LCD Single Particle ($e^{\pm}$vs. $\pi^{\pm}$) Identification

Kaustuv Datta , Jayesh Mahapatra, Maurizio Pierini, Jean-Roch Vlimant

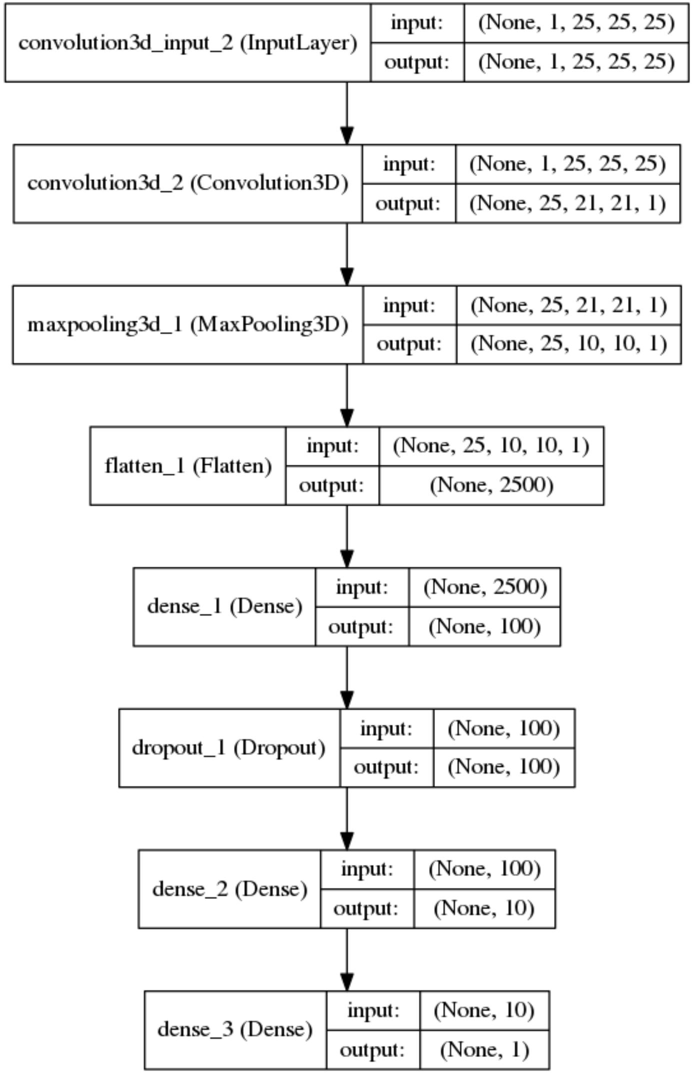
Workflow

- Data first generated as .root files with information of energy deposit on calorimeters, subsequently converted to HDF5 format
- Using a script to find the barycenter for events in HCAL and ECAL and create files for training
  - Extracted (ix, iy, iz) – indices in sub-detector that are used to number cells in the calorimeters, (X, Y, Z) – absolute spatial coordinates and energy for each event
  - Calculated weighted average in terms of energy to find barycenter of event in ECAL, get a 25x25x25 array of energies around those coordiantes
  - Pass Y, Z coordinates of ECAL barycenter to HCAL and get the surrounding 5x5x60 array of energies
  - HCAL and ECAL event saved as “images” using two separate keys in
  - “Target” key created to include information of the particle id, energy of hit, and momentum 3-vector
- Using data generator to feed in data to networks, with the capability to dynamically vary batch sizes
  - Data generator is useful for our large dataset – instead of loading everything on to memory only one file is open at a time
  - Events fed in individually till batch size is satisfied

Network Topologies

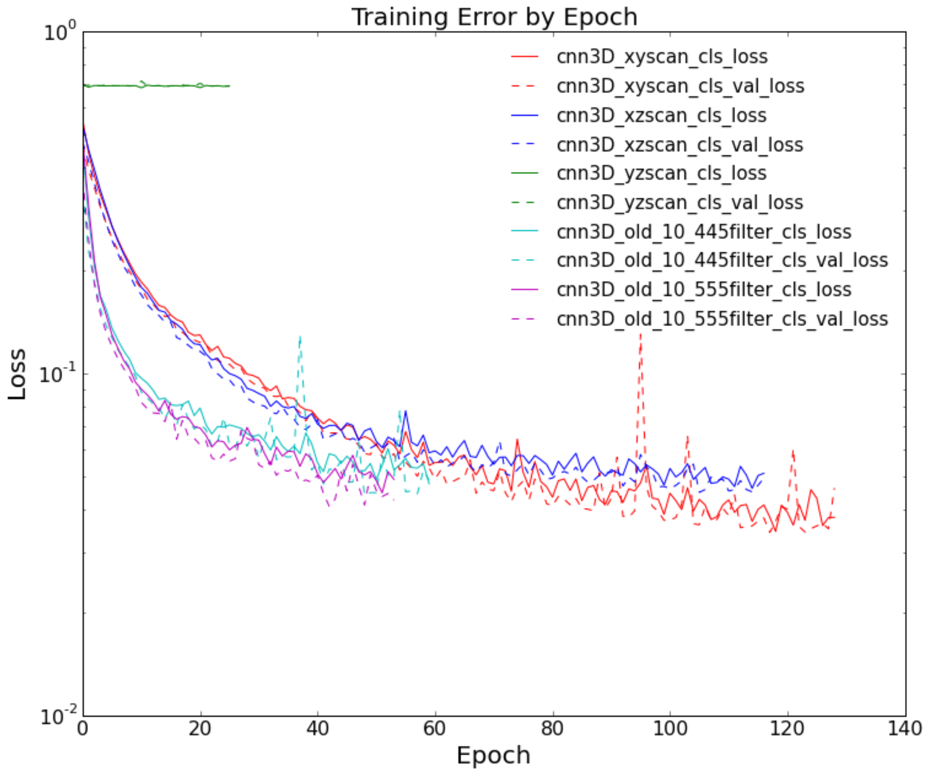
4 different 3D ConvNets:

- XY Scan (Full depth on Z axis), filter dimensions - (5,5,25) (summary of network shown alongside)
- XZ Scan (Full depth on Y axis, filter dimensions - (5,25,5)
- YZ Scan (Full depth on X axis), filter dimensions - (25,5,5)
- Other models (filter dimensions - (5,5,5) & (4,4,5).

