

Meilensteine der neuronalen Netze

1 Anfänge

Die Anfänge neuronaler Netze gehen auf Warren McCulloch und Walter Pitts zurück. Diese beschrieben 1943 ein neurologisches Netz, das jede logische und arithmetische Funktion berechnen kann. 1947 wiesen sie darauf hin, dass ein solches Netz zur räumlichen Mustererkennung eingesetzt werden kann.

- 1949 formulierte Donald O. Hebb seine Hebb'sche Lernregel, die in ihrer allgemeinen Form fast allen neuronalen Lernverfahren zugrunde liegt. Karl Lashley kam 1950 zu der These, dass der Prozess der Informationsspeicherung verteilt im Gehirn realisiert wird.

KÜNSTLICHE NEURONALE NETZWERKE

Das Gehirn im Computer

Künstliche neuronale Netze ahmen die Informationsverarbeitung des Gehirns im Computer nach. Das ist nicht etwa nur interessant für Biologen oder Neurowissenschaftlicher. Inzwischen haben neuronale Netze einen hohen praktischen Nutzwert – und glänzen besonders dann, wenn es gilt Neues zu lernen.

VON DR. KLAUS MANHART

Neuronale Netze retten bei der Minenräumung Leben, helfen bei Wettervorhersage und Klimaforschung und warnen rechtzeitig, bevor Kunden abwandern.

Wo normale Computer-Algorithmen versagen, müssen neuronale Netze ran. Dann etwa, wenn nur wenig explizites Wissen über ein Problem vorliegt. Oder wenn es nur aufwendig oder gar nicht zu formalisieren ist. Auch wenn Rechner lernen sollen oder der Datenwust so groß ist, dass kein Forscher mehr durchblickt, schlägt die Stunde der neuronalen Netze. Vor allem Klassifizierung, Vorhersage und Mustererkennung sind die klassischen Einsatzgebiete von Neuronetzen. Ohne dass man explizite Vorschriften programmieren müsste, ist so ein neuronales Netz etwa in der Lage, Tendenzen aus Börsenkursen abzulesen, Personen zu klassifizieren oder Gesichter zu erkennen.

Netze, die selbstständig Muster erkennen, können sogar helfen, Leben zu retten. Forscher der University of Liverpool trainierten z. B. ein neuronales Netz darauf, die Muster von Landminen in den Daten von Radar und Akustiksensoren – mit hoher Rate – zu erfassen. Beim Deutschen Krebsforschungszentrum spürt ein neuronales Netz im menschlichen Erbgut wichtige Gensequenzen auf. An der Universität Frankfurt wird die Technologie auf das Wetter und zentrale Fragen der Klimaforschung angewendet. Aber auch bei ganz praktischen Problemen sind Neuronetze nützlich. Schon lange helfen sie beim Data Mining, wenn es gilt, in ei-

ner Vielzahl von Daten Regelmäßigkeiten zu entdecken. So sind Unternehmen etwa in der Lage, mit neuronalen Netzen aus bereits abgewanderten Kunden Merkmale zu extrahieren, die diese gemeinsam haben. Selektiert man über diese Faktoren Kunden des Kundenbestands, erhält man die abwanderungsgefährdeten Kunden. Diese können dann gezielt „umsorgt“ werden.

Auch in alltäglichen Anwendungen lassen sich die künstlichen Netze effektiv einsetzen. Die Virens Scanner von Bitdefender nutzen seit Jahren neuronale Netze, um Spam abzuwehren. Das lernende Netz wird in den Laboren trainiert, indem viele Nachrichten damit analysiert werden. Ähnlich einem Kind in der Schule lernt der Filter, Spam-Mails und erlaubte Mails zu unterscheiden. Die empfangenen Mails werden anhand der Ähnlichkeiten untersucht, auf die der Filter trainiert worden ist.

Vorbild Gehirn

Das Faszinierende und Neue an neuronalen Netzen ist, dass sie ohne explizit vorgegebene Regeln neues Wissen lernen können. Ein kurzes Training und das Füttern des Netzes mit Beispielen genügen. Nach dem Lernen können die Ergebnisse auch für nicht Gelerntes abgerufen werden. Dieses Lernen an Beispielen erspart eine aufwendige Erstellung von Regeln und Algorithmen.

