

Mehrschichtiges Perzeptron (MLP)

Aufgabe 1

Bitte implementieren Sie ein mehrschichtiges Perzeptron (MLP). Das MLP soll eine Eingangsschicht, eine versteckte Schicht und eine Ausgabeschicht haben. Das von Ihnen zu implementierende MLP soll über folgende Features verfügen:

- Anzahl der Neuronen in den Schichten konfigurierbar
- Transferfunktion der Neuronen der Ausgabeschicht wählbar
- Lernrate und Momentum frei wählbar
- Single oder Batch Update wählbar
- Während des Trainings wird der quadratische Fehler für die Trainingsmenge für jeden Zyklus berechnet.

Anmerkung:

- Ein Trainingszyklus setzt sich aus Propagation- und Back-Propagation-Berechnungen für alle Muster der Trainingsmenge zusammen.

Aufgabe 2

Bitte trainieren Sie mit Ihrer Implementierung eines MLP das XOR-Problem. Bitte verwenden Sie zur Überprüfung Ihrer Implementierung für die Initialisierung der Gewichte die Werte aus der Datei „Init_MLP.txt“. Verwenden Sie für das Training bitte folgende Einstellungen:

- Threshold für jedes verarbeitende Neuron
- Aktivierung für Bias Neuron ist konstant 1
- Batch Update
- Lernschrittweite: $\eta=0.8$
- Momentum Term: $\alpha=0.9$

Trainieren Sie bitte bis der quadratische Fehler über alle Muster E geringer ist als 0.01. Legen Sie die berechneten Gewichtswerte bitte in einer formatierten Datei ab.

Wie viele Epochen werden für das Training unter den gegebenen Randbedingungen benötigt?

Aufgabe 3

Stellen Sie bitte die von Ihnen berechnete Lösung für das XOR-Problem in einer Ebene dar, so dass insbesondere erkennbar wird, ob mit Ihrer Lösung die beiden Klassen linear oder nichtlinear separiert werden. Generieren Sie für diese Darstellung Muster, die in ihren Komponenten von den 4 Mustern des XOR-Problems abweichen und damit die Fläche abdecken.

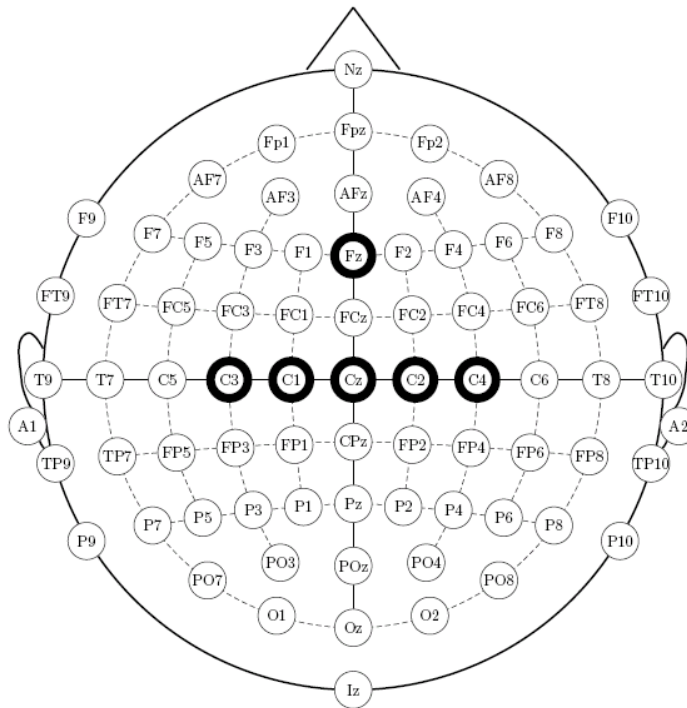
Aufgabe 4

Die technische Schnittstelle zwischen einem menschlichen Gehirn und einem Computer wird als Brain-Computer Interface (BCI) bezeichnet. Mit einem BCI können Computer mit Hilfe von neuronalen Aktivitäten, die durch Gedanken oder ausgeführte Aktivitäten ausgelöst wurden, gesteuert werden. Vor diesem Hintergrund wird auch die aktuelle Thematisierung des Forschungsfeldes in den Medien als „Gedankenlesen“ verständlich. Für die Realisierung eines BCI