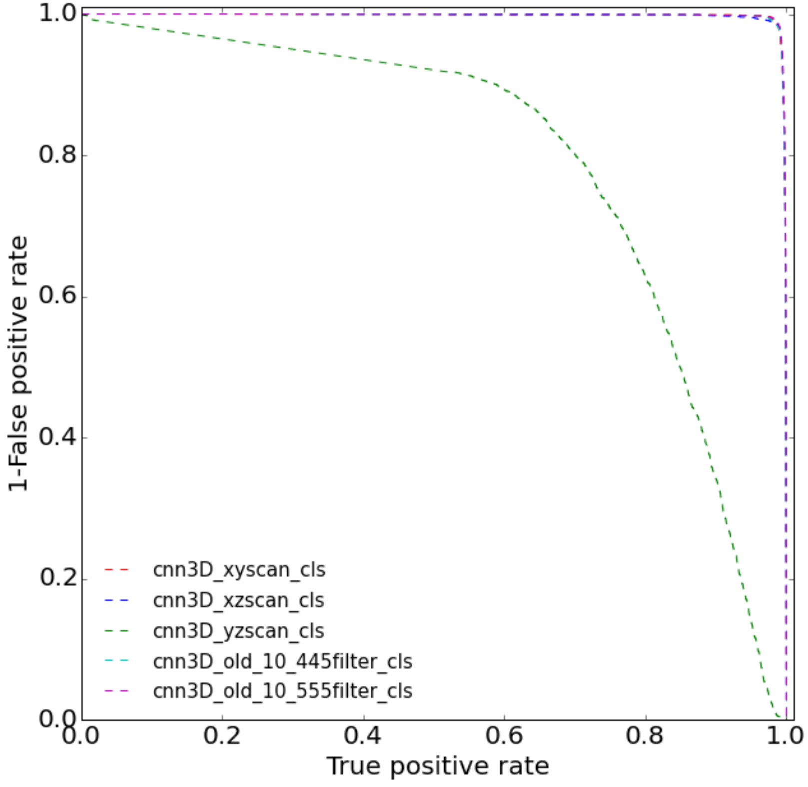


Loss Curves

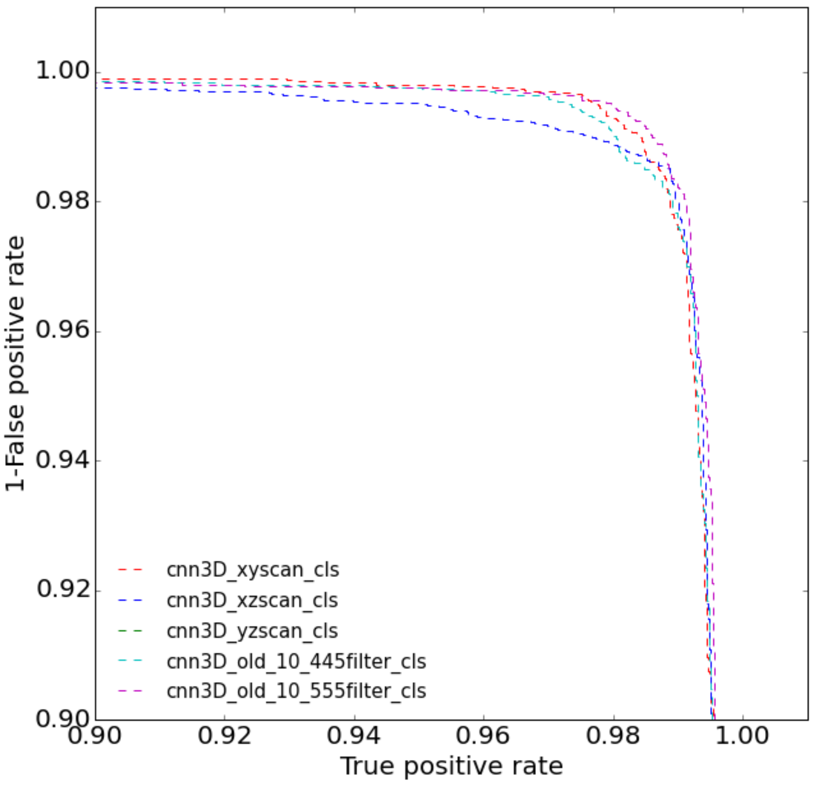
- Models trained with early-stopping (patience = 10 epochs)
- The xy,xz scan models train over 120 epochs.
- The old models with filter shapes of (4,4,5) & (5,5,5) train till 60 epochs and have final training loss nearly equal to the XZ Scan
- The yzscan model does not train well and hardly goes near 40 epochs
- The training losses also differ by a lot



