13

Generalization

13

Generalization

A cat that once sat on a hot stove
will never again sit on a hot stove
or on a cold one either.

Mark Twain

Generalization

- The network input-output mapping is accurate for the training data and for test data never seen before.

- The network interpolates well.

Cause of Overfitting

Poor generalization is caused by using a network that is too complex (too many neurons/parameters). To have the best performance we need to find the least complex network that can represent the data (Ockham's Razor).

Ockham's Razor

Find the simplest model that explains the data.

Problem Statement

Training Set

$$\{\mathbf{p}_1, \mathbf{t}_1\}\ ,\ \{\mathbf{p}_2, \mathbf{t}_2\}\ ,\ \dots\ ,\ \{\mathbf{p}_Q, \mathbf{t}_Q\}$$

Underlying Function

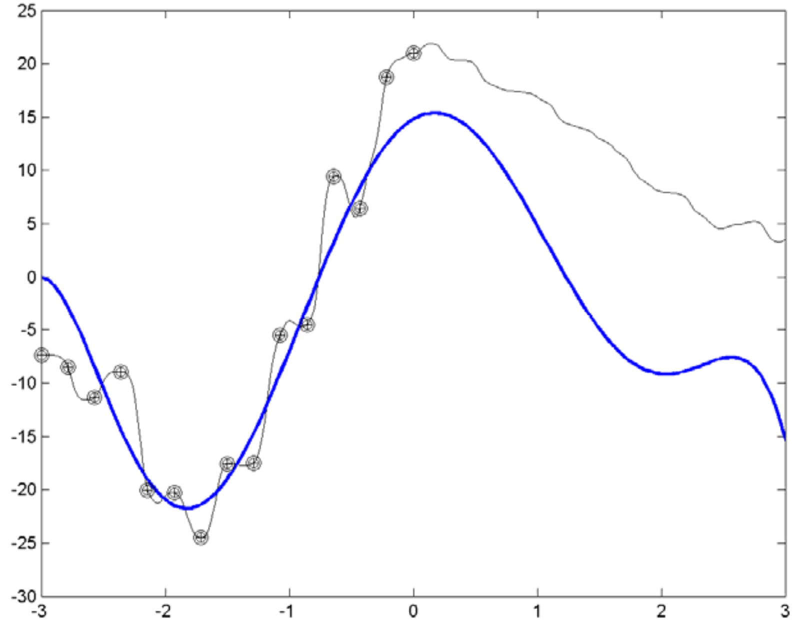
$$\mathbf{t}_q = \mathbf{g}(\mathbf{p}_q) + \varepsilon_q$$

Performance Function

$$F(\mathbf{x}) = E_D = \sum_{q=1}^{Q} (\mathbf{t}_q - \mathbf{a}_q)^T (\mathbf{t}_q - \mathbf{a}_q)$$

