Deep Learning

week 2

The Plan

- Discuss homework:
  - NN&DL book, ch1 + ch2
  - ideas for pet projects
- Watch (short) talk by Hinton
- talk, read, code, repeat

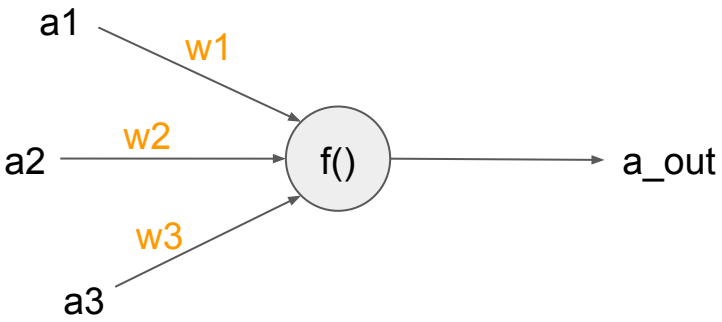
Neural Networks & Deep Learning

Chapters:

1. Basics (neurons, networks)
2. Basic learning (backprop)
3. Improvements
4. Intuitive proof function learning
5. Why is DL hard
6. Deep Learning

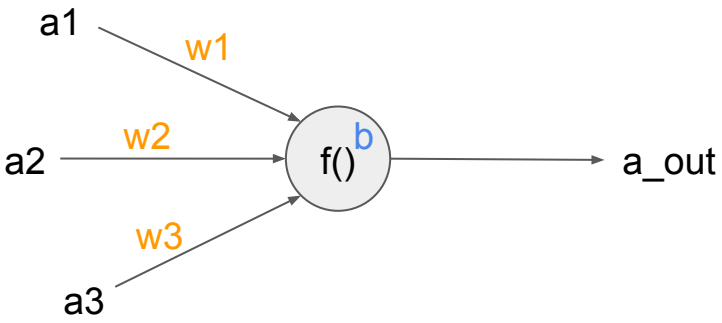
Neural Networks & Deep Learning - Ch1 + Ch2

Neurons



Neural Networks & Deep Learning - Ch1 + Ch2

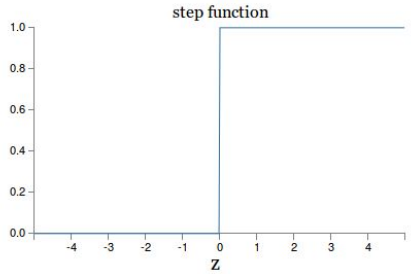
Neurons



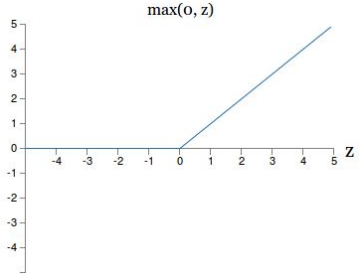
a_out = f(w·a + b)

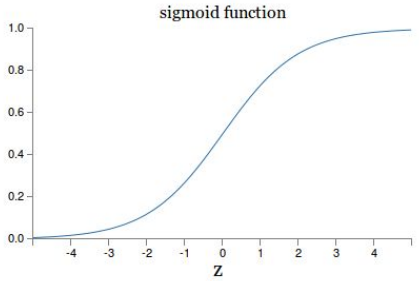
Choices for F()