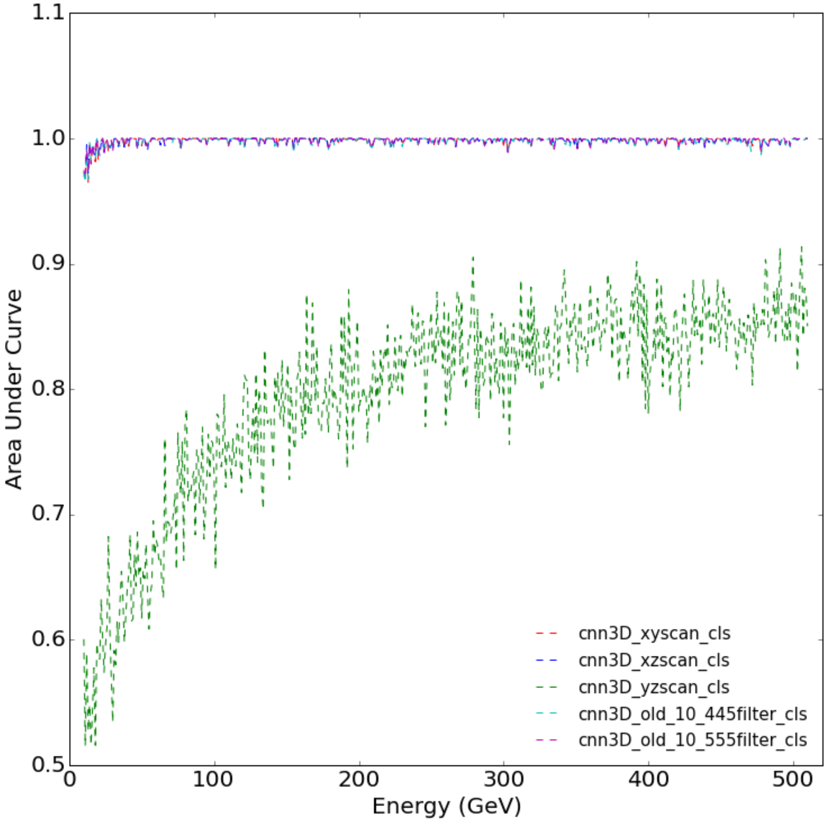
Results: ROC Curve



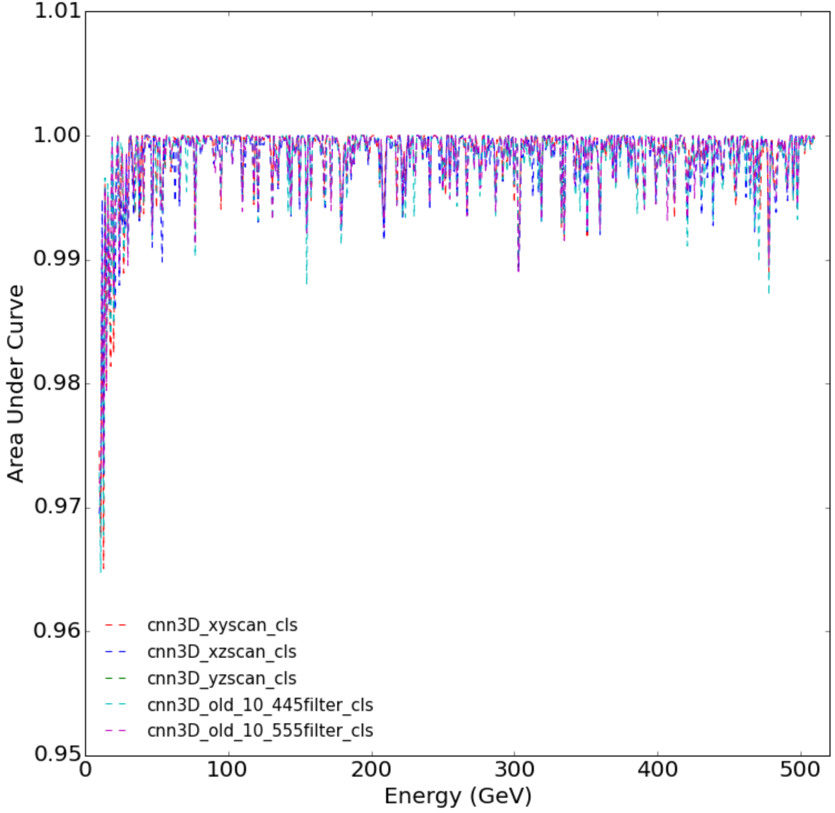
Zoomed in curve at 90% true positive rate



