Promising Hyperparameter Configurations for Deep Fully Connected Neural Networks to Improve Image Reconstruction in Proton Radiotherapy

Sokhna A. York
Center for Data, Mathematics, and Computational Sciences, Goucher College

Alina M. Ali
Department of Mathematics and Statistics University of Houston–Downtown

David C. Lashbrooke Jr.
Department of Mathematics and Department of Statistics, Purdue University

Rodrigo Yepez-Lopez
Department of Computer Science American University

Carlos A. Barajas
Department of Mathematics and Statistics University of Maryland, Baltimore County

Matthias K. Gobbert
Department of Mathematics and Statistics University of Maryland, Baltimore County

Jerimy C. Polf
Department of Radiation Oncology University of Maryland School of Medicine

***Abstract*—Proton therapy is a unique form of radiotherapy that utilizes protons to treat cancer by irradiating cancerous tumors while avoiding unnecessary radiation exposure to surrounding healthy tissues. Real-time imaging of prompt gamma rays can be used as a tool to make this form of therapy more effective. The use of Compton cameras is one proposed method for the real-time imaging of prompt gamma rays that are emitted by the proton beams as they travel through a patient's body. The non-zero time resolution of the Compton camera, during which all interactions are recorded as occurring simultaneously, causes the reconstructed images to be noisy and insufficiently detailed to evaluate the proton delivery for the patient. Deep Learning has been a promising method used to remove and correct the different problems existing within the Compton Camera's data. Previous papers have demonstrated the effectiveness of using deep fully connected networks to correct improperly ordered gamma interactions within the data. We do a moderately large hyperparameter grid search to find a promising set which yields competitive performance but contains fewer neurons making it compact. The studies which have many neurons, many layers, and a non-zero dropout rate have the best testing accuracy. These many neuron and many layer networks still have significantly fewer total neurons than the current neural network implementation. If given considerably more training time these compact networks could yield equal, if not superior, testing accuracy when compared to larger networks. More improvements are still needed for clinical use and we are currently experimenting with recurrent neural networks to test the viability of this type of architecture for this application.**

I. Introduction

Because of its many advantages, proton therapy has been increasingly growing in popularity as a form of cancer treatment. Most types of radiation work with a similar objective to damage the cellular DNA of target cancer cells that reside in the nucleus of every cell. Although X-ray therapy is able to deliver dosage at the tumor site, the radiation continues to travel through the body until it exits the other side. This may potentially cause harm to healthy surrounding tissues and organs that are unnecessarily exposed to radiation. By contrast, proton therapy has the advantage to deliver radiation dosage directly at the tumor site without travelling further posterior into the body. This advantageous characteristic of proton therapy is called the Bragg Peak [13]. The Bragg Peak makes it possible to spare any surrounding healthy tissues from unnecessary radiation damage. In order to take full advantage of all of the perks that proton therapy has to offer, we must have an efficient technique to image the prompt gamma rays in real-time as they travel through the patient's body.

The Compton camera is one promising technique to get real-time imaging of the proton beam, by detecting prompt gamma rays emitted along the path of the beam [10]. Unfortunately, some of the data produced by the Compton camera is not always usable due to the non-zero time resolution of the Compton camera, during which all interactions are recorded as occurring simultaneously. The use of the raw data by the Compton camera with modern reconstruction algorithms yield noisy and insufficiently detailed images to evaluate the proton delivery for the patient.

Fully connected neural networks have shown promise in their ability to detect interaction for true triples in [14]. They note that the neural network has comparable accuracy to classical Compton camera sequencing techniques. They go on to highlight that there are superior methods to neural networks and the classical method but they are too computationally intensive for any real-world use case. A different usage of neural networks was seen in [11] where the authors simply trained a network to determine whether data was "good" or

NSF (grant no. OAC–2050943 et al.), NIH (award number R01CA187416).
978-1-6654-3902-2/21/$31.00 ©2021 IEEE

"bad" and then used the "good" data for reconstruction. They use true triples and false events and noticed that the network had respectable accuracy and improved the good data to bad data ratio that can be used for reconstruction. Both of these examples use simpler Monte-Carlo simulations, shallow neural networks, and do not consider how neural networks can play a role in a broader clinical application. When dosage rates change, the amount of data pollution increases greatly to the point that there is more unusable data than there is usable data [10].

This work's specific purpose differs greatly from some of the previous work done on this topic. The earliest form of work was seen in tech. report [6] which started training a neural network to order a single interaction with the hope of expanding the neural network to do multi-task classification for interaction ordering. Inevitably this settled on just a single-task classification method using ordering permutations and published a portion of those results in [4]. Then in [5] the neural network structure was changed from a shallow but wide design to a long but thin design and also trained using a python generator which feeds one input type (called double/triple/double-to-triple/false) at a time in an attempt to improve accuracy without changing the neural network structure. It is important to note that up to this point all the doubles-to-triples, false triples, and false doubles had been generated during a preprocessing step rather than coming directly from a Monte-Carlo simulation. In the tech. report [2], several changes had been made to the training process, data generation, preprocessing, and network design. Doubles and Triples had been separated into their own categories and a specialized neural network was trained for each. All data was generated using a Monte-Carlo simulation instead of stitched together during preprocessing. A residual block structure was implemented to allow for a deep fully connected neural network instead of shallow one. The results for the doubles only neural network were published in [9] and the results for the true triples plus doubles-to-triples were published in [3]. The final aim of [2] was to create a neural network which is capable of quickly and accurately ordering interactions for triples and doubles while also detecting false couplings. The ability to reorder and detect false couplings is known to improve reconstruction quality and can allow for Compton cameras to be used in a clinical setting. The clinical viability and impact of the neural network on reconstructions is the focus of [12]. At this point we have seen that deep neural networks have great success when ordering interactions and detecting false couplings. In this publication, we report the promising attempts to create a more compact neural network than is seen in [2]. We define a "compact network" as a network with similar or superior performance but containing fewer total parameters and fewer neurons than the previous networks with similar function. The fewer neurons the network has, the more computationally cheap it is to use, and the faster the network can classify records.

