Classifying Particle Interactions with LiquidO using Deep Learning

Kyle Singer

Supervisor: Mark Chen

Department of Physics, Engineering Physics and Astronomy, Queen's University

1. Introduction

In 1956, the first ever electron flavoured anti-neutrino was observed using a (hydrocarbon) liquid scintillator detector (LSD) [1]. Transparent LSDs work by placing light emission detectors, typically photo-multiplier tubes (PMTs) around the outside of the system to detect photons produced during collisions. While transparent LSDs have been proven effective at detecting neutrinos, they struggle with particle energies below 10 MeV. LiquidO is a new type of LSD that uses an opaque (highly scattering) liquid with PMTs running through the liquid. Examples of the images created can be seen in Figure 1. In order to classify the thousands to millions of images produced, a machine learning approach was proposed. The goal is to classify the photon images with a 99% classification rate.

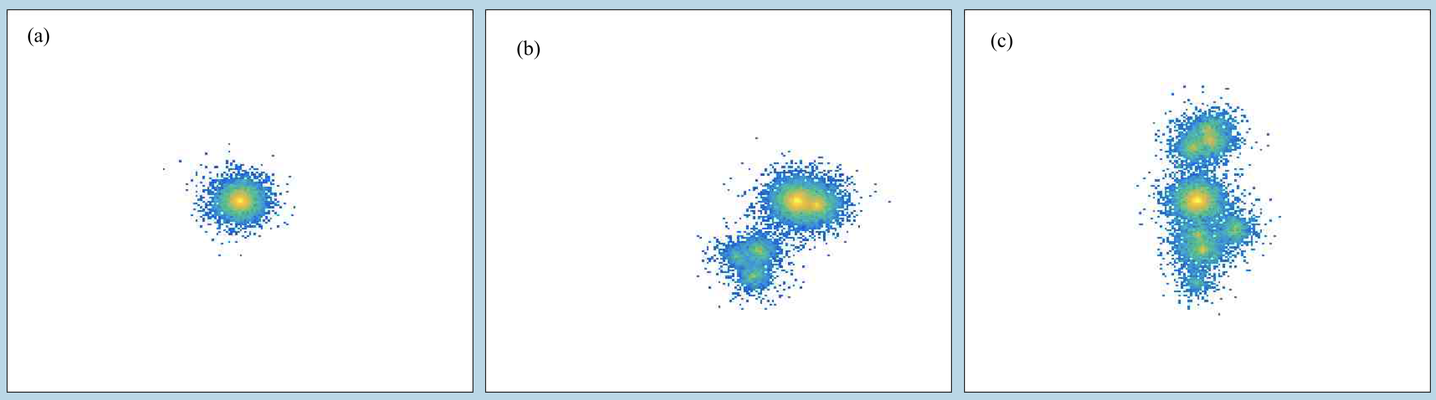

Figure 1: (a) simulation results for an electron with a 1 MeV system. (b) simulation results for a gamma ray with a 2.5 MeV system. (c) simulation results for a positron with a 1 MeV system.

5. Maintainability

The software designed and implemented was done so with a focus on readability and maintainability. A description of the project, instructions on running the code and reproducing the results, as well as some resources and the model structure are all recorded in a GitHub repository. The repo includes all files required needed to train and test the model including step-by-step instructions. The repository which includes the final thesis paper can be found using the QR code below.

2. Model Structure

The model constructed to train on is a Convolutional Neural Network (CNN) called Visual Geometry Group 16 (VGG16). The structure for the model can be seen in Figure 2.

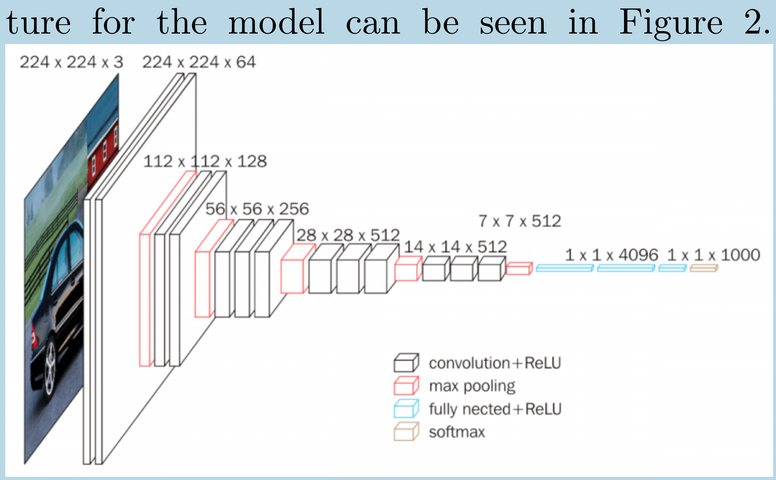

Figure 2: VGG16 CNN. All convolutions (blue) are performed with a 7x7 kernel (filter) size [4].

3. Optimizer

The Adam (adaptive moment estimation) optimizer, was designed based on the methodology around stochastic gradient descent (SGD) but with some improvements. Adam computes individual learning rates for different parameters that can be modified and are based on approximations for first and second moments of the gradients [2]. Adam was chosen as the optimizer as it is well-suited for machine learning (ML) based problems which use large datasets in combination with high-dimensional parameters [2].

4. Cost Function

Cross-entropy ($H(P,Q)$) refers to the average number of bits required to represent an event from one distribution (Q) in another (P) and is seen in Eq. 1. Here, H is the cross-entropy function, P & Q are the expected and predicted probabilities respectively, and x is the class [3].

$$H(P,Q) = -\sum_{x}^{X} P(x) * log(Q(x)) \qquad (1)$$

A cross-entropy cost function was chosen for this model of CNN as the model is built to solve a multi-class classification problem.

