New York Taxi Fare Challenge

Till Bungert

Abstract

1 Introduction

2 Related Work

3 Methods

3.1 Entity Embeddings

We map categorical variables with C categories represented by indices $c \in \{0, 1, \dots, C-1\}$ to real-numbered vectors $\mathbf{x}_c \in \mathbb{R}^n$

$$\textit{Embedding} : \{0, 1, \dots, C-1\} \to \mathbb{R}^n, c \mapsto \textit{Embedding}(c) = \mathbf{x}_c. \tag{1}$$

These embedding layers are implemented as lookup tables. The vector associated with each index is a parameter of the model and is learnd jointly with the rest of the model.

If the input to our model is a mixture of continuous and categorical variables as is the case here, we learn one embedding layer for each of the categorical variables and concatenate the vector components of each embedding output together with the continuous variables to one vector. This concatenated vector then serves as the input to the rest of the model.

3.2 Neural Networks

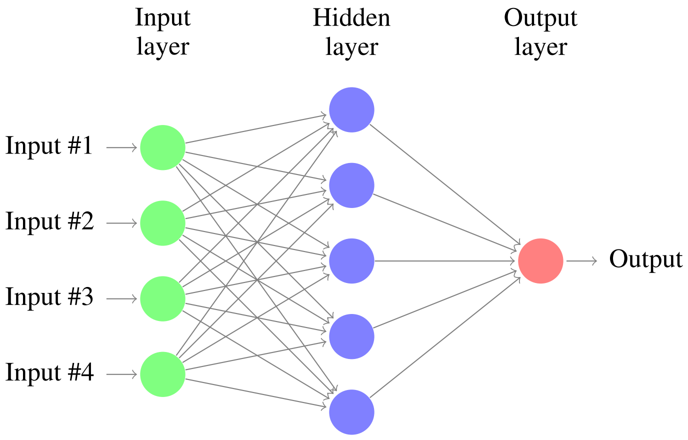

Figure 1: 2-Layer neural network

Feed-forward neural networks, sometimes called multilayer perceptrons, are one of many machine learning models designed to approximate some function f^*. They define a mapping $y = f(x, \theta)$ where θ is learned to result in the best approximation.

The name feed-forward neural network stems from the fact that they consist of intermediary functions called layers, that are chained together. The length of the chain of these intermediary functions gives the depth of the network. As f^*, f and all intermediary functions are vertor valued, the dimensionality of the vector gives the width of the layer. If a layer is not the input or output layer it is called hidden. By depicting each vector component as a node, neural networks can be described by directed acyclic graphs as in Figure 1 [2].

In most cases, each layer consists of a linear function $y = w^T x$, where w are called the weights of that layer and are part of θ. As a chain of linear functions is still a linear function, we need something to make the neural network nonlinear to learn general functions. To acomplish this, each layer is associated with a nonlinear function called the activation function, that is applied to the result of the linear function. Popular choices for activation functions are the rectified linear unit (ReLU)

$$\mathit{ReLU}(x) = \max(0, x) \tag{2}$$

and the softmax [2].

3.3 Gaussian Processes

Gaussian processes are nonparametric [1].

3.4 Tree-based methods

4 Experiments

4.1 Data Set

4.2 Model Setups

4.2.1 Simple Neural Network

4.2.2 Deep Neural Network

4.2.3 Gaussian Processes

4.2.4 Gradient Boosting

4.3 Results

5 Conclusion

References

[1] D. Barber. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.

[2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. `http://www.deeplearningbook.org`. MIT Press, 2016.