The remainder of this publication is organized as follows: Section II contains more in depth information about the distinct advantages that proton therapy provides versus other forms of radiotherapy. Section III describes how Compton camera imaging can be used to improve the restraints of proton therapy and how the misordered interactions and the presence of false events can limit the usage of the Compton camera. Section IV gives a background on neural networks, as well as explaining the network's conception. Section VI describes 288 different hyperparameters studies, some promising results, and how the network performed on different beam energies. Section VII discusses the impact, shortcomings, and overall successes associated with our goals and results as well as ongoing work.

II. Proton Therapy

Radiation therapy is a form of cancer treatment that uses high doses of radiation that act to kill cancer cells and ultimately the tumor. X-ray therapy is one common technique used for cancer treatment. In this type of therapy, the majority of the radiation dosage is delivered upon entering the body. Because of this, the tumor does not receive as high of a concentrated dose as it should. In addition, X-rays will continue to travel posterior into the human body until it exits out the other side. This is not ideal as there is no need for extra radiation exposure within the body. Proton therapy on the other hand is more efficient in this manner. Rather than depositing the majority of the dosage at the entry site, proton therapy works to deposit the majority of the dosage at the tumor site itself, thus making the process more effective. Proton therapy also has an advantage over X-ray therapy in the sense that the proton beam travels no further posterior into the body than the site of the tumor allowing for minimal exposure to surrounding tissue.

The proton therapy delivery process is quite unique. The protons start their journey inside of a vacuum tube that connects to a linear accelerator that leads to a cyclic particle accelerator called a synchrotron. This is where the protons stay until their energies reach a point at which they can reach any given depth inside the body of the patient. Once this energy level is reached, protons navigate through a beam-transport system that consists of many magnets that guide the protons to their target location. There are two tools involved in this guiding process. The nozzle-like aperture, acts to shape the beam of protons while the compensator acts to shape the protons into a 3-D shape which allows the protons to travel as deep as they need to into the body to reach the tumor. Once the proton beam has reached the depth of the tumor, the beam deposits the majority of its energy into the tumor.

Depending on the size of the tumor, the beam may have to kill the tumor cells layer by layer. When delivering a dosage to a tumor, the professional who is treating the patient will create what is called a safety margin. This safety margin enlarges the treatment area to ensure that all parts of the tumor are guaranteed to receive dosage. The safety margin is needed to account for slight movements in the patient during treatment as well as sligthly different positioning of the patient from one treatment to the next over several weeks.

If real-time information on the trajectory of the proton beam through the patient's body were available during a treatment, the safety margin could be smaller. The use of Compton cameras is one proposed method for the real-time imaging of prompt gamma rays that are emitted by the proton beams as they travel through the body.

III. Compton Camera

A. Background on Compton Camera

Compton Cameras are multistage detectors that generate images of gamma rays through Compton scattering, [12]. As the protons penetrate the human body, they interact with atoms in the body, leading to prompt gamma rays emission. As those gamma rays exit the body, some of them collide with the modules in the Compton Camera. Modules of the camera then measure the energy deposited by the prompt gamma and its position as it passes through different detection stages of the camera. For each Compton scatter the camera records x-, y-, z-coordinates and the energy level of the scatter. The readout of interactions in a single period is called an event. The raw output data from the camera for each interaction is in the form (e_i, x_i, y_i, z_i) where $i = 1, 2, 3$, and e_i is the energy level.

Image reconstruction algorithms exist that can recover the path of the proton beam from the Compton camera data. The Compton camera's capability to reconstruct full 3D images of the proton beam range could be used with the patient's CT to compare the planned treatment dose and make adjustments. Radiotherapy treatment requires a conformity between the treatment plan and the treatment delivery, making sure that patient's bone and soft tissue landmarks are aligned as they were at the time of treatment planning [13]. Having a patient change position, wiggle, scratch, look the other way, or any other subtle movement could cause disruption in the treatment plan. By obtaining reliable information regarding the patient from the reconstructed images, clinicians have the opportunity to better ensure that the entire tumor receives the exact dose as planned while making sure surrounding healthy tissues are safe.

B. Image Reconstruction and Representation of Scattering Events

During image reconstruction, existing algorithms require that the interactions within an event happen sequentially. The modules residing in the camera that record interactions have a non-zero time-resolution. This means that the time it takes for the camera to record interactions is slower than the time it takes gamma rays to pass through the camera. This results in the interactions being detected as happening simultaneously. Reconstruction algorithms assume that interactions happen from a single prompt gamma ray source, yet the Compton camera may combine several single scatters into one event. This noisy data makes the reconstruction more difficult, because the image reconstruction algorithms have set prerequisites that are not met by the Compton Camera data output. Usually, at lower energy levels, the raw data recorded by the Compton Camera can still be directly used with adept algorithms for reconstruction. However, it has been observed that at higher energy levels, such as those used during proton radiotherapy, the raw data has proven to be ineffective for image reconstruction [9]. At higher energy levels, the proton beams tend to emit increased numbers of prompt gamma rays, which then increases the likelihood for the camera to record false events, making the reconstructed images noisy.

The scattering events can be classified into five groups: True Triples, True Doubles, False Triples, False Doubles, and Double to Triple. A True Double and True Triple implies that the gamma-ray interacted exactly with the camera twice or three times, respectively. These are the events needed for the reconstruction of the images. A False Double event consists of two separate gamma rays interacting simultaneously with the camera, thus recorded as a True Double. Similarly, a True Triple encloses three separate gamma rays that interacting with the camera at the same time, thus falsely recorded by the module of the camera. These two False events need to be removed from the data, as they cause noise and corruption. Lastly, the Double to Triple occurs when two interactions were made by a single gamma ray and another interaction made by a different gamma-ray. The last interaction should be removed from the event for a better reconstruction process.

IV. Deep Learning

A. Introduction to Deep Learning

Deep learning is a subfield of Machine Learning. It uses a successive multi-layer structure, thus the "deep" in the name, called neural network. Neural networks serve the purpose of helping the model learn by identifying patterns present in the data and classifying information based on the learned patterns. A fully connected deep neural network (DNN) contains a series of fully connected layers that work to connect each node or "neuron" in one layer to each in the next layer. Figure 1 exhibits the architecture of a fully-connected network. A DNN is composed of an input layer, which takes in the data, hidden layers that specifically transforms the data using some function and an output layer that returns a specific format of the transformed data. Normally, a data set is divided into two parts, a training set and a testing set. Within the training set, the data is divided into training data and validation data. The training data are used to find an optimal set of connection weights, the test data are used to choose the best network configuration, and once an optimal network has been found, a validation set is required in order to test the true generalization ability of the model [7]. When the entire training data goes through the network is called an epoch.