6. Results

Once the model structure, optimizer, cost function, and method of implementation (PyTorch) were chosen, the CNN was constructed. Initially trained and tested with 18,000 images (9,000 training, 9,000 testing), the model achieved a 99.4% accuracy. The dataset was then extended to 30,000 images achieving a 99.8% accuracy. To challenge the model, images were randomly shifted ± 20 units from the origin where they were previously being generated from. Testing with this new set resulted in a $\sim$75.3% accuracy, falling below the 99% goal and showing the limitations of the model. The model was then trained on the shifted dataset though still falling short with a $\sim$98% when tested on both shifted and non-shifted images. The ending parameters of the 99.8% model were then used as the new starting point for training with the results on the shifted and non-shifted images being 99.29% (test loss 0.027) and 99.34% (test loss 0.024) respectively. The final model ended with 134,281,283 parameters (weights and biases) all of which are trainable, and with a total model size of 719.60 MB.

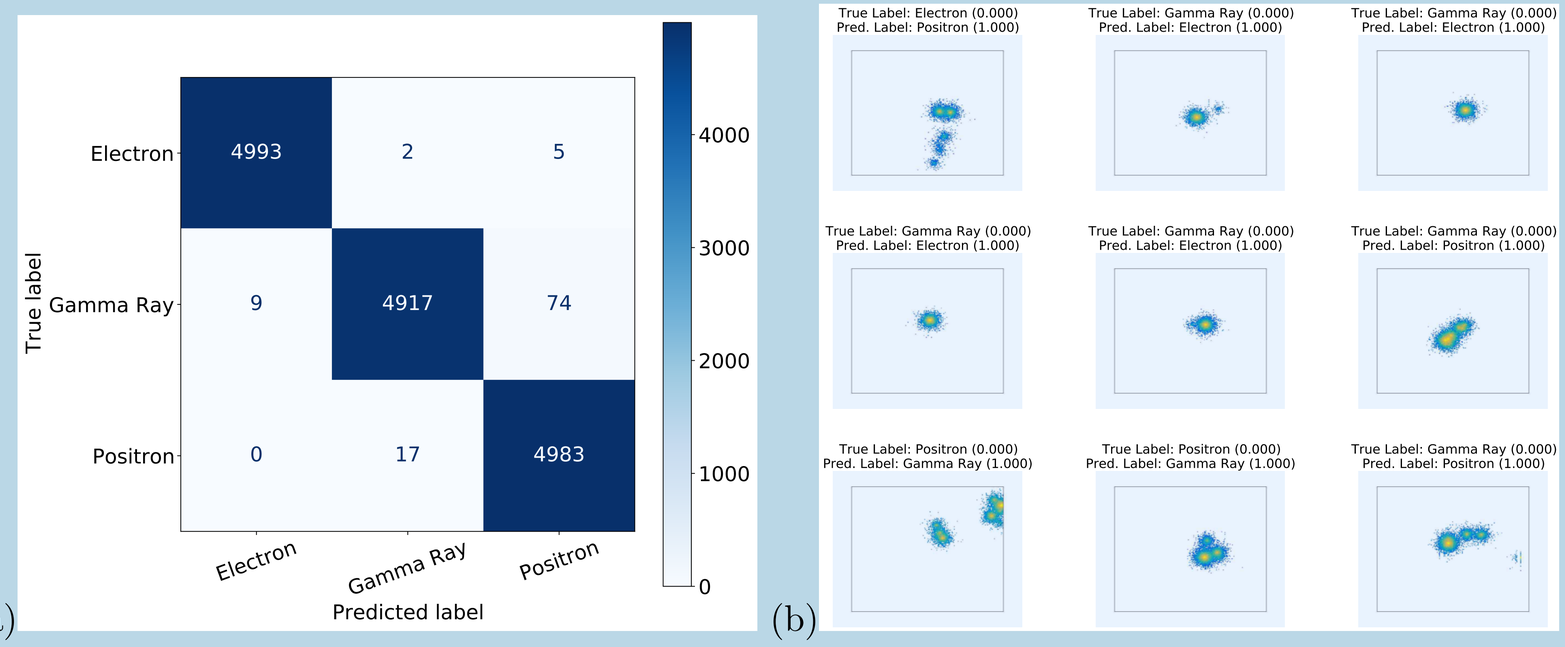

Figure 3: (a) Confusion matrix plotting the true class label of the image against the predicted label. (b) A subset of images misclassified by the VGG16 model from (a).

7. Conclusion

Using a VGG16 CNN combined with an Adam optimizer and cross-entropy cost function for training, the 99% accuracy goal, where no more than 90 images would be misclassified for a 9,000 image dataset was not only achieved but exceeded. The model was able to reach the goal on both shifted and non-shifted images with one model. The model was trained on shifted images where the initial parameter values began with the final parameters of model trained on 15,000 non-shifted images. Moving forward, the model can be improved by diversifying the training set to minimize misclassifications.

References

[1] Cowan, C., Reines, F., Harrison, F., Kruse, H. & McGuire, A. Detection of the free neutrino: A Confirmation. *Science*. **124** pp. 103-104 (1956)

[2] Kingma, D. & Ba, J. Adam: A method for stochastic optimization. *ArXiv Preprint ArXiv:1412.6980*. (2014)

[3] Mc., C. Machine learning fundamentals (I): Cost functions and gradient descent. (Towards Data Science,2017,11), https://towardsdatascience.com/machine-learning-fundamentals-via-linear-regression-41a5d11f5220

[4] Kurama, V. A Review of Popular Deep Learning Architectures: AlexNet, VGG16, and GoogleNet. (Paperspace-Blog,2020)