Regression: DNN / CNN Studies

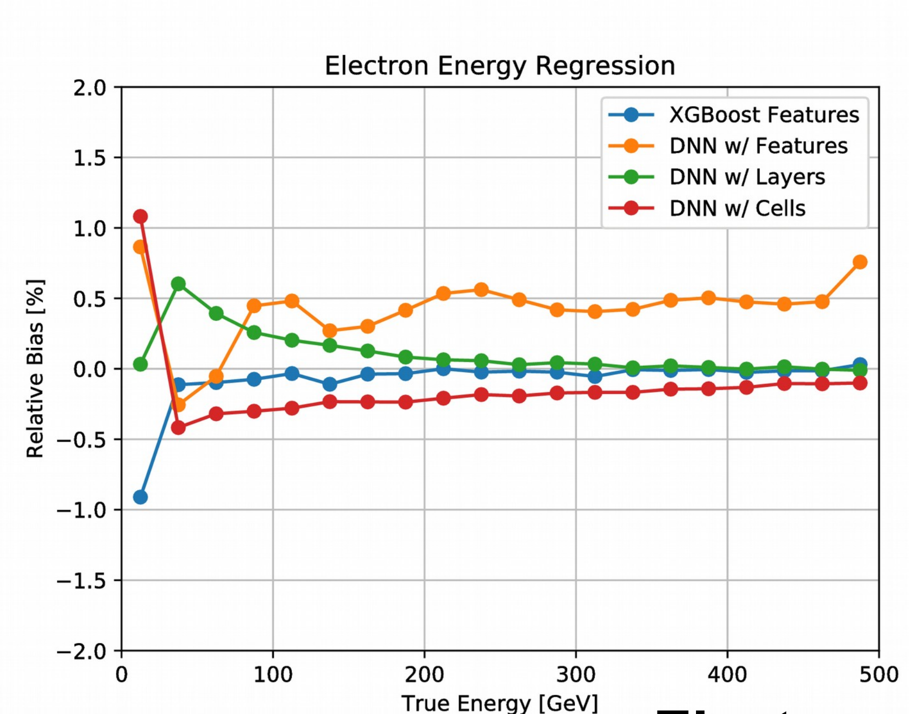
Dominick Olivito (UCSD)