We run the data through the network for thousands of epochs, in order to improve its accuracy. There is a way to evaluate the network after each epoch to track the network's learning ability throughout the learning process. Moreover, deep learning models do not evaluate the entire data in the model at once, rather they separate it in different batches. One of the biggest challenges in machine learning is the dynamics between generalization and optimization. Optimization refers to the process of adjusting a model to get the best performance

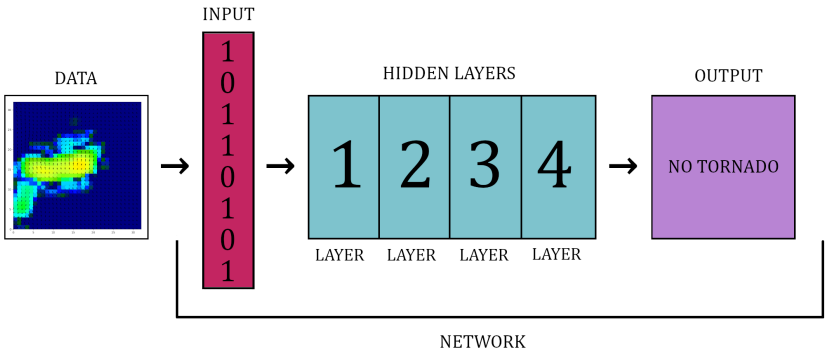

Fig. 1. A fully connected neural network architecture.

possible on the training data, whereas generalization is how the network performs on data it had never seen before. At the beginning of the training phase, generalization and optimization are correlated and the network has not learned all the patterns in the training data yet. This phenomenon is referred to as underfitting. After a certain number of epochs, the network's generalization stagnates and there is no improvement, this is called overfitting [7]. It performs well on the training data, but does not do as well on new data that it has not trained on rather the network starts to memorize the data. For this application, the network should have the ability to transform each scatter event in a way that it orders the interactions originated from the same prompt gamma ray in the correct order.

B. Fully Connected Residual Blocks

Neural networks, especially fully connected ones, break down once they start becoming notably deep and complex [2]. The issues encountered with deep networks is that the values become smaller and smaller until getting to zero and like non-zero values as you go deeper and deeper. During back propagation we start to see the gradient becoming like-zero causing little to no update to existing weights which causes learning stagnation. This phenomenon is discussed more in details in [8] where they further explain the process. The takeaway from [8] is that they create ResNet, a network built from "residual blocks" as a solution to this issue. A visual representation of a residual block can be seen in Figure 2. Consider some record x. We pass it as an input to a small group of n layers with their own activators. The result of the layer output we can call y. Finally we concatenate x and y, using addition. This addition operation helps push non-zero values through the forward propagation process which helps keep input data to each block new and non-zero. This also helps prevent vanishing gradients during the back propagation process. We use this residual block for our deep connected network. Our residual blocks only use fully connected layers with post-activation concatenation for the classification of prompt gamma events. The fully connected residual blocks allows us to create a thin yet super deep fully connected neural network which avoids problems, mentioned above.

V. Hardware and Software

We used the Graphic Processing Unit (GPU) clusters in the taki system in the UMBC High Performance Computing Facility (www.hpcf.umbc.edu) for our hyperparameter studies. For the studies that have 256 neurons or 256 layers we use the gpu2018 partition. This 1 GPU node has four NVIDIA Tesla V100 GPUs (5120 computational cores over 84 SMs, 16 GB onboard memory) connected by NVLink, two 18-core Intel Skylake CPUs, and 384 GB of memory (12×32 GB DDR4 at 2666 MT/s). For all other studies that has fewer than 256 neurons or layers, we use the gpu2013 partition which has 18 hybrid CPU/GPU nodes, each with two NVIDIA K20 GPUs (2496 computational cores over 13 SMs, 4 GB onboard memory), two 8-core Intel E5-2650v2 Ivy Bridge CPUs (2.6 GHz clock speed, 20 MB L3 cache, 4 memory channels), and 64 GB of memory (8×8 GB DDR3). For the gpu2018 we use 1 GPU per job, a time limit of 4 to 5 hours, and 35 GB of memory. For the gpu2013 we use 2 GPUs per job, a time limit of 4 to 16 hours, and MaxMemPerNode memory. the neural network backbone [9]. The training was done using Keras' Model.fit method on two NVIDIA K20 GPUs

This network was built using Tensorflow v2.4.0 (www.tensorflow.org) with the bundled Keras module8. We also used scikit-learn v0.22.1 (https://scikit-learn.org/stable/) to preprocess and normalize the data. Moreover pandas v1.0.4 (https://pandas.pydata.org/) and numpy v1.19.5 (www.numpy.org) were also used to help preprocess the data. Finaly we used the matplotlib v3.2.1 (www.matplotlib.org) library to graph our results.

VI. Results

For our studies, we trained the neural network on a data set that was generated using a Monte Carlo simulation and that consisted of 1,821,255 records and 15 features. These features represent spatial coordinates, Euclidean distance, and energy deposition for each interaction. An interaction is a grouping of three spatial coordinates and an energy level. Each row is either a triple, double-to-triple, or a false triple and consists of three interactions each. Our training data set only consisted of True Triples, Double-to-Triple scatter, and False events. Furthermore, when testing the neural network we

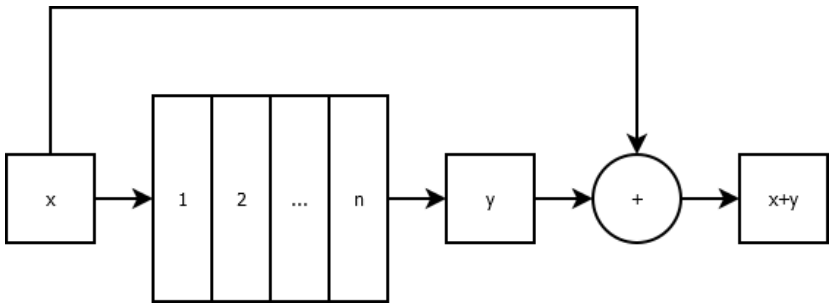

Fig. 2. Our fully connected residual block takes an input and passes it through n layers eventually adding it to the output of the n layers

used datasets that used 150MeV (Mega electron Volt) beams with three different dosage rates: 20kMU (kilo Monitor Unit), 100kMU, and 180kMU. The larger kMU values correspond to more intense dosage rates.

