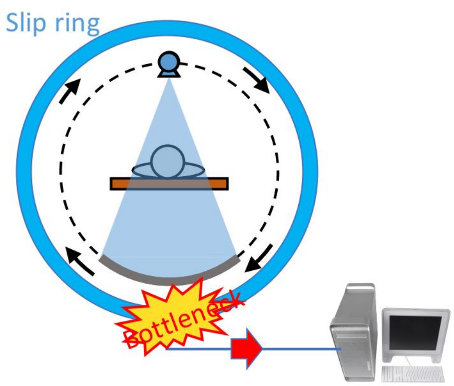
Neural network for compression of photon counting detector projection data

Picha Shunhavanich
Departments of Bioengineering and Radiology,
Stanford University
picha@stanford.edu

Code: https://github.com/pichashun/projdata

Introduction

With many attractive attributes, photon counting detectors (PCDs) with many energy bins are being considered for clinical CT systems. In practice, a large amount of projection data acquired for multiple energy bins must be transferred in real time through slip rings and data storage subsystems, causing a bandwidth bottleneck problem. The higher resolution of these detectors and the need for faster acquisition additionally contribute to this issue. Thus, the system could benefit from compression of the acquired data in the gantry before transmission.



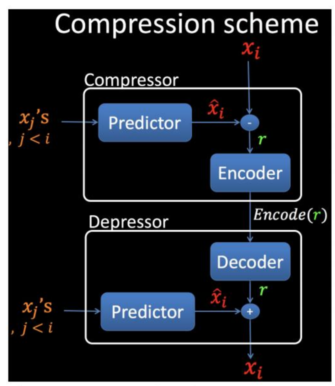


Fig. 1 **Left:** Illustration of bandwidth bottleneck from data transmission. **Right:** Compression scheme

In this work, we built a neural network to use as a predictor of a compression scheme. As illustrated in Figure 1 (right), an approach to image compression is to make a prediction of the sample value with a predictor that can be employed at both ends of the transmission, and to encode and transmit the prediction error rather than the high bit-width sample value itself. The decompressor uses the same predictor as the compressor, decodes the transmitted residual and subtracts it from the prediction. The data required to make the prediction must be available to both sides of the transmission. Our compression method is therefore divided into two parts: prediction (estimating the value of the data to be compressed) and encoding (reducing the number of bits needed to represent the prediction error before transmission). *We use a neural network as a predictor*. Therefore, the input of our network is a projection data (i.e. an image with 5 color channels) with all previous pixels available as an example shown in Figure 2., and the output is the projection data with the prediction of the next value, x_i. Note that during training and evaluation we can input the full image as shown in Figure 3 (left) as we control the availability of the pixels for each time point by masking convolutional filters in the network (e.g. with mask in Figure 3 (right)).

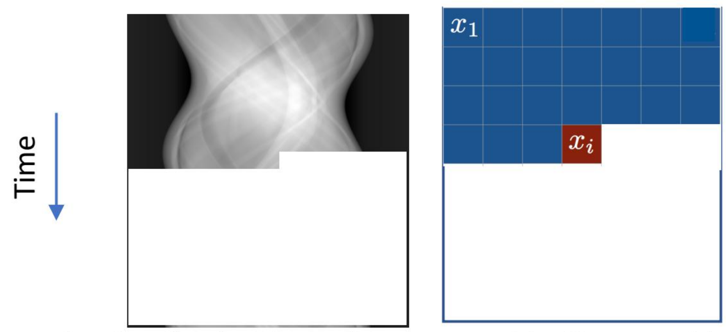


Fig. 2 **Left:** Example of projection data of one energy bin (5 energy bins/color channels in total) at one time point. **Right:** Example of image pixels available (x_j 's, $j < i$) (blue) for predicting value of the next pixel (x_i: *output*) (red)

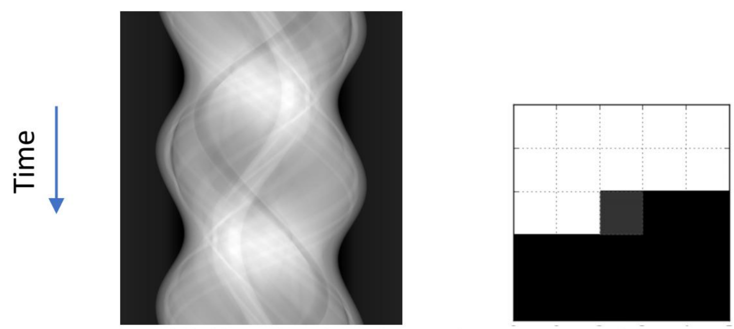


Fig. 3 **Left:** Example of full projection data of one energy bin. **Right:** Example of mask

Related work

We previously proposed a compression algorithm for PCDs projection data [1, 2] with a predictor based on the physics of X-ray signals, which did not employ deep learning. The neural network in this work could improve the performance of our compression method. To our knowledge, there has yet been other work that apply deep learning to this problem. However, there are deep learning work for RGB images that can be adapted to this problem, in particular, a PixelCNN in Ref. [3].

Dataset and Features

From CT images in NIH Clinical Center's DeepLesion dataset, we use our computer simulation to generate the projection data of PCDs (each with size 903x1000x5 pixels) assuming 1mm^2 detector pixel size, 1000 views per rotation, and 5 energy bins. Each scan was used to create 12 patches of size 78x78x5 pixels to be treated as sample data (converted to sinogram, down sampled by a factor of 4, and rescaled to range [0, 255]). The computer simulation and data preprocessing are done in MATLAB.

