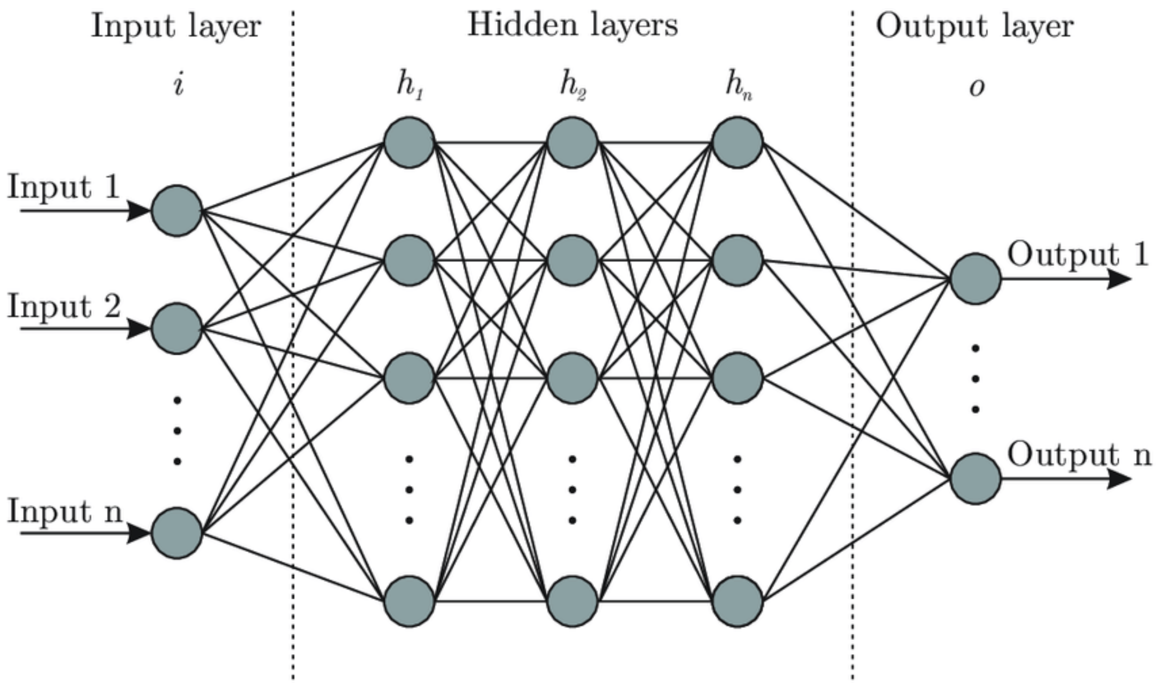
Speaker: Maria Francisca Queirós
Supervisor: Marcin Stolarski
LIP | Laboratório de Instrumentação e Física Experimental de Partículas
September 11, 2020

