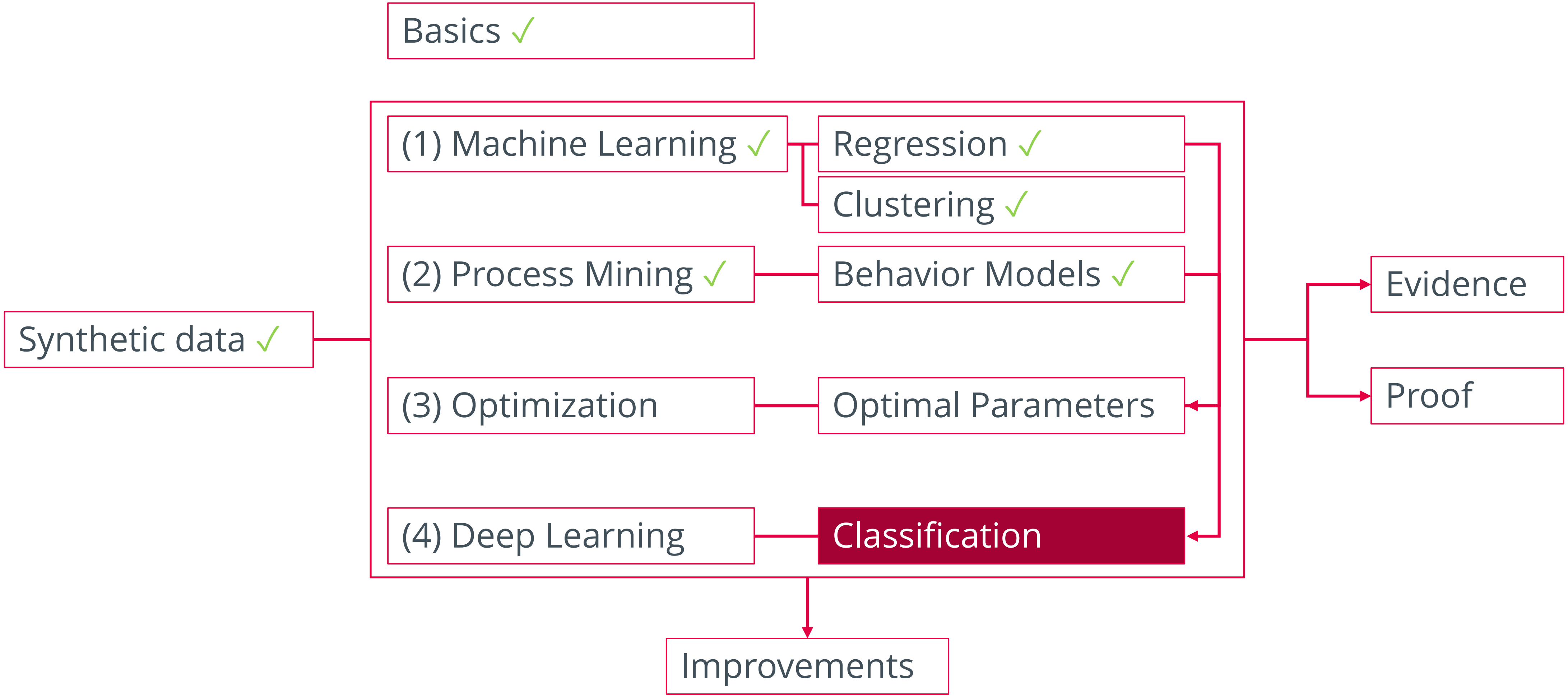
TECHNISCHE HOCHSCHULE
OSTWESTFALEN-LIPPE
UNIVERSITY OF
APPLIED SCIENCES
AND ARTS

