Exclusive channel study: Neural Network

Razakamiandra Rado F. (University of Antananarivo)
Advisors: Guang Yang, David Martinez

April 15th, 2021

Overview

- Goal
- The data
- Event selection
- Neural Network
- Conclusion and Next step

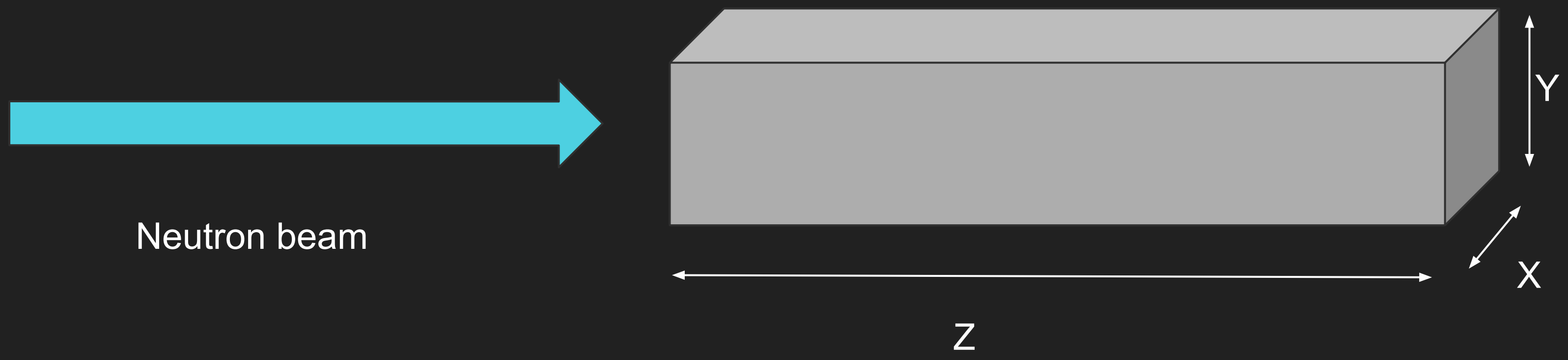
Goal

- Separate neutron induced pion and proton productions

- Compare the results with those obtained by using Hough transform

The data

- In this study we are using the data from a simulation of the SuperFGD prototype(24cm x 8cm x 48cm) shot by a neutron beam
- Neutron beam energy : 0-800 MeV uniformly distributed



Event selection

- We consider only the pion and proton events for this study
- pion events are our priority

In an event:

- ❖ if there is a pion (selected by the PDG code), this is a pion event
- ❖ if not, if there is a proton, this is a proton event

Neural Network

training and validation data → Train and evaluate the model

testing data → Predict

Build a Neural Network model → Train and evaluate the model → Predict

Input data(training, validation and testing): hit position of pion and proton events with energy deposit > 0.2 MeV in XY, YZ and XZ views.

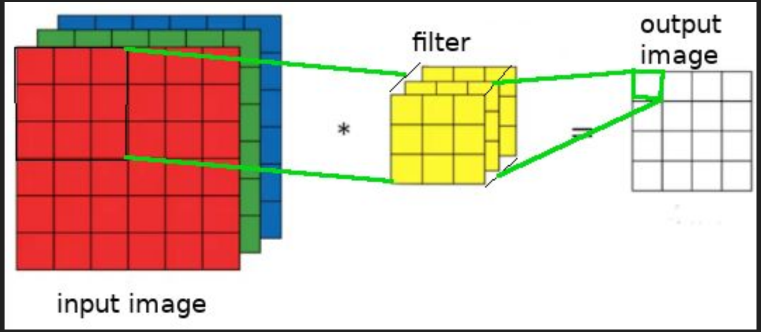
The color represents the energy deposit

WE ONLY USE XZ view

Neural Network

- We are building a Convolutional Neural Network (CNN) model to classify the images.
- Structure of a CNN: convolutional layer, pooling layer, flattening and a fully connected layer

➢ Convolutional layer: performs the convolution operation between the input image and a filter



