MACHINE LEARNING FOR DATA QUALITY MONITORING (DQM) AT CMS

PHYSICS RESEARCH SEMESTER ABROAD

MENTOR | FEDERICO DE GUIO, Ph.D.

STUDENT | GUILLERMO A. FIDALGO RODRÍGUEZ

OBJECTIVES AND MY CONTRIBUTION

- The project aims at applying recent progress in Machine Learning techniques to the automation of the DQM scrutiny for HCAL
  - Focus on the Online DQM.
  - Compare the performance of different ML algorithms.
  - Fully supervised vs semi supervised approach.

THE CHALLENGE

- Deciding the best architecture of the network is key
  - Too little and it may not be able to learn (underfitting)
  - Too big and it may learn to only identify very specific and/or unnecessary features (overfitting)
- There is no rule of thumb
  - Many, many, many…… possible combinations.

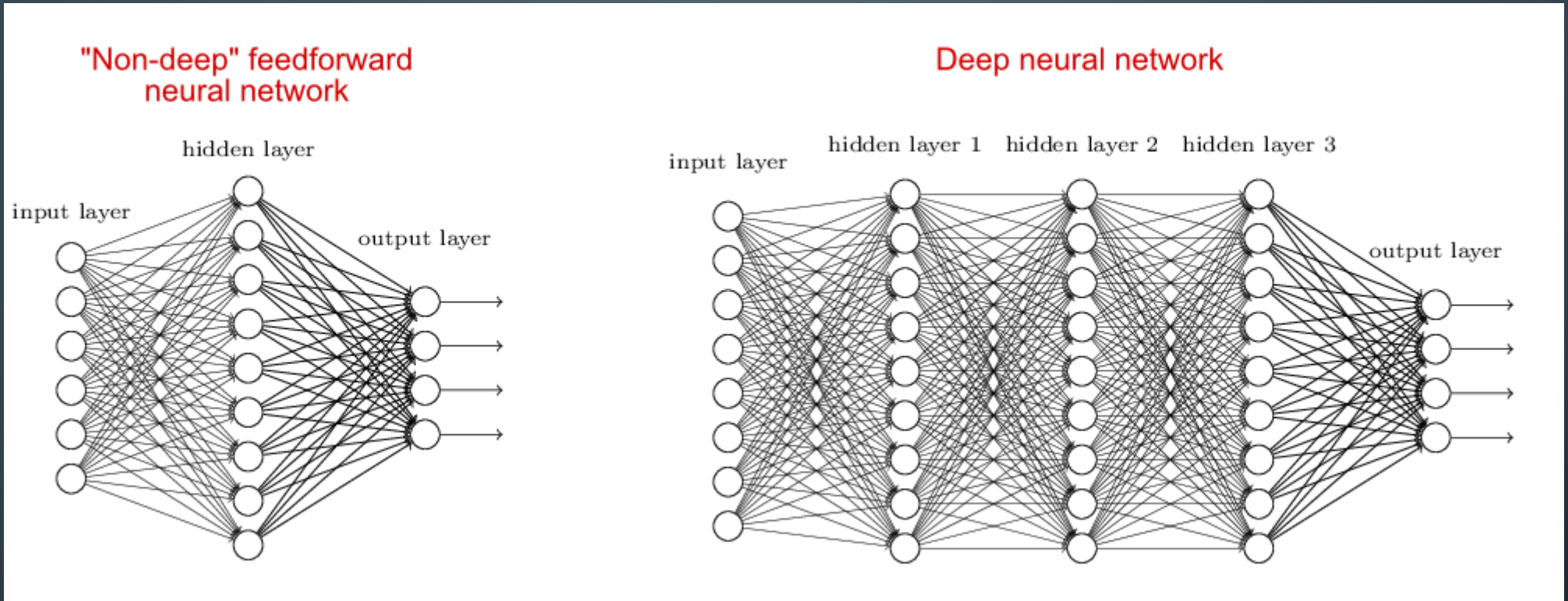
WHAT I'VE BEEN DOING

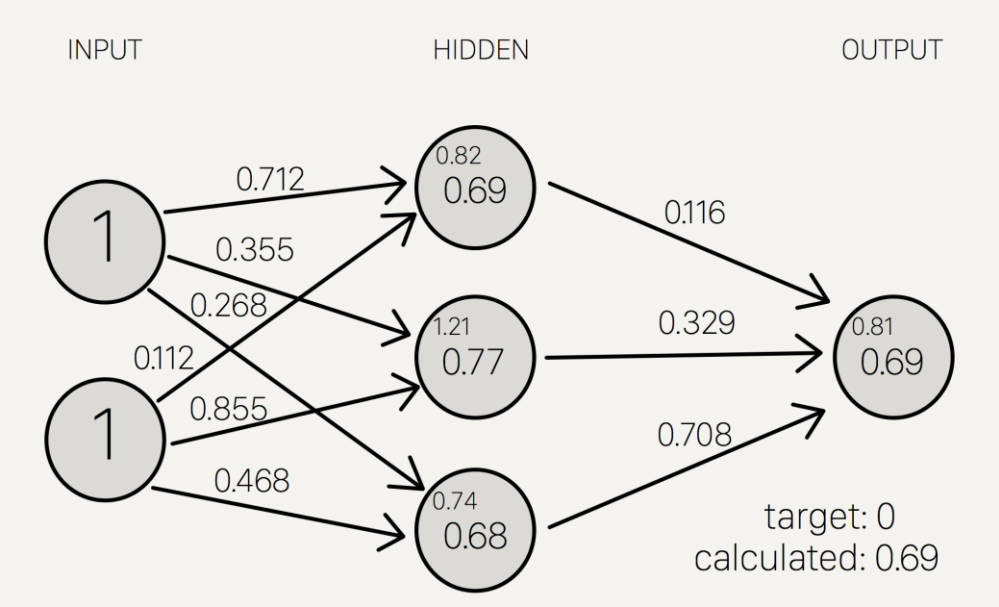
SKLEARN

- Pre-defined models
  - Logistic Regression
  - MLP
- Not much control over the model's architecture

KERAS

- Make your own models
  - A bit sophisticated
- Neural Networks
  - Deep Convolutional
    - Best with image recognition


