

Use of Deep Learning to Classify Compton Camera Based Prompt Gamma Imaging for Proton Radiotherapy

CyberTraining: Big Data + High-Performance Computing + Atmospheric Sciences

Jonathan N. Basalyga¹, Gerson C. Kroiz¹, Research assistant: Carlos A. Barajas¹,
Faculty mentor: Matthias K. Gobbert¹, Clients: Paul Maggi² and Jerimy Polf²

¹Department of Mathematics and Statistics, University of Maryland, Baltimore County

²Department of Radiation Oncology, University of Maryland School of Medicine

Technical Report HPCF-2020-14, hpcf.umbc.edu > Publications

Abstract

Real-time imaging has potential to greatly increase the effectiveness of proton beam therapy for cancer treatment. One promising method of real-time imaging is the use of a Compton camera to detect prompt gamma rays, which are emitted along the path of the beam, in order to reconstruct their origin. However, because of limitations in the Compton camera's ability to detect prompt gammas, the data are often ambiguous, making reconstructions based on them unusable for practical purposes. Deep learning's ability to detect subtleties in data that traditional models do not use make it one possible candidate for the improvement of classification of Compton camera data. We show that a suitably designed neural network can reduce false detections and misorderings of interactions, thereby improving reconstruction quality.

Key words. Proton beam therapy, Prompt gamma imaging, Compton camera, Machine learning, Deep learning.

1 Introduction

Proton beams' primary advantage in cancer treatment as compared to other forms of radiation therapy, such as x-rays, is their finite range. The radiation delivered by the beam reaches its maximum, known as the Bragg peak, at the very end of the beam's range. Little to no radiation is delivered beyond this point. By exploiting the properties of the Bragg peak it is possible to only irradiate cancerous tissues, avoiding any damage to the healthy surrounding tissues [9]. However, without some way to image proton beams in real time limitations exist in our ability to take full advantage of the dose delivery properties of the proton Bragg peak. This is due to uncertainties in the beam's position in the body relative to important organs that should not be irradiated.

The Compton camera is one method for real time imaging, which works by detecting prompt gamma rays emitted along the path of the beam. By analyzing how prompt gamma rays scatter through the camera, it is possible to reconstruct their origin. However, the raw data the Compton camera outputs does not explicitly record the sequential order of the interaction data which represents scatterings of a single prompt gamma ray. In addition, it often records false events, which mislabel scatterings of distinct prompt gamma rays as originating from a single ray. These problems make reconstructions based on Compton camera data noisy and unusable for practical purposes [9].

We approach this problem by constructing a neural network. Neural networks represent transformations of data. The network is trained by passing data through it then updating it systematically so as to reduce the loss of its output compared with some desired output. Doing this properly can create a model that exploits subtleties in the data which traditional models are unable to use [4].

In order to reduce the noise present in Compton camera data, we train a neural network to process the data so as to reduce false events and correctly order interactions within events. We then evaluate how this network affects the quality of prompt gamma-based proton beam reconstructions. We first classify a single interaction in an event, rather than the entire event, and achieve an accuracy nearing 90% after using an appropriate input data format and sufficient network complexity. We then approach the task of classifying an entire event, and reach an accuracy of over 80%. This demonstrates the feasibility of using a neural network to accurately classify events.

Prompt gamma emission reconstruction is a problem that we have done previous work on, which can be seen in [1, 3, 5, 6]. In all works we took the CCI reconstruction algorithm and a collection of prompt gamma emission events and attempted to recreate the original proton beam in 3D space. In [1] we focused specifically on improving the reconstruction speed for code which was originally serial in nature. We took the serial code and implemented multi-threading to take full advantage of a multi-core system. In [3, 5, 6] we took the multi-threaded code and implemented an MPI system which allowed for use of a high performance computing environment. The main focus of these works was about increasing the speed of reconstruction and enabling the algorithm to see timing improvements in a high performance computing environment where we can use many nodes with many cores available. It is important to note that these works did not address how the gamma emissions were grouped or classified prior to reconstruction. We created a conic section using all three interactions. If the conic section's properties were beyond certain thresholds, the event was assumed to be bad and was thrown out. In this technical report we specifically discuss the classification of prompt gamma events via a neural network. The network which results from this work would occur before the actual reconstruction process and deals with an entirely different part of the Compton camera to reconstruction pipeline.

The remaining sections of this report are organized as follows: Section 2 introduces proton beam therapy for cancer treatment and its current limitations. Section 3 discusses how Compton camera imaging can be used to overcome the limitations of proton beam therapy and explains how the presence of false events and misordered interactions in Compton camera data limits its practical usage. Section 4 gives a brief overview of neural networks. Section 5 describes the training of a series of networks to classify false events and order interactions. Section 6 presents our conclusions on this work

2 Proton Beam Therapy

Proton beam therapy was first proposed as a cancer treatment by Robert Wilson in 1946 [10]. To a first order approximation, the radiation dosage emitted by a proton beam is inversely proportional to the kinetic energy of the particles within the beam. Because the beam's particles lose kinetic energy as they traverse the patient, the amount of radiation delivered by the beam is low at its entry point, gradually rising until the beam nears the end of its range, at which point the delivered dosage rapidly reaches its maximum. This point of maximum dosage is called the Bragg peak. Little to no radiation is delivered beyond the Bragg peak. These characteristics of proton beam therapy give it a distinct advantage over x-rays. Exploiting its finite range, medical practitioners can confine the radiation of the beam to solely areas affected by cancerous tumors. Vital organs beyond the tumor can be spared [1, 5, 9].