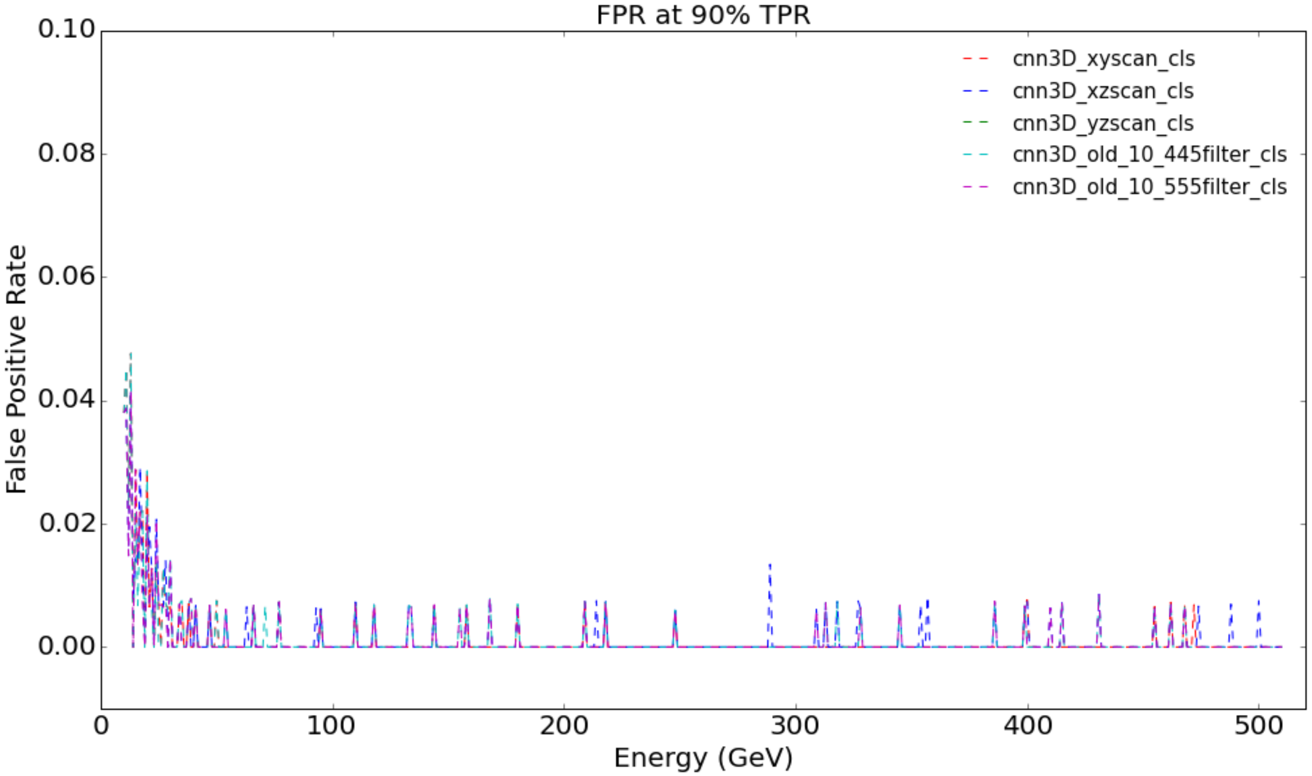
Results: Area Under ROC Curve



Zoomed in (but requires rebinning for better understanding of plot)



Results: Variation of FPR with Energy



Discussion

- Classification of electrons vs. charged pions may be too easy a problem for topologies used
- Spread of performance is being studied currently with 5-fold cross-validation, for the xy, yz, xz-scan models
- Need to look performance as a function of the amount of data used in training (given a fixed test sample) to get idea about over/under complexity of models
- Previous attempt at photon vs. neutral pion discrimination (using both ECAL and HCAL data) with branched convolution topologies gave unsatisfactory results – this needs to be attempted again with a revised approach
- Development of mpi-learn class to function in tandem with custom data-generator for networks to parallelize learning over multiple GPUs – will allow to tackle photon identification problem which requires more complex topologies, in addition to speeding up any learning in general

Future Work

- With around 20 trainings for each model we can observe if there is a Gaussian-distributed performances or not
- Try different training splits (0.01, 0.1, 0.25, 0.5, etc.) for data – all aforementioned models were trained on 0.7 of the available data, 0.2 was validation set and 0.1 was test set

Backup Slides

Current Dataset (e vs ch. Pi)

- Information of showers from single particles hitting ECAL surface
- Data stored as energy deposits per pixel around barycenter, in barrel region of calorimetry over a flat energy spectrum between 10-510 GeV

Photon identification slides, as backup info

Problems we are addressing with this project

- Energy Regression
  - Using neural networks (NN) to carry out regression over a discrete energies(10-109GeV) and, more recently, continuous spectrum of energies (10-500 GeV)
  - Relevant inputs, as per current dataset, provided as simulated photon and pion energy showers (events) recorded on highly granular Electromagnetic and Hadronic Calorimeter (ECAL, HCAL henceforth) geometries
  - Relevant output – energy of an identified photon hit

- Particle Classification
  - Identification of particles, as either photons (signal) or pions (background), in ECAL and HCAL events
- Simultaneous Regression and Classification
  - Tackling both regression and classification using branched network topologies

Experimental Setup

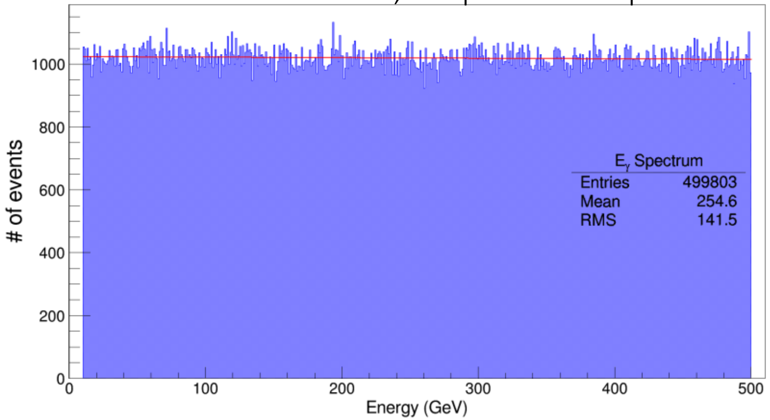
- Titans machine at Caltech
  - Two available NVIDIA GTX TitanX
- CSCS (Swiss National Supercomputing Centre) GPU cluster
  - Multiple available NVIDIA Tesla K20X
  - Slower than TitanX's, leading to longer training times, but multiple jobs can be run at same time due to good distribution of workload
- All work carried out using the Keras deep learning library for Python, running on Theano backend
- Some prototyping and inference work done on titans before and after extensive training on CSCS
- LCD Datasets generated (by Maurizio) consisting of single-particle showers in barrel of 3D geometry of high-granularity Linear Collider Detector ECAL, and HCAL

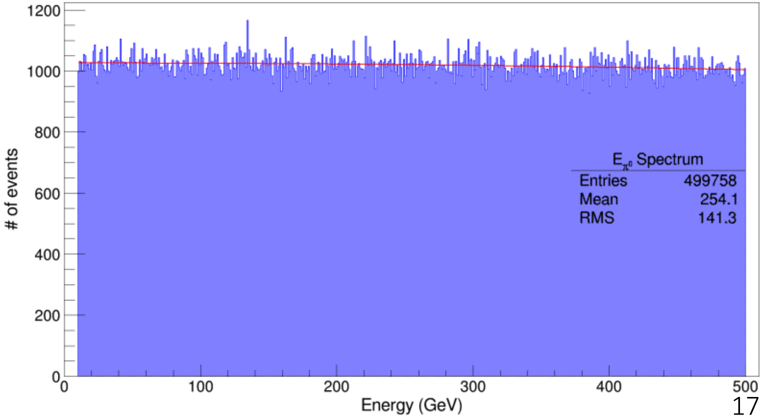
Workflow

- Data first generated as .root files with information of energy deposit on calorimeters, subsequently converted to HDF5 format
- Using a script to find the barycenter for events in HCAL and ECAL and create files for training
  - Extracted (ix, iy, iz) – indices in sub-detector that are used to number cells in the calorimeters, (X, Y, Z) – absolute spatial coordinates and energy for each event
  - Calculated weighted average in terms of energy to find barycenter of event in ECAL, get a 24x24x25 array of energies around those coordiantes
  - Pass Y, Z coordinates of ECAL barycenter to HCAL and get the surrounding 4x4x60 array of energies
  - HCAL and ECAL event saved as “images” using two separate keys in
  - “Target” key created to include information of the particle id, energy of hit, and momentum 3-vector
- Using data generator to feed in data to networks, with the capability to dynamically vary batch sizes
  - Data generator is useful for our large dataset – instead of loading everything on to memory only one file is open at a time
  - Events fed in individually till batch size is satisfied