Welcome

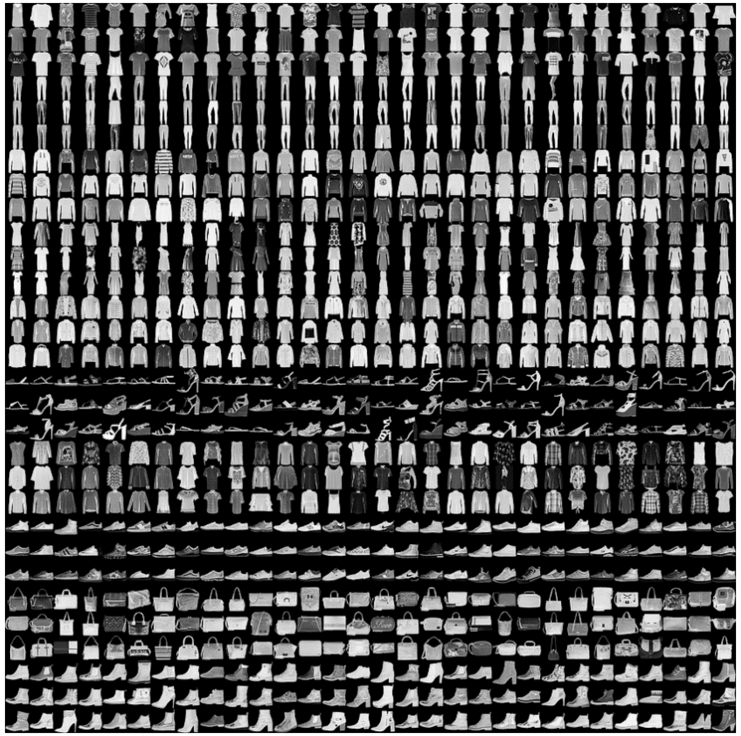
to Advanced Topics in Algorithms

Summary: Advanced Topics in Algorithms

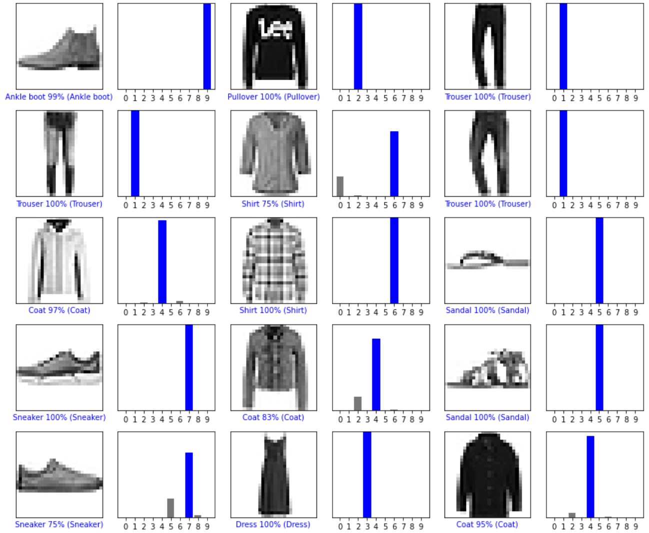


Revision: [Deep] Learning: Classification

Fashion-MNIST is a dataset of Zalando's article images



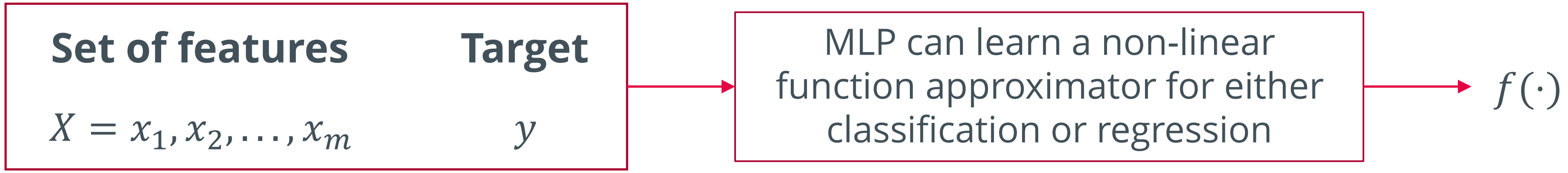
https://arxiv.org/pdf/1708.07747.pdf



XIAO, Han; RASUL, Kashif; VOLLGRAF, Roland. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Neural network models

Multi-layer Perceptron (MLP)
MLP is a supervised learning algorithm that learns a function $f(\cdot): R^m \rightarrow R^o$ by training on a dataset, where m is the number of dimensions for input and o is the number of dimensions for output [1].