Figure 2.1 shows two horizontal cross-sections of the chest comparing how radiation is delivered by x-ray therapy and by proton beam therapy. At the top of each image is the vertebral body, which contains a tumor that should be irradiated. Since the heart, which is at the bottom center, is still healthy, any radiation delivered to it should be avoided. In the case of x-ray therapy the heart

lies directly in the path of the x-rays. For proton beam therapy, however, all radiation is confined to just the vertebral body. The greater level of precision that proton beam therapy possesses allows for higher dosages of radiation to be delivered to cancerous tissues with minimal damage to healthy tissues. This can lead to better patient outcomes [9].

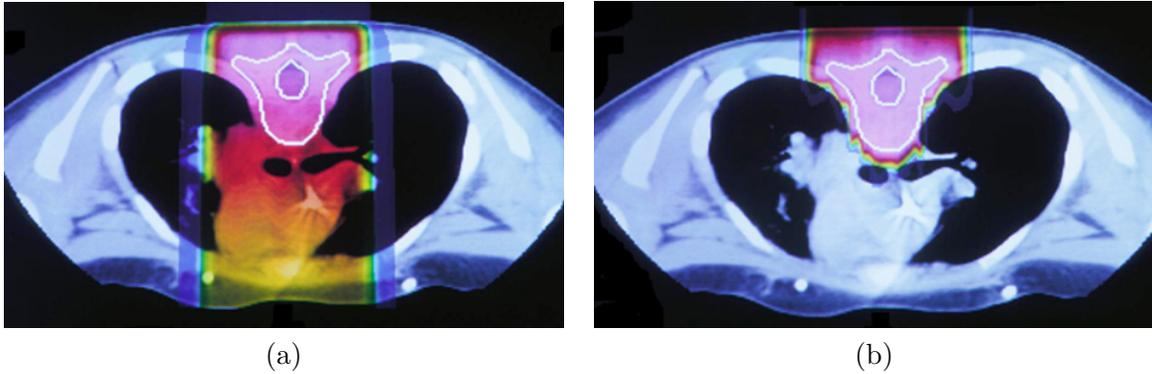


Figure 2.1: (a) X-ray treatment as compared to (b) proton beam treatment.

While the characteristics of proton beam therapy explained above would in principle greatly reduce the negative effects of radiation therapy, there are still practical limitations. In current practice the patient’s body is imaged before undergoing treatment in order to map the position of the tumor. Because proton beam therapy consists of multiple sessions over a period of one to five weeks, the relative size and position of the tumor within the patient’s body may change as surrounding tissues swell, shrink, and shift as a response to radiation. Therefore, whenever using proton beams a margin of error must be added to the position of the Bragg peak in order to fully irradiate the tumor. This rules out certain beam trajectories that would otherwise minimize damage to healthy tissue [9].

Figure 2.2 compares two possible beam trajectories through a cross-section of the chest [9]. In this case the heart, outlined in purple, is positioned at top-center of the figure and a tumor, outlined in green is located next to it. The optimal trajectory, shown in the left image, uses a single beam, which is represented as the space between the dashed white lines, to fully irradiate the tumor while stopping before reaching the heart. However, due to uncertainty in the exact location that the

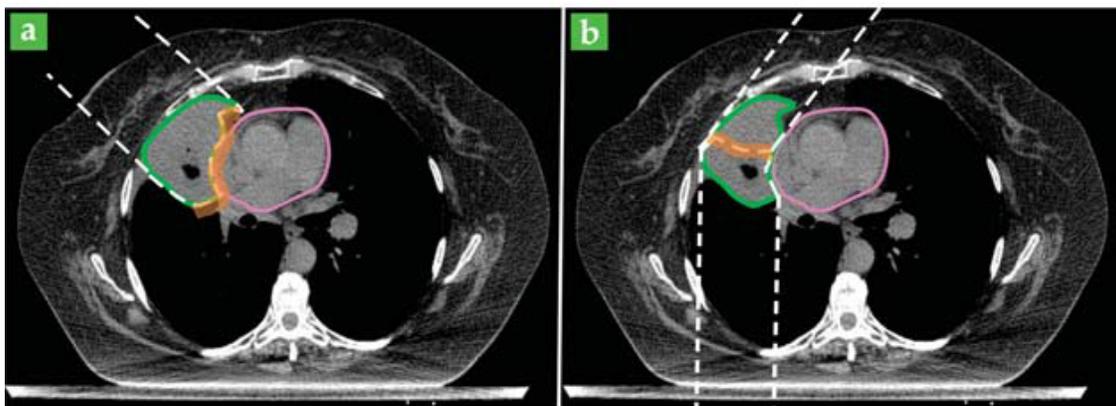


Figure 2.2: (a) Optimal proton beam trajectory. (b) Suboptimal trajectory necessary to protect heart.

Bragg peak occurs (and the beam stops), represented as an orange strip at the end of the beam partially overlapping with the heart, to fully cover the tumor would mean possibly irradiating the heart. Therefore, in practice the trajectory in the right image using two beams is used. Because this trajectory passes through the lungs, delivering a small dose of radiation to them, it is considered suboptimal [9].