A. Baseline Hyperparameter Studies

A total of 288 studies were run based on a combination of hyperparameters. In each of our studies, the validation rate, the number of epochs, and the residual block size were held constant at 0.2, 1024, and 8 respectively, to act as a control. The normalization method used also stayed consistent throughout all 288 studies. The energy values were normalized using a power transformer (Yeo-Johsnon), while the spatial coordinates were normalized using `MaxAbsScaler` from the sklearn library. Other hyperparameters such as dropout rate, number of neurons per layer, number of layers, and batch size were changed for each study. The values used for the changing variables are as follows:

- Drop out rate: 0, 0.1, 0.4
- Number of neurons per layer: 32, 64, 128, 256
- Number of layers: 8, 16, 32, 64, 128, 256
- Batch size: 1024, 2048, 4096, 8192

Thus, the total number of studies is $(3)(6)(4)(4) = 288$.

After all 288 studies were complete, each study that reported a peak validation accuracy greater than 0.76 was considered to be promising. We report here on one promising study in detail; more studies are detailed in [1]. This subsection contains a *hyperparameters table*, a *training and validation plot*, and three *confusion matrices* of the MCDE model test1 at 20kMU, 100kMU, and 180kMU dosage rates. The *hyperparameter table* in Table I lists validation rate, dropout rate, neurons per layers, number of layers, batch size, and epochs.

In the *training and validation plot* in Table 3, the accuracy, denoted as a decimal percent, is plotted against the number of epochs. The peak validation accuracy is displayed at the top of each plot in addition to the total wall clock time the study took to run. The blue line represents the training accuracy and the orange line represents the validation accuracy. We can see that both the training and validation accuracies start at 40% at 0 epochs, then proceed to increase significantly in the first few epochs, where both accuracies increase to 60%. Then, the slope becomes flatter as the network struggles to learn more information about the data.

The three *confusion matrices* are provided in Figures 4, 5, 6. In these confusion matrices, the left most column is the correct input class and the percentage in the other columns represents the amount of data put into the class at the top of the column. These percentages sum to 100% in each row. Each cell is colored from white to dark green proportional to the maximum value in the entire matrix. The darkest entry in each row is the dominant classification of the input class. Within the matrices, there are thirteen different labels: 123, 132, 213, 231, 312, 321, 124, 214, 134, 314, 234, 324, and 444. Label 123 represents the case where all three interactions are correctly ordered and no adjustments need to be made. Label 132 represents the case where the third interaction should happen second and the second interaction should happen third. Label 213 represents the circumstance the second interaction should happen first, the first should occur second and the third is correctly labeled. Label 231 represents the scenario where the third interaction should be first, the second interaction should be third and the first interaction should be second. The label 312 represents the case where the first interaction should be third, the second interaction first whereas the last interaction should be second. The label 321 is the scenario where the first interaction should happen third, the second is labeled correctly, and the third interaction should happen first. The label 124 indicates that the first two interactions are a correctly ordered double and the third interaction should be disposed of. The label 214 indicates that the first two interactions are a incorrectly ordered double and the third interaction should be disposed of. The label 234 indicates that the last two interactions are a correctly ordered double and the first interaction should be disposed of. The label 324 indicates that the last two interactions are a incorrectly ordered double and the first interaction should be disposed of. The label 134 indicates that the first and third interactions are a correctly ordered double and the second interaction should be disposed of. The label 314 indicates that the first and third interactions are a incorrectly ordered double and the second interaction should be disposed of. The thirteenth label 444, is a false event, where the three interactions should be thrown away.

Figure 4 is a confusion matrix created from our fully connected network, whose parameters are displayed in Table I. It classifies the MCDE model test1 150MeV 20kMU beam. We can observe that the maximum classification in each class

TABLE I
HYPERPARAMETERS USED IN SECTION VI-A.

Validation Rate	0.2
Dropout Rate	0
Neurons per Layer	64
Layers	256
Batch Size	2048
Epochs	1024

is the input class itself in dark green. The accuracies for the true triples range from 66.3% to 74.6%., the double-to-triples' accuracies range from 62.2% to 71.5%, and the false events are $\approx 63\%$. Which means that when given any input class the will classify it the majority of the time. However there is still some discussion to be had about the dominant misclassifications of events. For some true triples the second highest classification percent is the double-to-triple version of itself. For instance, if we look at input class 132, the second highest at the 134 in the top column. This means that the network is able to classify correctly the first two interactions, but might have a harder time distinguishing the last interaction from a falsely coupled single. The other dominant misclassification case is where the second highest percentage is still a true triple but not correctly ordered. Finally we observe that the true triples are rarely classified as false events. For the double-to-triple data, we see that the second-highest ordering is the reverse ordered double-to-triple. This suggest that the network is able to correctly identify the interaction that does not belong in the event. However, it fails to order them once it takes out the single interaction that is not part of the pair. The third most likely misclassification of the doubles-to-triples are false events, which means that the network will drop just those events from the data. This is helpful for the reconstruction phase as it decrease noise in the data even if the data could have been used if correctly classified. There is still a large amount of doubles-to-triples that are still incorrectly labeled as true triples that will leads to noise in the reconstruction. For the false events we notice that second and third highest misclassification percentages are doubles-to-triples. When this happens we are unintentionally adding pure noise to the reconstruction algorithm leading to poorer results. Even though the percentages in Figures 4, 5 and 6 are different, the trends and conclusions drawn from the network's performance remain the same.

