Bayesian learning of neural networks

Johan Suykens

K.U. Leuven, ESAT-STADIUS
Kasteelpark Arenberg 10
B-3001 Leuven (Heverlee), Belgium
Email: johan.suykens@esat.kuleuven.be
http://www.esat.kuleuven.be/stadius

Lecture 5

Overview

- Bayes theorem
- Model comparison, Occam's razor principle
- regression using a multilayer perceptron
- parameters and hyperparameters
- levels of inference
- prior, posterior, likelihood, evidence
- error bars and predictions
- automatic relevance determination

Bayes theorem (1)

- Events A, B
  Model assumption $\mathcal{H}$

- Bayes Theorem:

$$P(B|A,\mathcal{H}) = \frac{P(A|B,\mathcal{H})P(B|\mathcal{H})}{P(A|\mathcal{H})}$$

(notation $P(A|B,\mathcal{H})$ means: probability for having A **given** that we have B and $\mathcal{H}$)

Bayes theorem (2)

- Model $\mathcal{H}$ parameterized by parameter vector w
  (e.g. interconnection weights of multilayer perceptron)

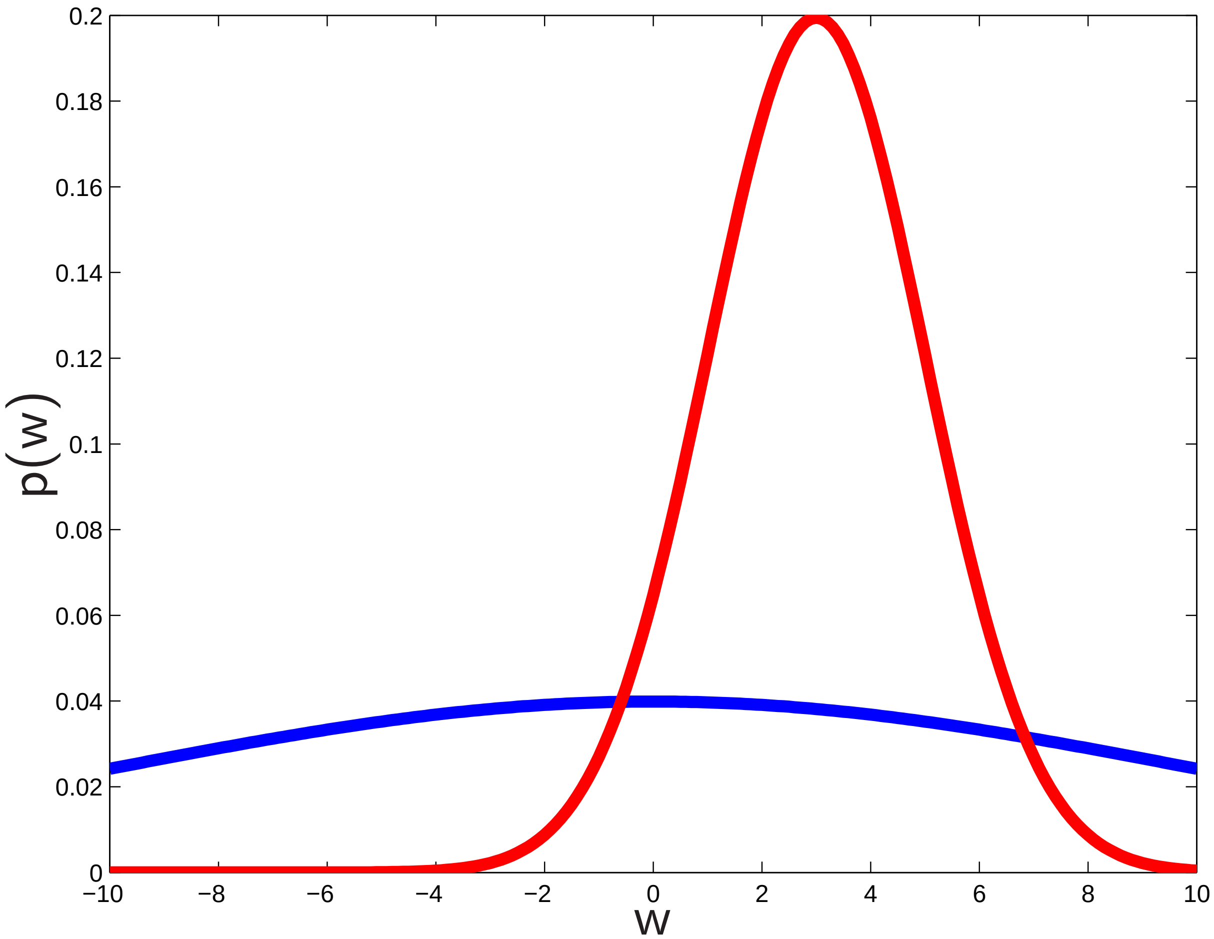
- Bayes Theorem:

