Multi-Layer Recurrent Neural Networks for the Classification of Compton Camera Based Imaging Data for Proton Beam Cancer Treatment

Joseph Clark
Dept. of Mathematics and Computing
Lander University

Anaise Gaillard
Dept. of Computational and Data Sciences
George Mason University

Justin Koe
Dept. of Electrical Engineering
The Cooper Union

Nithya Navarathna
Dept. of Biological Sciences
U. of Maryland, Baltimore County

Daniel J. Kelly
Dept. of Mathematics and Statistics
U. of Maryland, Baltimore County

Matthias K. Gobbert
Dept. of Mathematics and Statistics
U. of Maryland, Baltimore County

Carlos A. Barajas
Dept. of Mathematics and Statistics
U. of Maryland, Baltimore County

Jerimy C. Polf
Dept. of Radiation Oncology
U. of Maryland School of Medicine

***Abstract*—Proton beam 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 the proton beam can make this form of therapy more precise and safer for the patient during delivery. 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. Unfortunately, some of the Compton camera data is flawed and the reconstruction algorithm yields noisy and insufficiently detailed images to evaluate the proton delivery for the patient. Previous work used a deep residual fully connected neural network. The use of recurrent neural networks (RNNs) has been proposed, since they use recurrence relationships to make potentially better predictions. In this work, RNN architectures using two different recurrent layers are tested, the LSTM and the GRU. Although the deep residual fully connected neural network achieves over 75% testing accuracy and our models achieve only over 73% testing accuracy, the simplicity of our RNN models containing only 6 hidden layers as opposed to 512 is a significant advantage. Importantly in a clinical setting, the time to load the model from disk is significantly faster, potentially enabling the use of Compton camera image reconstruction in real-time during patient treatment.**

***Index Terms*—Proton Beam therapy, Compton camera, Image reconstruction, Deep residual neural network, Recurrent neural network**

I. Introduction

Because to its many advantages, proton beam therapy has gained popularity as a form of cancer treatment. Most types of radiation therapies work with the objective to damage the cellular DNA of target cancer cells that reside in the nucleus of every cell. X-ray therapy is able to deliver dosage at the tumor site, but its 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 beams have a finite range that can be controlled and they deposit the majority of their energy just before they stop. This sharp energy increase of the proton beam right before stopping is known as the Bragg peak. Since almost no radiation is delivered beyond the Bragg peak, healthy tissue can be spared from unnecessary radiation [1]. In order to take full advantage of these properties of proton therapy, we must have an efficient technique to image the prompt gamma rays produced by the beam in real-time as they travel through the patient's body. A Compton camera can be used to detect the prompt gamma rays emitted when the proton beam travels through the body, and an algorithm is available to reconstruct the beam's image from the prompt gamma data, which then provides an indirect image of the proton beam. Unfortunately, a lot of the raw data of the Compton camera is flawed and the reconstruction algorithm yields noisy and insufficiently detailed images to evaluate the proton delivery for the patient [2], [3].

Machine learning can be used to clean the raw Compton camera data by identifying and removing false data before image reconstruction [2], [3]. Research efforts to clean the Compton camera data have led to the use of neural networks. Shallow networks like the one in [2] use 1 to 2 hidden layers to perform simple classifications of simulated prompt gamma data under ideal conditions that do not represent the irradiation conditions encountered during clinical proton beam radiotherapy. This *shallow* network in [2] is a binary classification network that simply determines which event data are true events and should be used for reconstruction and which are false events that should not be used for reconstruction. This is improved upon in [3] using the *deep* residual fully connected neural network described in [4] for triple event classification. This neural network consists of 64 residual blocks with 8 fully

connected layers per block yielding a total of 512 hidden layers. Each layer had 256 neurons, a 45% dropout rate, and used leaky ReLU activation. More detailed results and discussions about the impact of neural network processing on the use and viability of Compton camera based imaging in clinical proton radiotherapy are the focus of [3], while providing details on the network and its training are the focus of [4]. The full capabilities of the described neural network are specified in [5], where preprocessing the data, all classification capabilities, and postprocessing output data are described in detail. Other recent work [6], [7] focused on hyperparameter studies on the deep residual fully connected neural network from [4], varying batch sizes, neurons, and layers. The use of recurrent neural networks (RNNs) is proposed in [6], since they use recurrence relationships in sequence data sets to make potentially better predictions. The potential for RNNs to be an improvement over feedforward neural networks (FNNs) is shown in [8].

In this work, we test RNN architectures using two different recurrent layers because of their potential for classifying sequence data, the Long Short-term Memory (LSTM) (discussed in Section III-A1) and the Gated Recurrent Unit (GRU) (discussed in Section III-A2). The LSTM uses memory cells with gates and a carry track to encode and learn from sequence data. The GRU uses two gating units to encode and learn from sequence data. The goal in this change in type of network architecture is to examine data as a sequence of interactions rather than one single event, but preliminary results do not show any benefit. We use models with 4 GRU layers and with 4 LSTM layers and achieve similar testing accuracy as the deep residual fully connected model from [4]. The model with 4 GRU layers outperforms the deep residual fully connected model in 3 classes but has a larger gap (within 10%) in accuracy in the other 10 classes. The model with 4 LSTM layer outperforms the previous deep residual fully connected model in only 2 classes but has a smaller gap (within 6%) in accuracy in the other 11 classes. Although the deep residual fully connected model achieves a slightly higher accuracy in nearly every class, the simplicity of our RNN models containing only 6 hidden layers (4 recurrent and 2 fully connected) as opposed to 512 is an advantage. And importantly in a clinical setting, the time to load the model from disk is significantly faster, potentially enabling the use of Compton camera image reconstruction in real-time during patient treatment.

