Theoretical Concepts of Machine Learning
Part 2

Sepp Hochreiter

Institute of Bioinformatics
Johannes Kepler University, Linz, Austria

Outline

7 Optimization Techniques

Outline

8 Bayes Techniques

8.1 Likelihood, Prior, Posterior, Evidence

8.2 Maximum A Posteriori Approach

8.3 Posterior Approximation

8.4 Error Bars and Confidence Intervals

8.5 Hyper-parameter Selection: Evidence Framework

8.6 Hyper-parameter Selection: Integrate Out

8.7 Model Comparison

8.8 Posterior Sampling

9 Linear Models

9.1 Linear Regression

9.2 Analysis of Variance

9.3 Analysis of Covariance

9.4 Mixed Effects Models

9.5 Generalized Linear Models

9.6 Regularization

Chapter 7

Optimization Techniques

Error Minimization and Model Selection

Model class: parameterized models with parameter vector $\boldsymbol{w}$

Goal: to find the optimal model in parameter space

optimal model: optimizes the objective function

objective function: is defined by the problem to solve
- empirical error
- complexity

to find the optimal model: search in the parameter space

Search Methods and Evolutionary Approaches

In principle any search algorithm can be used

random search: randomly a parameter vector is selected and then evaluated -- best solution kept

exhaustive search: parameter space is searched systematically

will find the optimal solution for a finite set of possible parameters
do not use any dependency between objective function and parameter
→ singular solution

In general there are dependencies between objective function and parameters which should be utilized

Search Methods and Evolutionary Approaches

first dependency: good solutions are near good solutions
(objective is not continuous)

stochastic gradient: locally optimize the objective function

genetic algorithms: a stochastic gradient corresponds to mutations which are small

another dependency: good solutions share properties (independent!)

genetic algorithms: - "crossover mutations" combines parts of different parameter vectors
- solutions have independent *building blocks*

Search Methods and Evolutionary Approaches

other evolutionary strategies to optimize models:

- genetic programming
- swarm algorithms
- ant algorithms
- self-modifying policies
- Optimal Ordered Problem Solver
  (class is predefined by the search algorithm,
  model class is not parameterized)
  the latter modify their own search strategy

parallel search: different locations in the parameter space

Search Methods and Evolutionary Approaches

simulated annealing (SA): overcome the problem of local optima

- find the global solution with probability one if the annealing process is slow enough

- jumps from the current state into another state where the probability is given by the objective function

- objective function = energy function

- energy function follows the Maxwell-Boltzmann distribution

- sampling is similar to the Metropolis-Hastings algorithm

- temperature: which jumps are possible
  beginning: high temperature → large jumps into energetically worse regions
  end: low temperature → small jumps into energetically favorable regions

Search Methods and Evolutionary Approaches

Advantages:
- discrete problems and non-differentiable objective functions
- very easy to implement

Disadvantages:
- computationally expensive for large parameter spaces
- depend on the representation of the model

parameter vector of length W: 2^W decisions to make to decide whether to in- or decrease components

Search Methods and Evolutionary Approaches

in the following: the objective function is differentiable with respect to the parameters

→use of gradient information

Already treated: convex optimization (only one minimum)

Gradient Descent

objective function: $R(g(.;\boldsymbol{w}))$ is a differentiable function

parameter vector $\boldsymbol{w}$ is W-dimensional

model: g

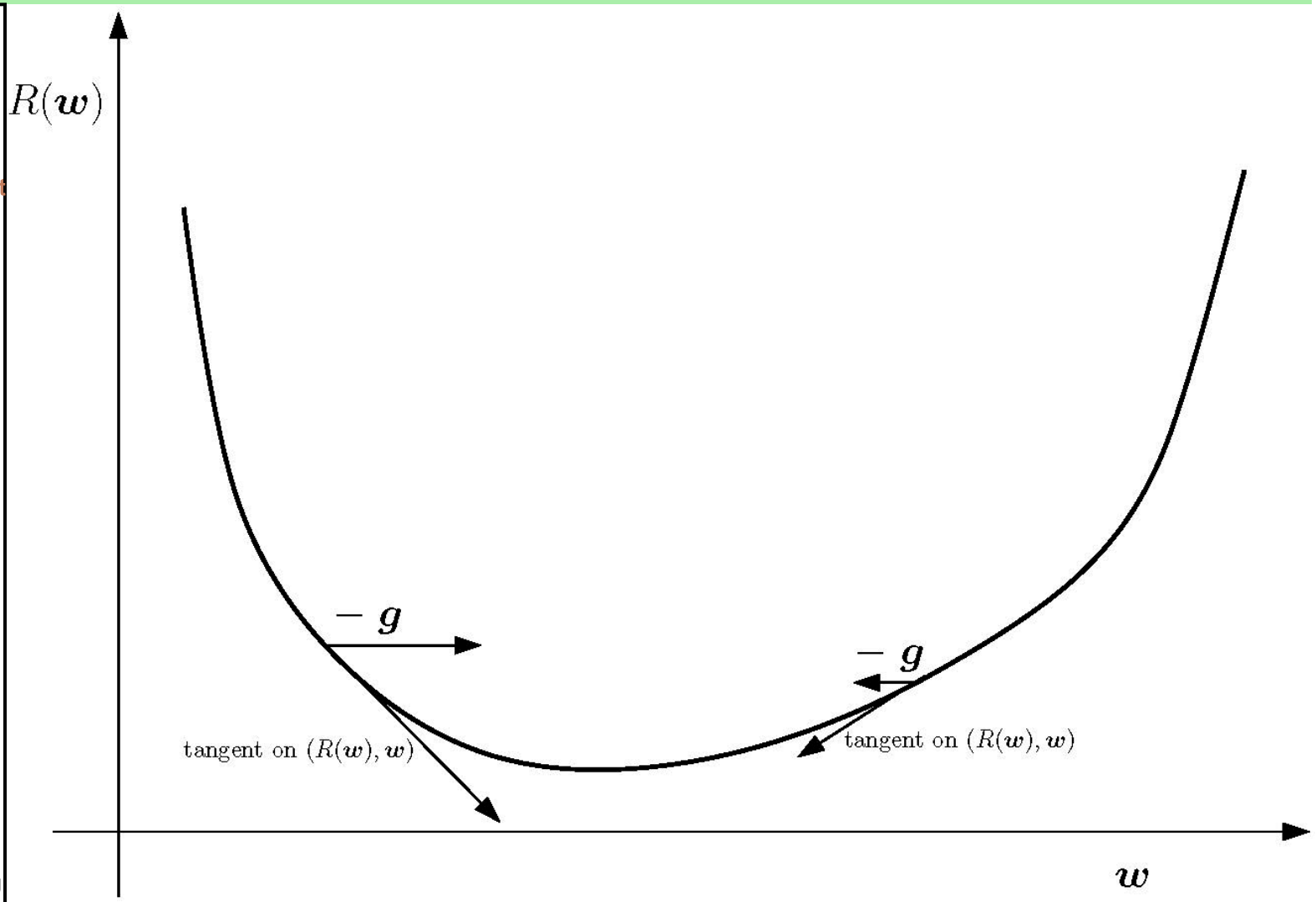
for simplicity: $R(\boldsymbol{w}) = R(g(.;\boldsymbol{w}))$

gradient:
$$\frac{\partial R(\boldsymbol{w})}{\partial \boldsymbol{w}} = \nabla_{\boldsymbol{w}} R(\boldsymbol{w}) = \left(\frac{\partial R(\boldsymbol{w})}{\partial w_1}, \ldots, \frac{\partial R(\boldsymbol{w})}{\partial w_W} \right)^T$$

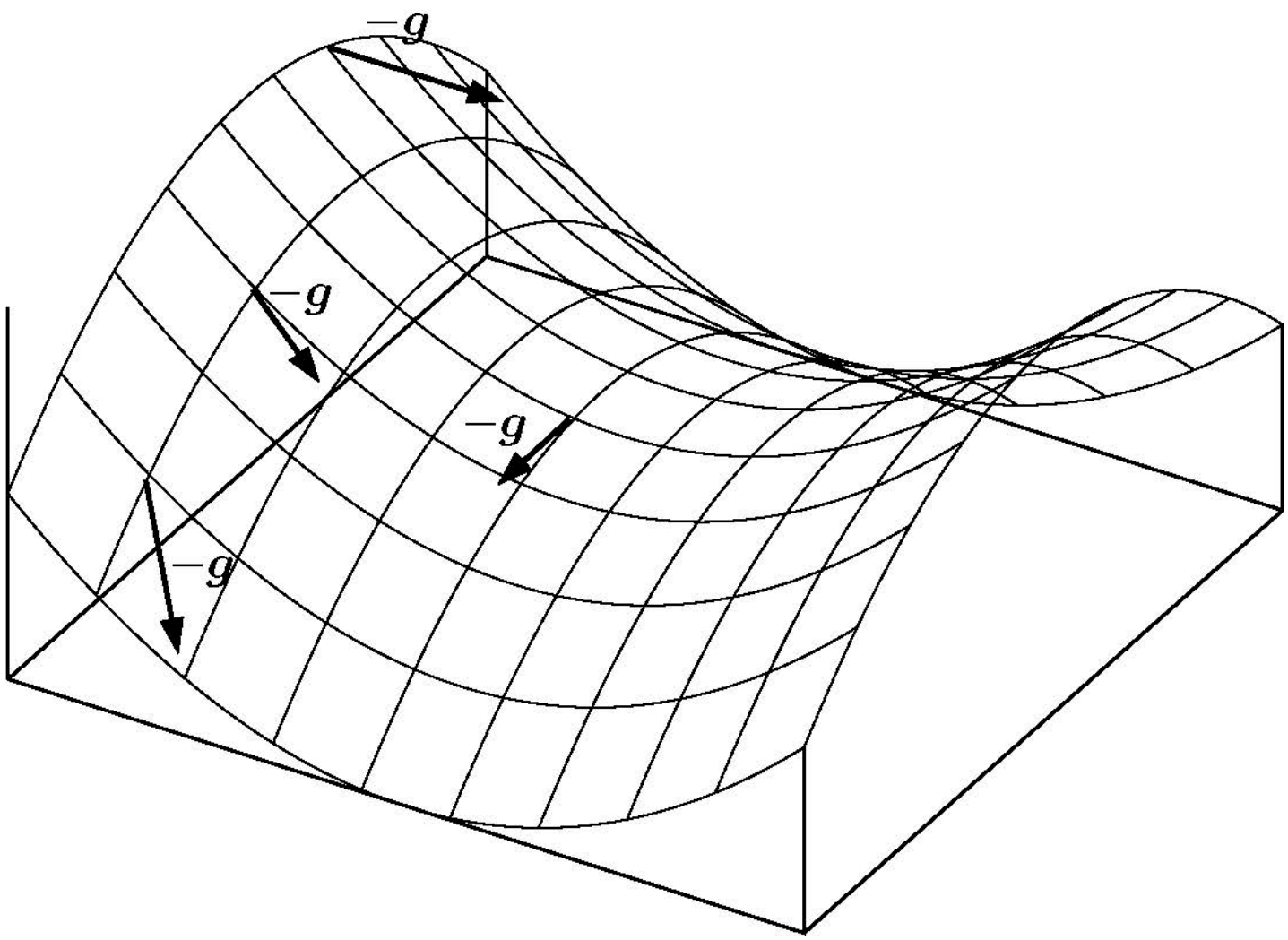
short: $\boldsymbol{g} = \nabla_{\boldsymbol{w}} R(\boldsymbol{w})$

negative gradient: direction of the steepest descent of the objective

Gradient Descent



Gradient Descent



Gradient Descent

gradient is locally → small step in the negative gradient direction

learning rate $0 < \eta$: controls step-size

gradient descent update:

$$\Delta \boldsymbol{w} = -\eta \nabla_{\boldsymbol{w}} R(\boldsymbol{w})$$

$$\boldsymbol{w}^{\text{new}} = \boldsymbol{w}^{\text{old}} + \Delta \boldsymbol{w}$$

momentum term: - gradient descent oscillates
- flat plateau
→ learning slows down

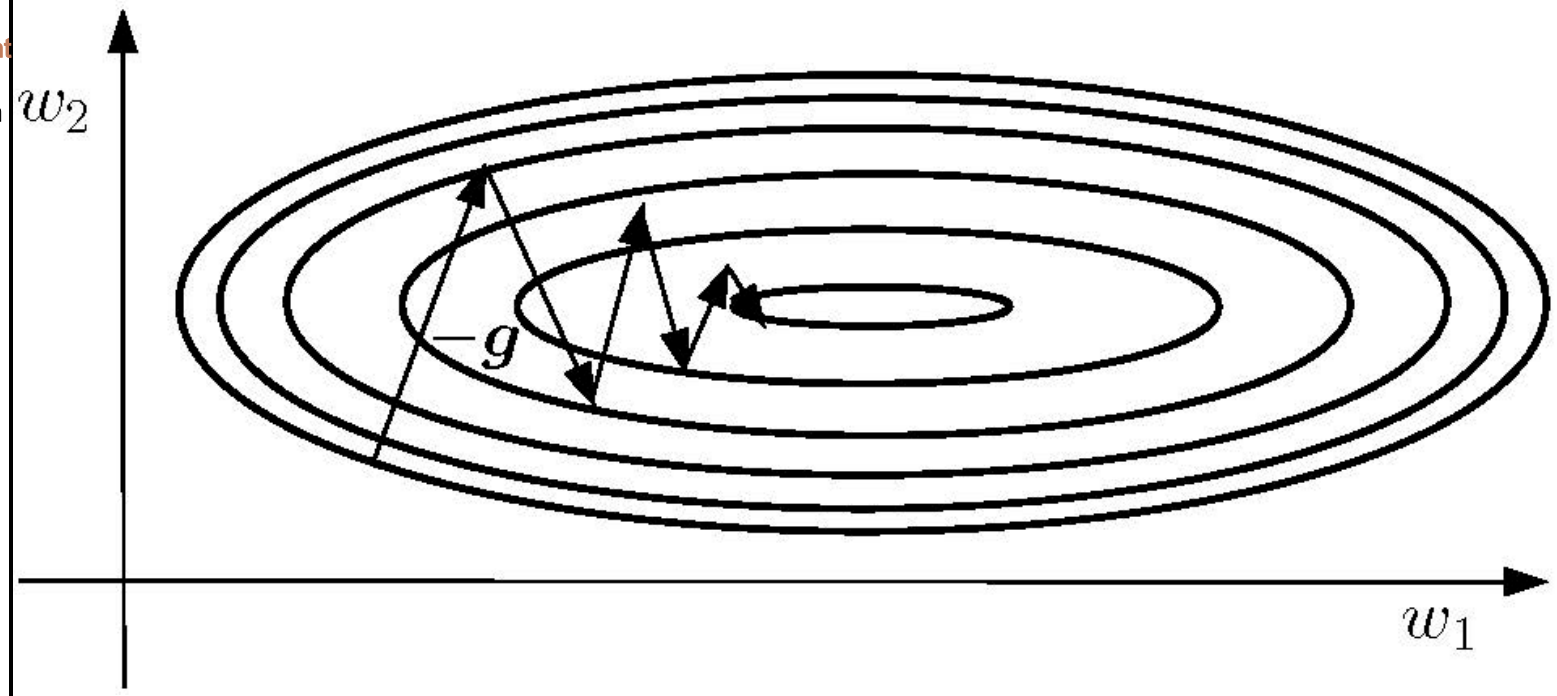
$$\Delta^{\text{new}} \boldsymbol{w} = -\eta \nabla_{\boldsymbol{w}} R(\boldsymbol{w})$$

$$\boldsymbol{w}^{\text{new}} = \boldsymbol{w}^{\text{old}} + \Delta^{\text{new}} \boldsymbol{w} + \mu \Delta^{\text{old}}$$

