

Project work part II: Questions on Deep Learning

Prof. Dr.-Ing. Stache

Fill out the following:

Name: Gustav Willig

Matrikel-Nr. / Student-ID: 197332

I confirm that the work I handed in is my own work. All sources are cited.

I know that my work will be checked for plagiarism, and I accept that any occurrence of plagiarism and/or non-cited sources will lead to a grade of 5.0 for this part. The due date for this work to upload to ILIAS is July 22, time: 23:59, uploads after this due are not accepted.

Questions:

Q1: What is an embedding, how is it determined, what does the parameter “size” of an embedding mean, to which parameter does it correspond in a neural network? (4 P)

A:

Als **word-embedding** bezeichnet man die Abbildung einer beliebigen Anzahl von Wörtern auf Wortvektoren mit fester Länge. Das Besondere dabei ist, dass diese Vektoren auch die Semantik und den Kontext des Wortes berücksichtigen. Zudem kann der Abstand zwischen zwei Vektoren die Beziehung zwischen diesen beschreiben z.B. ist der Abstand zwischen Katze und Hund geringer als zu Hovercraft.

Embeddings ist ein **fully-connected layer**, welcher neben der Semantik auch den Kontext des Inputs berücksichtigt. Die Gewichte werden als **embedding weights** bezeichnet (siehe Skip-Grams-Solution.ipynb).

Der Parameter **size** bezeichnet die Form der **embedding** Matrix z.B. bei 10000 Wörtern und 300 versteckten Neuronen (= **hidden units**) beträgt die **size** 10000 x 300.

Die **size** korrespondiert mit dem **input layer** und den **hidden units**.

Q2: What does the parameter lstm_size in contrast to lstm_layers mean? (2 P)

A:

LSTM steht für **long short-term memory**.

lstm_size: Anzahl der Einheiten in der versteckten Schicht in der LSTM-Zelle. Gängige Werte hierfür sind 128, 256, 512 (siehe jupyter-notebook „Sentiment_RNN_Solution.ipynb“).

Unterschied zwischen lstm_layers: **lstm_layers** ist die Anzahl der LSTM-Schichten in einem Netz (siehe jupyter-notebook „Sentiment_RNN_Solution.ipynb“).

Q3: What is a typical range of the number of LSTM-layers in a recurrent neural network?

(1 P)

A:

Es gibt keine typische Range, es kommt immer auf das Problem und die verfügbare Hardware für das Training an. Normalerweise (siehe jupyter-notebook „Sentiment_RNN_Solution.ipynb“) startet man bei einer Schicht und fügt Neue hinzu, falls eine Unteranpassung (= **underfitting**) vorliegt.

Q4: What would happen if we used all the data we have for training a network?

(1 P)

A:

Das Modell würde beim Training, Validieren und Testen (da dieselben Daten verwendet werden) eine Genauigkeit von 100% erreichen. Somit ist eine richtige Validierung des Modells nicht möglich. Was dazu führt, dass **overfitting** nicht erkannt wird. Außerdem wird das Modell höchstwahrscheinlich eine schlechte Genauigkeit aufweisen, wenn unbekannte Daten verwendet werden.

Q5: What is the purpose of a validation dataset and how does it contrast to the test dataset?

(2 P)

A:

Zweck der Validierungsdaten: Mit den Validierungsdaten wird nach jeder Epoche überprüft, wie gut die Prädikationen des veränderten Modells sind.

Unterschied zu Testdaten: Testdaten sollen niemals zum Training des Netzwerkes eingesetzt werden (**golden rule**). Diese dienen dazu an der Endversion des Modells zu überprüfen, ob das Modell die zugrundliegende Struktur der Daten erkannt hat.

Q6: What is meant by the principle of weight sharing in the context of convolutional neural networks and what is the effect of weight sharing?

(2 P)

A:

Weight sharing bedeutet, dass Neuronen einer Schicht dieselben Gewichte benutzen. Wie in Abbildung 1 zu sehen ist nutzen alle Neuronen derselben Schicht denselben Gewichtsvektor. Der Einsatz von **weight sharing** erfolgt oft in **convolutional layers** (Fei-Fei, Li; Johnson, Justin; Yeung, o. J.).