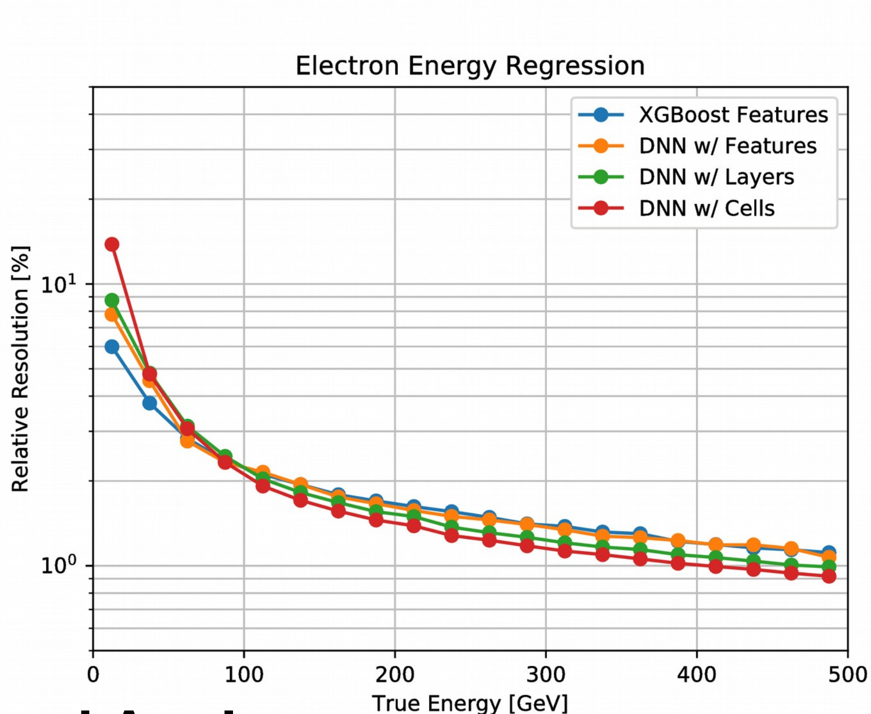
Overview

- Try building up complexity of DNN / CNN Regression networks
  - Feature-based DNN with same features as XGBoost
    - ECAL_E, HCAL_E, ECALMomentZ1
  - Layer-based DNN
    - since depth in Z was the main feature to improve baseline BDT, try summing in X,Y and only using energy in layers of Z
  - Cell-based DNN, try different window sizes in local XY
  - Cell-based CNN, similar architecture to NIPS paper
    - Try different window sizes
    - Try different numbers of convolution filters
    - Try skip connections for ECAL_E, HCAL_E
- Using skip connections by default for ECAL_E, HCAL_E variables
  - Seems to speed convergence
- All this on fixed angle Electon samples

Performance: DNN Inputs

- Since last week, added DNN w/ layers curve
  - Slightly better than cells at lowest energy
  - At high energy slightly worse than cells, better than features

