

Mario Neises

Aktuelle Entwicklung und Probleme bei  
**künstlichen neuronalen Netzen**  
in Bezug auf  
**Deep Learning**

# Inhalt

- Einführung
- Grundlagen
- *Zwischenfazit*
- Deep Learning
- Probleme
- *Fazit*

## Einführung: führende Forscher

- Geoffrey E. Hinton (Google)
- Yann LeCun (Facebook)
- Yoshua Bengio

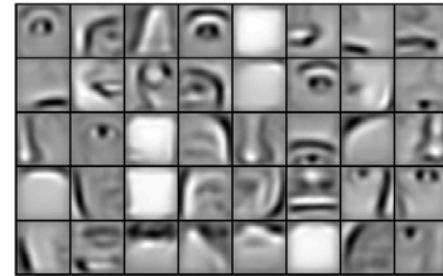


Bildquellen: <http://www.cs.toronto.edu/~hinton/>, <http://yann.lecun.com/>, [http://www.iro.umontreal.ca/~bengioy/yoshua\\_en/](http://www.iro.umontreal.ca/~bengioy/yoshua_en/)

# Einführung: Klassifizierung

Anforderungen an (künstliche) neuronale Netze:

- Generalisierung
- Lernfähigkeit
- Fehlertoleranz
- Parallelisierbarkeit

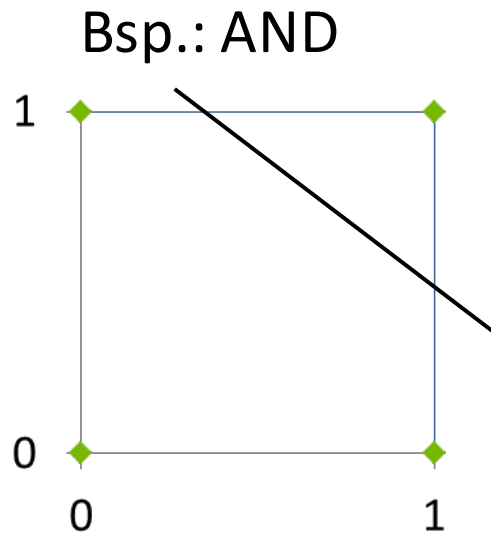


Merkmalsextraktion

Bildquelle: H. Lee et al., "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations", 2009, Abbildung 3

