

Wie denken denkende Maschinen?

Swiss ICT Symposium, KKL Luzern, 14.11.2017

Thilo Stadelmann



Swiss Alliance for
Data-Intensive Services



datalab
www.zhaw.ch/datalab

Was? → Wie? → Wo?

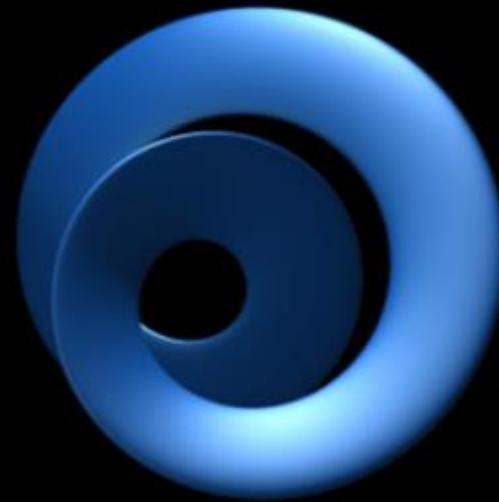


1

Was ist passiert?
(Eine kurze Geschichte der letzten Monate)

Google Acquires Artificial Intelligence Startup DeepMind For More Than \$500M

Posted Jan 26, 2014 by Catherine Shu (@catherineshu)

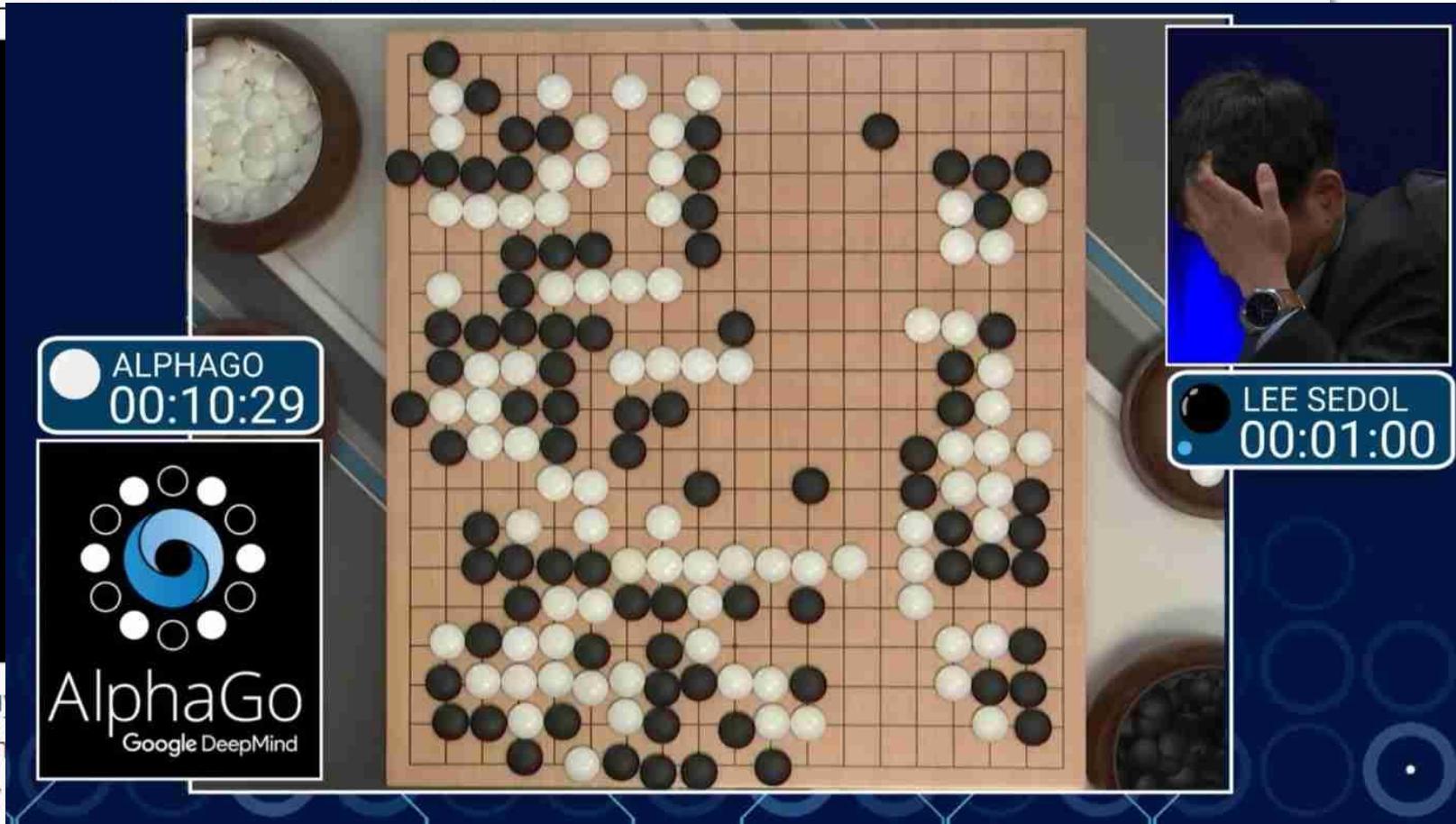


Google will buy London-based artificial intelligence company DeepMind. The Information reports that the acquisition price was more than \$500 million, and that Facebook was also in talks to buy the startup late last year. DeepMind confirmed the acquisition to us, but couldn't disclose deal terms.

The acquisition was originally confirmed by Google to Re/code.

Google Acquires Artificial Intelligence Startup DeepMind For More Than \$500M

Posted Jan 26, 2014 by Catherine Shu (@catherineshu)



Google will buy reports that they in talks to buy couldn't disclose deal terms.

The acquisition was originally confirmed by Google to Re/code.

Google Acquires Artificial Intelligence Startup DeepMind For More Than \$500M

Posted Jan 26, 2014 by Catherine Shu (@catherineshu)



Google will buy reports that they in talks to buy couldn't disclose deal terms.

The acquisition was originally confirmed by Google to Re/code.



Google Acquires Artificial Intelligence Startup DeepMind For More Than \$500M

Posted Jan 26, 2014 by Catherine Shu (@catherineshu)



The graph illustrates the rapid growth of AlphaGo Zero's Elo rating over a 40-day period. The Y-axis represents the Elo Rating, ranging from -2000 to 5000. The X-axis represents time in days, from 0 to 40. Three data series are plotted: AlphaGo Zero 40 blocks (blue line), AlphaGo Lee (green dots), and AlphaGo Master (blue dots). AlphaGo Zero 40 blocks starts at approximately -1800 and rises sharply to about 4800 by day 10, then continues to rise more gradually to nearly 5200 by day 40. AlphaGo Lee and AlphaGo Master are shown as horizontal dashed lines at approximately 3500 and 4700 respectively.

40 days

AlphaGo Zero surpasses all other versions of AlphaGo and, arguably, becomes the best Go player in the world. It does this entirely from self-play, with no human intervention and using no historical data.

— AlphaGo Zero 40 blocks ••••• AlphaGo Lee ••••• AlphaGo Master

Alphago
Google DeepMind

At last – a computer program that can beat a champion Go player **PAGE 484**

ALL SYSTEMS GO

CONSERVATION
SONGBIRDS A LA CARTE
Illegal harvest of millions of Mediterranean birds
PAGE 452

RESEARCH ETHICS
SAFEGUARD TRANSPARENCY
Don't let openness backfire on individuals
PAGE 459

POPULAR SCIENCE
WHEN GENES GOT 'SELFISH'
Dawkins's cutting card forty years on
PAGE 462

NATURE.COM/NATURE
28 January 2016 410
Vol. 529 No. 7587

047

The acquisition was originally confirmed by Google to Re/code.



Computing

Algorithm Clones Van Gogh's Artistic Style and Pastes It onto Other Images, Movies

A deep neural network has learned to transfer artistic styles to other images.

by Emerging Technology from the arXiv May 10, 2016

The nature of artistic style is something of a mystery to most people. Think of Vincent Van Gogh's *Starry Night*, Picasso's work on cubism, or Edvard Munch's *The Scream*. All have a powerful, unique style that humans recognize easily.



Computing

Algorithm Clones Van Gogh's Artistic Style and Pastes It onto Other Images, Movies

A deep neural network has learned to transfer artistic styles to other images.

by Emerging Technology from the arXiv May 10, 2016

The nature of artistic style is something of a mystery to most people. Think of Vincent Van Gogh's *Starry Night*, or Edvard Munch's *The Scream*. Humans recognize easily.



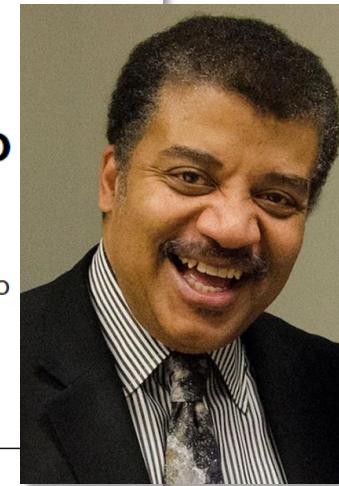


Computing

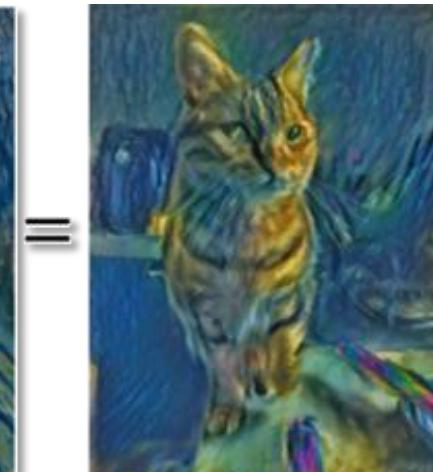
Algorithm Clones Van Gogh's Artistic Style and Pastes It onto Other Images, Movies

A deep neural network has learned to transfer artistic styles to other images.

by Emerging Technology from the arXiv May 10, 2016



The nature of artistic style is something of a mystery to most people. Think of Vincent Van Gogh's *Starry Night*, or Edvard Munch's *The Scream*. Humans recognize easily.





Computing

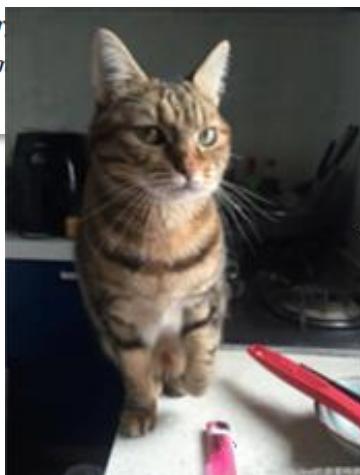
Algorithm Clones Van Gogh's Artistic Style and Pastes It onto Other Images, Movies

A deep neural network has learned to transfer artistic styles to other images.

by Emerging Technology from the arXiv May 10, 2016



The nature of artistic style is something of a mystery to most people. Think of Vincent Van Gogh's *Starry Night*, or Edvard Munch's *The Scream*. Humans recognize easily.





Deep neural networks can now transfer the style of one photo onto another

And the results are impressive

by James Vincent | @jvincent | Mar 30, 2017, 1:53pm EDT

f SHARE

t TWEET

in LINKEDIN

Computing

Algorithm
Artistic
Other In

A deep neural n
other images.

by Emerging Tech

The nature of art
of Vincent Van C
Edvard Munch's
humans recogni



Original photo

Reference photo

Result

You've probably heard of an AI technique known as "style transfer" — or, if you haven't heard of it, you've seen it. The process uses neural networks to apply the look and feel of one image to another, and appears in apps like [Prisma](#) and [Facebook](#). These style transfers, however, are stylistic, not photorealistic. They look good because they look like they've been painted. Now a group of researchers from Cornell University and Adobe have augmented

Ad closed by Google

Report this ad

AdChoices ▾



NOW TRENDING

...und die Liste liesse sich fortsetzen!

 Brandon Amos About Blog



Image Completion with Deep Learning in TensorFlow

August 9, 2016



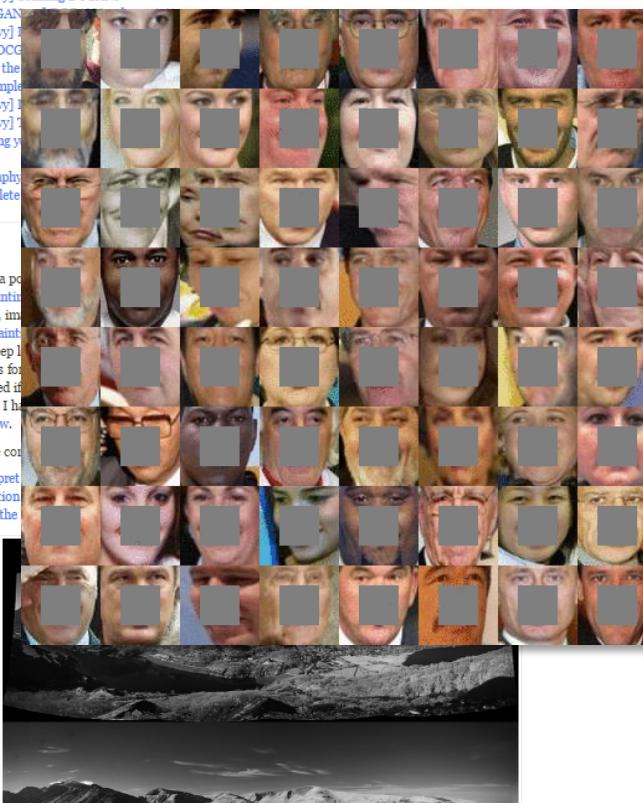
- Introduction
- Step 1: Interpreting images as samples from a probability distribution
 - How would you fill in the missing information?
 - But where does statistics fit in? These are images.
 - So how can we complete images?
- Step 2: Quickly generating fake images
 - Learning to generate new samples from an unknown probability distribution
 - [ML-Heavy] Generative Adversarial Net (GAN) building blocks
 - Using $G(z)$ to produce fake images
 - [ML-Heavy] Training DCGANs
 - Existing GANs
 - [ML-Heavy] Implementing DCGANs
 - Running DCGANs
- Step 3: Finding the right completion
 - Image completion
 - [ML-Heavy] 1
 - [ML-Heavy] 2
 - Completing your face
- Conclusion
- Partial bibliography
- Bonus: Incomplete images

Introduction

Content-aware fill is a popular technique for image completion and inpainting. It does content-aware fill, image completion, and semantic image inpainting. Semantic Image Inpainting shows how to use deep learning to do content-aware fill. Some deeper portions of this section can be skipped if you're not interested in learning from images of faces. If you are interested in learning about image completion, then you should skip this section.

We'll approach image completion in three steps:

1. We'll first interpret images as samples from a probability distribution.
2. This interpretation will allow us to quickly generate fake images.
3. Then we'll find the right completion.



...und die Liste liesse sich fortsetzen!

Brandon Amos [About](#) [Blog](#)

[Image Completion with Deep Learning in TensorFlow](#)
August 9, 2016

- [Introduction](#)
- [Step 1: Interpreting images as samples from a probability distribution](#)
 - How would you fill in the missing information?
 - But where does statistics fit in? These are images.
 - So how can we complete images?
- [Step 2: Quickly generating fake images](#)
 - Learning to generate new samples from an unknown probability distribution
 - [ML-Heavy] Generative Adversarial Net (GAN) building blocks
 - Using $G(z)$ to produce fake images
 - [ML-Heavy] Training DCGANs
 - Existing GANs
 - [ML-Heavy] 1
 - [ML-Heavy] 2
 - Running DCGANs
- [Step 3: Finding the right model](#)
 - Image completion
 - [ML-Heavy] 1
 - [ML-Heavy] 2
 - [ML-Heavy] 3
 - Completing your images
- [Conclusion](#)
- [Partial bibliography](#)
- [Bonus: Incomplete](#)

Introduction

Content-aware fill is a powerful technique for image completion and inpainting. It can do content-aware fill, image completion, semantic image inpainting, and more. "Semantic Image Inpainting with Generative Adversarial Networks" shows how to use deep learning to fill in some deeper portions of images. This section can be skipped if you're not interested in learning about image completion with TensorFlow.

