Übung zur Vorlesung

Maschinelles Lernen 1

Wintersemester 2010/2011

Abteilung Maschinelles Lernen
Institut für Softwaretechnik und
theoretische Informatik
Fakultät IV, Technische Universität Berlin
Prof. Dr. Klaus-Robert Müller
Email: klaus-robert.mueller@tu-berlin.de

Blatt 11

Abgabe bis Mittwoch, den 26. Januar 2011 um 12:00

Die Lösungen bitte im **Postfach** von Dr. Konrad Rieck abgeben und **elektronisch** an `konrad+lehre@mlsec.org` senden.

RBF-Netze

In der Vorlesungen wurden Radialbasisfunktionsnetze (RBF-Netze) behandelt. Zur Erinnerung: Ein RBF-Netz mit m Zentren berechnet die Funktion

$$\hat{f}(x) = \sum_{i=1}^{m} k(x, \mu_i)\alpha_i, \qquad \text{wobei} \qquad k(x, z) = \exp\left(-\frac{\|x - z\|^2}{2\sigma}\right).$$

Wir betrachten den Fall, dass die Gewichte $\alpha_i \in \mathbb{R}$ und die Zentren $\mu_i \in \mathbb{R}$ gelernt werden, die Kernbreite σ jedoch fest bleibt. Das Netz hat also die Parameter $\theta = (\alpha_1, \mu_1, \ldots, \alpha_m, \mu_m)$. Während des Lernens soll der quadratische Fehler

$$E(\theta) = \frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{f}_\theta(x_j))^2$$

auf den Trainingsdaten minimiert werden. Für einen Trainingspunkt (x, y) lauten die Gradienten

$$\frac{\partial E}{\partial \alpha_i} = \epsilon \cdot k(x, \mu_i), \qquad \frac{\partial E}{\partial \mu_i} = \frac{1}{\sigma}\epsilon \cdot \alpha_i k(x, \mu_i)(x - \mu_i),$$

mit $\epsilon = y - \hat{f}(x)$.

1. **Implementation eines RBF-Netzes (25 Punkte)** Implementiere das RBF-Netz mit adaptiven Zentren μ_i und Gewichten α_i. Schreibe

   - eine Funktion `init_rbfnet`, die das Netz zufällig initialisiert,
   - eine Funktion `update_rbfnet`, die das Netz mit Trainingspunkten updatet und
   - eine Funktion `apply_rbfnet`, die Vorhersagen auf Testpunkten macht.

   Verwende für die Implementierung das Programmskelett. Tipps: (a) Für die Initialisierung setze $\alpha_i = \frac{1}{m}$ und wähle die Zentren zufällig im Interval $(-\pi, \pi)$.

2. **Anwendung eines RBF-Netzes (5 Punkte)** Teste das RBF-Netz auf dem folgenden Datensatz. Ziehe $x_1, ..., x_{100}$ aus einer uniformen Verteilung auf $(-\pi, \pi)$. Sei

   $$y_i = \text{sinc}(4x_i) + r_i \quad \text{mit} \quad r_i \sim \mathcal{N}(0, 0.04) \quad \text{und} \quad \text{sinc}(x) = \begin{cases} \frac{\sin(x)}{x} & x \neq 0 \\ 1 & x = 0. \end{cases}.$$

   Finde durch Probieren drei Werte von σ, die eine geschätzte Funktion liefern, die stark underfittet/overfittet/die wahre Funktion gut approximiert.

```
function sheet11
clf;

% Configuration
iters = 200;
sigmas = [ ... ];
eta = 0.1;
centers = 20;

% Generate data
[X,Y] = gen_data(100);

% Compute RBF network for different sigmas
for s = 1:length(sigmas)

   % Init network
   C = init_rbfnet(sigmas(s), centers);

   % Train network in chunks of 10
   err = zeros(1, iters);
   for i = 1:iters
       idx = randperm(length(X));
       idx = idx(1:10);

       C = update_rbfnet(C, X(idx), Y(idx), eta);
       P = apply_rbfnet(C, X);

       % Determine error
       err(i) = mean((Y - P).^2);
   end

   % Make nice plots
   subplot(2,3,s); hold on;
   plot(X,Y,'.'); plot(X,P,'r-');
   subplot(2,3,s+3);
   plot(err);
end


% Init RBF network
% Input:
%   m     Number of RBF centers
%   w     Kernel width (sigma)
% Output:
%   C     Struct with alphas, mus and sigma
%
function C = init_rbfnet(w, m)
   C.sigma = ...
   C.alphas = ...
   C.mus = ...

% Apply RBF network
% Input:
%    C    Struct with alphas, mus and sigma
%    X    Testing data (1 x n)
% Output:
%    Y    Predictions (1 x n)
%
function Y = apply_rbfnet(C, X)
   ...
```

```
% Update RBF network
% Input:
%    C:    Struct with alphas, mus and sigma
%    X:    Training data (1 x n)
%    Y:    Training labels (1 x n)
%    eta:  Learning rate
% Output:
%    C:    Struct with updated alphas, mus and sigma
function C = update_rbfnet(C, X, Y, eta)
   ...


% Generate data
% Input:
%    n    Number of data points
% Output:
%    X    Data points (1 x n)
%    Y    Labels (1 x n)
function [X,Y] = gen_data(n)
   ...
```

Für Fragen zum Übungsblatte bitte in der Google Group http://groups.google.com/group/ml-tu registrieren und die Frage an die Mailingliste stellen.