The remainder of this work is organized as follows: Section II provides background on proton beam therapy to treat cancer and the use of a Compton camera to image promp gammas. Section III details the basics of machine learning and recurrent neural networks, while also providing details on the LSTM and GRU. Section IV outlines the hardware and software we use to carry out this research project. Section V contain the results of our work. Section VI contains our conclusions and future work.

II. Application Background

A. Proton Beam Therapy

Radiation therapy is a form of cancer treatment that uses high doses of radiation to kill cancer cells. X-ray therapy, a form of radiation therapy, is a common technique used for cancer treatment, where 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, which is another form of radiation therapy, 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.

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 slightly 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 and an optimal path could be used. 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.

B. Compton Camera

The Compton camera is a multi-stage detector that produces data used to generate images of proton beams used in proton beam therapy [4]. As protons from the beam enter the body, they interact with cells in the body causing the emission of prompt gamma rays. Some of these gamma rays will collide with the Compton camera. An interaction is when a prompt gamma collides with a stage of the Compton camera. For each interaction, 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 $i = 1, 2, 3$ for the three stages of the Compton camera, 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 scan 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 [1]. 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.

Prompt gammas are emitted at speeds close to the speed of light consequently the camera is unable to detect the true ordering of interactions in an event. The false events cause noise in the image created impacting the usefulness of the image [4]. Next we describe the three different type of Compton camera scatters.

a) True Triples: In the True Triples event, the Compton camera will detect the path of a single prompt gamma occurring in some order. However, it is possible that the true path is some other ordering. There are a total of 6 total combinations of True Triple scatters: 123, 132, 213, 231, 312, 321 and, as the data stands, only the 123 ordering is usable.

b) Double-to-Triples (DtoT): In the DtoT event, the Compton camera will detect the path of a single prompt gamma as a true triple. However, in reality, there were two prompt gammas who had varying paths. One prompt gamma could have detected as the first and third interaction and the second prompt gamma could have been mistaken as the second interaction. Similar to true triples, there are a total of 6 misdetection orderings: 124, 134, 214, 234, 324, 314. The second prompt gamma interaction is classified as "4" in the misdetection orderings. In this case, without processing the data, all 6 orderings are unusable.

c) False Triples: In a false triples event, the Compton camera will detect a true triple whereas in reality, there were actually three different prompt gammas. As a result, this entire event provides no insight into the path of a single prompt gamma and must be discarded.

d) The Need for Machine Learning: In order to make proton beam therapy more effective, real-time imaging is needed to verify location and effectiveness of the proton beam, in particular the location of the Bragg peak. Machine learning is capable of classifying which type of scatter event occurred based upon data provided by the Compton camera. These classifications lead to removal of unusable data which will clean the resulting image. A clearer image allows for treatment verification. A sufficiently accurate machine learning model could produce an image that is clear enough to be used in proton beam therapy as a form of treatment verification. A machine learning algorithm will need approximately 90% testing accuracy to be useful for clinicians.

In current practice, the patient's body is imaged before undergoing treatment in order to map the position of the tumor. A plan for how to target and treat the tumor with the proton beam is then developed. The course of proton beam radiation therapy itself then follows, and consists of the delivery of the planned treatment in multiple treatment sessions over a period of one to five weeks. Machine learning models would be used to greatly improve the reconstructed images of the delivered proton beam in real-time. The model is loaded as part of the beam imaging software at the start of the day by the operator and is then used to clean the Compton camera data prior to reconstruction of the beam image for each patient during treatment.

Additional details on the application are provided in the report [9].

III. Machine Learning

Machine learning is a type of artificial intelligence where a machine is trained to identify specific trends and patterns to make predictions from data. In the case of Compton camera data, the machine learning algorithm will try to predict the appropriate class for a scatter event. The main benefit of machine learning is its efficiency in producing results that would take humans alone much longer. There are four different ways that a machine can be taught: supervised, unsupervised, semi-supervised, and reinforcement. Supervised learning is a form of learning where the machine is provided a labeled data set that has regular input data as well as the desired output data. This allows the machine to produce a model that has been fitted appropriately. Unsupervised learning is used when one wants to identify hidden patterns within an unlabeled data set. This allows the machine to identify any trends it finds in the data without special instruction. Semi-supervised learning is a mixture of supervised and unsupervised where the model is provided some labeled data and a large amount of unlabeled data. Reinforcement learning is similar to the way humans learn where the machine will interact with the data and there will be either a positive or negative reward depending on whether the machine does something the programmer wants or not. The method used in this study is supervised learning because the data set contains both the data from the scatter event and the corresponding label of which event scatter took place.

A. Recurrent Neural Networks

Recurrent neural networks (RNNs) are an efficient neural network used for time series tasks. They work similar to a coupling process in biology. They rely on information from the previous system or "loop" to move forward with the next. In this type of neural networks, the sequence or order of the network is very important. The system can be read and executed differently if the elements of both series are in different orders. In the case of RNNs, elements include an input layer, hidden layers, and an output layer.

RNNs use back-propagation through time to illustrate gradients. The difference between RNN back propagation and other methods such as in a feed forward network is that sum errors are necessary at each time step because of the shared parameters throughout the network. There are several types of RNNs that are distinguished by the pathways between inputs

and outputs. RNNs may also contain activation functions that allow a neuron to translate the input into a specific output. Finally, there are a few RNN structures that vary depending on the desired use. There are bidirectional recurrent neural networks, long short-term memory, and gated recurrent units. Bidirectional recurrent networks rely on future data to generate predictions.

RNNs are a viable option for Compton camera data because of their ability to encode information about previous events. Shaping an event in the Compton camera as a sequence of three interactions each with five features, we have transformed the data produced by the Compton camera to a sequence. Using the sequence of interactions the RNN will be able to predict the ordering of interactions, i.e., the appropriate scatter.

