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

Jonathan N. Basalyga[1]

In collaboration with: Gerson C. Kroiz[1], RA: Carlos A. Barajas[1], Faculty: Matthias K. Gobbert[1], Clients: Paul Maggi[2], Jerimy Polf[2]

[1]Department of Mathematics and Statistics, University of Maryland, Baltimore County
[2]Department of Radiation Oncology, University of Maryland School of Medicine

CyberTraining, 2020

Acknowledgments: NSF (CyberTraining, MRI), NIH, UMBC, HPCF, CIRC, Math & Stat

Maryland Proton Treatment Center

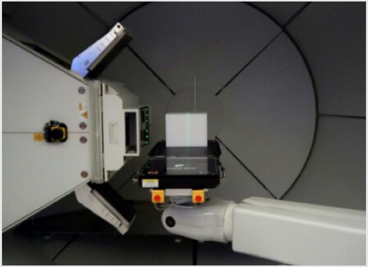
Image credit: Maryland Proton Treatment Center

This work was done in collaboration with the Maryland Proton Treatment Center, located in Baltimore. Opened in 2016, the center was the first in the Maryland/DC region to offer proton therapy for cancer treatment. In the past four years it has trained more than 200 health care professionals in proton therapy and, with its state of the art facilities and four treatment rooms, has been able to treat over 2,000 patients.

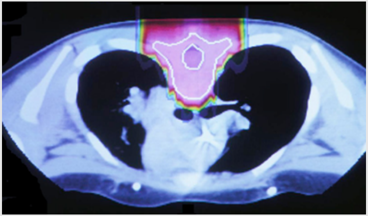
`www.mdproton.com`

Proton Beam Therapy

Proton beams' advantage in cancer research is their finite range. They reach their highest dose just before they stop, at what is called the Bragg peak. Little to no radiation is delivered beyond this point.



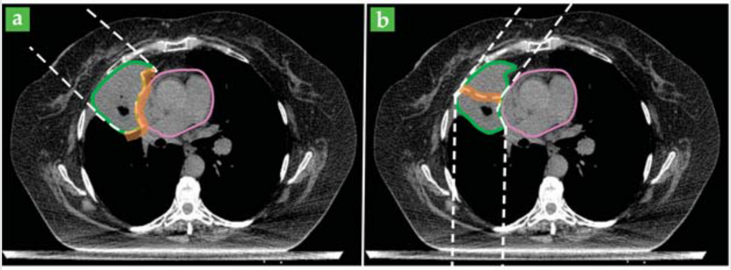
Treatement table in
MD Proton Treatment Center
Image credit: Maggi, 2019



Radiation levels in
proton beam therapy
Image credit: Public domain

The Need for Real-Time Imaging

- Uncertainties in the beam's position limit proton beam therapy's advantages.
- Imaging the beam in near real time would reduce uncertainties and allow the advantages of the Bragg peak to be fully exploited.



(a) Optimal trajectory. Notice orange bar representing uncertainty intersects the heart (b) Suboptimal trajectory necessary to protect heart. A low dosage irradiates healthy lung tissue.
Image credit: Polf, Parodi. Physics Today, 2015

Image Reconstruction using Compton Camera

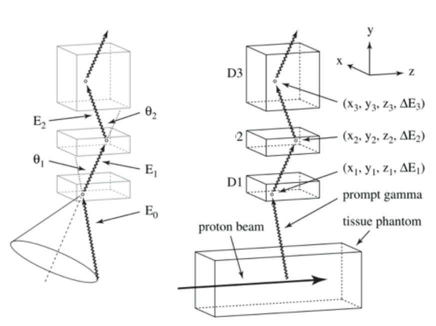


Image credit: Peterson, Robertson, Polf. Phys Med Biol, 2010

- Nuclear reactions between beam and tissue produce prompt gamma rays.
- Compton camera records the position and energies of each interaction.
- By analyzing how prompt gammas scatter through the Compton camera we can reconstruct their origin, thereby imaging the beam.

Limitations of the Compton camera

- It simply records events as single, double, or triple scatters.
- It cannot determine the correct ordering of camera events.
- It can't determine if a double or triple scatter event was triggered by prompt gammas originating from different physics events that happened to enter the camera at about the same time.