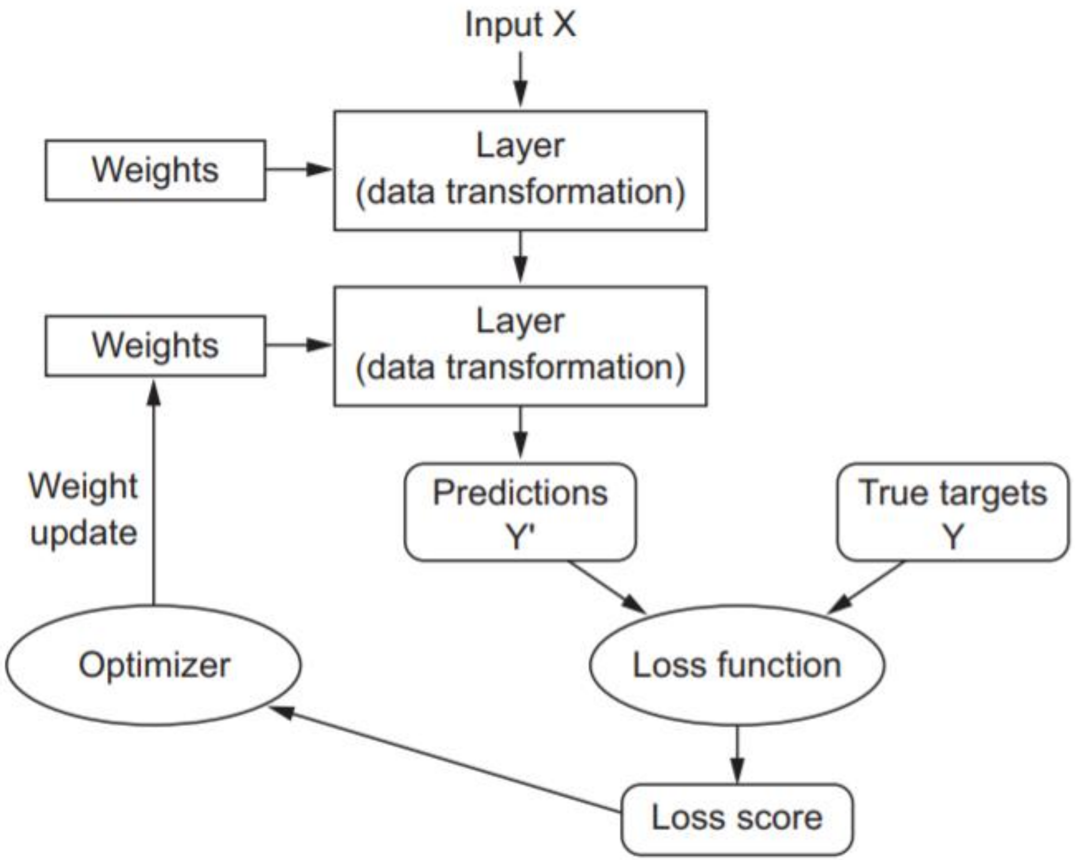
COMPASS acceptance obtained using Machine Learning Techniques

Objective of this internship

- Using Neural Networks to produce the acceptance of COMPASS experiment;
- Learning more about the COMPASS experiment;
- Learning more about the use of Neural Network.