- One color channel of the input image is stored in a 2D array where each element is the pixel's color code
- The distance between consecutive application of the filter on the image is referred to **strides**
- We can also add zero padding to the input image.
- The filters weights are initialized once and updated as the learning process progresses.

number of filters, filter size, padding, strides, filters weights initializer and **activation function** are parameters we need to tune

Neural Network

- ➢ Pooling layer: selects which information we will use for the next steps in the network
  - We need to specify the pooling size
- ➢ Flattening: transforms the multidimensional array to one dimensional array
- ➢ Fully connected layer: output of the model

Evaluation of the performance of the model

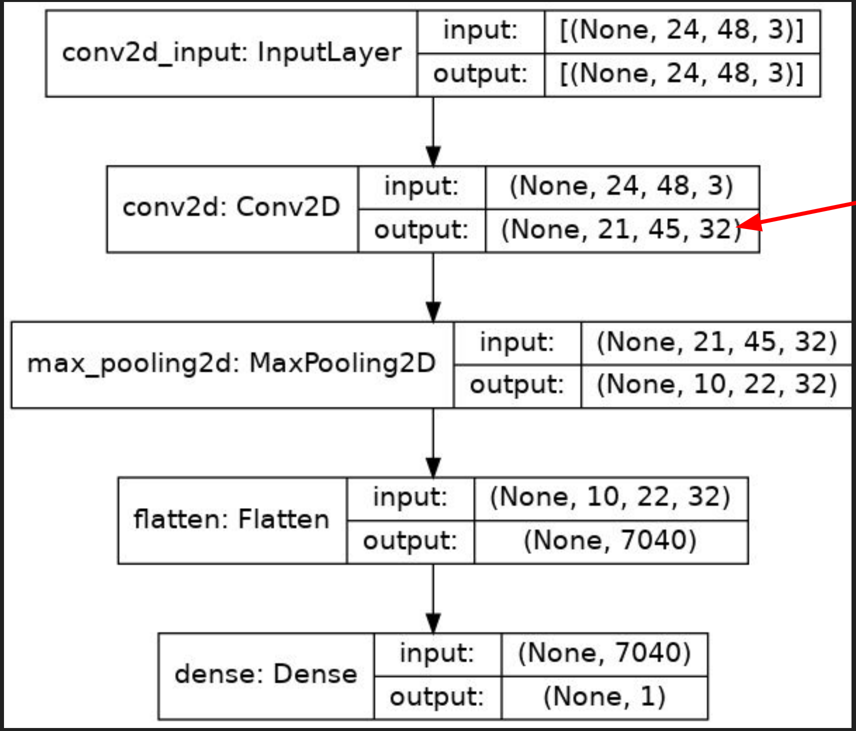
$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

Loss function: the quantity that is minimized during training

Neural Network

Example of model with one convolutional layer with parameters tuning

We only used the XZ view to train the model.



- **input shape:** (height, width, n_channels). Here we are using RGB images so n_channels=3 (Red, Green and Blue)

