Deep learning with raw data from Daya Bay

Samuel Kohn
UC Berkeley & Lawrence Berkeley National Laboratory
on behalf of
the Daya Bay Collaboration
and
MANTISSA-HEP Machine Learning @ NERSC

Two kinds of questions to ask

Neural networks (NNs) can be used to answer a variety of questions

What is the best combination of quantities (E, position, Δt) to distinguish between signal and background?

* Use *fully connected*, *supervised* neural networks (e.g. TMVA)

What information is contained in the the raw signals of a detector?

- distinction between event classes?

- more than one style of "good" signal events?

- What features allow us to make these distinctions?

* Use *convolutional autoencoder* neural networks (e.g. this study) [1, 2]

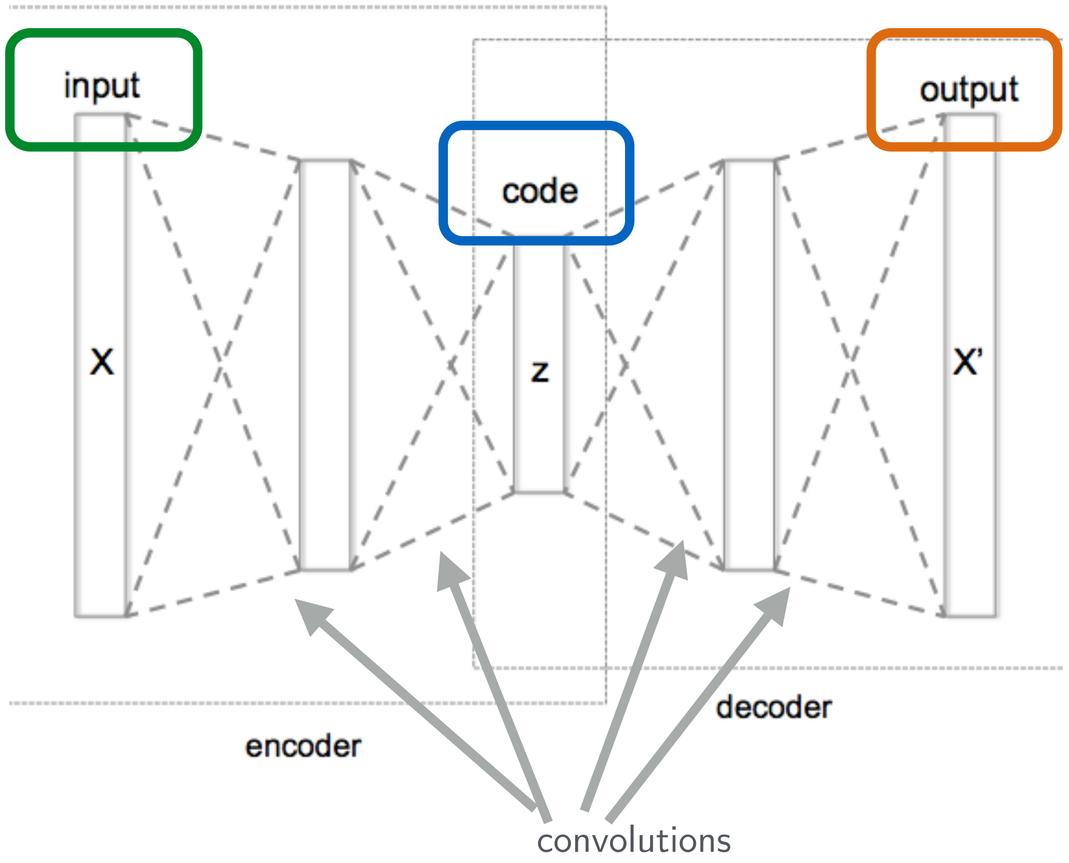
Convolutional autoencoders

Does not require labels for training!