Neural Network

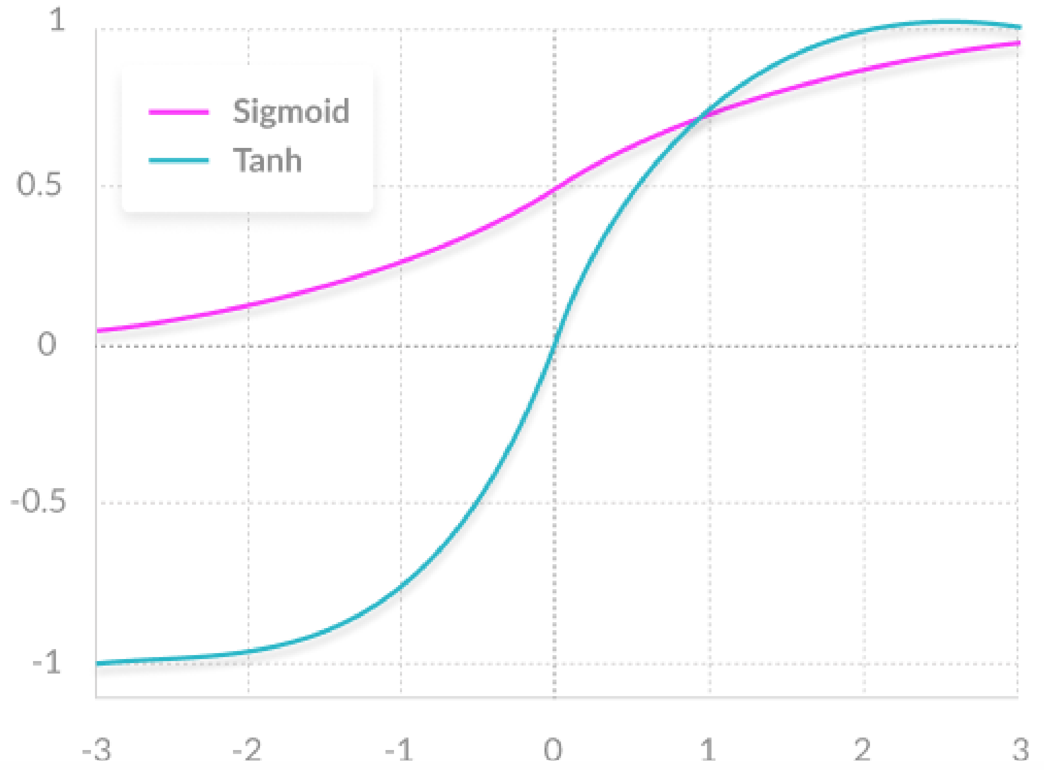




Neural Network

Activation function

- Transforms the input to some other form;
- Attached to each neuron;
- Determines whether it should be activated or not;
- Helps normalization of the ouput.



Loss function

$$L = -\frac{1}{N}\sum_i y_i \log p_i + (1 - y_i)\log(1 - p_i)$$

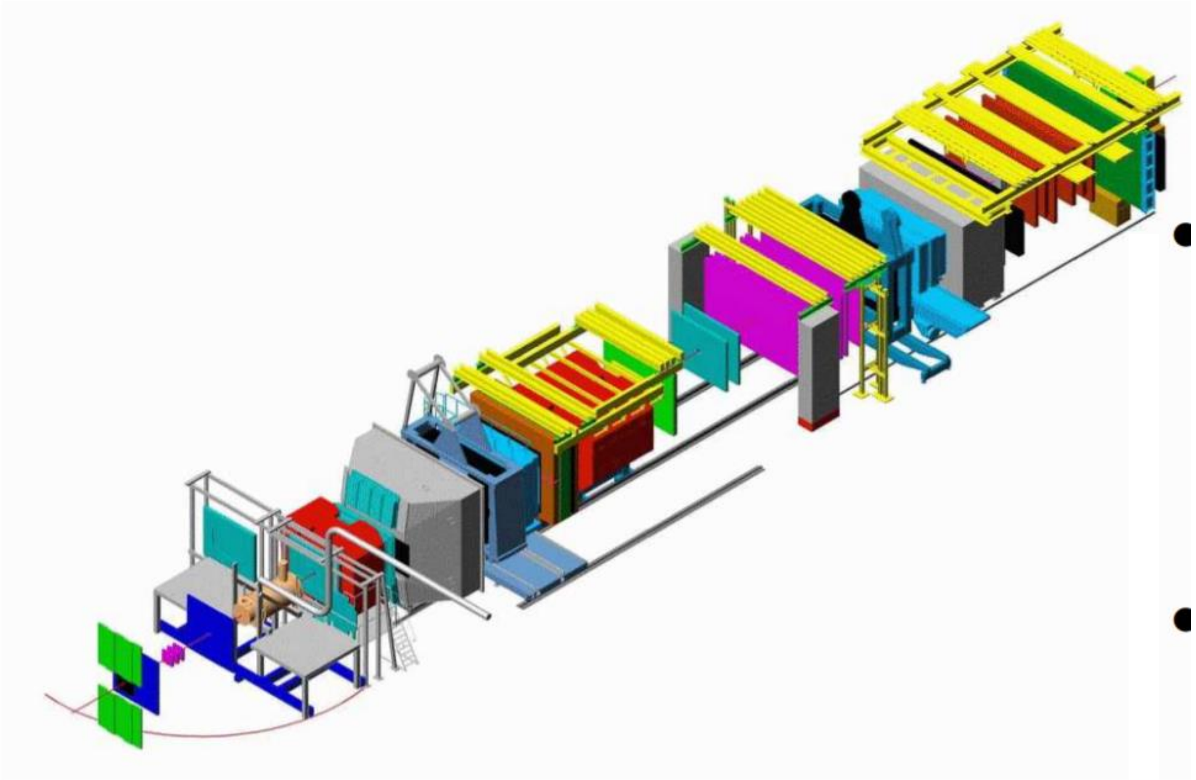
Gradient descendent

$$w^{t+i} = w^t - \eta \nabla L$$

Compass

- Fixed target experiment;
- General purpose spectrometer;
- Muon and hadron beams;
- Polarised target (longitudinally and transversely polarised NH3 ans 6LiD).

