

Title

- Text

Exercise 7: Neural Networks

MAD

bacdavid@student.ethz.ch

Outline

1. Information
2. Goals
3. Theory/ Recap
4. Exercises

Information

General

- Lecture material & problem sets available [here](#)
- Tutorial material available [here](#)

Specific

- There is a second power point available towards the subject from last year.
- Please have a look at it for more information.

Goals

Goals of Today

- Know how neurons work
- Know how we can stack neurons to a neural network
- Understand how we can optimize neural networks efficiently

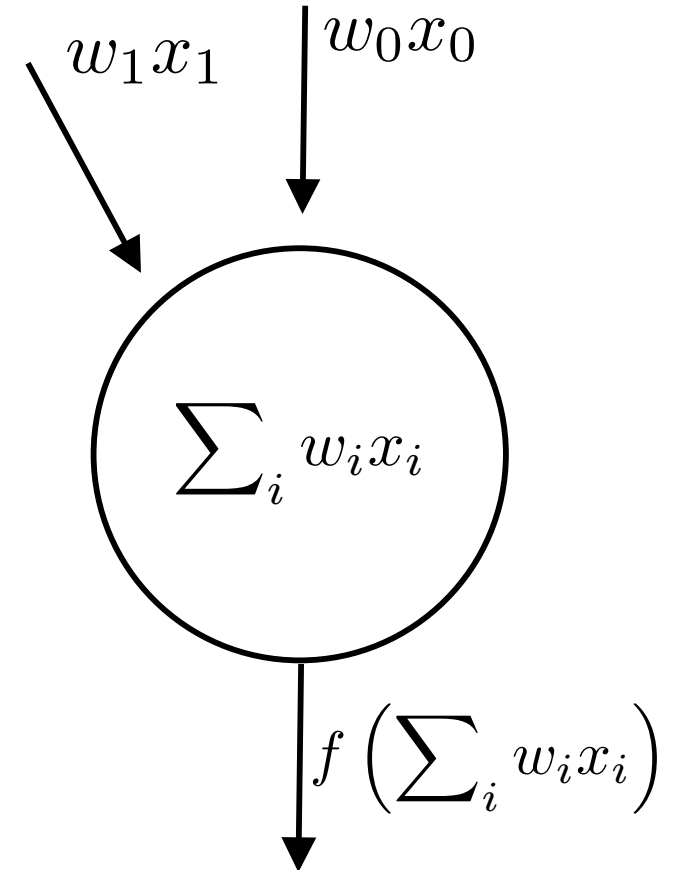
Theory / Recap

Neurons

1. Each input x_i is weighted by w_i
2. Then summed
3. Then an activation function f is applied
4. $f(\sum w_i x_i)$ constitutes the input for the next layer

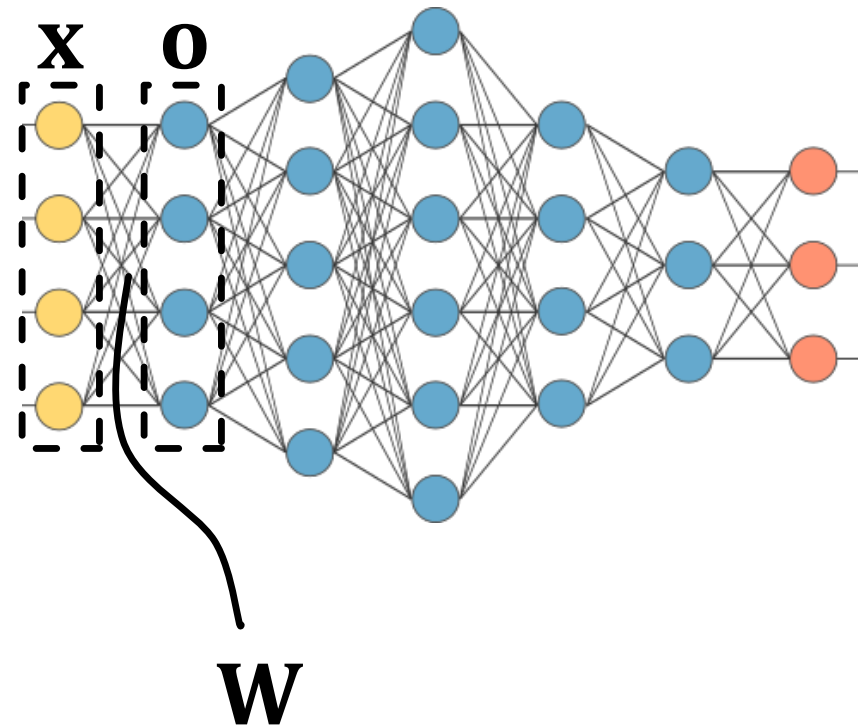
Note

- $\sum w_i x_i = \mathbf{w}^T \mathbf{x}$ “dot product”
- Usually there is a bias “offset” term b : $\sum w_i x_i + b$
- Notation is different in the exercises



