Generalization

Johan Suykens

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

Lecture 4

Overview

- interpretation of network outputs
- bias and variance
- bias-variance trade-off
- selection of regularization constant
- cross-validation
- complexity criteria
- pruning
- committee networks

Interpretation of network outputs (1)

- Training data: $\{x_n, t_n\}_{n=1}^{N}$, $x_n \in \mathbb{R}^m$ input data, $t_n \in \mathbb{R}$ target data
- $y(x_n; w)$ static model with output $y \in \mathbb{R}$ and weights w
- The goal is not to memorize data but rather to model the underlying generator of the data, characterized by $p(x, t)$.

Interpretation of network outputs (1)

- Training data: $\{x_n, t_n\}_{n=1}^{N}$, $x_n \in \mathbb{R}^m$ input data, $t_n \in \mathbb{R}$ target data
- $y(x_n; w)$ static model with output $y \in \mathbb{R}$ and weights w
- The goal is not to memorize data but rather to model the underlying generator of the data, characterized by $p(x, t)$.
- $p(x, t)$ joint probability density of inputs x and targets t. One has

$$p(x, t) = p(t|x)p(x)$$

with $p(t|x)$ probability density of t given a particular value of x and $p(x)$ unconditional density of x.

Interpretation of network outputs (2)

- **Generalization error:**
  Consider cost E in the limit $N \to \infty$ (infinite data set size)

$$
\begin{aligned}
E &= \lim_{N\to\infty} \frac{1}{2N} \sum_{n=1}^{N} \{y(x_n; w) - t_n\}^2 \\
&= \tfrac{1}{2} \int \int \{y(x; w) - t\}^2 p(t, x)\, dt dx \\
&= \tfrac{1}{2} \int \int \{y(x; w) - t\}^2 p(t|x) p(x)\, dt dx
\end{aligned}
$$

Interpretation of network outputs (3)

- Define conditional averages $\langle t|x\rangle = \int tp(t|x)dt, \langle t^2|x\rangle = \int t^2 p(t|x)dt.$

  One has

$$\begin{aligned}\{y-t\}^2 &= \{y-\langle t|x\rangle + \langle t|x\rangle - t\}^2 \\ &= \{y-\langle t|x\rangle\}^2 + 2\{y-\langle t|x\rangle\}\{\langle t|x\rangle - t\} + \{\langle t|x\rangle - t\}^2\end{aligned}$$

Interpretation of network outputs (3)

- Define conditional averages $\langle t|x\rangle = \int tp(t|x)dt, \langle t^2|x\rangle = \int t^2 p(t|x)dt.$

  One has

$$\begin{aligned} \{y-t\}^2 &= \{y-\langle t|x\rangle + \langle t|x\rangle - t\}^2 \\ &= \{y-\langle t|x\rangle\}^2 + 2\{y-\langle t|x\rangle\}\{\langle t|x\rangle - t\} + \{\langle t|x\rangle - t\}^2 \end{aligned}$$

  and obtains

$$\begin{aligned} E &= \tfrac{1}{2}\int \{y(x;w)-\langle t|x\rangle\}^2 p(x)dx + \tfrac{1}{2}\int \{\langle t^2|x\rangle - \langle t|x\rangle^2\}p(x)dx \\ &= \boxed{1} + \boxed{2} \end{aligned}$$

  $\boxed{1}$: At the minimum w^* of error function: $y(x;w^*) = \langle t|x\rangle$, i.e. the output approximates the conditional average of the target data.

  $\boxed{2}$: intrinsic noise on the data and sets lower limit on achievable error

Bias and variance (1)

- Practice: we have only one specific and finite data set D.
  Eliminate the dependency on a specific data set D by

$$\mathcal{E}_D[\{y(x) - \langle t|x\rangle\}^2]$$

  where $\mathcal{E}_D$ denotes the ensemble average.
  Question: how close is the mapping to the desired one ?

