STADS

Machine Learning Course
Workshop IV: Computer Vision

In **Kooperation** mit

Prof. Dr. Martin Schlather
Lehrstuhl für Stochastik und ihre Anwendungen

Prof. Dr. Leif Döring
Lehrstuhl für Stochastik

Course structure

Part I: Basics of Machine Learning

04.10 – Introduction to machine learning
11.10 – Neural networks

Part II: Computer Vision (CV)

18.10 – Convolutional Neural Networks
25.10 – Practical application of CNNs
01.11 – CV Challenge presentation

Part III: Natural Language Processing (NLP)

08.11 – Recurrent Neural Networks
15.11 – Practical application of RNNs
22.11 – NLP Challenge Presentation

Agenda

STADS | Students' Association for Data Analytics & Statistics

Motivation

Problem

Model

Implementation with Tensorflow

Agenda

STADS | Students' Association for Data Analytics & Statistics

- Motivation
- Problem
- Model
- Implementation with Tensorflow

The aim of today's event

Deep Learning is a very modern subject area in which a lot of research is currently being done. Therefore, it makes sense to constantly educate yourself and learn new concepts.

1. Formulate question
2. Insight into a scientific paper
3. Implementation

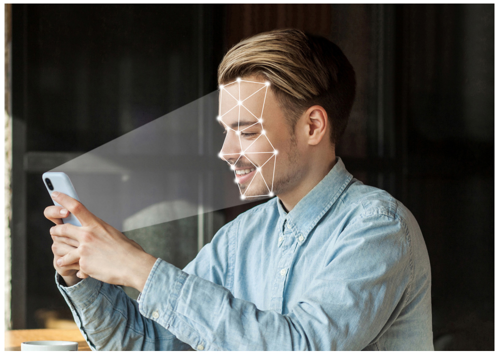
Learning a new concept from the field of ML

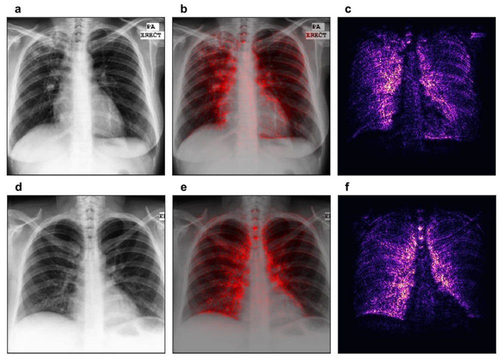
STADS Students' Association for Data Analytics & Statistics

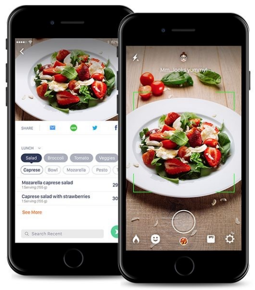
Image verification

Image verification
Image verification is a branch of computer vision. The goal is to decide whether the same object or the same person can be seen in two or more images.

Applications
1. Face-ID 2. Medicine 3. Detection of wrong photos 4. Signature recognition 5. ...





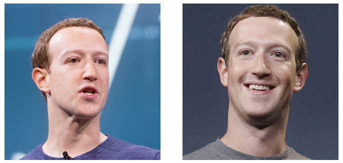


Face recognition

Face recognition in the context of machine learning allows computers to recognize/compare faces in photos and videos.

Our goal: Artificial neural network that can classify whether a given face is similar to another face

