An Introduction to Deep Learning

Rajiv Sambasivan[1]

[1]Chennai Mathematical Institute

Supervised Learning - Additive Models

- A common modeling approach to supervised learning is an additive model

$$y(x) = \sum_j f_j(x)$$

- Here, $y(x)$ is the response, x represents a vector of predictors

Additive Models

- The intuition is that we can model the response y in terms of additive components $f_j(x)$.
- We do not model combinations of the variables explicitly.
- This approach works well in many situations, has many attractive properties.

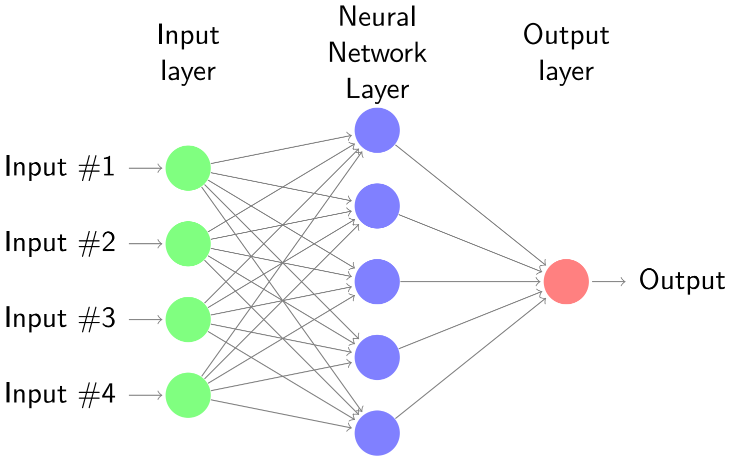
Interactions - combinations of variables

- A complimentary idea is to model the response in terms of combination of the variables.

$$
\begin{aligned}
y(x) = f_1(w_{11}.x_1 + w_{21}.x_2 + \ldots + w_{m1}.x_m) + \\
f_2(w_{12}.x_1 + w_{22}.x_2 + \ldots + w_{m2}.x_m) + \ldots
\end{aligned}
$$

- This idea goes by the name of **projection pursuit** and is useful to build sophisticated supervised learning models (see chapter 11, [Friedman et al., 2001])

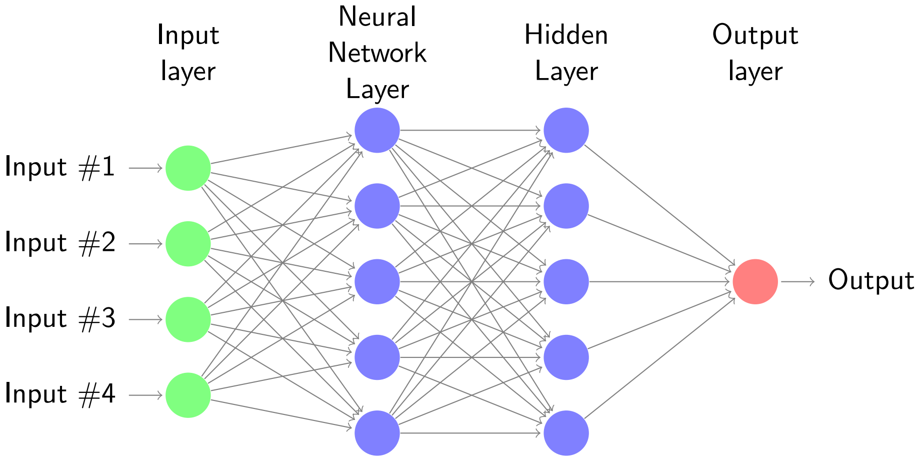
Neural Networks - Projection Pursuit



Neural Networks - Schematic

- Input Layer - captures x
- Neural Network Layer - applies f_i, these are called **activation functions**. Many available - sigmoid, relu etc..
- Output Layer - regress or apply decision function to inputs from the previous layer.

Deep Neural Networks - Schematic



Convex Optimization - A Brief Detour

- Optimization is frequently used to arrive at solutions in machine learning.
- Typical problem template involves a model which has some parameters.
- Optimal model parameters are those that minimize/maximize the **loss** or another **objective** function associated with the solution.
- The **loss** associated with the solution is a modeling decision.

Convex Optimization - Demonstration with Regression and Classification

- We will illustrate the application of convex optimization to determine model parameters with two simple toy examples - one classification and one regression problem.
- The mean square error will be the loss function used in regression and the log-likelihood will be used for the classification problem
- The extension to larger problems is similar - just many more parameters due to model complexity.

Neural Networks - Problem Structure

- Neural Networks are highly parametrized solutions
- Fully connected solutions are not optimal for many practical problems.
- Developing a solution typically requires leveraging special characteristics of the problem.
- If samples occur over a reqular grid like in image data or some engineering applications, a type of neural network called a convolutional neural network (CNN) can be used.
- CNN's reduce the parameter space by using parameter sharing. (see [Goodfellow et al., 2016])

Neural Networks - Problem Structure (contd.)

- Sequences are another example of structure in the problem that is exploited.
- Word sequences, time series etc. are examples of problems with a sequential structure
- This structure is utilized to reduce the parameter space - the number of weights that we have to learn
- The number of layers, the number of neurons per layer, the type of initialization to use with the layer can all affect the effectiveness of the model.

Neural Networks

- The weights are the parameters we need to learn
- Weights are learned using a technique called **back propogation**.
- Basic template involves:
  1. Initialize weights
  2. Evaluate the solution (a.k.a forward pass)
  3. Compute the loss
  4. Update the weights (backward propogation)

Summary

Thanks!

References I

Friedman, J., Hastie, T., and Tibshirani, R. (2001).
The elements of statistical learning, volume 1.
Springer series in statistics New York, NY, USA:.

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016).
Deep learning, volume 1.
MIT press Cambridge.