Pulses Characterization from Raw Data for CDMS

Physical Sciences

Francesco Insulla (06210287), Chris Stanford (05884854)

December 14, 2018

Abstract

In this project we seek to take a time series of current amplitudes collected by by phonon-sensitive detectors called QETs (Quasiparticle-trapping-assisted Electrothermal-feedback Transition-edge-sensors) and identify the start time of the pulse. Currently, our techniques for registering a pulse are not accurate, and often classify detector noise as a pulse. This issue is significant, because the separation of signal and noise is critical to the success of the experiment, and is easily generalized to any other detector of this type. Furthermore, the problem of processing data to find pulses, and characterizing them, occur in many other fields were signals need to processed, therefore it has the potential for a wider range of uses. We implement various Machine Learning models and found the most success with PCA+FCNN.

1 Introduction

The Cryogenic Dark Matter Search (CDMS) research group seeks to directly detect the most frequent form of matter in the universe: Dark Matter. To do so, we study the behaviour inside semiconducting crystals at cryogenic temperatures. When a dark matter particle or another form of radiation interacts with the crystal, a cloud of electrons and holes is produced at the interaction site. These charges are then drifted through the crystal by an applied electric field, and produce phonons that are collected by phonon-sensitive detectors called QETs (Quasiparticle-trapping-assisted Electrothermal-feedback Transition-edge-sensors). Once the a signal pulse is received, we seek to determine the start time of the interaction from the raw data. We use logistic regression, a shallow fully connected neural networks (FCNN), linear and kernelized principle component analysis (PCA) with FCNN, and convolutional recurrent neural networks (CNN+RNN).

2 Related Work

Dr. Andrew Watson's dissertation [1] uses a two-step Principal Component Analysis / Multi-Dimensional Fit (PCAMDF) method to reconstruct the location of interactions within a dark matter detector. His research has similar goals to ours and inspired use to focus on PCA as a technique since because it worked for him. However our methods and dataset are different. Dr. Watson uses data from a detector that has 19 channels while we are limited to only 2 channels and we seek to find the start time of the pulse. Furthermore, while we use PCA for one of our methods, we feed the resulting projections into a neurla network instead of a simpler multi-dimensional fit. Our colleague, To Chin Yu, has also done work on position-recon with a similar dataset to Dr. Watson, using CNNs and Residual Networks (ResNets).

3 Dataset Creation

While we dont have enough real data [1] to train on, we have do have a Monte Carlo simulation of our experiment built with G4CMP[2]. Combining the results from simulating with real noise, we created our dataset as represented by Figure 1.

