

Maschinelles Lernen: Neuronale Netze

Ideen der Informatik

Kurt Mehlhorn

Adrian Neumann



mp max planck institut
informatik

16. Januar 2014

- Biologische Inspiration
- Stand der Kunst in Objekterkennung auf Bildern
- Künstliche Neuronale Netze
 - Künstliche Neuronen
 - Neuronale Netze
 - Realisierung von Und, Oder und Negation
- Trainieren von Neuronalen Netzwerken
 - Prinzip
 - Beispiel: Exklusives Oder
 - Buchstaben
- Bildererkennung (Deep Convolutional Networks)

Stand der Kunst: Klassifikation (Krizhevsky et al, 2012)



mite

container ship

motor scooter

leopard

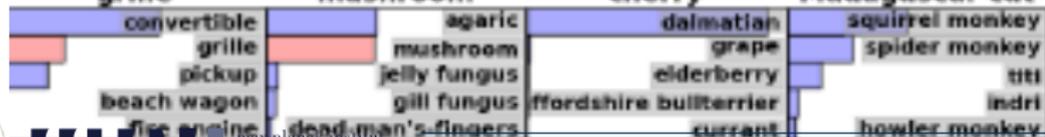


grille

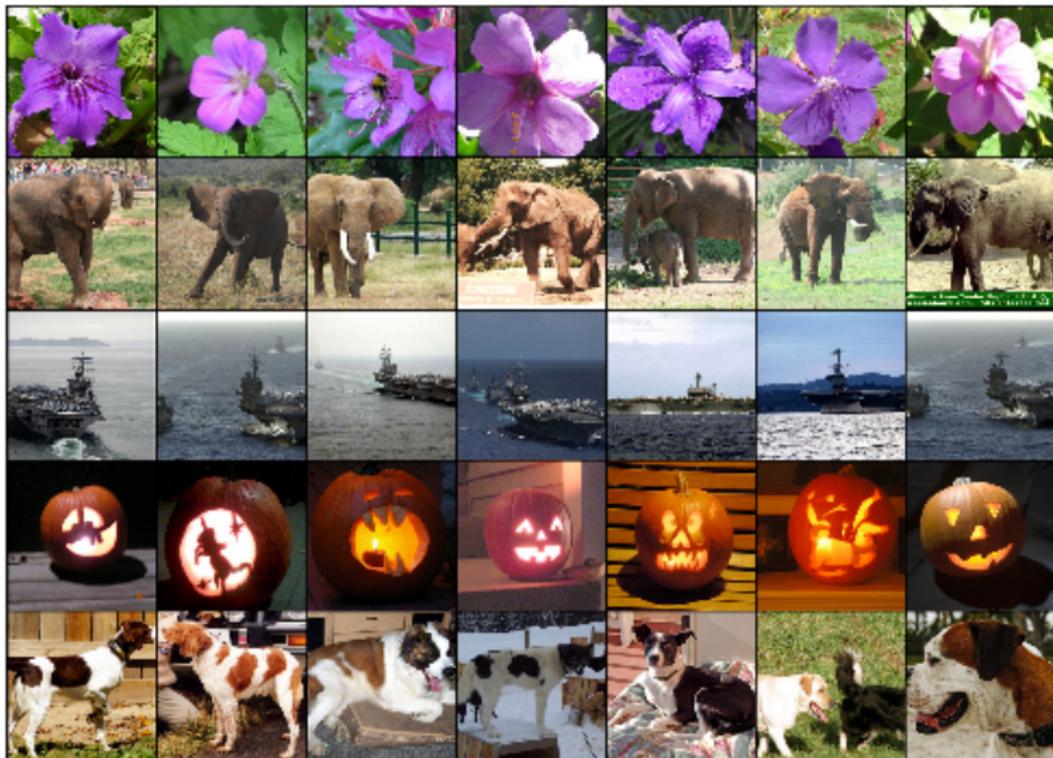
mushroom

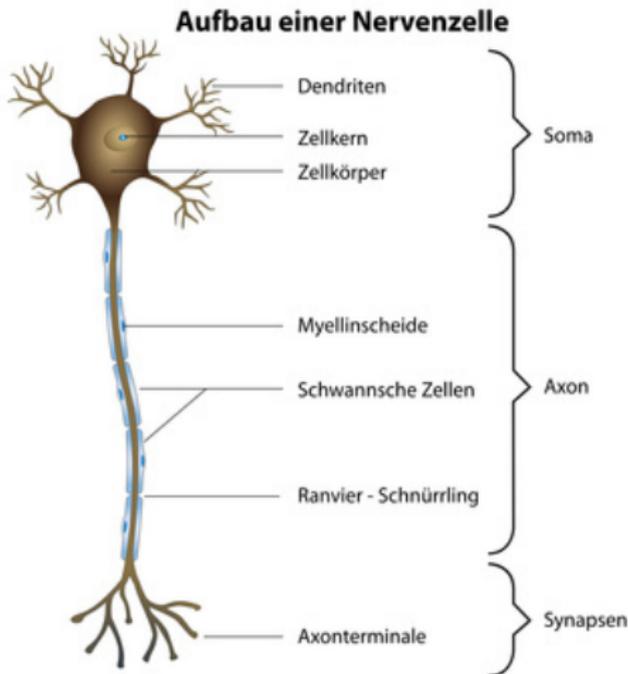
cherry

Madagascar cat



Stand der Kunst: Suche (Krizhevsky et al, 2012)





Axonterminale = Input

Dendriten = Output

Zelle feuert, wenn
Gesamterregung
einen Schwellwert
übersteigt

Inputs können auch
hemmend sein.

Visueller Kortex ist schichtenweise aufgebaut; 6 Schichten.

Neuronen der ersten Schicht bekommen Input von einem kleinen Feld von Sehzellen, Neuronen höherer Schichten von einem kleinen Feld der davorliegenden Schicht.

One-Learning Algorithm Hypothese

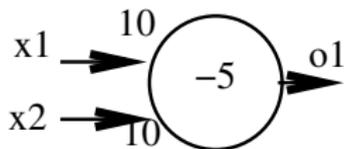


