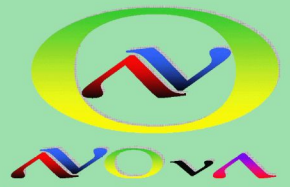


UNIVERSITY OF Cincinnati

Using Modern Deep Learning Techniques to Categorize Neutrino Interactions

Adam Aurisano
University of Cincinnati

IML LHC Machine Learning WG
14 April 2016

NuMI Off-Axis ν_e Appearance Experiment

NOνA is a long-baseline neutrino oscillation experiment located 14 mrad off-axis from the NuMI beam designed to measure:

ν_e appearance

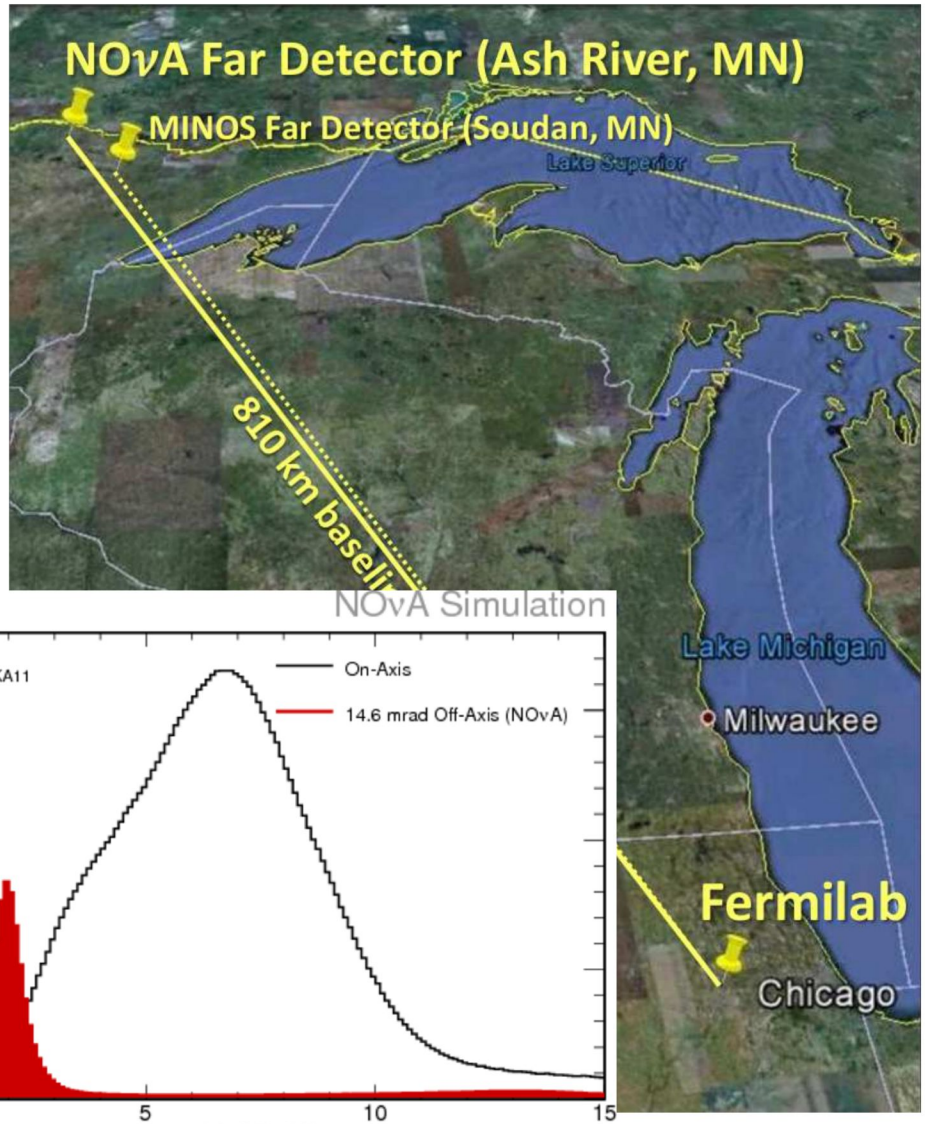
- θ_{13}
- Mass hierarchy
- θ_{23} octant
- CP violation

ν_μ disappearance

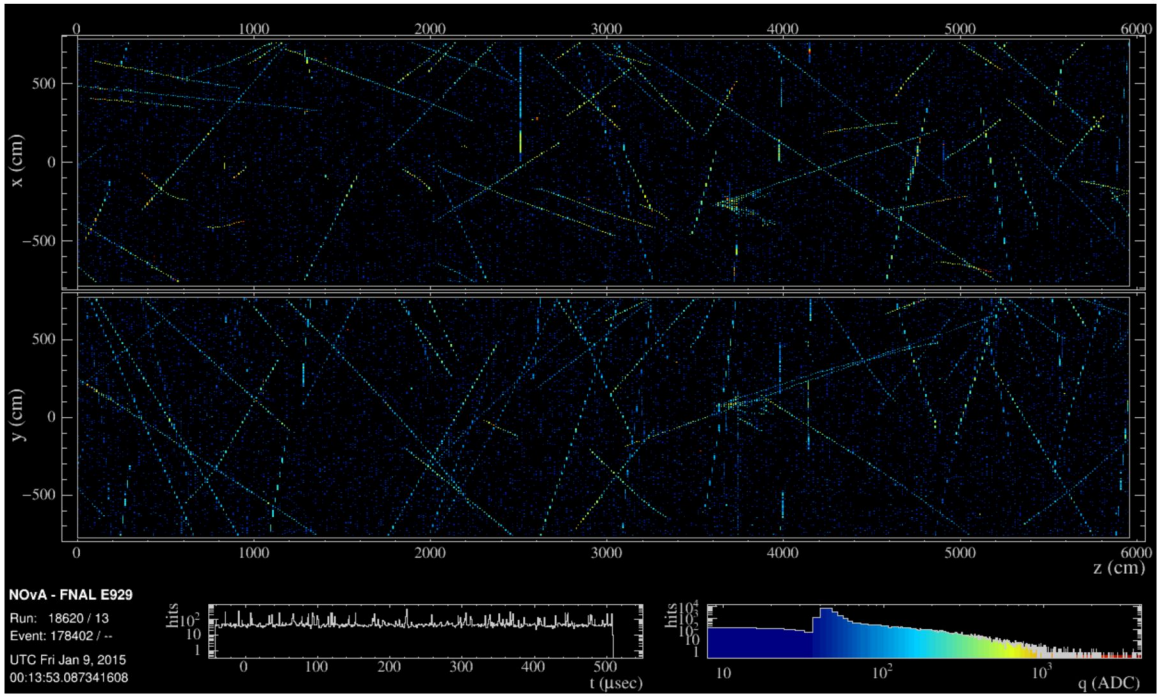
- Improved precision on $|\Delta m^2_{32}|$ and θ_{23}

Others

- Cross-sections
- Steriles
- Supernovae
- Exotics

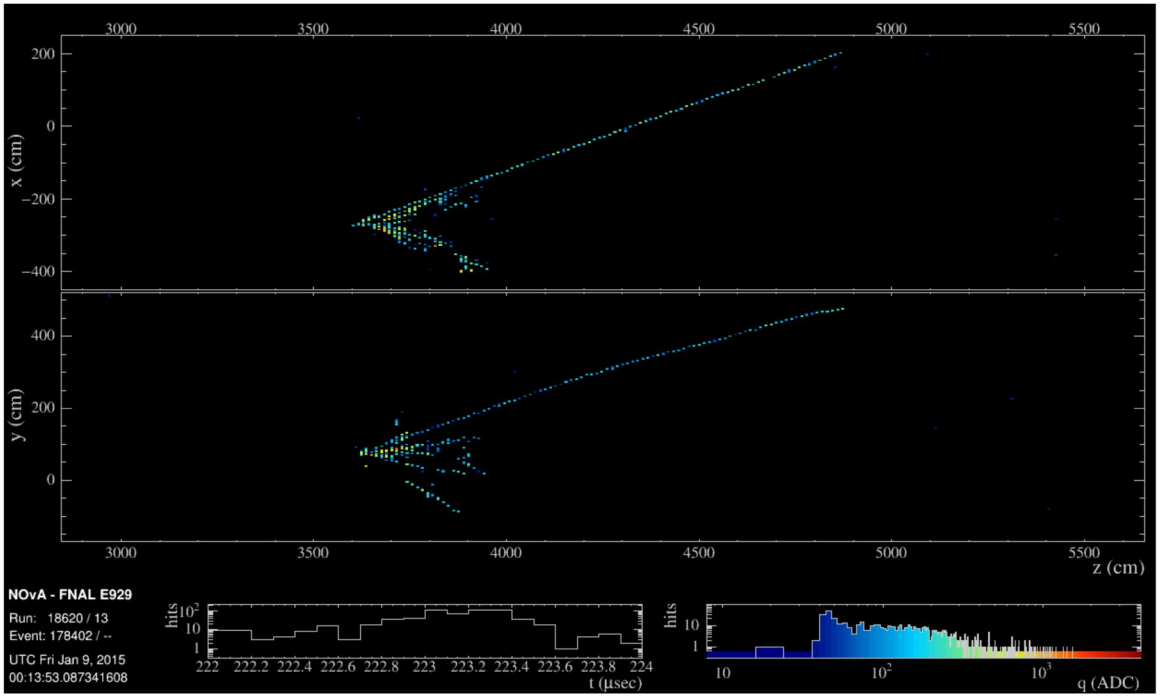


Can You Find the Neutrino?



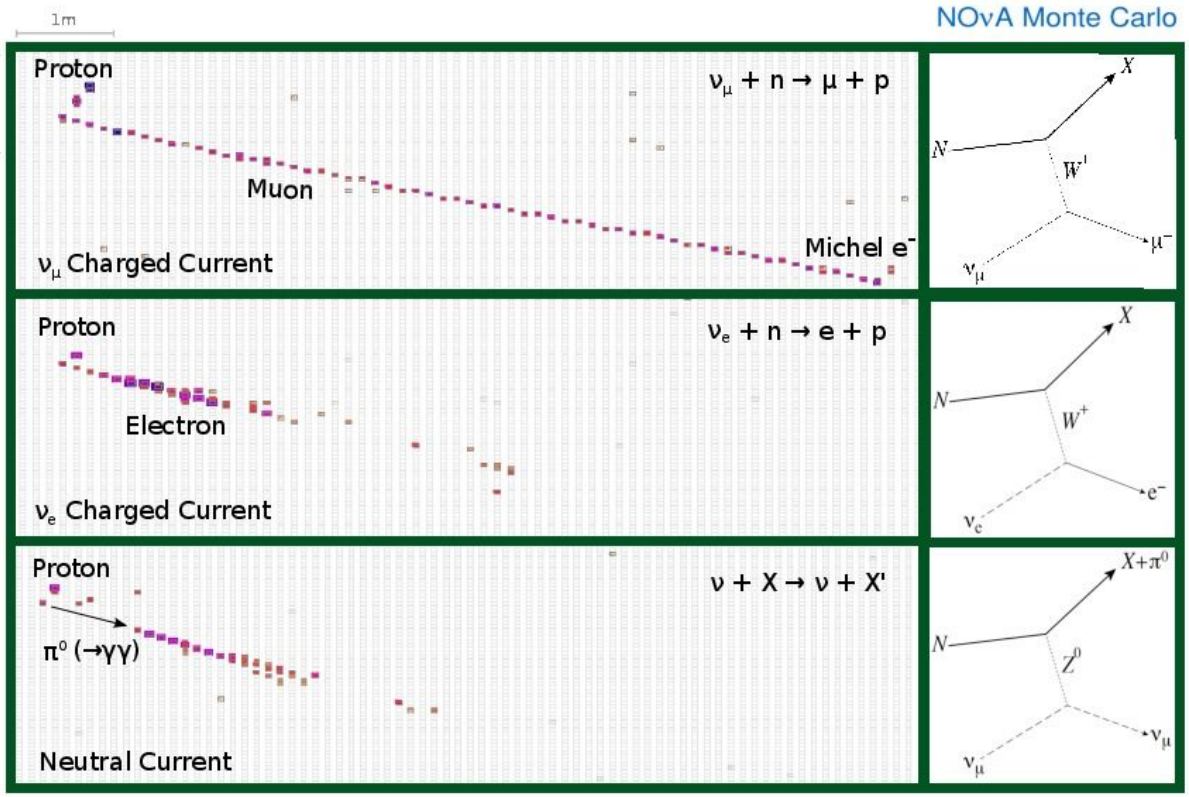
- At NOvA, data is taken in 550 μs intervals.
- Most of the activity is from cosmic rays
  - 100,000's cosmic rays/second
  - 100's ν_μ/year
  - 10's ν_e/year

Zooming in...



