Introduction to neural networks

Glenn Bruns
CSUMB

Much material in this deck from Géron, Hands-on Machine Learning with Scikit-Learn and TensorFlow

Learning outcomes

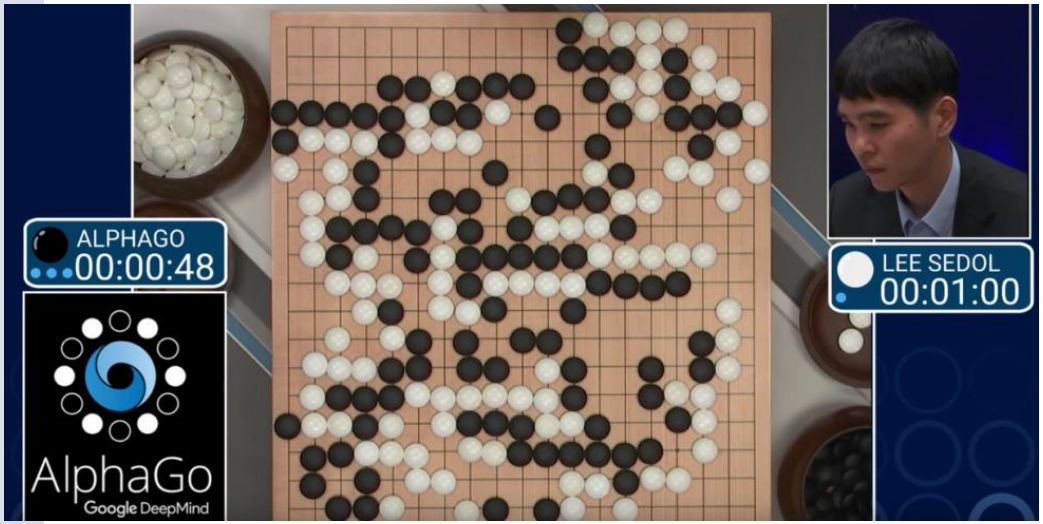
After this lecture you should be able to:

1. Define artificial neuron, perceptron, linear threshold unit, multi-layer perceptron, deep neural network, activation function
2. Define the logistic, hyperbolic, and ReLU activation functions
3. Explain the perceptron learning rule
4. Apply the softmax activation function

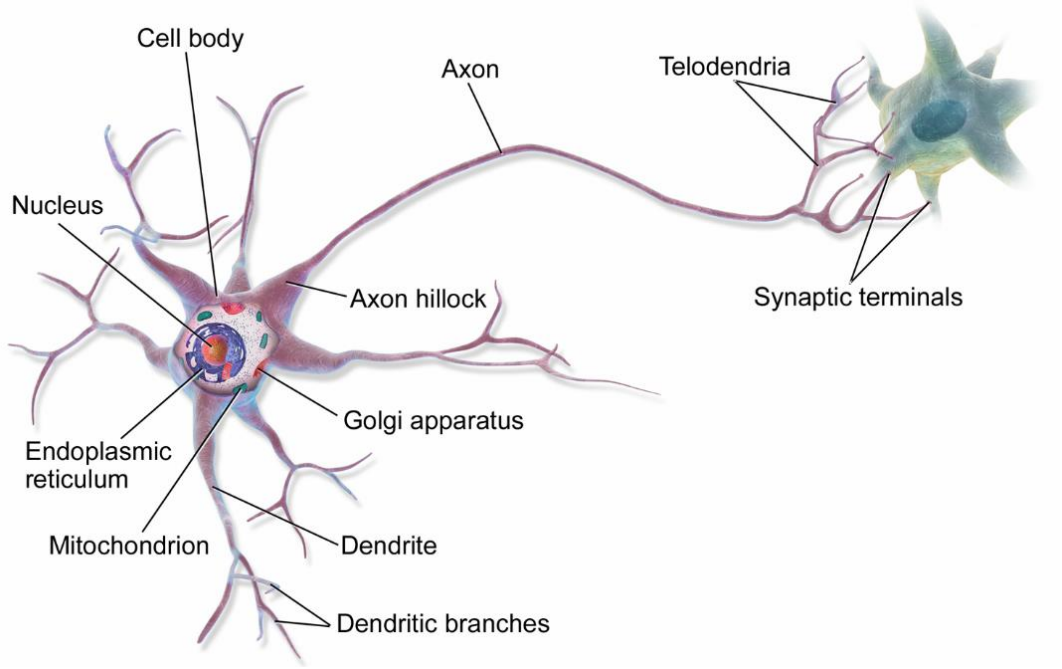
Neural networks and Deep Learning

Deep Learning is a model/technology that has enabled great leaps in machine learning.