We'll approach image completion in three steps:

1. We'll first interpret what's missing.
2. This interpretation will help us find the right model.
3. Then we'll find the right model.

The Unreasonable Effectiveness of Recurrent Neural Networks

May 21, 2015

There's something magical about Recurrent Neural Networks (RNNs). I still remember when I trained my first recurrent network for image Captioning. Within a few dozen minutes of training my first baby model (with rather arbitrarily-chosen hyperparameters) started to generate very nice looking descriptions of images that were on the edge of making sense. Sometimes the ratio of how simple your model is to the quality of the results you get out of it blows past your expectations, and this was one of those times. What made this result so shocking at the time was that the common wisdom was that RNNs were supposed to be difficult to train (with more experience I've in fact reached the opposite conclusion). Fast forward about a year. I'm training RNNs all the time and I've witnessed their power and robustness many times, and yet their magical outputs still find ways of amusing me. This post is about sharing some of that magic with you.

We'll train RNNs to generate text character by character and ponder the question "how is that even possible?"

By the way, together with this post I am also releasing code on [Github](#) that allows you to train character-level language models based on multi-layer LSTMs. You give it a large chunk of text and it will learn to generate text like it one character at a time. You can also use it to reproduce my experiments below. But we're getting ahead of ourselves; What are RNNs anyway?

Recurrent Neural Networks

Sequences. Depending on your background you might be wondering: What makes Recurrent Networks so special? A glaring limitation of Vanilla Neural Networks (and also Convolutional Networks) is that their API is too constrained: they accept a fixed-sized vector as input (e.g. an image) and produce a fixed-sized vector as output (e.g. probabilities of different classes). Not only that, these models perform this mapping using a fixed amount of computational steps (e.g. the number of layers in the model). The core reason that Recurrent nets are more exciting is that they allow us to operate over sequences of vectors. Sequences in the input, the output, or in the most general case both. A few examples may make this more concrete:

VIOLA:

Why, Salisbury must find his flesh and thought
That which I am not aps, not a man and in fire,
To show the reining of the raven and the wars
To grace my hand reproach within, and not a fair are hand,
That Caesar and my goodly father's world;
When I was heaven of presence and our fleets,
We spare with hours, but cut thy council I am great,
Murdered and by thy master's ready there
My power to give thee but so much as hell:
Some service in the noble bondman here,
Would show him to her wine.

KING LEAR:

O, if you were a feeble sight, the courtesy of your law,
Your sight and several breath, will wear the gods
With his heads, and my hands are wonder'd at the deeds,
So drop upon your lordship's head, and your opinion
Shall be against your honour.

On the left, a recurrent network generates images of digits by learning to sequentially add color to a canvas (Gregor Elia); on the right, a recurrent network generates images of digits by learning to sequentially add color to a canvas (Gregor Elia);



...und die Liste liesse sich fortsetzen!

Brandon Amos [About](#) [Blog](#)

[Facebook](#) [Twitter](#) [Google+](#) [RSS](#)

Image Completion with Deep Learning in TensorFlow

August 9, 2016

[Twitter](#) [Facebook](#) [Google+](#) [LinkedIn](#) [StumbleUpon](#) [Email](#)

- [Introduction](#)
- [Step 1: Interpreting images as samples from a probability distribution](#)
 - How would you fill in the missing information?
 - But where does statistics fit in? These are images.
 - So how can we complete images?
- [Step 2: Quickly generating fake images](#)
 - Learning to generate new samples from an unknown probability distribution
 - [ML-Heavy] Generative Adversarial Net (GAN) building blocks
 - Using $G(z)$ to produce fake images
 - [ML-Heavy] Training DCGANs
 - Existing GANs
 - [ML-Heavy] 1
 - Running DCGANs
- [Step 3: Finding the right completion](#)
 - Image completion
 - [ML-Heavy] 1
 - [ML-Heavy] 2
 - Completing your face
- [Conclusion](#)
- [Partial bibliography](#)
- [Bonus: Incomplete](#)

Introduction

Content-aware fill is a powerful technique for image completion and inpainting. It allows us to fill in missing parts of an image while maintaining visual coherence. In this post, I'll show how to use deep learning to perform content-aware fill, inspired by the work of ["Semantic Image Inpainting"](#). This section can be skipped if you're not interested in learning about image completion with TensorFlow. We'll approach image completion in three steps:

1. We'll first interpret the image.
2. This interpretation will help us find the right completion.
3. Then we'll find the completion.

Well approach image completion in three steps:

1. We'll first interpret the image.
2. This interpretation will help us find the right completion.
3. Then we'll find the completion.

The Unreasonable Effectiveness of Recurrent Neural Networks

May 21, 2015

There's something magical about Recurrent Neural Networks (RNNs). I still remember when I trained my first recurrent network for image captioning. Within a few dozen minutes of training my first baby model (with rather arbitrarily-chosen hyperparameters) started to generate very nice looking descriptions of images that were on the edge of making sense. Sometimes the ratio of how simple your model is to the quality of the results you get out of it blows past your expectations, and this was one of those times. What made this result so shocking at the time was that the common wisdom was that RNNs were supposed to be difficult to train (with more experience I've in fact reached the opposite conclusion). Fast forward about a year: I'm training RNNs all the time and I've witnessed their power and robustness many times, and yet their magical outputs still find ways of amusing me. This post is about sharing some of that magic with you.

We'll train RNNs to generate text character by character and ponder the question "how is that even possible?"

By the way, together with this post I am also releasing code on [GitHub](#) that allows you to train character-level language models based on multi-layer LSTMs. You give it a large chunk of text and it will learn to generate text like it one character at a time. You can also use it to reproduce my experiments below. But we're getting ahead of ourselves; What are RNNs anyway?

Recurrent Neural Networks

Sequences. Depending on your background you might be wondering: What makes Recurrent Neural Networks so special? A glaring limitation of Vanilla Neural Networks (and also Convolutional Networks) is that their API is too constrained: they accept a fixed-sized vector as input (e.g. an image), and produce a fixed-sized vector as output (e.g. probabilities of different classes). Not only that: These models perform this mapping using a fixed amount of computational steps (e.g. the number of layers in the model). The core reason that recurrent nets are more exciting is that they allow us to operate over sequences of vectors. Sequences in the input, the output, or in the most general case both. A few examples may make this more concrete:

VIOLA:

Why, Salisbury must find his flesh and thought
That which I am not aps, not a man and in fire,
To show the reining of the raven and the wars
To grace my hand reproach within, and not a fair are hand,
That Caesar and my goodly father's world;
When I was heaven of presence and our fleets,
We spare with hours, but cut thy council I am great,
Murdered and by thy master's ready there
My power to give thee but so much as hell:
Some service in the noble bondman here,
Would show him to her wine.

KING LEAR:

O, if you were a feeble sight, the courtesy of your law,
Your sight and several breath, will wear the gods
With his heads, and my hands are wonder'd at the deeds,
So drop upon your lordship's head, and your opinion
Shall be against your honour.

On the left, a recurrent neural network generates images of digits by learning to sequentially add color to a canvas (Gregor El al.); On the right, a recurrent network generates images of digits by learning to sequentially add color to a canvas (Gregor El al.);



the morning paper

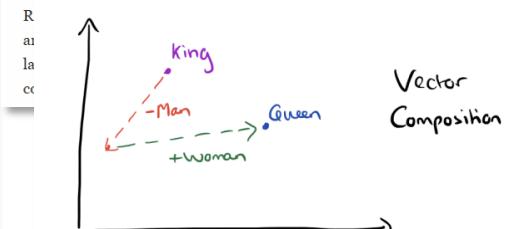
The amazing power of word vectors

APRIL 21, 2016

For today's post, I've drawn material not just from one paper, but from five! The subject matter is 'word2vec' – the work of Mikolov et al. at Google on efficient vector representations of words (and what you can do with them). The papers are:

- ★ [Efficient Estimation of Word Representations in Vector Space](#) – Mikolov et al. 2013
- ★ [Distributed Representations of Words and Phrases and their Compositionality](#) – Mikolov et al. 2013
- ★ [Linguistic Regularities in Continuous Space Word Representations](#) – Mikolov et al. 2013
- ★ [word2vec Parameter Learning Explained](#) – Rong 2014
- ★ [word2vec Explained: Deriving Mikolov et al.'s Negative Sampling Word-Embedding Method](#) – Goldberg and Levy 2014

From the first of these papers ('Efficient estimation...') we get a description of the *Continuous Bag-of-Words* and *Continuous Skip-gram* models for learning word vectors (we'll talk about what a word vector is in a moment...). From the second paper we get more illustrations of the power of word vectors, some additional information on optimisations for the skip-gram model (hierarchical softmax and negative sampling), and a discussion of *analogies* and *metaphors* to illustrate the third paper's findings.



...und die Liste liesse sich fortsetzen!

Brandon Amos About Blog

Image Completion with Deep Learning in TensorFlow

August 9, 2016



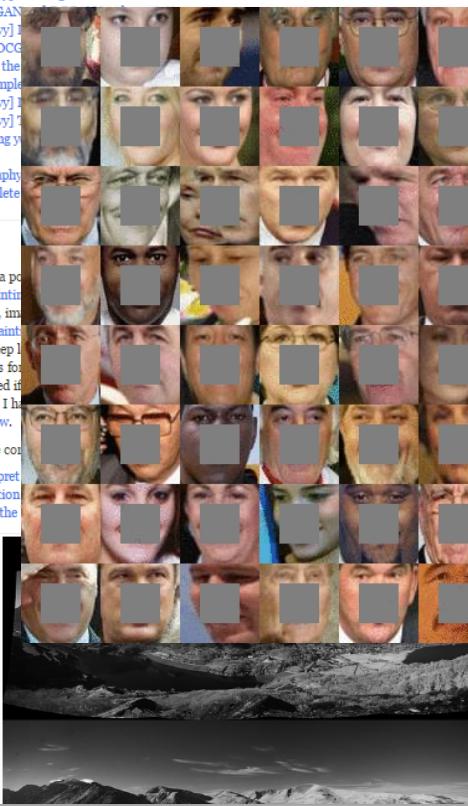
- Introduction
- Step 1: Interpreting images as samples from a probability distribution
 - How would you fill in the missing information?
 - But where does statistics fit in? These are images.
 - So how can we complete images?
- Step 2: Quickly generating fake images
 - Learning to generate new samples from an unknown probability distribution
 - [ML-Heavy] Generative Adversarial Net (GAN) building blocks
 - Using $G(z)$ to produce fake images
 - [ML-Heavy] Training DCGANs
 - Existing GANs
 - [ML-Heavy] DCGANs
 - Running DCGANs
- Step 3: Finding the right samples
 - Image completion
 - [ML-Heavy] 1
 - [ML-Heavy] 2
 - Completing your images
- Conclusion
- Partial bibliography
- Bonus: Incomplete

Introduction

Content-aware fill is a powerful technique for image completion and inpainting. It can be used to content-aware fill, images. "Semantic Image Inpainting" shows how to use deep learning to fill in some deeper portions of images. This section can be skipped if you're not interested in learning about image completion in TensorFlow.

We'll approach image completion in three steps:

1. We'll first interpret the image.
2. This interpretation will help us find the missing parts.
3. Then we'll find the missing parts.



Andrij Karpathy blog About Hacker's guide to Neural Networks

The Unreasonable Effectiveness of Recurrent Neural Networks

August 21, 2015



TECH Nvidia AI Generates Fake Faces Based On Real Celebs

BY STEPHANIE MLOT 10.21.2017 :: 10:00AM EST

32 SHARES



I'm getting a distinctly mid-90s "The Rachel" vibe from the woman in the top left corner (via Nvidia)

STAY ON TARGET

AI Shelley Pens Truly Creepy Horror Stories—And You Can Help

Neural Network Serves Up Truly Frightening Halloween Costume Ideas

Celebrity scandals are about to get a lot more complicated.

Nvidia has developed a way of producing photo-quality, AI-generated human profiles—by using famous faces.

the morning paper

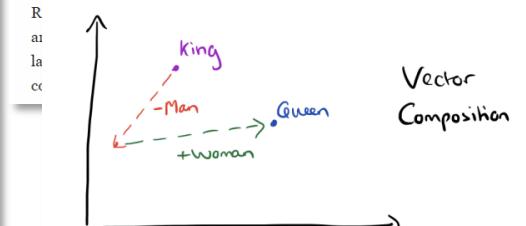
The amazing power of word vectors

APRIL 21, 2016

For today's post, I've drawn material not just from one paper, but from five! The subject matter is 'word2vec' – the work of Mikolov et al. at Google on efficient vector representations of words (and what you can do with them). The papers are:

- ★ Efficient Estimation of Word Representations in Vector Space – Mikolov et al. 2013
- ★ Distributed Representations of Words and Phrases and their Compositionality – Mikolov et al. 2013
- ★ Linguistic Regularities in Continuous Space Word Representations – Mikolov et al. 2013
- ★ word2vec Parameter Learning Explained – Rong 2014
- ★ word2vec Explained: Deriving Mikolov et al.'s Negative Sampling Word-Embedding Method – Goldberg and Levy 2014

From the first of these papers ('Efficient estimation...') we get a description of the *Continuous Bag-of-Words* and *Continuous Skip-gram* models for learning word vectors (we'll talk about what a word vector is in a moment...). From the second paper we get more illustrations of the power of word vectors, some additional information on optimisations for the skip-gram model (hierarchical softmax and negative sampling), and a discussion of *n-grams* and *softmax* to measure the third moment of distributions.

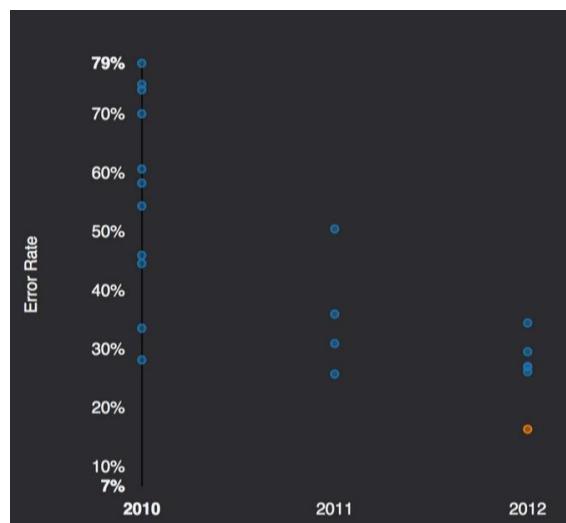


Was ist passiert?