False



True

False

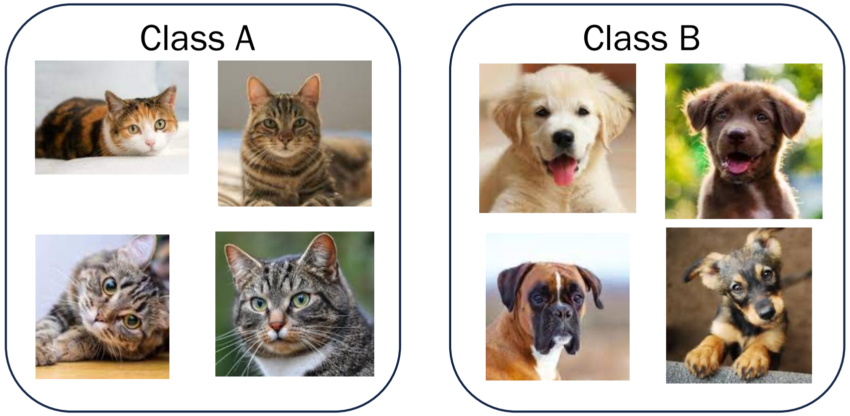
True

Addition	Since a video is just a sequence of photos, such a neural network can also be useful for face recognition in videos.

PROBLEM

One-Shot-Learning

"usual" classification	One-shot classification
There are many data points for each category	For each category there are many data points



One-shot learning can be used to learn features about previously unknown classes

STADS | Students' Association for Data Analytics & Statistics

Motivation

Problem

Model

Implementation with Tensorflow

Source

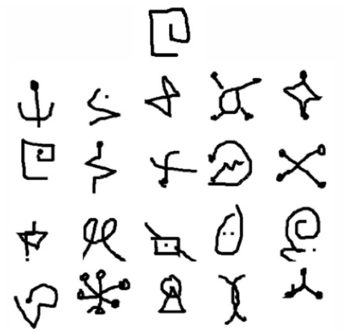
So-called Siamese neural networks are suitable for the problem we are considering. For this purpose, we implement a special network, which was presented in 2015 by Koch, Zemel and Salakhutdinov at the 32nd International Conference on Machine Learning.

Siamese Neural Networks for One-shot Image Recognition

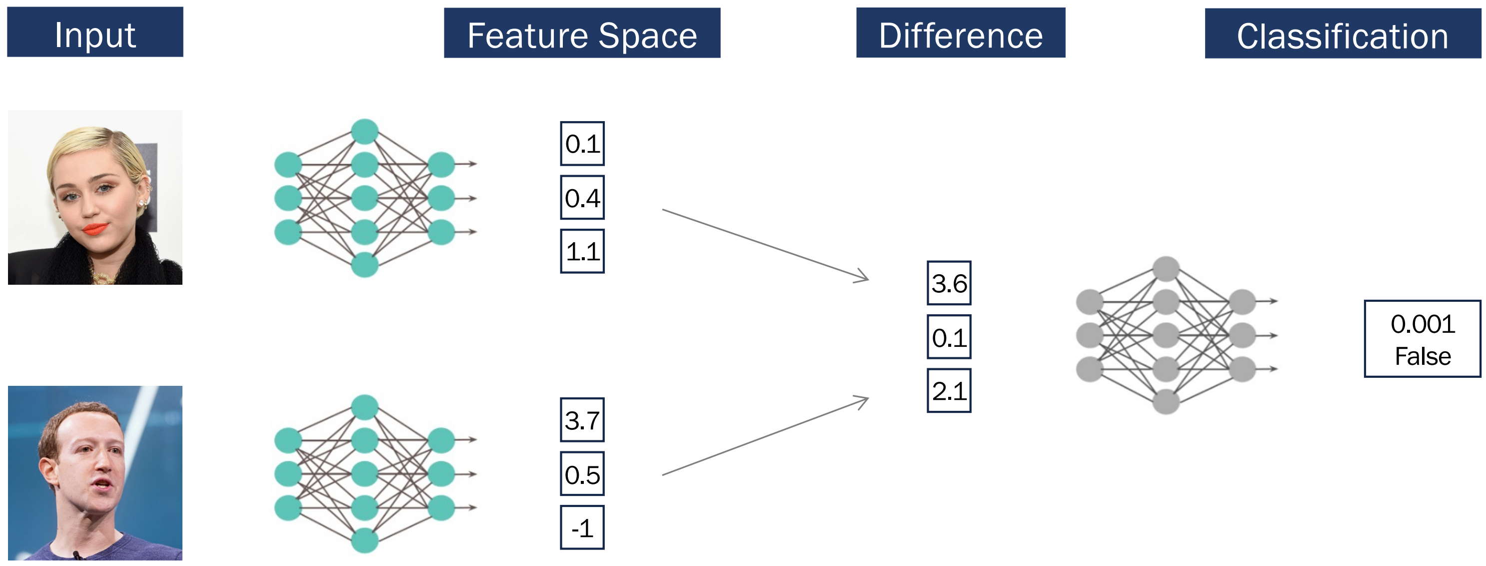
Gregory Koch GKOCH@CS.TORONTO.EDU
Richard Zemel ZEMEL@CS.TORONTO.EDU
Ruslan Salakhutdinov RSALAKHU@CS.TORONTO.EDU
Department of Computer Science, University of Toronto. Toronto, Ontario, Canada.