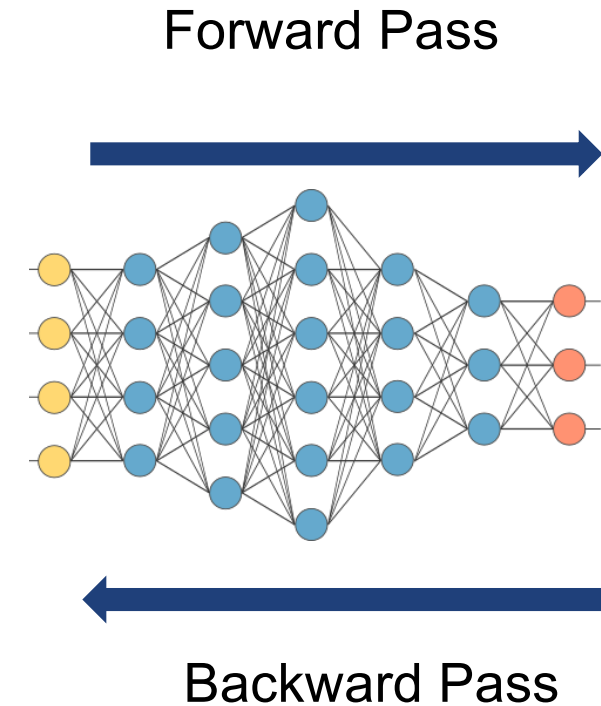
Neural Networks

- Universal function approximator
- Can approximate any function
- Layers of interconnected neurons
- Using lin. alg. notation:
 1. $f(\mathbf{w}_1^T \mathbf{x}) = o_1$ & $f(\mathbf{w}_2^T \mathbf{x}) = o_2$ etc.
 2. Stack: $\hat{f}(\mathbf{W}\mathbf{x}) = \mathbf{o}$



Operations on Neural Network

- Forward pass: read out $\mathbf{y} = \mathbf{F}^{\text{NN}}(\mathbf{x})$
- Backward pass: gradient descent



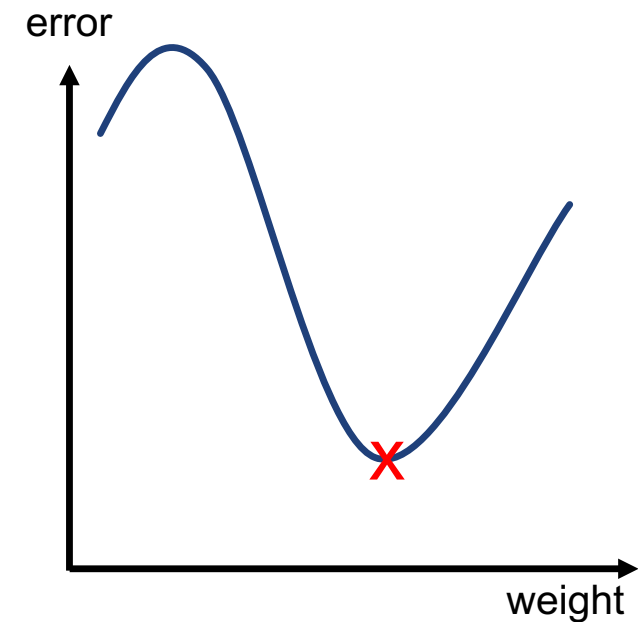
Gradient Descent

- Gradient Descent is used to adjust the weights w_i as such that they produce the output we want
- What do we want? Minimize $error = (y_{target} - y(w))^2$
- Update:

$$w \leftarrow w - \eta \cdot \frac{d \text{error}(w)}{dw}$$

Questions

- Will we find a global minimum?
- What does the derivative intuitively mean?
- What's the derivative in higher dimensions?