We have seen that the networks ability to classify events of all input classes is good enough to warrant further experimentation but not nearly as high as we suspect they could be. The percentages in Figure 4, 5, 6 are still 5% to 10% worse than the more complex networks seen in [2] for all input classes. The positive side is that while our network is worse it is significantly faster and more memory efficient since it uses fewer layers and neurons per layer than the more complex networks. Additionally neither our network nor more complex networks meet the 80% to 90% real-world use threshold.

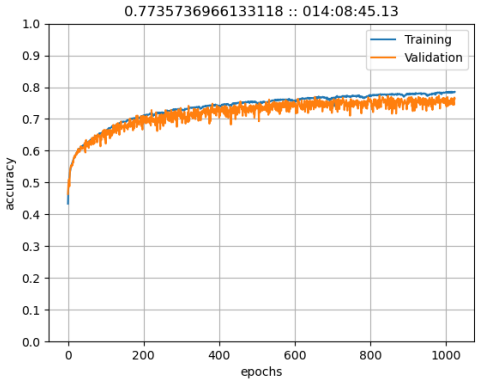

Fig. 3. Training and validation plot using a fully connected network in Section VI-A.

B. Reducing the Hyperparameter Space to Enable Long Running Studies

In this section we used the results from Section VI-A to pick a reduced set of hyperparameters for searching while using a larger number of epochs. The idea is that we pick our best performers and give them more time to learn and possibly plateau. This should give us additional insights into what configurations are optimal for learning with a more compact network.

A total of 12 studies were run based on a our reduced combination of hyperparameters. The validation rate and the residual block size as well as the normalization method used stayed consistent with Section VI-A, but we are training for 4096 epochs now. Our varied hyperparameters such as dropout rate, number of neurons per layer, number of layers, and batch size were reduced in scope changed for each study. The values used for the changing variables are as follows:

- Drop out rate: 0.1
- Number of neurons per layer: 64, 128
- Number of layers: 64, 128
- Batch size: 2048, 4096, 8192

Thus, the total number of studies is $(1)(2)(2)(3) = 12$. The most promising study is reported here in detail; more studies are detailed in [1].

Figure 7 is a training and validation plot for our fully connected network, whose parameters are displayed in Table II. As we can see at 0 epochs we have a validation accuracy of a little less than 40%, then we have a sudden spike to a little over 60% at around 100 epochs. After that, the network seems to learn steadily with a slight increase over 3000 epochs, as it drops more neurons.

Figure 8 is a confusion matrix created from our fully connected networkclassifying the MCDE model test1 150MeV 20kMU beam. We can observe that the maximum classification in each class is the input class itself in dark green. The accuracies for the true triples range from 70% to 78%, the doubles-to-triples' accuracies range from 62% to 75%, and the false events are $\approx 66\%$. Which means that when given

	123	132	213	231	312	321	124	214	134	314	234	324	444
123	66.3	8.1	2.1	3.4	3.0	2.7	8.1	0.6	0.1	0.1	3.5	1.4	0.6
132	3.8	71.2	2.6	2.2	2.9	3.1	0.3	0.1	7.8	0.8	1.4	3.3	0.5
213	3.3	3.5	70.6	3.4	2.0	2.9	1.1	6.7	4.4	1.2	0.3	0.0	0.6
231	1.5	3.2	4.8	71.1	3.0	5.4	0.1	0.4	1.6	2.5	4.8	1.1	0.3
312	2.7	2.7	2.3	2.7	74.6	4.0	3.0	1.5	0.9	5.1	0.1	0.2	0.3
321	2.5	3.3	2.8	2.2	5.9	72.1	1.4	2.8	0.0	0.3	0.7	5.6	0.3
124	3.5	0.4	0.6	0.1	3.2	2.1	71.5	9.3	0.9	0.4	0.4	1.7	5.8
214	0.8	0.3	4.5	0.3	2.3	3.3	13.6	66.8	0.5	1.2	0.8	0.5	5.0
134	0.4	4.0	3.7	2.5	0.4	0.1	1.0	0.6	71.3	8.8	1.8	0.5	5.0
314	0.1	0.8	2.1	5.1	6.8	0.4	0.3	1.4	8.9	66.5	0.6	0.9	6.1
234	2.6	2.4	0.3	7.6	0.1	1.5	0.8	1.1	1.3	0.7	62.2	13.8	5.7
324	1.3	4.6	0.2	0.8	0.2	8.0	1.1	0.6	0.9	0.6	8.9	67.6	5.2
444	0.6	0.3	0.9	0.6	0.6	0.3	6.3	6.6	4.1	5.0	8.5	6.3	59.9

Fig. 4. Confusion Matrix using a fully connected network trained on triples, double to triples, and false data from a 150MeV beam over 1024 epochs in Section VI-A. The testing data used is from the MCDE model test1 150MeV 20kMU beam.

	123	132	213	231	312	321	124	214	134	314	234	324	444
123	66.8	6.1	2.7	2.7	2.9	3.8	7.3	0.5	0.3	0.1	4.6	1.9	0.4
132	3.6	71.2	3.1	1.9	2.1	1.8	0.2	0.0	9.6	1.2	1.3	3.4	0.5
213	3.9	3.6	68.1	4.8	3.3	2.5	1.1	5.9	4.7	1.3	0.4	0.0	0.4
231	1.3	3.0	4.8	73.0	1.8	5.6	0.3	0.3	1.7	2.0	4.4	1.0	0.6
312	1.9	2.3	2.7	2.7	76.9	4.2	2.9	1.1	0.4	4.4	0.0	0.4	0.1
321	2.1	3.4	3.8	2.5	7.3	70.7	1.3	3.7	0.0	0.2	0.6	4.1	0.4
124	5.3	0.4	0.8	0.1	3.4	2.4	70.8	8.9	0.6	0.4	0.4	1.5	5.0
214	0.8	0.1	4.8	0.5	1.7	4.0	14.0	65.5	0.8	1.1	0.9	0.7	5.1
134	0.2	4.9	3.0	2.1	0.7	0.1	0.5	0.7	70.0	10.4	1.6	0.4	5.4
314	0.1	0.5	1.6	4.3	6.1	0.2	0.3	1.4	10.0	68.1	0.6	1.1	5.8
234	2.8	2.2	0.4	6.2	0.2	2.0	1.1	0.7	1.4	0.7	63.4	13.6	5.4
324	1.2	4.6	0.1	1.3	0.5	6.7	1.3	0.7	0.8	0.5	8.4	67.4	6.6
444	0.5	0.3	0.2	0.6	0.6	0.6	5.5	5.7	5.6	5.0	4.6	5.5	65.4

Fig. 5. Confusion Matrix using a fully connected network trained on triples, double to triples, and false data from a 150MeV beam over 1024 epochs in Section VI-A. The testing data used is from the MCDE model test1 150MeV 100kMU beam.