3 Compton Camera Imaging

3.1 Introduction to the Compton camera

In order to exploit the full advantages of proton therapy, many researchers are investigating methods to image the beam in real time as it passes through the patient's body [9]. One proposed method for real time imaging is by detecting prompt gamma rays that are emitted along the path of the beam using a Compton camera.

As the proton beam enters the body, protons in the beam interact with atoms in the body, emitting prompt gamma rays. These prompt gamma rays exit the body, some of which entering the Compton camera. Modules within the Compton camera record interactions with energy levels above some trigger-threshold. These modules have a non-zero time-resolution during which all interactions are recorded as occurring simultaneously. For each interaction (that is, Compton scatter) an (x, y, z) location and the energy deposited are recorded. The collection of all interaction data that a camera module collects during a single readout cycle is referred to as an event. [8].

In principle it is possible to use the data that the Compton camera outputs (paired with a suitable reconstruction algorithm) in order to image the proton beam, however this has been shown to only be feasible at low energy levels. At the higher energy levels more typical of proton beam therapy, reconstructions of the beam are far too noisy to be helpful. This is a result of two main limitations in how the Compton camera records events [8]:

- Reconstruction methods typically require that all interactions in an event be chronologically ordered by their occurrence. However, as noted above, due to the camera's non-zero time-resolutions, the camera records all interactions within an event as occurring simultaneously. Therefore, the order of interactions that it outputs is arbitrary.
- Reconstruction methods also assume that all interactions in an event correspond to the same prompt gamma ray. However, since the Compton camera classifies all scatters occurring in the same module during the same readout cycle as belonging to the same event, should two prompt gamma rays enter the same module of the Compton camera during the same readout cycle, the camera would record the resulting interactions in a single event. This results in events that do not correlate to any actual physical event, which are referred to as false events.

At the higher energy levels typically used in treatment, proton beams emit a larger number of prompt gamma rays per unit time, increasing the likelihood of false events. Also, prompt gamma rays are more likely to scatter at higher energy levels, leading to more multi-scatter events, which, as explained above will be unordered. These two effects greatly diminish the accuracy of Compton camera reconstructions at high energy levels, making them unusable [8].

3.2 The representation of events

Multiscatter events can be classified into five categories. A False Triple event consists of three interactions which all originate from separate prompt gamma rays that happened to enter the

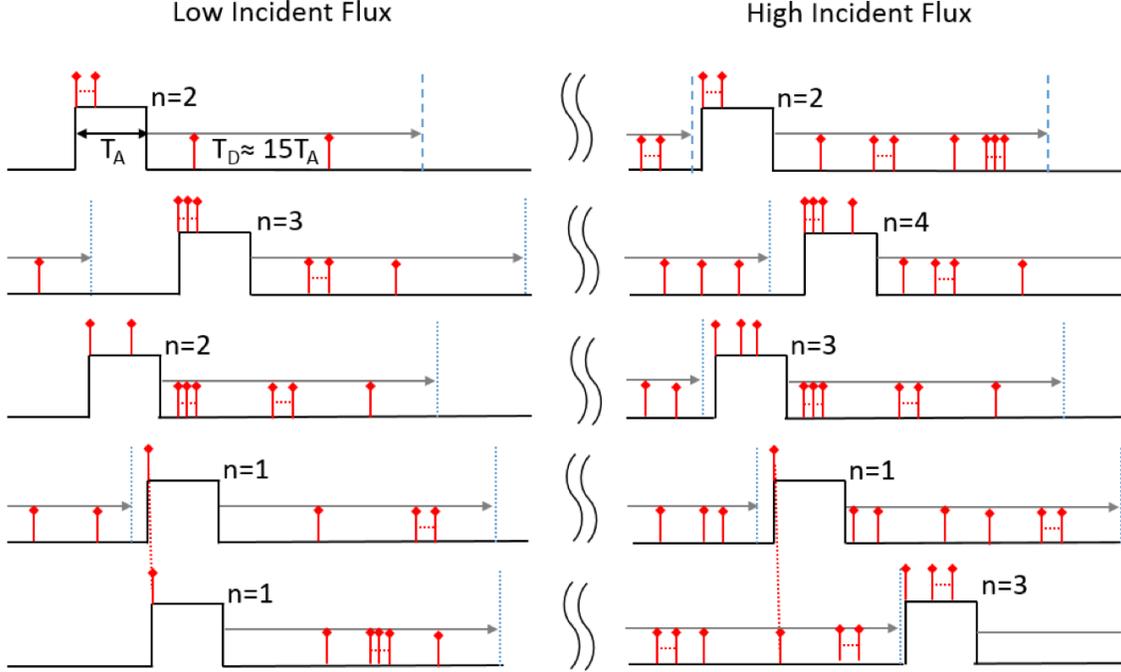


Figure 3.1: An illustration of events.

same module of the camera at the same time. These should be removed from the data before reconstruction. Similarly, False Double events contain two interactions originating from separate prompt gamma rays. These too should be removed. A Double to Triple event contains two interactions corresponding to the same prompt gamma ray, and one interaction from a different prompt gamma ray. The non-corresponding interaction should be removed before reconstruction. The two remaining categories of events are true double and true triple events, which, once properly ordered, can be used for reconstruction.