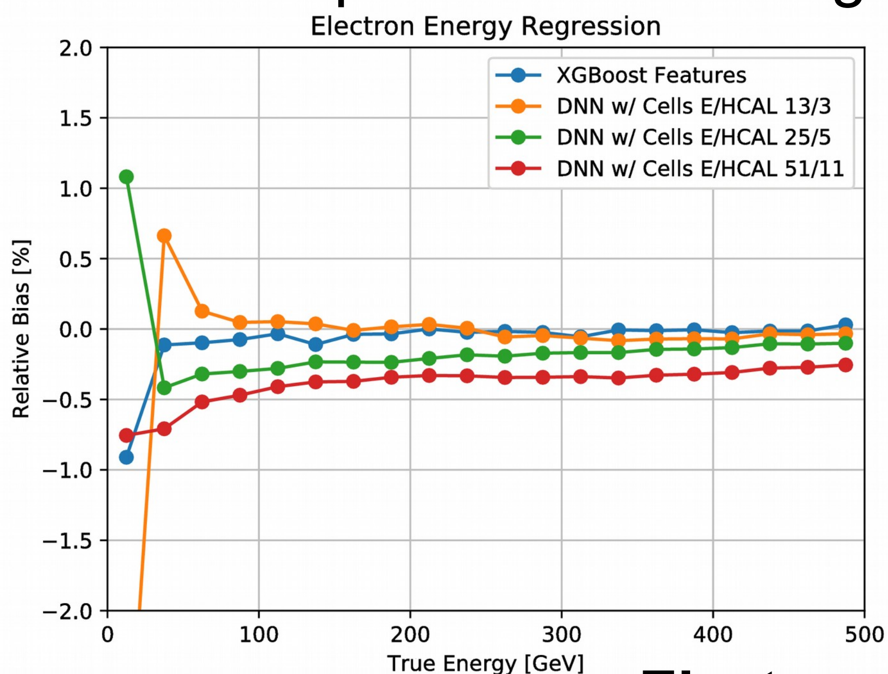


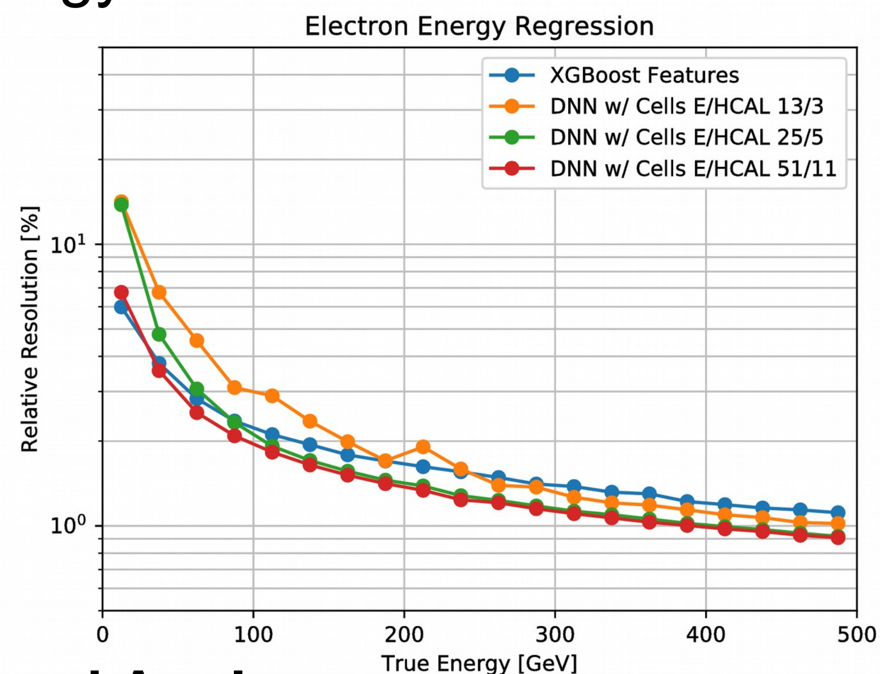


Electrons, Fixed Angle

Performance: DNN Window Size

- Nominal window size: 25 in XY for ECAL, 5 in XY for HCAL
- Worse performance going to smaller window sizes like 13/3
- Going to larger window size 51/11, better performance at low energies
  - Matches now XGBoost with features. Note that features include energy sum over 51/11 size window
  - Same performance at high energy as 25/5 windows