$$P(w|D,\mathcal{H}) = \frac{P(D|w,\mathcal{H})P(w|\mathcal{H})}{P(D|\mathcal{H})}$$

  meaning

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

From prior to posterior



Starting from a prior distribution $p(w)$, using the data D, one obtains the posterior distribution $p(w|D)$.

Model comparison

- Consider two alternative models $\mathcal{H}_1, \mathcal{H}_2$ and data D
- From

$$
\begin{aligned}
P(\mathcal{H}_1|D) &= \frac{P(D|\mathcal{H}_1)P(\mathcal{H}_1)}{P(D)} \\
P(\mathcal{H}_2|D) &= \frac{P(D|\mathcal{H}_2)P(\mathcal{H}_2)}{P(D)}
\end{aligned}
$$

Model comparison

- Consider two alternative models $\mathcal{H}_1, \mathcal{H}_2$ and data D
- From

$$
\begin{aligned}
P(\mathcal{H}_1|D) &= \frac{P(D|\mathcal{H}_1)P(\mathcal{H}_1)}{P(D)} \\
P(\mathcal{H}_2|D) &= \frac{P(D|\mathcal{H}_2)P(\mathcal{H}_2)}{P(D)}
\end{aligned}
$$

one obtains

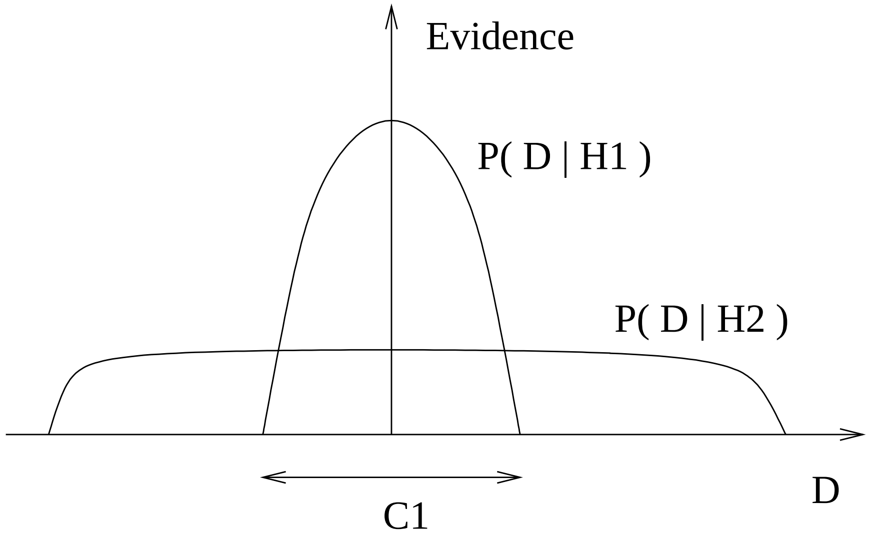
$$
\frac{P(\mathcal{H}_1|D)}{P(\mathcal{H}_2|D)} = \frac{P(\mathcal{H}_1)}{P(\mathcal{H}_2)} \frac{P(D|\mathcal{H}_1)}{P(D|\mathcal{H}_2)}
$$

Occam's razor principle (1)