Set of features $X = x_1, x_2, \ldots, x_m$

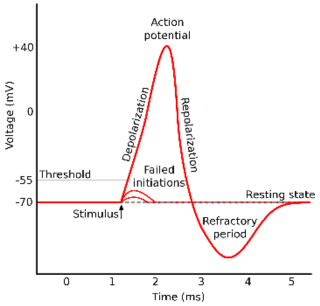
Target y

MLP can learn a non-linear function approximator for either classification or regression

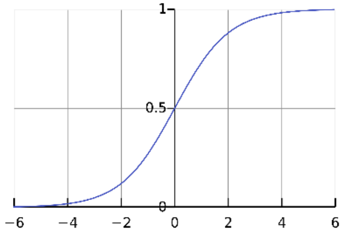
$f(\cdot)$

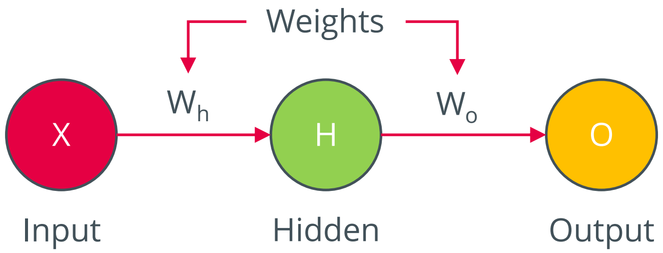
Neurons

Action potential



Sigmoid function





Weights

W_h W_o

X H O

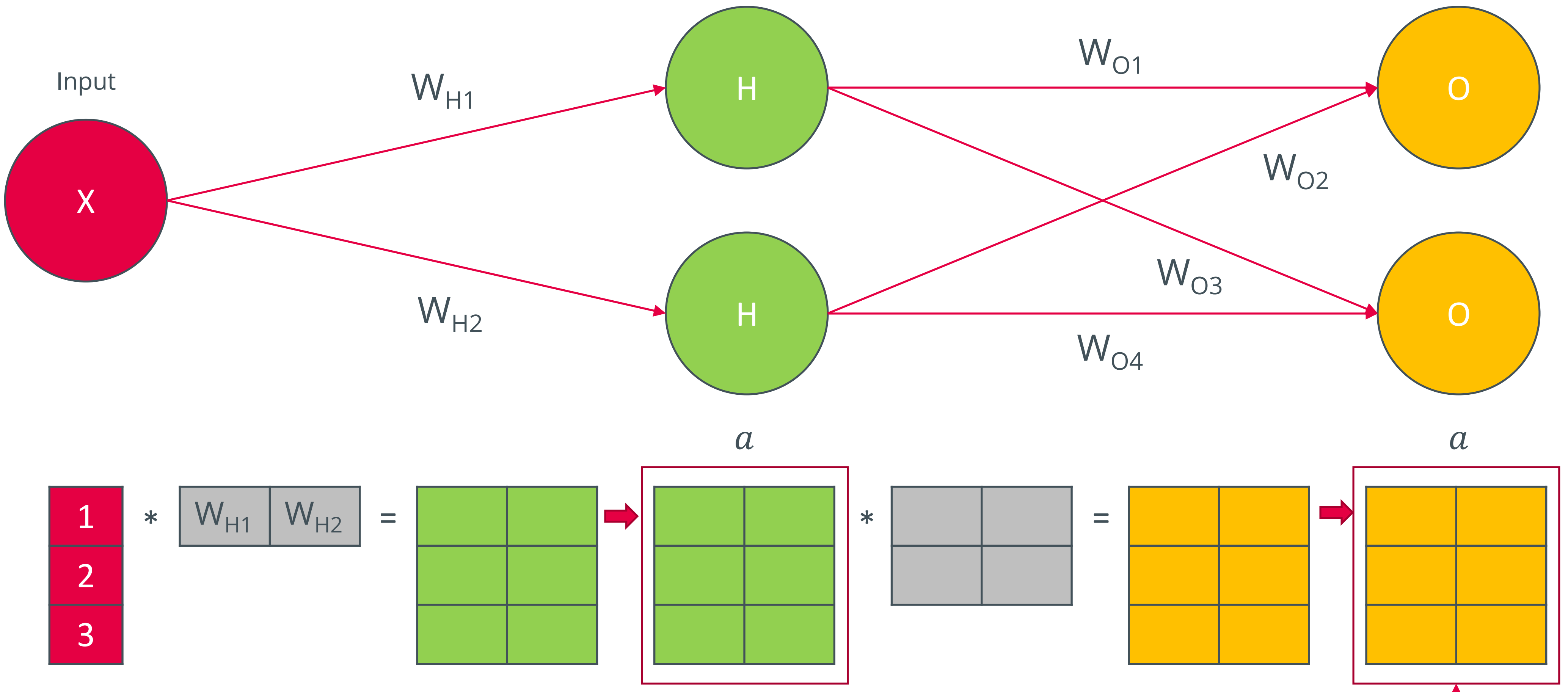
Input Hidden Output

$f(X) = a(a(XW_h)W_o)$

a = activation function

[1] https://scikit-learn.org/stable/modules/neural_networks_supervised.html
[2] https://en.wikipedia.org/wiki/Action_potential#/media/File:Action_potential.svg