It has especially helped on hard problems like image classification, speech recognition, and expert-level Go.



How does the brain work?



a biological neuron

Your brain contains a huge network of interconnected neurons.

Neurons receive signals from other neurons.

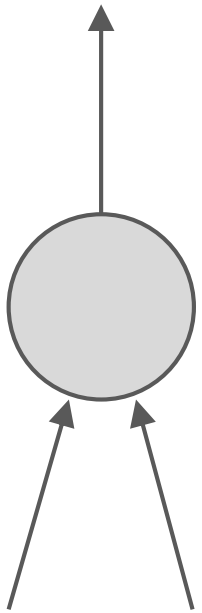
When a neuron receives enough signals from "input" neurons within a short period, it fires its own signal.

Artificial neurons

A first mathematical model of neurons, by McCulloch and Pitts.

artificial neuron:

- one or more binary inputs
- one binary output
- output activated when a certain number of inputs are active



example:

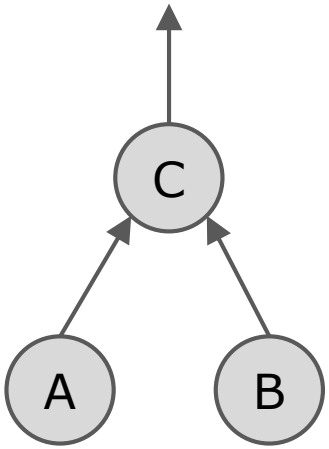
- two inputs, one outputs
- activate output when both inputs active

Logical computation with neurons

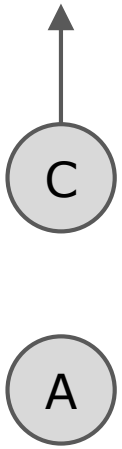
Question: how "powerful" is an artificial neuron?

E.g., can it model logical operations, like 'and' and 'or'?

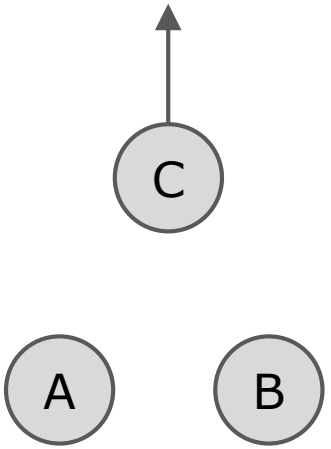
(let's use artificial neurons in which both inputs must be active for output to be activated)



C = A ^ B



C = A ?

