Deep learning in NOvA and DUNE

Rutherford Appleton Laboratory
January 16, 2019

Chris Backhouse
University College London

Neutrino physics in a nutshell
NOvA and DUNE experiments
Neural networks
Deep learning
NOvA and DUNE applications
Future directions

Neutrino physics in a nutshell
NOvA and DUNE experiments
Neural networks
Deep learning
NOvA and DUNE applications
Future directions

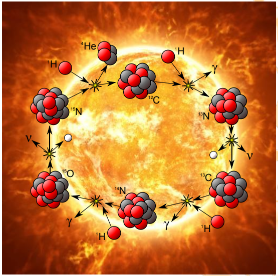
Using Modern Deep Learning Techniques to Categorize Neutrino Interactions
– A. Aurisano @ SLAC

Convolutional Neural Networks in Neutrino Analyses – A. Radovic @ FNAL

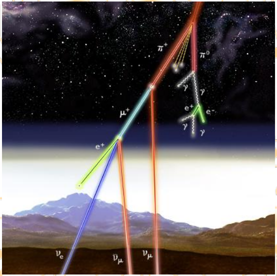
Deep Learning Applications on NOvA
– F. Psihas @ DPF 2017

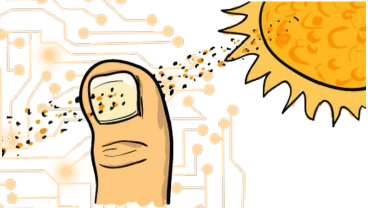
Neutrinos are everywhere

Solar



Atmospheric





FACT: about 65 million neutrinos pass through your thumbnail every second.

Reactor

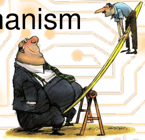
Supernova

- Second most abundant particle in the universe
- But we know almost nothing about them
- Only interact via the weak force
- Need powerful sources and huge detectors

Neutrinos are unique

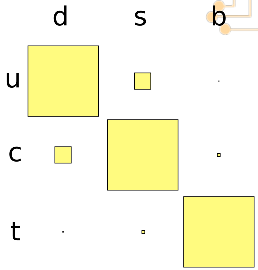
- Far lighter than the quarks and charged leptons
- May get their masses by a different mechanism

$$m_{\mathrm{EW}}^2/m_\nu \sim 10^{15}\,\mathrm{GeV} \sim m_{\mathrm{GUT}}$$

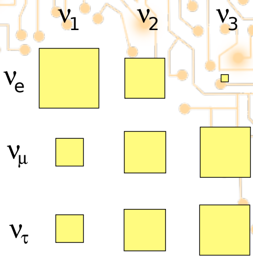


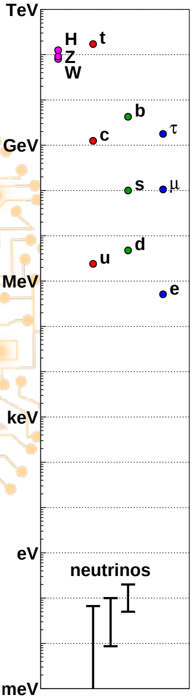
- Very different mixing structure to quarks
- Most of what we know comes from neutrino **oscillations** arising from this mixing

Quarks



Neutrinos

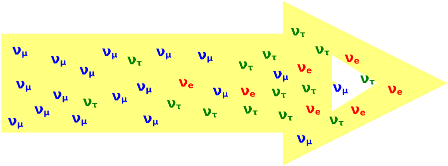


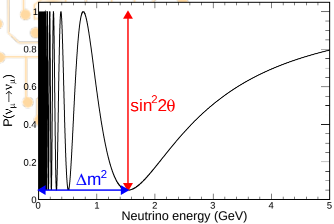


Neutrino mixing and oscillation

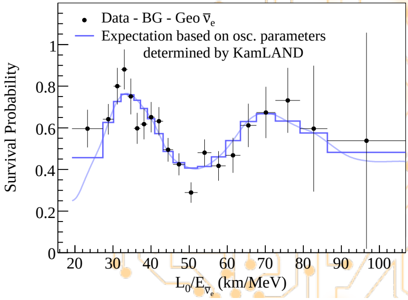
- Neutrinos mix, like quarks
$$|\nu_\alpha\rangle = \sum_i U^\star_{\alpha i}|\nu_i\rangle$$
- Unlike quarks, mixings large

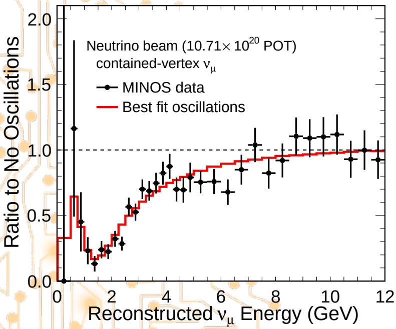
$$|\nu_\alpha\rangle = \cos\theta|\nu_1\rangle + \sin\theta|\nu_2\rangle \quad \rightarrow \quad P(\nu_\alpha \rightarrow \nu_\alpha) = 1 - \sin^2 2\theta \sin^2\left(\frac{\Delta m^2 L}{4E}\right)$$

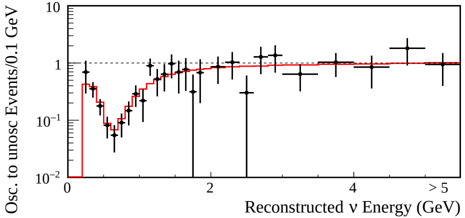


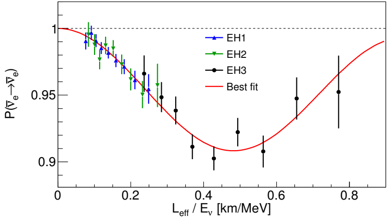


Oscillation structure





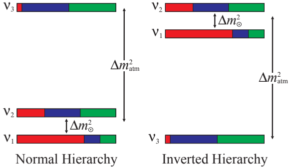




NOvA

- Powerful ν_μ beam from Fermilab
- Measure flux in Near Detector
- Measure again at Far Detector for $P(\nu_\mu \to \nu_\mu)$ and $P(\nu_\mu \to \nu_e)$

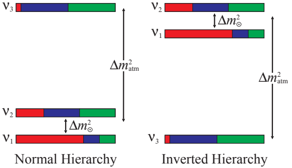
- World's highest power ν beam
- Longest baseline of any expt. maximizes sensitivity to mass ordering

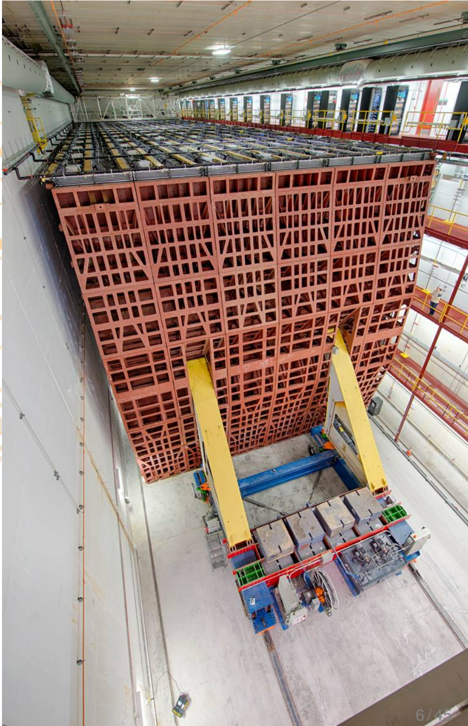


NOvA

- Powerful ν_μ beam from Fermilab
- Measure flux in Near Detector
- Measure again at Far Detector for $P(\nu_\mu \to \nu_\mu)$ and $P(\nu_\mu \to \nu_e)$

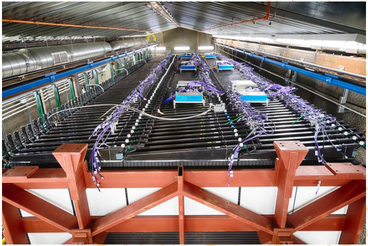
- World's highest power ν beam
- Longest baseline of any expt. maximizes sensitivity to mass ordering

