Artificial Neural Networks

Mitchell Corbett and Matthew Galbraith

April 7, 2015

Outline

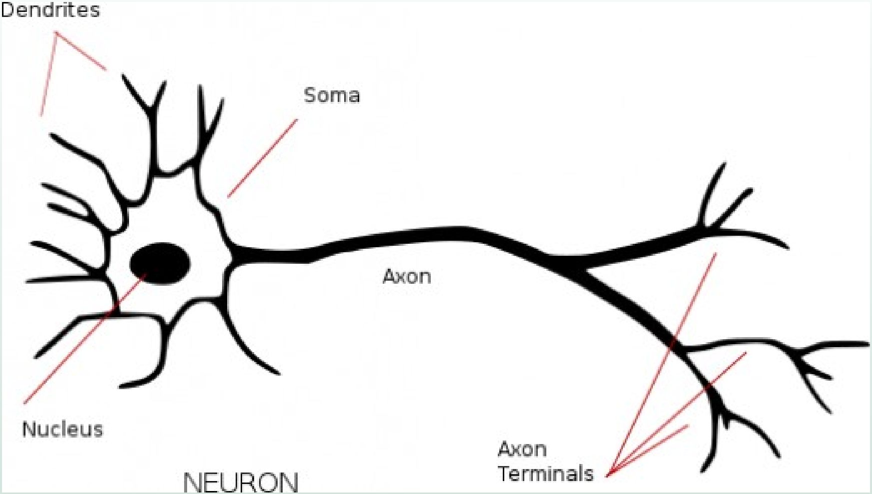
Outline

What is a Neural Network?

- What is a Neural Network?

Biological Inspirations

- In Biology, the Neuron is a fundamental unit of the Central Nervous System. It contains 3 parts. The Cell Body, Dendrites, and an Axon.
  - The Cell Body contains the essential parts of the cell
  - Dendrites are short fibers that receive signals.
  - The Axon is a long projecting branch which sends signals. It can branch in to many other cells.


Dendrites
Soma
Axon
Nucleus
Axon
Terminals
NEURON

DARPA NN Study

- "...a Neural Network is a system composed of many simple processing elements operating in parallel whose function is determined by network structure, connection strength and the processing performed at computing elements or nodes"

Haykin (1994)

- "A neural network is a massively parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for use."

Applications

Classification Problems:

- Character Recognition
- Labelling images

Regression:

- Recognize facial keypoint
- Image Processing

- With that, let's dive in!

What the heck is a Neuron

- We need to discuss how we can model a Neuron computationally.
- A basic neuron has 3 key components. **Synapses**, an **Adder**, and an **Activation Function**.

Components of a Basic Neuron

Synapses

Synapses or "Links" each have a weight assigned to them. If we have an input signal coming into a neuron, we will want to weight them differently.

Adder

Used for summing the input signals which have been weighted. This is also called a Summing Junction.

Activation Function

Used to limit the output of a neuron. Essentially, the activation function requires that a sufficiently strong output is reached. This is very similar to the neurons in our brains, which fire or activate in response to a change in electrical charge.

Activation Functions

There are a few different choices for activation functions. Certain ones will lead to different models.

Threshold function

A function that is 0 or 1 depending on the sign of the input.

Piecewise linear

An approximation of non-linear behavior:

$$\phi(v) \begin{cases} 1 & v \geq \frac{1}{2} \\ v & \frac{1}{2} > v > -\frac{1}{2} \\ 0 & v \leq -\frac{1}{2} \end{cases}$$

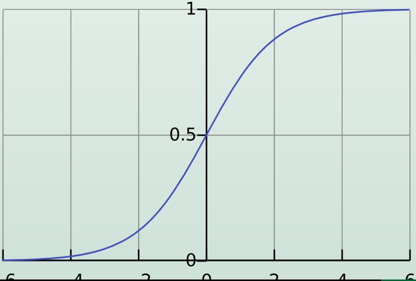
Activation Functions

Sigmoid Function

The typical choice for activation. It has the desirable property of being continuous and strictly increasing, making it a good fit for an activation function.

$\phi(v) = \frac{1}{1+exp(-v)}$

$tanh(v_i)$ is a similar function that fits this description.



Other Components of a Basic Neuron

We can introduce a parameter to the sigmoid function, to control the steepness, but what if we wish have a stronger output.