müssen somit neuronale Aktivitäten gemessen und klassifiziert werden. Die klassifizierten Ereignisse werden zur Steuerung eines Computers verwendet. Für die Realisierung eines BCI existieren zwei sehr unterschiedliche Vorgehensweisen. Die erste Möglichkeit besteht in der Definition von typischen Signalmustern, die von den Nutzern für die Verwendung des BCI erlernt werden müssen. Die alternative Vorgehensweise besteht in der Adaption des BCI an den Nutzer. Hierzu müssen die Nutzer in einem Trainingszyklus die für die Steuerung vorgesehenen Gedankenabläufe bzw. Aktivitäten mehrfach ausführen, so dass eine Adaption der Parameter des BCI an die spezifischen Ausprägungen der neuronalen Aktivitäten des Nutzers erfolgen kann. Die zweite Vorgehensweise bietet für den Nutzer einen höheren Komfort, ist wissenschaftlich deutlich anspruchsvoller und entspricht dem Vorgehen der datengetriebenen Modellierung. Aus diesen Gründen wird im Kontext der Modellbildung und Simulation die Vorgehensweise der Adaption der Parameter des BCI an den Nutzer verfolgt. Für die nicht-invasive Messung der neuronalen Aktivitäten, d.h. der Hirnströme, kann bspw. ein Elektroenzephalogramm (EEG) oder ein Magnetoenzephalogramm (MEG) zum Einsatz kommen. Aktuell kommen für den Aufbau von BCIs meist EEGs zu Einsatz, da diese auf Grund ihrer Größe transportabel sind und eine sehr große Verbreitung haben, so dass die Anschaffungskosten attraktiv sind. Im Forschungsvorhaben wurde mit Daten gearbeitet, die mit Hilfe von EEGs gemessen wurden.

Es existiert eine internationale Forschungsszene, die sich mit der Erforschung von BCIs beschäftigt. Mit dem Ziel eine Vergleichbarkeit der vielfältigen Forschungsergebnisse zu erreichen, werden seit 2002 im zweijährigen Turnus vom Fraunhofer Institut für Rechnerarchitektur und Softwaretechnik in Berlin BCI Klassifikationswettbewerbe durchgeführt. Hierzu werden Daten zur Adaption der BCIs zur Verfügung gestellt und die Teilnehmer des Wettbewerbs müssen die Klassifikationsergebnisse für Daten einreichen, die ohne Information über die zu Grunde liegenden Ereignisse zur Verfügung gestellt werden. Die historischen Daten der Wettbewerbe stehen zur allgemeinen Verfügung. Für die Praktikumsaufgabe sollen Sie die Daten eines Datensatzes des zweiten Wettbewerbs mit Hilfe des von Ihnen realisierten KNN klassifizieren

Manuel Möbius hat die Daten für Sie aufbereitet, so dass die Durchführung der Aufgabe stark vereinfacht wird. Sie können natürlich auch die unverfälschten Daten des Wettbewerbs für die Aufgabe verwenden. Manuel Möbius hat mit Hilfe von Experimenten Elektroden identifiziert, die für die Klassifikation sehr gut geeignet sind. Die aufbereiteten Datensätze enthalten Daten von sechs Elektroden (FZ, C3, C1, CZ, C2, C4). Die Lokalisierung der Elektroden können Sie dem folgenden Schaubild entnehmen.



Sie finden zwei Datensätze mit jeweils Trainings- und Testdaten. Der erste Datensatz enthält Rohdaten, d.h. es wurde keine Parametrisierung durchgeführt. Der zweite Datensatz enthält parametrisierte Daten. Die Parametrisierung erfolgte mit Hilfe einer PCA (principal component analysis). Hierbei haben wir Ihnen die ersten 15 Hauptkomponenten zur Verfügung gestellt. Mit dem zweiten Datensatz sollten deutlich bessere Ergebnisse erreichbar sein. Es stehen 400 Trainings- und 80 Testmuster zur Verfügung. Die Testdaten dürfen natürlich für das Training nicht verwendet werden.

Jedes zu klassifizierende Muster der Rohdaten besteht aus $6 * 151$ Werten. Jedes Muster der parametrisierten Daten besteht aus $6 * 15$ Werten. Zu Beginn jedes Musters wurde die Klassenzugehörigkeit (rechts oder links) abgelegt, wobei „0 1“ links codiert und „1 0“ rechts codiert.

Bitte realisieren Sie einen Klassifikator für die Rohdaten und einen Klassifikator für die parametrisierten Daten. Messen Sie die Leistungsfähigkeit Ihrer Klassifikatoren für die Testmengen.