1) Long Short-Term Memory: A Long Short-Term Memory (LSTM) neural network is a type of RNN that is traditionally used for natural language processing and time series forecasting. The unique aspect of LSTM is that it contains a memory cell. This memory cell is used to store certain pieces of information that may be needed later in the training process, called a state. The memory cell has three gates to determined the state: forget gate, input gate, and output gate. The forget gate controls what stored information can be forgotten. The input gate controls what information should be used to change the state of the memory cell, and the output gate controls which part of that information is needed at a given time. As stated previously, RNNs use the output of one step and carry it over into the next step in addition to the new data input. The different gates classify the needed and unneeded information, and the new state is outputted for the next step. The memory cell was added to combat the main issue with RNNs which is long-term dependency where as more and more information is fed into the RNN, it becomes less effective in learning because the network cannot remember everything.

2) Gated Recurrent Unit: A Gated Recurrent Unit is essentially a streamlined version of the LSTM in Section III-A1. The GRU has gating units that modulate the flow of information inside of the unit. The GRU factors in the previous short-term dependency with a reset gate by using a linear interpolation between the previous activation function value and the current one. The GRU also factors in previous long-term dependencies with an update gate by taking a linear sum between existing state and the newly computed state. Unlike the LSTM, the GRU does not have separate memory cells.

IV. Hardware and Software

We used the GPU cluster ada in the UMBC High Performance Computing Facility. The ada system has 3 distinct node types. Four nodes each with 8 Nvidia RTX 2080 Ti GPUs each with 11GB GPU memory. Seven nodes with 8 Nvidia Quadro RTX 6000 GPUs each with 24GB of GPU memory. Two nodes each with 8x Nvidia Quadro RTX 8000 GPUs each with 48GB memory. Each node has 384 GB of CPU memory (12×32 GB DDR4 at 2933 MT/s) except the two RTX 8000 nodes which have 768GB of CPU memory(12×64GB DDR4 at 2933 MT/s).

Networks built on ada were built with the software package Anaconda3 and Tensorflow v2.6.0 with the bundled Keras module.

V. 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 1,443,993 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 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. Both the training and testing datasets were reshaped to be sequentially read. Therefore each record of 15 features was reshaped to 3 interactions of 5 features each: three spatial coordinates, Euclidean distance, and energy deposition. Each record is fed into the neural network as a sequence of 3 interactions. The testing data contains 37,151 testing data points for 20kMU/min, 17,425 for 100kMU/min, and 12,254 for 180kMU/min from MCDE model test 1 150MeV.

Previous research explored fully connected networks in depth. We explore recurrent neural networks using the LSTM and GRU layers. We begin by examining the number of epochs, the batch size, and the learning rate. We then explore the number of layers and number of neurons in order to determine a promising configuration for a recurrent neural network. RNNs with both GRU and LSTM models are examined. These studies lead us to the use of 4 recurrent layers of 128 neurons with a batch size of 2048 and learning rate of 10^{-3}. Table I shows the constant parameters for all RNN studies.

TABLE I: Constant RNN parameters

Hyperparameter	Value
Recurrent Layer Activation	Tanh
Final activation	Softmax
Output Layer	13 Neurons
Optimizer	Nadam
Loss Function	Categorical Crossentropy
Train,Validation Split	0.8/0.2

Our goal is to discover how long these models could be trained before plateauing. Testing models with more epochs than 512 showed an increase in validation accuracy. However, after 1024 epochs the model learns very slowly. A 4 GRU layer model with 1024 epochs has a validation accuracy of 71% and the same model with 8192 epochs has a validation accuracy of 77%.

A learning rate scheduler is used to change the learning rate during model training. One possible learning rate schedule is a step schedule which changes the learning rate at certain

epochs. This can be done using a Keras callback which will adjust the learning rate during training. A piece-wise function such as the one in Equation (1) can represent a step learning schedule. We use i to represent the current epoch and p to represent the total number of epochs, then L is a function of epochs and determines our learning rate at the i^{th} epoch. Our initial studies using a step learning rate schedule showed that a learning rate schedule could make approximately a 6% to 7% increase in accuracy while preventing overfitting.

The impact of the learning rate schedule on accuracy and generalization lead us to study a 32,000 epoch model with the learning rate schedule that also has 2 dense layers of 128 and 64 neurons respectively. We test these parameters on both the models with LSTM and the models with GRU layers. The final models will then be tested using a new data set and will have confusion matrices made to verify their accuracy. A confusion matrix contains all 13 misdetection orderings as well as a percentage that is determined by how frequently the model classified each event correctly or incorrectly. The main diagonal of a given matrix shows the percentage of correct classifications of the network. All other entries in the matrix are percentages where the network incorrectly classified an event.

A. 32,000 Epoch Network with Learning Rate Schedule

The 4 LSTM layer model has a very high training and validation accuracy. The dense layers for the LSTM model have the ReLU activation function for both layers. This causes us to believe the model could be a possible improvement over the previous model. We notice the rise in accuracy at the epochs where the learning rate is lowered. We also notice how training and validation accuracy converge at the end with the lowering learning rates. The model has a final validation accuracy of 89% which is a significant improvement over all previous studies. The model however is overfit. There is a significant difference in the validation accuracy and the testing accuracy.

The dense layers for the GRU model have the leaky ReLU activation function for both layers the parameters are the same for the GRU. The GRU layers produce a model with a slightly lower final validation accuracy of 86%. However the model performs different on the test data. Ultimately, this model is still overfit with the large difference in validation and testing accuracy. The overfitting of these models tells us that we should try and apply some regularization to them such as dropout layers in order to make the model more general. Regularization techniques will help bring the testing accuracy closer to the validation accuracy.