- Clustering in space and time creates slices
  - Groups of hits likely to be causally related
  - Lets us separate neutrino events, cosmic rays, and noise
  - ...But it still doesn't tell use what each slice actually is – for that we need a classifier.

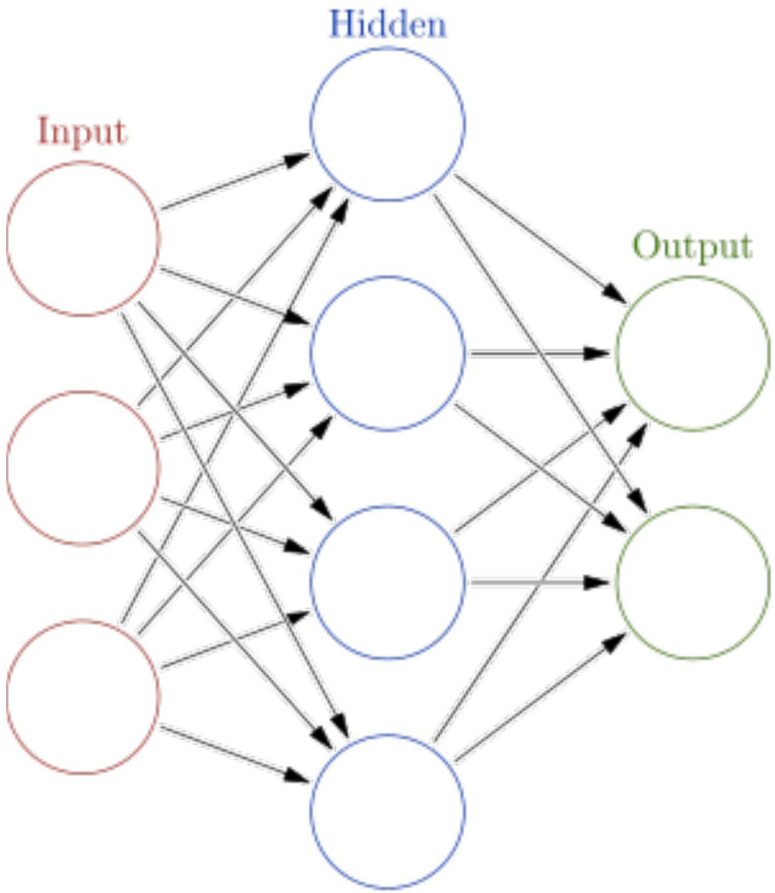
Event Topologies



More complicated events can contain multiple charged pions make it more difficult to separate these event types.

- Excellent spatial granularity.
- Muons look like long tracks
- Low Z material allows for seeing the development of electron showers.
- NC can look very similar to electron showers, but there is sometimes a gap between the start of the event and the beginning of the π^0 shower.

Review: Traditional Neural Nets



- Traditional neural nets are powerful machine learning tools in wide spread use throughout high energy physics.
- Nodes are organized in layers
- Each node performs a weighted sum on the outputs of all nodes in the previous layer, and the result pushed through a non-linear function – often a sigmoid function.
- Optimize the weights using an iterative learning procedure.
  - Minimize an error function by comparing the ground truth to the prediction for a set of training data.

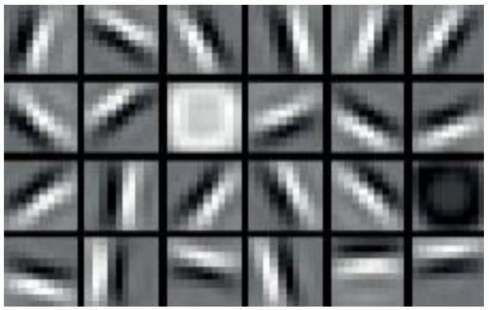
Deficiencies of Traditional Neural Networks

- In theory, a single layer network with a sufficient number of nodes can approximate most functions to arbitrary precision.
- Due to the fully connected nature of traditional neural nets, number of free parameters increases sharply with additional nodes.
- For the same reason, does not scale well to raw data.
  - Important to reduce raw data down to a few, powerful, engineered features to use as input.
  - Requires significant expertise and is limited to our imagination in developing new features.
- Large number of free parameters is computationally complex to train and evaluate.
  - Also risks learning the training set exactly while failing to generalize to other data.
- Multi-layer networks can often approximate a function with fewer nodes than a single layer network.
  - Maybe deeper is better?

Deep Learning

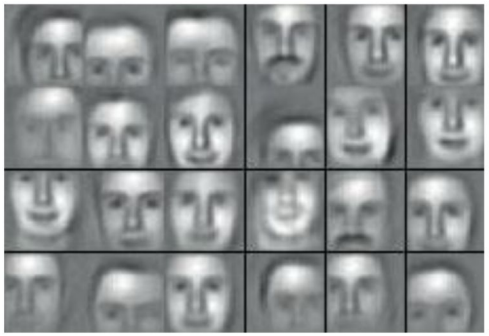
Raw input

Low level features



Mid level features

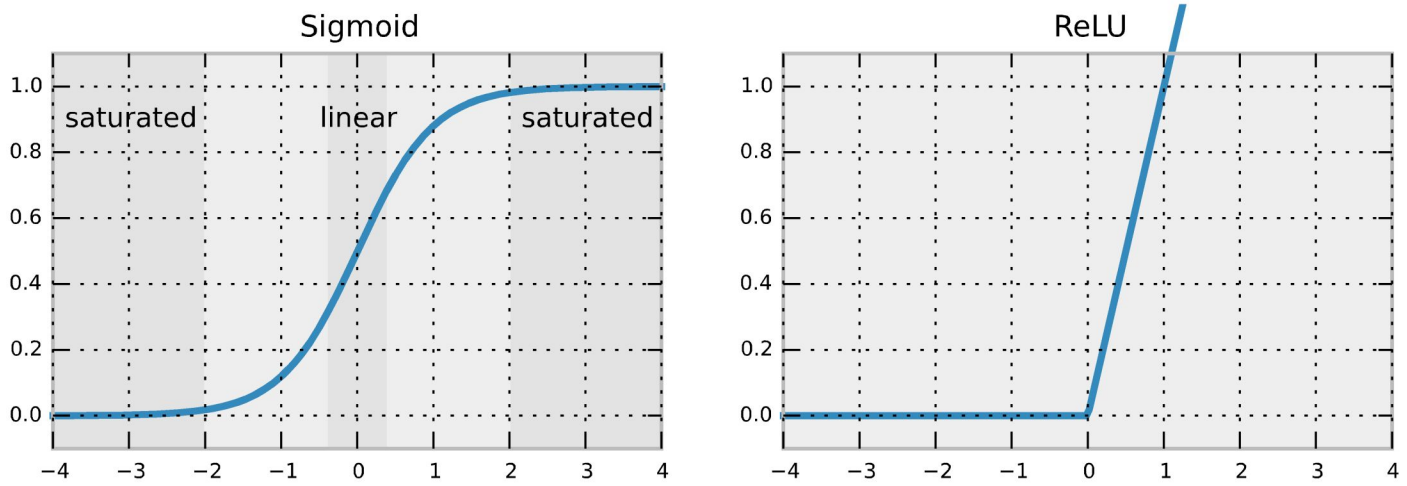
High level features



developer.nvidia.com/deep-learning-courses

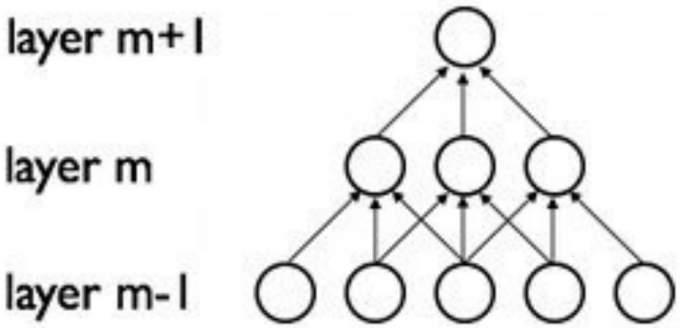
- Deep learning is a new paradigm that has caused a renaissance in the machine learning community.
- Use sparsely connected neurons to allow for many hidden layers.
- Deep structure extracts increasingly complex features from the input data instead of needing engineered features.

Activation Functions



- Traditional neural nets typically use saturating non-linear functions like the sigmoid.
- These functions have a small useful range.
  - Optimization techniques based on the steepest gradient descent get stuck if weights enter the saturated region.
  - Proper initialization of the weights required sophisticated pre-training to keep weights in the trainable region.
    - Either building up and training the deep structure one layer at a time or initializing with restricted Boltzmann machines that went through unsupervised training.
- New non-saturating non-linear functions like ReLU make it possible to train deep networks with no pre-training since they can not get stuck on the positive side.

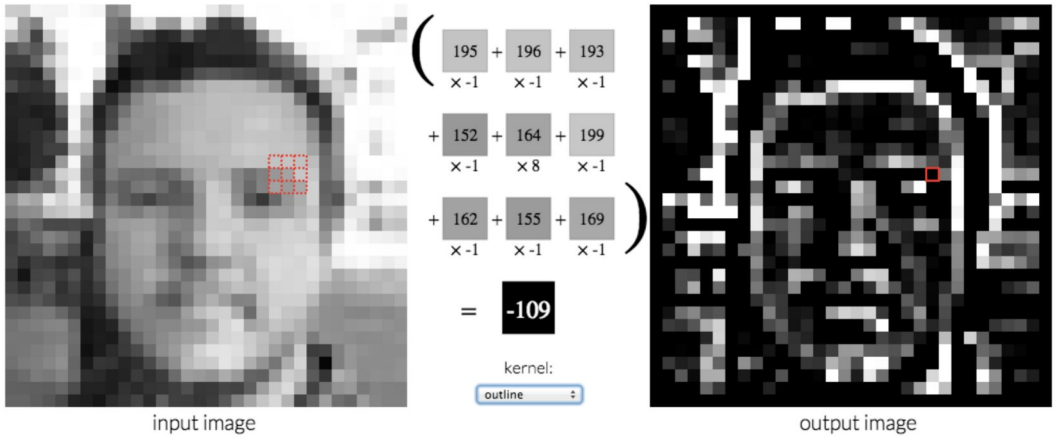
Convolutional Neural Nets



http://deeplearning.net/tutorial/lenet.html

- Convolutional neural nets are a very successful deep learning method.
- Inspired by research showing that the cells in the visual cortex are only responsive to small portions of the visual field - “receptive field”.
- Some cells collect information from small patches – sensitive to edge-like features.
- Other cells collect information from large patches.
- Effectively, these cells are applying convolutional kernels across the visual field.

