An Introduction to Deep Learning

By Rahil Mahdian – December 12-13, 2017
Outline

- Machine Learning
- Learning Strategies
- Neural Network Learning
- Deep Learning
- Feed Forward Network
- Problems of Deep Learning
- AutoEncoders
- Restricted Boltzmann Machines
- Convolutional Neural Networks
- Recurrent Neural Networks (RNNs, LSTM)
- Deep Learning Applications
Scope of Machine Learning

Machine learning deals with the problem of extracting features from data so as to solve many different predictive tasks:

- Forecasting (e.g. Energy demand prediction, finance)
- Imputing missing data (e.g. Netflix recommendations)
- Detecting anomalies (e.g. Security, fraud, virus mutations)
- Classifying (e.g. Credit risk assessment, cancer diagnosis)
- Ranking (e.g. Google search, personalization)
- Summarizing (e.g. News zeitgeist, social media sentiment)
- Decision making (e.g. AI, robotics, compiler tuning, trading)
When to Apply Machine Learning

- Human expertise is absent (e.g. Navigating on Mars)
- Humans are unable to explain their expertise (e.g. Speech recognition, vision, language)
- Solution changes with time (e.g. Tracking, temperature control, preferences)
- Solution needs to be adapted to particular cases (e.g. Biometrics, personalization)
- The problem size is too vast for our limited reasoning capabilities (e.g. Calculating webpage ranks, matching ads to Facebook pages)

Nando de Freitas, Oxford
Machine Learning Pioneering (C. Shannon 1961)
Learning Types

Supervised Learning
- data: \( \{x^{(t)}, y^{(t)}\} \)
- setting: \( x^{(t)}, y^{(t)} \sim p(x, y) \)
- Example
  - classification
  - regression

UnSupervised Learning
- data: \( \{x^{(t)}\} \)
- setting: \( x^{(t)} \sim p(x) \)
- Example
  - distribution estimation
  - dimensionality reduction

Semi-Supervised Learning
- data: \( \{x^{(t)}, y^{(t)}\} \)
- setting: \( x^{(t)} \sim p(x, y) \)
  - \( x^{(t)} \sim p(x) \)
Machine Learning vs Deep Learning

Machine Learning in Practice

- Describing your data with features a computer can understand
  - Domain specific, requires Ph.D. level talent
- Learning algorithm
  - Optimizing the weights on features
Perceptron — Single Neuron element, Rosenblatt 1958
Neural Networks (MLP)

Rule of thumb:
- the number of training examples should be at least five to ten times the number of weights of the network.

Other rule:
\[ N > \frac{|W|}{(1 - a)} \]
- \(|W|\) = number of weights
- \(a\) = expected accuracy on test set
Training schemes (SGD, Batch, MiniBatch)
MLP - Function Approximation

2-layer:

\[ x \in \mathbb{R}^d \rightarrow w^{(1)} \rightarrow g^{(1)}(.) \rightarrow w^{(2)} \rightarrow g^{(2)}(.) \rightarrow y \in \mathbb{R}^n \]

An MLP with single hidden layer can approximate any function.
Deep Motivation - Different Layers of Abstraction

[picture from Simon Thorpe]
Deep Neural Networks

Deep neural networks learn hierarchical feature representations.
Feed Forward Neural Networks

\[ y = g^3(W^3 h^2 + b^3) \]
\[ W^3 \]
\[ h^2 = g^2(W^2 h^1 + b^2) \]
\[ W^2 \]
\[ h^1 = g^1(W^1 x + b^1) \]
\[ W^1 \]
\[ x \]
Training DNNs - Backpropagation
Deep NN - Training Problem

The **back-propagation** encounters the three following difficulties in the training process of deep neural networks:

- Vanishing Gradient- *output error fails to reach the farther back nodes*
- Overfitting
- Computational Load
Vanishing Gradient Solutions
Overfitting – Generalization Problem
Model Complexity

- **Underfit**: High Bias - Low variance
- **Trade-off fit**: Medium Bias - Medium variance
- **Overfit**: Low Bias - High variance
Among competing hypotheses, the one with the fewest assumptions should be selected.

