Ludwig-Maximilians-Universität
HS: Logik, Gehirn, Information
Dozenten: Prof. Dr. Lauth; Dr. Brückner
Referenten: Jacob Halatek; Christian Müller

1 Simple Recurrent Networks

1.1 Feed-Forward und seine Grenzen

Feed-Forward-Netze verarbeiten Information parallel: Ihre Input-Einheiten bekommen alle zur gleichen Zeit Information und verarbeiten diese.

Repräsentation von Zeit, bzw. Information, die zeitlich geordnet ist, ist damit nur bedingt möglich.

Mögliche Lösung kann ein vorgeschalteter Zwischenspeicher (Buffer) sein, der die Information aufnimmt und dann an das neuronal Netz weitergibt. Diese Lösung wirft weitere Fragen auf:

1. Gibt es solche Zwischenspeicher in biologischen Systemen?
2. Wann weiß ein solches Netz, dass der Input abgeschlossen ist und bearbeitet werden kann?
3. Ein Zwischenspeicher setzt eine feste Grenze für die Dauer von verarbeitbaren Mustern/Inputs.
   Außerdem legt diese Lösung (in Anbetracht von 2. und 3. nahe, dass Muster immer gleichlang sind. => Diese Regelmäßigkeit ist bspw. bei Zeichenfolgen der Sprache nicht gegeben.
4. Mit diesem Ansatz lässt sich nicht zwischen relativer und absoluter zeitlicher Position unterscheiden. Beispiel können hierfür folgende zwei Bitfolgen sein:

   $$[011100000]$$

   $$[000111000]$$

   Handelt es sich bei diesen beiden Bitfolgen um das gleiche Muster, das lediglich in der Zeit verschoben ist? Oder (als geometrische Vektoren betrachtet) sind dies zwei deutlich voneinander unterschiedene Bitfolgen?

1.2 Neuronale Netze mit einfachem Feedback

Erfolgversprechender ist es, Zeit durch den Effekt zu repräsentieren, den sie auf die Verarbeitung hat. Eine solche Repräsentation von Zeit hätte den großen Vorteil, dass sie dem neuronalen Netz implizit ist und nicht erst simuliert werden müsste.

- Ein solches Verarbeitungssystem hätte dynamische Eigenschaften, die auf zeitliche Sequenzen reagieren.

- Um auf zeitliche Sequenzen als solche reagieren zu können, muss ein neuronales Netz einen Speicher bekommen.

Feedback innerhalb eines neuronalen Netzes bietet diese Möglichkeiten. Ein möglicher Entwurf eines solchen neuronalen Netzes mit einfachem Feedback (Simple-Recurrent Network, oder: SRN) besteht darin, zu jeder versteckten Schicht eines Multi-Layer Perceptrons (MLP) eine entsprechende Kontext-Schicht hinzuzufügen, die wie folgt verschaltet ist:

1. Anzahl der Kontextzellen entspricht der Anzahl verdeckter Zellen

2. Die Rückkopplungsverbindung hat das feste Gewicht 1 (ist im Lernprozess nicht veränderbar!)

Damit enthalten die Kontextzellen den letzten Zustand der verdeckten Schicht und erlauben somit einen zeitlichen Bezug.

1.3 Beispiele

1. **XOR – MLP vs. SRN**
   Wenn XOR als eine Bitfolge von jeweils zwei Bit, die der XOR-Input sind, und einem folgenden Bit, das der XOR-Output ist, dargestellt wird, ist ein SRN in der Lage, das jeweils dritte Bit (also den XOR-Output) vorherzusagen.

2. **Buchstabenfolgen**
   Diese Art von Vorhersage ist auch bei komplexeren Eingaben möglich. Jeffrey L. Elman ließ ein SRN Wörter als Buchstabenfolgen einlesen. Nachdem das SRN trainiert war, ließ es indirekt – über die Fehlerquote in der Vorhersage – die Struktur des Inputs in der Zeit sichtbar werden.

2 Hopfield-Netze und Boltzmann-Maschinen

2.1 Begriffe und Wiederholung

Feed Forward Netze In einem FFN wir die Informationsverarbeitung in *Layern* organisiert. Ein FFN bildet einen n-dimensionalen Inputvektor auf einen k-dimensionalen Outputvektor ab.