"Non-deep" feedforward neural network
hidden layer
input layer
output layer
Deep neural network
input layer
hidden layer 1
hidden layer 2
hidden layer 3
output layer


INPUT
HIDDEN
OUTPUT
1
1
0.712
0.355
0.268
0.112
0.855
0.468
0.82
0.69
1.21
0.77
0.74
0.68
0.116
0.329
0.708
0.81
0.69
target: 0
calculated: 0.69

HOW A DEEP NEURAL NETWORK SEES

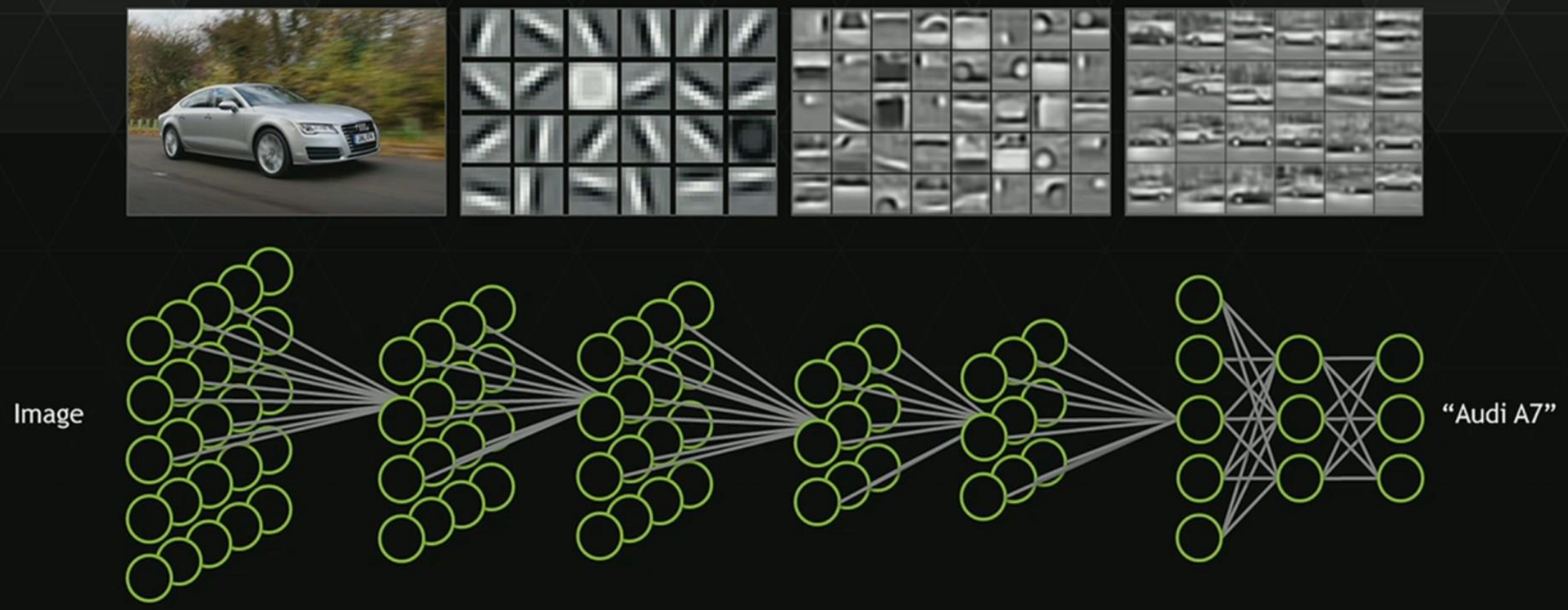
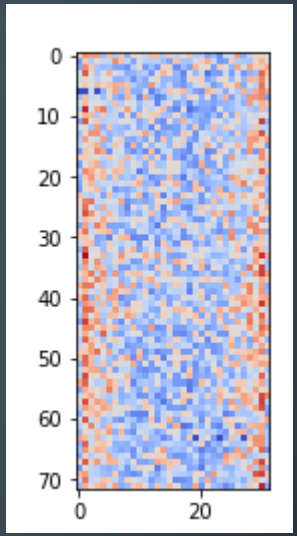
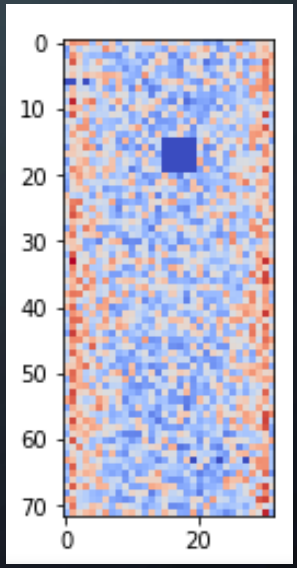


Image source: "Unsupervised Learning of Hierarchical Representations with Convolutional Deep Belief Networks" ICML 2009 & Comm. ACM 2011.
Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Ng.

PROGRESS SO FAR

- Able to make simple architectures through trial and error
- Measuring the performance
- Results have been promising for simple problems
  - 5x5 hot region with fixed location
  - 5x5 hot region with random location
- Still figuring out how to train the model with a multiple types of problems





WITH SIMPLE (BINARY, FIXED LOCATION) PROBLEM

SIMPLE ARCHITECTURE

```
model = Sequential([
    Conv2D(64, kernel_size=(2, 2), activation='relu', input_shape=(input_shape),data_format='channels_last'),
    Flatten( ),
    Dense(2,activation='softmax')
])
model.compile(loss='categorical_crossentropy',optimizer='adadelta',metrics=['accuracy'])
```

```
Epoch 20/20
1198/1198 [==============================] - 5s 4ms/step - loss: 8.0321 - acc: 0.5017 - val_loss: 8.0993 - val_acc: 0.4975
```

