Neuronale Netze (SS 2002), 29.5.

Training

- **Further gradient descent methods:**
  - Manhattan training: the derivative determines only the sign but not the size of the steps

    Pros and Cons: faster in plateaus, possibly less oscillation for large derivatives, but a little less stable and possible oscillation; still a limited effect, learning rate is still crucial
  - SuperSAB, DeltaBARDelta (only offline): use a separate, adaptive learning rate for each weight

    Pros and Cons: nice idea, learning rate is no longer critical, because it is adaptive; the effect might be cancelled by the size of the derivative, possible instabilities for rapid changes
  - RProp (only offline): use a separate, multiplicative adapted learning rate for each weight, ignore the size of the derivatives, take back bad steps

    Pros and Cons: very robust and very fast method, one of the best (and simplest) training methods for FNNs; but: overfitting very easily possible, possibly local optima, only offline method
- **line-search methods:**
  - Steepest descent: compute the gradient and optimize via line search into that direction

    Pros and Cons: no learning rate; but the gradient is not optimum even for quadratic functions, oscillation for not rotation-symmetric functions
  - conjugate gradient descent: search in a better direction, i.e.

    $$\text{search direction} = \text{gradient} + \alpha \text{ old search direction}$$

    α is determined such that the gradient in the new search direction is orthogonal to the old search direction (in second approximation)

Pros and Cons: no learning rate, fast, guaranteed convergence for quadratic functions after —W— steps; unpredictable behavior for functions which are dissimilar to quadratic functions

- **second order methods**
  - Newton method: $\vec{w}_{t+1} = \vec{w}_t - H^{-1}(\vec{w}_t) grad(\vec{w}_t)$ where H = matrix of second derviatives

    Pros and Cons: no learning rate, but very inefficient due to matrix incersion
  - Quickprop: approximate H by a diagonal matrix, derivatives by difference quotients

    Pros and Cons: gives a fast method similar to backprop where the 'learning rate'is automatically determined from second order information; very instable with respect to the situation because of a very rough approximation
- **Global optimization methods** like simulated annealing, taboo search, genetic algorithms, evolutionary algorithm, modified random search, ...