Input: $\mathbf{x} \in \mathbb{R}^n$

Output: $\mathbf{y} \in \mathbb{R}^k$

Verschaltung und Aktivierung: $\mathbf{xW} = \mathbf{a} \qquad sign(\mathbf{a}) = \mathbf{y} \qquad \mathbf{W} \in \mathbb{R}^{n \times k}$

2.2 Motivation und Einführung

2.2.1 Natürliche neuronale Netze

- Das Gehirn besteht aus 10^{10}-10^{11} Neuronen
- Nur etwa 0,1% der kortikalen Pyramidenzellen sind Input-/Outputschichten zugeordnet
- Jedes Neuron bildet ungefähr 10^4 synaptische Verbindungen aus und bekommt von ebensovielen Neuronen Input
- Damit sind die meisten Neuronen in rekursiven Netzwerken organisiert und über 2-4 Synapsen selbstgekoppelt

2.2.2 Konsequenz der Rückkopplung

- Keine klare Layer-Struktur
- Keine linearen Prozesse
- Informationsverarbeitung durch Dynamik des Systems

2.2.3 Qualitative Aspekte der Dynamik

Dynamik beschreibt die *Entwicklung eines System* in Raum und Zeit in Form von Differenz- oder Differentialgleichungen.[1]

Stabilität Bei der Untersuchung von dynamischen Systemen spielt die Frage nach der Existenz von *Fixpunkten* eine große Rolle: Gibt es einen Zustand x', so dass sich der Zustand $x(t') = x'$ des Systems für alle $t > t'$ nicht mehr ändert?

Attractor Neuronal Networks (ANN) Bezug zu neuronalen Netzen: Unterliegt ein neuronales Netz *dynamischem Verhalten*, so lässt sich *Information* allgemein in Form von *Fixpunkten*(Attraktoren) repräsentieren. Dabei entspricht ein *neuronales Netz* einer bestimmten „*Energielandschaft*".

2.3 Assoziative Netze

2.3.1 Übergang: Feed-Forward => Rekurrente Netze

In einem (bidirektionalen) assoziativen Netz ist der Output vollständig an den Input gekoppelt, i.e.

$$\mathbf{x}_i \in \{-1, 1\}^n, \mathbf{y}_i \in \{-1, 1\}^k \quad \text{Zeilenvektoren}$$

[1]**Raum:** Allg. ein abstrakter Raum der von den *Variablen des Systems* aufgespannt wird. Zum Beispiel den Zuständen aller Neuronen.
Zeit: (1) *Diskret* $t \in N$ führt auf Differenzengleichungen: $x_{t+1} = f(x_t)$. (2) *Kontinuierlich* $t \in R$ führt auf Differentialgleichungen: $\dot{x} = f(x, t)$

- 1.Zeitschritt: $\mathbf{y}_0 = sign(\mathbf{x}_0\mathbf{W})$
- 2.Zeitschritt: $\mathbf{x}_1^T = sign(\mathbf{W}\mathbf{y}_0^T)$
- 3.Zeitschritt: $\mathbf{y}_1 = sign(\mathbf{x}_1\mathbf{W})$
- …
- n.Zeitschritt: $\mathbf{y}_{n-1} = sign(\mathbf{W}\mathbf{x}_{n-1})$
- n+1.Zeitschitt: $\mathbf{x}_n^T = sign(\mathbf{W}\mathbf{y}_{n-1}^T)$

2.3.2 Stabilität und Lernen

Stabilität in autoassoziativen Netzen: Das Ziel ist, $\mathbf{W}$ derart zu trainieren, dass das neuronale Netz für verschiedene Inputvektoren $\mathbf{x}_i$ gegen bestimmte Zustände $(\mathbf{x}'_i, \mathbf{y}'_i)$ konvergiert: Aktiviert man ein Netzwerk mit dem Input $\mathbf{x}$ so nimmt es nach endliche vielen Zeitschritten den Zustand $(\mathbf{x}', \mathbf{y}')$ ein.

Analogie beim Menschen: Ein bestimmter Reiz verursacht eine bestimmte Errinerung/Assoziation. Dies geschieht nicht instantan. Es vergeht Zeit: Assoziation ist ein Prozess.

