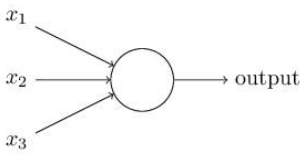
Neural Networks and Deep Learning- Michael Nielsen CH1 - original

- Intro
  - Learning algorithms
    - Suit for the secnarios where it's hard to make precise rules to make decisions
    - The idea is that the neural network uses the examples to automatically infer rules for recognizing handwritten digits.
    - Normally, the more the examples, the better NN learns
  - What to do
    - A handwriting digital number recognizer
    - we'll discuss how these ideas may be applied to other problems in computer vision, and also in speech, natural language processing, and other domains.
  - What are the key ideas about neural networks
    - two important types of artificial neuron
      - the perceptron
      - the sigmoid neuron
    - the standard learning algorithm for neural networks
      - stochastic gradient descent.
  - Throughout, I focus on explaining
    - why things are done the way they are,
    - and on building your neural networks intuition.
    - Amongst the payoffs, by the end of the chapter we'll be in position to understand what deep

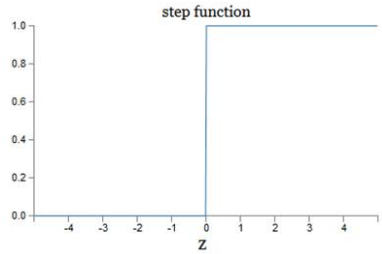
learning is, and why it matters.

- the perceptron
  - Structure



    - Inputs: searval binary
    - Output: one binary
    - Weight: real No., same count as inputs
    - Threshold: one real No.
  - How it works
    - Decesion rules·

$$\text{output} = \begin{cases} 0 & \text{if } \sum_j w_j x_j \leq \text{ threshold} \\ 1 & \text{if } \sum_j w_j x_j > \text{ threshold} \end{cases} \quad (1)$$



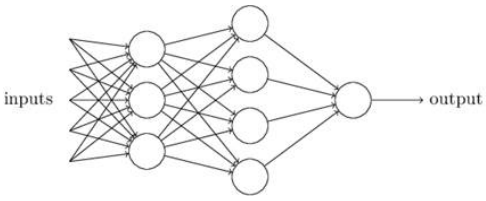
    - Simplified by b≡−threshold

$$\text{output} = \begin{cases} 0 & \text{if } w \cdot x + b \leq 0 \\ 1 & \text{if } w \cdot x + b > 0 \end{cases} \quad (2)$$

      - the bias as a measure of how easy it is to get the perceptron to output a 1. or in more biological terms, the bias is a measure of how easy it is to get the perceptron to fire.
    - A way you can think about the perceptron is that it's a device that makes decisions by weighing up evidence.

- By varying the weights and the threshold, we can get different models of decision-making.

- a complex network of perceptrons could make quite subtle decisions:



- The perceptrons in the later layer make decisions based on the outputs of the previous layer, which means the later layers can make decisions at a more complex and abstract level.

- Where perceptrons can be used
  - To make decisions
  - to compute the elementary logical functions , the underlying computation, e.g. AND, OR, and NAND
    - networks of perceptrons can compute any logical function
    - perceptrons are also universal for computation, because NAND gates are universal for computation

- What is unique with perceptron
  - we can devise learning algorithms which can automatically tune the weights and biases of a network of artificial neurons.
  - So that it can learn to solve problems

- the sigmoid neuron

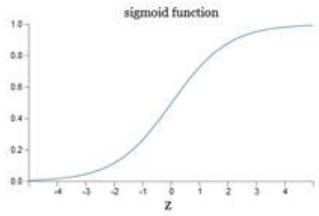
- Whay we need sigmoid instead of perceptron
  - What gives the ability to learn
    - small change in weight causes only a small corresponding change in the output. And then we can modify the weights and biases to get the desired output.
    - But in fact, a small change in the weights or bias of any single perceptron in the network can sometimes cause the output of that perceptron to completely flip, say from 0 to 1. besides, one change may influence the whole network in a complex way.
    - Here comes sigmoid neuron
- Intro
  - Sigmoid neurons are similar to perceptrons, but modified so that small changes in their weights and bias cause only a small change in their output.
- Structure
  - inputs, x1,x2,…. Real No.
  - Params
    - weights for each input, w1,w2,…,
    - an overall bias, b.
  - Output: $\sigma(w \cdot x+b)$ - Real No.·