NEXT ARCHITECTURE

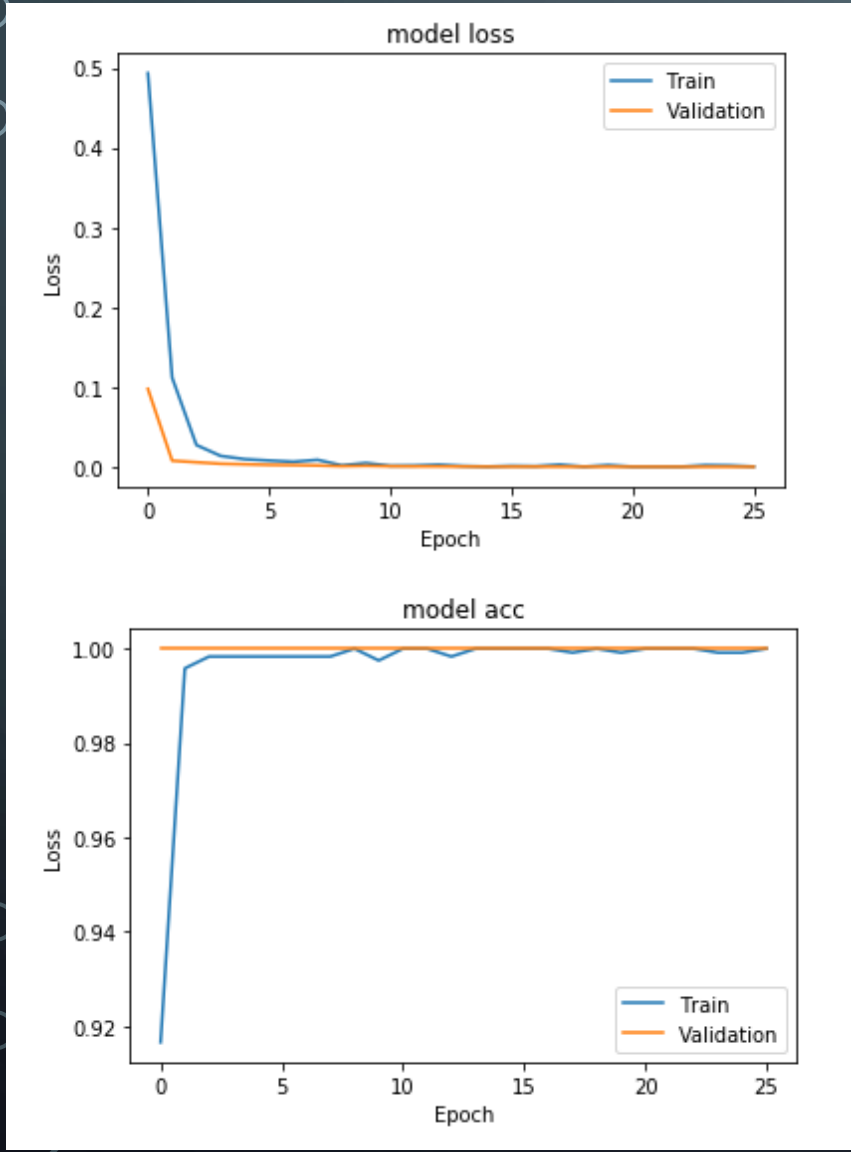
```
model = Sequential([
    Conv2D(64, kernel_size=(2, 2), activation='relu', input_shape=(input_shape),data_format='channels_last'),
    Conv2D(64, (2, 2), activation='relu'),
    MaxPooling2D(pool_size=(2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    Conv2D(64, (2, 2), activation='relu'),

    Flatten( ),
    Dense(2,activation='softmax')
])
model.compile(loss='categorical_crossentropy',optimizer='adadelta',metrics=['accuracy'])
```

```
Epoch 20/20
1198/1198 [==============================] - 66s 55ms/step - loss: 8.0860 - acc: 0.4983 - val_loss: 8.0188 - val_acc: 0.502
```

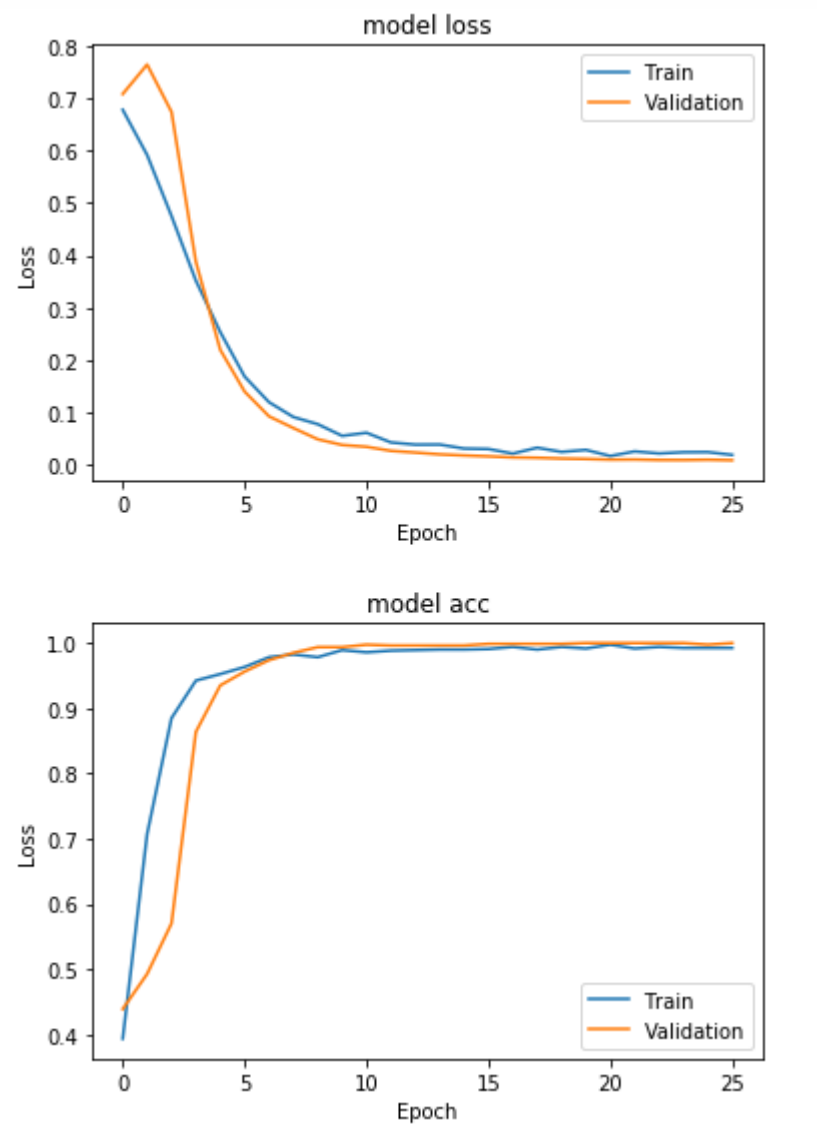
MY ARCHITECTURE

```
model = Sequential([
BatchNormalization(input_shape=input_shape))
Conv2D(8, kernel_size=(3, 3), strides=(2, 2), activation='relu')
Conv2D(8, kernel_size=(3, 3), strides=(2, 2), activation='relu')

Dropout(0.25)

Flatten()

Dense(2, activation='softmax')

])
```

