Possible Architectures of Neural Networks for Particle Classification and Reconstruction in SPD ECAL

Andrei Maltsev (JINR, Dubna)

SPD Physics & MC Meeting #21

23.03.2022

Motivation

The ECAL reconstruction algorithm must:

- be fast enough to be used in the online filter
- separate clusters from π^0/γ to reject background events
- be robust against ECAL miscalibrations
- (ideally) be interpretable

ECAL: ~25k cells → need parallelizable algorithms → convolutional neural networks (CNN) is a natural choice

Convolutional neural networks

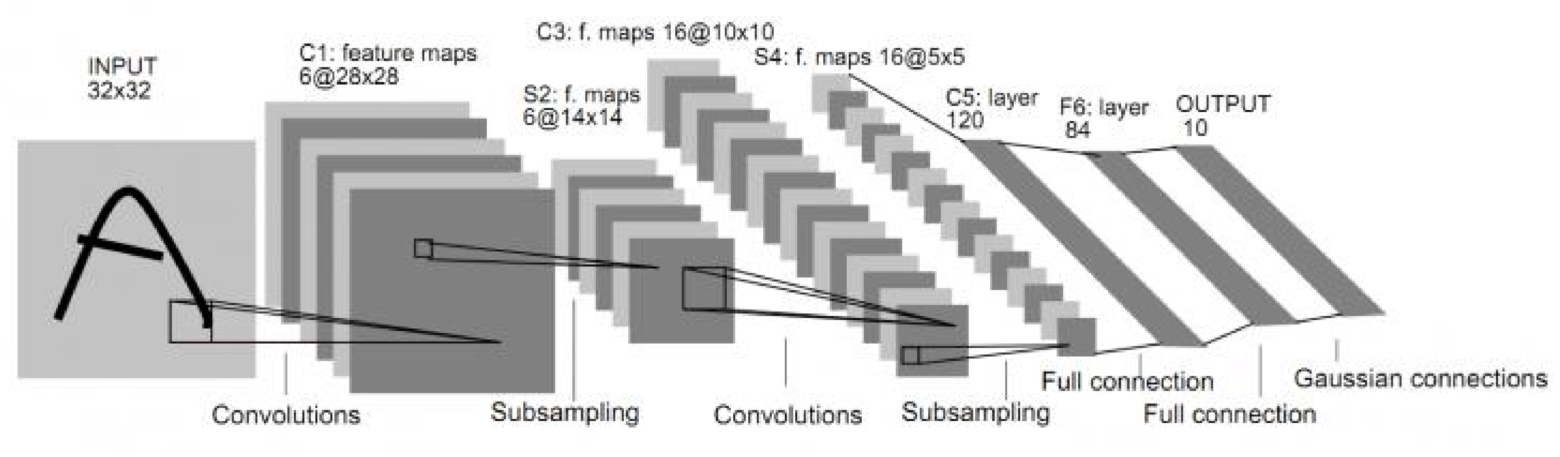
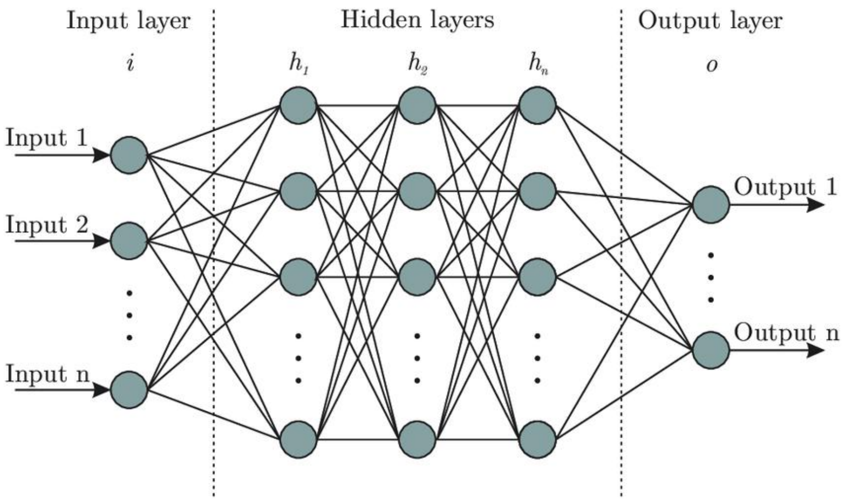


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.



Gradient-based learning applied to document recognition, Y.Lecun et al.

