Konvergenz von Hopfield-Netzen

Matthias Jauernig

1. August 2006

Zusammenfassung
Die nachfolgende Betrachtung bezieht sich auf das diskrete Hopfield-Netz und hat das Ziel, die Konvergenz des Verfahrens zu zeigen. Leider wird dieser Beweis in Büchern und Internet-Artikeln zu unausführlich geführt, das möchte ich mit diesem Beitrag bereinigen.

1 Einführung in das diskrete Hopfield-Netz

Das Hopfield-Netz ist ein Modell eines Neuronalen Netzes, welches die Simulation von Speicherungsvorgängen im menschlichen Gehirn zum Ziel hat und dadurch eingegebene (vielleicht verrauschte) Muster zu bereits gespeicherten Mustern zuordnen soll. Eingaben und Ausgaben sind (zumindest hier) bipolar, d.h. entsprechen der Menge $\{-1, 1\}$.

Der Gesamtinput x'_k eines Neurons N_k berechnet sich bei einem Netz mit n Neuronen aus dem Skalarprodukt

$$x'_k = \sum_{i=1}^{n} w_{ij} \cdot x_i$$

Dementsprechend ergibt sich der Aktivierungszustand a_k aus der Signum-Funktion:

$$a_k = sign(x'_k) = \begin{cases} 1 & \text{für } x'_k \geq 0 \\ -1 & \text{für } x'_k < 0 \end{cases}$$

Die Ausgabe y_k ergibt sich aus dem Aktivierungszustand a_k des Neurons k. Aufgrund der vollständigen Vernetzung im Hopfield-Modell ist der Output-Vektor aller Neuronen des Netzes gleich dem Input-Vektor eines Neurons im nächsten Takt. Es gibt sowohl stetige als auch diskrete Interpretationen des Hopfield-Modells. Hier möchte ich mich auf die diskrete Variante beschränken, womit uns ein taktweises Arbeiten gegeben ist. Entsprechend dieser Überlegungen gilt:

$$x(t+1) = y(t) = a(t) \qquad \text{wenn t>0 den momentanen Zeittakt angibt}$$

Die Gewichtsmatrix W ist symmetrisch und die Elemente der Hauptdiagonalen betragen 0. Dies ist eine wichtige Voraussetzung für die Konvergenz des Verfahrens und kann veranschaulicht werden durch einen ungewichteten Netzgraphen, in dem kein Neuron eine Verbindung zu sich selbst besitzt, jedoch zu allen anderen Neuronen.

2 Die Energiefunktion

Die im Hopfield-Netz verwendete Energiefunktion entspringt dem *Ising-Modell* in der Physik. Dieses beschreibt die Berechnung des Gesamtfeldes eines magnetischen Materials anhand der Spins der darin enthaltenen Atome. Im Hopfield-Netz wird die Funktion verwendet, um die Konvergenz des Verfahrens zu zeigen. Wird in einem Berechnungsschritt ein gespeichertes Muster bzw. ein Neben-Minimum (bei korrelierten Mustern) erreicht, so fällt die Energiefunktion in ein lokales Minimum.

Die Energiefunktion ist definiert als:

$$E = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}a_i a_j$$

3 Arbeitsweise zur Musterassoziation

Bei dem hier vorgestellten Hopfield-Modell lassen sich k Muster im Netz zur späteren Wiedererkennung speichern, indem Teilmatrizen $W^{(l)}$ (l. . . Index des zu speichernden Musters $y^{(l)}$) durch $W^{(l)} = y^{(l)} \cdot y^{(l)^T}$ berechnet und summiert werden:

$$W = \frac{1}{k}\sum_{l=1}^{k} W^{(l)}$$

Dabei müssen die Muster aber unkorreliert sein, ansonsten kommt es leicht zu Neben-Minima, in welche die Energiefunktion fallen kann.

Das Netz arbeitet so, dass im 1. Zeittakt das zu erkennende Muster als Input an die Neuronen angelegt wird. Im 2. Zeittakt sind es durch die vollständige Vernetzung die Outputs der Neuronen selbst, die als Input verwendet werden.

Man unterscheidet 2 Arbeitsweisen:

1. **synchron**: Die Zustände *aller* Neuronen werden parallel aktualisiert, d.h. alle Neuronen berechnen ihre Ausgabe anhand der Outputs im letzten Durchlauf.

2. **asynchron**: Hier wird der Zustand *eines* Neurons N_k aktualisiert. Ist dies geschehen, wird ein nächstes Neuron N_l ausgewählt und dieses mit dem aktuellen Netzzustand (und dem neuen Zustand von N_k) aktualisiert. Die Neuronen müssen dabei nicht sequentiell ausgewählt werden, dies kann auch zufällig geschehen. Wichtig ist nur, dass die Wahrscheinlichkeit der Auswahl für jedes Neuron gleich ist, d.h. im Durchschnitt alle Neuronen in etwa gleich oft aktualisiert werden. Für diese Arbeitsweise lässt sich die Konvergenz zeigen und sie ist es, auf welche sich die nachfolgende Betrachtung bezieht.

Dieser Algorithmus: „Neuron auswählen, Zustand aktualisieren, nächstes Neuron auswählen, ...“ wird solange durchgeführt, wie sich ein Zustand ändert. Ändert sich der Zustand von keinem Neuron des Netzes, so gilt der aktuelle Netzzustand als stabil. In diesem Fall ist ein lokales Minimum der Energiefunktion vorhanden und der dabei erreichte Netzzustand (die Ausgabe der Neuronen) entspricht entweder einem gespeicherten Muster oder einem Neben-Minimum der Energiefunktion, welches durch Korrelation (also Überschneidungen) von gespeicherten Mustern entstehen kann.

