8. Übungsblatt

Aufgabe 38 RNN

Betrachten Sie ein RNN mit einer versteckten Schicht, die wie folgt aktiviert wird :

$$h_{t+1} = Wx_t + Uh_t + b$$

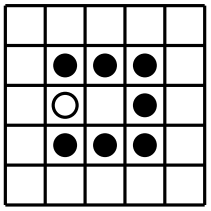
$$W, U \in \mathbb{R}^{4\times4}, b \in \mathbb{R}^{4\times1}$$

In den folgenden Skizzen betrachten wir den Agenten ∘, der sich entlang des Pfades von • bewegen soll. Die Eingabe x_t in das RNN ist die Umgebung des Agenten, wobei sich der Vektor wie folgt zusammensetzt:

$$x_t = \begin{bmatrix} \uparrow \\ \rightarrow \\ \leftarrow \\ \downarrow \end{bmatrix} \in \{0,1\}^4$$

Hierbei soll ein Eintrag 1 sein, wenn sich in der jeweiligen Richtung ein • befindet. Die Ausgabe h_t soll dabei analog zu x_t die nächste Richtung des Agenten angeben. In der folgenden Anwendung wird nach dieser Ausgabe der Agent in die Richtung bewegt und beeinflusst die Eingabe x_{t+1}

Bestimmen sie für das Labyrinth unten mit $h_0 = (1,0,0,0)^T$ die Gewichtsmatrizen W, U, b, sodass der Agent dem Pfad aus • folgt.



Könnte ein normales MLP dieses Problem lösen? Was können wir über die Berechnungsfähigkeit der RNNs ableiten?

Aufgabe 39 Gradienten in RNNs

Eine besondere Herausforderung bei RNNs ist das bestimmen eines geeigneten Gradientens. Dabei ist ein Aspekt das Lernen von Abhängigkeiten zwischen Elementen am Anfang und am Ende der Sequenz z.B. Subjekt und Verb in einem Nebensatz.
Wir betrachten nun als Vereinfachung nur die Verbindungen zwischen versteckten Neuronen mit linearer Aktivierungsfunktion.

$$h_t = U^t \cdot h$$

Weiterhin sei $f(h_k)$ der Fehler, den die versteckte Schicht macht. Bestimmen Sie den Gradienten der Fehlerfunktion für U in Abhängigkeit von $k > 1$. Nehmen Sie dazu an, dass U eine Diagonalenmatrix ist.

- Wie verhält sich der Gradient, wenn $U = U_1$ bzw. U_2 ist, für sehr große t?

$$U_1 = \begin{bmatrix} 1.2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1.1 \end{bmatrix}, U_2 = \begin{bmatrix} 0.8 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.9 \end{bmatrix}$$

- Warum sind diese Werte problematisch für Gradienten-basiertes Training?

- Was sind mögliche Strategien, die bei den Problemen helfen können ?

Aufgabe 40 Architekturen

Schlagen Sie für die folgenden Anwendungsgebiete jeweils eine Architektur vor und erklären Sie warum die jeweilige Architektur geeignet ist.

- Sie haben eine Sammlung von den Werken von Shakespear und wollen Texte im gleichen Stil generieren. Dabei sollen Sie auf der Ebene von Buchstaben arbeiten.
- Sie haben eine Sammlung von Katzen- und Hundebilder und wollen bei neuen Bildern entscheiden, ob es sich um einen Hund oder eine Katze handelt.
- Sie sollen ein Model entwickeln, das markiert, ob ein Buchstabe in einem Text zur wörtlichen Rede gehört oder nicht.
- Sie sollen ein Model entwickeln, das Englisch in Russisch und Deutsch übersetzt.
- Sie sollen ein Model entwickeln, das in der Lage ist einen Code zu zu reduzieren und den ursprünglichen Code zu rekonstruieren.

Aufgabe 41 Bonus: Automatische Differenzierungsframework (6)

Dieses Mal wollen wir Convolution differenzierbar implementieren, da es mitverantwortlich ist für den Erfolg von Deep Learning. Dabei gehen wir wie folgt vor :

a) Implementieren Sie als erstes den forward pass der convolution auf numpy arrays in der Funktion h_conv2d. Dabei machen wir folgende Annahmen:

   - Der Stride ist immer 1.
   - Der Kernel hat immer eine ungerade Größe.
   - Es gibt kein Padding.
   - Der Dilationfaktor ist 1.

- Das Bild hat die Dimensionen $Batch \times channels \times height \times width$.
- Der Kernel hat die Dimensionen $out\_channels \times channels \times kernel\_height \times kernel\_width$.

b) Implementieren Sie nun den backward pass auf der Basis von numpy arrays in der Funktion h_conv2d_grad. Diese Funktion soll den Gradienten bezüglich des Bildes und des Kernels für die implementierte convolution zurückgeben.

c) Implentieren Sie die differenzierbare conv2d Funktion mit Hilfe ihrer bereits implementierten Funktionen.

d) Implentieren Sie den Conv2D Layer analog zum Linearen Layer. Hier können Sie noch einen Bias in die Convolution einfügen.

e) Nun nutzen Sie ihren implementierten Layer in der Experimentvorlage um ein CNN zu trainieren, das MNIST löst. Den Datensatz bekommen Sie unter `http://yann.lecun.com/exdb/mnist/`. Dieser Datensatz dient zur Erkennung von handgeschriebenen Zahlen.