Neural networks and deep learning

Status	
Link	http://neuralnetworksanddeeplearning.com/chap1.html
ee	

Humans have visual cortex which contains millions of neurons

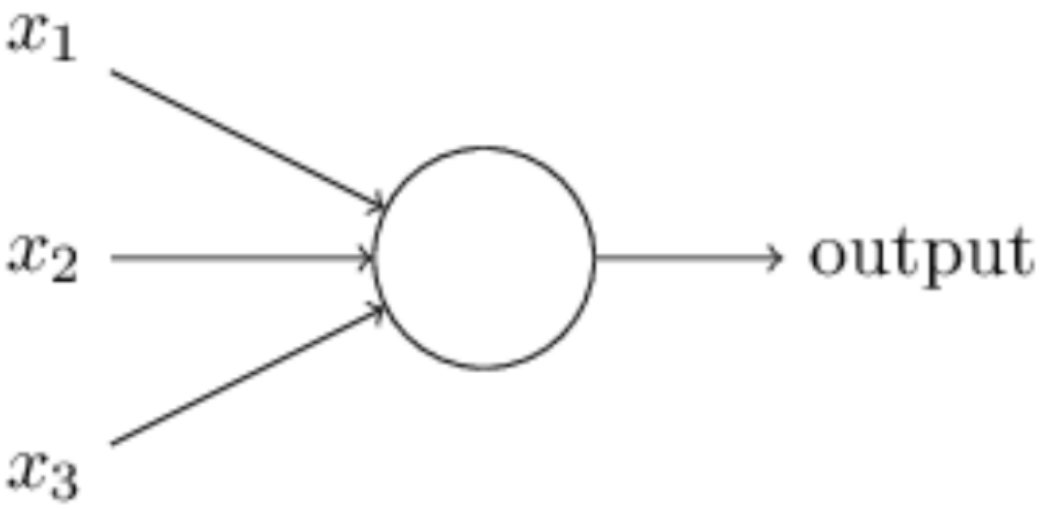
- Really hard problem to read hand digits with an algorithm
  - too many exceptions and special cases
- Machine learning takes a large data set (training data)
- “develops a system which can learn from those training examples”
- Hand writing recognition is good! used by banks

Perceptrons

- artificial neurons
- One artificial neuron, called sigmoid neuron

How do perceptrons work?

Takes a bunch of binary inputs, and produces a single binary output

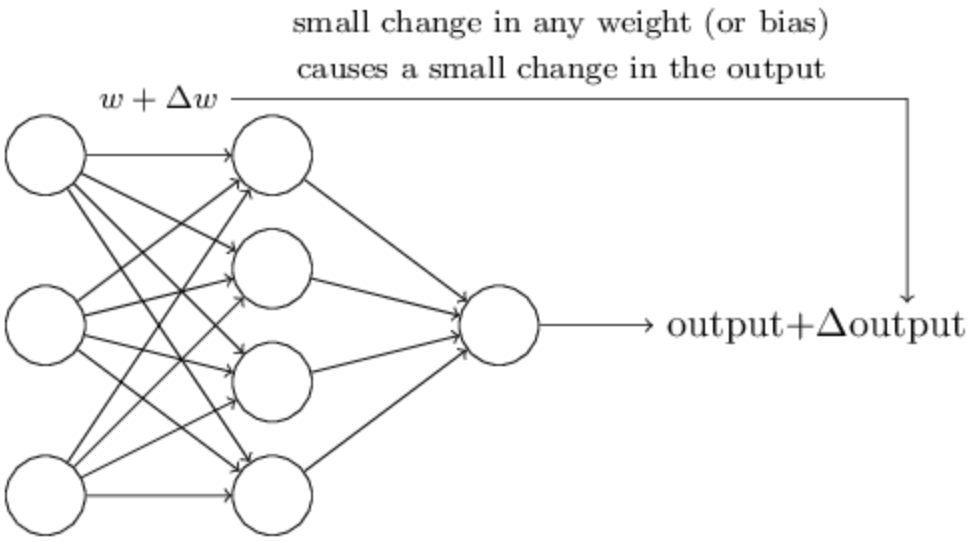


- Assign each input a weight, w_n. Let's call each input x_n
- The output is the sum of $\sum_i^n w_i x_i$ for each perceptron
- The weight makes its corresponding input more/less important than other inputs
- If the weighted sum is $\geq$ some threshold value, output 1
- If the weighted sum is $<$ the same threshold value, output 0
- This is the same as the dot product!
  - $\sum_i^n w_i x_i = w \cdot x$
- To simplify math in the future we define a new value
- BIAS
  - $b \equiv -\text{threshold value}$.
  - $w \cdot x - b \geq 0$ output 1
  - $w \cdot x - b < 0$ output 0
- What makes perceptrons different from NAND gates?