Poor Generalization

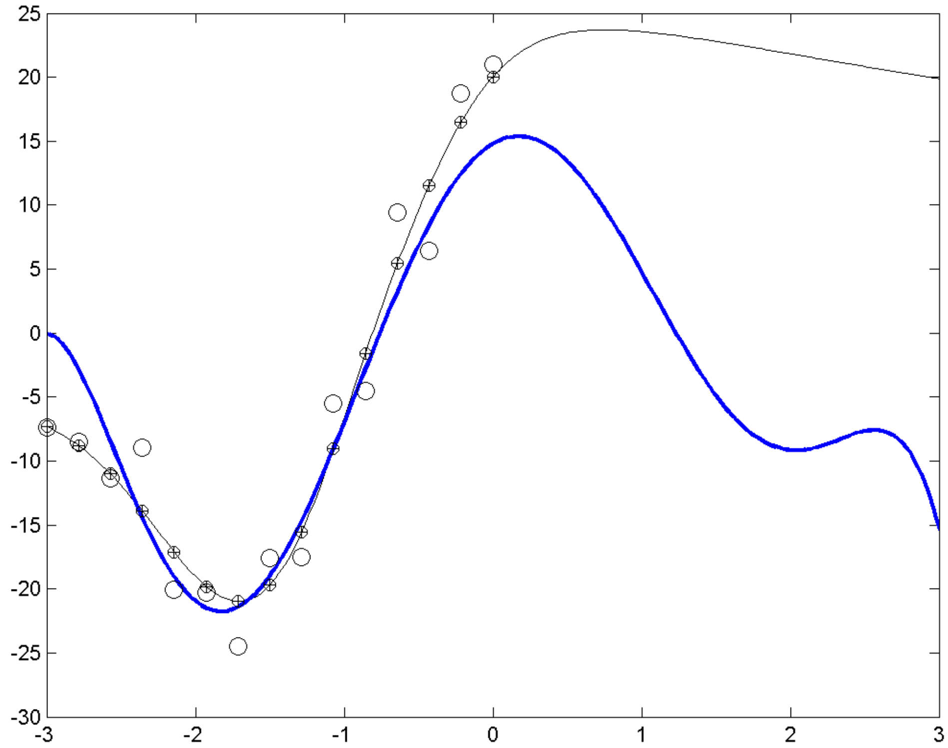
Overfitting Extrapolation



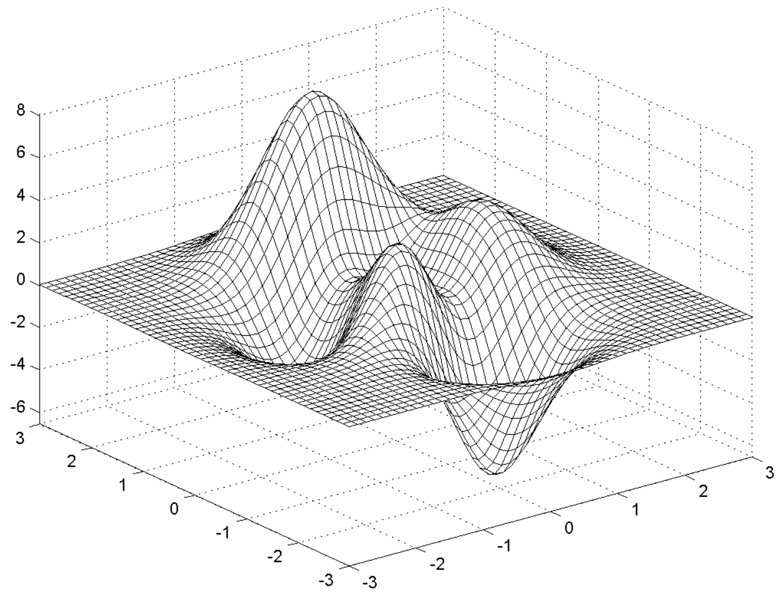
Interpolation

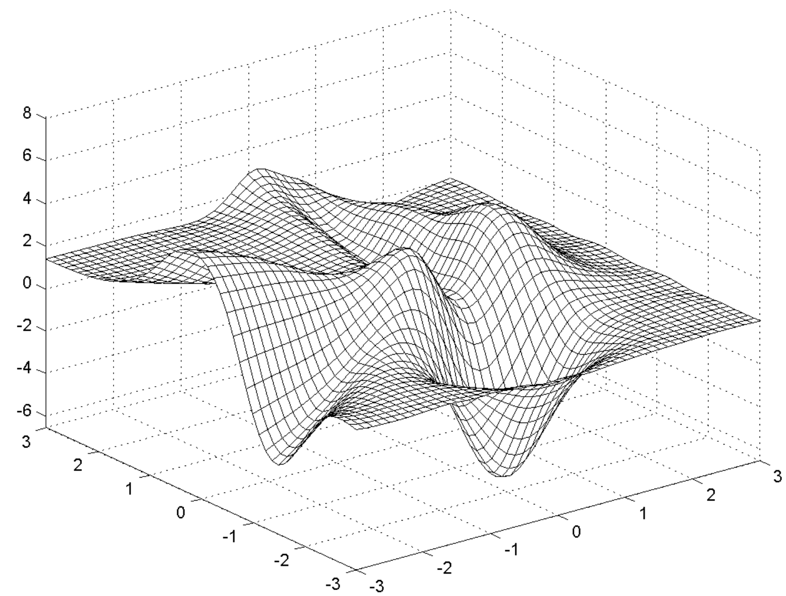
Good Generalization

Interpolation Extrapolation



Extrapolation in 2-D





Measuring Generalization

Test Set

- Part of the available data is set aside during the training process.
- After training, the network error on the test set is used as a measure of generalization ability.
- The test set must never be used in any way to train the network, or even to select one network from a group of candidate networks.
- The test set must be representative of all situations for which the network will be used.

