The Convolutional Visual Network for Identification of NOvA Events.

An implementation of Convolutional Neural networks and its applications on neutrino interaction events.

Fernanda Psihas

Indiana University

For the NOvA Collaboration
The neutrino flavor eigenstates undergo oscillations as they propagate.

Oscillations measurements rely on flavor identification.

Given the small cross sections of neutrinos, they are statistics limited by nature i.e. 33 signal events in two years of data.

NuMI Off-axis $\nu_e$ Appearance Experiment

We produce a beam of mostly $\nu_\mu$
The NOVA Detectors

Charged particles are detected through the scintillation light produced in each cell.

PVC extrusions filled with liquid scintillator.
Neutrino Interactions

Neutrino interactions are **flavor conserving**, thus, they can be identified from the outgoing particles.

**Charged Current Interactions**

**Neutral Current Interactions**
Signature Data Events

\[ \nu \times X \]

\[ q (\text{ADC}) \]

\[ 10 \]

\[ 10^2 \]

\[ 10^3 \]

\[ 10^4 \]

\[ \nu_e \]

\[ \mu \]

\[ p \]

\[ \pi \]

\[ \gamma \]

\[ 1 \text{m} \]

\[ 1 \text{m} \]

\[ q (\text{ADC}) \]
Traditional Reconstruction

Use the topology and magnitude of the energy depositions.
Takes advantage of the granularity and time resolution of our detectors.

Isolate the event

Define clusters

Fit trajectory

We isolate individual interactions using time and space correlation of the hits.

Groups of hits can be clustered as following the path of same particle starting at the interaction point.

When necessary we can fit an assumed trajectory for each cluster of hits.
Traditional ID Methods

Mostly focused on identifying the lepton in the event. Extracted features (i.e. track length and scattering for muons, topology of energy depositions for electromagnetic showers)

* Require Previous reconstruction.
* Features are pre-defined, based on MC or test data.

Example: The Likelihood ID method

* Reconstruct electron shower.
* Find likelihoods from it’s dE/dx profiles compared to particle hypotheses.

Likelihoods → Traditional Neural Network
ID with Convolutional Neural Networks

**Premise:** Rather than select a set of features a priori, let a deep learning network extract features and draw correlations.

**In practice:** Use “images” of our events to train Convolutional Neural Networks (CNNs) to identify neutrino interactions.

Disentangle the identification from traditional reconstruction.

Allow for features apart from those based in our assumptions of the physics.***

Explore the potential of deep learning beyond event identification.

[Image of raw data, low-level features, mid-level features, high-level features]

https://developer.nvidia.com/deep-learning-courses
ID with Convolutional Neural Networks

**Premise:** Rather than select a set of features a priori, let a deep learning network extract features and draw correlations.

**In practice:** Use “images” of our events to train Convolutional Neural Networks (CNNs) to identify neutrino interactions.

- Disentangle the identification from traditional reconstruction.
- Allow for features apart from those based in our assumptions of the physics.
- Explore the potential of deep learning beyond event identification.

In absence of test data, these methods rely on the simulations*.

Known features of trained networks like over-training and saturating loss functions.
Network Layers

The simplest form of a CNN includes convolutional layers, max pooling layers, and MLP layers.

Pooling Layers:
Down-sampling is done by performing operations (average, max, etc) on the feature maps while still preserving the information.
Kernel Renormalization:
Kernels evolve as the training progresses through renormalization. This process uses non saturating functions.

Dropout:
Randomly reset weights, effectively removing whole nodes at each step. **Encourages complex dependence** and discourages overtraining.
**Neutrino Event CVN:** Siamese network architecture based on GoogLeNet.

- Inspired by siamese architectures to allow the network to learn from features on each 2D view of the event.
- We train on Fermilab's Wilson cluster GPUs (2 K40s)
- Trained on 4.7 million simulated events of all neutrino interaction types plus cosmic rays

**Inception modules:**
Network in network model with kernels of multiple dimensions

- **1x1 convolutions**
- **3x3 convolutions**
- **5x5 convolutions**
- **1x1 convolutions**
- **3x3 max pooling**


**X View**
- Convolution 7x7, stride 2
- Max Pooling 3x3, stride 2
- LRN

**Y View**
- Convolution 7x7, stride 2
- Max Pooling 3x3, stride 2
- LRN

**Inception Module**
- Convolution 1x1
- Max Pooling 3x3, stride 2
- LRN

**Aug Pooling**
- 6x5
- Softmax Output
MRE (Muon Removed - Electron):
Select a muon neutrino interaction with traditional ID methods.