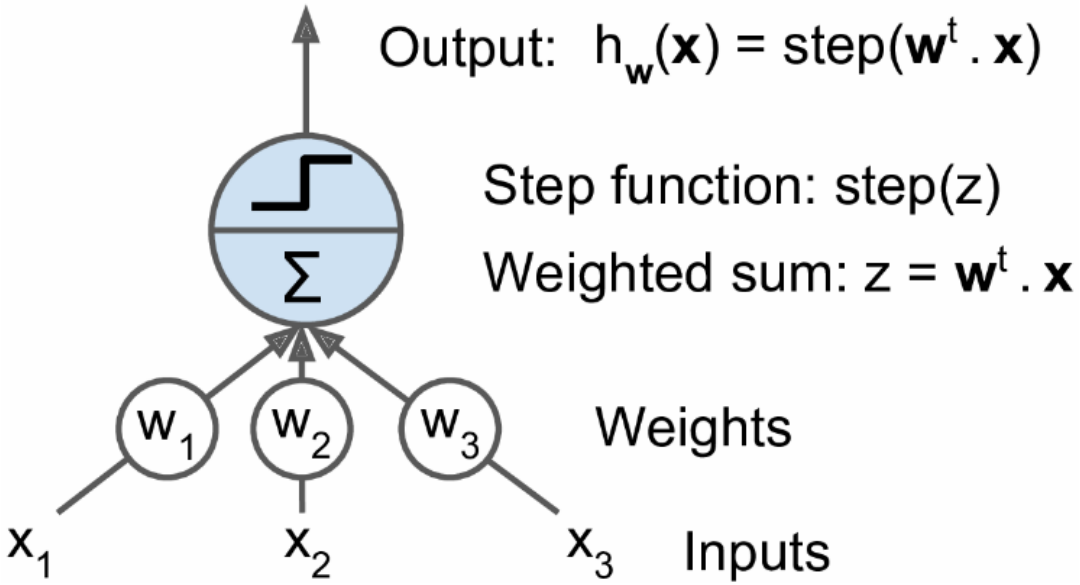


C = A V B ?

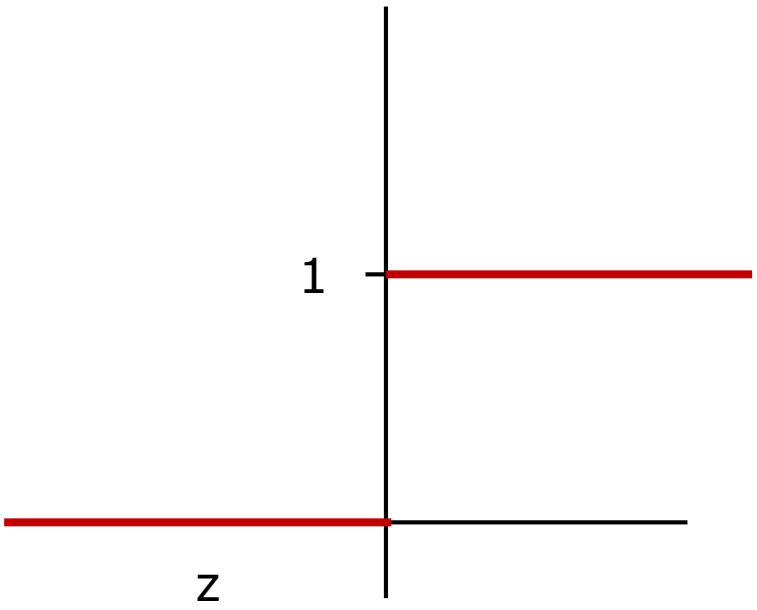
Linear Threshold Unit LTU

An LTU is another kind of artificial neuron.

- It has numeric, not boolean, inputs and outputs.
- It computes a weighted sum of its inputs.
- The output is a step function applied to the weighted sum.



Step functions



two common step functions:

$$\text{heaviside}\,(z) = \begin{cases} 0 & \text{if } z < 0 \\ 1 & \text{if } z \geq 0 \end{cases}$$

$$\text{sgn}\,(z) = \begin{cases} -1 & \text{if } z < 0 \\ 0 & \text{if } z = 0 \\ +1 & \text{if } z > 0 \end{cases}$$

Training an LTU involves finding good values for the w coefficients.

