

## Übungsblatt 11

Ausgabe: 03.07.2017  
 Abgabe: 10.07.2017

Diesmal benötigen Sie die zu scikit-learn kompatible Python-Bibliothek scikit-neuralnetwork sowie scipy, numpy und theano in Version 0.7 (bei neueren Versionen bestehen zurzeit noch Kompatibilitätsprobleme zu scikit-neuralnetwork):

- [https://scikit-neuralnetwork.readthedocs.io/en/latest/guide\\_installation.html](https://scikit-neuralnetwork.readthedocs.io/en/latest/guide_installation.html)
- <https://www.scipy.org/install.html>
- <http://www.numpy.org/>
- <http://deeplearning.net/software/theano/install.html>

Reichen Sie Ihren kommentierten Quellcode per E-Mail als \*.py- oder \*.ipynb-Datei ein:  
[holldack@thi.cs.uni-frankfurt.de](mailto:holldack@thi.cs.uni-frankfurt.de)

### Aufgabe 11.1 Neuronale Netze mit scikit-neuralnetwork

(32 Punkte)

In dieser Aufgabe wollen wir Feedforward-Netzwerke mit Schichten  $V_0, \dots, V_T$  und genau einem Ausgabegatter  $(T, 1)$  verwenden, um geometrische Formen im Einheitsquadrat zu lernen. In der Vorlesung haben wir gesehen, dass neuronale Netzwerke die Konjunktion von Threshold-Funktionen, d. h. den Schnitt von Halbräumen, ausdrücken können und auf Blatt 4 haben wir gezeigt, dass das schwache Konsistenzproblem für den Schnitt von Halbräumen NP-hart ist. Wir wollen dieses Ergebnis im Folgenden experimentell untersuchen.

- a) Konstruieren Sie mithilfe der Klasse `Classifier` ein möglichst einfaches Netzwerk, welches ein Beispiel  $(x_1, x_2) \in [0, 1]^2$  genau dann positiv klassifiziert, wenn  $x_1 \geq x_2$  gilt.

Untersuchen Sie den Einfluss der folgenden Parameter mithilfe sogenannter *Restarts*. Führen Sie die Experimente für jede Parameterwahl mehrfach durch und beobachten Sie, welchen Einfluss die Parameter auf den durchschnittlichen Trainings- bzw. Testfehler und die empirische Standardabweichung haben. Wie „stabil“ verhalten sich die Fehlerraten?

- i) Gatter-Typ (`Linear`, `Rectifier`, `Sigmoid` und `Tanh`)
  - ii) Anzahl der Iterationen (`n_iter`) und Schrittweite (`learning_rate`) des stochastischen Gradientenabstiegs
  - iii) Anzahl und Margin der Trainingsbeispiele
- b) Konstruieren Sie nun ein möglichst einfaches Netzwerk, welches ein Beispiel  $(x_1, x_2) \in [0, 1]^2$  genau dann positiv klassifiziert, wenn  $\frac{1}{2}x_1 \leq x_2 \leq 2x_1$  gilt, und untersuchen Sie analog zu Teilaufgabe a) den Einfluss der Parameter.
- c) Leiten Sie aus Ihren experimentellen Ergebnissen aus a) und b) jeweils möglichst gute Gewichte und Schwellenwerte für beide Netzwerke ab. (Die verwendeten Gatter dürfen Sie frei wählen.) Plotten Sie je eine Heatmap für die Wahrscheinlichkeit, dass ein Beispiel  $x \in [0, 1]^2$  positiv klassifiziert wird.

*Hinweis:* Um die Ausgabe Ihres Netzwerks für Ihre Wahl der Gewichte und Schwellenwerte zu bestimmen, genügt es, Algorithmus 12.2 im Skript zu implementieren.

*Hinweis zur Verwendung von scikit-neuralnetwork:*

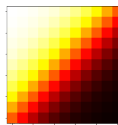
Je zwei Schichten  $V_i$  und  $V_{i+1}$  sind vollständig miteinander verbunden, d. h. für alle  $u \in V_i$  und  $v \in V_{i+1}$  existiert im Netzwerk die gewichtete Kante  $(u, v)$ . Verwenden Sie für die Ausgabe ein Softmax-Gatter, das die Wahrscheinlichkeit beschreibt, dass ein eingegebenes Beispiel  $x \in [0, 1]^2$  durch das Netzwerk positiv bzw. negativ klassifiziert wird. Intern besteht ein Softmax-Gatter aus zwei Gattern, die die beiden Klassifikationen repräsentieren. Schwellenwerte werden in jeder Schicht implizit durch Bias-Knoten dargestellt.

`Classifier` erwartet die Beispiele und Klassifikationen als numpy-Array. Verwenden Sie `numpy.array([1,2,3])`, um ein numpy-Array mit den Elementen 1, 2 und 3 zu erzeugen.

Sie können die Gewichte und Schwellenwerte mit der Funktion `get_parameters()` auslesen.

Weitere nützliche Links:

- <https://scikit-neuralnetwork.readthedocs.io/en/latest/index.html>
- [https://scikit-neuralnetwork.readthedocs.io/en/latest/guide\\_model.html#classification](https://scikit-neuralnetwork.readthedocs.io/en/latest/guide_model.html#classification)
- [https://scikit-neuralnetwork.readthedocs.io/en/latest/module\\_mlp.html](https://scikit-neuralnetwork.readthedocs.io/en/latest/module_mlp.html)
- <https://stackoverflow.com/questions/33282368/plotting-a-2d-heatmap-with-matplotlib>



**Abbildung 1:** Beispiel für eine suboptimale Heatmap für Teilaufgabe a)