**Research Plan: Improving Particle Discrimination in WIMP Dark Matter Detection Experiments Using Neural Networks**

**Brendon Matusch**

**Rationale**

Detection of dark matter is one of the foremost endeavors in modern particle physics. Due to unexpectedly large gravitational forces affecting the rotation of galaxies and the deflection of photons, it is hypothesized that there are a large number of “dark” particles that rarely interact via forces other than gravity. One of the most-researched candidates for such a particle is the Weakly Interacting Massive Particle, or WIMP.

The identification of background radiation events is a major hurdle to overcome in all experiments for WIMP detection. The current practice of manually developing a discriminator function to eliminate background events is challenging and time-consuming when available calibration data is frequently impure and present in limited quantities. Machine learning, particularly in the form of neural networks, has the potential to be a powerful solution, because it permits automation of the process. Machine learning systems may also locate patterns in the data that can be used for discrimination, but are not readily apparent to researchers.

However, there are several problems that must be overcome. For instance, impure data can be detrimental to the training of many types of models, and data often comes in formats that are non-trivial to process, such as audio and 3D geometries. It is not always clear which of these formats can be used to produce the most effective discriminator, nor is it obvious what kind of neural network will learn the most efficiently and accurately on a given data format.

If solutions to these issues can be found, it stands to reason that the pace of many dark matter research projects could be considerably improved. There could potentially be a great reduction in the engineering time that is spent on manually deriving a discriminator function every time some aspect of the experiment changes.

**Background on Experiments**

**PICO-60**

The PICO-60 experiment is a bubble chamber containing superheated C3F8. It is designed to detect the presence of WIMPs in the mass range of 3.3keV and above. It operates on the expectation that WIMPs will interact with atomic nuclei through the weak nuclear force, transmitting kinetic energy to them and producing energetic recoils.

It is engineered to be almost entirely insensitive to gamma rays and many other forms of low-energy radiation. Thus, there are two remaining sources of background radiation:

* Externally introduced neutrons, which are hypothesized to produce nuclear recoils identical to those created by WIMPs. However, unlike WIMPs, most neutrons scatter multiple times, producing multiple bubbles. A very small number of neutrons producing single-scatter events remain, and can be subtracted from the number of WIMP-like events detected.
* Alpha particles, which are emitted by nuclear decays inside the detector. These can be detected using a discriminator.

Because alpha events occur inside the detector, they cannot be easily removed by ignoring events near the walls of the detector. For development of discriminators, there is an available set of 99%-pure alpha background events collected in 2016 and 2017.

Since WIMPs have never been detected, there is of course no WIMP data available to calibrate discriminators. However, as WIMPs are hypothesized to produce nuclear recoils indistinguishable from those created by neutrons, a 90%-pure calibration set created with americium and californium neutron sources is used for development.

Sensory data for each recorded event comes from two piezoelectric microphones (piezos) mounted to the outside of the silica vessel, and also from three externally mounted cameras viewing the transparent detector at different angles.

A conventional discriminator known as the Acoustic Parameter (AP) was developed by the PICO-60 collaboration and verified based on physical modeling and empirical modification to discriminate accurately between alpha particles and nuclear recoils. It makes use of a banded Fourier transform of audio collected by the piezos.

In addition to AP, a multi-layer perceptron was trained on the data during past research and found to discriminate between alpha particles and nuclear recoils with 85% accuracy.

**DEAP-3600**

DEAP-3600 uses 3.3 tonnes of liquid argon as target material. Liquid argon is a scintillator; when struck by an ionizing particle it emits photons in a quantity proportional to the energy deposited. The photons are detected by 255 extremely sensitive light detectors (photomultipliers or PMTs) placed around the acrylic vessel containing the liquid argon. Based on the counts and timings of photons that reach each of the PMTs, it is possible to determine the energy and location of any event that occurs in the body of the detector. However, alpha events that occur in the neck are very difficult to isolate, because they overlap with the apparent energy range of expected WIMP candidates.

It is impractical to create significant amounts of clean calibration data in the DEAP-3600 detector. Thus, data from a Monte Carlo simulation (which was benchmarked using real-world calibration data) is used instead.

Once again, a conventional discriminator was developed by the DEAP-3600 collaboration. Because there is no clear signature of a neck alpha event, it is not possible to discriminate nearly as accurately as in the PICO-60 experiment. The best performance achieved in this way is removal of 99.6% of neck events, at the cost of 91% of hypothetical (simulated) WIMP events.

**Engineering Goal**

Despite the existence of working conventional discriminators in the PICO-60 and DEAP-3600 experiments, neural networks for separation of signal and background events are most certainly applicable to both of them.

In the case of PICO-60, my goal is to use available data to develop and validate a general machine learning system that has the potential to speed up the development of discriminators in future iterations of the PICO project, as well as other, similar dark matter experiments. By applying semi-supervised learning algorithms, I hope to find a system that will be applicable when even less accurately classified training data is available compared to PICO-60.