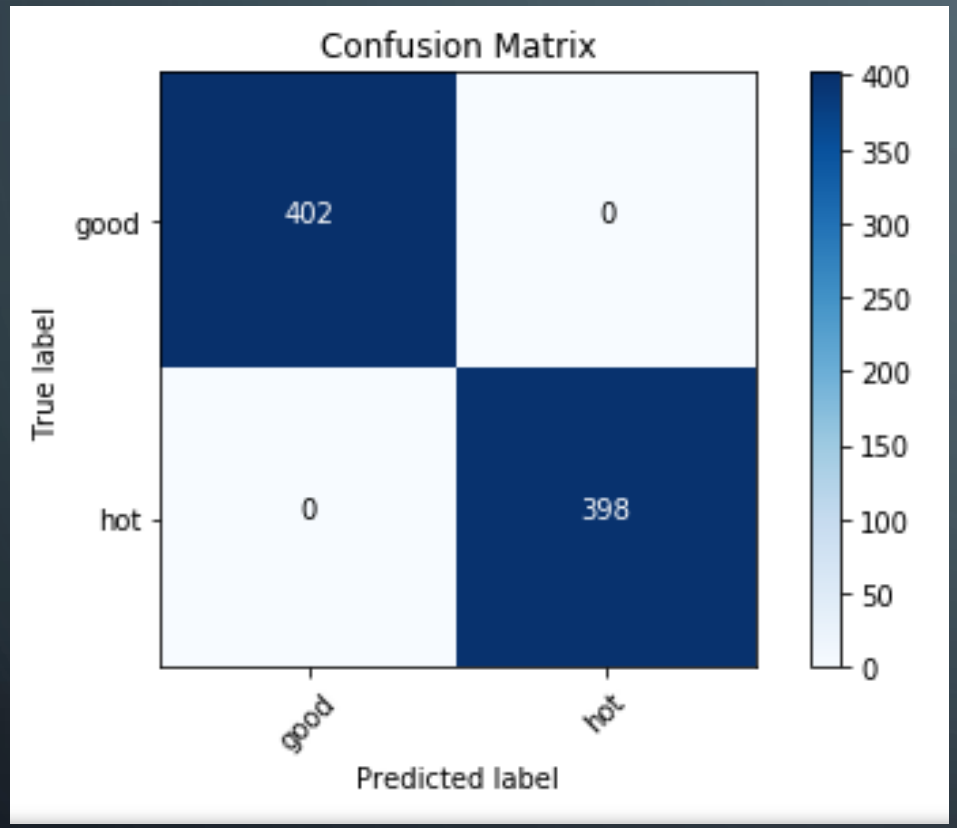
RESULTS



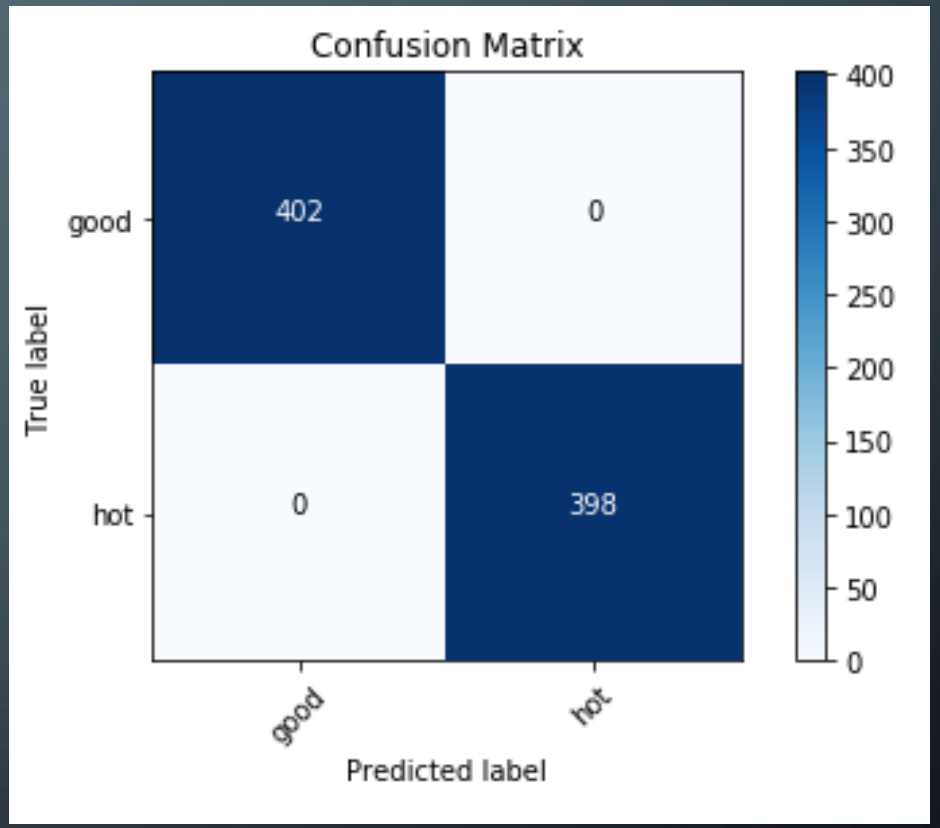
Fixed Location



Random Location



Fixed Location



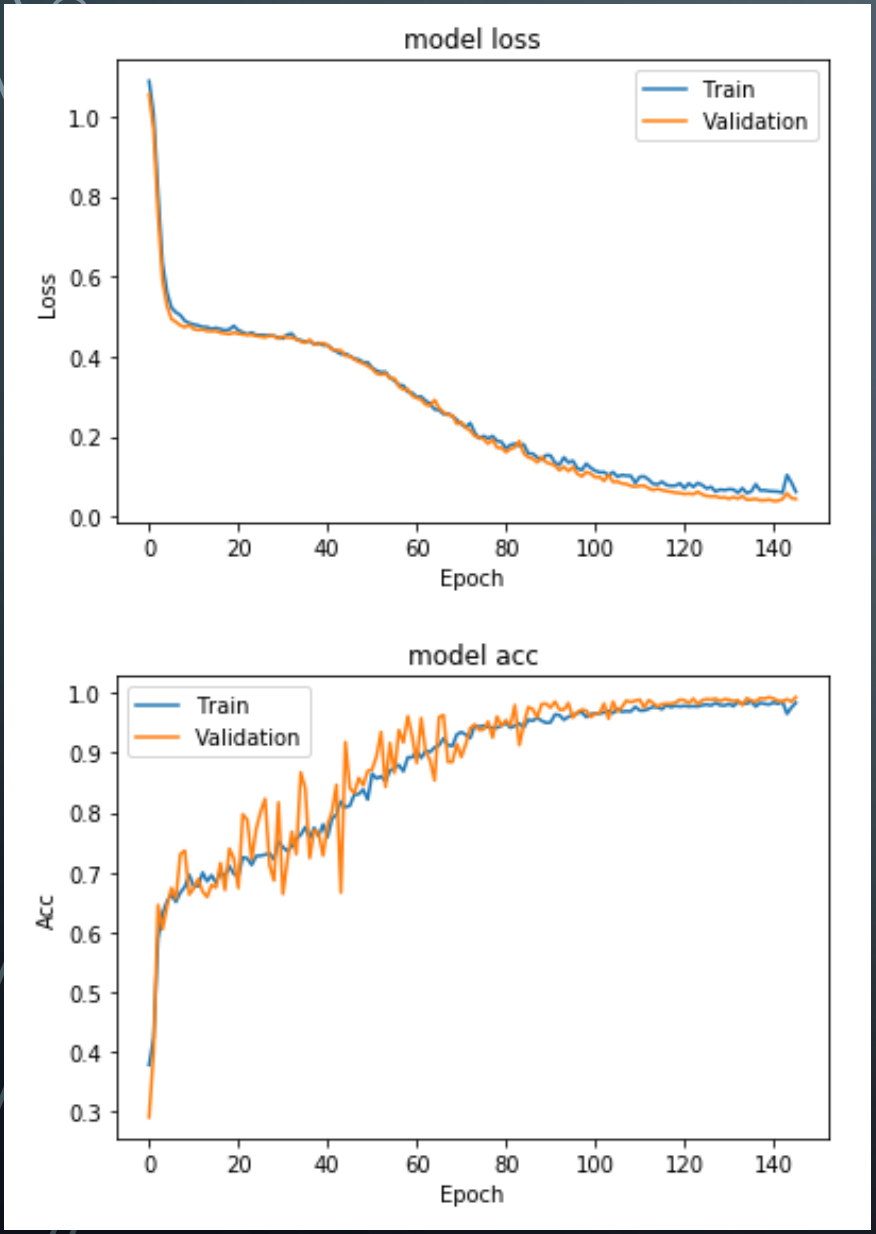
Random Location

