UNIVERSITY OF RHODE ISLAND

Department of Mathematics

Applied Mathematics and Scientific Computing Seminar

Location: Lippitt Hall 204

Time: Friday, April 26, 2019, 3:00pm

(refreshments at 2:50 p.m.)

Reducing the Dimensionality of Data with Neural Networks

by **Yi Xu**

Graduate student, Department of Mathematics, URI

Abstract: Principal Component Analysis (PCA) is a widely used method for dimensionality reduction. It is designed to find the direction in which the variance of the data points is greatest, and use these directions as coordinates to represent the various points in the dataset.

In this talk we describe the results in a paper by Hinton and Salakhutdinov *"Reducing the Dimensionality of Data with Neural Networks,"* Science, Volume 313, 2006. The authors use a non-linear version of PCA that uses an adaptive multi-layered coding network to transform data from high-dimensional to low-dimensional encoding, and a similar decoding network from low-dimensional encoding to refactor the data. The network is initialized with random weights, and the overall structure of the network is trained by minimizing the error between the original data and the reconstructed data. The gradient is calculated using the chain rule and the gradient is propagated backwards to update the network weights.

For autoencoder, the initial weight setting is very important. If it is too large, it will lead to a bad local optimum. If it is too small, it will lead to training difficulties. Only by finding a good initialization weight can make subsequent gradient algorithm converge to an ideal local solution. Finding such an initialization weight requires many types of algorithms to be tried for each layer, so this article introduces the process of *pretraining*. The pretrain process can be abstracted as: the model tries to learn one layer of feature detector, and then use the output of current layer as the input features of the next layer.

The whole deep learning autoencoder can be split into three stages: pretraining, unfolding, and fine-tuning. The pretraining process is to train layer by layer, the unfolding is to use the weights learned from pretraining into the encoder and decoder, and the fine-tuning means to use the backpropagation algorithm to slightly tune the autoencoder architecture's weights.

The experiments are conducted on the widely used MNIST dataset. The best reported error rates are 1.6% for randomly initialized backpropagation and 1.4% for support vector machines.