Technische Universität Braunschweig

Institut für Robotik und Prozessinformatik
Prof. J. Steil

Übungen für Grundlagen Maschinelles Lernen

SS 2019 Blatt 3

Übungstermin: 10.03.

Task 3.1 Gewichteter Fehler theoretisch: Nehmen Sie an, ein Datensatz $\vec{x}_n, t_n$ ist gegeben, wobei jeder Trainingsdatenpunkt n mit einem Faktor r_n gewichtet ist. Setze ein lineares Model $y_n = \vec{w}^T\phi(\vec{x}_n)$ an (für irgendein $\phi()$ und $n = 1..N$ Trainingsdaten) und betrachte die entsprechende Fehlerfunktion

$$E_D(\vec{w}) = \sum_{n=1}^{N} r_n \left(t_n - \vec{w}^T\phi(x_n)\right)^2 \tag{1}$$

Finde die Lösung für die optimalen Parameter $\vec{w}^\star$ und interpretiere die Parameter r_n im Sinne von (i) datenabhängiger Unsicherheit und (ii) Duplikationen von Datenpunktion.

Task 3.2 Gewichteter Fehler praktisch::

Verwenden Sie nun den Datensatz der Regressionsaufgabe von Blatt zwei und gewichten Sie die Punkte bei input -1/2 und 1/2 doppelt, 10-fach, 100-fach. Dazu können Sie das publizierte Matlab-Script leicht verändern. Verwenden Sie wieder Polynome zur Approximation und variieren Sie den Grad ? Was erwarten Sie, wie wird sich das optimale Modell mit der Wahl des Polynomgrades ändern.

Gewichten Sie dann auch noch den Punkt bei Eingabe -1 höher. Was ändert sich ? Welche Form wird das optimale Modell haben und warum ?

Task 3.3 Gewichteter Fehler: multiple lokale Modelle:

Nehmen Sie nun an, Sie generieren die Gewichte r_n aus einer Gaussfunktion mit einer Varianz β^2 (z.B. $\beta^2 = 1$) und Mittelwert μ. Welchen Effekt hat das auf die Modellierung ? Überlegen Sie dazu, welche Datenpunkte jeweils wichtig sind, wenn $\mu = -1/2, 0, 1/2$ ist.

Diese Überlegungen legen nahe, das Datenmodell als Summe mehrerer lokaler Modelle zu wählen. Wie könnten diese lokalen Modelle aussehen, wenn wir Gaussgewichtungen mit $\mu_1 = -1/2$ und $\mu_2 = 1/2$ und das Gesamtmodell als Summe von zwei lokalen Modellen wählen. Wie, wenn man drei lokale Modelle verwenden wollte ? Wo sollten dann die drei Mittelpunkte $\mu_{1,2,3}$ für die jeweiligen Gaussfunktionen liegen, mit denen die Gewichte erzeugt werden, und was für ein Modell (d.h. welche Wahl der jeweiligen Polynomgrade) wäre sinnvoll.

Task 3.4 Gradientenberechnungen für tanh-Modelle (sog. künstliche "Neuronen"): Zeige, dass gilt: $f(s) = tanh(\beta s)$ then $f'(s) = \beta(1 - f(s)^2)$. Berechne dann den Gradienten der folgenden Fehlerfunktion:

$\frac{\partial}{\partial w_i} E(\vec{w}) = \frac{\partial}{\partial w_i} 1/2 \sum_n (t_n - \tanh(\sum_{d=1}^{D} w_d x_d))^2,$

wobei wir einen einen D-dimensionalen Eingabevektor $\vec{x} \in R^D$ und eindimensionale Ausgabe y annehmen.