- *"Simple models should be preferred"*

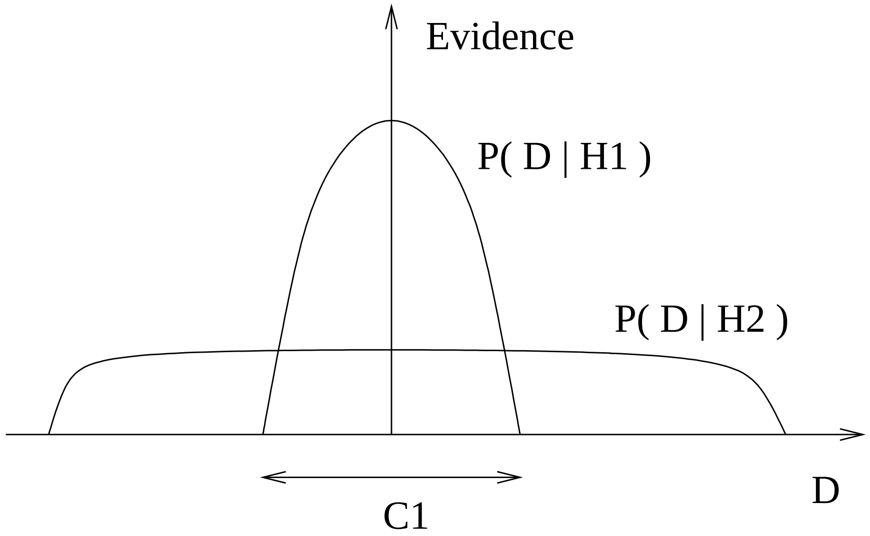
- Bayes Theorem embodies Occam's razor automatically!

Occam's razor principle (2)



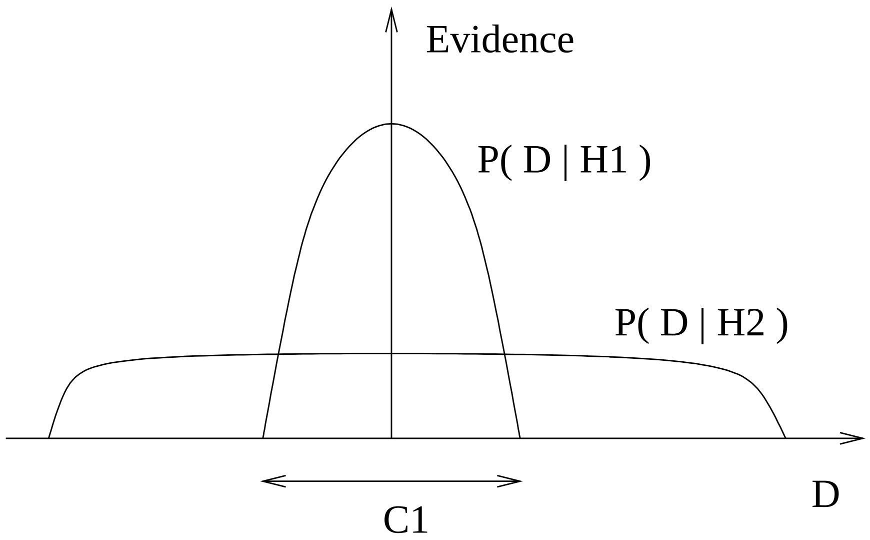
- Suppose $P(\mathcal{H}_1) = P(\mathcal{H}_2)$

Occam's razor principle (2)



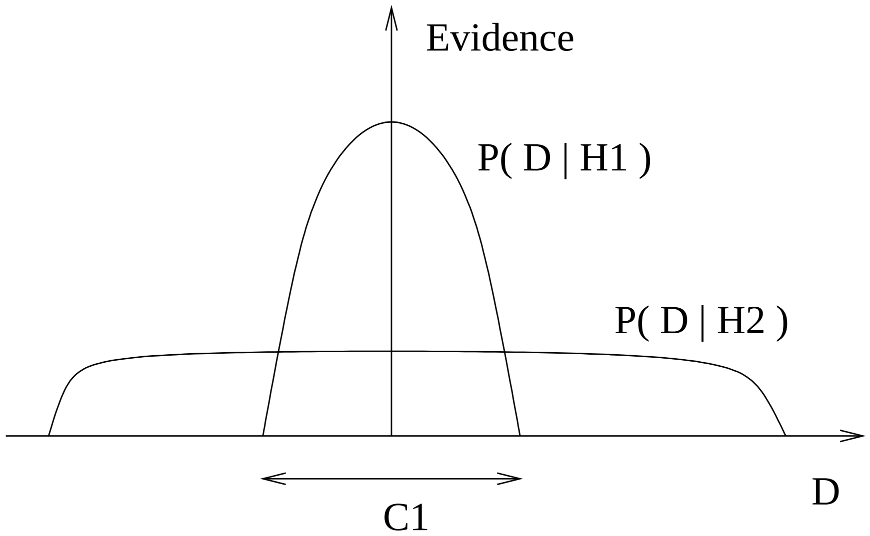
- Suppose $P(\mathcal{H}_1) = P(\mathcal{H}_2)$
- Simple model $\mathcal{H}_1$ makes a limited range of predictions (shown by $P(D|\mathcal{H}_1)$)