    - Not entirely sure how NAND gates are universal to computations, will look into this
- Perceptrons can tune the weights and biases without programmers doing it manually

Sigmoid Neurons

If you change the weights in the perceptrons, there will be a change in the output



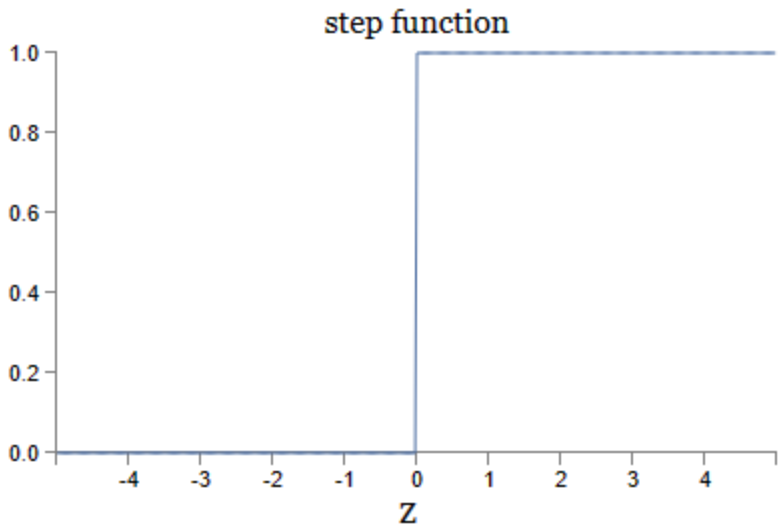
- This is wanted with machine learning
    - i.e. tune the weights and biases so the network is able to more accurately differentiate between 2 and 5
- The question is: how much to adjust the weights and biases?
- Here comes the… SIGMOID NEURON
- Same as perceptrons except that small changes in weights + biases correspond to small changes in the output
- Instead of outputting a 0 or 1, output $\sigma(w \cdot x + b)$, where σ is a function.
- We define the σ function, called the sigmoid function, s.t. $\sigma(z) = \dfrac{1}{1 + e^{-z}}$
- Here's it written explicitly (practicing my LaTeX)

$$\sigma(w \cdot x - b) = \frac{1}{1 + \exp(-\sum_{i=1}^{n} w_i x_i - b)}$$

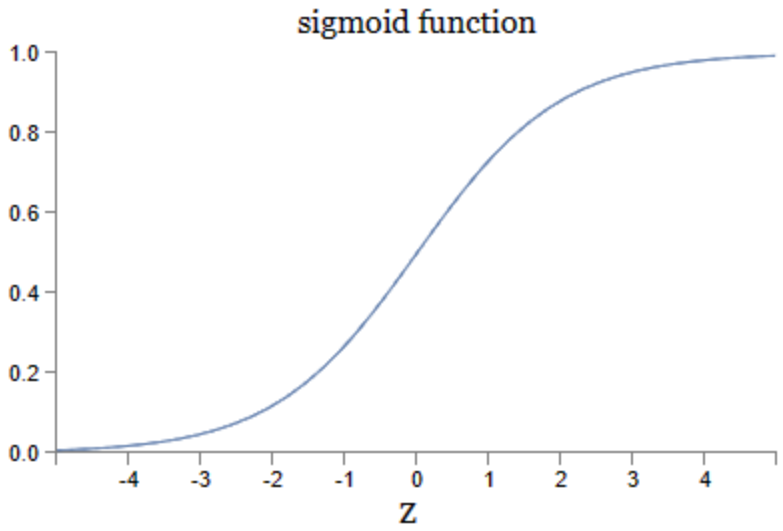
- ^ is for n inputs. The book I'm learning this from adopts a more elegant notation:
- $\sigma(w \cdot x - b) = \frac{1}{1 + \exp(\sum_j w_j x_j - b)}$

Where is this math stuff coming from?

- Let's model the perceptron in an input/output graph



- It can only output a 0 or a 1
- The sigmoid function is a smoothed out step function



Suppose you change the bias term b by some amount Δb, and the weight coefficients by Δw_i, where w_i is the ith weight. Then, the change in output is:

$$\Delta \text{output} \approx \sum_i \frac{\partial \text{ output}}{\partial w_i} \Delta w_i + \frac{\partial \text{ output}}{\partial b} \Delta b$$

(I wrote more on this in my linear algebra notebook. Check it out if confused in the future!)