Bias and variance (1)

- Practice: we have only one specific and finite data set D.
  Eliminate the dependency on a specific data set D by

$$\mathcal{E}_D[\{y(x) - \langle t|x\rangle\}^2]$$

  where $\mathcal{E}_D$ denotes the ensemble average.
  Question: how close is the mapping to the desired one ?

- In a similar fashion as before we write

$$\begin{aligned}
\{y(x) - \langle t|x\rangle\}^2 &= \{y(x) - \mathcal{E}_D[y(x)] + \mathcal{E}_D[y(x)] - \langle t|x\rangle\}^2 \\
&= \{y(x) - \mathcal{E}_D[y(x)]\}^2 + \{\mathcal{E}_D[y(x)] - \langle t|x\rangle\}^2 + \\
&\quad 2\{y(x) - \mathcal{E}_D[y(x)]\}\{\mathcal{E}_D[y(x)] - \langle t|x\rangle\}
\end{aligned}$$

Bias and variance (2)

- Expectation over ensemble of data sets:

$$\begin{aligned}\mathcal{E}_D[\{y(x) - \langle t|x\rangle\}^2] &= \{\mathcal{E}_D[y(x)] - \langle t|x\rangle\}^2 + \\ &\quad \mathcal{E}_D[\{y(x) - \mathcal{E}_D[y(x)]\}^2] \\ &= \boxed{1} + \boxed{2}\end{aligned}$$

$\boxed{1}$

$$(\text{bias})^2 = \frac{1}{2}\int \{\mathcal{E}_D[y(x)] - \langle t|x\rangle\}^2 p(x)dx$$

$\boxed{2}$

$$\text{variance} = \frac{1}{2}\int \mathcal{E}_D[\{y(x) - \mathcal{E}_D[y(x)]\}^2] p(x)dx$$

Example 1

In order to fix the ideas...

Generate 100 data sets by sampling $h(x)$ and adding noise.
$h(x)$ is the true underlying function to be estimated (which is known in this experiment, but in a practical situation would be unknown).

Example 1

In order to fix the ideas...

Generate 100 data sets by sampling $h(x)$ and adding noise.
$h(x)$ is the true underlying function to be estimated (which is known in this experiment, but in a practical situation would be unknown).

Estimate the mappings $y_i(x)$ for $i = 1, 2, ..., 100$
(e.g. 100 MLP's, one MLP for each generated data set).

Average response : $\overline{y}(x) = \frac{1}{100} \sum_{i=1}^{100} y_i(x)$

$$(\text{Bias})^2 = \sum_n \{\overline{y}(x_n) - h(x_n)\}^2$$

$$\text{Variance} = \sum_n \frac{1}{100} \sum_{i=1}^{100} \{y_i(x_n) - \overline{y}(x_n)\}^2$$

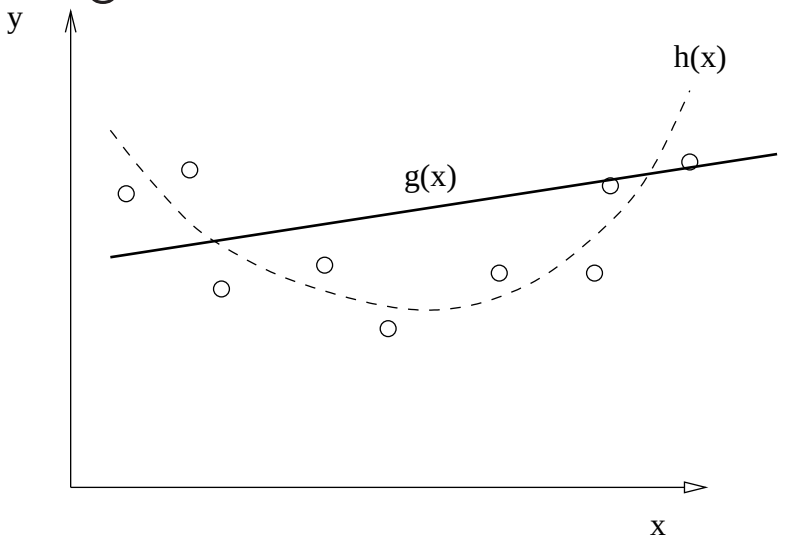
Example 2 (1)