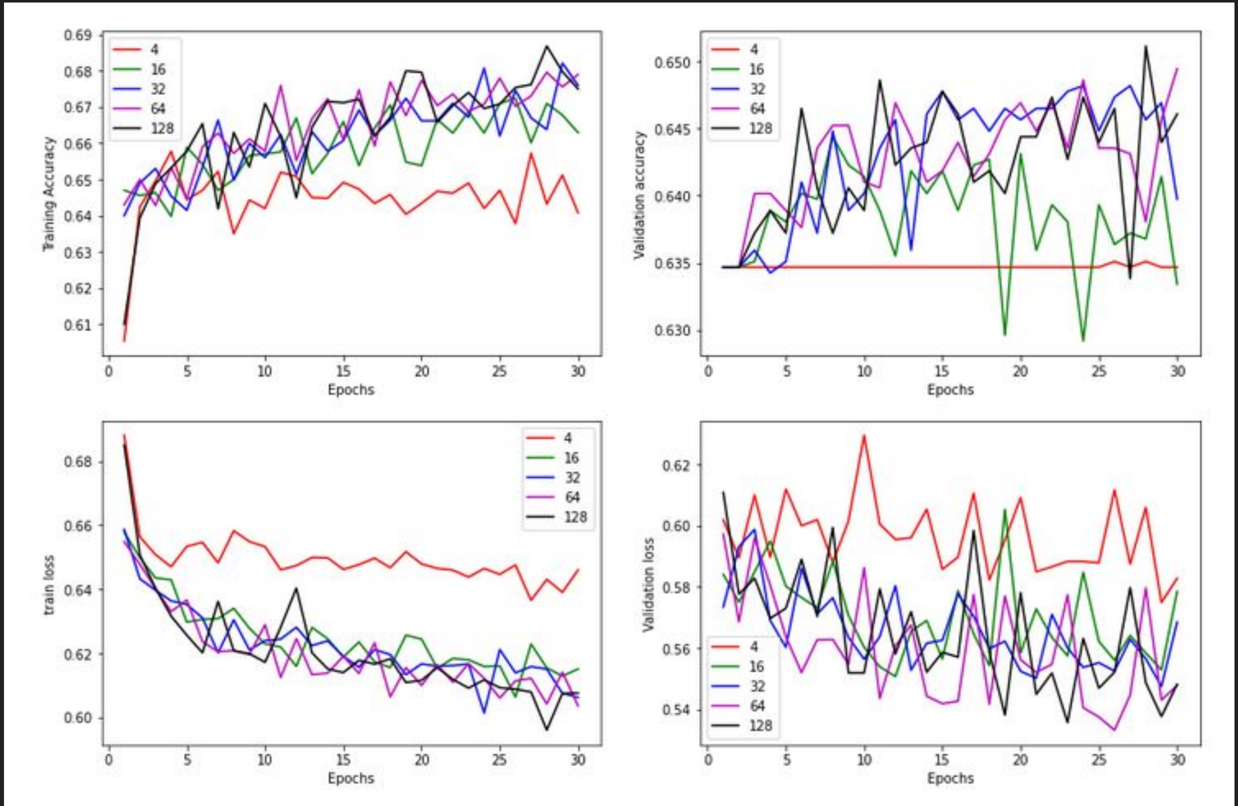
Number of filters

- Max pooling selects the maximum value in the pooling size
- output of the Flatten layer : 10*22*32
- Dense: fully connected layer, output layer of the model

To find the parameters, we tried different values and compared the accuracies and loss plots.
We picked up the highest accuracy and the lowest loss.

