A Very Brief Introduction to Neural Networks and Deep Learning

Ashley Lee, Isabel Restrepo, & Paul Stey

December 7, 2017

Table of Contents

1 Background
- Neural Network Basics
- Neural Network Basics
- History
- Mechanics of Neural Networks

2 Species of Neural Networks
- Convolutional Neural Network
- Recurrent Neural Network
- Advantages of Neural Networks
- Disadvantages of Neural Networks
- Perceptron

3 Applications of Neural Networks
- Computer Vision
- Natural Language Processing
- Speech Recognition

4 Summary
- When to use Neural Networks

What is a neural network?

1. A species of directed acyclic graphs (usually)
2. “Universal function approximator”
3. “An engineering solution to a statistics problem”

What do neural networks do?

Like many other statistical or machine learning models (e.g., GLM, random forests, boosting), neural networks:

1. Attempt to approximate a data-generating mechanism
2. Can be used for classification problems
3. Can be used for regression problems
4. Can also be used for dimension reduction like principal components analysis (PCA)

History of Neural Networks

The history of neural networks is long and tumultuous.

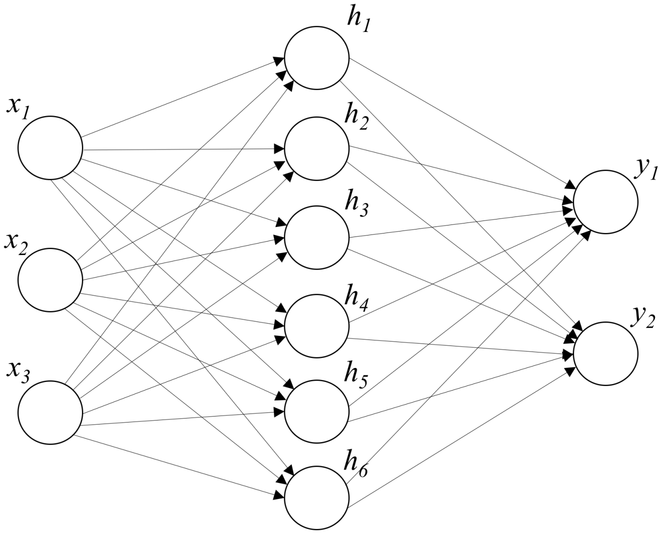
1. McCulloch and Pitts (1943) "*A Logical Calculus of Ideas Immanent in Nervous Activity*"
2. Rosenblatt (1958) "*The Perceptron: A Probabilistic Model For Information Storage And Organization In The Brain*"

More Recently

Neural networks are experiencing a major resurgence. There are at least two reasons.

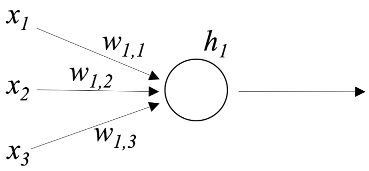
1. Better algorithms for back-propagation
2. GPUs are well suited to building neural networks
   - Matrix multiplies can be made embarrassingly parallel
   - GPUs have much better memory bandwidth
3. More labeled data

The Multilayer Perceptron



Single Neuron

A single neuron takes inputs, x_j, and applies the weights, $w_{\cdot j}$ to the input by computing the dot product of the vectors x and w. The result is the input to the "activation" function.



Activation Functions

The notion of an activation function comes again from the theoretical relationship to neurons in the brain.

Activation functions are analogous to “link” functions in generalized linear models (GLMs).

In fact, one common activation function is the sigmoid function, which is just our old friend the logistic function which you are using when you fit logistic regression models.

Purpose of Activation Functions

There are a few reasons we use activation functions.

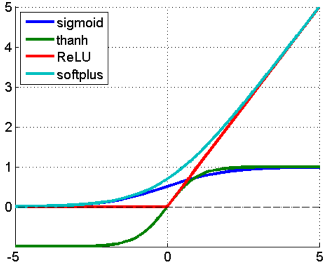
The most basic reason is that—like link functions in GLMs—we want to take some linear predictor and transform it so that it is bounded appropriate. For instance, the value of logistic function is in the range $(0, 1)$.

And the second key reason is that this allows us to introduce non-linearities. Recall that a neural network (like many statistical or machine learning models) is trying to approximate a data-generating mechanism. So we are trying to approximate a function that might be very complex and include many non-linearities.

Common Activation Functions

Some common activation functions include the following:

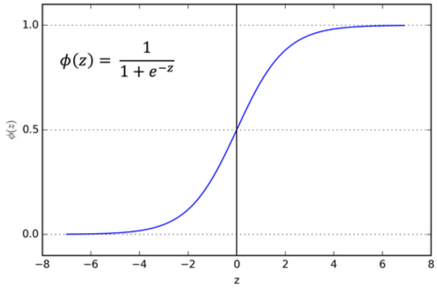
1. Sigmoid (i.e., logistic)
2. Hyperbolic tangent: $tanh$
3. Rectified linear unit (ReLU)
4. softplus



Sigmoid Functions

Sigmoid function: $\phi(z) = \frac{1}{1+e^{-z}}$

1. Range between 0 and 1
2. Sometimes interpreted as probability
3. Special case of the softmax: $\psi(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$, which is used for vector-value outcome

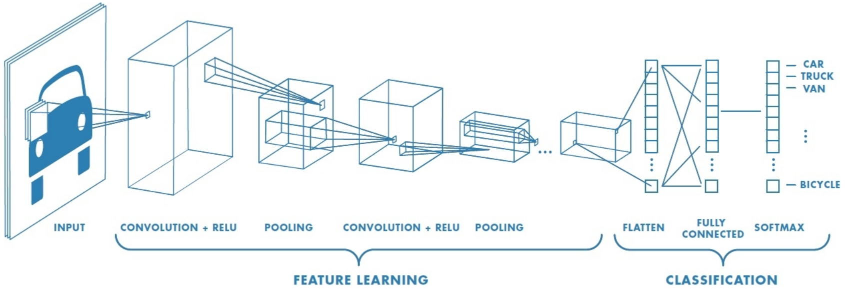


Activation Functions

Of these many activation functions, ReLUs—or some variation thereof—tend to be the most common.

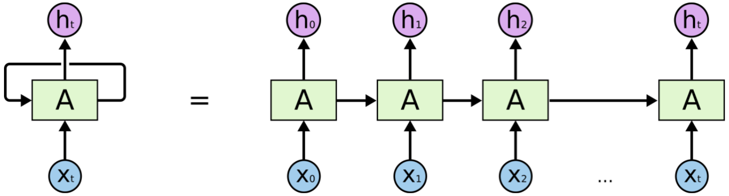
Sigmoid (and probably $tanh$) are not frequently used in the hidden layers more because of the vanishing gradient problem (discussed later).

Convolutional Neural Networks



Source: https://www.mathworks.com/discovery/convolutional-neural-network.html

Recurrent Neural Networks



When to use neural networks

Conditions under which you might consider using neural networks:

1. Have a huge amount of labeled training data
2. Image classification (with huge amount of labeled images)
3. Certain NLP tasks
4. Some signal processing problems

When *not* to use neural networks

Probably should **not** use neural networks when:

1. You have specific hypotheses you want to test
   - E.g., "*Drug X improves condition Y*".
2. Interested in estimating the effect of some variable(s) on some outcome variable
3. You have highly structured data and/or few features

In the case of (1.) and (2.), a traditional statistical model is better. In the case of (3.), using some ensemble-of-trees method will give as-good or better results with minimal tuning.