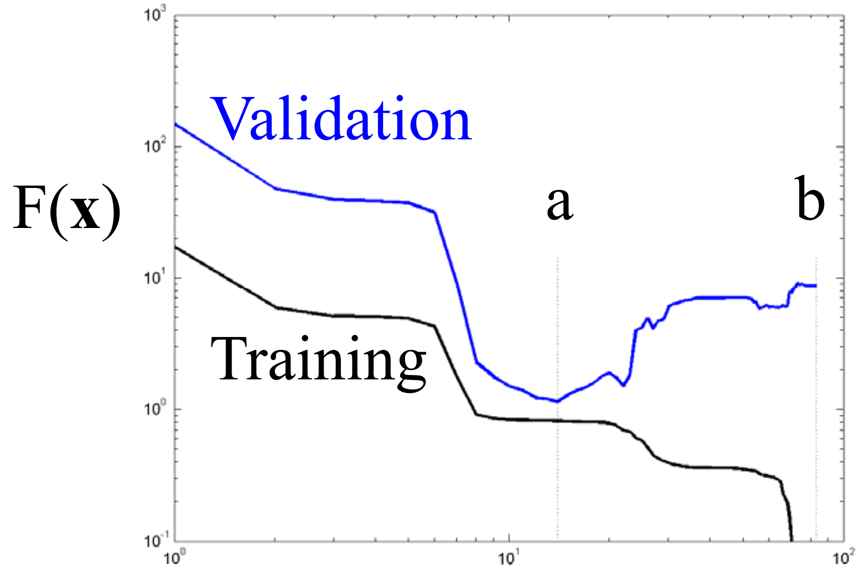
Methods for Improving Generalization

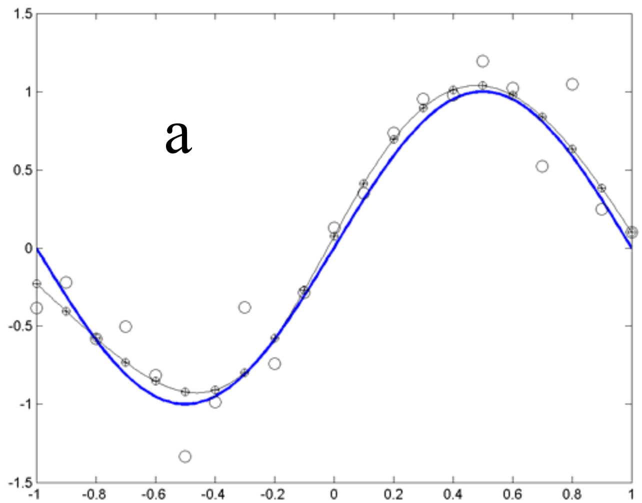
- Pruning (removing neurons) until the performance is degraded.
- Growing (adding neurons) until the performance is adequate.
- Validation Methods
- Regularization

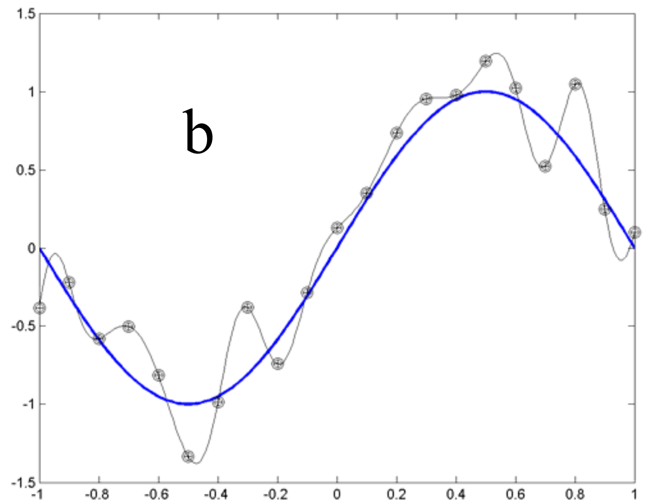
Early Stopping

- Break up data into training, *validation*, and test sets.
- Use only the training set to compute gradients and determine weight updates.
- Compute the performance on the validation set at each iteration of training.
- Stop training when the performance on the validation set goes up for a specified number of iterations.
- Use the weights which achieved the lowest error on the validation set.