2 Blütezeit

- Von 1957-1958 entwickelten Frank Rosenblatt und Charles Wightman den ersten erfolgreichen Neurocomputer, mit dem Namen Mark I Perceptron. Basis war ein einfaches Perzeptron aus einem einzelnen künstlichen Neuron mit anpassbaren Gewichtungen und einem Schwellenwert.
- 1960 stellten Bernhard Widrow und Marcian E. Hoff das ADALINE (ADaptive Linear Neuron) vor. Dieses Netz erreichte als erstes weite kommerzielle Verbreitung. Das neuronale Netz lernte mit der Delta-Regel.

3 Zusammenbruch

- 1969 gaben Marvin Minsky und Seymour Papert eine genaue mathematische Analyse des Perceptrons. Sie zeigten auf, dass wichtige Probleme nicht gelöst werden können. Die Folge war ein vorläufiges Ende der Forschungen.

4 Wiederaufbau und Renaissance

- 1982 beschrieb Teuvo Kohonen die nach ihm benannten selbstorganisierenden Karten. Im selben Jahr stellte John Hopfield das Modell der Hopfield-Netze vor.
- 1985 veröffentlichte John Hopfield eine Lösung des Travelling-Salesman-Problems durch das Hopfield-Netz.

5 Jetztzeit

- In den Neunzigerjahren wurden künstliche neuronale Netze der zweiten Generation in Wissenschaft und Technik gut eingeführt und weit verbreitet. Aktuell beschäftigt sich die Wissenschaft vor allem mit sogenannten „Spiking neural networks“.

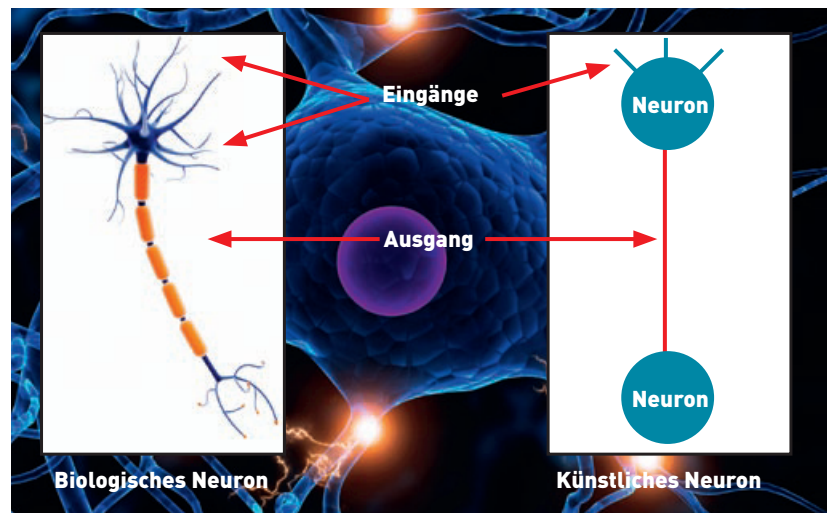
Vorbild für diese Art des maschinellen Lernens ist das menschliche Gehirn. Der modernen Hirnforschung zufolge sind die Inhalte unseres Denkapparats in neuronalen Netzen gespeichert – daher auch der Name. Dieses biologische Netz im menschlichen Kopf besteht Schätzungen zufolge aus 100 Milliarden von Nervenzellen – den Neuronen. Sie sind darauf spezialisiert, Informationen als elektrische Impulse zu empfangen und an andere Zellen weiterzuleiten. Um den Informationsaustausch zu gewährleisten, sind die Zellen über Nervenstränge, die Synapsen, eng miteinander verbunden. Durch insgesamt 100 Billionen Synapsen entsteht ein hochkomplexes neuronales Netzwerk, bei dem jedes Neuron im Schnitt mit 1000 anderen verbunden ist. Während Neuronen sehr einfach aufgebaut und zu beschreiben sind, führt ihr Zusammenwirken zu sehr komplexen Phänomenen. Wenn wir etwas lernen, werden die Verbindungen zwischen den Nervenzellen, die für den Lernvorgang nötig sind, verändert. Manche dieser Synapsen werden gestärkt, andere geschwächt. Die Anpassung ist entscheidend für das Funktionieren eines Netzwerks und des zentralen Mechanismus, über den wir in der Lage sind, zu lernen und Probleme zu lösen. Mit dem Computer lässt sich dieser Vorgang in Form von ähnlichen Modellen simulieren. Weil das Trainieren des Netzes viel Zeit kostet, beschränken gängige Software-Implementierungen die Zahl der Neuronen allerdings auf ein paar Dutzend bis einige Hundert.

