Übungsblatt 13

Aufgabe 1

Vergleichen Sie LSTM und RNN mit jeweils einer verborgenen Schicht hinsichtlich ihrer Fähigkeit, Long-Term-Dependencies zu erkennen. Erzeugen Sie dazu einen Datensatz mit m Zeitreihen

$$(x_1, \ldots, x_n, y)$$

der Länge $n + 1$ mit folgenden Eigenschaften:

- Für 50% der Zeitreihen gilt $x_1 = y = -0.5$. Für die anderen 50% gilt $x_1 = y = 0.5$.
- Die Werte $x_2, \ldots, x_n$ folgen einer Gleichverteilung aus dem Intervall $[-0.3, +0.3]$.
- Die Sequenzen $(x_1, \ldots, x_n)$ sind die Merkmalsvektoren, die in das Netz eingegeben werden. Die Werte y sollen vom Netz vorhergesagt werden.

Verwenden Sie für den Vergleich der Leistungsfähigkeit beider RNN-Varianten den RMSE. Experimentieren Sie mit verschiedenen Längen n und verschiedener Anzahl von verborgenen Neuronen. Was beobachten Sie?

Frage: Warum lässt sich dieses Problem mit linearer Regression einfach lösen?

Aufgabe 2

In dieser Aufgabe sollen Sie ein Character-Level Language Model mit LSTM in PyTorch implementieren, um neuen Text zu generieren. Verwenden Sie dazu das Buch `moby_dick.txt` (oder ein anderes Buch Ihrer Wahl).

Gehen Sie wie folgt vor:

1. Lesen Sie die Datei `moby_dick.txt` in Python ein.
2. Führen Sie eine Tokenisierung auf Character-Level durch, um eine Liste von eindeutigen Zeichen (Tokens) zu erhalten. Diese Liste von Tokens ist das Vokabular.
3. Erstellen Sie ein Mapping von jedem Token des Vokabulars zu einem Index und umgekehrt.
4. Fassen Sie das eingelesene Dokument als eine Liste von Zeichen auf. Konvertieren Sie die Liste der Zeichen in eine Liste von Indizes unter Verwendung des Mappings.
5. Erzeugen Sie einen Trainingsdatensatz wie in Blatt 12, Aufgabe 3. Merkmalsvektoren `x` bestehen aus n aufeinanderfolgenden Zeichen (Integers) $x_{t-n+1}, ..., x_t$ und x_{t+1} ist das zugehörige Ausgabelabel y.

6. Implementieren Sie ein LSTM-Modell, das Merkmalsvektoren der Länge n als Eingabe akzeptiert und das nächste Zeichen vorhersagt. Trainieren Sie das Modell mit dem Datensatz.
7. Verwenden Sie das trainierte Modell, um einen neuen Text zu generieren. Wählen Sie ein zufälliges Eingabebeispiel aus der Trainingsmenge aus und lassen Sie dann das Modell die nächsten 1000 Zeichen erzeugen.

Hinweise:

- Normalisieren Sie die Merkmalsvektoren, so dass die Werte der Merkmale zwischen 0 und 1 liegen.
- Auf eine Einbettung (z.B. One-Hot-Encoding) kann verzichtet werden.
- Es handelt sich um ein Klassifikationsproblem.