Der ImageNet Wettbewerb



1000 Kategorien
1 Mio. Beispiele

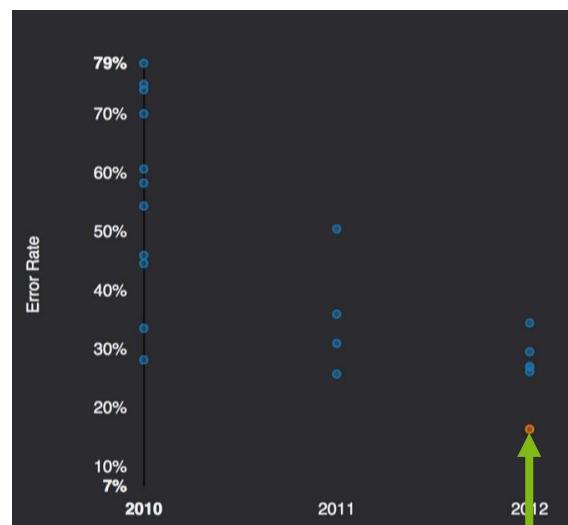


Was ist passiert?

Der ImageNet Wettbewerb



1000 Kategorien
1 Mio. Beispiele



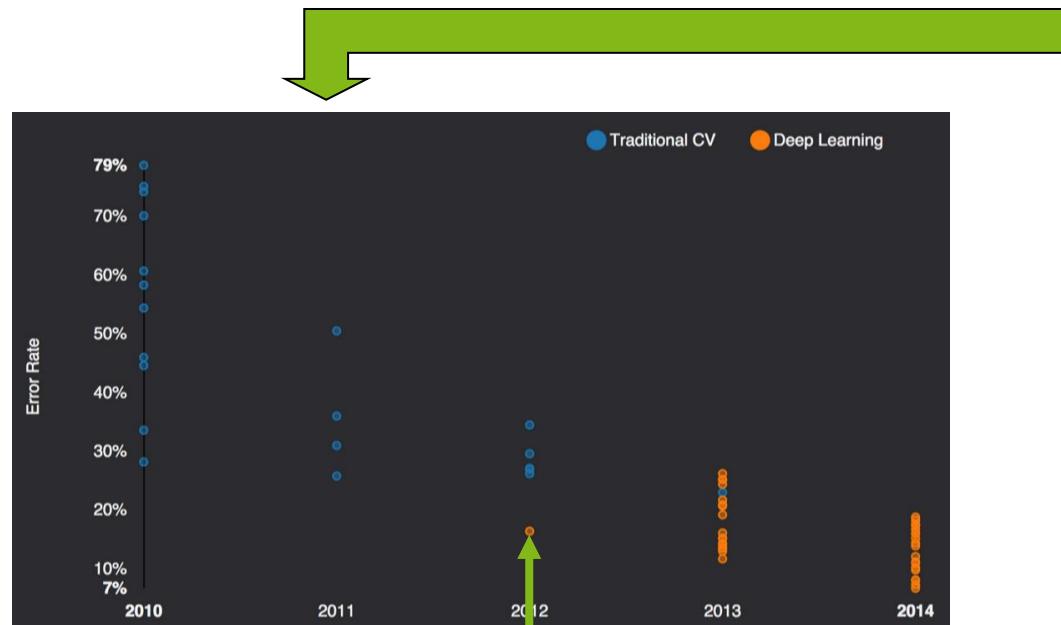
A. Krizhevsky verwendet als erster ein
sog. «Deep Neural Network» (CNN)

Was ist passiert?

Der ImageNet Wettbewerb



1000 Kategorien
1 Mio. Beispiele



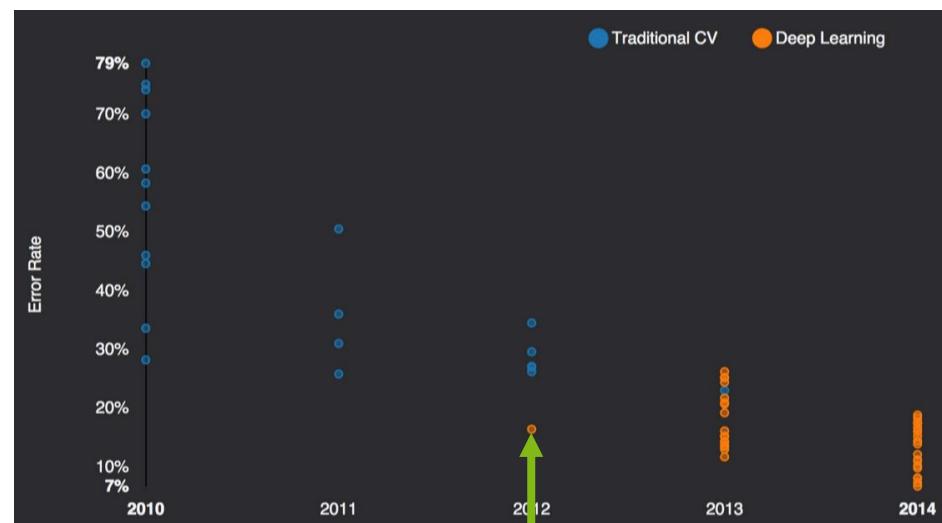
A. Krizhevsky verwendet als erster ein
sog. «Deep Neural Network» (CNN)

Was ist passiert?

Der ImageNet Wettbewerb



1000 Kategorien
1 Mio. Beispiele



A. Krizhevsky verwendet als erster ein
sog. «Deep Neural Network» (CNN)

2015: Computer haben "Sehen" gelernt

4.95% Microsoft (06. Februar)
→ Besser als Menschen (5.10%)

4.80% Google (11. Februar)

4.58% Baidu (11. Mai)

3.57% Microsoft (10. Dezember)

Was? → Wie? → Wo?



2

Wie geht das?

Grundlage

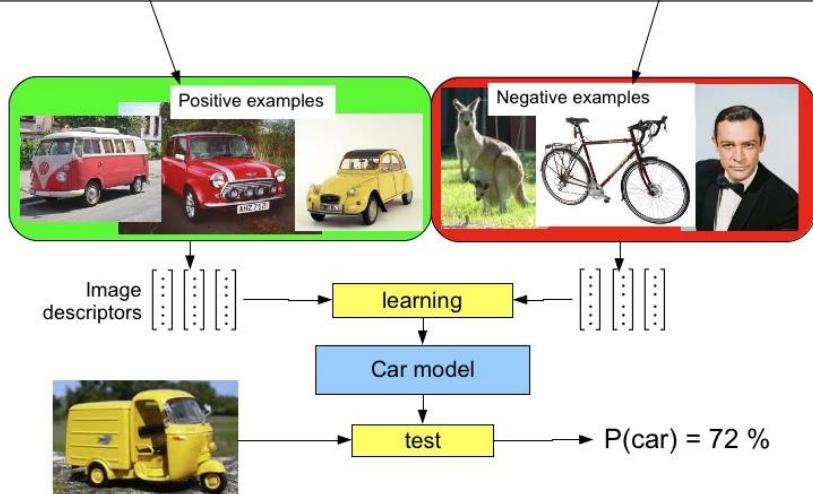
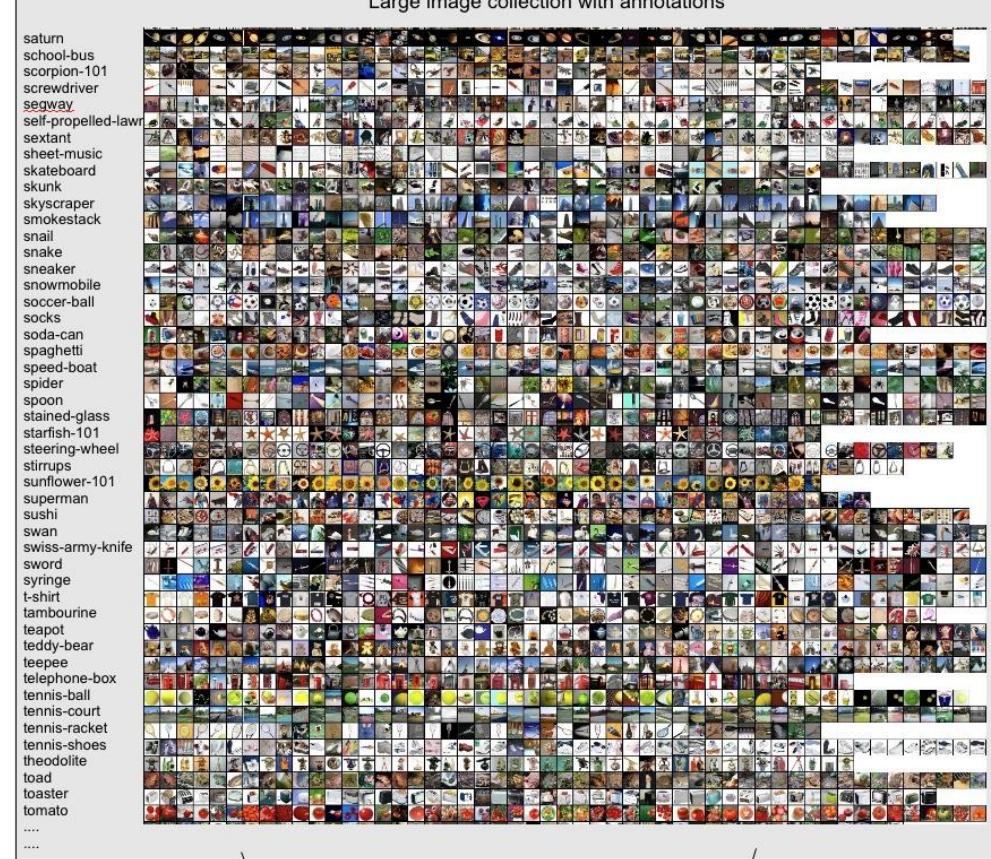
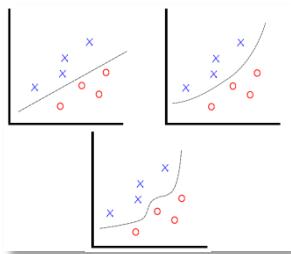
Induktives überwachtes Lernen

Annahme

- Ein an *genügend viele* Beispiele angepasstes Modell...
- ...wird auch auf unbekannte Daten **generalisieren**

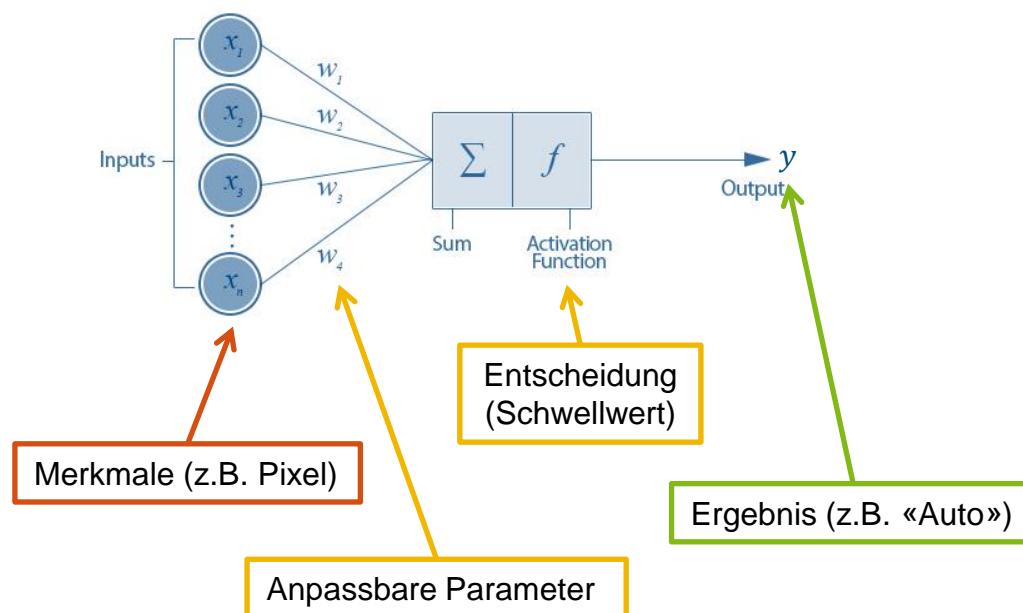
Methode

- **Suchen der Parameter einer gegebenen Funktion...**
- ...so dass für alle Beispiele Eingabe (Bild) auf Ausgabe («Auto») abgebildet wird

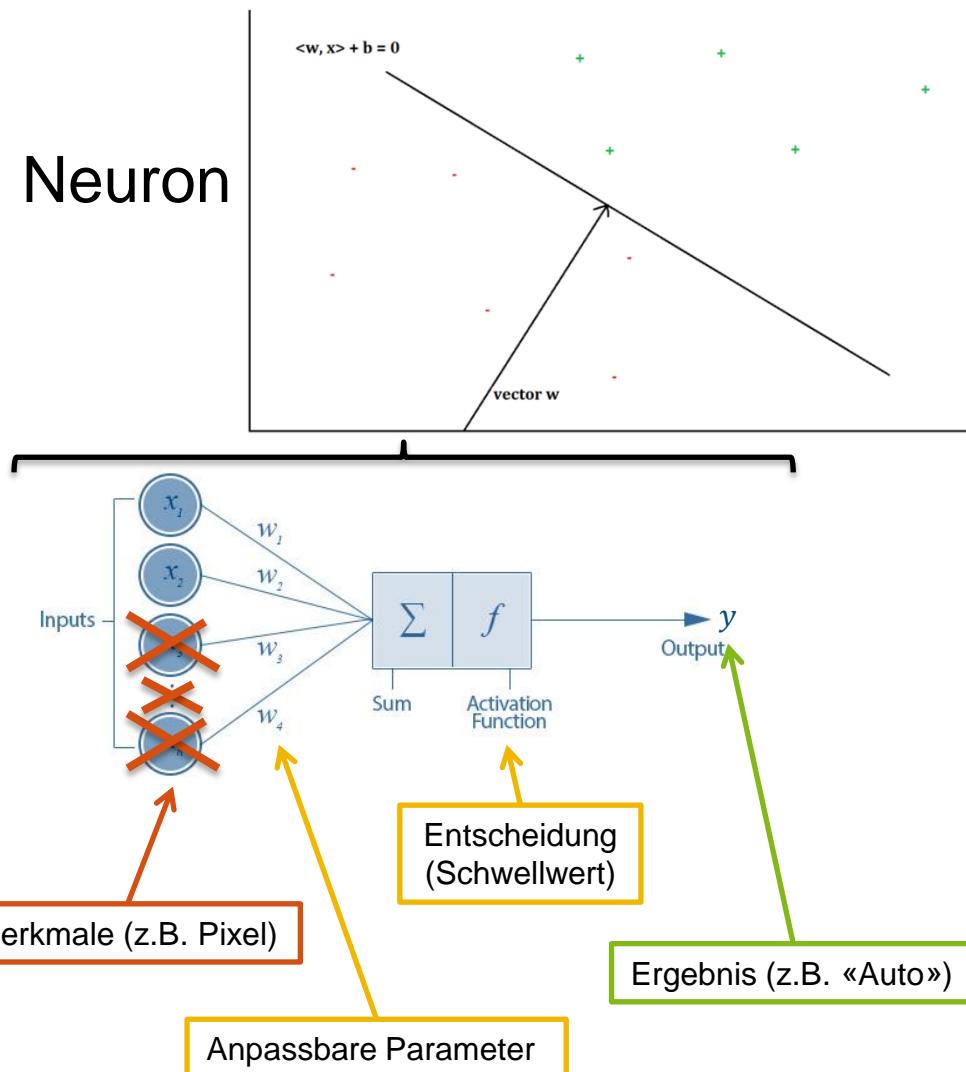


Suche der Parameter einer Funktion?

