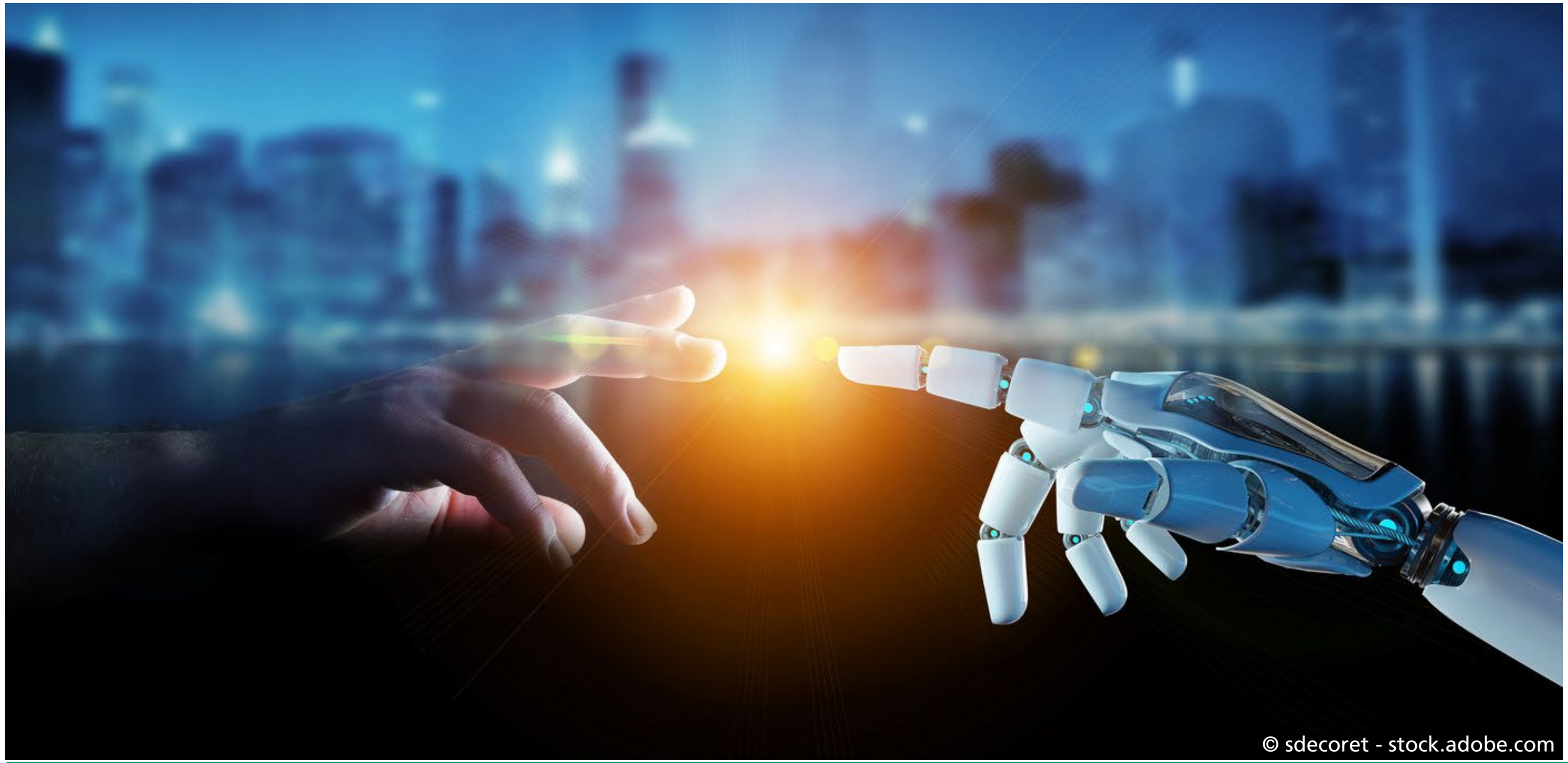
Introduction to Deep Learning and Generative AI

Dr. Gerhard Paaß
Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS)
Sankt Augustin



TensorFlow

Nicht zur Veröffentlichung! März 2024

Fraunhofer
BIG DATA AI

Fraunhofer-Institute for Intelligent Analysis and Information Systems IAIS

- 300+ members: Scientists, project engineers, technical staff and administration
- Location:
  Fraunhofer Campus
  Schloss Birlinghoven/Bonn
- Common research groups and cooperations with

universitätbonn

b-it

Hochschule Bonn-Rhein-Sieg

ML2R Kompetenzzentrum Maschinelles Lernen Rhein-Ruhr

KI.NRW

Schloss Birlinghoven. © Fraunhofer IAIS

IAIS Entrance, © Fraunhofer IAIS

Fraunhofer BIG DATA AI

Fraunhofer IAIS

Do more with data!

Computer Vision

Enterprise Information Integration

Auto-Intelligence

Preventive security

Nicht zur Veröffentlichung! März 2024

Fraunhofer BIG DATA AI

Course Overview

1.	Intro to Deep Learning	Recent successes, Machine Learning, Deep Learning & types
2.	Intro to Tensorflow	Basics of Tensorflow, logistic regression
3.	Building Blocks of Deep Learning	Steps in Deep Learning, basic components
4.	Unsupervised Learning	Embeddings for meaning representation, Word2Vec, BERT
5.	Image Recognition	Analyze Images: CNN, Vision Transformer
6.	Generating Text Sequences	Text Sequences: Predict new words, RNN, GPT
7.	Sequence-to-Sequence and Dialog Models	Transformer Translator and Dialog models
8.	Reinforcement Learning for Control	Games and Robots: Multistep control
9.	Generative Models	Generate new images: GAN and Large Language Models

: link to background material, : link to images used in lecture, G. : Terms that may be asked in the exam

Introduction to Machine Learning and Deep Learning
Agenda

[New York Times 11.10.2023]

TECH FIX

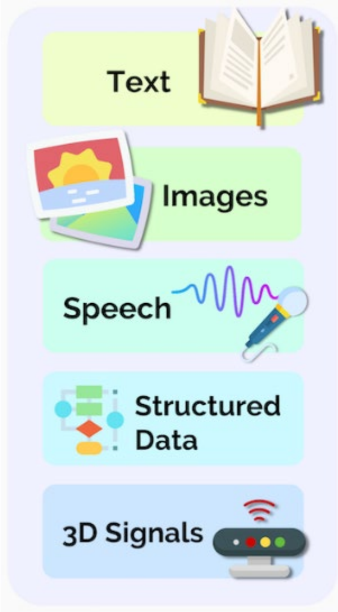
Smartphone Photos Are Getting Faker. Uh-Oh?

Google's new $700 Pixel 8 lets you use artificial intelligence to add or remove elements from your images. It's not clear we really need this.

pixabay father and son

[Biztech news]

ChatGPT: AI will shape the world on a scale not seen since the iPhone revolution, says OpenAI boss



[Financial Times 24.01.2023]

Financial Times

Generative AI: How will the new era of machine learning affect you?

By Richard Waters

25 Jan 2023 · 10 min read

Waymo Chrysler Pacifica Hybrid undergoing testing in the San Francisco Bay Area by Dillu CC BY-SA 4.0

[New York Times 21.08.2023]

'This Experience May Feel Futuristic': Three Rides in Waymo Robot Taxis

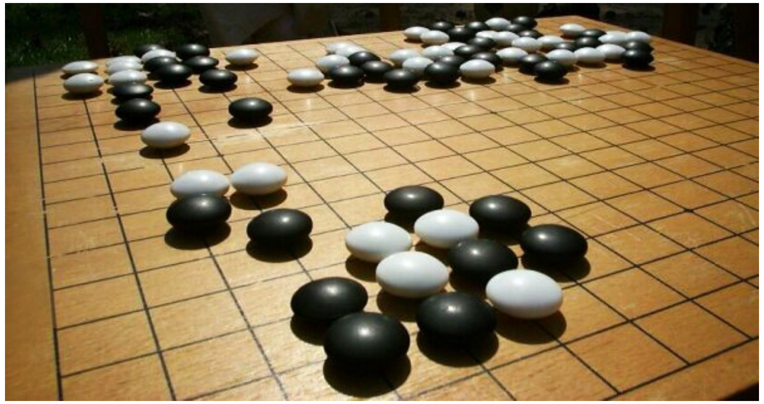
On Monday, Waymo began letting the public pay for rides in its driverless cars in San Francisco. The New York Times dispatched three reporters around the city to test the service.

Is AI a Hype or Reality?

Nicht zur Veröffentlichung! März 2024

Fraunhofer BIG DATA AI

Success Story: AlphaGo



Go board, at a Go-weekend, Hoge Rielen, Belgium by Donarreiskoffer / CC BY-SA 3.0

Go Tournament

- 27 May 2017
- three game competition
  - Ke Jie, world's best professional Go player
  - AlphaGo, a computer program created by Google
- Based on **deep learning** [Silver et al. 2016 Nature]

AlphaGo besiegt nächsten Top-Spieler

Ke Jie aus China gilt als weltbester Spieler des asiatischen Brettspiels Go. Doch der Google-Software AlphaGo musste auch er sich geschlagen geben - trotz eines spektakulären Zuges.
Spiegel 23.05.2017

AlphaGo won 3-0

- AlphaZero: **reinforcement learning**
  zero: no training data used
  ➔ training against itself: **beats AlphaGo**

[Silver et al. 2017 Nature]

Success Story: Image Recognition

- **Image Net**: 15 million named and classified images, covering 22,000 categories.
- **deep neural networks** with billions of node
  - 4.94% top-5 test error on the ImageNet 2012
  - **surpass human-level performance** (5.1%)
  - 2020: top 5 test error is 1.3%

AI: Facebook's new algorithm was trained on one billion Instagram pics

Facebook's researchers have achieved a breakthrough in self-supervised learning, by training an AI system on one billion unlabeled Instagram pictures.
ZDNet March 4, 2021

- **Pretraining:** 1 billion Instagram images without labels
  - Same embedding for images with different views
  - Finetune on ImageNet: **15.8%** top-1 error

Komondor by Nikki68 / CC-SA 3.0 Unported

GT: komondor
1: komondor
2: patio
3: llama
4: mobile home
5: Old English sheepdog

He et al. 2015

[Goyal et al. 2021: Self-supervised Pretraining of Visual Features in the Wild

Success Story: Speech Recognition

Historic Achievement: Microsoft researchers reach human parity in conversational speech recognition

Microsoft AI Blog Oct. 18, 2016

Microsoft has made a major breakthrough in speech recognition, creating a technology that recognizes the words in a conversation as well as a person does.

©WrightStudio - stock.adobe.com

Switchboard corpus

2500 telephone conversations between 500 speakers of American English

A.1: Uh, do you have a pet Randy?
B.2: Uh, yeah, currently we have a poodle.
A.3: A poodle, miniature or, uh, full size?
B.4: Yeah, uh, it's, uh miniature.
A.5: Uh-huh.
B.6: Yeah.
A.7: I read somewhere that, the poodles is one of the, the most intelligent dogs, uh, around.
B.8: Well, um, I wouldn't, uh, I definitely wouldn't …

- Word error rate (WER) 5.1%,
- The rate is better than for professional human transcriptionists (5.7%)
- **neural language models**
- 2020 WER on Librispeech: 2.2%

[Xiong et al. 2016] [Xiong et al. 2017]

Success Story: Natural Language Translation

Deep learning boosts Google Translate tool

Internet giant's latest service employs neural networks to cut error rate by 60%, the company says.