Remove the muon hits and replace them with a single simulated electron of matching momentum.

Data/MC comparisons show less than 1% difference in efficiency.

<table>
<thead>
<tr>
<th>PID</th>
<th>Sample</th>
<th>Preselection</th>
<th>PID</th>
<th>Efficiency</th>
<th>Efficiency diff %</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVN</td>
<td>Data</td>
<td>262884</td>
<td>188809</td>
<td>0.718222</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MC</td>
<td>277320</td>
<td>199895</td>
<td>0.720809</td>
<td>-0.36%</td>
</tr>
</tbody>
</table>
CVN Performance on $\nu_e$

Implemented in NOvA’s main analysis for the results shown this summer at Neutrino 2016

<table>
<thead>
<tr>
<th>Total bkg</th>
<th>NC</th>
<th>Beam $\nu_e$</th>
<th>$\nu_\mu$ CC</th>
<th>$\nu_\tau$ CC</th>
<th>Cosmogenic</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.2</td>
<td>3.7</td>
<td>3.1</td>
<td>0.7</td>
<td>0.1</td>
<td>0.5</td>
</tr>
</tbody>
</table>

76% Purity, 73% Efficiency and an equivalent increased exposure of 30%

33 events selected with estimates background of ~8

NOvA Preliminary
Ongoing Work CVN and Reconstruction

Using the existing reconstruction.
Classify clusters by particle ID

Original CVN network modified to take 4 views (event + prong)

Trained on 50% purity prongs from all events no preselection

Room for improvement in classification and network optimization
Contributing to reconstruction.

There are CNN implementations in the literature for pixel by pixel classification using semantic segmentation.

In our events that means classify individual hits by the particle which caused them.

Initial studies are ongoing to compare the performance of SS to traditional clustering and the existing particle CVN identifier.
Summary

Convolutional Visual Network

CVN is our implementation of CNNs for neutrino event classification.

- It effectively increases our exposure by 30% compared to traditional ID methods.
- Studies show promise on other analyses, like the muon neutrino disappearance.
- Currently being used for multiple physics analysis.
- NOvA’s nue appearance analysis is the first implementation of a CNN in a HEP result.

CVN Paper: “A Convolutional Neural Network Neutrino Event Classifier”

A.Aurisano et. al. JINST 11 (2016) no.09, P09001

NOvA’s Latest results: Neutrino 2016 “New Results from NOvA” LINK

An implementation of CVN for cluster/particle classification is in testing stages.

There is ongoing work for hit classification using semantic segmentation.

Ongoing studies are learning about the interplay between traditional reconstruction and image classification techniques.
These are not the slides you’re looking for
CVN | Convolutional Visual Network

Is NOvA's implementation of Convolutional Neural Networks

Neutrino Event CVN:
Classifier for events in a sampling calorimeter by neutrino interaction type.

For the Electron Neutrino analysis:
76% purity, 73% efficiency and a 30% equivalent increase in exposure.

NOvA's nue appearance analysis is the first implementation of a CNN in a HEP result.

**CVN Paper:** “A Convolutional Neural Network Neutrino Event Classifier” A.Aurisano et. al. JINST 11 (2016) no.09, P09001

CVN for Reconstruction:
Ongoing studies to identify hit by hit in an event. This type of identification could influence the existing approaches at reconstruction.

Summary

MC Simulation
Neutrino Interactions

Neutrino interactions are **flavor conserving**, thus, they can be identified from the outgoing particles.

Charged Current Interactions

Neutral Current Interactions
Neutrino Oscillations

The neutrino flavor eigenstate oscillations are described by the PMNS matrix.

\[
\begin{pmatrix}
\nu_e \\
\nu_\mu \\
\nu_\tau
\end{pmatrix} =
\begin{pmatrix}
U_{e1}^* & U_{e2}^* & U_{e3}^* \\
U_{\mu1}^* & U_{\mu2}^* & U_{\mu3}^* \\
U_{\tau1}^* & U_{\tau2}^* & U_{\tau3}^*
\end{pmatrix}
\begin{pmatrix}
\nu_1 \\
\nu_2 \\
\nu_3
\end{pmatrix}
\]

Flavor Eigenstates

Mass Eigenstates

\[|\nu_i(t)\rangle = e^{-iHt}|\nu_i(0)\rangle\]

\[P(\nu_\alpha \rightarrow \nu_\beta) = \left| \sum_i U_{\beta i} U_{\alpha i}^* e^{-i m_i^2 L/E} \right|^2\]

Oscillation probability

The goal of oscillations experiments is to determine the PMNS parameters via oscillation probabilities.

