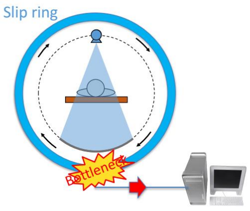
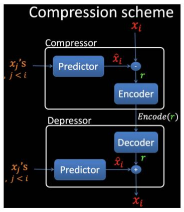
Neural network for compression of photon counting detector projection data

Picha Shunhavanich; Departments of Bioengineering and Radiology,
Stanford University, Stanford CA 94305; Correspondence: picha@stanford.edu

Problem description and motivation

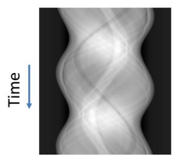
With many attractive attributes, photon counting detectors(PCDs) with many energy bins are being considered for clinical CT systems. In practice, a large amount of projection data acquired for multiple energy bins must be transferred in real time through slip rings and data storage subsystems, causing a bandwidth bottleneck problem. In this work, we built a neural network to use as a predictor of a compression scheme that encodes and transmit the prediction error rather than the high bit-width sample value itself.

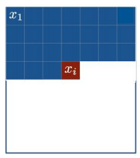




Data and Features

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). Each scan was used to create 12 patches of size 78x78x5 pixels to be treated as sample data (processed and rescaled to range [0, 255]).

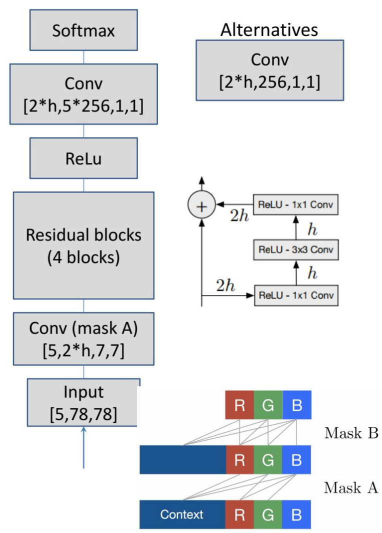




Left: Example of projection data (image to be compressed) of one energy bin.
Right: Example of image pixels available (x_j's, $j < i$: *input of a neural network*) (blue) for predicting value of the next pixel (x_i: *output*) (red)

Models

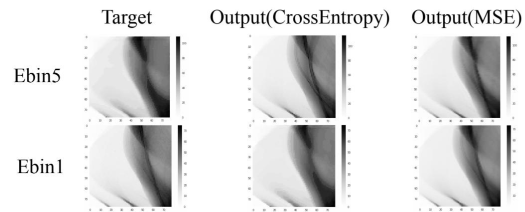
We implemented a PixelCNN using PyTorch based on the paper: Oord, Aaron van den, Nal Kalchbrenner, and Koray Kavukcuoglu. "Pixel recurrent neural networks." The input, filters, and masks were modified to have 5 color channels. The original model classifies output value into 256 classes (at each pixel location and color channel). We also changed the last layer of the model to use Mean Squared Error loss as our goal is to reduce prediction errors.



Results

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.

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



Discussions

The results of the model were reasonable. The model captures the structures in the images well. 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. An encoder could be developed to incorporate the model characteristics.

References

Oord, Aaron van den, Nal Kalchbrenner, and Koray Kavukcuoglu. "Pixel recurrent neural networks." arXiv preprint arXiv:1601.06759 (2016).
https://github.com/jzbontar/pixelcnn-pytorch
http://sergeiturukin.com/2017/02/22/pixelcnn.html