Early Stopping Example







Regularization

Standard Performance Measure

$$F = E_D$$

Performance Measure with Regularization

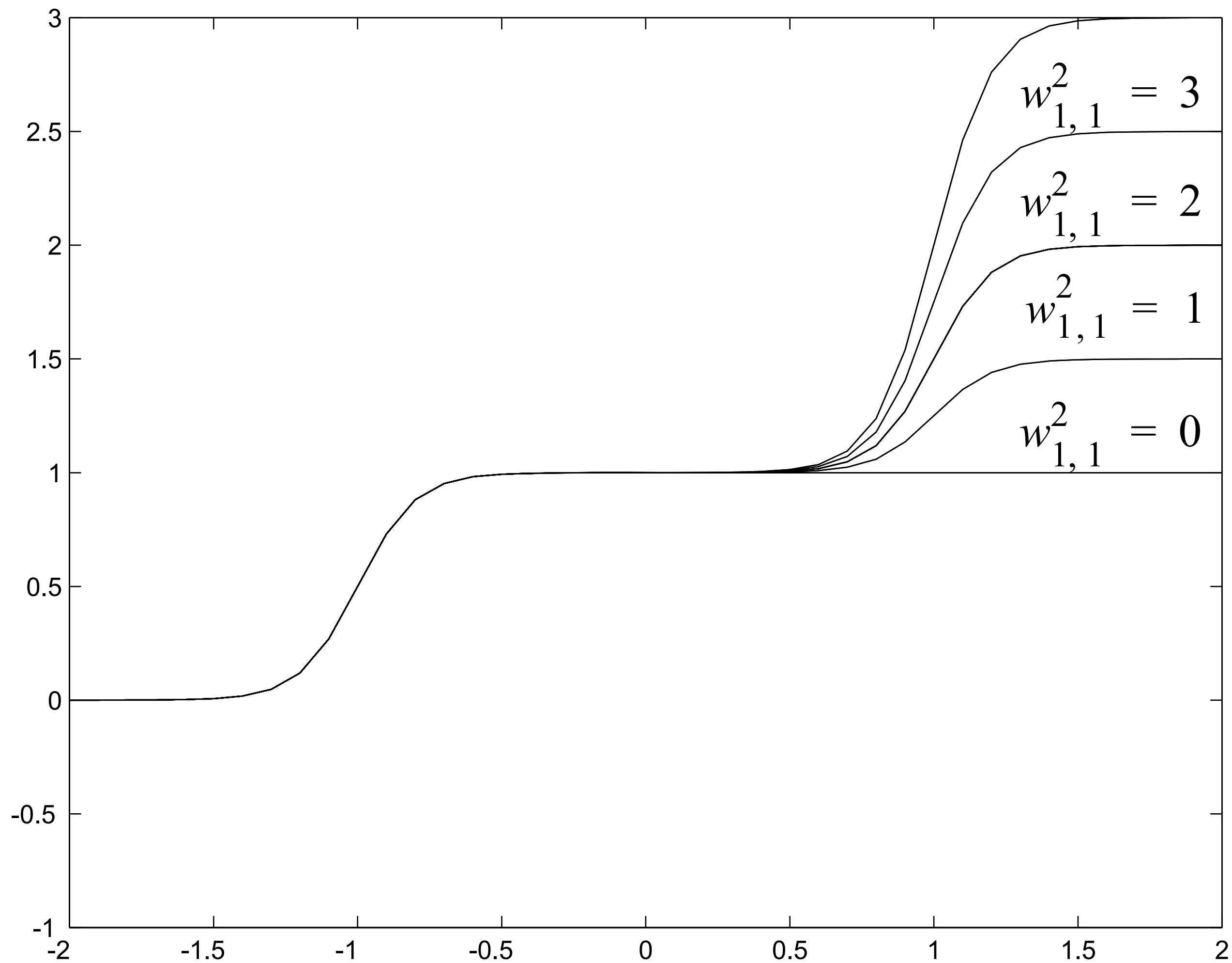
$$F = \beta E_D + \alpha E_W = \beta \sum_{q=1}^{Q} (\mathbf{t}_q - \mathbf{a}_q)^T (\mathbf{t}_q - \mathbf{a}_q) + \alpha \sum_{i=1}^{n} x_i^2$$

