

Neuronen trainieren mit stetigen Transferfunktionen

Adaptive Systeme

Prof. Dr. Nikolaus Wulff

22. November 2019

1 Nichttriviale Hebb'sche Lernregel

In dieser Übung wird die Hebb'sche Lernregel

$$\Delta \mathbf{W} = \mu(\vec{d} - \vec{y})\vec{x}^T \quad (1)$$

mit $0 < \mu$ für andere Klassen von Transferfunktionen als der Stufenfunktion erweitert.

1. Wie lautet die Verallgemeinerung der Hebb'schen Lernregel für eine stetige Transferfunktion anstelle der Heavyside Stufenfunktion zur Approximation der Neuronausgabe?
2. Berechnen Sie $\sigma'(x)$ für die Fermi- und die erweiterte tanh- Transferfunktionen. Drücken Sie die Lösung $\sigma'(x)$ mittels geeigneter Umformungen durch $\sigma(x)$ aus.

Interpretieren Sie die nun gewonnenen Lösungen:

- (a) Welchen Verlauf stellt $\sigma'(x)$ dar? Plotten Sie hierfür $\sigma(x)$ und $\sigma'(x)$ in einem Diagramm.
- (b) Was bedeutet diese Lösung anschaulich für das Verständnis eines Neurons und des zugehörigen Lernprozess?

Diese Frage ist eine Mischung aus Mathematik und anschaulich intuitiven Verständnis des Lernens generell und der Entwicklung von Algorithmen im Besonderen.

3. Implementieren Sie passend zu den gewonnenen Gleichungen einen Lernalgorithmus zur automatischen Adaption der Gewichte W_{jk} .
4. Testen Sie Ihren Lernalgorithmus an Hand der schon bekannten AND und OR Probleme.

2 Klassifikation der Einheitskugel

Sie wollen normierte Vektoren $\vec{x} \in \mathbb{R}^3$ der Einheitskugel, d.h. $\|\vec{x}\| = 1$, nach den Orientierungen "oben/unten", "links/rechts" und "vorne/hinten" klassifizieren.

Trainieren Sie hierzu ein neuronales Netz – bzw. genauer eine Neuronenschicht –, das den in der vorhergehenden Aufgabe erstellten Lernalgorithmus verwendet und diese Klassifikationsaufgabe löst.

Stellen Sie sich hierzu folgende Fragen:

1. Wie viele Eingänge und wie viele Ausgänge benötigen Sie.
2. Auf welchem Wertebereich müssen Sie nun operieren, wird immer noch eine Abbildung $\{0, 1\}^n \rightarrow \{0, 1\}^m$ realisiert.
3. Wie viele Trainingsvektoren benötigen Sie bzw. wie sollten diese aussehen? Wie lassen sich diese geeignet generieren oder erstellen...?
4. Testen Sie nach dem Trainingsabschluß, d.h. nach richtiger Erkennung aller Muster aus dem Trainings-sample, den Erfolg des Training mit
 - (a) normierten Vektoren, die nicht in der ursprünglichen Trainingsmenge enthalten waren,
 - (b) mit unnormierten Vektoren,

d.h. in beiden Fällen mit Vektoren, die dem Netz auf Grund des Trainings unbekannt und neu sind.

Wie gut werden diese klassifiziert bzw. wie groß ist die Fehlerrate? Hängt diese Fehlerrate irgendwie von dem verwendeten Trainingssatz ab, bzw. fällt Ihnen nach einigem Nachdenken womöglich ein besserer evt. anderer Trainingssatz ein? Sollte der Trainingssatz irgendwelche statistischen (oder sonstigen) Eigenschaften haben?

5. Wie sieht nach dem erfolgreichen Training die Gewichtsmatrix \mathbf{W} aus? Lässt sich diese Matrix irgendwie sinnvoll in Beziehung setzen zu den in der Vorlesung erwähnten Hyperebenen, welche die Muster diskriminieren bzw. klassifizieren sollen?