One hidden layer is the "semantic space" containing an encoding of the input in fewer pixels (bottleneck)

Train NN to "reconstruct" (recover) original image from the encoding

Network learns features, patterns, and ways to distinguish images—*without labels*



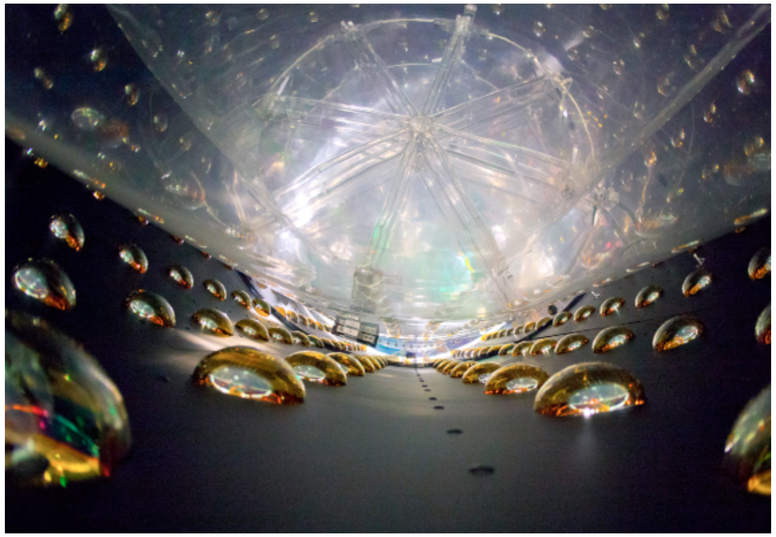
Data source

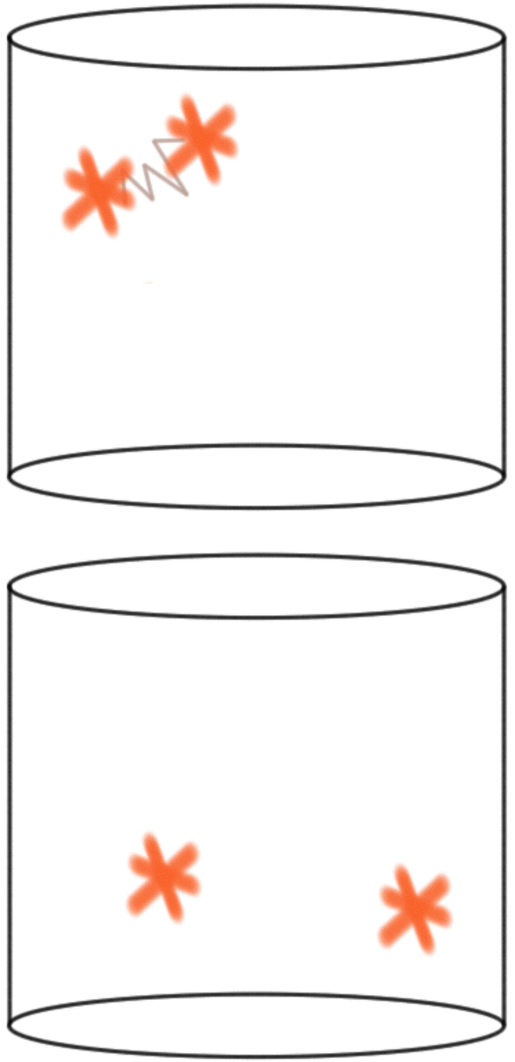
Daya Bay reactor antineutrinos [3]

Cylindrical detector

8 rings × 24 columns of PMTs

2×10^6 antineutrinos in 4 years



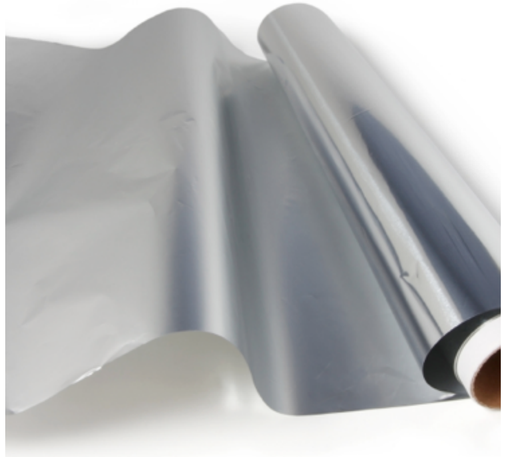


Inverse beta decay (IBD): correlated pair of signals from antineutrino event

Lithium-9: correlated pair of signals from nuclear decay (*tricky* background)