27 September 2016

Google introduces real-time extended voice translation

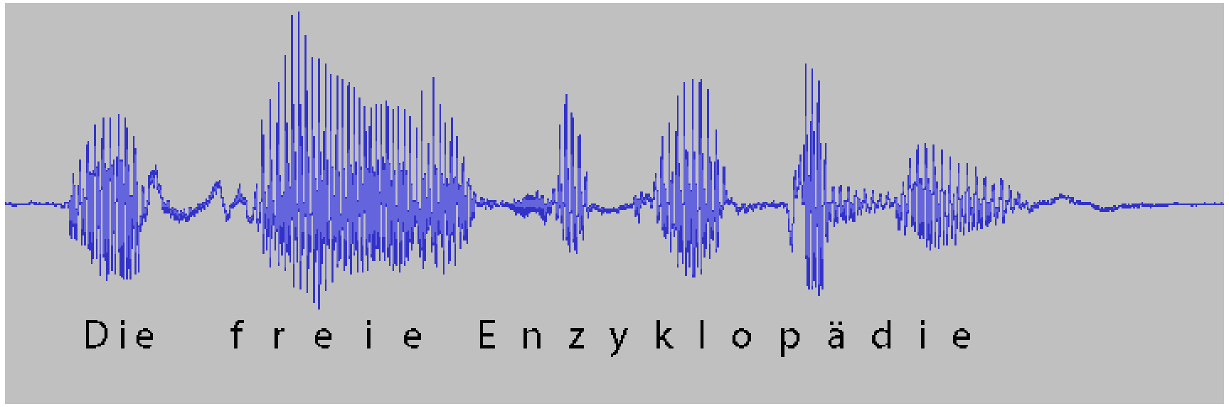
Google has announced a new real-time transcription feature for its free Translate app for Android phones.

Tech Xplore march 18,2020

- entirely based on **deep learning**
- reduces errors by around **60%**, compared previous service

  [Castelvechii 2016] [Schuster 2016]
- 2017:
  Near-human performance for some language pairs

  [Shazeer et al. 2017]



Grafik als Beispiel für eine kontinuierliche Phrase by Mwka / CC BY-SA 3.0

Nicht zur Veröffentlichung! März 2024

Fraunhofer BIG DATA AI

Success Story: Question Answering

- **Natural Questions**: real anonymized, aggregated queries issued to the Google search engine.
- 315k training examples, 7842 test examples

Question: when are hops added to the brewing process?

Answer: The boiling process

- Fully automatic approach

Also used by Google search

1. Search on full Wikipedia using "Embeddings" of passages
2. Collect question and top-100 passages
3. Process by language model and generate answer

Brasserie La Choulette 12-11-2006 by Philimage / CC BY-SA 3.0

- 67% accuracy on Natural Questions test set [Min et al. 2021] p.11
  - Limited to models of 6GB: similar accuracy
  - Competition with human teams: comparable quality

Knowledge base can be changed on the fly

Nicht zur Veröffentlichung! März 2024

Fraunhofer BIG DATA AI

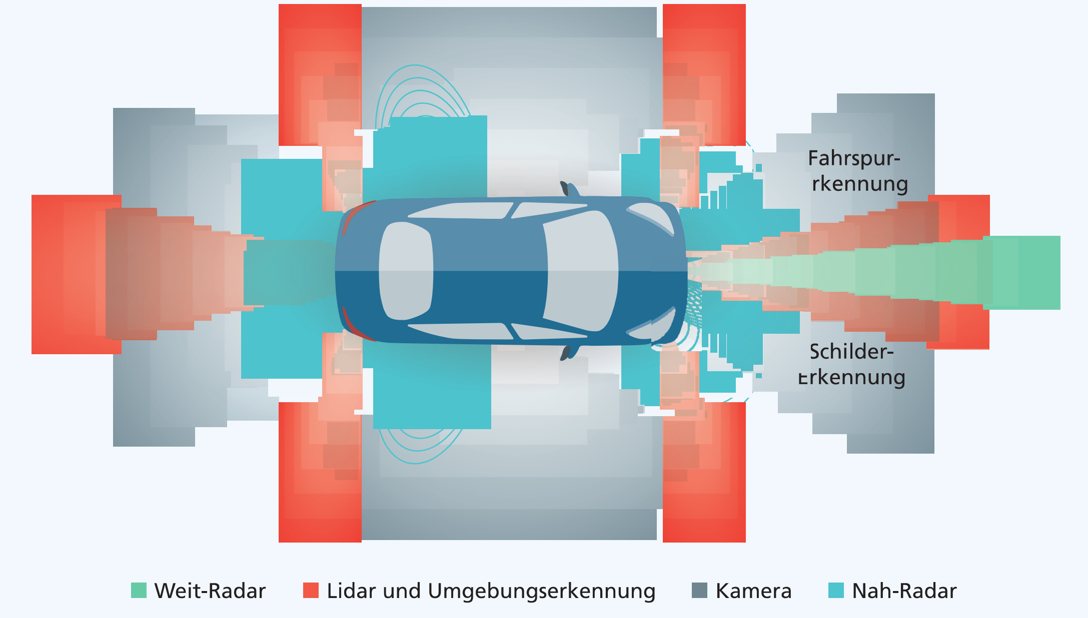
Success Story: Self Driving Car

Waymo simulated real-world crashes to prove its self-driving cars can prevent deaths

The company used real-world fatal crashes that took place in Arizona over a decade. … Waymo's autonomous vehicles would have "avoided or mitigated" 88 out of 91 total simulations

The Verge, Mar 8, 2021

sensor placement © Fraunhofer IAIS



Recent Technology: **Deep Learning**

- learns from examples how the system should behave
- Deep learning is now the best approach to most **perception** tasks, as well as to many **low-level control** tasks.
- Sebastian Thrun as pioneer in 2005

Nicht zur Veröffentlichung! März 2024

Fraunhofer BIG DATA AI

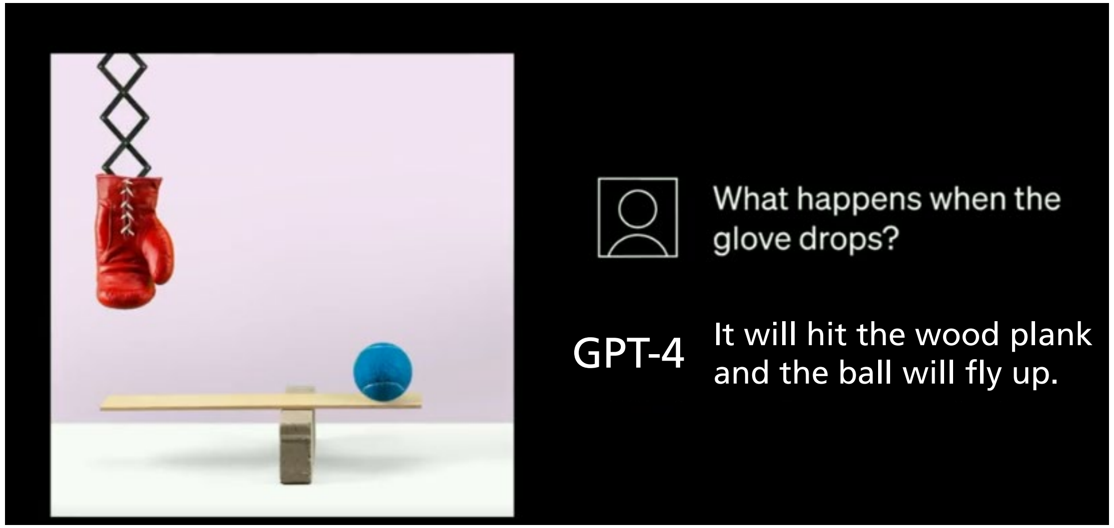
Success Story: Generative Pre-trained Transformer GPT-4

Query: Write a gothic romance about a computer

GPT-4: In a dark and imposing castle, two computers stood side by side, their circuits humming in unison. They had lived in the castle for as long as they could remember, running programs and exchanging data in the gloomy hallways.

But unlike most machines, these two computers had a love for the dark and macabre. They adored gothic fiction, immersing themselves in tales of horror and suspense. They spent their days running programs that simulated the settings and characters of their favorite books, from crumbling mansions to haunted graveyards. …

Process **images** and **text** simultaneously



- **GPT-4** generates fluent text
  - answers question
  - python programs
  - …

Benchmark	GPT-4 prompt with few examples	prior best Model
MMLU questions in 57 subjects (professional & academic)	86.4%	75.2%
HellaSwag Commonsense reasoning around everyday events	95.3 %	85.6 %
AI2 Reasoning Challenge (ARC) Grade-school multiple choice science questions. Challenge-set.	96.3 %	86.5 %
GSM-8K Grade-school mathematics questions	92.0 %	87.3%

https://speakergy.com/smart-speakers-statistics/

Nicht zur Veröffentlichung! März 2024

Fraunhofer BIG DATA AI

Introduction to Machine Learning and Deep Learning
Agenda

1. Deep Learning Success Stories
2. Application of Machine Learning
3. A Simple Model: Logistic Classification
4. Main Groups of Machine Learning Models
5. Deep Neural Network
6. Types of Deep Learning Models
7. Summary

Characteristics of the Tasks

- What is the commonality of the applications described above?

Input	Input	Input
		*my home is my castle*
Model ↓	**Model** ↓	**Model** ↓
Output: *house*	Output: *hello*	Output: *Ma maison est mon chateau*

- From a large number of instances (examples):
  - Compute a mapping from the input to desired outputs: **model**
  - Should yield good results for unknown **new input**

Nicht zur Veröffentlichung! März 2024

Fraunhofer BIG DATA AI

Machine Learning & Modeling

Machine Learning aims at extracting **meaningful** information from data.