Occam's razor principle (2)



- Suppose $P(\mathcal{H}_1) = P(\mathcal{H}_2)$
- Simple model $\mathcal{H}_1$ makes a limited range of predictions (shown by $P(D|\mathcal{H}_1)$)
- The more complex model $\mathcal{H}_2$ is able to predict a larger variety of data sets

Occam's razor principle (2)

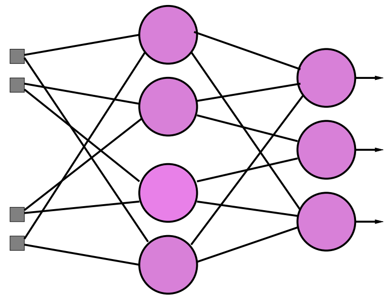


- Suppose $P(\mathcal{H}_1) = P(\mathcal{H}_2)$
- Simple model $\mathcal{H}_1$ makes a limited range of predictions (shown by $P(D|\mathcal{H}_1)$)
- The more complex model $\mathcal{H}_2$ is able to predict a larger variety of data sets
- If the data fall in region C_1 the model $\mathcal{H}_1$ is more probable because

$$\frac{P(\mathcal{H}_1|D)}{P(\mathcal{H}_2|D)} = \frac{P(D|\mathcal{H}_1)}{P(D|\mathcal{H}_2)} > 1$$

Regression using a multilayer perceptron

- **Training data** $D = \{(x^{(n)}, t^{(n)})\}_{n=1}^{N}$
  Input data $x^{(n)}$, target data $t^{(n)}$



- **Model** $\mathcal{H}$ given by:

$$y(x; w)$$

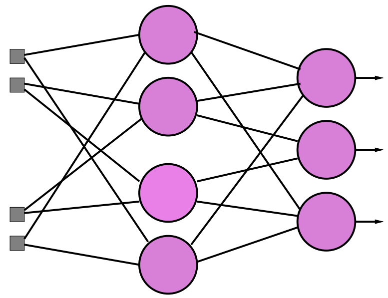
  is parametrized by w (interconnection weights)

-

Regression using a multilayer perceptron

- **Training data** $D = \{(x^{(n)}, t^{(n)})\}_{n=1}^{N}$
  Input data $x^{(n)}$, target data $t^{(n)}$

- **Model** $\mathcal{H}$ given by:



$$y(x; w)$$

  is parametrized by w (interconnection weights)

- **Objective:**

$$\min_{w} M(w) = \beta E_D(w) + \alpha E_W(w)$$

  with

$$E_D(w) = \frac{1}{2} \sum_{n=1}^{N} [t^{(n)} - y(x^{(n)}; w)]^2$$

$$E_W(w) = \frac{1}{2} \sum_{j} w_j^2$$

  (keep also the weights small when minimizing the training error!)

Connection objective function and Bayes theorem (1)

- **Bayes theorem:**

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

$$\Rightarrow \quad \log \text{Posterior} = \log \text{Likelihood} + \log \text{Prior} - \log \text{Evidence}$$

-

Connection objective function and Bayes theorem (1)

- **Bayes theorem:**

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

$$\Rightarrow \quad \log \text{Posterior} = \log \text{Likelihood} + \log \text{Prior} - \log \text{Evidence}$$

- Relate this to the **objective**:

$$\min_w M(w) = \beta E_D(w) + \alpha E_W(w)$$

with

$$\begin{array}{rcl} E_D(w) & \leftrightarrow & \log \text{Likelihood} \\ E_W(w) & \leftrightarrow & \log \text{Prior} \end{array}$$

Connection objective function and Bayes theorem (2)

- $\min_w M(w) = \beta E_D(w) + \alpha E_W(w)$ or
  $\max_w -M(w) = -\beta E_D(w) - \alpha E_W(w)$
  Relate this to "$\max \ \log \text{Posterior}$"

- Hence

$$\begin{array}{rcl} -M(w) & \leftrightarrow & \log \text{Posterior} \\ -\beta E_D(w) & \leftrightarrow & \log \text{Likelihood} \\ -\alpha E_W(w) & \leftrightarrow & \log \text{Prior} \end{array}$$

