Forschungsmodul: Komplexe Systeme

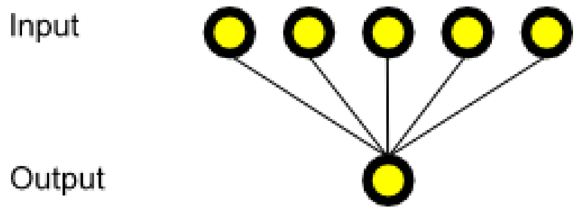
Bericht zur Vorlesung vom 08. November 2007 von Jan-Philip Gehrcke

---

Künstliche neuronale Netzwerke (KNN)

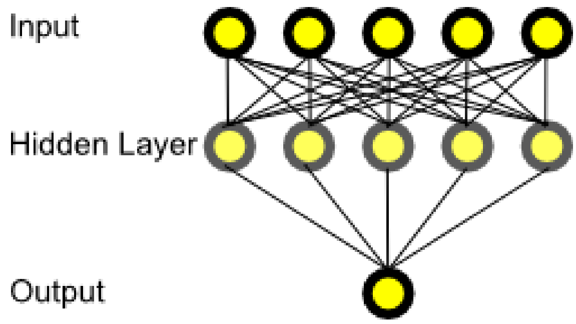
Architektur

Um neuronale Netzwerke zu beschreiben und künstlich zu modellieren, ist es erforderlich eine gewisse Architektur eines solchen KNN festzulegen. Am anschaulichsten sind hierbei sogenannte feedforward-Netzwerke, in denen die Daten in einer festgelegten Richtung fließen. Das simpelste Beispiel ist die Grundversion des Perzeptrons, welches aus verschieden gewichteten Eingängen (Inputs) und einem Ausgang (Output) besteht, der zwei Zustände kennt. Der Wechsel zwischen diesen Zuständen soll bei einem bestimmten Schwellenwert der Inputsumme stattfinden.



Dieses Modell passt auf das reale einzelne Neuron: die Eingänge entsprechen hierbei den Synapsen, welche das eingehende Signal über die Dendriten weitergeben zum Zellkörper, wo es verrechnet wird. Schließlich wird das Ausgangssignal über das Axon der Zelle nach außen geleitet. Überschreitet die Summe der Eingangssignale den Schwellwert, “feuert” das Neuron.

Das simple Perzeptron kann für komplexere Strukturen einfach erweitert werden, indem zusätzliche Schichten (sog. hidden layers) eingebaut werden. Im feedforward-Netzwerk ist dann jedes Neuron mit allen nächsten verbunden.



Modellierung

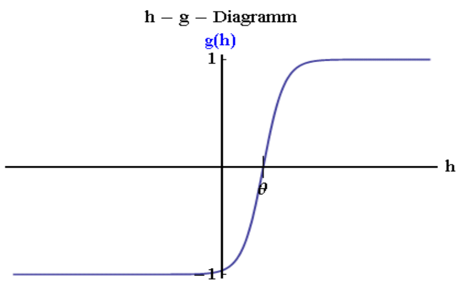
Die Funktion eines einfachen KNN mit einem Ausgang ist mathematisch zu modellieren. Die Summe h der Eingänge wird mithilfe des Skalarproduktes zwischen N-dimensionalem Inputvektor $\vec{x}$ und dem Kopplungsvektor $\vec{J}$ bestimmt. So geht jede Eingangskomponente x_i mit ihrem jeweiligen Gewicht (Kopplungsstärke) J_i in h ein:

$$\vec{x} \cdot \vec{J} = \sum_{i=1}^{N} x_i J_i = h$$

h wird auch als inneres Feld bezeichnet. Nun kann man eine Aktivierungsfunktion $g(h)$ für das Perzeptron aufstellen, wobei g den Zustand des Ausgangs beschreibt. Nimmt man für den ausgeschalteten Zustand (Neuron in Ruhe) $g=-1$ und für den eingeschalteten Zustand (Neuron feuert) $g=1$ an, kann man $g(h)$ mit einer sigmoiden Funktion, z.B. folgendermaßen, modellieren:

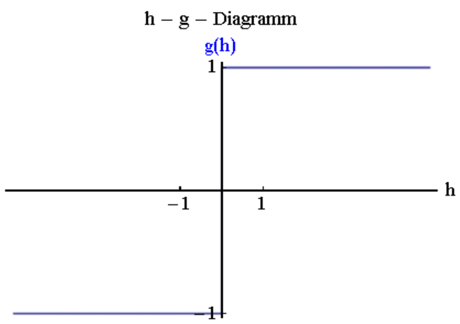
$$g(h) = \tanh(\beta(h-\theta))$$

Dabei ist θ der Schwellwert und β die Steilheit der Kurve um θ herum. Die Aktivierungsfunktion könnte dann folgendermaßen aussehen ($\beta = 2$):



Das Perzeptron als Logikelement

Lässt man β gegen unendlich gehen und transformiert h so, dass $\theta = 0$ wird (prägt man die Schwelle des Perzeptrons also implizit ein), dann lässt sich $g(h)$ als $g(h) = sign(h)$ ausdrücken. Mit dieser Formulierung führten Warren McCulloch und Walter Pitts das Neuron mit mehreren Eingängen und einem Ausgang als Logikelement ein.