- Neuron hat k eingehende Kanten.
- Eingabewerte x_1 bis x_k liegen zwischen 0 und 1.
- $k + 1$ Gewichte (Parameter) w_0, w_1, \dots, w_k .
- w_0 heißt Grundgewicht (Bias); w_i ist die Wichtung von x_i .
- Ausgabe = $g(w_0 + w_1 x_1 + \dots + w_k x_k)$.
- g = Sigmoid Funktion.
- Sigmoid Funktion ist differenzierbare Approximation einer Stufenfunktion.

$$g(z) = \frac{1}{1 + e^{-z}}$$

- $g(0) = 1/2$.
- $g(1) = 0.73$
- $g(4) = 0.95$
- symmetrisch zu $(0, 1/2)$.
- differenzierbare Approximation einer Stufe von 0 nach 1 an der Stelle 0.
- wenn man e^{-z} durch e^{-10z} ersetzt, wird Flanke steiler.

Realisierung von Oder und Nund (Negiertem Und)



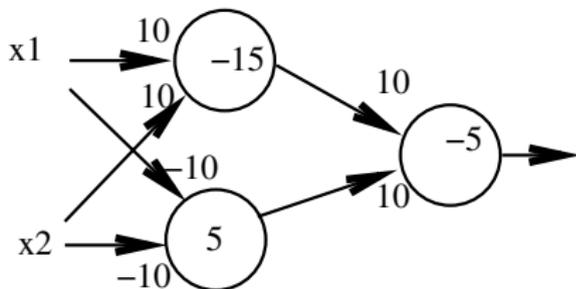
$$o_1 = g(-5 + 10x_1 + 10x_2) \approx x_1 \vee x_2$$

$$o_2 = g(15 - 10x_1 - 10x_2)$$

x_1	x_2	$o_1 =$	$o_1 \approx$	$o_2 =$	$o_2 \approx$
0	0	$g(-5)$	0		
0	1				
1	0				
1	1				

Komplexeres Beispiel

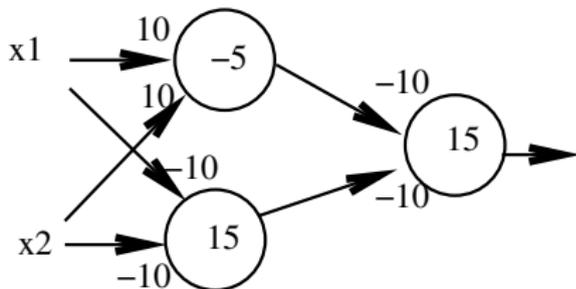
Welche Boolesche Funktion wird durch dieses Neuronale Netz berechnet?



x_1	x_2	$z_1 =$	$z_1 \approx$	$z_2 =$	$z_2 \approx$	$o =$	$o \approx$
0	0	$g(-15)$	0				
0	1						
1	0						
1	1						

Aufgabe

Welche Boolesche Funktion wird durch dieses Neuronale Netz berechnet?