$\Rightarrow$

$$\begin{array}{rcl} \exp(-M(w)) & \leftrightarrow & \text{Posterior} \\ \exp(-\beta E_D(w)) & \leftrightarrow & \text{Likelihood} \\ \exp(-\alpha E_W(w)) & \leftrightarrow & \text{Prior} \end{array}$$

Connection objective function and Bayes theorem (2)

- $\min_w M(w) = \beta E_D(w) + \alpha E_W(w)$ or
  $\max_w -M(w) = -\beta E_D(w) - \alpha E_W(w)$
  Relate this to "$\max\ \log \text{Posterior}$"

- Hence

$$\begin{array}{rcl} -M(w) & \leftrightarrow & \log \text{Posterior} \\ -\beta E_D(w) & \leftrightarrow & \log \text{Likelihood} \\ -\alpha E_W(w) & \leftrightarrow & \log \text{Prior} \end{array}$$

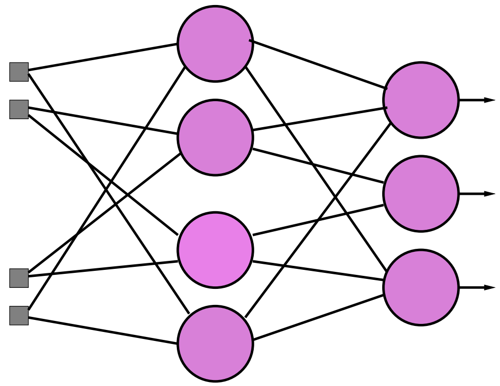
  $\Rightarrow$

$$\begin{array}{rcl} \exp(-M(w)) & \leftrightarrow & \text{Posterior} \\ \exp(-\beta E_D(w)) & \leftrightarrow & \text{Likelihood} \\ \exp(-\alpha E_W(w)) & \leftrightarrow & \text{Prior} \end{array}$$

  Using normalization factors Z_M, Z_D, Z_W one obtains

$$\begin{array}{rcl} \exp(-M(w))/Z_M & = & \text{Posterior} \\ \exp(-\beta E_D(w))/Z_D & = & \text{Likelihood} \\ \exp(-\alpha E_W(w))/Z_W & = & \text{Prior} \end{array}$$

Inference: different hierarchical levels



- Training of a multilayer perceptron: **which are all unknowns?**

  1. parameters w: all unknown interconnection weights
  2. hyperparameters α, β: regularization constants to be determined
  3. number of hidden units, leading to different models $\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, ...$

- the Bayesian inference is treated at different **hierarchical levels**

Levels of inference

- Level 1: parameters w

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

- Level 2: hyperparameters α, β

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

- Level 3: model comparison

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

Levels of inference

- Level 1: parameters w

$$\max_{w} \ \text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

- Level 2: hyperparameters α, β

$$\max_{\alpha,\beta} \ \text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

- Level 3: model comparison

$$\max_{\mathcal{H}_1,\mathcal{H}_2,\mathcal{H}_3,\mathcal{H}_4,\ldots} \ \text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

Levels of inference

- Level 1: parameters w

$$P(w|D,\alpha,\beta,\mathcal{H}) = \frac{P(D|w,\alpha,\beta,\mathcal{H})P(w|\alpha,\beta,\mathcal{H})}{P(D|\alpha,\beta,\mathcal{H})}$$

- Level 2: hyperparameters α, β

$$P(\alpha,\beta|D,\mathcal{H}) = \frac{P(D|\alpha,\beta,\mathcal{H})P(\alpha,\beta|\mathcal{H})}{P(D|\mathcal{H})}$$

- Level 3: model comparison

$$P(\mathcal{H}|D) = \frac{P(D|\mathcal{H})P(\mathcal{H})}{P(D)}$$

Levels of inference

- Level 1: parameters w

$$P(w|D,\alpha,\beta,\mathcal{H}) = \frac{P(D|w,\alpha,\beta,\mathcal{H})P(w|\alpha,\beta,\mathcal{H})}{\mathbf{P}(\mathbf{D}|\alpha,\beta,\mathcal{H})}$$

- Level 2: hyperparameters α, β

$$P(\alpha,\beta|D,\mathcal{H}) = \frac{\mathbf{P}(\mathbf{D}|\alpha,\beta,\mathcal{H})P(\alpha,\beta|\mathcal{H})}{\mathbf{P}(\mathbf{D}|\mathcal{H})}$$