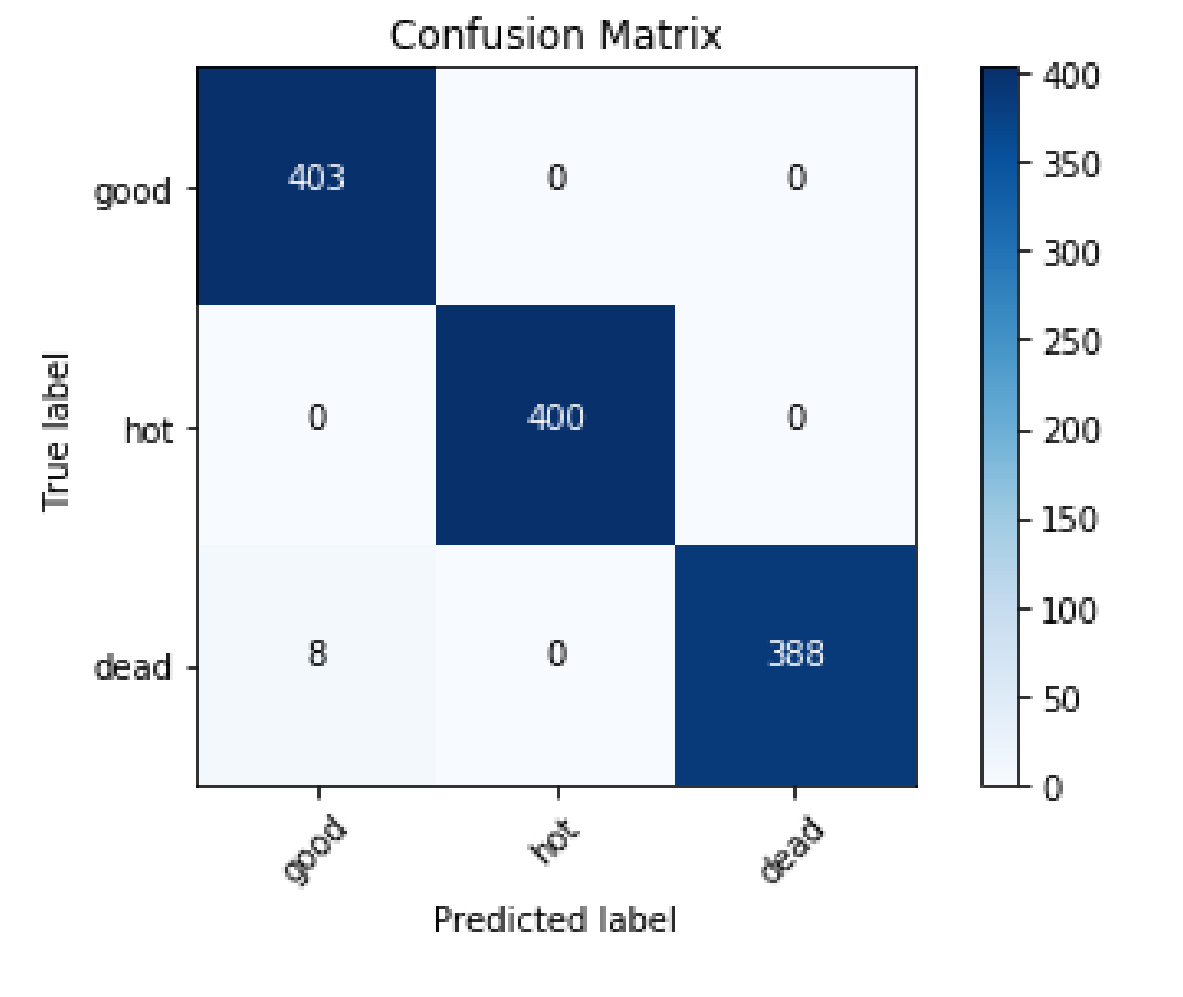
NOW LET'S MAKE THE PROBLEM A BIT HARDER

- With random position
- Multiclass problem (good ,hot, dead)
- Same arch.

RESULTS

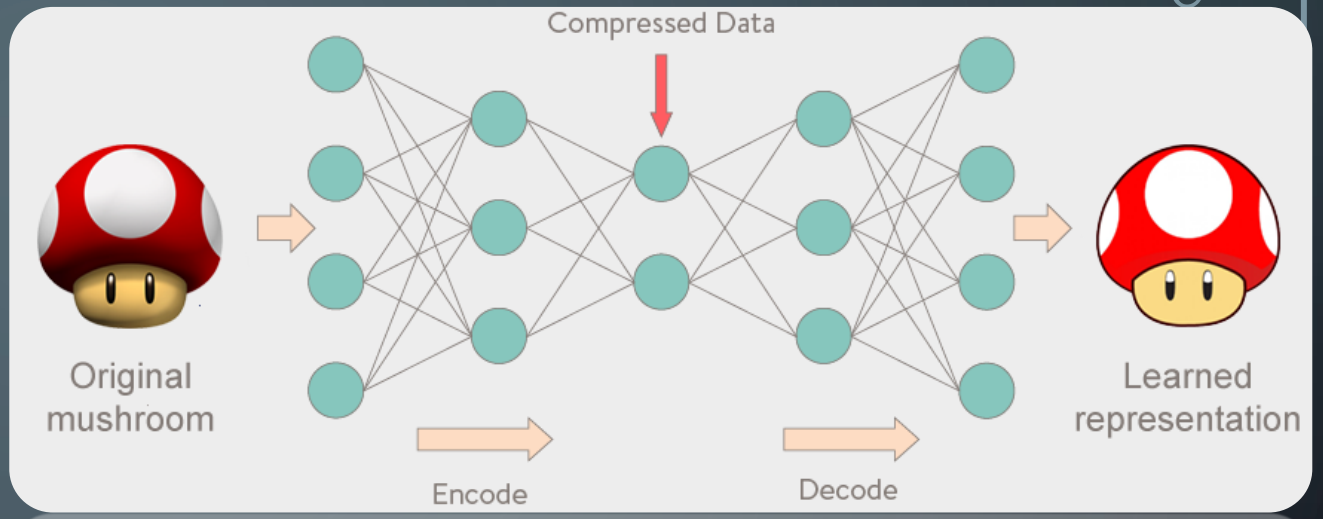
```
Epoch 141/150
