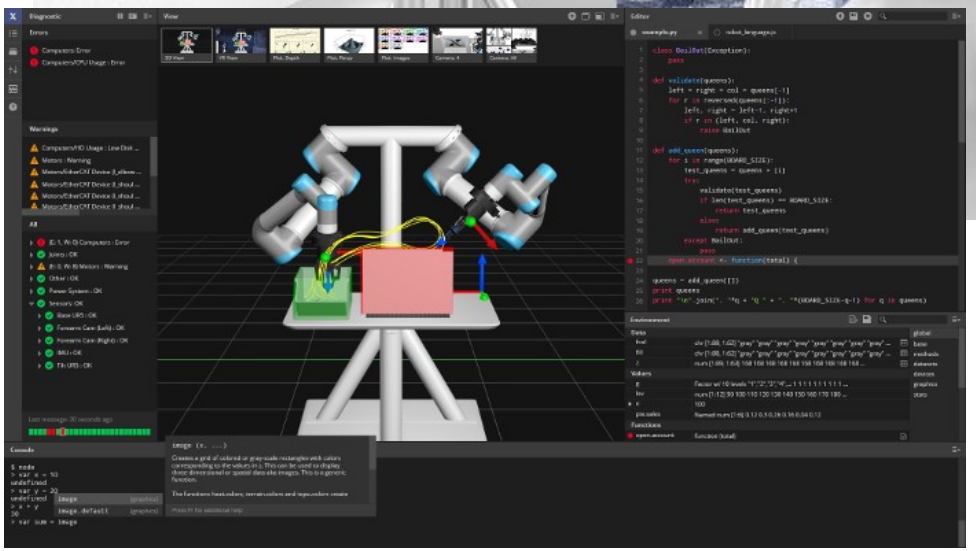
# Deep down the rabbit hole

## Deep Learning mit künstlichen neuronalen Netzen

Andreas Köpf
Xamla Robotics Team
PROVISIO GmbH
andreas.koepf@provisio.com
@neurosp1ke

# We are hiring!

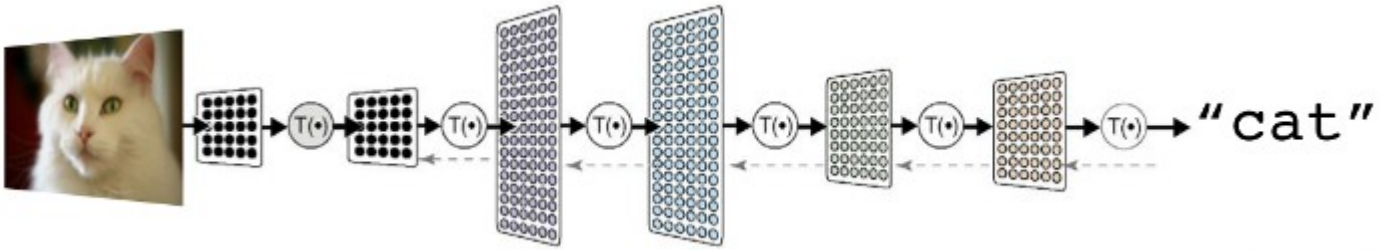



Wir suchen Developer mit Machine Learning, Computer Vision oder Robotik Erfahrung.

Bei Interesse: jobs@xamla.com

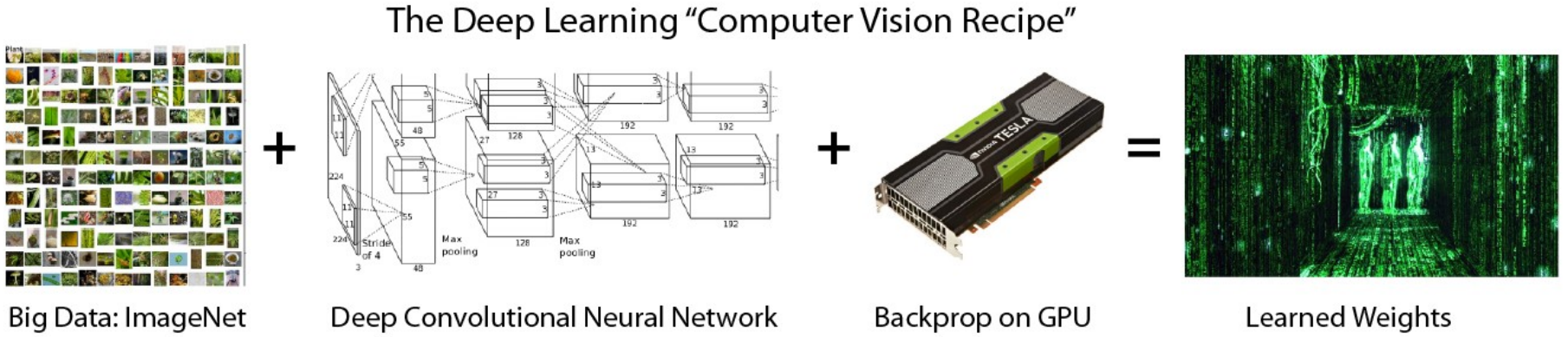
# Überblick

- Was ist Deep Learning?
- Multilayer Perceptron
- ConvNets
- Rekurrente Netze
- Visualisierungen
- Tools & Links



# Was ist Deep Learning?

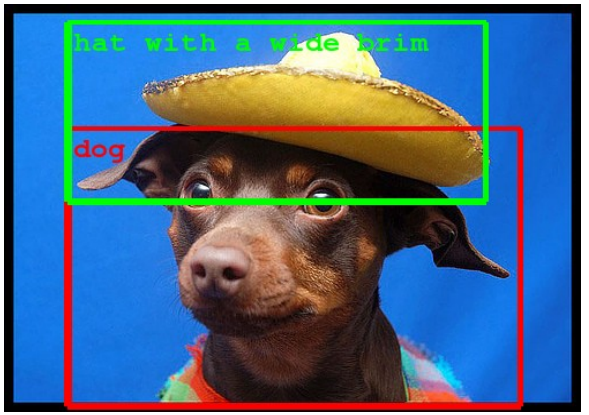
- Schichtenweise aufgebaute maschinelle Lernverfahren, insb. verschiedene Neuronale Netze mit mehr als 3 Schichten (z.B. ConvNet, DBN, RNN)
- Ziel: Automatische Generierung von einfach trennbaren Merkmalen aus hochdimensionalen Eingaben
- Training von großen (> 100 Mio Parameter) und tiefen (>10 Schichten) Netzen

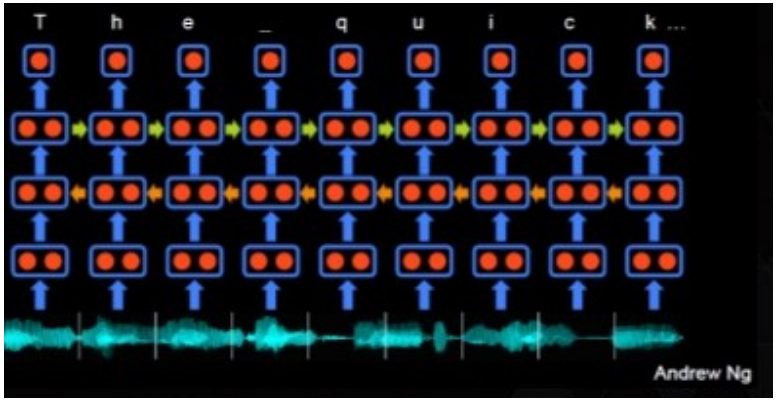


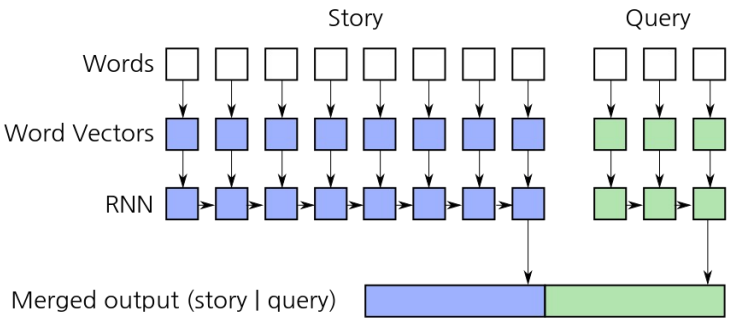

Bildquelle: http://www.computervisionblog.com/2015/05/deep-learning-vs-big-data-who-owns-what.html

# Prominente Einsatzgebiete

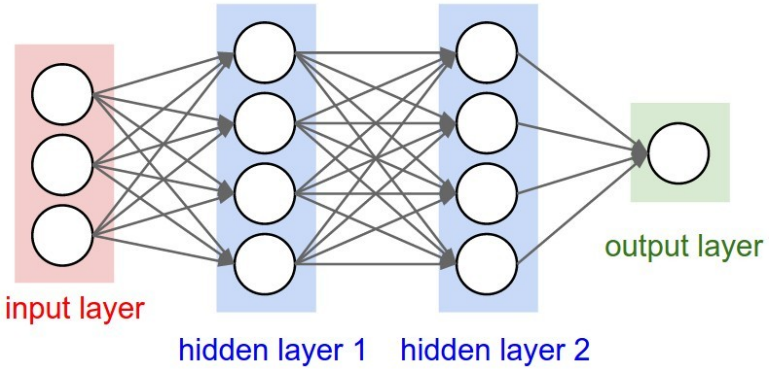
- Maschinelles Sehen (ConvNet)
  - Bildklassifizierung
  - Objekterkennung
- Sequenzverarbeitung (RNN)
  - Akustische Spracherkennung
  - Maschinelle Übersetzung
  - Sprachmodelle und Textverständnis
  - Verarbeitung biologischer Sequenzen (DNA)