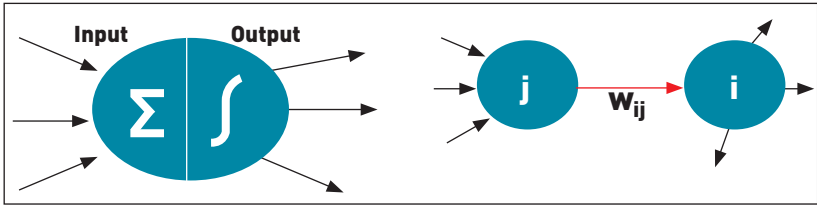
Das Neuron – Elementarteilchen des Hirns

Auch das künstliche neuronale Netz besteht aus Neuronen und Nervenfasern – im Computer sind die Neuronen einfache Rechenelemente, die auch Units, Einheiten oder Knoten heißen. Sie nehmen Informationen aus der Umwelt oder von anderen Units auf (Input) und geben sie in modifizierter Form weiter (Output). Wie ein natürliches Neuron können künstliche Neurone Impulse aussenden („feuern“) und empfangen. Jedes

Neuron errechnet dann aus den Inputs der anderen Neuronen, ob es im nächsten Zeitschritt aktiv oder nicht aktiv sein wird. Die Verbindungen zwischen den Units, die Kanten, können dabei wie im natürlichen Vorbild stärker oder schwächer sein. Diese Verbindungsstärke wird durch ein Gewicht ausgedrückt. Je größer das Gewicht, desto größer ist der Einfluss einer Einheit auf eine andere Einheit. Ein positives Gewicht übt auf ein anderes Neuron einen erregenden, verstärkenden Einfluss aus, ein negatives einen hemmenden. Der zweite Faktor, der den Einfluss auf ein Neuron festlegt, ist die Signalstärke des sendenden Neurons. Beide Werte – Gewicht und Signalstärke – werden dann miteinander multipliziert: Je stärker also das Aktivitätslevel der sendenden Einheit und je höher das Gewicht der Verbindung zwischen den beiden Neuronen ist, umso höher ist der Input auf das empfangende Neuron. Da ein Neuron den Input von mehreren Neuronen empfängt, muss nun nur noch der Gesamt-Input berechnet werden. Der ergibt sich im einfachsten Fall durch Addi-

Natürliche und künstliche Neuronennetze: Künstliche Netze ahmen Grundprinzipien ihrer biologischen Vorbilder nach.





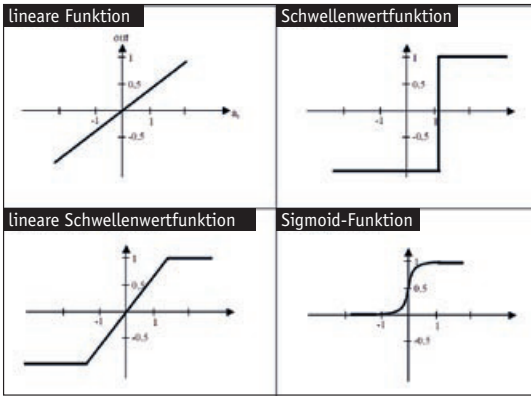
Die Komponenten des Neurons (links): Ein Neuron hat vier Komponenten. (1) Im Input-Teil wird ein Neuron durch andere Neurone aktiviert, (2) die Summationsfunktion summiert alle Inputs zu einem Aktivierungspotenzial auf, (3) die Schwellwertfunktion konvertiert das Aktivierungspotenzial in ein Ausgangsmaß und (4) im Output-Teil wird die Aktivierung an andere Neurone weitergegeben. Die Neuronen-Verbindung (rechts): Zwischen den Neuronen bestehen Verbindungen mit unterschiedlichen Gewichten.

tion der Einzel-Inputs. Das Ergebnis ist das Aktivitätslevel oder die Gesamterregung, die sich durch den Input aller Eingabeneurone ergeben hat. Ein Beispiel: Angenommen, Neuron j im Bild hat eine Aktivierung von 1, das Gewicht der Verbindung zwischen den beiden Units i und j betrage 3. Der Input der Einheit i ist dann $1 \times 3 = 3$. Bestünden noch zwei weitere Kanten von j zu i mit einem Input von 1 bzw 2 wäre der Gesamtinput für i die Summe $3 + 1 + 2 = 6$.