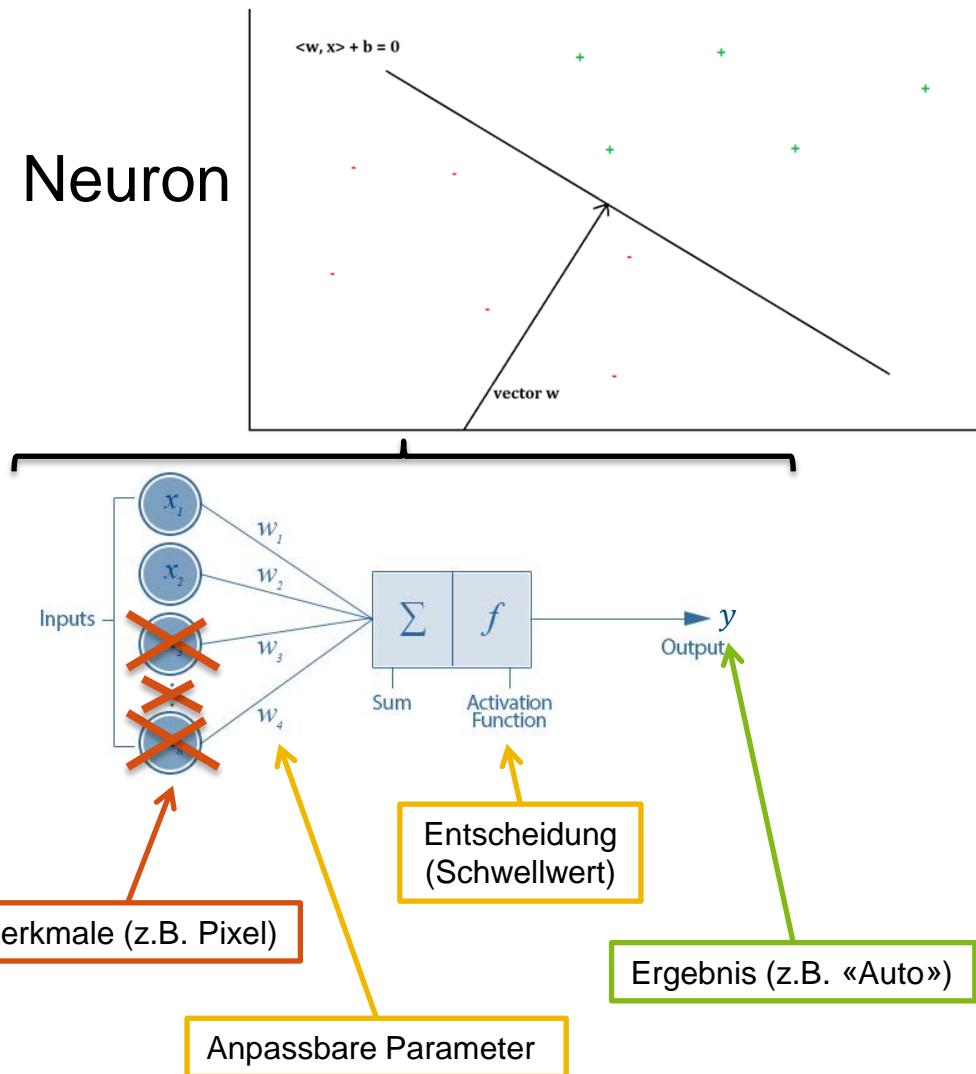
Neuron



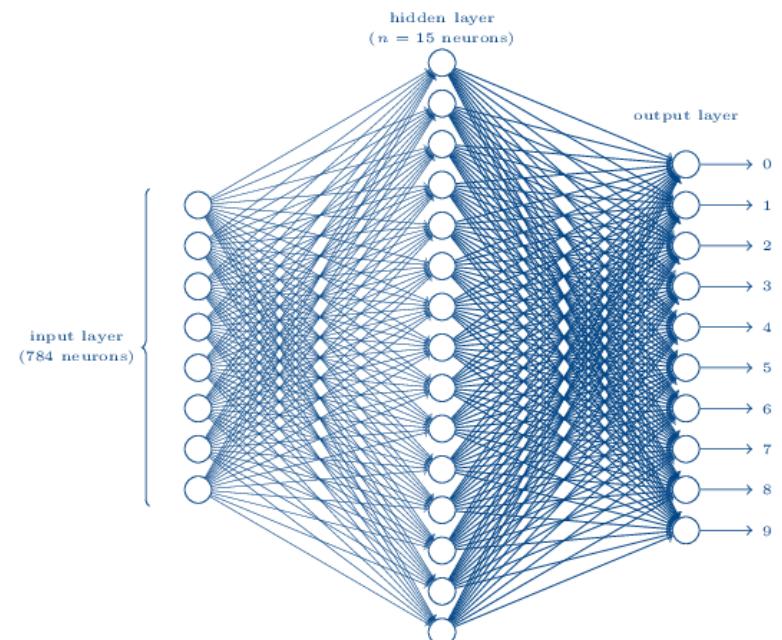
Suche der Parameter einer Funktion?



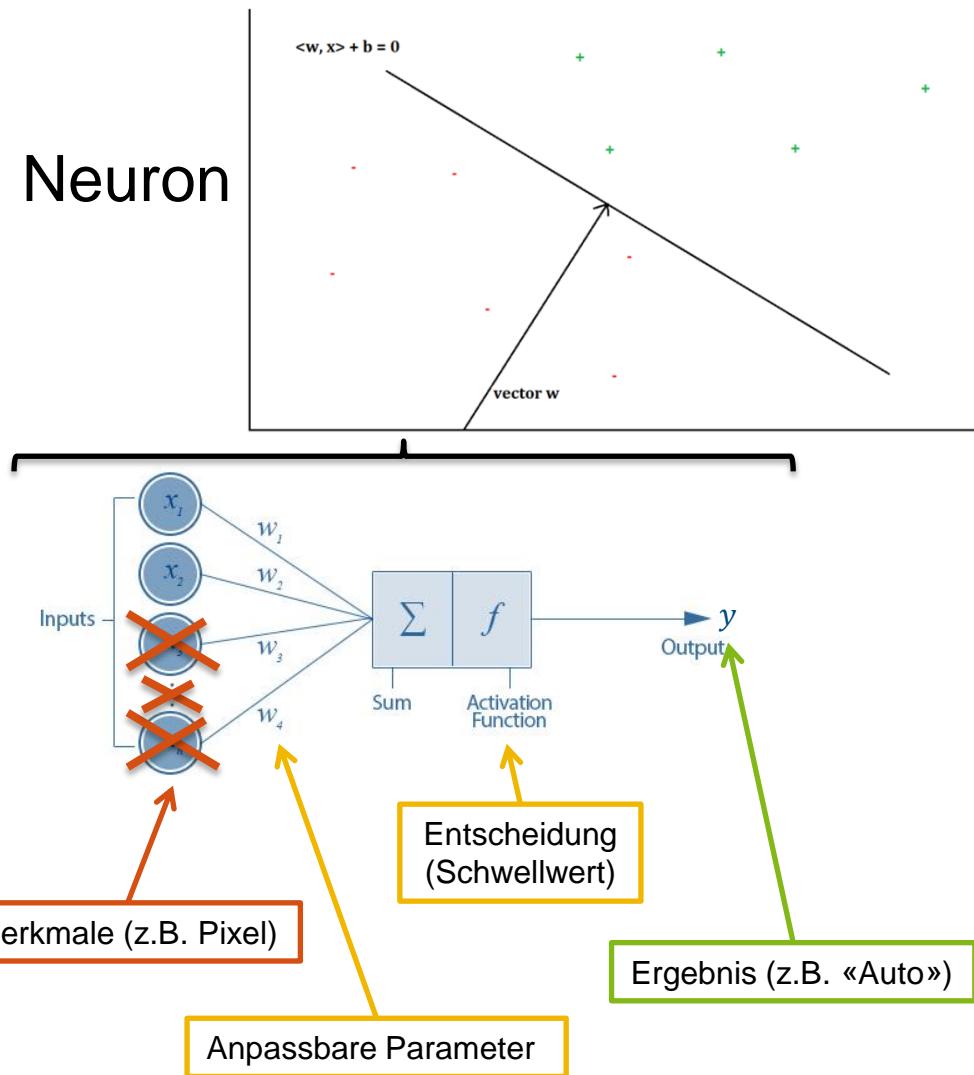
Suche der Parameter einer Funktion?



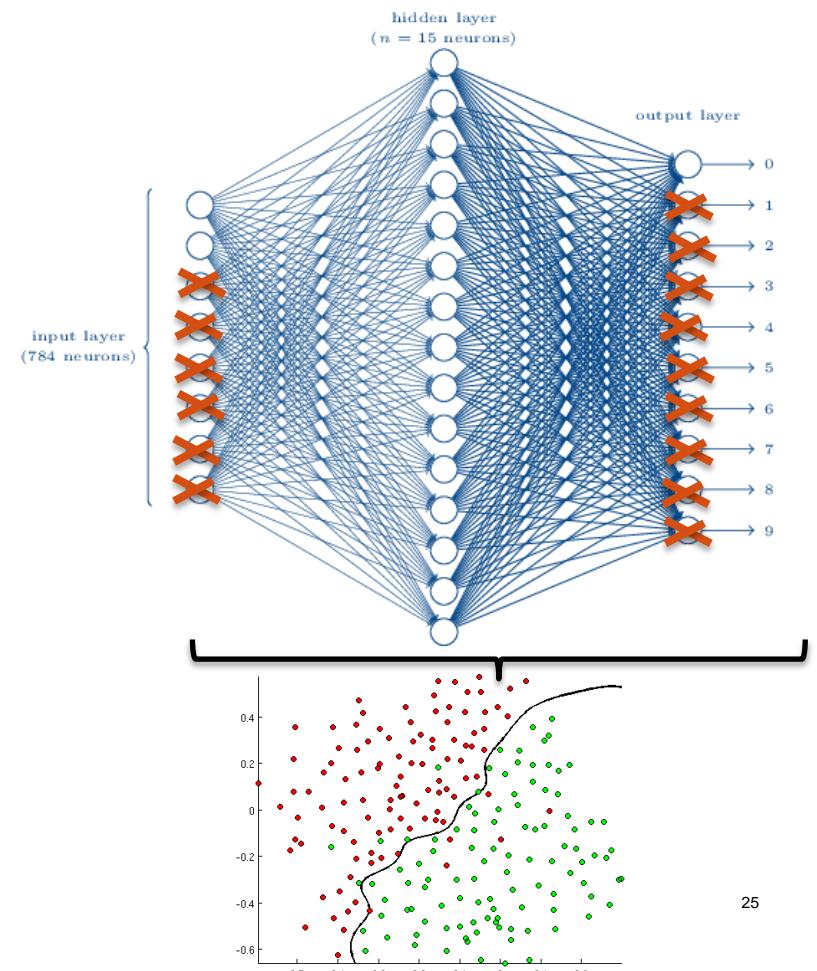
Neuronales Netz



Suche der Parameter einer Funktion?

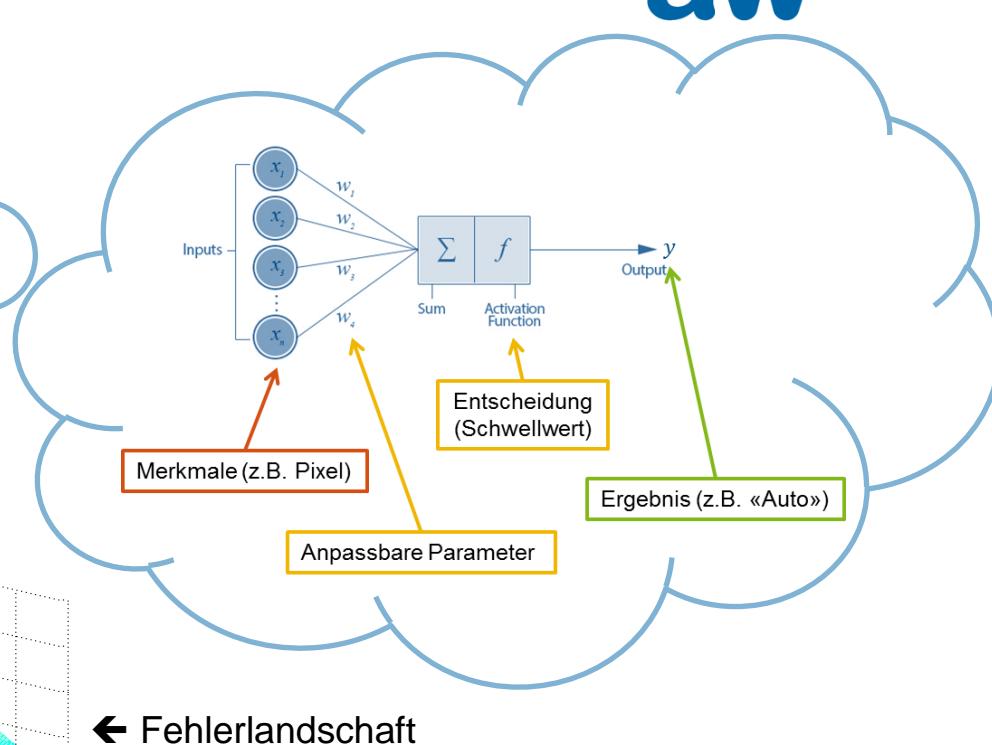
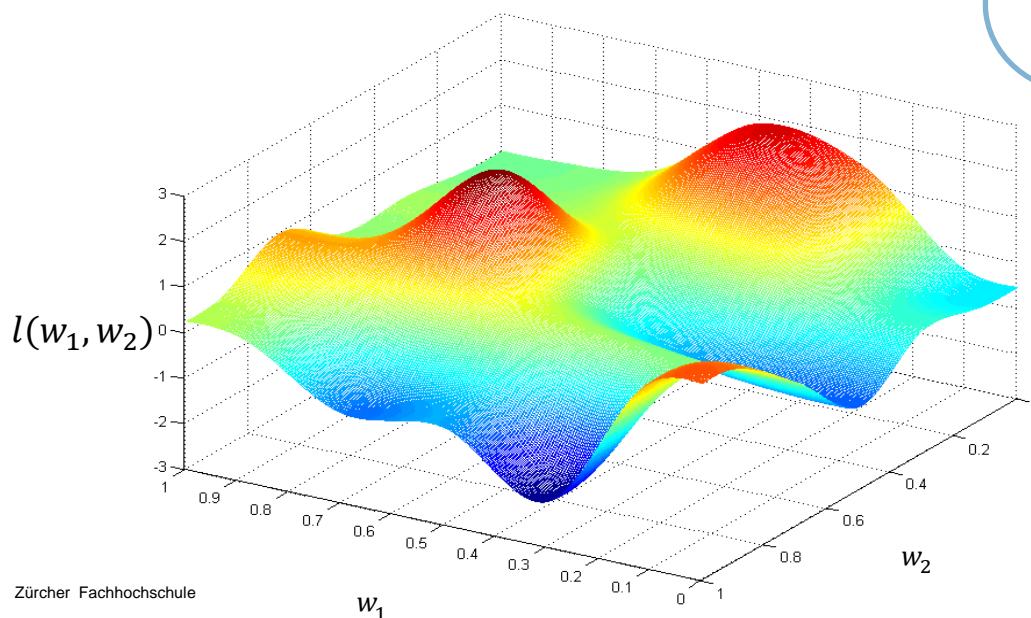


Neuronales Netz



Suche der Parameter einer Funktion?