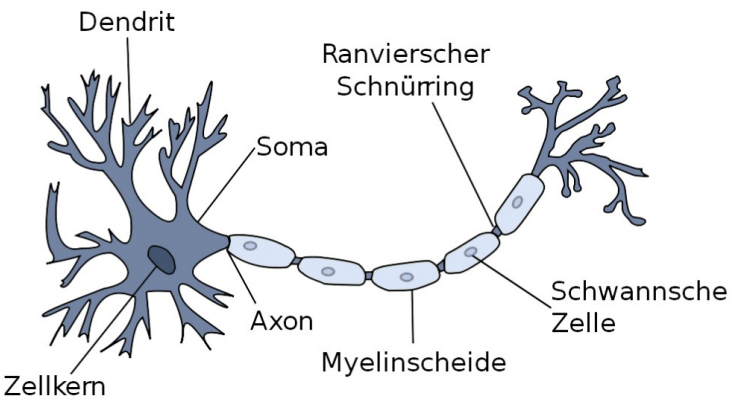



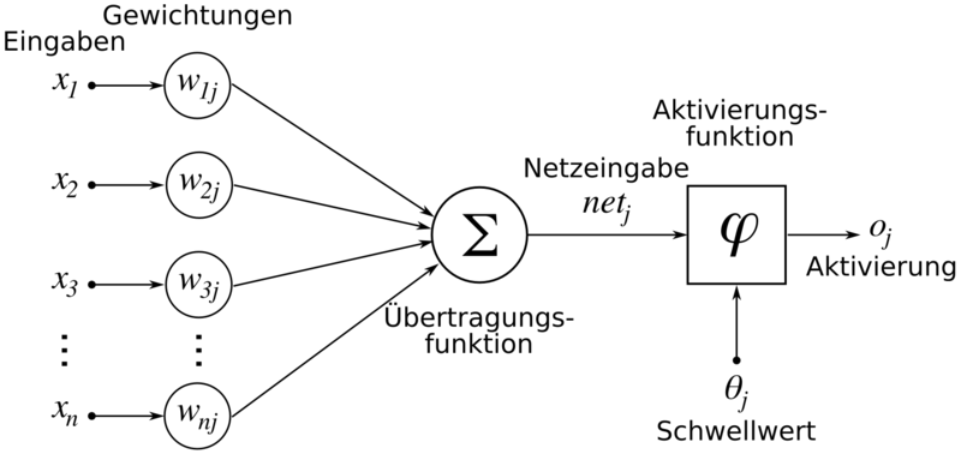



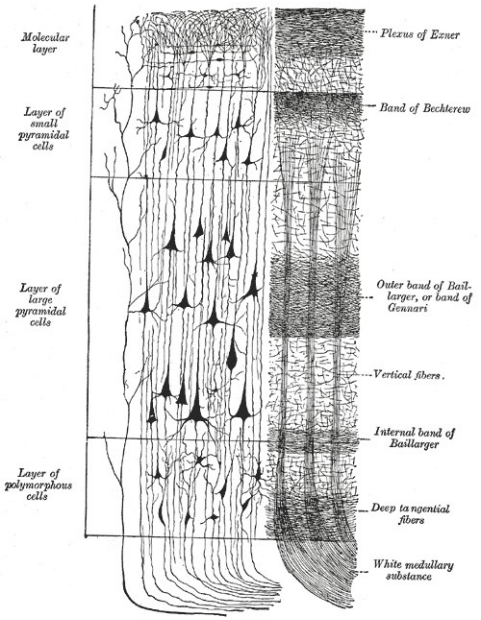
# Multilayer Perceptron

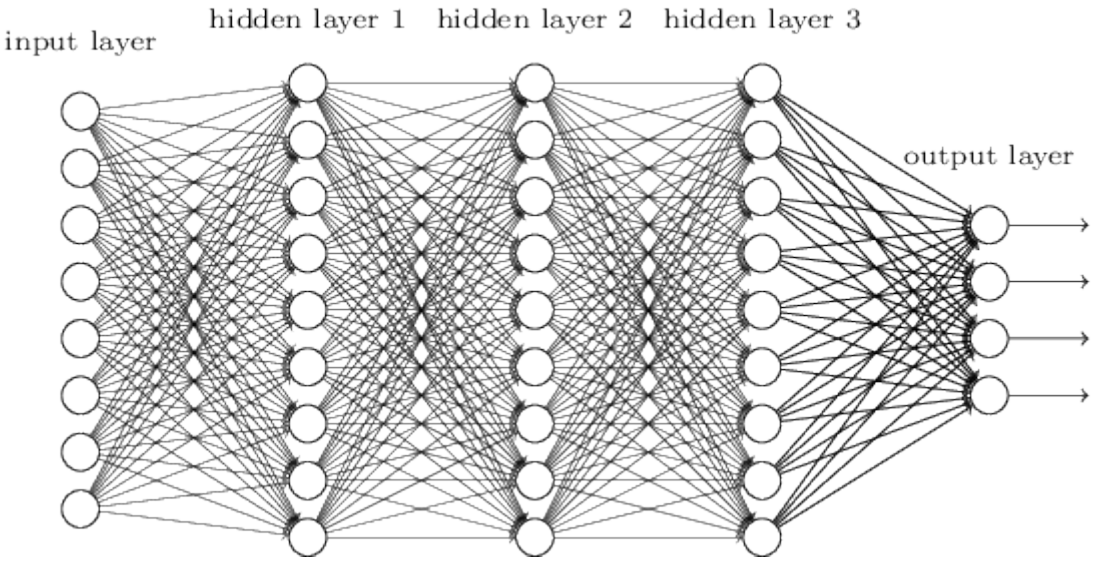


# Biologische vs. künstliche Neuronen



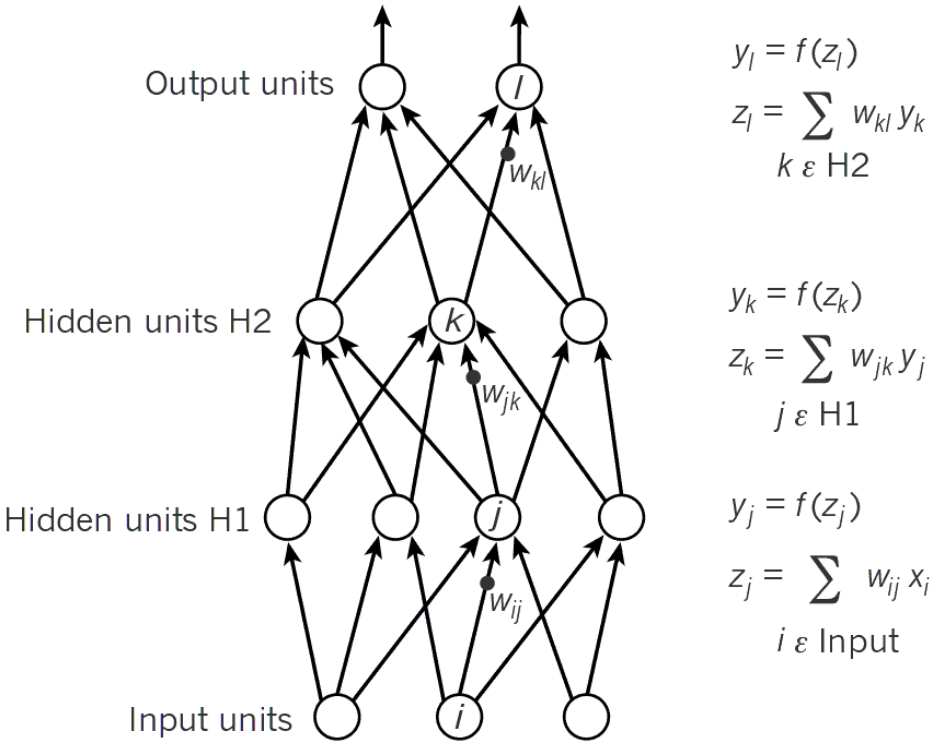



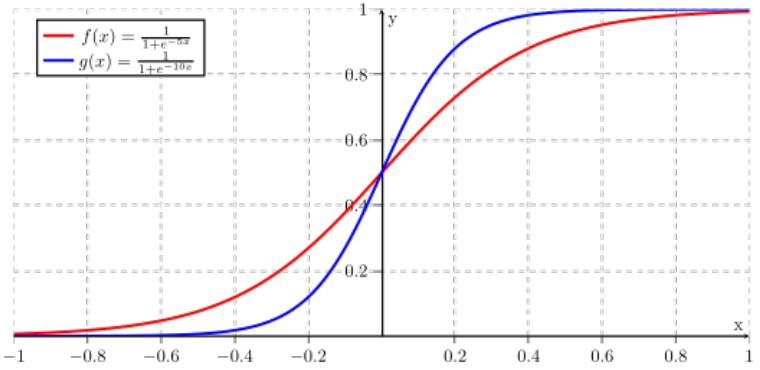

Bildquellen:
David Kriesel, http://www.dkriesel.com/science/neural_networks
Wikipedia von Chrislb, https://de.wikipedia.org/wiki/K%C3%BCnstliches_neuronales_Netz
https://de.wikipedia.org/wiki/Gro%C3%9Fhirnrinde
http://neuralnetworksanddeeplearning.com/chap5.html

# Multilayer Perceptron (MLP)

- **Universeller* Funktionsapproximator**
- Feed-Forward, Schicht zu Schicht vollverknüpft
- Neuron: gewichtete Summe der Vorschicht-Ausgaben + Bias durchläuft Transferfunktion
- Training mit Hilfe des Gradienten der Fehlerfunktion, analytisch mit Kettenregel bestimmbar (Backprop)
- Verschiedene Transferfunktionen möglich, z.B. tanh, Softsign, ReLU, Softmax, Linear, ...

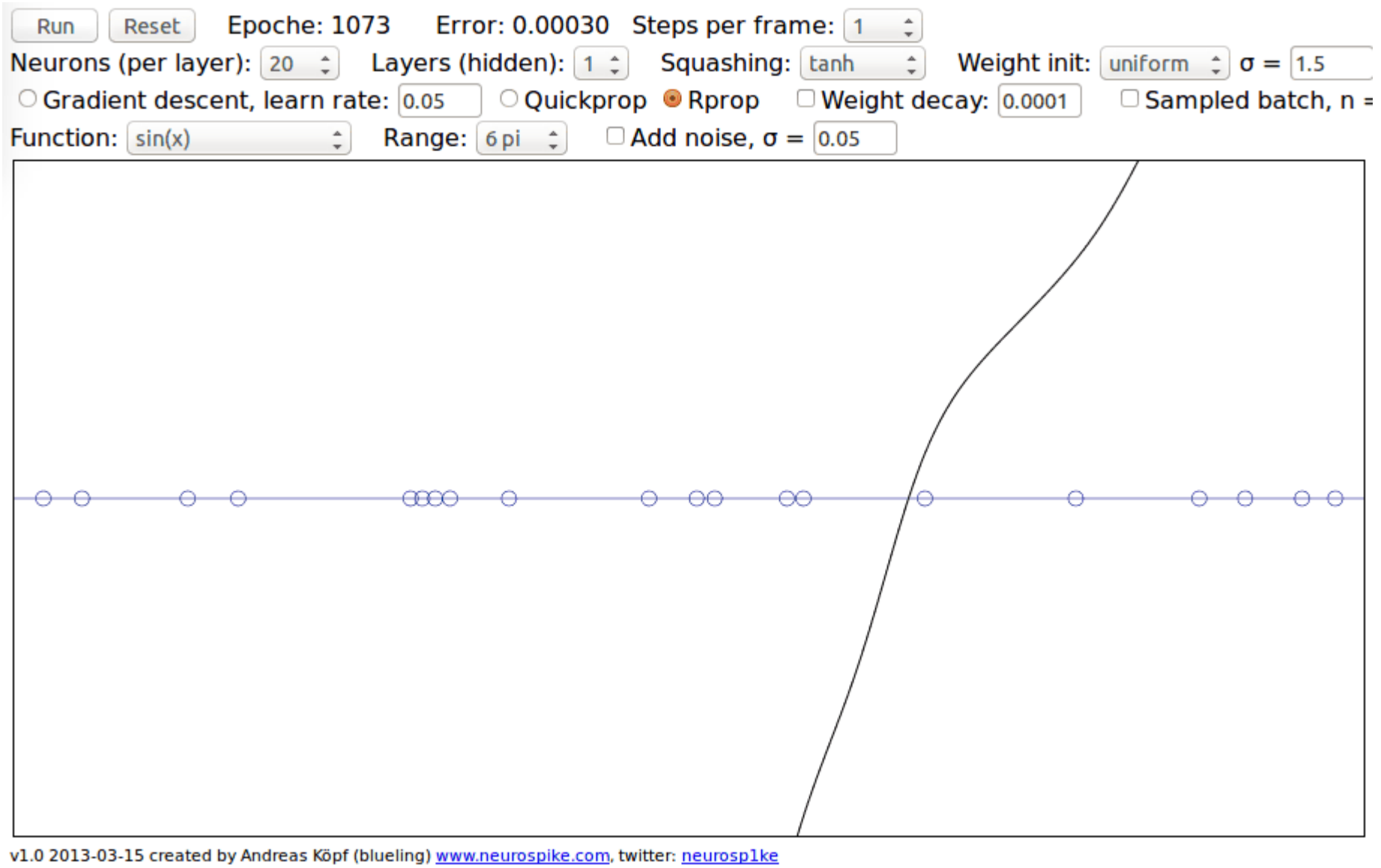



Lesetipp:
“Ein kleiner Überblick über Neuronale Netze” D. Kriesel
http://www.dkriesel.com/science/neural_networks

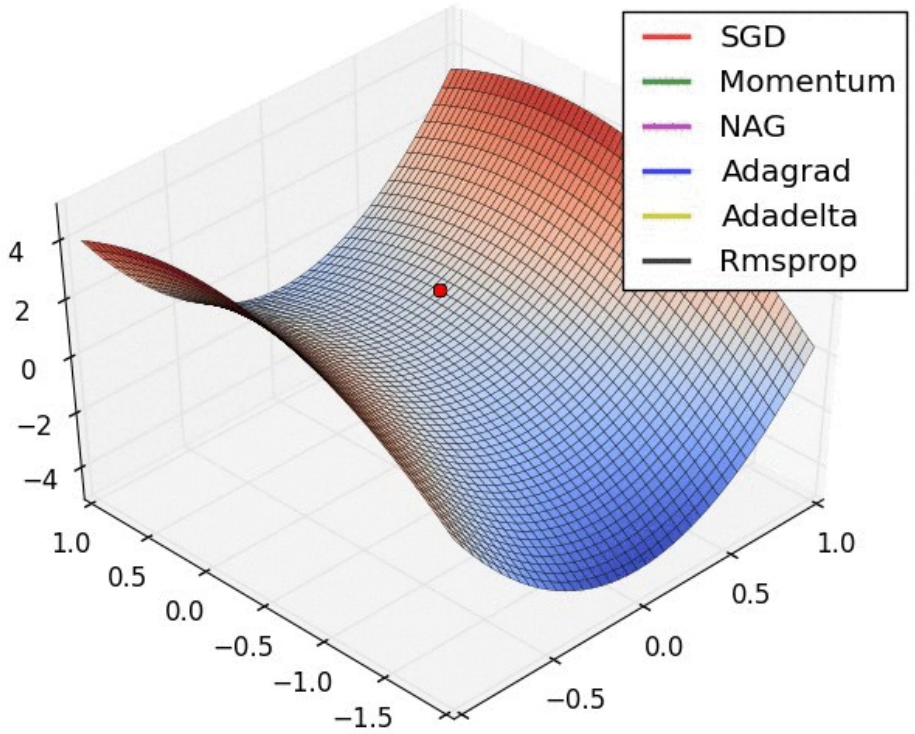


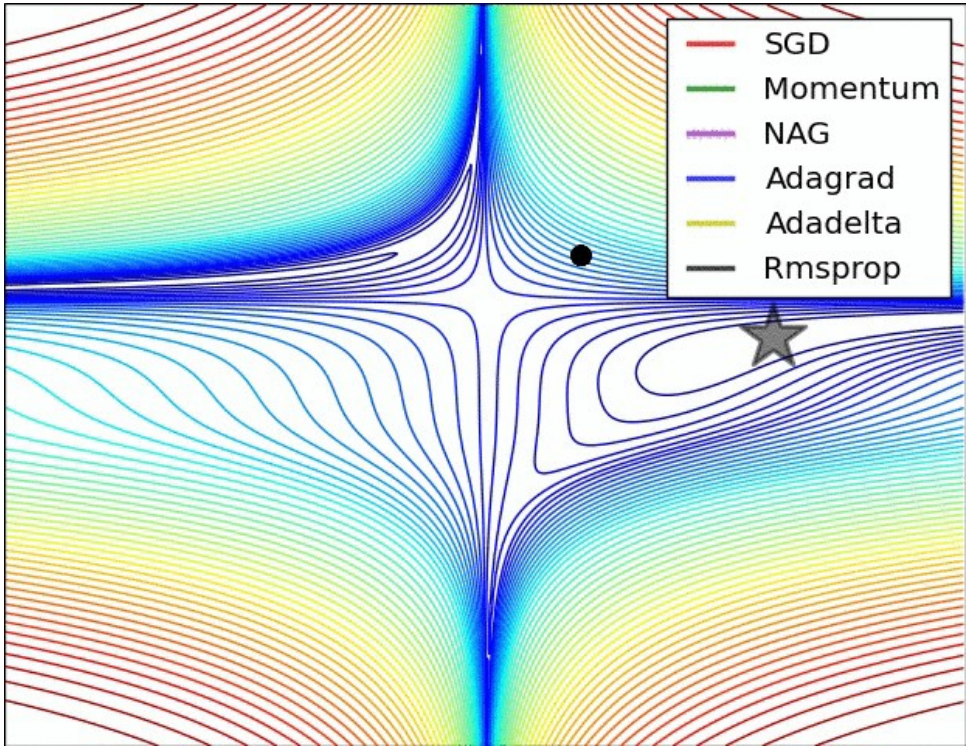

Bildquellen:
MLP forward pass: doi:10.1038/nature14539
Fermi-Funktion: Wikipedia, „Sigmoid-function“ von MartinThoma

