

# Computational Intelligence

M. Sc. Andreas Buschermöhle, M. Sc. Jan H. Schoenke

## Übungsblatt 7 - Neuronale Netze III: Autoencoder-Netze und Rekurrente Netze

Abgabe bis Mittwoch, 25.06.2014, 12:00 Uhr

---

In diesem Übungsblatt werden Algorithmen in Matlab geschrieben und zur Auswertung ein- und zweidimensionale Plots angefertigt. Erstellen Sie im Dateiordner Ihrer Gruppe in Stud.IP einen Ordner mit dem Namen **Blatt 7** und laden Sie zu zweit sowohl die m-Files, als auch die fertigen Matlab-Figures mit sinnvoller Achsenbeschriftung rechtzeitig in diesen Ordner hoch. Geben Sie in Vips je nach Aufgabenstellung die textuelle Interpretationen Ihrer Ergebnisse sowie ggf. die Namen der Dateien an, die sich auf die jeweilige Aufgabe beziehen.

---

### Aufgabe 7.1: Autoencoder-Netze (15 + 15 = 30 P)

In dieser Aufgabe soll ein Autoencoder-Netz umgesetzt werden, dass die eindimensionale Parametrierung einer zweidimensionalen Kurve erlernt. Aus einer Menge an Trainingsdaten, die der 2D-Kurve entstammen, wird ein Netz trainiert, das in einer verdeckten Schicht nur ein Neuron hat und als Ausgabe die Eingabe wiedergibt, sodass die Komprimierung auf eine eindimensionale Information erzwungen wird. Dann kann der hintere Teil des Netzes vom einzelnen Neuron zur Ausgabeschicht genutzt werden, um die parametrisierte Kurve auszuwerten.

- a) Erstellen Sie ein Autoencoder-Netz, dass mit dem Sinus auf dem Wertebereich  $[0, 2\pi]$  trainiert wird. Verwenden Sie für das Training 100 zufällig gewählte Daten von der Kurve. Die Eingabe an das Netz sind die x- und y-Koordinaten der Datenpunkte, die identisch als Ausgabe erwartet werden. Welche Topologie haben Sie für ihr Netz gewählt und wie gut konnte die Identität damit erlernt werden?
- b) Reduzieren Sie nun das trainierte Autoencoder-Netz auf den auswertenden Teil und prüfen Sie die Ausgabe des Netzes für Eingaben aus dem Wertebereich  $[-1, 1]$ , indem Sie die Resultate plotten. Liegen Die Datenpunkte auf der gewünschten Kurve und ist ein monotoner Zusammenhang zwischen der Kurve und den Eingabewerten zu erkennen?

*Tipp: Eine Variante, die Eingabe an eine verdeckte Schicht zu simulieren, besteht darin, alle vorherigen Gewichte auf 0 zu setzen, die Eingabe auf 0 zu setzen und das Gewicht des Bias-Neurons für die verdeckte Schicht direkt mit der Eingabe zu belegen.*

Die Zeitreihenvorhersage ermöglicht es, aus vergangenen und gegenwärtigen Informationen zukünftige Informationen vorherzusagen. Am Beispiel der Mackey-Glass-Gleichung soll diese Anwendung von Neuronalen Netzen untersucht werden. Die Mackey-Glass-Gleichung ist als Differenzialgleichung gegeben:

$$\frac{dx}{dt} = \beta \cdot \frac{\theta^n \cdot x(t - \tau)}{\theta^n + x(t - \tau)^n} - \gamma \cdot x(t)$$

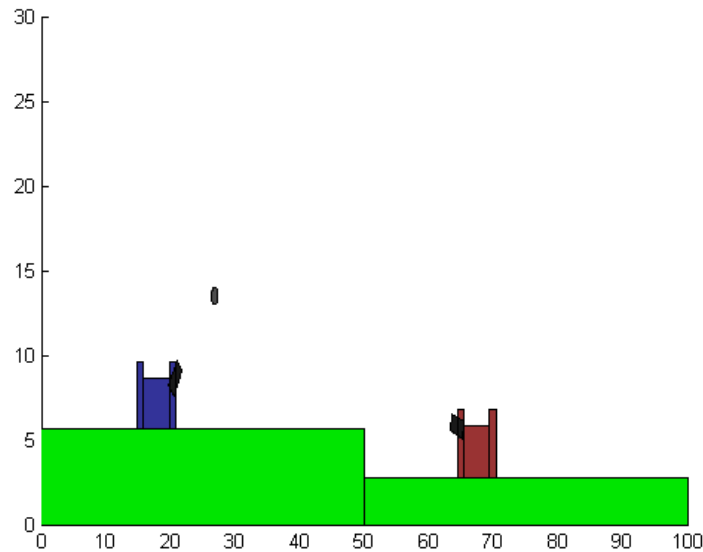
Eine Lösung mit einfacher Näherung der Differentialgleichung ist in Stud.IP gegeben und ermöglicht es, einen zeitlichen Verlauf ausgehend von Anfangswerten zu erhalten:

```
x = mackeyglass(x0, xt0, len)
```

Die Lösung enthält bereits Standardwerte für die Parameter der Gleichung und benötigt nur die Anfangswerte für  $x(t = 0)$ ,  $x(t < 0)$  für den Rückgriff in den ersten  $\tau$  Zeitschritten sowie die Anzahl an Zeitschritten, für die die Gleichung ausgewertet werden soll. Verwenden Sie als Startwerte  $x(t = 0) = 1$  und  $x(t < 0) = 0$ .