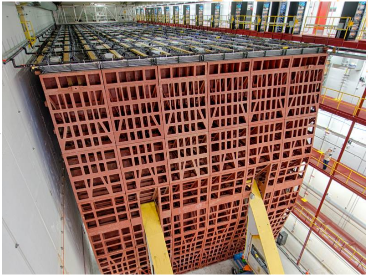




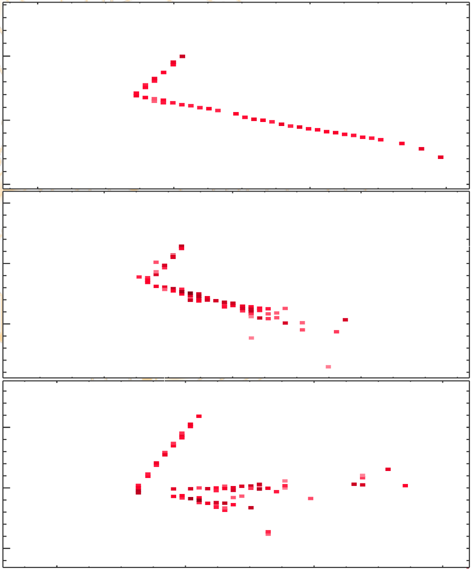
NOvA

ν_μ



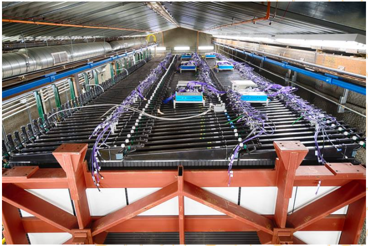


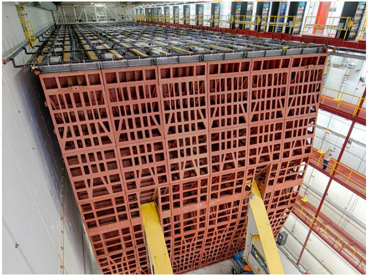
ν_μ
$+$
ν_e



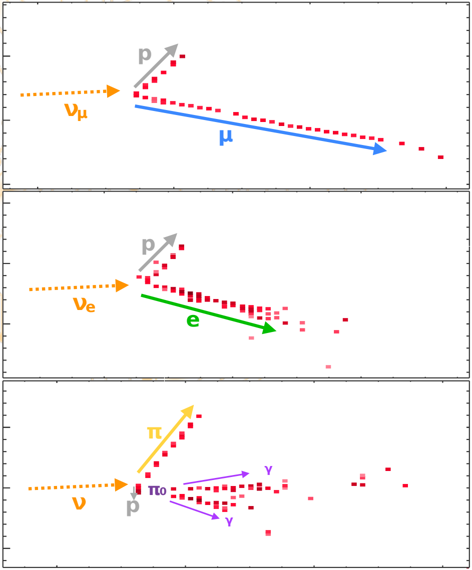
NOvA

ν_μ



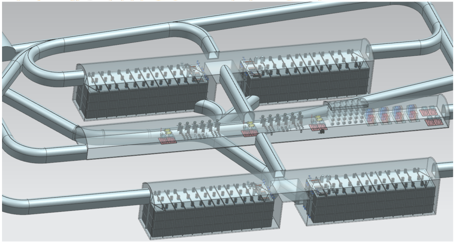


$\nu_\mu + \nu_e$



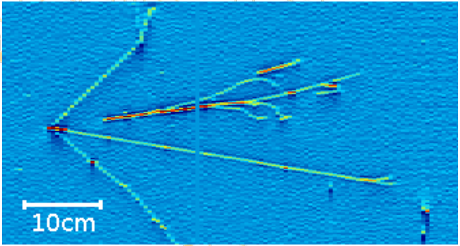
DUNE

- **More powerful** beam
- **Longer** baseline
- **Deep** underground
- **Larger** detector
- **Finer** segmentation



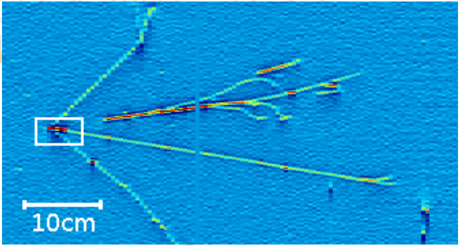
DUNE

- **More powerful** beam
- **Longer** baseline
- **Deep** underground
- **Larger** detector
- **Finer** segmentation



DUNE

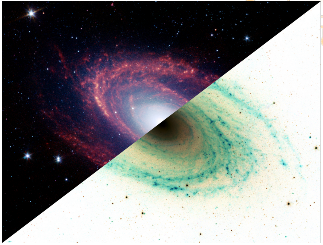
- **More powerful** beam
- **Longer** baseline
- **Deep** underground
- **Larger** detector
- **Finer** segmentation

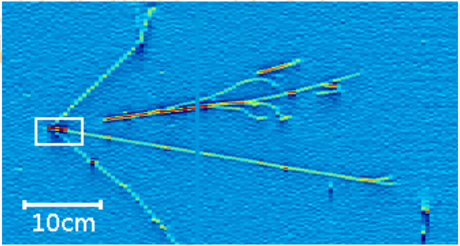


DUNE

- **More powerful** beam
- **Longer** baseline
- **Deep** underground
- **Larger** detector
- **Finer** segmentation

- Primary goal to discover if $\nu/\bar{\nu}$ oscillations differ (5σ level)

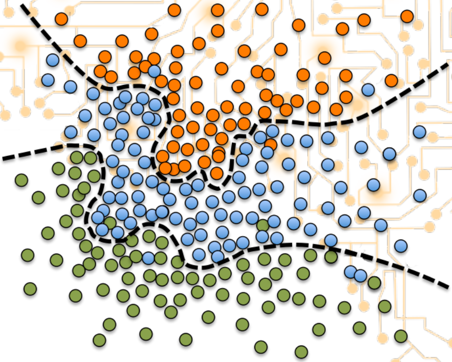




Classifiers

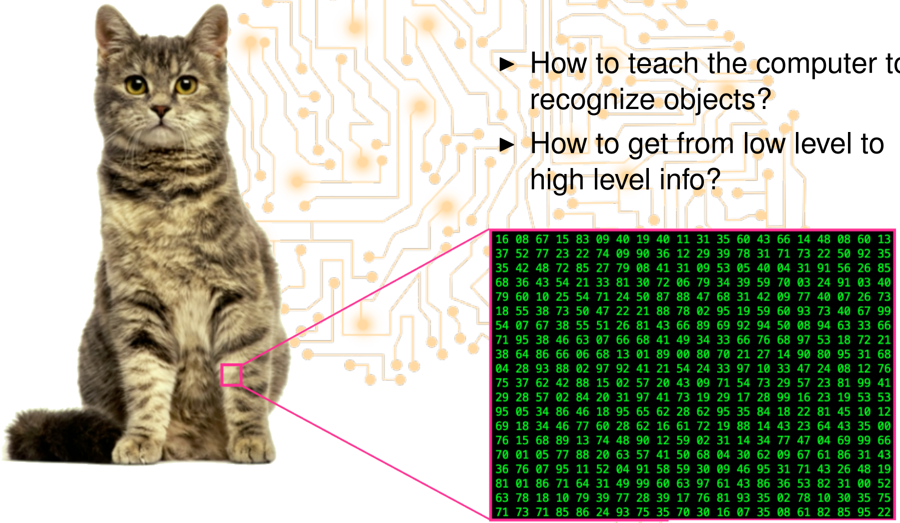
- In general want $P(\text{physics}|\text{observations})$

- Want to reduce huge-dimensional space
- Smooth P so don't need infinite MC stats



Classifiers

- How to teach the computer to recognize objects?
- How to get from low level to high level info?