Suppose $t_n = h(x_n) + \epsilon_n$ with true function $h(x)$ estimated by $y(x)$.
Consider two extreme cases:

- **Extreme case 1:** Fix $y(x) = g(x)$ independent of any data set. Then:

$$\mathcal{E}_D[y(x)] = g(x) = y(x)$$

$\Rightarrow$ zero variance, but large bias



Example 2 (2)

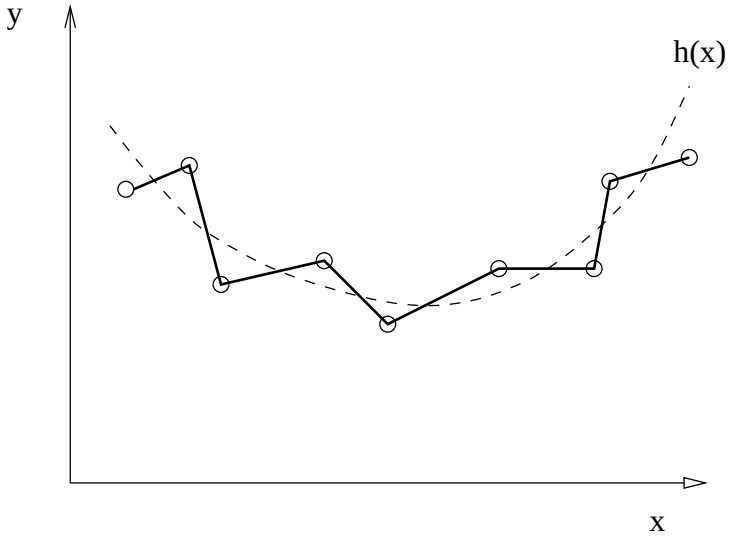
- **Extreme case 2:** Consider exact interpolant of data. Then zero bias:

$$\mathcal{E}_D[y(x)] = \mathcal{E}_D[h(x) + \epsilon] = h(x) = \langle t|x \rangle$$

but large variance:

$$\mathcal{E}_D[\{y(x) - \mathcal{E}_D[y(x)]\}^2] = \mathcal{E}_D[\{y(x) - h(x)\}^2] = \mathcal{E}_D[\epsilon^2]$$

$\Rightarrow$ zero bias, but large variance.

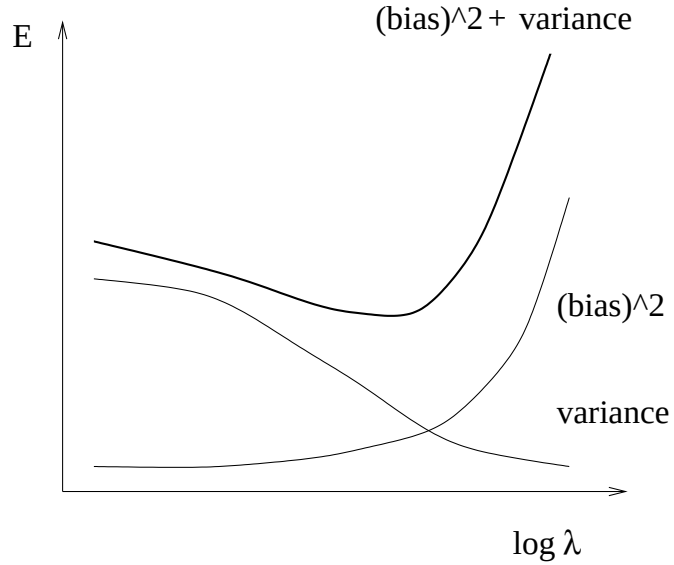


Ridge regression: bias-variance trade-off (1)

Ridge regression

$$\min_{\theta} J_{ridge}(\theta) = \frac{1}{2} e^T e + \frac{1}{2} \lambda \, \theta^T \theta \,, \quad \lambda > 0$$

