Maschinelles Lernen
Aufgabenblatt 04

Prof. Dr. David Spieler
Hochschule München

14. Oktober 2019

Aufgabe 1 (Perceptronklassifikator) *Angenommen Sie haben den Perzeptronklassifikator*

$$f(\mathbf{x}) : \mathbb{R}^2 \rightarrow \{0, 1\}$$

mit dem Parametern $\mathbf{w}_0 = 2$, $\mathbf{w}_1 = -0.4$ *und* $\mathbf{w}_2 = 0.5$ *trainiert.*

1. *Welches geometrische Objekt ist die Entscheidungsoberfläche?*

2. *Zeichnen Sie die Entscheidungsoberfläche und markieren Sie den Halbraum, welcher positiv klassifiziert wird* $f(\mathbf{x}) = 1$ *mit einem Pluszeichen* $+$ *und den negativen Halbraum* $f(\mathbf{x}) = 0$ *mit einem Minuszeichen* $-$.

3. *Was müssten Sie an* $f(\mathbf{x})$ *ändern, um eine entgegengesetzte Klassifikation zu erhalten, also Klasse* 0 *bei Daten der aktuellen Klasse* 1 *und umgekehrt?*

4. *Berechnen Sie die Gewichte eines Perzeptronklassifikators, wenn Sie wissen, dass die Punkte* $(3, 0)^T$ *und* $(0, 3)^T$ *auf der Entscheidungsoberfläche liegen und dass der Ursprung* $(0, 0)^T$ *negativ klassifiziert wird. Sie können einen beliebigen Wert für* $\mathbf{w}_0$ *bestimmen solange die anderen Eigenschaften erfüllt sind.*

Aufgabe 2 (Lernen von Perzeptronen) *Wir möchten einen Perzeptronklassifikator*

$$f(\mathbf{x}) : \mathbb{R}^2 \rightarrow \{0, 1\}, f(\mathbf{x}) = \alpha(\mathbf{w}_1\mathbf{x}_1 + \mathbf{w}_2\mathbf{x}_2 + \mathbf{w}_0)$$

mit den Gewichten $\mathbf{w} = [\mathbf{w}_0, \mathbf{w}_1, \mathbf{w}_2]$ *trainieren andhand der Datenpunkte* $\mathbf{d}^{(1)} = [1, 2]^T$, $\mathbf{d}^{(2)} = [2, 3]^T$ *und* $\mathbf{d}^{(3)} = [2, 0]^T$ *mit den Klassen* 1, 0 *und* 1.

1. *Zeichnen Sie die drei Datenpunkte in ein Koordinatensystem. Wählen Sie für Klasse* 0 *Datenpunkte die Farbe Rot und für Klasse* 1 *Datenpunkte die Farbe Blau.*

2. *Angenommen, die Gewichte sind anfangs* $\mathbf{w} = \mathbf{0}$, *zeichnen Sie in das obige Koordinatensystem die Entscheidungsoberfläche in den gleichen Farben wie oben ein.*

3. *Führen Sie das Training schrittweise durch, bis alle Datenpunkte korrekt klassifiziert werden. Dies wird einige Iterationen dauern. Sie können sich mit einem Python-Skript behelfen. Was ist der endgültige Wert von* $\mathbf{w}$*? Wieviele Iterationen wurden benötigt?*

4. *Zeichnen sie nochmals die drei Datenpunkte in ein Koordinatensystem und zeichnen Sie die Entscheidungsoberfläche in den bekannten Farben.*

Aufgabe 3 (Perzeptron - Lernalgorithm) *Nun möchten wir die Perzeptronklassikfikation auf einen größeren Datensatz anwenden. Hierfür verwenden wir das Jupyter Notebook `ex04_03.ipynb`.*

1. *Führen Sie die Zellen bis* Implementierung *aus.*

2. *Welche besondere Eigenschaft besitzt der Datensatz, wenn Sie sich den Scatterplot ansehen?*

3. *Implementieren Sie die Funktionen `perceptron_classify`, `perceptron_learn_step`, `perceptron_accuracy` und `perceptron_learn`.*

4. *Trainieren Sie das Perzeptron anhand des geladenen Datensatzes. Wieviele Iterationen wurden benötigt, um eine 100%-ige Genauigkeit zu erzielen. Was sind die finalen Gewichte?*

Aufgabe 4 (Adaline Klassifikator) *Jetzt möchten wir einen Adaline Klassifiaktor auf dem gleichen Datensatz trainieren. Hierfür verwenden wir das Jupyter Notebook `ex04_04.ipynb`.*

1. *Implementieren Sie den Adaline Klassifikations- und Lernalgorithmus.*

2. *Wenden Sie den Adaline Lernalgorithmus auf den Datensatz `linsep.csv` an. Wieviele Iterationen werden benötigt, um eine 100%-ige Genauigkeit zu erreichen? Was sind die finalen Gewichte?*

3. *Experimentieren Sie mit verschiedenen Werten für* η*. Was passiert und warum?*