COMPASS



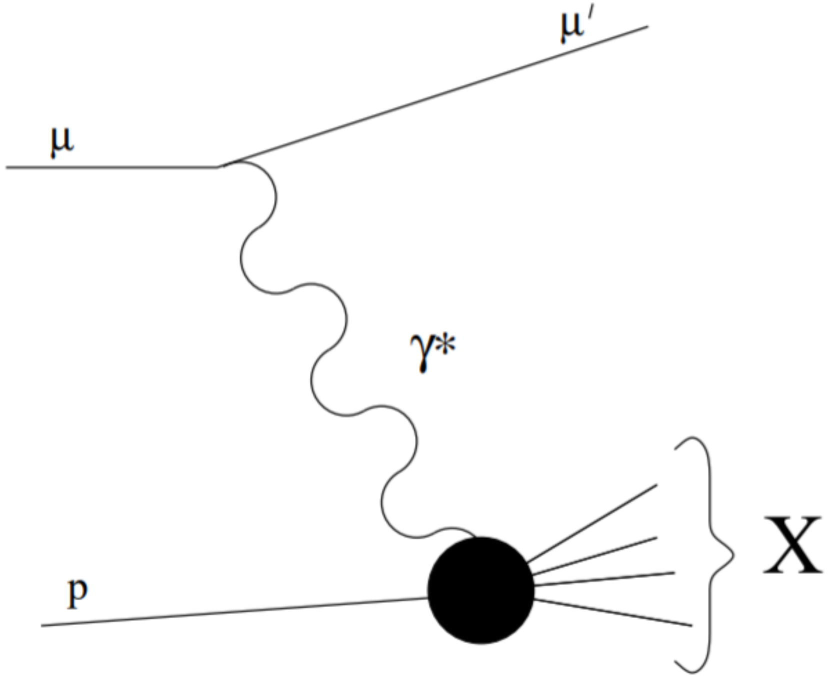
- DETECTOR
  - two stage spectrometer
  - 60 m length
  - 2 (3) magnets
  - about 350 detector planes

- POLARIZED TARGET
  - 6LiD (NH3) target
  - 2-3 cells (120 cm total length)
  - ± 50% (90%) polarization
  - polarization reversal every 8h-24h
- POLARIZED BEAM
  - μ + at 160 GeV/c (200 GeV/c in 2011)
  - polarization –80 %
- FEATURES
  - angular acceptance: ±70 mrad (±180 mrad from 2006)
  - track reconstruction: p > 0.5 GeV/c
  - identification h, e, μ: calorimeters and muon filters
  - identification: π, K, p (RICH) p > 2, 9, 18 GeV/c respectively