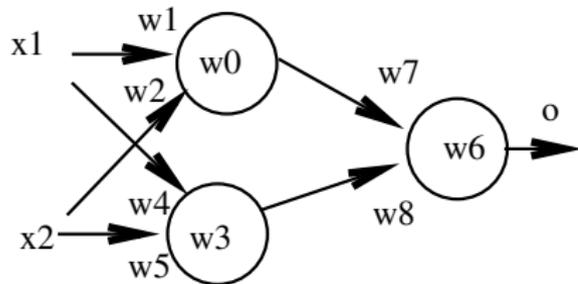


x_1	x_2	$z_1 =$	$z_1 \approx$	$z_2 =$	$z_2 \approx$	$o =$	$o \approx$
0	0	$g(-5)$	0				
0	1						
1	0						
1	1						

Training (Rummelhart/Hinton 86)

Bisher: Wir haben bestimmt, welche Funktion durch ein Netz mit gegebenen Parameterwerten ausgerechnet wird.

Nun: Wir möchten, dass das Netz eine bestimmte Funktion realisiert. Wie müssen wir die Parameter einstellen?



x_1	x_2	o
0	0	1
0	1	0
1	0	0
1	1	1

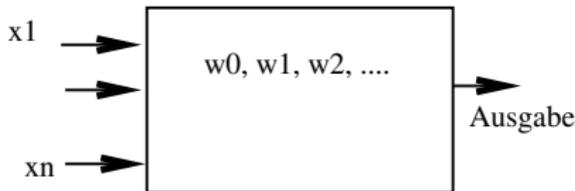
Aufgabe: finde w_0 bis w_8 , so dass die gegebene Funktion berechnet wird.

Sprechweise: das Netz „lernt“ die Parameter.

Jedes Trainingsbeispiel (für ein Netz mit n Eingaben und einem Ausgang) besteht aus einem Eingabevektor $x = (x_1, \dots, x_n)$ und einer Ausgabe y . Alle Werte liegen zwischen 0 und 1.

W = Menge aller Parameter (alle w 's).

$h_W(x)$ = Ausgabe (Hypothese) des Netzes mit Parametersatz W bei Eingabe $x = (x_1, \dots, x_n)$



Training soll w 's finden, so dass Hypothesen des Netzes und korrekte Ausgaben möglichst gut übereinstimmen.

Was heißt gut übereinstimmen?

Training III

Fehler am Trainingsbeispiel (x, y) : $(y - h_W(x))^2$

Gesamtfehler E = Summe der Einzelfehler über alle Trainingsbeispiele

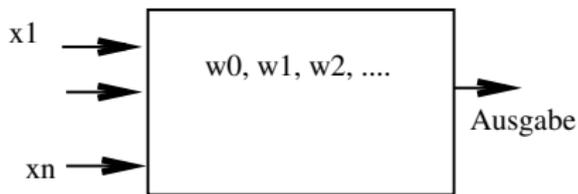
Ziel des Trainings ist die Bestimmung eines Parametersatzes, der den Gesamtfehler klein macht.

Bestimmung von (lokalen) Minima durch Iterationsverfahren

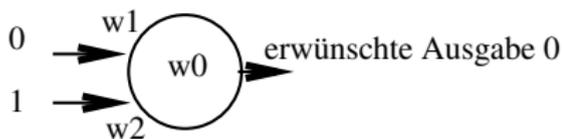


Der Trainingsalgorithmus (Back Propagation)

- 1) Initialisiere die Parameter mit kleinen zufälligen Werten
- 2) Solange nicht zufrieden, d.h. Gesamtfehler E zu groß
 - für jeden Parameter tue: ändere Parameter ein bißchen, so dass Gesamtfehler kleiner wird.



Beispiel für Trainingsschritt



Parameterwerte: $w_0 = 0.2$, $w_1 = 0.1$ und $w_2 = -0.2$.

Trainingsbeispiel: an Eingabe $(0, 1)$ soll die Ausgabe 0 sein. Im allgemeinen gibt es viele Trainingsbeispiele.