Accidental: uncorrelated pair of signals (*easy* background)

Questions to examine

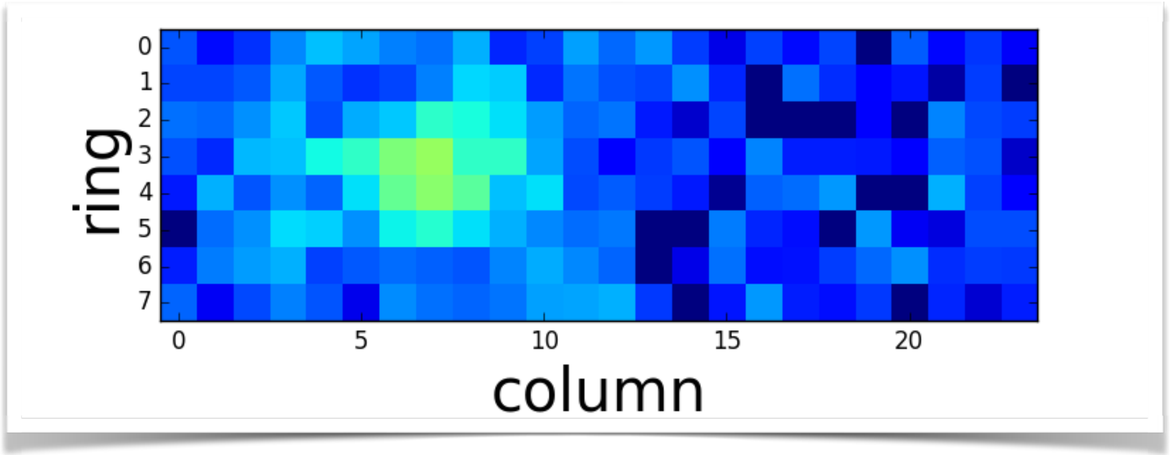


"Unroll" cylindrical detector into 8 × 24 pixel map of PMT charges and hit times for each detector trigger

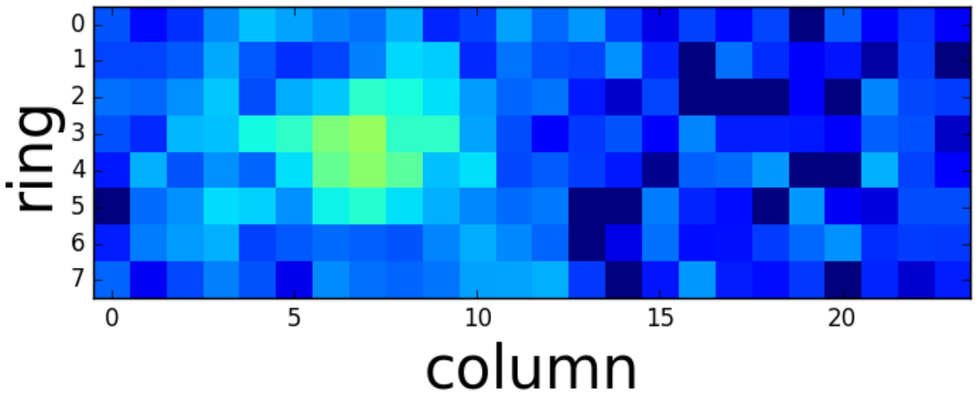
What information is contained in the PMT hitmaps?

- distinction between IBDs and accidentals? Lithium-9?
- more than one style of IBDs?

What features allow us to make these distinctions?



