hochschule mannheim

EMB-Lab, Fak. N

Architekturen & Layer II
- Sequentielle Daten, Sonstige Anwendungen

DLM – Deep Learning Methoden

Benjamin Kraus

Kevin Höfle, Prof. Dr. Marcus Vetter

Mannheim, 18.11.2019

Inhalt

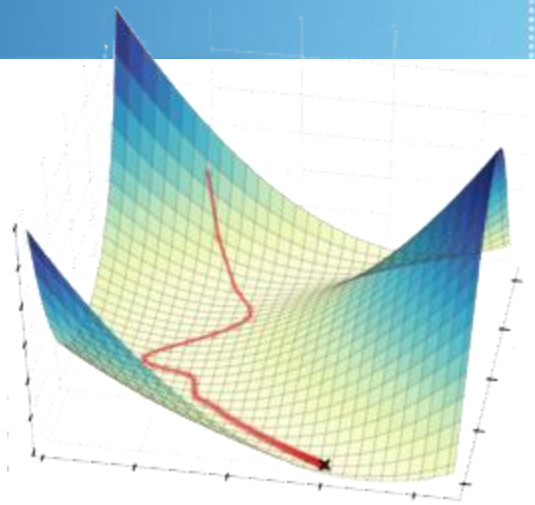
- Wiederholung

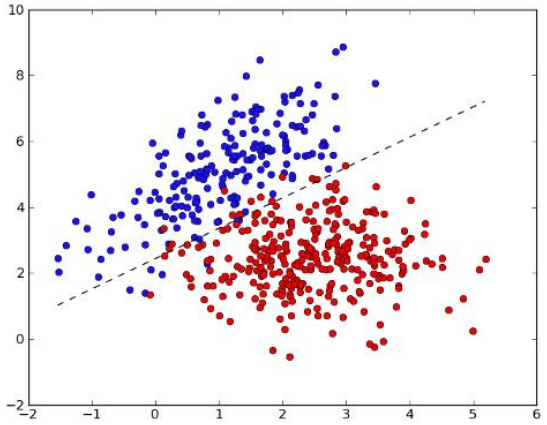
- Sequentielle Probleme
- WaveNet & n-D Faltung
- RNN
- Backpropergation Through Time
- LSTM
- GRU usw.

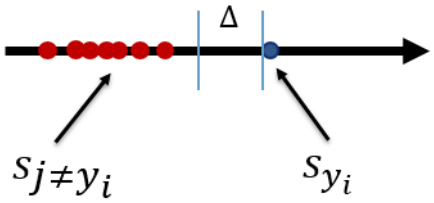
- Embeddings & Feature Space Interpolation

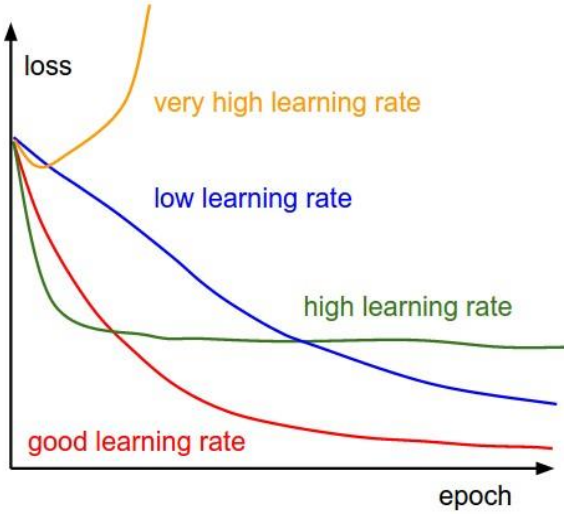
Recap

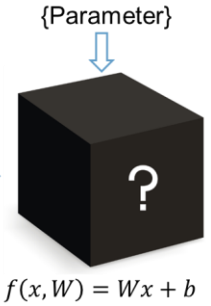
- Lineare Klassifikation
- Loss Function
- Gradienten Abstieg
- Back Propergation







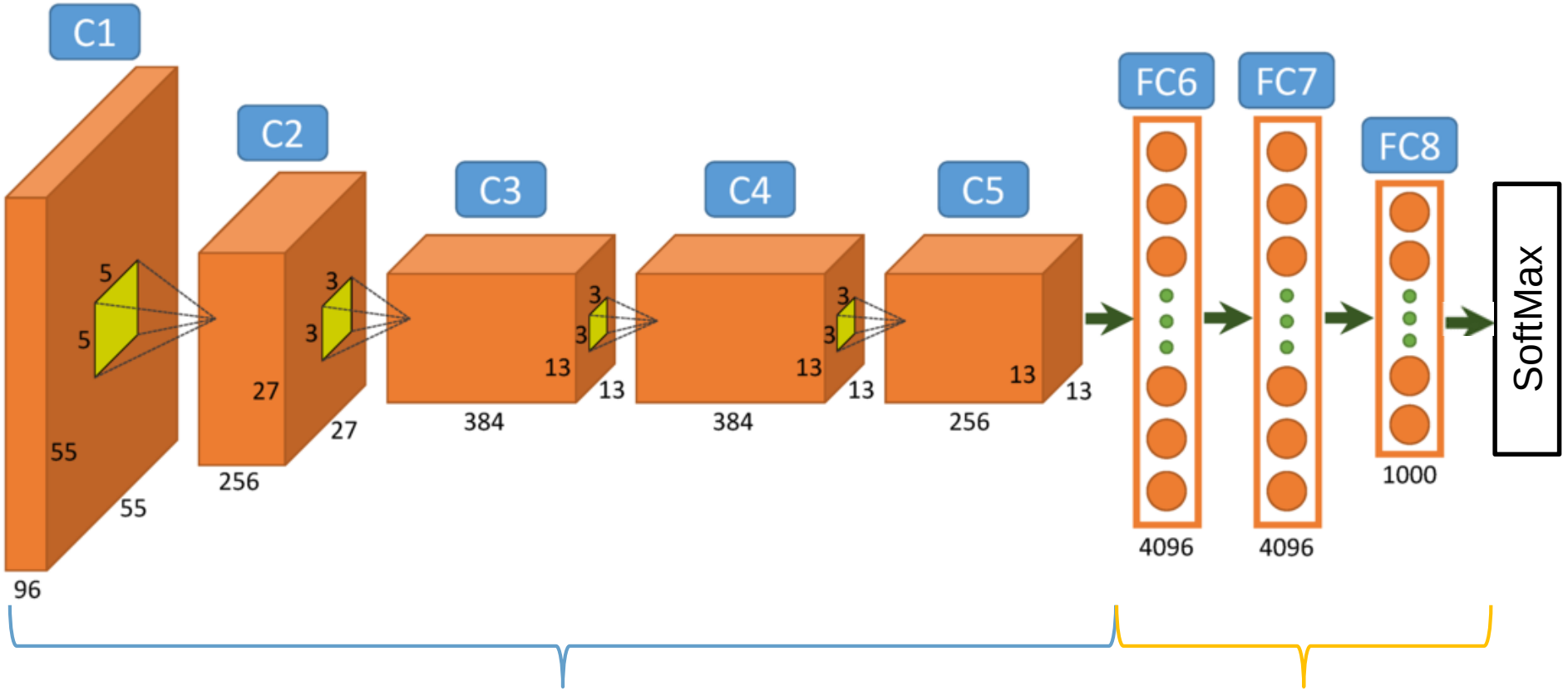




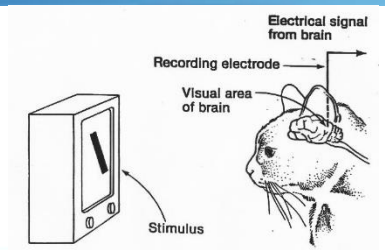
{scoreCat = 2,5 }
{scoreDog = 3,1}
{scoreCar = -0,1}
…

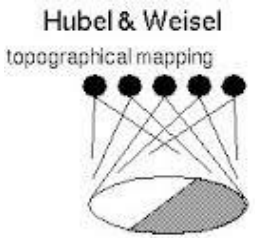
Neuronale Netze Recap

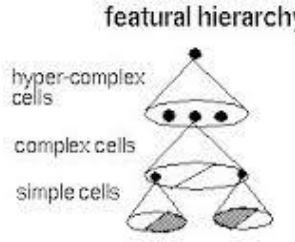
- Conv Layer und Dense Layer

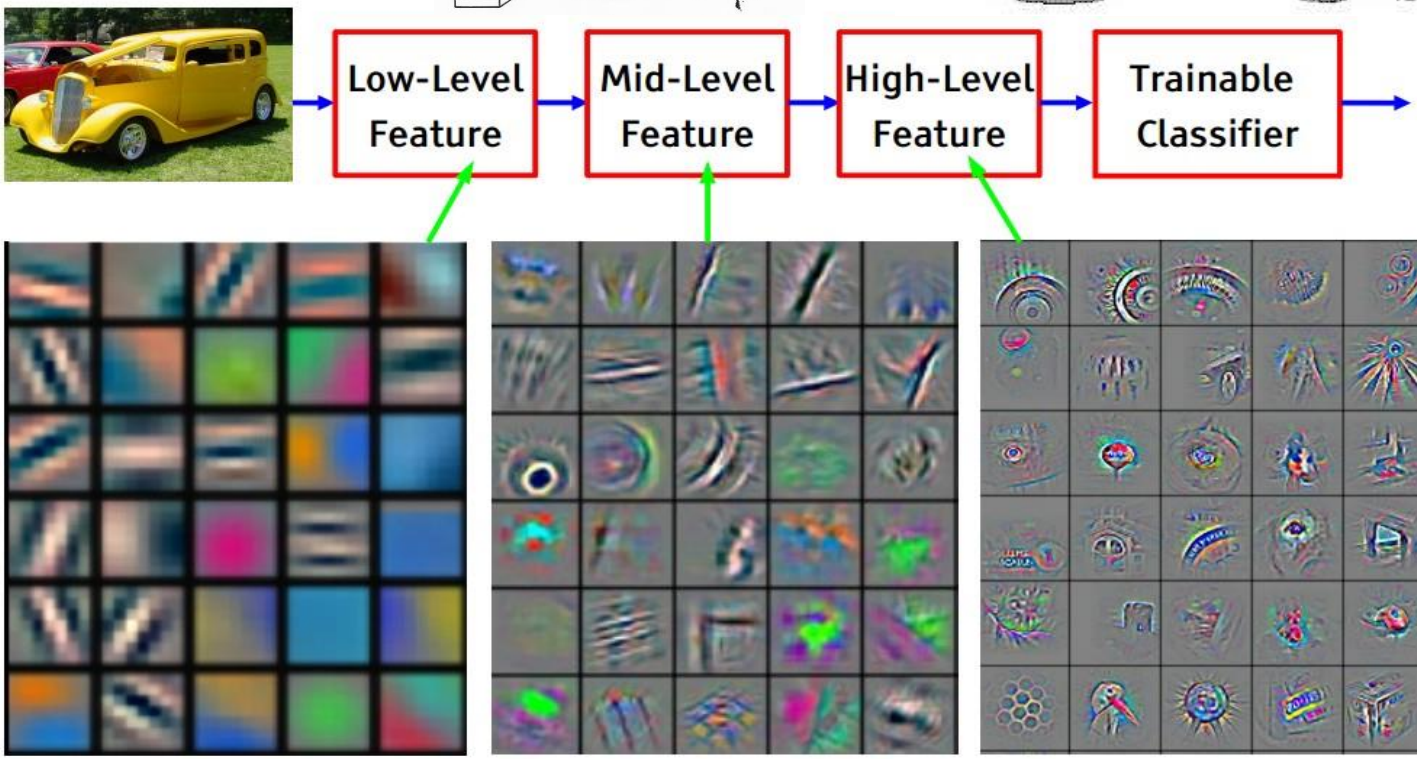


Neuronale Netze Recap





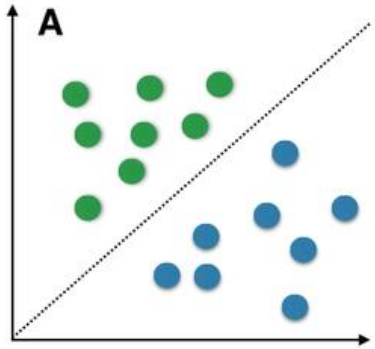


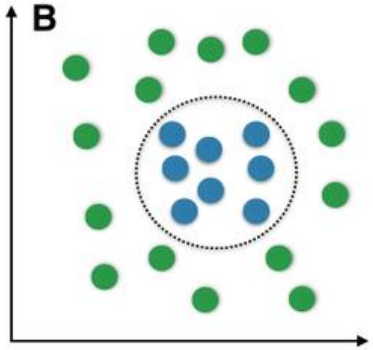


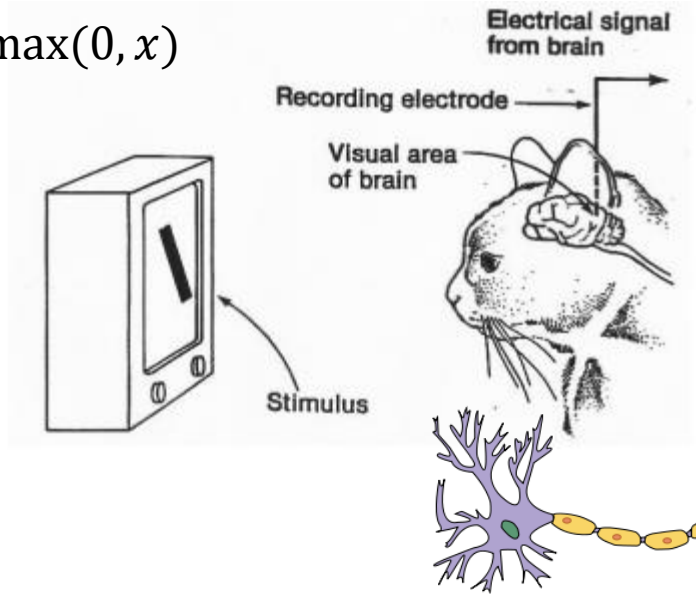
Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Neuronale Netze Recap

- Activation Functions:
  - Sigmoid $\sigma(x) = 1/(1 + e^{-x})$
  - Tanh $f(x) = tanh(x)$
  - ReLU $f(x) = \max(0, x)$



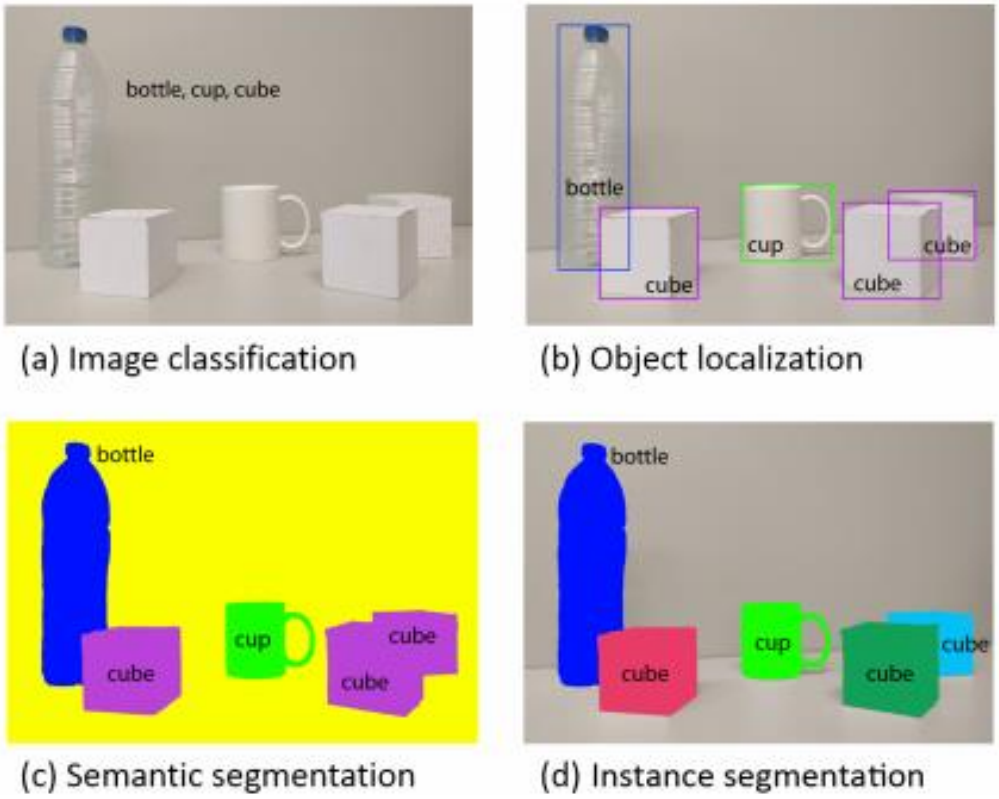




[Daniel Smilkov and Shan Carter]

Neuronale Netze Recap

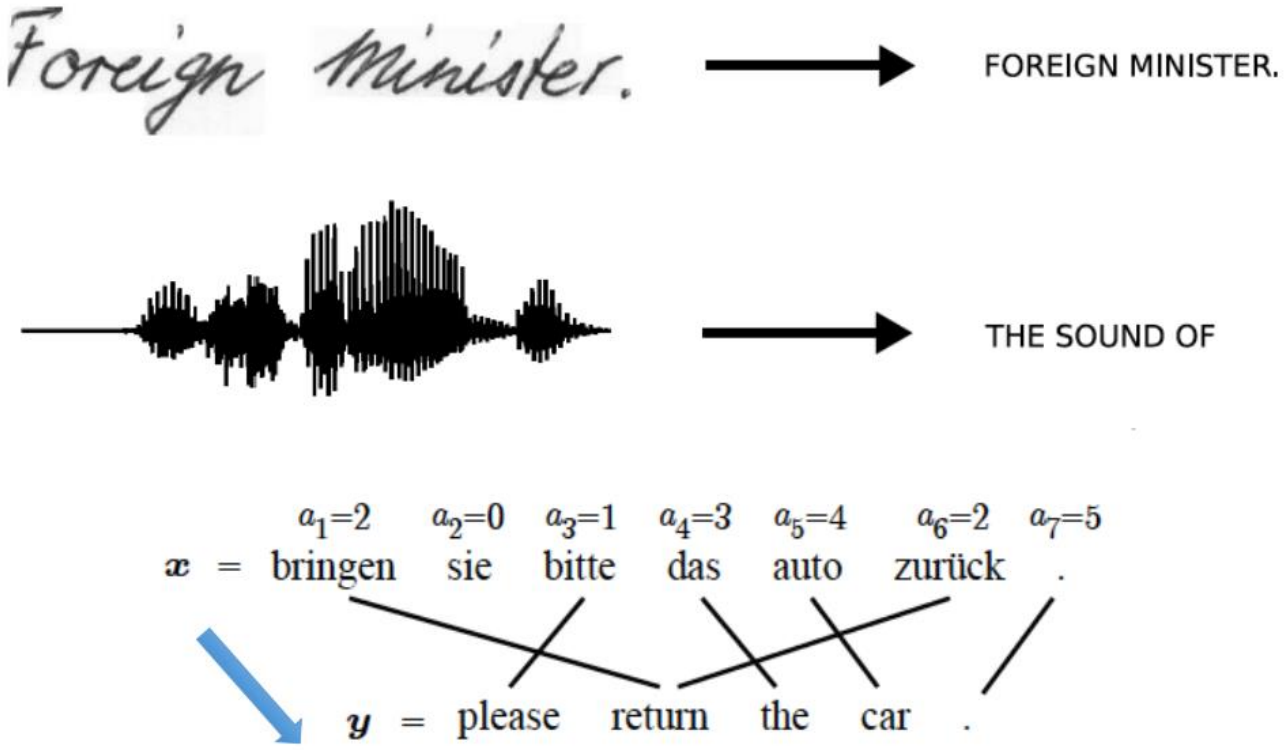
- Klassifikation: VGG, ResNet, GoogLeNet
- Lokalisation: R-CNN, YOLO
- Segmentierung: DeconvNet, UNet