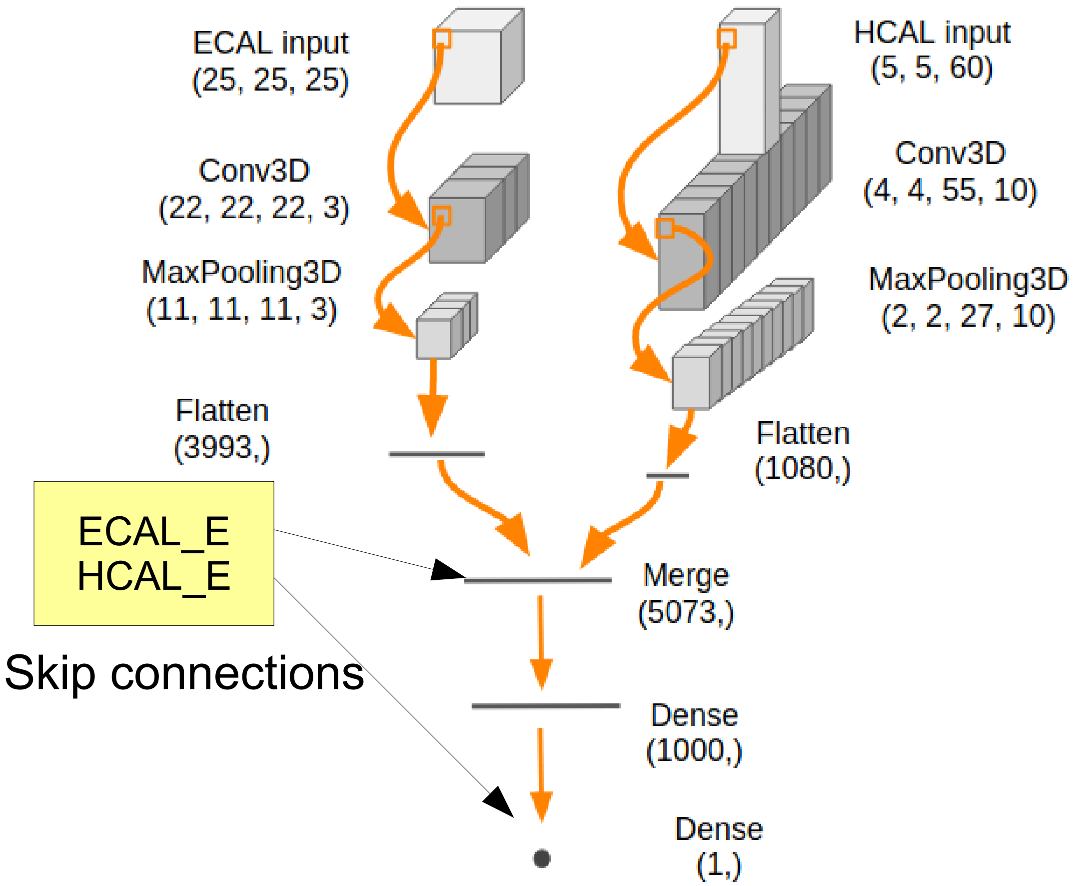


Electrons, Fixed Angle

NIPS CNN Architecture

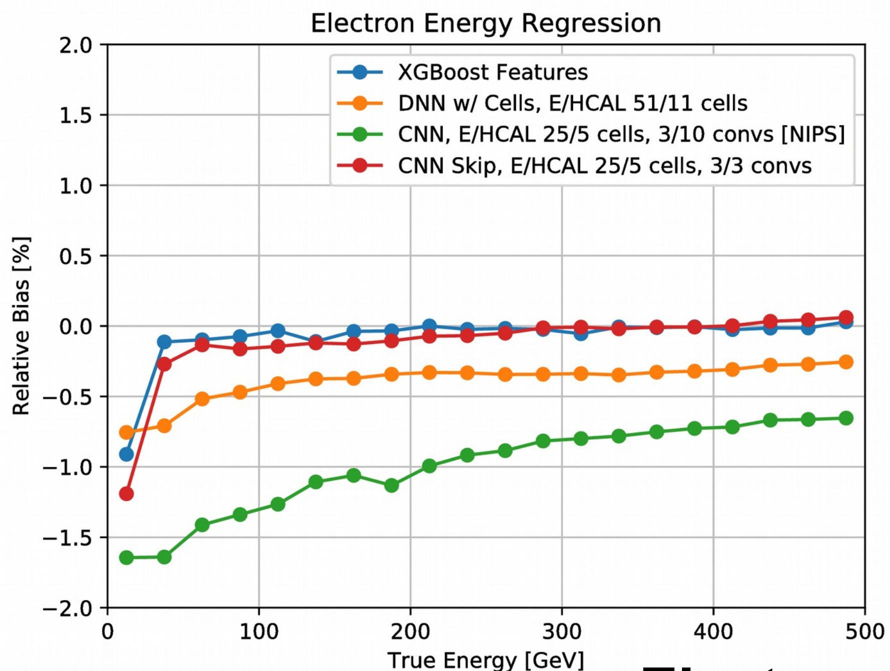
- Starting from NIPS, vary:
  - Add skip connections
  - Input window size
  - Number of conv filters

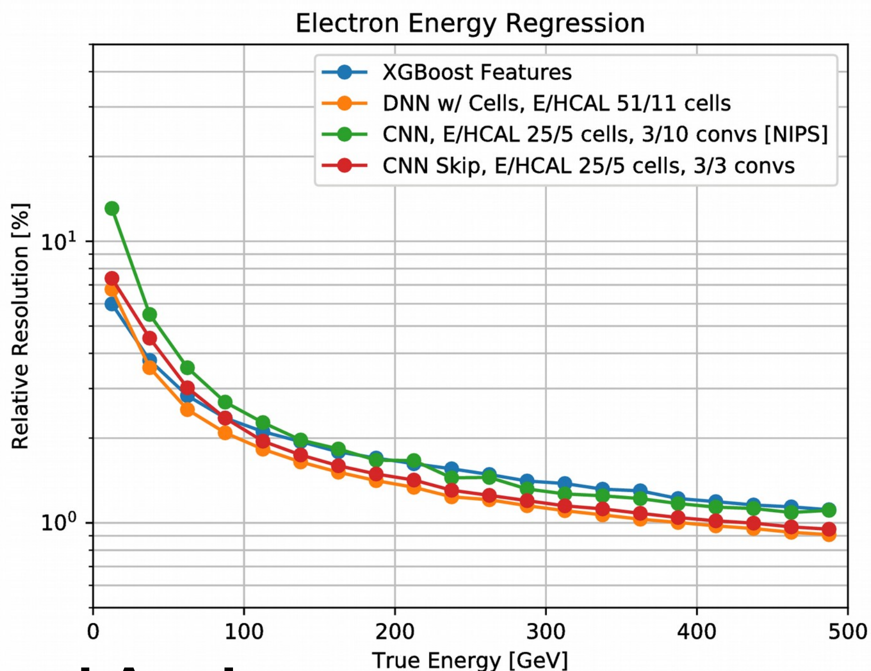
- Could also vary:
  - Conv filter size
  - Max pooling
  - Number of conv layers
  - Number / width of dense layers