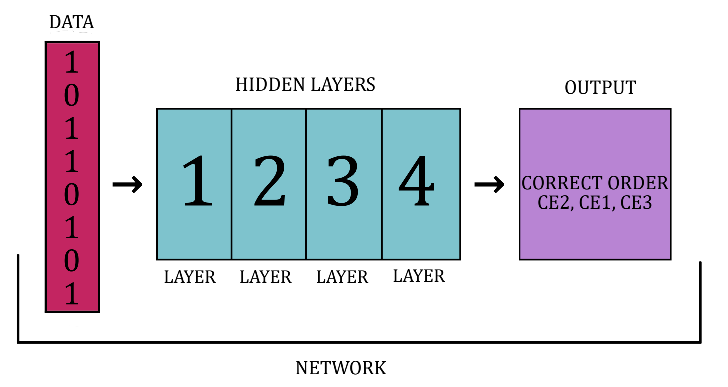
Note the difference between a camera event, which is a tuple of location data and the energy level of a single interaction that the camera records, and an event, which is all camera events read in a single readout cycle.

Idea

Use a neural network to preprocess the data:

- Neural networks can exploit subtleties in data that traditional models don't use
- We can train a neural network to differentiate true scatters from false scatters and determine the actual scattering ordering.
- The output of the neural network can be fed into a reconstruction algorithm.

What is a Neural Network?



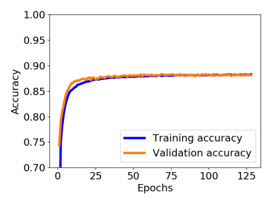
- A neural network takes some input and passes it through a sequence of transformations called hidden layers and a final output layer to produce an output of some specified form.

Chollet, *Deep Learning with Python*, 2018

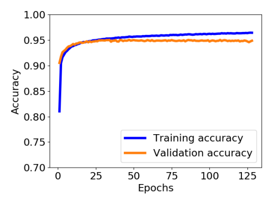
- Each hidden layer is of the form $g(Ax + b)$ where A and b are tensors (the weights) and g is some non-linear function called the activation function. The whole neural network represents some function f which is the composition of all layers.
- The number of layers and their activation functions are fixed.
- Given some loss function we would like minimize the loss between the output of the neural network and what the output "should be".
- By updating the weights of the neural network using an algorithm called Gradient Descent we can find the weights with minimal loss. This is called training the neural network.
- We can improve performance of the network by training it with the same data multiple times. Each pass through the data is referred to as an epoch.
- Deep Learning refers to the use of many hidden layers.

Chollet, *Deep Learning with Python*, 2018

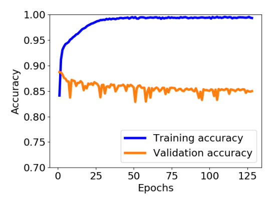
Overfitting vs. Underfitting



Underfitting



Good fitting



Overfitting

- One of the primary challenges in deep learning is avoiding underfitting and overfitting
- A network underfits when it could be improved by more training. In this case there is still information in the data the network hasn't internalized.
- A network begins to overfit when its performance on testing and validation data decreases, even though it performs well on training data. This is because the network has begun memorizing patterns in the training data that don't generalize.

Hardware and Software Used

- HPCF2018:
  - 1 GPU node (gpunode001) containing four NVIDIA Tesla V100 GPUs connected by NVLink and two 18-core Intel Skylake CPUs
  - The node has 384 GB of memory (12 x 32 GB DDR4 at 2666 MT/s) and a 120 GB SSD disk
- HPCF2013:
  - 18 hybrid CPU/GPU nodes (gpunode[101-118]), each two NVIDIA K20 GPUs (2496 computational cores, 5 GB onboard memory) and two 8-core Intel E5-2650v2 Ivy Bridge CPUs (2.6 GHz clock speed, 20 MB L3 cache, 4 memory channels)
  - Each node has 64 GB of memory (8 x 8 GB DDR3) and 500 GB of local hard drive
  - The nodes are connected by a QDR (quad-data rate) InfiniBand switch
- Software: Python 3.7.6, Tensorflow 2.1.0, Keras 2.2.4-tf, NumPy 1.18.1, sklearn 0.20.dev0