Figure 3.1 shows a schematic of the Compton camera as it records events. The left side shows events produced at low energy levels and the right shows higher energy levels. Each row represents an independent module of the camera. The red arrows represent scatters, with those originating from the same prompt gamma ray being connected by a dotted line. A single readout cycle within a module of length T_A is represented by a raised pulse. The value n is how many interactions occur during the readout cycle. Looking at just the left side, the first two rows show a True Double and True Triple event, respectively. The third row shows a False Double event consisting of two scatters originating from different prompt gamma rays. The fourth and fifth rows show two True Single events that consist of separate scatters by the same prompt gamma ray. The right side representing higher energy levels shows a far greater proportion of false events.

The raw data output by the Compton camera contains the information shown in Figure 3.2(a). The entire matrix represents an entire event, while each row represents a single interaction. There are three rows because an event can contain up to three interactions. The variable e_i represents the energy level of the i th interaction, where $i = 1, 2, 3$, while (x_i, y_i, z_i) represents the corresponding position. Note that data representing double events still contains three rows even though double events contain only two interactions. This is necessary, since the networks we introduce in Section 5 expect input data to be a consistent size. To account for the missing interaction the third row will be zeroed out. Similarly, single events would have two rows zeroed out, however, single events have

e_1	x_1	y_1	z_1
e_2	x_2	y_2	z_2
e_3	x_3	y_3	z_3

(a)

e_1	x_1	y_1	z_1	$\delta e_{1,2}$	$\delta r_{1,2}$
e_2	x_2	y_2	z_2	$\delta e_{2,3}$	$\delta r_{2,3}$
e_3	x_3	y_3	z_3	$\delta e_{3,1}$	$\delta r_{3,1}$

(b)

Figure 3.2: (a) The initial input format representing a single event. (b) The appended input format including distances and energy distances between each interaction.

only one possible ordering, so there is no need to classify them.

To improve the performance of our networks, we find it useful to use the appended data shown in Figure 3.2(b). In this version we add the distances $\delta r_{i,j}$ and the differences in energy levels $\delta e_{i,j}$ between the i th and j th interactions, where $i, j = 1, 2, 3$. Since these values have physical significance with regards to the ordering of interactions, explicitly including them in the data makes it easier for the networks to learn. For double events, the non-relevant $\delta r_{i,j}$ and $\delta e_{i,j}$ are set to zero. We use the value zero rather than NaN because our networks require that all data consist of real numbers.

4 Deep Learning

We propose to train a neural network to process the data output by the Compton camera by removing false events and properly ordering the camera events within events.

The structure of a fully connected neural network is shown in Figure 4.1 [2]. The network contains three main components: an input layer which accepts the data, hidden layers which each perform some transformation on the data, and an output layer which returns the transformed data in some prescribed format [4]. We would like to train the neural network to transform the provided data in some useful way. In the case of the data output by the Compton camera, we would like the neural network to transform each multi-scatter event so that it contains only interactions originating from the same prompt gamma ray, and so that these interactions are in the correct order.

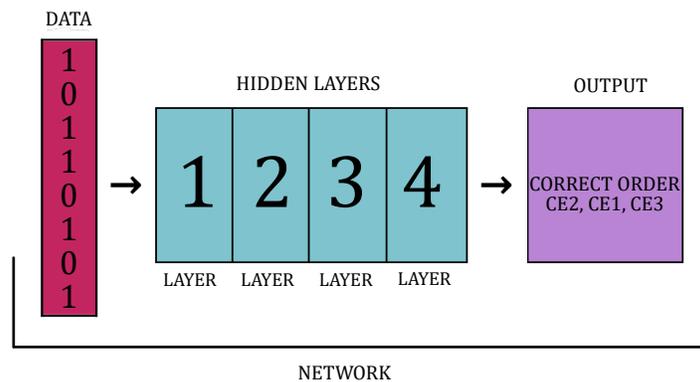


Figure 4.1: The structure of a fully connected neural network.

Figure 4.2 shows the training and testing process for a neural network during what is called supervised learning [2]. Supervised learning refers to training the neural network using labels for the data which provide what the transformed data is supposed to look like. By feeding data into

the network and comparing it with the corresponding labels using a suitable loss function, we can calculate the current loss of the neural network. The neural network can then be updated using an optimization function. After training the network, it is then tested on data it has not seen before. If the network performs well on data it was not trained on, this indicates that the network’s model generalizes well and can be used on real-life data.

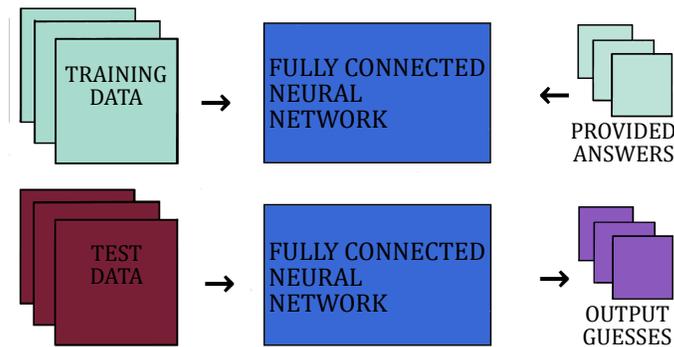
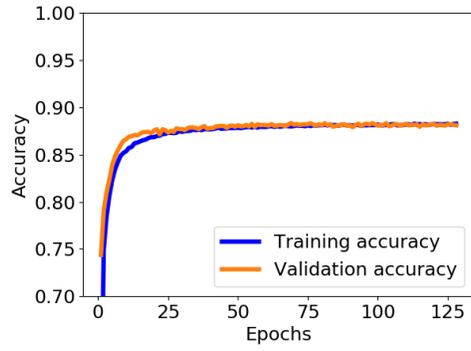


Figure 4.2: The training and testing process for a fully connected neural network.