	123	132	213	231	312	321	124	214	134	314	234	324	444
123	69.7	7.5	2.9	1.2	2.4	1.7	7.9	0.2	0.2	0.2	3.4	2.2	0.5
132	4.6	72.1	2.4	1.2	2.2	2.6	0.0	0.2	7.9	0.7	1.4	3.6	1.0
213	2.9	1.4	73.3	4.6	3.1	0.7	1.9	6.0	3.6	1.4	0.2	0.2	0.5
231	1.4	3.1	4.1	72.3	3.4	5.3	0.2	0.0	1.2	2.9	5.1	1.0	0.0
312	2.9	3.1	1.0	2.2	74.9	3.9	3.1	1.4	1.0	5.1	0.0	0.2	1.2
321	2.9	3.6	2.7	1.0	4.6	72.3	1.7	5.5	0.0	0.2	0.7	4.3	0.5
124	4.4	0.5	0.3	0.2	3.6	1.9	71.0	9.5	1.0	0.6	0.6	0.7	5.8
214	1.0	0.1	4.2	0.7	2.1	3.2	14.0	67.0	0.5	1.4	0.7	0.3	5.0
134	0.4	5.2	2.7	2.0	1.3	0.0	0.7	0.6	72.4	8.0	1.5	0.7	4.4
314	0.2	0.6	1.4	4.2	6.4	0.5	0.4	1.3	9.1	68.5	0.3	0.7	6.4
234	2.8	2.1	0.4	5.6	0.3	1.4	0.6	0.4	1.5	0.6	65.1	12.8	6.4
324	1.4	4.3	0.1	1.0	0.3	6.8	2.1	0.3	0.6	0.6	9.7	66.2	6.4
444	0.5	1.0	0.3	0.6	0.3	0.7	5.8	5.7	5.1	5.7	6.3	4.4	63.5

Fig. 6. Confusion Matrix using a fully connected network trained on triples, double to triples, and false data from a 150MeV beam over 1024 epochs in Section VI-A. The testing data used is from the MCDE model test1 150MeV 180kMU beam.

TABLE II
HYPERPARAMETERS USED IN SECTION VI-B.

Validation Rate	0.2
Dropout Rate	0.1
Neurons per Layer	128
Layers	128
Batch Size	8192
Epochs	4096

any input class the will classify it the majority of the time. However there is still some discussion to be had about the dominant misclassifications of events. For the true triples, the matrix shows that the second-highest classification accuracies are either a true triple, or the corresponding double-to-triple event. This means that the network recognizes true events but struggles to reorder the events correctly; other times, it is able to identify the ordering of the first two interactions but improperly decouples the third interaction. Both these cases could cause pollution in the data used for reconstruction. We also observe that the true triples are barely classified as false events. For the doubles-to-triples, we can see that the second-highest percentage is another double-to-triple event. In most cases we see that it correctly determines the falsely coupled interaction but cannot determine the correct the order of the remaining pair. Even so there are some still some cases where doubles-to-triples are also classified as true triples. The presence of this extra interaction will cause noise during reconstruction. We observe that the false events are classified correctly most of the time but occasionally they are classified as doubles-to-triples. Any false data that leaks into the reconstruction data will always cause noise during reconstruction.

By choosing a study whose dropout rate was greater than 0 we fixed the discrepancy between validation and testing accuracy seen in Section VI-A. Even when only using 0.1 dropout, just a small amount of dropout has brought our confusion matrices' accuracies much closer to our validation accuracy seen in the training and validation plots. If we had picked something larger we may have not seen any favorable results within 4096 epochs. The triples now have comparable accuracy to a more complex network but the doubles-to-triples and false data are still 1% to 7% worse than the more complex network. This is an improvement over our previously trained networks which used no dropout rate. Our training time for these studies are high but we cannot locate bugs within our code or TensorFlow. The results are unaffected by this increase in training time. Currently this network configuration still does not have the classification performance for real-world use but is with more tuning could approach the lower bounds.

VII. CONCLUSIONS AND FUTURE WORK

Section VI contains the promising highlights of our attempts to create a more compact neural network than is seen in [2]. We define a "compact network" as a network with similar or superior performance but contains fewer total parameters and fewer neurons than the previous networks with similar function. The fewer neurons the network has, the more computationally cheap it is to use, and the faster the network can classify records. If a network is sufficiently cheap one may not even need a GPU for real-world use! Under this same idea we refer to the networks in [2] as "complex networks" because the networks contain significantly more neurons and parameters than the networks we are attempting to create here. To create a more compact network we conduct hyperparameter studies using the grid search method.

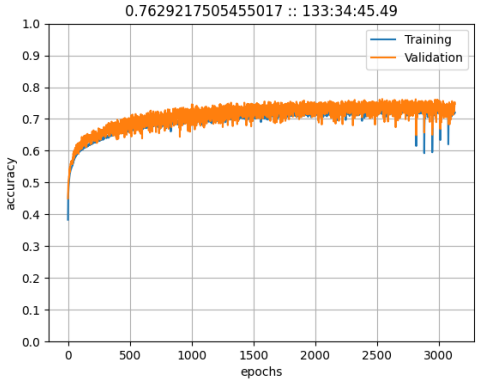

Fig. 7. Training and validation plot using a fully connected network in Section VI-B.