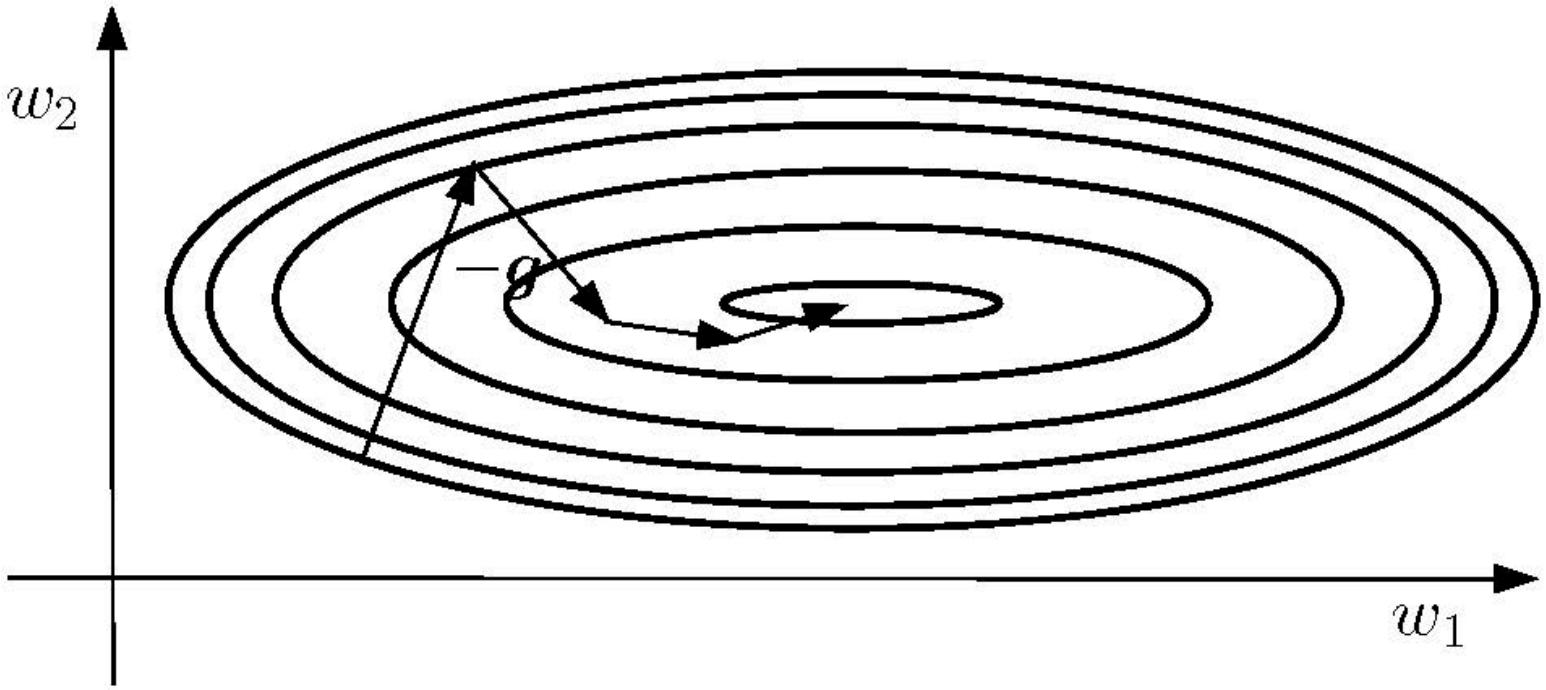
$$\Delta^{\text{old}} = \Delta^{\text{new}}$$

momentum factor or **momentum parameter**: $0 \leq \mu \leq 1$

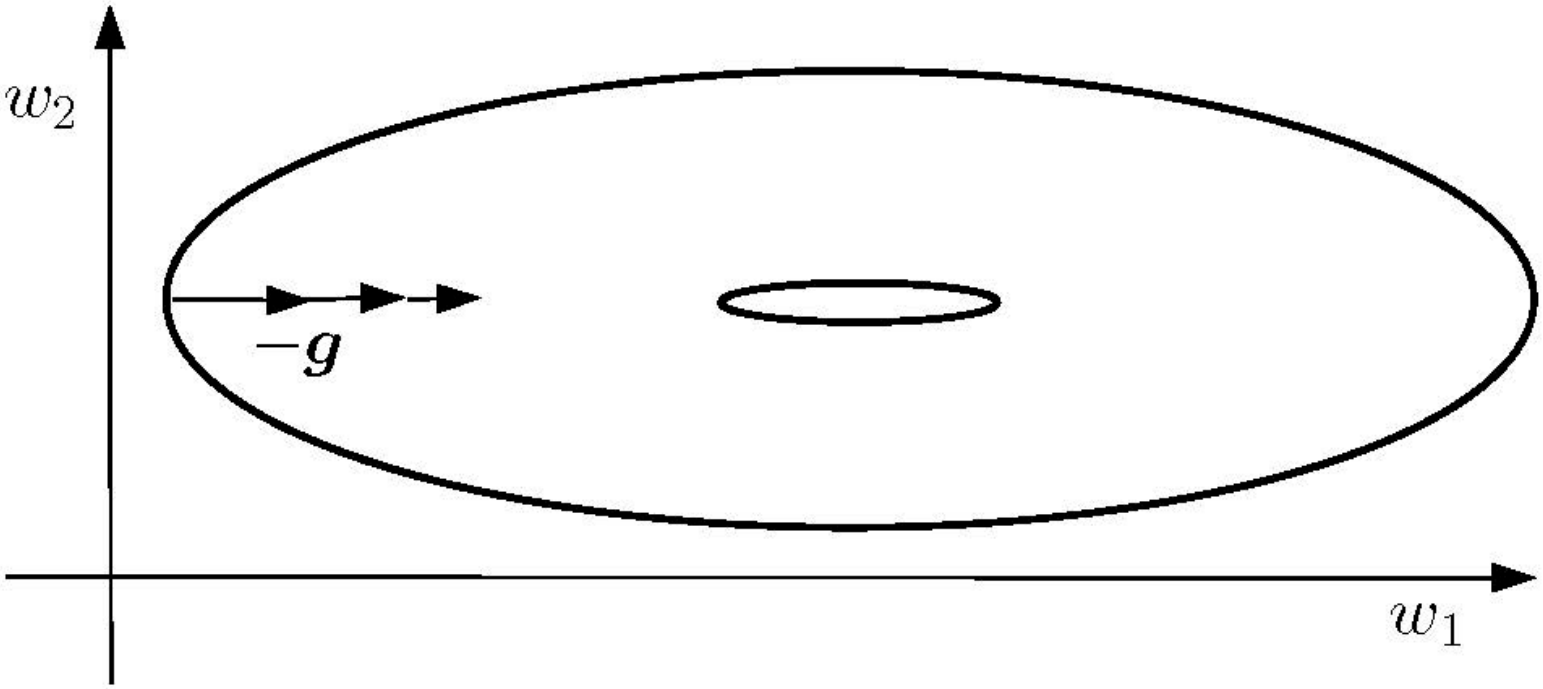
Gradient Descent



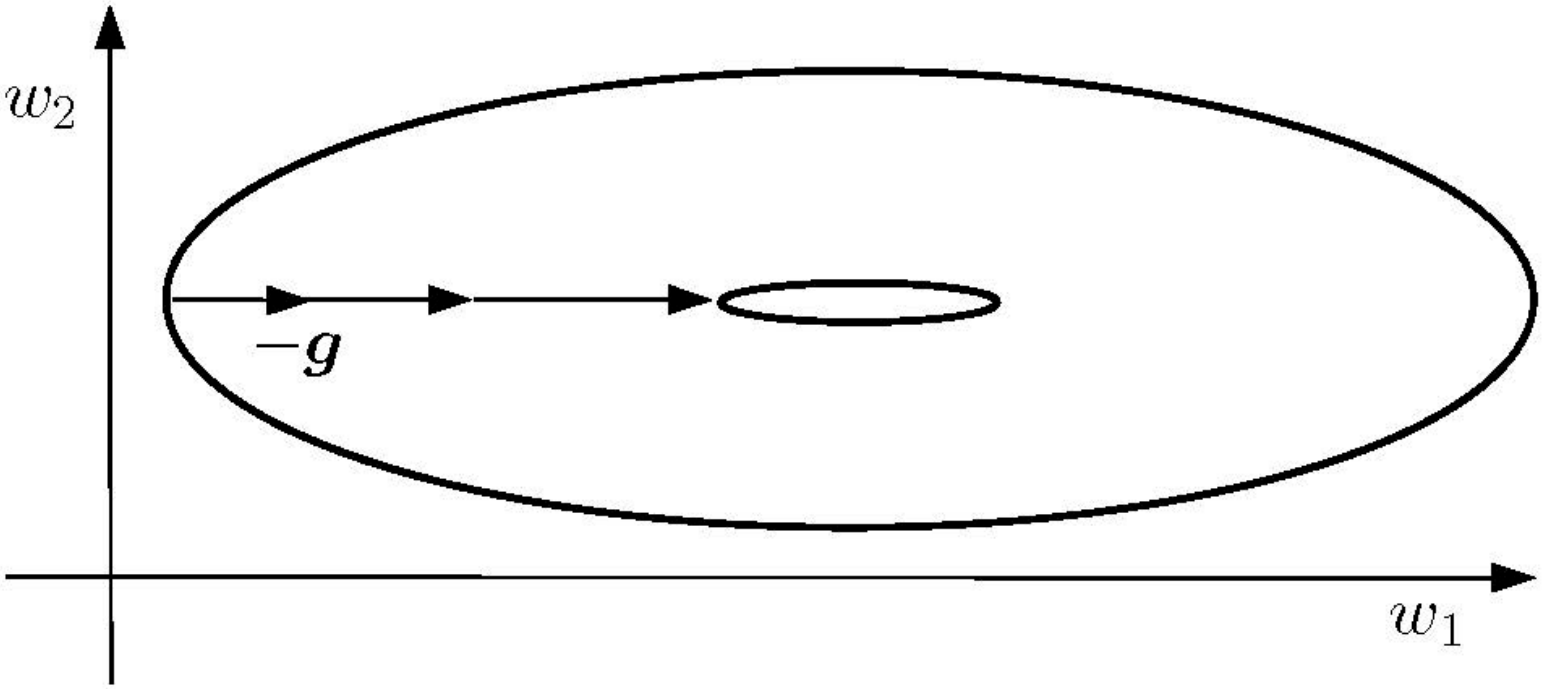
Gradient Descent



Gradient Descent



Gradient Descent

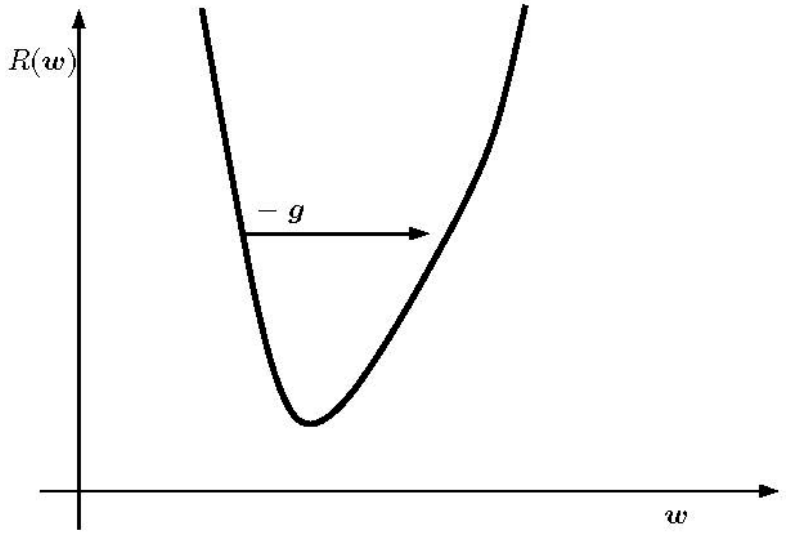


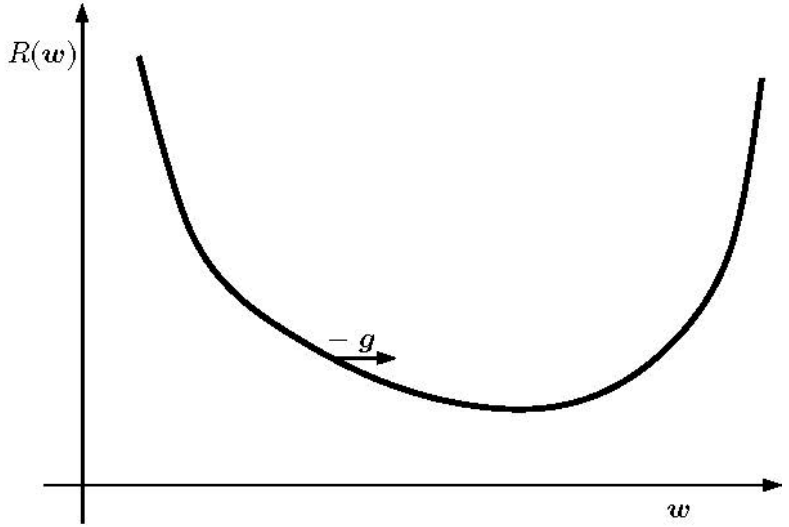
Step-size Optimization

step-size should be adjusted to the curvature of the error surface

flat curve: large step-size

steep minima: small step-size





Step-size Optimization: Heuristics

Learning rate adjustments: resilient backpropagation (Rprop)

change of the risk $\Delta R = R(\boldsymbol{w} + \Delta \boldsymbol{w}) - R(\boldsymbol{w})$

learning rate adjustment: $$\eta^{\text{new}} = \begin{cases} \rho\, \eta^{\text{old}} & \text{if} \quad \Delta R \leq 0 \\ \sigma\, \eta^{\text{old}} & \text{if} \quad \Delta R > 0 \end{cases}$$

$\rho > 1 \quad \sigma < 1$ typical: $\rho = 1.1, \quad \sigma = 0.5$

Step-size Optimization: Heuristics

Largest eigenvalue of the Hessian

Hessian $\boldsymbol{H}$: $H_{ij} = \frac{\partial^2 R(\boldsymbol{w})}{\partial w_i \, \partial w_j}$

Largest eigenvalue $e_{\max}$ bounds the learning rate:

$$\eta \leq \frac{2}{\lambda_{\max}}$$

maximal eigenvalue: matrix iteration $\boldsymbol{a} = \sum_{i=1}^{W} \alpha_i \, \boldsymbol{e}_i$

$$\boldsymbol{H}^s \, \boldsymbol{a} = \sum_{i=1}^{W} \lambda_i^s \, \alpha_i \, \boldsymbol{e}_i \approx \lambda_{\max}^s \, \alpha_{\max} \, \boldsymbol{e}_{\max}$$

Important question:
how time intensive is it to obtain the Hessian or the product of the Hessian with a vector?

