

- 4.5 Perceptron-Lernalgorithmus
- 4.6 Die Trenngeraden bzw. Hyperplanes oder auch Hyperebenen für das Beispiel
- 4.7 scikit-learn-kompatibler Estimator
- 4.8 scikit-learn-Perceptron-Estimator
- 4.9 Adaline
- 4.10 Zusammenfassung
- 4.11 Referenzen

5 Mehrschichtige

neuronale Netze

- 5.1 Ein echtes Problem
- 5.2 XOR kann man lösen
- 5.3 Vorbereitungen für den Start
- 5.4 Der Plan für die Umsetzung
- 5.5 Das Setup (`>>class<<`)
- 5.6 Die Initialisierung (`>>__init__<<`)
- 5.7 Was für zwischendurch
(`>>print<<`)
- 5.8 Die Auswertung (`>>predict<<`)
- 5.9 Die Verwendung

5.10 Zusammenfassung

6 Lernen im mehrschichtigen Netz

6.1 Wie misst man einen Fehler?

6.2 Gradientenabstieg an einem Beispiel

6.2.1 Gradientenabstieg – die Idee

6.2.2 Algorithmus für den Gradientenabstieg

6.3 Ein Netz aus sigmoiden Neuronen

6.4 Der coole Algorithmus mit Vorwärts-Delta und Rückwärts-

Propagation

6.4.1 »__init__«-Methode

6.4.2 »predict«-Methode

6.4.3 »fit«-Methode

6.4.4 »plot«-Methode

6.4.5 Alles im Konzert

6.5 Ein »fit«-Durchlauf

6.5.1 Initialisierung

6.5.2 Forward

6.5.3 Output

6.5.4 Hidden

6.5.5 ΔW_{kj}

6.5.6 ΔW_{ji}

6.5.7 W_{ji}

6.5.8 W_{kj}

6.6 Zusammenfassung

6.7 Referenz

7 Convolutional Neural Networks

7.1 Aufbau eines CNN

7.2 Der Kodierungsblock

7.2.1 Convolutional Layer

7.2.2 Activation Function

7.2.3 Pooling Layer

7.2.4 Überlappen, ausfüllen und
Schrittlänge

7.3 Der Prädiktionsblock

7.3.1 Flatten

7.3.2 Softmax

7.4 Trainieren von Convolutional Neural Networks

7.4.1 Das Problem der