- Level 3: model comparison

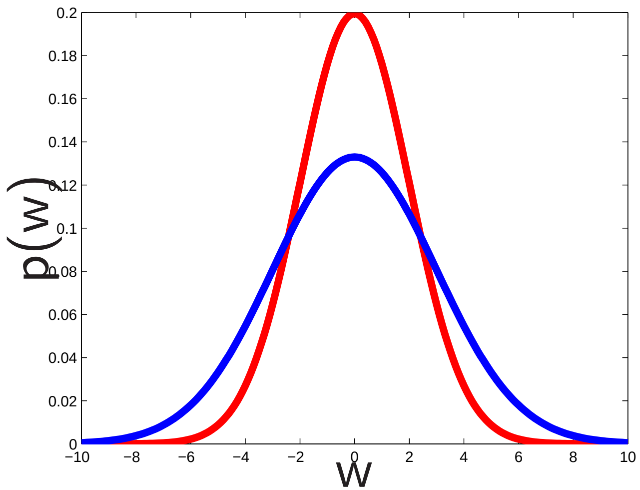
$$P(\mathcal{H}|D) = \frac{\mathbf{P}(\mathbf{D}|\mathcal{H})P(\mathcal{H})}{P(D)}$$

Level 1

- **Prior distribution:**

$$P(w|\alpha, \mathcal{H}) \propto \exp(-\alpha \sum_j w_j^2)$$

α large: many emphasis on making the weights small
α small: less emphasis on making the weights small



Level 1

- A **Taylor approximation** of the log posterior is considered at the maximum posterior w_{MP} solution:

$$P(w|D,\mathcal{H}) \simeq P(w_{MP}|D,\mathcal{H}) \exp(-\frac{1}{2}\Delta w^T A \Delta w)$$

with Hessian matrix at w_{MP}:

$$A = -\nabla^2 \log P(w|D,\mathcal{H})|_{w_{MP}}$$

note: the Hessian matrix contains second order derivatives

- Posterior at Level 1:

$$P(w|D,\alpha,\beta,\mathcal{H}) \propto \exp[-M(w_{MP}) - \frac{1}{2}(w - w_{MP})^T A (w - w_{MP})]$$

Level 2

- Use the fact that the evidence of level 1 equals the likelihood of level 2 to propagate the results from level 1 towards level 2

- One obtains formulas for an optimal choice of α_{MP}, β_{MP} by solving the following set of nonlinear equations (implicit in α_{MP}, β_{MP}):

$$\left\{ \begin{array}{lcl} \alpha_{MP} & = & \frac{\gamma}{2E_W(w_{MP})} \\ \beta_{MP} & = & \frac{N-\gamma}{2E_D(w_{MP})} \end{array} \right.$$

-

Level 2

- Use the fact that the evidence of level 1 equals the likelihood of level 2 to propagate the results from level 1 towards level 2

- One obtains formulas for an optimal choice of α_{MP}, β_{MP} by solving the following set of nonlinear equations (implicit in α_{MP}, β_{MP}):

$$\left\{ \begin{array}{lcl} \alpha_{MP} & = & \frac{\gamma}{2E_W(w_{MP})} \\ \beta_{MP} & = & \frac{N-\gamma}{2E_D(w_{MP})} \end{array} \right.$$

- An important role is played by the **effective number of parameters**:

$$\gamma \quad = \quad k - \alpha_{MP}\mathrm{trace}(A^{-1}) = \sum_{i=1}^{k} \frac{\lambda_i}{\lambda_i + \alpha} \leq k$$

with k the total number of parameters and λ_i the eigenvalues of $\beta\nabla^2 E_D$.

Level 3

- If the posterior is well approximated by a Gaussian:

$$P(D|\mathcal{H}_i) \simeq P(D|w_{MP}, \mathcal{H}_i) \times P(w_{MP}|\mathcal{H}_i) \det{}^{-1/2}(A/2\pi)$$

