

Reducing the Dimensionality of Data with Neural Networks

Assignment 4 COMP30230

Ersi Ni

15204230

April 25, 2017

Geoffrey Hinton et al. published a landmark paper in the science magazine in 2006, describing a neural network based data compression and reconstruction scheme called "autoencoder". Furthermore in this paper, the authors have provided a break-through solution to tackle the problem when training neural networks with many hidden layers.

Training many hidden layer neural network using gradient descent has many challenges. Due to gradients descent being a greedy solution, training with large weights will often have gradients trapped in local minima; while training with small weights will take exorbitant amount of effort because gradient vanishing in early stage. In the paper Hinton et al. proposed a pretraining scheme that solves this problem.

The data compression, or dimensionality reduction scheme is powered by stacking binary features trained with restricted boltzmann machine on top of each other. Each layer is a stochastic feature compression of the layer below it. These feature layers are built layer by layer, from bottom to top. The result from this building process retains a set of weights of the entire network structure that is friendly to gradient descent. Hence the term "pretraining".

Hinton et al. demonstrated this method's result using a synthetic data set of 2D image with a known intrinsic dimensionality and highly non-linear pixel intensity. "autoencoder" is put up to a comparison with PCA and showed superiority. Further the authors demonstrated the model's reconstruction's performance on MNIST character image dataset and on face image data set. For all these, "autoencoder" clearly out-performed PCA.

To demonstrate the flexibility of the model, the authors performed evaluation on text data, as well as the usage of pretraining scheme in classification and regression problem. They showed that in MNIST classification task, a pretrained deep network broke the previous record in lowest error rate.