The discriminator used for DEAP-3600 is not known to be optimal, so it is possible the application of machine learning will lead to improvements in performance. Alternately, it is possible that the machine learning system will perform at the same level as the conventional discriminator and provide evidence that it is close to optimal. My goal is thus to determine which of these two cases is true.

**Procedures**

**PICO-60**

For the PICO-60 experiment, there are a number of different input formats that can potentially be used as input data for a neural network. The format used in past research on the experiment is a Fourier transform integrated into eight frequency bands. This was used as the input to a conventional discriminator known as the Acoustic Parameter (AP), as well as to a multi-layer perceptron that was applied experimentally as part the original PICO-60 study.

There are three other data formats that will be investigated in the context of PICO-60:

1. A full-resolution analogue to the banded Fourier transform, using all values in the high-resolution curve directly without any integration
2. A full-resolution audio waveform, which includes time domain information (such as how the frequency distribution changes over time) which may be useful
3. Images captured by cameras in the detector, which have been used only for reconstructing the positions of events but are not proven to be inapplicable for background event identification

Using whichever of these data formats is observed to produce the most effective discriminator, the intent is to experiment with a wide variety of neural network hyperparameters, including the following (depending on the type of neural network used):

* Dropout and L2 regularization
* Number of layers (convolutional and dense)
* Number and size of convolutional filters

In general, these hyperparameters will be optimized using grid searches, in which a discrete search space is defined for each hyperparameter, and every possible combination of hyperparameter values within their respective search spaces is tested. Each configuration will be tested multiple times so that variations in the randomly selected validation set will have minimal effect.

In addition to a variety of hyperparameters applied to dense and convolutional neural networks (CNNs) trained with conventional supervised learning, novel semi-supervised learning algorithms will be applied to improve performance. The essential concept of semi-supervised learning is that a limited amount of labeled (classified) training events is used alongside unlabeled events, which help the network more accurately learn a decision boundary between the two classes.

More specifically, the problem in the PICO-60 experiment is that the training data is impure, so it is not desirable to fit perfectly (which would amount to finding some element of the audio, such as noise, that differs based on when the data was collected). Rather, one would hope that a machine learning system could ignore the segments of the data that are impure and rather learn to separate particle types.

I hypothesize that semi-supervised learning will be able to improve the effectiveness of training by taking advantage of the neural network’s *most confident* predictions (those where the output is extremely close to either 0 or 1) on unlabeled data. I expect that these predictions should correlate with the actual particle types, and that incorrect predictions caused strictly by overfitting will be less confident. I have invented two semi-supervised learning algorithms, that I intend to test, that should take advantage of this predicted property:

* Iterative cluster nucleation: This is an algorithm that, given a set of labeled data *X* and a set of unlabeled data *U*, iteratively determines labels for examples in *U* and adds them to *X*. It applies the following procedure:
  + Train a neural network for a small number of epochs on *X* (perhaps 30).
  + Run inference on the entirety of *U*; call the vector of predictions *P*.
  + Find all values in *P* that are very close to either 0 or 1 (suppose less than 0.05 or greater than 0.95), representing high confidence.
  + For each of the corresponding examples in *U*, assign it a label according to which of the two predictions it is given, and add it to *X*.
  + Repeat, training on the newly updated labeled set *X*.
* Gravitational differentiation: This is a related algorithm that can be thought of as a more analog alternative to iterative cluster nucleation. Rather than making a binary decision on whether or not to add a given example to the labeled training set, it makes use of a distortion function that calculates final-layer derivatives for each example in *U* such that an example with a prediction of 0.5 (neutral) has no effect on training, and unlabeled examples with predictions closer to 0 or 1 have progressively more effect on the training process.  
  This can be formalized in the following piecewise exponential function *GravDiff*, where *p* is the neural network’s prediction on the training example in question, *ψ* is a parameter that defines the degree of distortion, and *g* is a parameter that defines the overall influence of unlabeled examples on the training process:

**DEAP-3600**

For the DEAP-3600 experiment, large amounts of simulated training data are available. This means data efficiency systems such as semi-supervised learning are less applicable than PICO-60. However, the format of the data presents a very different set of challenges. There are two main open questions, with respect to the optimal machine learning system to apply:

1. Each of the photomultiplier tubes receives multiple photons at different times. What is the best way to encode this information for a neural network? I will attempt to use both the total number of photons received by each PMT, and the time of arrival of the first photon at each PMT.
2. The 255 photomultiplier tubes are arranged in a mostly-hexagonal lattice on the surface of the spherical vessel, minus a large hole in the top. What is the best neural network architecture to process this spatial arrangement of data points? I have considered the following options:
   * A dense neural network is an obvious and simple choice, but it will likely require more parameters than something optimized for this spatial arrangement (for the same reasons that dense neural networks are suboptimal for image processing compared to convolutional neural networks). It should thus have a greater tendency to overfit and train more slowly.
   * A common CNN cannot be applied without deforming the spherical topology, because they are designed to operate on flat rectangular images, and (as we know from world maps) a sphere cannot be projected onto a rectangle without distortion. However, the distortions introduced by world maps do not prevent us from reading the map, and similarly, applying a cylindrical map projection to the sphere may allow a conventional CNN to work effectively.
   * I have invented a novel neural network architecture called a topological CNN, which may be an effective solution to this problem. It is much like a conventional CNN, except that rather than a kernel being convolved over a square grid, a kernel is convolved over an arbitrary geometric lattice of polygonal nodes (hexagonal, in this case).  
     Each node is assigned six neighbors in a clockwise order, and a kernel consists of all node within a certain distance of its center node. (Distance is defined by the minimum number of connections that must be taken to get from one node to the next.) In a topological convolutional layer, such a kernel is formed at every possible center node, and the output of the layer is an identical topology (excepting nodes at the edges where kernels can not be formed without hitting the edge of the topology).  
     To ensure that the orientation of every kernel is the same (that is, that they are not randomly rotated in different positions on the topology), the clockwise order of the connections of each node must be rotated such that the top node (the node with the highest vertical position) comes first. This way, orientations of kernels will be symmetrical along the vertical axis (on opposite sides of a sphere, for instance).

Both of these questions will be approached in a similar manner as with PICO-60. I will start by implementing and validating each of these concepts, proceed by manually optimizing and testing various configurations for each of the networks and input methods discussed, and finally seek a conclusive result by running a grid search for the optimal hyperparameters.

**Implementation Details**

All programming for this study was done in Python 3. Keras, running on a TensorFlow backend, was used for all machine learning tasks. NumPy and SciPy were used for linear algebra and signal processing. ROOT, scikit-image, and scikit-learn were used for data loading and storage. Matplotlib was used for data visualization.

**Data Analysis**

In general, the performance of machine learning discriminators will be measured using four metrics:

1. Precision: the proportion of events classified as background that are actually background events. High precision means that the events the discriminator classifies as background events are very pure (that is, very few signal events are incorrectly removed).
2. Recall: the proportion of background events that are classified as background. High recall means that almost all background events are removed, and that there are very few of them remaining in the set of predicted signal events.
3. Accuracy: the proportion of the network’s classifications that are correct. High accuracy means that there are very few misclassifications in general.
4. Class-wise standard deviation (CWSD): the mean of the standard deviations of network predictions on the sets of background and signal events individually. This represents the confidence of the network. Even if accuracy is high, the predictions of the network (decimal numbers from 0 to 1) may be spread out. Low CWSD means that the network’s predictions are tightly clustered near 0 and 1, which is desirable because it means there are few events for which the network’s predictions are questionable.

To calculate these metrics, I need access to highly accurate classifications for each of the events. In the DEAP-3600 experiment, because the training data is simulated, I have access to perfectly accurate ground truths for training. In the PICO-60 experiment, the outputs from the conventional AP discriminator are used for comparison.

Note that at no point during the training process are the AP classifications provided to the neural network. The intent of this project is to develop a general system that will work for future experiments where a conventional discriminator is not available. Thus, AP is used to validate the predictions of the network, but only the impure classifications given by the calibration runs are used for training data.

Once each of the configurations produced by the grid searches have been analyzed, I will compare the most effective individual configurations from the various general architectures (supervised learning, semi-supervised learning, dense neural network, topological CNN, et cetera) to answer my research questions for both PICO-60 and DEAP-3600.

**Risk and Safety**

There are no safety concerns related to this software project. All interaction with radioactive calibration sources was done by professionals, years prior to this project.

**Bibliography**

1. Kiefer, J. and J. Wolfowitz. “Stochastic Estimation of the Maximum of a Regression Function". *Ann. Math. Statist.* 23.3 (Sept. 1952): 462-466. Web.
2. Brun, Rene and Fons Rademakers. “ROOT - An Object Oriented Data Analysis Framework". *AIHENP '96 Workshop, Lausanne.* 1996. 81-86. http://root.cern.ch/.
3. Hunter, J. D. “Matplotlib: A 2D graphics environment". *Computing In Science & Engineering* 9.3 (2007): 90-95. Print.
4. Pedregosa, F., et al. Scikit-learn: “Machine Learning in Python". *Journal of Machine Learning Research* 12 (2011): 2825-2830. Print.
5. Kingma, Diederik P. and Jimmy Ba. “Adam: A Method for Stochastic Optimization". *CoRR* abs/1412.6980 (2014). arXiv: 1412.6980. Web.
6. Van der Walt, Stefan, et al. “scikit-image: image processing in Python". *PeerJ* 2 (June 2014): e453. Web.
7. Dai, Wei, et al. “Very Deep Convolutional Neural Networks for Raw Waveforms". *CoRR* abs/1610.00087 (2016). arXiv: 1610.00087. Web.
8. Amole, C., et al. “Dark Matter Search Results from the PICO 60 C3 F8 Bubble Chamber". *Phys. Rev. Lett.* 118 (25 June 2017): 251301. Web.