- a) Implementieren Sie ein MLP, das den jeweils nächsten Zeitschritt aus dem aktuellen Wert und dem Wert vor  $\tau = 17$  Schritten vorhersagt. Nutzen Sie die ersten 300 Daten zum Training des Netzes. Sagen Sie dann in jedem Schritt mit dem MLP den nächsten Wert voraus und vergleichen Sie diesen mit der tatsächlichen Ausgabe. Bilden Sie hierzu den mittleren quadratischen Fehler (MSE) über die nächsten 1000 Schritte.
- b) Zum Vergleich sollen zwei naive Ansätze für die Prädiktion umgesetzt werden. Der erste Ansatz liefert als Vorhersage  $x_p$  den Wert des aktuellen Zeitschritts:  $x_p(t + 1) = x(t)$ . Der zweite Ansatz nutzt die Änderung zwischen den letzten beiden Werten um eine lineare Extrapolation zu berechnen:  $x_p(t + 1) = x(t) + (x(t) - x(t - 1))$ . Nutzen Sie diese beiden Verfahren, um ebenfalls den MSE für 1000 Vorhersagen nach Schritt 300 zu bestimmen. Interpretieren Sie ihre Ergebnisse im Vergleich zur MLP-Vorhersage.
- c) In den beiden ersten Teilaufgaben wurde nur eine Ein-Schritt-Vorhersage getroffen, in der immer wieder aktuelle reale Daten vorliegen. Schwieriger ist eine stabile Vorhersage über mehrere Schritte, da hier das Vorhersage-Ergebnis wieder als Eingabe für die nächste Vorhersage genutzt wird und kein Abgleich mit der Realität stattfindet. Verwenden Sie Ihr trainiertes MLP, um ab Schritt 300 die nächsten 1000 Schritte vorherzusagen, wobei als Eingabe das Ergebnis der zuvor getroffenen Vorhersage genutzt wird. Bestimmen Sie auch hier den MSE und plotten Sie zusätzlich den Verlauf ihrer Vorhersage im Vergleich zum tatsächlichen Verlauf der Daten. Interpretieren Sie Ihre Ergebnisse.

In dieser Aufgabe soll das Konzept der Vorhersage von Daten mittels MLP angewendet werden. Hierzu wurde der Grabenkrieg aus Übungsblatt 5 leicht abgewandelt. Die beiden Schlösser stehen nun in jeder Runde auf einem Hügel mit zufälliger Höhe. Dadurch ist es nicht mehr ohne weiteres möglich, aus dem Winkel und der Schussstärke auf die Entfernung zu schließen, die zurückgelegt wird. Als Trainingsdaten liegen nun nicht mehr die Winkel, Schussgeschwindigkeit und Entfernung vor, sondern der Verlauf der Geschwindigkeit der Kannonenkugel über die Zeit. In einem Cell-Array sind die Verläufe für alle bisher getätigten Schüsse eines Spiels gespeichert. Über 100 Spiele soll wiederum ein selbst programmierter Spieler gegen den naiven Gegner, der bereits aus Blatt 5 bekannt ist, antreten.



Für diese Aufgabe liegen im Stud.IP die folgenden drei Dateien: `Grabenkrieg.m` bildet den Rahmen für das Spiel und soll nicht verändert werden. `shootBall.m` stellt die Spielphysik und -visualisierung zur Verfügung und soll nicht verändert werden. `trainAndEvalShooter.m` stellt die Intelligenz des Spielers dar und soll in den folgenden Aufgabenteilen ergänzt werden.

- Entwickeln Sie ein MLP, dass aus den bisher abgefeuerten Schüssen lernt, die Zeitreihe der Geschwindigkeiten in x- und y-Richtung vorherzusagen. Wie haben Sie die Netzstruktur gewählt und das Training aufgesetzt?
- Zusammen mit der gegebenen Information über die Positionen der Schlösser und die Höhe der Hügel kann nun iterativ mit dem trainierten MLP der Auftreffpunkt ermittelt werden. So kann gezielt eine Einstellung für den Winkel und die Geschwindigkeit gefunden werden (z.B. über eine Partikel-Schwarm-Optimierung oder Evolutionäre Strategie), mit der der Gegner getroffen werden kann. Setzen Sie diese Methode der Wahl der Schussbedingungen um und beschreiben Sie Ihr Vorgehen.