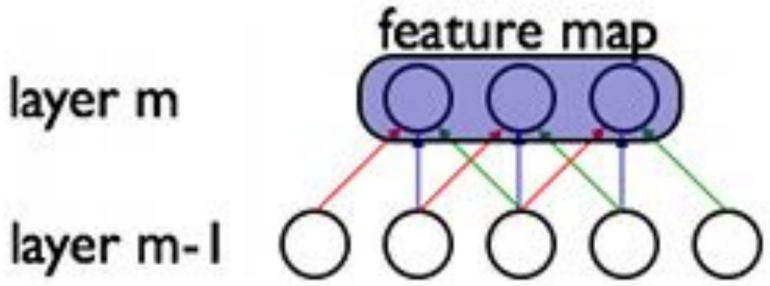
Convolutional Kernels

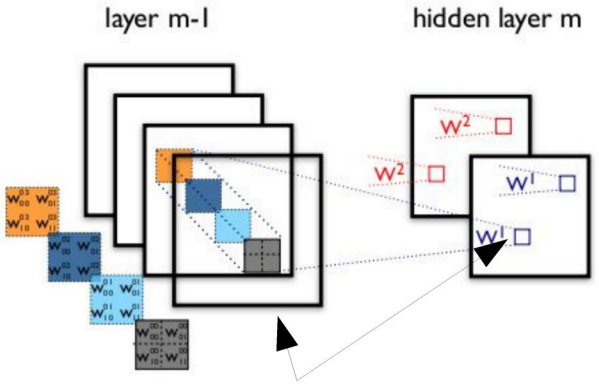


- Convolutional kernels are well known in computer graphics.
- Kernels transform images.
  - The one above outlines objects in the image.
- Many common kernels exist, but it we want to learn optimal kernels directly from the data.

http://setosa.io/ev/image-kernels/

Convolutional Layers



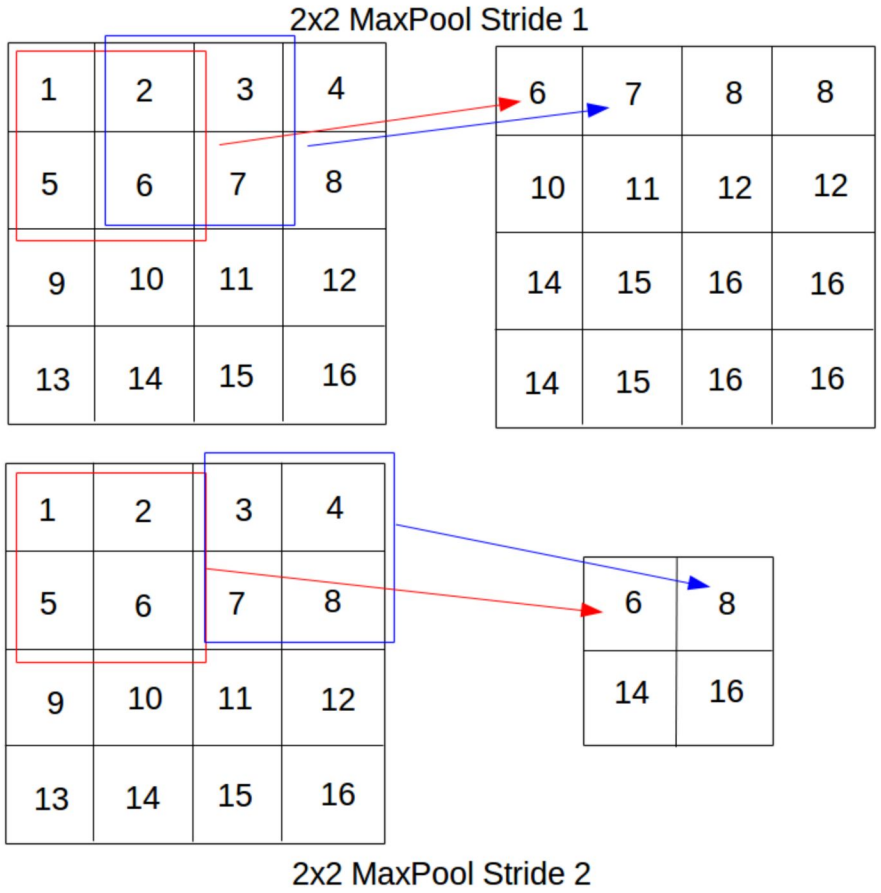


Each pixel is the result of a tensor dot product of the weights with a tower of patches in the incoming feature maps

- Each kernel we create stays the same as we apply it across the image.
  - Weight sharing reduces the number of free parameters, lowering the risk of overtraining.
- Each convolutional layer trains an array of kernels which produce corresponding feature maps.
- Weights going from layer to the next are a 4D tensor of NxMxHxW
  - N is number of incoming feature maps
  - M is the number of outgoing feature maps
  - H and W are the height and width of the outgoing convolutional kernels.
- The next layer applies kernels to combine the information in a receptive field across feature maps in the previous layer to create new feature maps.

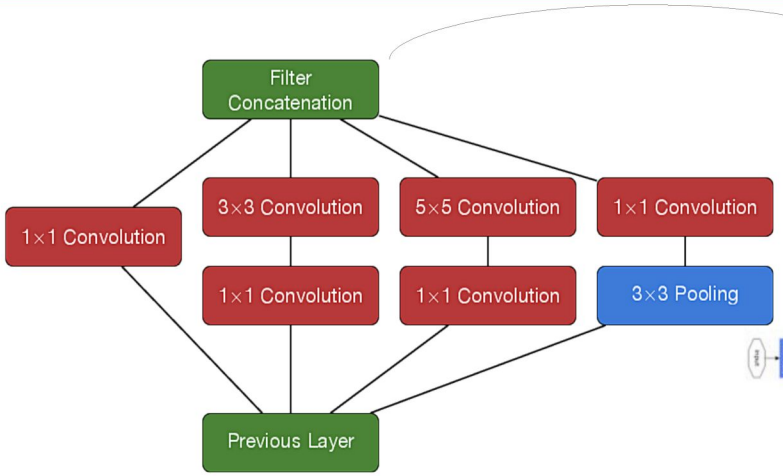
http://deeplearning.net/tutorial/lenet.html

Pooling Layers

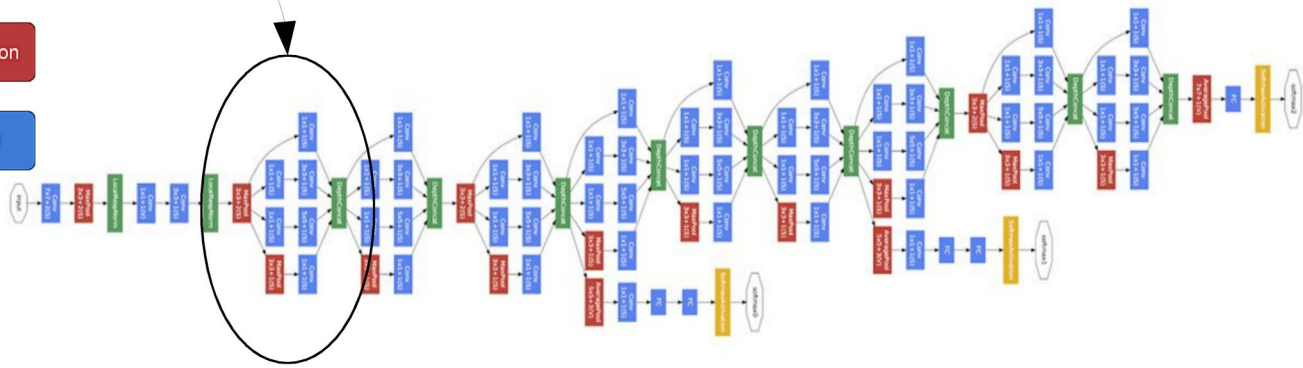


- Pooling is a technique to down sample information.
  - Output pixel is either maximum value of a patch of input pixels (max pooling) or the average (average pooling).
- Can be thought of as a type of smoothing to remove less significant information.
- Can either be strided or unstrided
  - Controls how much information is lost
- Number of output feature maps is the same as the number of input maps.

GoogLeNet



C. Szegedy, et. al., Going Deeper with Convolutions, arXiv:1409.4842 (2014)



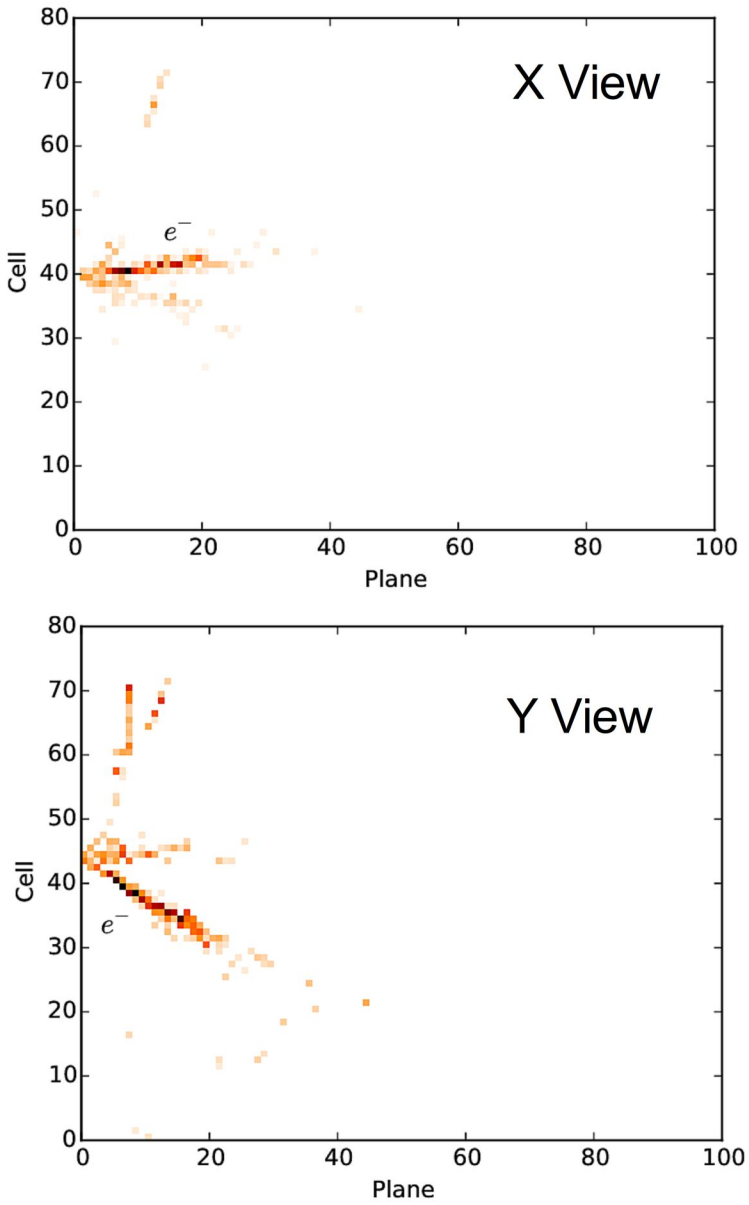
- Due to the rise of relatively cheap GPUs, it has become possible to make increasingly complex network-in-network models.
- The GoogLeNet architecture is composed of a series of inception modules.
  - Outputs of the previous layer fans out to several convolutional layers with different kernel size.
  - Applies max pooling to downsample in feature map height and width and 1x1 convolutions to downsample the previous stack of feature maps into a smaller set of feature maps.
    - Designed to get maximum identification power out of as few operations as possible.
- In the ILSVRC 2014 image classification task, the correct classification was not one of the top 5 ranked categories out of 1,000 only 6.67% of the time.

Caffe

- Caffe (caffe.berkeleyvision.org) is deep learning framework developed by the Berkeley Vision and Learning Center.
- Comes pre-packaged with a large variety of layer types.
  - Vision layers: convolutional, pooling
  - Activation layers: sigmoid, ReLU, tanh, power
  - Others: inner product (traditional fully connected layer), concatenation, splitting, dropout
- Able to run on GPUs without any extra effort (only change one line in the configuration file)
- Comes with a variety of models produced by different computer vision groups for image classification contests.
- Tried several – LeNet, AlexNet, VGG, but GoogLeNet performed the best.
- All training done on the Wilson cluster at Fermilab, thanks to the support of the Scientific Computing Division
- Integrated with the Fermilab product distribution system and with the art framework through a custom module.