[Grafik: Garcia-Garcia]

Sequentielle Probleme

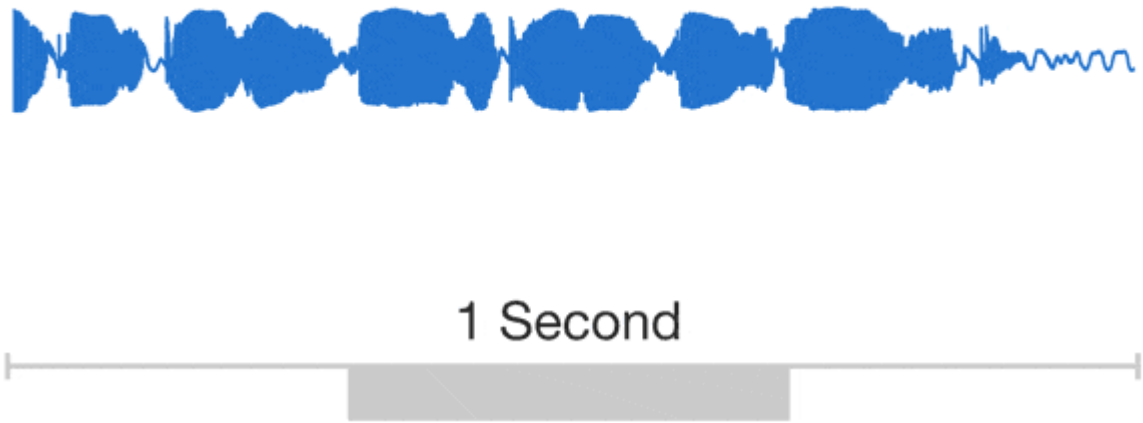
- Es gibt eine Vielzahl von Problemen bei welchen zeitliche Abfolge eine Rolle spielt



Credit: Alex Graves, Kevin Gimpel, Dhruv Batra

Sequentielle Probleme

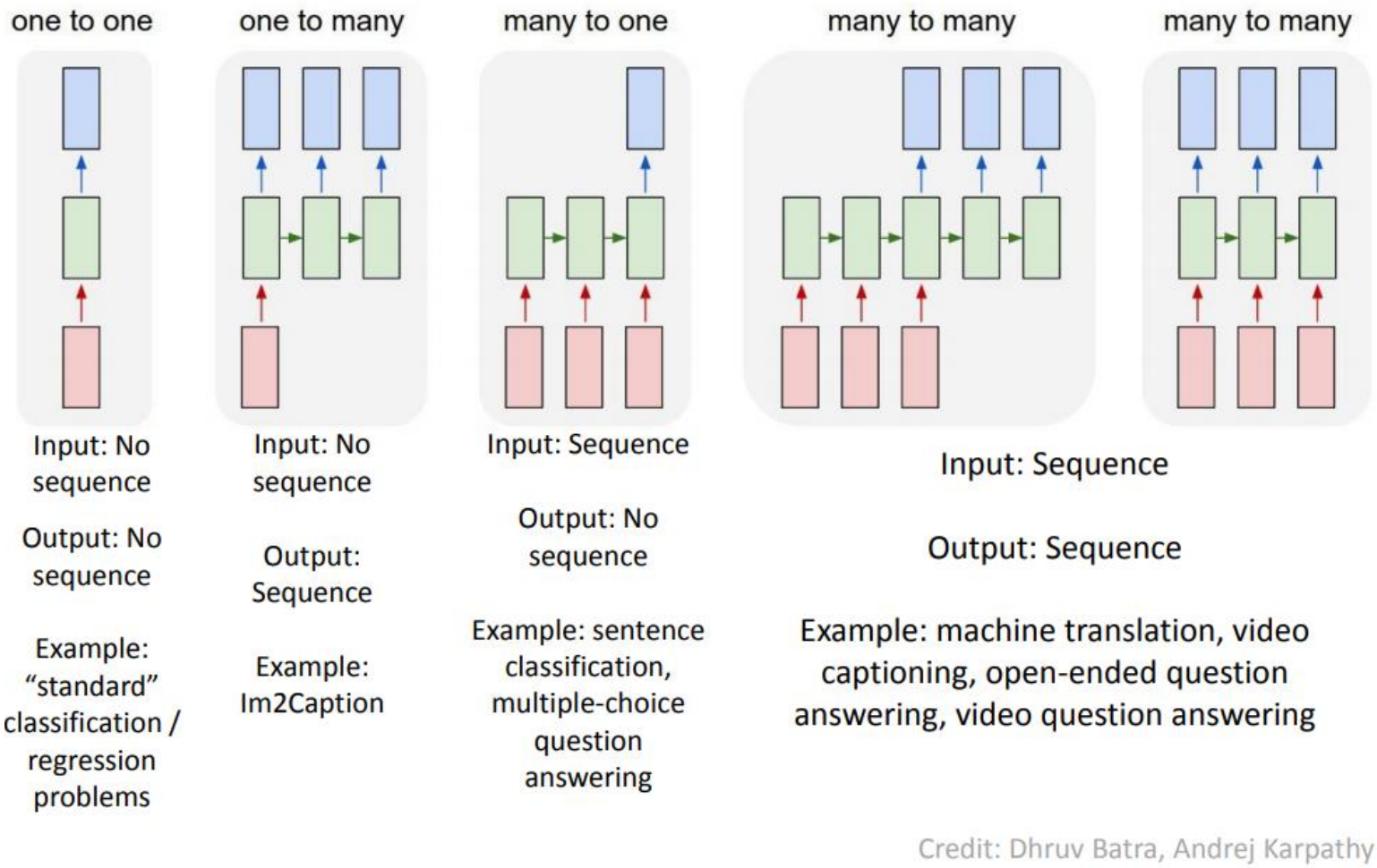
- Unterschiedliche Abstraktionsebenen
- Bsp. Rede:
  - Rede -> Kapitel -> Satz -> Wort ->Laut -> Schwingung -> Abtastwert



[Grafik: Dielemann, Oord et al.]

Sequentielle Probleme

- Arten von seq. Problemen:



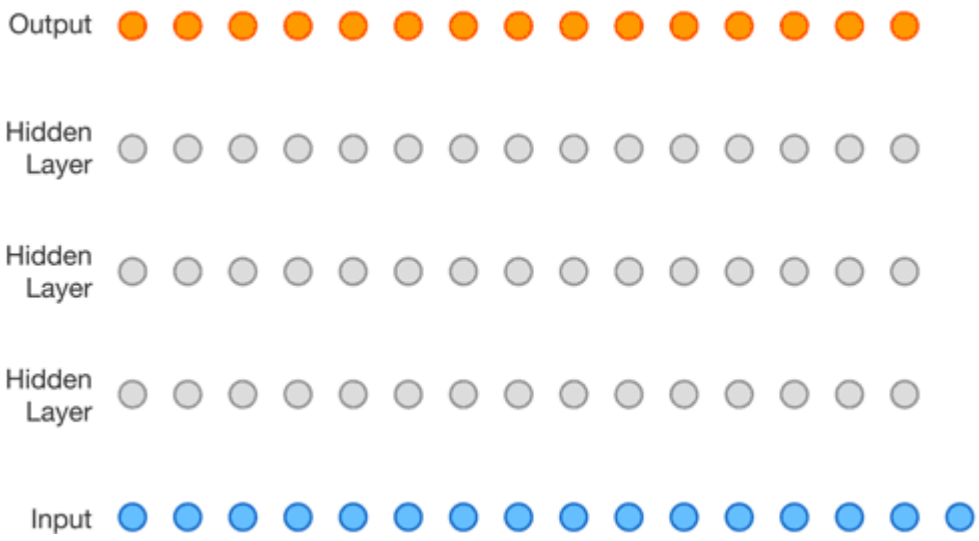
Credit: Dhruv Batra, Andrej Karpathy

Sequentielle Probleme

- Sequenz zu Sequenze (many to many)
- z.B: Sprache rein Sprache raus.

Figure 1: A second of generated speech.

- Ansatz mit Faltung:
  - Zeit ist auch nur eine Dimension über die man Falten kann



[Grafik: Oord et al.]

Sequentielle Probleme

- Problem: Das Receptive Field von Neuronen ist begrenzt
- Ein Neuron kann nur Information aus benachbarten Abtastwerten sehen

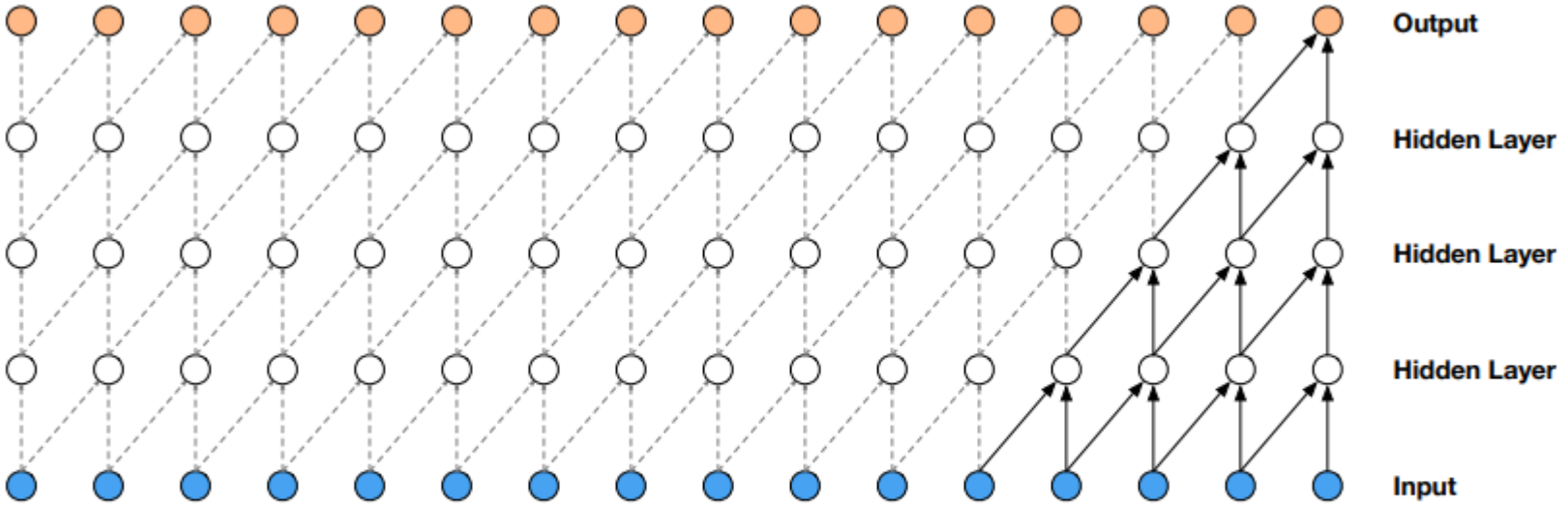


Figure 2: Visualization of a stack of causal convolutional layers.

[Grafik: Oord et al.]

Sequentielle Probleme

- Lösung: dilated Convs

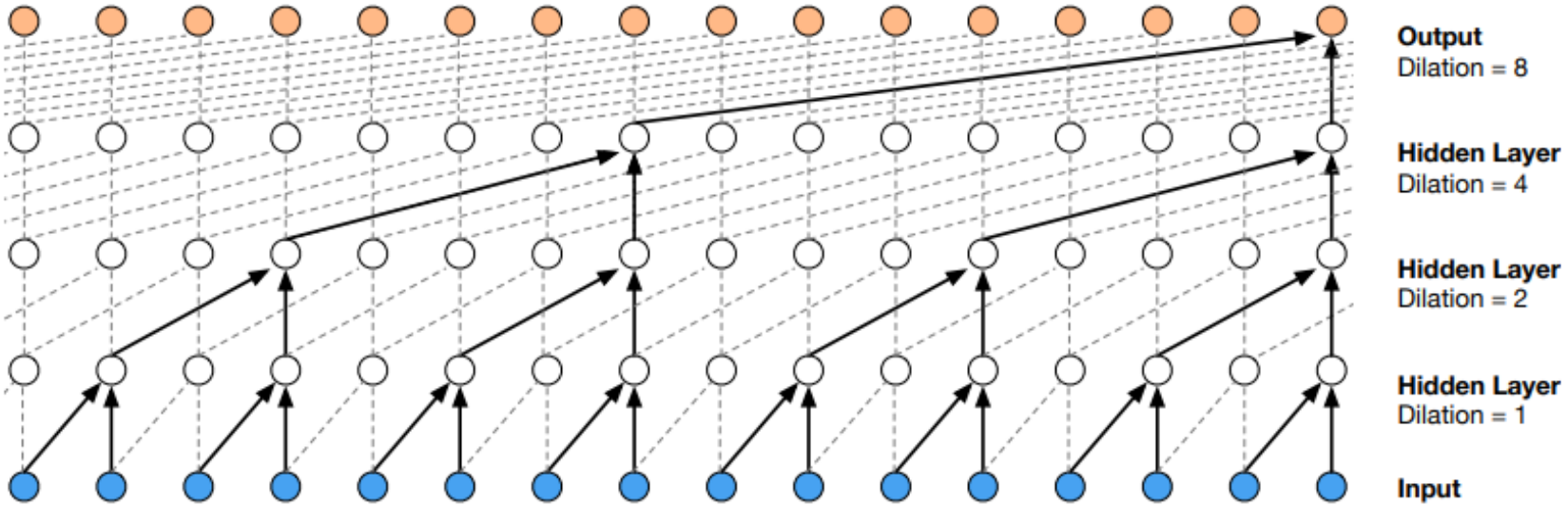
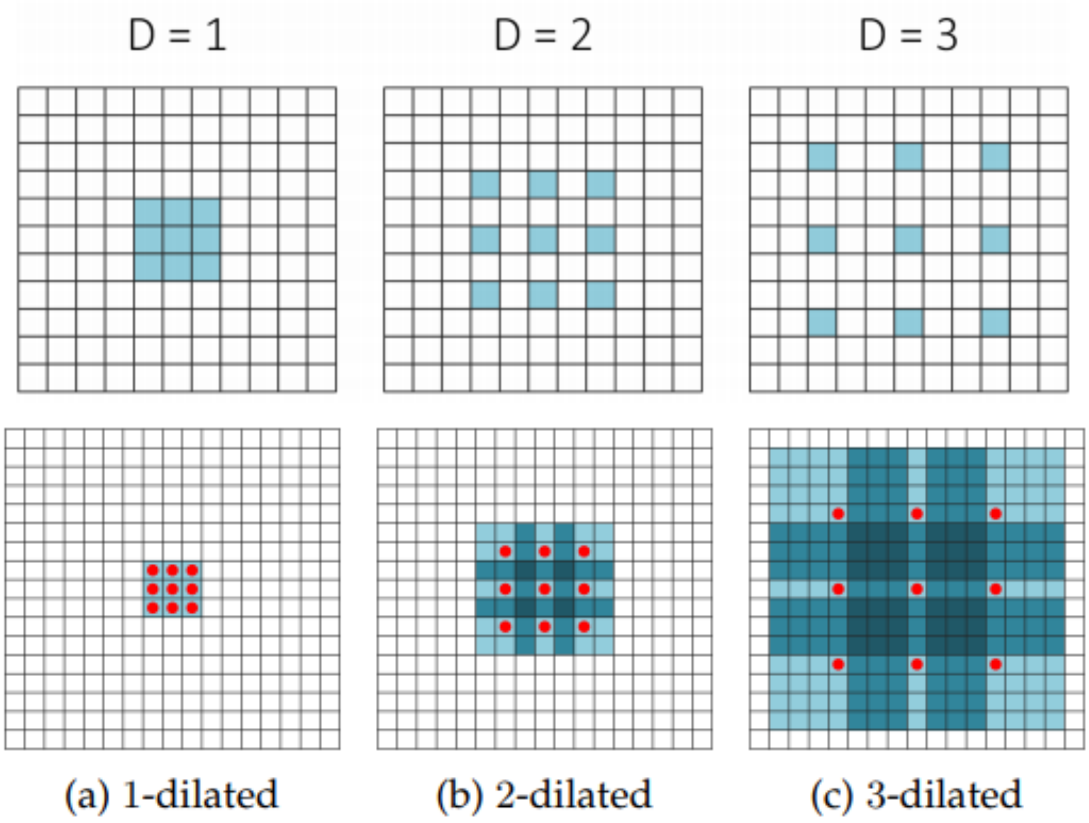


Figure 3: Visualization of a stack of *dilated* causal convolutional layers.

