

Maschinelles Lernen und Data Mining
Sommersemester 2010
Übungsblatt 3

Besprechung des Übungsblattes am Freitag, den 04.06.2010

Aufgabe 3-1 Lineare Regression mit Gauss'schem Rauschen

Gegeben sei ein Datensatz D mit $d_i = (x_{i,1}, \dots, x_{i,M}, y_i)^T$ auf N Datenpunkten mit M Variablen, dessen Zielgröße y linear von \mathbf{X} abhängt. Aufgrund von technischen Ungenauigkeiten wurden die Eingangsvariablen von \mathbf{X} jedoch nur verrauscht aufgenommen, d.h.:

$$y_i = x_i^T \mathbf{w} + \epsilon_i,$$

wobei ϵ_i den Rauschfehler von Datenpunkt i darstellt. Nehmen wir weiter an, dass ϵ gaussverteilt ist:

$$P(\epsilon_i) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}\epsilon_i^2}.$$

Damit können wir die Verteilung von y in Abhängigkeit der Variablen \mathbf{X} und des Modells \mathbf{w} darstellen als

$$P(y_i|x_i, \mathbf{w}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(y_i - x_i^T \mathbf{w})^2}.$$

- a) Bestimmen Sie den Parameter $\hat{\mathbf{w}}$ der die Wahrscheinlichkeiten der Trainings-Daten $P(D|\mathbf{w})$ maximiert. Verwenden Sie hierfür den *Maximum-Likelihood Schätzer*: $\hat{\mathbf{w}}^{\text{ML}} = \arg \max_{\mathbf{w}} P(D|\mathbf{w})$.
Bei der Bestimmung der Likelihoodfunktion können Sie davon ausgehen, dass \mathbf{w} unabhängig von den Eingangsdaten \mathbf{X} verteilt ist.
- b) Eine beliebige a priori Verteilungsannahme für Zufallsvariablen in einem Bayes'schen Ansatz ist

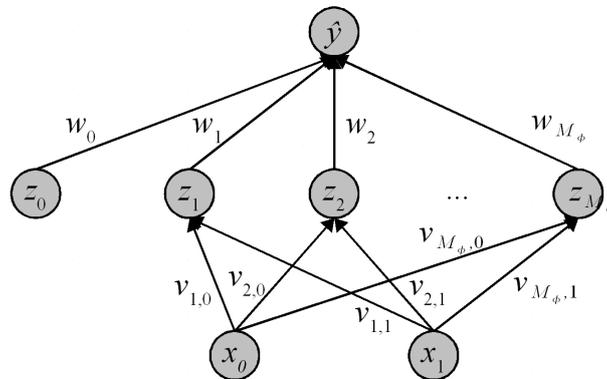
$$P(\mathbf{w}) = \frac{1}{(2\pi\alpha^2)^{\frac{M}{2}}} e^{\left(-\frac{1}{2\alpha^2} \sum_{j=0}^{M-1} w_j^2\right)}$$

Berechnen Sie den Parameter $\hat{\mathbf{w}}$, der den Ausdruck $P(\mathbf{w})P(D|\mathbf{w})$ maximiert. Ergibt sich dadurch eine neue Interpretation des λ -Termes aus der regulierten Kostenfunktion (*penalized least squares (PLS)*)?

b.w.

Aufgabe 3-2 Ein einfaches neuronales Netz
schriftlich bearbeiten

Unten abgebildet sehen Sie ein zweischichtiges neuronales Netz mit einem Eingabeneuron $x \in \mathbb{R}$ und je einem Biasneuron $x_0 = z_0 = 1$ (d.h. $\mathbf{x}_i = (1, x_{i,1})^T$) in der Eingabeschicht und der versteckten Schicht.



Als Aktivierungsfunktion der versteckten Neuronen verwenden wir einen Sigmoiden, also

$$z_h = \phi(\mathbf{x}_i, \mathbf{v}_h) = \frac{1}{1 + \exp\left(-\sum_{j=0}^M v_{h,j} x_{i,j}\right)},$$

das Ausgangsneuron \hat{y} wird wie üblich über eine Linearkombination gebildet.