Wann Neuronen feuern

Der Input von anderen Zellen führt also in jedem Neuron zu einer Reizung oder Aktivierung. Diese muss aber nicht zu einer Reaktion führen. Jede Zelle besitzt eine Aktivierungs- oder Transferfunktion. Diese entscheidet, ob und wie stark ein Signal ausgesendet wird. Sie spezifiziert also, was das Neuron mit den Signalen tut. Meist hat die Transferfunktion einen Schwellenwert. Ob das empfangende Neuron auf den eingehenden Input reagiert, hängt maßgeblich vom Schwellenwert dieses Neurons ab. Liegt die Gesamterregung des Neurons über dieser Schwelle, feuert das Neuron. Die Schwellenwert-Funktion konvertiert also das Maß der Aktivierung in ein Ausgangsmaß. Sie kann im einfachsten Fall binär sein und nur die Werte 0 (nicht aktiv) und 1 (aktiv) annehmen: Sie wechselt am Schwellenwert von einem Wert zum anderen. Eine lineare Schwellenwertfunktion kann sinnvoll sein, wenn ein niedriger Input nicht als Signal weitergeleitet werden soll. Läge der Schwellenwert von Zelle i im obigen Beispiel bei 1 – wie in der Grafik – würde das Neuron also feuern, weil der Gesamtinput 6 größer ist als der Schwellenwert. Im Fall einer binären Funktion würde Z mit Stärke 1 feuern. Diese Aktivität kann nun wieder als Eingangssignal für andere Neuronen dienen, die daraus ihren Zustand im darauf folgenden Zeitschritt berechnen. Alles was bisher beschrieben wurde, findet parallel statt. Dies macht neuronale Netze sehr schnell.

Aktivierungsfunktion: Die vier häufigsten Aktivierungsfunktionen. Auf der x-Achse ist der Aktivitätslevel der Zelle abgetragen, auf der y-Achse der abgegebene Output.

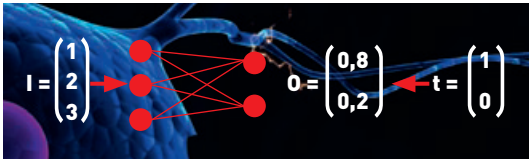


Lernen und Trainieren

Lernen erfolgt in neuronalen Netzen in der Regel durch Veränderung der Gewichte zwischen den Neuronen. Lernregeln geben an, wie das Netz lernen soll, für eine vorgegebene Eingabe eine gewünschte Ausgabe zu produzieren. Zunächst lernt das Netz in der Trainingsphase anhand des vorgegebenen Materials. Dabei unterscheidet man grundsätzlich überwachtes („supervised“) und nicht-überwachtes („unsupervised“) Lernen. Beim nicht überwachten Lernen wird kein Output vorgegeben, die Gewichtsveränderung erfolgt hier abhängig von der Ähnlichkeit der Gewichte mit den Inputs. Beim überwachten Lernen gibt man dem Netz Beispiele in beliebiger Reihenfolge und wiederholt das Ganze. Für jedes Beispiel ist bekannt, was die gewünschte Ausgabe sein soll. Stimmt die Ausgabe des Netzes für ein Beispiel mit der gewünschten überein, dann braucht nichts getan zu werden. Weichen tatsächliche und gewünschte Ausgabe voneinander ab, dann müssen die Gewichte im Netz so verändert werden, dass sich der Fehler bei der Ausgabe verringert. Dieser Prozess erfolgt im Idealfall so lange, bis alle Beispiele richtig berechnet werden. Der ganze Lernprozess ist also ein iterativer Vorgang, bei dem ein spezieller Algorithmus die Gewichte so einstellt, dass der Output möglichst genau dem bekannten Ergebnis entspricht. Ist die Trainingsphase abgeschlossen, erfolgt ein Test. Dieser überprüft auf Basis der modifizierten Gewichte aus dem Training, ob das Netz etwas gelernt hat. Dazu gibt man neue Input-Daten in das System ein und kontrolliert, welchen Output das Netz berechnet. Mit den neuen Daten kann man sehen, ob eine Generalisierung auf neue Reize funktioniert, ob das Netz also in der Lage ist, über die Lerndaten hinaus Aufgaben zu lösen. Um die Gewichte zu verändern, braucht man Lernregeln. Sie geben an, wie die Veränderungen genau vorgenommen werden sollen. Eine Lernregel ist die Hebb-Regel, benannt nach dem Psychologen Donald Hebb. Sie besagt: „Je häufiger ein Neuron A gleichzeitig mit Neuron B aktiv ist, umso bevorzugter werden die beiden Neuronen aufeinander reagieren („what fires together that wires together“). Das Gewicht zwischen zwei Units wird dann verändert, wenn beide Einheiten aktiv sind. Eine andere Regel lernt aus Fehlern und löscht zunehmend nicht erwünschtes Verhalten. Diese Delta-Regel beruht auf dem Vergleich zwischen dem gewünschten und dem tatsächlichen Output. Ist die Aktivität einer Einheit geringer als vom Trainer vorgesehen, werden die Gewichte zwischen sendender und empfangender Einheit erhöht. Hier wird ein errechneter Betrag zu den Verbindungsgewichten bei jedem Trainingsschritt hinzuaddiert. Ist die Aktivität hingegen zu groß, schwächt man alle positiven Inputs und verringert deren Gewicht, indem man einen bestimmten Betrag abzieht.