LCD Dataset

- Previous dataset included events in ECAL, at discrete energies over range of 10-109 GeV, and only the regression problem was addressed on this dataset
- Current dataset contains information of energy deposits in barrel region, of ECAL and HCAL, over a flat energy spectrum between 10-500 GeV, from single particles hitting ECAL surface and subsequently showering
  - Including HCAL information is important to image the tail of hadron showers, and see more of higher energy electron showers continuing into the HCAL from the ECAL
  - Increased generalization of methodology
  - More data is extremely important from the deep learning' point of view
  - Total of about 500,000 photon and pion events each

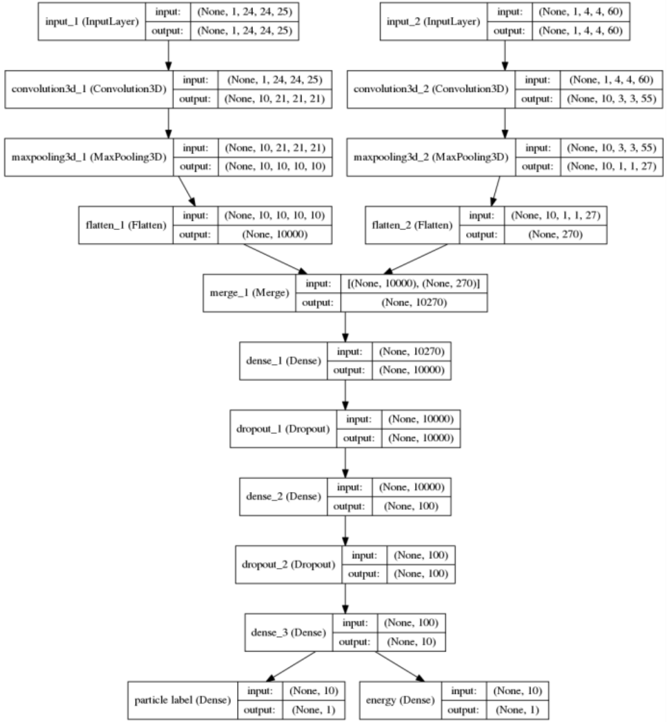




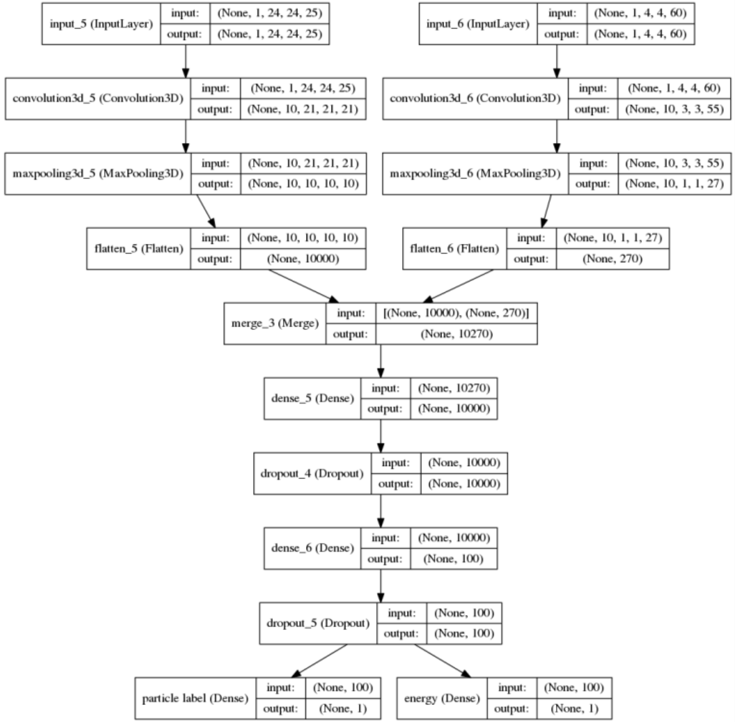
Data Pre-Processing for Training

- Initial dataset was 50 photon and 50 pion files, each with a flat energy spectrum, of about 10,000 events per file
- Photon and pion files were merged and 100 files of 10,000 events each were created
- Files were shuffled 500 times, opening two random files at any given time and using the same random seed to shuffle events to not lose correspondence between images and targets
- Regression+Classification and Classification only models were trained on these shuffled files
- Regression only models were trained on files containing only photon events