Step-size Optimization: Heuristics

Individual learning rate for each parameter

$$\Delta w_i = -\eta_i \, g_i \qquad g_i = \left[\frac{\partial R(\boldsymbol{w})}{\partial \boldsymbol{w}}\right]_i$$

parameters not independent from each other
if $\eta_i > 0$ then the error decreases

delta-delta rule: $\Delta \eta_i = \gamma \, g_i^{\text{new}} \, g_i^{\text{old}}$

delta-bar-delta rule: $\Delta \eta_i = \begin{cases} \kappa & \text{if } \bar{g}_i^{\text{old}} \, g_i^{\text{new}} > 0 \\ -\phi \, g_i^{\text{new}} & \text{if } \bar{g}_i^{\text{old}} \, g_i^{\text{new}} \leq 0 \end{cases}$

where $\bar{g}_i^{\text{new}} = (1 - \theta) \, g_i^{\text{new}} + \theta \, \bar{g}_i^{\text{old}}$

$\bar{g}_i$ is an exponentially weighted average

disadvantage: many hyper-parameters

Step-size Optimization: Heuristics

Quickprop

developed in the context of neural networks ("backpropagation")

$$\Delta^{\text{new}} w_i = \frac{g_i}{g_i^{\text{old}} - g_i^{\text{new}}} \Delta^{\text{old}} w_i$$

Taylor expansion

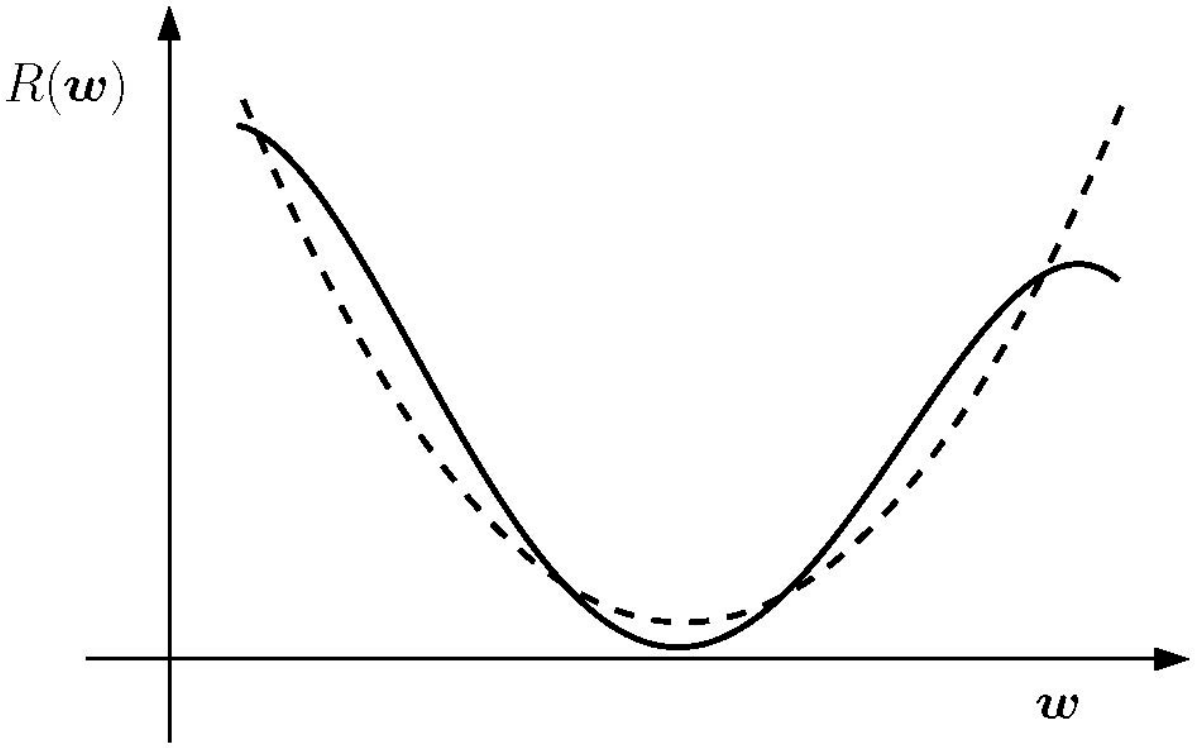
$$R(w_i + \Delta w_i) = R(w_i) + \frac{\partial R(\boldsymbol{w})}{\partial w_i} \Delta w_i +$$

$$\frac{1}{2} \frac{\partial^2 R(\boldsymbol{w})}{(\partial w_i)^2} (\Delta w_i)^2 + O\left((\Delta w_i)^3\right) =$$

$$R(w_i) + g_i \Delta w_i + \frac{1}{2} g_i' (\Delta w_i)^2 + O\left((\Delta w_i)^3\right)$$

Step-size Optimization: Heuristics

quadratic approximation



Step-size Optimization: Heuristics

minimize $R(w_i + \Delta w_i) - R(w_i)$ with respect to Δw_i

zero derivative: $\Delta w_i = -\frac{g_i}{g_i'}$

Now approximate $g_i' = g_i'(w_i)$ by $g_i'(w_i^{\text{old}})$

difference quotient: $g_i' = \frac{g_i^{\text{new}} - g_i^{\text{old}}}{\Delta^{\text{old}} w_i}$

$$g_i^{\text{old}} = g_i(w_i - \Delta^{\text{old}} w_i)$$

Line Search

update direction $\boldsymbol{d}$ is given $\Delta \boldsymbol{w} = \eta \, \boldsymbol{d}$

minimize $R(\boldsymbol{w} + \eta \, \boldsymbol{d})$ with respect to η

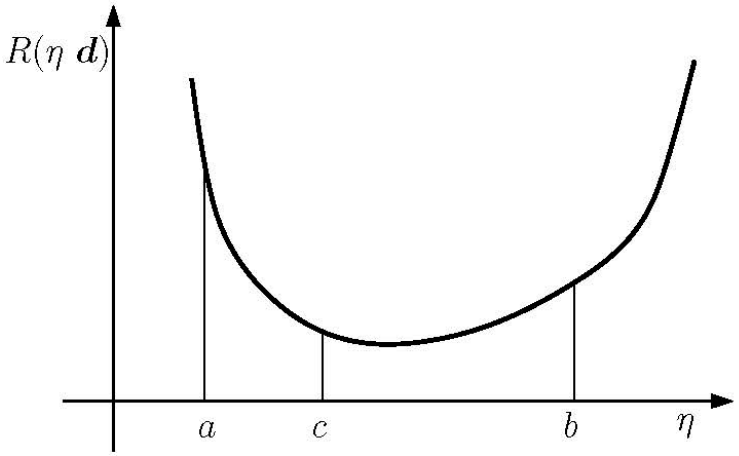
quadratic functions with minimum $\boldsymbol{w}^*$: $\eta = \dfrac{\boldsymbol{g}^T \boldsymbol{g}}{\boldsymbol{g}^T \boldsymbol{H}(\boldsymbol{w}^*) \boldsymbol{g}}$

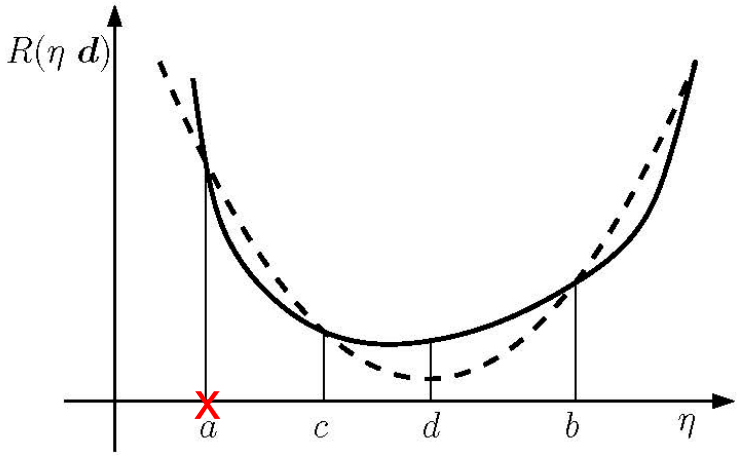
However: 1) valid only near the minimum and
2) $\boldsymbol{H}(\boldsymbol{w}^*)$ unknown

Line Search

Line search:

1. Fits a parabola though three points and determines its minimum
2. Adds minimum point to the points
3. Point with largest $\boldsymbol{R}$ is discharged





Line Search

Algorithm: Line-Search

BEGIN initialization $a_0, b_0, c_0; R(a_0) > R(c_0);$
$R(b_0) > R(c_0)$, Stop=false, $i = 0$
END initialization
BEGIN line search
while STOP=false **do**
fit quadratic function through a_i, b_i, c_i
determine minimum d_i of quadratic function
if stop criterion fulfilled, e.g. $|a_i - b_i| < \epsilon$ or $|R(b_0) - R(c_0)| < \epsilon$ **then**
Stop=true
else

$$c_{i+1} = d_i$$
$$b_{i+1} = c_i$$
$$a_{i+1} = \begin{cases} a_i & \text{if } R(a_i) \leq R(b_i) \\ b_i & \text{if } R(a_i) > R(b_i) \end{cases}$$

end if
$i = i + 1$
end while
END line search

Optimization of the Update Direction

The default direction is the negative gradient: $-\boldsymbol{g}$

However there are better approaches

Newton and Quasi-Newton Method

The gradient vanishes at the global minimum $\boldsymbol{w}^*$

$$\nabla_{\boldsymbol{w}} R(\boldsymbol{w}^*) = \boldsymbol{g}(\boldsymbol{w}^*) = \boldsymbol{0}$$

Taylor series:

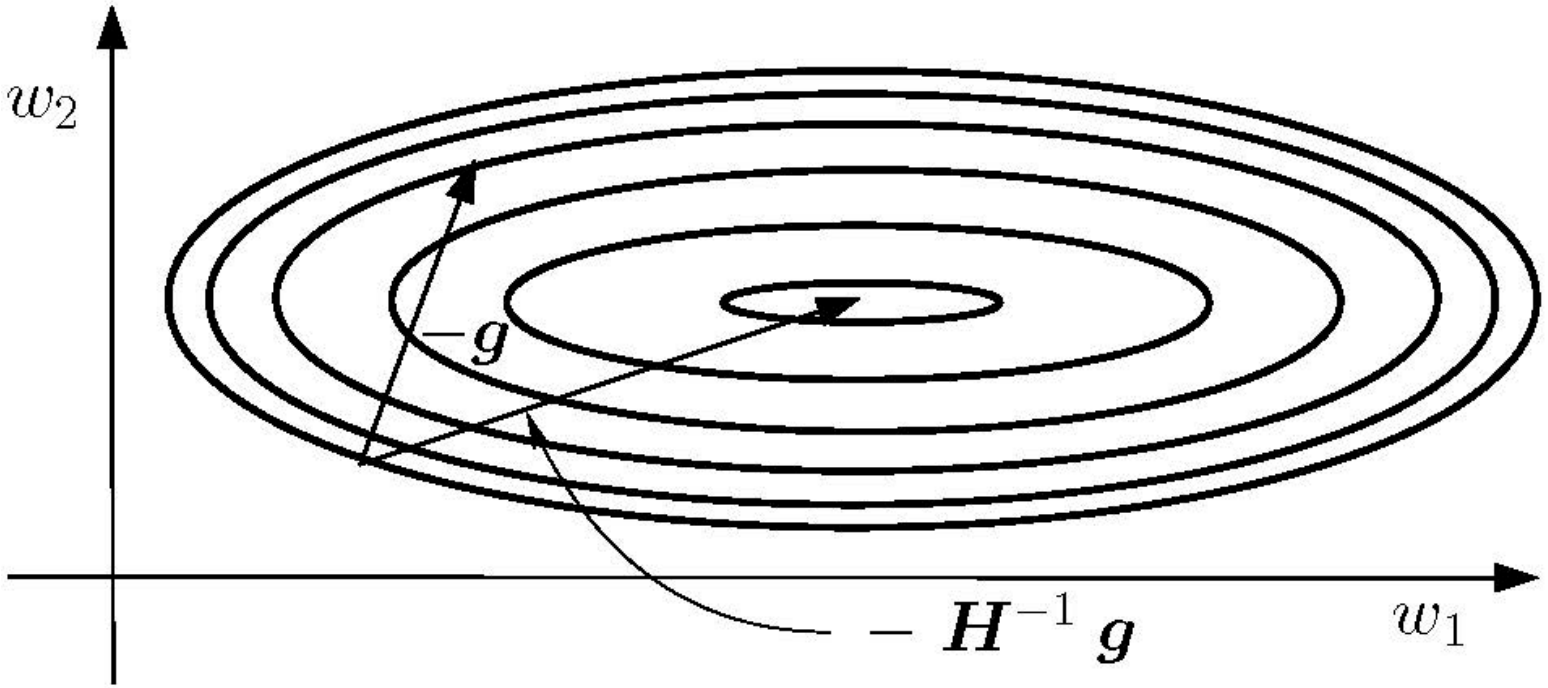
$$R(\boldsymbol{w}) = R(\boldsymbol{w}^*) + \frac{1}{2} (\boldsymbol{w} - \boldsymbol{w}^*)^T \boldsymbol{H}(\boldsymbol{w}^*) (\boldsymbol{w} - \boldsymbol{w}^*) + O\left(\|\boldsymbol{w} - \boldsymbol{w}^*\|^3\right)$$

quadratic approximation: $\boldsymbol{g} = \frac{\partial R}{\partial \boldsymbol{w}} = \boldsymbol{H}(\boldsymbol{w}^*) (\boldsymbol{w} - \boldsymbol{w}^*)$

Thus: $\boldsymbol{w}^* = \boldsymbol{w} - \boldsymbol{H}^{-1} \boldsymbol{g}$

Newton direction: $\boldsymbol{H}^{-1} \boldsymbol{g}$

Newton and Quasi-Newton Method



Newton and Quasi-Newton Method

Disadvantages

1. computationally expensive: inverse Hessian
2. only valid near the minimum (Hessian is positive definite)

model trust region approach: model is trusted up to a certain value

$\boldsymbol{H} + \lambda \boldsymbol{I}$ compromise between gradient and Newton direction

Hessian can be approximated by a diagonal matrix

Newton and Quasi-Newton Method

Quasi-Newton Method

Newton equation: $\boldsymbol{w}^{\text{new}} - \boldsymbol{w}^{\text{old}} = -\boldsymbol{H}^{-1}\left(\boldsymbol{g}^{\text{new}} - \boldsymbol{g}^{\text{old}}\right)$

quasi-Newton condition

Broyden-Fletcher-Goldfarb-Shanno (BFGS) method

$\boldsymbol{w}^{\text{new}} = \boldsymbol{w}^{\text{old}} + \eta\, \boldsymbol{G}^{\text{old}}\, \boldsymbol{g}^{\text{old}}$ where η is found by line search and

$$\boldsymbol{G}^{\text{new}} = \boldsymbol{G}^{\text{old}} + \frac{\boldsymbol{p}\,\boldsymbol{p}^T}{\boldsymbol{p}^T\,\boldsymbol{v}} - \frac{\left(\boldsymbol{G}^{\text{old}}\,\boldsymbol{v}\right)\,\boldsymbol{v}^T\boldsymbol{G}^{\text{old}}}{\boldsymbol{v}^T\boldsymbol{G}^{\text{old}}\,\boldsymbol{v}} + \left(\boldsymbol{v}^T\boldsymbol{G}^{\text{old}}\,\boldsymbol{v}\right)\,\boldsymbol{u}\,\boldsymbol{u}^T$$

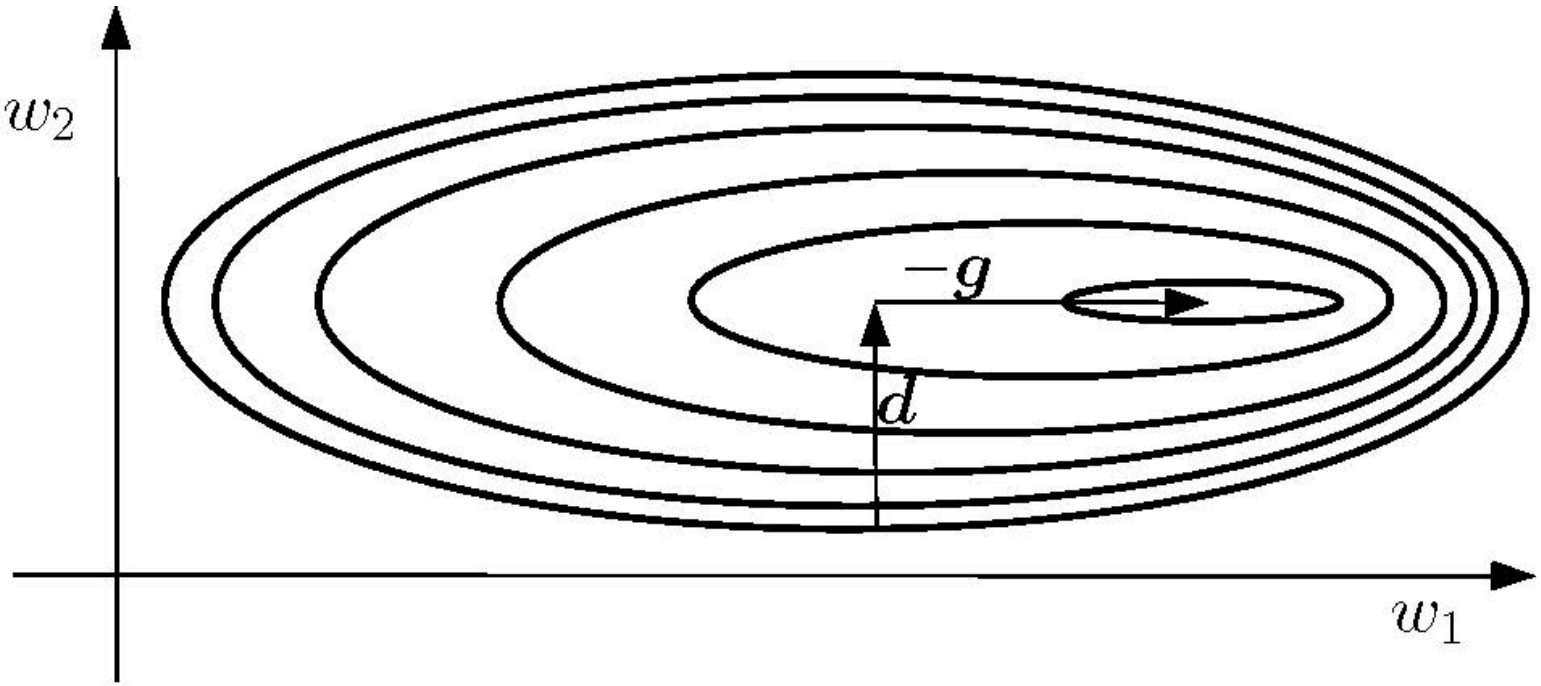
$\boldsymbol{G}$ approximates the inverse Hessian, therefore the update is more complicated

$$\boldsymbol{p} = \boldsymbol{w}^{\text{new}} - \boldsymbol{w}^{\text{old}}$$
$$\boldsymbol{v} = \boldsymbol{g}^{\text{new}} - \boldsymbol{g}^{\text{old}}$$
$$\boldsymbol{u} = \frac{\boldsymbol{p}}{\boldsymbol{p}^T\boldsymbol{v}} - \frac{\boldsymbol{G}^{\text{old}}\boldsymbol{v}}{\boldsymbol{v}^T\boldsymbol{G}^{\text{old}}\,\boldsymbol{v}}$$