# Einführung: Klassifizierbarkeit

- Lineare Separierbarkeit



$$1 * w_0 + 0 * w_1 + 0 * w_2 \leq 0$$

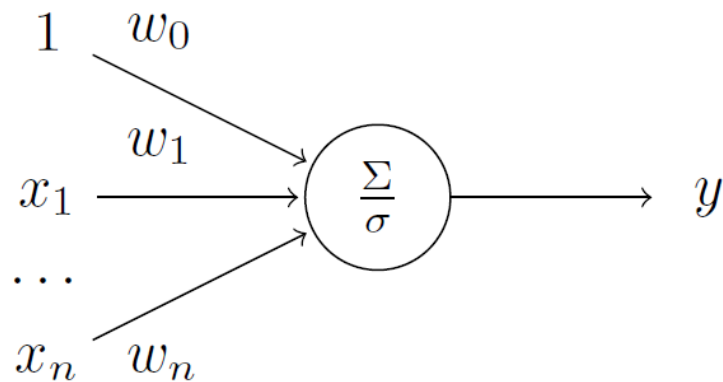
$$1 * w_0 + 0 * w_1 + 1 * w_2 \leq 0$$

$$1 * w_0 + 1 * w_1 + 0 * w_2 \leq 0$$

$$1 * w_0 + 1 * w_1 + 1 * w_2 \geq 0$$

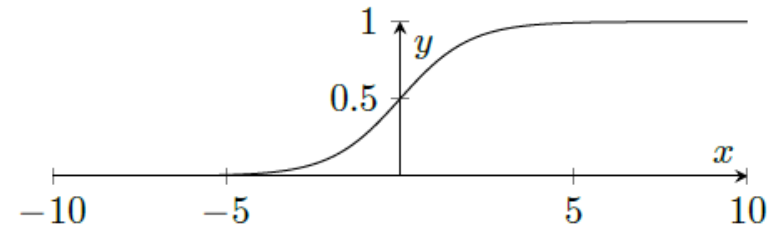
# Grundlagen: Bausteine

- Neuron



$$\sum_{i=0}^n (w_i * x_i) = \vec{w} * \vec{x}$$

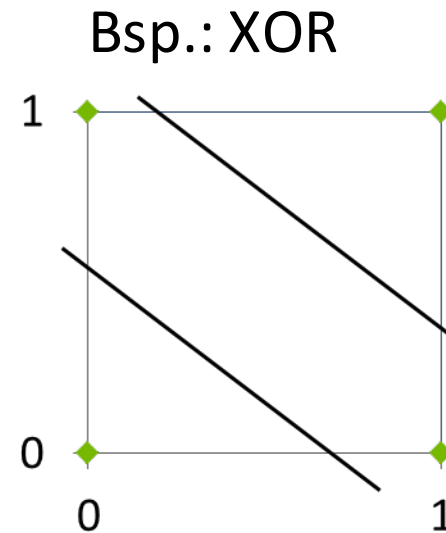
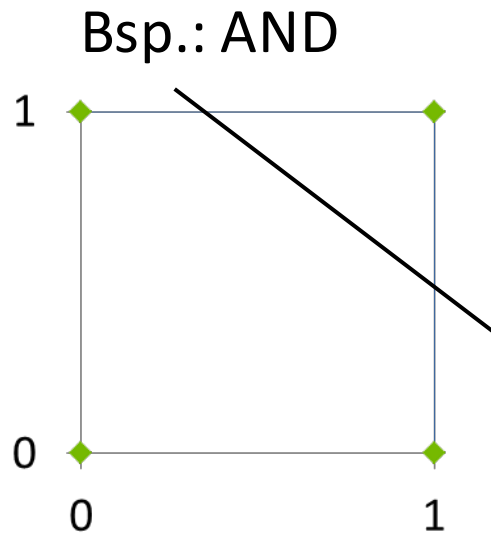
- mit Aktivierungsfunktion



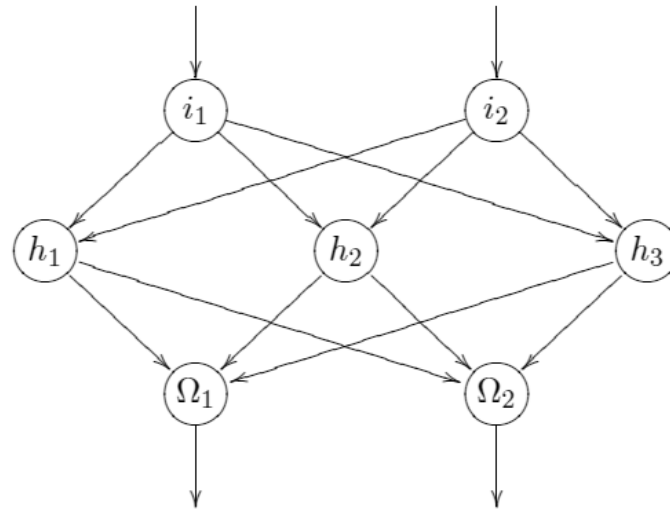
$$\sigma(\vec{x} * \vec{w})$$

# Einführung: Klassifizierbarkeit

- Lineare Separierbarkeit
- Nicht l. Separierbarkeit



# Grundlagen: Netze



$\vec{r}$	$i_1$	$i_2$	$h_1$	$h_2$	$h_3$	$\Omega_1$	$\Omega_2$
$i_1$							
$i_2$							
$h_1$							
$h_2$							
$h_3$							
$\Omega_1$							
$\Omega_2$							

