# Masters Presentation

## Latent Variable Machine Learning Algorithms: Applications in a Nuclear Physics Experiment

Robert Solli

University of Oslo, Expert Analytics AS

December 12, 2019

# Outline

1. Introducing the Active Target Time Projection Chamber (AT-TPC)
2. Challenges with traditional analysis of AT-TPC data - and the evangilization of Machine Learning
   (i) Recap of central literature
   (ii) Introducing thesis problem statements
3. An introduction to central Machine Learning concepts
4. The Auto-Encoder neural network
5. Results
6. Summary, Conclusion and Outlook

# AT-TPC

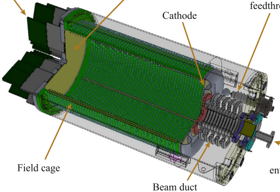



Figure: Diagram of the AT-TPC[1]

The AT-TPC is an experiment set up at the rare isotopes facility on the Michigan State University campus. The AT-TPC is commissioned to capture reactions with exotic nuclei.

[1] J. Bradt et al. "Commissioning of the Active-Target Time Projection Chamber". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 875 (2017), p. 65. DOI: 10.1016/J.NIMA.2017.09.013. URL: https://www.sciencedirect.com/science/article/pii/S0168900217309683.

# AT-TPC

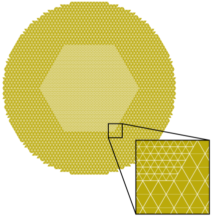


Figure: Detector pad plane of the AT-TPC[2]

Each triangle represents spatial discrete regions of the detector surface. The pad-plane consists of some $10^4$ sensor pads on a circle with $r = 29cm$

[2]Bradt et al., "Commissioning of the Active-Target Time Projection Chamber".

# AT-TPC Data

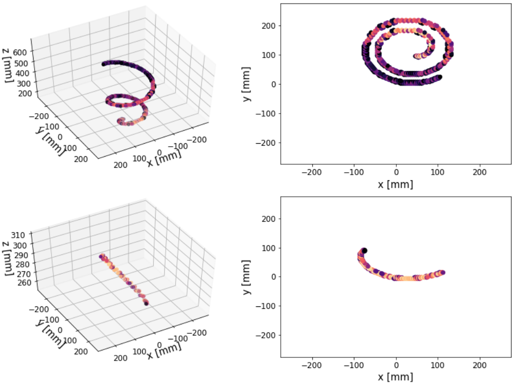
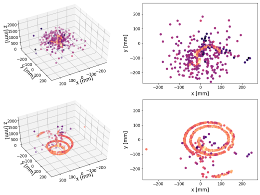
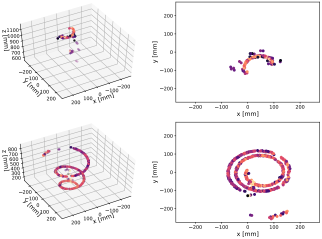



Figure: Simulated AT-TPC data in an experiment with $^{46}$Ar

# AT-TPC Data




(a) Unfiltered events from the $^{46}$Ar experiment




(b) Noise-filtered events from the $^{46}$Ar experiment

Significant noise levels in the experiment, from unknown sources. Some 60% of the recorded events are from unidentified reactions.

# Challenges with the AT-TPC

I. Expensive integration - for each event a fit is computed.
II. Assumptions of the integration technique:
   (i) Each event is fit against parameters of the event of interest,
   (ii) The integration is sensitive to Noise and Breaks in the tracks.
III. In some experiments researchers are unable to identify samples of the positive class of events.

The amount of data is also significant: the experiment generates on the order of $10^5$ events per hour running.

## Idea

Solve the problem by training deep neural networks.

# Previous Work

- Work on applying ML to this data started with a supervised learning project by Kuchera et al.[3].
- The authors explored a *supervised classification* problem of identifying reactions when ground-truth labels available.
- By fine tuning pre-trained networks the authors achieve very impressive performance.

One of the open questions is then, can we segment the events based on reactions without the ground truth labels?

---

[3] Michelle. P. Kuchera et al. "Machine learning methods for track classification in the AT-TPC". In: *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 940 (2019), p. 156. DOI: 10.1016/j.nima.2019.05.097. URL: https://linkinghub.elsevier.com/retrieve/pii/S0168900219308046.

# ML background

Machine Learning (ML) is an amorphous set of algorithms for pattern recognition and function approximation. Bred as a mix between computer science and statistical learning theory, it includes algorithms like:

I. Linear Regression
II. Logistic Regression
III. Random Forest Classifiers
IV. Genetic Algorithms

And many others...