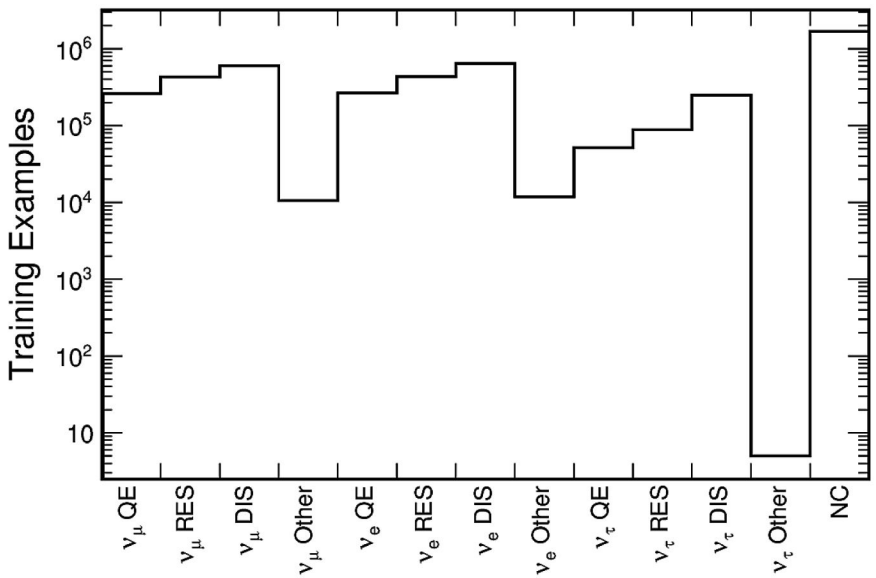
Y. Jia et. al., Caffe: Convolutional Architecture for Fast Feature Embedding, arXiv:1408.5093 (2014)

Constructing Input Images



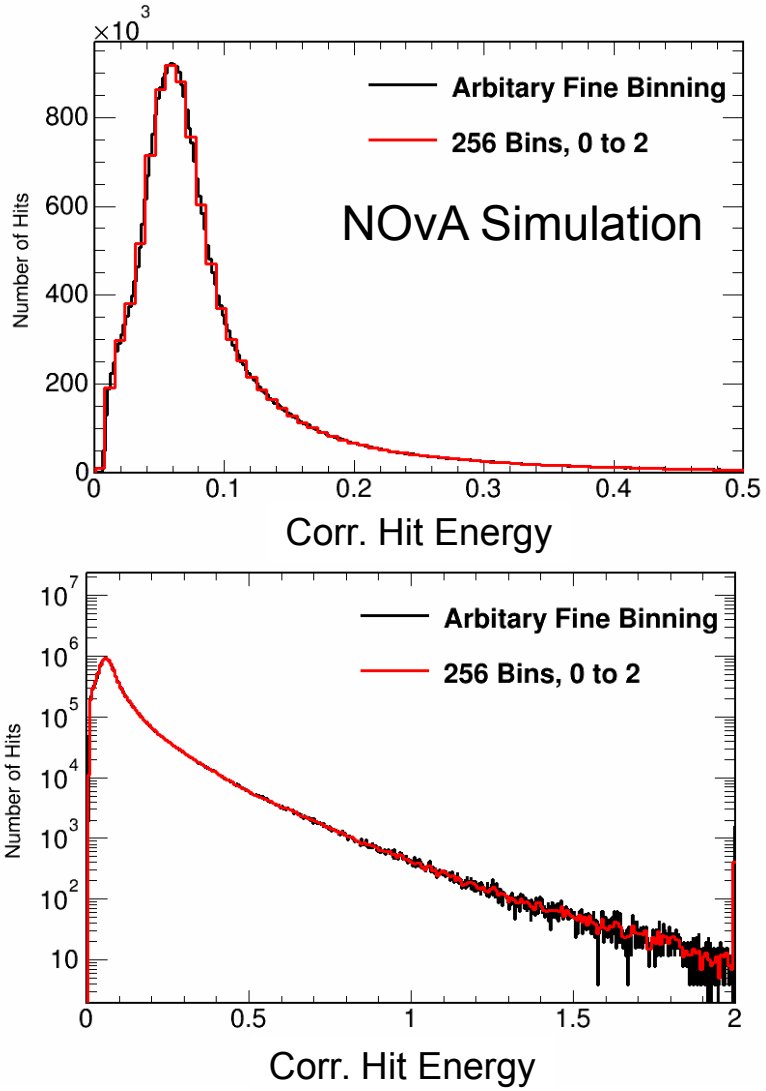
- For our input, we construct “pixel maps” from hits in a slice.
  - No reconstruction other than initial hit clustering.
  - All slices with at least 15 distinct hits were used in training.
- The X view is composed of all planes with vertical cells
  - A projection on the x,z plane
- The Y view is composed of all planes with horizontal cells.
  - A projection on the y,z plane
- We take all hits in a 100 plane by 80 cell box for each view
  - ~14.52 m deep and ~4.18 m wide

Building Training and Testing Datasets



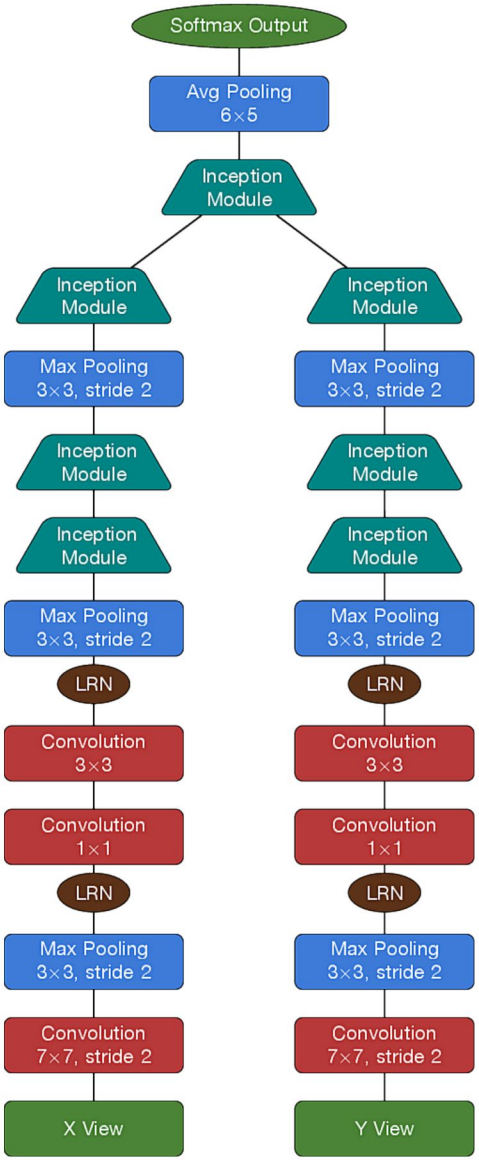
- Take existing simulations containing ν_μ, ν_e, and ν_τ events.
- Construct a pixel array pixels[c][h][w] where c indexes the view, h indexes cells, and w indexes planes from pixel maps.
- Push into LevelDB databases: 80% for training and 20% for testing.
- Rescale the corrected hit energy to go from 0 – 255 and recast to 8 bits to save space.
- Almost all hits have exactly zero energy (suppressed in these plots).
- Distribution very different from natural images.

Building Training and Testing Datasets



- Take existing simulations containing ν_μ, ν_e, and ν_τ events.
- Construct a pixel array pixels[c][h][w] where c indexes the view, h indexes cells, and w indexes planes from pixel maps.
- Push into LevelDB databases: 80% for training and 20% for testing.
- Rescale the corrected hit energy to go from 0 – 255 and recast to 8 bits to save space.
- Almost all hits have exactly zero energy (suppressed in these plots).
- Distribution very different from natural images.

Convolutional Visual Network (CVN)



- GoogLeNet showed the most promise in our early testing.
- Treating each view as channels of an image does not make sense.
  - Initial convolutional layer would have linearly combined unrelated information in each view.
- Instead, create a "siamese" GoogLeNet variant.
  - Split the views early and double the architecture. Each parallel GoogLeNet learns separate features. These are merged together at the end before going through fully connected layers.
- The architecture attempts to categorize events as $\{\nu_\mu, \nu_e, \nu_\tau\}$ x {QE,RES,DIS} or cosmic rays.
  - QE (quasi-elastic): Incoming neutrino interacts with a single nucleon. Results in an out-going lepton + a recoil nucleon
  - RES (resonance): Interaction of the neutrino excites a Δ resonance, often leading to a pion in the final state (in addition to the lepton and nucleon).
  - DIS (deep inelastic scattering): The interaction causes fragmentation of the nucleus.
  - These are approximately in order of increasing complexity, though final state interactions make this more complicated.
- In principle, this architecture a universal classifier (rather than ν_e only).

Making CVN Robust

- What is overtraining?
  - Networks contain large numbers of parameters – sometimes they learn how to classify the training data exactly at the expense of generalizing well to new data.
  - Can be seen if the evaluation of testing data begins to diverge from that of training data.
- Convolutional neural networks tend to already be more robust due to having fewer trainable parameters compared to fully connected networks.
- In our case, our training data is entire synthetic
  - Must also make sure we generalize from simulation to data.
- Techniques use to prevent overtraining
  - Early stopping
    - Hard to make rigorous – will not use with CVN.
  - Regularization
  - Dropout
  - Data Augmentation

Regularization

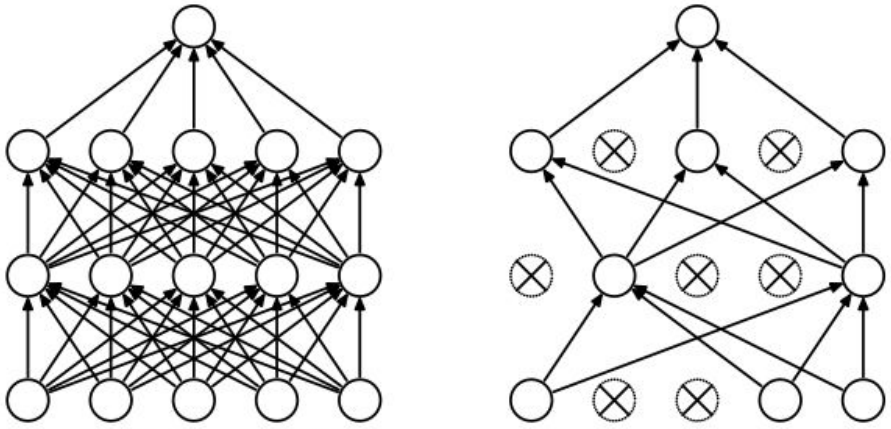
- Add a term to the error function to be minimized of the form:

$$\frac{1}{2}\lambda \sum w_i^2$$

- Decreases the number of effective free parameters.
- Prevents any weight from being too large unless there is strong evidence that it needs to be.
  - Makes it difficult for the network to finely tune weights to perfectly categorize training examples.

Moody, J., et al. "A simple weight decay can improve generalization." Advances in neural information processing systems 4 (1995): 950-957.

Dropout



(a) Standard Neural Net (b) After applying dropout.

- In the fully connected layer in CVN, we apply the dropout technique.
  - At each iteration, randomly set 40% of weights to zero and scale the rest up by 1/(1 – 0.4).
  - Since no weight is reliably in use with any other weight, weights can not be strongly correlated.
  - Preventing weight co-adaptation strongly promotes generalization.
  - Can be thought of as an ensemble of smaller networks.

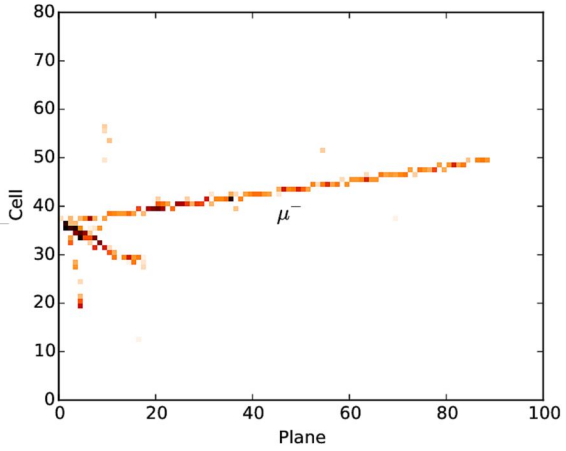
Srivasta et al, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”, JMLR 2014

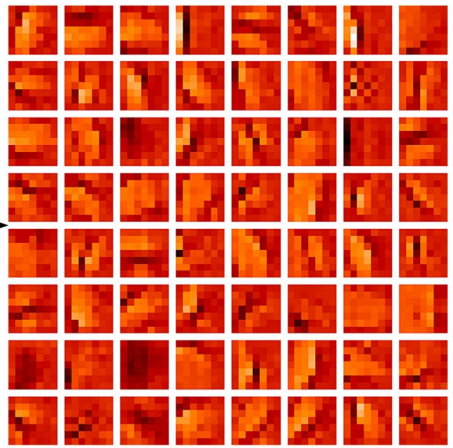
Data Augmentation

- Unlike with natural images, our training set is entirely synthetic.
  - In addition to normal concerns about overtraining, we must make sure that the network adequately generalizes from simulation to data.
- Reflecting images across the y-axis effectively doubles the dataset.
- Augment the simulated training set by applying systematic shifts.
  - Apply channel by channel multiplicative jitter to make the network less sensitive to calibration uncertainties.