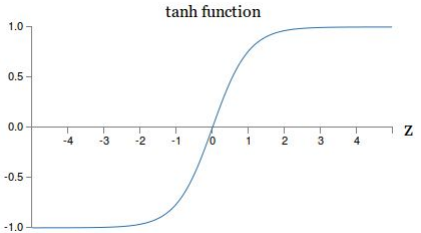
perceptron



rectified linear neuron





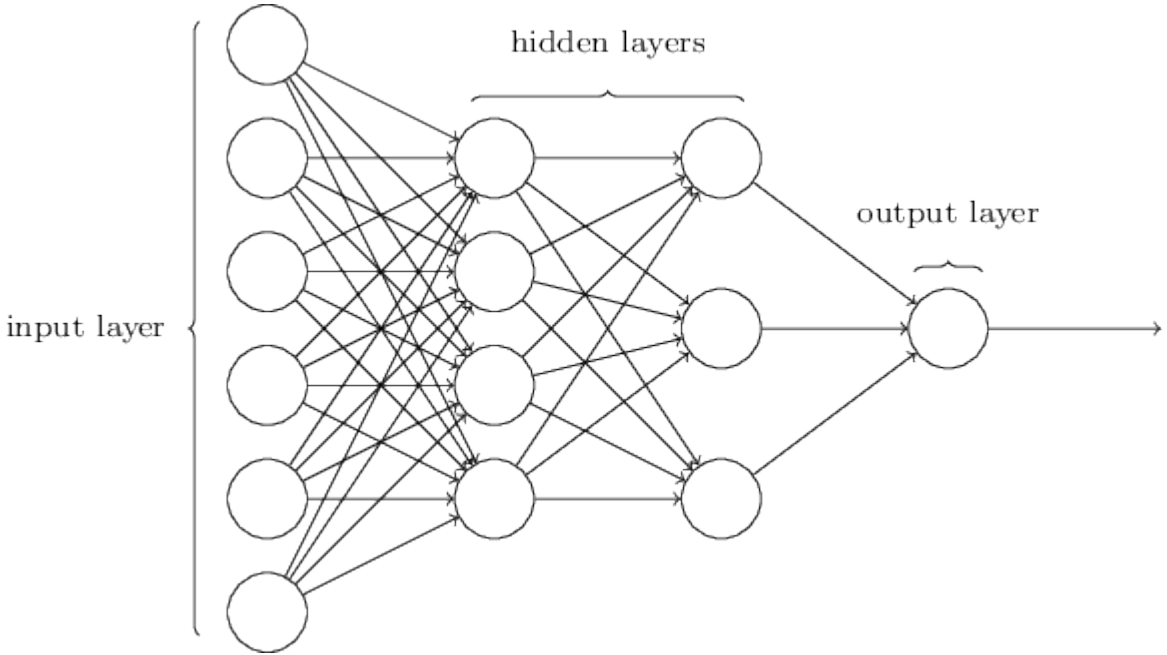


probabilities

$$softmax(x^{(k)}) = \frac{e^{x^{(k)}}}{\sum_j e^{x^{(j)}}}$$

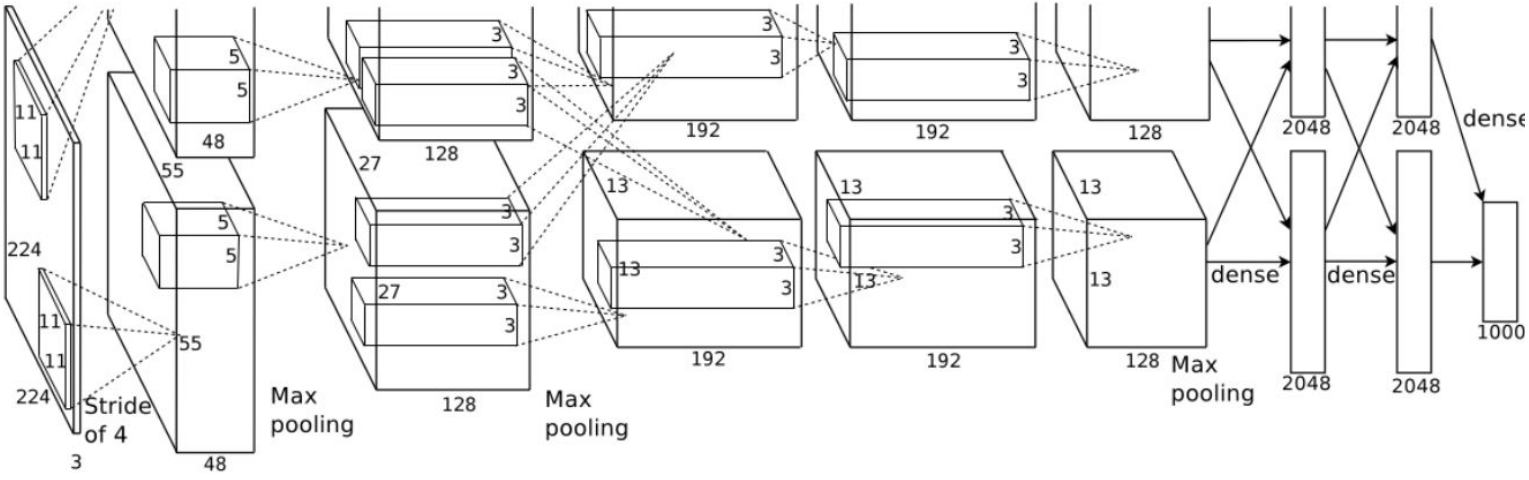
Neural Networks & Deep Learning - Ch1 + Ch2