Network implementation

Input data structure:

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

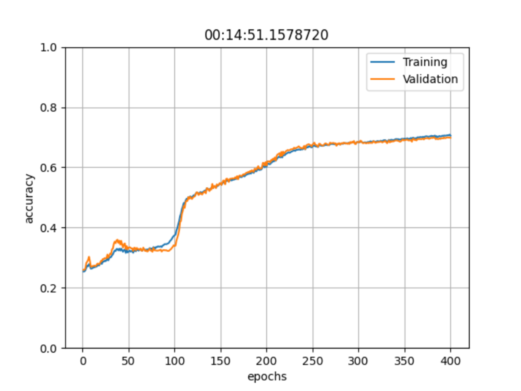
- The data is arranged such that we have an energy level for each camera event.
- We have a position in 3D space produced by the camera. These will serve as our input data.
- We also have a lot of sundry information like: the correct order, initial energy level, was it a partial scattering, and many other usable features which we can use as output data.
- The output is the correct slot in a vector, e.g. [1,0,0,0], where the position of the 1, created by softmax activation, determines pos 1, pos 2, pos 3, or false (pos 4).
- The first version of the network only classifies a single camera event.

This network is fully connected using residual skips.

Failure to Learn

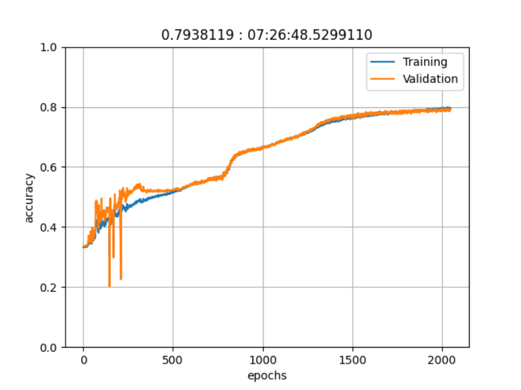
- With an 8 layer fully connected network, the network had a prediction rate of 20%.
- Despite thousands of epochs it refused to learn and grid searching hyperparameters yielded no meaningful results.

Normalization



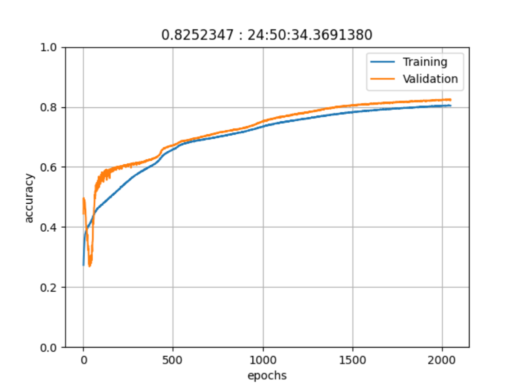
- Simply normalizing the data caused the network to start learning almost instantly.
- The accuracy seemed to level out at 70%

Increasing Epochs



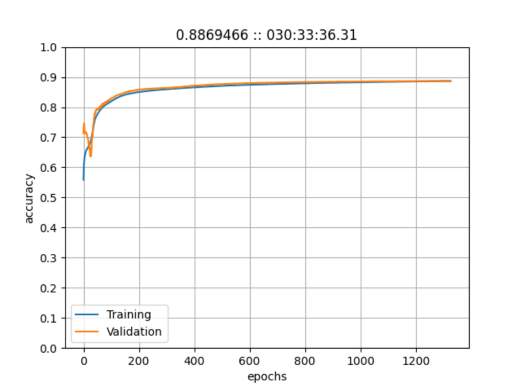
- By pushing to 2000 epochs we were able to seen accuracy of nearly 80%!
- However since accuracy and validation are tightly overlapping we concluded that the network was underfitting and increased the number of layers

Adding Network Complexity



- By increasing the number of layers to 16 and changing the activator to SeLU we noticed that the initial accuracy was considerably higher.
- The final accuracy is also slightly higher at 82%

Training on More Data

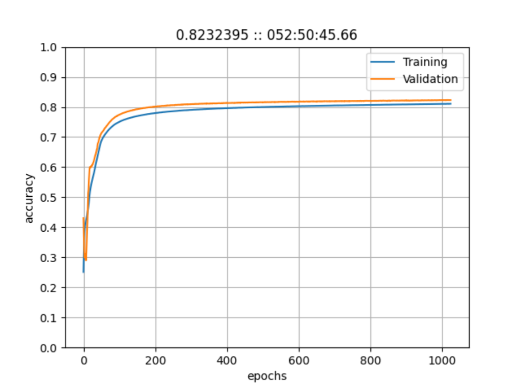


- The most substantive change was using $5\times$ more data for classfication and two computed vaues.
  - Euclidean distance between all camera events with respect to each other.
  - Energy difference with respect to each other.