# 1D Funktionsapproximation Demo

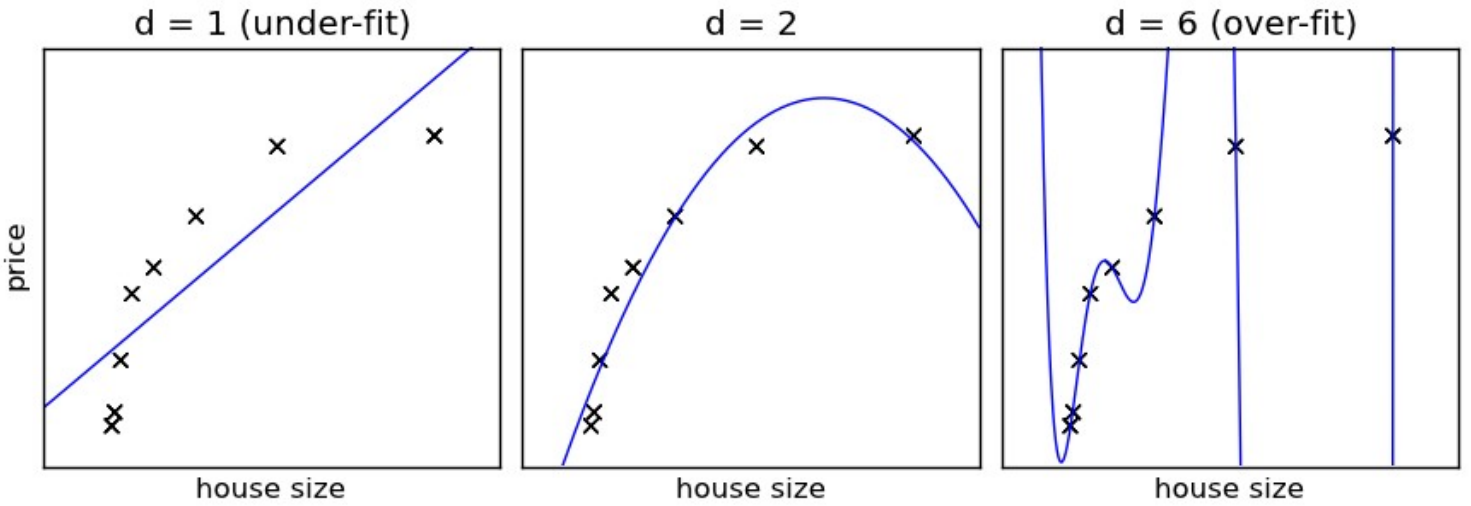


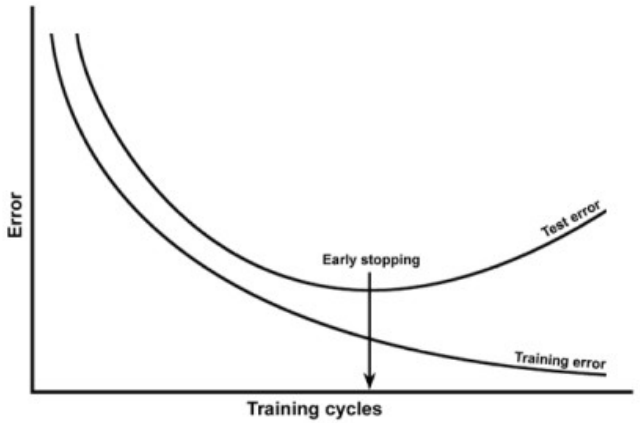
# Optimierung mit Gradientenabstieg

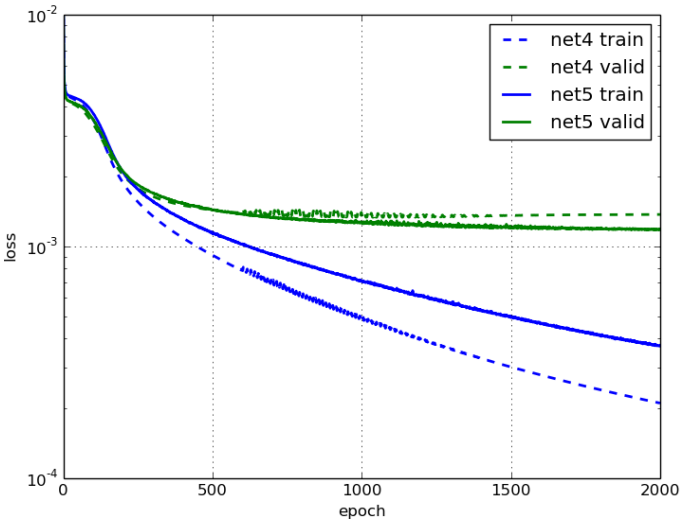


# Overfitting-Problematik

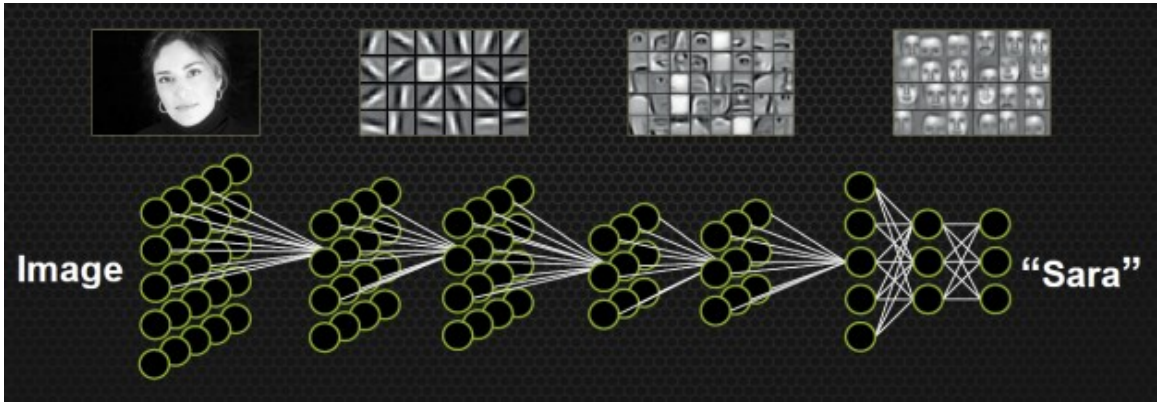


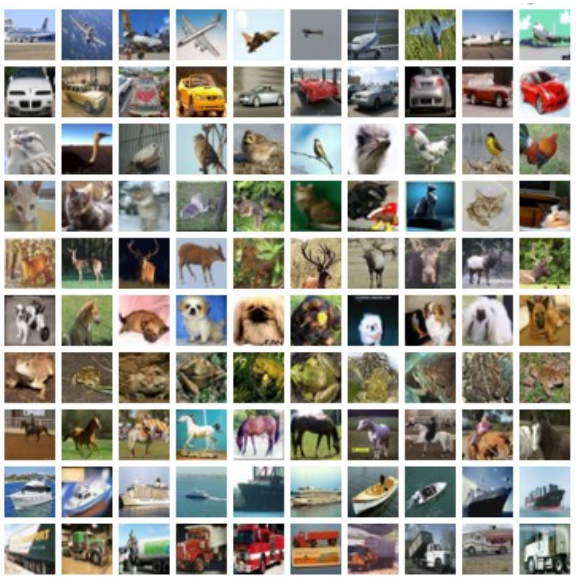




Bildquellen:
http://ogrisel.github.io/scikit-learn.org/sklearn-tutorial/tutorial/astronomy/practical.html
http://stats.stackexchange.com/questions/131233/neural-network-over-fitting

# Convolutional Neural Networks (ConvNets)

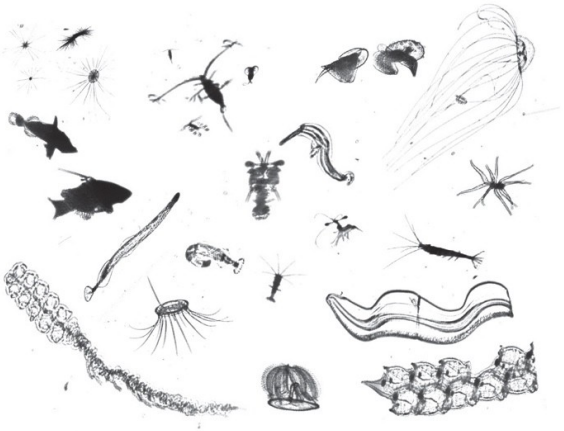


# Klassifikation



CIFAR-10 (60.000, 10 Klassen):
http://torch.ch/blog/2015/07/30/cifar.html

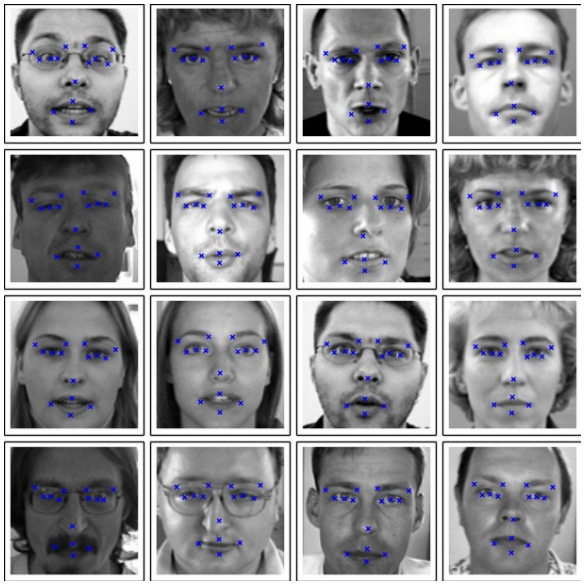

GTSRB (50.000, 43 Klassen):
http://benchmark.ini.rub.de/



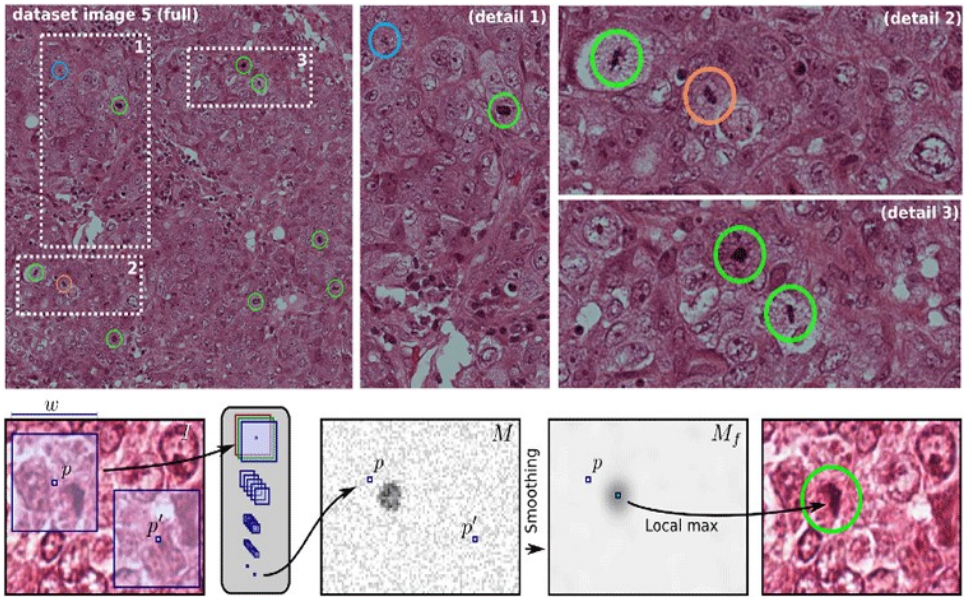
NDSB (30.000, 121 Klassen):
https://www.kaggle.com/c/datascibowl
http://benanne.github.io/2015/03/17/plankton.html

ILSVRC (>1 Mio, 1000 Klassen):
http://image-net.org/challenges/LSVRC/2015/

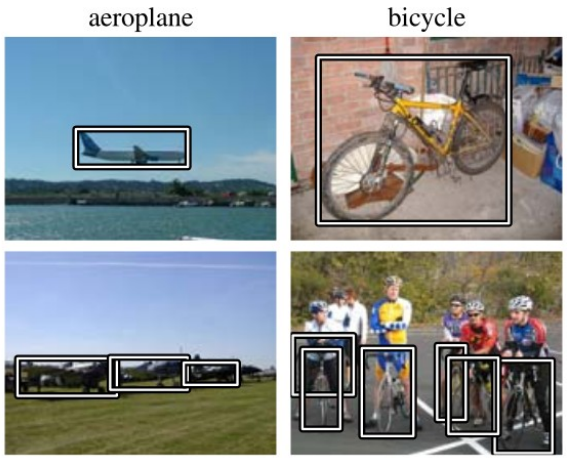
# Regression / Detection



Facial Keypoint Detection
http://danielnouri.org/notes/2014/12/17/using-convolutional-neural-nets-to-detect-facial-keypoints-tutorial/




Mitosis Detection in Breast Cancer
http://people.idsia.ch/~ciresan/data/miccai2013.pdf



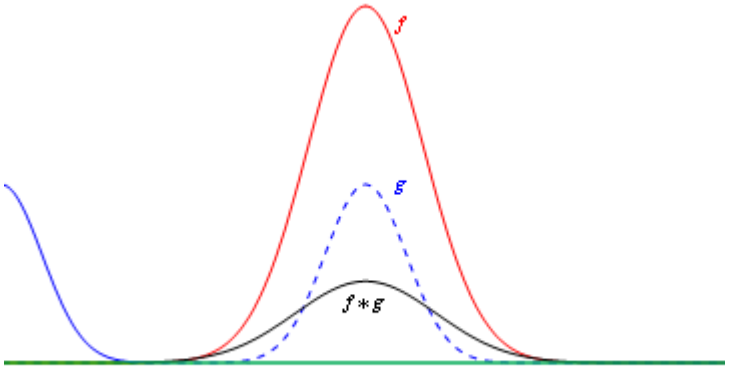

Pascal VOC Detection Challenge
http://host.robots.ox.ac.uk/pascal/VOC/pubs/everingham10.pdf