Sequentielle Probleme

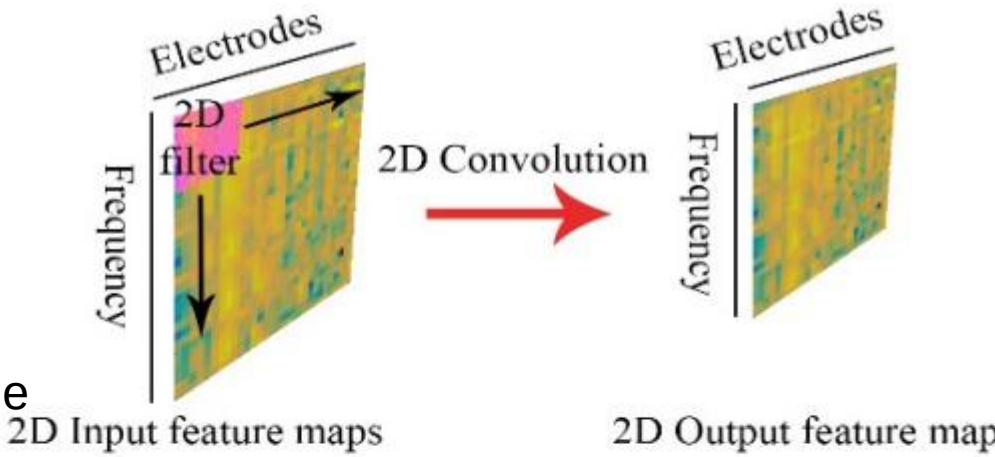
- Lösung: dilated Convs
- Funktioniert auch auf Bildern:
- Vergrößert auch hier das Receptive Field

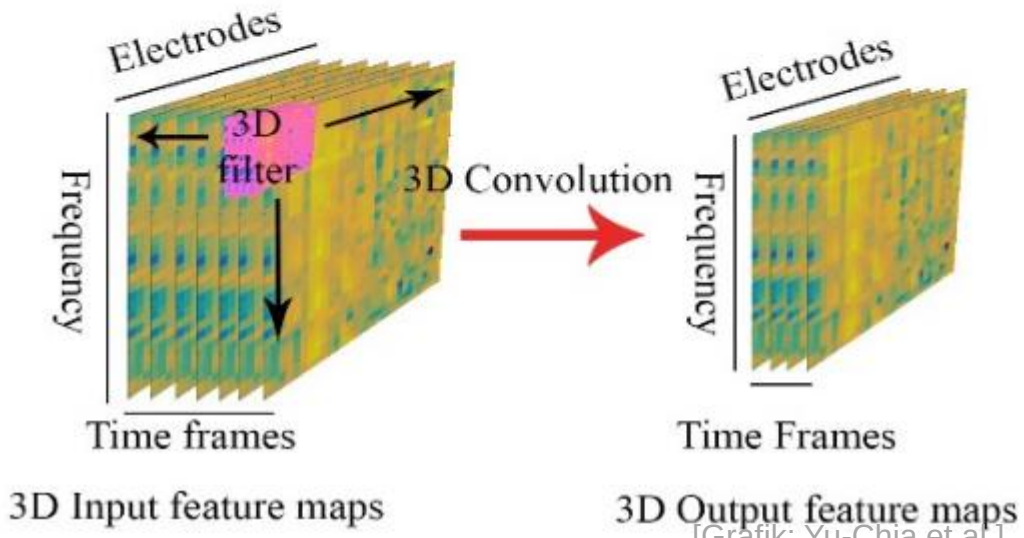


[Grafik: Yu et al.]

3-D Conv

- Zeit ist nur eine weitere Dimension:
- Bilder sind HxBxC
  - z.B. FullHD
  - 1920x1080x3
- Videos sind aufeinander folgende Bilder
  - z.B. 2sec FullHD bei 60fps
  - 120x1920x1080x3

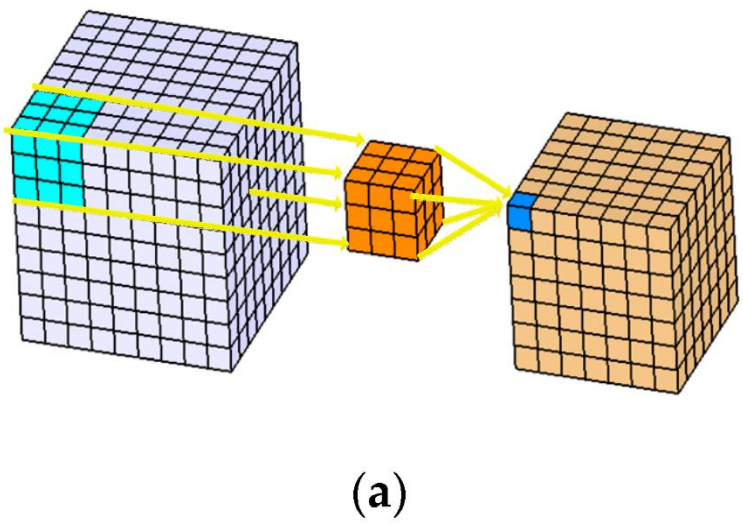




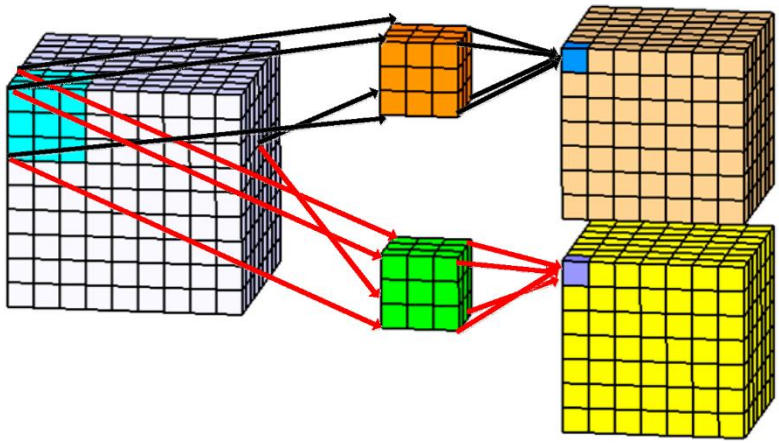
[Grafik: Yu-Chia et al.]

3-D Conv

- Filter Kernel sind 3D (z.B. 3x3x3 (x1 bei einem Channel))
- Strides sind 3D
- Faltungsergebnisse sind 3D + Anzahl der Filter Kernels
  z.B. T x H x W x C



(a)

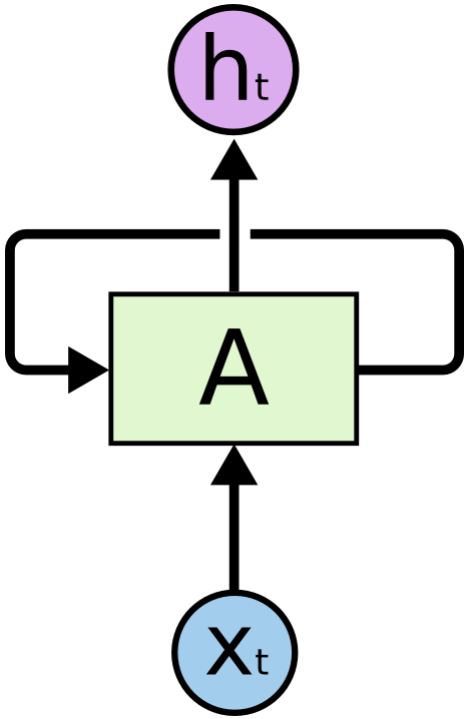


(b)

[Grafik: Mei et al.]

Recurrent Neural Networks

- NN mit „Loops“
- Gut geeignet für sequentielle Probleme
- speech recognition, language modeling, translation, image captioning…

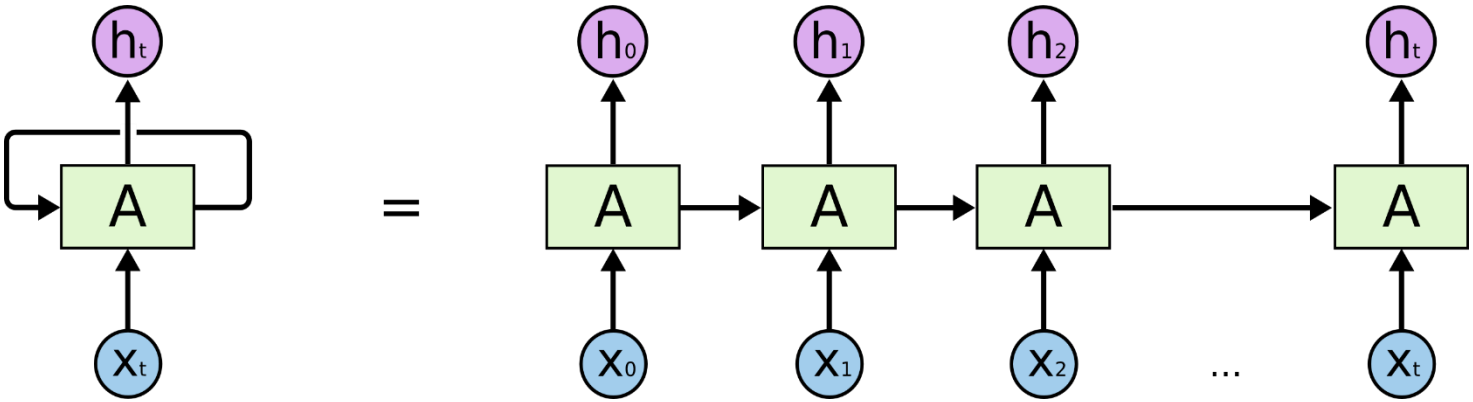


[Chris Olah, Google Brain, Blog post]

Recurrent Neural Networks

- ANN mit „Loops“
- Loops können über die Zeitachse ausgerollt werden
- Eine lineare Kette entsteht:

  (und wird schnell „tief“)

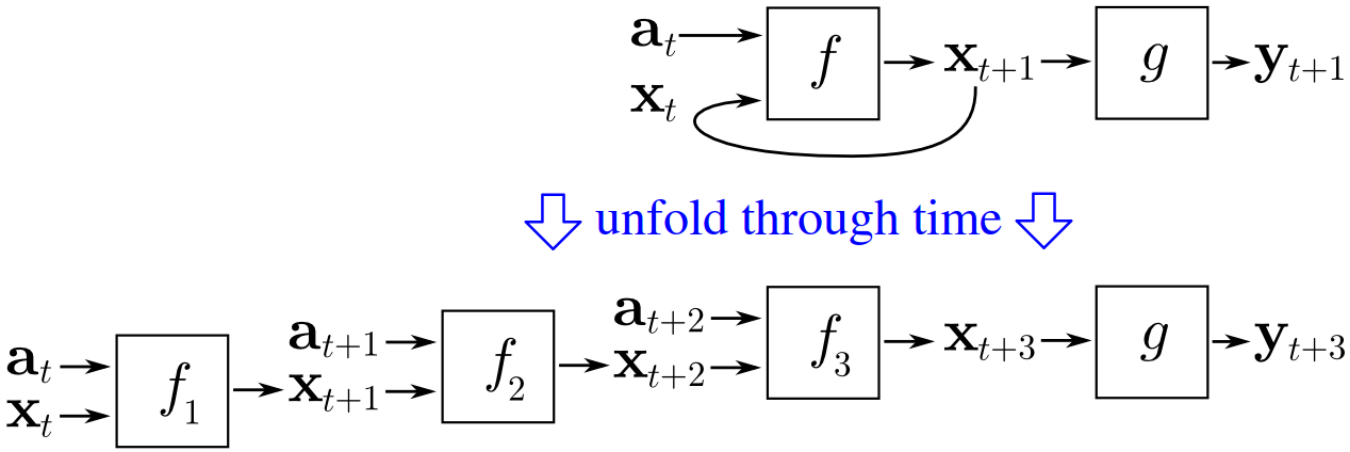


[Chris Olah, Google Brain, Blog post]

Recurrent Neural Networks

- Das Netz kann dann mit BPTT trainiert werden

(Backpropergation Through Time)



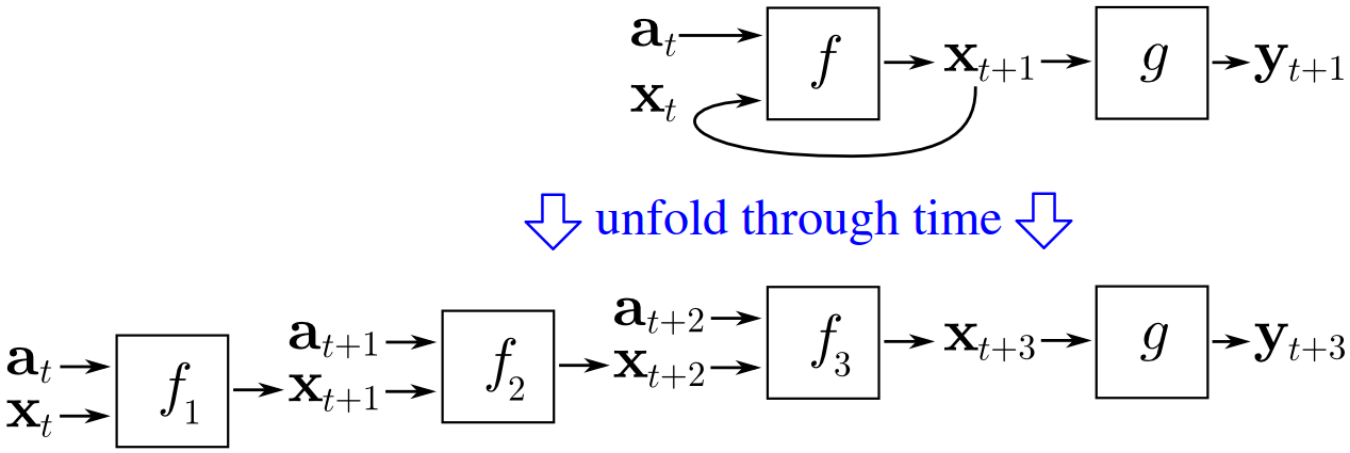
[Wiki: Backpropagation through time]

Recurrent Neural Networks

- Das Netz kann dann mit BPTT trainiert werden

(Backpropergation Through Time)

- Dies skaliert aber bei langen Sequenzen schlecht.
  - Zum Training werden alle Aktionen bis zum letzten Zeitschritt gebraucht: $\langle \mathbf{x}_t, \mathbf{a}_t, \mathbf{a}_{t+1}, \mathbf{a}_{t+2}, \ldots, \mathbf{a}_{t+k-1}, \mathbf{y}_{t+k} \rangle$

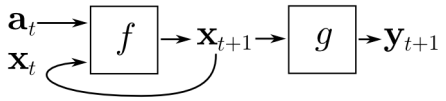


[Wiki: Backpropagation through time]

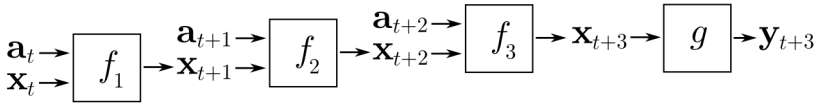
Recurrent Neural Networks

- Pseudo-code:

```
Back_Propagation_Through_Time(a, y)    // a[t] is the input at time t. y[t] is the output
    Unfold the network to contain k instances of f
    do until stopping criteria is met:
        x = the zero-magnitude vector;// x is the current context
        for t from 0 to n - k         // t is time. n is the length of the training sequence
            Set the network inputs to x, a[t], a[t+1], ..., a[t+k-1]
            p = forward-propagate the inputs over the whole unfolded network
            e = y[t+k] - p;           // error = target - prediction
            Back-propagate the error, e, back across the whole unfolded network
            Sum the weight changes in the k instances of f together.
            Update all the weights in f and g.
            x = f(x, a[t]);           // compute the context for the next time-step
```



unfold through time

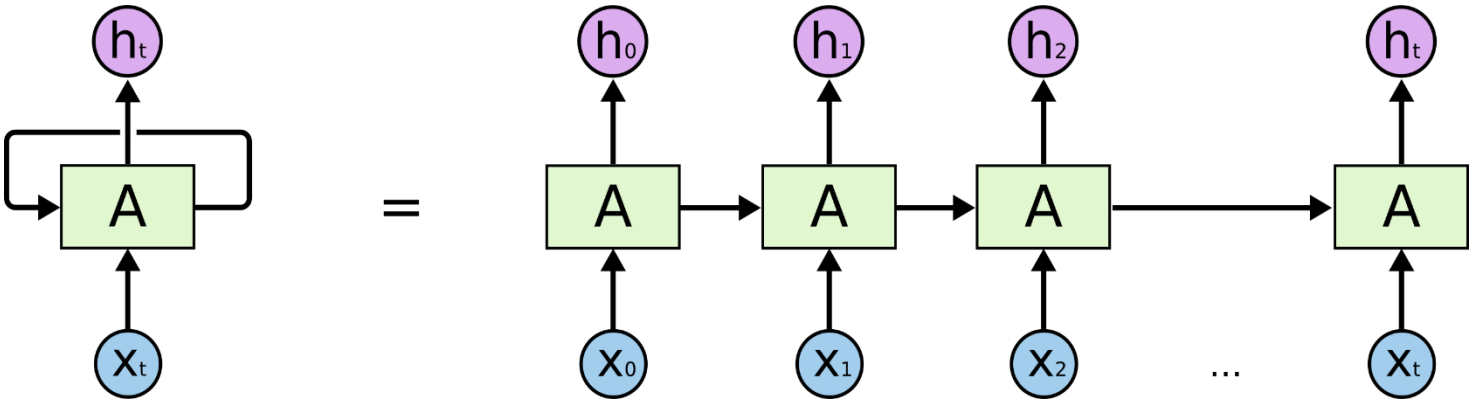


[Wiki: Backpropagation through time]

Recurrent Neural Networks

- ANN mit „Loops“
- Loops können über die Zeitachse ausgerollt werden
- Eine lineare Kette entsteht:

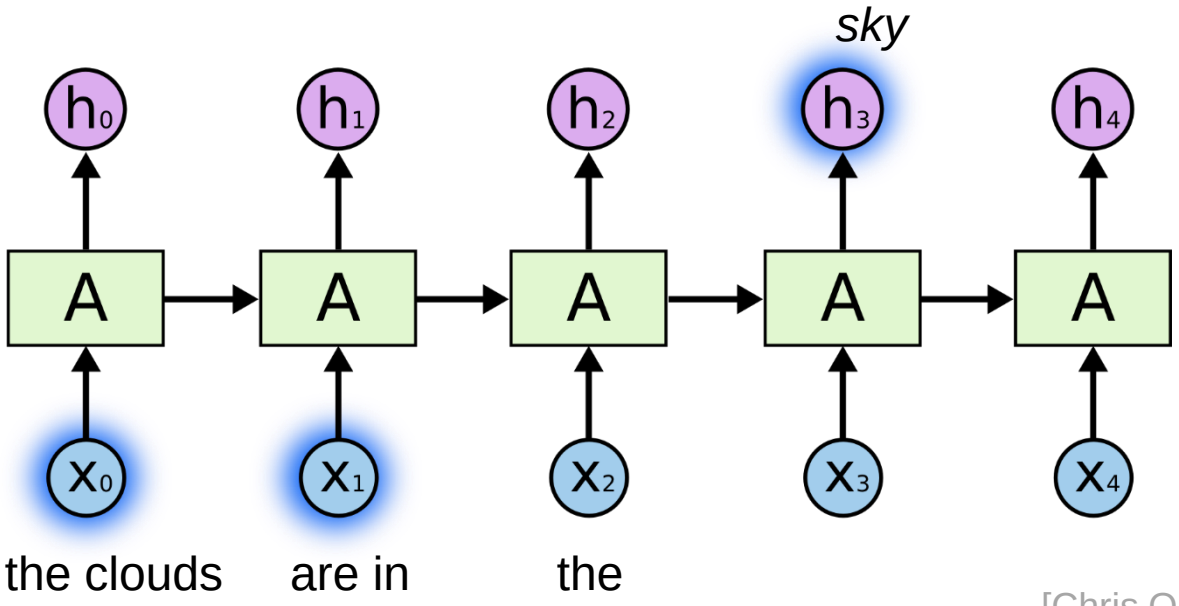
  (und wird schnell „tief“)
- Es entsteht das **„Problem of Long-Term Dependencies“**