Neural Network

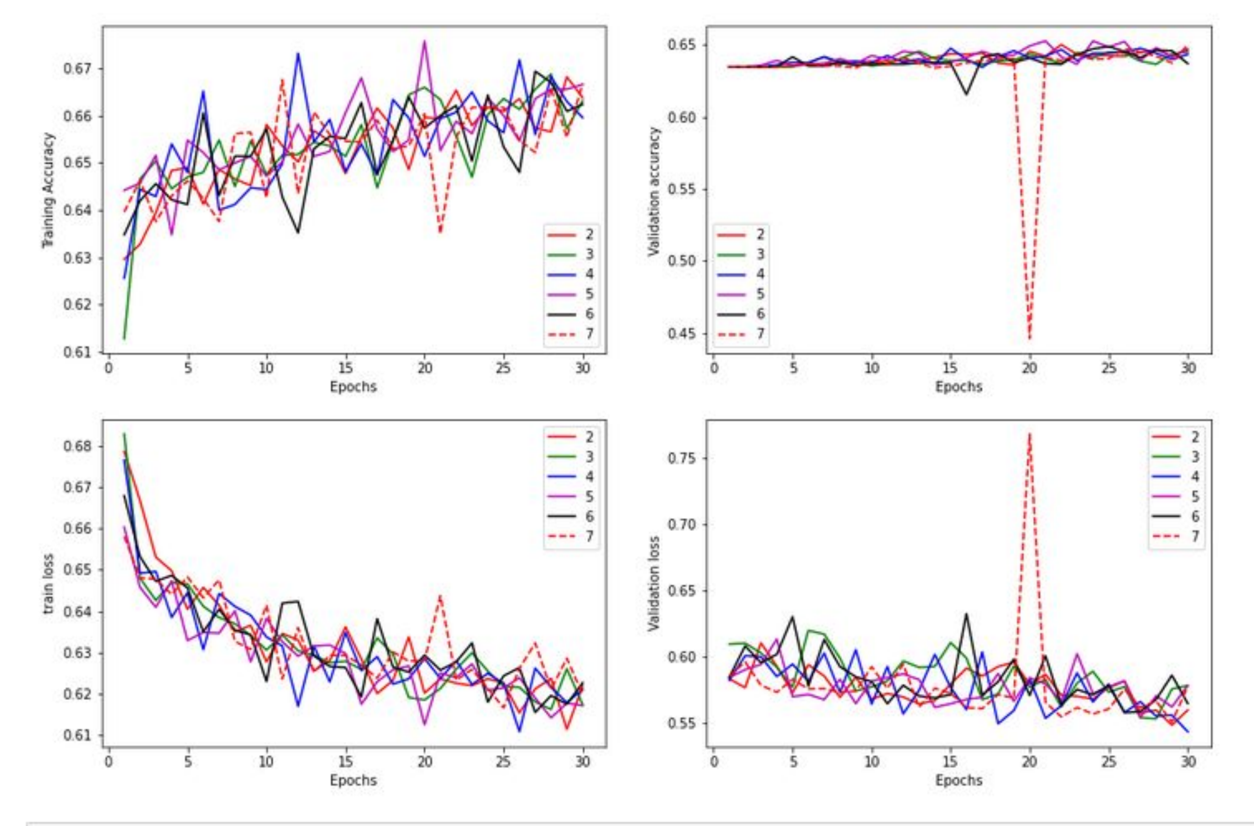
Finding the number of filters



The accuracy and loss are among the best for the blue line in the left figures which corresponds to filters = 32.

Neural Network

Finding the filter's size



To find the filter size, we are using the previous chosen number of filters which is 32.

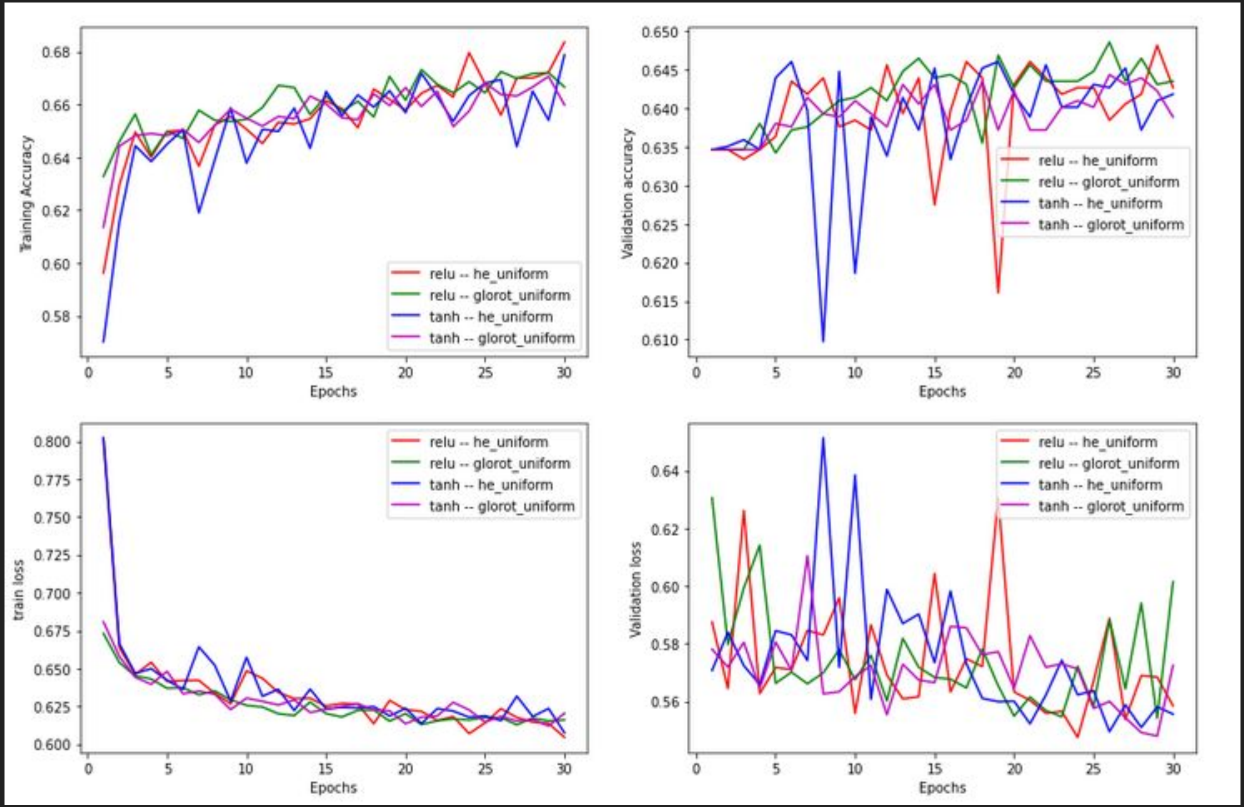
2, 3, … here mean 2x2, 3x3, and so on.