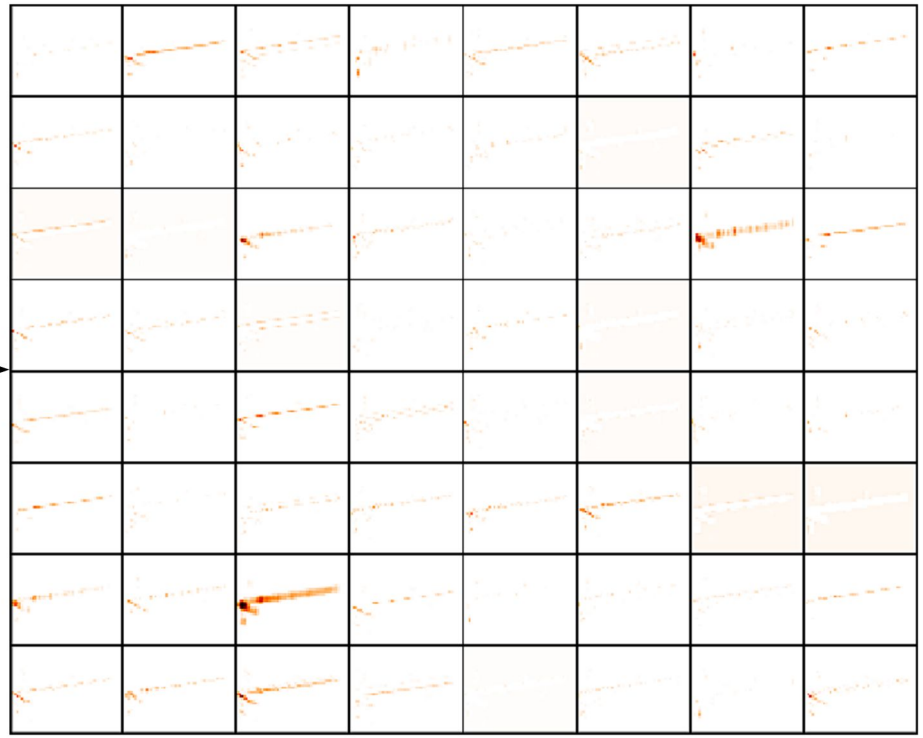
Understanding the Net

Input image





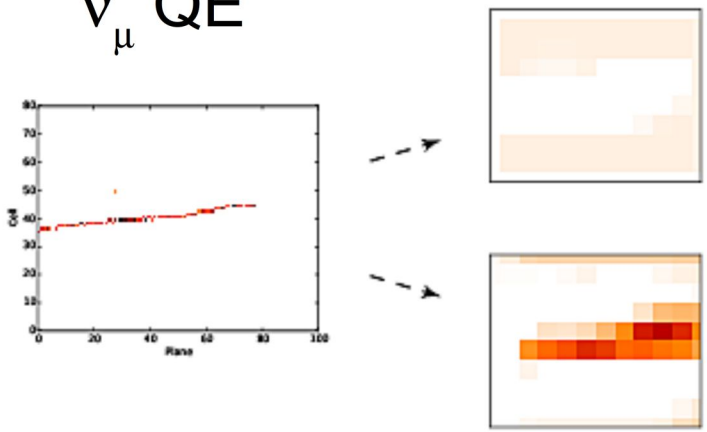
Weights from 7x7 convolutional layer



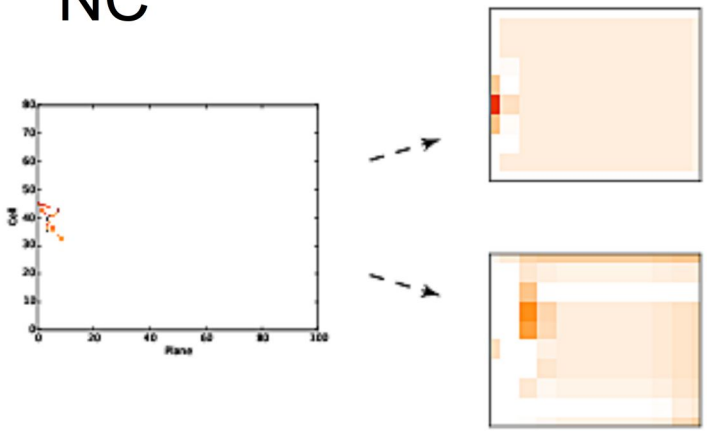
→ Convolutions subsample, sharpen, blur, etc the input image
→ Some feature maps highlight the muon track or hadronic activity.

Understanding the Net

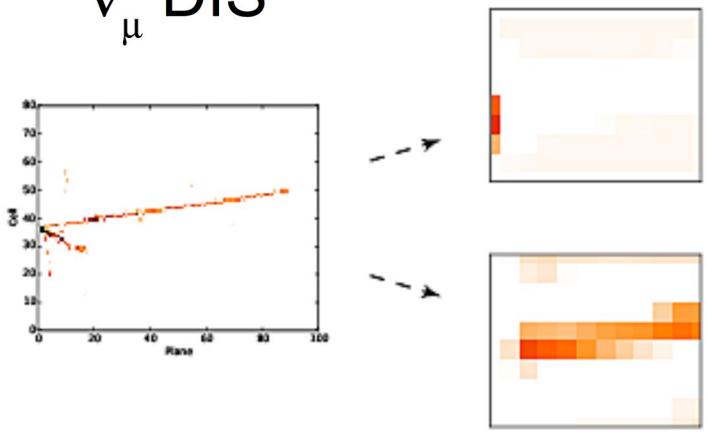
ν_μ QE



NC



ν_μ DIS



A look at two feature maps produced by the first inception module on the y-view branch.

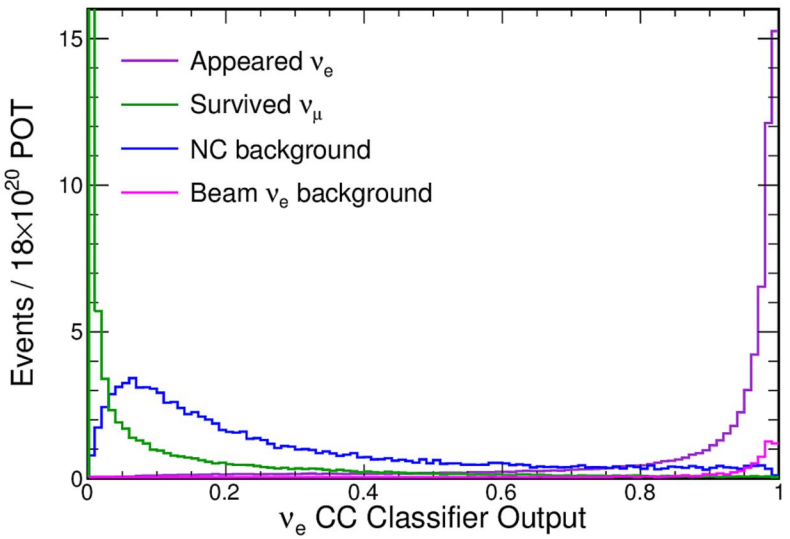
→ Top feature map sensitive to hadronic activity.

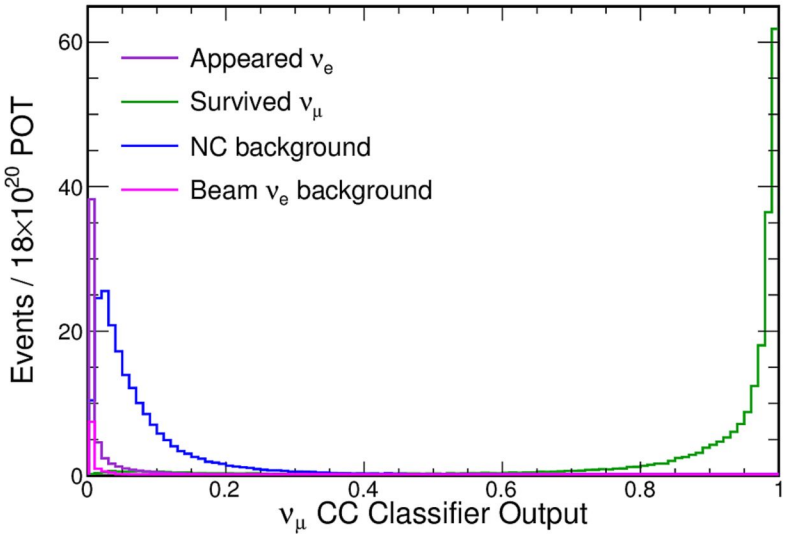
→ Bottom feature map sensitive to muon tracks.

Evaluation as a ν_μ/ν_e Selector

- Our architecture can be used as a ν_μ or ν_e selector similar to current selectors by summing over all interaction type outputs for the relevant flavor.
- Evaluate over a sample statistically independent of the training and testing samples.
- Weight by the simulated NOvA flux and neutrino oscillation probabilities using global best fits of oscillation parameters.
- Apply pre-selection criteria used in the NOvA ν_e analysis (arXiv:1601.05022) and ν_μ analysis (PRD 93 (2016) 051104) designed to reject cosmics backgrounds while keeping most neutrino interactions contained within the detector.

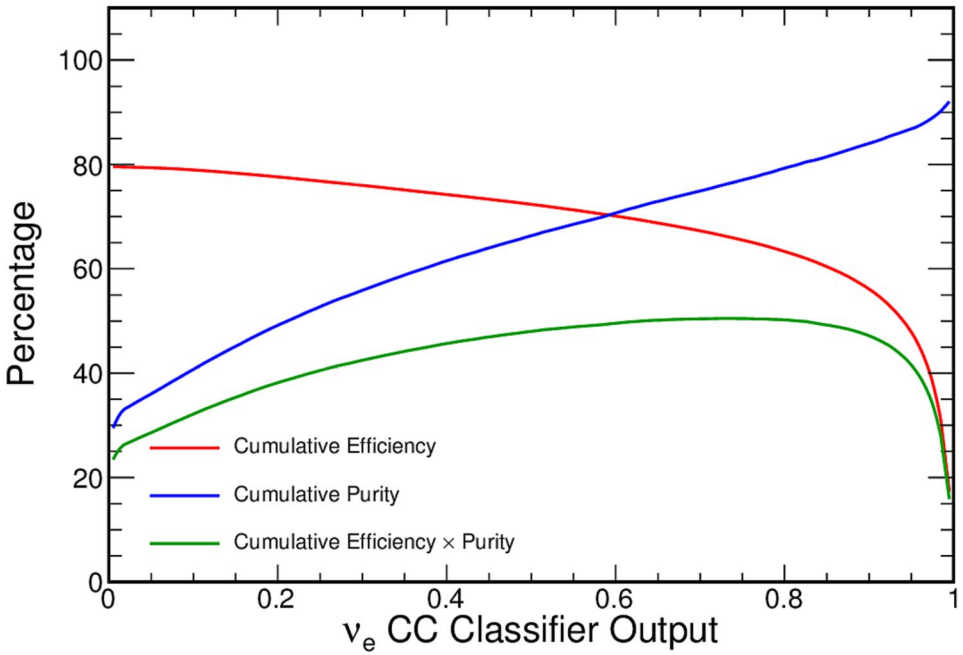
PID Spectra





- 18×10^{20} protons-on-target (3 nominal years of running), full 14-kton Far Detector.
- Primary background for ν_e sample is electron neutrinos produced promptly by the beam.
  - Irreducible background.
- Virtually all backgrounds are removed by the ν_μ classifier.

Performance as a ν_e Selector

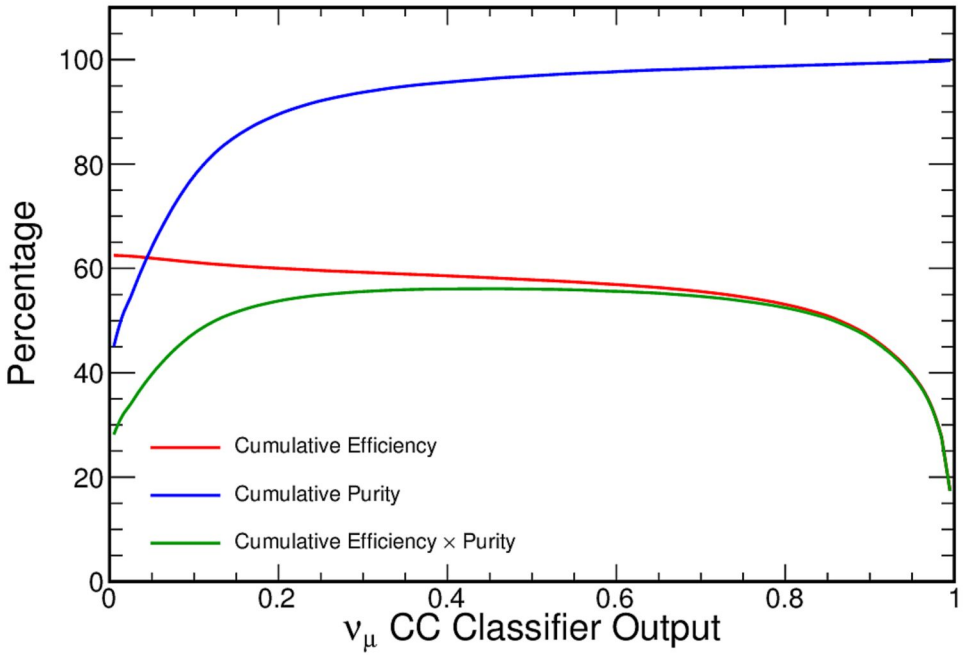


Efficiency and purity as functions of minimum allowed CVN output.