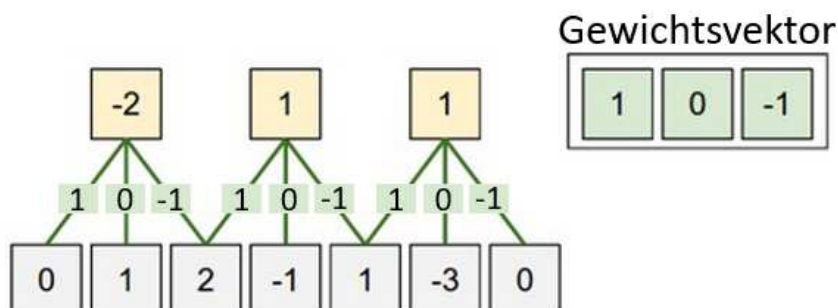


Abbildung 1: Weight sharing (Fei-Fei, Li; Johnson, Justin; Yeung, o. J.)

Effekt von **weight sharing**:

⇒ Verringerung der Anzahl an Parametern und somit eine Reduktion der Komplexität

⇒ Verringerung der Gefahr von **overfitting** (Modell kann besser generalisieren)

Q7: You observe during training of a network that the validation error increases whereas the training error decreases. What happened? What can you do about it? (2 P)

A:

What happened?

Das ist ein klares Zeichen dafür, dass **overfitting** vorliegt. Zu Beginn verbessert sich das Modell mit zunehmender Trainingsdauer, wenn das Training allerdings zu lange dauert kommt es zu dem oben beschriebenen Effekt, dass die Trainingsgenauigkeit steigt während die Validierungsgenauigkeit abnimmt. Einer der Gründe ist, dass das Modell sich zu sehr an die Trainingsdaten angepasst hat und nicht mehr das zugrundeliegende Muster lernt.

What can you do about it?

- Die Anzahl der Parameter des Netzes reduzieren. Dies führt zu einer Reduzierung der Komplexität, wodurch die Gefahr von **overfitting** verringert wird.
- Mehr Trainings- und Validierungsdaten nutzen (durch den Einsatz von Generatoren können aus den vorhandenen Originalbildern zusätzliche Bilder erzeugt werden).
- Einfügen von Dropoutschichten
- Einsatz von **cross-validation** (z.B. mit **k-fold** aus **sklearn**)

Q8: What is meant by data augmentation? (1 P)

A:

Als **data augmentation** wird die Erhöhung des vorhandenen Bestands an Validierungs- und Trainingsdaten bezeichnet. **Data augmentation** wird oft angewandt um den Bestand an Images zu erhöhen. Aus einem Originalbild können durch drehen und flippen neue Datensätze geschaffen werden (siehe untere Abbildung).

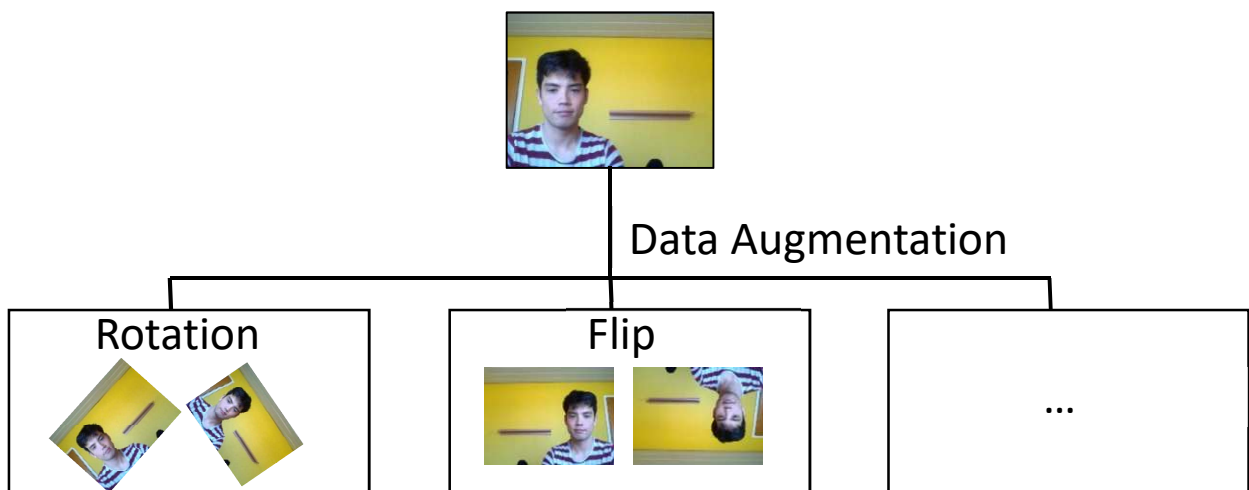


Abbildung 2: Darstellung von data augmentation

Literaturverzeichnis

Fei-Fei, Li; Johnson, Justin; Yeung, S. (o. J.). CS231n Convolutional Neural Networks for Visual Recognition. Abgerufen 22. Juli 2018, von <http://cs231n.github.io/convolutional-networks/>