All filter size except 7x7 and 6x6 can be accepted.

For the next slides, I chose 4x4 because using 5x5 can decrease the size of the image too quickly and 2x2 and 3x3 can give us a large number of parameters of the model

Neural Network

Test of the activation function and the kernel initializer



Glorot uniform: uniform distribution over [-r, r] where

$$r = \sqrt{\frac{6}{n_{in} + n_{out}}}$$

n_in: number of input neurons
n_out: number of output neurons

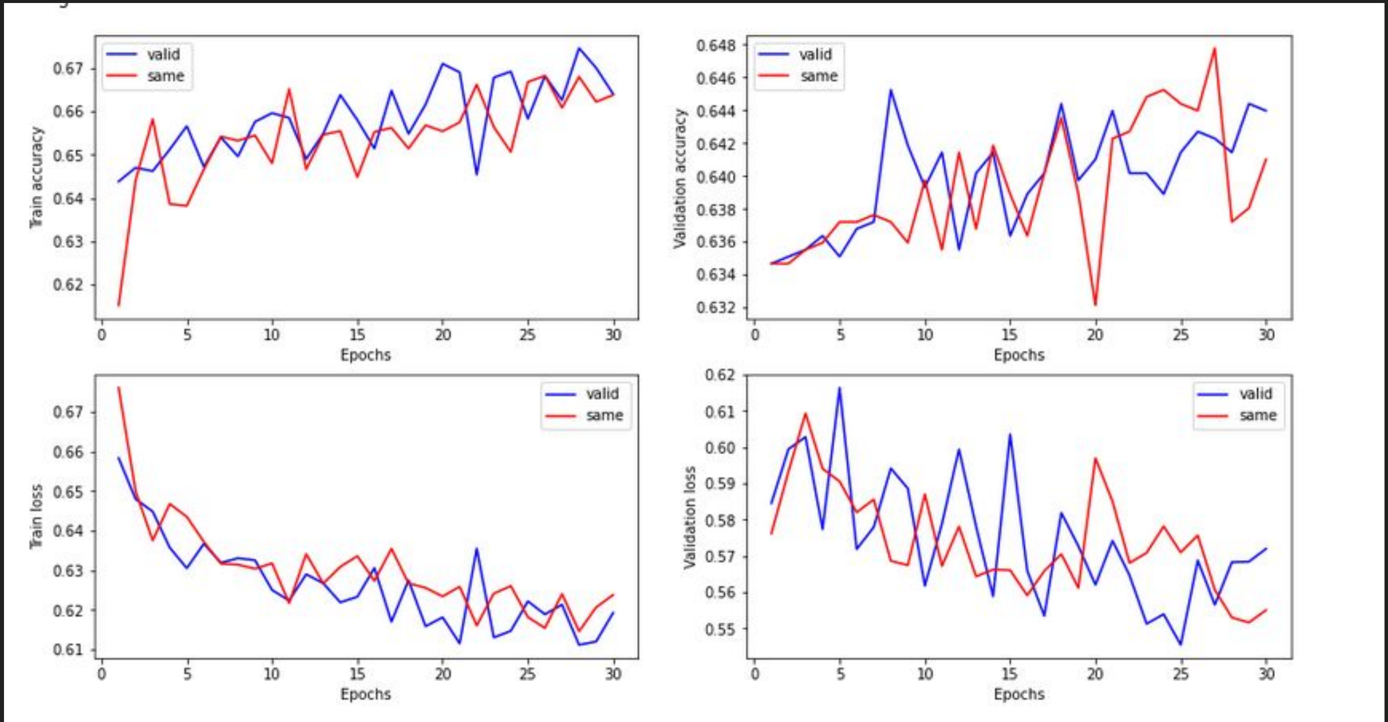
He uniform: variant of Glorot uniform

$$r = \sqrt{2}\sqrt{\frac{6}{n_{in}+n_{out}}}$$