Complexity Penalty

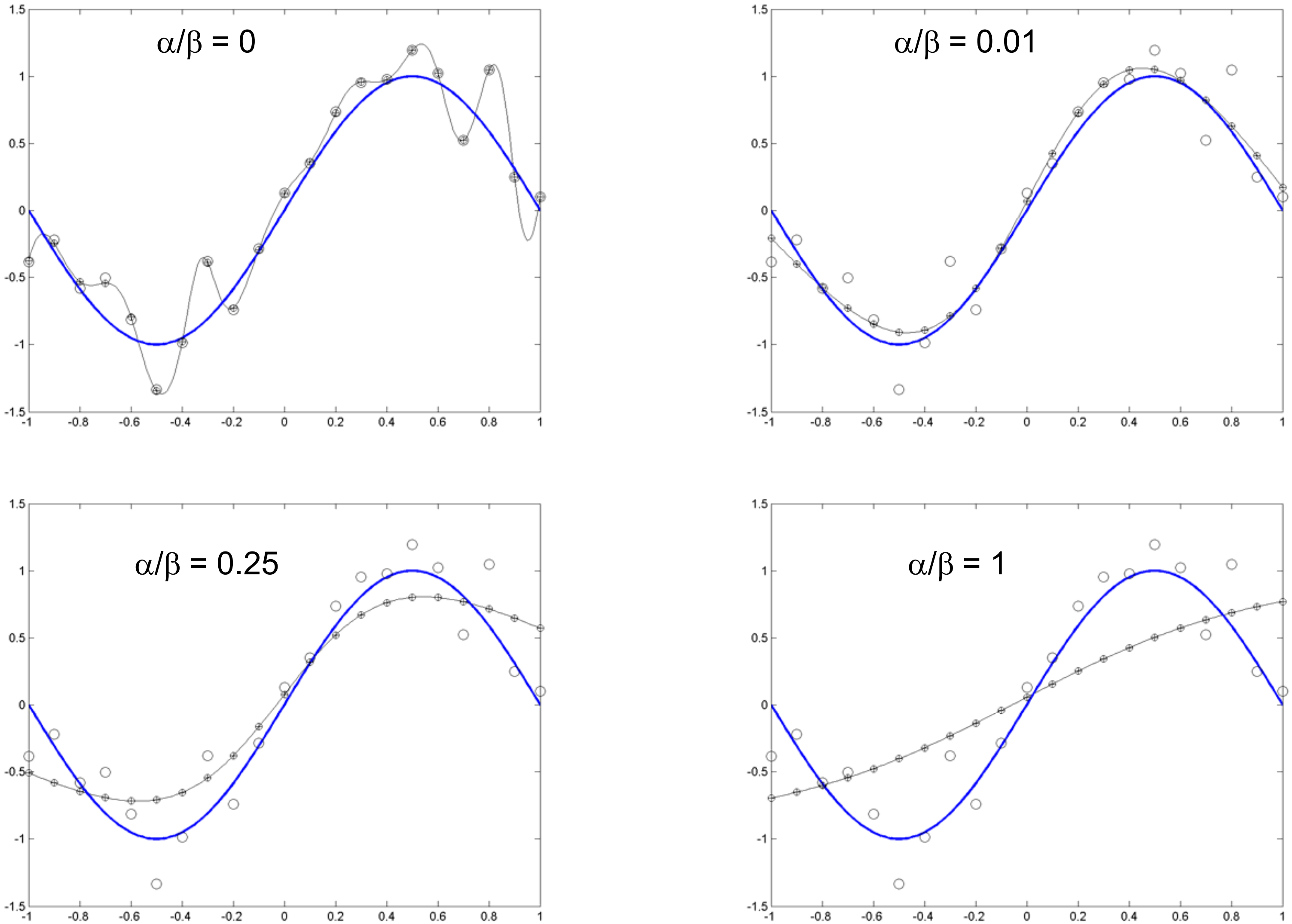
(Smaller weights means a smoother function.)

13

Effect of Weight Changes



Effect of Regularization



13

Bayes' Rule

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

$P(A)$ – Prior Probability. What we know about A before B is known.

$P(A|B)$ – Posterior Probability. What we know about A after we know the outcome of B.

$P(B|A)$ – Conditional Probability (Likelihood Function). Describes our knowledge of the system.