Epoch 00141: val_loss improved from 0.04021 to 0.03862, saving model to best_weights.hdf5
 - 1s - loss: 0.0626 - acc: 0.9844 - val_loss: 0.0386 - val_acc: 0.9908
```

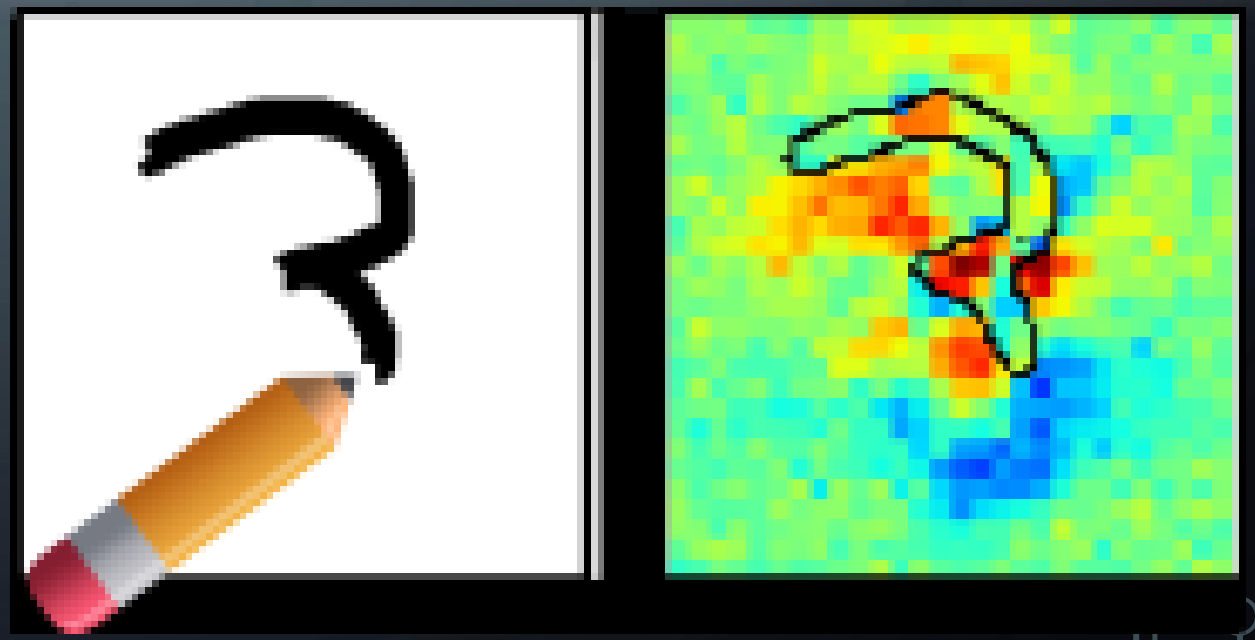




WHAT'S NEXT?