- also quick note about the activation function chosen. Sigmoid function activation function is used because it is easy to differentiate.

Mid chapter exercises

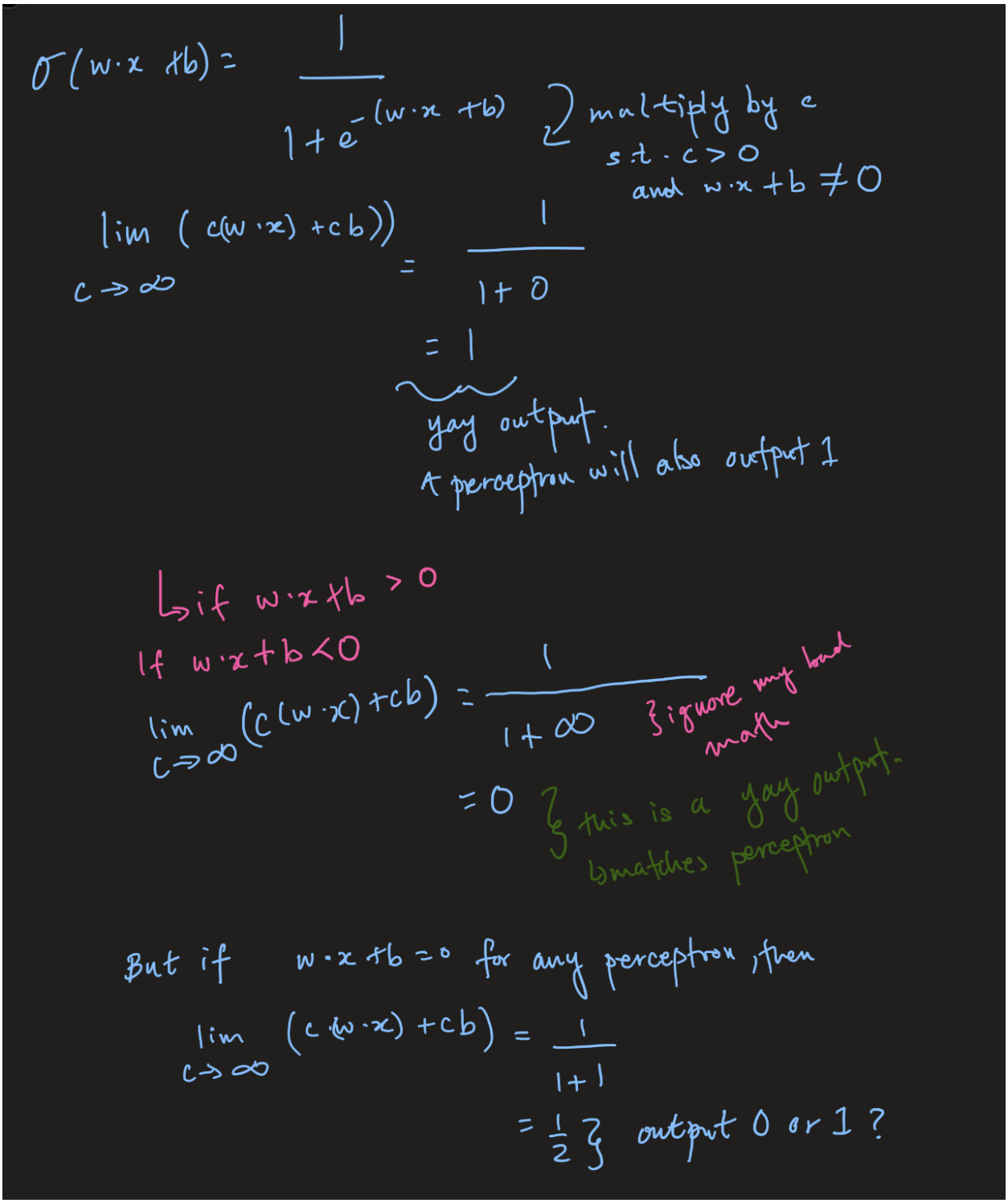
- **Sigmoid neurons simulating perceptrons, part I**

  Suppose we take all the weights and biases in a network of perceptrons, and multiply them by a positive constant, $c > 0$. Show that the behaviour of the network doesn't change.