[Chris Olah, Google Brain, Blog post]

Recurrent Neural Networks

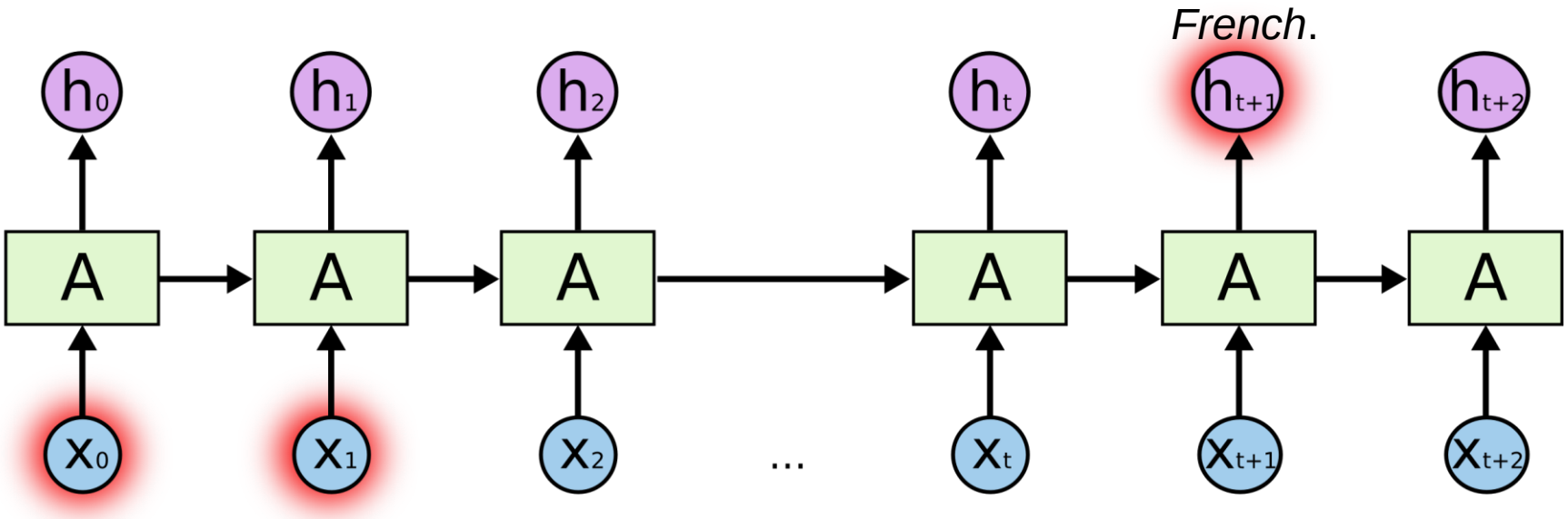
- **Problem of Long-Term Dependencies**
  - Die Distanz der Abhänigkeit eines Eingabe zu einer Ausgabe
  - Sprachsemantik Beispiel: „the clouds are in the *sky*”
    - ➢Kontext ist nahe beim Problem
    - ➢Suchraum nach relevanter Information ist klein



[Chris Olah, Google Brain, Blog post]

Recurrent Neural Networks

- **Problem of Long-Term Dependencies**
  - Die Distanz der Abhänigkeit eines Eingabe zu einer Ausgabe
  - Sprachsemantik Beispiel: „I grew up in France… I speak fluent *French*.”



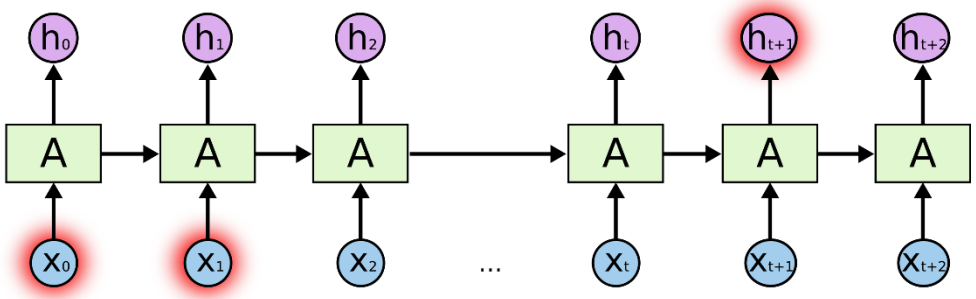
[Chris Olah, Google Brain, Blog post]

Recurrent Neural Networks

- **Problem of Long-Term Dependencies**

  - Beim Training muss über viele Schritte ausgerollt werden
  - Der Suchraum nach relevanter Information steigt
  - Zusätzlich neigen RNN stark zu vanishing / exploding Gradients

  - Stark vereinfacht, Detailliere Untersuchungen:
  - Hochreiter91 & Bengio94

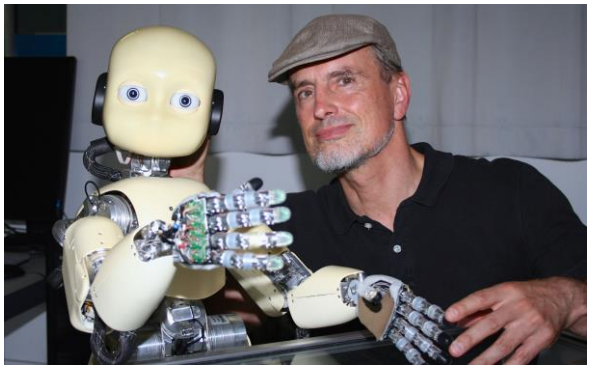


[Chris Olah, Google Brain, Blog post]

Long Short Term Memory networks

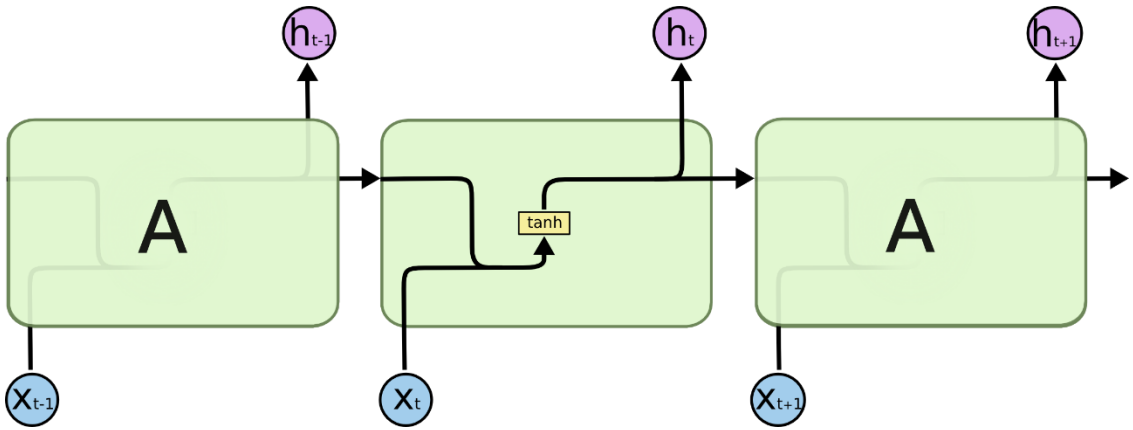
- **Long Short Term Memory networks**
  - Beschrieben 1997 von
  - Schmidhuber (Schweizer Forschungsinstituts für Künstliche Intelligenz IDSIA.)
  - Hochreiter (TU München)

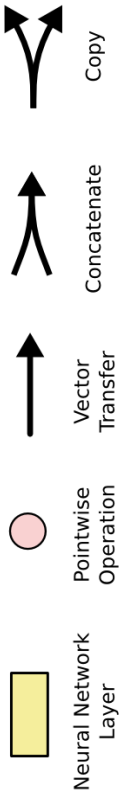
  … soll diese Probleme beheben



Long Short Term Memory networks

- LSTM:
- Spezialisierung von RNNs
  - RNN hat einen Layer in der “Loop”

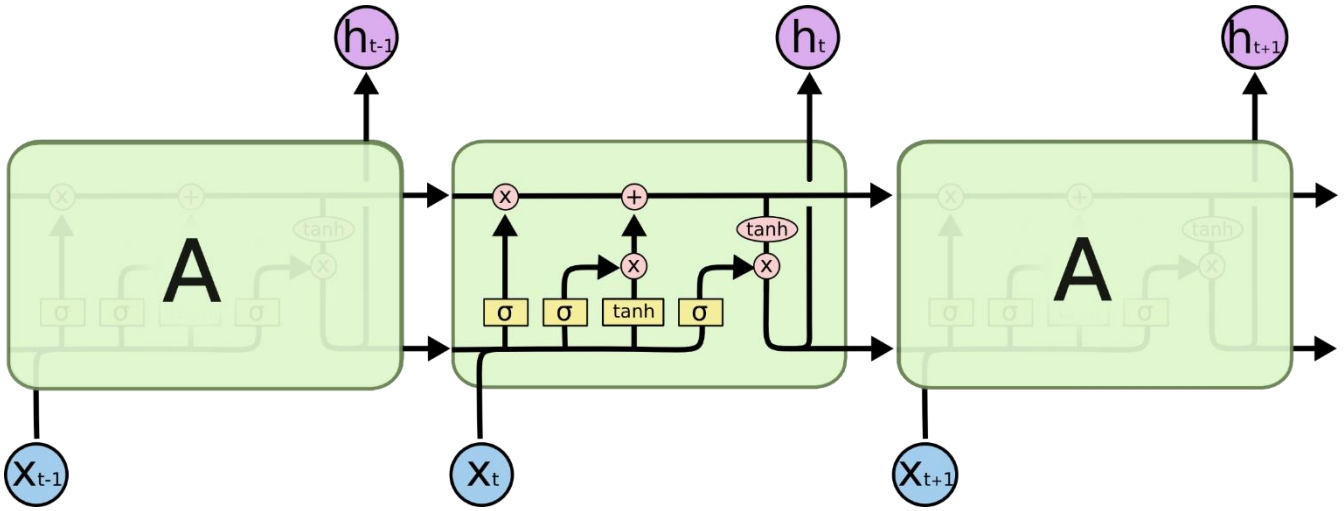


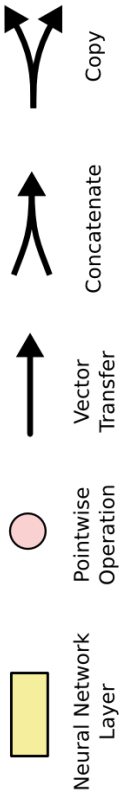


[Chris Olah, Google Brain, Blog post]

Long Short Term Memory networks

- LSTM ist ein wenig komplizierter:
  - Vier NN-Layer (manchmal auch Sublayer genannt)
  - Diese haben je ihre eigenen Gewichte/Bias…

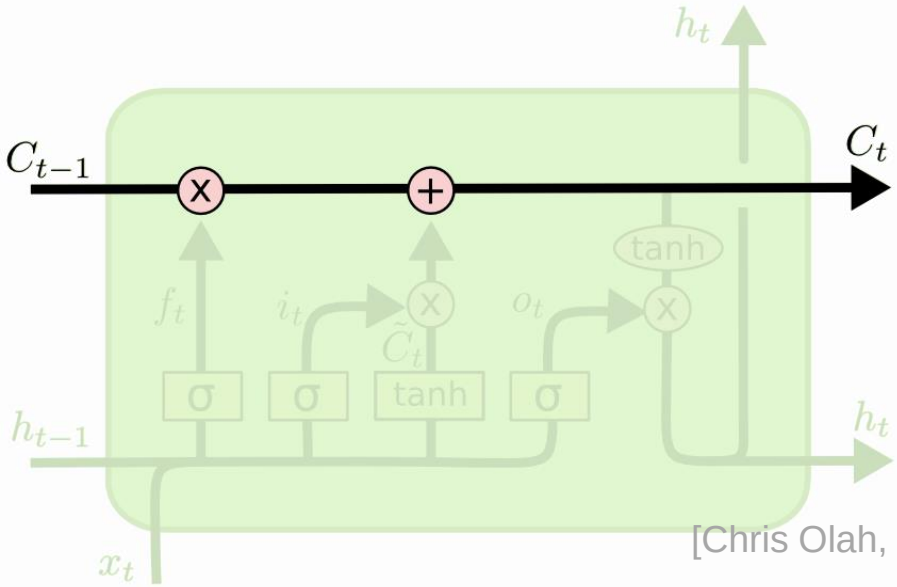


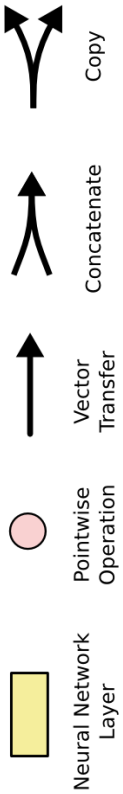


[Chris Olah, Google Brain, Blog post]

Long Short Term Memory networks

- Der Cell State:
  - Informationsfluss durch die komplette Stuktur
  - Nur einige wenige Lineare Operationen

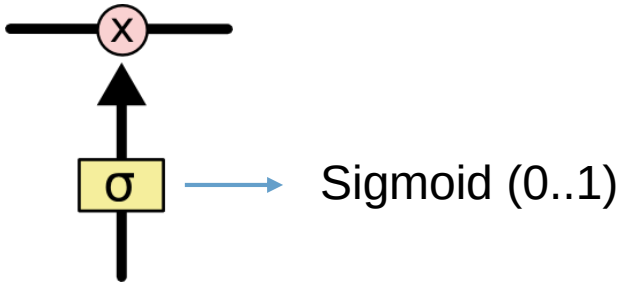




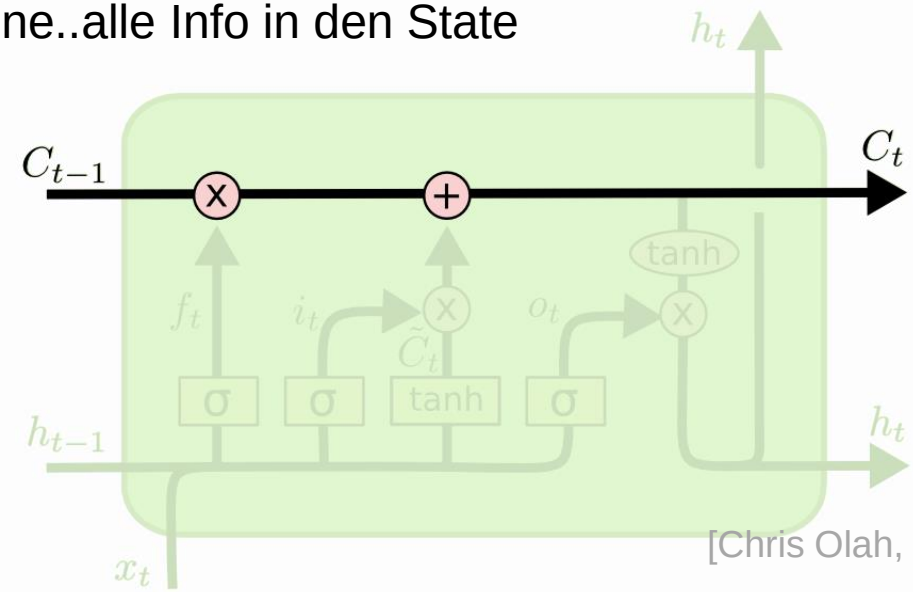
[Chris Olah, Google Brain, Blog post]

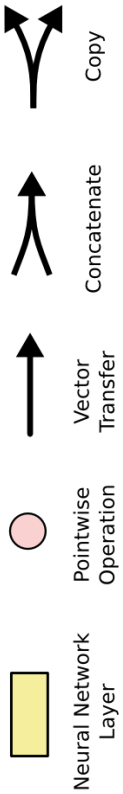
Long Short Term Memory networks

- Der Cell State:
  - Information werden durch Gates hinzugefügt oder entfernt



  - Lässt keine..alle Info in den State



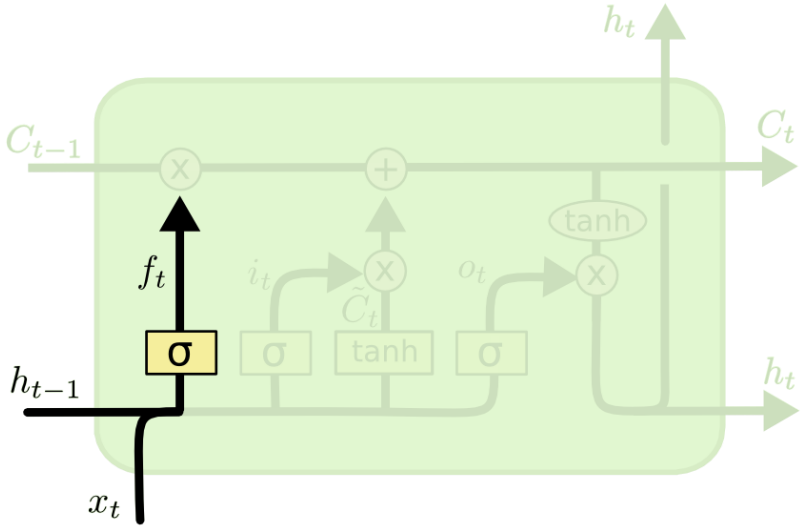


[Chris Olah, Google Brain, Blog post]

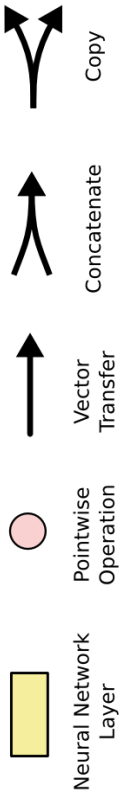
Long Short Term Memory networks

- Gates Step by Step:
  - Forget Gate: Wirft Infromation aus dem State Flow
  - Sprachsemantik Beispiel:

  “the cell state might include the gender of the present subject, so that the correct pronouns can be used. When we see a new subject, we want to forget the gender of the old subject.”[Chris Olah]



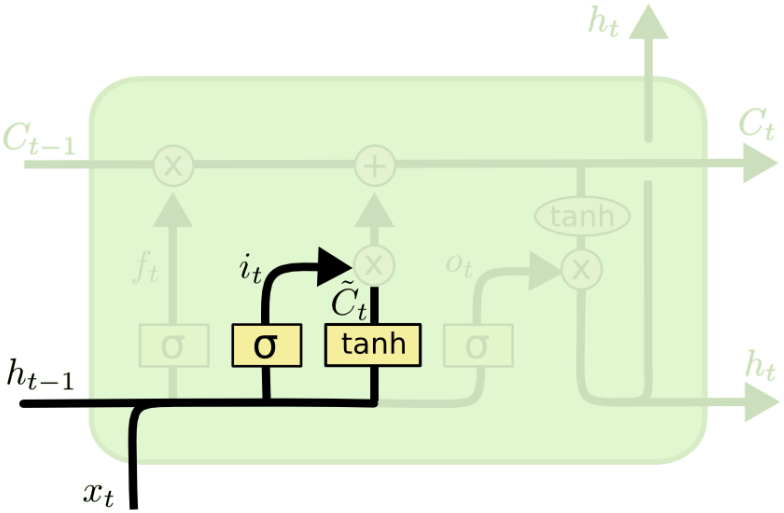
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$