Methods

We implemented a PixelCNN using PyTorch based on Ref. [3]. The input, filters, and masks were modified to have 5 color channels. Mask A and B of the model are adjusted to assume the availability of only context for energy bin 1 output, availability of context and energy bin 1 for energy bin 2 output, availability of context and energy bin 1 and 2 for energy bin 3 output, … , availability of context and energy bin 1 to 4 for energy bin 5 output.
The original model classifies output value into 256 classes (at each pixel location and color channel). We used 4 residual blocks in the model. We also changed the last layer of the model to use Mean Squared Error loss as our goal is to reduce prediction errors. The diagram of the model is shown in Figure 4.
We started with the base code from Ref. [4] that use binary image and no residual blocks, modify the code for RGB mask from Ref. [5], and wrote the rest of the code ourself.

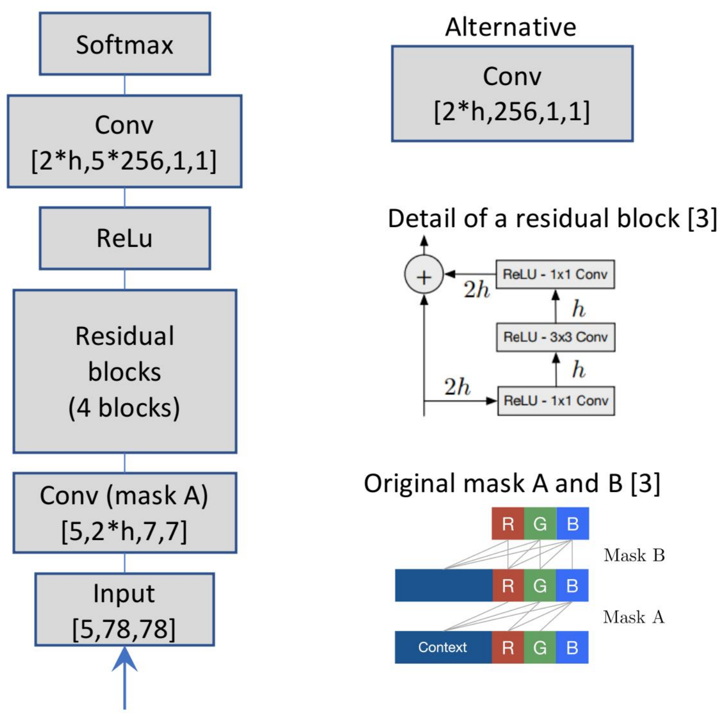


Figure 4. Diagram of the model

Experiments/Results/Discussion
Two types of loss function were experimented: CrossEntropyLoss (NLL) and Mean Squared Error. The learning rate used in both were 0.001. Larger learning rates were also tested but had significantly worse results. The batch size is 10 as it regularized well. We used 8000 samples for training and 2000 samples for testing (dev set, no test set).

As shown in Table 1 and Figure 5, the results of the model were reasonable. The models capture the structures in the images well. The model with MSE loss function outperform the model with CrossEntropyLoss in terms of visualization and average prediction errors. Better results could be gained from a larger network, running the model longer or more sample data. We would want the prediction error to be as low as possible for better compression.

Table 1. Results of the models

Model	Train (8000 samples)	Test (2000 samples)
CrossEntropyLoss(NLL)	2.73	2.76
Mean Squared Error	7.42	7.94

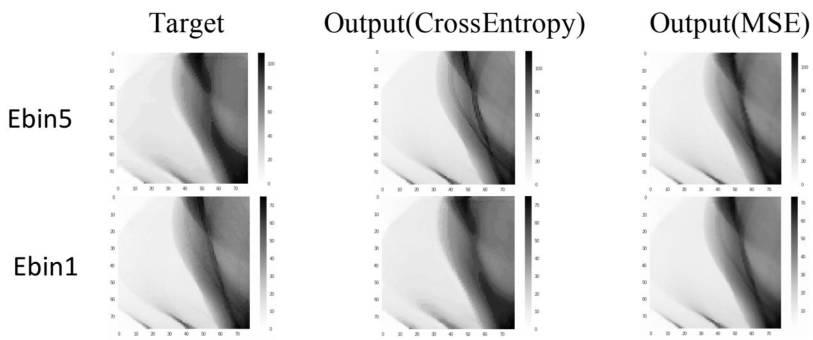


Figure 5. Example results of energy bin 1 and 5

Conclusion/Future work

In conclusion, the modified PixelCNN can effectively be used as a predictor in a compression scheme. The mean squared error loss function outperform the CrossEntropyLoss for our purpose of reducing prediction errors. For future work, an encoder could be developed to incorporate the model characteristics.

References

1. Shunhavanich, P., & Pelc, N. J. (2016, March). Lossless compression of projection data from photon counting detectors. In *Medical Imaging 2016: Physics of Medical Imaging* (Vol. 9783, p. 97831J). International Society for Optics and Photonics.
2. Shunhavanich & Pelc, N. J. (2016, July). Lossy Compression of Projection Data from Photon Counting Detectors. The International Conference on Image Formation in X-Ray Computed Tomography.
3. Oord, A. V. D., Kalchbrenner, N., & Kavukcuoglu, K. (2016). Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*.
4. https://github.com/jzbontar/pixelcnn-pytorch
5. https://github.com/rampage644/wavenet/blob/master/wavenet/models.py
6. http://sergeiturukin.com/2017/02/22/pixelcnn.html