Abstract

The process of learning good features for machine learning applications can be very computationally expensive and may prove difficult in cases where little data is available. A prototypical example of this is the *one-shot learning* setting, in which we must correctly make predictions given only a single example of each new class. In this paper, we explore a method for learning *siamese neural networks* which employ a unique structure to naturally rank similarity between inputs. Once a network has been tuned, we can then capitalize on powerful discrimina-

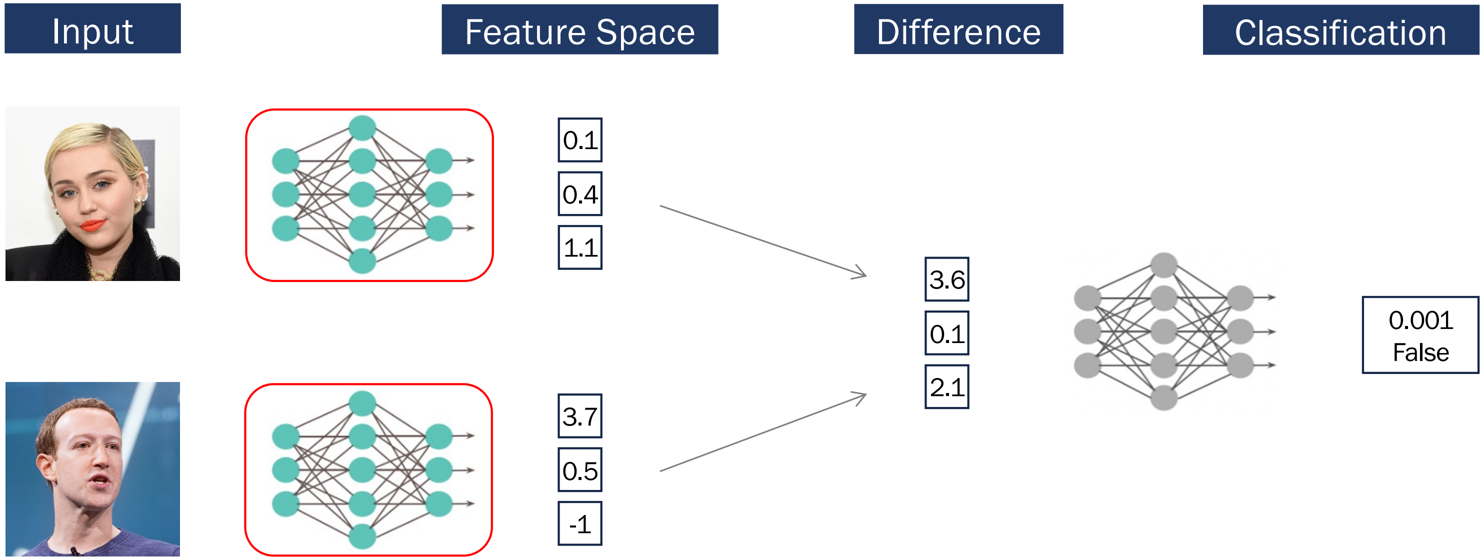


Siamese neural network - simplified sketch



two input images → Transformation into two vectors → Calculate similarity of vectors → Classification