In the related concept of overfitting, excessively complex models are affected by statistical noise (a problem also known as the bias-variance trade-off), whereas simpler models may capture the underlying structure better and may thus have better predictive performance.

**Hoeffding’s inequalities:**

\[
n \geq \frac{1}{2\epsilon^2} \log \frac{2 \cdot 2^{32P}}{\delta} = \frac{1}{2\epsilon^2} \left( \log \frac{2}{\delta} + 32P \log 2 \right)
\]

- Number of sufficient samples
- $|R_{\text{true}}(g) - R_n(g)| \leq \epsilon$
- Failure rate
- True error rate
- Empirical error rate
- # of model parameters

**Occam's razor:** William of Ockham (c. 1287–1347)
Overfitting Solutions – Dropout & Regularization

**Dropout:**
Train some of the randomly selected nodes rather than the entire network \( \approx \text{Regularization} \)

**Rule of thumb:**
50% of hidden layers, and 25% for the input layer

**Regularization:**
Add a norm of the weights to the cost function. (L1-norm, L2-norm)

Data Augmentation is also a way to avoid overfitting; i.e., adding noise, translating data, etc.
AutoEncoder - Nonlinear dimensionality reduction

Decoder
\[ \hat{x} = o(\hat{a}(x)) \]
\[ = \text{sigmoid}(c + W^T h(x)) \]
for binary inputs

Encoder
\[ h(x) = g(a(x)) \]
\[ = \text{sigmoid}(b + Wx) \]
Restricted Boltzmann Machines (RBM)

Energy function: \[ E(x, h) = -h^T W x - c^T x - b^T h \]
\[ = - \sum_j \sum_k W_{j,k} h_j x_k - \sum_k c_k x_k - \sum_j b_j h_j \]

Distribution: \[ p(x, h) = \exp(-E(x, h))/Z \]
\[ = \exp(h^T W x + c^T x + b^T h)/Z \]
\[ = \exp(h^T W x) \exp(c^T x) \exp(b^T h)/Z \]

Hugo Larochelle
Unsupervised Pre-training — another solution

- Train one layer at a time, from first to last, with unsupervised criterion
- Fix the parameters of previous hidden layers
- Previous layers viewed as feature extraction
Convolution Neural Network (Lecun et al. 1993, LeNet)
Convolution NN - Architecture
ConvNet – How it works?

Vincent Vanhoucke - Google
Convolutional NN — (LeCun, Fukushima)

CNN called LeNet by Yann LeCun (1998)

AlexNet - 2012
ConvNet for Speech
CNN Structures

AlexNet
- Input
- Conv
- Conv
- Pool
- Conv
- Pool
- FC
- FC
- Softmax

VGGNet
- Input
- Conv
- Conv
- Pool
- Conv
- Pool
- Conv
- Pool
- Conv
- Pool
- Conv
- Pool
- FC
- FC
- FC
- Softmax

- Image input
- Convolutional layer
- Max-pooling layer
- Fully-connected layer
- Softmax layer
LSTM and RNNs – sequential data
LSTM Training
RNNs & Multi-Hypothesis Tracking- BeamSearch

Vincent Vanhoucke- Google

The quick ?
Captioning & Translation - RNNs

Vincent Vanhoucke - Google
Deep Learning Applications: Computer Vision
DNN Application - Caption Generation

- A car is parked in the middle of nowhere.
- A wooden table and chairs arranged in a room.
- There is a cat sitting on a shelf.
- A ferry boat on a marina with a group of people.
- A little boy with a bunch of friends on the street.

(Kiros, Salakhutdinov, Zemel, TACL 2015)
Wrap Up

- Machine Learning Influence
- Learning Neural Networks
- Deep Learning Motivation, Problems, Solutions
- Unsupervised Neural Networks: AutoEncoders, Restricted Boltzmann Machines
- Deep Learning Training Solutions
- Feed Forward Neural Networks as MLPs
- Convolutional Neural Networks
- Recurrent Neural Networks for sequential data
- LSTMs as a generalization of RNNs
- Applications of DNNs
Thanks for attending the Talk.

Questions?