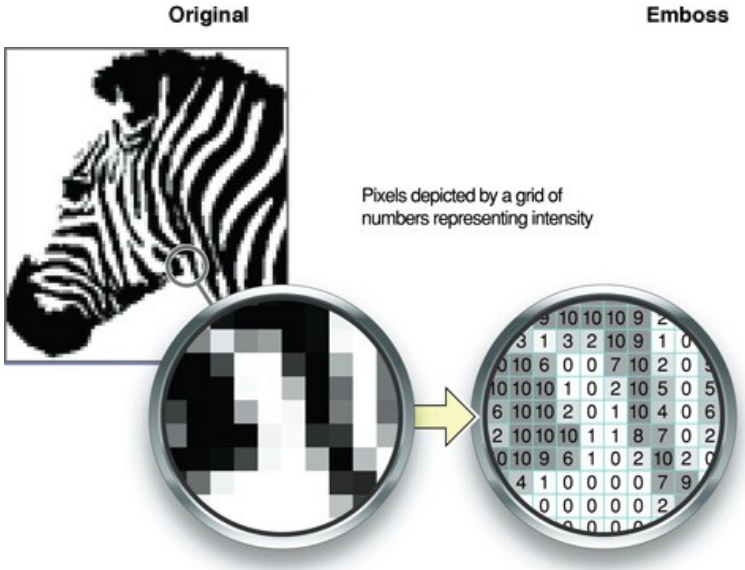
Populäre Detection-Verfahren auf Basis von ConvNets:

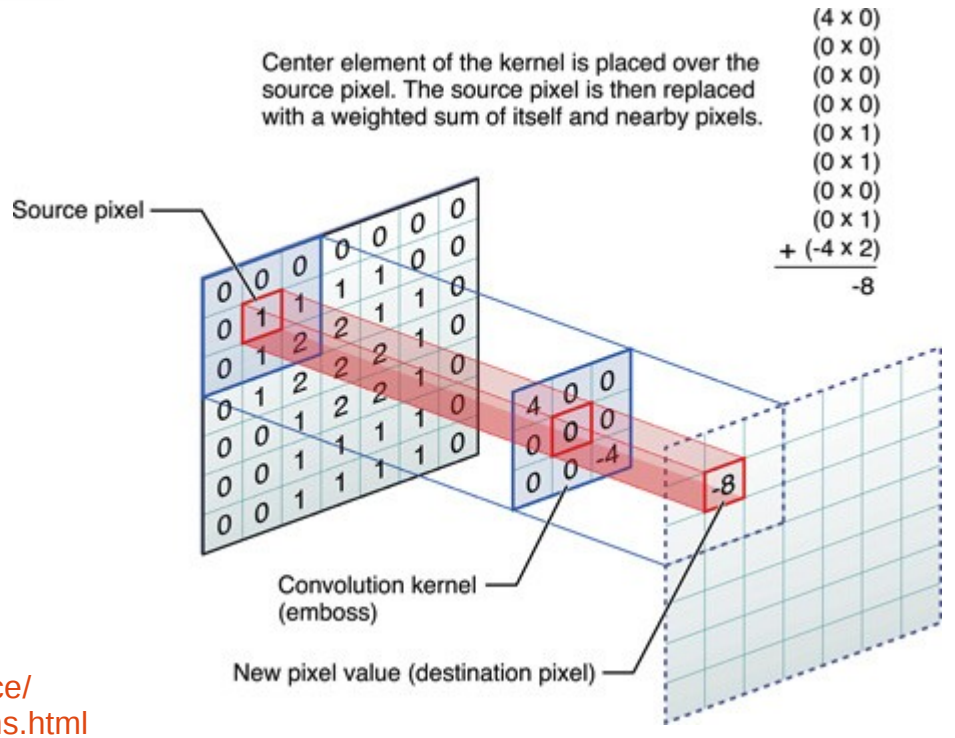
- Overfeat
  http://arxiv.org/abs/1312.6229
- (Faster-)RCNN
  http://arxiv.org/abs/1506.01497

# Technische Details von ConvNets

# Convolution = Filtern



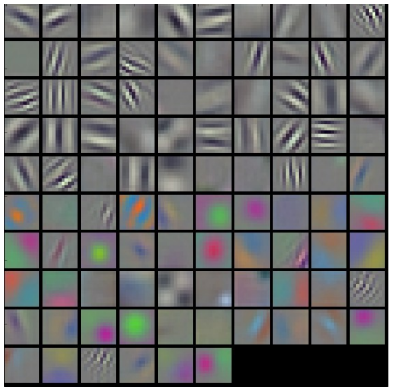




Bildquellen:
https://developer.apple.com/library/ios/documentation/Performance/Conceptual/vImage/ConvolutionOperations/ConvolutionOperations.html
https://de.wikipedia.org/wiki/Faltung_%28Mathematik%29

# ConvNets

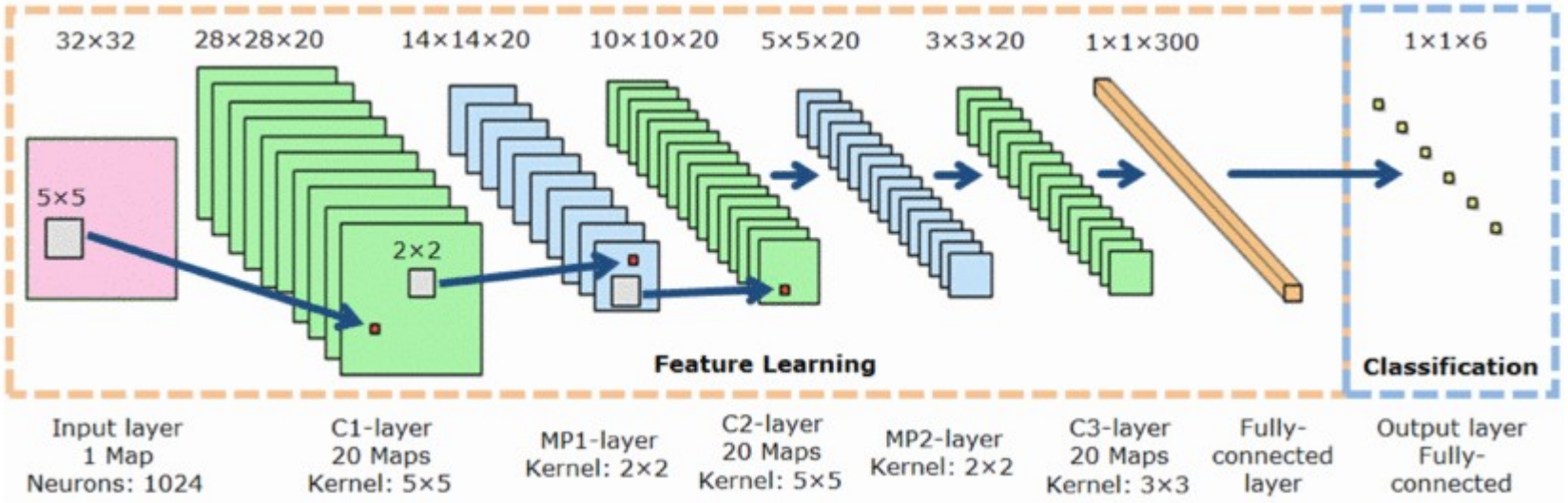
Yann LeCun

- Populärste Deep-Learning Technik
- Entwickelt in den 90ern: LeCun et al, 1998
- Neuronen mit lokalen rezeptiven Feldern
- Nicht-lineare Transfer-Funktionen (heute meist ReLU)
- Pooling/Sub-Sampling → Dimension Schrittweise reduzieren

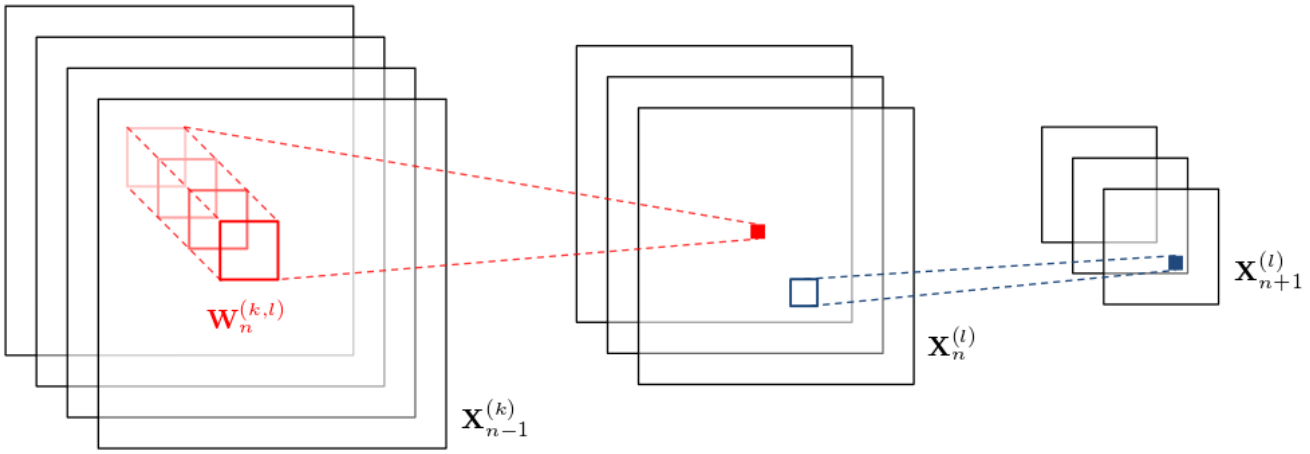


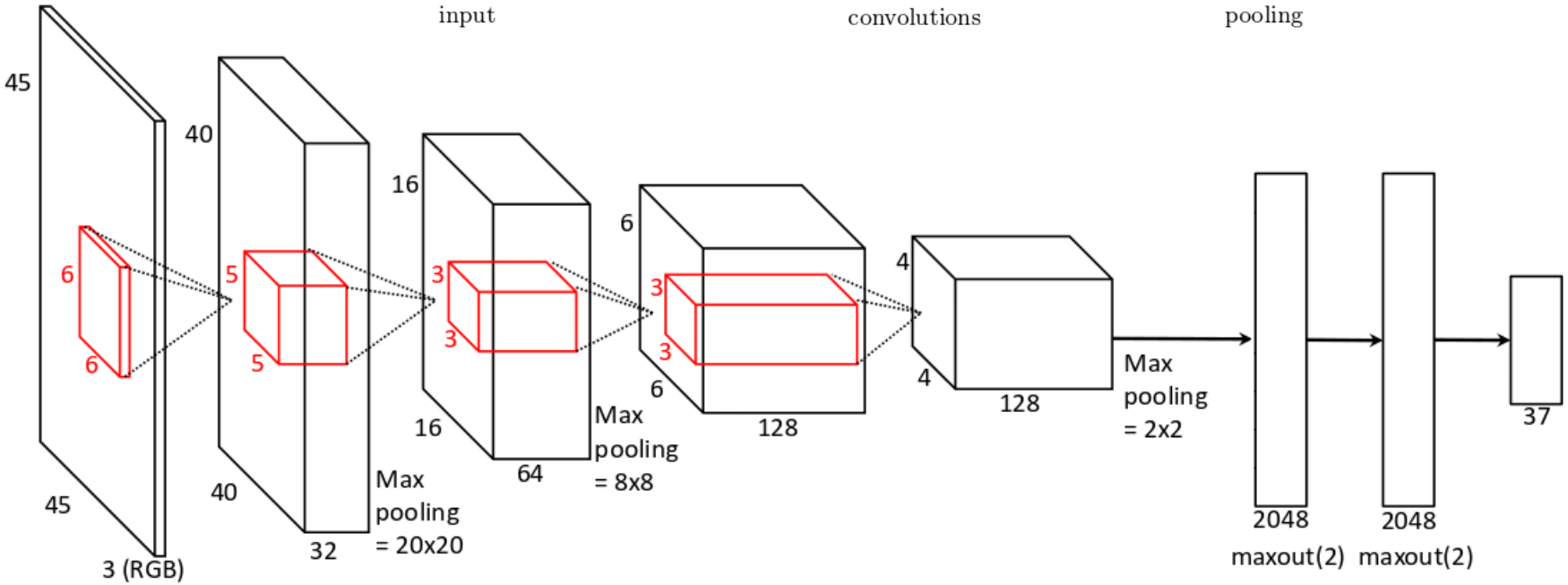
Typische Filter Conv1

- Vollverknüpfte Schichten nach mehreren Sub-Sampling-Schritten
- Ausgabeschicht, z.B. Softmax für Klassifikation



