Usecases of autoencoders

Quick overview

Dimensionality reduction

- Used to reduce the dimensionality of a dataset by figuring out complex patterns among the variables (~non linear PCA)

- Could be implemented as a pretransformation in TMVA

Cleaning input data

- **Denoise autoencoder:** the inputs before fed to the net is corrupted, such that the autoencoder learns to reconstruct the proper data (or find patterns in it) from data with a certain corruption level

- Could be implemented as a pretransformation in **TMVA**

Weight initialisation for a DNN

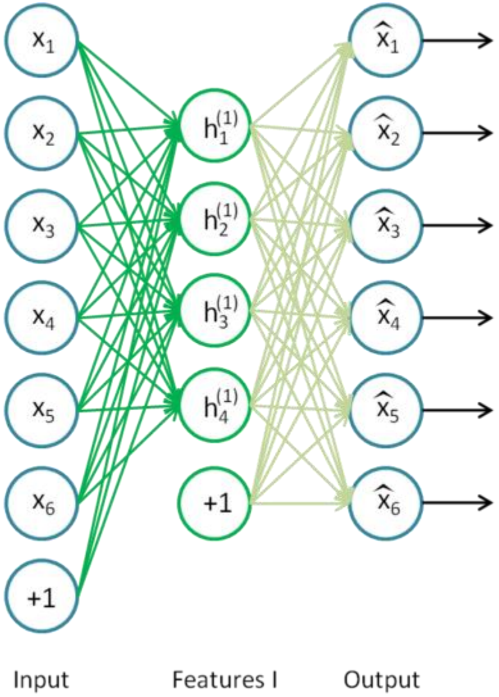
- Pretrain the layers of a DNN by autoencoders, which have the ability to find better than random initialisation, according to the data

Pretraining of DNN layers

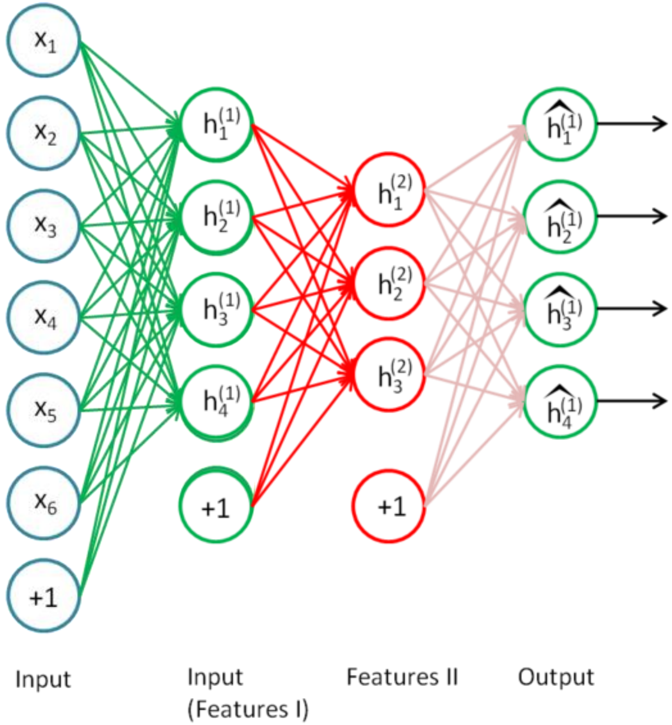
- Each layer of the DNN is made the hidden layer of a sparse autoencoder, which “pretrains” the layer (initializes the weights)
- This is done recursively, each pretrained layer being feed with data and becoming the inputs for the next sparse autoencoder

Source : http://ufldl.stanford.edu/wiki/index.php/Stacked_Autoencoders

Pretraining of DNN layers



Pretraining of DNN layers



Classification

- Add a **convolutional** layer at the end of the autoencoder to get classification (~CNN?) (use the learned features in the data to classifiy)

Anomaly detection

- Output (decoded) compared to the actual input, after autoencoder has been trained on real data. If too far from each other, it means the algorithm was not trained on this particular case.

About autoencoders

- For all applications, the data needs to be correlated