$P(B)$ – Marginal Probability. A normalization factor.

13

Example Problem

- 1% of the population have a certain disease.
- A test for the disease is 80% accurate in detecting the disease in people who have it.
- 10% of the time the test yields a false positive.
- If you have a positive test, what is your probability of having the disease?

Bayesian Analysis

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

A – Event that you have the disease.

B – Event that you have a positive test.

$P(A) = 0.01$

$P(B|A) = 0.8$

$P(B) = P(B|A)P(A) + P(B|\sim A)P(\sim A) = 0.8\ 0.01 + 0.1\ 0.99 = 0.107$

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} = \frac{0.8 \times 0.01}{0.107} = 0.0748$$

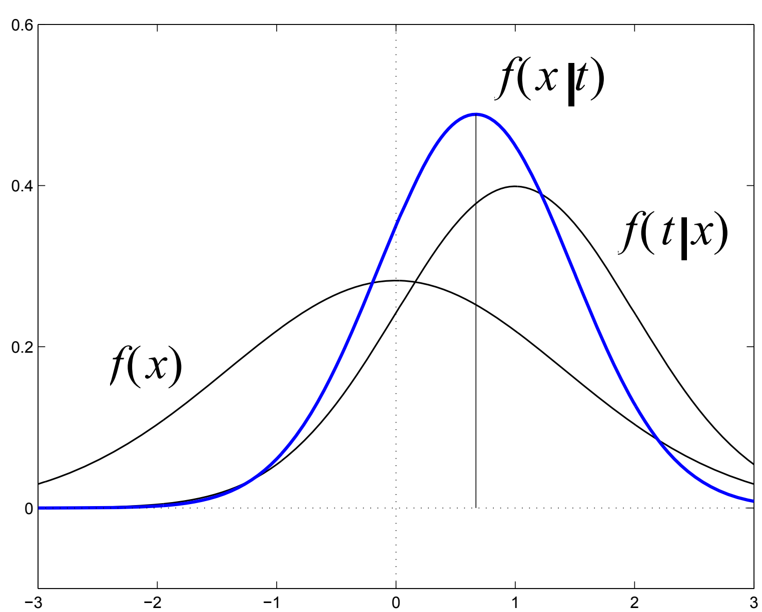
Signal Plus Noise Example

13

$$t = x + \varepsilon$$

$$f(\varepsilon) = \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{\varepsilon^2}{2\sigma^2}\right) \qquad f(x) = \frac{1}{\sqrt{2\pi}\sigma_x} exp\left(-\frac{x^2}{2\sigma_x^2}\right)$$

$$f(t|x) = \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{(t-x)^2}{2\sigma^2}\right) \qquad f(x|t) = \frac{f(t|x)f(x)}{f(t)}$$



13

NN Bayesian Framework

(MacKay 92)

MP — Posterior

ML — Likelihood

Prior

$$P(\mathbf{x} \mid D, \alpha, \beta, M) = \frac{P(D \mid \mathbf{x}, \beta, M)\, P(\mathbf{x} \mid \alpha, M)}{P(D \mid \alpha, \beta, M)}$$

Normalization (Evidence)

D - Data Set

M - Neural Network Model

$\mathbf{x}$ - Vector of Network Weights

Gaussian Assumptions

Gaussian Noise

$$P(D|\mathbf{x}, \beta, M) = \frac{1}{Z_D(\beta)} \exp(-\beta E_D) \qquad Z_D(\beta) = (2\pi\sigma_\varepsilon^2)^{N/2} = (\pi/\beta)^{N/2}$$

Gaussian Prior:

$$P(\mathbf{x}|\alpha, M) = \frac{1}{Z_W(\alpha)} \exp(-\alpha E_W) \qquad Z_W(\alpha) = (2\pi\sigma_w^2)^{n/2} = (\pi/\alpha)^{n/2}$$

$$P(\mathbf{x}|D, \alpha, \beta, M) = \frac{\frac{1}{Z_W(\alpha)}\frac{1}{Z_D(\beta)} \exp(-(\beta E_D + \alpha E_W))}{\text{Normalization Factor}} = \frac{1}{Z_F(\alpha, \beta)} \exp(-F(\mathbf{x}))$$

$$F = \beta E_D + \alpha E_W$$

Minimize F to Maximize P.