4 Konvergenz des Verfahrens

Das zu betrachtende Problem bezieht sich auf den stabilen Endzustand. Das Verfahren ist nur dann sinnvoll, wenn nach endlich vielen Schritten ein Netzzustand erreicht ist, sodass ein beliebiges Neuron ausgewählt werden kann, sich sein Zustand aber nicht mehr ändert.

Wann ändert sich der Zustand eines Neurons N_k? Dies ist der Fall, wenn 1.) der alte Zustand $a_k = 1$ ist und das Skalarprodukt $w_{.k}^T \cdot a < 0$. In diesem Fall hat die Signum-Funktion den Wert -1, so dass der neue Zustand $b_k = -1$ ist. 2.) kann der alte Zustand $a_k = -1$ sein und das Skalarprodukt ≥ 0. Hier ist der neue Zustand des Neurons $b_k = 1$, da die Signum-Funktion bei Argumenten größer gleich 0 den Wert 1 aufweist.

Zunächst soll gezeigt werden, dass sich bei der Zustandsänderung eines beliebigen Neurons N_k der Wert der Energiefunktion verringert. Daraus ergibt sich direkt ihre Konvergenz. Aus dieser Betrachtung heraus kann dann gezeigt werden, dass sich in endlich vielen Schritten ein stabiler Endzustand einstellen *muss*.

Es sei also angenommen, dass ein Neuron N_k betrachtet wird, dessen Zustand sich von a_k zu b_k ändert (also von -1 auf 1 oder umgekehrt). Die Energiefunktion habe *vor* der Aktualisierung den Wert E_1 und *nach* der Aktualisierung den Wert E_2. Diese ergeben sich aus:

$$2 \cdot E_1 = -\sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq k}}^{n} w_{ij} a_i a_j - \sum_{i=1}^{n} w_{ik} a_i a_k \tag{1}$$

$$2 \cdot E_2 = -\sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq k}}^{n} w_{ij} a_i a_j - \sum_{i=1}^{n} w_{ik} a_i b_k \tag{2}$$

Bei Subtraktion der Gleichungen 1 - 2 heben sich die beiden Doppelsummen auf, übrig bleiben nur die Terme für das geänderte Neuron N_k:

$$\begin{aligned} 2 \cdot E - 2 \cdot E_k &= -\sum_{i=1}^{n} w_{ik} a_i a_k + \sum_{i=1}^{n} w_{ik} a_i b_k \\ &= (-a_k + b_k) \cdot \underbrace{\sum_{i=1}^{n} w_{ik} a_i}_{\text{Skalarprodukt für } N_k} \end{aligned} \tag{3}$$

Wie ersichtlich ist, entspricht die übrig bleibende Summe gerade dem Skalarprodukt, welches zur Berechnung des Gesamtinputs von N_k zu ermitteln ist.

Um zu entscheiden, wie groß die Differenz sein kann, ist die rechte Seite der Gleichung zu untersuchen. Dabei gilt es 2 Fälle zu unterscheiden:

1. alter Zustand $a_k = 1$: In diesem Fall ist $b_k = -1$, da sich der Zustand geändert haben soll. Dies ist aufgrund der Verwendung der Signum-Funktion aber nur möglich, wenn das Skalarprodukt (der Gesamtinput, welcher der Summe in Gleichung 3 entspricht) kleiner als 0 ist. Da $-a_k + b_k = -2$ gilt, beträgt der Wert der rechten Seite >0. Daraus folgt: $E - E_k > 0 \equiv E > E_k$
2. alter Zustand $a_k = -1$: In diesem Fall ist $b_k = 1$ der geänderte Zustand. Das Skalarprodukt muss für diese Änderung größer oder gleich 0 sein. Da $-a_k + b_k = 2$ ist, beträgt der Wert der rechten Seite ≥ 0. Daraus folgt: $E - E_k \geq 0 \equiv E \geq E_k$.

Weiterhin ist die Energiefunktion nach unten beschränkt, da

$$E \geq -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} |w_{ij}|$$

Aus diesen Überlegungen folgt, dass bei jeder Zustandsänderung eines Neurons N_k der Energiewert des aktuellen Netzzustands fällt oder gleich bleibt. Da die Folge der Energiewerte nach unten beschränkt ist, folgt daraus direkt deren Konvergenz.

Damit ist aber noch nicht allzu viel über die Konvergenz des Netzzustandes gesagt, möchte man meinen. Auf den ersten Blick könnte man annehmen, dass es zu Zyklen in den

Netzzuständen kommen kann, da die Energiewerte auch gleich bleiben können. Dies ist aber nicht der Fall. Es ist ersichtlich, dass sich der Zustand bei n Neuronen maximal n-mal hintereinander von -1 auf 1 ändern kann und dadurch möglicherweise der Energiewert gleich bleibt. Spätestens dann muss ein Neuron jedoch seinen Zustand von 1 auf -1 ändern. Nur so könnte ein alter Netzzustand wiederhergestellt werden. Da dann aber (nach Fall 1) der Energiewert auf jeden Fall kleiner werden muss, kommt man auf keinen Fall mehr zu einem bereits betrachteten Zustand zurück.

Da es wie eben gezeigt keine Zyklen gibt, die Energiefunktion monoton fällt sowie nach unten begrenzt ist (also konvergiert) und nur eine endliche Zustandsmenge (bestehend aus 2^n Zuständen) existiert, *muss* auch die Folge der Netzzustände (also Output-Vektoren) konvergieren, also zu einem stabilen Endzustand gelangen.