# Convolution + Pooling

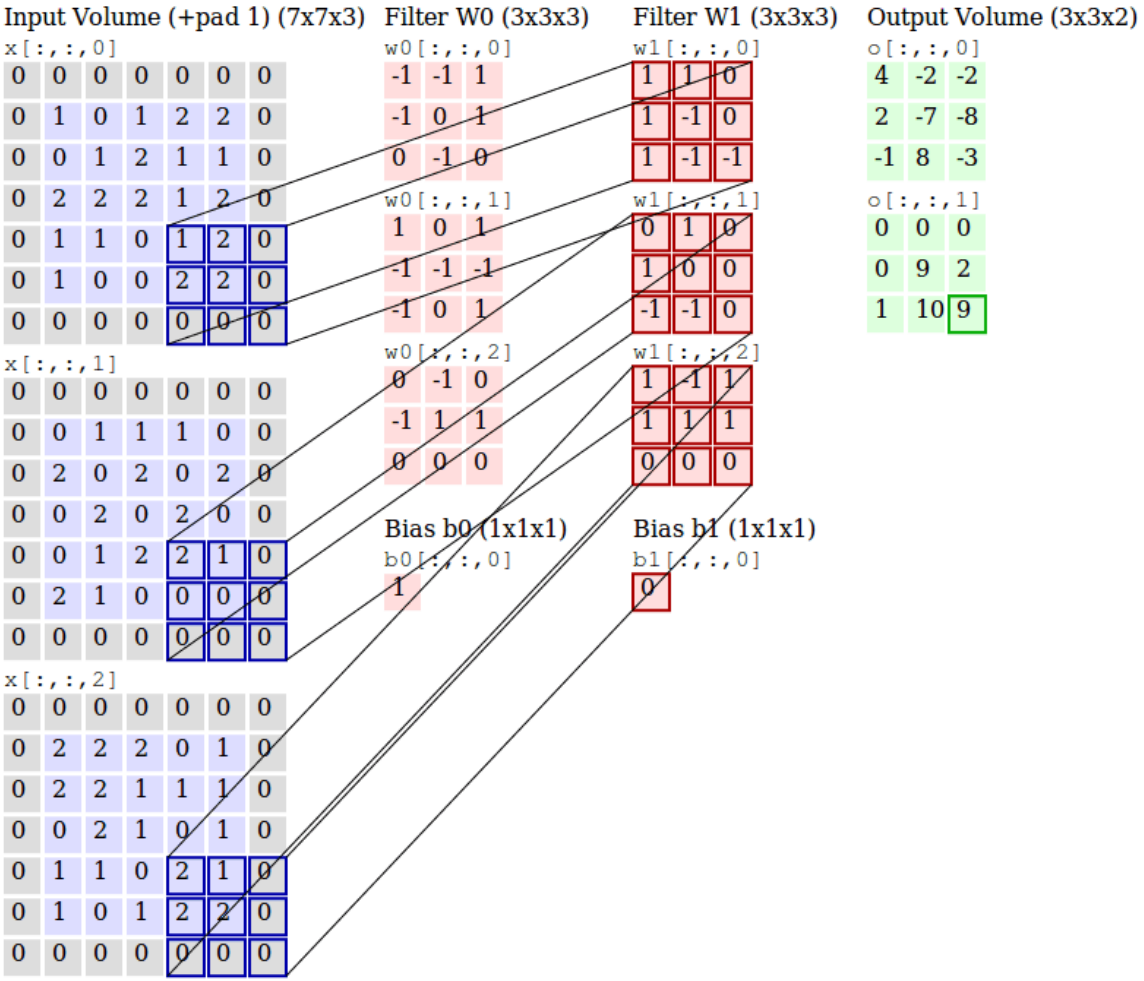



Bildquelle: (Sander Dieleman et al.) Rotation-invariant convolutional neural networks for galaxy morphology prediction
http://arxiv.org/abs/1503.07077

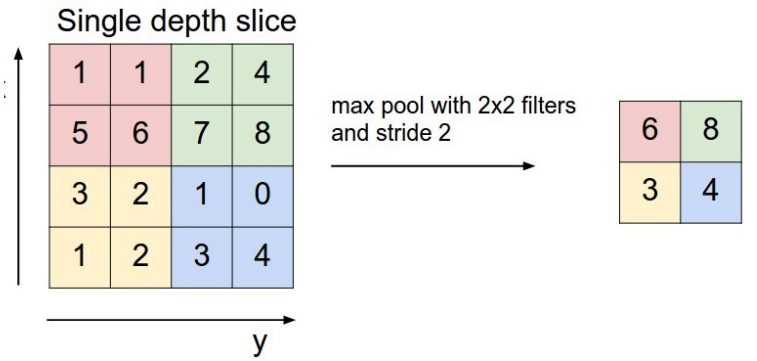
# Convolution Animation

Filter: 2; Kernel: 3x3; Stride 2,2; Padding 1,1;



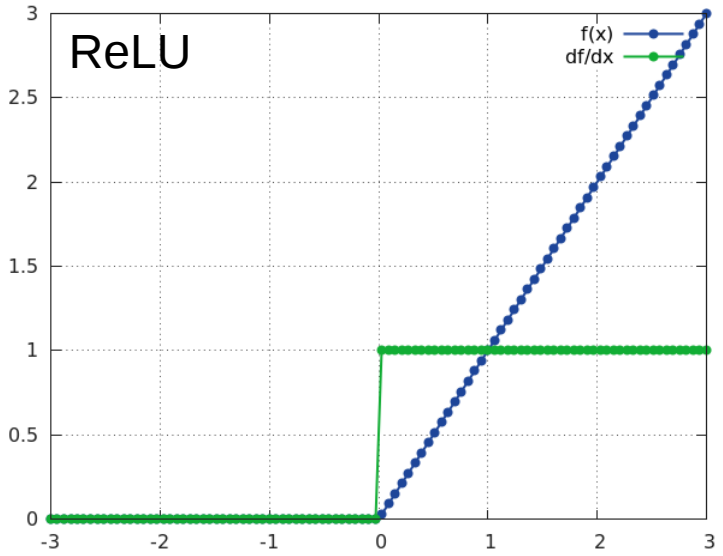

$$x_{ij}^{\ell} = \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} \omega_{ab} y_{(i+a)(j+b)}^{\ell-1}.$$

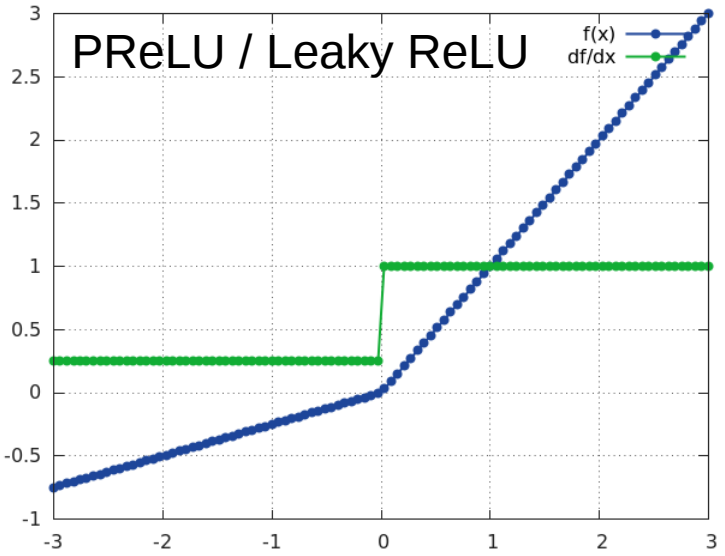
## 2x2 Max Pooling:

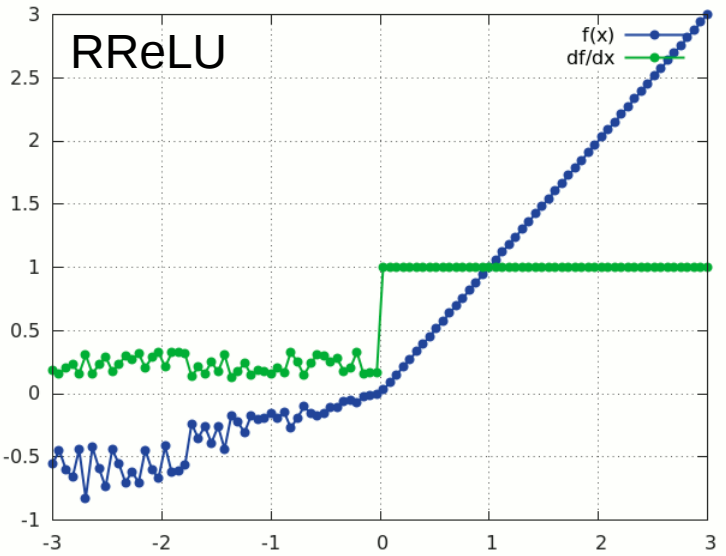



Bildquellen: Andrej Karpathy, https://cs231n.github.io/convolutional-networks/

# Rectified Linear Unit (ReLU)



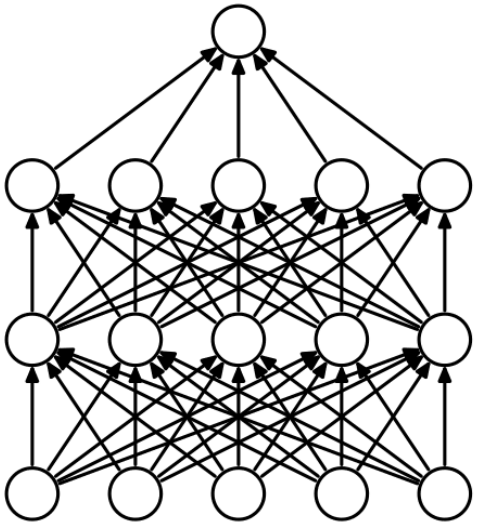

(Kaiming He et al.) Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification
http://arxiv.org/abs/1502.01852

(Bing Xu et al.) Empirical Evaluation of Rectified Activations in Convolutional Network
http://arxiv.org/abs/1505.00853

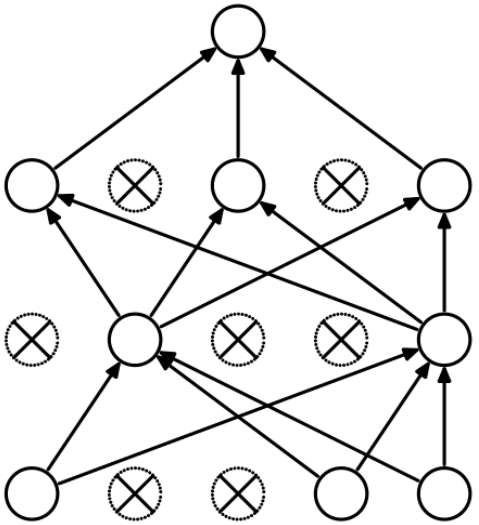
# Regularisierung mit Dropout

Ausgaben mit einer bestimmten Wahrscheinlichkeit (z.B. p=0.5) auf 0 setzten.

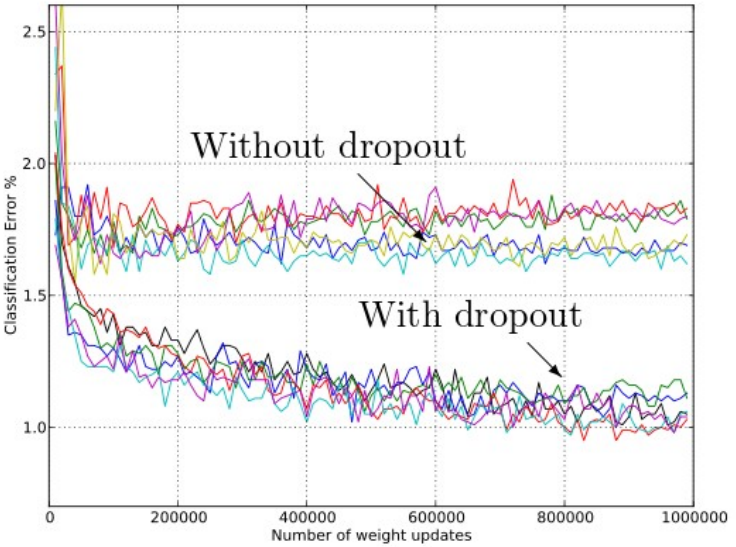
“provides a way of approximately combining exponentially many different neural network architectures. “
“each hidden unit (..) must learn to work with a randomly chosen sample of other units”



(a) Standard Neural Net



(b) After applying dropout.

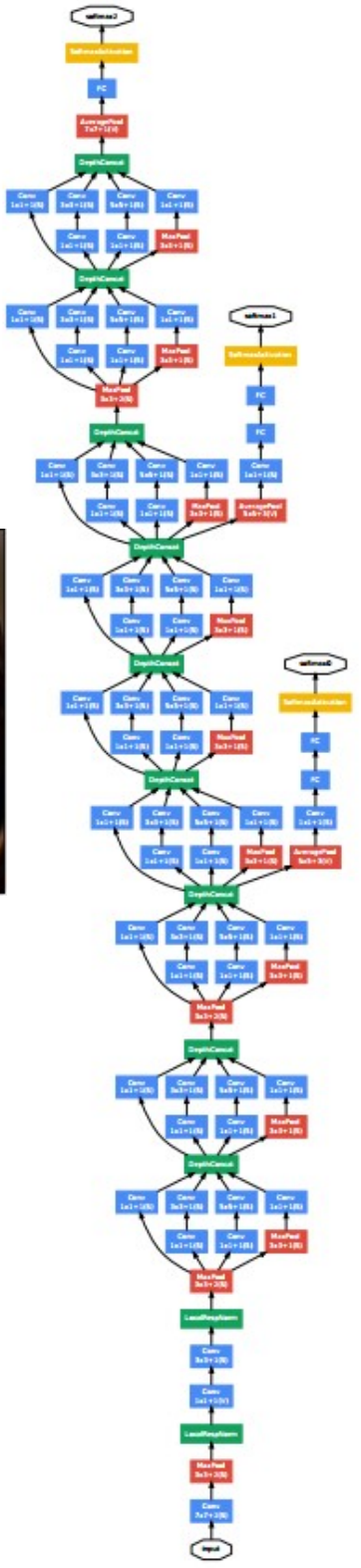



(Nitish Srivastava et al.) Dropout: A Simple Way to Prevent Neural Networks from Overtting
http://www.cs.toronto.edu/~rsalakhu/papers/srivastava14a.pdf

# Beispiel für tiefe ConvNet Architektur

# GoogLeNet

- Gewinner der ILSVRC 2014 in den Kategorien „Classification“ und „Detection“
- 22 Schichten tief
- 9 Inception Module
- 5 Mio Parameter
- Auxiliary Classifiers: Lösung für “vanishing gradient” Problem
- ReLU
- < 5% error auf ImageNet (1000 Klassen)
- Verwendet für die Indizierung in Google Photos