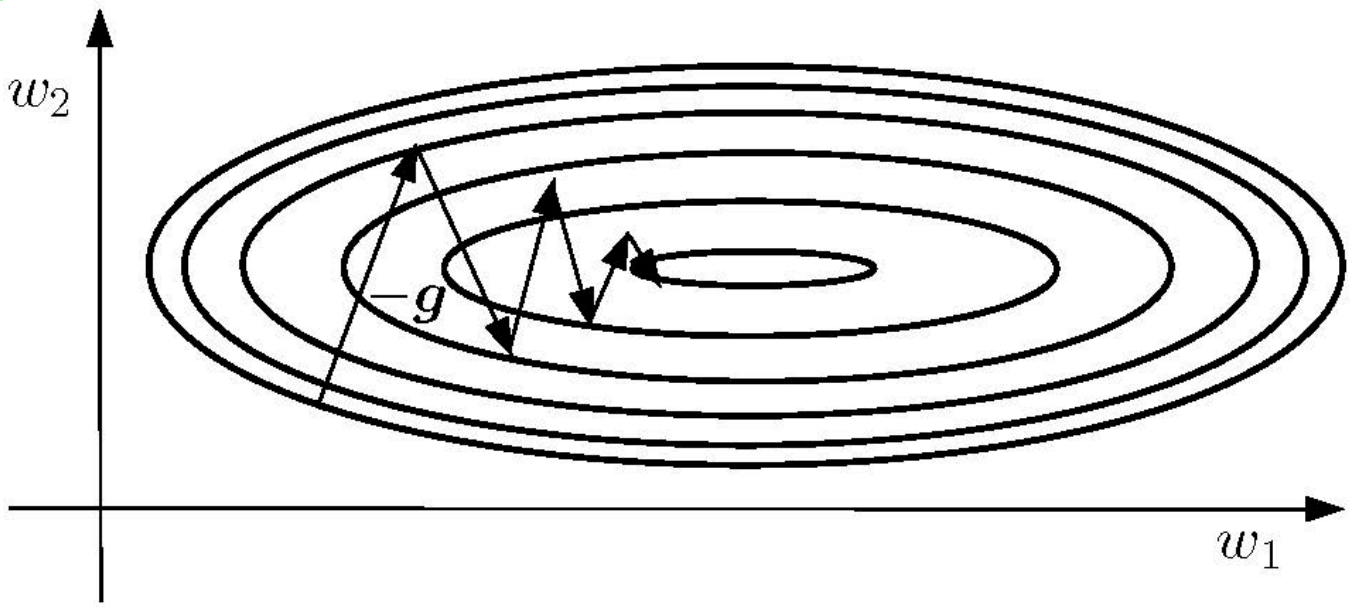
Conjugate Gradient

$R(\boldsymbol{w} + \eta\, \boldsymbol{d})$ is minimized with respect to η

$$\frac{\partial}{\partial \eta} R(\boldsymbol{w} + \eta\, \boldsymbol{d}) = 0 \quad \square \quad (\boldsymbol{g}^{\text{new}})^T\, \boldsymbol{d}^{\text{old}} = 0$$



Conjugate Gradient



still oscillations: different two-dimensional subspaces which alternate

avoid oscillations: new search directions are orthogonal to **all** previous

orthogonal is defined via the Hessian

in the parameter space the quadratic volume element is regarded

Conjugate Gradient

We require new gradient is orthogonal to old direction:

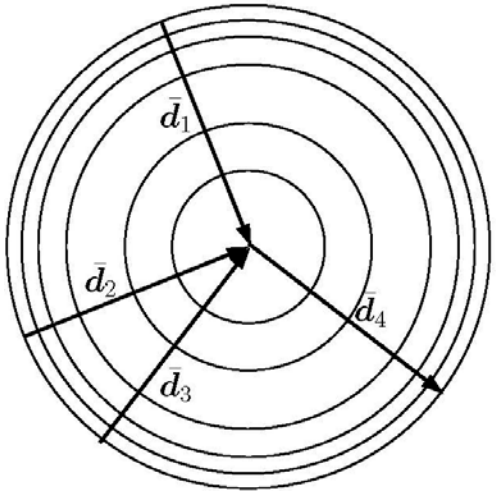
$$\boldsymbol{g}(\boldsymbol{w}^{\text{new}} + \eta \boldsymbol{d}^{\text{new}})^T \boldsymbol{d}^{\text{old}} = 0$$

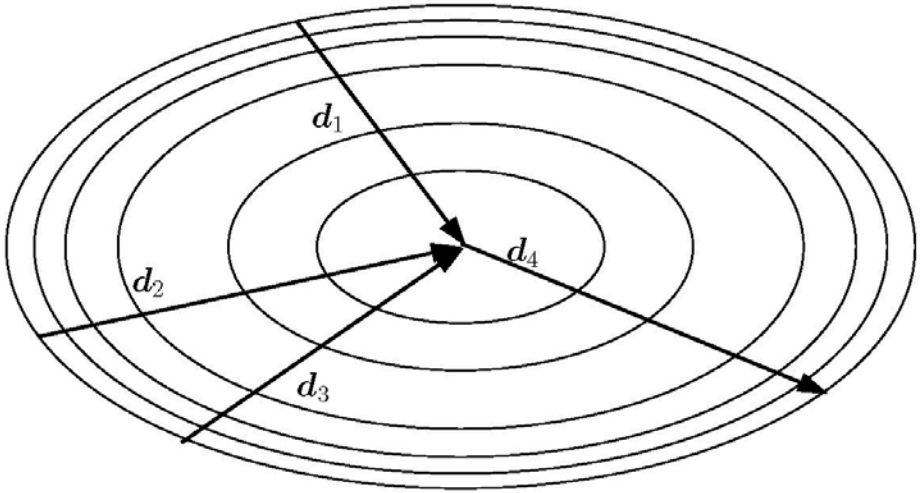
Taylor expansion:

$$\boldsymbol{g}(\boldsymbol{w}^{\text{new}} + \eta \boldsymbol{d}^{\text{new}})^T = \boldsymbol{g}(\boldsymbol{w}^{\text{new}}) + \eta \boldsymbol{H}(\boldsymbol{w}^{\text{new}}) \boldsymbol{d}^{\text{new}} + O(\eta^2)$$

and $\boldsymbol{g}(\boldsymbol{w}^{\text{new}})^T \boldsymbol{d}^{\text{old}} = 0$ follows:

$$(\boldsymbol{d}^{\text{new}})^T \boldsymbol{H}(\boldsymbol{w}^{\text{new}}) \boldsymbol{d}^{\text{old}} = 0$$ conjugate directions





$$\boldsymbol{d} = \boldsymbol{H}^{-1/2} \bar{\boldsymbol{d}}$$

$$\boldsymbol{d}_1^T \boldsymbol{H} \boldsymbol{d}_2 = 0 \qquad \boldsymbol{d}_3^T \boldsymbol{H} \boldsymbol{d}_4 = 0$$

Conjugate directions: orthogonal directions in other coordinate system

Conjugate Gradient

quadratic problem $R(\boldsymbol{w}) = \frac{1}{2}\, \boldsymbol{w}^T \boldsymbol{H} \boldsymbol{w} + \boldsymbol{c}^T \boldsymbol{w} + k$

$\boldsymbol{0} = \boldsymbol{H}\boldsymbol{w}^* + \boldsymbol{c} \qquad \boldsymbol{g}(\boldsymbol{w}^*) = \boldsymbol{H}\boldsymbol{w}^* + \boldsymbol{c} \qquad \boldsymbol{g}(\boldsymbol{w}_j) = \boldsymbol{g}_j = \boldsymbol{H}\boldsymbol{w}_j + \boldsymbol{c}$

conjugate directions are linearly independent, therefore

$$\boldsymbol{w}^* - \boldsymbol{w}_1 = \sum_{i=1}^{W} \eta_i \, \boldsymbol{d}_i$$

multiplied by $\boldsymbol{d}_j^T \boldsymbol{H}$

$$\boldsymbol{w}_j - \boldsymbol{w}_1 = \sum_{i=1}^{j-1} \eta_i \, \boldsymbol{d}_i$$

multiplied by $\boldsymbol{d}_j^T \boldsymbol{H}$

$\boldsymbol{H}\boldsymbol{w}^* = -\boldsymbol{c}$

$$-\boldsymbol{d}_j^T (\boldsymbol{c} + \boldsymbol{H}\boldsymbol{w}_1) = \sum_{i=1}^{W} \eta_i \, \boldsymbol{d}_j^T \boldsymbol{H} \, \boldsymbol{d}_i = \eta_j \, \boldsymbol{d}_j^T \boldsymbol{H} \, \boldsymbol{d}_j$$

$$\boxed{\boldsymbol{d}_j^T \boldsymbol{H} \, \boldsymbol{d}_i = 0} \qquad \boldsymbol{d}_j^T \boldsymbol{H} \, \boldsymbol{w}_j = \boldsymbol{d}_j^T \boldsymbol{H} \, \boldsymbol{w}_1$$

$\boldsymbol{g}_j = \boldsymbol{c} + \boldsymbol{H}\boldsymbol{w}_j$

Gives:

$$\boxed{\eta_j = -\frac{\boldsymbol{d}_j^T \boldsymbol{g}_j}{\boldsymbol{d}_j^T \boldsymbol{H} \, \boldsymbol{d}_j}}$$

Conjugate Gradient

$$\boldsymbol{w}_{j+1} = \boldsymbol{w}_j + \eta_j \, \boldsymbol{d}_j$$

search directions $\boldsymbol{d}_j$? We set

$$\boldsymbol{d}_{j+1} = -\boldsymbol{g}_{j+1} + \beta_j \, \boldsymbol{d}_j$$

Multiplying by $\boldsymbol{d}_j^T \boldsymbol{H}$ gives $\beta_j = \frac{\boldsymbol{g}_{j+1}^T \boldsymbol{H} \, \boldsymbol{d}_j}{\boldsymbol{d}_j^T \boldsymbol{H} \, \boldsymbol{d}_j}$

$$\boldsymbol{g}_{j+1} - \boldsymbol{g}_j = \boldsymbol{H} \, (\boldsymbol{w}_{j+1} - \boldsymbol{w}_j) = \eta_j \, \boldsymbol{H} \, \boldsymbol{d}_j$$

Hestenes – Stiefel :

$$\beta_j = \frac{\boldsymbol{g}_{j+1}^T \, (\boldsymbol{g}_{j+1} - \boldsymbol{g}_j)}{\boldsymbol{d}_j^T \, (\boldsymbol{g}_{j+1} - \boldsymbol{g}_j)}$$

Conjugate Gradient

$$\boldsymbol{d}_k^T \, \boldsymbol{g}_j \; = \; 0 \text{ for } k < j$$

gives

$$\boldsymbol{d}_j^T \, \boldsymbol{g}_j \; = \; - \, \boldsymbol{g}_j^T \, \boldsymbol{g}_j$$

Polak – Ribiere :

$$\beta_j \; = \; \frac{\boldsymbol{g}_{j+1}^T \left(\boldsymbol{g}_{j+1} \; - \; \boldsymbol{g}_j \right)}{\boldsymbol{g}_j^T \, \boldsymbol{g}_j}$$

Fletcher – Reeves :

$$\beta_j \; = \; \frac{\boldsymbol{g}_{j+1}^T \boldsymbol{g}_{j+1}}{\boldsymbol{g}_j^T \, \boldsymbol{g}_j}$$

Conjugate Gradient

Updates are mathematically equivalent but numerical different
The Polak-Ribiere has an edge over the other update rules

Conjugate Gradient (Polak-Ribiere)

BEGIN initialization $\boldsymbol{g}_0 = \nabla_{\boldsymbol{w}} R(\boldsymbol{w}_0)$, $i = 0$, $\boldsymbol{d}_0 = -\boldsymbol{g}_0$, Stop=false
END initialization
BEGIN Conjugate Gradient
while STOP=false **do**
determine η_i by line search
$\boldsymbol{w}_{j+1} = \boldsymbol{w}_j + \eta_j \boldsymbol{d}_j$
$\boldsymbol{g}_{i+1} = \nabla_{\boldsymbol{w}} R(\boldsymbol{w}_i)$
$\beta_j = \frac{\boldsymbol{g}_{j+1}^T (\boldsymbol{g}_{j+1} - \boldsymbol{g}_j)}{\boldsymbol{g}_j^T \boldsymbol{g}_j}$
$\boldsymbol{d}_{j+1} = -\boldsymbol{g}_{j+1} + \beta_j \boldsymbol{d}_j$
if stop criterion fulfilled, e.g. $\|g_{i+1}\|| < \epsilon$ or $|R(\boldsymbol{w}_{j+1}) - R(\boldsymbol{w}_j)| < \epsilon$ **then**
STOP=true
end if
$i = i + 1$
end while
END Conjugate Gradient

Conjugate Gradient

η are in most implementations found by line search

disadvantage of conjugate gradient compared to quasi-Newton:
the line search must be done precisely (orthogonal gradients)

advantage of conjugate gradient compared to quasi-Newton:
the storage is $O(W)$ compared to $O(W^2)$ for quasi-Newton

Levenberg-Marquardt Algorithm

This algorithm is designed for quadratic loss $R(\boldsymbol{w}) = \sum_{i=1}^{l} \left(e^i(\boldsymbol{w})\right)^2$

combine the errors e^i into a vector $\boldsymbol{e}$

Jacobian of this vector is $Z_{ij} = \frac{\partial e^i}{\partial w_j}$

linear approximation $\boldsymbol{e}(\boldsymbol{w}^{\text{new}}) = \boldsymbol{e}(\boldsymbol{w}^{\text{old}}) + \boldsymbol{Z}\left(\boldsymbol{w}^{\text{new}} - \boldsymbol{w}^{\text{old}}\right)$

Hessian of the loss function:

$$H_{jk} = \frac{\partial^2 R}{\partial w_j \, \partial w_k} = \sum_{i=1}^{l} \left(\frac{\partial e^i}{\partial w_j} \frac{\partial e^i}{\partial w_k} + e^i \frac{\partial^2 e^i}{\partial w_j \, \partial w_k} \right)$$

Small e^i or $e^i \frac{\partial^2 e^i}{\partial w_j \, \partial w_k}$ averages out (negative and positive e^i)

approximate the Hessian by $\boldsymbol{H} = \boldsymbol{Z}^T \boldsymbol{Z}$ (outer product)

Note that this is only valid for quadratic loss functions

Levenberg-Marquardt Algorithm

unconstraint minimization problem

$$\frac{1}{2} \left\| e(\boldsymbol{w}^{\text{old}}) + \boldsymbol{Z} \left(\boldsymbol{w}^{\text{new}} - \boldsymbol{w}^{\text{old}} \right) \right\|^2 + \lambda \left\| \boldsymbol{w}^{\text{new}} - \boldsymbol{w}^{\text{old}} \right\|^2$$

first term: minimizing the error
second term: minimizing the step size (for valid linear approximations)

Levenberg-Marquardt update rule

$$\boldsymbol{w}^{\text{new}} = \boldsymbol{w}^{\text{old}} - \left(\boldsymbol{Z}^T \boldsymbol{Z} + \lambda \boldsymbol{I} \right)^{-1} \boldsymbol{Z}^T \boldsymbol{e}(\boldsymbol{w}^{\text{old}})$$

Small λ gives the Newton formula while large λ gives gradient descent

The Levenberg-Marquardt algorithm is a model trust region approach

Predictor Corrector Methods for $R(w) = 0$

Problem: solve $R(\boldsymbol{w}) = 0$

Idea: simple update (predictor step) by neglecting higher order terms
then: correction by the higher order terms (corrector step)