- Plot ROC curves
- Compare it to other models
  - AE
- Why and exactly what is it learning?
- Can we make it work with something more realistic?
  - 1x1 bad region (channel)
  - Can it identify what values should be expected after each lumisection?





CULTURAL EXPERIENCE

- Paris

BACKUP

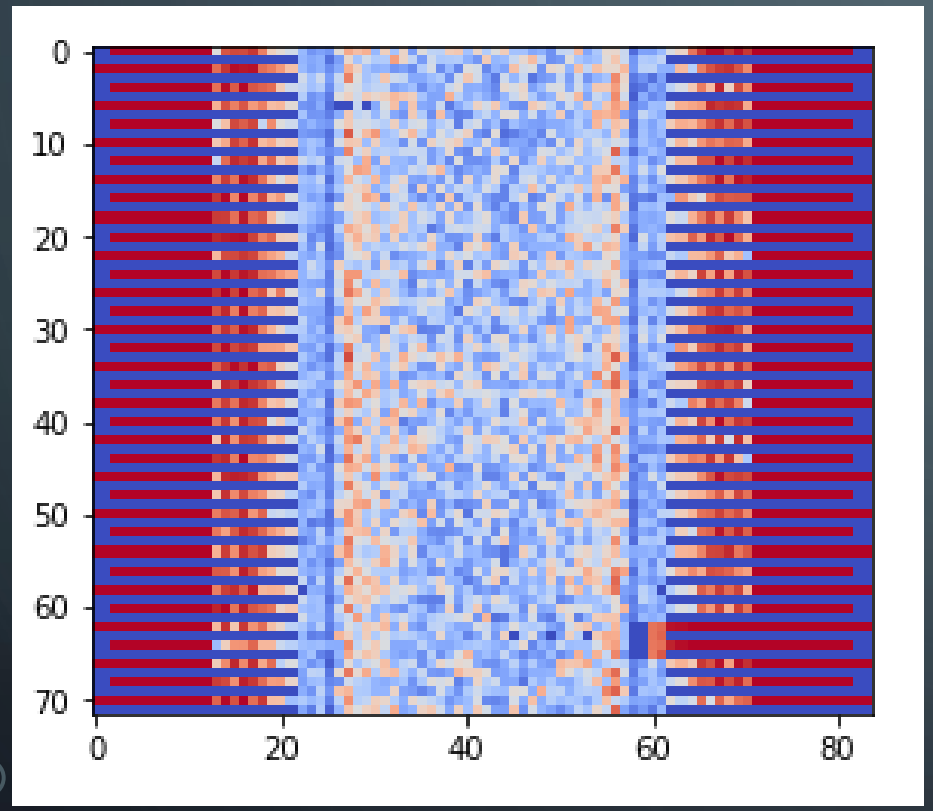
THE CHALLENGE

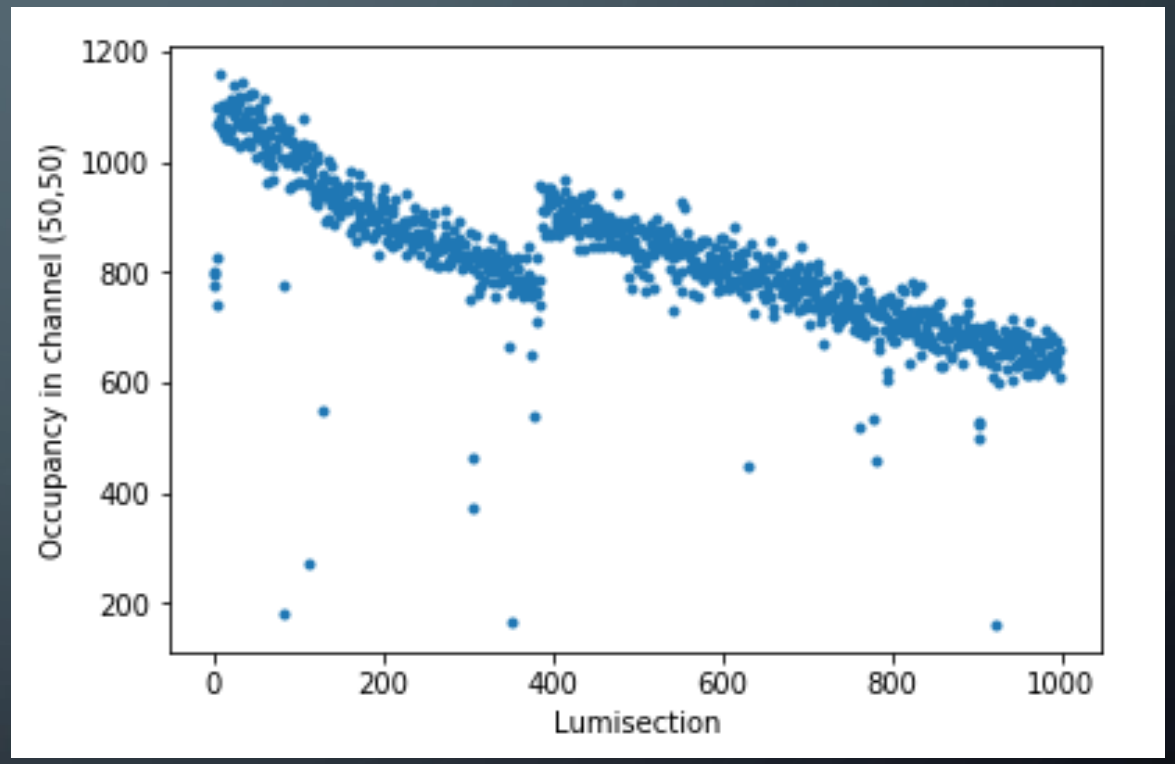
- You have to make sure that it behaves well in order to perform sensible data analysis.
- Reduce man power.
  - Shifters monitor constantly the quality of the data flow.
  - Discriminate between good and bad data to have high purity
  - Build something that helps the people to minimize the time needed to spot problems and save time examining hundreds of histograms
  - Build intelligence that analyzes the data and raises alarms in case of problems. Have quick feedback.