Hybrid network topologies explored

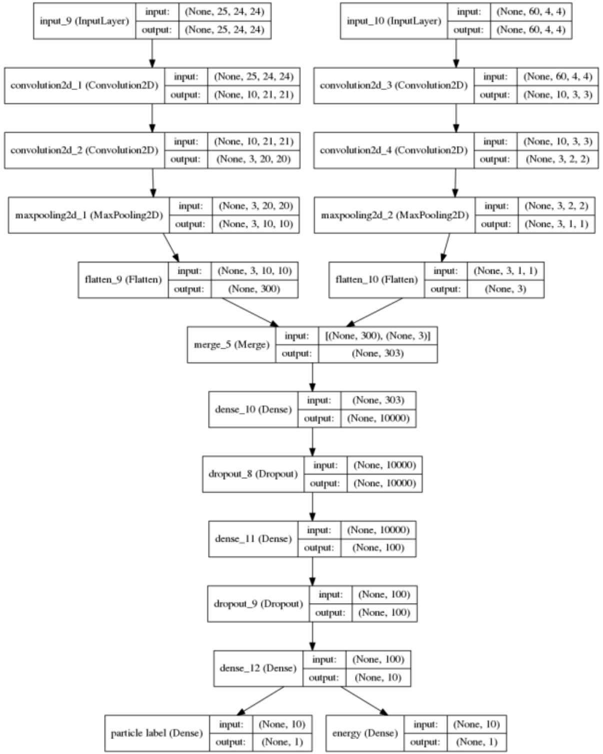


BCNN1_regcls

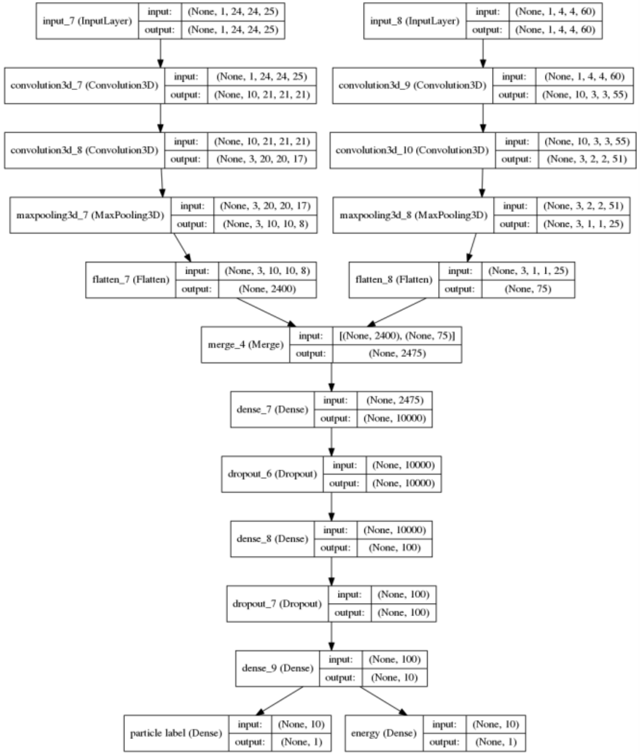


BCNN3_regcls

Network topologies (contd.)

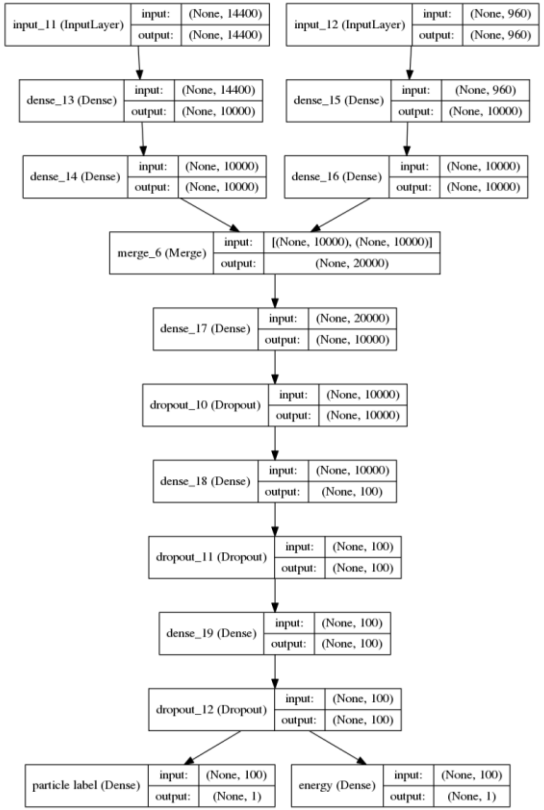


BCNN4_regcls



BCNN5_regcls

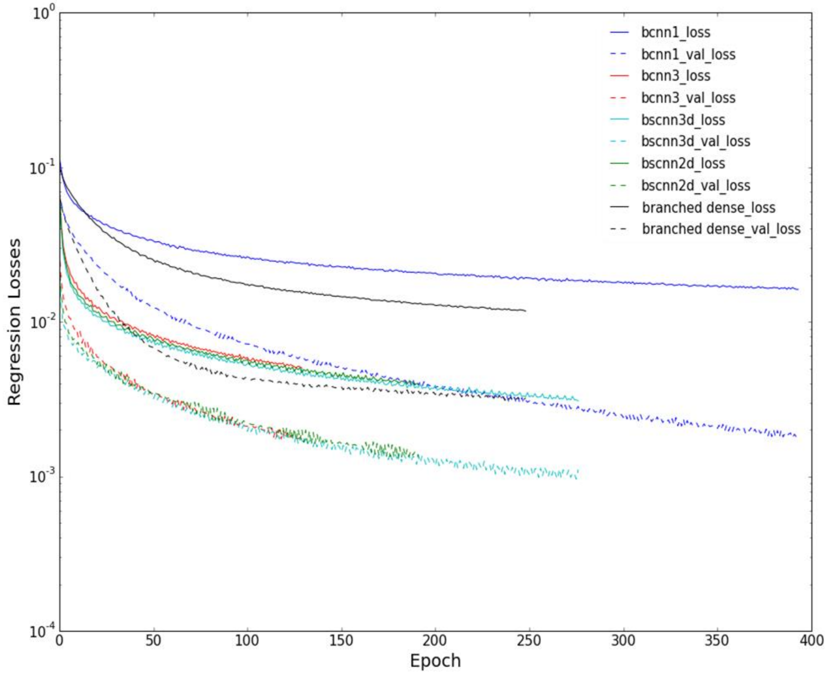
Network topologies (contd.)