Das Netz gibt aus:

$$h_W(0, 1) = g(0.2 + 0.1 \cdot 0 - 0.2 \cdot 1) = g(0) = 1/2.$$

Um den Fehler zu reduzieren, sollten wir $h_W(0, 1)$ verringern. Dazu müssen wir

$$z = w_0 + 0 \cdot w_1 + 1 \cdot w_2$$

verringern. Also w_0 und w_2 etwas verringern und w_1 gleich lassen.

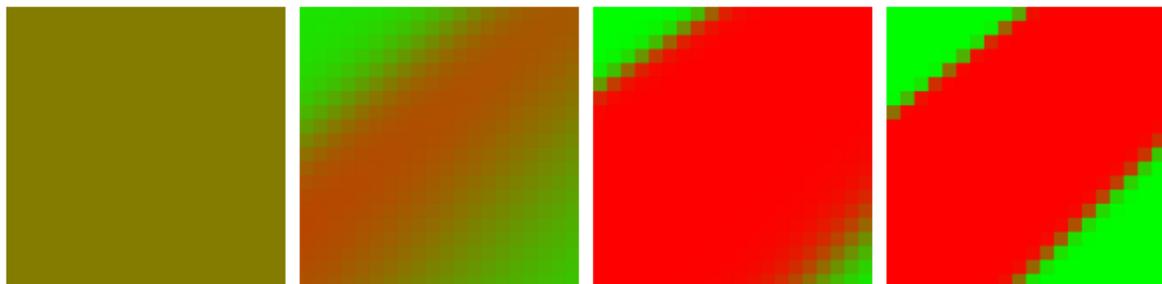
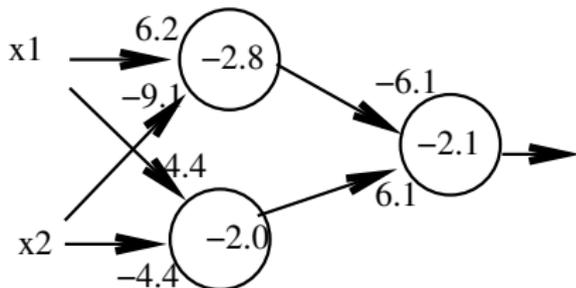
Beispiel für das Ergebnis eines Trainings

16 Trainingsbeispiele:

x_1 und x_2 in $\{0, 0.1, 0.9, 1\}$

$y = 1$ genau wenn $x_1 \approx x_2$

In 50 Tausend Iterationen wurden folgende Parameter bestimmt.



Unterscheide T und C

Aufgabe: in einem 25 x 25 Pixelbild befindet sich ein T oder C in einer der vier möglichen Orientierungen; fünf Pixel sind Eins, die anderen sind Null.

Bestimme ein neuronales Netz (einfacher Architektur), das T und C unterscheidet.

einfach: alle Neuronen der ersten Schicht schauen sich eine 3 x 3 Matrix an und sind identisch, d.h. haben die gleichen Parameter.

Ausgabeneuron hängt von allen Neuronen der ersten Schicht ab.

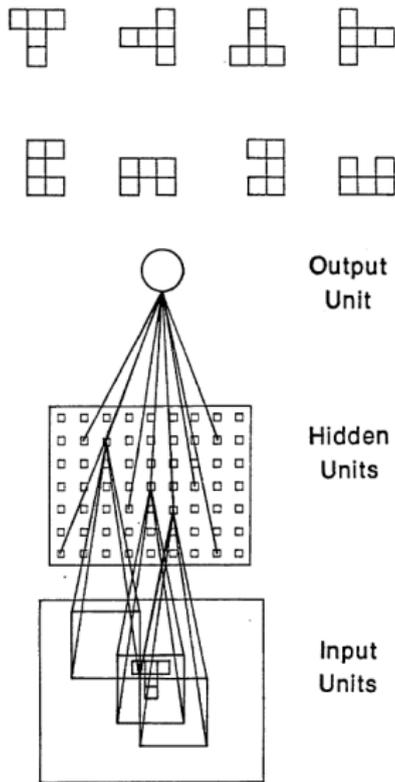
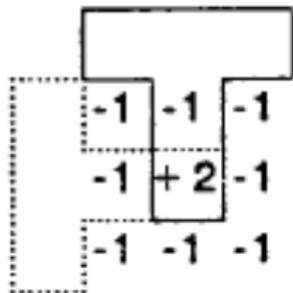


FIGURE 14. The network for solving the T-C problem. See text for explanation.