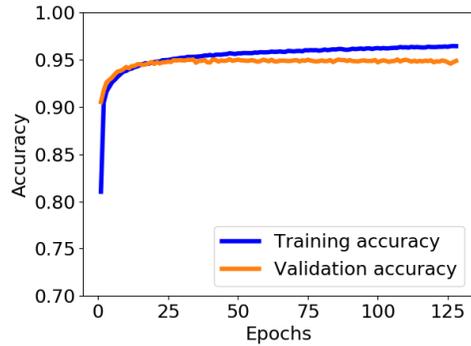
To improve the network’s performance, it is typical to train the network on all available data multiple times. One pass through all the training data is referred to as an epoch. Often, the network will be trained for hundreds or thousands of epochs. It is standard practice to set aside some data with which to evaluate the network after each epoch. These data are called the validation data. By evaluating the network at the end of each epoch it is possible to plot how the network’s performance improves over the training process, giving insight into whether or not the network has been fully trained. After the network has finished training, a final data set separate from the training data and validation data is used to test the network. This data set is referred to as the testing data.

One of the primary difficulties in deep learning is training a network so that it properly fits the data it is being applied to. When there is still information in the data that has not been incorporated into the network’s model this is referred to as underfitting. This occurs because the network has either not been trained enough, or because the network size is too small to fully process the data. When the network performs very well on the data it has been trained on but does not generalize to data that it has not been trained on, this is called overfitting. Overfitting occurs because the network has begun directly mapping inputs to outputs, that is, “memorizing” the data. Since large networks have a greater capacity to store information about the data, they are more likely to overfit. Therefore, using a larger network does not necessarily lead to better performance [4].

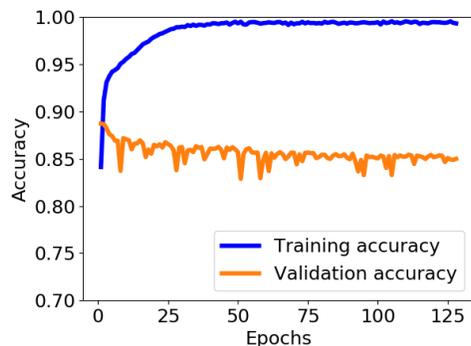
Figure 4.3 compares the accuracy curves of three networks which originating from a textbook example as they are being trained [4]. In each plot the blue curve represents training accuracy, that is, how well the network performs on training data at each epoch, while the orange curve represents validation accuracy, which measures how well the network performs on a validation set composed of data the network was not trained on. The left plot shows training and validation curves typical of underfitting. Since the network still has not fully internalized distinguishing features in the training data, it performs just as well on the validation data as it does on the training data. The



(a) underfitting



(b) suitable fitting



(c) overfitting

Figure 4.3: The training and validation accuracy curves representative of underfitting, a suitable fitting, and overfitting.

plot on the right shows an overfitting curve. Here there is a large gap between training accuracy and validation accuracy, indicating a lack of generalization. The plot in the center shows what the training and validations curves of a suitable fitting look like. The training and validation curves are just beginning to diverge, but are still very close. This occurs because the network has incorporated as much information from the model as it can, so any additional training either has no effect on the validation accuracy, or even lowers it.

5 Results

We now train a series of fully connected neural networks to classify events and properly order the interactions with an event. Subsection 5.1 presents networks that classify only a single interaction, whereas the network in Subsection 5.2 classifies and reorders a complete event. All of these networks use a large number of layers, as many as 24, so residual skips are used to support this network size. Dropout layers are used as well.

The networks in this work were trained using the GPU portion of the taki cluster, maintained by the UMBC High Performance Computing Facility, which has the following specifications:

- 1 GPU node from 2018 containing four NVIDIA Tesla V100 GPUs (16 GB onboard memory) connected by NVLink and two 18-core Intel Skylake CPUs. The node has 384 GB of memory (12×32 GB DDR4 at 2666 MT/s) and a 120 GB SSD disk.
- 18 hybrid CPU/GPU nodes from 2013, each containing two NVIDIA K20 GPUs (2496 computational cores, 5 GB onboard memory) and two 8-core Intel E5-2650v2 Ivy Bridge CPUs (2.6 GHz clock speed, 20 MB L3 cache, 4 memory channels). Each node has 64 GB of memory (8×8 GB DDR3) and 500 GB of local hard drive. The nodes are connected by a QDR (quad-data rate) InfiniBand switch.

5.1 Classifying a Single Camera Event

We first train a series of networks which only classify the first interaction of an event. Initially they use the data format shown in Figure 3.2(a) as their input. A softmax activation function is used to output the probability of each possible place of the interaction in the sequence of all camera events, that is, the probability that the interaction came first, second, third, or that it is actually not a part of the event at all.