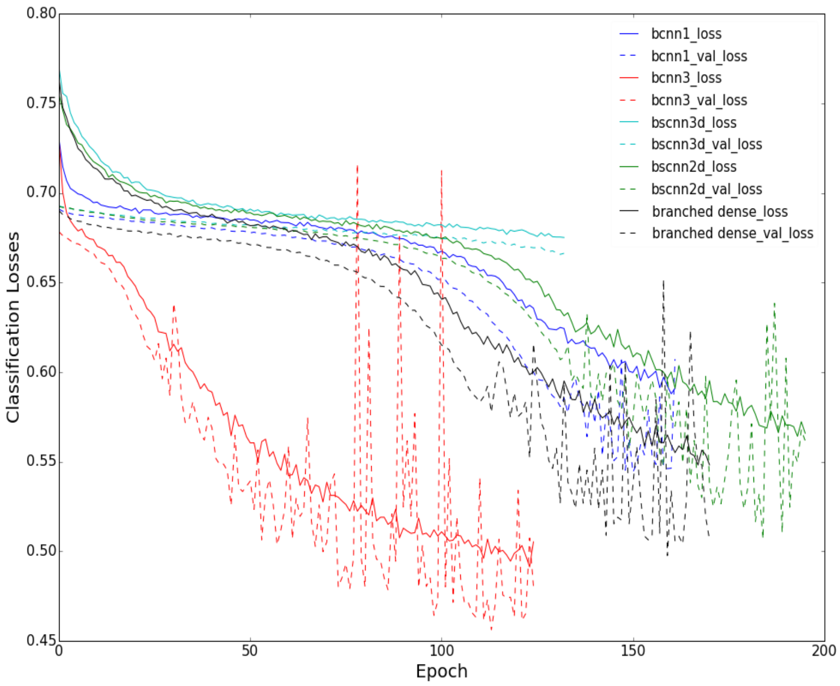
dense1_regcls

Training Losses (really funky loss plots, need to try again!)