# Why machine learning

I Builds on the work by Kuchera[4].

II Exploration of feasibility, modern computing resources has created a resurgence in ML with strong results from other fields.

[4] Kuchera et al., "Machine learning methods for track classification in the AT-TPC".

# Deep Learning

A special sub-branch of Machine Learning is the field of Deep Learning, or more broadly: Differentiable Programming
The premise of Deep Learning is to formulate an approximation to some unknown function $f(\mathbf{x})$ with a model, $\hat{f}$, that maintains some set of parameters $\{\theta\}$.
Some examples of the unknown function, $f(\mathbf{x})$, we would want to approximate are:

(a) a Hamiltonian of a system
(b) a function which determines the thermodynamic phase of a system
(c) a function which itentifies dog-species from a picture

# Deep Learning: Neural Networks

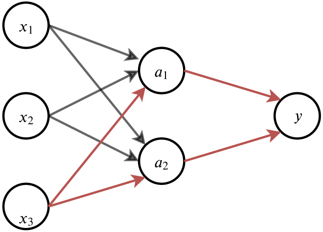
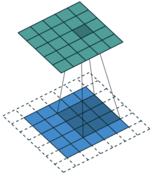



Figure: A neural network with three input nodes, two hidden nodes, and one output node

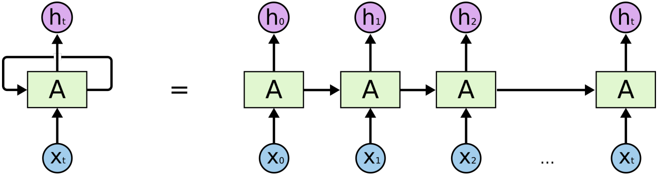
Each of the *activations* **a** are computed as a matrix product fed through a nonlinear activation *activation function* g

$$\boldsymbol{a}^{[1]} = g(\boldsymbol{x}\boldsymbol{\theta}^{[1]})_D \tag{1}$$

# Deep Learning: Network Types



(a) Convolutional architecture capturing local structures[5]




(b) Recurrent architecture for serialized applications[6]

---

[5] Vincent Dumoulin and Francesco Visin. "A guide to convolution arithmetic for deep learning". In: (Mar. 2016). arXiv: 1603.07285. URL: http://arxiv.org/abs/1603.07285.

[6] Ma Jianqiang. *All of Recurrent Neural Networks - Jianqiang Ma - Medium*. URL: https://medium.com/@jianqiangma/all-about-recurrent-neural-networks-9e5ae2936f6e (visited on 11/13/2019).

# Autoencoders

- Recall that we want to separate classes of reaction products
- Additionally - we assume that we have access to very little or no ground truth labelled data

### Idea

Learn the distribution over the events through two nonlinear maps which compress and inflate a representation of the events.

### Central Hypothesis

If we can create a compressed representation of an event from which we can reconstruct the even - the compression should be informative of the type of event that occurred.

# Autoencoders

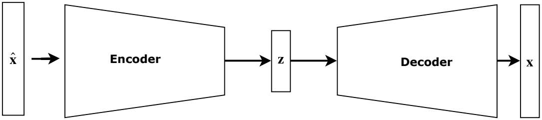



Figure: Autoencoder neural network schematic

An autoencoder is defined by an encoder/decoder pair of neural networks. We construct these such that

$$\dim(\hat{\mathbf{x}}) \gg \dim(\mathbf{z}), \tag{2}$$

and with the optimization objective

$$\mathcal{O} = \arg\min ||\hat{\mathbf{x}} - \text{Decoder}(\text{Encoder}(\hat{\mathbf{x}}))||_2^2. \tag{3}$$

# Implementing details

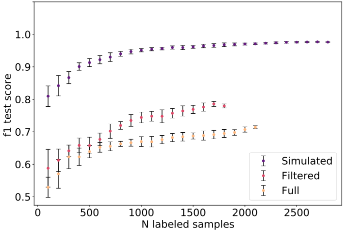
TBD

# Experiment

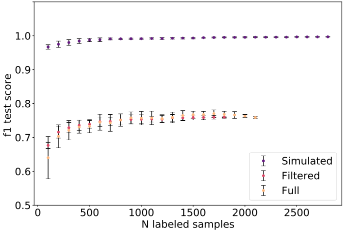
- We chose to represent the data as 2D projections, neglecting the z-axis.
- An autoencoder network is then fit to the events in an end to end manner.
- After traing, we extract the compressed (or *latent*) expressions of the data.
- With varying amounts of labelled ground-truth data we fit a very simple classifier to the latent expressions

We wish to construct latent spaces that are as close to trivially separable, and so use a logistic regression classifier.