Taylor series

$$R(\boldsymbol{w}^{\text{new}}) = R(\boldsymbol{w}^{\text{old}}) + S(\boldsymbol{w}^{\text{old}}, \Delta\boldsymbol{w}) + TS(\boldsymbol{w}^{\text{old}}, \Delta\boldsymbol{w})$$

$$S((\boldsymbol{w}^{\text{old}}, \boldsymbol{0}) = 0 \qquad T((\boldsymbol{w}^{\text{old}}, \boldsymbol{0}) = 0$$

predictor step

$$R(\boldsymbol{w}^{\text{old}}) + S(\boldsymbol{w}^{\text{old}}, \Delta\boldsymbol{w}) = 0 \rightarrow \Delta\boldsymbol{w}$$

corrector step

$$R(\boldsymbol{w}^{\text{old}}) + S(\boldsymbol{w}^{\text{old}}, \Delta\boldsymbol{w}) + TS(\boldsymbol{w}^{\text{old}}, \Delta_{\text{pred}}\boldsymbol{w}) = 0$$

Iterative algorithm

Convergence Properties

Gradient Descent

$$R\left(\boldsymbol{w}^{\text{old}}\right) - R\left(\boldsymbol{w}^{\text{new}}\right) =$$

$$R\left(\boldsymbol{w}^{\text{old}}\right) \left(\frac{\left(\boldsymbol{g}^T \boldsymbol{g}\right)^2}{\left(\boldsymbol{g}^T \boldsymbol{H}(\boldsymbol{w}^*) \boldsymbol{g}\right) \left(\boldsymbol{g}^T \boldsymbol{H}^{-1}(\boldsymbol{w}^*) \boldsymbol{g}\right)} \right)$$

Kantorovich inequality:

$$\frac{\left(\boldsymbol{g}^T \boldsymbol{g}\right)^2}{\left(\boldsymbol{g}^T \boldsymbol{H} \boldsymbol{g}\right) \left(\boldsymbol{g}^T \boldsymbol{H}^{-1} \boldsymbol{g}\right)} \geq \frac{4 \lambda_{\min} \lambda_{\max}}{\left(\lambda_{\min} + \lambda_{\max}\right)^2} \geq \frac{1}{\text{cond}(\boldsymbol{H})}$$

condition of a matrix

$$\text{cond}(\boldsymbol{H}) = \frac{\lambda_{\max}}{\lambda_{\min}}$$

Convergence Properties

Newton Method

the Newton method is quadratic convergent in $\left\| \boldsymbol{w}^{\text{old}} - \boldsymbol{w}^* \right\|$:

$$\boldsymbol{w}^{\text{new}} - \boldsymbol{w}^* = O\left(\left\| \boldsymbol{w}^{\text{old}} - \boldsymbol{w}^* \right\|^2 \right)$$

assumed that

$$\left\| \frac{1}{2} \boldsymbol{H}^{-1} \left(\boldsymbol{\xi}_1^T \boldsymbol{H}_1(\boldsymbol{w}^{\text{old}}) \, \boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_W^T \boldsymbol{H}_W(\boldsymbol{w}^{\text{old}}) \, \boldsymbol{\xi}_W \right)^T \right\| < 1$$

where

$$\boldsymbol{\xi}_i = \lambda \left(\boldsymbol{w}^* - \boldsymbol{w}^{\text{old}} \right), \quad 0 \leq \lambda \leq 1,$$

and $\boldsymbol{H}_i$ is the Hessian of $\boldsymbol{g}_i$

On-line Optimization

Until now fixed training set where update uses all examples
→batch update

Incremental update, where a single example is used for update
→ on-line update

A) overfitting
Cheap training examples → overfitting is avoided if the training size is large enough (empirical risk minimization)

large training sets → computationally too expensive

→ on-line methods

B) non-stationary problems
non-stationary problems (dynamics change)
→ on-line methods can track data dependencies

On-line Optimization

goal: find $\boldsymbol{w}^*$, for which $g(\boldsymbol{w}^*) = 0$

$$g(\boldsymbol{w}) = \mathrm{E}\left(f(\boldsymbol{w}) \mid \boldsymbol{w}\right)$$

with finite variance $\mathrm{E}\left((g - f)^2 \mid \boldsymbol{w}\right) < \infty$

Robbins-Monro procedure

$$\boldsymbol{w}^{i+1} = \boldsymbol{w}^i - \eta_i \, f(\boldsymbol{w}^i)$$

where $g(\boldsymbol{w}^i)$ is a random variable and the learning rate sequence η_i satisfies

$$\lim_{i \to \infty} \eta_i = 0$$ convergence

$$\sum_{i=1}^{\infty} \eta_i = \infty$$ changes are sufficient large to find the root

$$\sum_{i=1}^{\infty} \eta_i^2 < \infty$$ noise variance is limited

On-line Optimization

Theorem 1 (Robbins-Monro)

Under above conditions the above sequence converges to the root $\boldsymbol{w}^*$ of $\boldsymbol{g}$ with probability 1.

Robbins-Monro for maximum likelihood:

$$\frac{1}{l} \frac{\partial}{\partial \boldsymbol{w}} \sum_{i=1}^{l} \ln p(\boldsymbol{x}^i \mid \boldsymbol{w}) = 0$$

$$\lim_{l \to \infty} \frac{1}{l} \sum_{i=1}^{l} \frac{\partial}{\partial \boldsymbol{w}} \ln p(\boldsymbol{x}^i \mid \boldsymbol{w}) = \mathrm{E}\left(\frac{\partial}{\partial \boldsymbol{w}} \ln p(\boldsymbol{x}^i \mid \boldsymbol{w}) \right)$$

maximum likelihood solution is asymptotically equivalent to

$$\mathrm{E}\left(\frac{\partial}{\partial \boldsymbol{w}} \ln p(\boldsymbol{x}^i \mid \boldsymbol{w}) \right) = 0$$

On-line Optimization

Robbins-Monro procedure

$$\boldsymbol{w}^{i+1} = \boldsymbol{w}^{i} + \eta_i \frac{\partial}{\partial \boldsymbol{w}} \ln p(\boldsymbol{x}^{i+1} \mid \boldsymbol{w}) \mid_{\boldsymbol{w}^i}$$

online update formula for maximum likelihood

Convex Optimization

Convex Problems

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & \forall_i: \; c_i(\boldsymbol{x}) \leq 0 \\ & \forall_j: \; e_j(\boldsymbol{x}) = 0 \end{aligned}$$

f, c_i and e_j are convex functions

solution is a convex set
f is strictly convex: solution is unique

constraint convex minimization

SVM optimization problems

Convex Optimization

Lagrangian

$$L(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\mu}) = f(\boldsymbol{x}) + \sum_i \alpha_i \; c_i(\boldsymbol{x}) + \sum_j \mu_j \; e_j(\boldsymbol{x}) \qquad \alpha_i \geq 0$$

where α and μ are called "Lagrange multipliers"

Lagrangian also for non-convex functions

Convex Optimization

KKT conditions

feasible solution exists then the following statements are equivalent:
a) an $\boldsymbol{x}$ exists with $c_i(\boldsymbol{x}) < 0$ for all i (Slater's condition)
b) an $\boldsymbol{x}$ and $\alpha_i \geq 0$ exist such that $\sum_i \alpha_i \; c_i(\boldsymbol{x}) \; \leq \; 0$
(Karlin's condition)
Above statements a) or b) follow from the following statement
c) there exist at least two feasible solutions and a feasible $\boldsymbol{x}$ such that all c_i are strictly convex at $\boldsymbol{x}$ wrt. the feasible set (strict constraint qualification).

The saddle point condition of Kuhn-Tucker: if one of a) – c) holds then

$$L\left(\hat{\boldsymbol{x}}, \boldsymbol{\alpha}, \boldsymbol{\mu}\right) \; \leq \; L\left(\hat{\boldsymbol{x}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\mu}}\right) \; \leq \; L\left(\boldsymbol{x}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\mu}}\right)$$

is necessary and sufficient for $\left(\hat{\boldsymbol{x}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\mu}}\right)$ being a solution to the optimization problem.

Note, that ”sufficient” also holds for non-convex functions

Convex Optimization

From $L(\hat{\boldsymbol{x}}, \boldsymbol{\alpha}, \boldsymbol{\mu}) \leq L(\hat{\boldsymbol{x}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\mu}})$ follows that

$$\sum_i (\alpha_i - \hat{\alpha}_i)\ c_i(\hat{\boldsymbol{x}}) + \sum_j (\mu_j - \hat{\mu}_j)\ e_j(\hat{\boldsymbol{x}}) \leq 0$$

$\mu_j = \hat{\mu}_j, \alpha_i = \hat{\alpha}_i, \alpha_k = \hat{\alpha}_k + 1 \rightarrow c_k(\hat{\boldsymbol{x}}) \leq 0$

Analog for $e_k(\hat{\boldsymbol{x}}) = 0$ ➔ $\hat{\boldsymbol{x}}$ fulfills the constraints

$$\mu_j = \hat{\mu}_j,\ \alpha_i = \hat{\alpha}_i,\ \alpha_k = 0 \rightarrow \hat{\alpha}_k\ c_k(\hat{\boldsymbol{x}}) \geq 0$$

From above $c_k(\hat{\boldsymbol{x}}) \leq 0$ and $\hat{\alpha}_k \geq 0 \rightarrow \hat{\alpha}_k\ c_k(\hat{\boldsymbol{x}}) \leq 0$

$$\hat{\alpha}_i\ c_i(\hat{\boldsymbol{x}}) = 0$$

analog

$$\hat{\mu}_j\ e_j(\hat{\boldsymbol{x}}) = 0$$

“Karush-Kuhn-Tucker” (KKT) conditions

Convex Optimization

differentiable problems: minima and maxima can be determined

Theorem 1 (KKT and Differentiable Convex Problems) *A solution to the problem with convex, differentiable f, c_i, and e_j is given by $\hat{\boldsymbol{x}}$ if $\hat{\alpha}_i \geq 0$ and $\hat{\mu}_j$ exist which satisfy:*

$$\frac{\partial L(\hat{\boldsymbol{x}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\mu}})}{\partial \boldsymbol{x}} = \frac{\partial f(\hat{\boldsymbol{x}})}{\partial \boldsymbol{x}} + \sum_i \hat{\alpha}_i \frac{\partial c_i(\hat{\boldsymbol{x}})}{\partial \boldsymbol{x}} + \sum_j \hat{\mu}_j \frac{\partial e_j(\hat{\boldsymbol{x}})}{\partial \boldsymbol{x}} = 0$$

$$\frac{\partial L(\hat{\boldsymbol{x}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\mu}})}{\partial \alpha_i} = c_i(\hat{\boldsymbol{x}}) \leq 0$$

$$\frac{\partial L(\hat{\boldsymbol{x}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\mu}})}{\partial \mu_j} = e_j(\hat{\boldsymbol{x}}) = 0$$

$$\forall_i : \quad \hat{\alpha}_i \; c_i(\hat{\boldsymbol{x}}) = 0$$

$$\forall_j : \quad \hat{\mu}_j \; e_j(\hat{\boldsymbol{x}}) = 0$$

Convex Optimization

Constraints fulfilled by x and α:

$$f(\boldsymbol{x}) \;\geq\; f(\hat{\boldsymbol{x}}) \;\geq\; f(\boldsymbol{x}) \;+\; \sum_i \alpha_i \; c_i(\boldsymbol{x}) \qquad e_j(\boldsymbol{x}) = 0$$

KKT gap: $\sum_i \alpha_i \; c_i(\boldsymbol{x})$

Wolfe's dual to the optimization problem

$$\max_{\boldsymbol{x},\boldsymbol{\alpha},\boldsymbol{\mu}} \qquad f(\boldsymbol{x}) \;+\; \sum_i \alpha_i \; c_i(\boldsymbol{x}) \;+\; \sum_j \mu_j \; e_j(\boldsymbol{x})$$

$$\text{s.t.} \qquad \forall_i : \quad \alpha_i \;\geq\; 0$$

$$\frac{\partial L(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\mu})}{\partial \boldsymbol{x}} \;=\; 0$$

The solutions of the dual are the solutions of the primal
If $\frac{\partial L(\boldsymbol{x},\boldsymbol{\alpha},\boldsymbol{\mu})}{\partial \boldsymbol{x}} = 0$ can be solved for $\boldsymbol{x}$ and inserted into the dual,
then we obtain a maximization problem in α and μ

Convex Optimization

Linear Programs

$$\min_{\boldsymbol{x}} \quad \boldsymbol{c}^T \boldsymbol{x}$$
$$\text{s.t.} \quad \boldsymbol{A}\,\boldsymbol{x} + \boldsymbol{d} \leq \boldsymbol{0}$$

where $\boldsymbol{A}\,\boldsymbol{x} + \boldsymbol{d} \leq \boldsymbol{0}$ means $\forall_i : \sum_{j=1}^{l} A_{ij}\, x^j + d_j \leq 0$

Lagrangian $L = \boldsymbol{c}^T \boldsymbol{x} + \boldsymbol{\alpha}^T (\boldsymbol{A}\,\boldsymbol{x} + \boldsymbol{d})$

optimality conditions

$$\frac{\partial L}{\partial \boldsymbol{x}} = \boldsymbol{A}^T \boldsymbol{\alpha} + \boldsymbol{c} = 0$$
$$\frac{\partial L}{\partial \boldsymbol{\alpha}} = \boldsymbol{A}\,\boldsymbol{x} + \boldsymbol{d} \leq \boldsymbol{0}$$
$$\boldsymbol{\alpha}^T (\boldsymbol{A}\,\boldsymbol{x} + \boldsymbol{d}) = \boldsymbol{0}$$
$$\boldsymbol{\alpha} \geq \boldsymbol{0}$$

dual formulation

$$\max_{\boldsymbol{\alpha}} \quad \boldsymbol{d}^T \boldsymbol{\alpha}$$
$$\text{s.t.} \quad \boldsymbol{A}^T \boldsymbol{\alpha} + \boldsymbol{c} = \boldsymbol{0}$$
$$\boldsymbol{\alpha} \geq \boldsymbol{0}$$

Convex Optimization

dual of the dual

$$\min_{\boldsymbol{\alpha},\boldsymbol{\mu}} \quad \boldsymbol{c}^T \boldsymbol{x}'$$

$$\text{s.t.} \quad \boldsymbol{A}\,\boldsymbol{x}' + \boldsymbol{d} + \boldsymbol{\mu} = \boldsymbol{0}$$

$$\boldsymbol{\mu} \geq \boldsymbol{0}$$

$\boldsymbol{\mu}$ can chosen free if we ensure $\boldsymbol{A}\,\boldsymbol{x}' + \boldsymbol{d} \leq \boldsymbol{0}$

we obtain again the primal

Convex Optimization

Quadratic Programs

$$\min_{\boldsymbol{x}} \quad \frac{1}{2}\boldsymbol{x}^T\boldsymbol{K}\boldsymbol{x} + \boldsymbol{c}^T\boldsymbol{x}$$
$$\text{s.t.} \quad \boldsymbol{A}\,\boldsymbol{x} + \boldsymbol{d} \leq \boldsymbol{0}$$

Lagrangian

$$L(\boldsymbol{x}, \boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{K}\boldsymbol{x} + \boldsymbol{c}^T\boldsymbol{x} + \boldsymbol{\alpha}^T(\boldsymbol{A}\,\boldsymbol{x} + \boldsymbol{d})$$

optimality conditions

$$\frac{\partial L}{\partial \boldsymbol{x}} =$$
$$\boldsymbol{K}\,\boldsymbol{x} + \boldsymbol{A}^T\boldsymbol{\alpha} + \boldsymbol{c} = 0$$
$$\frac{\partial L}{\partial \boldsymbol{\alpha}} = \boldsymbol{A}\,\boldsymbol{x} + \boldsymbol{d} \leq \boldsymbol{0}$$
$$\boldsymbol{\alpha}^T(\boldsymbol{A}\,\boldsymbol{x} + \boldsymbol{d}) = \boldsymbol{0}$$
$$\boldsymbol{\alpha} \geq \boldsymbol{0}$$

dual

$$\min_{\boldsymbol{x}} \quad \frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{A}\boldsymbol{K}^{-1}\boldsymbol{A}^T\boldsymbol{\alpha} - \left(\boldsymbol{d}^T - \boldsymbol{c}^T\boldsymbol{K}^{-1}\boldsymbol{A}^T\right)\boldsymbol{\alpha}$$
$$\text{s.t.} \quad \boldsymbol{0} \leq \boldsymbol{\alpha}$$

Convex Optimization

Optimization of Convex Problems

constraint gradient descent

interior point methods (interior point is a pair $(\boldsymbol{x}, \alpha)$ which satisfies both the primal and dual constraints)

predictor-corrector methods: anneal KKT conditions (which are the only quadratic conditions in the variables)