Performance: CNN Skip

- NIPS CNN architecture gives similar results to XGBoost at high energy, worse at low energy
- Adding skip connections improves resolution and bias across full energy range
  - Number of convolution filters makes small difference, using less here for HCAL in Skip Connection network

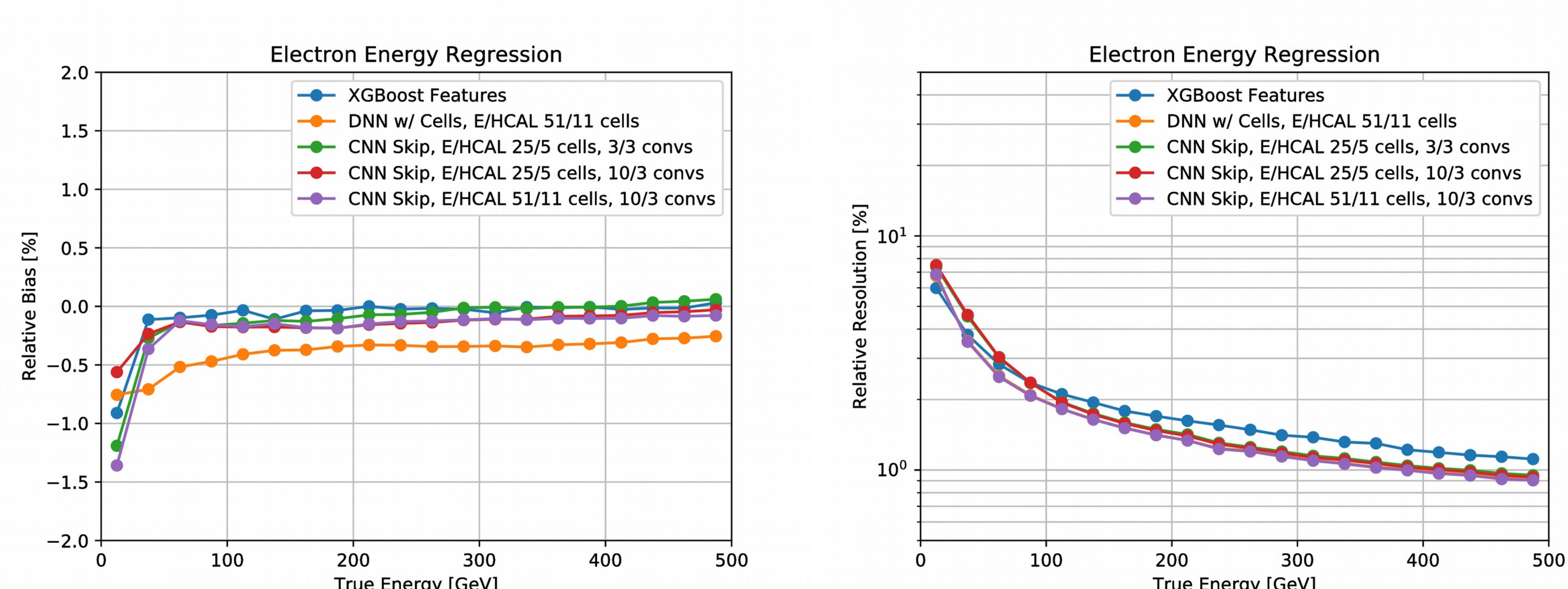




Electrons, Fixed Angle

Performance: CNN Window Size

- CNN also benefits at low energy (< 100 GeV) from going to larger window size
  - Resolution same as DNN, slightly better bias
- Again, number of convolutional filters doesn't seem to make much difference



Electrons, Fixed Angle

Summary / To Do

- Can improve performance at low energy (< 100 GeV) by using larger window size of 51 in ECAL, 11 in HCAL
  - Electrons have wider showers at low energy, especially below 20-30 GeV
- Loading data for training takes much longer though
  - Compared to 25 ECAL, 5 HCAL, takes about 5x longer
  - 1 hour / epoch instead of 12 minutes

- Skip connections improve CNN performance
- Other hyperparameter tuning hasn't made much impact

- Can try best configurations for other particle types
- Can also try to run best classification model for regression
  - Suggestions on which to try?