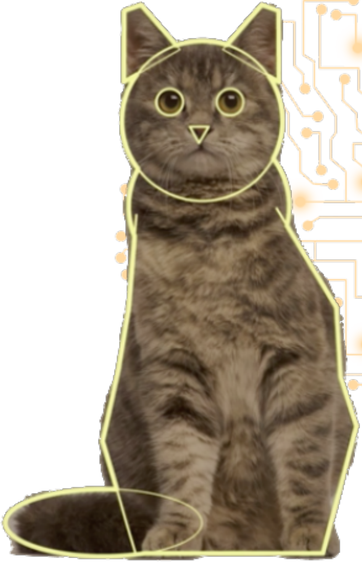
Credit: Fei Fei Li @ TED via Kazu Terao @ NNN18

Traditional approach

- Write algorithms to find features

Credit: Fei Fei Li @ TED via Kazu Terao @ NNN18

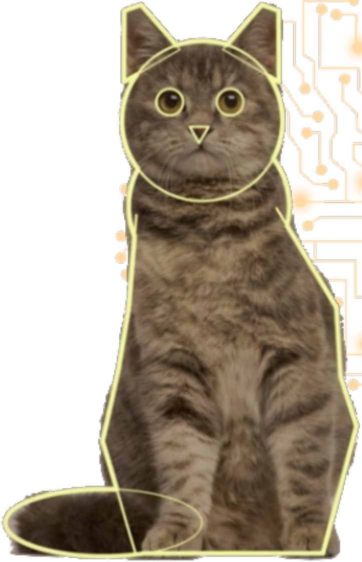
Traditional approach



- Write algorithms to find features
- Define object as feature combo
- Test
- Search for pathologies
- Add special-cases / new algorithms

Credit: Fei Fei Li @ TED via Kazu Terao @ NNN18

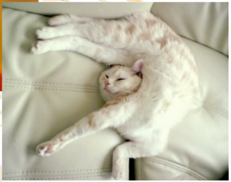
Traditional approach



- Write algorithms to find features
- Define object as feature combo
- Test
- Search for pathologies
- Add special-cases / new algorithms

Partial cat
(particle escaping fiducial volume)

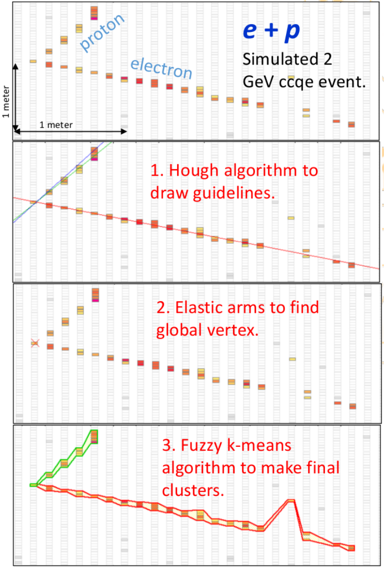
Stretching cat
(Nuclear FSI)



- How about cases like these?

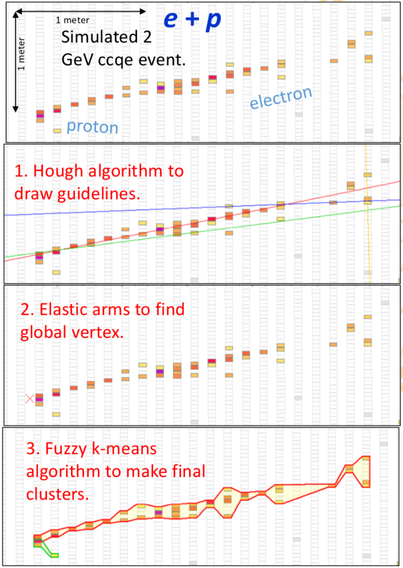
Credit: Fei Fei Li @ TED via Kazu Terao @ NNN18

NOvA event reconstruction



- First cluster hits in space and time
- Start with 2-point Hough transform
  - Line-crossing are vertex seeds
- ElasticArms finds vertex
- Fuzzy *k*-means clustering forms prongs
- ν_μ analysis uses a Kalman filter to reconstruct any muon track

NOvA event reconstruction

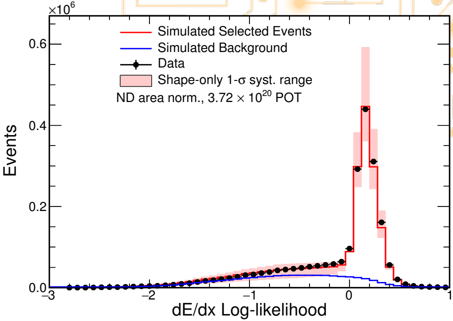


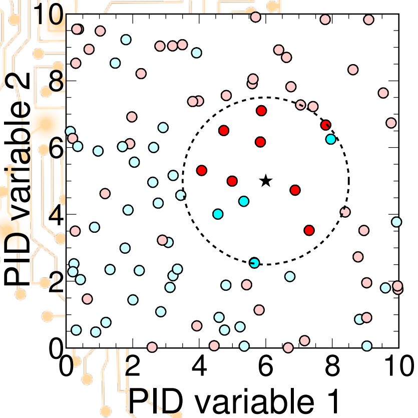
- First cluster hits in space and time
- Start with 2-point Hough transform
  - Line-crossing are vertex seeds
- ElasticArms finds vertex
- Fuzzy *k*-means clustering forms prongs
- ν_μ analysis uses a Kalman filter to reconstruct any muon track

NOvA "classic" PIDs

ν_μ PID

- kNN based on dE/dx and scattering LLs, track length, etc.

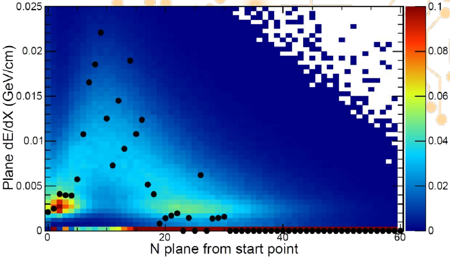


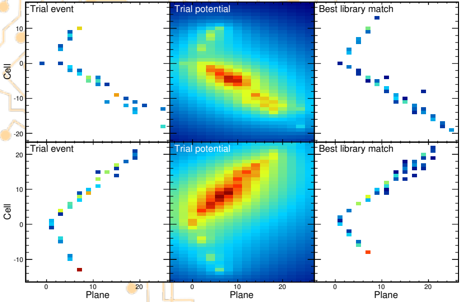


NOvA “classic” PIDs

ν_e PIDs

- LID: ANN based on shower dE/dx LLs
- LEM: “kNN” over library events + decision tree
- $\sim$ 70% LID/LEM overlap – room for improvement?

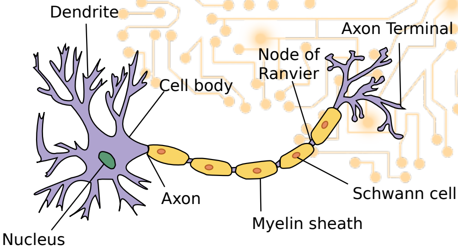


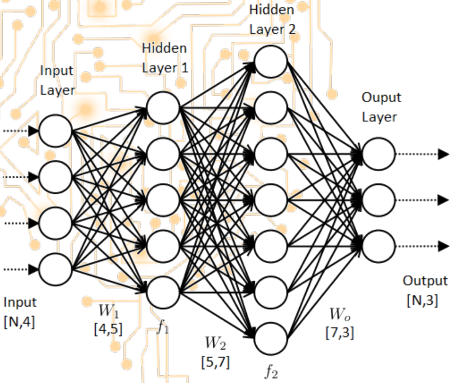


C. Backhouse, R. Patterson, NIM A778 (2015) 31-39

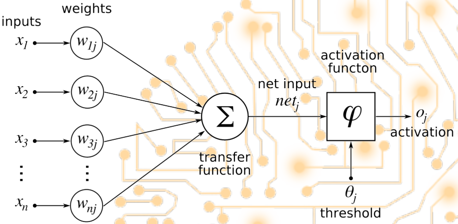
Artificial Neural Networks

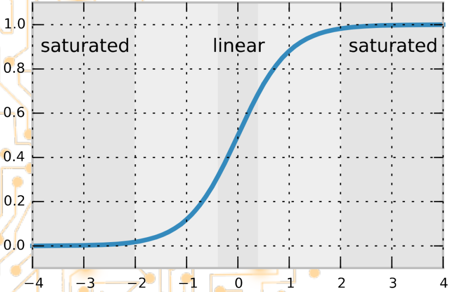
- Origins back in the 40s
- **Loosely** model the neurons in a brain