- Zeigen Sie, dass gilt: $\frac{\partial z_h}{\partial v_{h,j}} = x_{i,j} \cdot z_h \cdot (1 - z_h)$
- Drücken Sie den maximalen Wert von \hat{y} in Abhängigkeit von \mathbf{w} aus, wenn alle Ausgangsgewichte w_h ($h \in \{0, \dots, M_\phi\}$) positiv sind. Was ist der minimale Wert?
- Wie sieht \hat{y} aus, wenn $v_{h,j} = 0$ für alle $j \in \{0, \dots, M\}, h \in \{1, \dots, M_\phi\}$? Welche Funktion erhalten Sie für \hat{y} , wenn alle $v_{h,j} = c, c \neq 0$?

Aufgabe 3-3 Basisfunktionen von Neuronalen Netzen

Die Ausgabe eines neuronalen Netzes für einen Testvektor \mathbf{x}_i ist definiert durch $f(\mathbf{x}_i) = \sum_{h=0}^{M_\phi-1} w_h \phi_h(\mathbf{x}_i, \mathbf{v}_h)$.

Die Gewichte der einzelnen Neuronen können über die Backpropagation-Regel mit musterbasiertem Gradientenabstieg gelernt werden. In der Vorlesung wurden die neuronalen Netze mit sigmoiden Neuronen vorgestellt. Natürlich können auch andere Basisfunktionen verwendet werden.

- Welche Eigenschaften müssen diese Basisfunktionen erfüllen?
- Ist eine Linearkombination $\phi(\mathbf{x}_i, \mathbf{v}_h) = z_h = \sum_{j=0}^M v_{h,j} x_{i,j}$ hierfür geeignet? Begründen Sie.
- Ist die Anzahl der Parameter für $\phi(\mathbf{x}_i, \mathbf{v}_h)$ beschränkt? Können mehrere verschiedene Basisfunktionen für ein neuronales Netz verwendet werden?

b.w.

Aufgabe 3-4 Kernkombinationen
schriftlich bearbeiten

Um einen selbstdefinierten Kern $k(\mathbf{x}_i, \mathbf{x}_j)$ für $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^n$ anzuwenden, muss für gewöhnlich gezeigt werden, dass es sich auch tatsächlich um einen legitimen Kern handelt. Da es recht aufwendig sein kann zu zeigen, dass für k das *Mercer Theorem* zutrifft, wird häufig explizit das Mapping der impliziten Basistransformationen angegeben: $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$.

Eine weitere beliebte Variante die Gültigkeit einer Kernfunktion zu zeigen ist die Rückführung auf eine Kombination aus Kernels, da für einige Operationen \circ gilt, dass $k(\mathbf{x}_i, \mathbf{x}_j) = k_1(\mathbf{x}_i, \mathbf{x}_j) \circ k_2(\mathbf{x}_i, \mathbf{x}_j)$ ein legitimer Kernel ist.

Zeigen Sie dass für valide Kernelfunktionen $k_l(\mathbf{x}_i, \mathbf{x}_j)$ mit $l \in 1, \dots, n$ gilt:

- a) **Skalierung:** Für $a > 0$ ist $k(\mathbf{x}_i, \mathbf{x}_j) := a k_1(\mathbf{x}_i, \mathbf{x}_j)$ ein Kernel.
- b) **Summe:** $k(\mathbf{x}_i, \mathbf{x}_j) := k_1(\mathbf{x}_i, \mathbf{x}_j) + k_2(\mathbf{x}_i, \mathbf{x}_j)$ ist ein Kernel.
- c) **Linearkombination:** Für $w \in \mathbb{R}_+^n$ ist $k(\mathbf{x}_i, \mathbf{x}_j) := \sum_{l=0}^{n-1} w_l k_l(\mathbf{x}_i, \mathbf{x}_j)$ ein Kernel.
- d) **Produkt:** $k(\mathbf{x}_i, \mathbf{x}_j) := k_1(\mathbf{x}_i, \mathbf{x}_j) \cdot k_2(\mathbf{x}_i, \mathbf{x}_j)$ ist ein Kernel.
- e) **Potenz:** Für ein $p \in \mathbb{N}_+$: $k(\mathbf{x}_i, \mathbf{x}_j) := k_1(\mathbf{x}_i, \mathbf{x}_j)^p$ ist ein Kernel.