Bonus Slides

Samples / Details

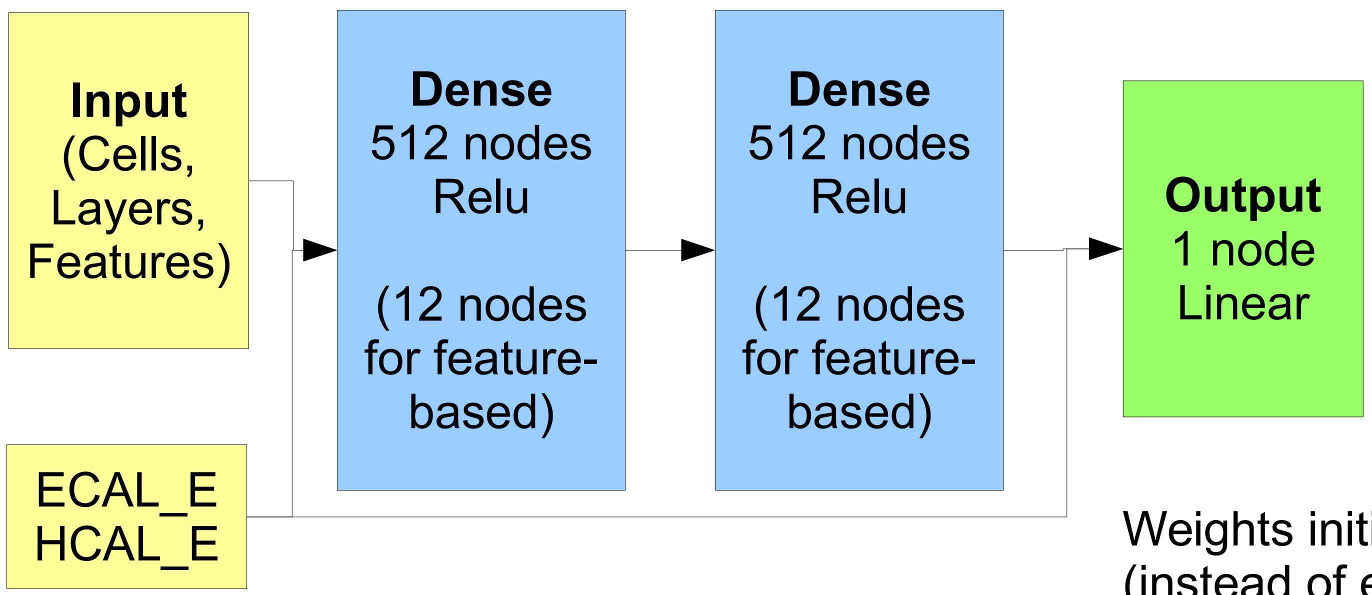
- Samples: new larger window samples, fixed angle, with features
  - On culture-plate at caltech:
    - /data/shared/LCDLargeWindow/fixedangle/*Escan/*.h5
    - /data/shared/LCDLargeWindow/varangle/*Escan/*.h5
  - Slimmed versions with only features (no images):
    - /data/shared/LCDLargeWindow/fixedangle/*Escan/merged_featuresonly/
    - /data/shared/LCDLargeWindow/varangle/*Escan/merged_featuresonly/
  - ~800k events, 70% train, 30% test

- Running XGBoost in python with:
  - maxdepth 3, up to 1000 rounds
  - Early stopping if test loss doesn't improve for 10 rounds

- Running DNNs / CNNs in pytorch, python3 using Triforce
  - Dropout 0.2
  - Adam, learning rate 0.001
  - L2 regularization 0.01 (“decayRate”)
  - Train for 5-10 epochs depending on window size

Skip Connections

- Basic idea: hardcode **Identity function** into network, to make other layers learn residual correction to identity
- Appropriate for our case: we know linear regression in ECAL_E, HCAL_E gets close to the right answer
- Performance is similar, training converges faster
  - For Feature Based NN, in 10-20 epochs instead of 40



Weights initialized to 1 (instead of e-02)