Machine Learning 1
Lecture 3.3 - Supervised Learning
Stochastic Gradient Descent

Erik Bekkers

(Bishop 3.1.3)
Stochastic gradient descent

- for $N \gg 1$ \( w_{ML} = \left( \Phi^T \Phi \right)^{-1} \Phi^T t \) is very costly to compute!
  - Needs to process all data \((x_1, \ldots, x_N)\) at once.
  - Matrix inversion of $M \times M$ matrix: \( O(M^3) \)

- Loss is a sum of error terms for each datapoint:

\[
E_D(w) = \sum_{i=1}^{N} E(x_i, t_i, w)
\]

\[
E(x_i, t_i, w) = \frac{1}{2} \left( t_i - w^T \phi(x_i) \right)^2
\]

- Approach for large dataset: stochastic gradient descent

approximating the total error with less data points

\( E_D \rightarrow \) a way for minimizing
Recap: The Gradient

- The gradient encodes all directional derivatives via scalar product:

\[ \nabla_w E := \frac{\partial}{\partial w} E = \left( \frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \ldots, \frac{\partial E}{\partial w_m} \right) \]

- The gradient is always perpendicular to the contours of a function.

- The gradient always points in the direction of steepest ascent.
Stochastic gradient descent

\[ E_D(w) = \sum_{i=1}^{N} E(x_i, t_i, w) \]

\[ E(x_i, t_i, w) = \frac{1}{2} (t_i - w^T \phi(x_i))^2 \]

- Stochastic gradient descent:
  - Initialize \( w^{(0)} \), choose learning rate \( \eta \).
  - Iterate over data points, and update

\[
w^{(\tau+1)} = w^{(\tau)} - \eta (\nabla_w E(x_i, t_i, w^{(\tau)}))^T
= w^{(\tau)} + \eta (t_i - \hat{w}^{(\tau)} \phi(x_i)) \phi(x_i)
\]

- If \( E_D(w) \) is convex in \( w \) and \( \eta \) is small enough: convergence