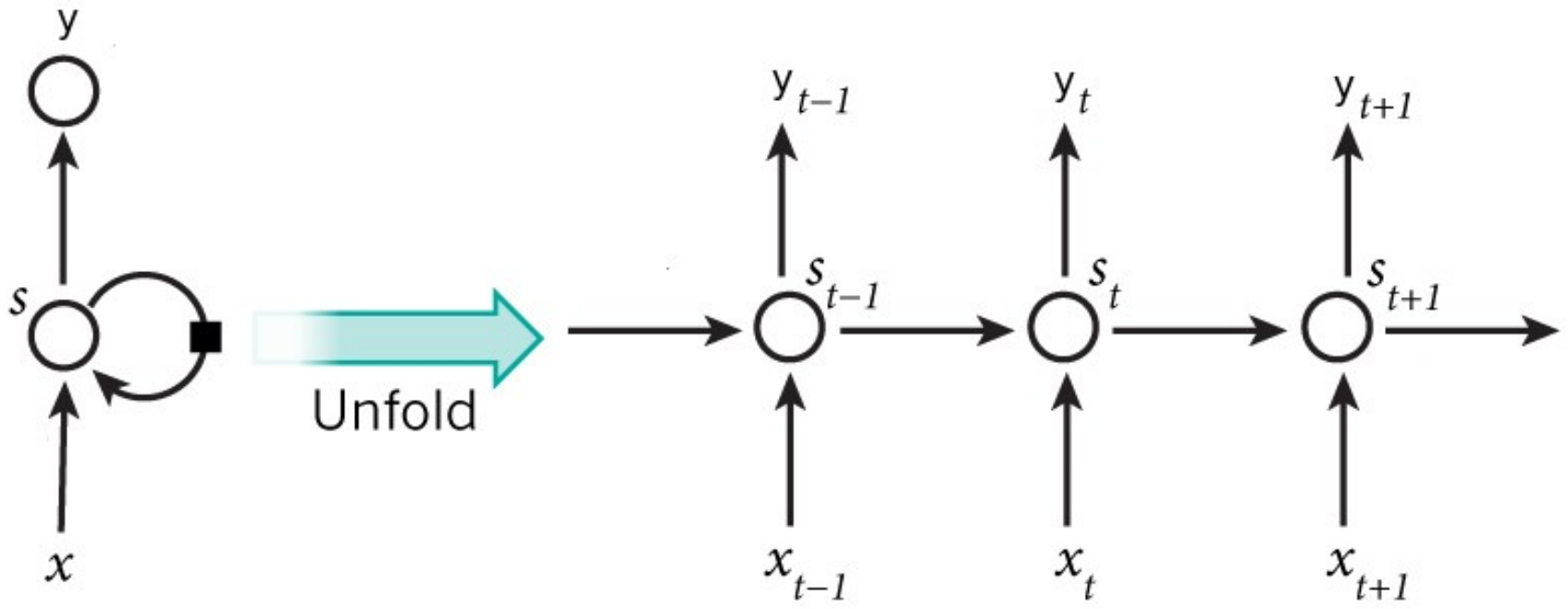



Paper: Going Deeper with Convolutions (C. Szegedy et al. 2014)

# Rekurrente Neuronale Netze (RNN)

# Rekurrente Neuronale Netze

Zusammenhang mit tiefen Netzen: “Unfolding in time”



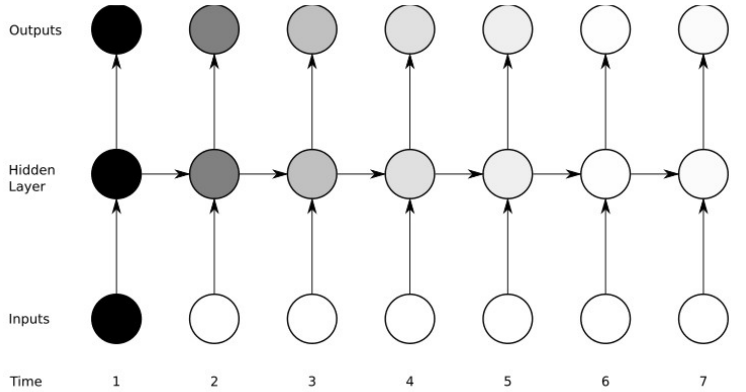
# Long Short Term Memory (LSTM)

**Problem:**

Bei Backpropagation Through Time (BPTT) wird Gradientensignal wiederholt mit Gewichtsmatrix multipliziert

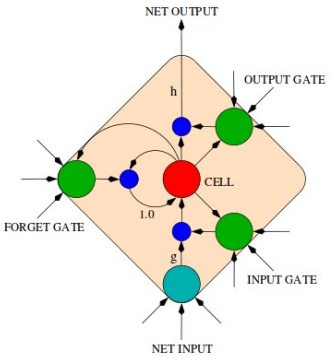
→ Gradientensignal wird exponentiell kleiner/größer, der Lernprozess wird langsam/divergiert (**vanishing/exploding Gradients**)

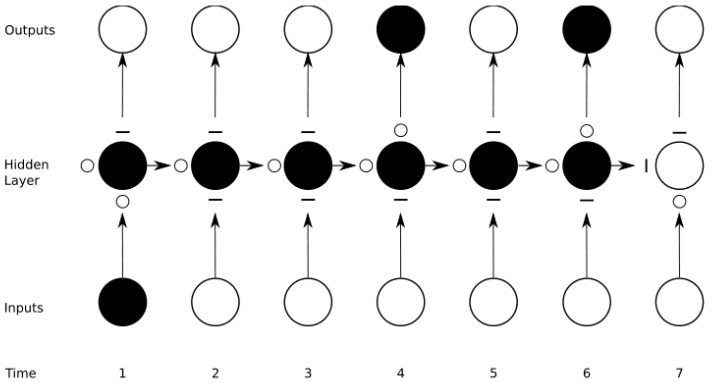
→ Trainieren von **Langzeitabhängigkeiten** in den Daten sehr **schwierig**!




**Lösung:**

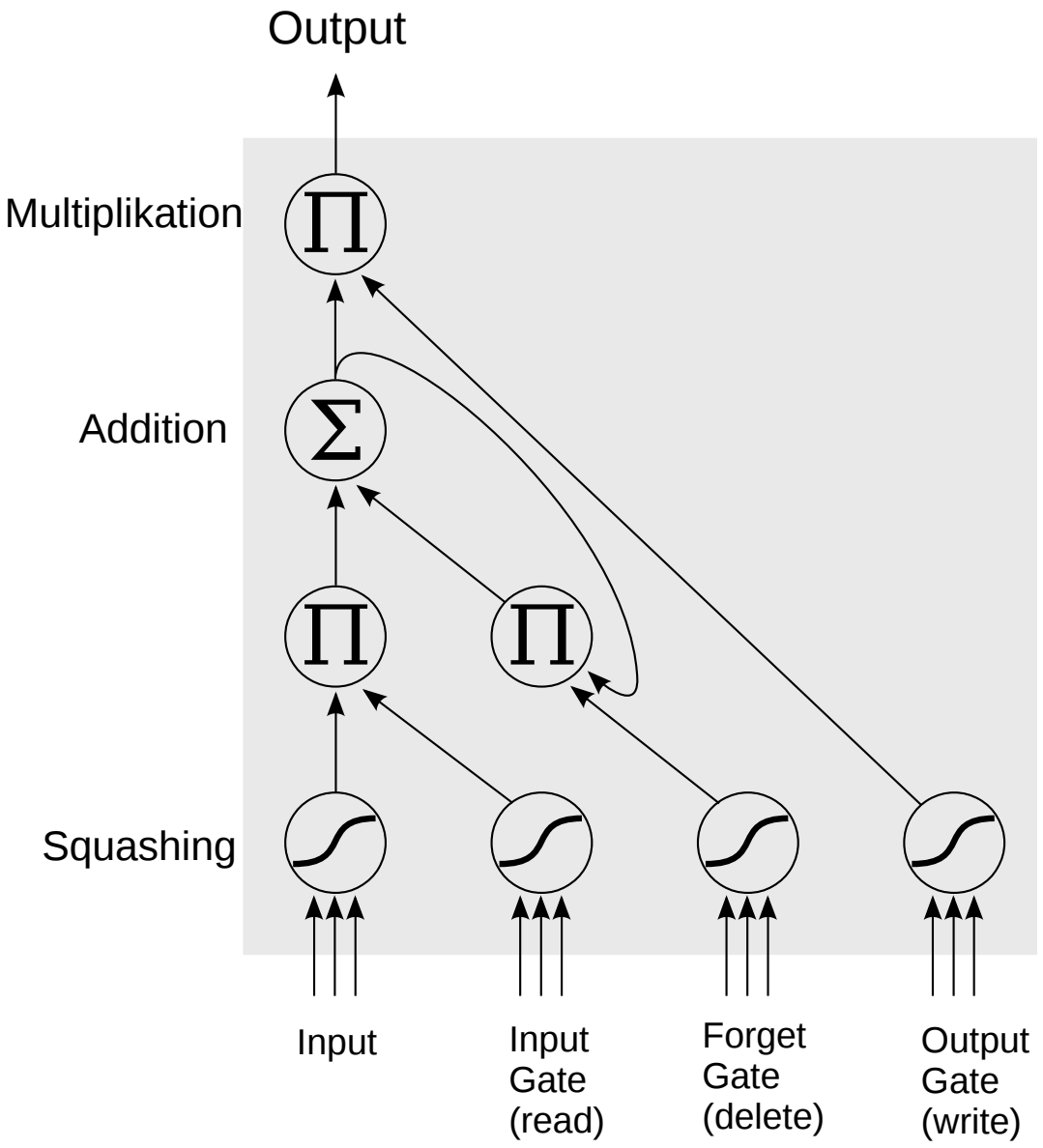
“Memory Cells” mit Gates




(Felix Gers) Dissertation:
Long Short-Term Memory in Recurrent Neural Networks
(Alex Graves et al.)
A Novel Connectionist System for Unconstrained Handwriting Recognition

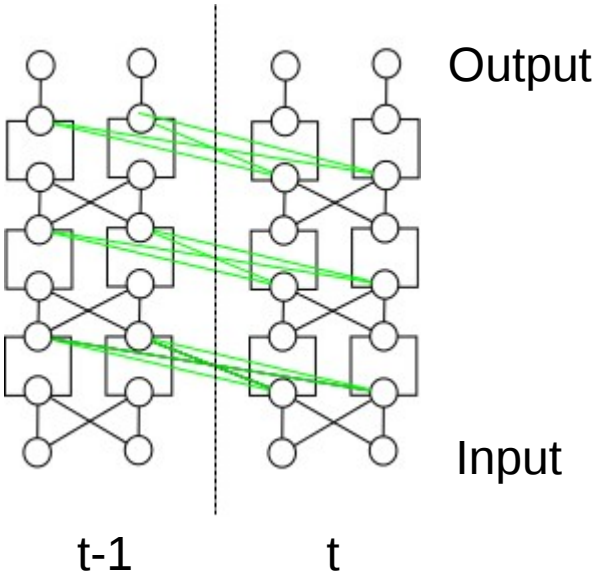
# LSTM Block (*Memory Cell*)




Input: Ausgaben aus vorherigem Zeitschritt t-1 + neue Netzeingaben

Squashing: Transferfunktion
Gate-Squashing: Sigmoid [0, 1]
Input-Squashing: tanh [-1, 1]

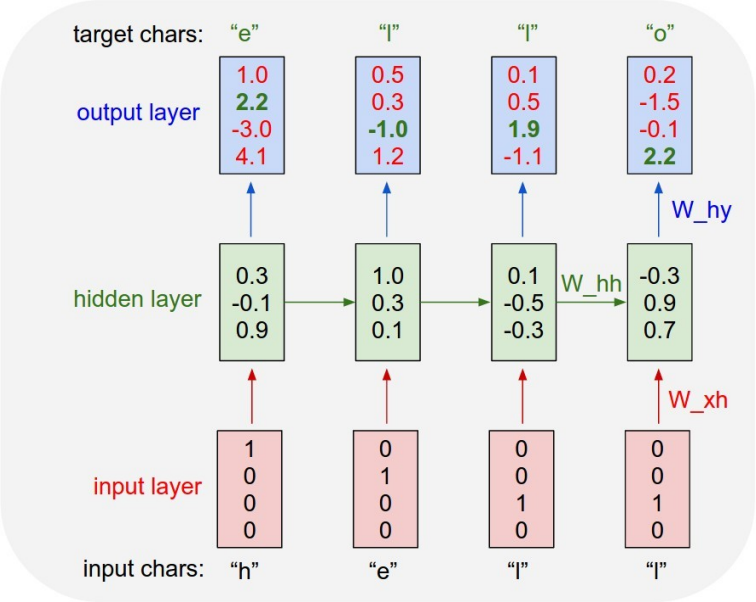
**Mehrschichtige LSTM Netze**



# RNN Anwendungen

# Zeichenbasiertes Sprachmodell

## Char-RNN




Trainingsziel für "output layer":
grüne Zahlen hoch, rote Zahlen niedrig
→ Wahrscheinlichkeiten über Softmax

- Ziel: Vorhersage des nächsten Zeichens mit RNN
- 1-of-k Encoding (Onehot)
- Beste Ergebnisse mit LSTM (vs. GRU, Standard-RNN)
- Lernt Langzeitabhängigkeiten und Kontext, z.B. Satzzeichen, Klammern
- Zeichenwahrscheinlichkeiten können zur Generierung verwendet werden

(Andrej Karpathy) The Unreasonable Effectiveness of Recurrent Neural Networks
http://karpathy.github.io/2015/05/21/rnn-effectiveness/
Code: https://github.com/karpathy/char-rnn

# Demos: Obama RNN, XML

„The United States will step up to the cost of a new challenges of the American people that will share the fact that we created the problem. They were attacked and so that they have to say that all the task of the final days of war that I will not be able to get this done. The promise of the men and women who were still going to take out the fact that the American people have fought to make sure that they have to be able (...)

Thank you very much. God bless you, and God bless the United States of America.“

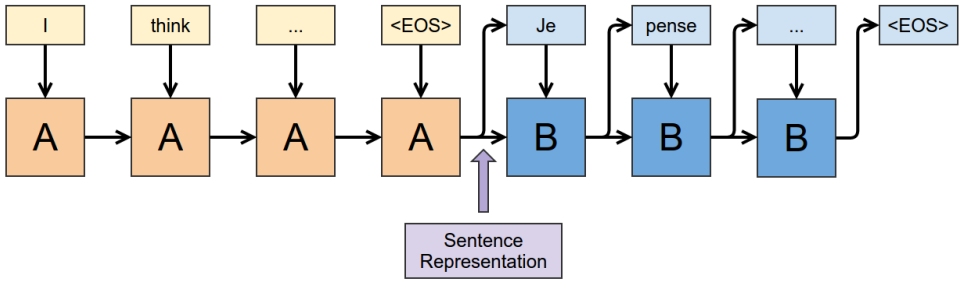
Obama-RNN — Machine generated political speeches.
https://medium.com/@samim/obama-rnn-machine-generated-political-speeches-c8abd18a2ea0

```
<page>
  <title>Antichrist</title>
  <id>865</id>
  <revision>
    <id>15900676</id>
    <timestamp>2002-08-03T18:14:12Z</timestamp>
    <contributor>
      <username>Paris</username>
      <id>23</id>
    </contributor>
    <minor />
    <comment>Automated conversion</comment>
    <text xml:space="preserve">#REDIRECT [[Christianity]]</text>
  </revision>
</page>
```

Mit char-rnn generiertes XML,
Tags werden korrekt geschlossen.