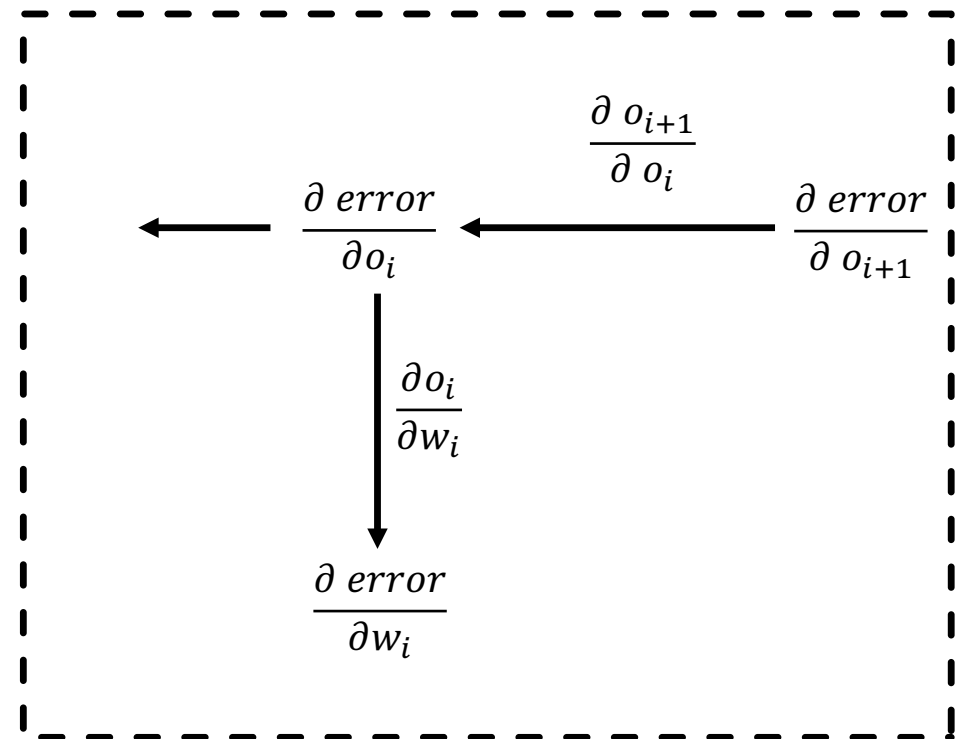


Back Propagation

- Very efficient way to calculate the gradient for G.D.
- How does it work?
 1. Do a forward pass: Check what output is being produced.
 2. Compare the output with the target (check the error)
 3. Check how every weight is responsible for the produced error: Gradient.
 4. Use the chain rule to propagate the error back through the network and adjust the weights accordingly (learning rate η)

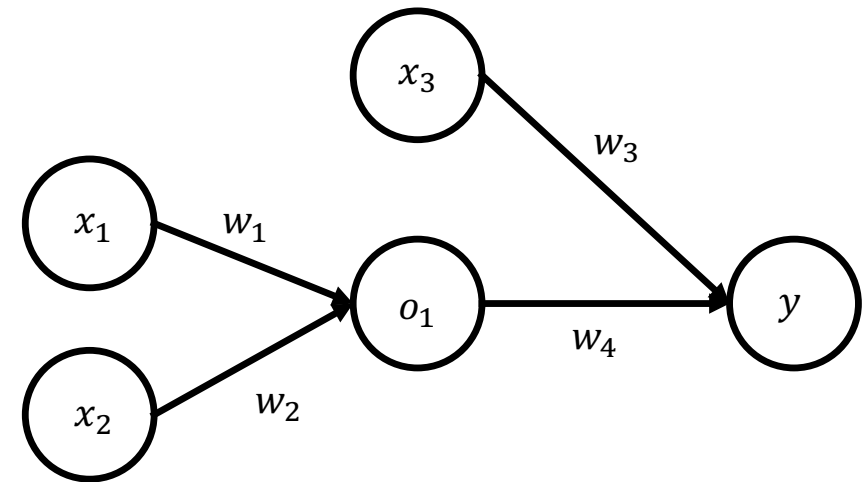
$$\frac{\partial \text{error}}{\partial w_i} = \underbrace{\frac{\partial \text{error}}{\partial o_i}} \cdot \frac{\partial o_i}{\partial w_i}$$

$$\frac{\partial o_{i+1}}{\partial o_i} \cdot \frac{\partial \text{error}}{\partial o_{i+1}}$$



Example 1: Neural Network computations

- o_1 uses a $f(x) = \max(x, 0)$ (called ReLU) activation function, y a linear one
- The neural network on the right can be written as layers:
 $o_1 = \hat{f}(\mathbf{W}_1 \mathbf{x})$ and $y = \mathbf{W}_2 \mathbf{o}$. What are the entries in \mathbf{W}_1 , \mathbf{W}_2 , \mathbf{x} , and \mathbf{o} ?
- Define the error as $(y_{target} - y)^2$. Compute $\frac{d \text{error}}{dw_1}$.
- Assume $w_1 = w_2 = w_3 = w_4 = 1$ and $x_1 = x_2 = x_3 = 1$ and $\eta = 0.1$ and $y_{target} = 2$. Compute one gradient update step as $w_1 \leftarrow w_1 - \eta \cdot \frac{d \text{error}(w_1)}{dw_1}$.



Exercises

Exercise 1

- Different notation used:

x_i : Input

w_{ij} : Weight connecting neuron i and j

z_j^k : Output of neuron j in layer k

$o_j^k = \sigma(z_j^k)$: Activated output of neuron j in layer k

$\sigma(\cdot)$: Activation function

h_{k-1} : Number of neurons in the layer $k - 1$

b_j^k : Bias of neuron j in layer k (can be neglected for simplicity)

- Derive some of the aspects of back propagation: plug in & check what drops out

ie.
$$\frac{\partial z_j^k}{\partial w_{ij}^k} = \frac{\partial}{\partial w_{ij}^k} (\sum_{i'} w_{i'j}^k o_{i'}^{k-1}) = o_i^{k-1} \quad (\text{ex. 1a})$$

Exercise 2

- Simple introduction to tensorflow
- Tensorflow is different than what you are used to terms programming:
 1. Set up a graph with all the functions that you will use afterwards
 2. Feed values into the graph; Values «travel» through graph and are returned
- Please have a look at the additional material: There is an example file.

Questions?