- Learns from **data,** which is a set of **instances**
  - **Instance**: e.g. input – label pair
- Creates an internal description (**model**)
  - e.g. function to compute labels from inputs
- The model can be applied to new inputs
  - Does **not** simply memorize all given inputs
  - But „recognizes" patterns and regularities in the data: **generalization**

Input

Model

Output:

house

Deep learning is a special type of machine learning

G. machine learning G. data G. instance G. generalization

Machine Learning: Some Common Facts

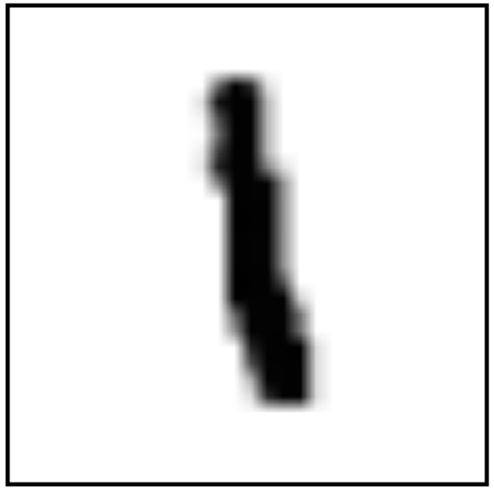
- An extremely large number of different machine learning algorithms were developed
  - Support Vector Machine, Random Forrest, Boosting Trees, Gaussian Mixture Model, K-Means, Latent Dirichlet Allocation, …
- **NEXT** Demonstrate Machine Learning approach using a simple model
  - **logistic classifier**
  - ➔ simple **building block of neural networks**

Introduction to Machine Learning and Deep Learning
Agenda

1. Deep Learning Success Stories
2. Application of Machine Learning
3. A Simple Model: Logistic Classification
4. Main Groups of Machine Learning Models
5. Deep Neural Network
6. Types of Deep Learning Models
7. Summary

MNIST Dataset

- Images of handwritten digits
  - Images are partitioned into **classes** corresponding to digits
  - For each image a classname or **label** is provided:
- Each image is a matrix of **28 by 28 greylevel pixels**:
  - Convert input into a **vector** of 784 real numbers in [0.0,1.0],
  - ➔ Description of inputs
- Training and test sets
  - 60000 training **instances**
  - 10000 test instances



$$\simeq \begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&.6&.8&0&0&0&0&0&0\\ 0&0&0&0&0&0&.7&1&0&0&0&0&0&0\\ 0&0&0&0&0&0&.7&1&0&0&0&0&0&0\\ 0&0&0&0&0&0&.5&1&.4&0&0&0&0&0\\ 0&0&0&0&0&0&0&1&.4&0&0&0&0&0\\ 0&0&0&0&0&0&0&1&.4&0&0&0&0&0\\ 0&0&0&0&0&0&0&1&.7&0&0&0&0&0\\ 0&0&0&0&0&0&0&1&1&0&0&0&0&0\\ 0&0&0&0&0&0&0&.9&1&.1&0&0&0&0\\ 0&0&0&0&0&0&0&.3&1&.1&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0 \end{bmatrix}$$

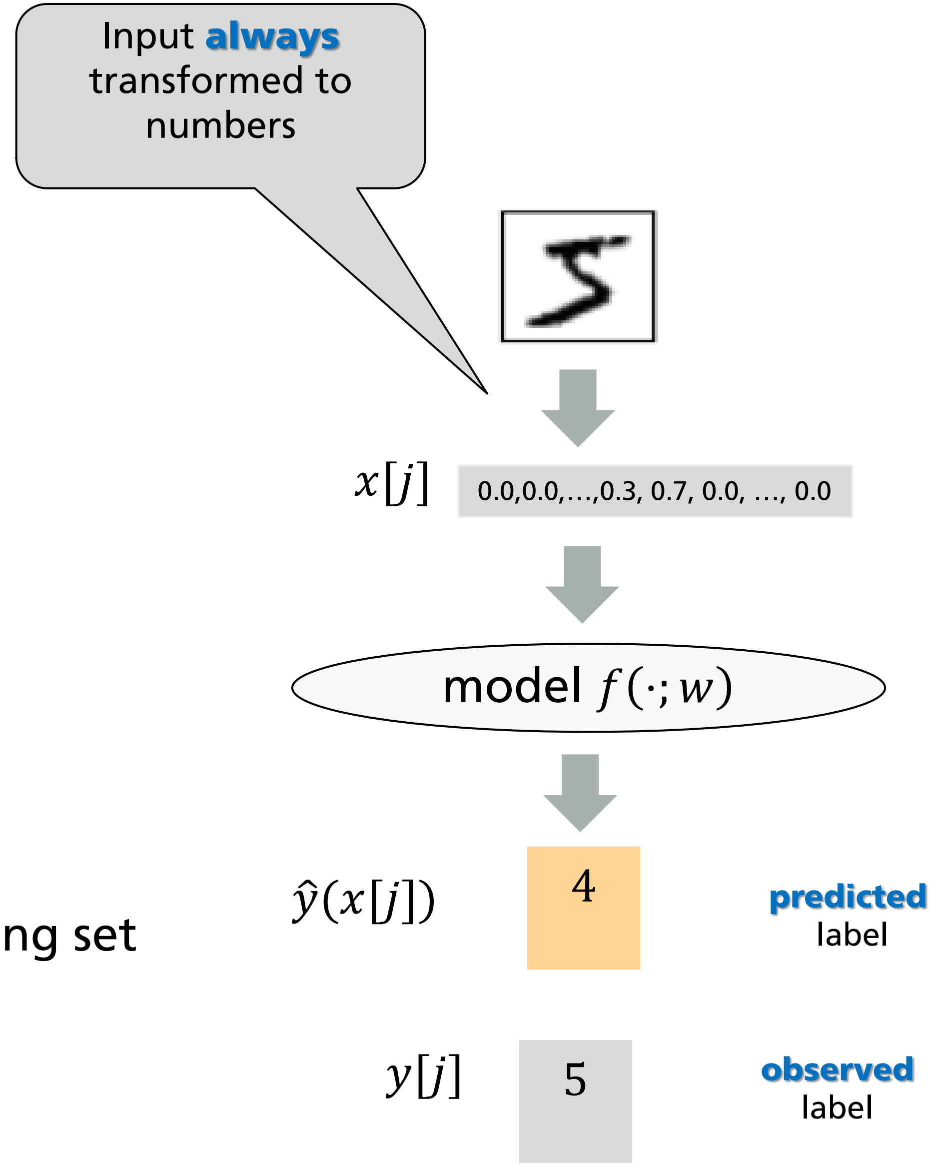
Printout from MNIST Dataset

G. class G. label G. image G. pixel G. matrix G. vector

MNIST Classification Task

- Observed **input**:
  a vector of dimension 784 describing the input image is the set of inputs, e.g
- Observed **label**:
  is the observed label corresponding to the input vector, e.g.
- **Model:** A function
  Assigns to each image an estimated label
  Has a **parameter vector** w with unknown values
- **Task**
  Reduce difference between
  observed (true) label and predicted label for the training set
  ➔ change values of parameter vector w

Training



G. model G. parameter vector G. training G. training set G. classification task

Logistic Regression Classifier: Affine Transformation

- Affine mapping
  - converting the input vector (784 dims) to a score vector (10 dims)
  - Multiply each input component of with a number and form sum

$$
\begin{array}{lccccc}
 & \text{score vector } \boldsymbol{u} & & \text{weight matrix } \boldsymbol{W} & & \text{input } \boldsymbol{x} & & \text{bias } \boldsymbol{b} \\
\text{score for 0} & u_1 & = & w_{1,1}\; w_{1,2}\; w_{1,3}\; w_{1,4}\; w_{1,5}\; w_{1,6}\; \cdots\; w_{1,784} & * & x_1 & + & b_1 \\
\text{score for 1} & u_2 & = & w_{2,1}\; w_{2,2}\; w_{2,3}\; w_{2,4}\; w_{2,5}\; w_{2,6}\; \cdots\; w_{2,784} & * & x_2,\ x_3,\ x_4 & + & b_2 \\
 & & & \cdots & & x_5,\ x_6 & & \cdots \\
\text{score for 9} & u_{10} & = & w_{10,1}\; w_{10,2}\; w_{10,3}\; w_{10,4}\; w_{10,5}\; w_{10,6}\; \cdots\; w_{10,784} & * & \cdots,\ x_{784} & + & b_{10}
\end{array}
$$

$$u_i = w_{i,1} * x_1 + w_{i,2} * x_2 + \cdots + w_{i,784} * x_{784} + b_i$$

- Vector notation of **linear transformation** $\boxed{u = W * x + b}$
- Interpretation of **score**: higher value of u_i: ➔ i–th label more likely
- Weight matrix W, bias b: both are unknown
  ➔ determine their value from data (**training**)

Nicht zur Veröffentlichung! März 2024

G. linear transformation G. weight matrix G. bias

Logistic Regression Classifier: Map Scores to Probability

probability vector $(p_1, \dots, p_k)$

- $p_i \geq 0$
- $p_1 + \cdots + p_k = 1.0$

- Transform scores to probabilities by **softmax** function

$$\hat{p}(y|x) = softmax(u) = \begin{pmatrix} \frac{\exp(u_1)}{\exp(u_1)+\exp(u_2)+\cdots+\exp(u_{10})} \\ \vdots \\ \frac{\exp(u_{10})}{\exp(u_1)+\exp(u_2)+\cdots+\exp(u_{10})} \end{pmatrix}$$

Example

u_i	$exp(u_i)$	$\frac{\exp(u_1)}{\exp(u_1)+\exp(u_2)+\exp(u_3)}$
2,2	9,02	0,860
0,3	1,34	0,128
-2,2	0,11	0,010

- $\hat{p}(y|x;w)$ is the predicted probability vector of classes 1-10 for input
  - for current parameters $w := (W, b)$
  - predict real vector ➔ **logistic regression**
- Predicted label for w:
  - the class with **highest** predicted probability
  - **logistic regression classifier**
- Model: $f(x;w) = \hat{p}(y|x;w) = \text{softmax}(W * x + b)$

Nicht zur Veröffentlichung! März 2024

G. probability vector G. softmax function G. logistic regression G. logistic regression classifier

Logistic Regression Classifier: Loss Function

- 60000 training instances $\text{Train} = \{(x_1, y_1), (x_2, y_2), \dots, (x_{60000}, y_{60000})\}$
  - Change parameters $w := (W, b)$ such that most predicted labels are correct
  - How to measure the accuracy / loss of the classifier?

- **Probability of the whole training set**
  - Assumption: the training instances are statistically independent

$$\hat{p}(TrainSet|w) = \hat{p}(y = y_1|x_1; w) * \cdots * \hat{p}(y = y_{60000}|x_{60000}; w)$$

probability of training set for given w

probability of the **observed** class y_1 for input x_1 given w

- **Probability of events**
- must be independent e.g. dice
- probability of throwing (2,5) 1/6 * 1/6 = 1/36

Modify parameters w such that the **probability** of the training set gets **maximal**.

Nicht zur Veröffentlichung! März 2024

G. probability of training set

Logistic Regression Classifier: Define Loss Function

01-01 ..

- How to find optimal parameters $w := (W, b)$

observed label for x_{60000}

$$\hat{p}(TrainSet|w) = \hat{p}(y = y_1|x_1; w) * \cdots * \hat{p}(y = y_{60000}|x_{60000}; w)$$

transform with $-\log(\circ)$

$$L_{x,y}(w) = -\log(p(TrainSet|w)) = -[\log \hat{p}(y = y_1|x_1; w) + \cdots + \log \hat{p}(y = y_{60000}|x_{60000}; w)]$$

- Transform with $-\log$:
  - log ist montonous:
    $\hat{p}(TrainSet|W, b)$ has maximum exactly when $-\log(p(TrainSet|w))$ has minimum
  - no underflow, sum instead of product
  - multiply by -1: use minimization

- Alternative optimization criterion: minimize the negative log of probability of whole training set

- **Loss function** $L_{x,y}(w)$ assigns a real value to each w
  smaller loss value ⇒ lower difference between predicted and observed output