Processes

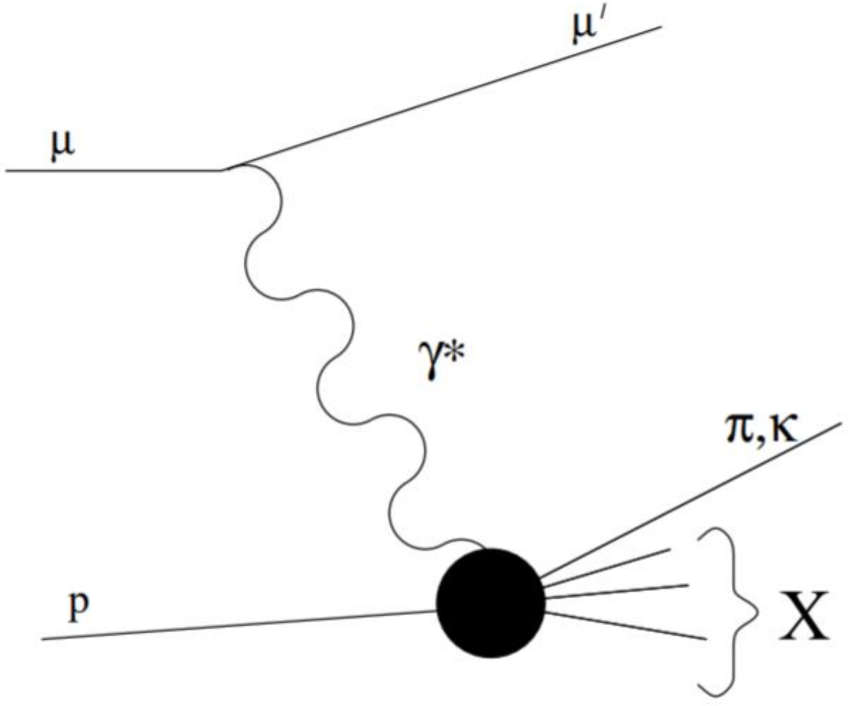
Deep Inelastic Scattering (DIS)

- Variables: x,Q2 or y



Semi-Inclusive Deep Inelastic Scattering(SIDIS)

- There is a hadron in the sample
- Variables: x ,Q2 ,y, z



Compass experiment

What may happen to the particles:

- Some may decay;
- Some re-interact and lost;
- Appearance of holes in the detectors;
- …

We only see between 30-70% of the generated particles

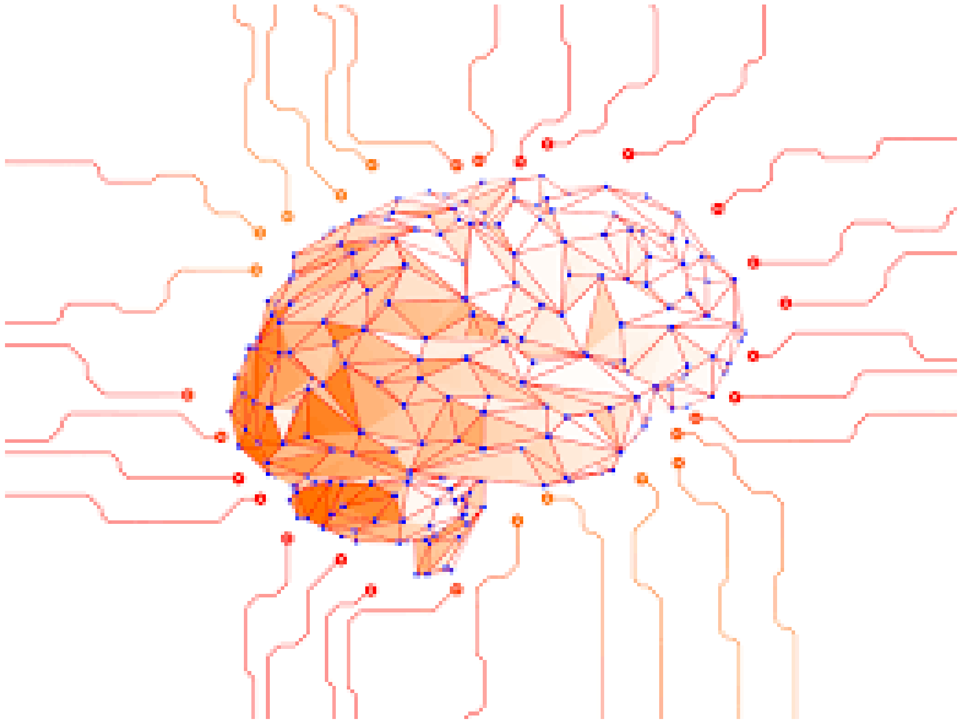
Acceptance

Compass acceptance and Machine Learning

- Check if the output of NN is similar to the one that we have from MC;
- We compare the output of NN and the ouput of MC;
- We compare the acceptance generated by NN and by MC

