FEATURES LEARNING FOR DEEP LEARNING MODELS

Autoencoder & Stacked Autoencoder

Ahmed Hani, Ibrahim Sharaf, Abdelrahman Hamdy

August 1, 2017

FCIS'17 Machine Learning Course

Today's Objectives

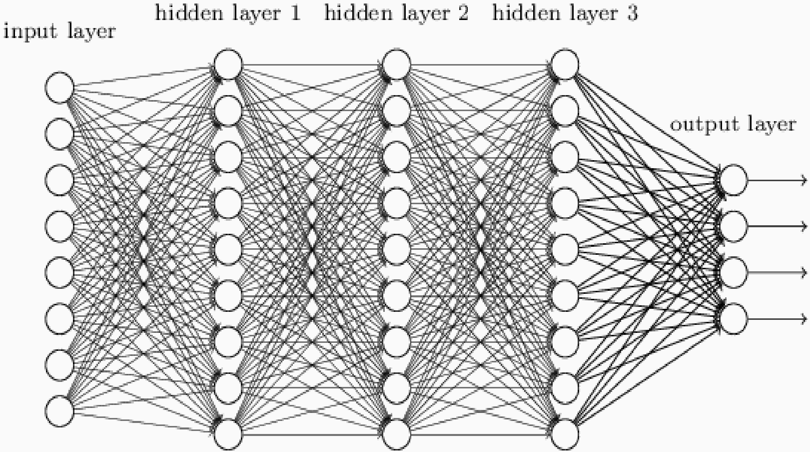
Learning

- Deep Learning Pipeline
- Transforming Raw Data to Features
- Autoencoder - Powerful Features Learning and Dimensionality Reduction model

To Do

- Implement a simple Autoencoder using Tensorflow
- Build end-to-end Model for MNIST Dataset
- Getting started with Keras - High Level Deep Learning Framework

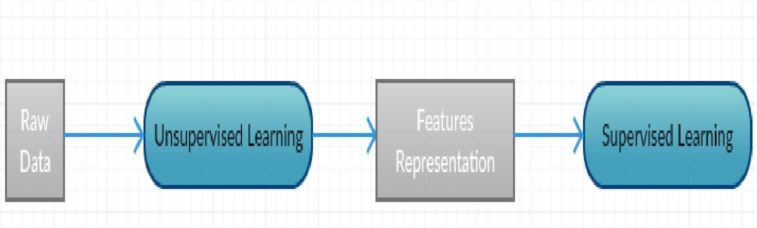
Why Deep Learning?



Why Deep Learning? (Cont.)

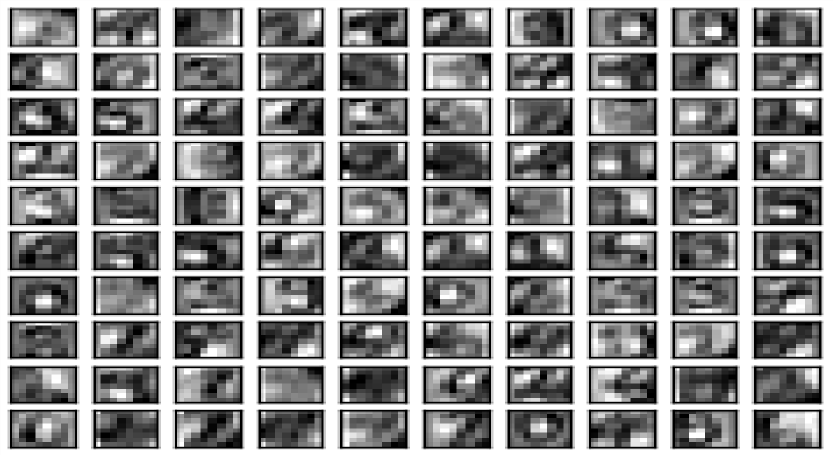
- Performance Issues with very Deep Neural Networks
- **Vanishing/Exploding** Gradient of Backpropagation over the network layers
- Slow Training Process

Deep Learning Pipeline



Features Learning

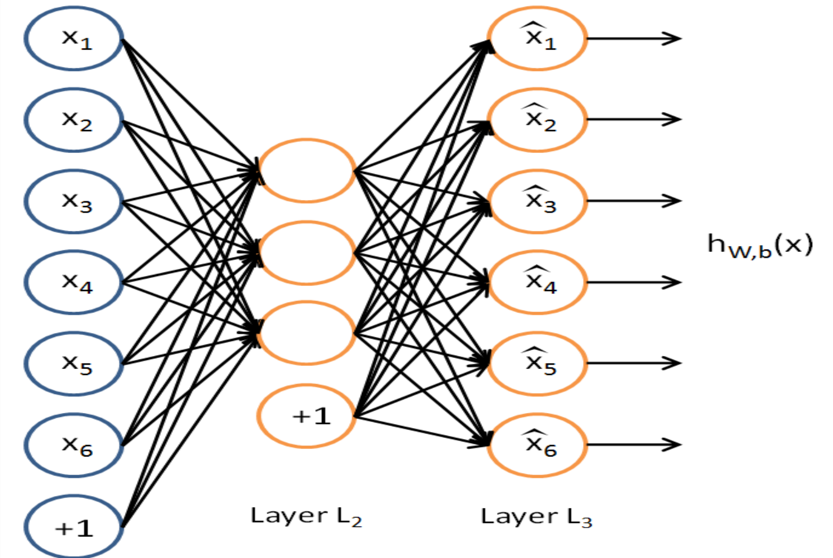
Features Learning (Cont.)