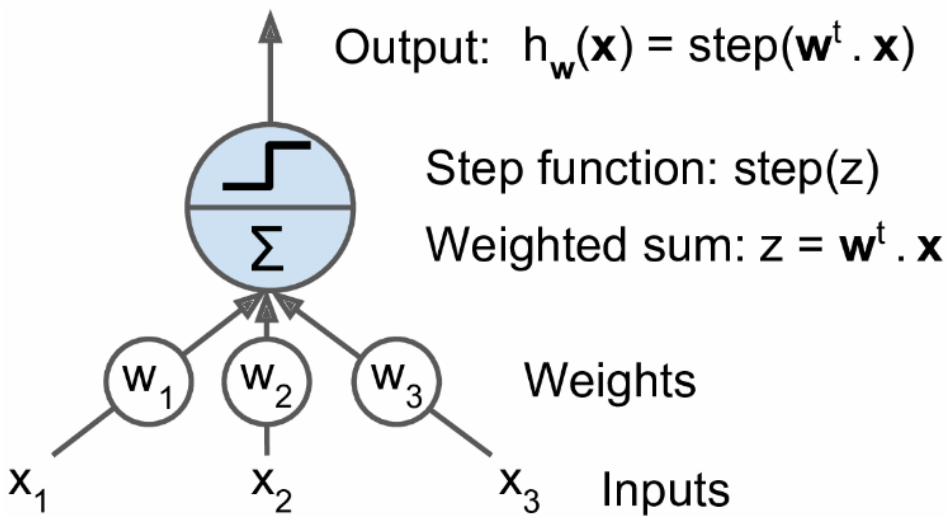
Training an LTU

Idea: reinforce connections that lead to the correct output.

LTU is fed on training instance at a time.

For each instance it makes a prediction.

When a prediction is wrong, connection weights from the inputs are adjusted.

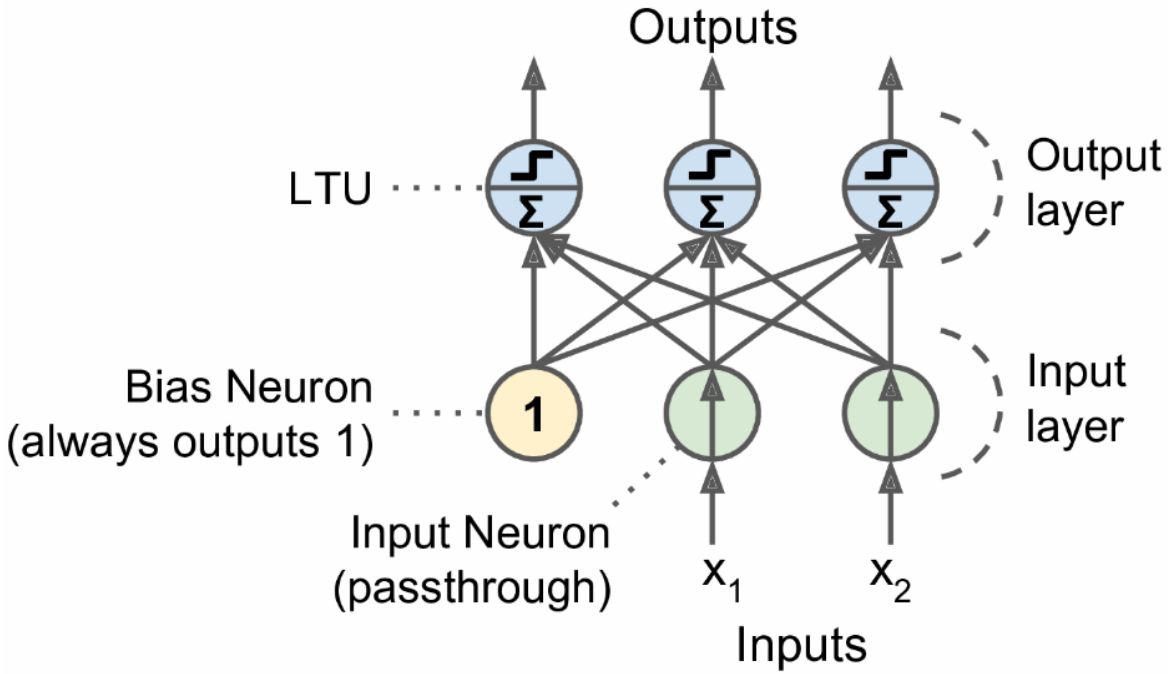


Assume step function is 'sign' function.
Assume input x_1 is a bias term.

```
predict = sgn(w.dot(x))
if predict != y:
   w = w + eta * y * x
```

Perceptrons

A perceptron is a single layer of LTUs, with each neuron connected to all the inputs.



The figure shows a perceptron with 2 inputs and 3 outputs.

The "input neurons" output whatever input they are fed.

Multi-Layer Perceptron

Interest in perceptrons dwindled in 1969.

- Minsky and Paper showed that they could not solve some simple problems
- for example, modeling XOR (exclusive or)

AI research moved to other things like logic, problem solving, and search.

But – stacking perceptrons gives them more power.