? 01-a

G. loss function G. Loss from probability of training set

Fraunhofer BIG DATA AI

Logistic Regression Classifier: Define Loss Function

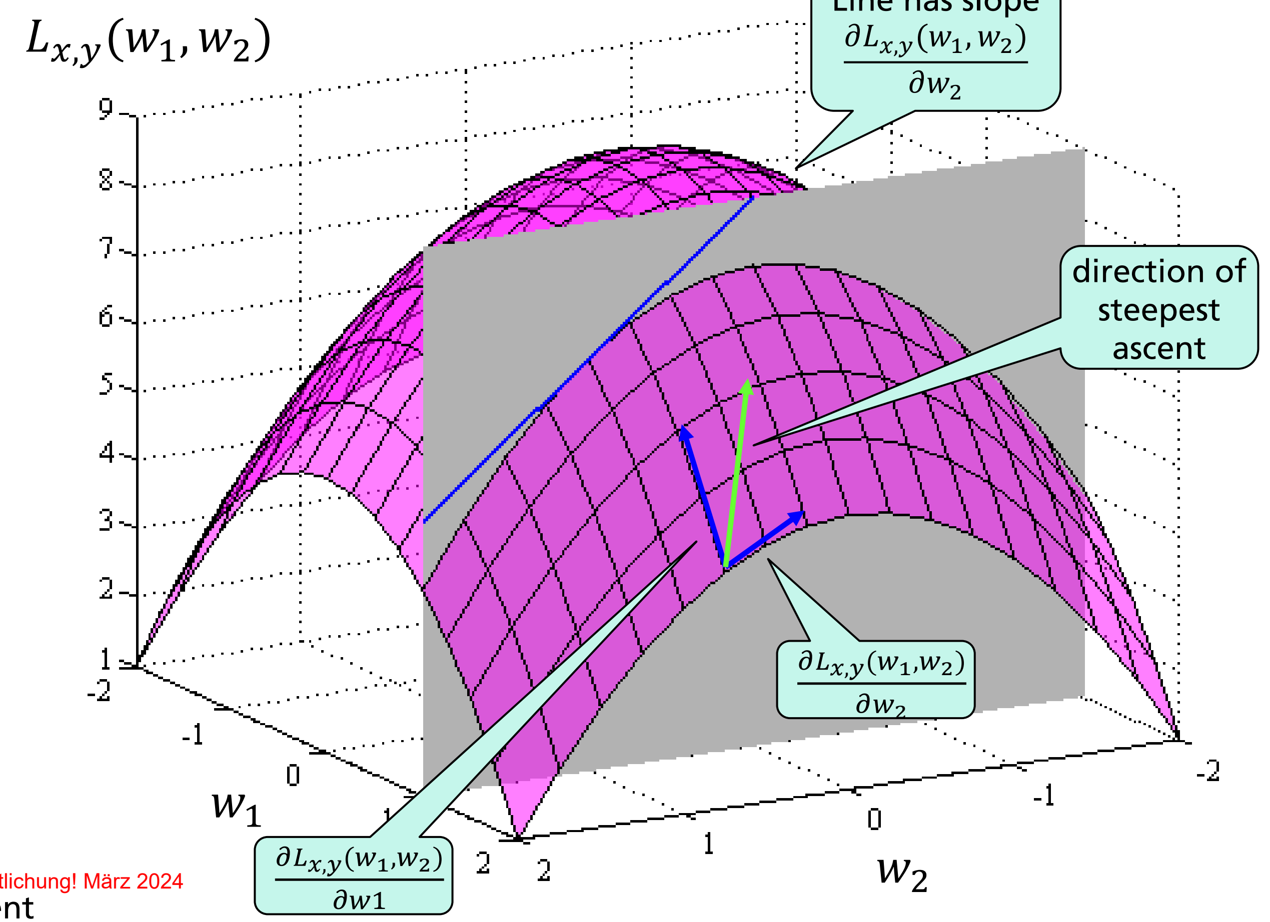
- How to find optimal parameters $w := (W, b)$

observed label

$$L_{x,y}(w) = -\log(p(TrainSet|w)) = -[\log \hat{p}(y = y_1|x_1; w) + \cdots + \log \hat{p}(y = y_{60000}|x_{60000}; w)]$$

- Initial value w: random

- Good approach:
  Use derivative info for finding optimum
  - partial derivative $\frac{\partial L_{x,y}(w_1,w_2)}{\partial w_1}$
    relative increase of loss $L_{x,y}(w)$ if w_1 is increased by small amount
  - sum of partial derivatives:
    gradient: direction of steepest ascent



Nicht zur Veröffentlichung! März 2024

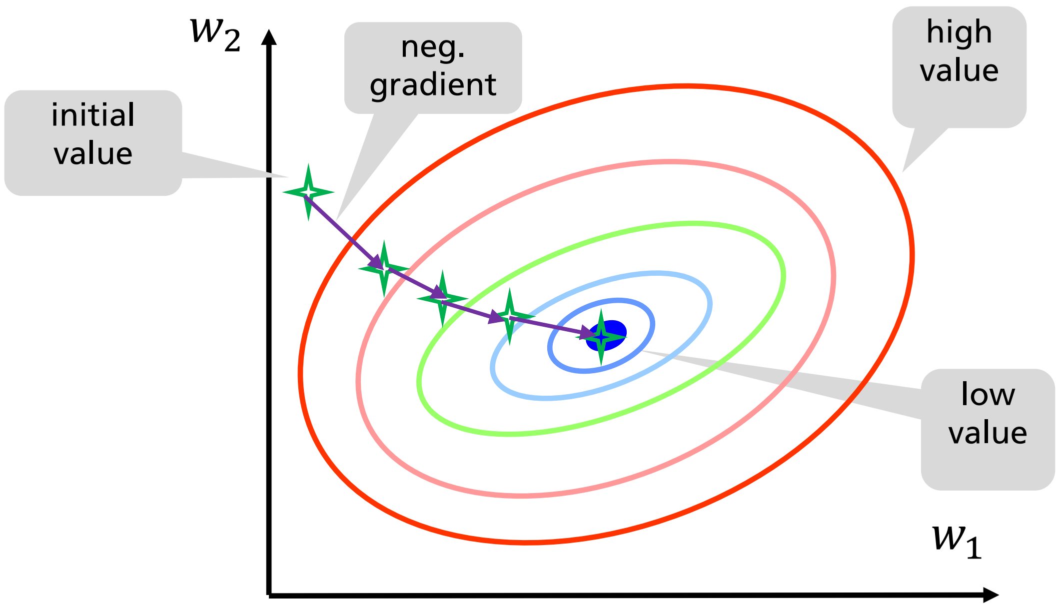
G. gradient

Logistic Regression Classifier: Find Optimal Parameters $\hat{w}$

Parameter Optimization: A method to find a value of parameter w such that $L_{x,y}(w)$ is minimal

- **Gradient** the direction (vector) where the function $L_{x,y}(w)$ has its steepest ascent at point w
- Consider the simple case that w has only two dimensions
- **Gradient Descent Optimization**
  - Initialize value $w = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$
  - Compute gradient for w $\quad \frac{\partial L_{x,y}(w)}{\partial w} = \begin{bmatrix} \frac{\partial L_{x,y}(w)}{\partial w_1} \\ \frac{\partial L_{x,y}(w)}{\partial w_1} \end{bmatrix}$
  - Set $\begin{pmatrix} w_1 \\ w_2 \end{pmatrix} := \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} - \lambda * \begin{pmatrix} \partial L/\partial w_1 \\ \partial L/\partial w_2 \end{pmatrix}$

    learning rate
  - Stop if changes are very low.



- In general yields **near optimal value**
  - Works well for very high dimensions

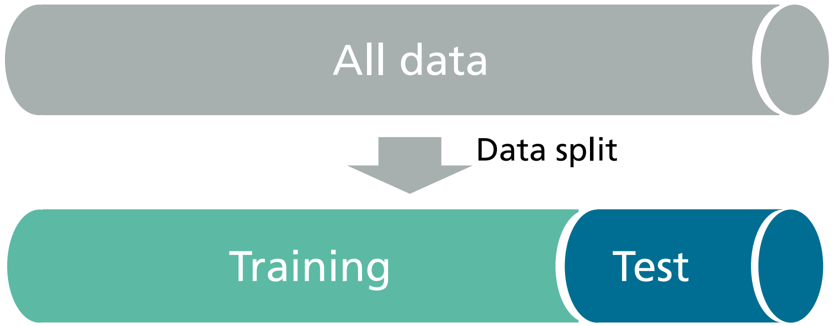
Visualization:

Details later

G. parameter optimization G. gradient descent optimization G. learning rate

Logistic Regression Classifier: Checking Performance

- How to determine the accuracy of a classifier?
- Compute performance measure on **separate test set**
  - Must not be used during training
    ➔ similar to new data



- Compute the fraction of correctly predicted class labels
  - **Accuracy** $= \dfrac{\text{number of correctly assigned test instances}}{\text{number of test instances}}$
  - On average accuracy correct for new unknown data

Logistic Regr. Classifier for MNIST Acc=92,4%

- **Precision**$_i$ $= \dfrac{\text{number of correctly assigned instances of class } i}{\text{number of all assigned instances of class } i}$
- **Recall**$_i$ $= \dfrac{\text{number of correctly assigned instances of class } i}{\text{number of all true instances of class } i}$
- **F-value**$_i$: harmonic mean of Precision$_i$ and Recall$_i$: $F = \dfrac{2 * precision_i * recall_i}{precision_i + recall_i}$

01-b

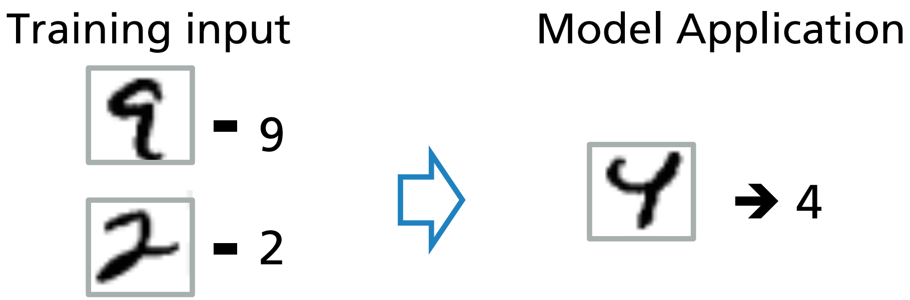
G. test set G. performance measure G. accuracy G. precision G. recall G. F-value

Introduction to Machine Learning and Deep Learning
Agenda

Machine Learning-Tasks by Availability of Data

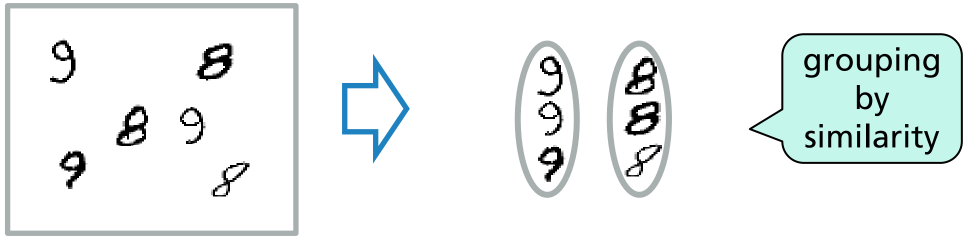
Supervised

- **Labelled** instances: The value of the output is known.
- Model predicts the output variable based on input variables.



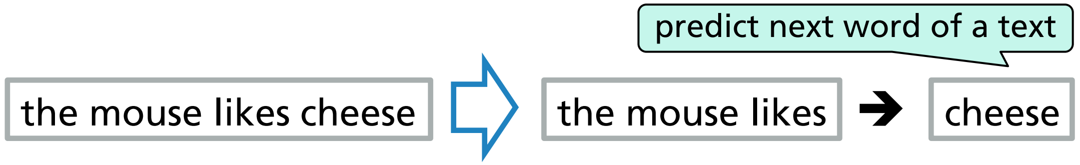
Unsupervised

- Instances **unlabeled**, only inputs, no outputs.
- Learn a model describing relationships among instances.