A



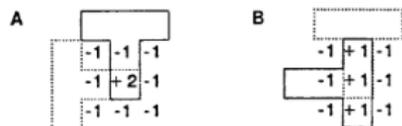
Wenn die Mitte des Filter auf dem Fuß des T's liegt, liefert der Filter eine +1.

Bei einem C erzeugt der Filter immer einen Wert kleiner gleich 0.

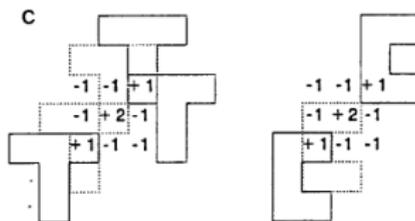
Neuron der ersten Schicht =

$$g(-5 + 20 \cdot \text{Mittelpixel} - 10 \cdot \text{Summe der Randpixel})$$

Ausgabeneuron = Oderverknüpfung aller Neuronen der ersten Schicht



Training fand vier verschiedene Lösungen.



Aufgabe: finde heraus, wie B und C funktionieren.

D ist besonders interessant: es funktioniert weil ein C 20 Rezeptoren überlappt und ein T 21 Rezeptoren.

D -2 -2 -2
-2 -2 -2
-2 -2 -2

0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	1	1	1	0
0	1	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0

- 1.2 Millionen Trainingsbeispiele aus 1000 Klassen.
- Eingabe sind 224 x 224 Pixel jeweils mit 3 Farbwerten (0 bis 255)
- Ausgabeschicht hat 1000 Neuronen, eins pro Klasse
- Netzwerk hat 8 Schichten, 650000 Neuronen, 60 Millionen Parameter
- die ersten 5 Schichten sind konvolutional
 - konvolutional: Kopien des gleichen Neurons schauen auf Teilbilder
- Training dauerte 1 Woche
-