Quelle: David Kriesel, Ein kleiner Überblick über Neuronale Netze, 2007, S. 43

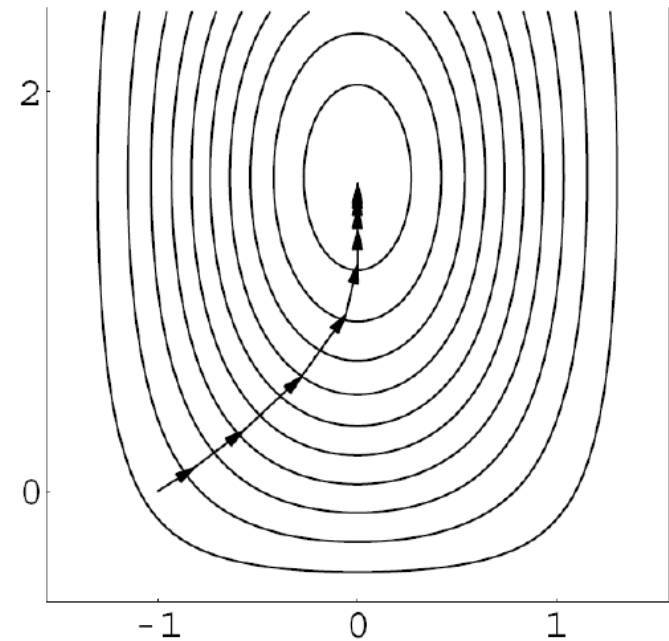
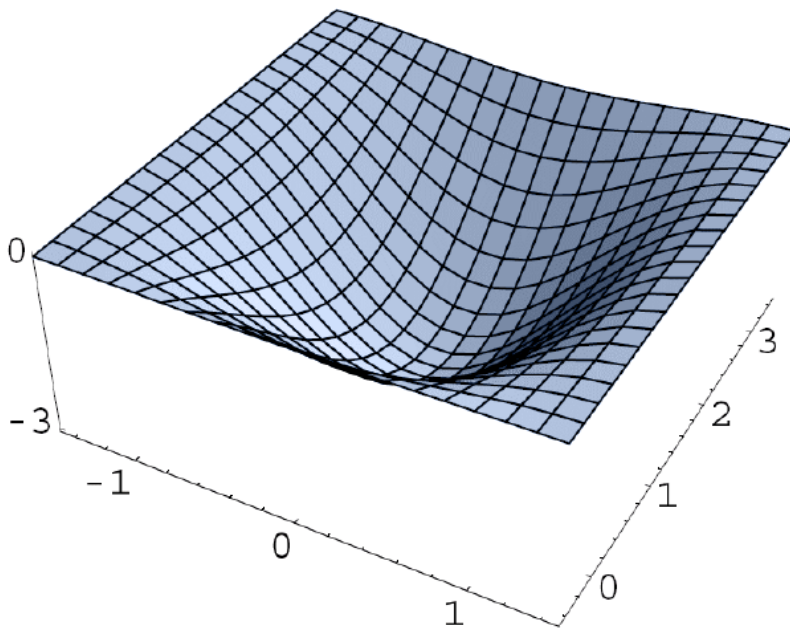


## Grundlagen: Lernen

- Lernen: Anpassen der Gewichte
- Training: Bekannte Ausgabe zur Eingabe
- Fehler =  $\text{Ausgabe}_{\text{Bekannt}} - \text{Ausgabe}_{\text{Berechnet}}$