Figure 5.1(a) shows the training and validation accuracy of our initial version of the network after being trained for 400 epochs on normalized data. This network is able to learn, capping out at an accuracy around 70%, however the network underfits the data. This is indicated in the plot by the fact the validation curve still lies on top of the training curve at the end of training.

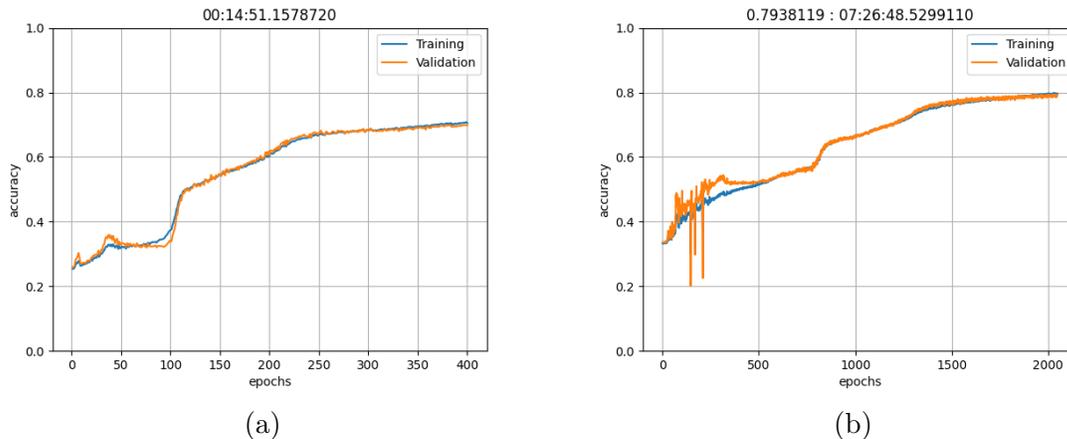


Figure 5.1: Training accuracy and validation accuracy of initial version of the network trained on normalized data for (a) 400 epochs and (b) 2000 epochs.

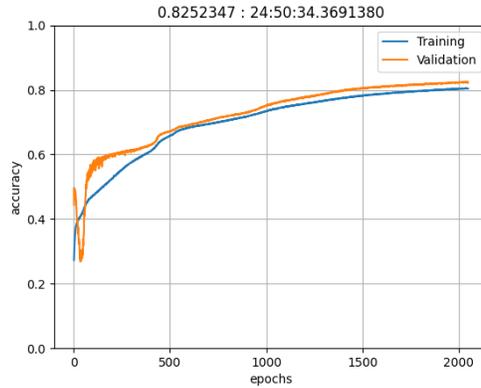


Figure 5.2: Training accuracy and validation accuracy of 16 layer network using a SeLU activation function trained on normalized data for over 2000 epochs.

We take two actions to eliminate the underfitting. The first is to increase the number of epochs the network is trained on to 2000. The results of this are seen in Figure 5.1(b). The network achieves a higher accuracy, this time around 80%, but the training and validation curves still lie on top of each other, indicating underfitting. Therefore we increase the network complexity by using 16 hidden layers. This is shown in Figure 5.2. At this point our network begins to fit the data, as seen by the small gap between the training and validation curve. Validation accuracy comes to slightly more than 80% by the end of training. Note that the validation curve lies above the training curve. This is explained by our usage of dropout layers.

The next improvement we make to the network's performance comes by using the appended input format shown in Figure 3.2(b). By making this change, as well as using five times as much data as is used by the previous networks, the network is able to reach an accuracy of about 88%, as seen in Figure 5.3. However, in this case the network is still underfitting after 1200 epochs.

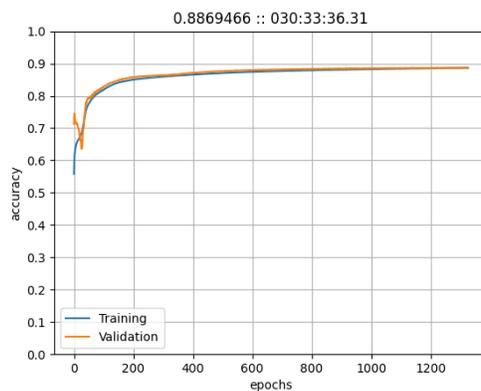


Figure 5.3: Training accuracy and validation accuracy of 16 layer network using a SeLU activation function trained on 5 times as much data, appended with Euclidean distances and energy differences, for over 1200 epochs.

5.2 Classifying a Complete Event

In order to classify all the interactions in an event simultaneously, thereby classifying the event itself, we train a new network which predicts the correct permutation of interactions. For example, the output “314” indicates a Double to Triple event where the correct interactions are the first and third interactions of input data, and the correct ordering of their occurrences places the third interaction from the input as the first to actually occur, then the first interaction from the input as the second to actually occur. The network continues to use a softmax activation function, so its output is the probability of every possible permutation of interactions. Our input layer consists of a dense 15 neuron layer, corresponding to the data structure show in Figure 3.2(b). The hidden layers are 24 fully connected layers with a 4096 square dimension. Our output layer is a dense 24 neuron layer, corresponding to the 24 possible permutations. We use a residual skip every four layers. We fix learning rate at 10^{-6} and batch size at 8192, and we train for over 1000 epochs.