Geometrische Interpretation

Die N-dimensionalen Vektoren $\vec{x}$ sind Informationsbeispiele oder Muster, die zusammen mit dem Kopplungsvektor $\vec{J}$ in einem Raum $\mathbb{R}^N$ beschrieben werden können. Es kann dann eine Hyperebene durch den Ursprung dieses Mausterraumes geben, sodass für alle Vektoren $\vec{x}_o$ “oberhalb” dieser Ebene $g(h)=1$ und für alle Vektoren $\vec{x}_u$ unterhalb der Ebene $g(h)=-1$ gilt. Dieses Verhalten des Perzeptrons entspricht allgemein einer binären Klassifikation, auch Dichotomie genannt, welche verschiedene Eingangsmuster in zwei sich ausschließende Fälle einteilt.

Linear separable Klassifikation

Das oben genannte Hyperebenen-Beispiel für eine Dichotomie entspricht einer linear separablen Klassifikation.

Eine Klassifikation eines Mustersatzes

$$\begin{pmatrix} \vec{x}^{(1)} \\ S_1 \end{pmatrix} \cdots \begin{pmatrix} \vec{x}^{(P)} \\ S_P \end{pmatrix}$$

mit $S = g(h)$ heißt linear separabel, wenn ein $\vec{J} \in \mathbb{R}^N$ existiert, sodass

$$S_\mu = sign(\vec{J} \cdot \vec{x}^{(\mu)}) \quad für\, \mu = 1, \dots , P \quad .$$

Dass nicht alle Dichotomien linear separabel sind, kann man sich anschaulich anhand der Verknüpfung $S(\vec{c}) = sign(x_1) \;\; XOR \;\; sign(x_2)$ vorstellen: hier lässt sich keine Hyperebene finden, die die Ergebnisse trennt. Durch nichtlinear transformierende Vorbehandlungen lassen sich Mengen jedoch manchmal linear separabel machen.

Speicherkapazität des Perzeptrons

Hat man verschiedene Muster $\vec{x}^{(1)}, \dots , \vec{x}^{(P)} \in \mathbb{R}^N$ in allgemeiner Lage, d.h. jede Untermenge von $\{\vec{x}^{(1)}, \dots , \vec{x}^{(P)} \in \mathbb{R}^N\}$ ist linear unabhänig, stellt man fest, dass es 2^P verschiedene Dichotomien gibt, die Eingangsmuster einzuteilen. Nun möchte man die Anzahl $C(P,N)$ der Dichotomien finden, welche eine linear separable Klassifikation sind. Darunter kann man sich die Anzahl der möglichen Kopplungsvektoren $\vec{J}$ vorstellen. Mit der Anzahl dieser steigt auch das Speichervermögen des Netzwerkes. Durch geometrische Überlegungen für kleine N kann man eine Rekursionsformel für $C(P,N)$ aufstellen, die auch allgemein beweisbar ist:

$$C(P+1,N) = C(P,N) + C(p,N-1)$$

Die entsprechende Lösung lautet dann:

$$C(P,N) = 2 \sum_{k=0}^{N-1} \binom{P-1}{k}$$

Wenn die Dimension N größer ist als die Zahl der Muster P, ergibt sich, wie erwartet, $C = 2^P$. Die Wahrscheinlichkeit $w(P, N)$, dass eine zufälle Dichotomie eine linear separable Klassifikation ist, erhält man also durch:

$$w(P, N) = \frac{C(P, N)}{2^P}$$

Beschreibt man P durch $P = \alpha\, N$, ergibt sich:

$$\lim_{N \to \infty} w(\alpha\, N, N) = \left\{ \begin{matrix} 1 & \textit{für} & \alpha < 2 \\ \frac{1}{2} & \textit{für} & \alpha = 2 \\ 0 & \textit{für} & \alpha > 2 \end{matrix} \right\}$$

Wenn bei sehr großen N also höchstens doppelt soviele Muster wie Input-Kanäle vorhanden sind, sind nahezu alle möglichen Dichotomien linear separable Klassifikationen, was ein großes Speichervermögen bedeutet.