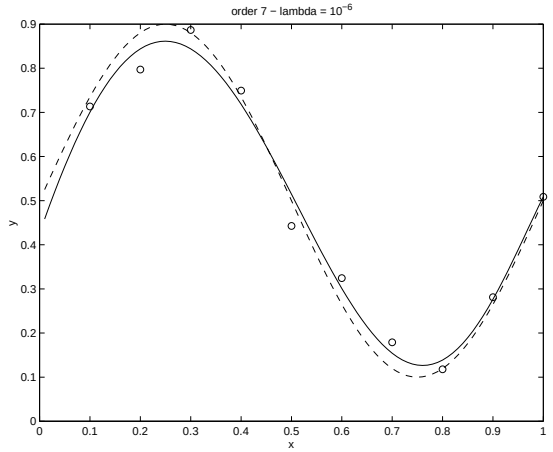
Bias-variance trade-off



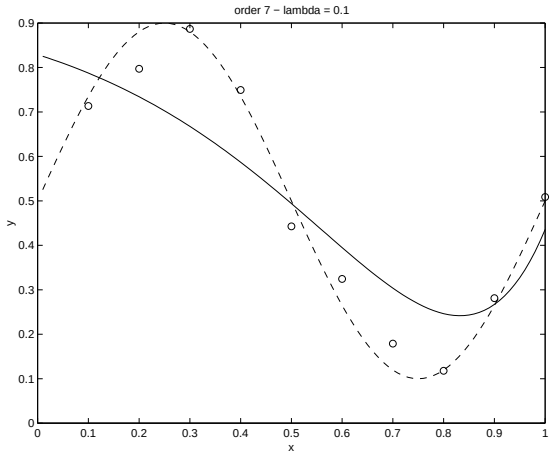
Try to minimize both bias and variance

Ridge regression: bias-variance trade-off (2)

Regularization for polynomial of order 7: $\lambda = 10^{-6}$ (oscillation is avoided)



$\lambda = 0.1$ (too much regularization)



Effective number of parameters: bias-variance trade-off

Effective number of parameters:
The number

$$(\#\lambda_j) > \nu$$

is related to the *effective* number of parameters

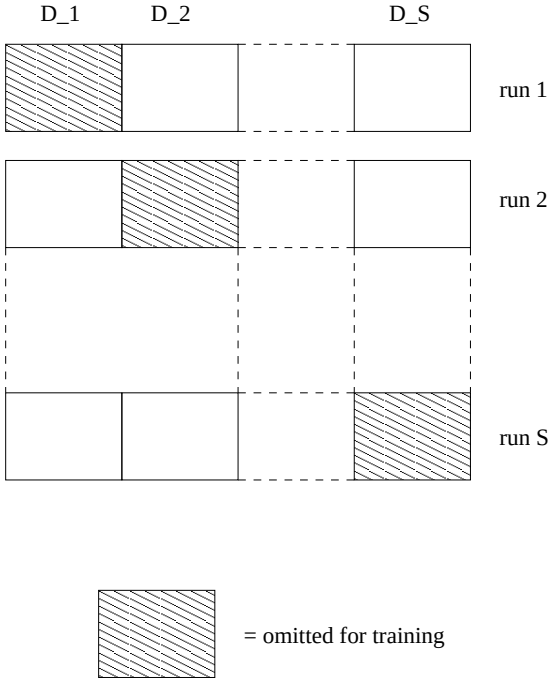
ν large - small model structure
small variance
large bias

ν small - large model structure
large variance
small bias

Cross-validation

Divide training set into S segments and train in each run on $S-1$ segments.
Test on the sum of segments that were left out in the S runs.