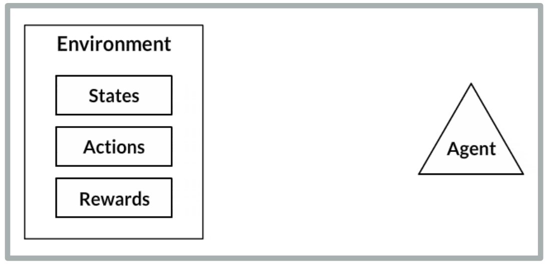
Self-Supervised

- Instances are **unlabeled** and contain different parts.
- Predict part of the instances from other parts.
- Can learn the internal structure of data.



Reinforcement Learning

- The agent specifies an action and receives a state and a reward at each time
- Only after a long time the final reward is available



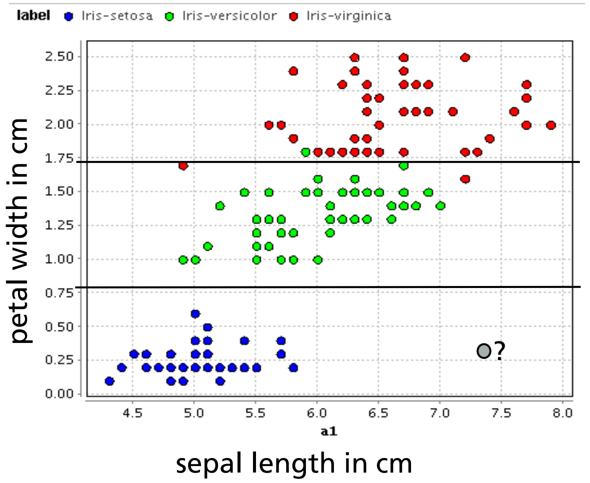
G. supervised learning G. unsupervised learning G. self-supervised learning G. dynamic control

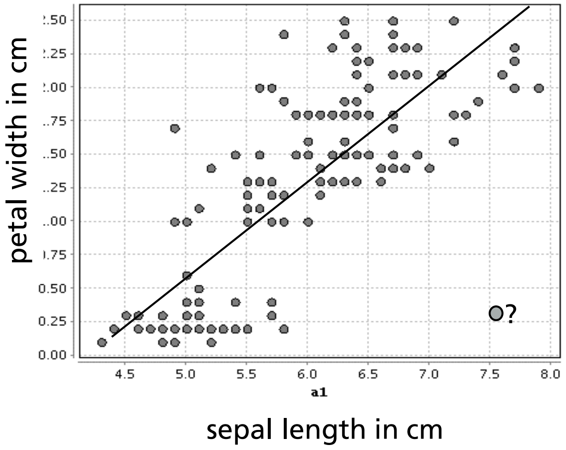
Machine Learning-Types by Form of Result

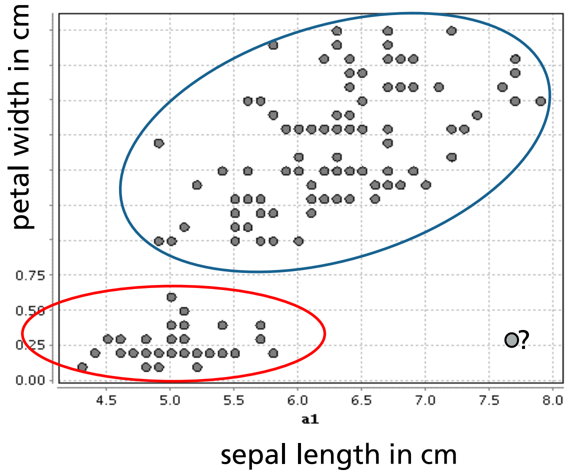
A number of pre-specified **classes**.
To which class does an instance belong?

Given a value of variable what is the **continuous** value of variable ?

Are there any **groups** of instances which are close to each other?







Classification

Regression

Clustering

Supervised

Unsupervised

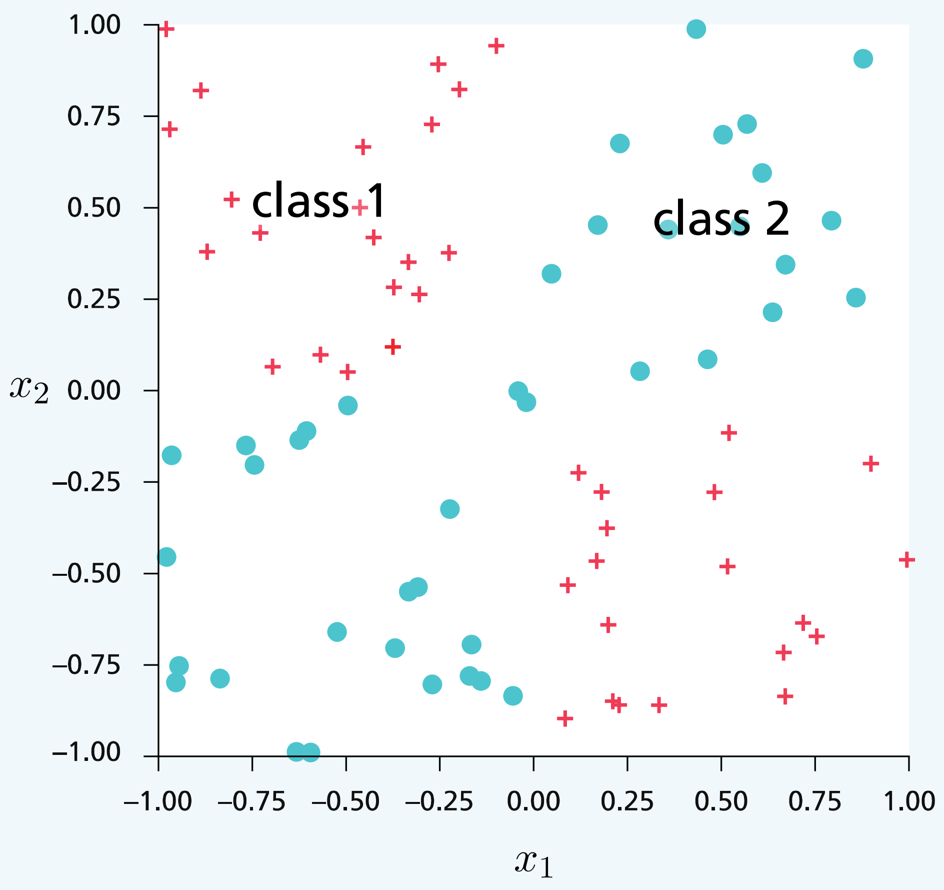
G. classification G. regression G. clustering

Fraunhofer BIG DATA AI

Introduction to Machine Learning and Deep Learning
Agenda

1. Deep Learning Success Stories
2. Application of Machine Learning
3. A Simple Model: Logistic Classification
4. Main Groups of Machine Learning Models
5. Deep Neural Network
6. Types of Deep Learning Models
7. Summary

Specific Class Setup



http://playground.tensorflow.org

class probabilities

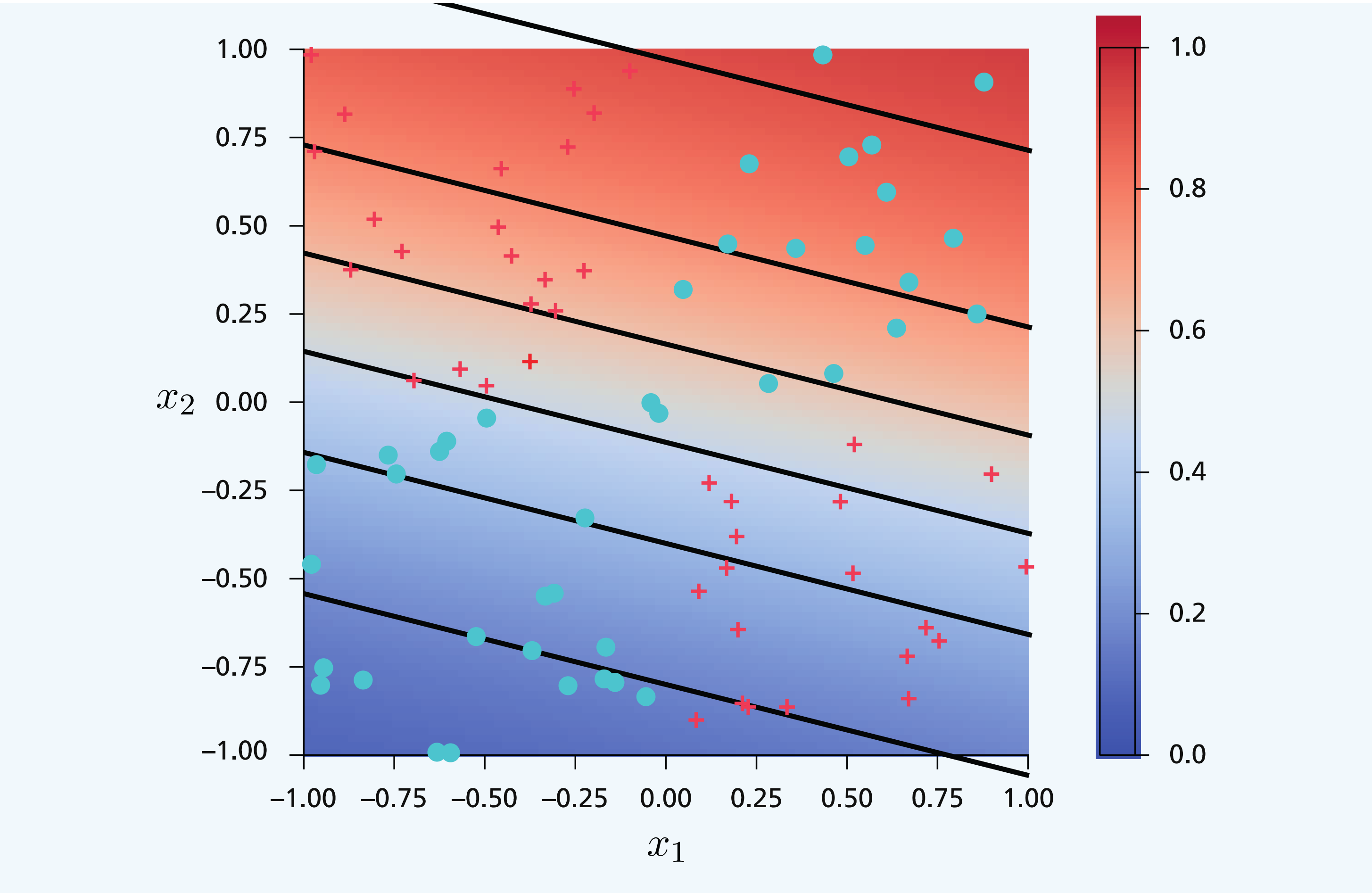
$$x = (x_1, x_2) \rightarrow W *\circ + b \rightarrow softmax(\circ) \rightarrow \begin{bmatrix}0.2\\0.8\end{bmatrix}$$

Nicht zur Veröffentlichung! März 2024

Linear Classifiers are Limited

- Best model with logistic regression
- Cannot separate classes in the data

- Representational power of logistic classifiers
  - cannot separate classes with nonlinear separating planes in the input space