Chapter 8

Bayes Techniques

Bayes Techniques

probabilistic framework for the empirical error and regularization

Focus: neural networks but valid for other models

Bayes techniques allow

- to introduce a probabilistic framework
- to deal with hyper-parameters
- to supply error bars and confidence intervals for the model output
- to compare different models
- to select relevant features
- to make averages and committees

Likelihood, Prior, Posterior, Evidence

training data $\{\boldsymbol{z}^1, \ldots, \boldsymbol{z}^l\}$ $(\boldsymbol{z}^i = (\boldsymbol{x}^i, y^i))$ $\{\boldsymbol{z}\} = \{\boldsymbol{z}^1, \ldots, \boldsymbol{z}^l\}$

matrix of feature vectors $\boldsymbol{X} = \left(\boldsymbol{x}^1, \ldots, \boldsymbol{x}^l\right)^T$

vector of labels $\boldsymbol{y} = \left(y^1, \ldots, y^l\right)^T$

training data matrix $\boldsymbol{Z} = \left(\boldsymbol{z}^1, \ldots, \boldsymbol{z}^l\right)$

Likelihood $\mathcal{L}$: $\mathcal{L}(\{\boldsymbol{z}\}; \boldsymbol{w}) = p(\{\boldsymbol{z}\}; \boldsymbol{w})$

probability of the model $p(\boldsymbol{z}; \boldsymbol{w})$ to produce the data set

iid data: $\mathcal{L}(\{\boldsymbol{z}\}; \boldsymbol{w}) = p(\{\boldsymbol{z}\}; \boldsymbol{w}) = \prod_{i=1}^{l} p(\boldsymbol{z}^i; \boldsymbol{w})$

Likelihood, Prior, Posterior, Evidence

supervised learning: $p(\boldsymbol{z}; \boldsymbol{w}) = p(\boldsymbol{x})\ p(y \mid \boldsymbol{x}; \boldsymbol{w})$

$$\mathcal{L}(\{\boldsymbol{z}\}; \boldsymbol{w}) = \prod_{i=1}^{l} p(\boldsymbol{x}^i) \prod_{i=1}^{l} p(y^i \mid \boldsymbol{x}^i; \boldsymbol{w})$$

$\prod_{i=1}^{l} p(\boldsymbol{x}^i)$ is independent of the parameters

sufficient to maximize: $\mathcal{L}(\{y\} \mid \{\boldsymbol{x}\}; \boldsymbol{w}) = \prod_{i=1}^{l} p(y^i \mid \boldsymbol{x}^i; \boldsymbol{w})$

likelihood is conditioned on $\boldsymbol{w}$

likelihood $p(\{\boldsymbol{z}\} \mid \boldsymbol{w})$

Likelihood, Prior, Posterior, Evidence

Overfitting: training examples with likelihood
other data with probability zero
sum of Dirac delta-distributions

avoid overfitting: some models are more likely in the world

prior distribution: $p(\boldsymbol{w})$

from prior problem knowledge without seeing the data

Likelihood, Prior, Posterior, Evidence

Bayes formula $$p(\boldsymbol{w} \mid \{\boldsymbol{z}\}) = \frac{p(\{\boldsymbol{z}\} \mid \boldsymbol{w})\ p(\boldsymbol{w})}{p_{\boldsymbol{w}}(\{\boldsymbol{z}\})}$$

posterior distribution: $p(\boldsymbol{w} \mid \{\boldsymbol{z}\})$

evidence: $p_{\boldsymbol{w}}(\{\boldsymbol{z}\}) = \int_W p(\{\boldsymbol{z}\} \mid \boldsymbol{w})\ p(\boldsymbol{w})\ d\boldsymbol{w}$

normalization constant:
accessible volume of the configuration space (statistical mechanics)
error moment generating function

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$$

Likelihood, Prior, Posterior, Evidence

if $p(\boldsymbol{w})$ is indeed the distribution of model parameters in real world

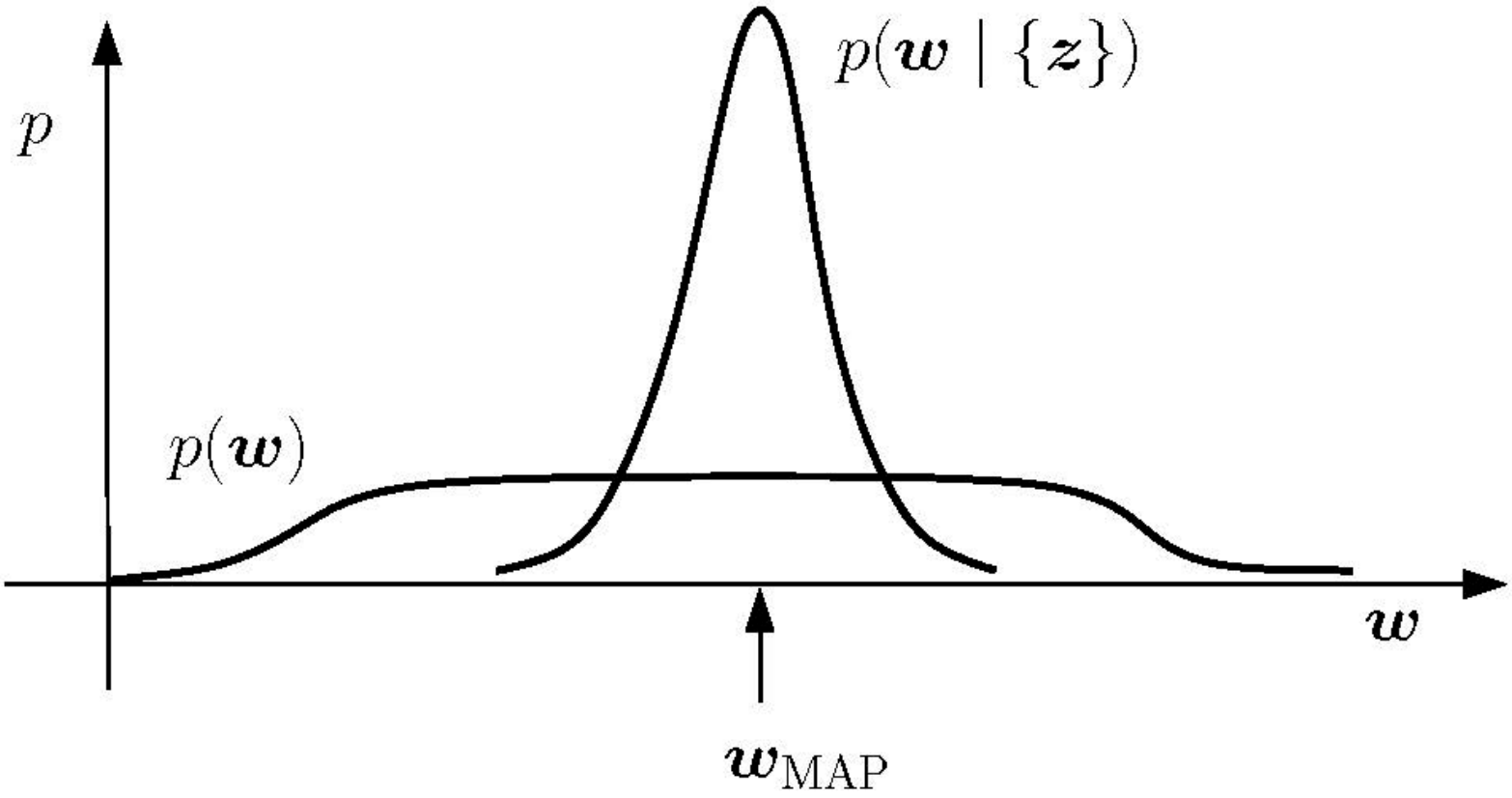
$$p_{\boldsymbol{w}}(\{\boldsymbol{z}\}) = p(\{\boldsymbol{z}\})$$

The real data is indeed produced by first choosing $\boldsymbol{w}$ according to $p(\boldsymbol{w})$ and then generating $\{\boldsymbol{z}\}$ through $p(\{\boldsymbol{z}\} \mid \boldsymbol{w})$

data in real world is not produced according to $p(\boldsymbol{w})$
therefore $p_{\boldsymbol{w}}(\{\boldsymbol{z}\})$ is not the distribution of occurrence of data
but gives the probability of observing data with the model class

Maximum A Posteriori Approach

Maximum A Posteriori Approach (MAP):
maximal posterior $p(\boldsymbol{w} \mid \{\boldsymbol{z}\})$



Maximum A Posteriori Approach

Prior: weight decay $\Omega(\boldsymbol{w}) = \|\boldsymbol{w}\|^2 = \boldsymbol{w}^T \boldsymbol{w} = \sum_{ij} w_{ij}^2$

Gaussian weight prior

$$p(\boldsymbol{w}) = \frac{1}{Z_w(\alpha)} \exp\left(-\frac{1}{2}\,\alpha\,\|\boldsymbol{w}\|^2\right)$$

$$Z_w(\alpha) = \int_W \exp\left(-\frac{1}{2}\,\alpha\,\|\boldsymbol{w}\|^2\right) d\boldsymbol{w} = \left(\frac{2\,\pi}{\alpha}\right)^{W/2}$$

α hyper-parameter: trades error term against the complexity term

Maximum A Posteriori Approach

other weight decay terms:

$$\Omega(\boldsymbol{w}) = \|\boldsymbol{w}\|_1 \quad \text{Laplace distribution}$$

$$p(\boldsymbol{w}) = \frac{1}{Z_w(\alpha)} \exp\left(-\frac{1}{2}\,\alpha\,\|\boldsymbol{w}\|_1\right)$$

$$Z_w(\alpha) = \int_W \exp\left(-\frac{1}{2}\,\alpha\,\|\boldsymbol{w}\|_1\right) d\boldsymbol{w}$$

$$\Omega(\boldsymbol{w}) = \log\left(1 + \|\boldsymbol{w}\|^2\right) \quad \text{Cauchy distribution}$$

$$p(\boldsymbol{w}) = \frac{1}{Z_w(\alpha)} \exp\left(-\frac{1}{2}\,\alpha\right)\left(1 + \|\boldsymbol{w}\|^2\right)$$

$$Z_w(\alpha) = \int_W \exp\left(-\frac{1}{2}\,\alpha\right)\left(1 + \|\boldsymbol{w}\|^2\right) d\boldsymbol{w}$$

Maximum A Posteriori Approach

Gaussian noise models

$$p(\{\boldsymbol{z}\} \mid \boldsymbol{w}) = \frac{1}{(2\,\pi)^{d/2}\,|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}\,(\boldsymbol{y} - \boldsymbol{X}\,\boldsymbol{w})^T\,\boldsymbol{\Sigma}^{-1}(\boldsymbol{y} - \boldsymbol{X}\,\boldsymbol{w})\right)\,p(\{\boldsymbol{x}\})$$

and for $\boldsymbol{\Sigma} = \sigma^2\,\boldsymbol{I}$

$$p(\{\boldsymbol{z}\} \mid \boldsymbol{w}) = \frac{1}{(2\,\pi)^{d/2}\;\sigma^d} \exp\left(-\frac{1}{2\,\sigma^2}\,(\boldsymbol{y} - \boldsymbol{X}\,\boldsymbol{w})^T\,(\boldsymbol{y} - \boldsymbol{X}\,\boldsymbol{w})\right)\,p(\{\boldsymbol{x}\})$$

$R_{\text{emp}} = (\boldsymbol{y} - \boldsymbol{X}\,\boldsymbol{w})^T\,(\boldsymbol{y} - \boldsymbol{X}\,\boldsymbol{w})$ mean squared error

Maximum A Posteriori Approach

negative log-posterior

$$-\log p(\boldsymbol{w} \mid \{\boldsymbol{z}\}) = -\log p(\{\boldsymbol{z}\} \mid \boldsymbol{w}) - \log p(\boldsymbol{w}) + \log p_{\boldsymbol{w}}(\{\boldsymbol{z}\})$$

where $p_{\boldsymbol{w}}(\{\boldsymbol{z}\})$ does not depend on $\boldsymbol{w}$

maximum a posteriori

$$\tilde{R}(\boldsymbol{w}) = \frac{1}{2\,\sigma^2} R_{\text{emp}} + \frac{1}{2}\alpha\,\Omega(\boldsymbol{w}) = \frac{1}{2}\,\beta\,R_{\text{emp}} + \frac{1}{2}\,\alpha\,\Omega(\boldsymbol{w})$$

where $\beta^{-1} = \sigma^2$

$$R(\boldsymbol{w}) = \tilde{R}(\boldsymbol{w})\;2\;\sigma^2 \qquad \lambda = \sigma^2\;\alpha$$

$$R(\boldsymbol{w}) = R_{\text{emp}} + \lambda\,\Omega(\boldsymbol{w})$$ this is exactly weight decay

empirical error plus complexity: maximum a posteriori

Maximum A Posteriori Approach

Likelihood: exponential function of empirical error

$$p(\{\boldsymbol{z}\} \mid \boldsymbol{w}) = \frac{1}{Z_R(\beta)} \exp\left(-\frac{1}{2}\,\beta\, R_{\text{emp}}\right)$$

Prior: exponential function of the complexity

$$p(\boldsymbol{w}) = \frac{1}{Z_w(\alpha)} \exp\left(-\frac{1}{2}\,\alpha\, \Omega(\boldsymbol{w})\right)$$

Posterior:

$$p(\boldsymbol{w} \mid \{\boldsymbol{z}\}) = \frac{1}{Z(\alpha,\beta)} \exp\left(-\frac{1}{2}\,(\alpha\, \Omega(\boldsymbol{w}) + \beta\, R_{\text{emp}})\right)$$

where

$$Z(\alpha,\beta) = \int_W \exp\left(-\frac{1}{2}\,(\alpha\, \Omega(\boldsymbol{w}) + \beta\, R_{\text{emp}})\right)\, d\boldsymbol{w}$$

Posterior Approximation

approximate the posterior: Gaussian assumption

$$\tilde{R}(\boldsymbol{w}) \;=\; \frac{1}{2\,\sigma^2} R_{\text{emp}} \;+\; \frac{1}{2}\alpha\;\Omega(\boldsymbol{w}) \;=\; \frac{1}{2}\;\beta\;R_{\text{emp}} \;+\; \frac{1}{2}\;\alpha\;\Omega(\boldsymbol{w})$$

Taylor expansion of $R(\boldsymbol{w})$ around its minimum $\boldsymbol{w}_{\text{MAP}}$

$$\tilde{R}(\boldsymbol{w}) \;=\; \tilde{R}(\boldsymbol{w}_{\text{MAP}}) \;+\; \frac{1}{2}\;(\boldsymbol{w} \;-\; \boldsymbol{w}_{\text{MAP}})^T\,\boldsymbol{H}\;(\boldsymbol{w} \;-\; \boldsymbol{w}_{\text{MAP}})$$

where the first order derivatives vanish at the minimum
$\boldsymbol{H}$ is the Hessian

posterior is now a Gaussian

$$p(\boldsymbol{w} \mid \{\boldsymbol{z}\}) \;=\; \frac{1}{Z}\;\exp(-\;\tilde{R}(\boldsymbol{w})) \;=$$

$$\frac{1}{Z}\;\exp\left(-\;\tilde{R}(\boldsymbol{w}_{\text{MAP}}) \;-\; \frac{1}{2}\;(\boldsymbol{w} \;-\; \boldsymbol{w}_{\text{MAP}})^T\,\boldsymbol{H}\;(\boldsymbol{w} \;-\; \boldsymbol{w}_{\text{MAP}})\right)$$

where $\boldsymbol{Z}$ is normalization constant

Posterior Approximation

Hessian for weight decay

$$\boldsymbol{H} = \frac{1}{2\,\sigma^2}\,\boldsymbol{H}_{\text{emp}} + \alpha\,\boldsymbol{I} = \frac{\beta}{2}\,\boldsymbol{H}_{\text{emp}} + \alpha\,\boldsymbol{I}$$

where $\boldsymbol{H}_{\text{emp}}$ is the Hessian of the empirical error (the factor ½ vanishes for the weight decay term)

normalization constant:

$$Z(\alpha,\beta) = \exp\left(-\tilde{R}(\boldsymbol{w}_{\text{MAP}})\right)\,(2\,\pi)^{W/2}\,|\boldsymbol{H}|^{1/2}$$

