Formeln

Lineare Regression

Linearer Zusammenhang zwischen den Eingabevariablen x und der Ausgabevariable y wird modelliert.

Hypothesenfunktion:
$h_{\theta(x)} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n$

Kostenfunktion (MSE):
$J(\theta) = \frac{1}{2n} \sum_{i=1}^{n} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$

Ziel:
Finde Parameter θ um J zu minimieren
$\min J(\theta)$

Multivariat:
Mehrere Features $x_1, x_2, \ldots, x_n$

Polynom-Regression:
$h_{\theta(x)} = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \ldots$

Gradient Descent

Update-Regel:
$\theta_j := \theta_j - \alpha \ \texttt{frac}\{\partial\}\{\partial\theta_j\} J(\theta)$

Für lineare Regression:
$\theta_j := \theta_j + \alpha \ \texttt{frac}\{1\}\{n\} \sum_{\{i=1\}}^{n} \left(y^{\{(i)\}} - h_{\theta(x^{\{(i)\}})}\right) \cdot x_j^{\{(i)\}}$

Lernrate α:
Zu groß $\longrightarrow$ Divergenz,
zu klein $\longrightarrow$ langsame Konvergenz

Logistische Regression

Sigmoidfunktion:
$g(z) = \texttt{frac}\{1\}\{1 + e^{\{-z\}}\}$

Hypothese:
$h_{\theta(x)} = g(\theta^T x)$

Klassifikation:
$h_{\theta(x)} ge 0.5 \rightarrow$ Klasse 1
$h_{\theta(x)} < 0.5 \rightarrow$ Klasse 0

Entscheidungsgrenze:
$\theta_0 + \theta_1 x_1 + \theta_2 x_2 = 0$

Nicht-linearität durch Features wie x_1^2, $x_1 x_2$, dots

Regularisierung

Kostenfunktion mit L2-Regularisierung:
$J(\theta) = \texttt{frac}\{1\}\{2n\} \sum \left(h_{\theta(x^{\{(i)\}})} - y^{\{(i)\}}\right)^2 + \lambda \sum_{\{j=1\}}^{d} \theta_j^2$

Effekt von λ:
- $\lambda = 0 \longrightarrow$ kein Penalty
- großes $\lambda \longrightarrow$ starke Bestrafung, Underfitting

Bias-Term θ_0 wird oft nicht regularisiert

Support Vector Machines

Ziel:
$m \underset{\{w,b\}}{\in} \left(\texttt{frac}\{1\}\{2\} \ |w|^2 + C \sum xi_i\right)$

Nebenbedingungen:
$y^{\{(i)\}}\left(w^T x^{\{(i)\}} + b\right) ge 1 - xi_i$ mit $xi_i ge 0$

C kontrolliert Trade-off:
großes $C \longrightarrow$ weniger Fehler,
kleines $C \longrightarrow$ größerer Margin

Kernel-Trick:
z. B. $K(x, x') = e^{\{-\gamma \ |x - x'|^2\}}$ (RBF-Kernel)

Neuronale Netzwerke

Feedforward:
$z^{\{(l+1)\}} = \theta^{\{(l)\}} a^{\{(l)\}}$
$a^{\{(l+1)\}} = g\left(z^{\{(l+1)\}}\right)$

Backpropagation:
$\delta^{\{(L)\}} = a^{\{(L)\}} - y$
$\delta^{\{(l)\}} = \left(\theta^{\{(l)\}}\right)^T \delta^{\{(l+1)\}} . * g'\left(z^{\{(l)\}}\right)$

Gradientenabstieg:
$\theta^{\{(l)\}} := \theta^{\{(l)\}} - \alpha \delta^{\{(l)\}} a^{\{(l-1)\}}$

Aktivierungsfunktionen:
Sigmoid, Tanh, ReLU, Leaky ReLU, Softmax

Convolutional Neuronal Networks

Modell Evaluation

Entscheidungsbäume

Pricipal Component Analysis (PCA)

von Eva Hölzl