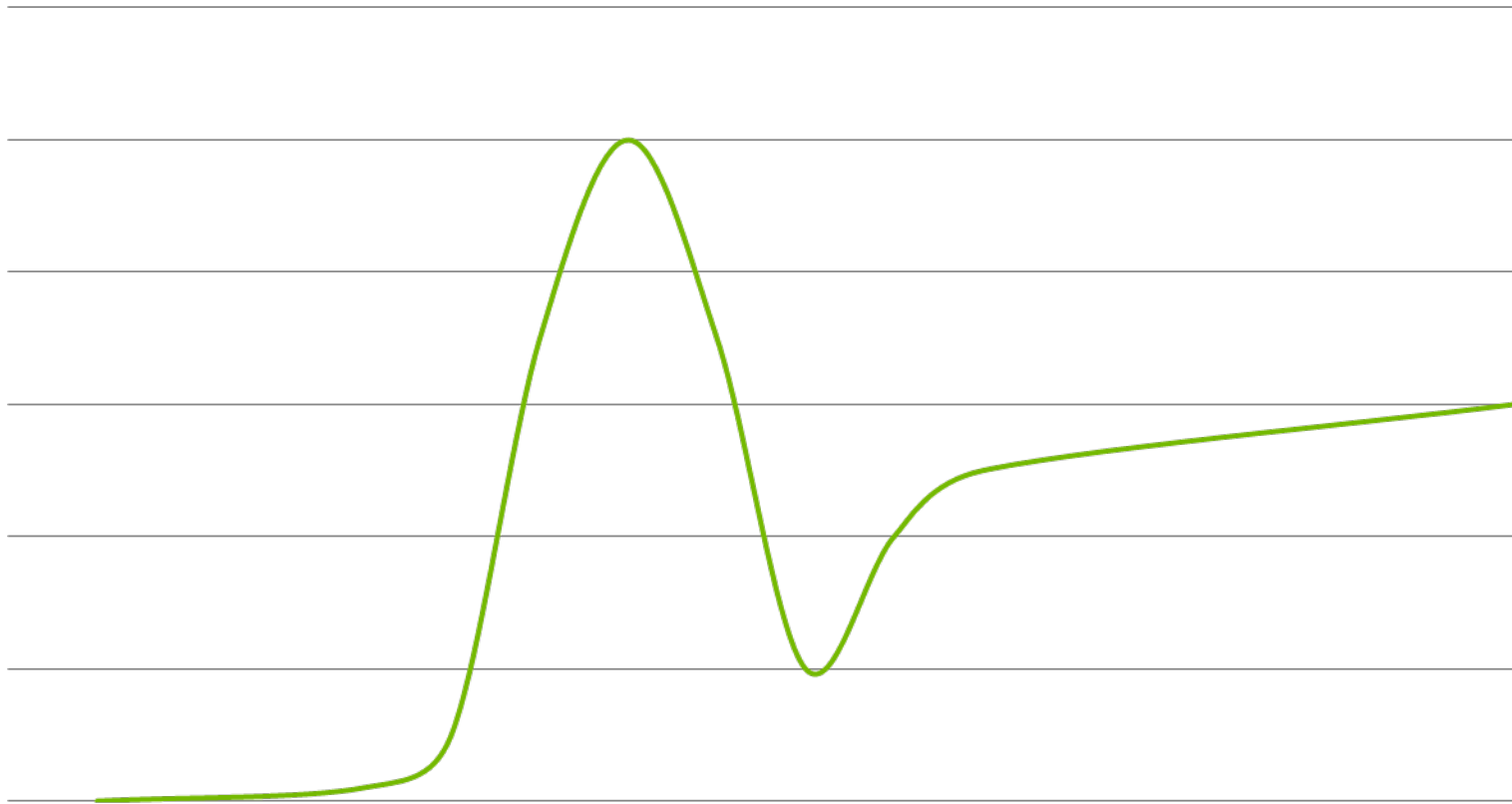
# Grundlagen: Selbstständiges Lernen

- Gradientenabstieg



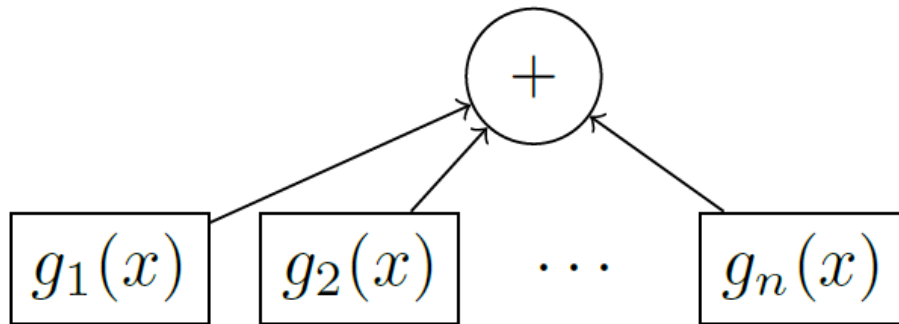
Quelle: David Kriesel, Ein kleiner Überblick über Neuronale Netze, 2007, S. 64