Error Bars and Confidence Intervals

confidence intervals for model outputs
how reliable is the prediction

Output distribution $p(y \mid \boldsymbol{x}, \{\boldsymbol{z}\}) = \int_W p(y \mid \boldsymbol{x}, \boldsymbol{w})\ p(\boldsymbol{w} \mid \{\boldsymbol{z}\})\ d\boldsymbol{w}$

where we used the posterior distribution $p(\boldsymbol{w} \mid \{\boldsymbol{z}\})$
and a noise model $p(y \mid \boldsymbol{x}, \boldsymbol{w})$

Gaussian noise model (one dimension)

$$p(y \mid \boldsymbol{x}, \boldsymbol{w}) = \frac{1}{Z_R(\beta)} \exp\left(-\frac{\beta}{2}\ (g(\boldsymbol{x}; \boldsymbol{w}) - y)^2\right)$$

$$Z_R(\beta) = \left(\frac{2\,\pi}{\beta}\right)^{l/2} \qquad \beta = \frac{1}{\sigma^2}$$

Error Bars and Confidence Intervals

Linear approximation

$$g(\boldsymbol{x}; \boldsymbol{w}) = g(\boldsymbol{x}; \boldsymbol{w}_{\text{MAP}}) + \boldsymbol{g}^T (\boldsymbol{w} - \boldsymbol{w}_{\text{MAP}})$$

$$p(y \mid \boldsymbol{x}, \{\boldsymbol{z}\}) \propto \int_W \exp\left(- \underbrace{\frac{\beta}{2} \left(y - g(\boldsymbol{x}; \boldsymbol{w}_{\text{MAP}}) + \boldsymbol{g}^T (\boldsymbol{w} - \boldsymbol{w}_{\text{MAP}})\right)^2}_{p(y \mid \boldsymbol{x}, \boldsymbol{w})} - \underbrace{\frac{1}{2} (\boldsymbol{w} - \boldsymbol{w}_{\text{MAP}})^T \boldsymbol{H} (\boldsymbol{w} - \boldsymbol{w}_{\text{MAP}})}_{p(\boldsymbol{w} \mid \{\boldsymbol{z}\})} \right) d\boldsymbol{w}$$

Using the fact that the integrant is a Gaussian in $\boldsymbol{w}$, this gives

$$p(y \mid \boldsymbol{x}, \{\boldsymbol{z}\}) = \frac{1}{\sqrt{2\,\pi}\,\sigma_y} \exp\left(- \frac{1}{2\,\sigma_y^2} \left(y - g(\boldsymbol{x}; \boldsymbol{w}_{\text{MAP}})\right)^2 \right)$$

where $\sigma_y^2 = \frac{1}{\beta} + \boldsymbol{g}^T \boldsymbol{H}^{-1} \boldsymbol{g} = \sigma^2 + \boldsymbol{g}^T \boldsymbol{H}^{-1} \boldsymbol{g}$

inherent data noise σ^2 plus the approximation uncertainty $\boldsymbol{g}^T \boldsymbol{H}^{-1} \boldsymbol{g}$

Error Bars and Confidence Intervals

To derive the formula for this Gaussian, we use following identity:

$$\int_W \exp\left(-\frac{1}{2}\boldsymbol{w}^T\boldsymbol{A}\boldsymbol{w} + \boldsymbol{v}^T\boldsymbol{w}\right) d\boldsymbol{w} =$$

$$(2\pi)^{W/2}\,|\boldsymbol{A}|^{-1/2}\,\exp\left(\frac{1}{2}\boldsymbol{v}^T\boldsymbol{A}^{-1}\boldsymbol{v}\right)$$

After collecting terms in $(\boldsymbol{w} - \boldsymbol{w}_{\mathrm{MAP}})$ the quadratic part is

$$\boldsymbol{A} = \beta\,\boldsymbol{g}\,\boldsymbol{g}^T + \boldsymbol{H}$$

and the linear part is

$$\boldsymbol{v} = \beta\,(y - g(\boldsymbol{x};\boldsymbol{w}_{\mathrm{MAP}}))\,\boldsymbol{g}$$

and the constant term is

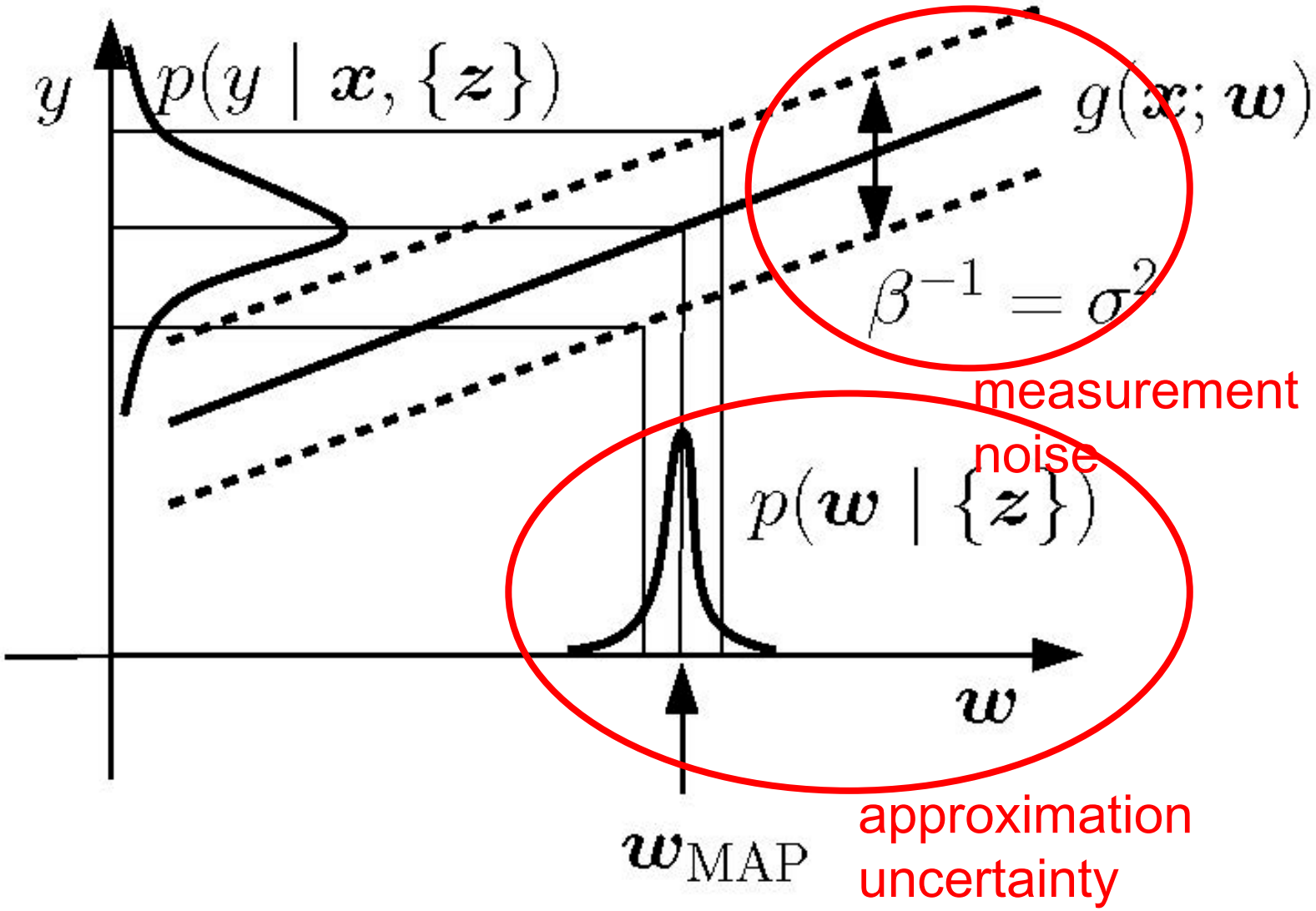
$$c = \beta\,(y - g(\boldsymbol{x};\boldsymbol{w}_{\mathrm{MAP}}))^2$$

We obtain in the exponent:

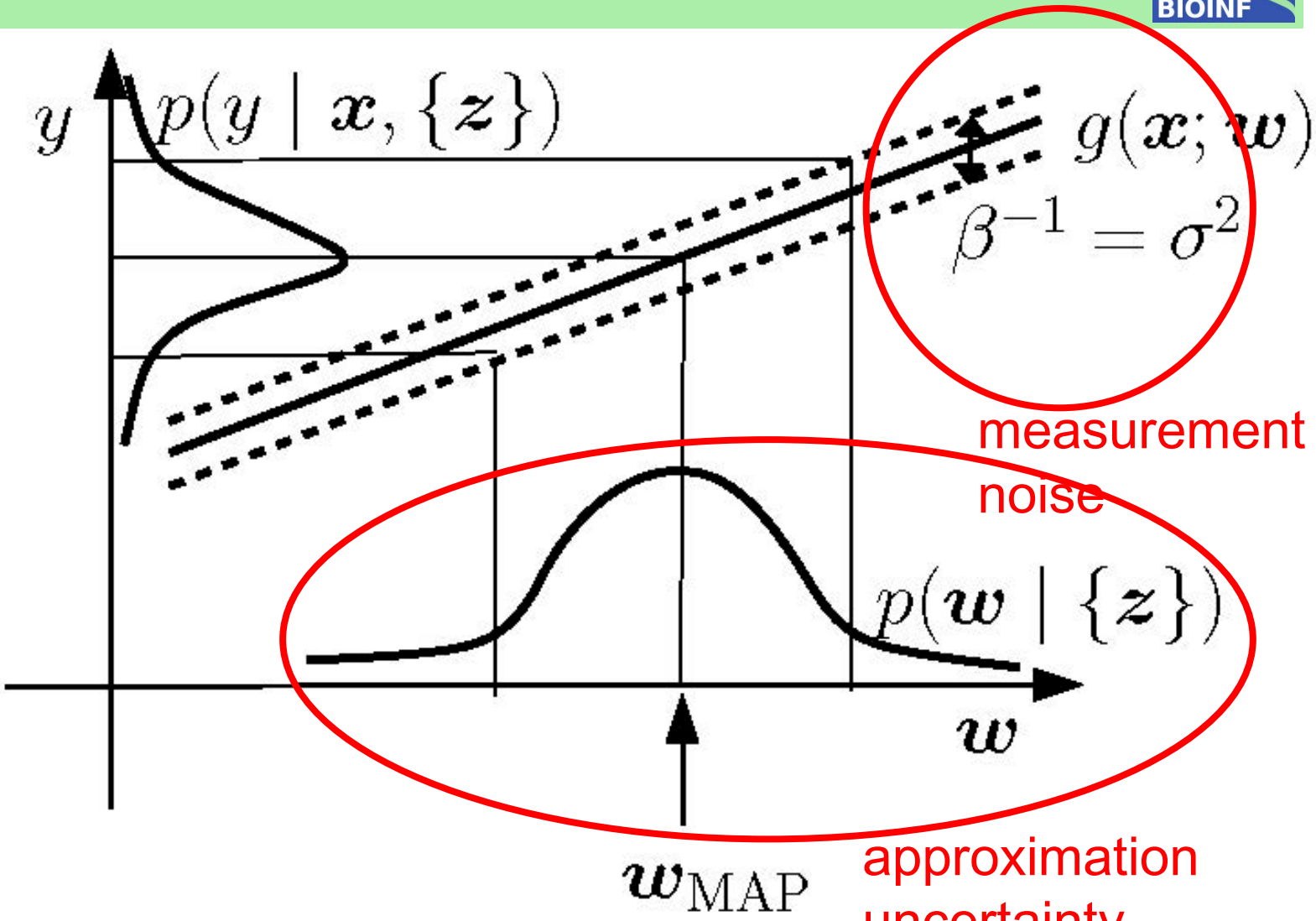
$$-\frac{1}{2}\boldsymbol{v}^T\boldsymbol{A}^{-1}\boldsymbol{v} + \frac{1}{2}c = \frac{1}{2}\,(y - g(\boldsymbol{x};\boldsymbol{w}_{\mathrm{MAP}}))^2\left(\beta - \beta^2\boldsymbol{g}^T\left(\boldsymbol{H} + \beta\boldsymbol{g}\boldsymbol{g}^T\right)^{-1}\boldsymbol{g}\right)$$

$$\left(\beta - \beta^2\boldsymbol{g}^T\left(\boldsymbol{H} + \beta\boldsymbol{g}\boldsymbol{g}^T\right)^{-1}\boldsymbol{g}\right)^{-1} = \frac{1}{\beta} + \boldsymbol{g}^T\boldsymbol{H}^{-1}\boldsymbol{g}$$

Error Bars and Confidence Intervals



Error Bars and Confidence Intervals



Hyper-parameter Selection: Evidence Framework

hyper-parameters α and β

β: assumption on the noise in the data

α: assumption on the optimal complexity relative to empirical error

integrating out α and β → marginalization

$$p(\boldsymbol{w} \mid \{\boldsymbol{z}\}) =$$

$$\int_{S_\alpha} \int_{S_\beta} p(\boldsymbol{w}, \alpha, \beta \mid \{\boldsymbol{z}\})\, d\alpha\, d\beta =$$

$$\int_{S_\alpha} \int_{S_\beta} p(\boldsymbol{w} \mid \alpha, \beta, \{\boldsymbol{z}\})\, p(\alpha, \beta \mid \{\boldsymbol{z}\})\, d\alpha\, d\beta$$

compute the integrals → later in next section

Hyper-parameter Selection: Evidence Framework

approximate the posterior

assume: sharply peaked around the maximal values α_{MAP} and β_{MAP}

high values of $p(\alpha, \beta \mid \{\boldsymbol{z}\})$ and constant $p(\boldsymbol{w} \mid \alpha, \beta, \{\boldsymbol{z}\})$ with value $p(\boldsymbol{w} \mid \alpha_{\mathrm{MAP}}, \beta_{\mathrm{MAP}}, \{\boldsymbol{z}\})$

$$p(\boldsymbol{w} \mid \{\boldsymbol{z}\}) = p(\boldsymbol{w} \mid \alpha_{\mathrm{MAP}}, \beta_{\mathrm{MAP}}, \{\boldsymbol{z}\})$$

$$\int_{S_\alpha} \int_{S_\beta} p(\alpha, \beta \mid \{\boldsymbol{z}\}) \, d\alpha \, d\beta =$$

$$p(\boldsymbol{w} \mid \alpha_{\mathrm{MAP}}, \beta_{\mathrm{MAP}}, \{\boldsymbol{z}\})$$

searching for the hyper-parameters which maximize the posterior

Hyper-parameter Selection: Evidence Framework

posterior of

$$p(\alpha, \beta \mid \{\boldsymbol{z}\}) = \frac{p(\{\boldsymbol{z}\} \mid \alpha, \beta)\ p(\alpha, \beta)}{p_{\alpha,\beta}(\{\boldsymbol{z}\})}$$

prior for α and β: $p(\alpha, \beta)$

non-informative priors: equal probability for all values

marginalization over $\boldsymbol{w}$:

$$p(\{\boldsymbol{z}\} \mid \alpha, \beta) = \int_W p(\{\boldsymbol{z}\} \mid \boldsymbol{w}, \alpha, \beta)\ p(\boldsymbol{w} \mid \alpha, \beta)\ d\boldsymbol{w} =$$

$$\int_W p(\{\boldsymbol{z}\} \mid \boldsymbol{w}, \beta)\ p(\boldsymbol{w} \mid \alpha)\ d\boldsymbol{w}$$

Def. conditional probability:

$$p(\{\boldsymbol{z}\} \mid \boldsymbol{w})\ p(\boldsymbol{w}) = p(\{\boldsymbol{z}\})\ p(\boldsymbol{w} \mid \{\boldsymbol{z}\})$$

$$p(\{\boldsymbol{z}\} \mid \alpha, \beta) = \frac{Z(\alpha, \beta)}{Z_w(\alpha)\ Z_R(\beta)}$$

$$p(\{\boldsymbol{z}\} \mid \boldsymbol{w}) = \frac{1}{Z_R(\beta)} \exp\left(-\frac{1}{2}\ \beta\ R_{\text{emp}}\right)$$

$$p(\boldsymbol{w}) = \frac{1}{Z_w(\alpha)} \exp\left(-\frac{1}{2}\ \alpha\ \Omega(\boldsymbol{w})\right)$$

$$p(\boldsymbol{w} \mid \{\boldsymbol{z}\}) = \frac{1}{Z(\alpha, \beta)} \exp\left(-\frac{1}{2}\ (\alpha\ \Omega(\boldsymbol{w}) + \beta\ R_{\text{emp}})\right)$$

$$p(\boldsymbol{w} \mid \{\boldsymbol{z}\}) = \frac{Z_R(\beta)\ Z_w(\alpha)}{Z(\alpha, \beta)}\ p(\{\boldsymbol{z}\} \mid \boldsymbol{w})\ p(\boldsymbol{w})$$

$$p(\{\boldsymbol{z}\} \mid \boldsymbol{w})\ p(\boldsymbol{w}) = \frac{Z(\alpha, \beta)}{Z_R(\beta)\ Z_w(\alpha)}\ p(\boldsymbol{w} \mid \{\boldsymbol{z}\})$$