The measurable in these experiments is a count or energy spectrum.
Isolating neutrino interactions

The first step in our reconstruction is dividing an event (550 μs of data)
The first step in our reconstruction is dividing an event (550 μs of data) into slices (groups of hits with some time and space coincidence)

Isolating neutrino interactions
**Vertexing:** use lines of energy deposition formed with hough transforms to find intersections

**Clustering:** find clusters in angular space around the vertex and merge views via topology and prong dE/dx

**Tracking:** Trace particle trajectories using a kalman filter, example below
Performance on Cosmic Background

**CVN Neutrino Identification**

- **Veto**
- **Slice Quality**
- **Containment**
- **Preselection**
- **p_T/p**
- **Backward photon**
- **CVN**

**Graph Details**

- **X-axis**: CVN $\nu_e$ Classifier
- **Y-axis**: Events
- **Legend**:
  - Simulated Signal
  - Simulated Beam Background
  - Cosmic Data

**Additional Details**

- **POT equivalent**: $6.05 \times 10^{30}$
- **Livetime**: 353.69 seconds

**Graph Comparison**

- **Simulated Signal** vs. **Cosmic Data**
- **Performance on Cosmic Background**
Data Driven Tests - MRBrem

![Graph](image)

- NOvA Preliminary
- Muon-removed FD cosmic data
- Muon-removed FD cosmic MC

10^2 Events/120 sec vs CVN ν_μ classifier
Performance on NearDet Data

NOvA Preliminary

Reconstructed neutrino energy (GeV)

Events / 3.72 x 10^{20} POT

CVN $\nu_e$ classifier

ND data
Total MC
Flux Uncert.
NC
Beam $\nu_e$ CC
$\nu_\mu$ CC
CVN MC Efficiency

NOvA Simulation

Efficiency (%) vs. Reconstructed energy (GeV)

- Signal
- \( \nu_e \) CC
- NC
- Beam \( \nu_e \)

NOvA Simulation

Efficiency (%) vs. True visible energy (GeV)

- Signal
- \( \nu_e \) CC
- NC
- Beam \( \nu_e \)
CVN Classifier

4.7 million, minimally preselected simulated events, pushed into LevelDB databases: 80% for training and 20% for testing.

Rescale calibrated energy depositions to go from 0 to 255 and truncate to chars for dramatically reduced file size at no loss of information.

Fine tuned with 5 million cosmic data events taken from an out of beam time minimal bias trigger.

The architecture attempts to categorize events as \{\nu_\mu, \nu_e, \nu_\tau\} \times \{QE, RES, DIS\}, NC, or Cosmogenic.
CVN Performance

- Trained on 4.7 million simulated events of all neutrino interaction types plus cosmic rays.

- Training sample has minimal preselection.
NuMI Beam

NOvA Simulation

CVN Neutrino Identification

Fernanda Psihas
Muon Neutrino Analysis

![Graph showing CC Classifier Output](image)

- Appeared $\nu_e$
- Survived $\nu_\mu$
- NC background
- Beam $\nu_e$ background

Events / $18 \times 10^{20}$ POT

$\nu_\mu$ CC Classifier Output
Neutral Current Neutrino Analysis

**CVN NC Classifier**

- **ND Data**
- **Total Prediction**
- **NC 3 Flavor Prediction**
- $\nu_e$ CC Background
- $\nu_\mu$ CC Background

**NOvA Preliminary**

- **FD Data**
- **Total Prediction**
- **NC 3 Flavor Prediction**
- $\nu_e$ CC Background
- $\nu_\mu$ CC Background
- Cosmic Background

$\Delta m^2_{32} = 2.44 \times 10^{-3} \text{eV}^2$

$\theta_{12} = 8.5^\circ, \theta_{23} = 45^\circ$

$6.05 \times 10^{20} \text{POT-equiv.}$
Non Saturating Functions

More effective back propagation due to better weight initialization and saturation functions:

\[
\frac{\delta \sigma (x)}{\delta x} = \sigma (x) (1 - \sigma (x))
\]

Sigmoid gradient goes to 0 when x is far from 1. Makes back propagation impossible!

\[
ReLU (x) = \begin{cases} 
1 & \text{when } x > 0 \\
0 & \text{otherwise}
\end{cases}
\]

Use ReLU to avoid saturation.
In SGD we avoid some of the cost of gradient descent by **evaluating small batches of events one at a time**.

The performance of conventional gradient descent is approximated as the various noisy sub estimates even out, with the stochastic behavior even allowing for jumping out of local minima.