## Zwischenfazit

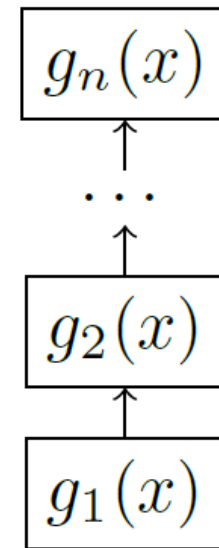


# Deep Learning

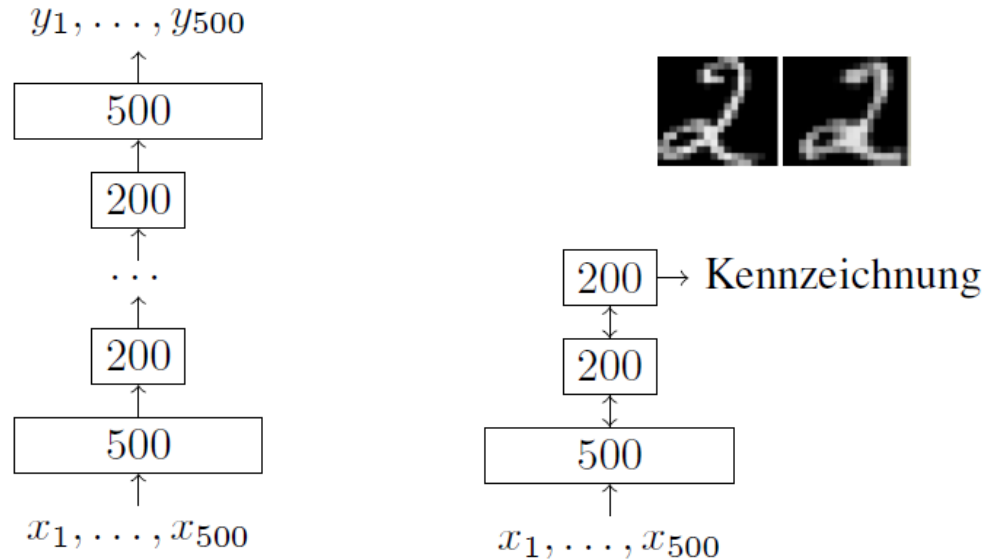
$$f \approx \sum_i^n g_i$$



$$f \approx g_1(g_2(\dots g_n))$$



# Deep Learning: Anwendungen



Deep Belief Nets: <http://www.cs.toronto.edu/~hinton/adi/index.htm>

ConvNetJS MNIST: <http://cs.stanford.edu/people/karpathy/convnetjs/demo/mnist.html>

# Probleme

1. spezifische Probleme
2. generelle Probleme

## Probleme: Spezifischer Natur

Beispiele:

- Stabilitäts-Plastizitäts-Dilemma
- Overfitting
- verschlechtert Erkennung
- spezifisch je nach Problem und Topologie
- vermeidbar / reduzierbar