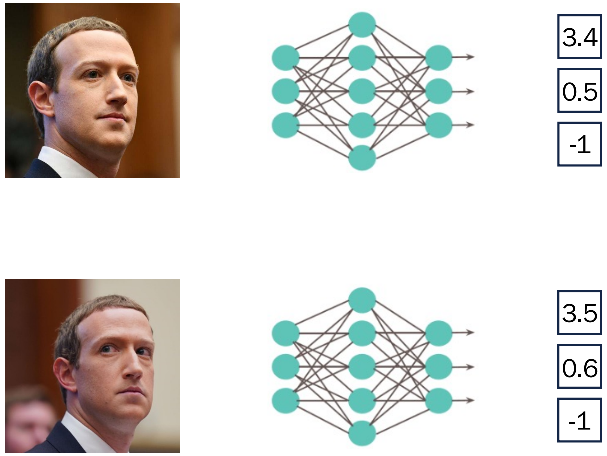
Siamese neural network - simplified sketch



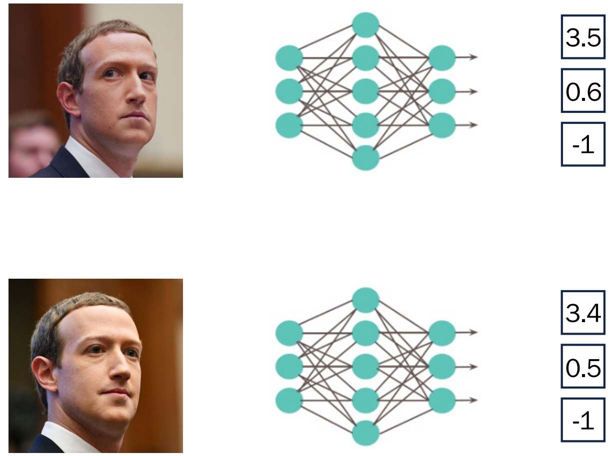
Two separate networks with the same parameters (hence "Siamese" NN).

Why two networks with the same parameters?

1. similar images → similar features



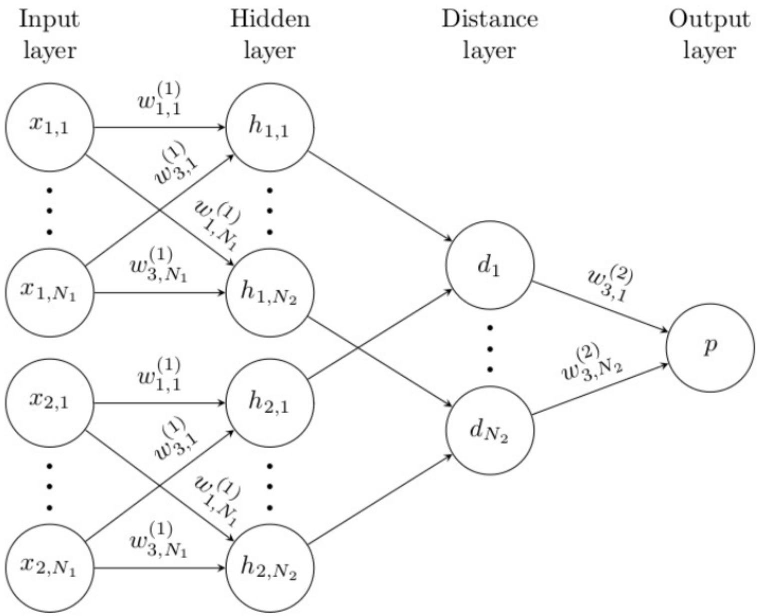
2. Symmetry (same features as in point 1.)