Artificial Neural Networks





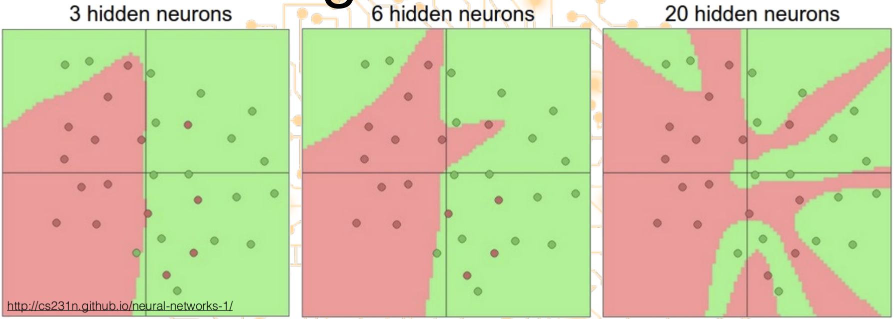
- N training events with input properties $\vec{x}_i$ and truth y_i
- Aim to minimize a loss function
- Squared error (regression):

$$L = \sum_i (y_i - f(\vec{x}_i))^2$$

- Cross entropy (classification):

$$L = \sum_i -y_i \log(f(\vec{x}_i)) - (1 - y_i) \log(1 - f(\vec{x}_i))$$

ANN training

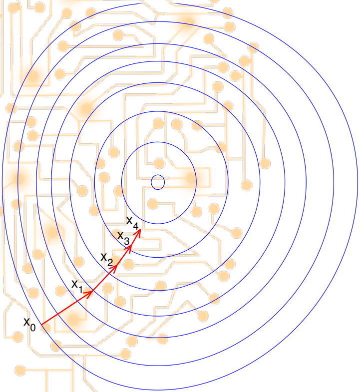


- Single layer with enough nodes can reproduce any function
  - Physicist's proof: use $2N$ neurons to build a delta function
- Multi-layer often need fewer nodes

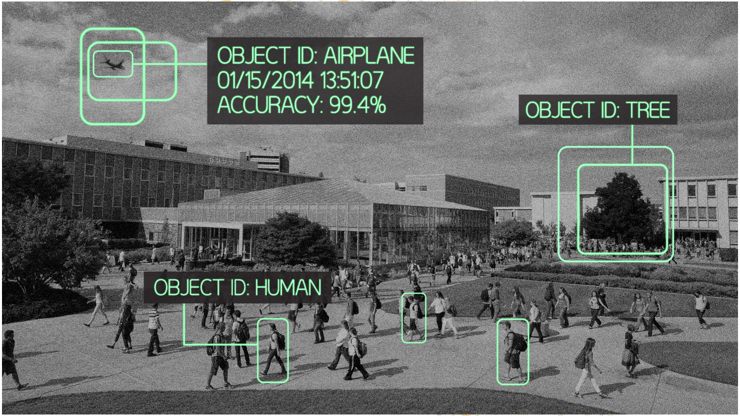
- How to train?
- Fully connected → number of parameters grows quickly

Backpropagation

- First applied to NNs in 1982
- Compute partial derivative of loss w.r.t. each weight $\frac{\partial L}{\partial w_i}$
- Optimize loss via gradient descent
- Adjust weights
  learning rate $\times$ gradient $\times$ loss
  $w_j' = w_j - \alpha \nabla_{w_j} L$
- Enjoyed a lot of success in HEP
- Recently overtaken by BDTs



Convolutional Neural Networks



- Recent advances in machine learning/computer vision
- Achieving near-human performance on image classification tasks
- Can we do better by classifying event-displays directly?

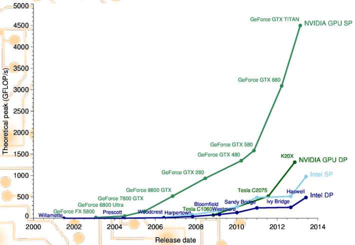
Deep learning

- *Deep* just means many hidden layers
- Can encode complex structures more efficiently
- Historically extremely difficult to train

- Various advances
  - GPUs - Bring more raw power to bear on training
  - Bigger training sets
  - Better weight initialization
  - Better nonlinearities
  - Stochastic gradient descent
  - Techniques to prevent overtraining
  - Convolutional networks – reduction in number of weights to train

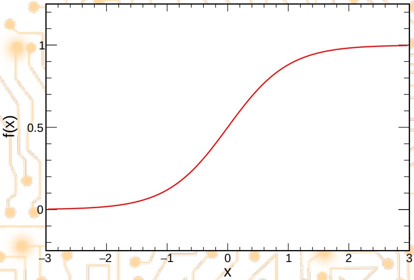
Deep learning

- *Deep* just means many hidden layers
- Can encode complex structures more efficiently
- Historically extremely difficult to train



- Various advances
  - GPUs - Bring more raw power to bear on training
  - Bigger training sets
  - Better weight initialization
  - Better nonlinearities
  - Stochastic gradient descent
  - Techniques to prevent overtraining
  - Convolutional networks – reduction in number of weights to train

Training improvements



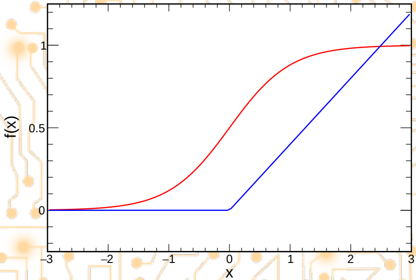
ReLU

- With traditional response function, saturated neuron $\partial L/\partial w_i \to 0$ stops training
- "Rectified linear unit" – more effective backpropagation
- Bonus: more efficient calculation

Stochastic gradient descent

- Training convenience: evaluate small batches of events
- Approximate result as noisy sub-estimates even out
- Bonus: can allow for jumping out of local minima

Training improvements



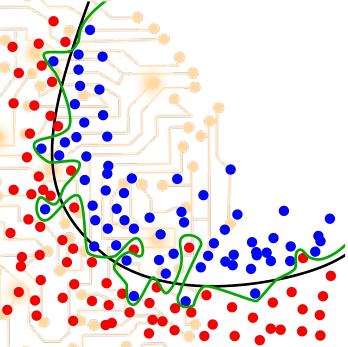
ReLU

- With traditional response function, saturated neuron $\partial L / \partial w_i \to 0$ stops training
- "Rectified linear unit" – more effective backpropagation
- Bonus: more efficient calculation

Stochastic gradient descent

- Training convenience: evaluate small batches of events
- Approximate result as noisy sub-estimates even out
- Bonus: can allow for jumping out of local minima

Training improvements



- Powerful classifiers risk overfitting

Regularization

- Add term $\lambda \sum w_i^2$ to loss
- Disfavours large weights

Dropout

- At each training iteration randomly set X% of weights to zero
- Weights not reliably used together so can't be strongly correlated

---

Moody *et al.* "A simple weight decay can improve generalization"
Srivasta *et al.* "Dropout: A Simple Way to Prevent Neural Networks from Overfitting"

Convolutional Neural Networks

$$\frac{1}{8}\begin{bmatrix} -1 & -1 & -1 \\ -1 & +8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Edge-detection kernel

- Early neurons in visual cortex sensitive to small “receptive field”
- **CNN** – deep neural network, inputs are the pixels of the image
- Enforce translational invariance → convolutions
- Learn optimal kernels direct from data

Convolutional Neural Networks

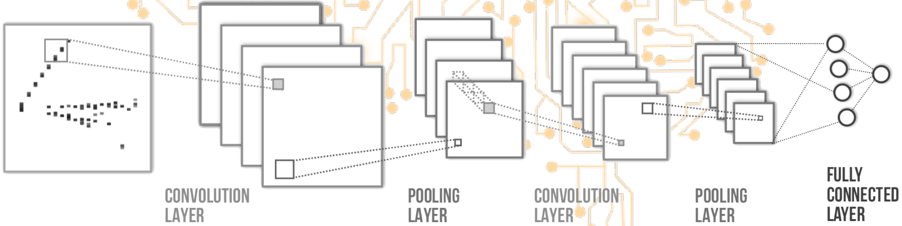
$$\frac{1}{8}\begin{bmatrix} -1 & -1 & -1 \\ -1 & +8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Edge-detection kernel

- Early neurons in visual cortex sensitive to small “receptive field”
- **CNN** – deep neural network, inputs are the pixels of the image
- Enforce translational invariance → convolutions
- Learn optimal kernels direct from data