Parameters

- Activation function(relu, sigmoid, tanh,elu, swish …)
- Number of neurons in each layer;
- Number of epochs
- Number of layers of NN;
- Optimizer;
- Dropout;
- Adding loops.



u20mqueiros@lipml01:~/NN

GNU nano 2.3.1 File: datamc2.py

```
mean=X_train_org.mean(axis=0, keepdims=True)
stdev=np.std(X_train_org,axis=0)

X_train =(X_train_org-mean)/stdev
X_test = (X_test_org -mean)/stdev     #test and training samples are normalized by the mean and stdev from the test sample



model = Sequential()
model.add(Dense(40, activation=swish, input_dim=10))   #input_dim differs from DIS and SIDIS samples
# model.add(Dense(40, activation=swish))
model.add(Dense(40, activation=swish))
model.add(Dense(40, activation=swish))
model.add(Dense(40,activation=swish))
model.add(Dropout(0.4))
model.add(Dense(1, activation='sigmoid'))  #for binary crossentropy output should have 'sigmoid'/'linear' activation functionmodel.summary()

NADAM=tf.keras.optimizers.Nadam(learning_rate=0.001,beta_1=0.999, beta_2=0.999 )

model.compile(loss='binary_crossentropy', optimizer=NADAM)
#model.compile(loss='mse', optimizer=NADAM) # mse loss (obs-exp)^2, also returns probability

totalx_min=100 # can be changed to lower values if too many models are saved

for x in range(0,70):
 print('xxxxx',x)
 model.fit(X_train, Y_train,batch_size=10000, epochs=5, verbose=1)
 loss=model.evaluate(X_test, Y_test,batch_size=1000000)
 if(loss<0.6230):
  test=model.predict(X_test[:], batch_size=1000000)

  a=np.linspace(0,0.50,11)
  totalx=0
  for i in a:
```

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

00:37

Compass acceptance and Machine Learning

Training

- Acceptance=Nreconstructed/Ngenerated
- Output of NN=Nreconstructed/(Nreconstructed+Ngenerated)
- Formule do acceptance = out/1-out

Exercises

- The output should be similar to 1.

Best results

```
-rw-r--r-- 1 u20mqueiros comp 111832 Aug 28 20:12 model-0.6215-72.46-405.56.hdf5
-rw-r--r-- 1 u20mqueiros comp 111832 Aug 28 20:13 model-0.6213-47.50-395.55.hdf5
```

```
-rw-r--r-- 1 u20mqueiros comp 111832 Aug 28 01:39 model-0.6213-38.93-388.90.hdf5
-rw-r--r-- 1 u20mqueiros comp 111832 Aug 28 01:47 model-0.6222-54.74-411.75.hdf5
-rw-r--r-- 1 u20mqueiros comp 111832 Aug 28 01:48 model-0.6218-49.98-418.08.hdf5
-rw-r--r-- 1 u20mqueiros comp 111832 Aug 28 01:48 model-0.6214-30.24-399.86.hdf5
```

```
-rw-r--r-- 1 u20mqueiros comp 111832 Aug 28 16:44 model-0.6212-65.79-386.51.hdf5
-rw-r--r-- 1 u20mqueiros comp 111832 Aug 28 16:57 model-0.6222-69.82-403.21.hdf5
-rw-r--r-- 1 u20mqueiros comp 111832 Aug 28 16:58 model-0.6219-44.69-414.80.hdf5
```

```
-rw-r--r-- 1 u20mqueiros comp 111832 Aug 25 15:44 model-0.6222-185.40-433.28.hdf5
-rw-r--r-- 1 u20mqueiros comp 111832 Aug 25 15:45 model-0.6225-43.33-441.98.hdf5
```

Aditional work

- Generative Adversarial Networks
- Genetic Algorithm
- Digit recognition

Conclusion

- There are some parametrization that are more stable than other;
- They have still a big chi-squared;
- But there are some results that can be used.

Thank you for your attention!