- Can optimize selection for discovery (S/sqrt(B)) or measurement (S/sqrt(S+B)).
- For the first analysis, we optimized for discovery.
- Using this optimization, CVN achieves a 40% improvement in efficiency while maintaining the same purity as compared to existing selectors.
  - Since the first analysis was statistics limited, efficiency gains directly translate to improved measurements of physics parameters (θ_{13}, mass hierarchy, etc)
- Primarily due to improved efficiency in selecting resonance and deep inelastic scattering interactions.

	CVN Selection Value	ν_e sig	Tot bkg	NC	ν_μ CC	Beam ν_e	Signal Efficiency	Purity
Contained Events	–	88.4	509.0	344.8	132.1	32.1	–	14.8%
$s/\sqrt{b}$ opt	0.94	43.4	6.7	2.1	0.4	4.3	49.1%	86.6%
$s/\sqrt{s+b}$ opt	0.72	58.8	18.6	10.3	2.1	6.1	66.4%	76.0%

Performance as a ν_μ Selector



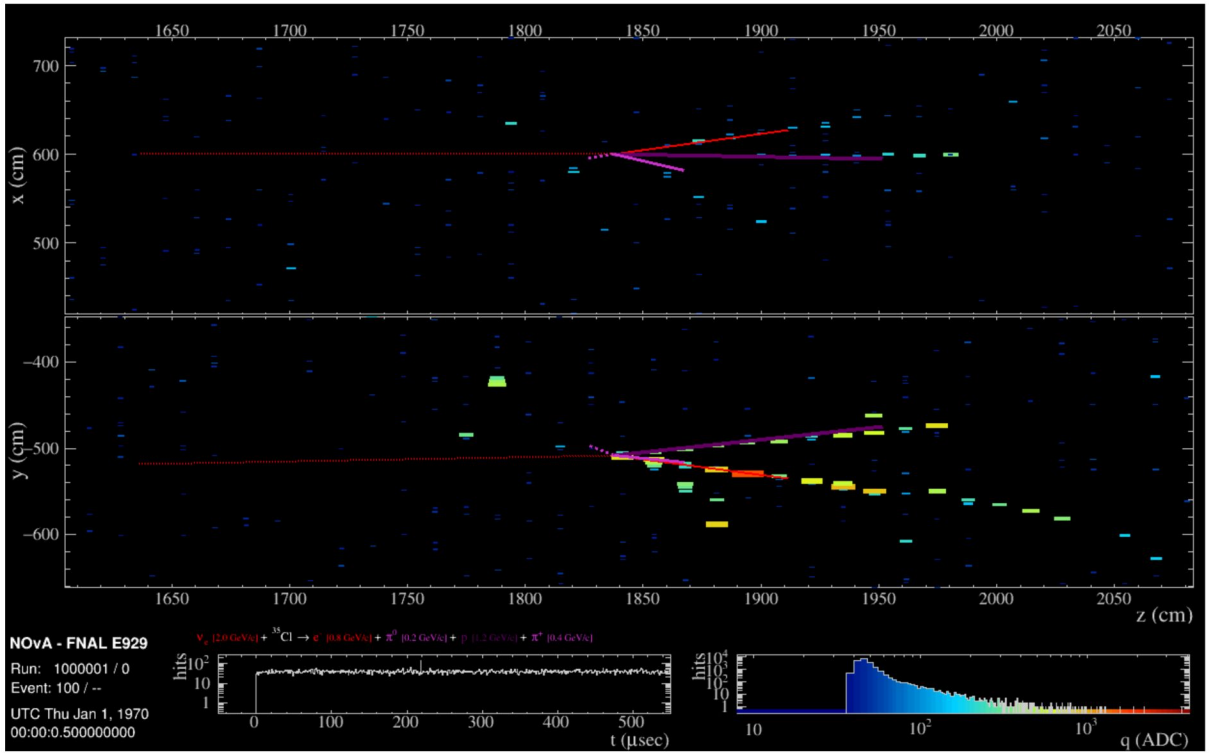
Efficiency and purity as functions of minimum allowed CVN output.

- Can optimize selection for discovery (S/sqrt(B)) or measurement (S/sqrt(S+B)).
- For the first analysis, we optimized for measurement.
- Using this optimization, CVN achieves slightly better efficiency and purity compared to the existing selector.
- Most ν_μ events are easily identified due to the long muon track.
  - Technique does not underperform when given well understood events.
  - Improved selection of events with very low energy muons.

	CVN Selection Value	ν_μ sig	Tot bkg	NC	Appeared ν_e	Beam ν_e	Signal Efficiency	Purity
Contained Events	–	355.5	1269.8	1099.7	135.7	34.4	–	21.9%
$s/\sqrt{b}$ opt	0.99	61.8	0.1	0.1	0.0	0.0	17.4%	99.9%
$s/\sqrt{s+b}$ opt	0.45	206.8	7.6	6.8	0.7	0.1	58.2%	96.4%

Event Selected Only by CVN

NOvA Simulation



- This simulated event was only selected by our architecture (not existing selectors).
- Note the pion nearly co-linear with the electron.
- These types of events are difficult to identify with traditional methods.

Architecture Improvements

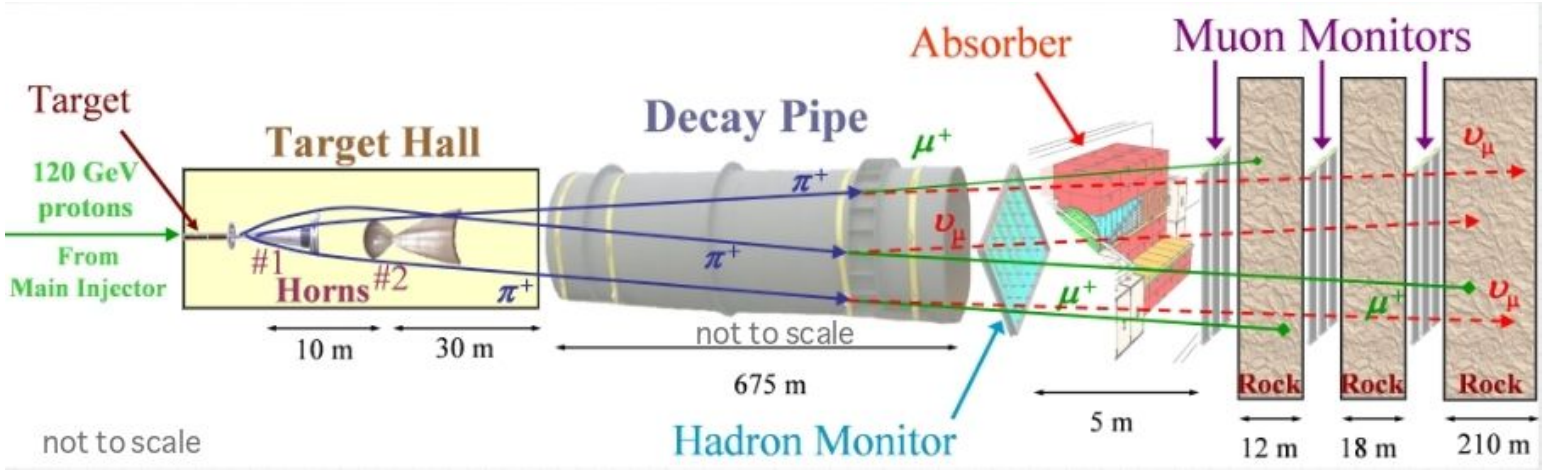
- Next step is to explore architecture improvements:
  - Weight sharing between views
    - Beam events are azimuthally symmetric, so the equivalents layers in the two view should use the same weights.
    - ~1/2 the number of free parameters
    - Disagreement between views may give discriminating power against cosmic rays.
  - Batch normalization (arXiv:1502.03167)
    - As data flows from layer to layer, it accumulates mean and variance shifts that slow training.
    - Makers of GoogLeNet created a layer that shifts the mean to zero and scales the variance to one over small batches of training samples.
    - Shown success at improving training speed, generalization, and final accuracy.
  - Improvements to the Inception module by replacing high order convolutions with a series of lower order ones (arXiv:1512.00567)
    - 7x7 convolutions can be replaced by 3 layers of 3x3 convolutions
    - 5x5 convolutions can be replaced by 2 layers of 3x3 convolutions
    - Other configurations like 1xN followed by Nx1 are also possible.
    - These configurations are computationally cheaper, and allow for extra non-linearities that improve the final accuracy.

Conclusions

- Modern deep learning techniques are very powerful.
- Using a modified GoogLeNet architecture, it is possible to build and train a neutrino event classifier that can achieve excellent signal and background separation for both ν_e appearance and ν_μ disappearance analyses.
  - Uses minimal reconstruction
  - In the ν_e case, achieved a 40% increase in efficiency with no loss in purity.
- GPU acceleration makes it possible to train complex networks in ~1 week.
- This technique is directly applicable to a number of analyses using other flavors or interaction types.
- Many more developments are in the works.
- More details available at A. Aurisano, A. Radovic, D. Rocco et. al. arXiv:1604.01444 (to be submitted to JINST).

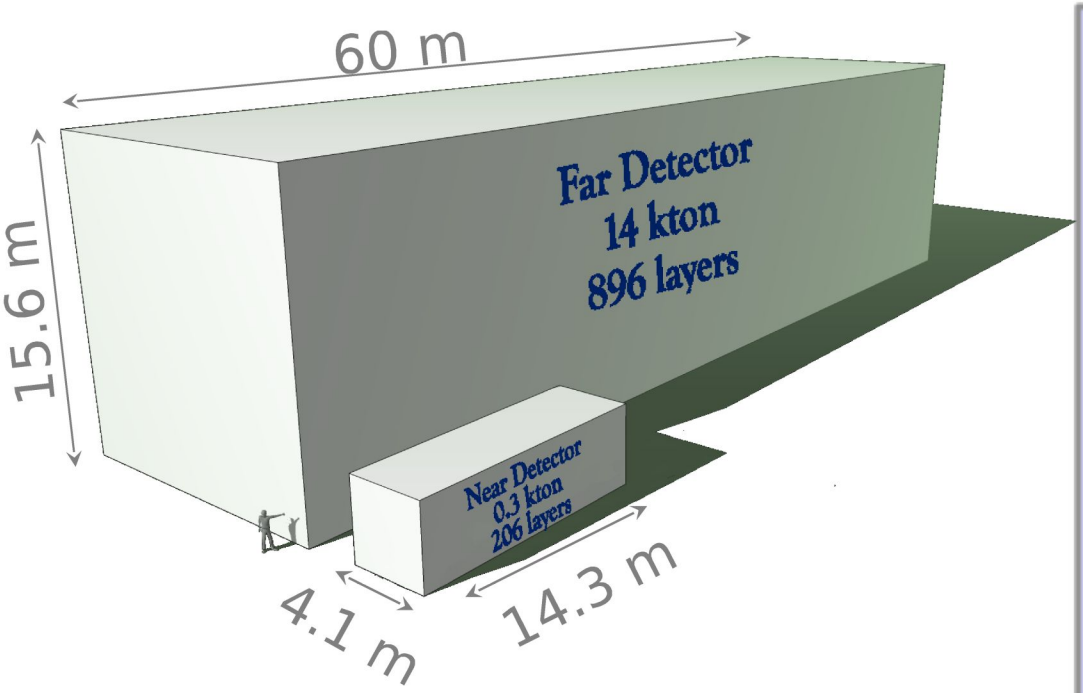
Backup

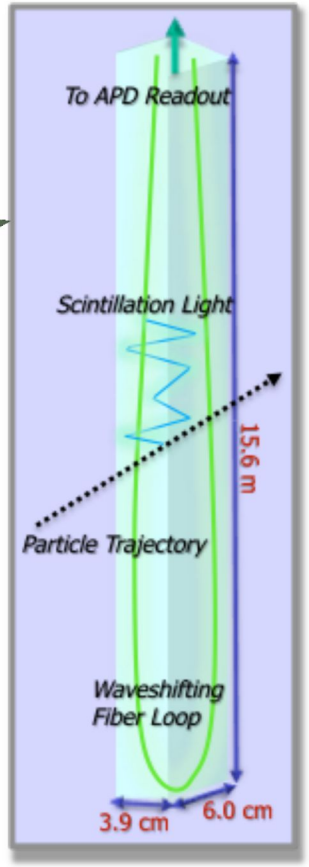
NuMI Beam



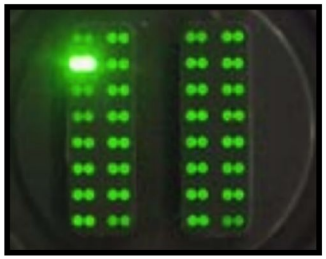
- Protons from the Main Injector at 120 GeV strike a graphite target.
- The resulting debris is focused by two magnetic horns.
- The particles travel down a decay pipe. At the end, only muons and neutrinos are left, and the muons are absorbed by rock.
  - The neutrinos are almost entirely muon neutrinos.