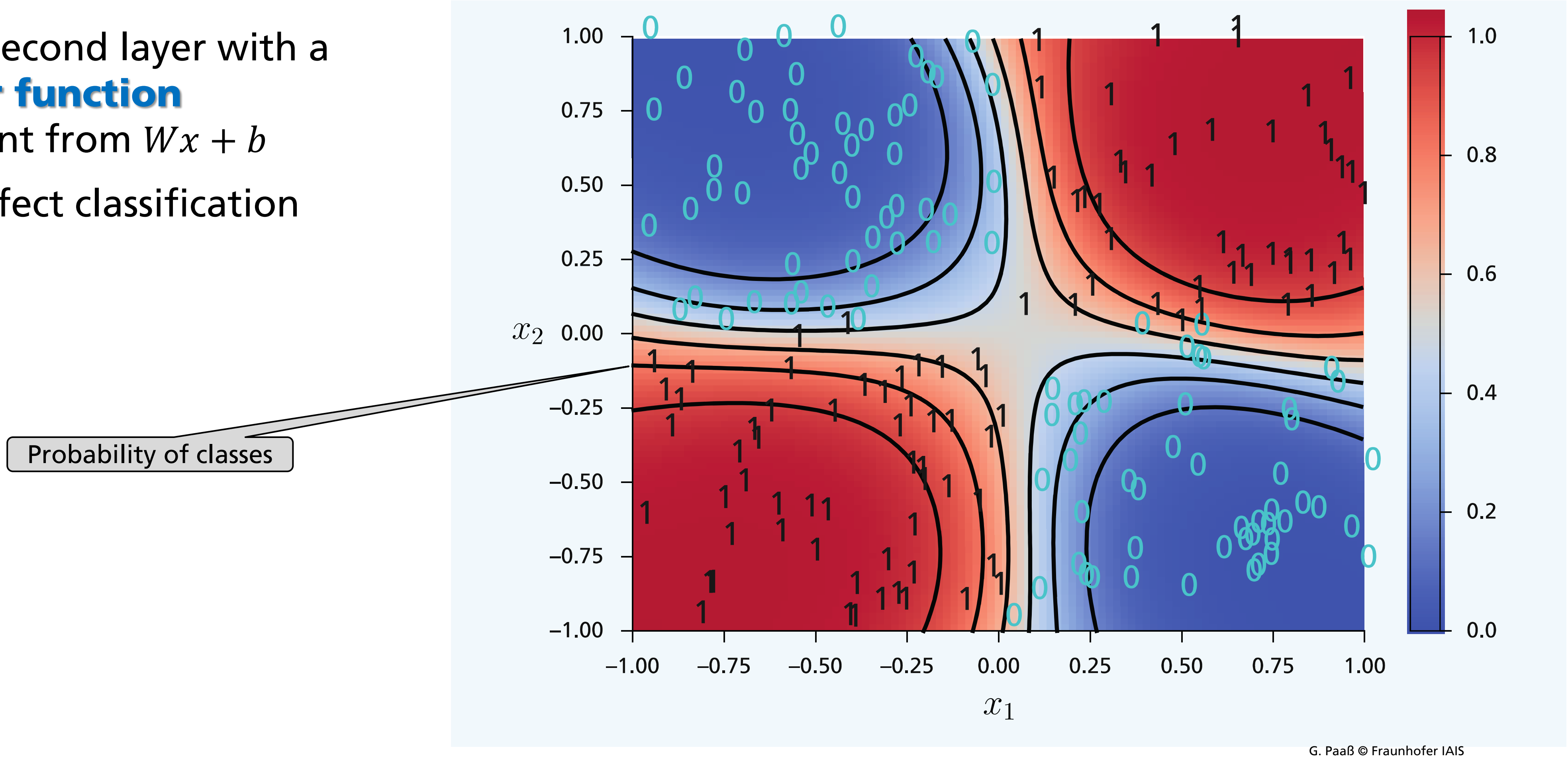


G. Paaß © Fraunhofer IAIS

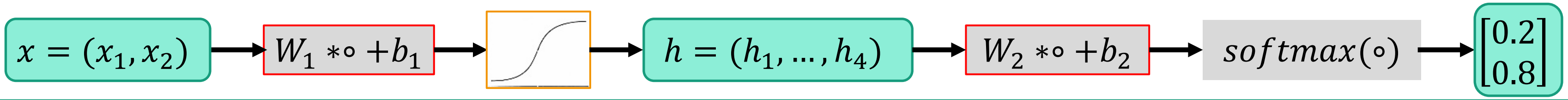
G. Representational power of logistic classifiers

The Effect of Nonlinearities

- Adding a second layer with a **nonlinear function** i.e. different from $Wx + b$
- Nearly perfect classification



G. Paaß © Fraunhofer IAIS



Nicht zur Veröffentlichung! März 2024

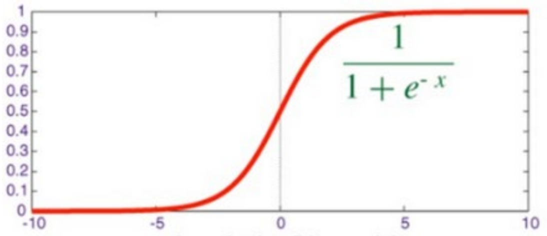
G. nonlinear function

Fraunhofer BIG DATA AI

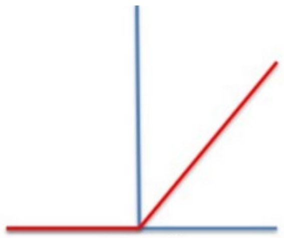
Linear Classifiers are Limited

functions different from $f(x) = Ax + b$

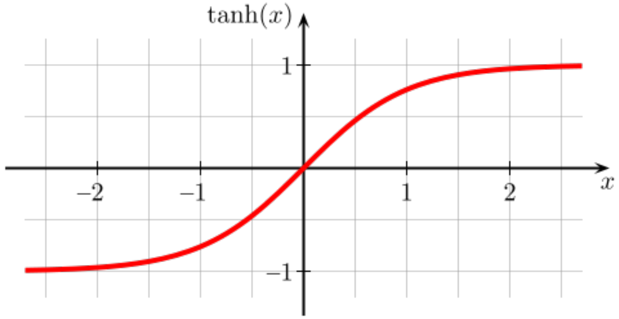
- Nonlinear **activation functions** are necessary
- Many types
  - Sigmoid $sig(x) = 1/(1 + \exp(-x))$
  - Rectified linear unit $ReLU(x) = \max(x, 0.0)$
  - Hyperbolic tangens $\tanh(x)$



Logistic Sigmoid



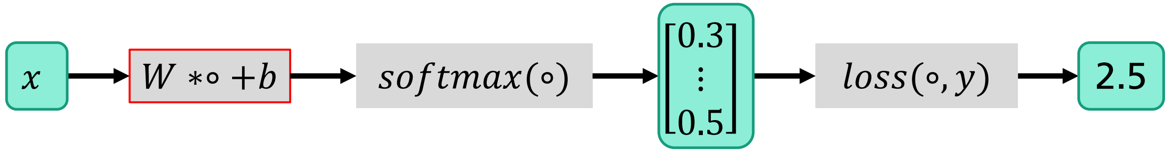
ReLU



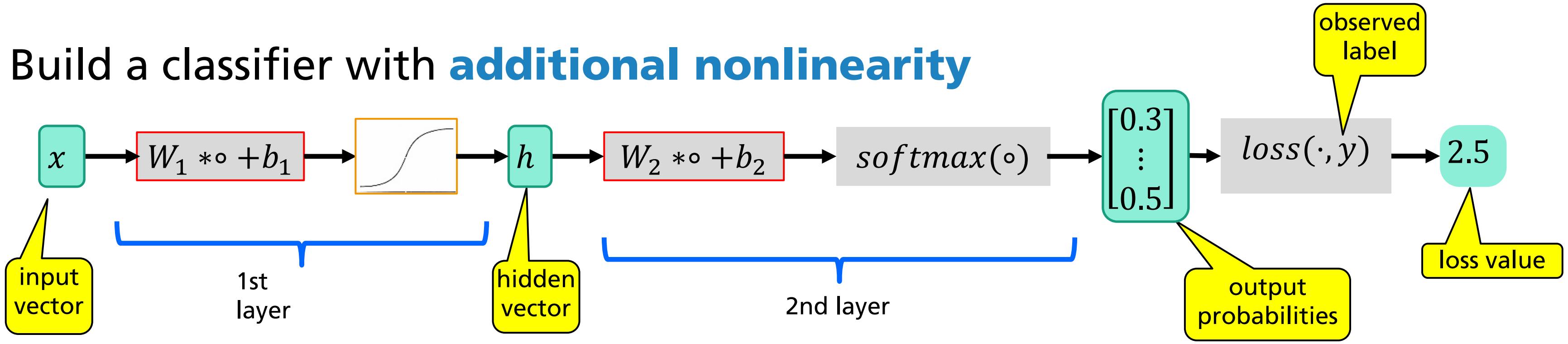
- apply nonlinear function to each component of a vector $(x_1, \cdots, x_k)$
  - e.g. Sigmoid $sig(x_1, \cdots, x_k) = \left(\frac{1}{1+\exp(-x_1)}, \cdots, \frac{1}{1+\exp(-x_k)}\right)$

 G. nonlinear activation function G. sigmoid function G. rectified linear unit G. hyperbolic tangent

Improving Neural Networks



- Build a classifier with **additional nonlinearity**



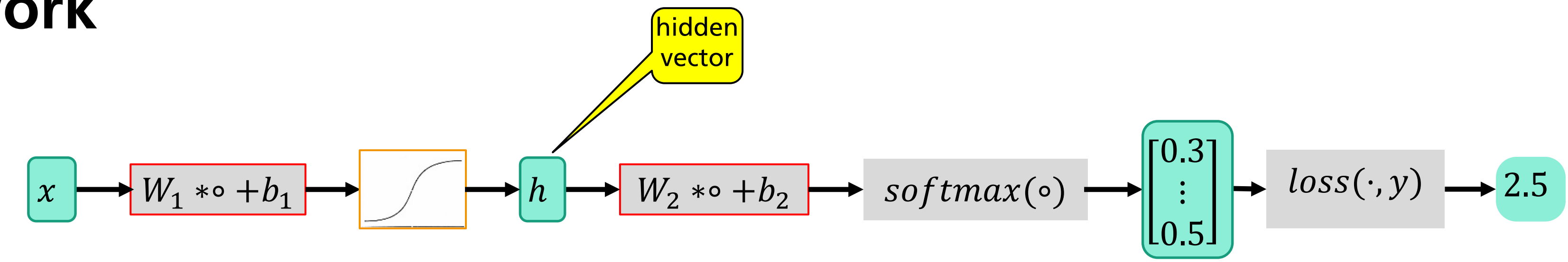
- The resulting function is **non-linear** because of the sigmoid
  - The dimension n_h of the **hidden vector** h may be selected
  - There are more parameters now: W_1, b_1, W_2, b_2
  - This is called a **multilayer neural network**

[Rumelhart & McClelland (1986)]

- Can approximate **arbitrary input-output relations** (with large enough hidden vector). This holds for all common types of non-linearities.