Study: IBD versus accidental



Accidentals are two uncorrelated signals that mimic an IBD event

- Background in Daya Bay: 1% of IBD sample is accidental [1]
- Well-understood background allows for evaluation of NN methods

Use autoencoder to analyze differences between IBDs and accidentals

Input data

- pair up prompt and delayed images to make a 2-channel image similar to RGB in a photo
- 50% IBD events, 50% accidental events

Architecture

Use a basic architecture for first study

Many opportunities for improvement

Input 2 channels representing prompt and delayed

8 × 24 pixels per channel

Bottleneck width of 16 “pixels”

Implementation details in backup

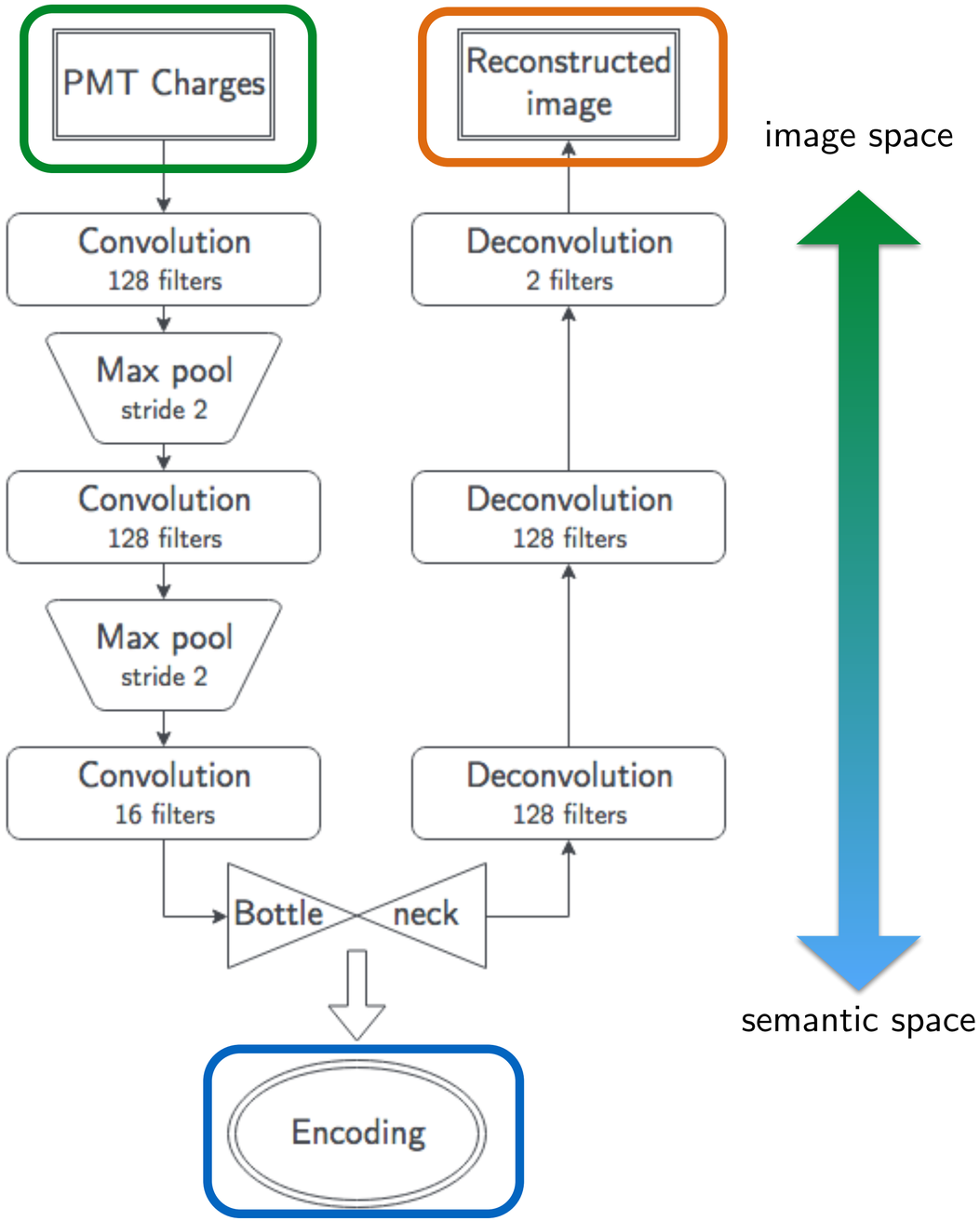


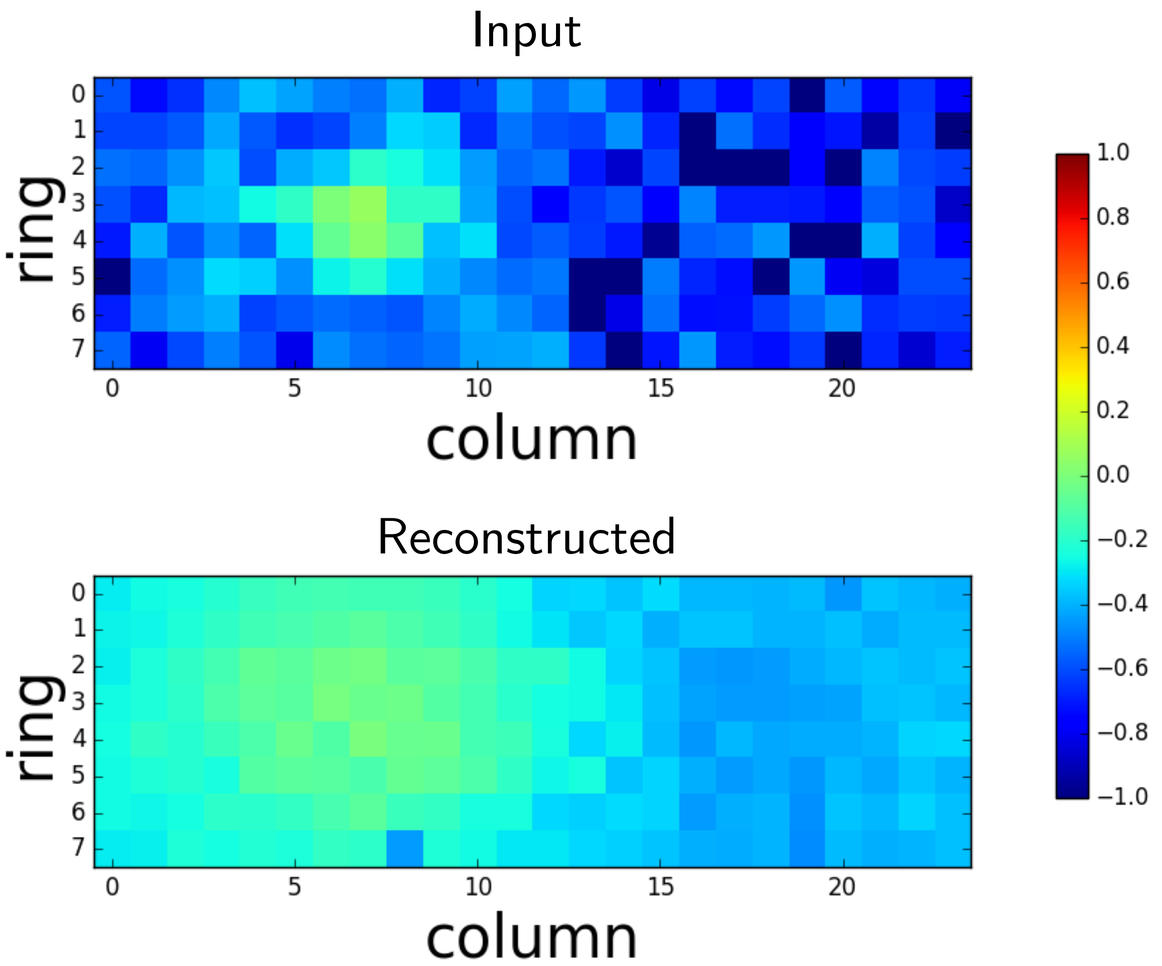
Image reconstructions

Zeroth-order evaluation of training

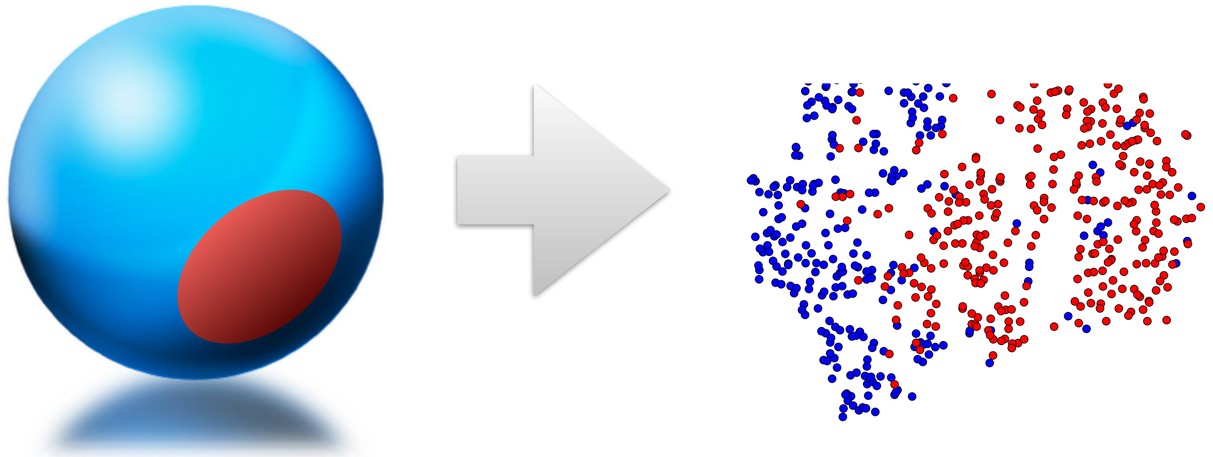
Qualitatively good reconstructions indicate the NN is learning how to encode the images

Does not accurately reconstruct fluctuations in PMT charge