Wie Rotkäppchen lernt

Zur Demonstration abschließend ein einfaches Beispiel, mit dem die Psychologen Jones und Hoskins 1987 zeigten, wie ein neuronales Netz das Verhalten von Rotkäppchen lernen kann. Der amerikanischen



Überwachtes Lernen: Wenn das Zielmuster t nicht mit dem Output-Muster o übereinstimmt, müssen die Gewichte zwischen den Zellen angepasst werden.

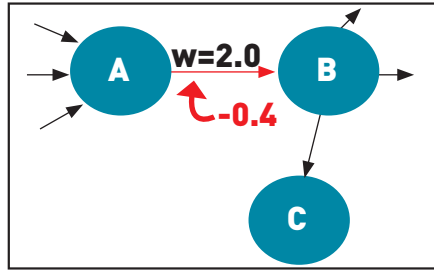
Originalquelle: Stephan Ruttloff

Variante des Märchens zufolge können Rotkäppchen drei Lebewesen mit typischen Merkmalen begegnen: ein Wolf (große Augen, große Ohren, große Zähne), die Großmutter (große Augen, freundlich, faltig) und ein Holzfäller (große Ohren, freundlich, schön).

Diesen sechs vorgegebenen Reizmerkmalen stehen sieben Verhaltensweisen gegenüber, die Rotkäppchen ausführen kann: weglaufen, schreien und den Holzfäller suchen (bei einer Begegnung mit dem Wolf), annähern, Essen anbieten und auf die Wange küssen (Begegnung mit der Großmutter) sowie annähern, Essen anbieten und flirten (Begegnung mit dem Holzfäller).

Das entsprechende neuronale Netz besteht aus sechs Input-Neuronen, wovon jedes Neuron für eine wahrgenommene Eigenschaft steht. Die Verhaltensweisen von Rotkäppchen werden durch die sieben Output-Neuronen repräsentiert. Jedes Input-Neuron der (Wahrnehmung) wird mit allen Output-Neuronen (Verhalten) verbunden. Zunächst sind die Verbindungsstärken zufällig, und das Netzwerk produziert nur unsinniges Verhalten. Das Rotkäppchen-Netzwerk soll jetzt mithilfe der Delta-Lernregel trainiert werden. Dazu legt man ein Input-Muster der Trainingsmenge an und vergleicht das gewünschte Output-Muster mit dem tatsächlichen Output. Anschließend wird die Verbindungsstärke zwischen einem Input- und einem Output-Neuron entweder erhöht oder erniedrigt, abhängig davon, ob das Input-Neuron aktiv war und zum gewünschten Output beigetragen hat. Dieses Verfahren führt zu einer schrittweisen Annäherung an die gewünschten Output-Muster und nach mehreren Trainingsdurchgängen sind alle Verbindungsstärken richtig eingestellt. Das Rotkäppchen zeigt das gewünschte Verhalten: Es läuft weg, schreit und hält nach dem Holzfäller Ausschau, wenn es ein Wesen mit großen Ohren, Augen und Zähnen sieht. Man kann das Verhalten von Rotkäppchen aber auch durch ein dreischichtiges Netzwerk simulieren. Diese Lösung von Aufgaben durch Mehrebenen-Netzwerke mit sogenannten Hidden Units oder Hidden Layers-versteckten Knoten – ist sehr häufig. Hidden Units sind Zellen, die keine direkte Verbindung zur Außenwelt haben. Man kann diese als Verkörperung von Konzepten – also als Zusammenfassung von bestimmten Merkmalen – ansehen wie „Großmutter“ und „Wolf“.