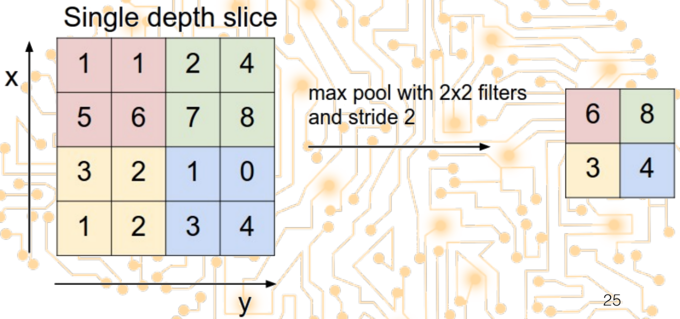
Convolutional Neural Networks

- Early CNN example: LeNet: Circa 1989
- Alternating convolution and max-pooling layers (downsampling)
- Finish with fully-connect network
- Max-pooling + convolution $\rightarrow$ translational invariance
- Convolutional layer trains N$\times$M$\times$W$\times$H coefficients



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

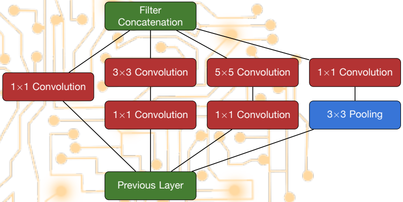
Pooling



- Pooling downsamples information (form of smoothing)
- Max or average of a patch of pixels
- Literal smoothing if stride=1

Inception modules

- GoogLeNet – 2014
- “Inception module”
- Combine different kernel sizes, keep number of maps under control with 1×1 convolutions
- Max pooling downsamples
- Reduce number of feature maps with $1 \times 1 \times N \rightarrow 1$

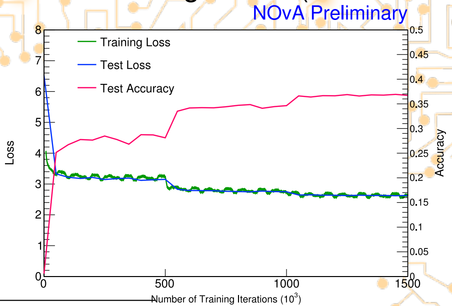


C. Szegedy *et al.*, “Going Deeper with Convolutions”, arXiv:1409.4842 (2014)

NOvA's network – CVN

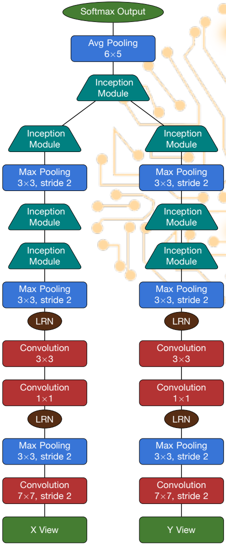
- **C**onvolutional **V**isual **N**etwork
- Turn NOvA events into pixel map: 100×80 (14.5m×4m) box
- Downsample charges to one byte (256 values)
- Inputs differ substantially to natural images *e.g.* many zero pixels

- Train to distinguish neutrino flavours (and interaction modes)
- 10 passes over 3.4m training events (1 week with two (k40) GPUs)



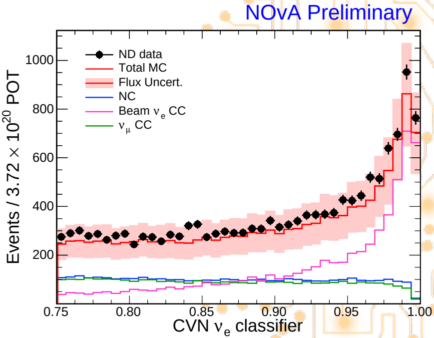
"A Convolutional Neural Network Neutrino Event Classifier" JINST vol 11 (2016)

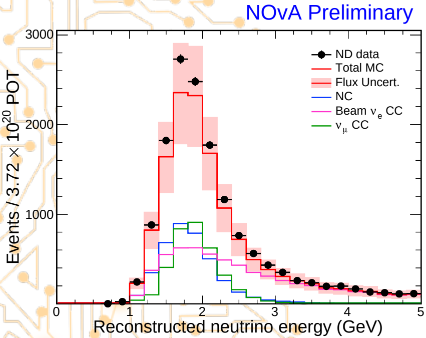
CVN architecture



- Usually have multiple “channels” for RGB
- Our views approx independent, don’t want linear combinations of unrelated info
- “Siamese” network, $\sim$ cut-down GoogLeNet
- Network topologies an intense research area
- Later CVN iterations have somewhat varying layer structures

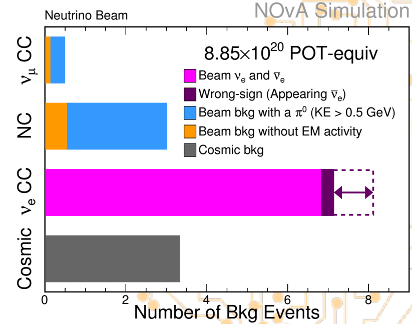
CVN performance

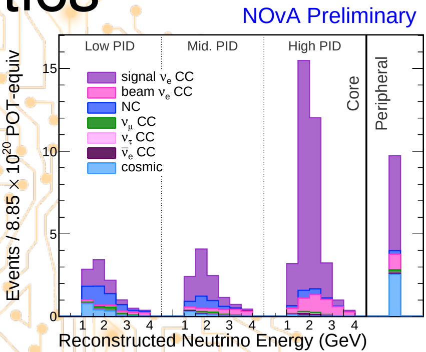


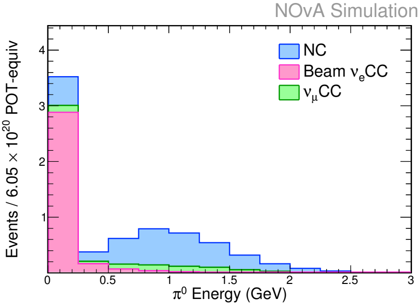


- Statistical power equivalent to collecting 30% more data
- Also improves ν_μ CC selection and adopted by NC group
- Systematic studies show same or less sensitivity to uncertainties
- Good data/MC agreement observed in Near Detector

CVN characteristics

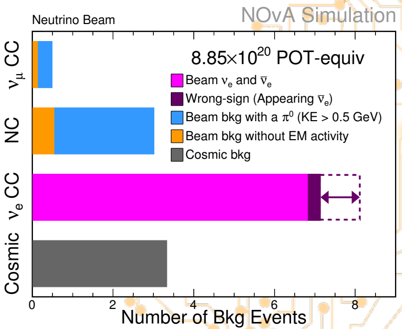


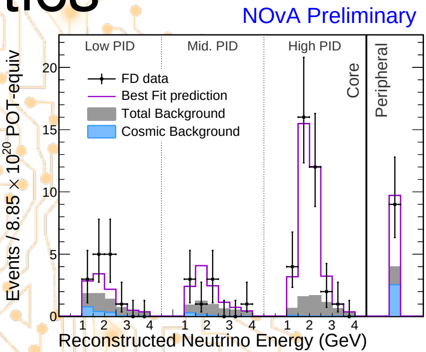


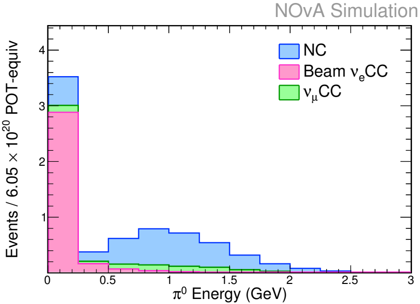


- Data analysis divides data into purity bins by CVN value
- Surviving backgrounds mostly contain energetic π^0 as expected

CVN characteristics



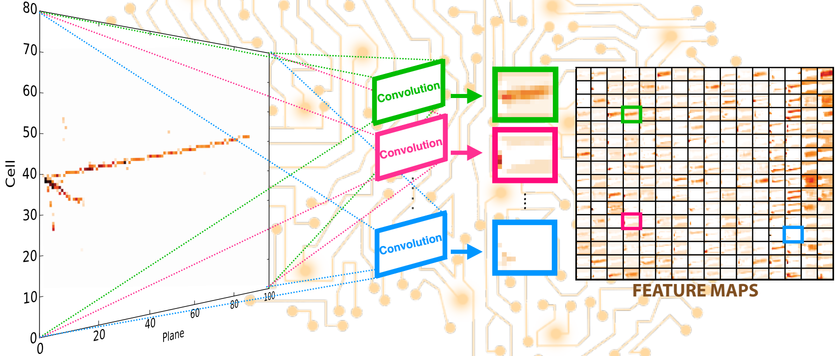




- Data analysis divides data into purity bins by CVN value
- Surviving backgrounds mostly contain energetic π^0 as expected