In Section VI-A we have seen that our networks' ability to classify events of all input classes is good enough to warrant further experimentation but there are still many problems that plague us. All of our promising studies that used a dropout rate of 0 had high peak validation accuracies but this performance did not carry over into the testing data. For example when we examine the peak validation in Figure 3 we have 77%. Yet, when we look at the confusion matrices we see that correct classification percentages fall into the mid 60's to low 70's. When we look at our worst results with the worst validation accuracy, not listed under our results in this work, we see that the common factor is a dropout rate greater than 0. This is not a huge surprise as dropout rate is known to cause lower initial accuracy in exchange for better network generalization when trained for more epochs. When using a dropout rate of 0 we cannot take the training and validation plots as the definitive proof of performance. Instead we would need to run hyperparameters studies which include dropout rates only greater than 0 while providing enough epochs for learning to potentially plateau. The first 288 studies yielded high performance during training and validation but substandard performance during testing. Even if the performance of these networks were, in general, not high enough to replace those in past works, we can still use their outcomes to guide our future hyperparameter decisions when trying to do more searches in the future.

In Section VI-B we attempted to demonstrate this by taking sets of hyperparameters that had the best performance and running them for longer with the hope that we may see improved testing performance. We also chose to only consider studies whose dropout rate was greater than 0 with the hope

	123	132	213	231	312	321	124	214	134	314	234	324	444
123	77.0	6.4	1.4	1.9	1.9	2.0	5.0	0.4	0.1	0.0	2.5	1.0	0.3
132	3.8	72.8	1.9	2.4	6.4	2.7	0.2	0.0	5.1	1.0	1.0	2.6	0.2
213	3.1	2.5	71.3	9.1	1.8	2.5	0.5	5.0	2.1	1.7	0.1	0.0	0.2
231	2.5	2.6	2.9	73.5	2.4	8.0	0.0	0.1	0.9	2.1	3.4	1.4	0.2
312	3.4	1.8	1.9	2.9	77.1	3.2	2.3	1.3	0.6	5.3	0.0	0.1	0.1
321	2.6	3.0	3.1	2.1	3.4	77.8	0.6	2.1	0.0	0.1	0.6	4.3	0.3
124	5.0	0.4	1.0	0.1	3.7	2.1	68.9	11.1	0.6	0.6	0.4	1.3	4.7
214	0.8	0.3	5.4	0.6	1.8	4.0	7.0	74.3	0.3	1.0	0.3	0.4	3.8
134	0.7	4.5	2.6	2.8	1.1	0.2	0.6	0.2	63.8	18.1	1.3	0.8	3.3
314	0.1	0.7	1.8	5.0	6.1	0.4	0.6	1.2	6.8	72.1	0.2	0.8	4.2
234	3.0	2.3	0.1	6.5	0.1	1.5	0.5	0.8	1.1	0.6	62.3	16.7	4.5
324	1.5	4.5	0.1	0.6	0.3	7.2	0.9	0.4	0.5	0.8	7.8	72.0	3.5
444	0.6	0.6	0.3	0.9	0.0	0.6	4.1	5.3	4.7	3.8	5.6	7.5	65.8

Fig. 8. Confusion Matrix using a fully connected network trained on triples, double to triples, and false data from a 150MeV beam over 4096 epochs in Section VI-B. The testing data used is from the MCDE model test1 150MeV 20K beam.

	123	132	213	231	312	321	124	214	134	314	234	324	444
123	77.0	7.2	1.8	1.7	1.5	1.7	3.9	0.8	0.0	0.0	2.5	1.6	0.2
132	3.5	73.1	1.5	2.2	6.3	1.5	0.0	0.0	6.5	1.3	0.9	2.7	0.3
213	4.3	1.9	66.6	12.5	2.3	1.9	0.7	5.0	2.3	1.9	0.2	0.1	0.1
231	1.8	2.6	2.5	76.5	1.7	8.5	0.1	0.0	1.1	1.5	2.5	1.1	0.1
312	2.9	1.3	2.8	2.8	78.0	3.8	2.0	1.0	0.4	4.8	0.0	0.1	0.1
321	2.6	3.2	3.6	2.2	3.6	76.7	0.8	3.1	0.0	0.2	0.3	3.5	0.3
124	6.7	0.3	0.6	0.0	3.2	2.1	71.0	9.8	0.7	0.4	0.4	1.0	3.7
214	1.0	0.1	5.6	0.6	1.6	4.6	7.2	73.0	0.5	0.9	0.3	0.3	4.4
134	0.3	5.5	2.8	2.2	0.8	0.0	0.5	0.5	65.3	16.3	1.3	0.2	4.2
314	0.1	0.7	1.6	4.2	5.6	0.3	0.2	0.7	7.3	73.7	0.4	0.9	4.5
234	3.9	2.4	0.5	5.2	0.2	1.9	0.4	0.5	0.8	0.2	61.0	18.1	4.7
324	1.3	2.8	0.1	1.0	0.5	7.1	0.6	0.5	0.6	0.5	7.1	72.4	5.5
444	0.8	0.3	0.3	0.8	0.6	0.8	4.1	4.7	5.5	5.0	4.1	4.4	68.6

Fig. 9. Confusion Matrix using a fully connected network trained on triples, double to triples, and false data from a 150MeV beam over 4096 epochs in Section VI-B. The testing data used is from the MCDE model test1 150MeV 100K beam.

	123	132	213	231	312	321	124	214	134	314	234	324	444
123	75.5	8.7	1.7	1.4	1.9	1.9	4.3	0.5	0.0	0.0	2.6	1.4	0.0
132	2.4	76.9	1.9	1.7	5.3	1.7	0.0	0.0	4.1	0.5	0.7	4.3	0.5
213	2.4	2.2	73.5	9.2	1.4	1.0	1.0	6.0	2.2	1.2	0.0	0.0	0.0
231	2.2	2.7	2.9	71.8	3.4	8.7	0.0	0.2	1.0	2.2	3.6	1.2	0.2
312	2.9	2.4	1.4	1.9	77.8	3.4	2.4	1.0	0.2	5.8	0.0	0.5	0.2
321	1.7	2.7	3.6	1.9	3.6	78.6	1.2	3.1	0.0	0.0	1.0	2.7	0.0
124	6.3	0.4	0.3	0.2	3.3	2.2	70.1	10.2	0.5	0.0	0.4	1.0	5.2
214	1.3	0.0	4.8	0.6	1.4	4.4	6.6	74.5	0.4	1.2	0.2	0.5	4.3
134	0.4	4.7	2.2	2.9	1.7	0.1	0.6	0.4	64.7	17.4	1.3	0.6	3.0
314	0.2	0.5	1.0	4.0	5.8	0.4	0.2	1.2	7.4	73.8	0.2	0.8	4.5
234	3.1	2.0	0.3	4.8	0.3	2.0	0.1	0.5	0.7	0.4	62.5	18.3	4.9
324	2.3	3.6	0.1	0.7	0.2	6.6	1.0	0.2	0.6	0.3	7.0	72.1	5.2
444	0.5	1.0	0.2	0.7	0.4	0.5	4.4	4.8	5.6	5.1	6.2	5.0	65.5