B. Regularization

We adjusted the number of epochs to be 16384 and added dropout layers between every hidden layer with a dropout rate of 20%. The model used Equation (1) as the learning rate schedule.

$$L(i) = \begin{cases} 10^{-3} & i \leq \frac{p}{8} \\ 10^{-4} & \frac{p}{8} < i \leq \frac{p}{4} \\ 10^{-5} & \frac{p}{4} < i \leq \frac{p}{2} \\ 10^{-6} & \frac{p}{2} < i \end{cases} \tag{1}$$

This helps regularize the model. While drastically reducing the model's validation accuracy to below 80%; the validation and testing accuracy are much closer. This model performs almost as well as the deep residual fully connected model for both the LSTM and the GRU. While the model is almost as accurate as the deep residual fully connected model. Its load time is 10s for the GRU model and 7s for the LSTM model which is an advantage. The fully connected model loads in 47s. Also having still only 4 recurrent layers and 2 dense layers is an advantage because there is a great deal more than can be done with such a simple model. Table III through Table VIII show the confusion matrices for the regularized GRU and LSTM models.

TABLE II: Comparison of GRU model with deep fully connected network.

Class	GRU	DRFCN	GRU - DRFCN
123	76.4	79.1	-2.7
132	79.4	76.0	3.4
213	73.5	76.4	-2.9
231	79.1	80.7	-1.6
312	83.1	82.4	0.7
321	76.2	76.5	-0.3
124	71.9	76.0	-4.1
214	74.0	75.0	-1
134	72.0	75.4	-3.4
314	78.3	75.4	2.9
234	63.9	73.6	-9.7
324	73.9	75.3	-1.4
444	63.5	72.6	-9.1

Comparing the GRU and LSTM models with the deep residual fully connected (DRFCN) model from [5] in each classification at the dosage rate of 100kMU/min, we see in Table II that the GRU model outperforms the deep fully connected model in three categories and is within 10% in all categories. Similarly, the LSTM only outperforms in two categories but is within 6% as shown in Table IX.

Another test was run on the 4 layer LSTM with 2 dense layers model except the number of neurons per dense layer was increased from 128 and 64 to 256 and 128. The dropout rate remained at 0.2. Table X shows the comparison results. This model is within 5% within every classification. Finally in Table XI we see a comparison of overall accuracy and the load times.

TABLE III: Confusion matrix for 4 GRU layer model with learning rate schedule Equation (1) trained on triples, double to triples, and false data from a 150MeV over 16384 epochs. 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	73.8	5.4	1.7	3.5	4.2	2.0	5.6	0.5	0.2	0.1	1.6	1.1	0.2
132	2.5	79.4	1.9	1.6	3.3	2.6	0.1	0.0	5.0	0.8	0.4	2.2	0.2
213	1.5	3.4	75.7	3.5	2.5	2.9	0.6	4.7	3.1	1.8	0.1	0.0	0.2
231	2.0	2.1	3.4	77.3	4.0	2.2	0.0	0.1	0.7	3.5	3.7	0.7	0.2
312	1.6	2.0	1.2	1.7	81.1	2.7	2.1	0.9	0.5	6.0	0.0	0.1	0.0
321	1.2	2.7	2.1	2.3	4.3	78.7	0.7	2.1	0.0	0.4	0.4	4.9	0.2
124	4.2	0.4	0.8	0.1	5.2	2.3	70.6	9.0	0.9	1.0	0.3	0.9	4.2
214	0.4	0.3	5.1	0.3	2.3	4.4	5.7	74.6	0.3	2.1	0.3	0.4	3.7
134	0.6	5.8	3.0	1.7	0.8	0.1	0.4	0.3	72.7	10.5	0.5	0.8	2.7
314	0.0	1.5	1.7	4.1	6.0	0.4	0.5	0.9	6.8	74.4	0.1	0.9	2.6
234	3.6	2.8	0.1	8.2	0.4	1.0	0.1	1.1	1.8	1.0	65.7	9.4	4.8
324	1.3	5.7	0.1	0.4	0.4	7.3	0.9	0.3	0.6	1.2	4.3	74.4	3.2
444	0.9	2.2	0.3	0.9	0.0	0.0	5.6	6.3	5.6	7.5	4.4	7.2	58.9

TABLE IV: Confusion matrix for 4 GRU layer model with learning rate schedule Equation (1) trained on triples, double to triples, and false data from a 150MeV over 16384 epochs. 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	76.4	4.4	1.8	3.1	3.5	1.7	4.5	0.8	0.2	0.0	2.0	1.4	0.1
132	2.0	79.4	1.5	1.2	3.5	2.0	0.0	0.0	6.8	0.6	0.6	1.9	0.3
213	2.0	3.7	73.5	3.7	3.2	1.9	0.5	4.9	3.6	2.7	0.2	0.0	0.1
231	1.9	2.2	3.1	79.1	3.1	1.9	0.0	0.0	1.2	3.4	3.4	0.6	0.1
312	1.1	1.7	1.5	2.1	83.1	2.7	1.7	0.6	0.3	4.7	0.0	0.2	0.1
321	0.8	3.3	2.7	2.8	5.6	76.2	0.6	3.2	0.0	0.6	0.3	3.8	0.2
124	5.9	0.5	0.5	0.1	5.3	1.6	71.9	7.7	0.8	0.7	0.2	1.3	3.4
214	0.7	0.2	5.8	0.4	2.5	4.2	5.6	74.0	0.5	2.1	0.2	0.5	3.4
134	0.3	6.5	2.8	1.7	0.7	0.0	0.4	0.5	72.0	11.0	0.6	0.4	3.2
314	0.1	0.5	1.6	2.9	5.8	0.2	0.1	0.7	5.9	78.3	0.1	1.1	2.8
234	4.7	3.1	0.5	8.0	0.2	0.9	0.4	0.5	1.1	0.8	63.9	10.3	5.4
324	1.2	4.7	0.2	0.9	0.4	6.1	0.7	0.3	0.8	1.3	5.4	73.9	4.1
444	1.0	0.9	0.4	0.6	0.9	1.0	4.2	4.6	6.4	8.3	3.0	5.2	63.5

TABLE V: Confusion matrix for 4 GRU layer model with learning rate schedule Equation (1) trained on triples, double to triples, and false data from a 150MeV over 16384 epochs. The testing data used is from the MCDE model test1 150MeV 180K beam.