Hyper-parameter Selection: Evidence Framework

Example

mean squared error and regularization by quadratic weight decay

$$Z(\alpha,\beta) = \exp\left(-\tilde{R}(\boldsymbol{w}_{\text{MAP}})\right) (2\pi)^{W/2} |\boldsymbol{H}|^{-1/2}$$

where $\tilde{R}(\boldsymbol{w}_{\text{MAP}}) = \frac{1}{2}\beta R_{\text{emp}} + \frac{1}{2}\alpha\,\Omega(\boldsymbol{w}_{\text{MAP}})$

$$Z_R(\beta) = \left(\frac{2\pi}{\beta}\right)^{l/2}$$

$$Z_w(\beta) = \left(\frac{2\pi}{\alpha}\right)^{W/2}$$

thus

$$\tilde{R}(\boldsymbol{w}) = \frac{1}{2}\beta R_{\text{emp}}(\boldsymbol{w}) + \frac{1}{2}\alpha\,\Omega(\boldsymbol{w})$$

$$\tilde{R}(\boldsymbol{w}) = \tilde{R}(\boldsymbol{w}_{\text{MAP}}) + (\boldsymbol{w}-\boldsymbol{w}_{\text{MAP}})^T \boldsymbol{H} (\boldsymbol{w}-\boldsymbol{w}_{\text{MAP}})$$

- first order derivative is zero
- $\exp(-R(\boldsymbol{w}))$ is a Gaussian with normalizing constant $Z(\alpha,\beta)$

$$\ln p(\{\boldsymbol{z}\} \mid \alpha,\beta) = -\frac{1}{2}\alpha\,\Omega(\boldsymbol{w}_{\text{MAP}}) - \frac{1}{2}\beta\,R_{\text{emp}} - \frac{1}{2}\ln|\boldsymbol{H}| + \frac{W}{2}\ln\alpha + \frac{l}{2}\ln\beta - \frac{l}{2}\ln(2\pi)$$

$(2\pi)^{W/2}$ cancels

$$\boldsymbol{H} = \frac{\beta}{2}\boldsymbol{H}_{\text{emp}} + \alpha\boldsymbol{I}$$

Hyper-parameter Selection: Evidence Framework

Assumption: eigenvalues λ_j of $1/2\ \boldsymbol{H}_{\text{emp}}$ do not depend on α

$$\frac{\partial}{\partial \alpha} \ln |\boldsymbol{H}| = \frac{\partial}{\partial \alpha} \ln \prod_{j=1}^{W} (\beta\ \lambda_j + \alpha) =$$

$$\frac{\partial}{\partial \alpha} \sum_{j=1}^{W} \ln (\beta\ \lambda_j + \alpha) = \sum_{j=1}^{W} \frac{1}{\beta\ \lambda_j + \alpha} = \text{Tr} \boldsymbol{H}^{-1}$$

Note, Hessian $\boldsymbol{H}$ was evaluated at $\boldsymbol{w}_{\text{MAP}}$ which depends on α
→ terms $\frac{\partial \lambda_j}{\partial \alpha}$ neglected

Hyper-parameter Selection: Evidence Framework

$$\frac{\partial}{\partial \alpha} \ln p(\{\boldsymbol{z}\} \mid \alpha, \beta) =$$

$$-\frac{1}{2}\,\Omega(\boldsymbol{w}_{\mathrm{MAP}}) - \frac{1}{2}\sum_{j=1}^{W}\frac{1}{\beta\,\lambda_j + \alpha} + \frac{1}{2}\,W\,\frac{1}{\alpha} = 0$$

which gives

$$\alpha\,\Omega(\boldsymbol{w}_{\mathrm{MAP}}) =$$

$$-\sum_{j=1}^{W}\frac{\alpha}{\beta\,\lambda_j + \alpha} + W = \sum_{j=1}^{W}\frac{\beta\,\lambda_j}{\beta\,\lambda_j + \alpha} = \gamma$$

$\Omega(\boldsymbol{w}_{\mathrm{MAP}})$ how far are weights pushed away from prior (zero) by data

term $\frac{\beta\,\lambda_j}{\beta\,\lambda_j + \alpha}$ is in [0;1]; 1 → data governed; 0 → prior driven

γ: effective number of weights which are driven by the data

Hyper-parameter Selection: Evidence Framework

Note, Hessian is not evaluated at the minimum of R_{emp} but at $\boldsymbol{w}_{\text{MAP}}$

eigenvalues λ_j are not guaranteed to be positive

→ terms $\frac{\beta\ \lambda_j}{\beta\ \lambda_j\ +\ \alpha}$ may be negative

derivative of the negative log-posterior with respect to β

$$\frac{\partial}{\partial \beta} \ln |\boldsymbol{H}| \ = \ \frac{\partial}{\partial \beta} \sum_{j=1}^{W} \ln (\beta\ \lambda_j\ +\ \alpha) \ =$$

$$\sum_{j=1}^{W} \frac{\lambda_j}{\beta\ \lambda_j\ +\ \alpha}$$

derivative set to zero: $\beta\ R_{\text{emp}} \ = \ l \ - \ \sum_{j=1}^{W} \frac{\lambda_j}{\beta\ \lambda_j\ +\ \alpha} \ = \ l \ - \ \gamma$

Hyper-parameter Selection: Evidence Framework

updates for the hyper-parameters:

$$\alpha^{\text{new}} = \frac{\gamma}{\Omega(\boldsymbol{w}_{\text{MAP}})}$$

$$\beta^{\text{new}} = \frac{l - \gamma}{R_{\text{emp}}(\boldsymbol{w}_{\text{MAP}})}$$

with these new hyper-parameters the new values of $\boldsymbol{w}_{\text{MAP}}$ can be estimated through gradient based methods

Iteratively again the hyper-parameters are updated and so forth

If all parameters are well defined: $\gamma = W$
If many training examples: $l >> W$

approximation for update formulae

$$\alpha^{\text{new}} = \frac{W}{\Omega(\boldsymbol{w}_{\text{MAP}})}$$

$$\beta^{\text{new}} = \frac{l}{R_{\text{emp}}(\boldsymbol{w}_{\text{MAP}})}$$

Hyper-parameter Selection: Integrate Out

Posterior: obtained by integrating out α and β

$$p(\boldsymbol{w} \mid \{\boldsymbol{z}\}) = \int_{S_\alpha} \int_{S_\beta} p(\boldsymbol{w}, \alpha, \beta \mid \{\boldsymbol{z}\}) \, d\alpha \, d\beta =$$

$$\int_{S_\alpha} \int_{S_\beta} p(\boldsymbol{w} \mid \alpha, \beta, \{\boldsymbol{z}\}) \, p(\alpha, \beta \mid \{\boldsymbol{z}\}) \, d\alpha \, d\beta =$$

$$\frac{1}{p_w(\{\boldsymbol{z}\})} \int_{S_\alpha} \int_{S_\beta} p(\{\boldsymbol{z}\} \mid \boldsymbol{w}, \beta) \, p(\boldsymbol{w} \mid \alpha) \, p(\alpha) \, p(\beta) \, d\alpha \, d\beta$$

where we used

$$p(\alpha, \beta \mid \{\boldsymbol{z}\}) = p(\alpha) \, p(\beta)$$

$$p(\boldsymbol{w} \mid \{\boldsymbol{z}\}) = \frac{p(\{\boldsymbol{z}\} \mid \boldsymbol{w}) \, p(\boldsymbol{w})}{p_{\boldsymbol{w}}(\{\boldsymbol{z}\})}$$

$$p(\{\boldsymbol{z}\} \mid \boldsymbol{w}, \alpha, \beta) = p(\{\boldsymbol{z}\} \mid \boldsymbol{w}, \beta)$$

$$p(\boldsymbol{w} \mid \alpha, \beta) = p(\boldsymbol{w} \mid \alpha)$$

Hyper-parameter Selection: Integrate Out

α and β are scaling parameters

- output range: β
- weights: α

Non-informative (uniform) on a logarithmic scale, therefore $p(\ln(\alpha))$ and $p(\ln(\beta))$ are constant

Using $p_x(\boldsymbol{x}) = p_g(g(\boldsymbol{x})) \left|\frac{\partial g}{\partial x}\right|$ gives

$$p(\alpha) = \frac{1}{\alpha}$$

$$p(\beta) = \frac{1}{\beta}$$

Hyper-parameter Selection: Integrate Out

prior over the weights

$$p(\boldsymbol{w}) = \int_0^\infty p(\boldsymbol{w} \mid \alpha)\, p(\alpha)\, d\alpha =$$

$$\int_0^\infty \frac{1}{Z_w(\alpha)} \exp(-\frac{1}{2}\, \alpha\, \Omega(\boldsymbol{w}))\, \frac{1}{\alpha}\, d\alpha =$$

$$(2\,\pi)^{-W/2} \int_0^\infty \exp(-\frac{1}{2}\, \alpha\, \Omega(\boldsymbol{w}))\, \alpha^{W/2-1}\, d\alpha = \frac{\Gamma(W/2)}{(2\,\pi\,\Omega(\boldsymbol{w}))^{W/2}}$$

where Γ is the gamma function

Analog $p(\{\boldsymbol{z}\} \mid \boldsymbol{w}) = \frac{\Gamma(l/2)}{(2\,\pi\,R_{\text{emp}})^{l/2}}$

Hyper-parameter Selection: Integrate Out

Bayes formula → negative log-posterior:

$$-\log p(\boldsymbol{w} \mid \{\boldsymbol{z}\}) = -\log p(\{\boldsymbol{z}\} \mid \boldsymbol{w}) - \log p(\boldsymbol{w}) + \log p_{\boldsymbol{w}}(\{\boldsymbol{z}\})$$

$$-\ln p(\boldsymbol{w} \mid \{\boldsymbol{z}\}) = \frac{l}{2} \ln R_{\text{emp}} + \frac{W}{2} \ln \Omega(\boldsymbol{w}) + \text{const}$$

compare with $-\ln p(\boldsymbol{w} \mid \{\boldsymbol{z}\}) = \frac{1}{2} \beta R_{\text{emp}} + \frac{1}{2} \alpha \Omega(\boldsymbol{w})$

Compute gradients with respect to parameters!
Comparing the coefficients on front of the gradients gives:

$$\alpha^{\text{new}} = \frac{W}{\Omega(\boldsymbol{w}_{\text{MAP}})}$$

$$\beta^{\text{new}} = \frac{l}{R_{\text{emp}}(\boldsymbol{w}_{\text{MAP}})}$$

iterative algorithm

$\boldsymbol{w}_{\text{MAP}}$ gradient descent

Model Comparison

compare model classes $\mathcal{M}$

Bayes formula $p(\mathcal{M} \mid \{\boldsymbol{z}\}) = \dfrac{p(\{\boldsymbol{z}\} \mid \mathcal{M})\ p(\mathcal{M})}{p_{\mathcal{M}}(\{\boldsymbol{z}\})}$

evidence for model class

$$p(\{\boldsymbol{z}\} \mid \mathcal{M}) = \int_W p(\{\boldsymbol{z}\} \mid \boldsymbol{w}, \mathcal{M})\ p(\boldsymbol{w} \mid \mathcal{M})\ d\boldsymbol{w}$$

Approximate posterior by a box around MAP value

$$p(\{\boldsymbol{z}\} \mid \mathcal{M}) \approx p(\{\boldsymbol{z}\} \mid \boldsymbol{w}_{\mathrm{MAP}}, \mathcal{M})\ p(\boldsymbol{w}_{\mathrm{MAP}} \mid \mathcal{M})\ \Delta\boldsymbol{w}_{\mathrm{MAP}}$$

Gaussian: $\Delta\boldsymbol{w}_{\mathrm{MAP}}$ estimated from Hessian $\boldsymbol{H}$

Model Comparison

$$\ln p(\{\boldsymbol{z}\} \mid \alpha, \beta) = -\frac{\alpha}{2}\,\Omega(\boldsymbol{w}_{\mathrm{MAP}}) - \frac{\beta}{2}\,R_{\mathrm{emp}} - \frac{1}{2}\ln|\boldsymbol{H}| + \frac{W}{2}\ln\alpha + \frac{l}{2}\ln\beta - \frac{l}{2}\ln(2\pi)$$

A more exact term (not derived) is

$$\ln p(\{\boldsymbol{z}\} \mid \mathcal{M}) = -\frac{\alpha_{\mathrm{MAP}}}{2}\,\Omega(\boldsymbol{w}_{\mathrm{MAP}}) - \frac{\beta_{\mathrm{MAP}}}{2}\,R_{\mathrm{emp}} - \frac{1}{2}\ln|\boldsymbol{H}| + \frac{W}{2}\ln\alpha_{\mathrm{MAP}} + \frac{l}{2}\ln\beta_{\mathrm{MAP}} + \ln M! + 2\ln M + \frac{1}{2}\ln\left(\frac{2}{\gamma}\right) + \frac{1}{2}\ln\left(\frac{2}{l-\gamma}\right)$$

M: number of hidden units in the network

→ posterior is only locally approximated, thus equivalent solutions in weights space exist

→ permutation invariant solutions

Posterior Sampling

Approximate integrals like $A(f) = \int_W f(\boldsymbol{w})\ p(\boldsymbol{w} \mid \{\boldsymbol{z}\})\ d\boldsymbol{w}$

sample weight vectors $\boldsymbol{w}_i$ according to $p(\boldsymbol{w} \mid \{\boldsymbol{z}\})$

$$A(f) \approx \frac{1}{L} \sum_{i=1}^{L} f(\boldsymbol{w}_i)$$

we cannot easily sample from $p(\boldsymbol{w} \mid \{\boldsymbol{z}\})$

simpler distribution $q(\boldsymbol{w})$ where we can sample from

$$A(f) = \int_W f(\boldsymbol{w}) \frac{p(\boldsymbol{w} \mid \{\boldsymbol{z}\})}{q(\boldsymbol{w})}\ q(\boldsymbol{w})\ d\boldsymbol{w}$$

$$A(f) \approx \frac{1}{L} \sum_{i=1}^{L} f(\boldsymbol{w}_i) \frac{p(\boldsymbol{w}_i \mid \{\boldsymbol{z}\})}{q(\boldsymbol{w}_i)}$$

$\boldsymbol{w}_i$ are sampled according to $q(\boldsymbol{w})$

Posterior Sampling

avoid the normalization of $p(\boldsymbol{w} \mid \{\boldsymbol{z}\})$

$$A(f) \approx \frac{\sum_{i=1}^{L} f(\boldsymbol{w}_i)\ \tilde{p}(\boldsymbol{w}_i \mid \{\boldsymbol{z}\})\ /\ q(\boldsymbol{w}_i)}{\sum_{i=1}^{L} \tilde{p}(\boldsymbol{w}_i \mid \{\boldsymbol{z}\})\ /\ q(\boldsymbol{w}_i)}$$

$\tilde{p}(\boldsymbol{w} \mid \{\boldsymbol{z}\})$: unnormalized posterior → product likelihood and prior

importance sampling

sample in regions with large probability mass
→ **Markov Chain Monte Carlo**

Posterior Sampling

Metropolis algorithm: random walk in a way that regions with large probability mass are sampled

$$\boldsymbol{w}^{\text{new}} = \boldsymbol{w}^{\text{candidate}} \text{ with probability}$$

$$\begin{cases} 1 & \text{if } p(\boldsymbol{w}^{\text{candidate}} \mid \{\boldsymbol{z}\}) > p(\boldsymbol{w}^{\text{old}} \mid \{\boldsymbol{z}\}) \\ \frac{p(\boldsymbol{w}^{\text{candidate}} \mid \{\boldsymbol{z}\})}{p(\boldsymbol{w}^{\text{old}} \mid \{\boldsymbol{z}\})} & \text{if } p(\boldsymbol{w}^{\text{candidate}} \mid \{\boldsymbol{z}\}) < p(\boldsymbol{w}^{\text{old}} \mid \{\boldsymbol{z}\}) \end{cases}$$

simulated annealing: can be used to estimate the expectation