OBJECTIVES AND MY CONTRIBUTION

- The project aims at applying recent progress in Machine Learning techniques to the automation of the DQM scrutiny for HCAL
  - Focus on the Online DQM.
  - Compare the performance of different ML algorithms.
  - Fully supervised vs unsupervised approach.

IMAGES

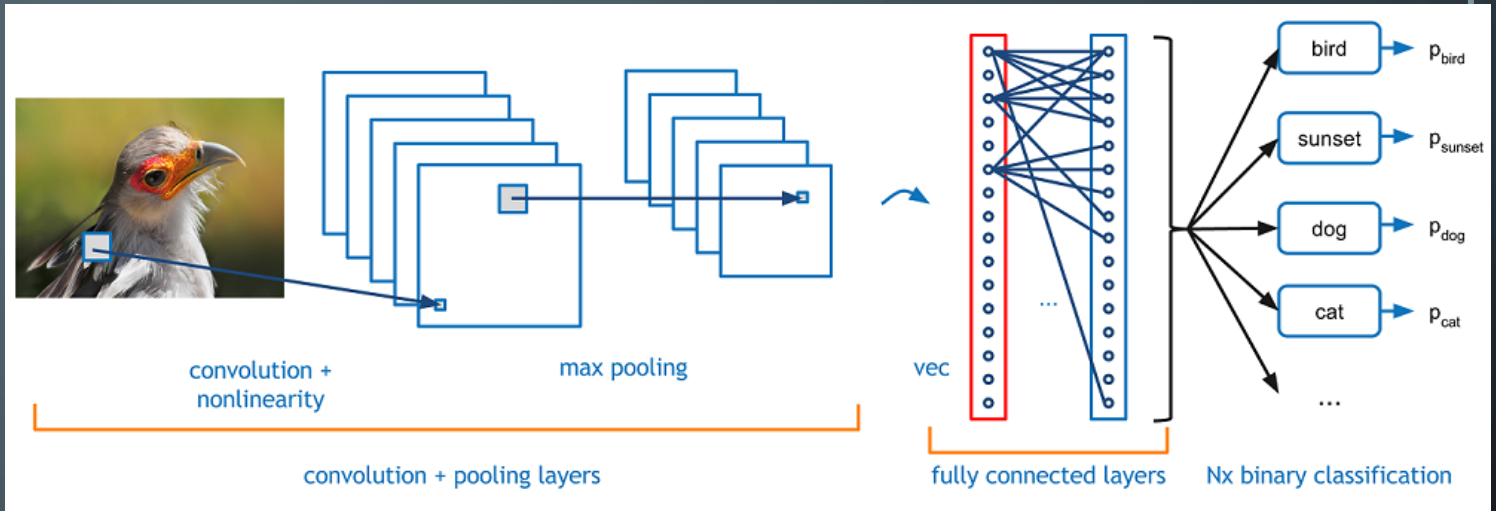

0
10
20
30
40
50
60
70
0
20
40
60
80

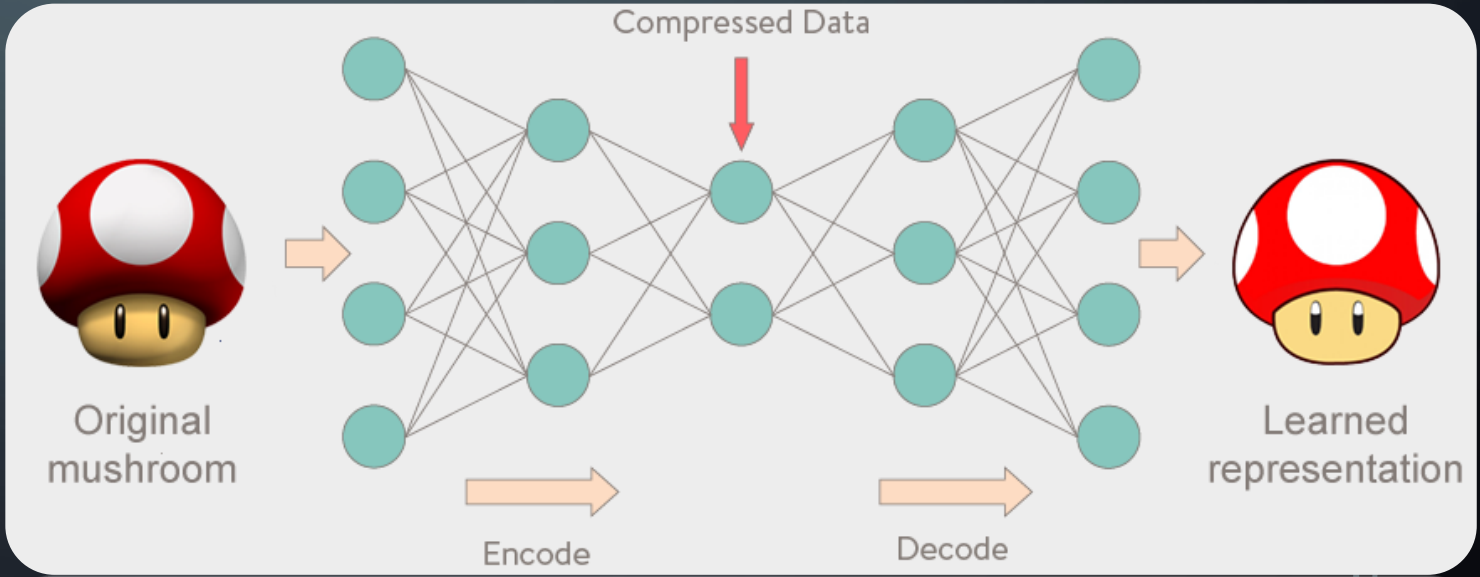

Occupancy in channel (50,50)
1200
1000
800
600
400
200
0
200
400
600
800
1000
Lumisection

WHAT'S NEXT?

- Familiarize with Keras
  - Creation of a model
  - Train it, test its performance
- Compare it to other models
  - CNN
  - AE

CNN





AE