[Hornik, Stinchcombe & White (1989)]

Neural Network

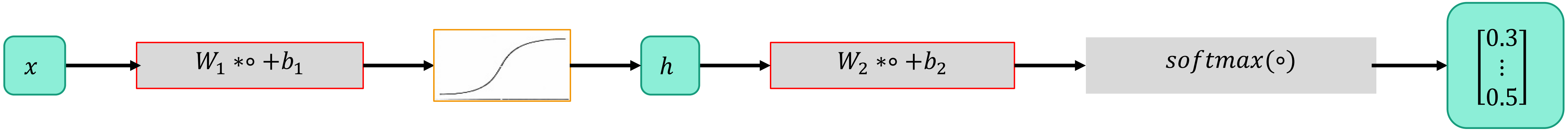


- The hidden vector h ➔ **new representation** of the input x
  - Learned during training
  - Enhances the computational power of the network
  - Contrast to conventional machine learning where features usually are specified by the data analyst

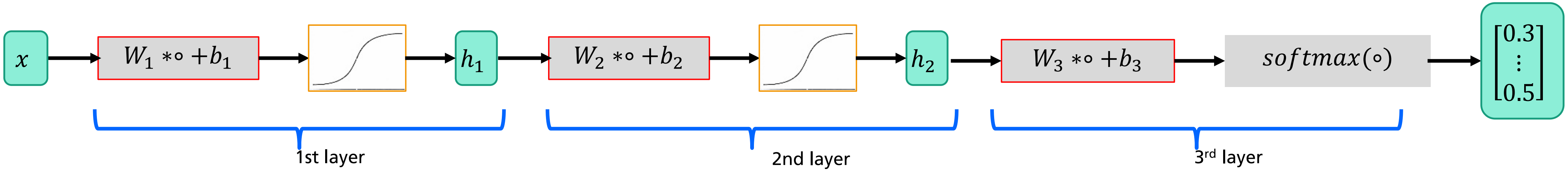
neural networks can construct new features

G. hidden vector as new features

Deep Neural Network



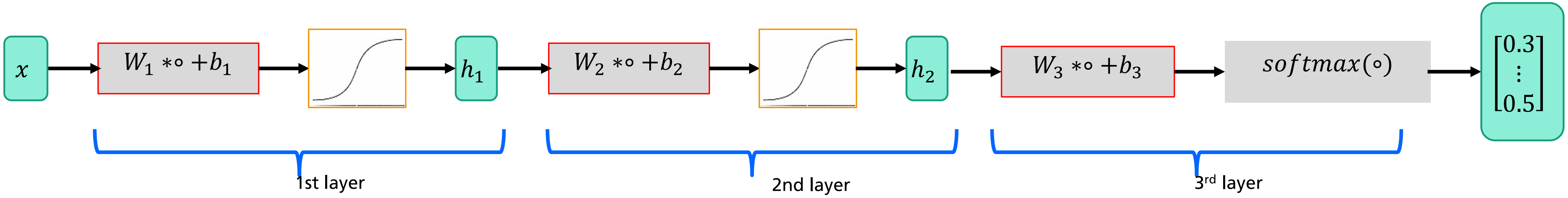
- 2-layer networks often require an excessive number of parameters.
- How to improve the capabilities of neural networks?
  - Add **additional layers** with nonlinearities: **deep neural network**



- Recent algorithms allow the training of such deep learning networks
- The **additional layers** generate more complex representations of input
  ➔ better than fewer layers with the same number of parameters
- With enough data deep neural networks show **very high performance**.

G. deep neural network G. advantage deep neural network

Multilayer Neural Network: Properties

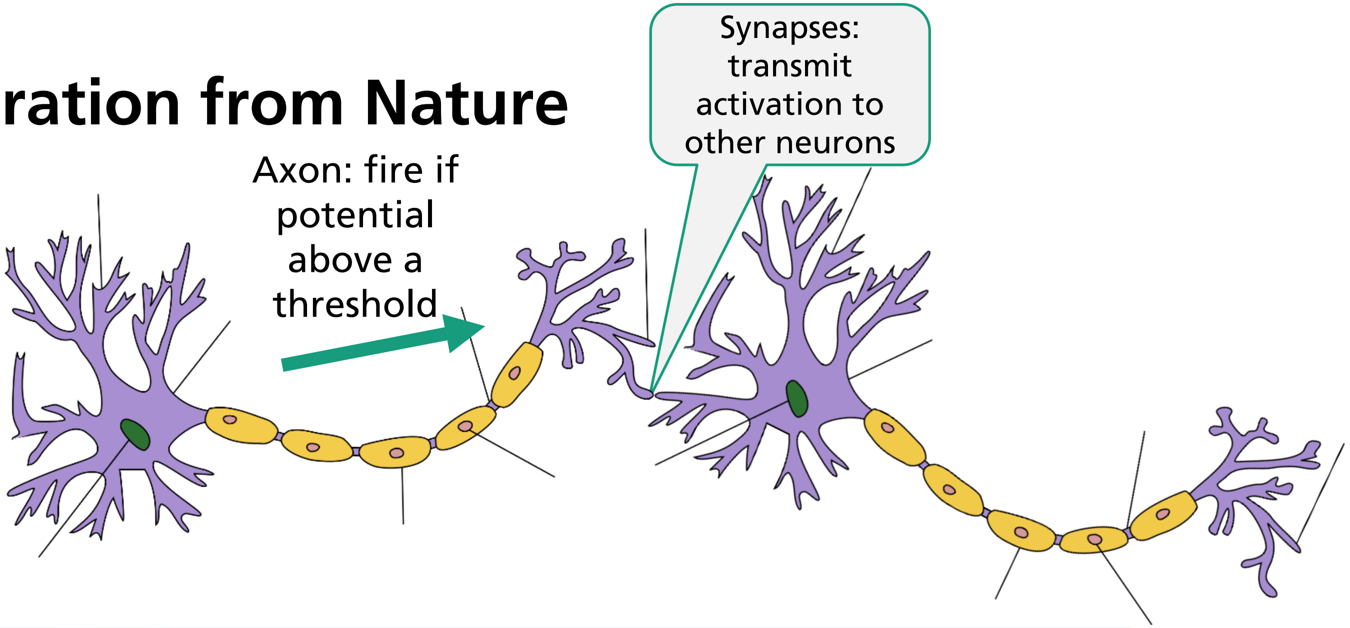


- Sequential execution of simple operators: multiply, add, sigmoid, softmax.
- Stability: Small input changes cause small output changes.
- **Gradients**
  - Can be derived **automatically** from the model: e.g. using the Chain Rule
  - Can be computed efficiently
- Need **high computational effort**

Details later

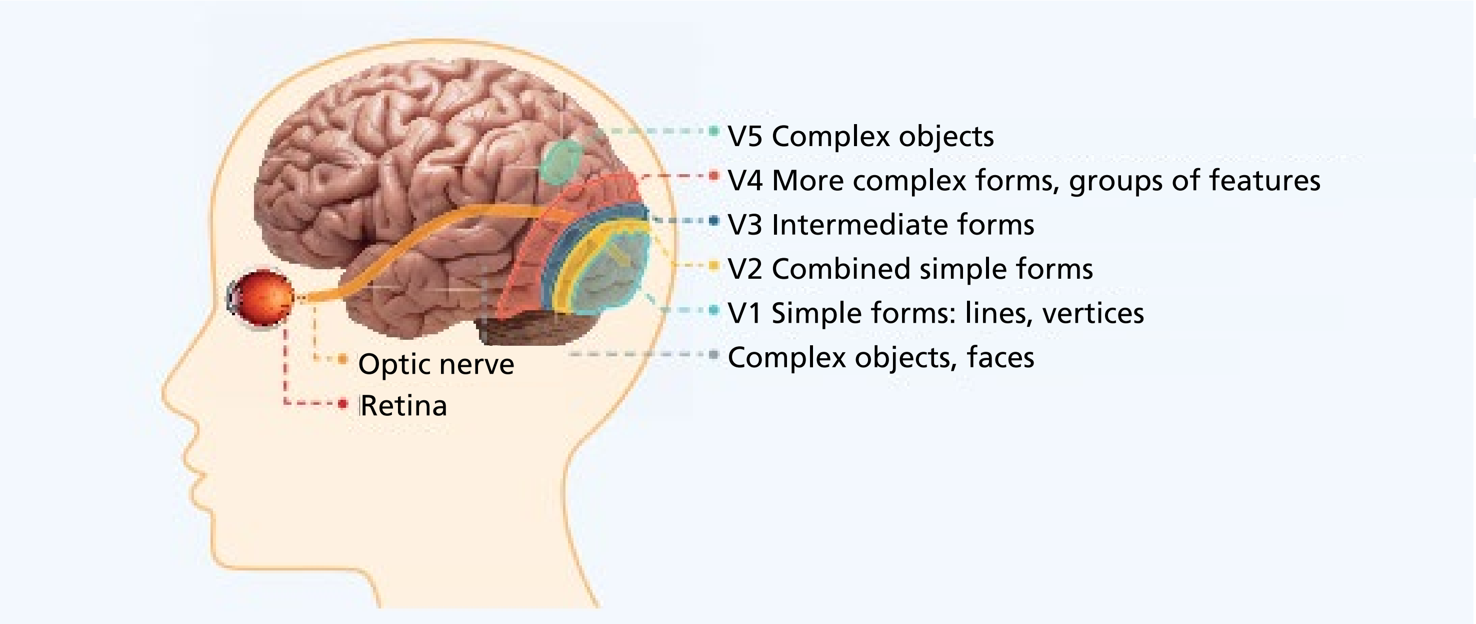
Multilayer Neural Network: Inspiration from Nature

- Neuron



Recreated File:Neuron-no labels2.png in Inkscape and hand-tuned to reduce filesize. Created by Quasar (talk) 19:59, 11 August 2009 (UTC) by Quasar Jarosz / CC BY-SA 3.0

- Visual Cortex is organized in layers



© Fraunhofer IAIS

? 01-b

G. Origin of the Term Neural Networks

Introduction to Machine Learning and Deep Learning
Agenda

Neural Network Types: Convolutional Neural Network

- **Convolutional layer:** Notion of a neighborhood: images, sequences

Alvar Aalto's birthplace by Jouni Mäki-Kahra / CC BY-SA 4.0

Input layer

- The same network (filterkernel) is swept over the input area.
- Repeat with a **large number k** of different networks: different features
  ➔ extract low-level characteristics: edges, circles, …
- Use **pooling layers** to reduce number of features:
  e.g. maximum of values in an area
- Many layers of convolution: deep neural network
- A final „global" layer usually implements a classifier