# Neural Machine Translation

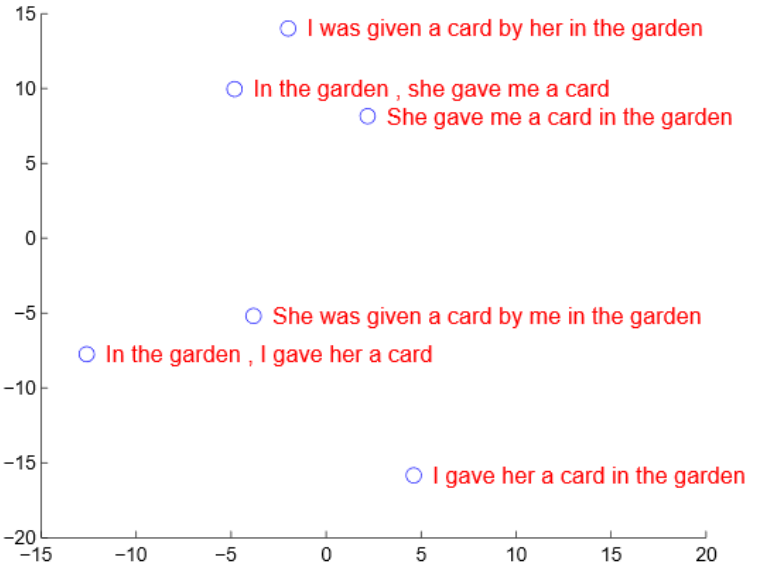
- Encoder → „Summary Vector“ → Decoder




- Deep LSTM (4 Schichten)
- 1000 Zellen pro Schicht
- 1000 dim. “Word-Embeding”

„We found deep LSTMs to significantly outperform shallow LSTMs“

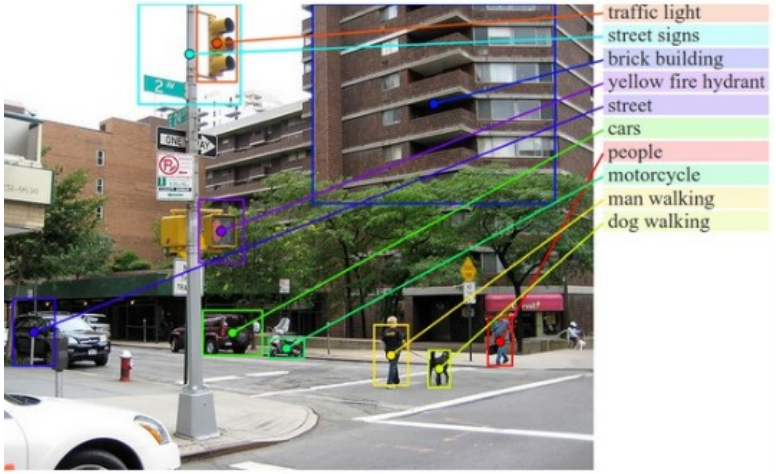
Ähnlich: Automatische Generierung von E-Mail Antworten (Gmail Smart Reply suggestion)

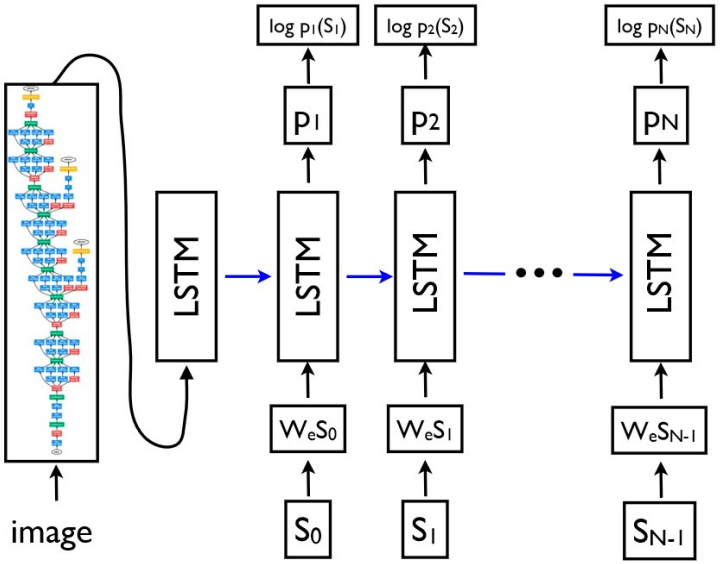



2D PCA der Summary Vektoren

(Ilya Sutskever, Oriol Vinyals, Quoc V. Le) Sequence to Sequence Learning with Neural Networks
http://arxiv.org/abs/1409.3215
Blogpost: Visualizing Representations: Deep Learning and Human Beings
http://colah.github.io/posts/2015-01-Visualizing-Representations/

# Generierung von Bildunterschriften

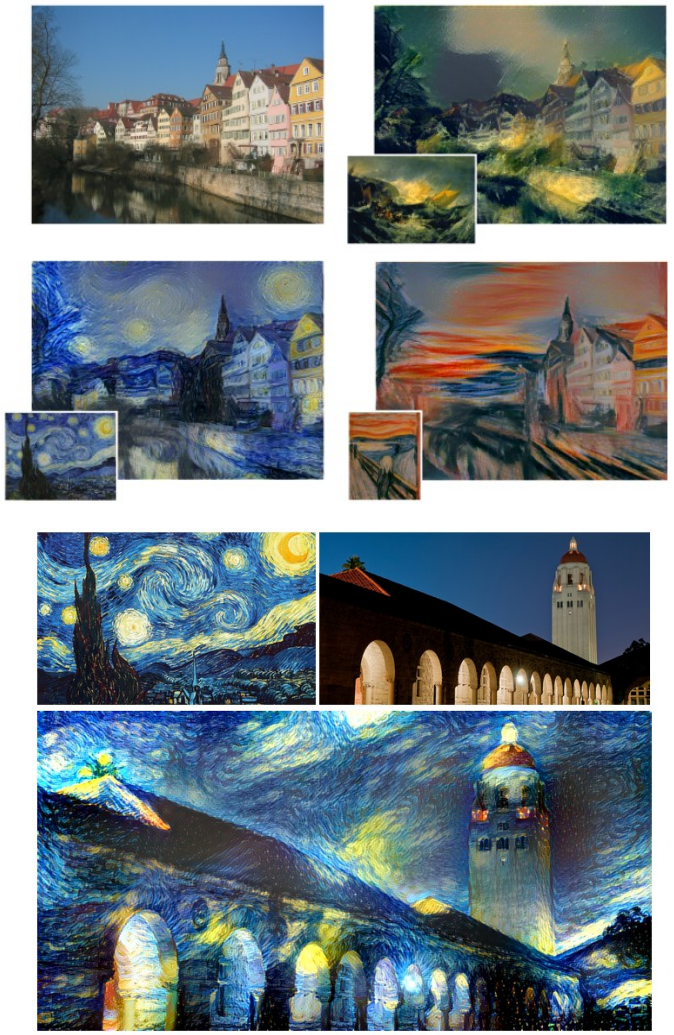

Karpathy: RCNN regions → bidirectional RNN

Vinyals: CNN as Image Encoder

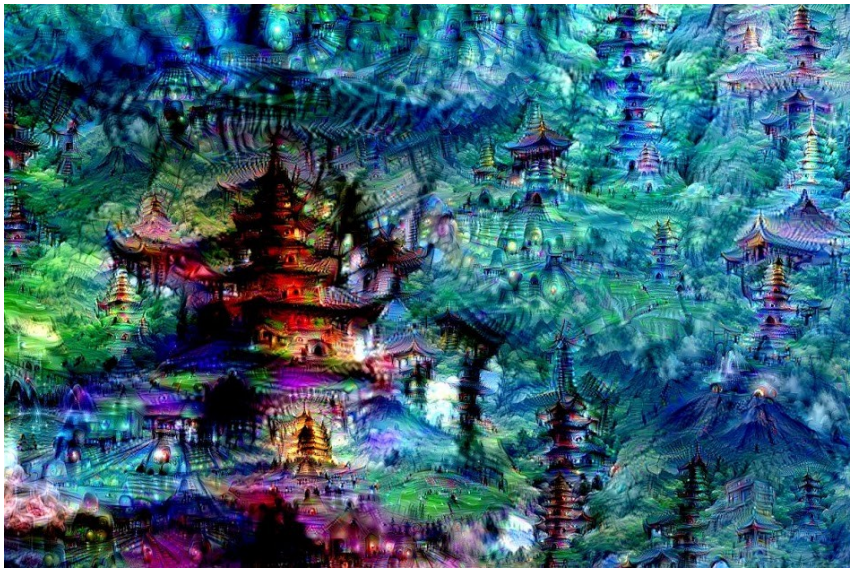
(Andrej Karpathy, Li Fei-Fei) Deep Visual-Semantic Alignments for Generating Image Descriptions
http://arxiv.org/abs/1412.2306
(Vinyals et al.) Show and Tell: A Neural Image Caption Generator
http://arxiv.org/abs/1411.4555

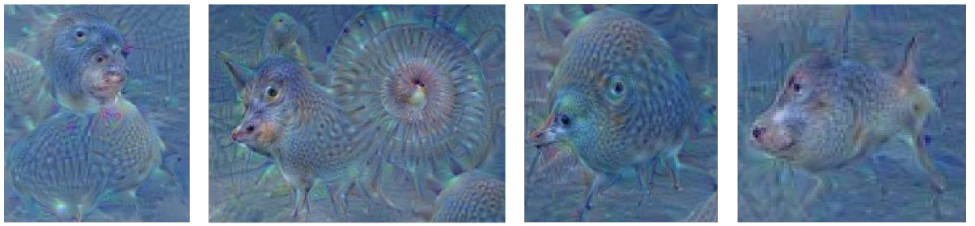
# Kunst & Visualisierungen

# Visuelle Kunst mit tiefen ConvNets



**StyleNet:**
http://arxiv.org/pdf/1508.06576v2.pdf
https://github.com/jcjohnson/neural-style





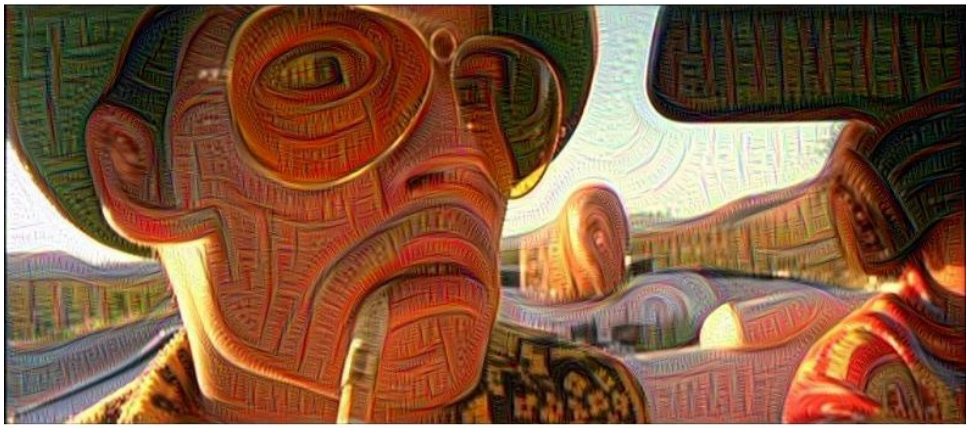
"Admiral Dog!" "The Pig-Snail" "The Camel-Bird" "The Dog-Fish"

**Deep Dream:**
**http://googleresearch.blogspot.co.uk/2015/06/inceptionism-going-deeper-into-neural.html**
https://github.com/google/deepdream