Bias

The Bias increases the net output of the activation function, effectively shifting the output.

Neuron Output

Given the inputs, and defined weights,

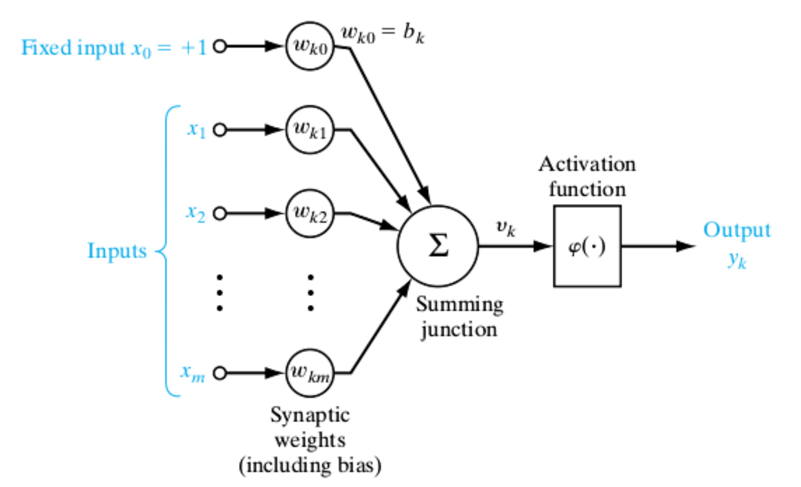
$$v_k = \sum_{j=1}^{m} w_{kj} x_j$$

is the linear combination of weighted inputs, as determined by the Adder, or Summing Junction.

$$y_k = \phi(v_k + b_k)$$

is then the outgoing signal from the neuron where b_k is the bias.

A basic Neuron



Single Layer Perceptron

- We now have the terminology to examine our first Neural Network! The Single Layer Perceptron! Yay!
- A Single Layer Perceptron (SLP) is a binary classifier that maps an input to a binary value.
- SLP is a single neuron, all on its own, using the Threshold function for activation.

Single Layer Perceptron

Mathematically, we define the SLP (Single-Layer Perceptron as follows:

SLP

The SLP is a function from $R^n - > \{0, 1\}$ such that:

$$f(x) \begin{cases} 1 & \mathit{if}\ w \cdot x + b > 0 \\ 0 & \mathit{otherwise} \end{cases}$$

Where w is the weight vector, $\cdot$ refers to the dot product and b is the bias.

Single Layer Perceptron - Learning

Definitions/Terminology

- $y = f(\mathbf{x})$ is the output
- $D = (x_1, d_1), \ldots, (x_s, d_s)$ is the training set of s samples where x_i is the input vector, and d_j is the desired output.
- As part of the input, we will represent the bias as a constant input , so $x_{j,0} = 1$ which has a corresponding weight, so we can train it.
- $w_i(t)$ is the weight vector corresponding to the ith input at time t.
- Also define a learning rate $0 < \alpha \leq 1$ which will dictate how much our weights can change with each update.

SLP Convergence

- The data in question must be linearly separable. By this we mean the existence of a hyperplane "Decision boundary" in which the two classes can be split apart.
- This plane will be where

$$\Sigma w_i x_i + b = 0$$

- The challenge is determining what the weights should be for the decision boundary to be correct.
- It's easy to see that this model is incapable of describing a non-linear decision boundary

SLP Convergence

- In fact, if the data is linearly separable, then it is guaranteed to converge.
- There is also an upper bound to the number of weight updates.

$$O(R^2/\gamma^2)$$

Where R is the maximum norm of an input vector, and γ is the margin of the hyperplane

References

- http://www.teco.edu/ albrecht/neuro/html/node16.html
- Haykin, S. (1999), Neural Networks: A Comprehensive Foundation, NY: Macmillan.
- Novikoff, A. B. (1962). On convergence proofs on perceptrons. Symposium on the Mathematical Theory of Automata, 12, 615-622. Polytechnic Institute of Brooklyn.
- DARPA Neural Network Study (U.S.). 1988. DARPA Neural Network Stdy. Widrow, Morrow, and Gschwendtner (Eds.). AFCEA Intl.

Smooth Transition

And with that... It's off to Matt!