[Chris Olah, Google Brain, Blog post]

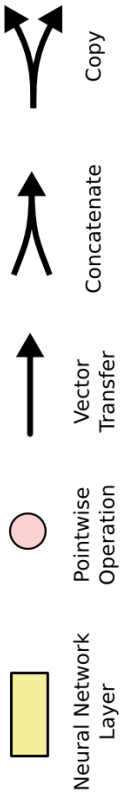
Long Short Term Memory networks

- Gates Step by Step:
  - Was in den State aufgenommen wird ist ein zweiteiliger Prozess
  - input gate layer entscheidet wo neue Values übernommen werden
  - Tanh-layer generiert die Kanidaten für diese neuen Values



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

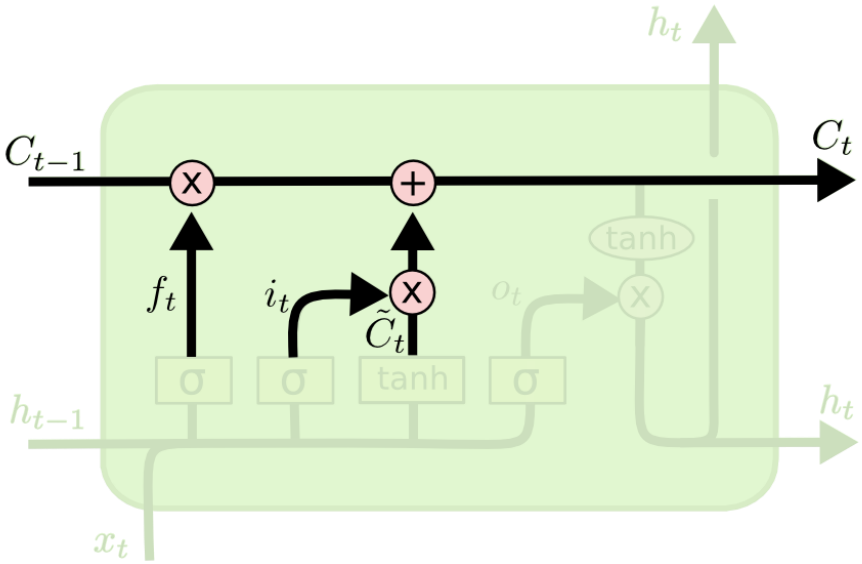


[Chris Olah, Google Brain, Blog post]

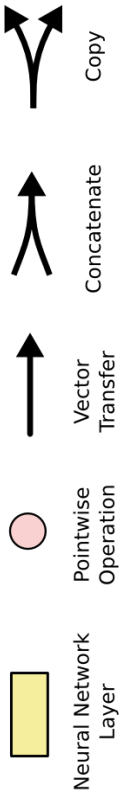
Long Short Term Memory networks

- Gates Step by Step:
  - State Update durchführen
  - Sprachsemantik Beispiel:

  “this is where we’d actually drop the information about the old subject’s gender and add the new information, as we decided in the previous steps.”[Olah]



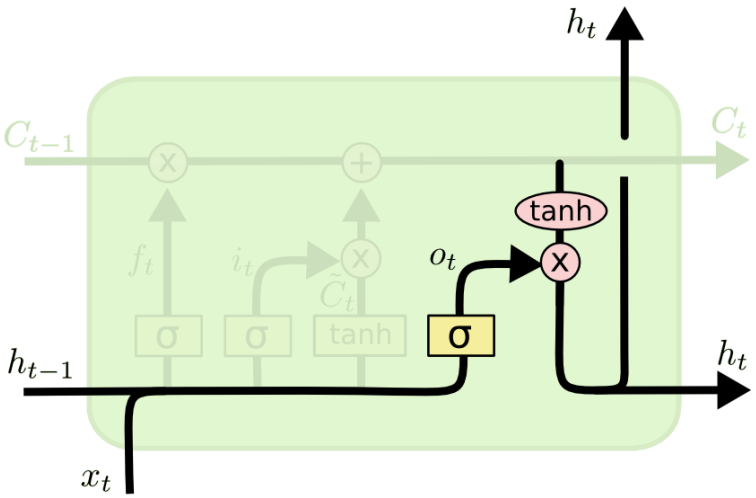
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$



[Chris Olah, Google Brain, Blog post]

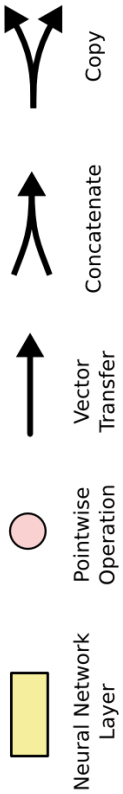
Long Short Term Memory networks

- Gates Step by Step:
  - Einen Output generieren:
    - Aus dem Cell State und dem eigentlichen RNN Output



$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

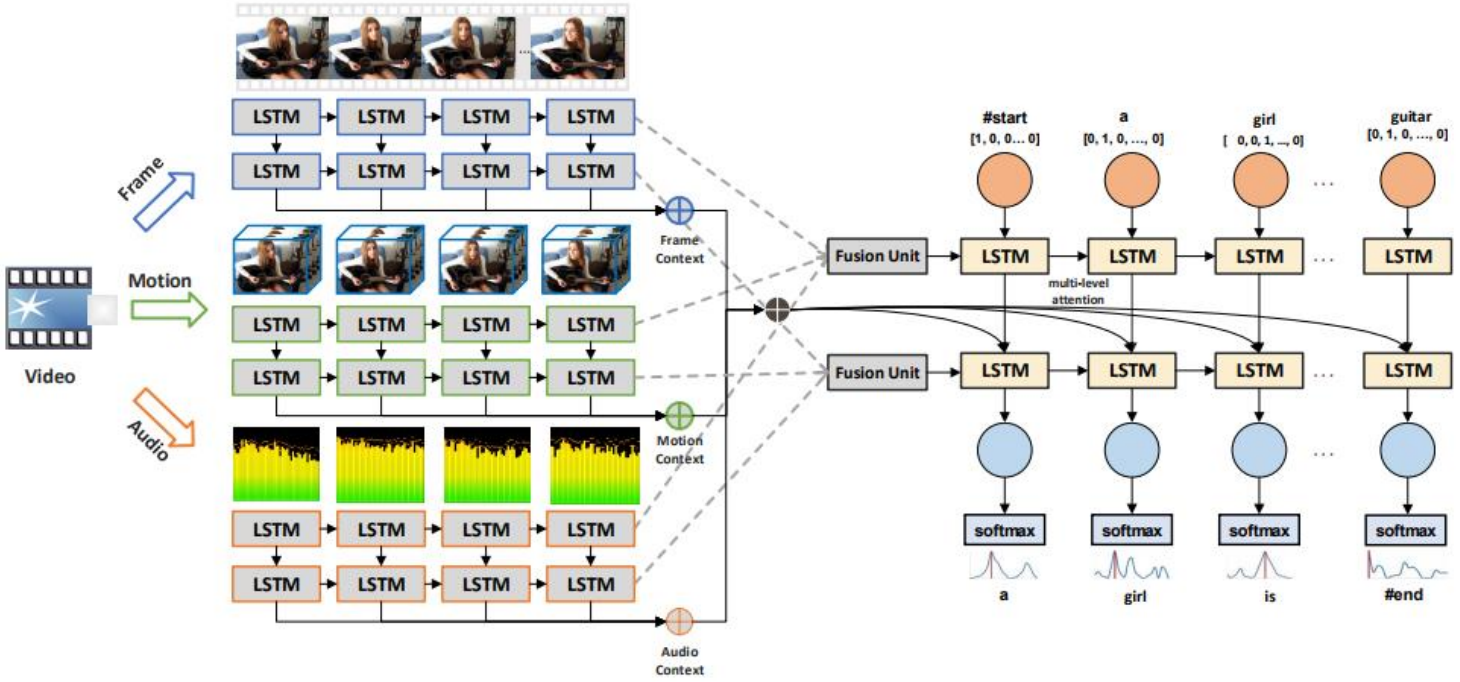
$$h_t = o_t * \tanh\left(C_t\right)$$



[Chris Olah, Google Brain, Blog post]

LSTM for Video Captioning

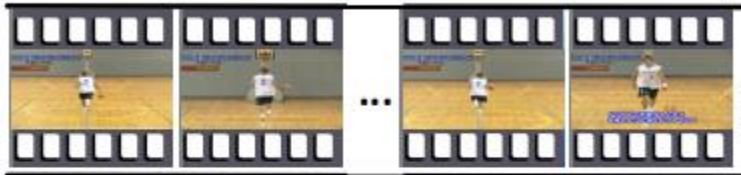
- Verstehen von Abläufen aus mehreren Quellen (z.B. Video+Audio)
  - Zeitliche Folge wird berücksichtig
  - Inputs werden fusioniert



[Xu; et.al.: Learning Multimodal Attention LSTM Networks for Video Captioning]

LSTM for Video Captioning

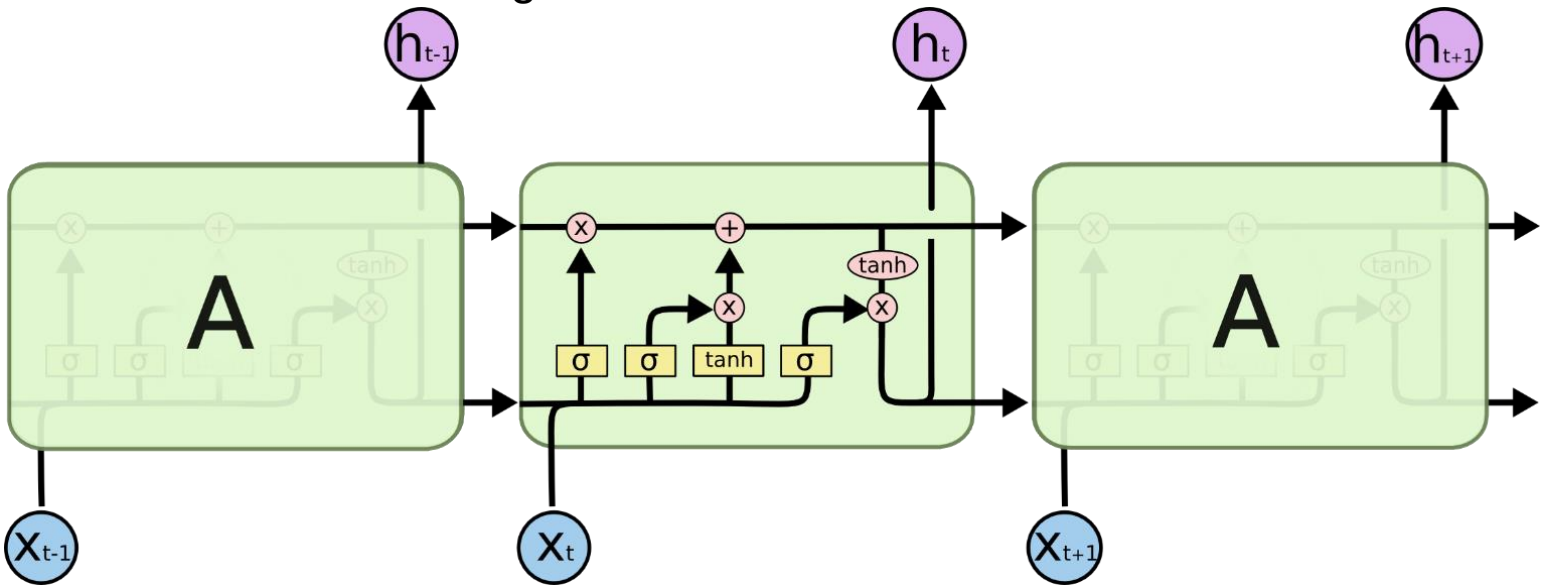
- Schätzungen von zeitlichen Vorgängen
  - Training mit vom Menschen annotierten Daten

	**MA-LSTM(G+C)(child-sum)**: a cat is eating	**Ground Truth**: 1. kitten is eating food 2. a cat is eating from a bowl 3. The animals are eating
	**MA-LSTM(G+C)(child-sum)**: a boy is running and playing basketball	**Ground Truth**: 1. a man runs while dribbling a basketball 2. a man slowly dribbles towards basket 3. a man is running and dribbling a basketball
	**MA-LSTM(G+C)(child-sum)**: a man is sitting and playing guitar	**Ground Truth**: 1. a man is playing the guitar on a park bench 2. a man is playing the guitar seated on a bench in an outdoor location 3. a man is sitting on a bench playing a guitar
	**MA-LSTM(G+C+A)(child-sum)**: a man is explaining cooking	**Ground Truth**: 1. a man is explaining about the preparation of naan 2. a man demonstrates how to make a good fish 3. chef explains how to make a meal
	**MA-LSTM(G+C+A)(child-sum)**: Men and women are dancing with music	**Ground Truth**: 1. a bunch of people dancing 2. a group of people are all dancing in a room 3. dancers dance to the beat of a love song

[Xu; et.al.: Learning Multimodal Attention LSTM Networks for Video Captioning]

Long Short Term Memory networks

- Fazit:
  - RNN sind mächtige Tools, haben aber Probleme
  - LSTM haben weniger Probleme
    - Sie implementieren etwas wie „Aufmerksamkeit“
    - Welche Dinge einer Sequenz sind wichtig für den Output?
    - Der Gradient fließt gut.



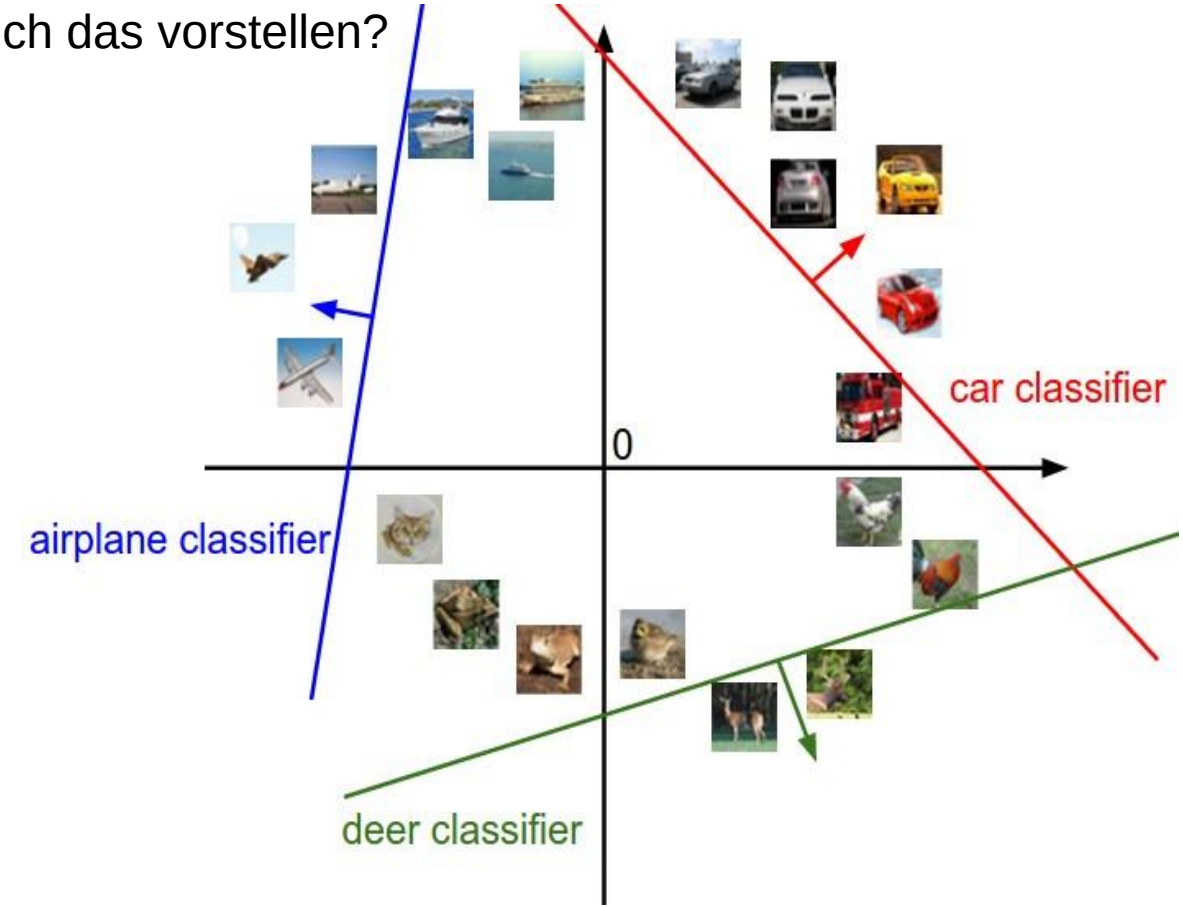
[Chris Olah, Google Brain, Blog post]

Fragen?

?

Embeddings & Feature Space

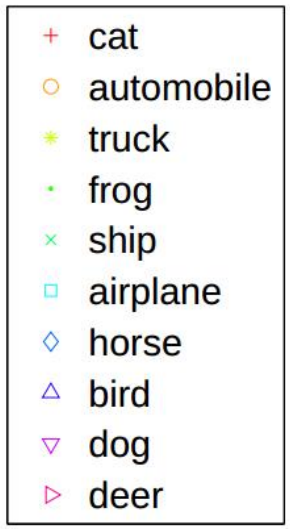
- Neuronale Netze transformieren den Feature Raum
- Wie kann man sich das vorstellen?