# Die Welt mit GoogLeNet's Augen

A

C



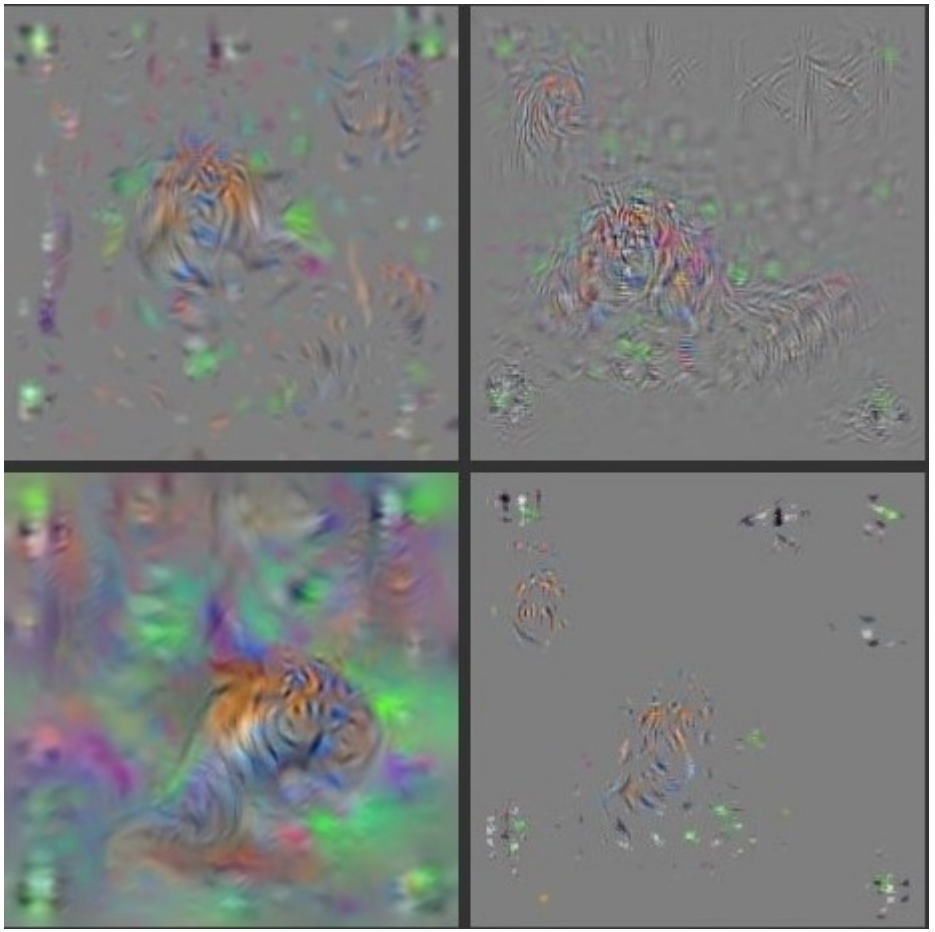
B

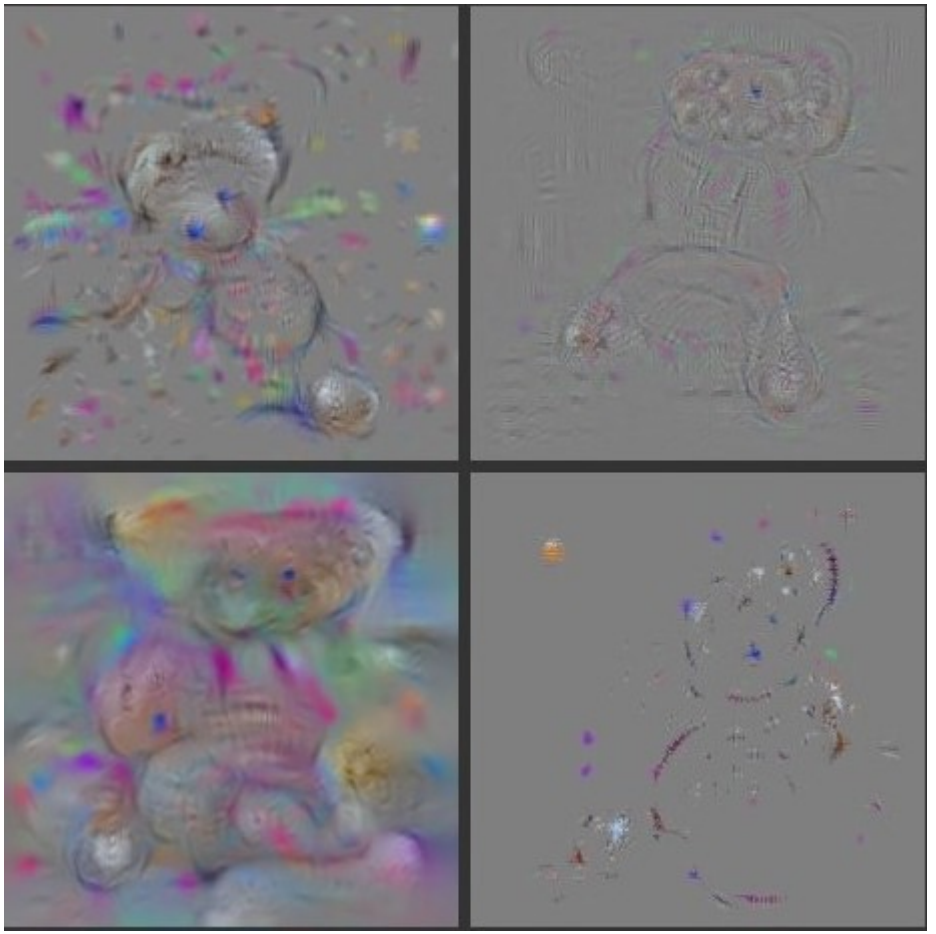
D

Maximierung der Aktivierung einer Schicht von GoogLeNet mit Gradientenaufstieg (B-D)
A: Original; B: conv2/3x3; C: inception_3b/5x5_reduce; D: inception_4c/output

(Adrian Rosebrock) Blog Post: bat-country: an extendible, lightweight Python package for deep dreaming with Caffe and Convolutional Neural Networks
http://www.pyimagesearch.com/2015/07/06/bat-country-an-extendible-lightweight-python-package-for-deep-dreaming-with-caffe-and-convolutional-neural-networks/

# Deep Quiz





Regularisierungstechniken:
“L2 decay”, “Gaussian blur” und “Clipping” von Pixeln mit kleiner Norm bzw. kleinem Beitrag

Video: https://www.youtube.com/watch?v=AgkfIQ4IGaM
Paper: (Jason Yosinski et al.) “Understanding Neural Networks Through Deep Visualization”
http://arxiv.org/abs/1506.06579

# Deep Learning Tools


http://torch.ch/
(basiert auf LuaJIT)

theano

http://deeplearning.net/software/theano/
(basiert auf Python)

Caffe

http://caffe.berkeleyvision.org/
(Python, C++)

TensorFlow™

http://www.tensorflow.org/
(Python, C++)

# Fragen

## Weiterführende Links

- Deep Learning in your browser
  http://cs.stanford.edu/people/karpathy/convnetjs/
- Deep Q-Learning
  https://www.cs.toronto.edu/~vmnih/docs/dqn.pdf
- Reinforcement Learning Learning in Robotics
  http://rll.berkeley.edu/deeplearningrobotics/
- Neural Turing Machine
  http://arxiv.org/abs/1410.5401
- Deep Speech
  http://arxiv.org/abs/1412.5567
- Generative Modelle
  http://arxiv.org/abs/1502.04623

# Bonus Slides

# Historisches & Schlüsselpersonen

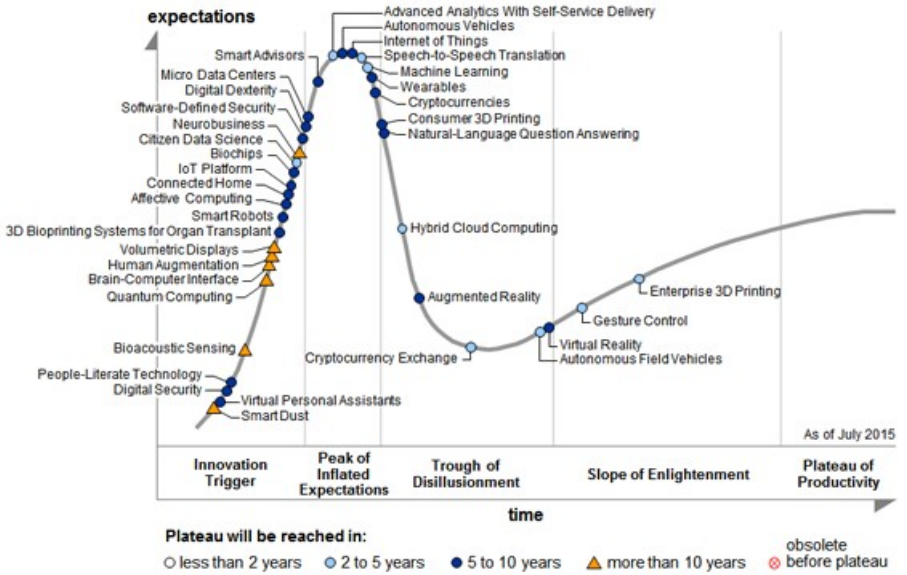
- 1957: Frank Rosenblatt entwickelt das Perceptron
- 1974: Paul Werbos führt Backpropagation für NN ein
- **1997: Hochreiter & Schmidhuber entwickeln LSTM**
- **1999: Yann LeCun entwickelt ConvNets**
- 2006: Geoffrey Hinton: Dimensionsreduktion mit tiefen neuronalen Netzen erscheint in Science
- 2009-2012: IDSIA (Schmidhuber) Gruppe gewinnt in Serie internationale ML Wettbewerbe

(v.l.n.r) Yann LeCun (Facebook),
Geoffrey Hinton (Google),
Yoshua Bengio,
Andrew Ng (Baidu),
Jürgen Schmidhuber (IDSIA)

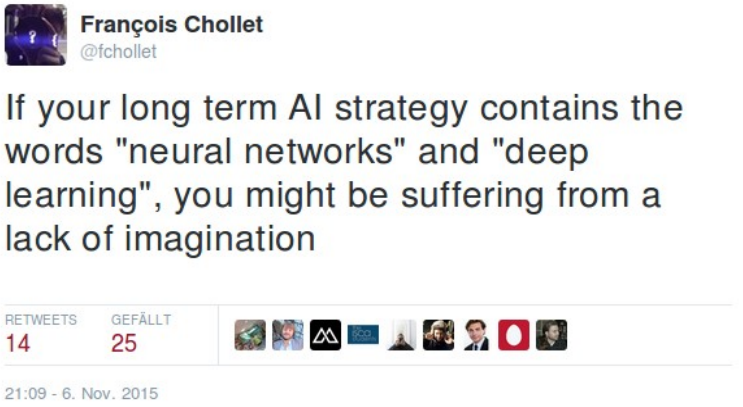
Historische Details von J. Schmidhuber:
Critique of Paper by "Deep Learning Conspiracy"
http://people.idsia.ch/~juergen/deep-learning-conspiracy.html

# Hype-Warnung

Gartner Hype-Zyklus Juli 2015:
Machine Learning im Hype-Maximum
“Tal der Enttäuschung” voraus







# OxfordNet (aka VGGNet)

- Beliebtes Basis-Modell für die Merkmalextraktion aus Bildern
- **138 Millionen Parameter** (Variante D)
- Gewinner des ILSVRC 2014 (ImageNet) Wettbewerbs in der Kategorie "Localization" und zweiter Platz in der Kategorie "Classification"
- Homogene Architektur: 3x3 Convs
- Jeweils ReLU nach Conv und FC
- Padding 1,1 (conv input = output size)
- 5 max-pooling Schichten
- 3 vollverknüpfte (FC) Schichten

| Input Size | VGG16 |
|---|---|
| 224 x 224 | conv 3x3, 64 |
| 224 × 224 | conv 3x3, 64 |
| 224 × 224 | max pooling 2x2 /2 |
| 112 × 112 | conv 3x3, 128 |
| 112 × 112 | conv 3x3, 128 |
| 112 × 112 | max pooling 2x2 /2 |
| 56 × 56 | conv 3x3, 256 |
| 56 × 56 | conv 3x3, 256 |
| 56 × 56 | conv 3x3, 256 |
| 56 × 56 | max pooling 2x2 /2 |
| 28 × 28 | conv 3x3, 512 |
| 28 × 28 | conv 3x3, 512 |
| 28 × 28 | conv 3x3, 512 |
| 28 × 28 | max pooling 2x2 /2 |
| 14 × 14 | conv 3x3, 512 |
| 14 × 14 | conv 3x3, 512 |
| 14 × 14 | conv 3x3, 512 |
| 14 × 14 | max pooling 2x2 /2 |
| 7 × 7 × 512 | fc, 4096 |
| 4096 × 1 | dropout 0.5 |
| 4096 × 1 | fc, 4096 |
| 4096 × 1 | dropout 0.5 |
| 4096 × 1 | fc, 1000 |
| 1000 | softmax |

Paper: Very Deep Convolutional Networks for Large-Scale Image Recognition
(Karen Simonyan, Andrew Zisserman)

# Fehlerfunktion

Exemplarisch: Kleinste Quadrate Fehlerfunktion (RMSE) mit Regularisierung

$$\min_{\mathbf{w}} \mathrm{E}(\mathbf{w}) \quad \text{mit} \quad \mathrm{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N} ||\mathbf{y}(\mathbf{x}_n, \mathbf{w}) - \mathbf{t}_n||^2 + \frac{\lambda}{2}||\mathbf{w}||^2$$

| | | | |
|---|---|---|---|
| $\{\mathbf{x}_n\}_{n=1,\ldots,N}$ | Input-Vektoren | $\mathbf{w}$ | Gewichte |
| $\{\mathbf{t}_n\}_{n=1,\ldots,N}$ | Target-Vektoren | $\lambda$ | Regularisierungsparameter |
| $\{\mathbf{y}_n\}_{n=1,\ldots,N}$ | Netz-Output-Vektoren $\mathbf{y}_n := \mathbf{y}(\mathbf{x}_n, \mathbf{w})$ | | |

**Gradientenschritt**: $\mathrm{w}^{\mathrm{new}} = \mathrm{w}^{\mathrm{old}} + \eta \cdot \frac{\partial}{\partial \mathrm{w}} \mathrm{E}(\mathrm{w})$

Berechnung der partiellen Ableitungen nach w per **Backpropagation**

# Neuronale Netze in der Praxis

## Stärken

- Nicht-lineare, universelle Funktionsapproximatoren
- Konzeptionell einfach & modular
- erlauben nachträgliches Finetuning mit mehr Daten
- effektiv für Mehrklassen-Probleme
- weltklasse bei Bildklassifikation und Objekterkennung
- Können als Merkmalgenerator verwendet werden
- Training jederzeit abbrechbar

## Schwächen

- langwieriges Training
- nicht interpretierbar
- Ergebnisse nicht immer reproduzierbar
- nicht vollständig theoretisch verstanden
- neigen zur Überanpassung (Overfitting), braucht Regularisierung
- viele Hyper-Parameter

# Tricks of the Trade

- Vorverarbeitung: Normalisierung der Trainingsbeispiele ($\mu = 0$, $\sigma = 1$), Whitening
- Bilddaten: Vorfilterungen wie Local-Contrast Normalization (LCN), Nutzung anderer Farbräume (z.B. YUV, Lab, HSL)
- ReLU Aktivierungsfunktionen
- Gewichtsinitialisierung (Xavier Glorot, orthogonal QR oder SVD, MSR)
- Regularisierung: Dropouts, Weight-Decay
- Data-Augmentation (Spiegelungen, 90˚ Rotationen, Elastic-Distortions etc.)
- Batchnormalisierung
- Spatial-Transformer Networks
- Mehrere 3x3 Convolutions in Folge
- 1x1 Convolutions = Network-in-Network (NiN), Inception Module
- Nutzung vortrainierter Modelle (z.B. auf ImageNet, siehe Caffe Model Zoo)