UC business Analytics R programming Guide - DL

Koji Mizumura

2019-07-01

Table of Contents

This is a practice of [UC business analytics R programming guide](http://uc-r.github.io/).

# Predictive Analysis

## Artificial Neural Network Fundamentals

Please refer [this page](http://uc-r.github.io/ann_fundamentals) for details.

Artificial neural networks (ANNs) describe a specific class of machine learning algorithms designed to acquire their own knowledge by extracting useful patterns from data. ANNs are function approximators, mapping inputs to outputs, and are composed of many interconnected computational units, called neurons. Each individual neuron possesses little intrinsic approximation capability; however, when many neurons function cohesively together, their combined effects show remarkable learning performance. This tutorial provides an introduction to ANNs and discusses a few key features to consider. Future tutorials will demonstrate how to apply ANNs.

### tl;dr

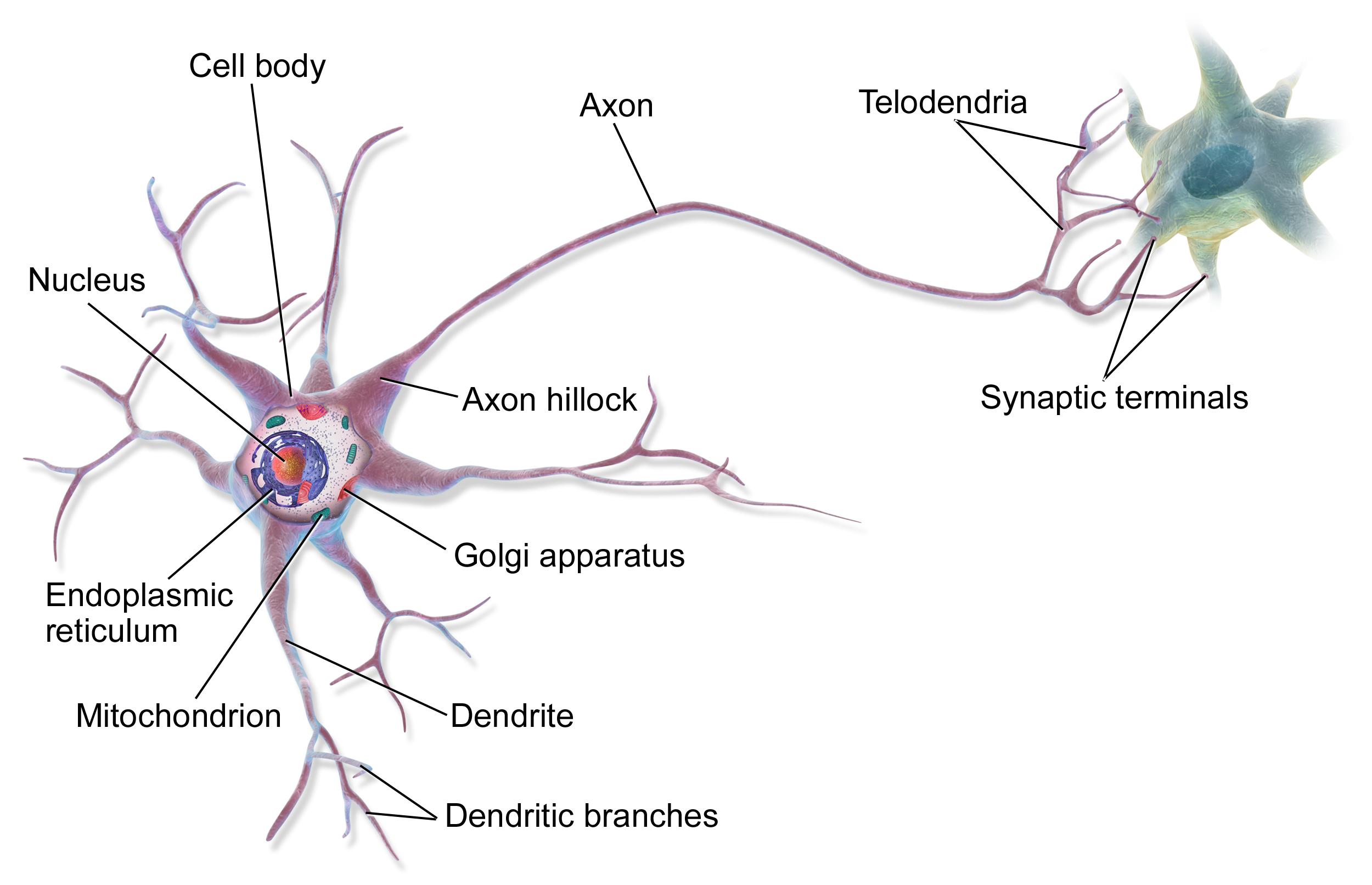
This tutorial provides a high level overview of ANNs, an analytical technique that is currently undergoing rapid development and research. To provide a robust introduction, this tutorial will cover:

1. [Biologic model](#ANN_Biologic_Model): A brief description of the biologic neuron, which ANNs attempt to mimic.
2. [Artificial Neuron](#ANN_Artificial_Neuron): The analytic analog to the biologic model.
3. [Activation functions](#ANN_Act_Func): How the artificial neuron decides to fire.
4. [How ANNs learn](#ANN_Model_Learning): Introducing the back-propagation algorithm.
5. [ANN hyperparameters](#ANN_Param): Dictating how well neural networks are able to learn.

### Biologic model

ANNs are engineered computational models inspited by the brain (human & animal). While some researchers used ANN to study animal brains, most researcheres view neural networks as being inspired by, not models of, neurological systems. Figure 1 shows the basic functional unit of the brain, a biologic neuron.

knitr::include\_graphics("image/Blausen\_0657\_MultipolarNeuron.png")



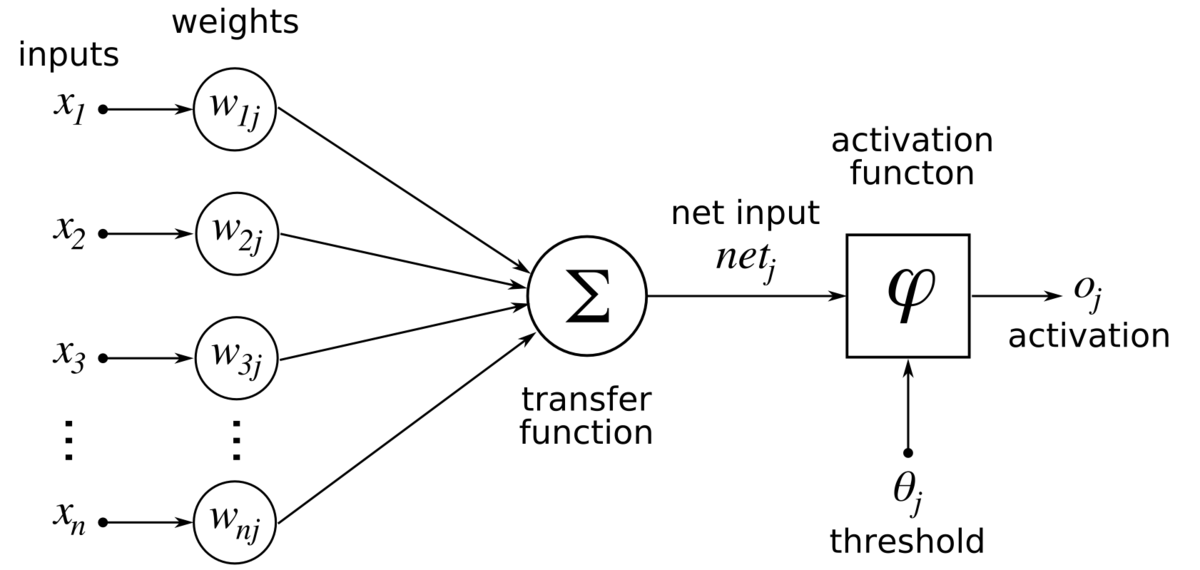
ANN neurons are simple representationsof their biologic counterparts. In thebiologic neuron figure please note the Dendrite, Cell body, and the Axon with the Synaptic terminals. In biologic systems, information (in the form of neuroelectric signals) flow into the neuron through the dendrites. If a sufficient number of input signals enter the neuron through the dendrites, the cell body generates a response signal and transmits it down the axon to the synaptic terminals. The specific number of input signals required for a response signal is dependent on the individual neuron. When the generated signal reaches the synaptic terminals neurotransmitters flow out of the synaptic terminals and interact with dendrites of adjoining neurons. There are three major takeaways from the biologic neuron:

1. The neuron only generates a signal if a sufficient number of input signals enter the neurons dendrites (all or nothing)
2. Neurons receive inputs from many adjacent neurons upstream, and can transmit signals to many adjacent signals downstream (cumulative inputs)
3. Each neuron has its own threshold for activation (synaptic weight).

### Artificial Neouron

The artificial analog of the biologic neuron is shown below in figure 2. In the artificial model the inputs correspond to the dendrites, the **transfer function**, **net input**, and **activation function** correspond to the cell body, and the **activation** corresponds to the axon and synaptic terminal.

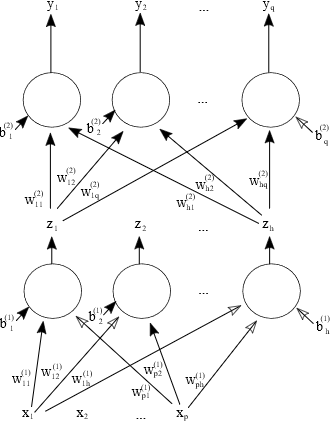
knitr::include\_graphics("image/1200px-ArtificialNeuronModel\_english.png")



The inputs to the artificial neuron may correspond to raw data values, or in deeper architecures, may be outputs from preceding artificial neurons. The transfer function sums all theinput together (cumulative inputs). If the summed input values reach a specified threshold, the activation function enerates an output signal (all or nothing). The output signal then moves to a raw output or other neurons depending on depending on specific ANN architecture.

This basic artificial neuron is combined with multiple other artificial neurons to create an ANN such as the ones shown in figure 3.

knitr::include\_graphics("image/Multilayer\_Feedforward\_Artificial\_Neural\_Network.png")



ANNs are often described as having an **Input** layer, **Hidden** layer, and **Output** layer. The input layer reads in data values from a user provided input. Within the hidden layer is where a majority of the “learning” takes place, and the output layer displays the results of the ANN.

In the bottom plot of the figure, each of the red input nodes correspond to an input vector . Each of the black lines with correspond to a weight , and describe how artificial neurons are connections to one another with the ANN.

The subscript identifies the source and the subscript describes to which artificial neuron the weight connects the source to. The green output nodes are the output vectors

Examination of the figure’s top-left and top-right plots show two possible ANN configurations. In the top-left, we see a network with one hidden layer with artificial neurons, input vectors and generates output vectors . Please note that the **bias** inputs to each hidden node, denoted by the . The bias term is a simple constant valued 1 to each hidden node acting akin to the grand mean in a simple linear-regression.

Each bias term in a ANN has its own associated weight . In the top-right ANN we have a network with two hidden layers. This network adds superscrip5vnotation to the bias terms and the weights to identify which layer each term belongs. Weights and biases with a superscript 1 act on connecting the input layer to the first layer of artificial neurons and terms with a superscript 2 connect the output of the second hidden layer to the output vectors.

The size and structure of ANN are only limited by the imagination of the analyst.

### Activation Functions

The capability of ANNs to learn *appriximately* any function(given sufficient training data examples) are dependent on the appropriate selection of the **Activation Function(s)** present in the network. Activation functions enables the ANN to learn non-linear properties present in the data.