Inside the black box – inspect

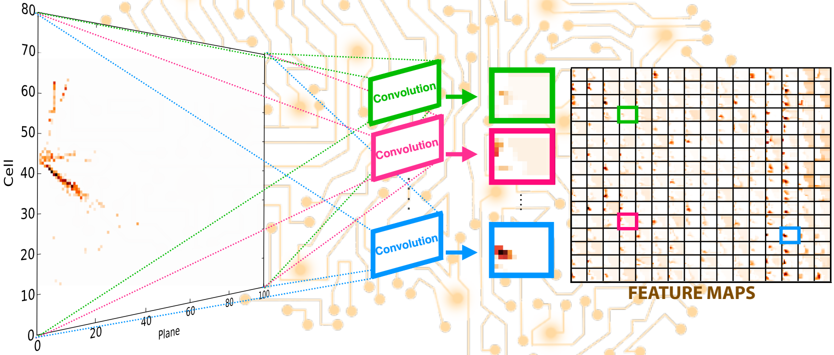
ν_μ CC



- Direct inspection of first network layer
- Some features sensitive to tracks, others showers

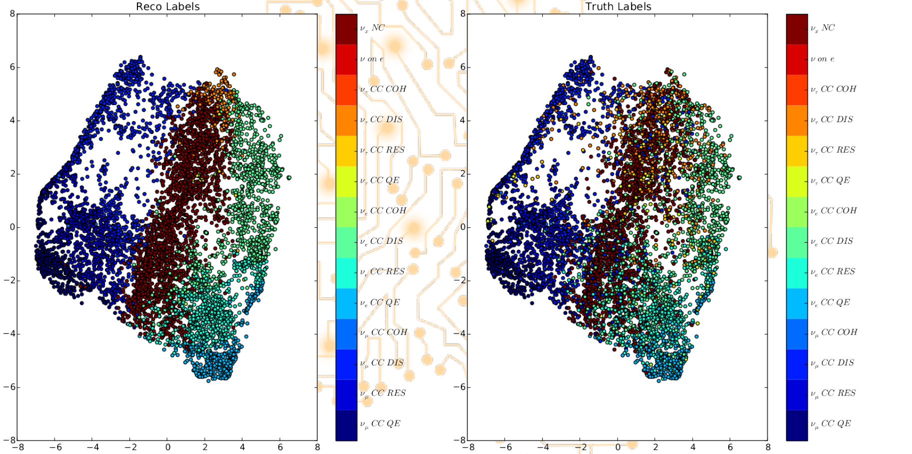
Inside the black box – inspect

ν_e CC



- Direct inspection of first network layer
- Some features sensitive to tracks, others showers

Inside the black box – t-SNE



- Lower-dimensional subspace contains much of the information
- *e.g.* principal components on CVN features
- Or non-parameteric "t-distributed stochastic neighbor embedding"

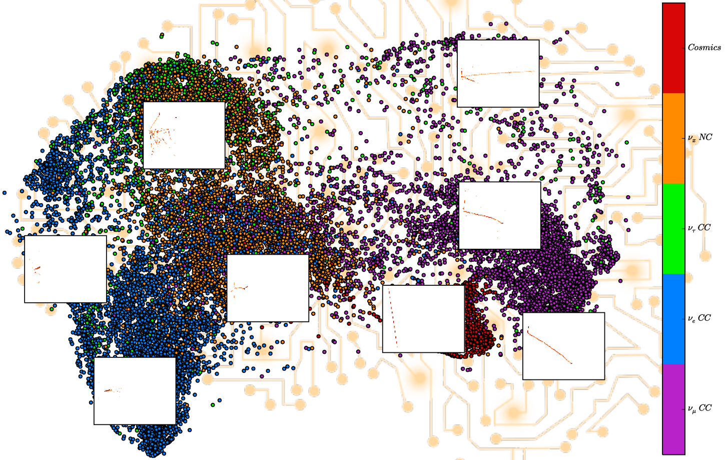
van der Maaten *et al.* "Visualizing High-Dimensional Data Using t-SNE"

Inside the black box – t-SNE

- Lower-dimensional subspace contains much of the information
- *e.g.* principal components on CVN features
- Or non-parameteric "t-distributed stochastic neighbor embedding"

van der Maaten *et al.* "Visualizing High-Dimensional Data Using t-SNE"

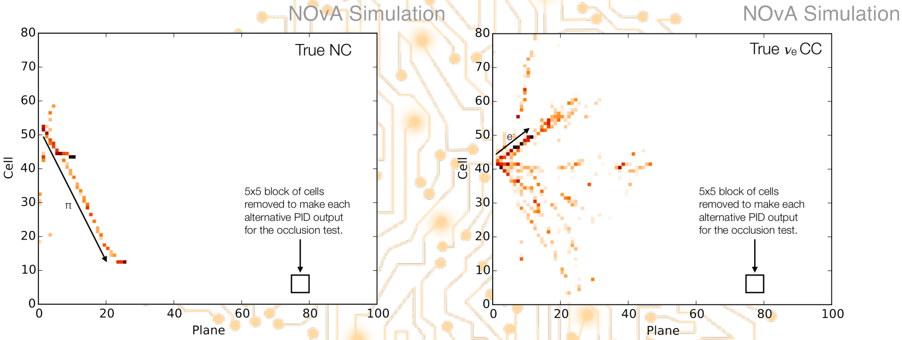
Inside the black box – t-SNE



- Lower-dimensional subspace contains much of the information
- *e.g.* principal components on CVN features
- Or non-parameteric "t-distributed stochastic neighbor embedding"

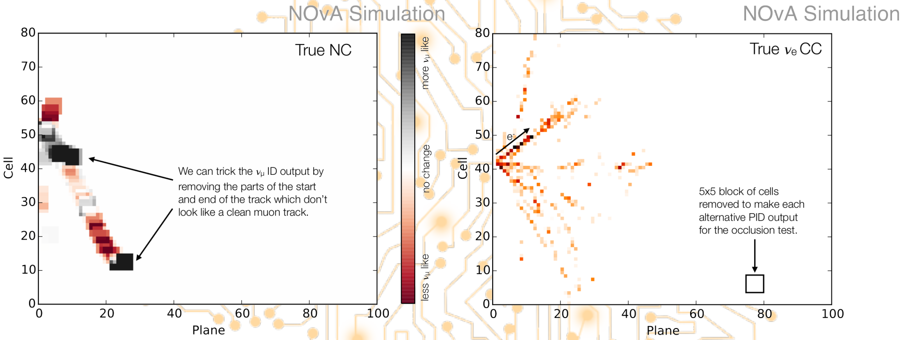
van der Maaten *et al.* "Visualizing High-Dimensional Data Using t-SNE"

Inside the black box – occlusion



- Which pixels in the input are important to the result?
- Which are irrelevant?

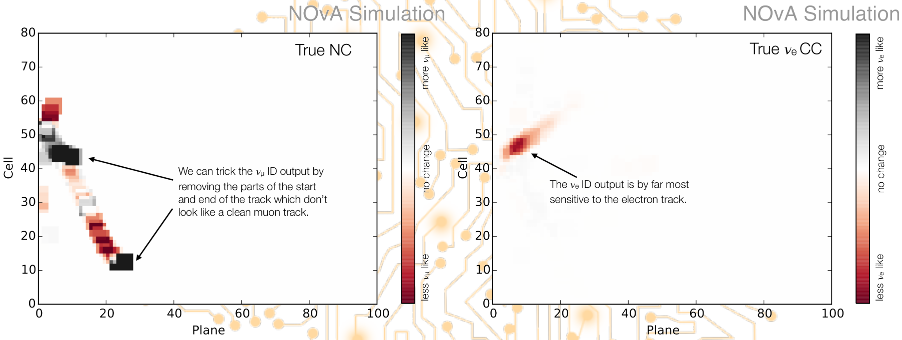
Inside the black box – occlusion



- Which pixels in the input are important to the result?
- Which are irrelevant?