## Probleme: Genereller Natur

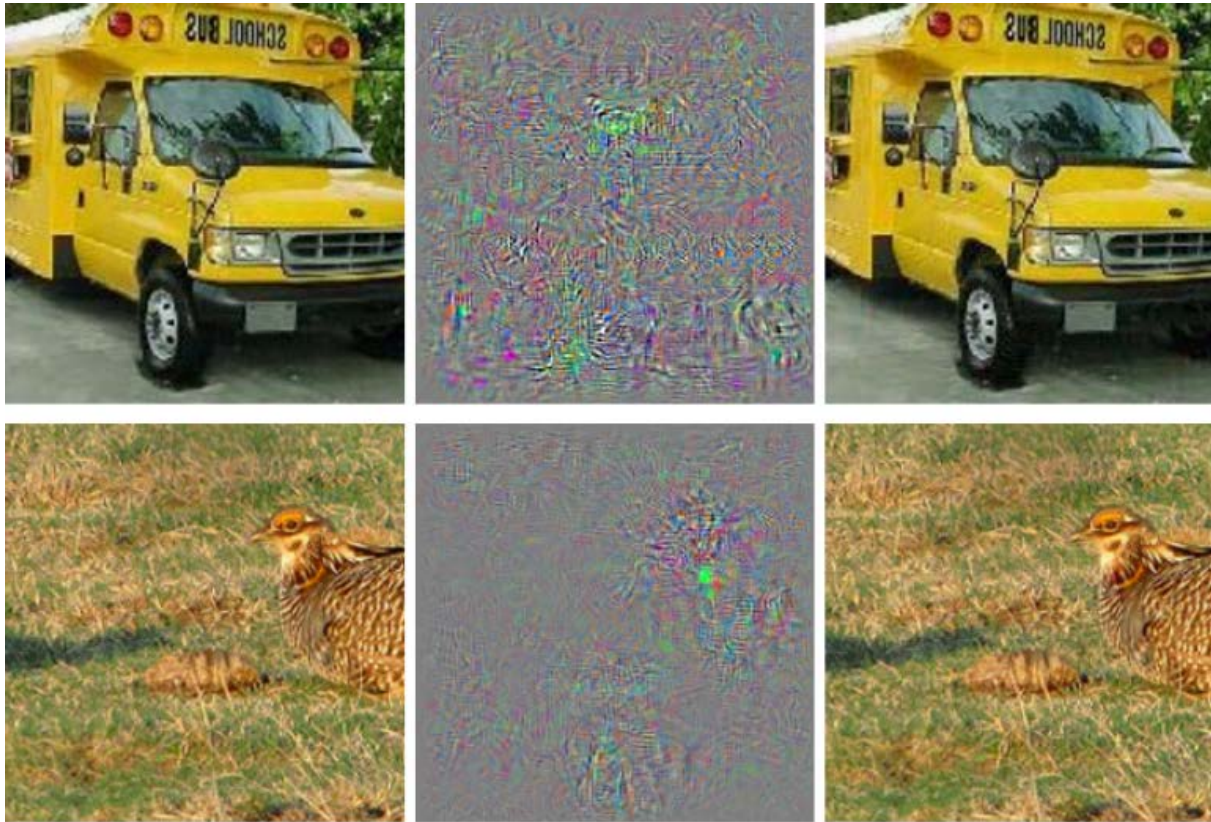
- keine Erkennung oder falsche Sicherheit
- generell bei Topologien und Daten: übertragbar

	x gehört zu $K^1$	x gehört nicht zu $K^1$
x ist K zugeordnet	richtig positiv	<b>falsch positiv</b>
x ist K nicht zugeordnet	<b>falsch negativ</b>	richtig negativ