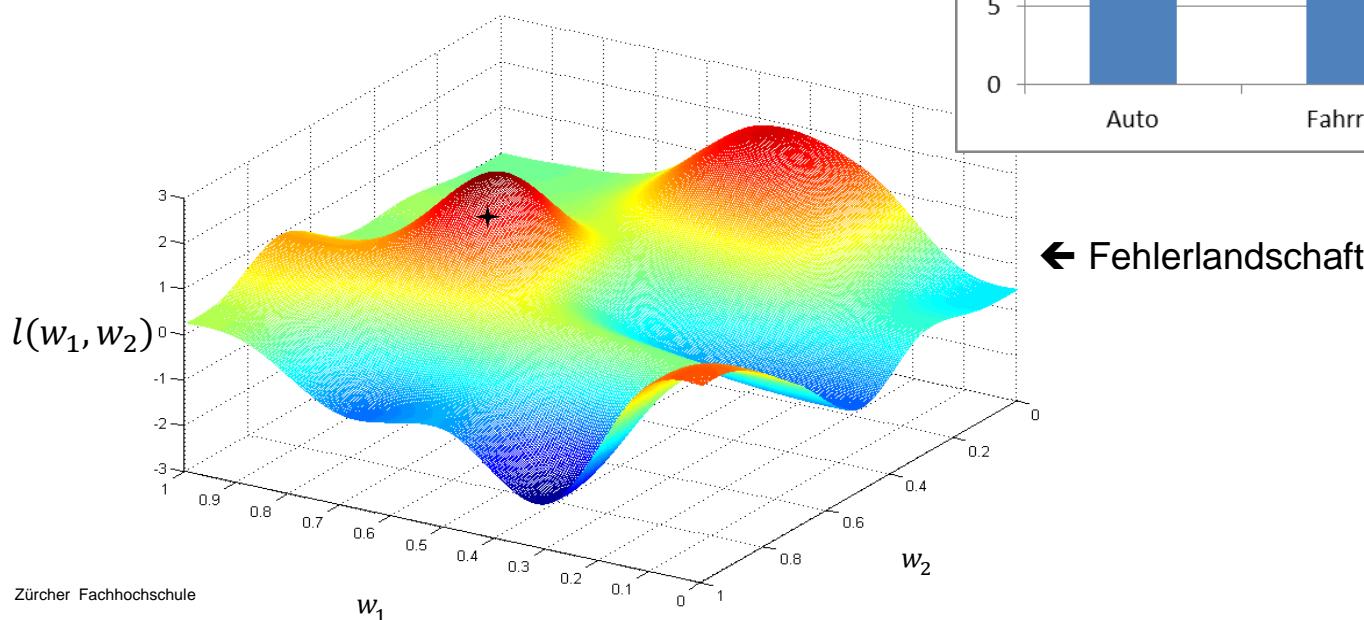
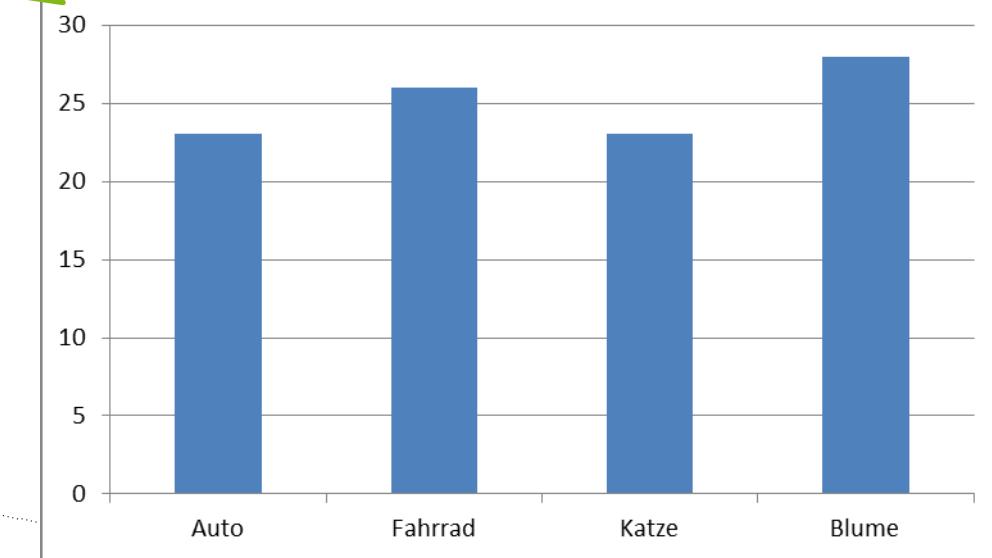
- Unser Neuronales Netz: $f_{\mathbf{W}}(\mathbf{x}) = \mathbf{y}$
mit Bild \mathbf{x} , echtem Resultat \mathbf{y} und Parametern \mathbf{W}
($\mathbf{W} = \{w_1, w_2\}$ anfangs zufällig gewählt)
- Fehlermass: $l(\mathbf{W}) = \frac{1}{N} \sum_{i=1}^N (f_{\mathbf{W}}(\mathbf{x}_i) - \mathbf{y}_i)^2$
Durchschnitt der quadratischen Abweichungen
über alle Bilder (Loss)



Suche der Parameter einer Funktion?

Wahrscheinlichkeit [%] für bestimmtes Ergebnis

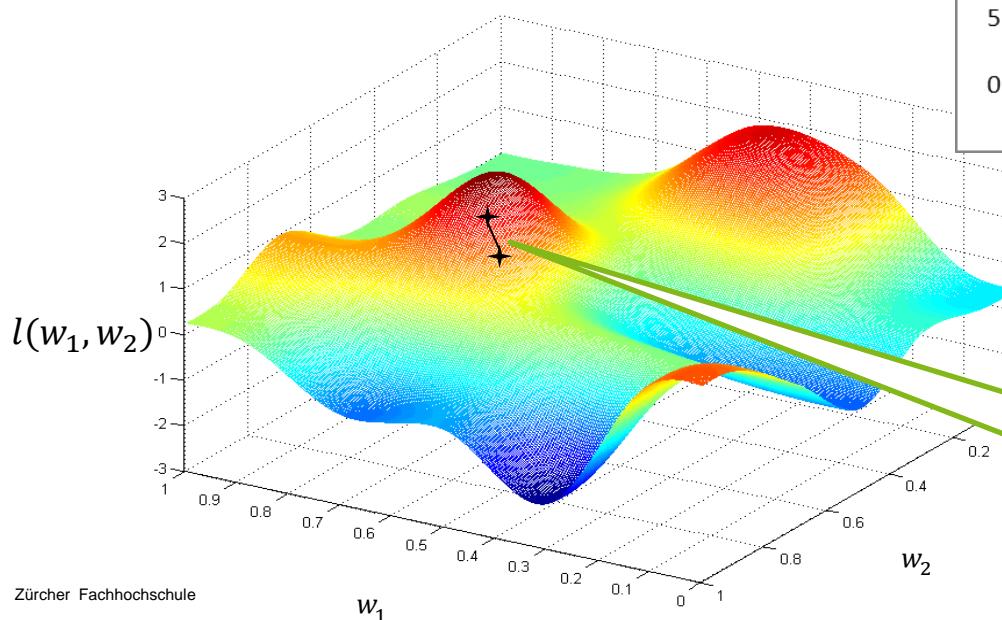
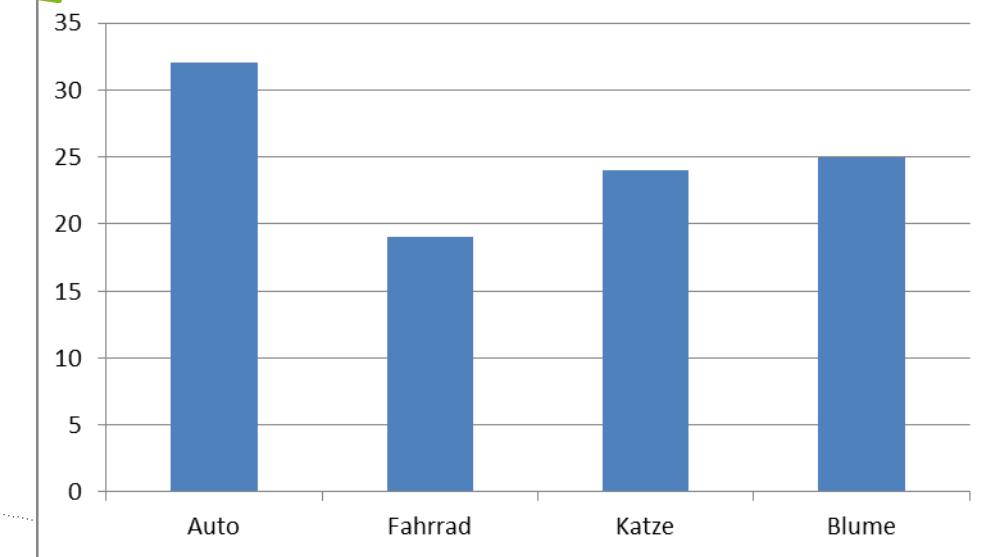
- Unser Neuronales Netz: $f_{\mathbf{W}}(\mathbf{x}) = \mathbf{y}$
mit Bild \mathbf{x} , echtem Resultat \mathbf{y} und Parametern \mathbf{W}
($\mathbf{W} = \{w_1, w_2\}$ anfangs zufällig gewählt)
- Fehlermass: $l(\mathbf{W}) = \frac{1}{N} \sum_{i=1}^N (f_{\mathbf{W}}(\mathbf{x}_i) - \mathbf{y}_i)^2$
Durchschnitt der quadratischen Abweichungen
über alle Bilder (Loss)



Suche der Parameter einer Funktion?

Wahrscheinlichkeit [%] für bestimmtes Ergebnis

- Unser Neuronales Netz: $f_{\mathbf{W}}(\mathbf{x}) = \mathbf{y}$
mit Bild \mathbf{x} , echtem Resultat \mathbf{y} und Parametern \mathbf{W}
($\mathbf{W} = \{w_1, w_2\}$ anfangs zufällig gewählt)
- Fehlermass: $l(\mathbf{W}) = \frac{1}{N} \sum_{i=1}^N (f_{\mathbf{W}}(\mathbf{x}_i) - \mathbf{y}_i)^2$
Durchschnitt der quadratischen Abweichungen
über alle Bilder (Loss)



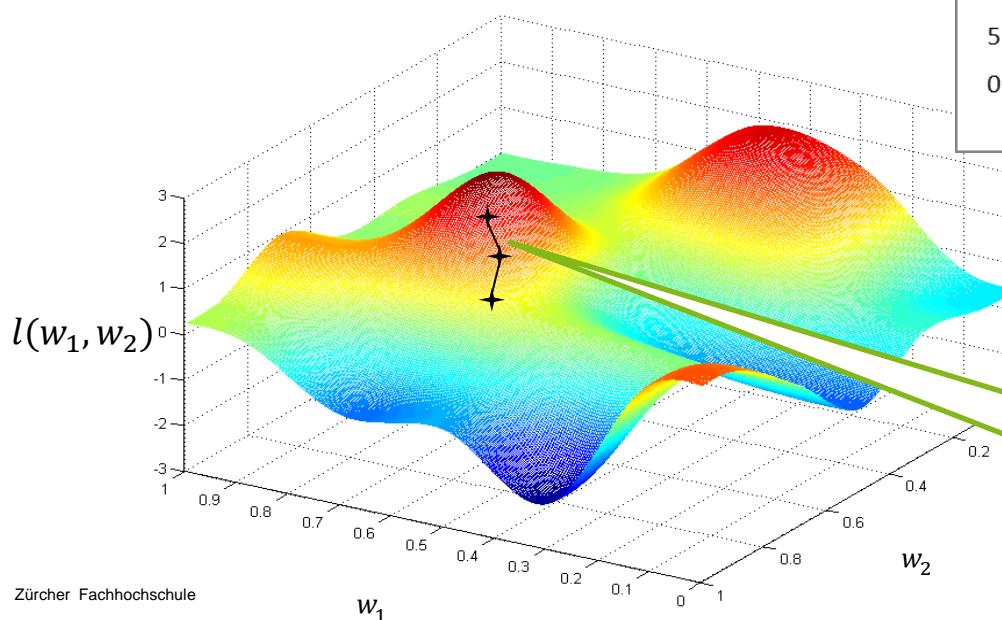
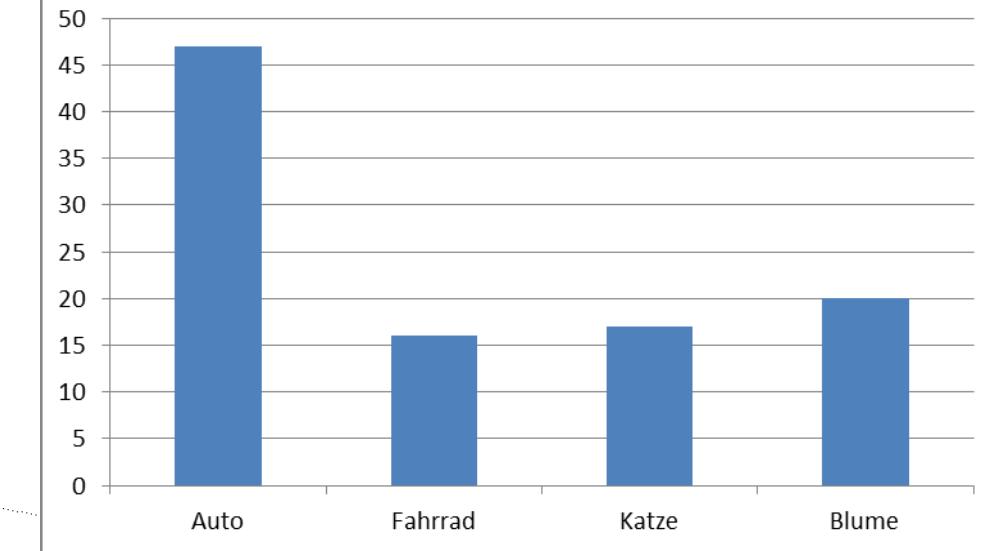
← Fehlerlandschaft

Methode: Anpassung der Gewichte
von f in Richtung der steilsten
Steigung (abwärts) von J

Suche der Parameter einer Funktion?