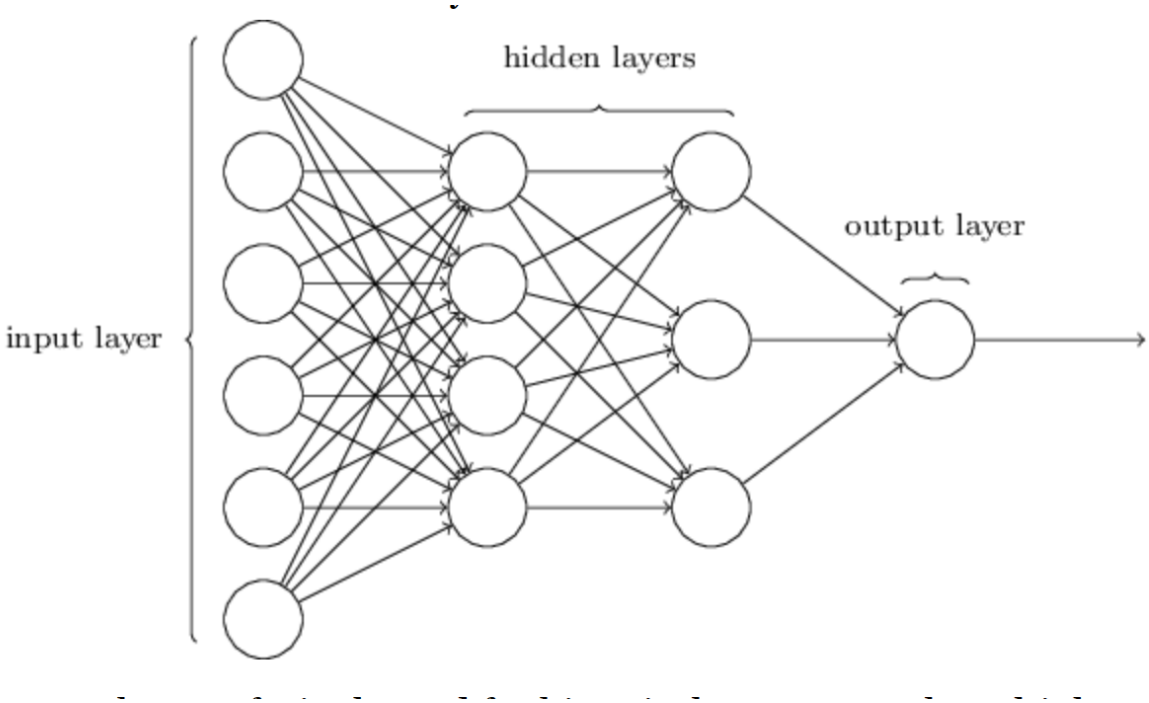
- what does behaviour of the network mean? I'm assuming it has to do with how the activation function works. In that case $\Delta \text{ output}$ would be $c\Delta\text{output}$.
- <actual solution>
- output is 0 if $w \cdot x + b \leq 0$. Multiply everything by c, and it outputs 0 iff $cw \cdot cx + cb \leq 0$. This is the same equation (just divide everything by c!). You can give the same argument for when the perceptron outputs 1 (if $w \cdot x > 0$)

- **Sigmoid neurons simulating perceptrons, part II**

  Suppose we have the same setup as the last problem - a network of perceptrons. Suppose also that the overall input to the network of perceptrons has been chosen. We won't need the actual input value, we just need the input to have been fixed. Suppose the weights and biases are such that $w \cdot x + b \neq 0$ for the input x to any particular perceptron in the network. Now replace all the perceptrons in the network by sigmoid neurons, and multiply the weights and biases by a positive constant $c > 0$. Show that in the limit as $c \to \infty$ the behaviour of this network of sigmoid neurons is exactly the same as the network of perceptrons. How can this fail when $w \cdot x + b = 0$ for one of the perceptrons?



Architecture of a neural network



- one input layer, two hidden layers, one output layer
- hidden just means “not an input not an output”

How do we design a neural network to recognize a 9, for example?

- for a 4096 pixel image, have 4096 input neurons
- each neuron takes an input of the brightness of a pixel, scaled between 0 and 1
- output layer is a single perceptron. If output layer outputs 0.5 or less, it is NOT a 9.
- Feedforward neural networks: the output of one layer is used as the input to the next layer; there are no loops in the network
- Recurrent neural networks: loops are allowed! neurons can’t all fire at the same time though. Still, more closer to how the human brain works so it’s interesting to research. Still less powerful than feedforward networks right now.