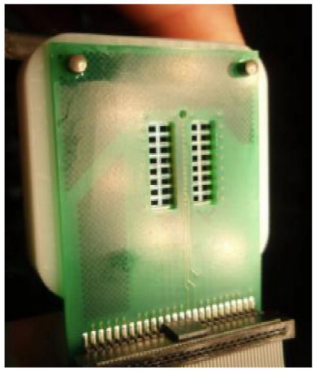
NOvA Design





Composed of alternating horizontal and vertical planes of liquid scintillator filled cells.



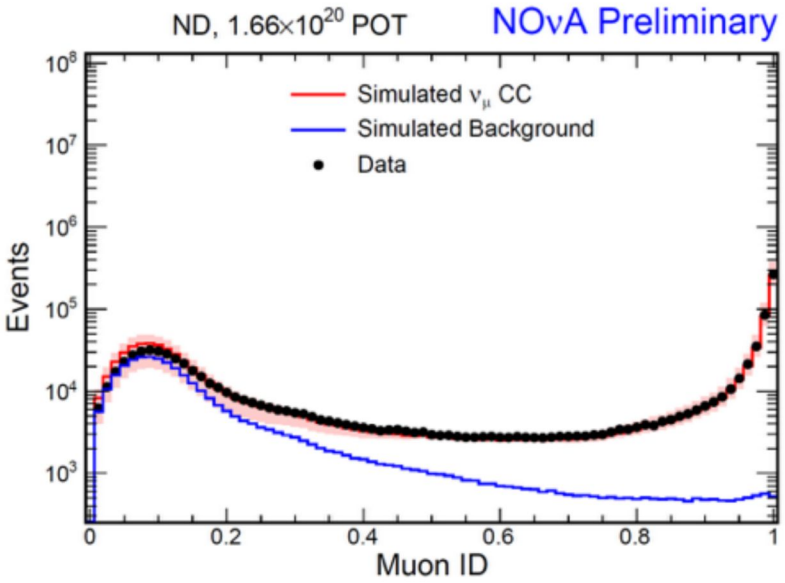


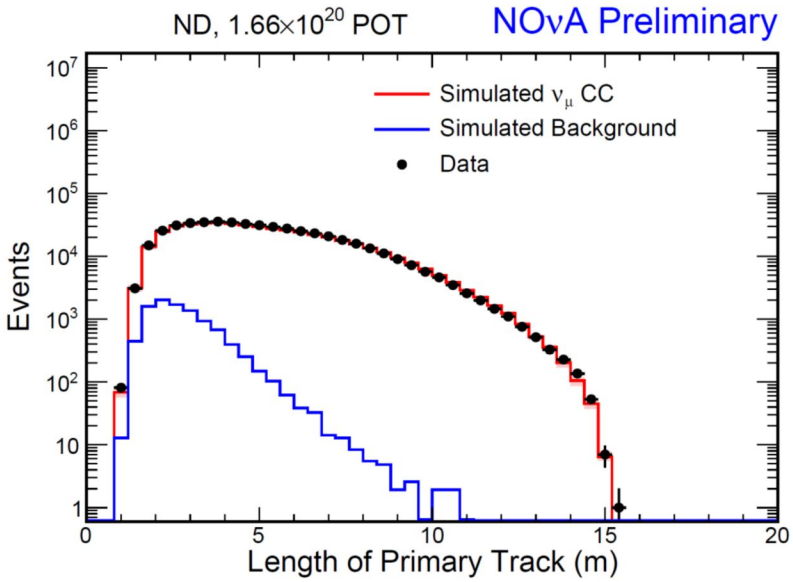
Wavelength shifting fibers carry light out of the cells to APDs

Far detector:
- 14 kton, low Z tracking calorimeter
- ~344,000 channels

Near detector:
- 0.3 kton
- Functionally equivalent to FD
- ~20,000 channels

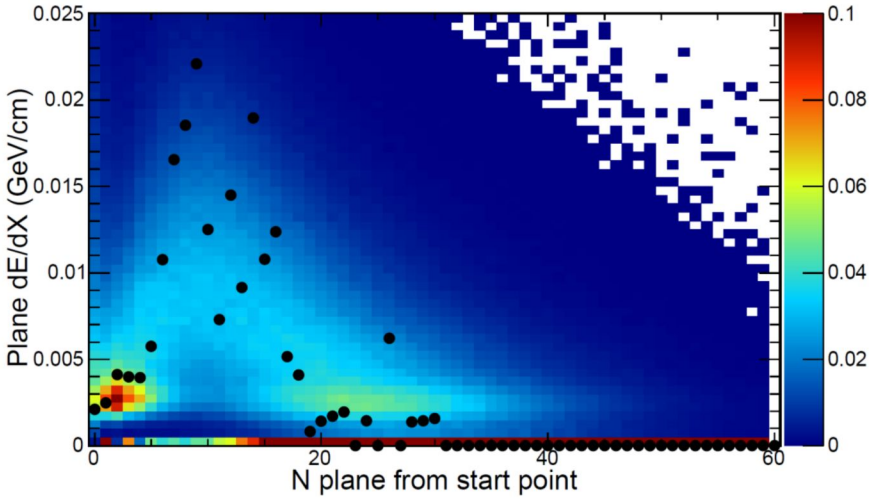
Current ν_μ Selector

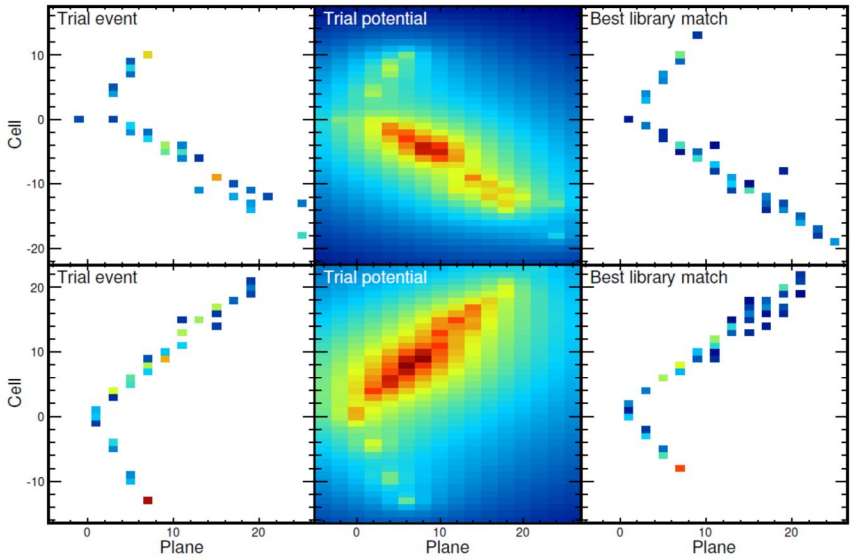




- ReMId: Use a 4 variable k-nearest-neighbor classifier to identify muon tracks.
  - track length
  - dE/dx along track
  - scattering along track
  - track-only plane fraction

Current ν_e Selectors



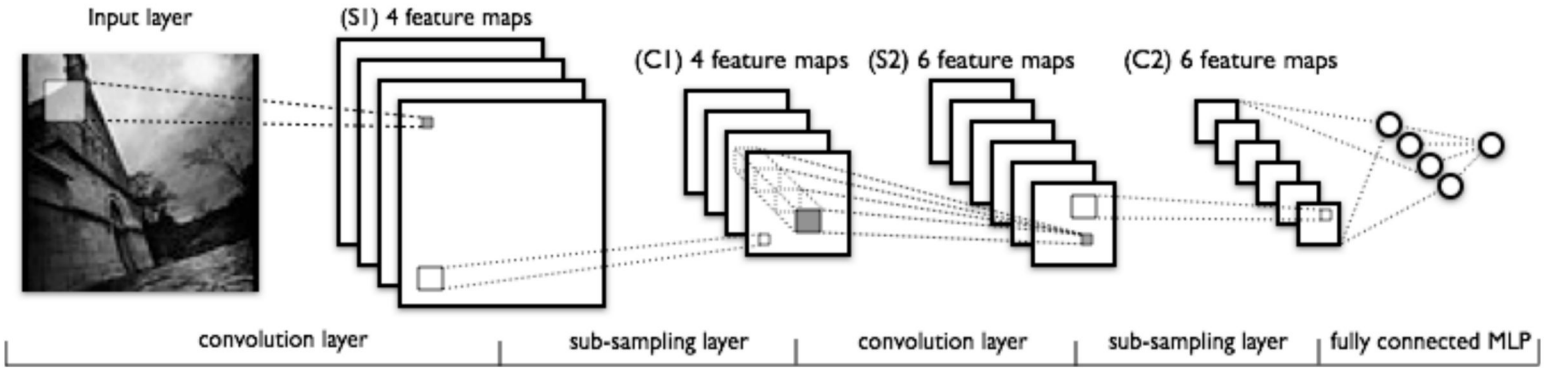


- LID
  - Calculates transverse and longitudinal dE/dx likelihoods for various particle hypotheses.
  - These, plus topological features, are fed into a standard neural network
- LEM
  - Finds best matches to a library of simulated events.
  - Properties of the best matches are fed into a decision tree.

Why Make a New Selector?

- We already have two ν_e selectors, why make another?
  - MC studies suggested that the events selected by LID and LEM only overlapped by ~70%.
  - Strongly suggests that there is information not being fully used.
- What properties would an ideal selector have?
  - Use inputs as close to raw data as possible to guard against reconstruction pathologies.
  - Make use of as much information as possible for each event.
  - Be robust against systematic uncertainties (like energy scale and scintillator non-linearities).
  - Provide classification values for all neutrino flavors, not just ν_e.
    - This would make a combination of ν_e appearance and ν_μ disappearance much easier.
- Deep learning provides a paradigm to achieve these goals.