Regression model losses

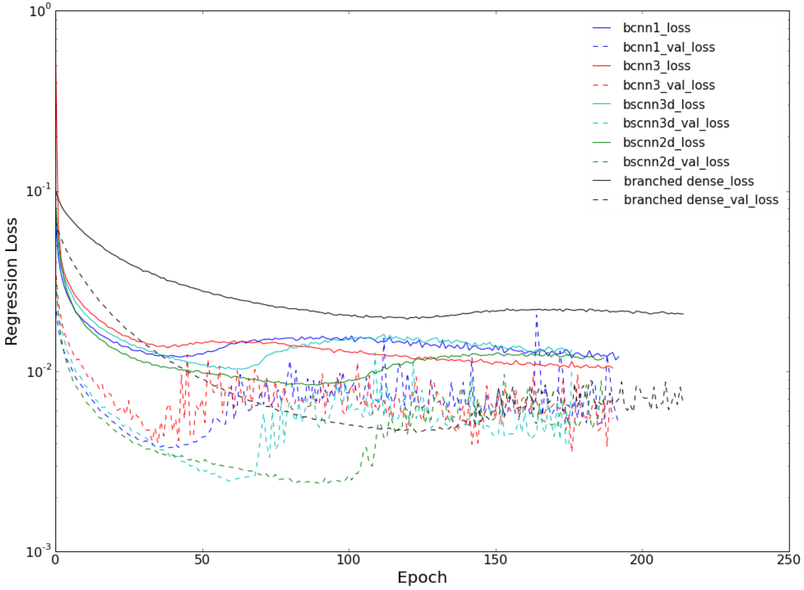


Classification model losses

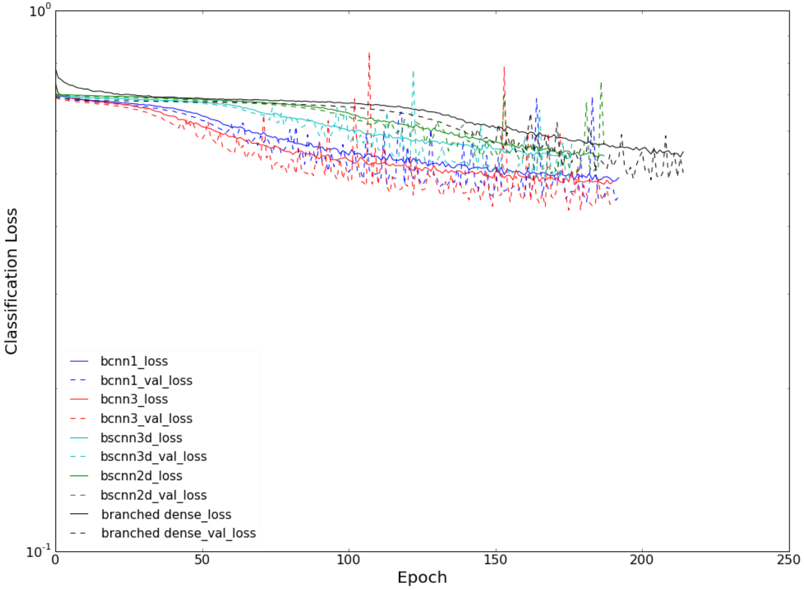


Training losses

Regression+Classification model's regression loss

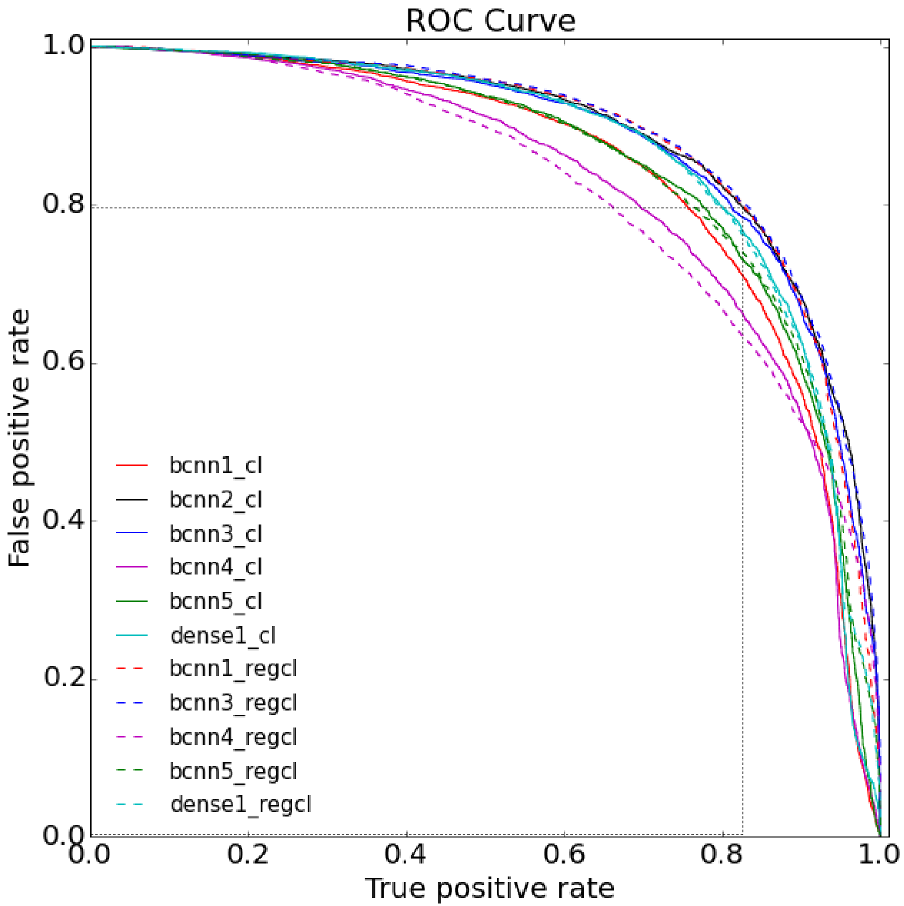


Regression+Classification model's classification loss



Initial results

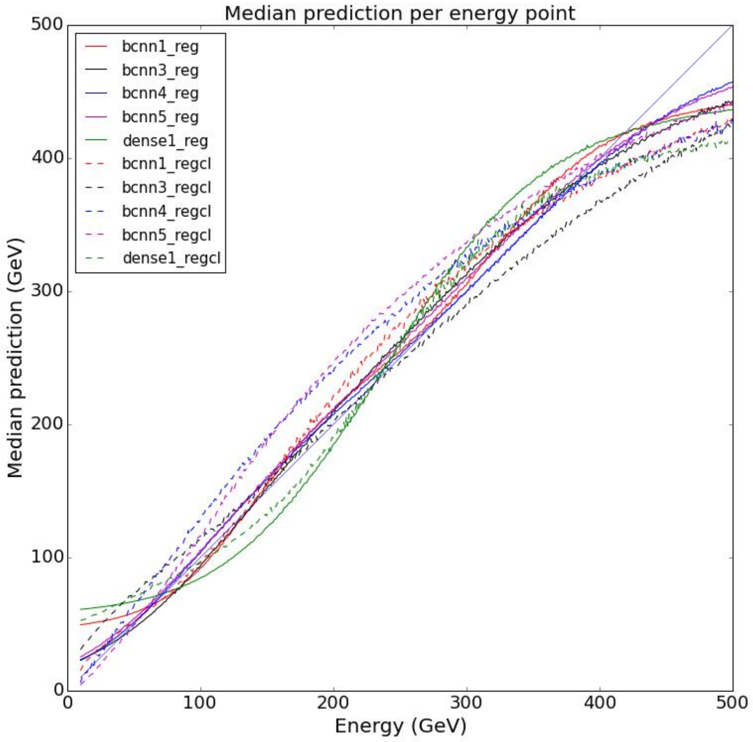
Classification performance

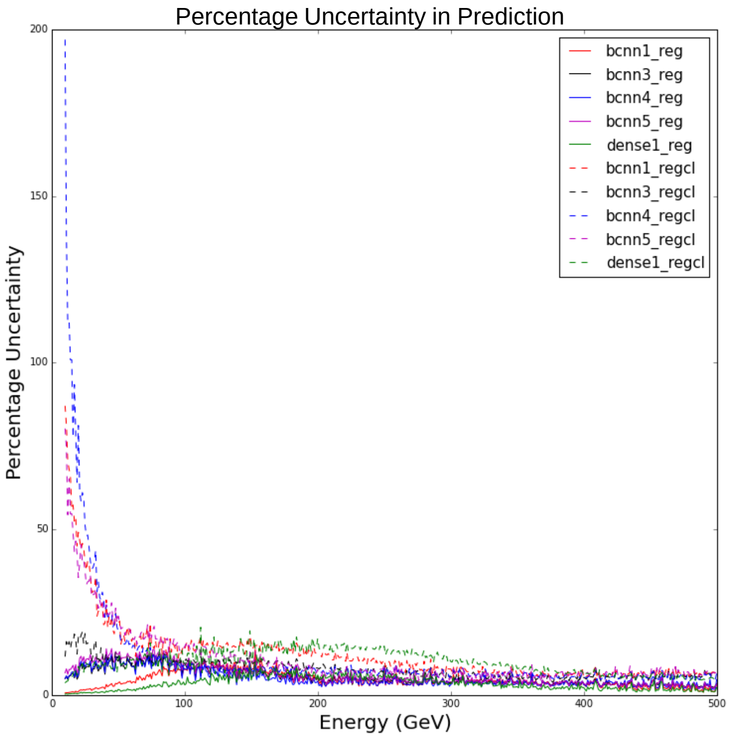


ROC Curve comparing the binary classification performance of the Classification (cl) and Regression+Classification (regcl) topologies trained on CSCS

- Best signal to noise ratio achieved: ~82% efficiency at ~20% false positive.
- Needs much more tweaking

Regression performance as a function of energy





Comparison of energy prediction performance of Regression (reg)
and Regression+Classification (regcl) topologies trained on CSCS

Next Deliverables and Work Going Forward

- Develop new topologies to achieve better performances
- Develop new classes to test regression performances over flat energy spectrum
  - Plot median relative error, zscore, percentage uncertainty of predictions and residual
- Develop new early-stopping classes
  - Some models definitely seem to not be training for enough epochs, especially the Regression+Classification models
  - Keras' vanilla EarlyStopping does not do a good enough job, since it monitors total validation loss of both outputs for dual output models
  - New EarlyStopping would need to monitor validation losses of both outputs, find a good balance to prevent over-fitting one branch with a higher magnitude of loss at the cost of under-training the other with smaller loss values