Klassifikation: Rechtecke = Erregung der Ausgabeneuronen



mite

container ship

motor scooter

leopard

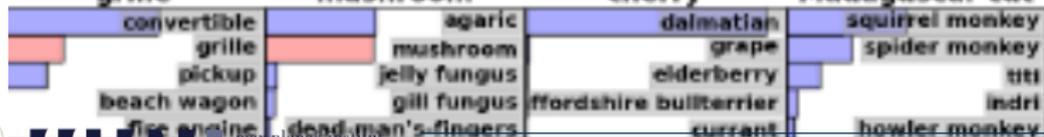


grille

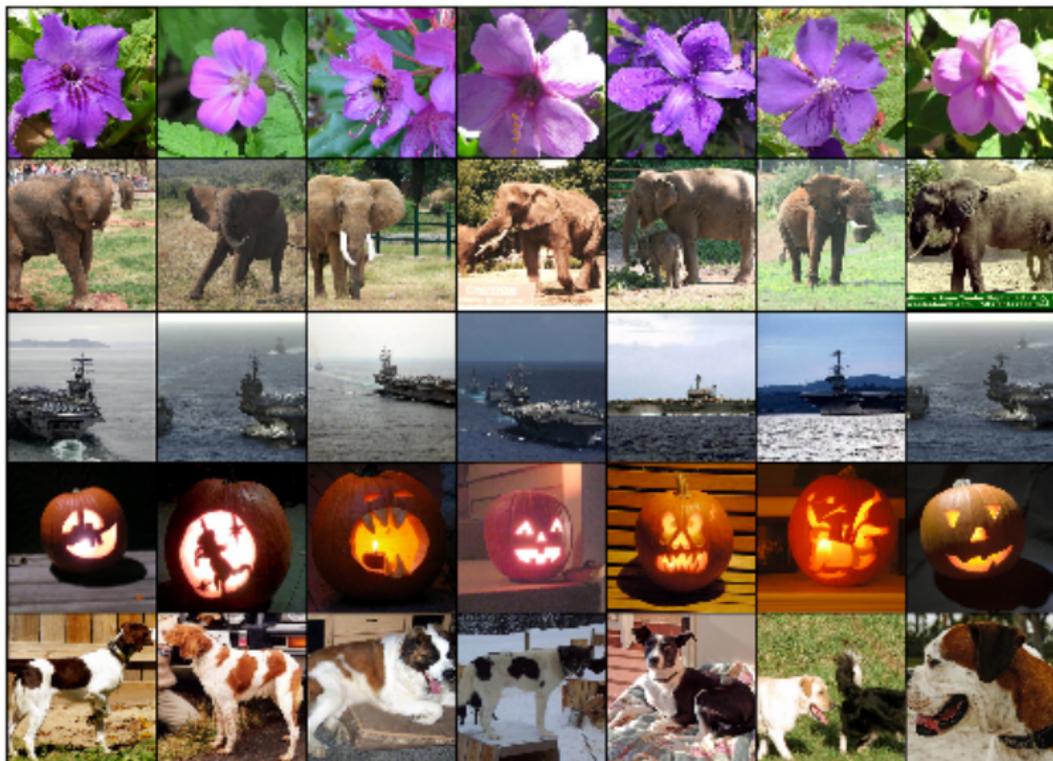
mushroom

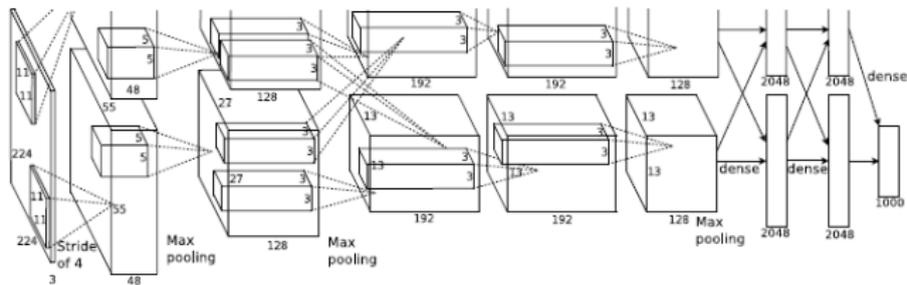
cherry

Madagascar cat

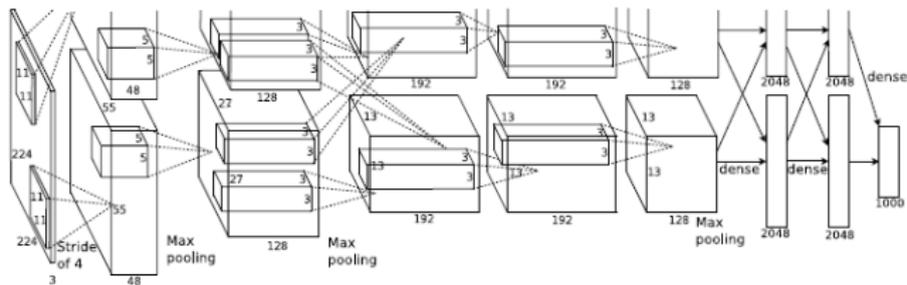


Suche: Suchbild in Spalte 1, Trainingsbilder mit ähnlichster Erregung der Ausgabeneuronen





- Eingabe: 224 x 224 mit jeweils 3 Farbwerten
- Schicht 1: 96 verschiedene Neuronen in 55 x 55 Matrix; jedes Neuron sieht 11 x 11 Feld der Eingabe; Shift von 4.
- Schicht 2: 256 verschiedene Neuronen in 27 x 27 Matrix; jedes Neuron sieht 5 x 5 Feld der Schicht 1, also 31 x 31 Feld der Eingabe.
- Schicht 3, 4, 5: ähnlich
- Schichten 6, 7: 4096 Neuronen, sehen ganze vorherige Schicht
- Schicht 8, Ausgabeschicht: 1000 Neuronen



- Eingabe: 224 x 224 mit jeweils 3 Farbwerten
- Schicht 1: 96 verschiedene Neuronen in 55 x 55 Matrix; jedes Neuron sieht 11 x 11 Feld der Eingabe; Shift von 4.
- Schicht 2: 256 verschiedene Neuronen in 27 x 27 Matrix; jedes Neuron sieht 5 x 5 Feld der Schicht 1, also 31 x 31 Feld der Eingabe.
- Informationsreduktion:
 - 11 x 11 Fenster der Eingabe, 3 Farbwerte: 363 Zahlen
 - erste Schicht: Ausgabe von 96 Neuronen, 96 Zahlen

Wie funktioniert das?

Neuronen der ersten Schicht entdecken Kanten, Linien, Bögen in 11 x 11 Feldern der Eingabe.