==> tanh with glorot uniform is better

Neural Network

Testing the Padding : 'same' or 'valid'

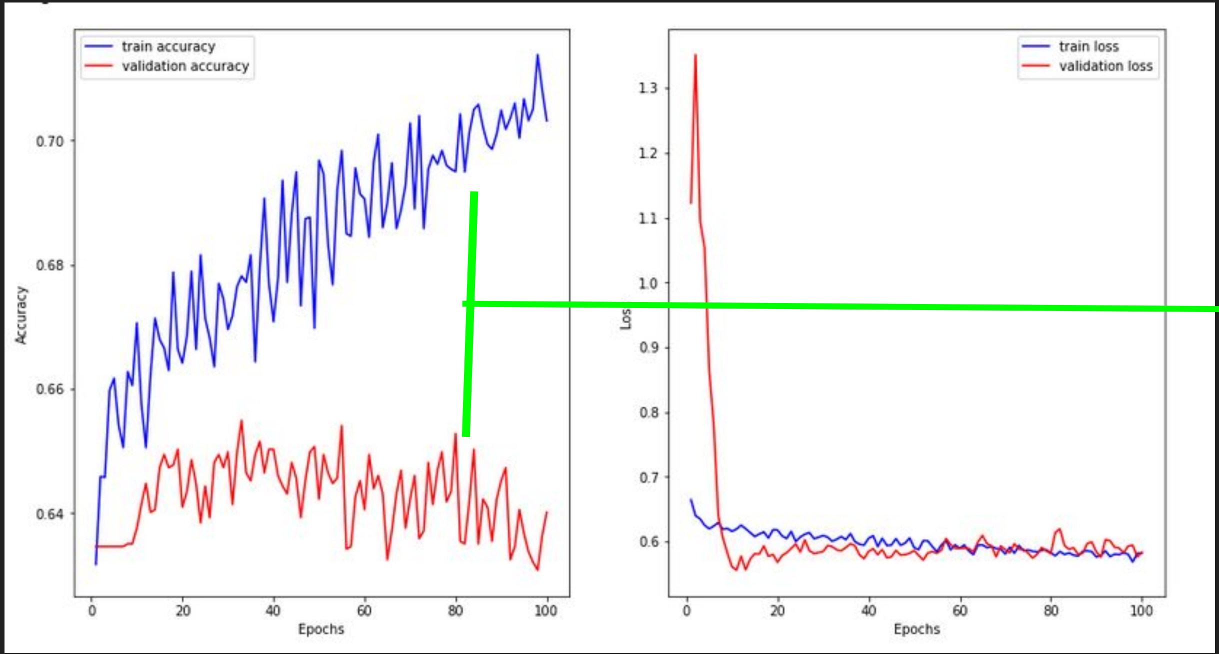


padding = 'valid' means do not apply any padding to the input image.
padding = 'same' means apply zero padding to the input image so that the output size is the same as the input size.

=>The results are similar so we choose no padding (valid)

Neural Network

Accuracy and loss using the chosen parameters:
number of filters = 32, filter size = 4x4, filter initializer = glorot uniform, activation function = tanh, padding = ‘valid’
The model was trained using the XZ views only.