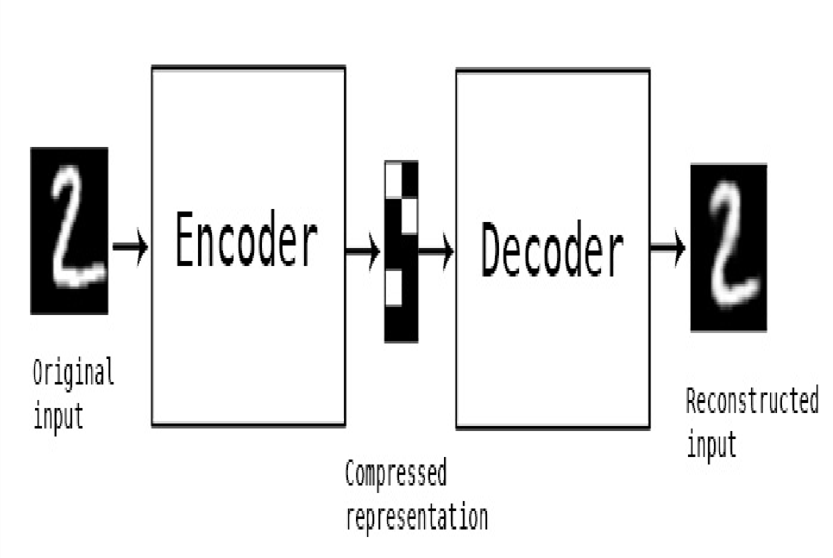
Features Learning (Cont.)

- Discover the representations needed for feature detection or classification from raw data
- The generated features are considered to be the **lower dimension** representation of the original features
- The generated features could or couldn't have the same original data statistical distribution
- Generative Model Examples: Restricted Boltzmann Machines (**RBMs**) and Deep Belief Networks (**DBNs**)
- Input-Reconstruction Model Examples: Vanilla Autoencoders (**AEs**) and Stacked Autoencoders (**SAEs**)

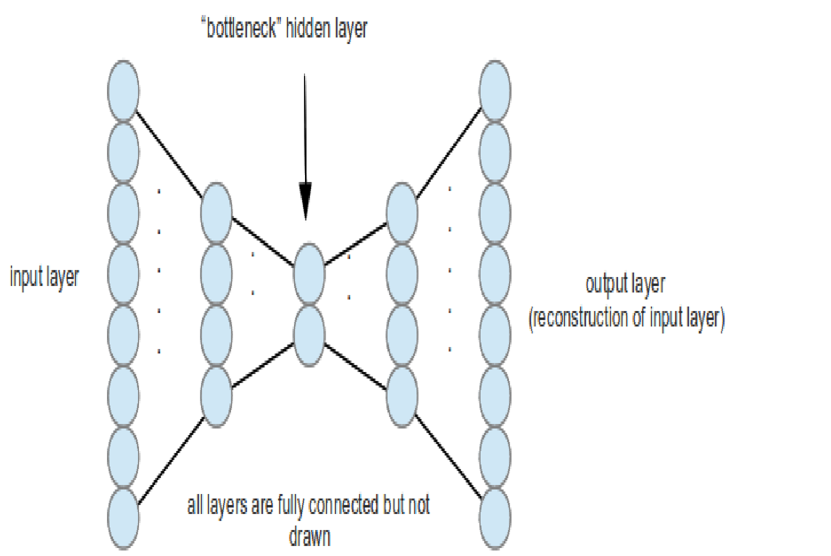
Vanilla Autoencoder Model



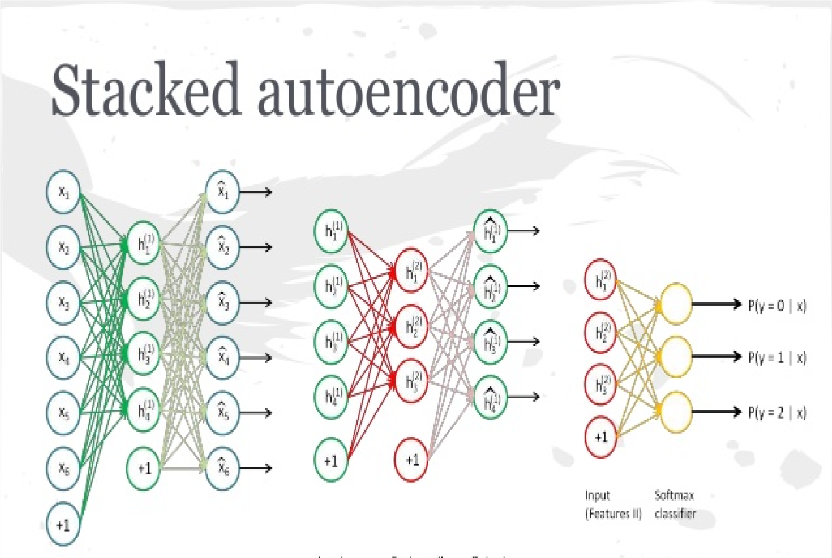
Vanilla Autoencoder Model (Cont.)



Deep Vanilla Autoencoder Model



Stacked Autoencoder Model



Autoencoder Notes

- Regularly, **Sigmoid** is always used as the hidden layer(s) activation function
- **Mean Squared Error** is used as the output layer loss function
- Backpropagation is used as the optimization algorithm, like any other Neural Network

Practical