- ν_μ PID most focused on cleanliness of track

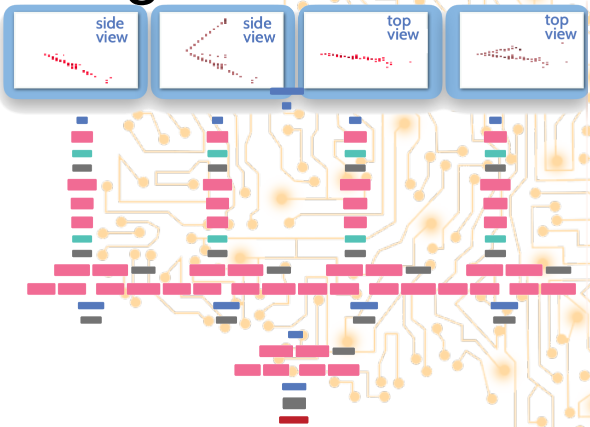
Inside the black box – occlusion

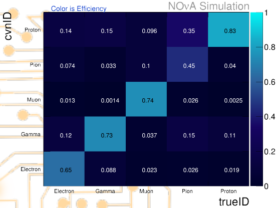


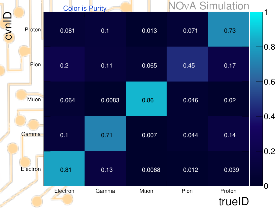
- Which pixels in the input are important to the result?
- Which are irrelevant?

- ν_μ PID most focused on cleanliness of track
- ν_e PID dominated by the EM shower

Prong CVN

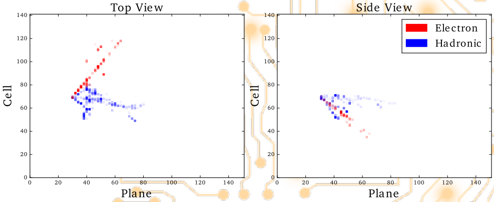


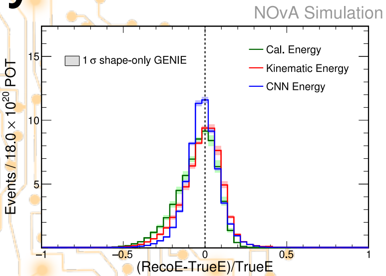




- Train network on individual prongs (from trad. reco) plus context
- Goal is to classify individual particles within the event
- Performance dependent on purity of traditional reconstruction
- In use for energy estimator, in future for xsec measurements
- Not to be confused with “final state CVN”

Regression energy estimator

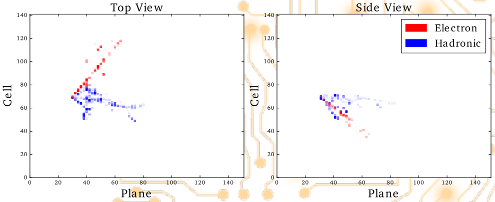


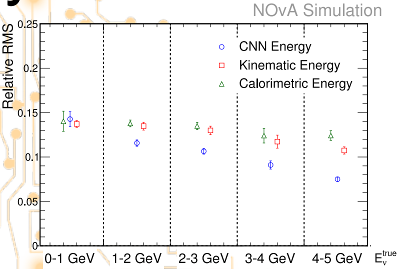


- Traditional technique attempts to seperate EM and hadronic hits, apply different scale factors
- 1m simulated ν_e interactions, flat across energies
- Train with loss $L = \frac{1}{N}\sum_i \left|\frac{f(x_i)-y_i}{y_i}\right|$
- Cautious about systematic biases
  - Haven't found anything dramatic yet

"Improved Energy Reconstruction in NOvA with Regression Convolutional Neural Networks", accepted by Phys Rev D

Regression energy estimator

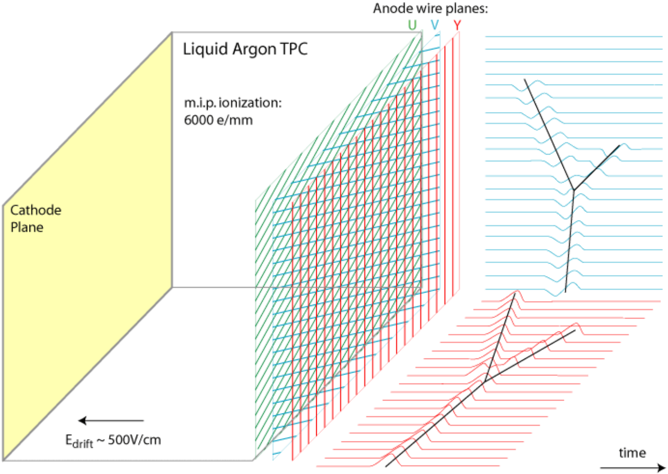


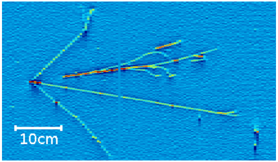


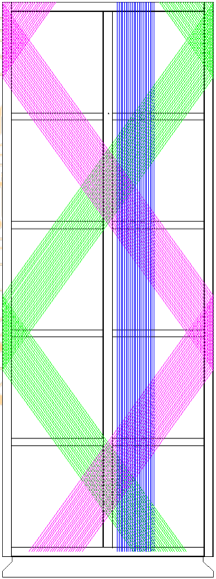
- Traditional technique attempts to seperate EM and hadronic hits, apply different scale factors
- 1m simulated ν_e interactions, flat across energies
- Train with loss $L = \frac{1}{N}\sum_i \left|\frac{f(x_i)-y_i}{y_i}\right|$
- Cautious about systematic biases
  - Haven't found anything dramatic yet

"Improved Energy Reconstruction in NOvA with Regression Convolutional Neural Networks", accepted by Phys Rev D

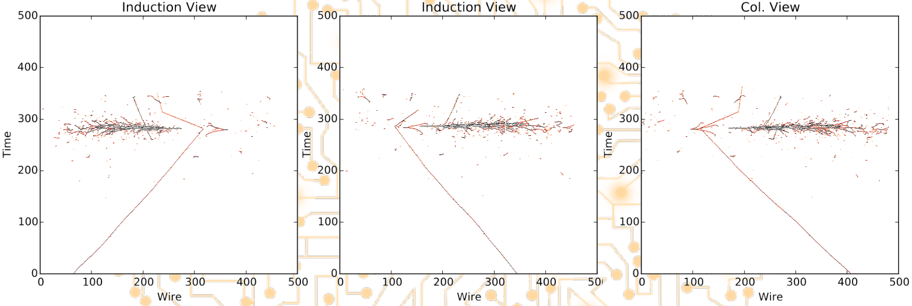
DUNE detector



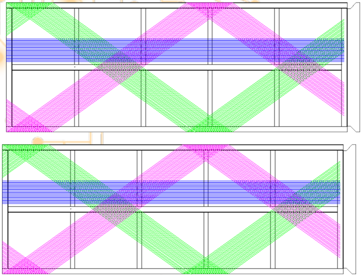




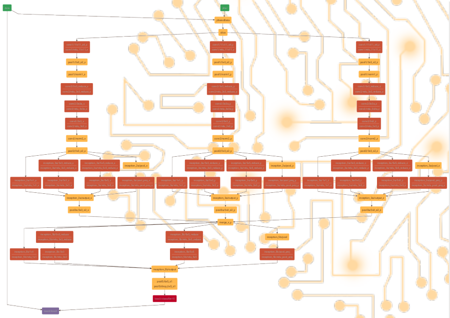
DUNE classifier

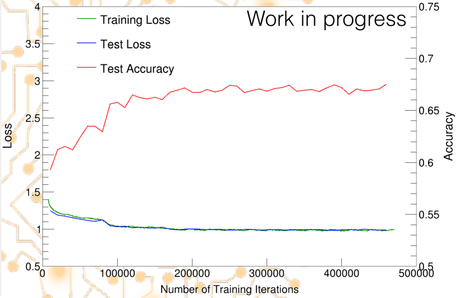


- Larger 500×500 map
- pixel = 1 wire (5mm) × 1.2ms
- "Unwrapping" wires into global space helps a lot