# Results




(a) Performance for an autoencoder with a naive architecture




(b) Performance using a representation of the events as seen by a VGG16 network

$$\text{recall} = \frac{TP}{TP + FP}, \quad \text{precision} = \frac{TP}{TP + FN}. \tag{4}$$

The $f1$ score is defined as the harmonic mean of precision and recall for each class. Formally we define it as

$$f1 = 2\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \tag{5}$$

# Clustering

- We demonstrated that we can construct high quality spaces that are linearly separable
- The next step then is a *clustering* task. Can we separate classes without knowing the ground truth labels?
- Clustering, while similar to classification have some principal differences.

We explored two autoencoder-based clustering algorithms, in addition to ordinary K-means clustering.

# K-means clustering

- Ordinary K-means on the two-dimensional representations would obviously not be productive because of the dimensionality of the problem.
- Instead we try to cluster the representation of the events as seen by an image classifying algorithm: VGG16

With $8e3$ output nodes the VGG16 representation is still very much high dimenisonal, but importantly it is sparse.

# K-means results

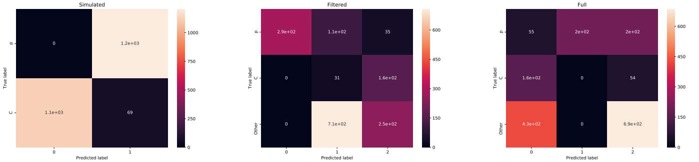
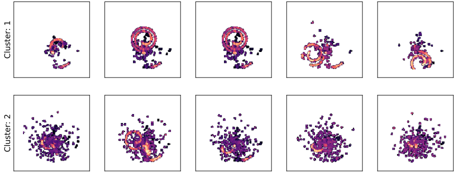



Figure: K-means clustering results using the VGG16 representation of the datasets

Cluster elements for clustering full data



# Mixture of Autoencoders

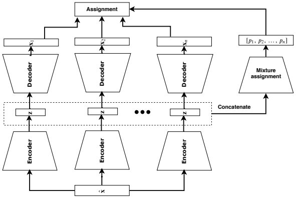
- One can also attempt clustering with Autoencoder based algorithms
- Niche field with only limited applications (MNIST, Reuters)

$$\mathcal{L}_x = \sum_j p_j^{(i)} \mathcal{C}(\boldsymbol{x}_j^{(i)}, \hat{\boldsymbol{x}}^{(i)})$$

$$S(\boldsymbol{p}^{(i)})_{\text{sample}} = -\sum_j p_j^{(i)} \log p_j^{(i)}$$

$$S(\{\boldsymbol{p}^{(i)}\}_i)_{\text{batch}} = \left(-\sum_j \bar{p}_j \log \bar{p}_j\right)^{-1},$$

$$\bar{\boldsymbol{p}} = \frac{1}{\beta} \sum_i^{\beta} \boldsymbol{p}^{(i)}$$



# Deep Clustering Results

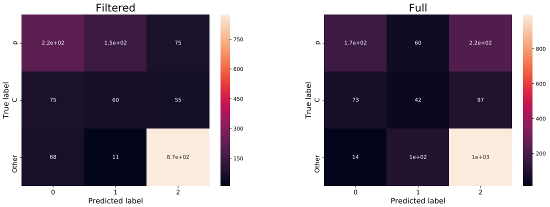
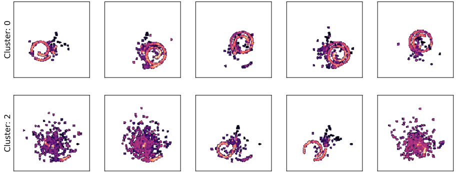



Figure: Top-performing mixae results for the real experimental data

Cluster elements for clustering of full data



# Summary and Outlook

- We have shown that Autoencoder models are suitable for both semi-supervised and unsupervised applications.
- Autoencoder clustering has a fundamental challenge with issues of stability and convergence. We still need labelled samples to verify.
- Including physical parameters increases the quality of the latent space.

Neural network models clearly have a place in this analyisis going forward: but connecting the anlysis to the physical properties of the system remains a challenge to solve.