Does reconstruct position and intensity of charge pattern



t-SNE evaluation



Examine encodings to look for patterns

Expect similar style events to have similar encodings

Use t-SNE algorithm to map N-dimensional encodings onto 2D plot [4]

Nearby points in N dimensions become nearby points in 2D plot

t-SNE plot

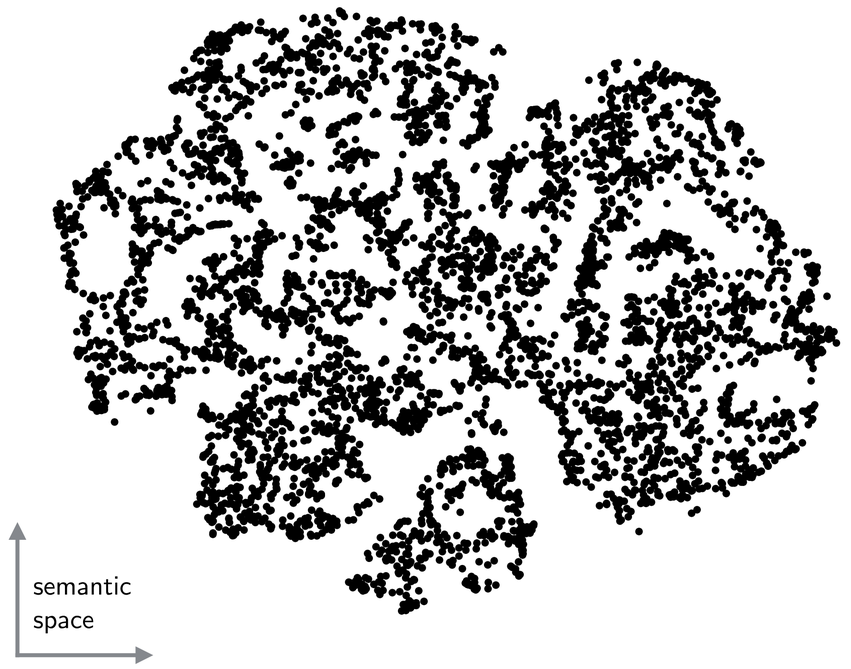
5120 data points

Each point represents the bottleneck encoding of one IBD or accidental event

Nearby points on this plot have similar encodings

Shows “semantic space”