Stabilität: Gesucht sind Paare von Vektoren $(\mathbf{x}'_i, \mathbf{y}'_i)$, die für gegebene Inputs $\mathbf{x}$ oder $\mathbf{y}$ *stationäre Zustände* bilden. Es gilt allgemein:

- $\mathbf{y}_i = sign(\mathbf{x}_i\mathbf{W})$
- $\mathbf{x}_{i+1}^T = sign(\mathbf{W}\mathbf{y}_i^T)$

Und damit lauten die *Stabilitätsbedingungen*:

- $\mathbf{y} = sign(\mathbf{W}\mathbf{x})$
- $\mathbf{x}^T = sign(\mathbf{W}\mathbf{y}^T)$

Hebb'sche Lernregel: Das Synapsengewicht soll *erhöht* werden wenn zwei Neuronen x_i und y_j gemeinsam aktiv, i.e. $x_i(t) = y_j(t) = 1$, *verringert* werden werden wenn dies nicht der Fall ist, i.e. $x_i(t) \neq y_j(t)$.

Dies legt folgende Definition nahe: Änderung des Synapsengewichtes w_{ij}:

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}$$

$$\Delta w_{ij} = \gamma x_i(t) y_j(t)$$

γ : Lernparameter

Die *Verschaltungsmatrix* ändert sich demnach mit: $\mathbf{W} = \mathbf{x}^T\mathbf{y}$

Vergleich mit den Stabilitätsbedingungen: Durch Hebb'sches Lernen lässt sich demnach eine Verschaltungsmatrix gewinnen die die geforderten Stabilitätskriterien erfüllt:

$$\mathbf{y} = sign(\mathbf{x}\mathbf{W}) = sign(\mathbf{x}\mathbf{x}^T\mathbf{y}) = sign(\|\mathbf{x}\|^2\mathbf{y}) = \mathbf{y}$$

$$\mathbf{x}^T = sign(\mathbf{W}\mathbf{y}^T) = \ldots = \mathbf{x}^T$$

Ergebnis: Hebb'sches Lernen kann genutzt werden um Vektorpaare $(\mathbf{x}, \mathbf{y})$ in einem assoziativen Netz zu speichern.

2.3.3 Energiefunktionen

Das Skalarprodukt zweier Vektoren ist eine *Zahl*. Es ist umso größer, je kleiner der *Winkel* zwischen beiden Vektoren ist. **Die Energiefunktion:**

1. Es sei $(\mathbf{x}, \mathbf{y})$ ein *stationärer Zustand*. $\mathbf{x}_0$ der Anfangswert.
2. Nach $n + 1$ Zeitschritten beträgt der Aktivierungsvektor $\mathbf{a}^T$: $\mathbf{a}^T = \mathbf{W}\mathbf{y}_n^T$
3. Dieser Zustand ist *stabil*, wenn $sign(\mathbf{a}) = \mathbf{x}_n$ gilt.
4. Dies ist erfüllt, wenn der *Winkel* zwischen $\mathbf{a}$ und $\mathbf{x}_n$ klein genug ist.
5. Das Skalarprodukt $\mathbf{x}_n\mathbf{a}^T = \mathbf{x}_n\mathbf{W}\mathbf{y}_n^T$ ist in diesem Fall groß.

Die *Energiefunktion* $E(\mathbf{x}_i, \mathbf{y}_i) = -\frac{1}{2}\mathbf{x}_i\mathbf{W}\mathbf{y}_i^T$ wird mit jedem Zeitschritt kleiner.

2.4 Hopfield-Netze

Ein **Hopfield-Netz** ist:

1. *vollständig verschaltet*, i.e. jedes Neuron ist mit allen anderen verbunden
2. *asynchron*, i.e. in jedem Zeitschritt wir nur der Zustand *eines* Neurons aktualisiert
3. *autoassoziativ*, i.e. $\mathbf{x} = \mathbf{y}$ mit
4. *symmetrischer* Verschaltungsmatrix $\mathbf{W} = \mathbf{W}^T$, i.e. $w_{ij} = w_{ji}$
5. *ohne Selbstkopplung*, i.e. $w_{ij} = 0$ für $i = j$

Zur Notwendigkeit der Bedingungen

1. Verschaltung und Selbstkopplung: Betrachte die Matrix:

$$\mathbf{W} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

   Jeder Zustand oszilliert in$\{-1, 1\}$

2. Symmetrie: Betrachte die Matrix:

$$\mathbf{W} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

Oszillation der Zustände in $\{(-1,1),(1,-1)\}$

2.4.1 Dynamik und Konvergenz

Energiefunktion für das Hopfield-Netz. Es sei:

1. $\mathbf{W} \in \mathbb{R}^{n \times n}$ die Verschaltungsmatrix
2. $\mathbf{x} \in \{-1,1\}^n$ der Zustandsvektor
3. $\Theta \in \mathbb{R}^n$ der Schwellenwert-Vektor