Figure 5.4(a) shows the training and validation accuracy curves of this new network. The network achieves a validation accuracy of about 82% after over 1000 epochs of training. This accuracy is lower than what the network classifying only individual interactions achieves, however this new network is performing a more complicated task.

Figure 5.4(b) shows the validation accuracy for each individual event type. This shows that the network performs better on some event types than others. The network classifies Double to Triple events with an accuracy of over 90%, False Doubles and False Triples are classified with an accuracy of slightly more than 80%, while True Doubles and True Triples are correctly classified less than 80% of the time. This is possibly due to an input class invariance, that is, the frequency of occurrence differs across event types, so the network can incorporate more information into its model about some input classes than it can about others.

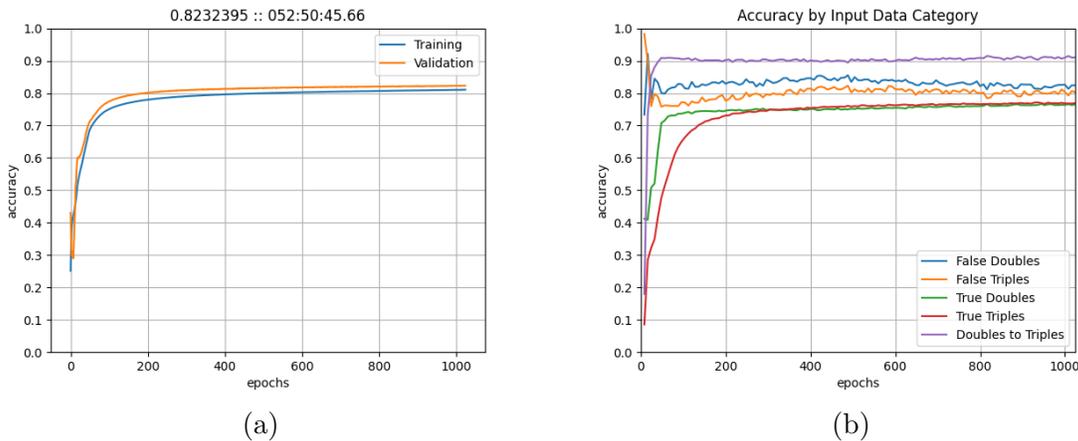


Figure 5.4: (a) Training accuracy and validation accuracy of network which classifies complete events, trained for over 1000 epochs. (b) Validation accuracy separated by input category.

6 Conclusions

The series of networks trained in Section 5.1 demonstrated that it was possible for a fully connected neural network to learn from the Compton camera data and classify single interactions at a reasonable accuracy. It was also shown that augmenting the interactions with information about distance and energy difference from other interactions in the event allows the network to learn more

quickly and attain a higher accuracy, even if this information could be directly deduced from the unaugmented data.

Using what we learned classifying single interactions, we showed in Section 5.2 that it is also possible to order and classify entire events at only a slightly lower accuracy. We found that the network classified some event types, such as double-to-triple events, at a much higher accuracy than it classified other events, like true doubles and true triples. This is likely due to class imbalances in the data.

To demonstrate the effect that classification has on proton beam reconstructions, Figure 6.1 shows three different simple back-projection reconstructions. Note that back-projection is one of the simplest reconstruction methods, and more sophisticated methods will produce clearer reconstructions [7]. Figure 6.1(a) was created using just the raw data output by the Compton camera. The large amount of noise in it makes it very difficult to identify the location of the Bragg peak. Figure 6.1(b) was created using Compton camera data that is classified at the same accuracy as in Figure 5.4(b). It demonstrates what effect the neural network in Section 5.2 would have on reconstructions. While there is still a great deal of noise, the shape of the beam is clearer and it is possibly to pick out the approximate location of the Bragg peak with the naked eye. Figure 6.1(c) uses perfectly classified Compton camera data to reconstruct the beam. It shows the limit of how much a neural network such as the one trained in Section 5.2 can improve reconstruction quality.

To our knowledge, this work represents the first attempt to use a neural network to classify and order Compton camera data. While the accuracy curves in Figure 5.4 and the reconstruction in Figure 6.1(b) clearly show room for improvement, they demonstrate the feasibility of applying neural networks to this problem. By exploring further network designs it may be possible to attain higher accuracies and produce better reconstructions.