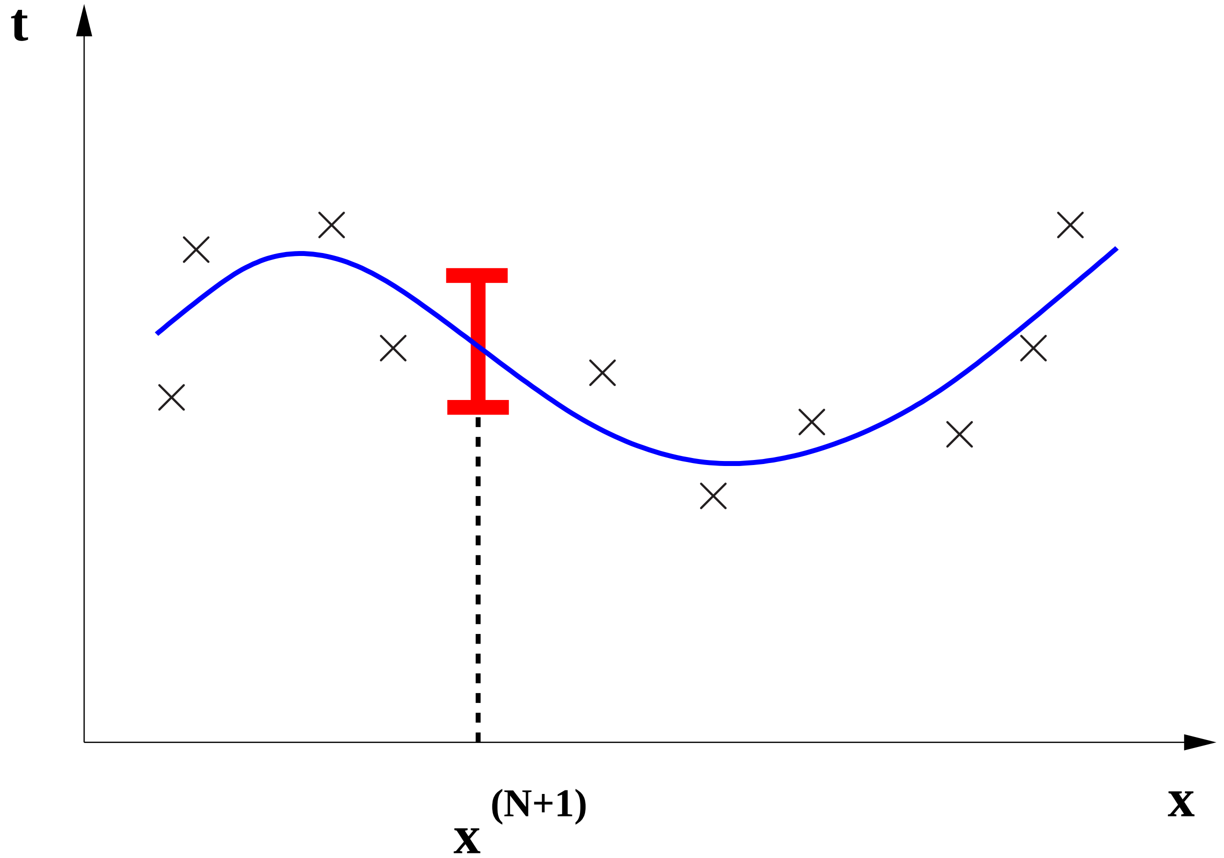
with Hessian $A = -\nabla^2 \log P(w|D, \mathcal{H}_i)$.

- Meaning:

Evidence $\simeq$ Best fit likelihood $\times$ Occam factor

- Different models $\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, ...$ are compared.
The model $\mathcal{H}_i$ that maximizes the evidence is selected.

Error bars and predictions



Training data: $(x^{(1)}, t^{(1)})$, $(x^{(2)}, t^{(2)})$, ... , $(x^{(N)}, t^{(N)})$
Given a new point $x^{(N+1)}$, what is the **estimated output value?**
What is the **uncertainty** on this prediction?

Error bars and predictions

- Prediction of a new target value $t^{(N+1)}$ for a new given input $x^{(N+1)}$
- Bayesian prediction involves **marginalization** over the uncertainty at all levels

$$P(t^{(N+1)}|D) = \sum_{\mathcal{H}_i} \int d\alpha d\beta \int d^k w P(t^{(N+1)}|w, \alpha, \beta, \mathcal{H}) P(w, \alpha, \beta, \mathcal{H}|D)$$

Try to find an approximation to this ...

Error bars and predictions

- **Assuming** fixed values α, β
  Assuming a local linearization of the output:

$$y(x^{(N+1)}; w) \simeq y(x^{(N+1)}; w_{MP}) + g(w - w_{MP})$$

  with sensitivity $g = \frac{\partial y}{\partial w}|_{x^{(N+1)}, w_{MP}}$.

-

Error bars and predictions

- **Assuming** fixed values α, β
  Assuming a local linearization of the output:

$$y(x^{(N+1)}; w) \simeq y(x^{(N+1)}; w_{MP}) + g(w - w_{MP})$$

  with sensitivity $g = \frac{\partial y}{\partial w}|_{x^{(N+1)}, w_{MP}}$.