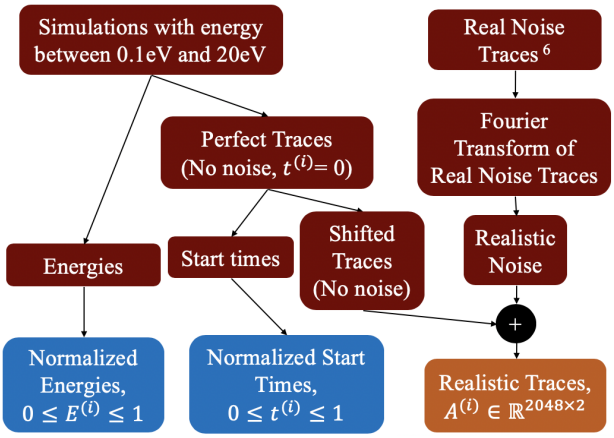

Figure 1: Pre-processing steps

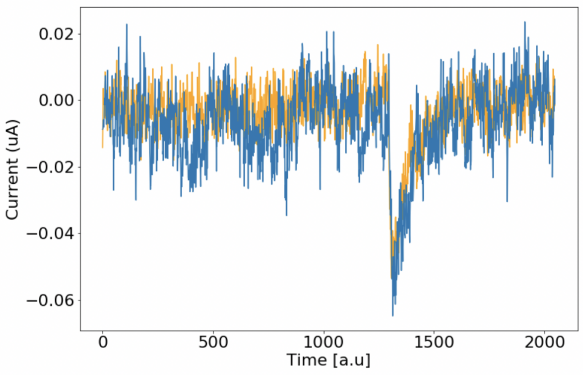

Figure 2: Sample trace

We first run simulations to generate events at energies between 0.1eV and 20eV with a step size of 0.1. The traces[3] that come out of the simulation are perfectly smooth and all the pulses start at the same time, so we add noise and shift the pulse to make them appear more similar to the real data. We do this by shifting the pulse on the time axis by a offset $d \sim \text{Uni}(0, 2048)$. To generate noise, we take the FFT (fast fourier transform) of real noise, add a random phase shift and then get the IFFT (inverse fast fourier transform). We then add this noise to the simulated traces resulting in time series similar to the one in Figure 2. We also process the files that have the information of the energy and position of the events, filtering the relevant information, and scaling the time to be in $[0, 1)$. Next, we correct the distribution of energies by eliminating examples where the frequency of the energy is abnormal. At the end of this process we are left with 39,458 total examples that we split intor Train, Dev, and Test using a 85-10-5 % split.

4 Features

In input features were traces with two channels: $A^{(i)} \in \mathbb{R}^{2048\times2}$. For the logistic regression and FCNN we flattened the trace to a 4096 dimensional vector. For the CNN+RNN we kept the shape of the trace. For PCA + FCNN model we flattened the traces and used PCA to find the first 1024 principle of the components (PCs) using 20% of the training set which explain 89.83% of the variance. We decided to try PCA because we plotted the correlation matrix of the traces (Figure 3) an noticed all the points of the trace are positively correlated. For the Kernel PCA, we used a

[1] By real data we mean data that is produced by the physical detector instead of by the Monter Carlo simulation
[2] https://github.com/kelseymh/G4CMP
[3] Each simulated event has an associated trace for each channel. A trace is a time series defined by an array of 2048 values where each value represents a current measured by the detector.

radial basis kernel, and 1024 PCs. We tried this too because we thought the relationship between the projected features could be non-linear.

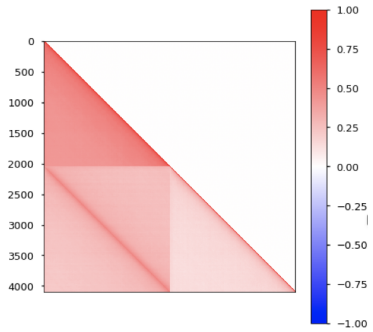

Figure 3: Correlation Matrix

5 Methods

We represent a trance example input as $A^{(i)} \in \mathbb{R}^{2048\times 2}$, the flattened version as $a^{(i)} = \text{Flatten}(A^{(i)}) \in \mathbb{R}^{4096}$ the true value of the time as $t^{(i)}$, and the prediction as $\hat{t}^{(i)}$. In all the models apart from logistic regression we minimize the mean squared error (MSE):

$$\frac{1}{m}\sum_{i=1}^{m}(t^{(i)} - \hat{t}^{(i)})^2,$$

although for applied purposes we are more interested in reporting the mean absolute error (MAE):

$$\frac{1}{m}\sum_{i=1}^{m}|t^{(i)} - \hat{t}^{(i)}|.$$

5.1 Baselines: Logistic Regression and Shallow FCNN

We begin the training/modelling phase by fitting a logistic regression as our most basic baseline. We use $a^{(i)}$s as inputs and $t^{(i)}$ discretized into one-hot encoded vector with 1048 classes as outputs. As a secondary baseline, and to get a sense of the power of neural networks on this task, we make a FCNN with one layer with 512 nodes.

5.2 PCA+FCNN

We perform PCA by finding the eigen-basis of the correlation matrix C:

$$C = \frac{1}{m}\sum_{i=1}^{m} a^{(i)}a^{(i)^T}$$

and then finding the projections by

$$\tilde{a}_k^{(i)} = V^{k^T}a^{(i)}$$

where V^{k^T} is the k-th principle. We then feed these projections into a FCNN explained in the next subsection.