LeNet-5 Model

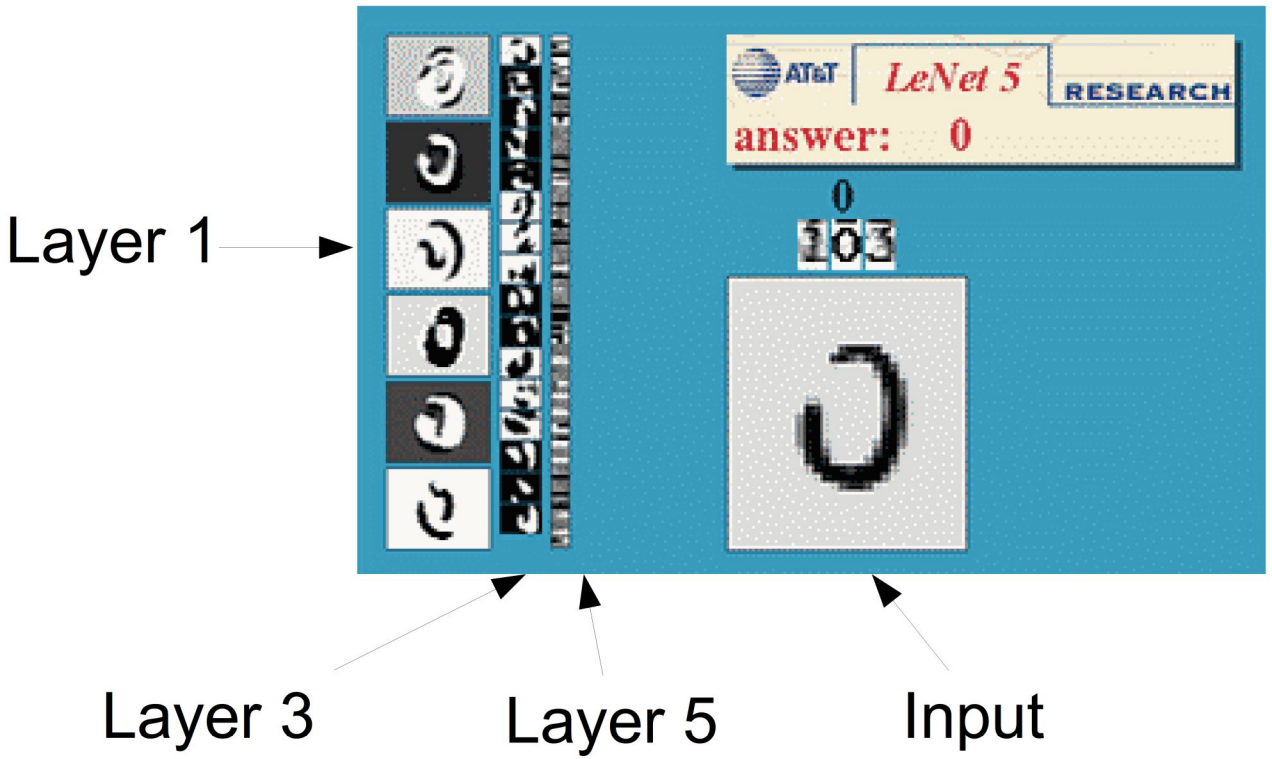


- CNNs have existed for a while now, but they've only recently become easy to use.
- One early example was the LeNet architecture.
- Composed of alternating convolutional and max pooling layers that ends in a fully connected MLP.
- Max pooling partitions the feature map into non-overlapping rectangles and downsamples by only keeping the maximum value contained in each.
- Max pooling + convolutional layers add a degree of translational invariance to the net.

Y. LeCun, L. Bottou, P. Haffner, Proceedings on the IEEE, 86(11), 2278-2324, (1998d)

LeNet-5 in Action

The LeNet-5 model was designed to identify handwritten digits in the MINST dataset



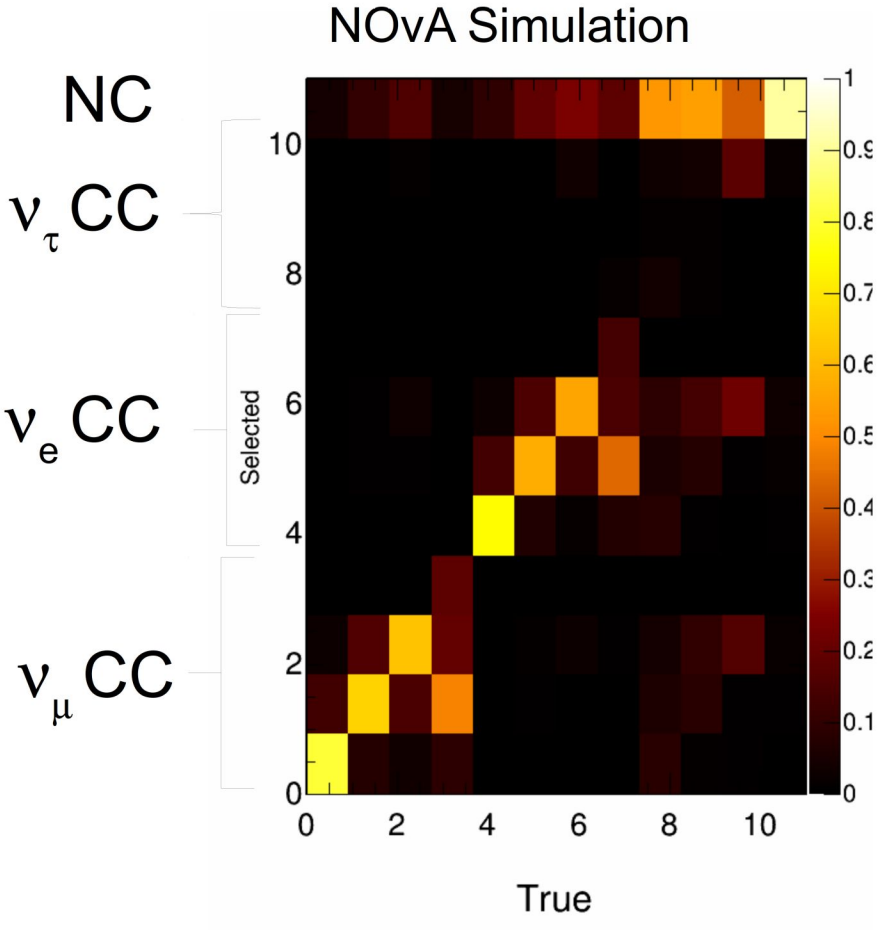
http://yann.lecun.com/exdb/lenet/

Each convolutional layer produces more feature maps but with smaller dimensions.

Features in the top layer are simple transformations of the input

Features become progressively more abstract as we move from input to output.

Initial Training

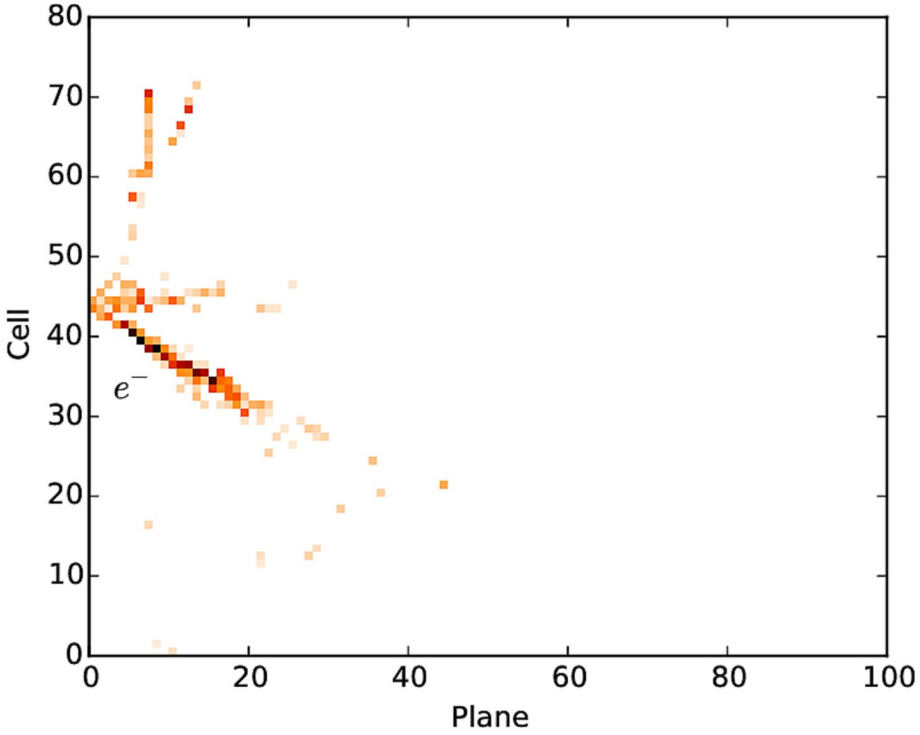


- The initial training consisted of ~10 passes over 3.4 million training images.
  - With two k40 GPUs, this took ~1 week.
- We can judge how well our model works by producing a confusion matrix.
  - This shows the relationship between the true event category and what the PID thought was the most likely event category.
- The matrix is mostly diagonal – events are mostly correctly identified.
- Mis-identified events mostly fall within blocks – while the interaction type is wrong, the selected neutrino flavor is still correct.

New Analysis Frontiers

- In addition to improving the ν_e and ν_μ analyses, this architecture is opening up new lines of investigation:
  - NC disappearance as a signature of sterile neutrinos.
  - ν_τ appearance at the near detector as a signature of sterile neutrinos.
    - Since ν_τ CC events only occur above 3.4 GeV, these events typically contain a large number of particles from nuclear fragmentation as well as the decay of the τ.
    - This analysis would be nearly impossible without CVN.
  - Classifying according to interaction type is potentially useful for cross-section measurements.

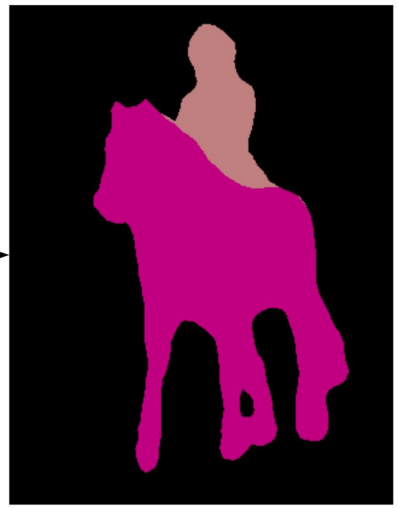
Prong Identification



- While CVN uses minimal reconstruction to classify whole events, it would still be useful to understand the particle content of the final state for each event.
- In NOvA reconstruction, objects with an identified start point and a direction are referred to as prongs.
- A prong reconstruction algorithm already exists.
  - Can we use a CVN-like classifier to classify prongs according to what particle created them?
  - Work ongoing, but it looks promising.

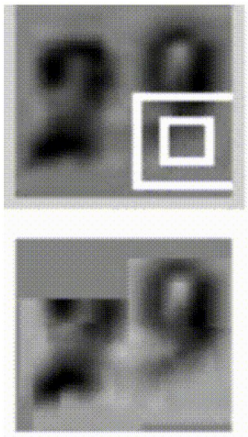
Semantic Segmentation

arXiv:1411:4038

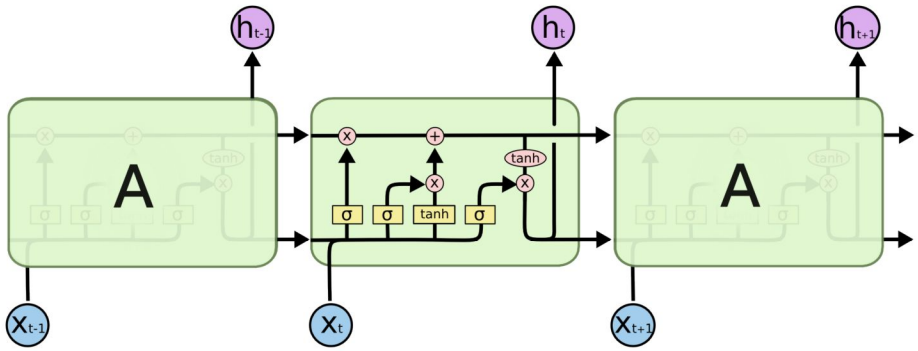


- Semantic segmentation refers to the process of labeling pixels according to what object they are a part of.
- Cutting edge research has demonstrated that this is possible using convolutional neural networks.
  - Extract information from all pixels across all layers corresponding to the pixel of interest (a hyper column of pixels)
- This could potentially turn reconstruction on its head
  - Instead of reconstructing objects and then identifying them, label all pixels according to what they likely came from first, and then use that information to assist reconstruction.
  - Instead of clustering in time and space, cluster in time, space, and PID.
  - Could potentially make it easier to reconstruct neutron interactions that are not well connected to the event vertex.

Recurrent Neural Networks



http://karpathy.github.io/2015/05/21/rnn-effectiveness/



http://colah.github.io/posts/2015-08-Understanding-LSTMs/

- Neural networks with loops in them are called recurrent.
- These architectures have a concept of time since information propagates around these loops once per time step.
- Ideal for problems with indeterminate input or output length.
- Has been used to scan over complicated images to decrease the computational complexity of each given subsample.
- Could be used to incorporate temporal information to distinguish up and downward going muon tracks (to look for atmospheric neutrinos) or to incorporate information from late interactions like neutrons and Michel electrons.
- Possibly also useful for online triggering.