	123	132	213	231	312	321	124	214	134	314	234	324	444
123	72.1	7.2	2.2	2.4	4.3	2.6	6.2	0.5	0.0	0.0	1.7	0.7	0.0
132	1.7	81.0	1.2	1.2	1.2	2.2	0.2	0.0	5.5	1.0	0.7	3.6	0.5
213	1.0	3.1	75.4	4.1	2.4	2.9	0.2	6.5	3.4	1.0	0.0	0.0	0.0
231	1.4	1.7	4.3	75.4	5.3	1.9	0.2	0.2	0.7	4.1	3.6	0.7	0.2
312	1.0	3.4	0.7	1.7	80.7	2.9	1.9	0.7	0.5	5.5	0.0	0.2	0.7
321	1.7	1.7	2.4	2.7	4.8	78.1	1.4	3.4	0.0	0.2	0.2	3.4	0.0
124	6.0	0.8	0.6	0.0	4.7	1.9	70.8	8.2	0.5	0.4	0.2	1.1	4.9
214	0.8	0.0	5.1	0.6	2.9	4.4	5.2	74.0	0.5	1.5	0.2	0.6	4.1
134	0.6	6.4	3.3	1.7	1.4	0.0	0.3	0.2	71.6	10.7	0.9	0.5	2.3
314	0.1	0.2	0.6	3.7	6.3	0.3	0.1	1.0	6.4	77.4	0.1	0.6	3.3
234	3.7	2.5	0.6	7.6	0.3	1.1	0.3	0.5	1.6	0.6	67.0	8.7	5.4
324	1.4	4.8	0.1	0.8	0.6	6.6	1.4	0.2	0.9	1.0	6.0	72.1	4.2
444	0.8	1.2	0.3	0.7	0.6	0.4	4.9	4.4	6.5	7.7	4.0	5.9	62.6

TABLE VI: Confusion matrix for 4 LSTM layer model with learning rate schedule Equation (1) trained on triples, double to triples, and false data from a 150MeV over 16384 epochs. 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	78.2	3.6	1.3	2.4	2.2	1.8	7.2	0.4	0.1	0.0	1.8	0.8	0.2
132	4.3	74.9	2.3	1.8	2.7	2.9	0.4	0.0	6.6	0.7	0.7	2.3	0.3
213	2.2	3.1	75.0	3.0	1.9	2.5	0.8	6.3	3.6	1.3	0.1	0.0	0.3
231	3.6	2.3	3.8	73.3	3.4	2.3	0.1	0.2	1.3	3.6	5.3	0.6	0.3
312	3.4	2.2	1.5	2.3	74.2	3.2	4.0	1.2	0.7	6.9	0.0	0.1	0.2
321	2.2	2.5	2.5	1.9	3.6	76.5	1.6	3.1	0.0	0.2	0.5	5.0	0.4
124	4.4	0.4	0.6	0.2	2.8	1.5	75.9	7.3	0.7	0.7	0.2	0.9	4.4
214	0.8	0.3	4.7	0.1	1.1	3.3	8.9	74.7	0.5	1.2	0.3	0.3	4.0
134	0.6	4.7	2.8	1.6	0.5	0.3	0.8	0.5	75.3	8.4	0.9	0.4	3.2
314	0.0	1.0	2.1	3.3	5.0	0.4	0.7	1.1	8.3	72.8	0.2	1.0	4.0
234	4.9	2.1	0.2	6.4	0.1	0.6	0.5	1.2	1.8	0.8	67.8	8.0	5.7
324	1.8	5.4	0.1	0.4	0.3	6.7	1.8	0.4	0.6	1.0	7.1	70.6	3.9
444	1.3	1.9	0.3	0.9	0.0	0.0	7.2	6.0	4.1	5.3	5.0	6.0	62.1

TABLE VII: Confusion matrix for 4 LSTM layer model with learning rate schedule Equation (1) trained on triples, double to triples, and false data from a 150MeV over 16384 epochs. 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	80.0	3.8	1.6	2.5	1.3	1.3	5.9	0.6	0.1	0.0	1.9	0.8	0.1
132	3.4	75.8	2.2	1.2	3.1	2.5	0.0	0.0	8.1	0.7	0.6	2.1	0.3
213	2.3	2.8	72.5	3.9	3.0	1.8	1.0	6.2	4.1	1.7	0.2	0.1	0.3
231	3.1	1.5	4.4	75.9	2.7	2.6	0.1	0.2	1.7	3.0	4.2	0.5	0.2
312	2.6	2.0	2.2	2.5	76.1	3.9	2.7	1.0	0.6	6.1	0.0	0.1	0.2
321	1.5	3.1	3.3	2.8	4.7	73.5	1.9	3.9	0.0	0.3	0.2	4.5	0.3
124	5.8	0.4	0.3	0.1	2.8	1.0	77.1	6.7	0.5	0.7	0.3	0.8	3.4
214	1.0	0.1	5.2	0.4	1.7	3.3	8.2	74.2	0.5	1.0	0.1	0.4	3.9
134	0.4	4.9	2.7	1.4	0.5	0.1	0.7	0.7	74.2	9.1	0.8	0.2	4.3
314	0.1	0.6	1.8	2.9	4.5	0.2	0.5	1.1	8.9	73.7	0.3	0.8	4.6
234	5.6	2.2	0.4	6.2	0.2	0.7	0.5	0.5	1.2	0.5	68.5	7.7	5.6
324	1.8	4.2	0.1	0.8	0.4	6.7	1.2	0.5	0.7	0.8	7.3	69.6	6.0
444	1.1	0.6	0.3	0.6	0.5	0.4	6.0	4.9	5.6	5.0	3.0	3.7	68.3

TABLE VIII: Confusion matrix for 4 LSTM layer model with learning rate schedule Equation (1) trained on triples, double to triples, and false data from a 150MeV over 16384 epochs. The testing data used is from the MCDE model test1 150MeV 180K beam.

