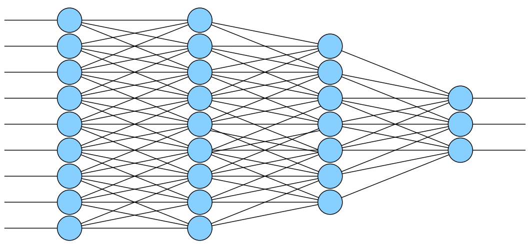
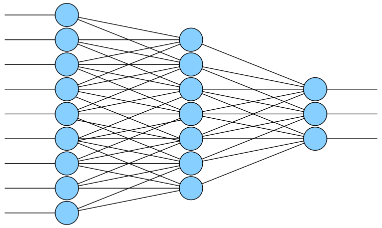
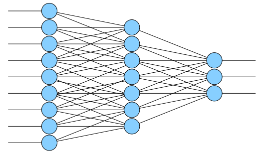
Alternatives to BackProp



- Multilayer feedforward network
- Arbitrary decision boundaries/functions



- Classification
- Function approximation
- Trained with prescribed outputs
- Batch or Incremental learning



Learning:

Minimize the error (E) as a function of all weights ($\vec{w}$)

1. Compute the direction in weight space where the error increases most: $\mathrm{grad}_{\vec{w}}(E)$
2. Change the weights in the opposite direction

$$w_i \leftarrow w_i - \eta \frac{\partial E}{\partial w_i}$$

- All computation can be performed locally
- Slow convergence
- Requires short step lengths

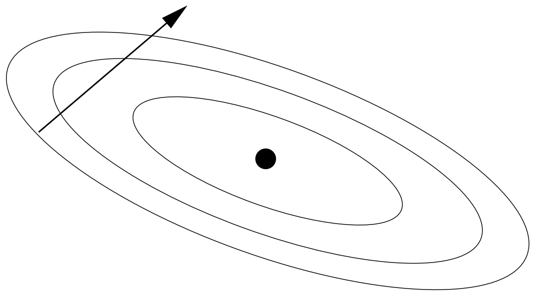
BackPropagation
- Basic Principle

Faster Alternatives
- The Problem
- The Solution

Measuring Performance
- Generalization

Regularization
- Limiting the Complexity
- Punishing Large Weights
- Optimal Pruning

Does the gradient point in the right direction?



- Incremental learning
- Large steps
- High-dimensional space

- Normalization
- De-correlation

These techniques only help on the global scale

Works for "toy-problems" but not when there is a lot of structure in the task

ANN fk
Örjan Ekeberg
BackPropagation
Faster Alternatives
The Problem
The Solution
Measuring Performance
Regularization

Idéa: Make use of the second derivative too

Ordinary gradient following

$$\Delta w = -\eta \frac{\partial E}{\partial w}$$

Newtons method

$$\Delta w = \left(\frac{\partial^2 E}{\partial w^2}\right)^{-1} \frac{\partial E}{\partial w}$$

Makes it necessary to invert a very large matrix!

ANN fk
Örjan Ekeberg
BackPropagation
Faster Alternatives
The Problem
The Solution
Measuring Performance
Regularization

Conjugate Gradient Method

- Established numerical method
- Incremental updates are made in directions where they do not counteract each other
- Does not require explicit computation of the second derivative

ANN fk
Örjan Ekeberg
BackPropagation
Faster Alternatives
The Problem
The Solution
Measuring Performance
Regularization

Conjugate Gradient Method:

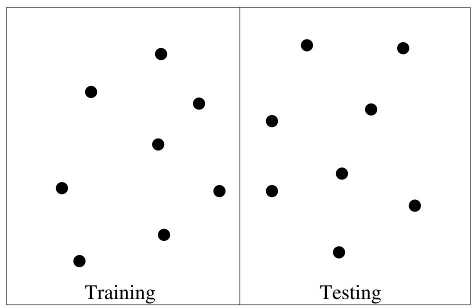
- Initiate:
  $\vec{r} \leftarrow -\frac{\partial E}{\partial \vec{w}}$
  $\vec{s} \leftarrow \vec{r}$
- Repeat:
  - Find η which minimizes $E(\vec{w} + \eta\vec{s})$
  - $\vec{w} \leftarrow \vec{w} + \eta\vec{s}$
  - $\vec{r} \leftarrow -\frac{\partial E}{\partial \vec{w}}$
  - $\beta = \max\left[\frac{\vec{r}^T \cdot (\vec{r} - \vec{r}_{old})}{\vec{r}_{old}^T \cdot \vec{r}_{old}}, 0\right]$
  - $\vec{s} \leftarrow \vec{r} + \beta\vec{s}$

How can one measure the performance of a neural network?

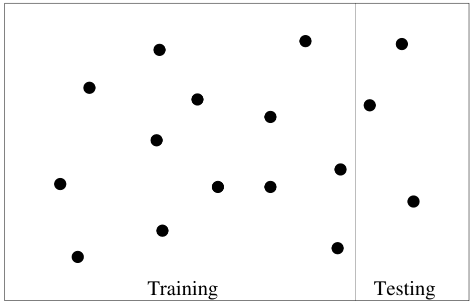
- Evaluation of a classifier
- Positive and negative errors
- The error for evaluation does not have to be the same as the error minimized during learning

Performance should always be measured on a separate test data set

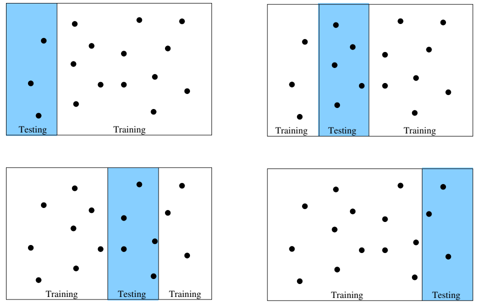
Separation of *training* and *testing* data



How large should the test data set be?



Maximal utilization of available data



Average over different partitionings

Many weights $\Rightarrow$ Bad generalization

Risk of making errors

- $\mathcal{E}_s$ — Empirical risk (measurable)
- $\mathcal{E}_c$ — Structural risk

$$R(\vec{w}) = \mathcal{E}_s(\vec{w}) + \lambda \mathcal{E}_c(\vec{w})$$

λ — regularization parameter

Augment the cost function with a complexity term

Weight Decay (Hinton, 1989):

$$\mathcal{E}_c = \sum_i w_i^2$$

Weight Elimination (Weigend et al., 1991):

$$\mathcal{E}_c = \sum_i \frac{(w_i/w_0)^2}{1+(w_i/w_0)^2}$$

Alternative technique

Remove "unnecessary" weights after training

Optimal Brain Damage (LeCun et al., 1990)

Idéa: Remove the least important weight

- Estimate how much the error increases when a weight is set to zero
- $\Delta w_i = -w_i$
- Error increase $-w_i \cdot \frac{\partial \mathcal{E}}{\partial w_i} = 0$ (since BP has converged) Oops
- Second derivative $\frac{w_i^2}{2} \cdot \frac{\partial^2 \mathcal{E}}{\partial w_i^2}$

ANN fk
Örjan Ekeberg
BackPropagation
Faster Alternatives
Measuring Performance
Regularization
Limiting the Complexity
Punishing Large Weights
Optimal Pruning

Optimal Brain Surgeon

(Hassibi et al., 1992)

- Improved version of Optimal Brain Damage
- Takes into account that other weights may need readjustment
- Requires an estimate of mixed second derivatives (Hessian-matrix)
- Can be efficiently estimated using the errors of the individual patterns

ANN fk
Örjan Ekeberg
BackPropagation
Faster Alternatives
Measuring Performance
Regularization
Limiting the Complexity
Punishing Large Weights
Optimal Pruning