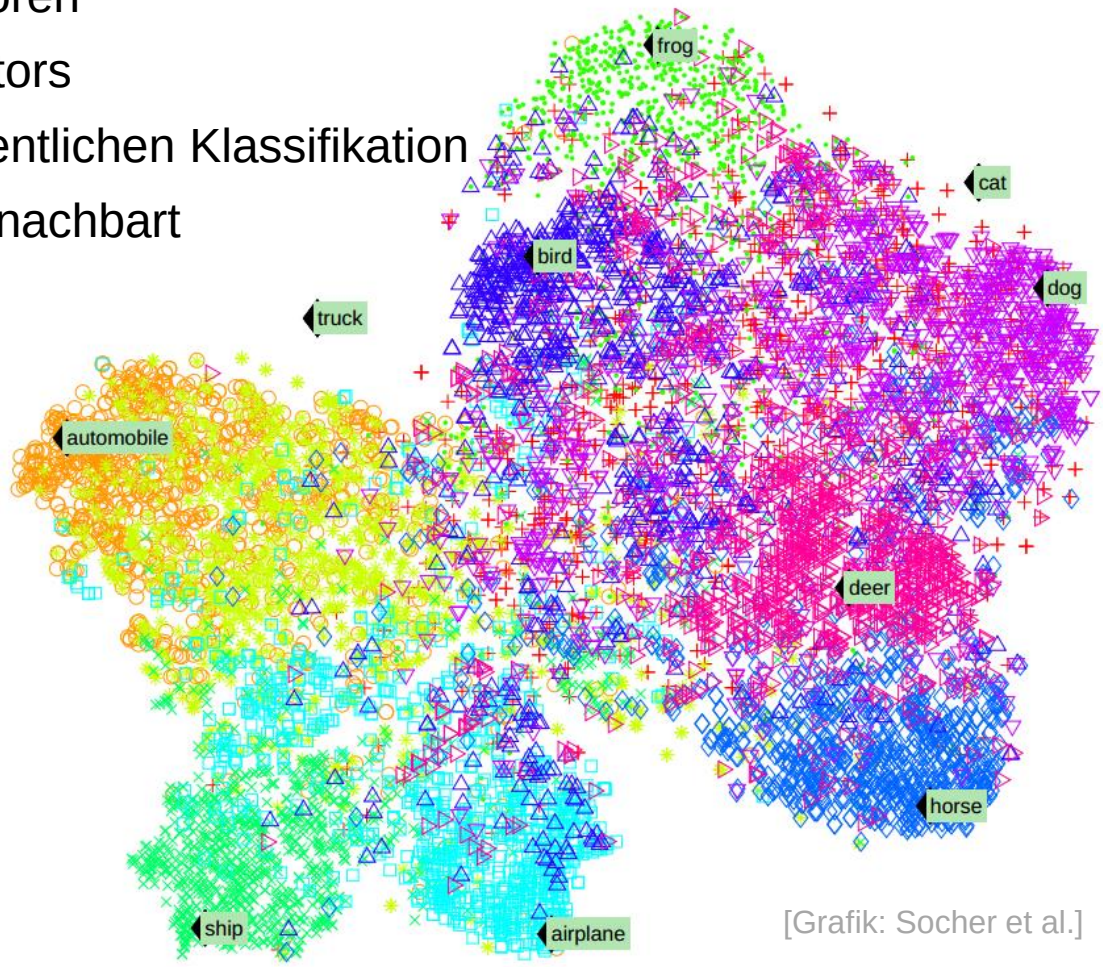


[Andrej Karpathy, Stanford Universitiy]

Embeddings & Feature Space

- Feature Space (CIFAR-10)
- Hoch Dimensionale Vektoren
- Am Ende Ihres Klassifikators
  - Letzter Layer vor der eigentlichen Klassifikation
- Ähnliche Klassen sind benachbart

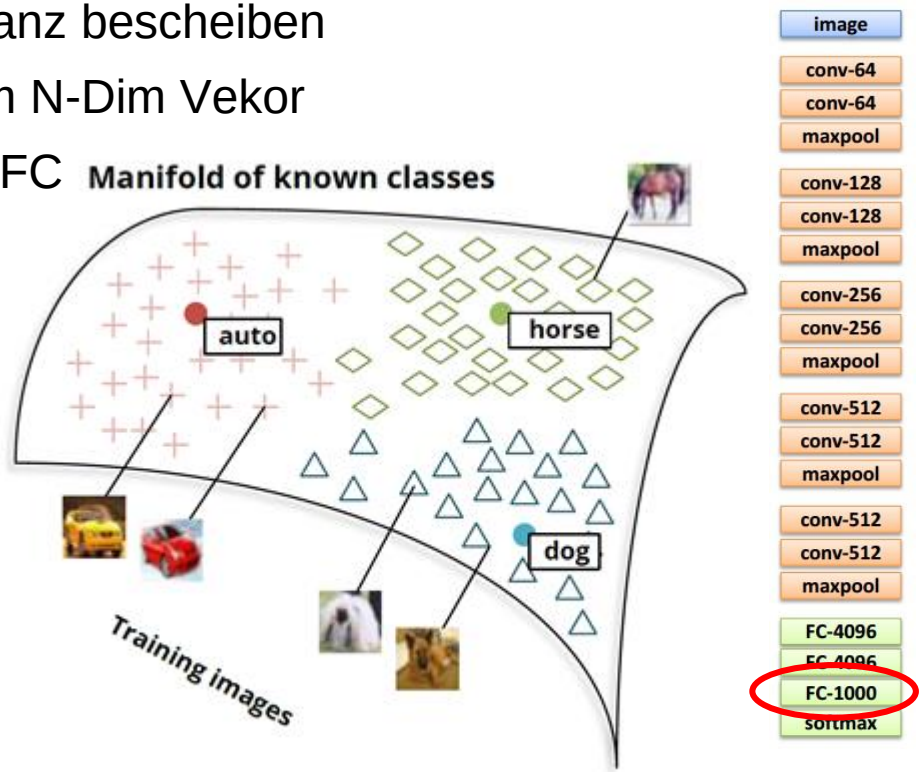




[Grafik: Socher et al.]

Embeddings & Feature Space

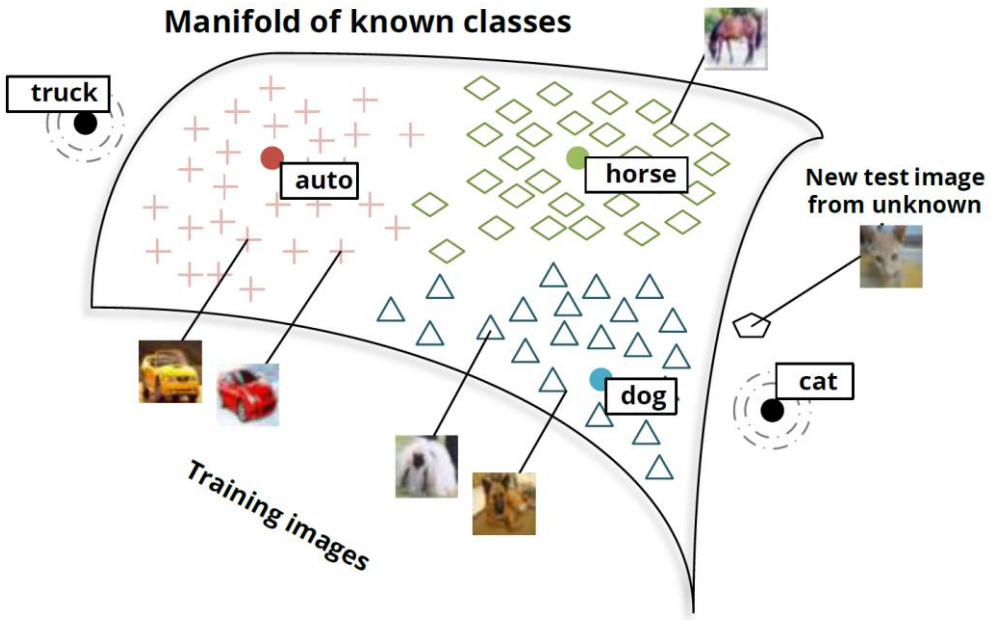
- Repressentanten einer Klasse liegen benachbart
- Ähnliche Klassen liegen benachbart
- Nachbarschaft lässt sich mit Distanz bescheiben
- Jede Aktivierung entspricht einem N-Dim Vekor
  - Mit N = Anzahl der Neuronen im FC



[Grafik: Socher et al.]

Embeddings & Feature Space

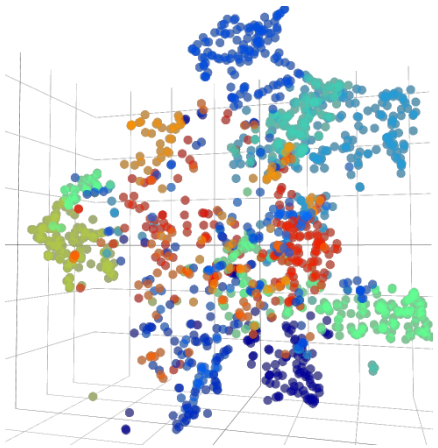
- Es werden unbekannte Klassen eingeben:

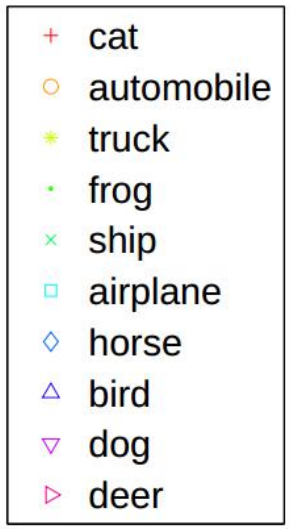


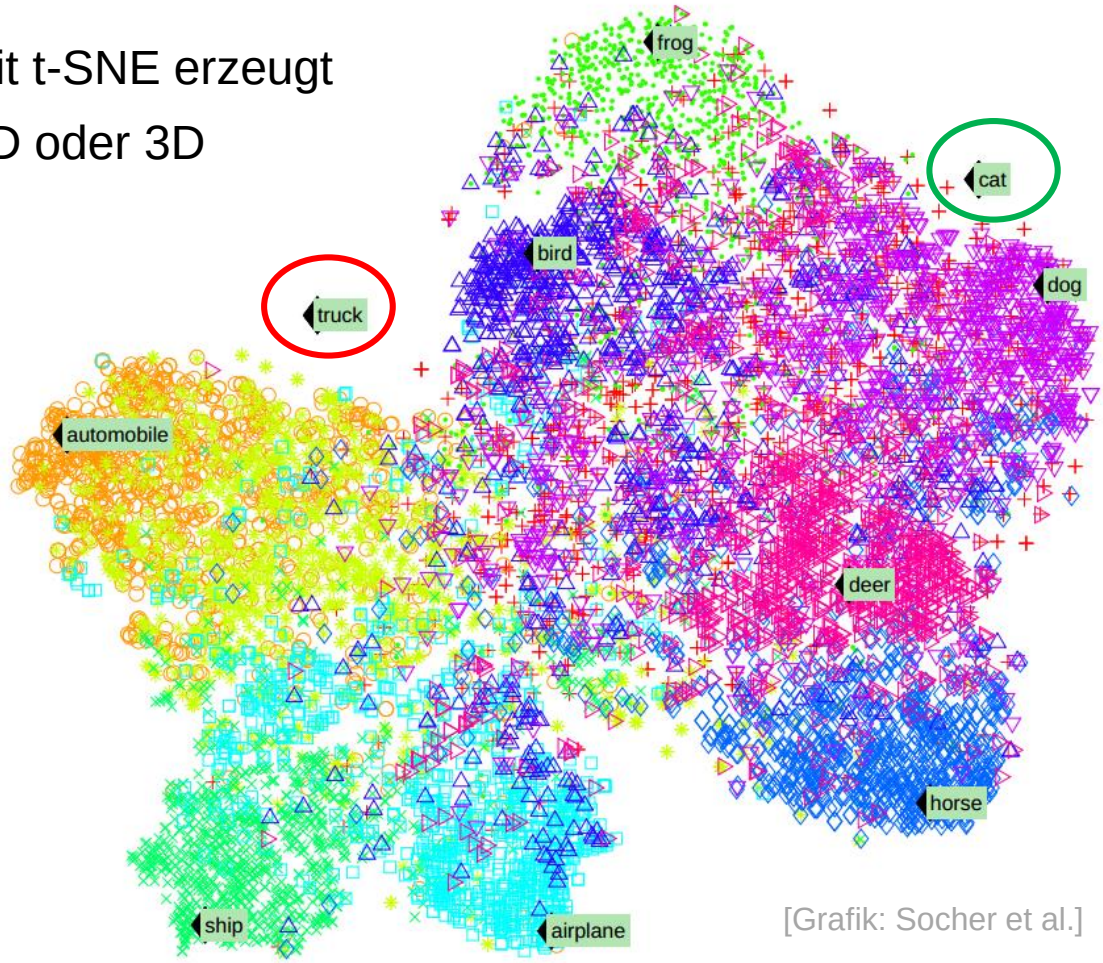
[Grafik: Socher et al.]

Embeddings & Feature Space

- Funktioniert tatsächlich!
- Die Darstellung hier ist mit t-SNE erzeugt
  - Projektion von N-D auf 2D oder 3D

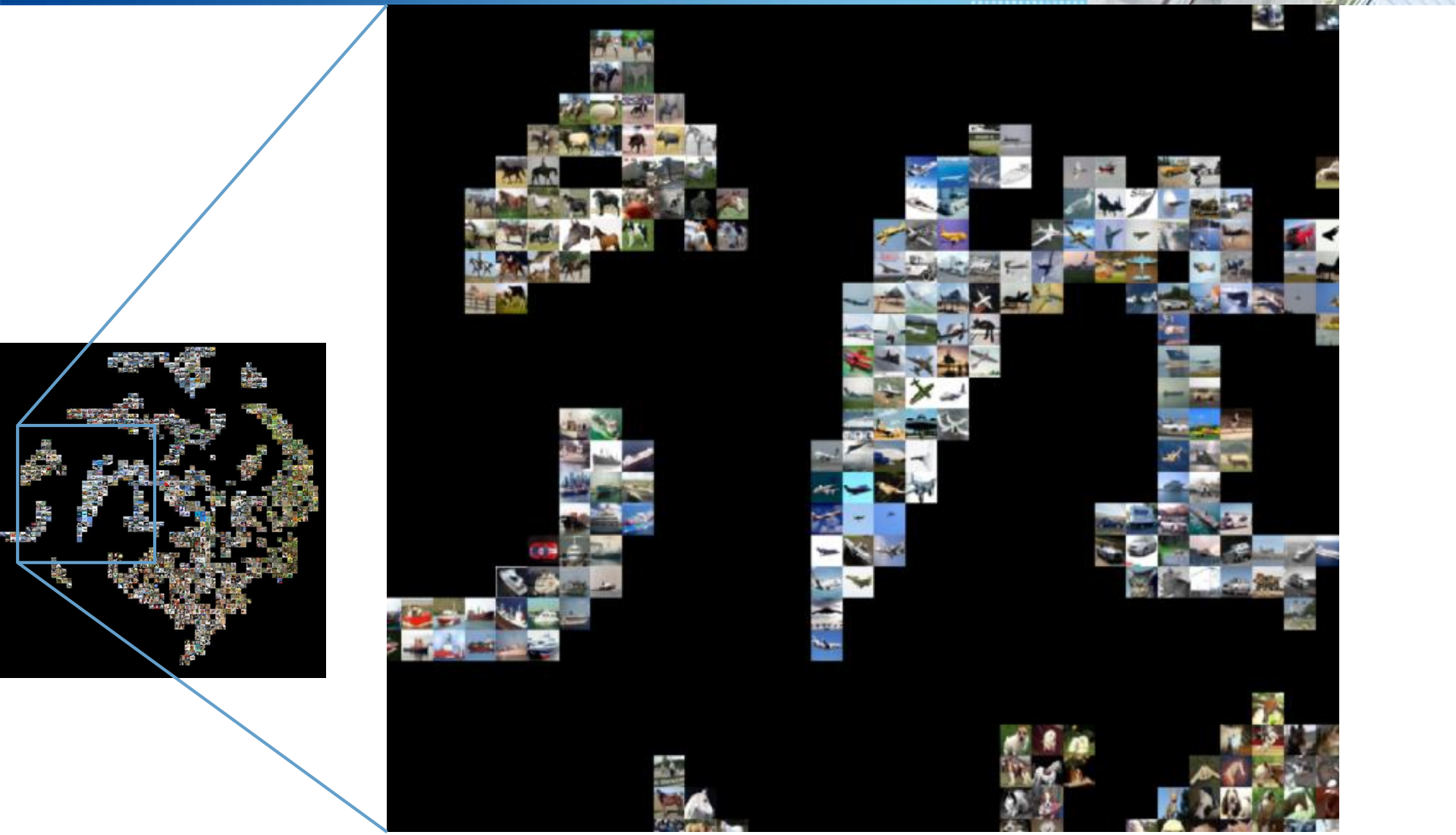






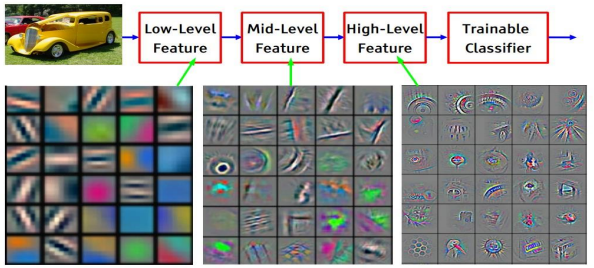
[Grafik: Socher et al.]

Embeddings & Feature Space



Embeddings & Feature Space

- Wie kann man das Interpretieren?



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Penultimate Layer

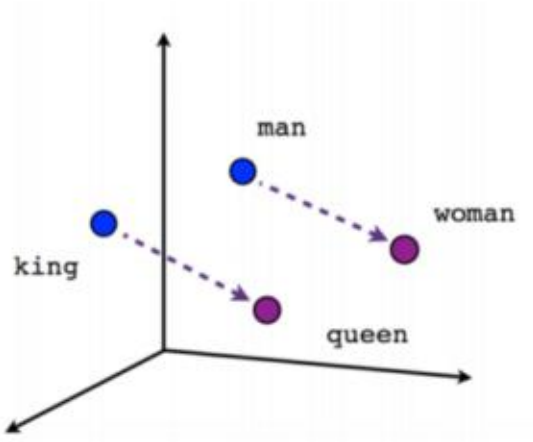
Dim1: Four legs?
Dim2: Straps?
Dim3: Brown & furry?
Dim4: Human leg?
Dim5: Standing in grass?
Dim6: Person holding it?
Dim7: Has laces?
...
Dim4096: In this sky?

Output Layer

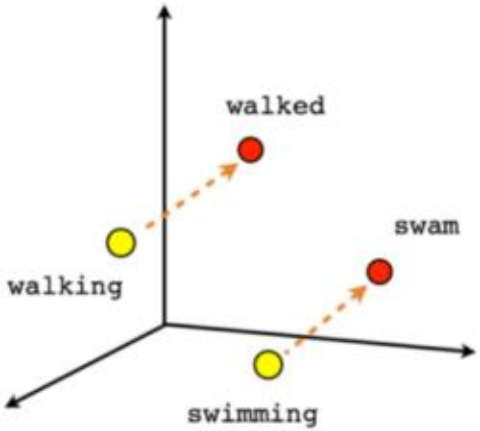
Dim1: Is this an aardvark?
Dim2: Is this an airplane?
Dim3: Is this an apple?
...
Dim 258: Is this a dress shoe?
...
Dim721: Is this a sandal?
...
Dim 1000: Is this a zebra?