Wahrscheinlichkeit [%] für bestimmtes Ergebnis

- Unser Neuronales Netz: $f_{\mathbf{W}}(\mathbf{x}) = \mathbf{y}$
mit Bild \mathbf{x} , echtem Resultat \mathbf{y} und Parametern \mathbf{W}
($\mathbf{W} = \{w_1, w_2\}$ anfangs zufällig gewählt)
- Fehlermass: $l(\mathbf{W}) = \frac{1}{N} \sum_{i=1}^N (f_{\mathbf{W}}(\mathbf{x}_i) - \mathbf{y}_i)^2$
Durchschnitt der quadratischen Abweichungen
über alle Bilder (Loss)



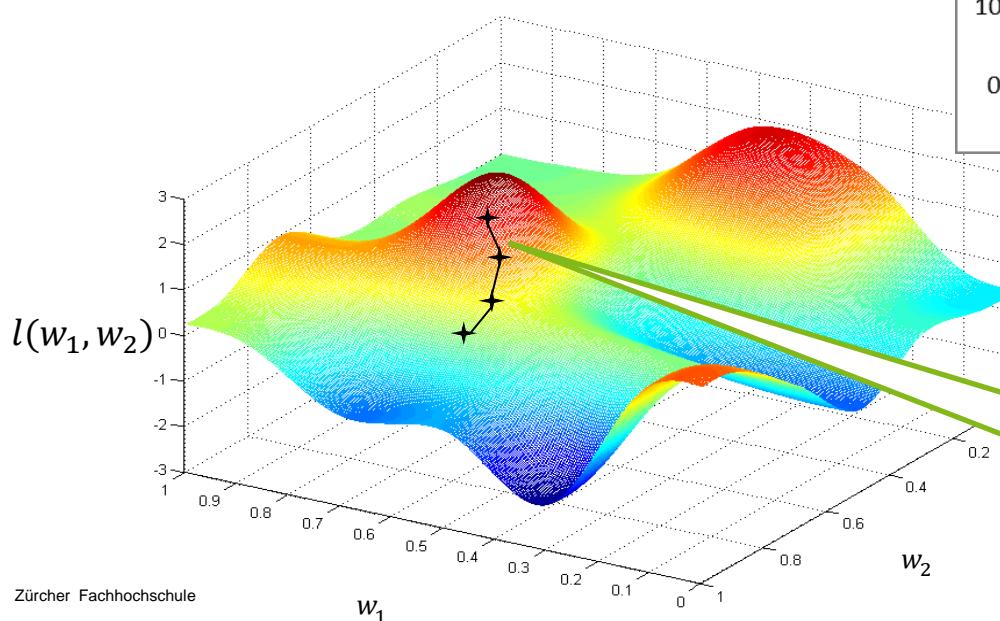
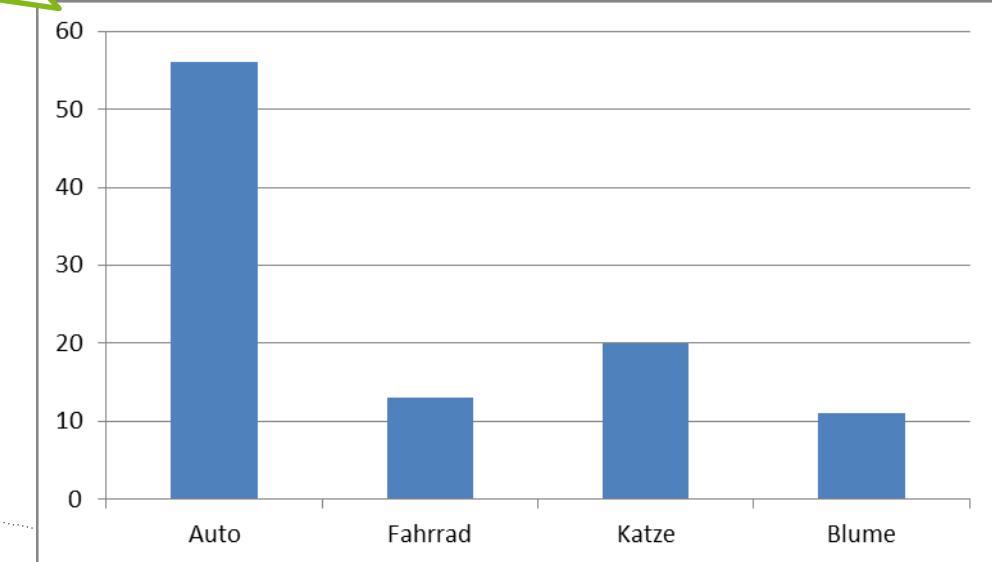
← Fehlerlandschaft

Methode: Anpassung der Gewichte
von f in Richtung der steilsten
Steigung (abwärts) von J

Suche der Parameter einer Funktion?

Wahrscheinlichkeit [%] für bestimmtes Ergebnis

- Unser Neuronales Netz: $f_{\mathbf{W}}(\mathbf{x}) = \mathbf{y}$
mit Bild \mathbf{x} , echtem Resultat \mathbf{y} und Parametern \mathbf{W}
($\mathbf{W} = \{w_1, w_2\}$ anfangs zufällig gewählt)
- Fehlermass: $l(\mathbf{W}) = \frac{1}{N} \sum_{i=1}^N (f_{\mathbf{W}}(\mathbf{x}_i) - \mathbf{y}_i)^2$
Durchschnitt der quadratischen Abweichungen
über alle Bilder (Loss)



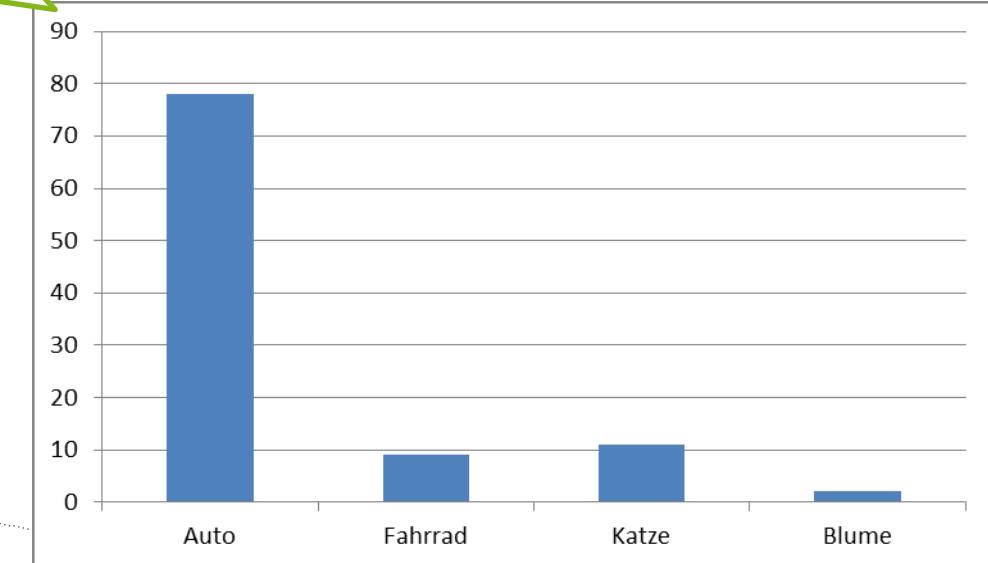
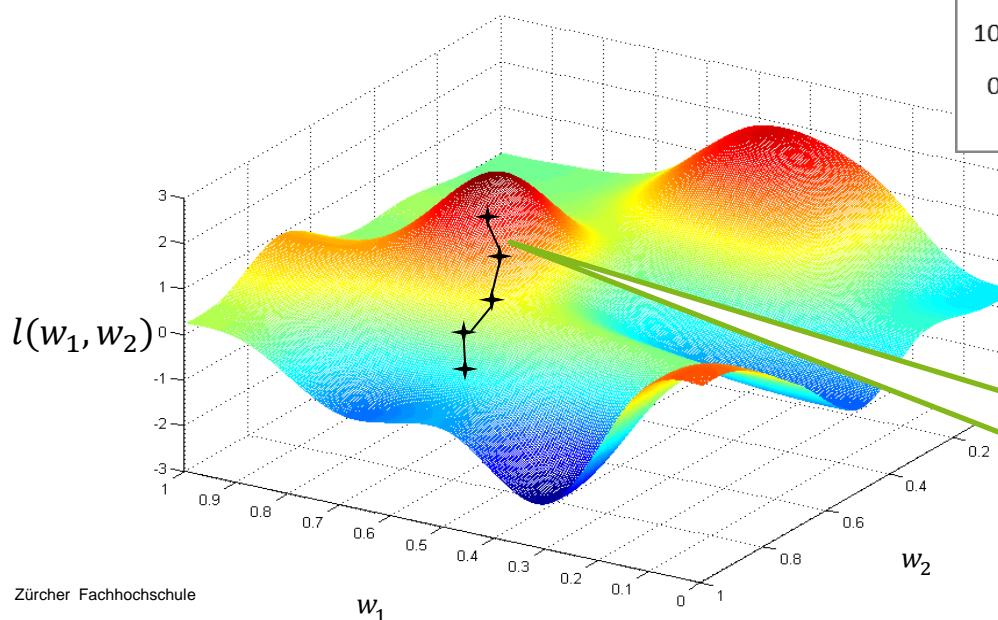
← Fehlerlandschaft

Methode: Anpassung der Gewichte
von f in Richtung der steilsten
Steigung (abwärts) von J

Suche der Parameter einer Funktion?

Wahrscheinlichkeit [%] für bestimmtes Ergebnis

- Unser Neuronales Netz: $f_{\mathbf{W}}(\mathbf{x}) = \mathbf{y}$
mit Bild \mathbf{x} , echtem Resultat \mathbf{y} und Parametern \mathbf{W}
($\mathbf{W} = \{w_1, w_2\}$ anfangs zufällig gewählt)
- Fehlermass: $l(\mathbf{W}) = \frac{1}{N} \sum_{i=1}^N (f_{\mathbf{W}}(\mathbf{x}_i) - \mathbf{y}_i)^2$
Durchschnitt der quadratischen Abweichungen
über alle Bilder (Loss)



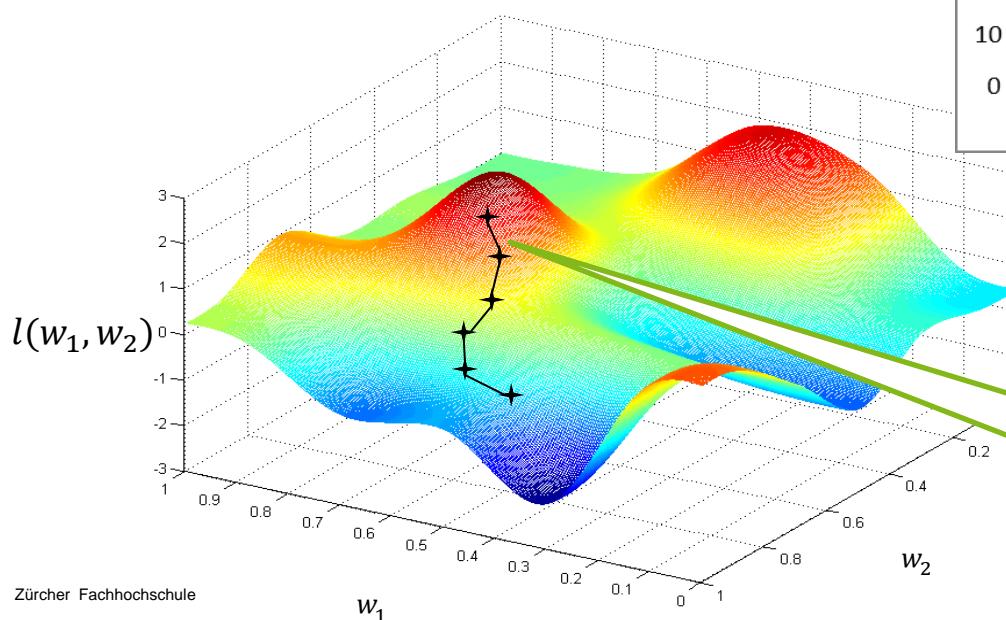
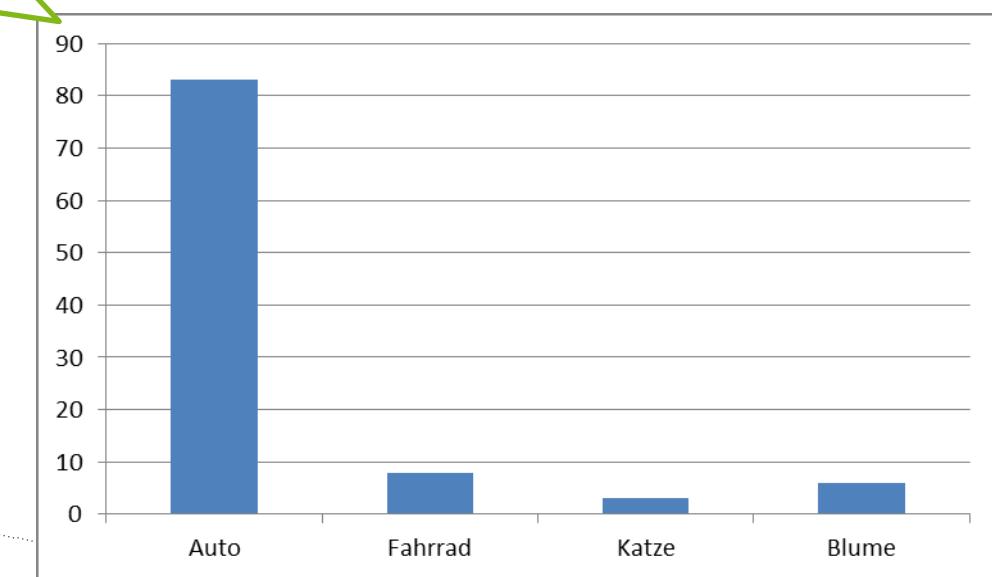
← Fehlerlandschaft

Methode: Anpassung der Gewichte
von f in Richtung der steilsten
Steigung (abwärts) von J

Suche der Parameter einer Funktion?

Wahrscheinlichkeit [%] für bestimmtes Ergebnis

- Unser Neuronales Netz: $f_{\mathbf{W}}(\mathbf{x}) = \mathbf{y}$
mit Bild \mathbf{x} , echtem Resultat \mathbf{y} und Parametern \mathbf{W}
($\mathbf{W} = \{w_1, w_2\}$ anfangs zufällig gewählt)
- Fehlermass: $l(\mathbf{W}) = \frac{1}{N} \sum_{i=1}^N (f_{\mathbf{W}}(\mathbf{x}_i) - \mathbf{y}_i)^2$
Durchschnitt der quadratischen Abweichungen
über alle Bilder (Loss)



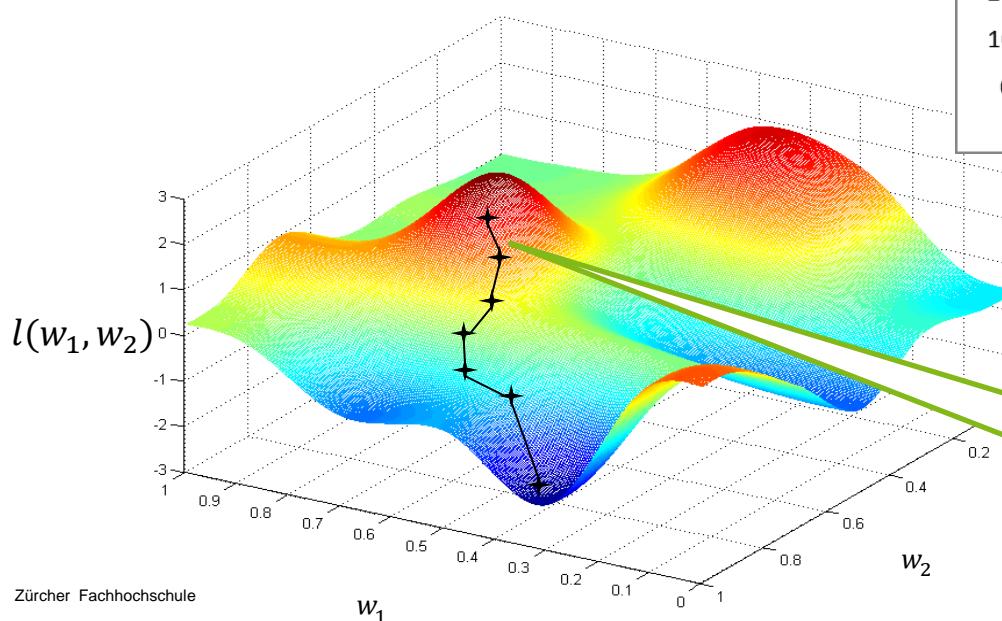
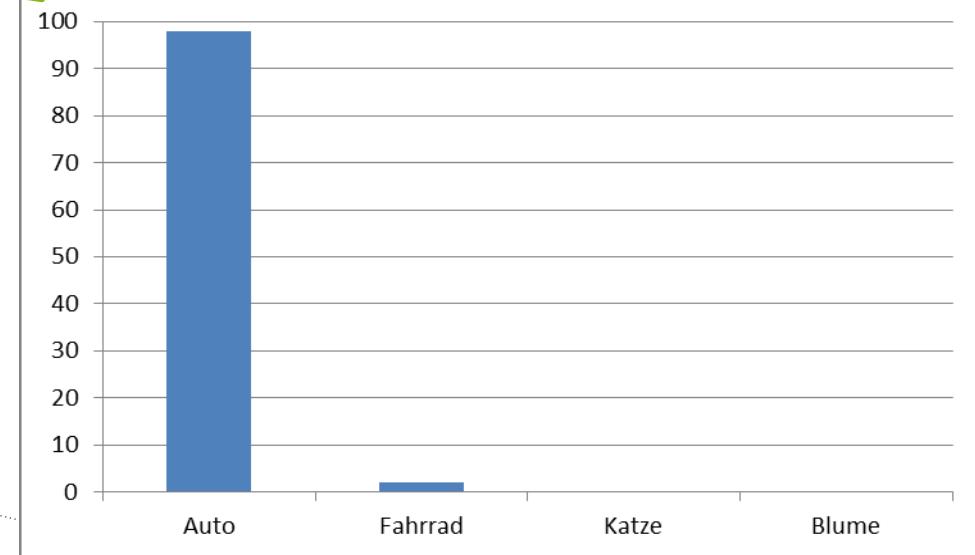
← Fehlerlandschaft

Methode: Anpassung der Gewichte
von f in Richtung der steilsten
Steigung (abwärts) von J

Suche der Parameter einer Funktion?