Typical choice: $S = 10$ (called 10-fold cross validation)
Extreme limit: $S = N$, i.e. N runs with $N - 1$ data points
(called leave-one-out cross-validation)



Complexity criteria (1)

- *In conventional statistics:*

  "prediction error (PE) = training error + complexity"

- *For linear models* (e.g. Akaike information criterion)

  $$\text{PE} = \text{MSE} + \frac{W}{N}\sigma^2$$

  where

N	number of training data
W	number of adjustable parameters
σ^2	variance of noise on data

Complexity criteria (2)

- *For nonlinear models* (Moody, 1992)
  Generalized prediction error (GPE):

  $$\text{GPE} = \text{MSE} + \frac{\gamma}{N}\sigma^2$$

  where

  $$\gamma = \sum_i \frac{\lambda_i}{\lambda_i + \nu}$$

  is the effective number of parameters.

  λ_i eigenvalues of Hessian of unregularized error
  ν regularization coefficient

  Eigenvalues $\lambda_i \ll \nu$ do not contribute to the sum.

Bayesian Learning

this is the topic of the next lecture

Pruning (1)

- **Optimal brain damage** (Le Cun, 1990)
  Consider error change due to small changes in weights:

$$\delta E = \sum_i \frac{\partial E}{\partial w_i} \delta w_i + \frac{1}{2} \sum_i \sum_j H_{ij} \delta w_i \delta w_j + \mathcal{O}(\delta w^3)$$

  where $H_{ij} = \frac{\partial^2 E}{\partial w_i \partial w_j}$.
  Assumption (after convergence): $\delta E \simeq \frac{1}{2} \sum_i H_{ii} \, \delta w_i^2$

- Measure relative importance of weights (saliency values):

$$H_{ii} w_i^2 / 2$$

Pruning (2)

- *Pruning algorithm:*

  1. Choose a relatively large initial network architecture.
  2. Train the network in the usual way until some stopping criterion is satisfied.
  3. Compute the saliencies $H_{ii}w_{ii}^2/2$.
  4. Sort weights by saliency and delete low-saliency weights
  5. Go to 2 and repeat until some overall stopping criterion is reached

- Optimal brain damage has been applied to recognition of handwritten zip codes (Le Cun), where networks with 10000 interconnection weights have been pruned by a factor 4.

Pruning (3)

- **Optimal brain surgeon** (Hassibi and Stork, 1993)

  Neglecting higher order terms one has

  $$\delta E = \frac{1}{2}\delta w^T H \delta w$$

  Setting weight $w_i = 0$ corresponds to $\delta w_i = -w_i$ or $e_i^T \delta w + w_i = 0$ where e_i is a unit vector.

- Consider the optimization problem

  $$\min_{\delta w} \delta E = \frac{1}{2}\delta w^T H \delta w \quad \text{subject to} \quad e_i^T \delta w + w_i = 0$$

  Lagrangian: $\mathcal{L}(\delta w, \lambda) = \frac{1}{2}\delta w^T H \delta w - \lambda(e_i^T \delta w + w_i)$

Pruning (4)

- Conditions for optimality:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial(\delta w)} = H\delta w - \lambda e_i = 0 \quad \rightarrow \quad \delta w = \lambda H^{-1} e_i \\ \frac{\partial \mathcal{L}}{\partial \lambda} = e_i^T \delta w + w_i = 0 \quad \rightarrow \quad \lambda e_i^T H^{-1} e_i = \lambda [H^{-1}]_{ii} = -w_i \end{cases}$$

  Hence $\delta w = -\frac{w_i}{[H^{-1}]_{ii}} H^{-1} e_i$ and $\delta E_i = \frac{1}{2} \frac{w_i^2}{[H^{-1}]_{ii}}$

- *Pruning algorithm:*

  1. Train a relatively large network to a minimum of the error function.
  2. Evaluate inverse Hessian H^{-1}.
  3. Evaluate δE_i for each value of i and select the value of i which gives the smallest increase in error.
  4. Update all the weights according to $\delta w = -\frac{w_i}{[H^{-1}]_{ii}} H^{-1} e_i$.
  5. Go to 3 and repeat until some stopping criterion is reached.

Alternative regularization terms and sparsity

- **Weight elimination** (Weigend, 1990)

$$\tilde{E} = E + \nu \sum_i \frac{(w_i/c)^2}{1 + (w_i/c)^2}$$

The algorithm is more likely to eliminate weights (i.e. putting weights to zero) than weight decay. A drawback is the choice of the additional tuning parameter c.