	123	132	213	231	312	321	124	214	134	314	234	324	444
123	77.6	6.2	1.9	1.2	1.9	1.7	6.0	0.5	0.0	0.0	2.2	0.7	0.0
132	3.1	78.8	1.7	1.2	1.4	1.7	0.5	0.0	5.8	1.2	0.7	3.4	0.5
213	1.2	2.2	75.4	4.8	1.2	3.1	1.2	6.0	4.1	0.5	0.2	0.0	0.0
231	3.4	1.7	5.3	71.6	4.1	2.4	0.0	0.0	1.2	4.3	4.8	0.5	0.7
312	3.9	2.4	1.0	1.7	73.7	3.1	4.8	1.0	0.5	6.7	0.0	0.2	1.0
321	1.7	2.2	2.7	1.9	3.1	76.9	3.1	3.6	0.0	0.2	0.7	3.4	0.5
124	5.4	0.8	0.6	0.1	3.0	1.1	76.6	6.6	0.5	0.3	0.3	0.9	3.8
214	1.0	0.0	4.1	0.5	1.8	3.3	8.8	74.2	0.4	0.7	0.3	0.2	4.7
134	0.5	5.0	3.5	1.3	0.8	0.1	0.6	0.4	73.7	8.8	1.0	0.8	3.6
314	0.0	0.2	0.9	3.0	5.4	0.3	0.5	1.3	8.3	74.3	0.2	0.8	4.8
234	4.4	1.7	0.6	5.8	0.2	0.9	0.3	0.6	1.0	0.6	71.2	6.4	6.1
324	2.5	4.4	0.0	0.6	0.5	6.7	2.0	0.4	0.5	0.6	6.5	69.9	5.4
444	0.6	1.0	0.3	0.7	0.3	0.4	6.8	4.9	5.8	5.1	4.7	4.2	65.3

TABLE IX: Comparison of LSTM model with deep fully connected network.

Class	LSTM	DRFCN	LSTM - DRFCN
123	80.0	79.1	0.9
132	75.8	76.0	-0.2
213	72.5	76.4	-3.9
231	75.9	80.7	-4.8
312	76.1	82.4	-6.3
321	73.5	76.5	-3
124	77.1	76.0	1.1
214	74.2	75.0	-0.8
134	74.2	75.4	-1.2
314	73.7	75.4	-1.7
234	68.5	73.6	-5.1
324	69.6	75.3	-5.7
444	68.3	72.6	-4.3

TABLE X: Comparison of the 4 layer LSTM with dense layers of 256 and 128 neurons with deep fully connected network at 100kMU dose rate.

Class	LSTM	DRFCN	LSTM - DRFCN
123	77.8	79.1	-1.3
132	76.7	76.0	0.7
213	75.8	76.4	-0.6
231	78.4	80.7	-2.3
312	77.8	82.4	-4.6
321	73.7	76.5	-2.8
124	74.5	76.0	-1.5
214	74.0	75.0	-1
134	71.5	75.4	-3.9
314	71.7	75.4	-3.7
234	71.8	73.6	-1.8
324	74.2	75.3	-1.1
444	68.2	72.6	-4.4

VI. Conclusions and Future Work

Results from the RNN (recurrent neural network) hyper-parameter study in Section V demonstrated that a learning rate scheduler benefits the model by increasing accuracy and efficiency. The learning rate schedule improves validation accuracy between 6% to 7%. Test results showed that at a higher number of epochs and with a smaller learning rate, the accuracy of the network increases. Due to the success of the learning rate scheduler, the LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit) models were trained using the scheduler. For these studies, a piece-wise function was created to illustrate the change in learning rate. From Section V-A, the maximum training accuracy for the model reached was 89% with 32,000 epochs.

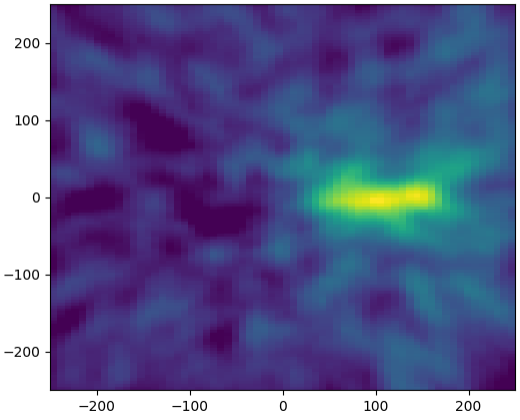
In Section V-B, we use dropout layers in between each recurrent layer to randomly zero out 20% of the neurons in each layer. A model with 4 GRU layers of 128 neurons and 2 dense layers of 128 and 64 neurons, respectively, has a testing accuracy of 73.4%. The model is able to load from its saved state to an active state, i.e., load from disk to GPU memory in 10s. A model with 4 LSTM layers of 128 neurons and 2 dense layers of 128 and 64 neurons, respectively, has a testing accuracy of 73.2%. The model is able to load from disk in 7s. The major advantage of this model are that it contains only 6 hidden layers which leaves a tremendous amount of space for further research and growth while already having a testing accuracy of 73%. Further, in real-time imaging, loading from disk is a potentially significant advantage when treating patients.

TABLE XI: Comparison of top performing models with the deep residual fully connected network (DRFCN) from [4].