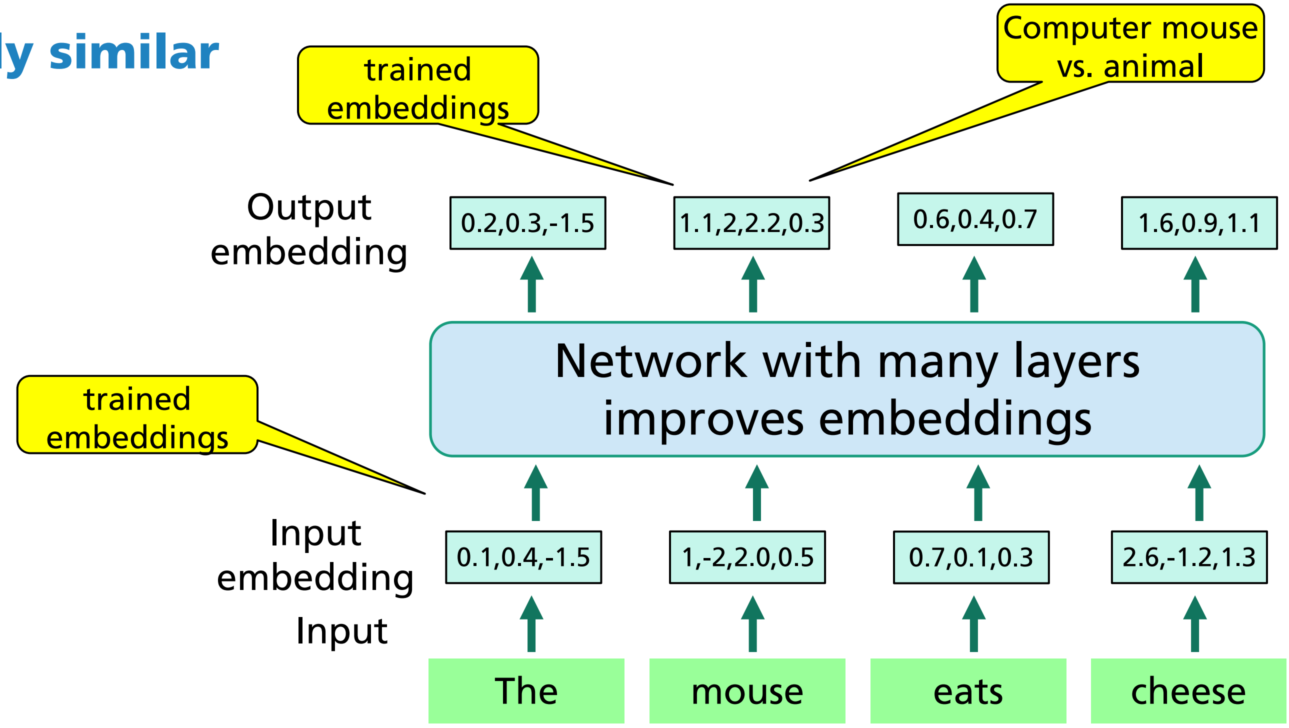
Details later

G. Filterkernel G. Convolutional layer G. pooling layer

Neural Network Types: Embedding Network

- To each word an **embedding vector** of size k is associated, e.g. $k = 100$
- Low distance of embeddings: corresponding words should be **semantically similar**
- Use information from all words of a text:
  - Improve input embeddings
  - Capture exact meaning of words
- No annotation required
  ➔ train on large text dataset

BERT



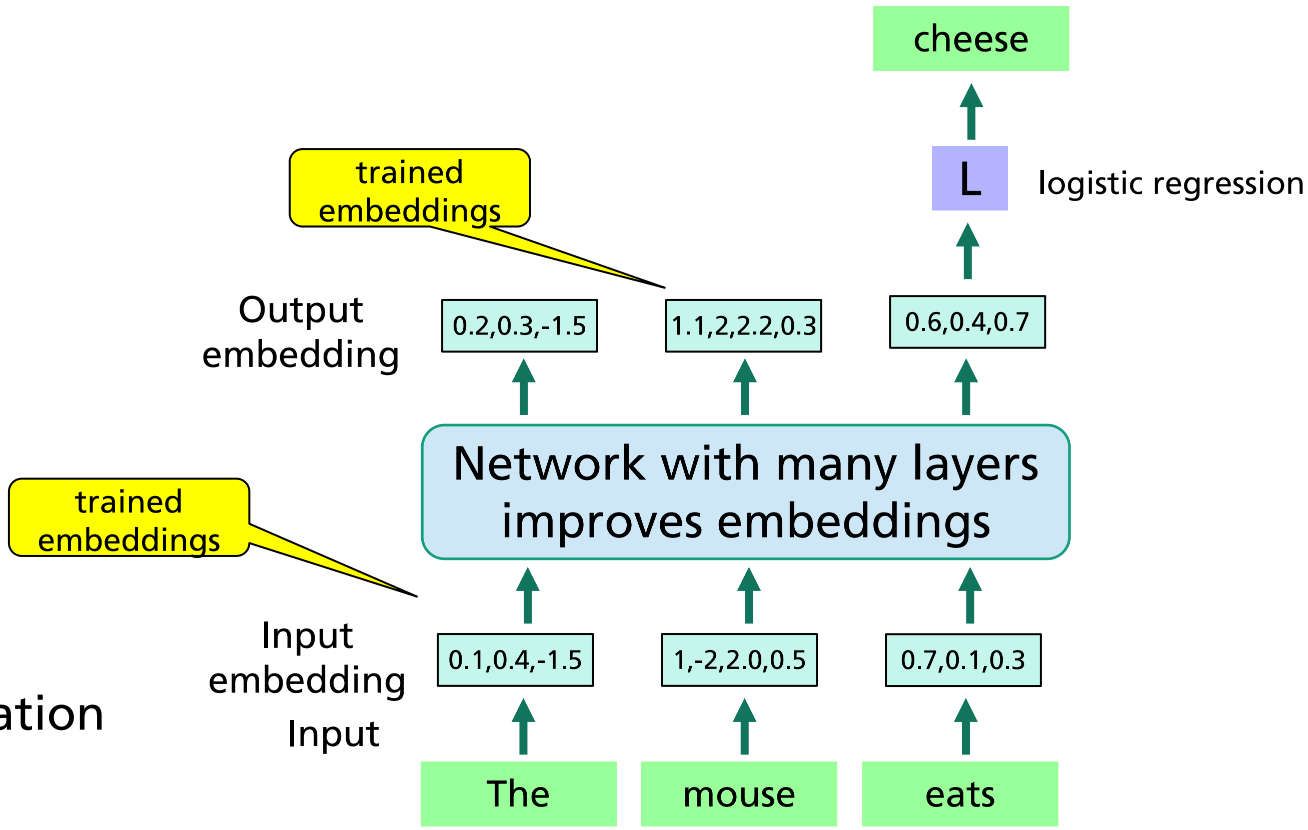
Details later

G. Word embedding G. embedding vector G. embedding network

Neural Network Types: Language Model

- **Language Model**
  - Task: predict the next word in a sentence

- Generate an embedding that contains all information about the next word
- Predict the probability distribution for the next word from the embedding
- Target: observed next word should get a maximal probability
  ➔ no annotation required,
- Many natural language applications
  language model, speech recognition, translation



Recurrent neural network

GPT

Nicht zur Veröffentlichung! März 2024

G. recurrent neural network

Fraunhofer BIG DATA AI

Neural Network Types: Reinforcement Learning

- **Dynamic Control**: Agent's actions affect subsequent data
  - No label is available, only a **delayed** „reward“
- Examples:
  - Game playing: chess, backgammon, go, atari video game
  - Manage an investment portfolio
  - Control a robot, a rolling mill, a self-driving car

[Chess pieces with colorful bokeh](#) by Mukumbura [CC BY-SA 2.0](#)

environment	state s_1	state reward	state s_3 reward r_3	state s_4 reward r_4
agent	action a_1	action a_2	action a_3	action a_4

- Deep learning: define value function $Value(action, state)$
  - ➔ maximal sum of rewards possible for $(action, state)$
  - Value function is complex: use deep neural network

©metamorworks - stock.adobe.com

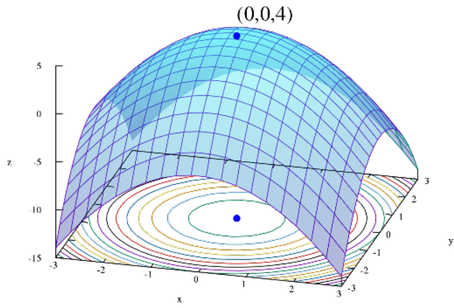
G. reinforcement learning task

Introduction to Machine Learning and Deep Learning
Agenda

Summary

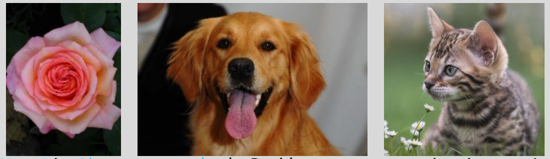
- **Machine Learning**: extract meaningful information from data.
  Steps:
  1. Compose a **model** function
  2. Define a **loss** function for the data:
     e.g. difference between predicted label and observed label
  3. **Minimize** the loss by optimization
  4. Apply the estimated model
- Deep Learning
  - compose a model from many modules: end-to-end differentiable
  - ➔ large **error reduction** in many pattern recognition tasks
- Advances in
  - Optimization algorithms
  - Annotated data
  - Computational hardware: GPUs
  - Available toolkits

Core Technology for Data Scientists

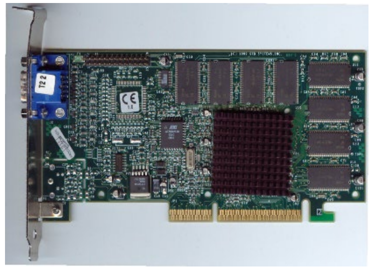


Graph of a given by $z = f(x, y) = -(x^2 + y^2) + 4$. The global maximum at $(x, y, z) = (0, 0, 4)$ is indicated by a blue dot. by IkamusumeFan CC BY-SA 4.0

ImageNet



"Rose" by Oberau-Online / CC BY 2.0 dog by David Locke (CC BY 2.0) andreaskrappweis / istockphoto.com



Voodoo3 2000 AGP card by Swaaye CC BY-SA 3.0

TensorFlow
TensorFlow Logo Apache License 2.0

Nicht zur Veröffentlichung! März 2024

G. machine learning steps

References

Blick 17, http://www.blick.ch/auto/news_n_trends/audi-q7-deep-learning-concept-das-auto-lernt-vom-fahrer-id6004071.html. Download on 06.01.2017

Castelvechii, D. "Deep Learning boosts Google translate Tool." Nature, 27.Sep. 2016

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. EMNLP 2014

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE International Conference on Computer Vision (pp. 1026-1034).

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. Neural networks, 2(5), 359-366.

Martin, J. "Why you might want to hold off buying Google Home" http://www.cio.com/. 7.Nov.16

Mikolov, T., and J. Dean. "Distributed representations of words and phrases and their compositionality." Advances in neural information processing systems (2013).

Microsoft NLP-group (2017): R-NET: Machine reading Comprehension with self-matching Networks

Reiley, C. "Deep Driving", MIT Technological Review, 18.Oct. 2016

Rumelhart, D.E; James McClelland (1986). Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Cambridge: MIT Press.

Schuster, M. "Zero-Shot Translation with Google's Multilingual Neural Machine Translation System", Google Research Blog , 22. Nov. 2016

Shazeer et al. 2017: Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer.

Silver, D. et al. (2016): Mastering the game of Go with deep neural networks and tree search. Nature 529, 484–489

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Advances in neural information processing systems (pp. 3104-3112).

Van Essen, D. C., & Gallant, J. L. (1994). Neural mechanisms of form and motion processing in the primate visual system. Neuron, 13(1), 1-10.

Xiong, W., et al. "Achieving Human Parity in Conversational Speech Recognition." arXiv preprint arXiv:1610.05256 (2016).

Xiong, W. et al. "The Microsoft 2017 Conversational Speech Recognition System.

Zhang, C., et al. "Mining User Intentions from Medical Queries: A Neural Network Based Heterogeneous Jointly Modeling Approach." WWW 2016

Disclaimer

Responsible contact: **Dr. Gerhard Paaß**, Fraunhofer IAIS, Sankt Augustin
E-mail: gerhard.paass@iais.fraunhofer.de