MODEL

Structure of the NN

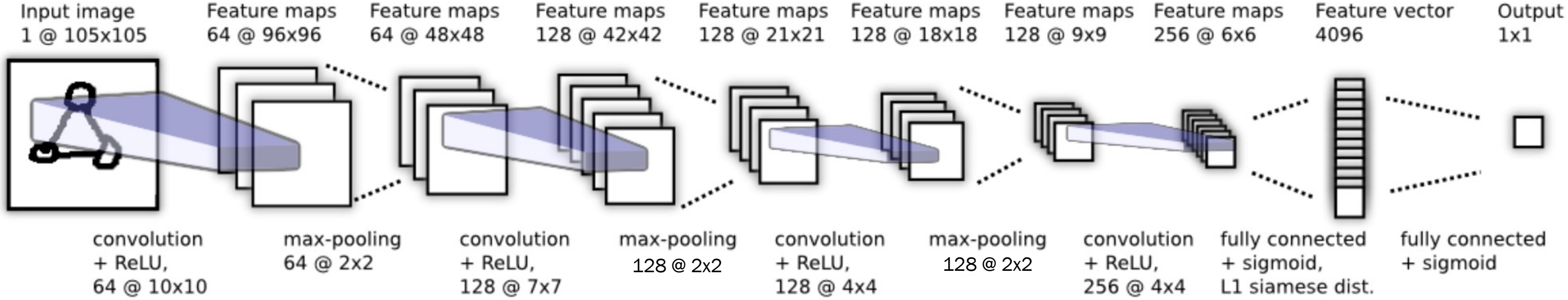
The structure of the Siamese neural network can be visualized as follows:



0.001
False

Structure in detail

The second twin is not shown, but immediately follows the penultimate layer.



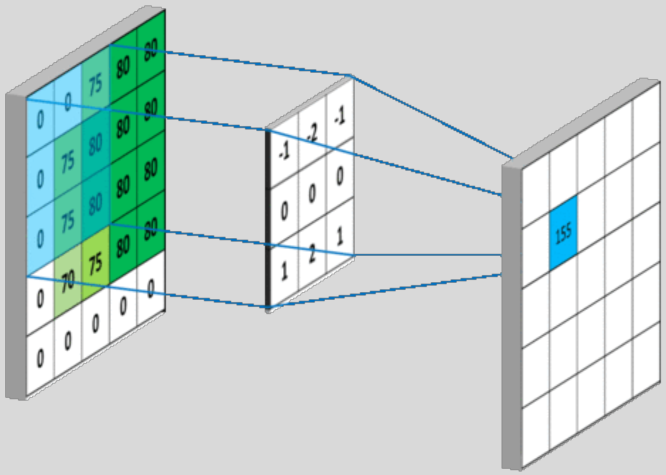
Operationen / Layers
1. Convolution 2. Max-Pooling 3. Fully Connected 4. L1 Distance

Activation functions
1. ReLU 2. Sigmoid

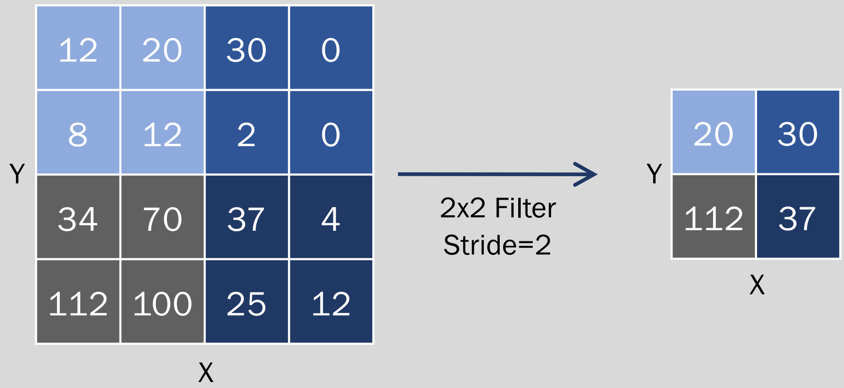
Repetition: Convolution & Max Pooling

Pooling and convolution are among the central operations in computer vision.

Convolution



Max-Pooling



https://mlnotebook.github.io/img/CNN/convZeros.png

L1 Distance Layer

L1 Distance Layer

We define the L1 Distance Layer as the component-wise absolute value of the difference between two vectors.

Explanation

We define the L1 distance layer as the component-wise absolute value of the difference between two vectors.