Fig. 10. Confusion Matrix using a fully connected network trained on triples, double to triples, and false data from a 150MeV beam over 4096 epochs in Section VI-B. The testing data used is from the MCDE model test1 150MeV 180K beam.

that this would help fix the discrepancy between validation and testing accuracy seen in the previous studies. In our extension studies we see that our best performing studies plateaued early in their training process. We see that the usage of even a small amount of dropout has brought our confusion matrices' accuracies much closer to our validation accuracy seen in the training and validation plots. The triples now have comparable accuracy to a more complex network but the doubles-to-triples and false data are still 1% to 7% worse than the more complex network. This is an improvement over our previously trained networks which used no dropout rate. Our training time for these studies was eerily high and we were not able to locate bugs or problems within TensorFlow; the results are unaffected by this strange increase in training time. The more compact networks show great promise in terms of achieving comparable accuracy to the best performing networks seen in [2]. Currently our networks still do not have the classification performance for clinical use. They also fall short in their classification accuracies, but if we can tackle the long training times then, with more hyperparameter tuning we can most likely create a network that is easier to train and cheaper to use than previous networks.

Particular studies, if given considerably more training time, could yield competitive, if not superior, testing accuracy to existing architectures while maintaining a simpler structure. More improvements are still needed for clinical use and we are currently experimenting with recurrent neural networks to test the viability of this type of architecture for this application.

ACKNOWLEDGMENTS

This work is supported by the grant "REU Site: Online Interdisciplinary Big Data Analytics in Science and Engineering" from the National Science Foundation (grant no. OAC–2050943). This work is 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. Co-author Carlos Barajas additionally acknowledges support as HPCF RA. 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.

REFERENCES

[1] Alina M. Ali, David Lashbrooke, Rodrigo Yepez-Lopez, Sokhna A. York, Carlos A. Barajas, Matthias K. Gobbert, and Jerimy C. Polf. Towards optimal configurations for deep fully connected neural networks to improve image reconstruction in proton radiotherapy. Technical Report HPCF–2021–12, UMBC High Performance Computing Facility, University of Maryland, Baltimore County, 2021.

[2] Carlos A. Barajas, Gerson C. Kroiz, Matthias K. Gobbert, and Jerimy C. Polf. Deep learning based classification methods of Compton camera based prompt gamma imaging for proton radiotherapy. Technical Report HPCF–2021–1, UMBC High Performance Computing Facility, University of Maryland, Baltimore County, 2021.

[3] Carlos A. Barajas, Gerson C. Kroiz, Matthias K. Gobbert, and Jerimy C. Polf. Using deep learning to enhance Compton camera based prompt gamma image reconstruction data for proton radiotherapy. *Proc. Appl. Math. Mech. (PAMM)*, in press (2021).

[4] Jonathan N. Basalyga, Carlos A. Barajas, Matthias K. Gobbert, Paul Maggi, and Jerimy Polf. Deep learning for classification of Compton camera data in the reconstruction of proton beams in cancer treatment. *Proc. Appl. Math. Mech. (PAMM)*, 20(1):e202000070, 2021.

[5] Jonathan N. Basalyga, Carlos A. Barajas, Gerson C. Kroiz, Matthias K. Gobbert, Paul Maggi, and Jerimy Polf. Improvements to the deep learning classification of Compton camera based prompt gamma imaging for proton radiotherapy. Technical Report HPCF–2020–29, UMBC High Performance Computing Facility, University of Maryland, Baltimore County, 2020.

[6] Jonathan N. Basalyga, Gerson C. Kroiz, Carlos A. Barajas, Matthias K. Gobbert, Paul Maggi, and Jerimy Polf. Use of deep learning to classify Compton camera based prompt gamma imaging for proton radiotherapy. Technical Report HPCF–2020–14, UMBC High Performance Computing Facility, University of Maryland, Baltimore County, 2020.

[7] François Chollet. *Deep Learning with Python*. Manning Publications Co., 2018.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[9] Gerson C. Kroiz, Carlos A. Barajas, Matthias K. Gobbert, and Jerimy C. Polf. Exploring deep learning to improve Compton camera based prompt gamma image reconstruction for proton radiotherapy. In *The 17th International Conference on Data Science (ICDATA'21)*, accepted (2021).

[10] Paul Maggi, Steve Peterson, Rajesh Panthi, Dennis 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.*, 65(12):125004, 2020.

[11] Enrique Muñoz, Ana Ros, Marina Borja-Lloret, John Barrio, Peter Dendooven, Josep F. Oliver, Ikechi Ozoemelam, Jorge Roser, and Gabriela Llosá. Proton range verification with MACACO II Compton camera enhanced by a neural network for event selection. *Sci. Rep.*, 11(1):9325, 2021.

[12] Jerimy C. Polf, Carlos A. Barajas, Gerson C. Kroiz, Stephen W. Peterson, Paul Maggi, Dennis S. Mackin, Sam Beddar, and Matthias K. Gobbert. A study of the clinical viability of a prototype Compton camera for prompt gamma imaging based proton beam range verification. In *AAPM Virtual 63rd Annual Meeting*, submitted (2021).

[13] Jerimy C. Polf and Katia Parodi. Imaging particle beams for cancer treatment. *Phys. Today*, 68(10):28–33, 2015.

[14] Andreas Zoglauer and Steven E. Boggs. Application of neural networks to the identification of the Compton interaction sequence in Compton imagers. In *IEEE Nuclear Science Symposium Conference Record*, 2007.