5.3 Kernelized PCA+FCNN

We perform the kernel trick on linear pca as follows:

$$\begin{aligned}\tilde{a}_k^{(i)} &= V^{k^T}\Phi(a^{(i)}) \\ &= \sum_{j=1}^{m}\left(c_j^k\Phi(a^{(j)})\right)^T\Phi(a^{(i)}) \\ &= \sum_{j=1}^{m} c_j^k K(a^{(j)}, a^{(i)})\end{aligned}$$

This was the structure used

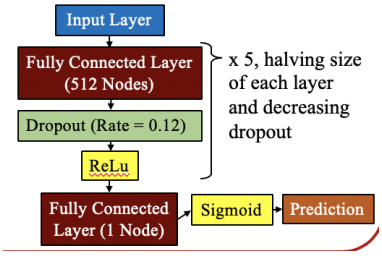

Figure 4: FCNN Structure

5.4 LSTM

Base on the suggestion of TA Ashwini Ramamoorthy, we implemented a Long Short-Term Memory (LSTM) neural network, as this type of NN is especially suited to dealing with time series data. We built the model with two layers of max pooling with a stride of 4 followed by two dense hidden layers of 256 and 16 nodes, respectively.

Training the LSTM proved to be much slower than previous methods. On the other hand, the LSTM predictions would converge withing a relatively smaller number of epochs (less than 20), so little training was required. Unfortunately, despite trying different structures of the model, we were unable to attain an mean average error with this method that came close to that of PCA+FCNN.

6 Results and Discussion

The results table represents the average MAE scaled to 2048 (the total number of bins). From this project were able to develop a methodology to construct an effective tool that we might use as part of physics experiment to determine the start-time of pulses measured by our detector. After constructing our dataset from our Monte Carlo simulations, we trained logistic regression, FCNN, CNN+LSTM, a standard PCA fed into FCNN, and KPCA with a radial basis kernel fed into a FCNN to predict $t^{(i)}$ and had the best Test set MAE with the standard PCA + FCNN method. Our goal was to get to a MAE around 1 or 2, however the lowest we ever got on training was 4. This is likely due to the fact that the pulses are so noisy which is why we chose this challenging problem in the first place. An important insight from this project was that more complex models dont always produce better results, as can be seen comparing the LSTM+CNN and KPCA+FCNN with the PCA+FCNN. Another lesson we learned was that producing the dataset and preparing it

Table of Scaled MAE	Logistic Regression (Discretized to 1024)	Shallow FCCN[7]	Linear PCA + FCNN	Radial Basis PCA + FCNN	CNN+LSTM
Training	66.78	47.38	15.24	27.59	116.77
Validation	72.78	188.52	17.91	73.12	120.34
Test	72.47	203.70	21.73	104.67	122.23

Figure 5: Final Results

for training can be the most time intensive step. Finally, while we didnt accomplish exactly what we set out to do we are content with out results and will continue improving on them.
If we had more time we would try other models, tune hyper-parameters more methodically, keeping track of all results, and use a larger dataset, with more examples per energy and also a wider range of energies.

7 Contributions

Task	**Who**
Write script to run simulations	Francesco I.
Process real noise to make new noise	Chris S.
Overlay noise and simulated traces	Chris S.
Filter relevant data/features from simulation outputs	Francesco I.
Preprocess data	Francesco I.
Train baseline model	Francesco I.
Write-up milestone	Francesco I.
Coding of various models	Francesco I.
Testing of various models	Chris S., Francesco I.
Poster and presentation	Francesco I.

8 Acknowledgements

We would like to thank To Chin Yu for suggesting this project and providing key insights about Machine Learning, the Northwestern SuperCDMS team for providing the MC simulation and the sample noise traces from which we generated our dataset and finally the CS299 TAs and Professors for giving us the opportunity to do this project and teaching us all the material.

References

[1] Watson, A. W. (2017). *Transverse position reconstruction in a liquid argon time projection chamber using principal component analysis and multi-dimensional fitting* (Order No. 10270707). Available from ProQuest Dissertations & Theses Global. (1906685475). Retrieved from https://search.proquest.com/docview/1906685475?accountid=14026