Feature Space

$$\begin{bmatrix} 0.1 \\ 0.4 \\ 1.1 \end{bmatrix}$$

$$\begin{bmatrix} 3.7 \\ 0.5 \\ -1 \end{bmatrix}$$

Difference

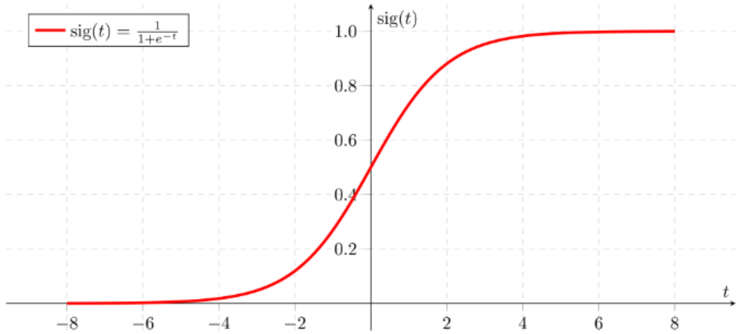
$$\boxed{3.6} = |0.1 - 3.7|$$
$$\boxed{0.1} = |0.4 - 0.5|$$
$$\boxed{2.1} = |1.1 - (-1)|$$

Repetition: ReLU & Sigmoid

In neural networks, the choice of activation functions plays a central role.

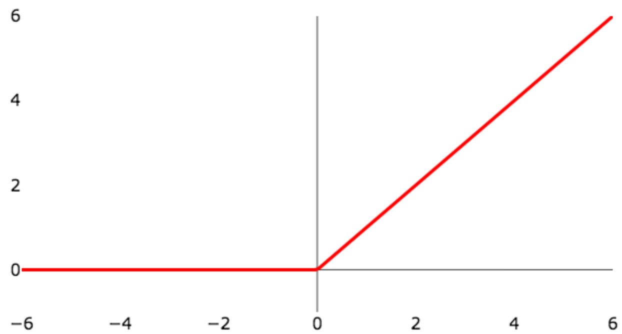
$\boldsymbol{Sigmoid}$

- Especially for classification
- Values between 0 and 1
  → Probabilities



$\boldsymbol{ReLU(t) = max(0, t)}$

- Negative values are set to zero
- Used in the Hidden Layers



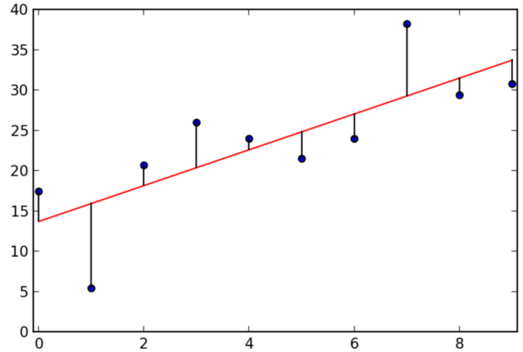
MODEL

Repetition: Loss function

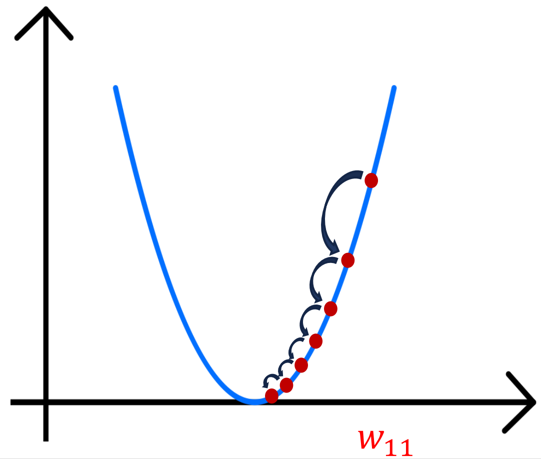
The loss function serves as a measure for the error of a forecast. Two weeks ago, the error sum of squares (SSR) was already introduced.

$$\boldsymbol{Beispiel\!: \ Sum\ of\ squared\ residuals\ (SSR)}$$

$$SSR = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$



The aim is to minimize the loss function



Loss-Function: Binary Cross-Entropy

For classification problems, binary cross-entropy is best.