Networks



Neural Networks & Deep Learning - Ch1 + Ch2

Deep Networks



2012

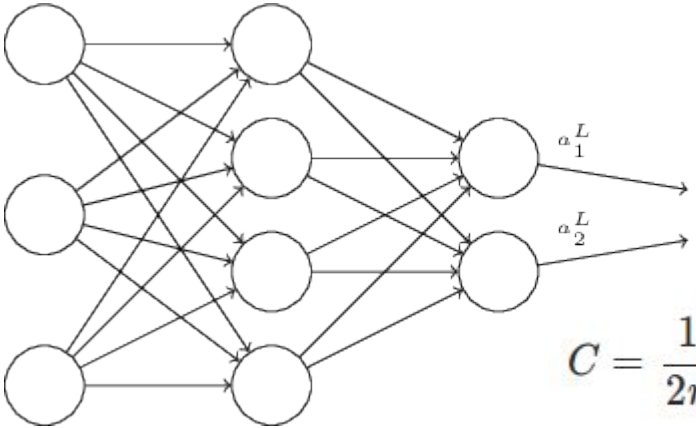
2014

2015

Neural Networks & Deep Learning - Ch1 + Ch2

Learning: backpropagation

- learning parameters (but hyper parameters up to engineer)
- stochastic (batched) gradient descent: avg partial derivatives
- intuitively: how does output error change for small change in parameter?
  but now all together in 1 fwd + bwd pass



$$\text{cost } C = C(a^L)$$

$$C = \frac{1}{2n} \sum_x \|y(x) - a^L(x)\|^2,$$

Summary: the equations of backpropagation

$$\delta^L = \nabla_a C \odot \sigma'(z^L) \quad \text{(BP1)}$$

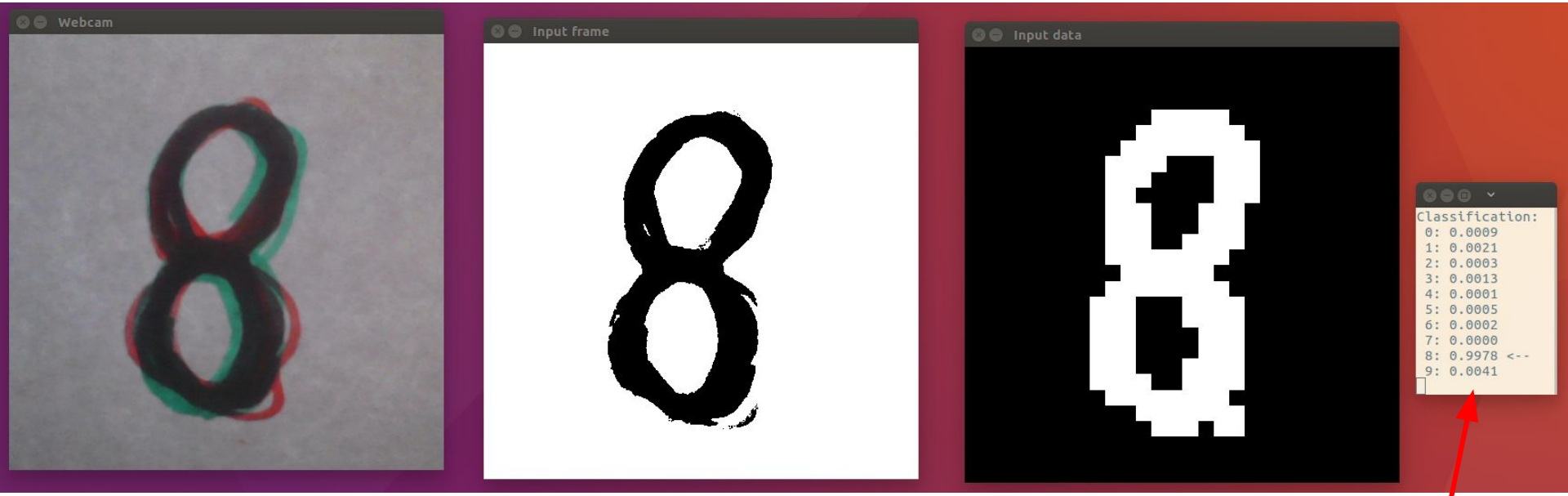
$$\delta^l = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(z^l) \quad \text{(BP2)}$$

$$\frac{\partial C}{\partial b_j^l} = \delta_j^l \quad \text{(BP3)}$$

$$\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l \quad \text{(BP4)}$$

Demo

Network 1 (vanilla implementation) from Nielson, trained on MNIST,
but with real-time classification on live webcam stream



Much harder with real data (centred, line thickness, rotation), but can be stable

DL intro Talk by Hinton

https://www.youtube.com/watch?v=l2dVjADTEDU

???

- talk, read, code, repeat