[3] https://en.wikipedia.org/wiki/Sigmoid_function#/media/File:Logistic-curve.svg

Matrix representation



Each row represents a prediction for a single observation in our training set

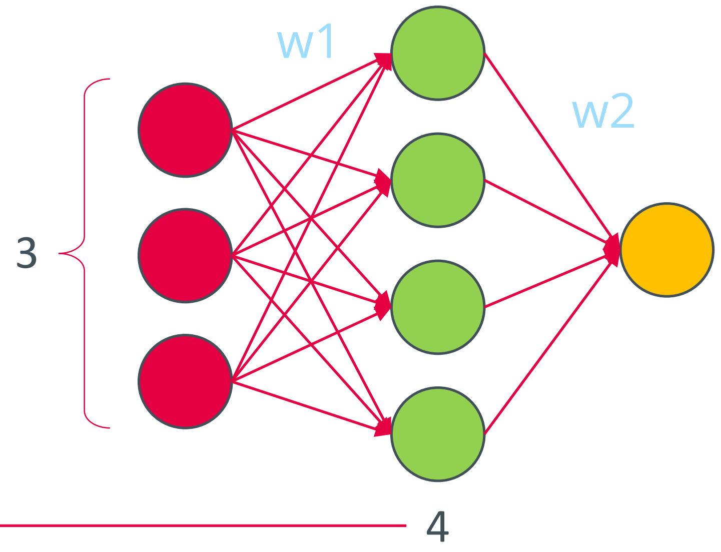
Backpropagation 1/2

Training data

```
x = np.array(([0, 0, 1], [0, 1, 1], [1, 0, 1], [1, 1, 1]), dtype=float)
y = np.array(([0], [1], [1], [0]), dtype=float)
```

x			y
0	0	1	0
0	1	1	1
1	0	1	1
1	1	1	0

3



```
w1 = np.random.rand(3, 4)
```

0.1980562	0.8675463	0.25031689	0.61791776
0.56455728	0.64895493	0.73713434	0.69868288
0.33841526	0.91953443	0.0589787	0.8731658