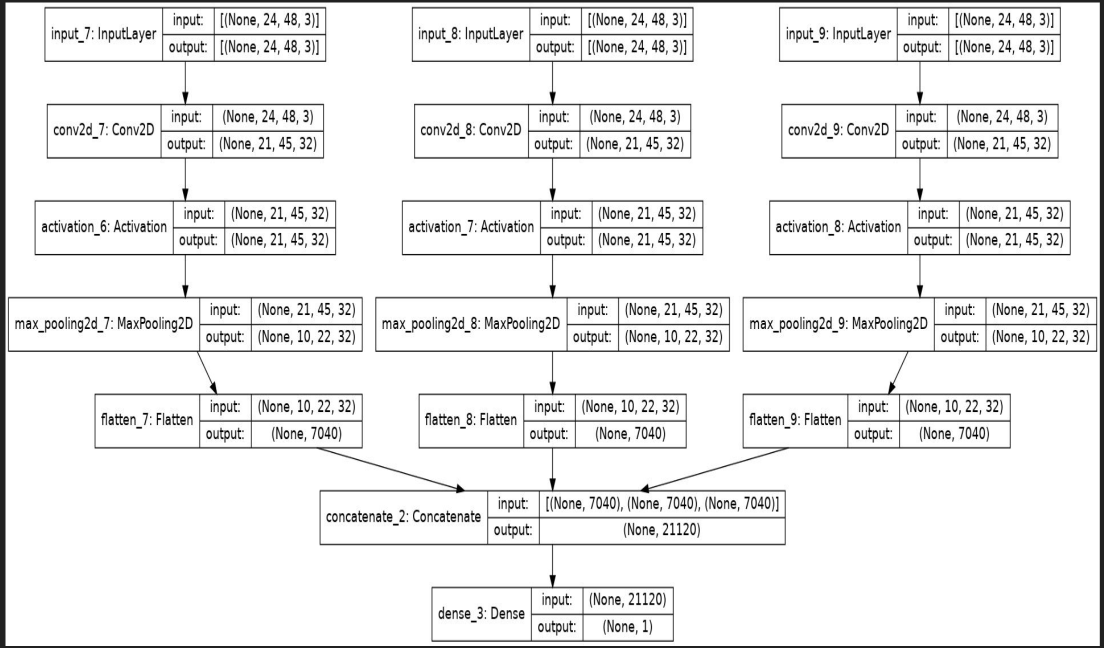


We need to fix this

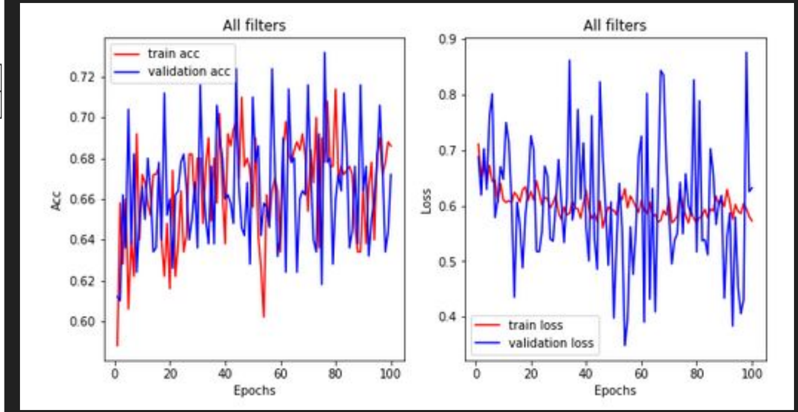
Neural Network

Using the three views at the same time to train the model

XY YZ XZ



Accuracy and loss using the chosen parameters:
number of filters = 32, filter size = 4x4, filter initializer = glorot uniform, activation function = tanh, padding = 'valid'



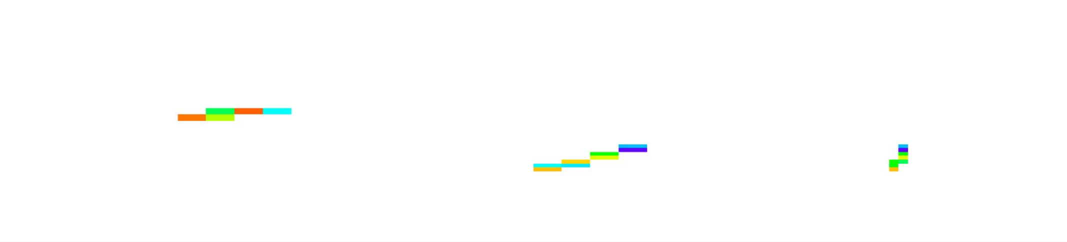
===> The accuracies are not optimal but better than those obtained with single view(XZ) as input.