$$L_p(x_1, x_2) = -\frac{1}{N}\sum_{i=1}^{N} y\left(x_1^{(i)}, x_2^{(i)}\right) \cdot \log\left(p\left(x_1^{(i)}, x_2^{(i)}\right)\right) + \left(1 - y\left(x_1^{(i)}, x_2^{(i)}\right)\right) \cdot \log\left(1 - p\left(x_1^{(i)}, x_2^{(i)}\right)\right)$$

Example for N = 1

$y\left(x_1^{(i)}, x_2^{(i)}\right) = 1$, $\mathrm{p}\left(x_1^{(i)}, x_2^{(i)}\right) = 0.99 \rightarrow L_p(x_1, x_2) = -\{1 \cdot \log(0.99) + 0 \cdot \log(0.01)\} \approx 0$ (small Loss)

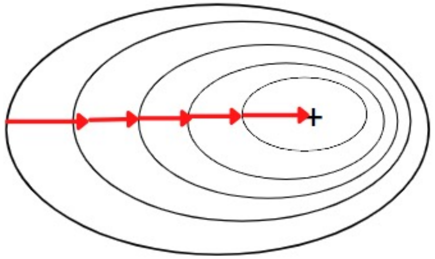
$y\left(x_1^{(i)}, x_2^{(i)}\right) = 1$, $\mathrm{p}\left(x_1^{(i)}, x_2^{(i)}\right) = 0.001 \rightarrow L_p(x_1, x_2) = -\{1 \cdot \log(0.001) + 0 \cdot \log(0.001)\} \approx 7$ (big Loss)

Designation	Explanation
N	Number of data points
$x_1^{(i)}, x_2^{(i)}$	Input of the NN, i.e., face 1 and face 2, to a data point i
$y(x_1^{(i)}, x_2^{(i)})$	Label: 1, if $x_1^{(i)}, x_2^{(i)}$ map the same face, 0 otherwise
p	Probability distribution given by the neural network
$p\left(x_1^{(i)}, x_2^{(i)}\right)$	Prediction of the NN for the similarity of $x_1^{(i)}$ and $x_2^{(i)}$

Stochastic Gradient Descent (SGD)

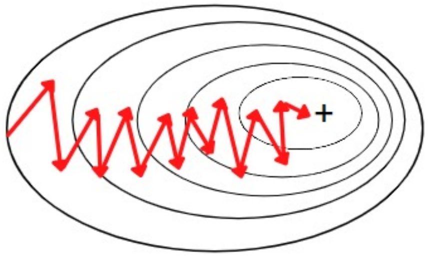
Repetition: Gradient Descent

In the classical gradient descent method, the loss function of the entire data set is calculated in each iteration. This leads to a "direct" convergence, but is much more computationally intensive.



Stochastic Gradient Descent

To reduce the computational effort, the data set is divided into many disjoint batches. In each iteration, the loss function of a single batch is computed in turn. It is important that the data set is randomly shuffled before the decomposition. The size of a batch is called the batch size.



Stochastic Gradient Descent (SGD)

Repetition: Gradient Descent

In the classical gradient descent method, the loss function of the entire data set is calculated in each iteration. This leads to a "direct" convergence, but is much more computationally intensive.

x_1	x_2	y	$\hat{y}$
5	2	21.3	22
3	6	14.5	14
7	-2	4.6	4
0.5	3	9.7	10

Iteration 1, 2

Stochastic Gradient Descent

To reduce the computational effort, the data set is divided into many disjoint batches. In each iteration, the loss function of a single batch is computed in turn. It is important that the data set is randomly shuffled before the decomposition. The size of a batch is called the batch size.

Batch Size = 2

x_1	x_2	y	$\hat{y}$
5	2	21.3	22
3	6	14.5	14
7	-2	4.6	4
0.5	3	9.7	10

Iteration 1 (rows 1–2)

Iteration 2 (rows 3–4)

STADS | Students' Association for Data Analytics & Statistics

Motivation

Problem

Model

Implementation with Tensorflow