- Then one has a predictive distribution with **mean**

$$y(x^{(N+1)}; w_{MP})$$

  and **variance**

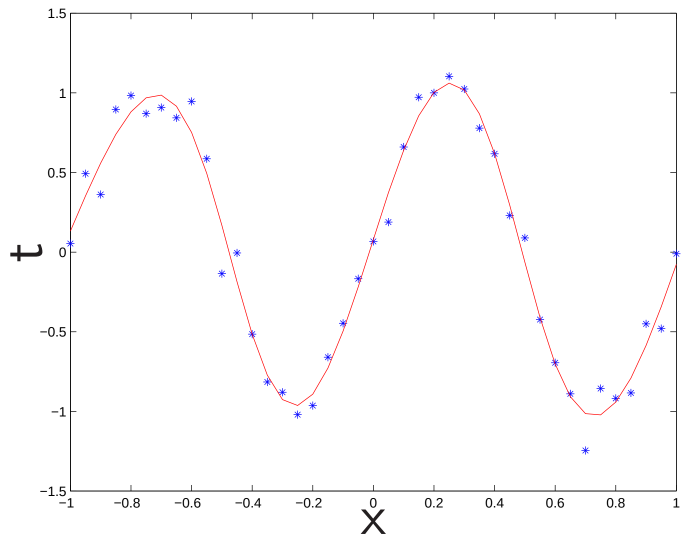
$$\sigma^2_{t|\alpha,\beta} = g^T A^{-1} g + \frac{1}{\beta}$$

  with $A = -\nabla^2 \log P(w|D, \alpha, \beta, \mathcal{H})$.

Matlab software demonstration

- demonstration of trainbr in Matlab
  http://www.mathworks.com/help/toolbox/nnet/trainbr.html

- x = [-1:0.05:1];
  t = sin(2*pi*x)+0.1*randn(size(x));
  net = newff(x,t,7,{},'trainbr'); % 7 hidden units
  net = train(net,x,t);
  y = sim(net,x);
  plot(x,t,'*'),hold,plot(x,y,'r-')

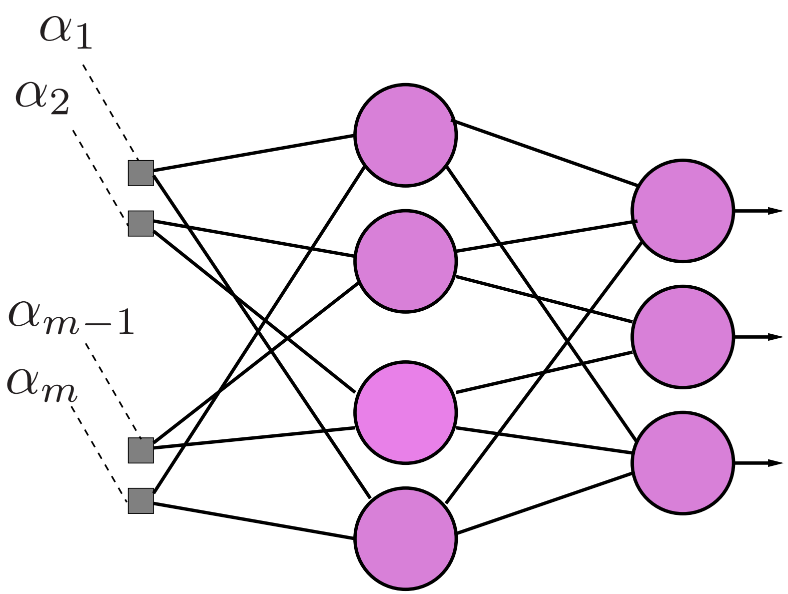


- The method **automatically** tunes the hyperparameters α, β!

Automatic relevance determination

- Assign different regularization constants $\alpha_1, \alpha_2, ..., \alpha_m$ to the weights that correspond to a particular input (m inputs in total).

- Do Bayesian inference at level 2 in $\alpha_1, \alpha_2, ..., \alpha_m, \beta$



Additional course material

This lecture is based on

- David MacKay, "Probable Networks and Plausible Predictions - A Review of Practical Bayesian Methods for Supervised Neural Networks"

see http://www.inference.phy.cam.ac.uk/mackay/
for related papers and book.