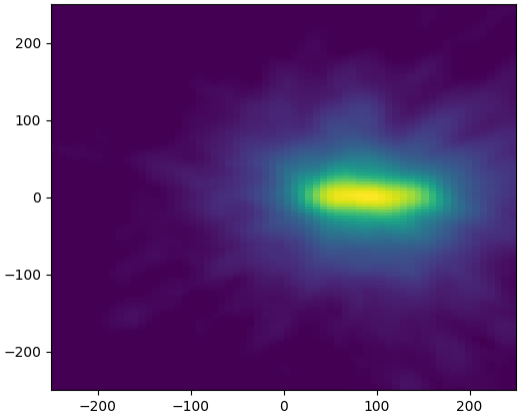
Model	Accuracy	Load Time
DRFCN (512 FCL)	75.8%	47s
4 LSTM w/ more neurons	74.4%	15s
4 GRU	73.4%	10s
4 LSTM	73.2%	7s

The key results of this work are summarized in Table XI. The Model column refers to the architecture of the model. The first row shows the results of the deep residual fully connected network (DRFCN) in [5]; this model has 512 fully connected layers (FCL). 4 LSTM w/ more neurons represents the 4 LSTM layer model with two dense layers of 256 and 128 neurons. 4 GRU represents the model with 4 GRU layers and 2 dense layers of 128 and 64 neurons. 4 LSTM represents the model with 4 LSTM layers and 2 dense layers of 128 and 64 neurons. The Accuracy column represents the overall testing accuracy of the model at the dosage rate of 100kMU/min. The Load Time column represents the observed wall clock time in seconds to load the model from its saved state to an active state, i.e., from disk to GPU memory. These measurements report observations on a reference computer, a basic laptop with an 11th Gen Intel Core i7–1165G7 CPU at 2.80 GHz with 16 GB of memory. The laptop has Intel Optane Memory H10 with 512 GB Intel QLC 3D NAND solid state drive connected by PCIe 3.0 x4 with NVMe interface. The GPU on the laptop is an Intel Iris Xe Graphics card. On a large cluster like taki or ada, described in Section IV, these times would in fact be *slower*, since the central rotating disk storage is much larger and connected only via network cables to the compute nodes. Even with high-performance fiber-optic cables, this is slower than direct connection from solid state storage inside a laptop. However, such direct connection and use of solid state storage is more realistic for the type of computer used in a clinical setting in a treatment room.

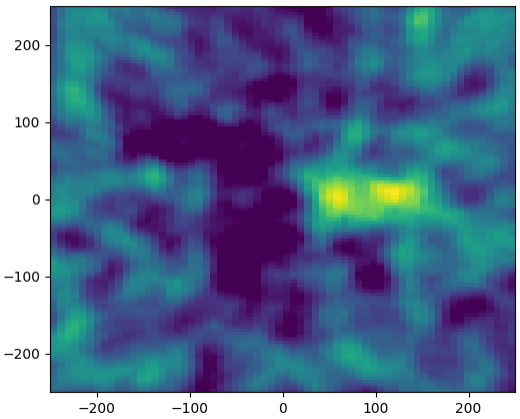
The DRFCN model has the highest accuracy of 75.8% with the load time of 47s. The models in the last two rows of the table have accuracies of 73.4% and 73.2% respectively while loading in 10s and 7s. These 4 GRU and 4 LSTM models are much simpler with only 6 hidden layers instead of 512. In particular, they have a factor 85 fewer layers while being only 2% less accurate. These two recurrent models are also 4 times faster to load from disk which is an advantage when treating the patient. This demonstrates the two recurrent models are much smaller than the DRFCN model but perform almost as accurately. Smaller models require less GPU memory to process similar amounts of data as well as process similar amounts of data in less time compared to larger models. This can save time and resources when in clinical use. In clinical use, the Compton camera software would be started-up and that process would include loading the neural network. Given



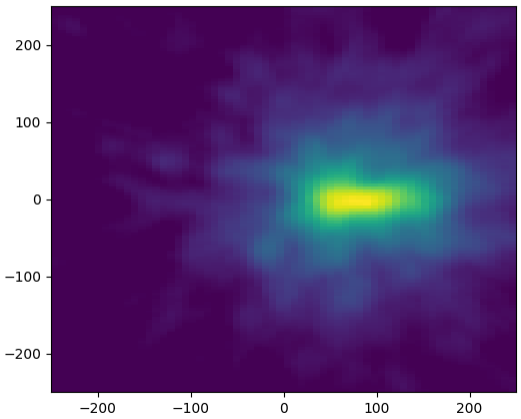
(a) 20kMU/min Uncleaned



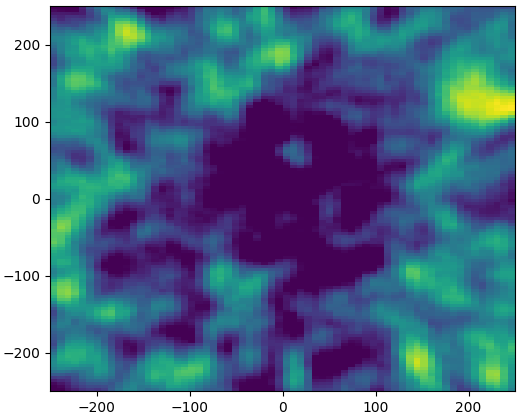
(b) 20kMU/min Cleaned



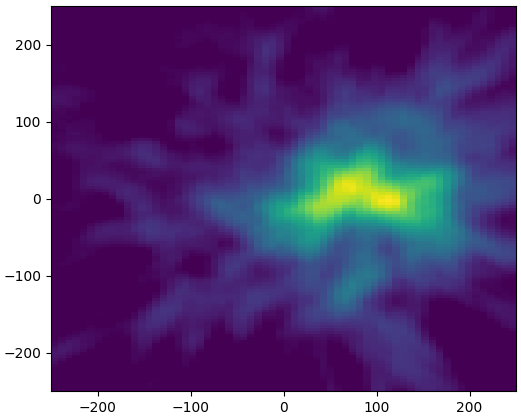
(c) 100kMU/min Uncleaned



(d) 100kMU/min Cleaned



(e) 180kMU/min Uncleaned



(f) 180kMU/min Cleaned

Fig. 1: The left column (a), (c), (e) uses testing data without the NN classification for data correction, called the "uncleaned" data. The right column (b), (d), (f) uses testing data with NN classification for data correction, called the "cleaned" data with the 4 layer GRU model described in Section V-B. Testing data used comes from MCDE model test1 150MeV.

the possibility of human error by the operator, a neural network that is quicker to load from disk and that processes data quicker would be advantageous. An error in the use of the neural network during treatment can be corrected quicker on the two smaller recurrent models.