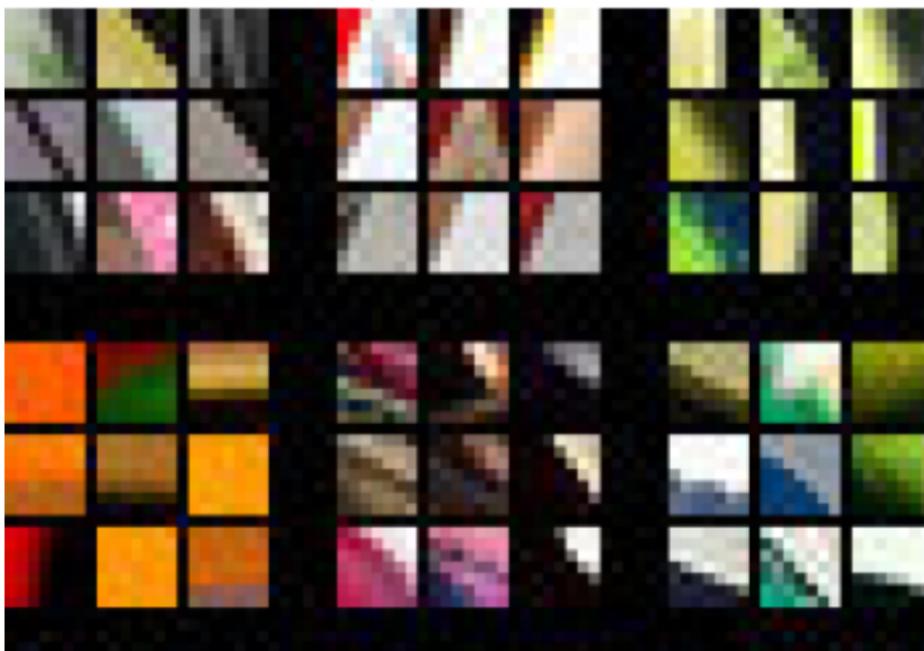


Bild zeigt Eingaben, bei denen 4 ausgewählte Neuronen der ersten Schicht besonders stark ansprechen.

Wie funktioniert das?

Neuronen der zweiten Schicht entdecken komplexere Merkmale in 31 x 31 Feldern der Eingabe.

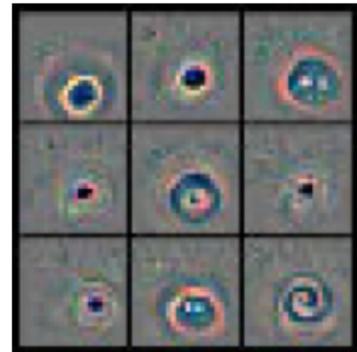
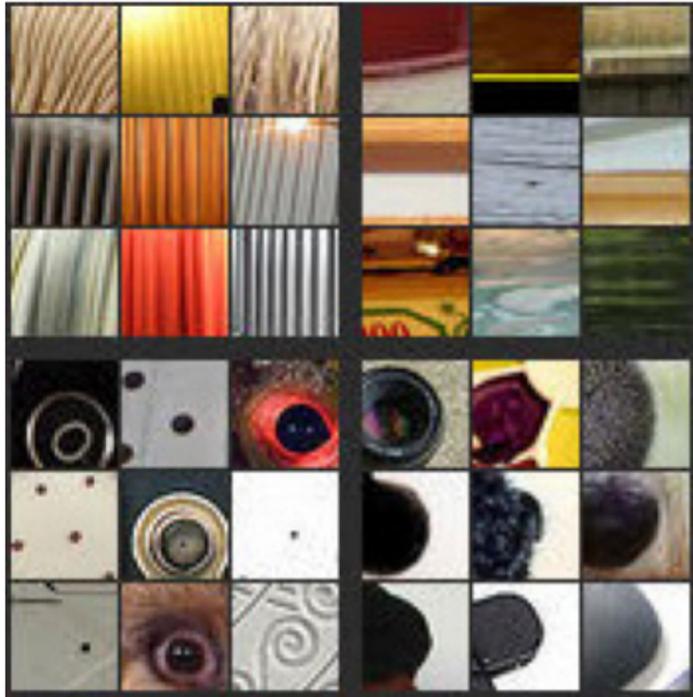


Bild zeigt Eingaben, bei denen 4 ausgewählte Neuronen der zweiten Schicht besonders stark ansprechen.

Wie funktioniert das?