Embeddings & Feature Space

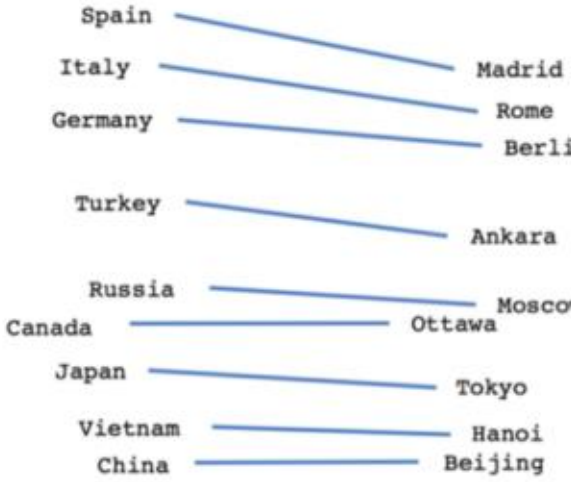
- Was bringt uns diese Ansicht?
  - Feature Space Interpolation



Male-Female



Verb tense

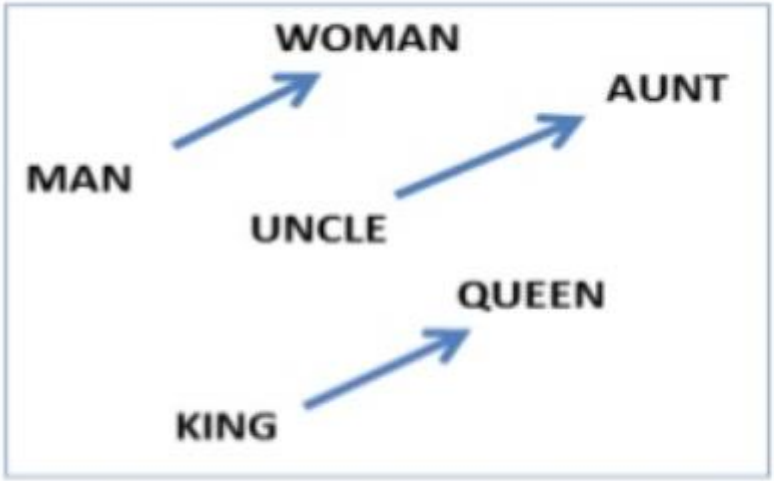


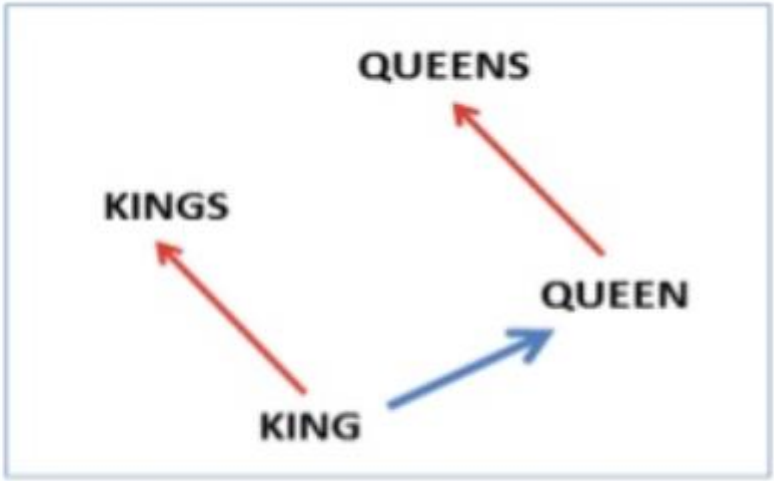
Country-Capital

[Grafik: Rutger Ruizendaal]

Embeddings & Feature Space

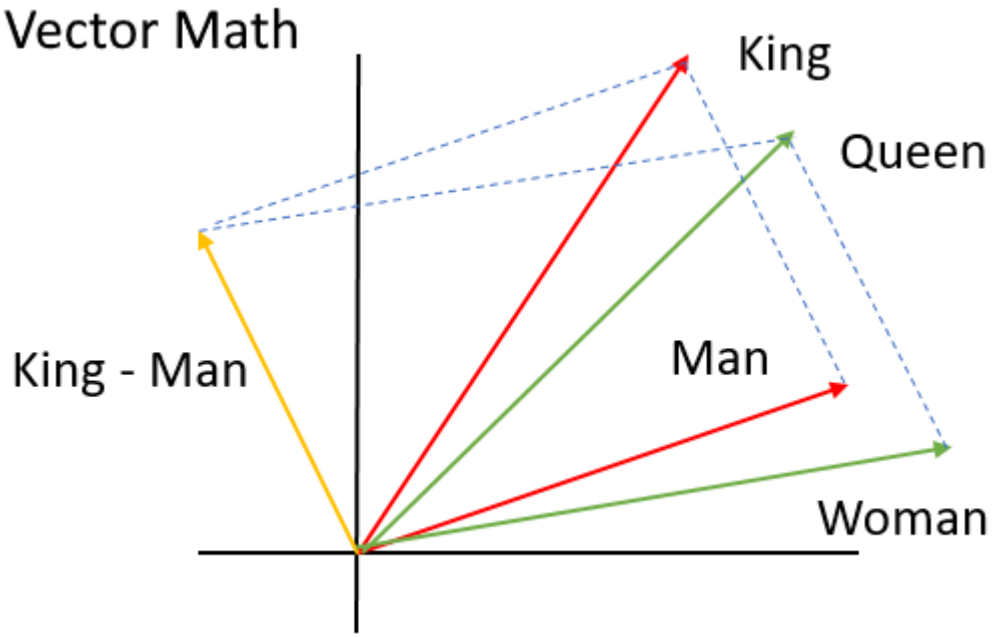
- Was bringt uns diese Ansicht?
  - Feature Space Interpolation





Embeddings & Feature Space

- Was bringt uns diese Ansicht?
  - Feature Space Interpolation



[Grafik: Mathworks]

Embeddings & Feature Space

- Word Embeddings

Expression	*Nearest token*
Paris - France + Italy	Rome
bigger - big + cold	colder
sushi - Japan + Germany	bratwurst
Cu - copper + gold	Au
Windows - Microsoft + Google	Android
Montreal Canadiens - Montreal + Toronto	Toronto Maple Leafs

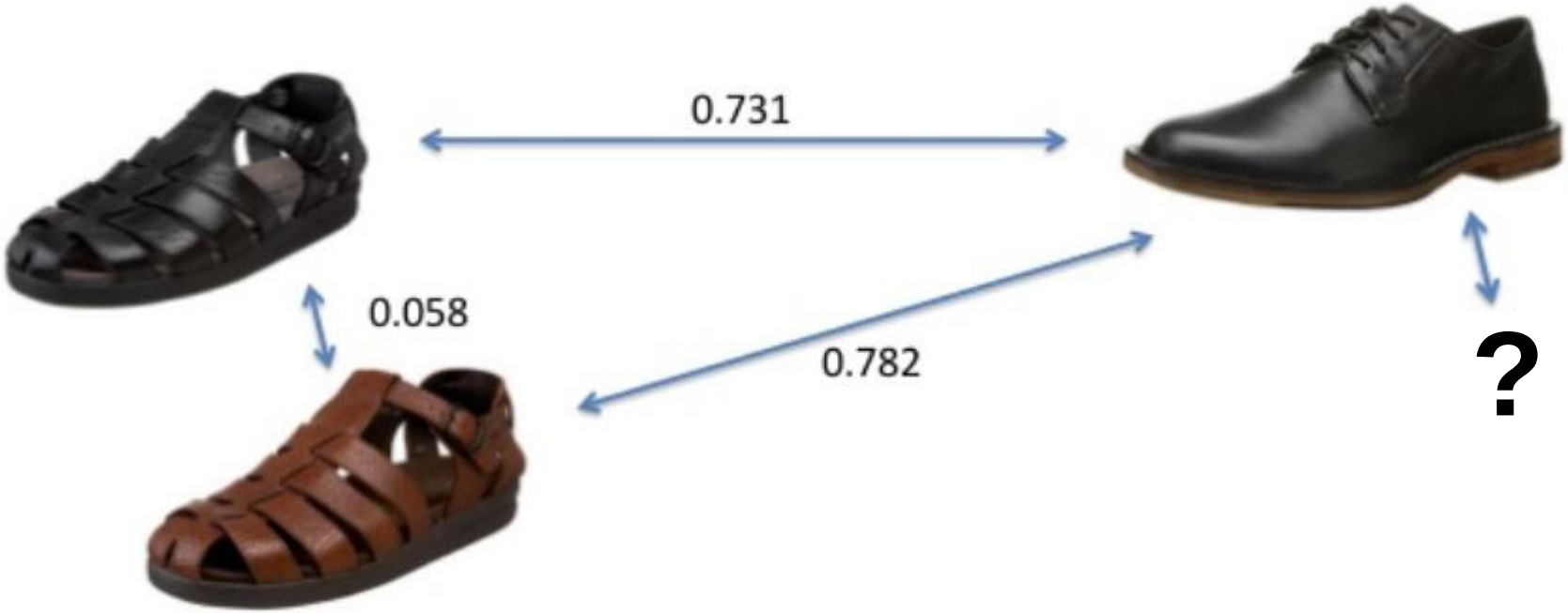
Embeddings & Feature Space

- Was bringt uns diese Ansicht?
  - Feature Space Interpolation

$$\text{W2v}(\text{“king”}) - \text{W2v}(\text{“queen”}) + \text{W2v}(\text{“aunt”}) = \text{W2v}(\text{“uncle”}) \approx \begin{bmatrix} 5.409 \\ 5.281 \\ -1.331 \\ 3.714 \\ -1.727 \\ -3.167 \\ -2.130 \\ 1.213 \\ -3.285 \\ \dots \\ -2.000 \end{bmatrix} \in \mathbb{R}^{128}$$

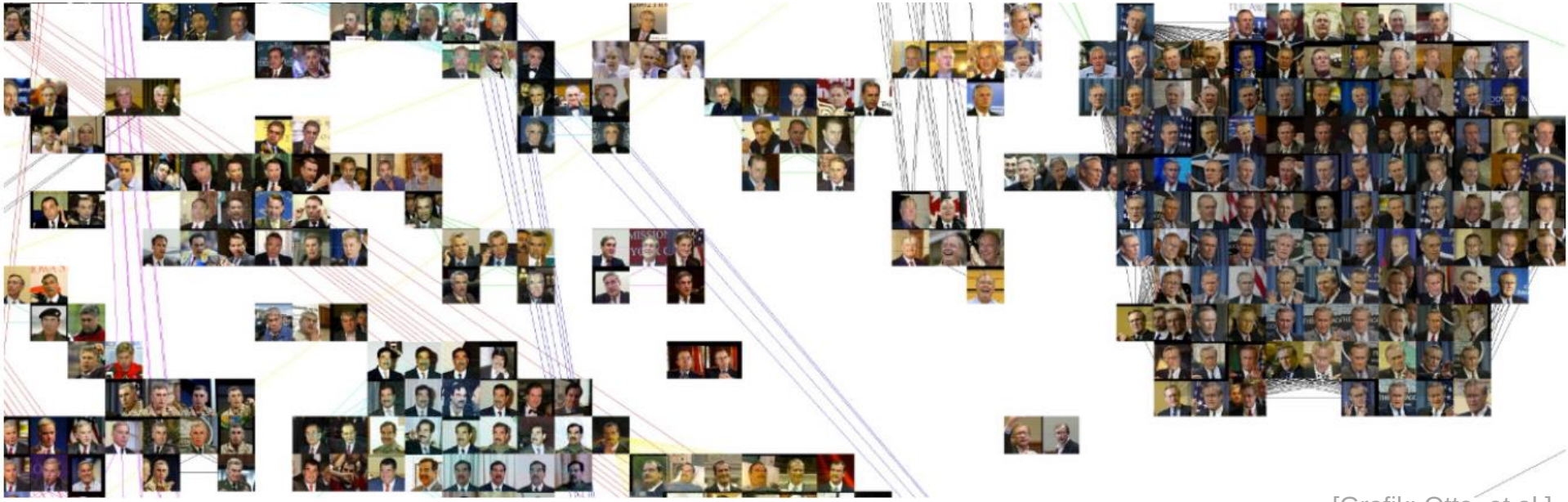
Embeddings & Feature Space

- Kann man das auch auf Bilder anwenden?

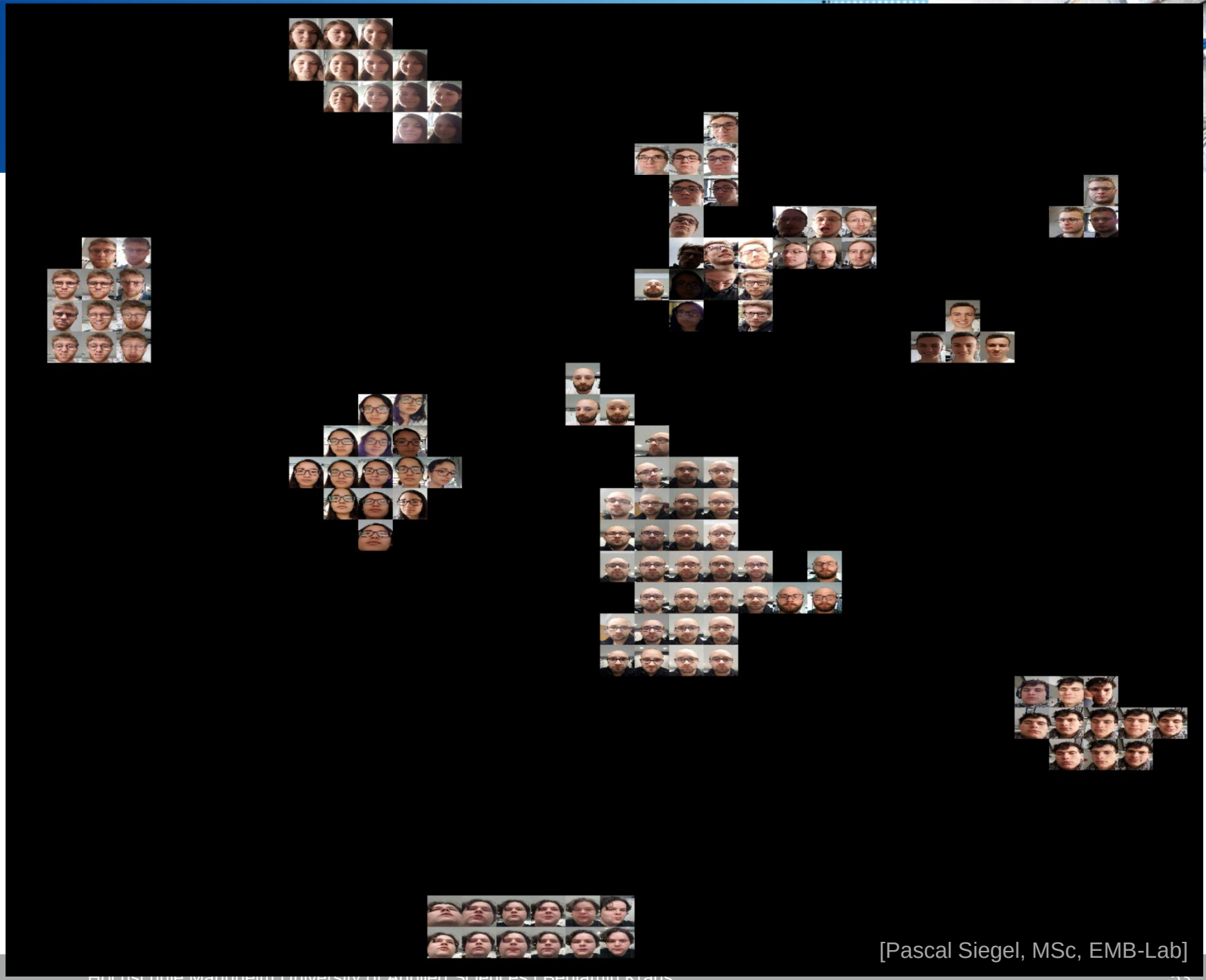


Embeddings & Feature Space

- Was bringt uns diese Ansicht?
  - Feature Space Vergleiche:
  - Netz auf Portrait-Fotos trainiert (Klassifikation)
  - Zwei Bilder von unbekannten Personen eingeben



[Grafik: Otto, et al.]



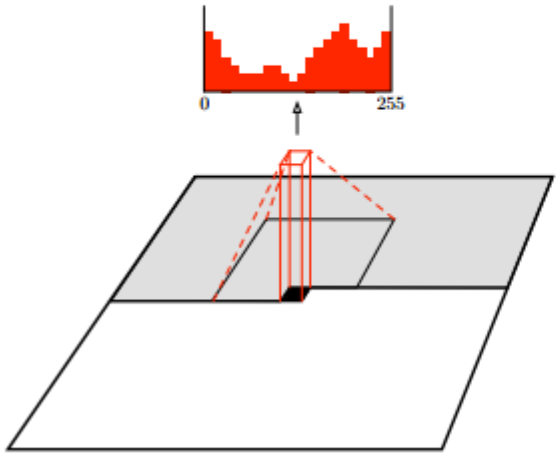
[Pascal Siegel, MSc, EMB-Lab]

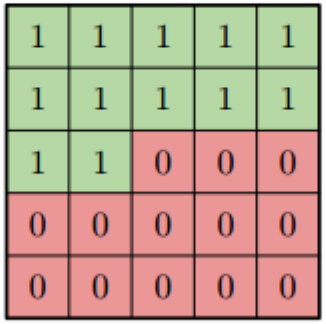
[Pascal Siegel, MSc, EMB-Lab]

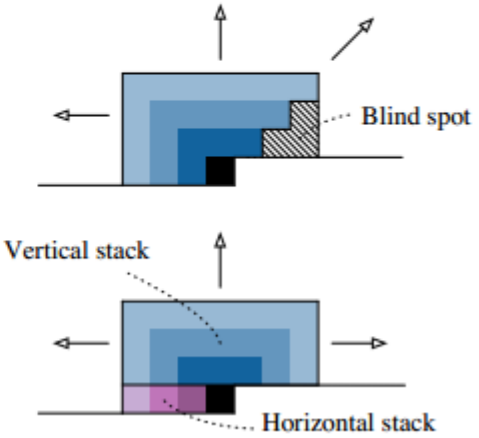
Embeddings & Feature Space

- PixelCNN zum Generieren von Bildern
- Training auf Portrait -> Embeddings
  - [Conditional Image Generation with PixelCNN Decoders, Oord et. al]

$$p(\mathbf{x}) = \prod_{i=1}^{n^2} p(x_i | x_1, ..., x_{i-1})$$







[Grafik: Oord et al.]

Embeddings & Feature Space

- PixelCNN zum Generieren von Bildern
- Conditioning on Portrait Embeddings

Figure 5: Linear interpolations in the embedding space decoded by the PixelCNN. Embeddings from leftmost and rightmost images are used for endpoints of the interpolation.