<sup>1</sup> Verifikation „x gehört zu K“ wird vom Menschen getroffen

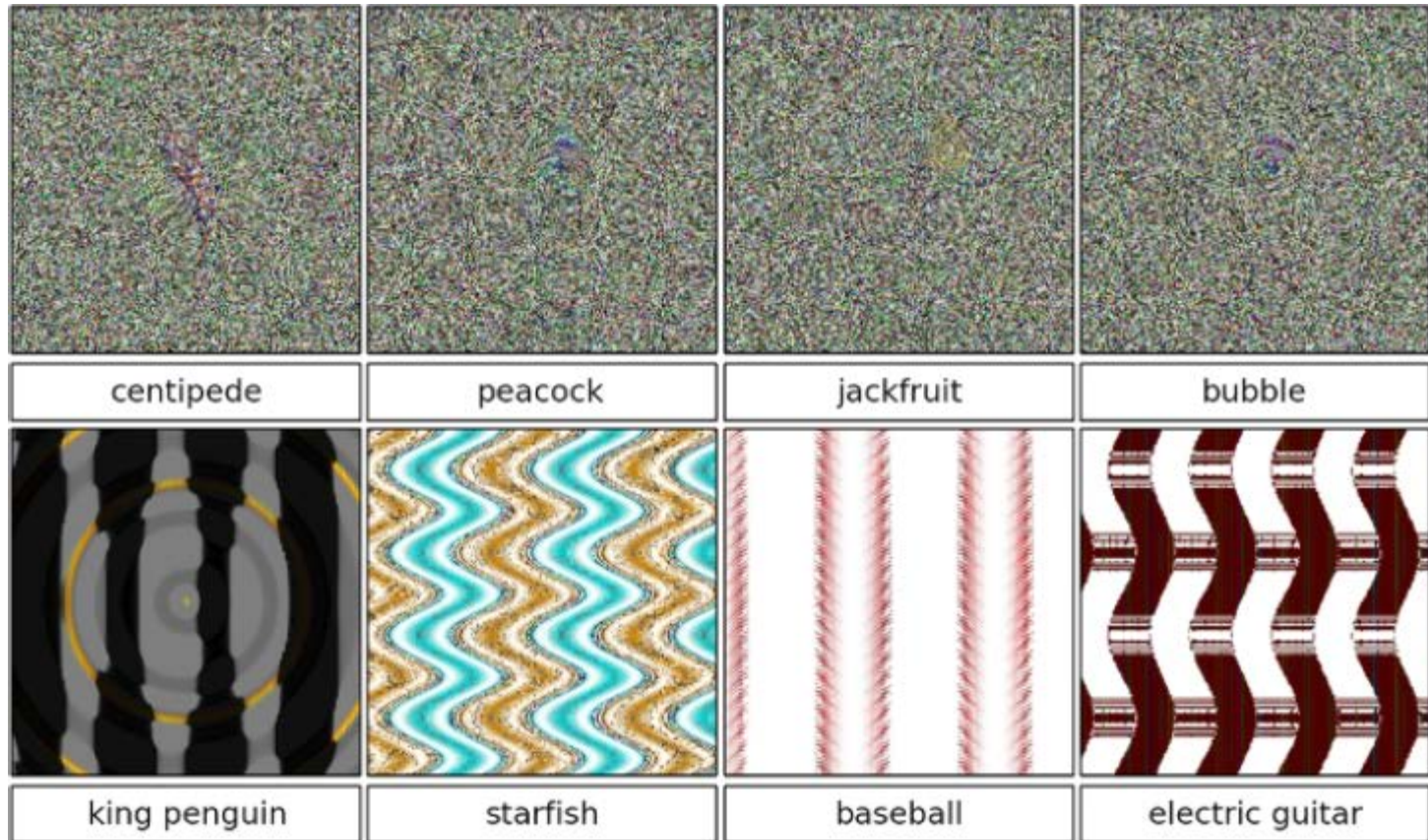


## Probleme: falsch negative Klassifikation



Bildquelle: C. Szegedy et al., "Intriguing properties of neural networks", Februar 2014, S. 6

## Probleme: falsch positive Klassifikation



Bildquelle: A. Nguyen, J. Yosinski, and J. Clune, Deep neural networks are easily fooled: High confidence predictions for unrecognizable images, **Dezember** 2014, S. 1

## Fazit

- Falsifizierung der bisherigen Arbeiten und Ergebnisse
- Extrahierte Merkmale  $\neq$  logische Struktur
- Fehler leicht zu produzieren und provozieren
- Praxiseinsatz von KNN schwierig, wenn
  - verhindern der Erkennung (falsch negativ)
  - falsche Sicherheit (falsch positiv)

gegeben

**Vielen Dank für Ihre  
Aufmerksamkeit!**