$$\sigma(z) \equiv \frac{1}{1+e^{-z}}. \tag{3}$$

- Prop.
  - The smoothness of σ means that small changes Δwj in the weights and Δb in the bias will produce a small change Δoutput

$$\Delta \text{output} \approx \sum_j \frac{\partial \text{output}}{\partial w_j} \Delta w_j + \frac{\partial \text{output}}{\partial b} \Delta b, \tag{5}$$

  - Very suitable to learn
    - it's the shape of σ which really matters, and not its exact form



    - The smoothness, the continuity enables ability to learn.
    - The main thing that changes when we use a different activation function is that the particular values for the partial derivatives in Equation (5) change. Which influence the learning speed only.
  - The output can be regarded as a probability or confildence
- Exercise
  - No change
  - Output = 0 is the decision boundary.

- The architecture of neural networks
  - Basic
    - Input layer/neuron
    - Output layer/neuron
    - Hidden layer: means nothing more than "not an input or an output"
    - multiple layer networks
    - these are sometimes called multilayer perceptrons or MLPs, despite being made up of sigmoid neurons, not perceptrons.
  - Design of NN
    - Input/output layer: #input/output
    - Hidden layer requires art, normally guided by some heuristics for hidden layer design [research]
      - e.g. trade-off between #neurons & training time
    - Types of NN
      - feedforward neural networks
        - No loops, information is always fed forward, never fed back. Loops are not allowed
      - recurrent neural networks
        - feedback loops are possible
        - 按时间，one N fires another for a limited duration，so

over time we get a cascade of neurons firing
    - Loops don't cause problems in such a model, since a neuron's output only affects its input at some later time, not instantaneously.
    - Recurrent neural nets have been less influential than feedforward networks, in part because of the less powerful learning algorithms (at least to date) . But they're much closer in spirit to how our brains work. And it's possible that recurrent network can solve more complex problems

---

- A simple network to classify handwritten digits
    - Procedure
        - First, we'd like a way of breaking an image containing many digits into a sequence of

separate images

- the standard learning algorithm for neural networks, known as stochastic gradient descent
- why things are done the way they are, and on building your neural networks intuition.

- Sub
  - what can be a prototype
    - As a prototype it hits a sweet spot: it's challenging - it's no small feat to recognize handwritten digits - but it's not so difficult as to require an extremely complicated solution, or tremendous computational power.
    - Furthermore, it's a great way to develop more advanced techniques, such as deep learning.
  - the NAND gate is universal for computation, that is, we can build any computation up out of NAND gates

- LAN
  - doing progressively more complex image processing.
  - Most people **effortlessly** recognize those digits as 504192. That ease is **deceptive.**
  - We carry in our heads a supercomputer, tuned by evolution over hundreds of millions of years, and superbly adapted to understand the visual world.
  - turn out to be not so simple to express algorithmically.
  - you quickly get lost in a **morass of exceptions and caveats** and

special cases.

- But this short program can recognize digits with an accuracy over 96 percent, **without human intervention.**
- loathe bad weather
- leftmost, bottommost
- The computational universality of perceptrons is **simultaneously reassuring and disappointing.** It's reassuring because it tells us that networks of perceptrons can be as powerful as any other computing device. But it's also disappointing, because it makes it seem as though perceptrons are merely a new type of NAND gate. That's **hardly big news!**
- Conventional vs. traditional
- This tuning happens in response to external stimuli, without direct intervention by a programmer.
- Radically vs. completely, totally
- Terrific
- the behaviour of the network on all the other images is likely to have completely changed in some hard-to-control way.
- the algebraic form of the sigmoid function turns out to be more of a technical detail than a true barrier to understanding.