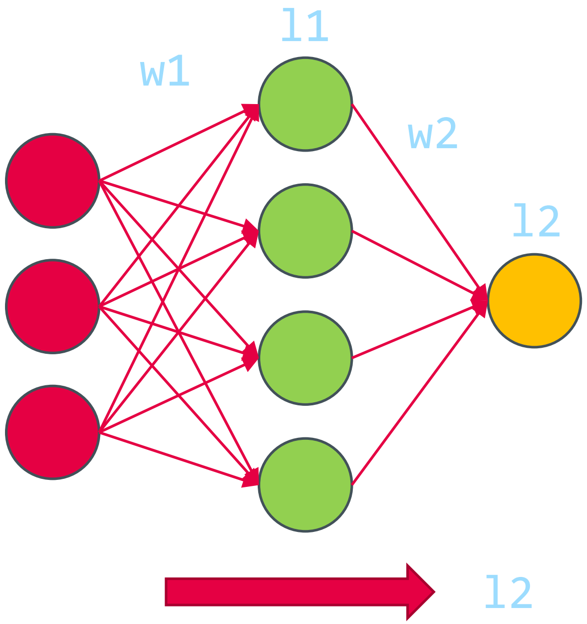
```
w2 = np.random.rand(4,1)
```

0.25678541
0.7534222
0.56365436
0.17331184

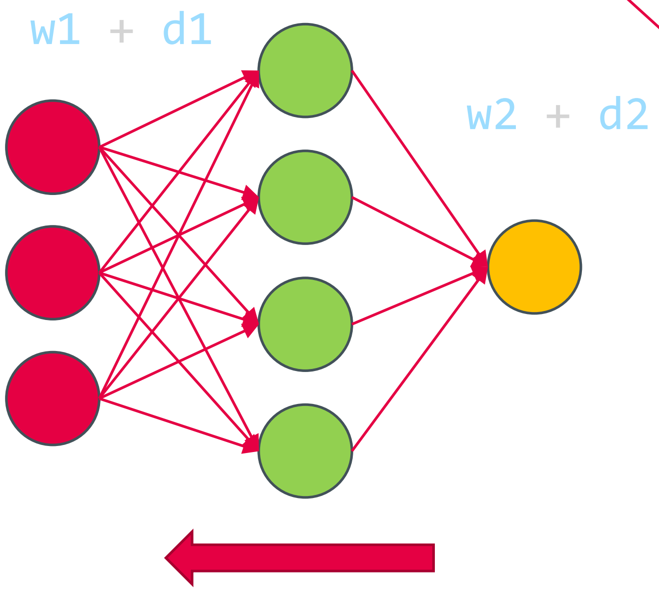
Backpropagation 2/2

Adjust each weight in the network in proportion to how much it contributes to overall error

```
def feedforward(X, w1, w2):
    l1 = a(np.dot(X, w1))
    l2 = a(np.dot(l1, w2))
    return l1, l2
```



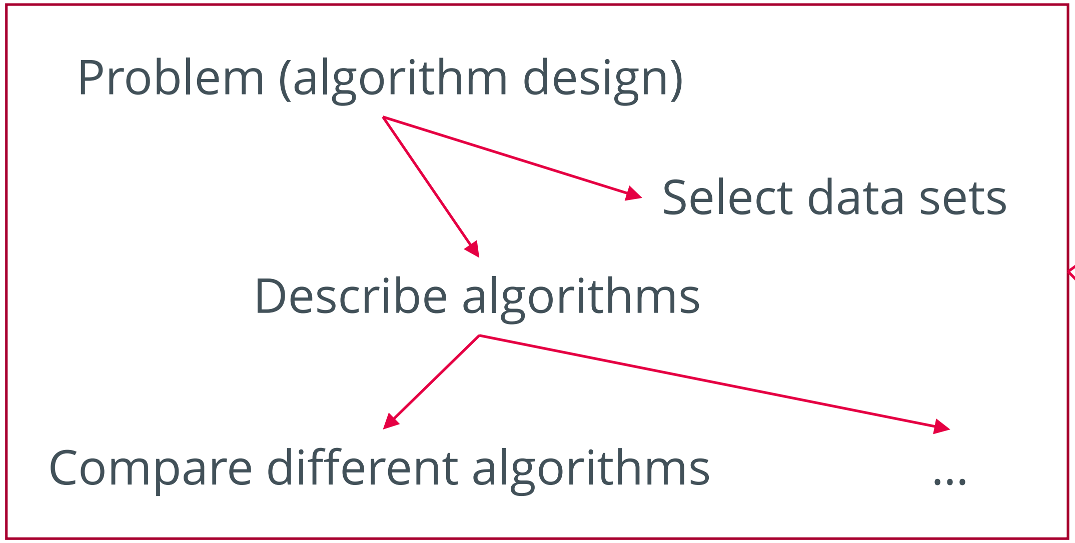
```
def backpropagation(w1, w2, l1, l2, y):
    d2 = np.dot(l1.T, 2*(y - l2) * ad(l2))
    d1 = np.dot(X.T, np.dot( 2 * (y - l2) * ad(l2), w2.T) * ad(l1))
    return w1 + d1, w2 + d2
```



C = 1/2 *(y - l2)^2

C’ = 2*(y - l2)

Exam 2022



Problem (algorithm design)

Select data sets

Describe algorithms

Compare different algorithms

...

Report:

- 6 Pages double-column IEEE format
- Deadline 01/31/2022

Presentation with code review

- 02/04/2022 or
- 02/11/2022

TH OWL

TECHNISCHE HOCHSCHULE
OSTWESTFALEN-LIPPE
UNIVERSITY OF
APPLIED SCIENCES
AND ARTS

Thank you!