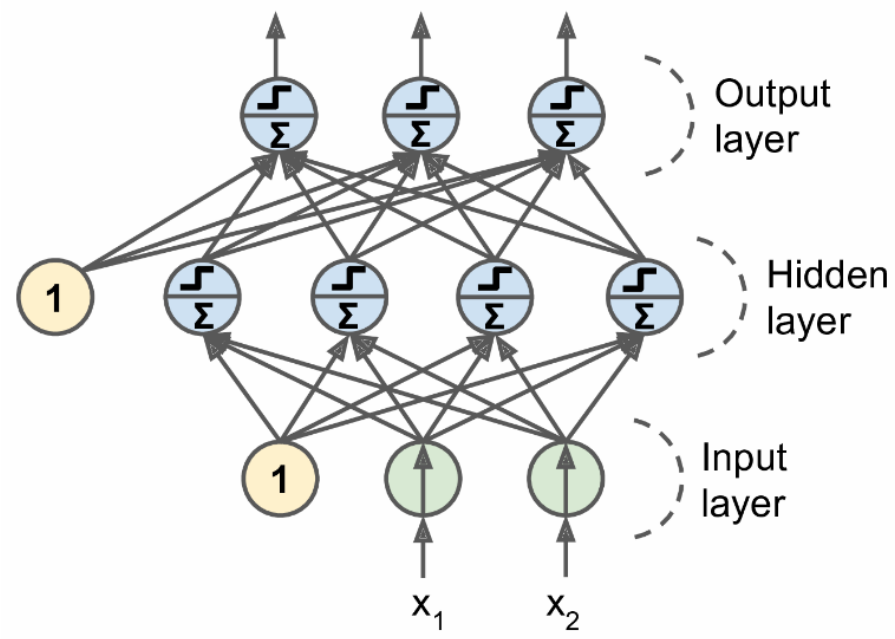
Multi-Layer Perceptron

A multi-layer perceptron has:

- one input layer
- one or more layers of LTUs (hidden layers)
- one final layer of LTUs (output layer)

Every layer but the output layer:

- includes a bias neuron
- is fully connected to the next layer



Training an MLP

In 1986, Rumelhart *et al* introduced the backpropagation training algorithm for MLPs.

It can be described as "gradient descent using reverse-mode autodiff".

Idea of backprop:

- for each training instance, first make a prediction (forward pass)
- measure the error
- go through each layer in reverse to measure the error contribution of each connection
- tweak the connection weights to reduce the error

Activation functions

Key to backprop:

- instead of step function, use logistic function:

$$\boldsymbol{\sigma(z)} = \frac{1}{1+\exp(-z)}$$

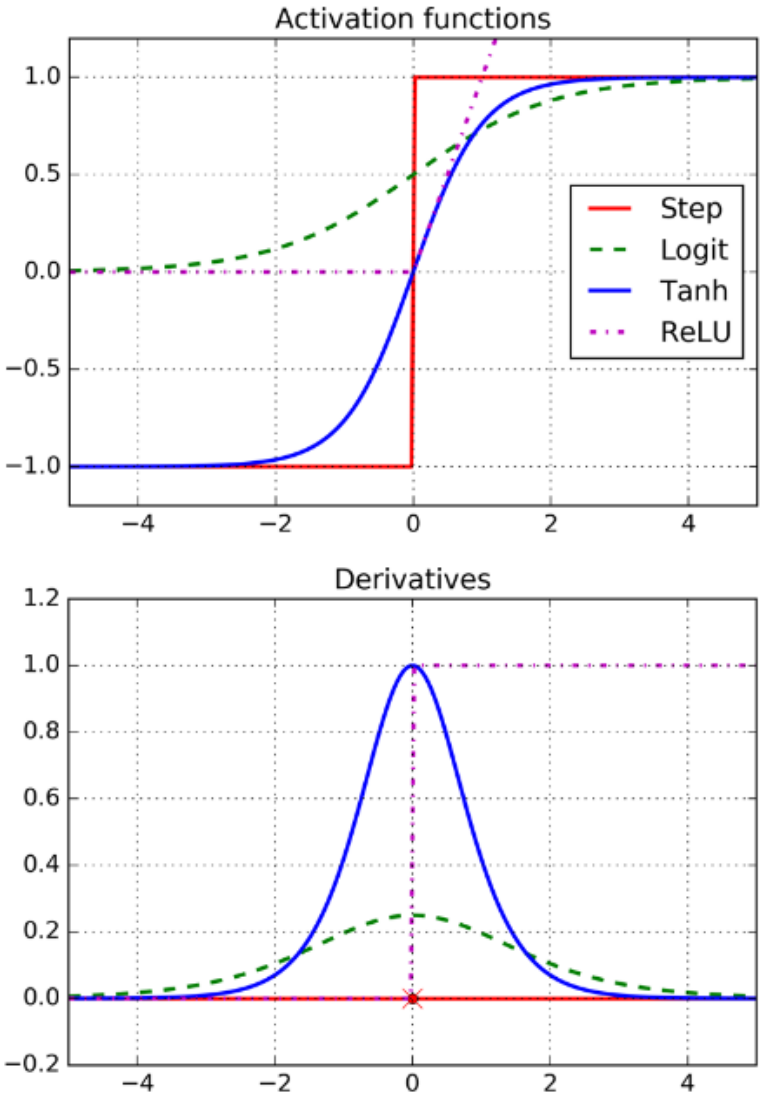
This is called an activation function. Other popular activation functions:

- hyperbolic tangent function:

$$\mathbf{tanh}(\boldsymbol{z}) = 2\sigma(2z) - 1$$

- ReLU function:

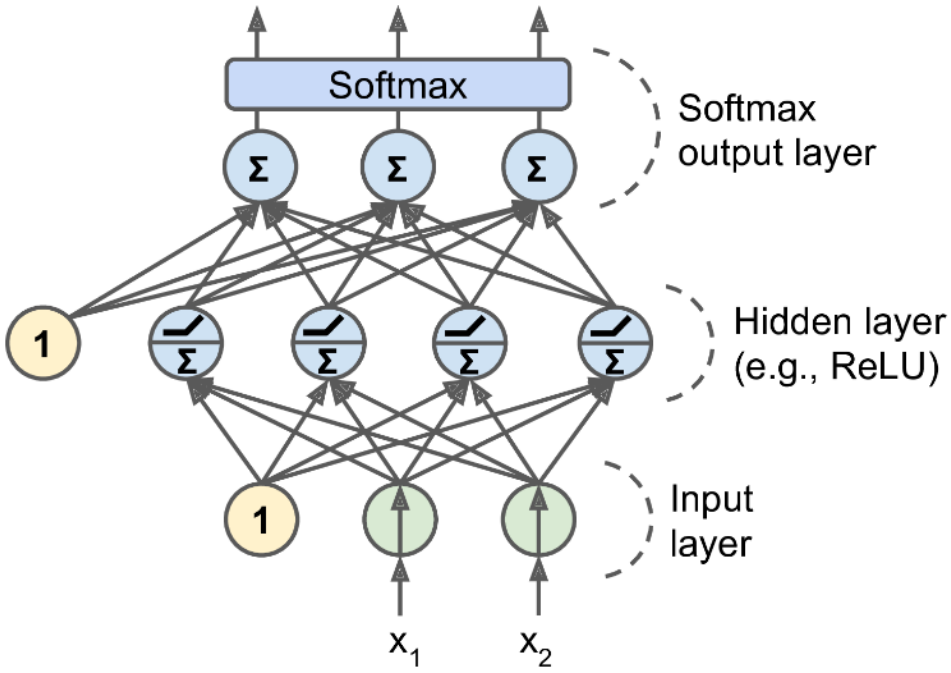
$$\boldsymbol{ReLU(z)} = \max(0, z)$$



MLP and classification

MLPs are often used for classification.

If classes are exclusive, the output layer is changed by replacing the individual activation functions with a single, shared softmax function.



softmax

$$\sigma(\mathbf{z})_j = \frac{\exp(z_j)}{\sum_{k=1}^{K} \exp(z_k)}$$

The outputs now correspond to probabilities for each class

Summary

- We've covered a bunch of basic concepts in neural nets, including:
  - linear threshold unit (LTU)
  - step function
  - perceptron
  - multi-layer perceptron (MLP)
  - activation function
- We've also discussed training of LTUs, perceptrons, and MLPs