Conclusion and Next step

- Neural network can help us separating pion and proton events from images but the model still needs to have a better accuracy.
- Now we are trying to improve the accuracy and the loss of the model by using the three views in the same image

pion

proton



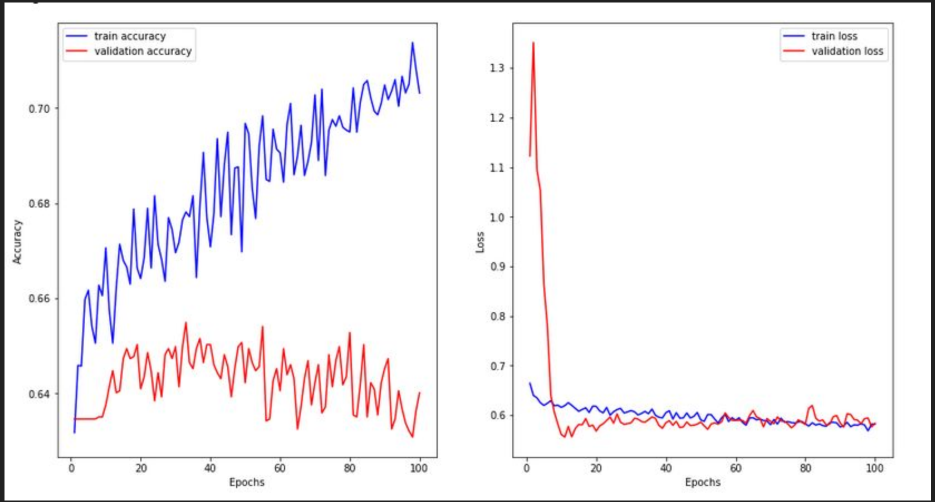
Backup

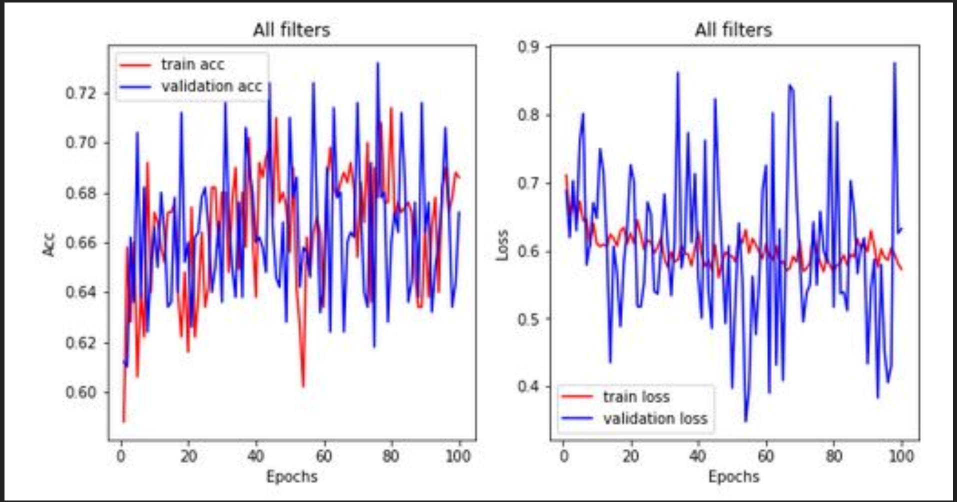
Loss function

$$-(y\log(p) + (1-y)\log(1-p))$$

- log : the natural log
- y: binary indicator(0 for pion, 1 for proton)
- p: predicted probability

Comparison between one view input and three views inputs





Plots for the model using XZ views only as input

Plots for the model using XY, YZ and XZ views as input