Neuronen der dritten Schicht entdecken noch komplexere Merkmale in wiederum größeren Feldern der Eingabe.

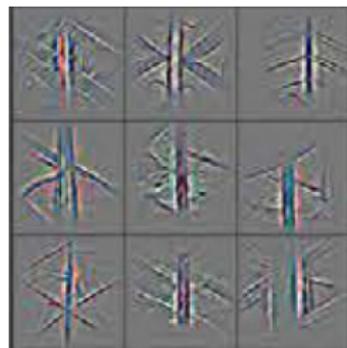
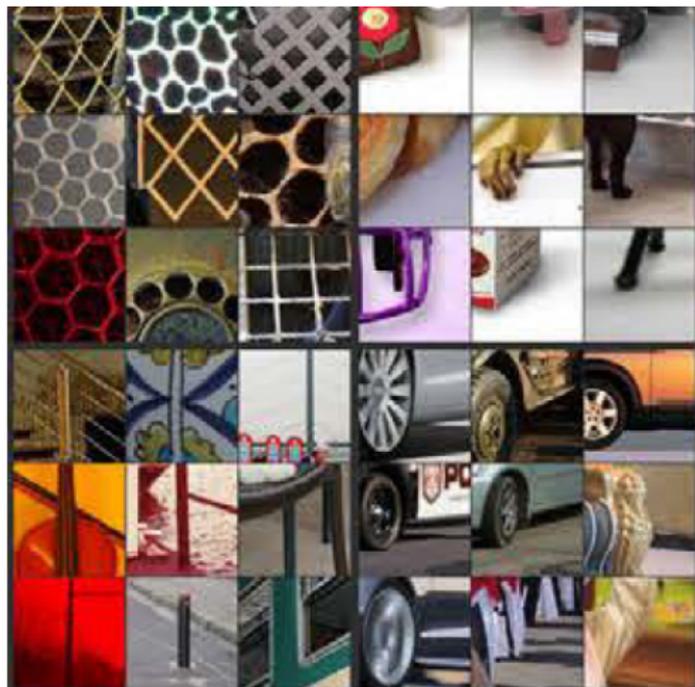


Bild zeigt Eingaben, bei denen 4 ausgewählte Neuronen der dritten Schicht besonders stark ansprechen.

Wie funktioniert das?

Neuronen der vierten Schicht entdecken noch komplexere Merkmale in wiederum größeren Feldern der Eingabe.

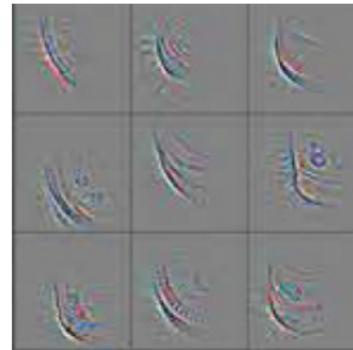


Bild zeigt Eingaben, bei denen 4 ausgewählte Neuronen der vierten Schicht besonders stark ansprechen.

Wie funktioniert das?

Neuronen der fünften Schicht entdecken noch komplexere Merkmale in wiederum größeren Feldern der Eingabe.

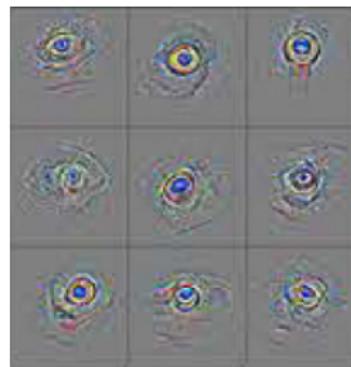


Bild zeigt Eingaben, bei denen 4 ausgewählte Neuronen der fünften Schicht besonders stark ansprechen.

Neuronale Netze mit vielen Schichten (deep networks) haben Durchbruch in Bilderkennung, Handschriftenerkennung, und Spracherkennung geschafft.

Je höher die Schicht, desto komplexere Merkmale werden erkannt; Merkmale auf einer Schicht sind Kombinationen von Merkmalen auf der vorherigen Schicht.

Training braucht sehr große Datensätze, die seit einigen Jahren durch soziale Netzwerke und Crowdsourcing zur Verfügung stehen.

Training ist aufwendig und dauert lange, aber das ist auch bei Gehirnen so (meine 18 Monate alten Enkel beginnen gerade die Verbindung zwischen dem Wort Elefant und dem Bild eines Elefanten herzustellen).