

Datum: 24.01.2006  
Prüfer: Prof. Helbig  
Dauer : ca. 25 Min  
Note : 2,0

Prof Helbig hatte an jenem Tag wohl schon mehrer Prüfungen gehabt. Daher stieg er direkt in die Prüfung ein, lediglich Ausweis wurde verlangt.

**Wie kann man neuronal Netze unterscheiden, wie sind diese aufgebaut, usw.?**

Hier habe angefangen, ein neuronales Netz zu beschreiben (Zuordnungsvorschrift von E nach A, math. Sicht als Graph, parallelen zu den biologischen Netzen)

Dann wollte er wissen, wie sich das nennt, diese Anordnung als Graph: Topologie

Wir gingen weiter zu den Funktionen im Neuron:

$F_{in}$  hat welche Parameter und welche Arten gibt es:

- Eingabe und Gewichte musste ich dann auch anmalen.
- Sigma, Pi, Min, max aufgezählt und Formeln aufgeschrieben. Achtung mit den Indizes
- Nettoinput

$F_{act}$  ebenfalls:

- $net_i$  und  $a(t)$
- Lineare, Schwellwert und sigmoide

Warum macht es keinen Sinn, ein mehrschichtiges Netz mit der linearen

Aktivierungsfunktion? Weil die Gewichte alle aufaddiert werden können und auf ein einschichtiges Netz reduziert werden kann

Warum nimmt man nicht immer die lineare Aktivierungsfunktion? Weil nicht alle Probleme linear separierbar sind UND weil bei der erweiterten Delta Regel die Ableitung benötigt wird. Diese gibt es nicht bei der Sprungfunktion oder ist null bei einfacher linearer Funktion.

Musste die Formel für Fermi Funktion aufschreiben und den Graphen zeichnen. Was steuert der Temperaturkoeffizient? Die Krümmung in den Kurven

Welche Arten von Ausgabefunktionen gibt es?

- Identität
- Signum (Hopfield)
- Maximum (Kohonen)

So, das wäre also eine Möglichkeit der Unterscheidung? Welche andere gibt es noch?

Topologie

FF 1. & 2. Ordnung, FB

Art des Lernens

Überwachtes Lernen & selbstorganisierend.

**Backpropagation:**

Wie arbeitet das Verfahren?

Fehlerfunktion für  $\delta_i$  aufschreiben, Gesamtfehler – die Quadrate waren sehr wichtig  
Netz aufmalen

Was wird bei dem Verfahren geändert? Gewichte

Was genau wird dem Netz beim Lernen übergeben? Eine Trainingsmenge mit Eingabe- und Ausgabevektoren

Welche Probleme gibt es mit dem Verfahren: Oszillation, Plateaus, lokale Minima

Was macht man bei lokalen Minimas? Gewichte neu initialisieren mit stochastischen Mitteln.

Was passiert, wenn die Lernrate zu groß gewählt wurde? Täler können übersprungen werden

Was verbessert das Gradient Reuse Verfahren? Besser Performanz da  $\delta_i$  nicht immer neu berechnet werden müssen.

## Hopfield

Aufmalen

Gibt es da eine Lernphase? Nein, nur ein init. Formel für die Gewichte, wichtig  $w_{ij} = 0$

Welches Verfahren aus der Physik wurde hier genutzt? Spingläser

Parallelen erklären: Spins können die Werte -1 und 1 annehmen. Diese haben das Bestreben, sich auszurichten. Zwischen den Spins bestehen Kopplungsstärken. Die Spins wirken nicht auf sich selbst, sprich die Kopplungsstärke ist Null. Die Spins richten sich so aus, dass Gesamtenergie des Netzes minimal wird  $E = \sum c_{ij}s_i s_j$ . Dieses wurde bei den Hopfield Netzen folgendermaßen umgesetzt: Neuronen haben nur binäre Werte -1 und 1, die Gewichte sind symmetrisch. Das Netz arbeitet so lange, bis die Ausgabe sich nicht mehr ändert, sprich analog zu den Spingläser, bis die Energiefunktion minimal ist.

Es wurde ab der Mitte recht hektisch, alles mögliche wurde abgefragt. Ich dachte, ich hatte einen guten Überblick, aber er hat es geschafft, mich aus der Ruhe zu bringen. Er sagte auch, der erste Teil wäre eine 1 gewesen, aber danach hab ich nur noch Murks gebaut. Er legt Wert auf exakte Formulierung (Sprich nicht nur Eingabe, sondern Eingabevektor). Die Zeit hatte für Kohonen nicht mehr gereicht. Wenn ich jetzt eine Empfehlung abgeben sollte: konzentriert Euch auf Def. Neuronal Netze, BackProp, Hopfield, Kohonen. Die anderen Netze verstehen, aber nicht unbedingt beherrschen.