conv layer with 128 features (3x3x5)
- 1 max pool (2x2)
- 1 FC layer with 1024 neurons
- Learning rate 5*1e-4

Data and CNN parameters

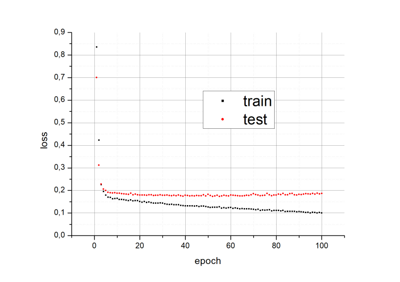
- Two classes: 0-3 and 4-7 spectators (15.8% centrality), 99500 events (15687 and 83813 respectively)
- The best perfomance was obtained with the dropout rate parameter set as 0.1 (only 10% of FC neurons remain unzeroed)
- 1 conv layer with 128 features (3x3x5)
- 1 max pool (2x2)
- 1 FC layer with 1024 neurons
- Learning rate 5*1e-4
- Batch size 100

Accuracy and loss

Two classes: 0-3 and 4-7 spectators (15.8% centrality), 99500 events (15687 and 83813 respectively)



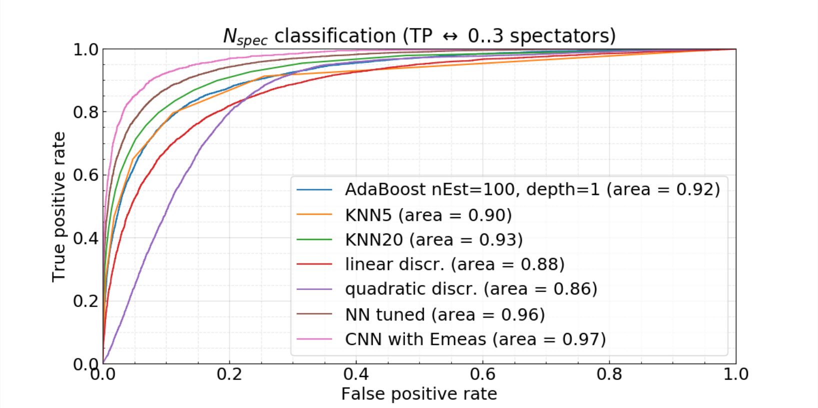
Accuracy (max 93.3% at 53 epoch)



Loss

ROC-curve and comparison with other ml methods

Measuring area under a ROC-curve is another method of defining the accuracy.



comparison of ROC-curves given different ml methods