Wahrscheinlichkeit [%] für bestimmtes Ergebnis

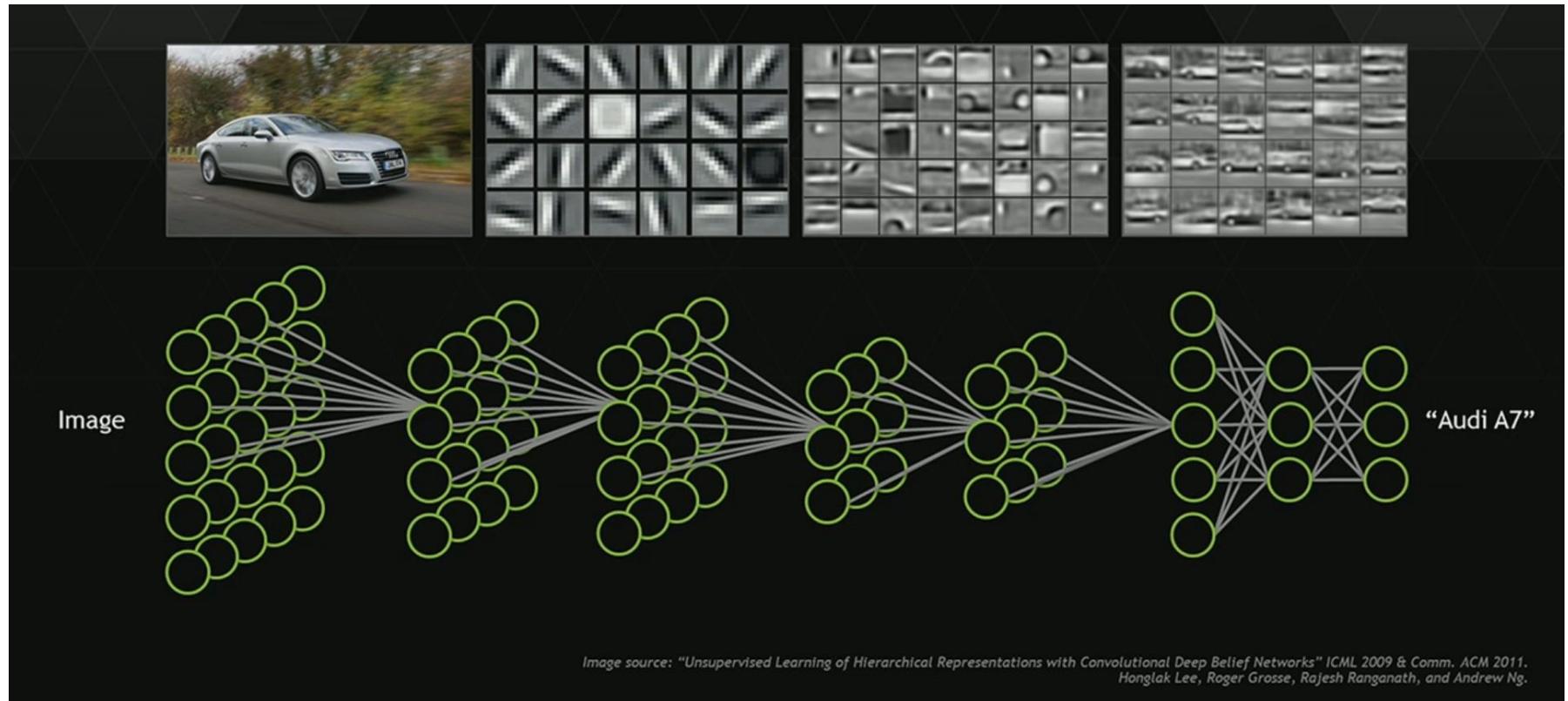
- Unser Neuronales Netz: $f_{\mathbf{W}}(\mathbf{x}) = \mathbf{y}$
mit Bild \mathbf{x} , echtem Resultat \mathbf{y} und Parametern \mathbf{W}
($\mathbf{W} = \{w_1, w_2\}$ anfangs zufällig gewählt)
- Fehlermass: $l(\mathbf{W}) = \frac{1}{N} \sum_{i=1}^N (f_{\mathbf{W}}(\mathbf{x}_i) - \mathbf{y}_i)^2$
Durchschnitt der quadratischen Abweichungen
über alle Bilder (Loss)



← Fehlerlandschaft

Methode: Anpassung der Gewichte
von f in Richtung der steilsten
Steigung (abwärts) von J

Was «sieht» das Neuronale Netz? Hierarchien komplexer werdender Merkmale



Quellen: <https://www.pinterest.com/explore/artificial-neural-network/>

Olah, et al., "Feature Visualization", Distill, 2017, <https://distill.pub/2017/feature-visualization/>.

Was «sieht» das Neuronale Netz? Hierarchien komplexer werdender Merkmale

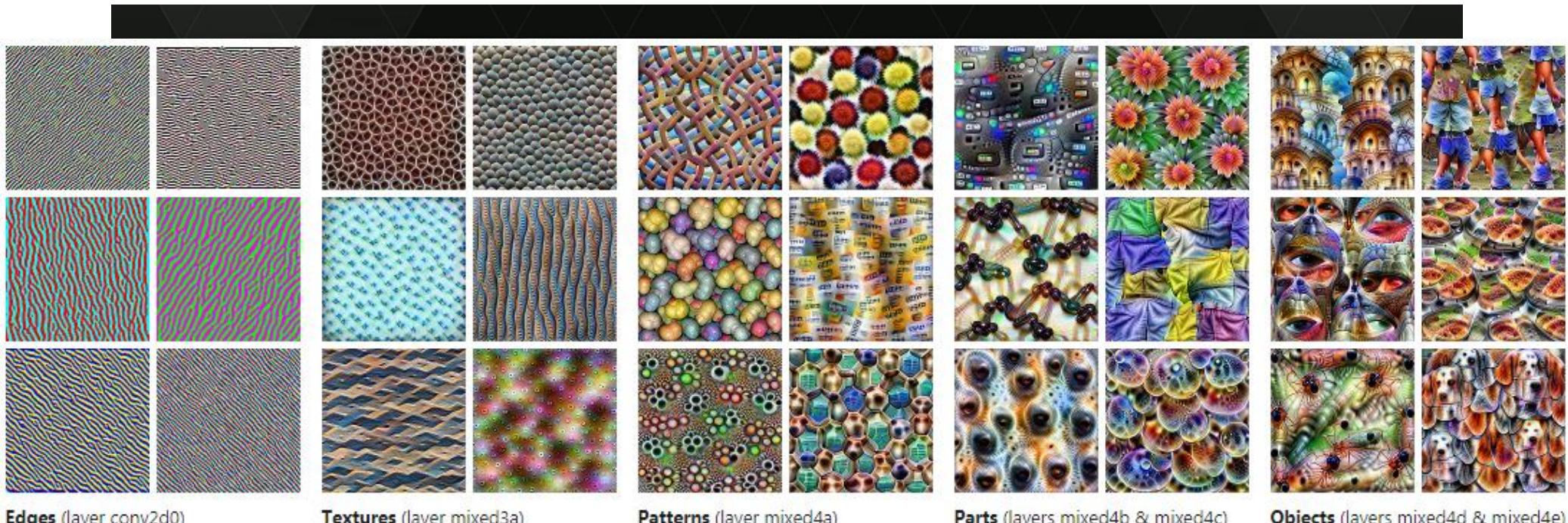


Image source: "Unsupervised Learning of Hierarchical Representations with Convolutional Deep Belief Networks" ICML 2009 & Comm. ACM 2011.
Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Ng.

Quellen: <https://www.pinterest.com/explore/artificial-neural-network/>

Olah, et al., "Feature Visualization", Distill, 2017, <https://distill.pub/2017/feature-visualization/>.

Wie schlussfolgert die Maschine? «Debugging» für Einblicke in die vermeintliche «Black Box»

Verdeutlichen ein Problem:

- Adversarial Examples



<https://blog.openai.com/adversarial-example-research/>

Wie schlussfolgert die Maschine? «Debugging» für Einblicke in die vermeintliche «Black Box»

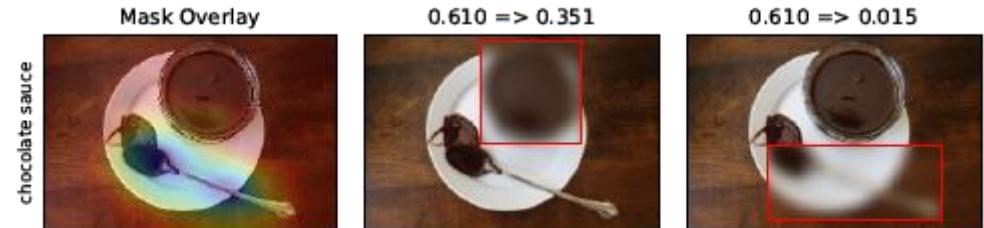
Verdeutlichen ein Problem:

- Adversarial Examples



Bieten eine Lösung:

- Saliency Maps



Ruth C. Fong & Andrea Vedaldi, «Interpretable Explanations of Black Boxes by Meaningful Perturbation», 2017

Was? → Wie? → Wo?

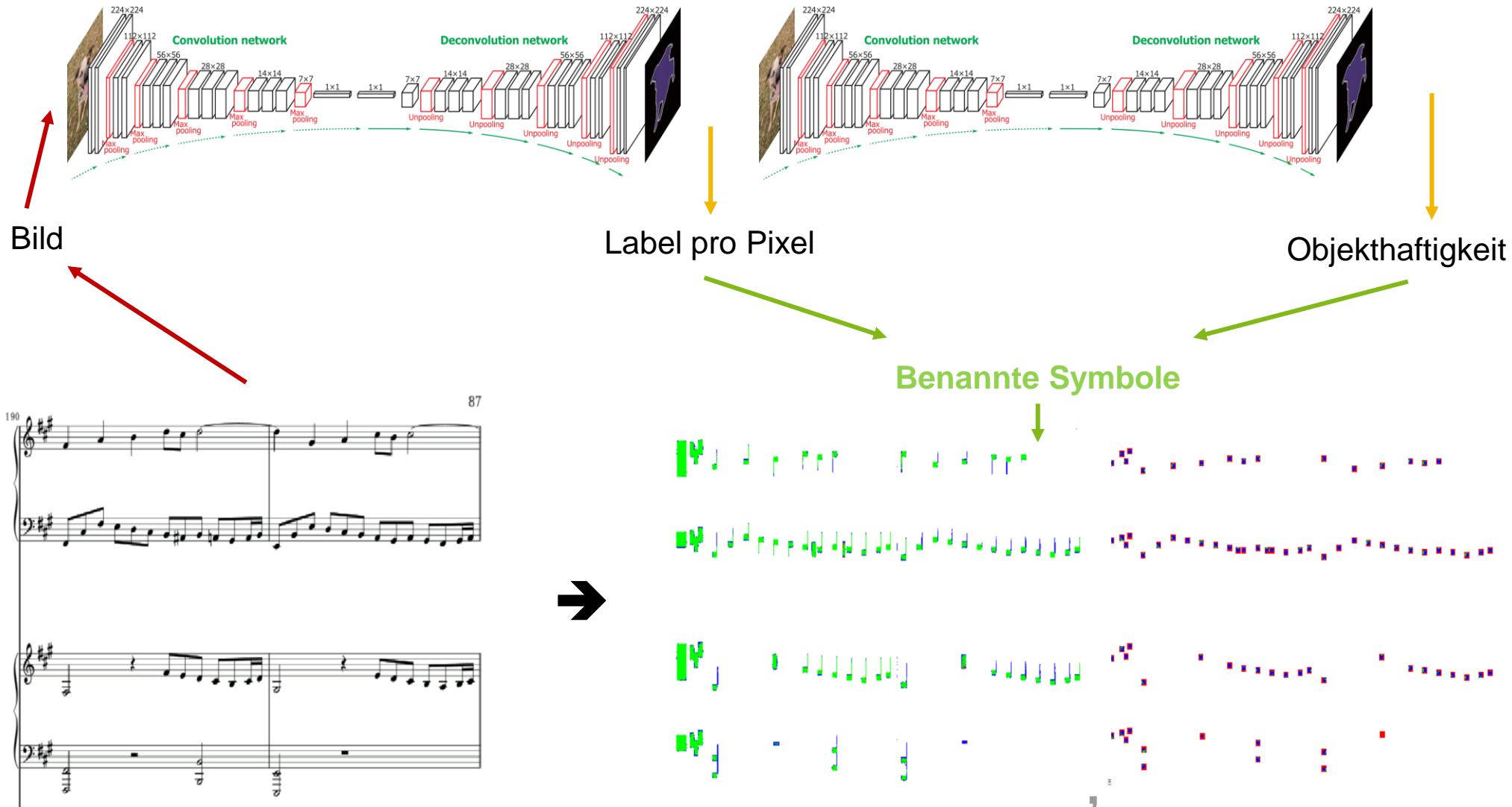


3

Wo wird das heute bereits eingesetzt?