Dann lässt sich eine *Energiefunktion* wie folgt definieren:

$$E(\mathbf{x}) = -\frac{1}{2}\mathbf{x}\mathbf{W}\mathbf{x}^T + \Theta\mathbf{x}^T$$

Als Summe:

$$E(\mathbf{x}) = -\frac{1}{2}\sum_{i,j=1}^{n} w_{ij}x_i x_j + \sum_{i=1}^{n}\Theta_i x_i$$

Beispiel: Ein **Flip-Flop** kann zwei logisch komplementäre Zustände speichern.

$$\mathbf{W} = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$$

$$\Theta_{1/2} = 0$$

$$E(\mathbf{x}) = x_1 x_2$$

Stabile Zustände: $(E(\mathbf{x}) = min)$

$$(1,-1),(-1,1)$$

2.4.2 Konvergenz

Ein Hopfield-Netz erreicht immer ein (lokales) Minimum: Die Energiefunktion

$$E(\mathbf{x}) = -\frac{1}{2}\mathbf{x}\mathbf{W}\mathbf{x}^T + \Theta\mathbf{x}^T$$

ist für gegebenen Anfangswert monoton fallend und nach unten beschränkt.

2.4.3 Anwendungen

- Mustererkennung (Bilderkennung, Texterkennung, Datenerkennung)
- Lösung kombinatorischer Probleme (Verteilung von Objekten unter Nebenbedingungen, Optimierung (Travelling Salesman)

2.4.4 Zusammenfassung: Hopfield

Vorteile:

- Rekkurrente Netze => *Assoziativer Speicher*
- Äquivalent zu ML-Perceptron (Lernen)
- Konvergenz auch für *verrauschten Input*
- Gegenüber FFN biologisch plausibler:
  Dynamik durch Rückkopplung und dadurch
- *Modellierung von „Zeit“*

Grenzen:

- Diskretes Modell => *Diskreter Zustandsraum*
- *Lokale Minima* sind stationäre Zustände

Lösungen:

- *Kontinuierliche* Modelle
- *Stochastische* Modelle

2.5 Boltzmann-Maschinen

2.5.1 Energieminimierung und Fluktuationen

- Beispiel aus der Molekularbiologie: Proteinfaltung
  Wie falte ich eine Kette von tausenden Aminosäuren in eine exakte Struktur?
  Struktur => Funktion
- Proteine als Ionenpumpen
  Was passiert nachdem ein Neuron gefeuert hat?
  Wer pumpt die Ionen aus der Zelle?
- Landschaft der freien Energie
  Komplexes System => Sehr viele lokale Gleichgewichtspunkte
- Fluktuationen
  Wie komme ich aus einem lokalen Minimum heraus?

2.5.2 Stochastik und Rückkopplung

Unterschied der Boltzmann-Maschine (BM) zu Hopfield-Netzen: *Wahrscheinlichkeit für Zustandswechsel*

Definition: Eine *Boltzmann-Maschine* ist ein Hopfield-Netz mit folgendem Zustandswechsel:

$$x_i = \begin{cases} 1 & \text{mit Wahrscheinlichkeit } p_i \\ 0 & \text{mit Wahrscheinlichkeit } 1 - p_i \end{cases}$$

und Wahrscheinlichkeiten p_i

$$p_i = \frac{1}{1 + e^{\frac{-\sum_{j=1}^{n} w_{ij} x_j - \Theta_i}{T}}}$$

Qualitatives Verhalten der BM – Abhängigkeit von T:

- $T \to 0 \;\Rightarrow$ *Diskretes Hopfield Modell*
- $T > 0 \;\Rightarrow\; p_i > 0$ auch für $\sum_{j=0}^{n} w_{ij} x_j - \Theta_i < 0$
  Neuronen können auch feuern wenn Schwellenwert nicht erreicht wurde

Analogie zum Gehirn

- Keine isolierten Teilnetze im Gehirn. Umgebung verursacht „*rauschen*".
- Kanalproteine unterliegen thermischen Fluktuationen: *Spontanes Feuern*.

Zusammenfassung

- Im Gehirn liegt *massive Rückkopplung* vor. Daher spielt die *Dynamik* und *Stabilität* von Neuronalen Netzen eine wichtige Rolle.
- Für Assoziative Netze existieren *Energiefunktionen*. Aus ihnen lassen Erkenntnisse über das *dynamische Verhalten* des Netzes gewinnen.
- *Lokale Minima* -„Störungen"- lassen sich durch stochastische *Fluktuationen* verlassen.