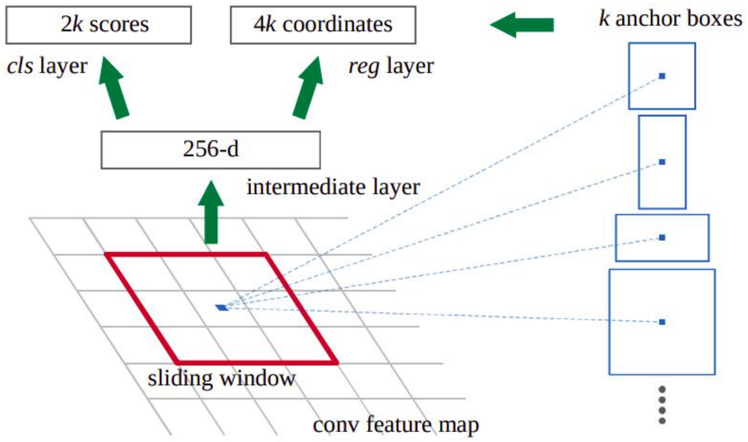
Facundo Bre et al.

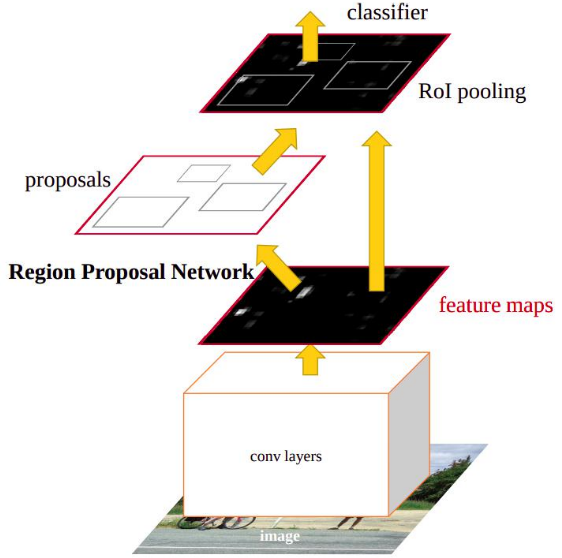
- “Sliding” window (kernel), here 5x5 pixels; multiple kernels correspond to multiple features
- basic features in the first layers, more complex features in the latter layers
- for regression (continious output)/classification (discrete output), the final layer is flattened
and fed as input to multi-layer perceptron or more complicated networks
- optimizing weights in kernels though training and backpropagation

Understandable how to process one image. But how to select multiple images (clusters) in e.g. ECAL?

Reconstruction network

Idea inspired by [Faster R-CNN](): two networks:





1) Region proposal network:

for each region, predict:

a) rough bounding box (could be omitted)

b) whether it's an object

originally, “anchors” of different aspect ratio were used, but could be omitted

2) Object detection network:

proposals → RoI pooling (could be omitted?)

predicts:

a) precise bounding box → energy, position

b) object class → PID

postprocessing: non maximum suppresion (to remove duplicate objects)

As a first step, try to use only PRN, but with PID and energy/position as outputs

Tests of the reconstruction layers

Convolutional layers and region selection replaced by simple search of local maxima

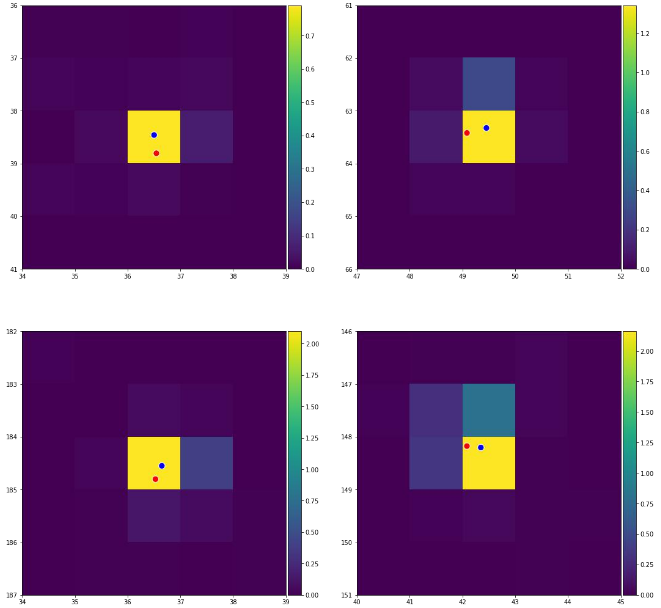
Network setup:

$N_{neurons}$ = [25(input), 50, 30, 10, 3/1(output: x/y/e)]

- Adam, lr = 0.01, β=(0.9, 0.999), 1400 iterations
- ~17k input 5x5 images, energies: 1,2,3,...,8 GeV, angles: [0,20]°
- MSELoss for position, modified loss function for energy resolution: taking into account energy resolution 4% $\oplus$ 6%/sqrt(E)
- two options: 3 outputs (x/y/e) or 3 separate networks for x/y/e

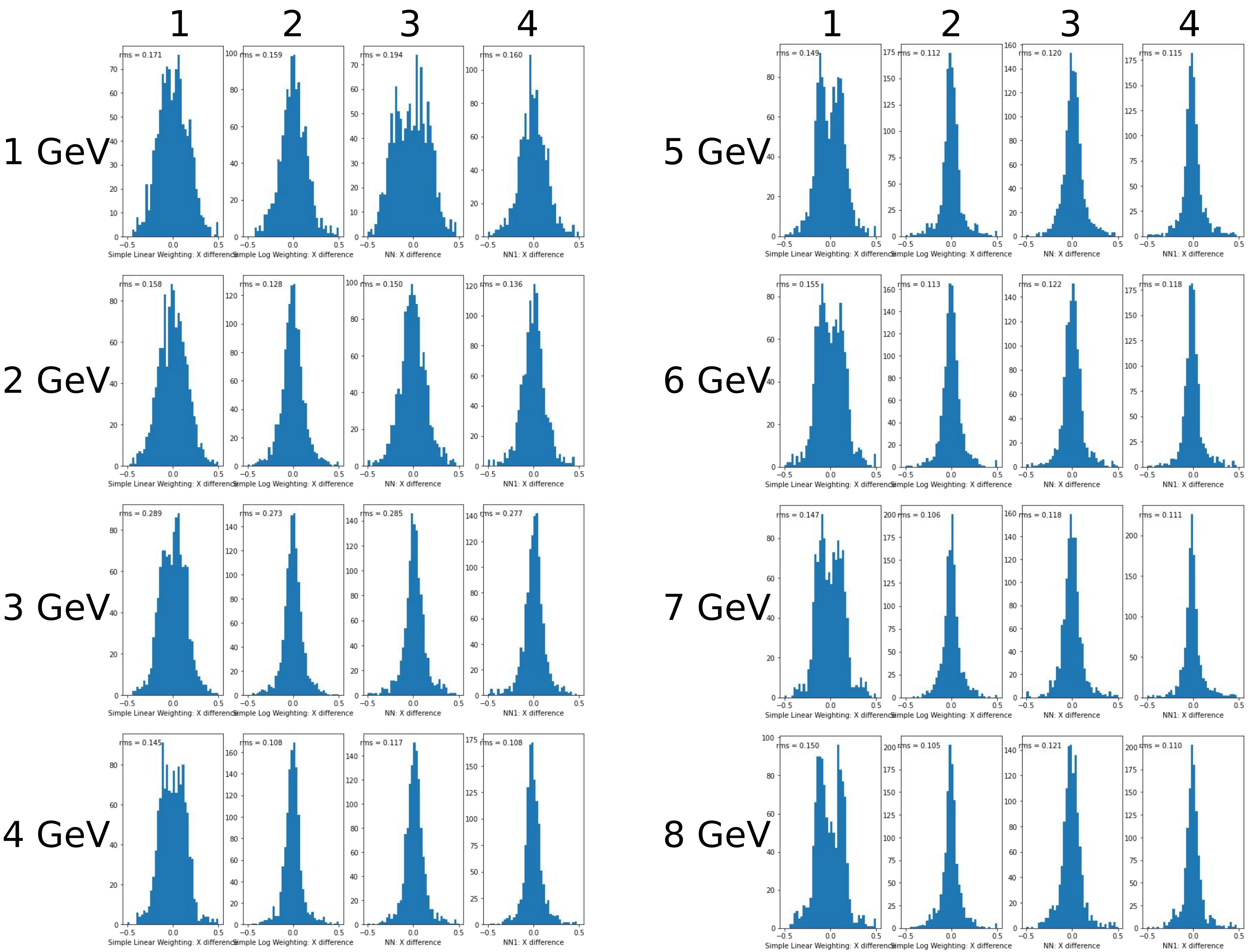
Performance compared to weigted average with linear or log. weights (cutoff parameter for log.weighting optimized for each energy)

hit position
weighted average(linear)