Im Beispiel geben die Input-Neuronen die wahrgenommenen Eigenschaften erst an die Neuronen des Hidden Layers weiter. Diese Zellen in der Zwischenschicht stellen durch Training ihre synaptischen Verbindungen so ein, dass der Input verallgemeinert und Input-Cluster gebildet werden, die ihrerseits wieder Output-relevant



Delta-Regel: Ist die Aktivität von Zelle B größer als erwünscht, wird die Verbindung zu anderen Zellen durch Abzug eines Betrags von den Gewichten geschwächt.

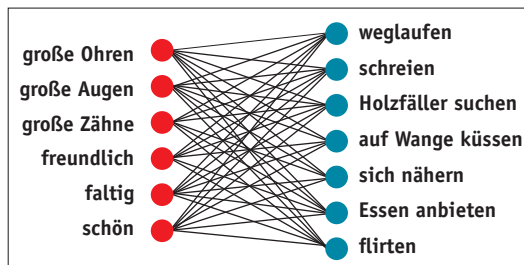
sind. Anders gesagt: Durch die Einführung eines Hidden Layers wird das Netz abstraktionsfähig, denn die wahrgenommenen Eigenschaften werden jetzt in Kategorien zusammengefasst. Ein Neuron des Hidden Layers repräsentiert den Prototyp einer dieser Kategorien. Der Input wird jetzt in drei Kategorien eingeteilt, und die drei Neuronen der Zwischenschicht repräsentieren ihre Prototypen, denen wir jetzt die Namen „Großmutter“, „Wolf“ und „Holzfäller“ geben können. pk

Serielle versus parallele Informationsverarbeitung

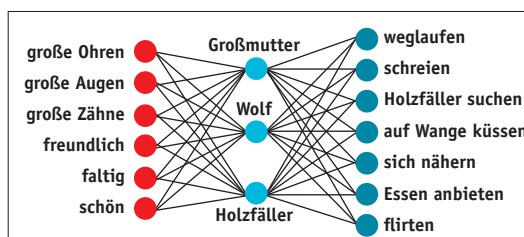
➤ Heute übliche Rechner basieren auf der von-Neumann-Architektur. Sie beziehen ihre Leistungsfähigkeit aus einem oder nur wenigen zentralen Prozessoren, die mit extrem hoher Geschwindigkeit Programme sequenziell, also Befehl um Befehl, abarbeiten. Doch die sequenzielle Abarbeitung entspricht nicht der Arbeitsweise neuronaler Netze.

Natürliche und künstliche neuronale Netze arbeiten massiv parallel und simultan. Obwohl einzelne Prozessorelemente nur eine geringe Verarbeitungsgeschwindigkeit aufweisen, ergibt die Parallelverarbeitung hohe Rechenleistungen. Serielle von-Neumann-Rechner sind deshalb nicht die ideale Architektur für neuronale Netze.

Die Vorteile neuronaler Netze kommen nur voll zur Geltung, wenn die zugrunde liegende Hardware ebenfalls hochgradig parallel ist. Eine Möglichkeit sind Parallelrechner, bei denen jeder Prozessor ein Neuron repräsentiert. Auf allen Parallelrechnersystemen gab es Bemühungen, künstliche neuronale Netze effizient zu implementieren. Bis heute werden neuronale Netze allerdings kaum auf Hardware-Basis realisiert, sondern in der Regel per Software auf klassischen Rechnerarchitekturen.



Ein zweischichtiges Netzwerk, welches das Verhalten von Rotkäppchen simuliert (nach Jones & Hoskins)



Ein dreischichtiges Netzwerk mit drei Kategorien als Hidden Layer (nach Jones & Hoskins)