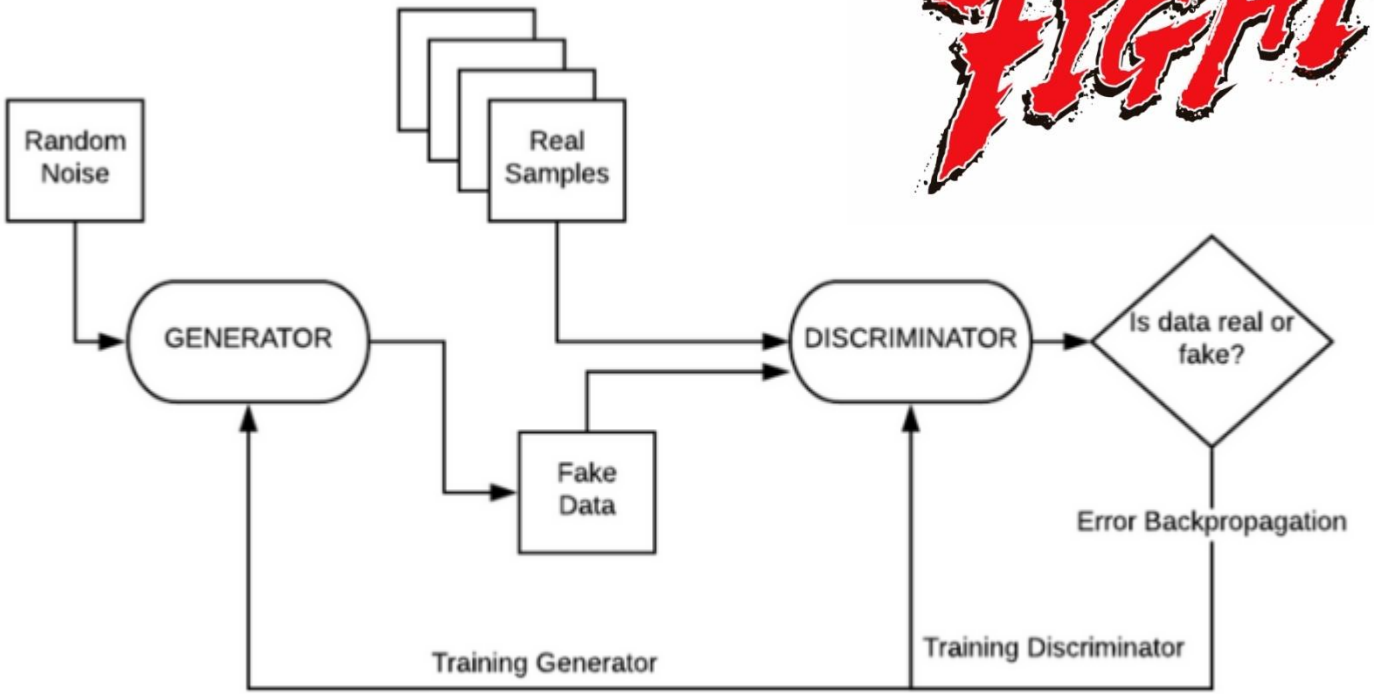
[Grafik: Oord et al.]

Embeddings

- Pix2Pix
- GAN
- Generative Adversarial Network

Generative Adversarial Network

- Trainiere G um „gefälschte“ Daten zu erzeugen
- Trainiere D so, dass es reale Daten von Fakes unterscheiden kann



Generative Adversarial Network

Discriminator outputs likelihood in (0,1) of real image

Minimax objective function:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log \underbrace{D_{\theta_d}(x)}_{\text{Discriminator output for real data x}} + \mathbb{E}_{z \sim p(z)} \log(1 - \underbrace{D_{\theta_d}(G_{\theta_g}(z))}_{\text{Discriminator output for generated fake data G(z)}}) \right]$$

- Discriminator (θ_d) wants to **maximize objective** such that D(x) is close to 1 (real) and D(G(z)) is close to 0 (fake)
- Generator (θ_g) wants to **minimize objective** such that D(G(z)) is close to 1 (discriminator is fooled into thinking generated G(z) is real)

Alternate between:

1. **Gradient ascent** on discriminator

$$\max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

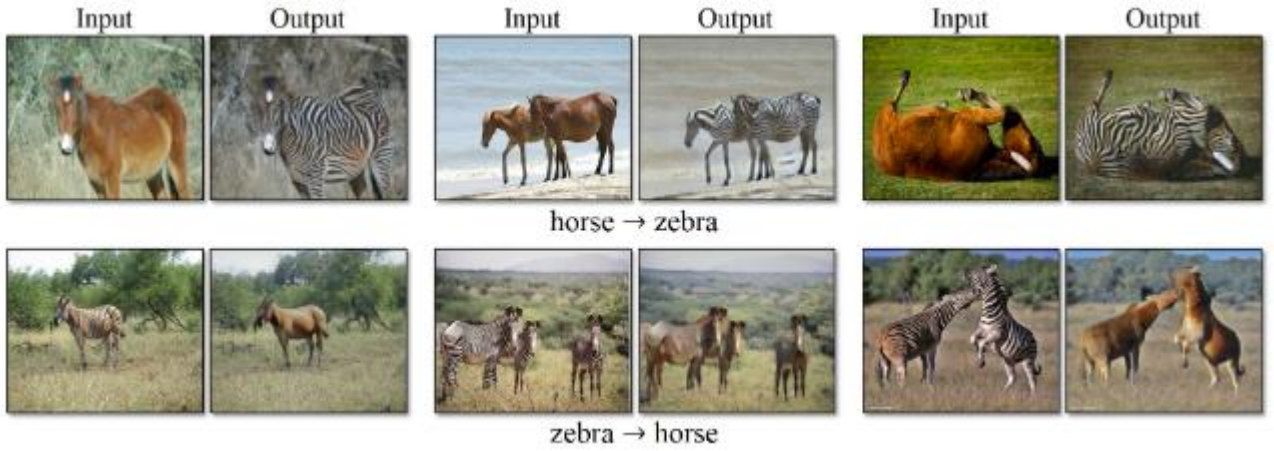
2. **Gradient descent** on generator

$$\min_{\theta_g} \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z)))$$

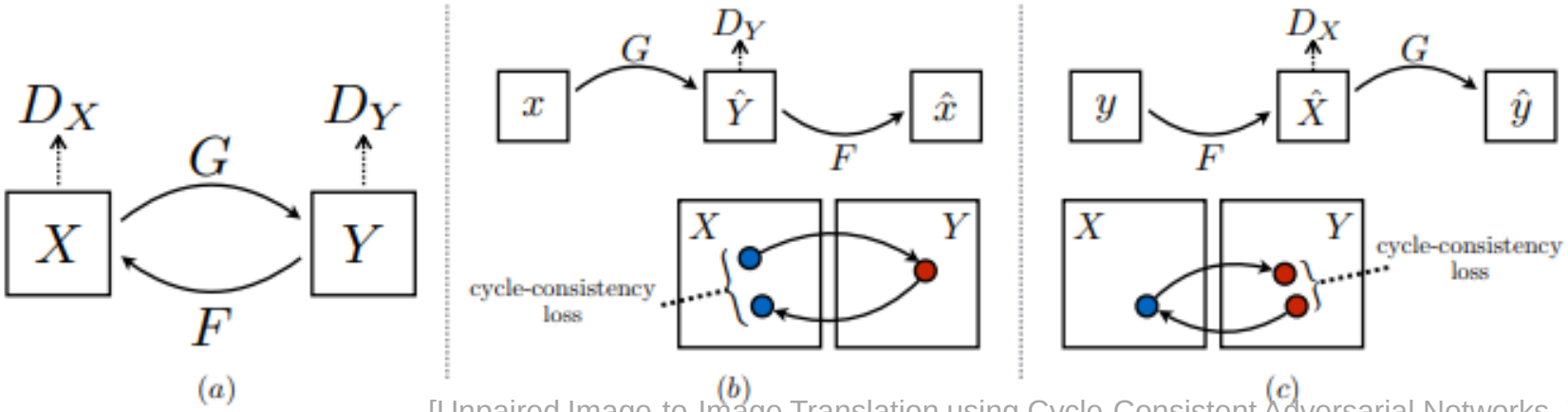
[Standford CS231n]

CycleGAN

- CycleGAN
- Für Daten mit Domain Info (z.B. class label)



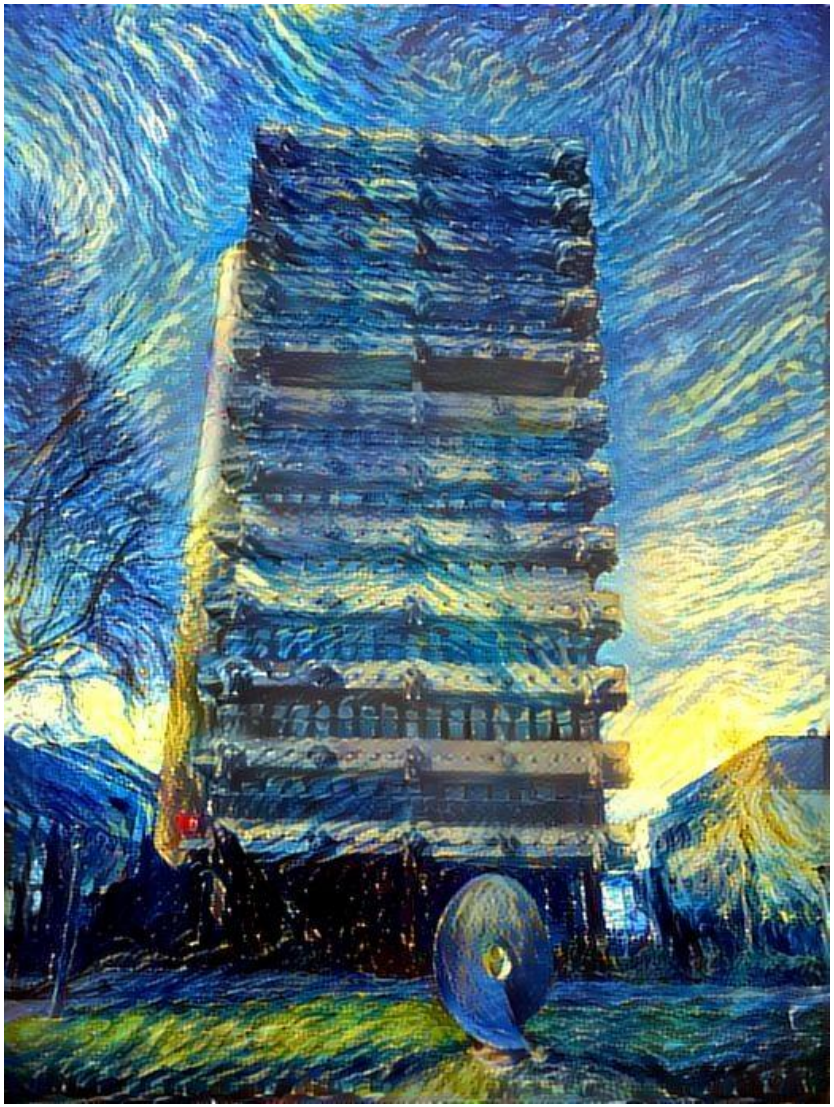
- Transformation in die Ziel-Domain und zurück, Dabei soll möglichst wenig Verlust entstehen.



[Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, Zhu]

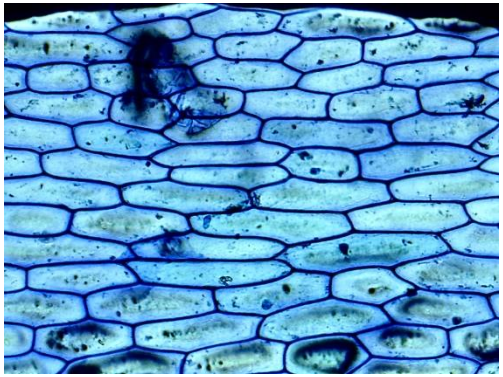
Embeddings & Feature Space

- Neural Style Transfer



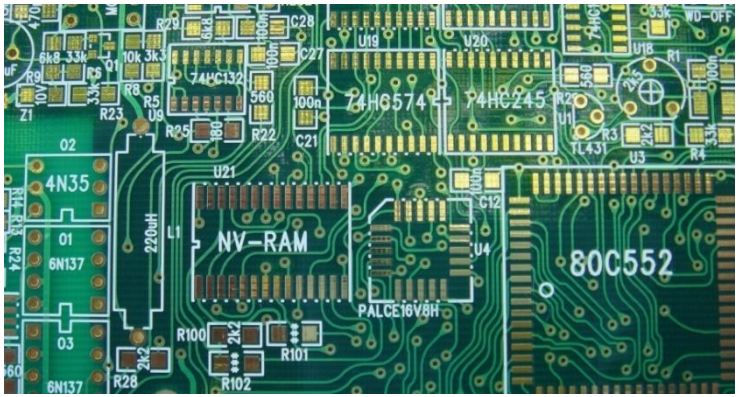
Embeddings & Feature Space

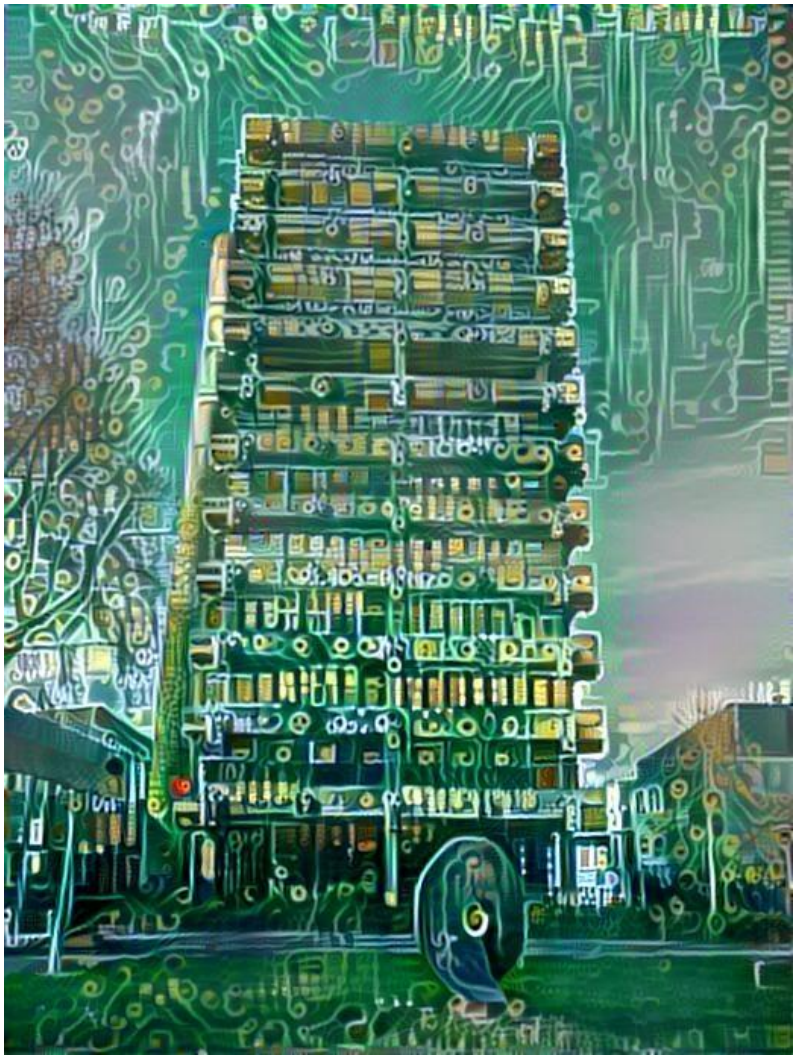
- Neural Style Transfer



Embeddings & Feature Space

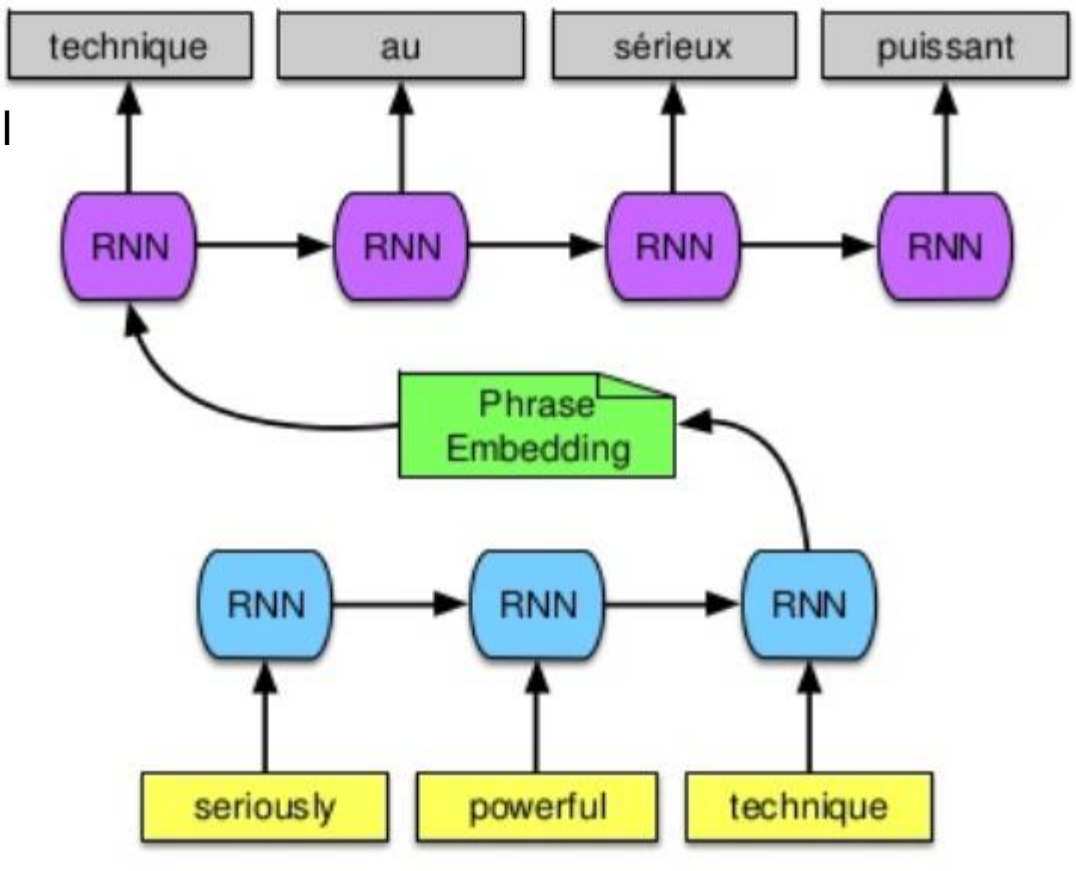
- Neural Style Transfer





Embeddings & Feature Space

- Seq2Seq
- Beispiel Übersetzung:
- Man kann den oberen Teil einfach per Zielsprache neu trainieren



Ausblick

- Wie geht es Weiter?
  - Wie Trainiere ich meine Netze richtig?
    - Überwachung des Trainingsvorgangs
    - Hyperparameter
    - Initialisierungen
    - Versuchsplanung
  - Semester Aufgabe