New Network Design

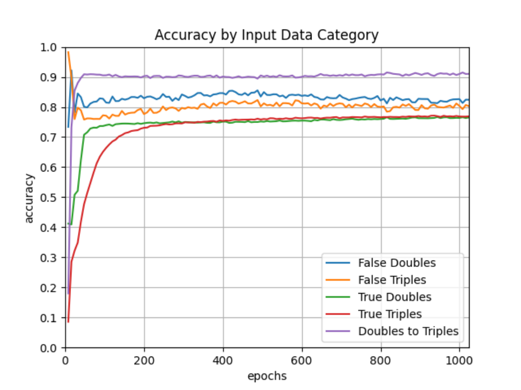
- The current version of the network is a deep fully connected network using 24 layers and residual skips in order to support this size.
- The network now classifies all camera events simultaneously. The output is permutation of camera events.
- Some examples of network output are ’132’ and ’431’. The first gives the correct order of a true triple event, the second is a Double to Triple where the correct camera events are 1 and 3 and their order is 3, 1.
- The network uses a softmax activation function, so it gives a probability for each possible permutation and the permutation with highest probability is selected as the output.

Training the New Class System



- After training for over two days accuracy reaches around 82%, lower than our previous network.
- However, this network classifies full events. If we had naively applied our previous network we might expect an accuracy of 68% (.88 $\times$ .88 $\times$ .88) due to compounded error.

Performance by Data Category



- The network performs best at classifying Doubles to Triples
- The network can classify each category with an accuracy of at least 70%

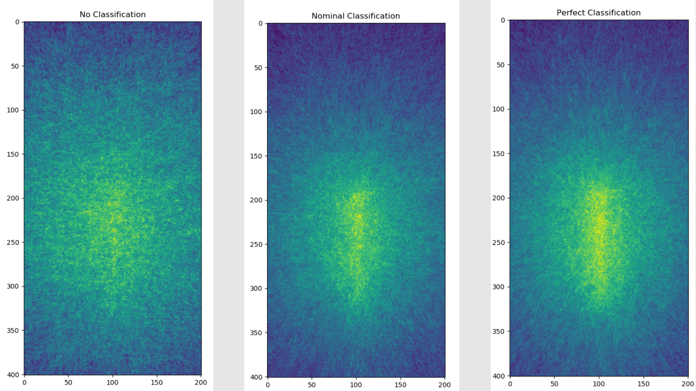
Training on Input Categories one at a time

Inspiration:

- GANs work by training two networks in competition with each other, a generator which creates artificial data, and a discriminator that attempts to classify real and artificial data.
- It's been found that descriminators learn best when first trained on all artificial data then trained on all real data, instead of feeding in both types of data mixed together.

Since our network is performing a similar task to the descriminator's, separating false data from true data, it's possible that by training on categories in isolation, that is, training on all data from one input category before moving on to the next, we can improve performance.

Classification's Effect on Reconstruction Quality



- By classifying data at the same accuracy that the network classifies data we see a noticeable improvement in reconstruction quality.
- These reconstructions are made with only the simplest reconstruction method, back projection.

Conclusions and Future Work

Conclusions

- There is still a large difference in accuracies between different input cases, however we have shown it is possible for a deep fully connected neural network to classify entire events with reasonable accuracy.
- Removing false events and correctly ordering true events can improve reconstruction quality.
- We've reached the maximum possible network complexity using a fully connected neural network.

Future Work

- To train on input categories one at a time, rather than training on all categories at the same time.
- To implement an RNN based approach in order to leverage the fact that each camera event is composed of a sequence of data.