DUNE classifier

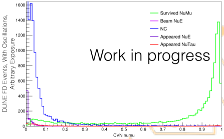




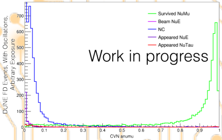
- Very similar to NOvA CVN, now triplet architecture

DUNE classifier

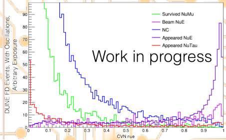
Neutrino Beam



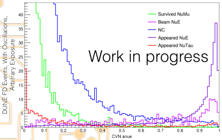
Anti-Neutrino Beam

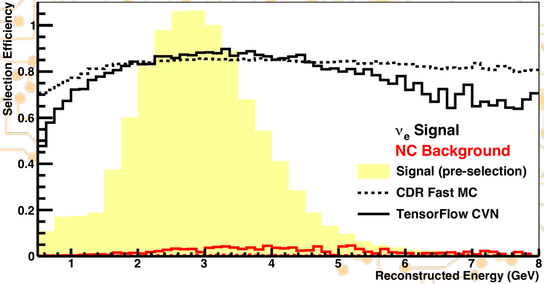


Neutrino Beam



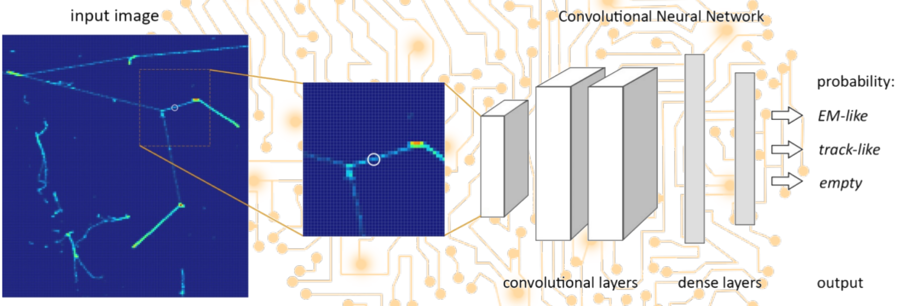
Anti-Neutrino Beam





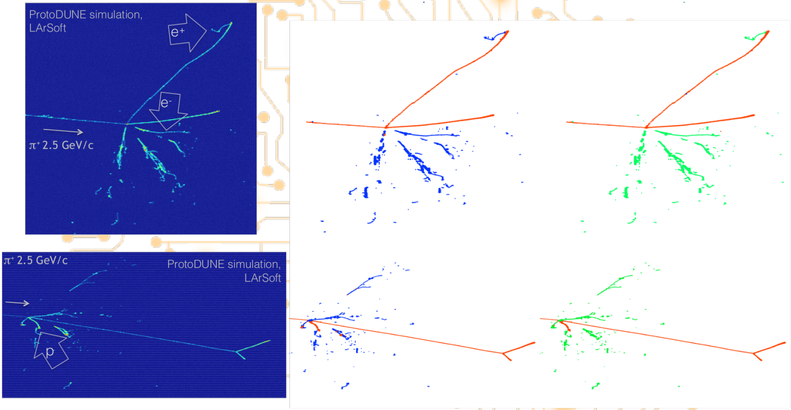
- Performance now exceeding conventional techniques and estimates from the DUNE CDR
- Will continue to investigate further improvements

DUNE track/shower CNN



- Choice of reconstruction algorithm guided by hit level classification
- Input small part of the image, classify central hit as trk vs shw
- Excellent performance

DUNE track/shower CNN



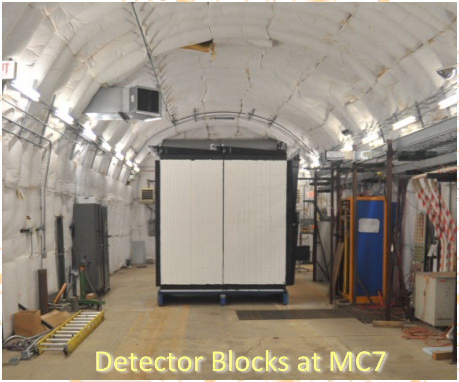
input: 2D ADC

CNN output:
EM-like (blue) / track-like (red)

MC truth:
EM-like (green) / track-like (red)

- Choice of reconstruction algorithm guided by hit level classification
- Input small part of the image, classify central hit as trk vs shw
- Excellent performance

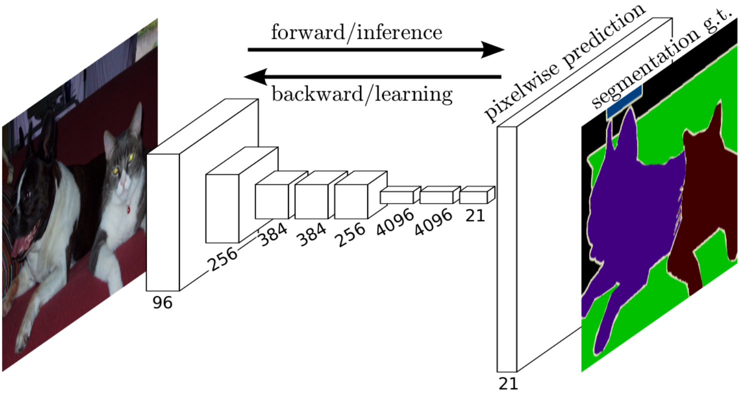
Future directions



- Improved training of Prong CVN using real testbeam data
- Can alleviate most concerns about overtraining to MC sample

- Deploy CNN energy estimator?
- Application of CNNs to vertex finding

Semantic segmentation



- Possibility to identify particles using deep learning techniques
- Replace conventional reconstruction stack completely

"Fully Convolutional Networks for Semantic Segmentation" arXiv:1411:4038

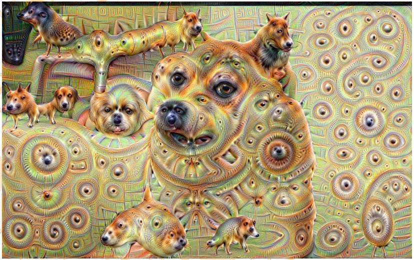
Semantic segmentation

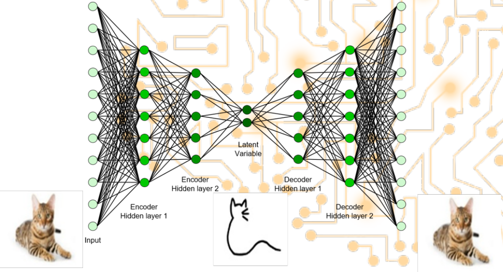
- Possibility to identify particles using deep learning techniques
- Replace conventional reconstruction stack completely

"Fully Convolutional Networks for Semantic Segmentation" arXiv:1411:4038

Generative Adversarial Nets

- If neural networks can hallucinate dogs, could they generate MC?



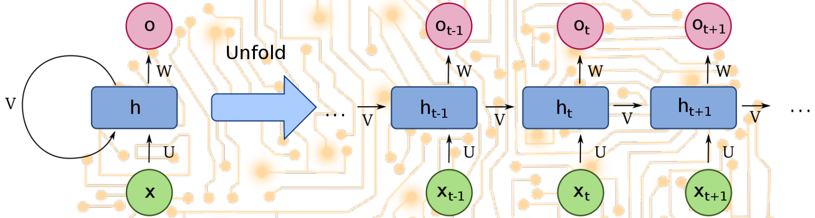


- Autoencoder aims to reproduce input image
- "Bottleneck" in the middle
- Derives latent variables

- Adversarial networks
  - One network generates events
  - A second tries to distinguish them from real data
  - Loss function is the success of the 1st in fooling the 2nd

"Learning to Pivot with Adversarial Networks" arXiv:1611.01046

Recurrent Neural Networks



- RNNs implement a form of memory
- Feed in slice of input data, plus output of previous iteration
- More sophisticated “LSTMs”

- A solution in search of a problem?
- Potentially useful for cosmic rejection
- Time-of-flight of muons tracks, delayed michels, neutrons

Conclusion

- Renaissance in machine learning
- New techniques and technologies
- Neutrino experiments on the leading edge

- Already performing excellently for core event classification tasks
- Exploring extensions in all directions

- Fermilab ML group – `machinelearning.fnal.gov`

- Extremely young and fast moving field in computer science
- Keep an eye on the literature for the next game-changer


Thank you!