- the axes do not represent physical quantities
- information is in the distance between data points



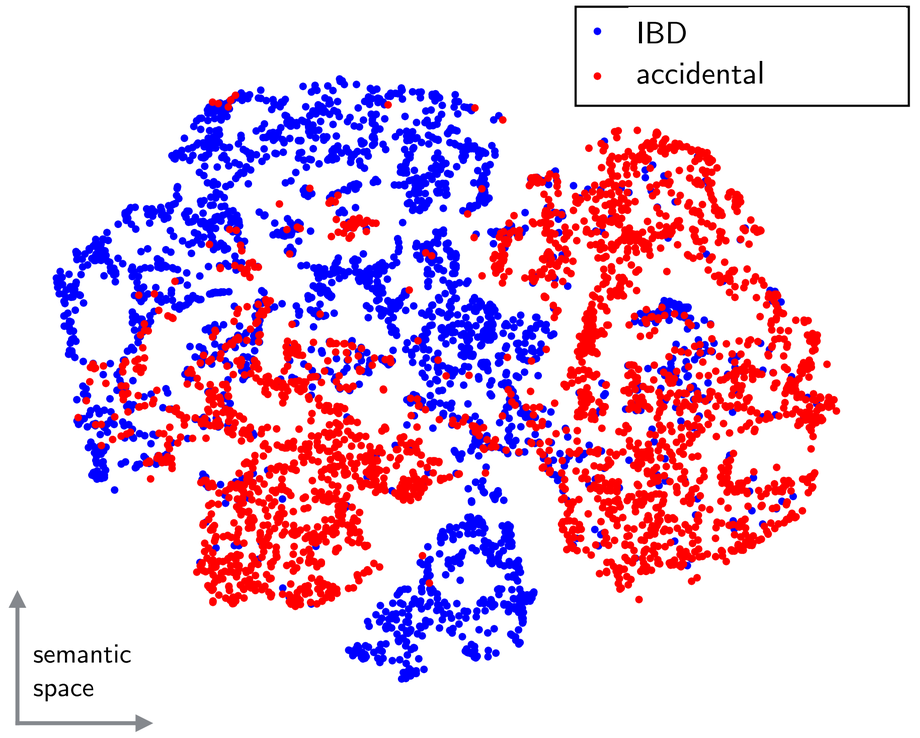
t-SNE plot: color-coded

Same 5120 data points

Color represents which data set the point belongs to (IBD or accidental)

NN *was not* given this information!

Separation of red and blue indicates *NN discovered different features* for IBD and accidentals events



Next steps

With this same NN

- Examine t-SNE clusters to understand why they clustered together
- Examine NN parameters: what is the NN looking for?

New NN:

- Incorporate more attributes (PMT times, Δt between prompt and delayed)
- Use advanced techniques such as guided backpropagation and InfoGAN [5, 6]

Compare results to supervised and (different) unsupervised convolutional NN also at LBNL [7]

References

[1] Proceedings of the IEEE, **86(11)** 2278 (1998)

[2] IEEE Conference on CVPR, 2528 (2010)

[3] Phys. Rev. Lett. **115**, 111802 (2015)

[4] Journal of Machine Learning Research **9**, 2579 (2008)

[5] J. Springenberg, et al. "Striving for Simplicity." arXiv:1412.6806

[6] X. Chen, et al. "InfoGAN." arXiv:1606.03657

[7] E. Racah, et al. "Revealing Fundamental Physics." arXiv:1601.07621

Backup

Computing details

Data sets: real Daya Bay data

- 9,000 event pairs from Daya Bay IBD set [98.5% pure]
- 9,000 "artificial" accidental event pairs (deliberately pair up signals far apart in time) [100%]

Computing resources

- Cori and Edison supercomputers @ NERSC
- Theano & Lasagne for NN architecture, convolutions, and training
- scikit-learn for data preprocessing and t-SNE analysis