- **L1-norm** on w instead of L2-norm:

$$\tilde{E} = E + \nu \sum_i |w_i|$$

gives sparsity, but non-differentiability of $|w_i|$ (this principle is used e.g. in lasso and compressed sensing methods, not in backpropagation).

Committee networks (1)

- Common approach: training many different networks and selecting the best one based on a validation set.
  Disadvantages:

  1. many training efforts are waisted
  2. generalization on the validation set has a random component due to noise on the data

- **Committee networks** (Perrone, 1993)
  The performance of the committee network can be better than the performance of the best single network.

-

Committee networks (1)

- Common approach: training many different networks and selecting the best one based on a validation set.
  Disadvantages:

  1. many training efforts are waisted
  2. generalization on the validation set has a random component due to noise on the data

- **Committee networks** (Perrone, 1993)
  The performance of the committee network can be better than the performance of the best single network.

- L trained networks $y_i(x)$, $i = 1, ..., L$ (e.g. L trained MLPs)
  True regression function $h(x)$ with

$$y_i(x) = h(x) + \epsilon_i(x) \quad i = 1, ..., L$$

  with average sum-of-squares error $E_i = \mathcal{E}[\{y_i(x) - h(x)\}^2] = \mathcal{E}[\epsilon_i^2]$.

Committee networks (2)

- **A simple committee network** (by taking average):

$$y_{COM}(x) = \frac{1}{L}\sum_{i=1}^{L} y_i(x)$$

- Error of committee network:

$$E_{COM} = \mathcal{E}[\,(\frac{1}{L}\sum_{i=1}^{L} y_i(x) - h(x))^2\,] = \mathcal{E}[\,(\frac{1}{L}\sum_{i=1}^{L} \epsilon_i)^2\,]$$

From Cauchy's inequality:

$$(\sum_{i=1}^{L} \epsilon_i)^2 \leq L\sum_{i=1}^{L} \epsilon_i^2 \;\Rightarrow\; E_{COM} \leq E_{AV}$$

where $E_{AV} = \frac{1}{L}\sum_{i=1}^{L} E_i = \frac{1}{L}\sum_{i=1}^{L} \mathcal{E}[\epsilon_i^2]$. Hence the variance is reduced by averaging over many networks.

Committee networks (2)

- **A weighted average committee network:**

$$\begin{aligned} y_{COM}(x) &= \sum_{i=1}^{L} \alpha_i y_i(x) \\ &= h(x) + \sum_{i=1}^{L} \alpha_i \epsilon_i(x) \end{aligned}$$

where $\sum_{i=1}^{L} \alpha_i = 1$. Consider the correlation matrix

$$C_{ij} = \mathcal{E}[\epsilon_i(x)\epsilon_j(x)]$$

In practice one uses a finite-sample approximation:
$C_{ij} = \frac{1}{N} \sum_{n=1}^{N} [y_i(x_n) - t_n][y_j(x_n) - t_n]$

Committee networks (3)

- Committee error:

$$\begin{aligned} E_{COM} &= \mathcal{E}[\{y_{COM}(x) - h(x)\}^2] \\ &= \mathcal{E}[\,(\sum_{i=1}^{L} \alpha_i \epsilon_i)\,(\sum_{j=1}^{L} \alpha_j \epsilon_j)\,] \\ &= \sum_{i=1}^{L}\sum_{j=1}^{L} \alpha_i \alpha_j C_{ij} = \alpha^T C \alpha \end{aligned}$$

- Optimal choice of α:

$$\min_{\alpha} \frac{1}{2}\alpha^T C \alpha \;\; \text{s.t.} \;\; \sum_{i=1}^{L} \alpha_i = 1$$

Optimal $\alpha = \frac{C^{-1}\vec{1}}{\vec{1}^T C^{-1}\vec{1}}$ with committee error $E_{COM} = 1/(\vec{1}^T C^{-1}\vec{1})$.