To illustrate the effect that network event classification can have on the PG images produced from the camera data, reconstructed PG images are shown in Figure 1 for the GRU model. In Figure 1, there are three rows of PG image reconstructions for each dose rate corresponding the the MCDE model test1 150MeV. The technical report [9] show reconstructed images for the 4 LSTM layer model. The images in the left column are the respective PG images reconstructed with raw data prior to NN classification, called the "uncleaned" data. The images in the right column are the respective PG images reconstructed with data after it has been corrected based on the NN classifications, called the "cleaned" data. Since each PG image is from data collected during delivery of the same 150MeV proton beam they will have the same position and range even though they are reconstructed from data collected at different dose rates. We observed an improved visual appearance of the beam in which the start point and end point are now easily distinguishable at all three dose rates. The method used to reconstruct these images is described in [4].

The 6 hidden layer model has a large space for improvement due to its simplicity. Transformer networks were briefly explored in [9] and its initial results did not increase accuracy as expected. However, the hyperparameter space is very large and there is still potential in finding the optimal combination and architecture. The bidirectional LSTM are also tested in [9], but did not show any improvement. More complex architectures than 4 recurrent layers and 2 dense layers should be explored in addition to more techniques of regularization. There is also room in the RNN and DRFCN merged models where, rather than stacking the RNN layers in front of the DRFCN, the RNN layers could be dispersed between the FCLs, placed inside the residual blocks, or placed behind the DRFCN. From the results of this work, it is still possible that the optimal configuration of hyperparameters has still not been achieved for the more complex recurrent architectures (RNNs with residual blocks and transformers). Therefore, hyperparameter searches and exploring different optimization techniques could increase the accuracy of those models.

ACKNOWLEDGMENT

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). Co-author Kelly additionally acknowledges support as HPCF RA. Co-author Polf acknowledges support from the NIH. 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, OAC–1726023, and CNS—1920079) 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] J. C. Polf and K. Parodi, "Imaging particle beams for cancer treatment," *Phys. Today*, vol. 68, no. 10, pp. 28–33, 2015. [Online]. Available: https://doi.org/10.1063/PT.3.2945

[2] E. Muñoz, A. Ros, M. Borja-Lloret, J. Barrio, P. Dendooven, J. F. Oliver, I. Ozoemelam, J. Roser, and G. Llosá, "Proton range verification with MACACO II Compton camera enhanced by a neural network for event selection," *Sci. Rep.*, vol. 11, no. 1, p. 9325, 2021. [Online]. Available: https://doi.org/10.1038/s41598-021-88812-5

[3] J. C. Polf, C. A. Barajas, S. W. Peterson, D. S. Mackin, S. Beddar, L. Ren, and M. K. Gobbert, "Applications of machine learning to improve the clinical viability of Compton camera based in vivo range verification in proton radiotherapy," *Front. Phys.*, vol. 10, p. 838273, 2022. [Online]. Available: https://www.frontiersin.org/article/10.3389/fphy.2022.838273

[4] C. A. Barajas, M. K. Gobbert, and J. C. Polf, "Deep residual fully connected neural network classification of Compton camera based prompt gamma imaging for proton radiotherapy," submitted (2022). [Online]. Available: http://userpages.umbc.edu/~gobbert/papers/Barajas_DigitalMedicine2022.pdf

[5] C. A. Barajas, "*Neural Networks for the Sanitization of Compton Camera Based Prompt Gamma Imaging Data for Proton Radiotherapy*," Ph.D. Thesis, Department of Mathematics and Statistics, University of Maryland, Baltimore County, 2022.

[6] A. M. Ali, D. Lashbrooke, R. Yepez-Lopez, S. A. York, C. A. Barajas, M. K. Gobbert, and J. C. Polf, "Towards optimal configurations for deep fully connected neural networks to improve image reconstruction in proton radiotherapy," UMBC High Performance Computing Facility, University of Maryland, Baltimore County, Tech. Rep. HPCF–2021–12, 2021. [Online]. Available: http://hpcf.umbc.edu

[7] S. A. York, A. M. Ali, D. Lashbrooke, R. Yepez-Lopez, C. A. Barajas, M. K. Gobbert, and J. C. Polf, "Promising hyperparameter configurations for deep fully connected neural networks to improve image reconstruction in proton radiotherapy," in *2021 National Symposium for NSF REU Research in Data Science, Systems, and Security (REU 2021 Symposium)*, in press (2021).

[8] F. Guo, "Comparison of feedforward and recurrent neural networks for predicting pavement roughness," 2021. [Online]. Available: cshub.mit.edu

[9] J. Clark, A. Gaillard, J. Koe, N. Navarathna, D. J. Kelly, M. K. Gobbert, C. A. Barajas, and J. C. Polf, "Sequence-based models for the classification of Compton camera imaging data for proton beam therapy," UMBC High Performance Computing Facility, University of Maryland, Baltimore County, Tech. Rep. HPCF–2022–12, 2022. [Online]. Available: http://hpcf.umbc.edu