Training: stochastic gradient descent with momentum

- Cost function: $\sum(x_{reco} - x_{input})^2$ (squared error)
- Learn rate: 0.01 (multiplies gradient to determine updates)

Denoising autoencoder

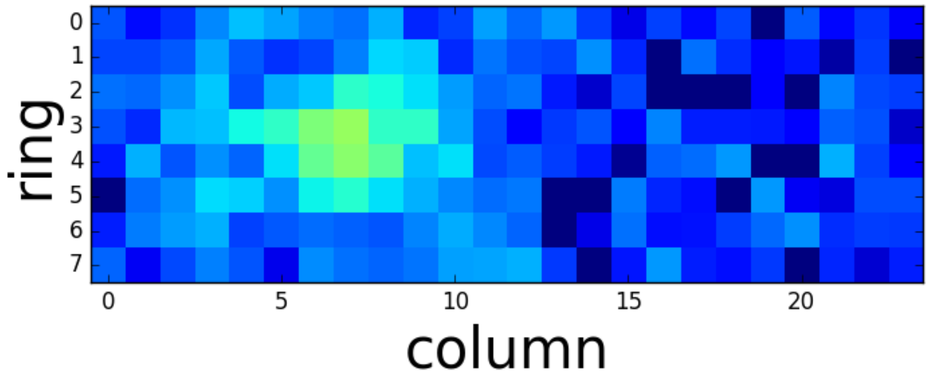
For this study I use a denoising convolutional autoencoder

Input is partially corrupted

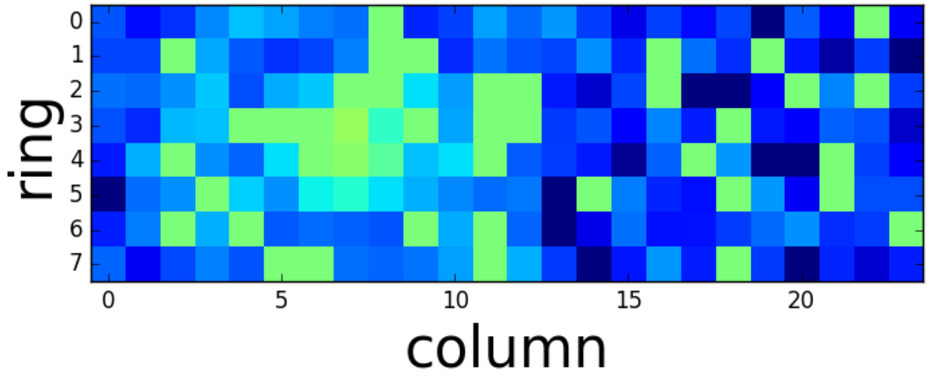
NN must interpolate missing pixels to recover the full image

Forces NN to learn more statistical dependencies between pixels

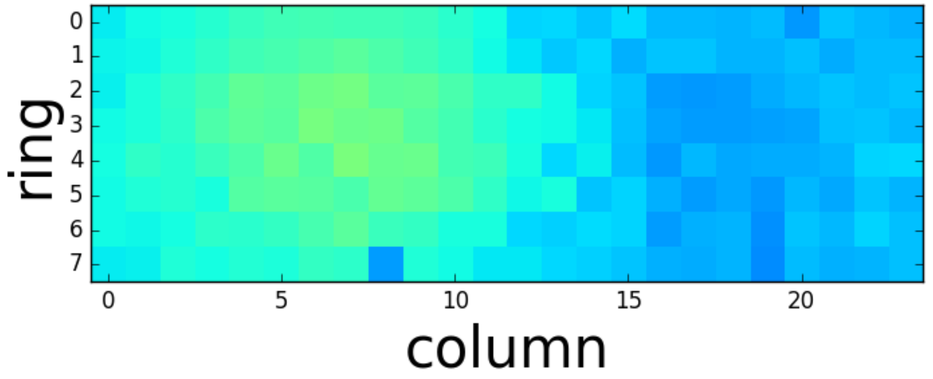
Predicting an arbitrary subset of pixels from the remainder means the NN “knows” what the image “should” look like



Input



Corrupted



Recon.