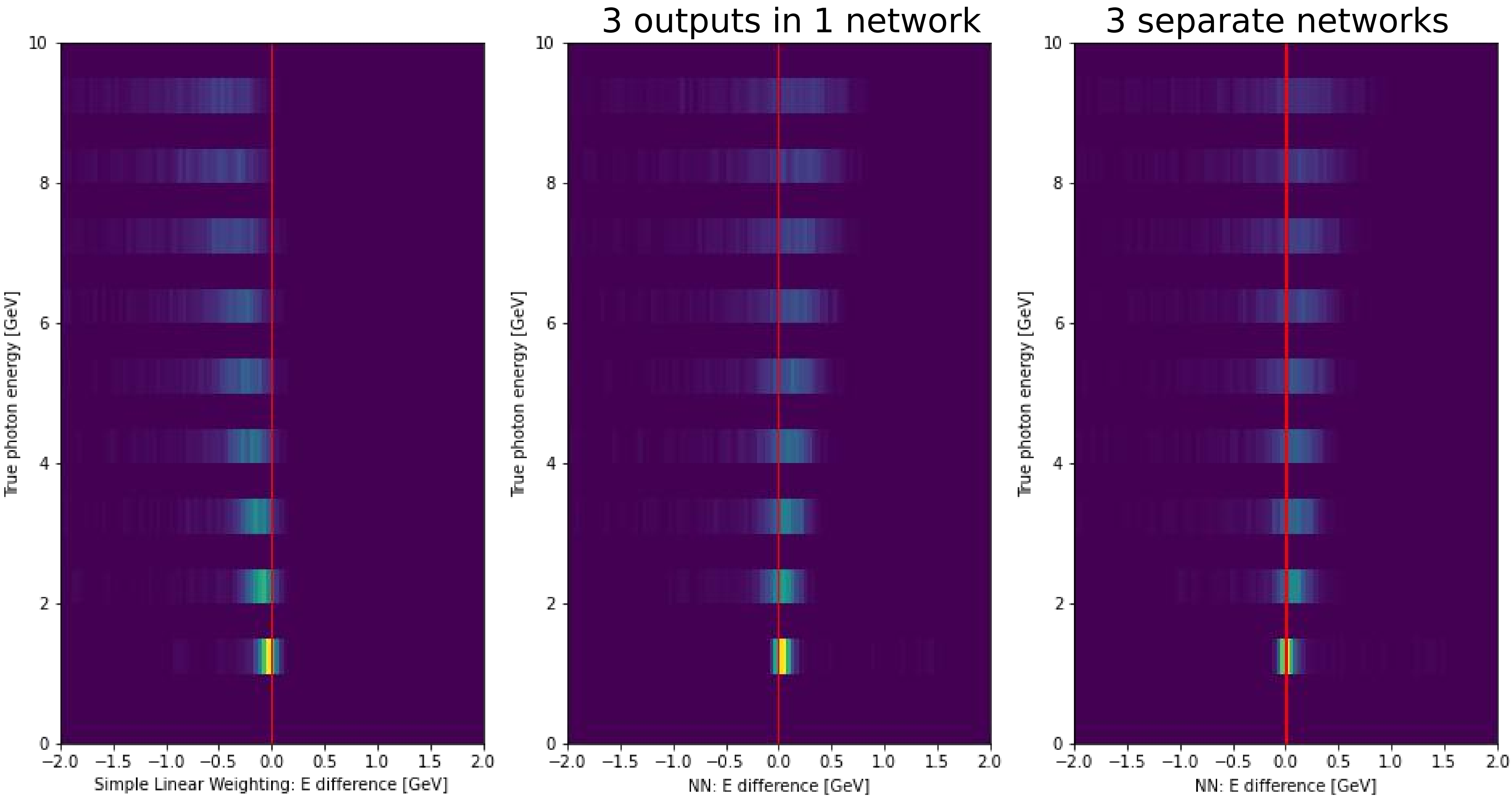
each bin = 1 cell

Performance of reconstruction network
Resolution in X(Y/Z/φ) variable in terms of cell size



Colums:
1) linear weighting
2) log.weighting
3) one network with 3 outputs
4) 3 separate networks

Performance of reconstruction network
Energy resolution



Conclusions and outlook

- Script to produce calorimeter array from SPDROOT output was set up
- Training of a simple network works for photons under small angles — the procedure works in general
- Next step: try implementing the CNN with common conv layers and separate layers for PID and position determination