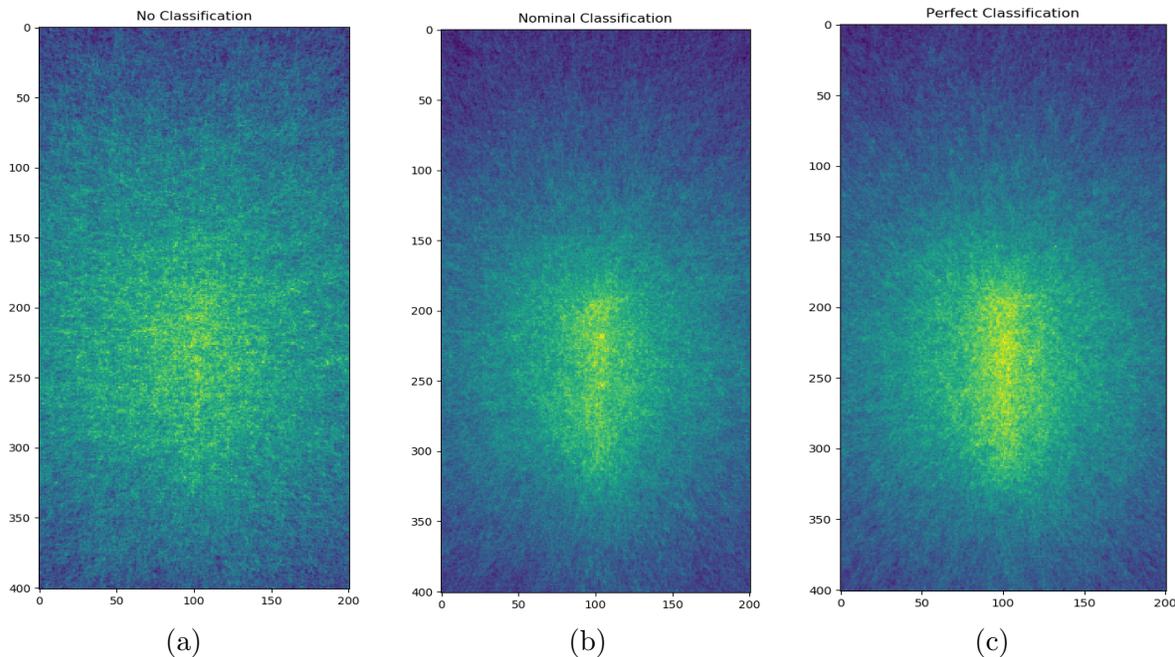


Figure 6.1: Reconstructions using (a) unclassified data, (b) data classified with the accuracies shown in Figure 5.4(b), and (c) data classified with perfect accuracy.

Acknowledgments

This work is supported by the grant “CyberTraining: DSE: Cross-Training of Researchers in Computing, Applied Mathematics and Atmospheric Sciences using Advanced Cyberinfrastructure Resources” from the National Science Foundation (grant no. OAC-1730250). The research reported in this publication was also supported by the National Institutes of Health National Cancer Institute under award number R01CA187416. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. The hardware used in the computational studies is part of the UMBC High Performance Computing Facility (HPCF). The facility is supported by the U.S. National Science Foundation through the MRI program (grant nos. CNS-0821258, CNS-1228778, and OAC-1726023) and the SCREMS program (grant no. DMS-0821311), with additional substantial support from the University of Maryland, Baltimore County (UMBC). See hpcf.umbc.edu for more information on HPCF and the projects using its resources. Co-author Carlos Barajas additionally acknowledges support as HPCF RA.

References

- [1] Fernando X. Avila-Soto, Alec N. Beri, Eric Valenzuela, Abenezer Wudenhe, Ari Rapkin Blenkhorn, Jonathan S. Graf, Samuel Khuvis, Matthias K. Gobbert, and Jerimy Polf. Parallelization for fast image reconstruction using the stochastic origin ensemble method for proton beam therapy. Technical Report HPCF-2015-27, UMBC High Performance Computing Facility, University of Maryland, Baltimore County, 2015.
- [2] Carlos A. Barajas. *An Approach to Tuning Hyperparameters in Parallel: A Performance Study Using Climate Data*. M.S. Thesis, Department of Mathematics and Statistics, University of Maryland, Baltimore County, 2019.
- [3] Johnmuel Casilag, James Della-Giustina, Elizabeth Gregorio, Aniebet Jacob, Carlos Barajas, Matthias K. Gobbert, Dennis S. Mackin, and Jerimy Polf. Development of fast reconstruction techniques for prompt gamma imaging during proton radiotherapy. Technical Report HPCF-2017-16, UMBC High Performance Computing Facility, University of Maryland, Baltimore County, 2017.
- [4] François Chollet. *Deep Learning with Python*. Manning, 2018.
- [5] James Della-Giustina, Carlos Barajas, Matthias K. Gobbert, Dennis S. Mackin, and Jerimy Polf. Hybrid MPI+OpenMP parallelization of image reconstruction in proton beam therapy on multi-core and many-core processors. In *Proceedings of the Symposium on High Performance Computing, HPC '18*, pages 1–11. Society for Computer Simulation International (SCS), 2018. article 11.
- [6] James Della-Giustina, Johnmuel Casilag, Elizabeth Gregorio, Aniebet Jacob, Carlos Barajas, Matthias K. Gobbert, Dennis S. Mackin, and Jerimy Polf. Speedup potential for reconstruction techniques for prompt gamma imaging during proton radiotherapy. *American Journal on Undergraduate Research*, 14(4):23–37, 2018.
- [7] Mirela Frandes, Bogdan Timar, and Diana Lungeanu. Image reconstruction techniques for Compton scattering based imaging: An overview [Compton based image reconstruction approaches]. *Curr. Med. Imaging*, 12(2):95–105, 2016.

- [8] Paul Maggi, Stephen W. Peterson, Rajesh Panthi, Dennis S. Mackin, Hao Yang, Zhong He, Sam Beddar, and Jerimy Polf. Computational model for detector timing effects in Compton-camera based prompt-gamma imaging for proton radiotherapy. *Phys. Med. Biol.*, online April 22, 2020.
- [9] Jerimy C. Polf and Katia Parodi. Imaging particle beams for cancer treatment. *Phys. Today*, 68(10):28–33, 2015.
- [10] Robert R. Wilson. Radiological use of fast protons. *Radiology*, 47(5):487–491, 1946.