Erkennung von Musiknotation

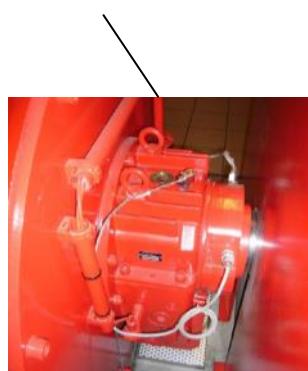
Grundlage für Digitalisierung in Orchestern und Musikschulen



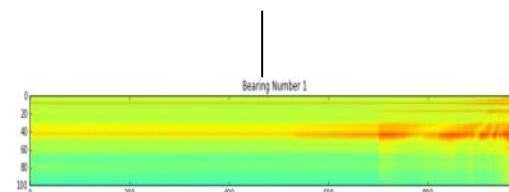
Datengetriebenes Condition Monitoring

Predictive Maintenance von Rotationsmaschinen

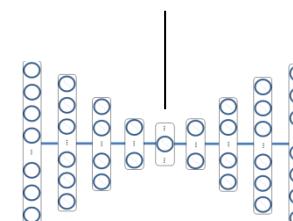
Vibrations-Sensor



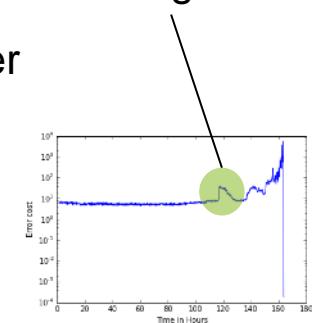
Merkmalsextraktion



z.B. neuronaler Autoencoder

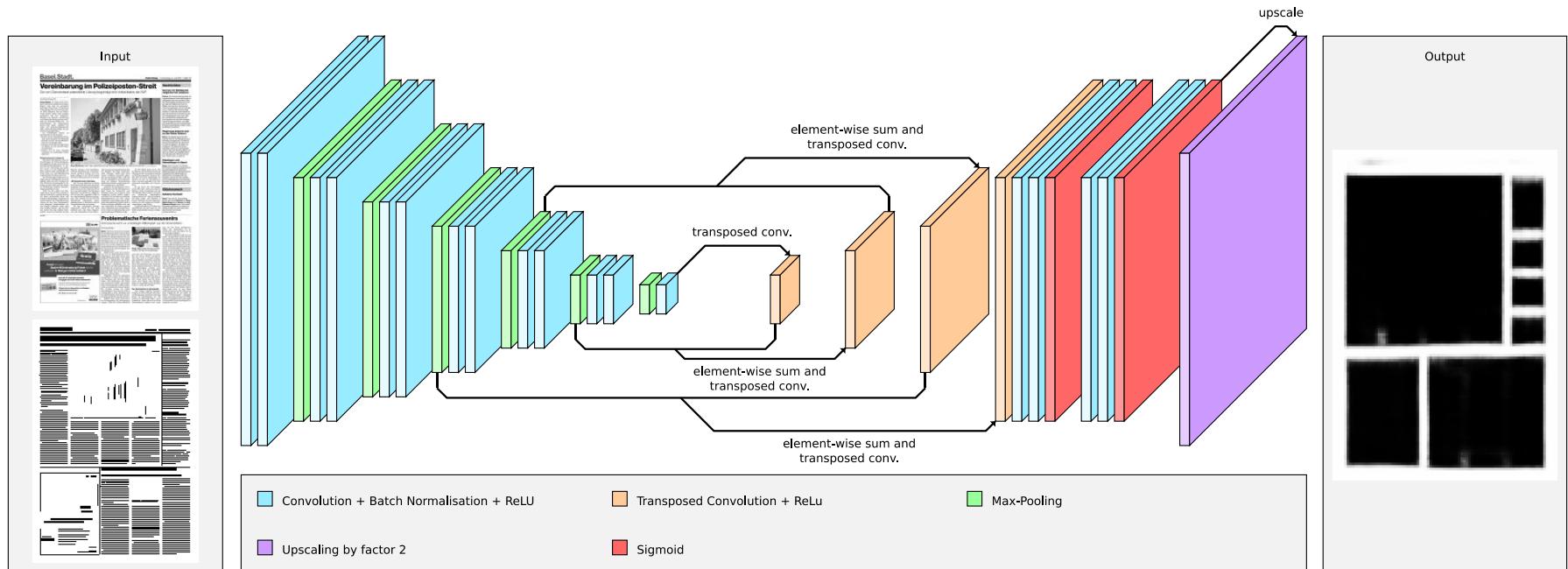


Früherkennung von Fehlern



Segmentierung von Zeitungsartikeln

Semiautomatische Medienbeobachtung



Schlussfolgerungen



- Deep Learning hat zu Paradigmenwechsel in *Mustererkennungsaufgaben* geführt
- Die Zeit vom Grundlagenresultat zur praktischer Anwendung beträgt wenige Monate
- Es gibt Methoden zum Hineinschauen in neuronale Black Boxes
- «Denkende rechnende» Maschinen sind trotzdem nur *insel(-hoch-)begabt*
→ Herausforderungen bestehen im Bereich Robustheit, Interpretierbarkeit, rechtl. Stellung



swiss group for artificial intelligence
and cognitive science



Swiss Alliance for
Data-Intensive Services

Mehr zu mir:

- Leiter ZHAW Datalab, Vice President SGAICO, Board Data+Service
- thilo.stadelmann@zhaw.ch
- 058 934 72 08
- www.zhaw.ch/~stdm



Mehr zum Thema:

- KI: <https://sgaico.swissinformatics.org/>
- Verband Data+Service Science: www.data-service-alliance.ch
- Gemeinsame Projekte: datalab@zhaw.ch

→ Fragen Sie gerne an.



ANHANG

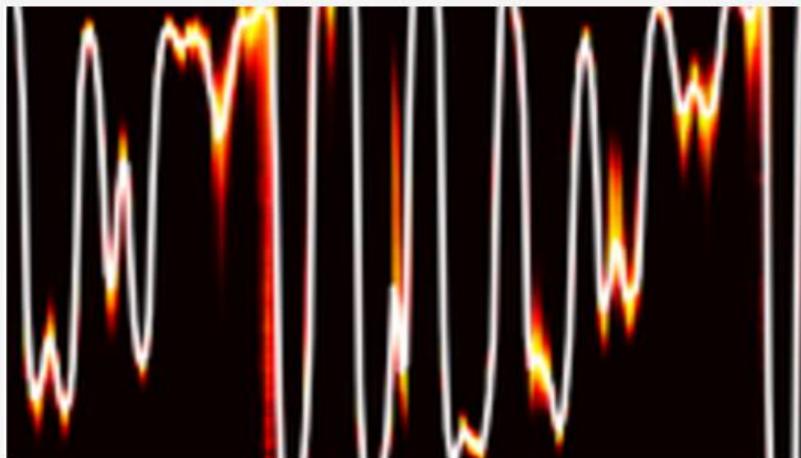
WaveNet lässt Computersprache natürlich klingen

von Henning Steier / 12.9.2016, 10:05 Uhr

Die Google-Tochter DeepMind hat ein neuronales Netz präsentiert, das Rechner fast wie Menschen klingen lässt. Es macht auch Musik.



KOMMENTARE



DeepMind lässt WaveNet Sprachwellen erzeugen. (Symbolbild: PD)

Die Google-Tochter DeepMind machte zuletzt mit ihrem [Sieg beim Spiel «Go» Schlagzeilen](#): Ihre Software AlphaGo schlug im Frühjahr einen der besten menschlichen Spieler, Lee Sedol. Nun hat das Londoner Unternehmen WaveNet präsentiert: Dieses neuronale Netz erzeugt Sprache, die sehr natürlich klingt – zumindest wenn man die im [Blogeintrag](#) des Unternehmens zu hörenden Klangbeispiele als Massstab nimmt. Man hat sogar das Gefühl, Atempausen zu hören.

MEISTGELESEN

Künstliche Intelligenz
Kein Google für jeden
[KOMMENTAR](#) / Henning Steier / 5.10.2016

Neue Produkte aus Mountain View
Google macht sich nicht nur im Wohnzimmer breit
Henning Steier / 4.10.2016

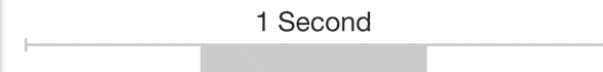
Dropbox
68 Millionen verschlüsselte Passwörter im Netz
5.10.2016



Generierte Sprache
«aus Texteingabe»



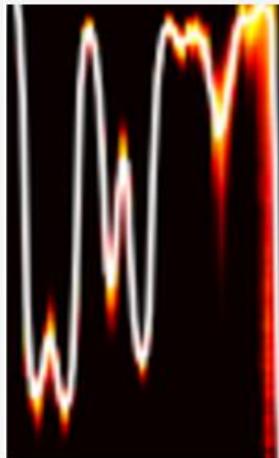
Generierte Musik
«ohne Inhaltsvorgabe»



WaveNet lässt Computergesproche natürlich klingen

von Henning Steier / 12.9.2017

Die Google-Tochter DeepMind macht auch Musik.



DeepMind lässt WaveNet Sprache

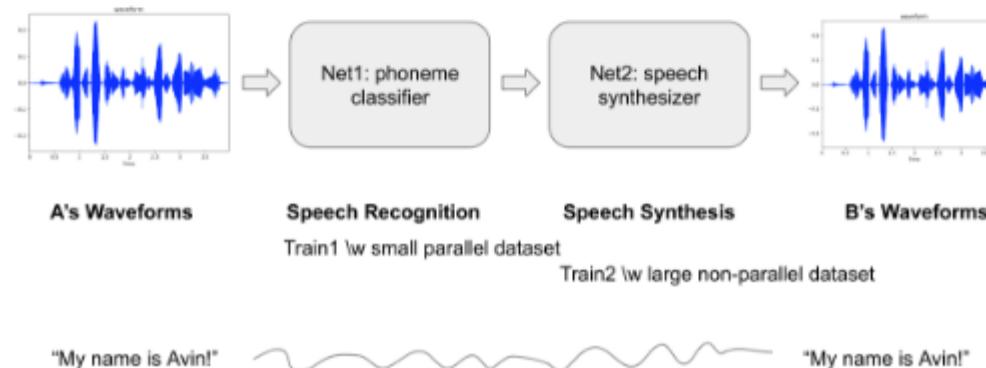
Die Google-Tochter DeepMind hat ein Spiel «Go» Schlagzeilen: einer der besten menschlichen Spieler schlägt einen der besten menschlichen Spieler. Londoner Unternehmen DeepMind erzeugt Sprache, die sehr gut klingt. Im Blogeintrag des Unternehmens wird der Massstab nimmt. Man hat

What if you could imitate a famous celebrity's voice or sing like a famous singer? This project started with a goal to convert someone's voice to a specific target voice. So called, it's voice style transfer. We worked on this project that aims to convert someone's voice to a famous English actress [Kate Winslet's voice](#). We implemented a deep neural networks to achieve that and more than 2 hours of audio book sentences read by Kate Winslet are used as a dataset.



Model Architecture

This is a many-to-one voice conversion system. The main significance of this work is that we could generate a target speaker's utterances without parallel data like <source's wav, target's wav>, <wav, text> or <wav, phone>, but only waveforms of the target speaker. (To make these parallel datasets needs a lot of effort.) All we need in this project is a number of waveforms of the target speaker's utterances and only a small set of <wav, phone> pairs from a number of anonymous speakers.



nerierte Sprache
is Texteingabe»

nerierte Musik
ine Inhaltsvorgabe»



Idee: Mehr Tiefe zum Lernen von Merkmalen

Klassische Bildverarbeitung

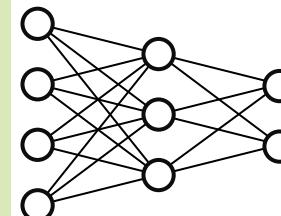


Merkalsextraktion
(SIFT, SURF, LBP, HOG, etc.)

(0.2, 0.4, ...)

(0.4, 0.3, ...)

Klassifikation
(SVM, Neuronales Netz, etc.)

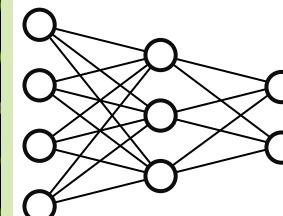
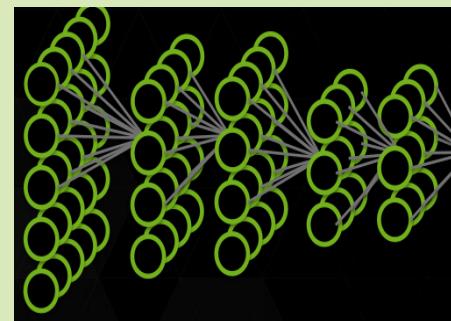


Containerschiff
Tiger
...

Mit Convolutional Neural Networks
(CNNs)

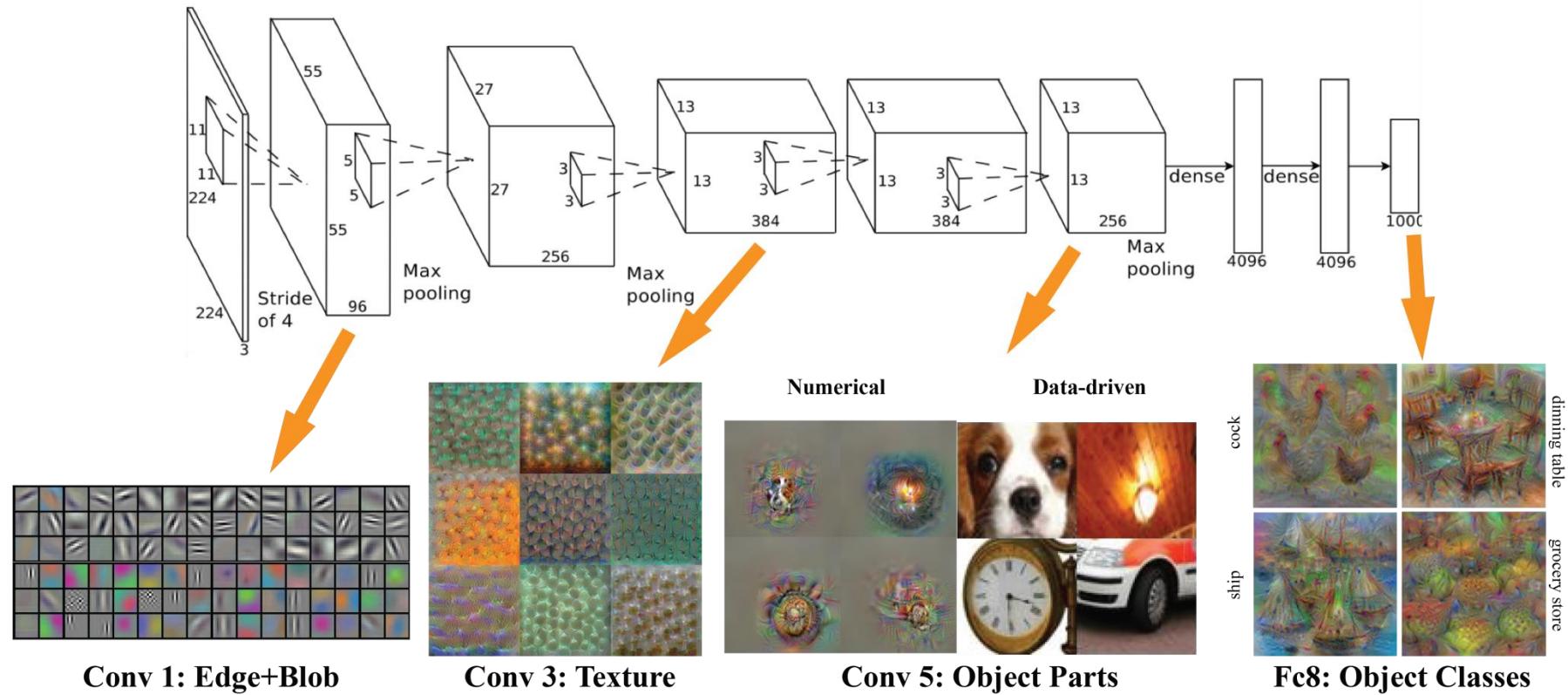


Nimmt rohe Pixel entgegen,
Merkmale werden mitgelernt!



Containerschiff
Tiger
...

Was «sieht» das Neuronale Netz? Hierarchien komplexer werdender Merkmale



Quelle: http://vision03.csail.mit.edu/cnn_art/data/single_layer.png