13 Optimizing Regularization Parameters

Evidence from First Level

Second Level of Inference

$$P(\alpha, \beta | D, M) = \frac{\overbrace{P(D|\alpha, \beta, M)}^{\text{Evidence from First Level}} P(\alpha, \beta | M)}{P(D|M)}$$

Evidence:

$$P(D|\alpha, \beta, M) = \frac{P(D|\mathbf{x}, \beta, M) P(\mathbf{x}|\alpha, M)}{P(\mathbf{x}|D, \alpha, \beta, M)}$$

$$= \frac{\left[\frac{1}{Z_D(\beta)} \exp(-\beta E_D)\right]\left[\frac{1}{Z_W(\alpha)} \exp(-\alpha E_W)\right]}{\frac{1}{Z_F(\alpha, \beta)} \exp(-F(\mathbf{x}))}$$

$$= \frac{Z_F(\alpha, \beta)}{Z_D(\beta) Z_W(\alpha)} \cdot \frac{\exp(-\beta E_D - \alpha E_W)}{\exp(-F(\mathbf{x}))} = \frac{Z_F(\alpha, \beta)}{Z_D(\beta) Z_W(\alpha)}$$

$Z_F(\alpha, \beta)$ is the only unknown in this expression.

Quadratic Approximation

Taylor series expansion:

$$F(\mathbf{x}) \approx F(\mathbf{x}^{MP}) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^{MP})^T \mathbf{H}^{MP} (\mathbf{x} - \mathbf{x}^{MP}) \qquad \mathbf{H} = \beta \nabla^2 E_D + \alpha \nabla^2 E_W$$

Substituting into previous posterior density function:

$$P(\mathbf{x} | D, \alpha, \beta, M) \approx \frac{1}{Z_F} \exp\left[-F(\mathbf{x}^{MP}) - \frac{1}{2}(\mathbf{x} - \mathbf{x}^{MP})^T \mathbf{H}^{MP} (\mathbf{x} - \mathbf{x}^{MP})\right]$$

$$P(\mathbf{x} | D, \alpha, \beta, M) \approx \left\{\frac{1}{Z_F} \exp(-F(\mathbf{x}^{MP}))\right\} \exp\left[-\frac{1}{2}(\mathbf{x} - \mathbf{x}^{MP})^T \mathbf{H}^{MP} (\mathbf{x} - \mathbf{x}^{MP})\right]$$

Equate with standard Gaussian density:

$$P(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^n \left|(\mathbf{H}^{MP})^{-1}\right|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}^{MP})^T \mathbf{H}^{MP} (\mathbf{x} - \mathbf{x}^{MP})\right)$$

Comparing to previous equation, we have:

$$Z_F(\alpha, \beta) \approx (2\pi)^{n/2} (\det((\mathbf{H}^{MP})^{-1}))^{1/2} \exp(-F(\mathbf{x}^{MP}))$$

13 Optimum Parameters

If we make this substitution for Z_F in the expression for the evidence and then take the derivative with respect to α and β to locate the minimum we find:

$$\alpha^{MP} = \frac{\gamma}{2E_W(\mathbf{x}^{MP})} \qquad \beta^{MP} = \frac{N-\gamma}{2E_D(\mathbf{x}^{MP})}$$

Effective Number of Parameters

$$\gamma = n - 2\alpha^{MP}\mathrm{tr}(\mathbf{H}^{MP})^{-1}$$

13 Gauss-Newton Approximation

It can be expensive to compute the Hessian matrix.

Try the Gauss-Newton Approximation.

$$\mathbf{H} = \nabla^2 F(\mathbf{x}) \approx 2\beta \mathbf{J}^T \mathbf{J} + 2\alpha \mathbf{I}_n$$

This is readily available if the Levenberg-Marquardt algorithm is used for training.

Algorithm (GNBR)

0. Initialize α, β and the weights.

1. Take one step of Levenberg-Marquardt to minimize $F(\mathbf{w})$.

2. Compute the effective number of parameters $\gamma = n - 2\alpha \mathrm{tr}(\mathbf{H}^{-1})$, using the Gauss-Newton approximation for $\mathbf{H}$.

3. Compute new estimates of the regularization parameters $\alpha = \gamma/(2E_W)$ and $\beta = (N-\gamma)/(2E_D)$.

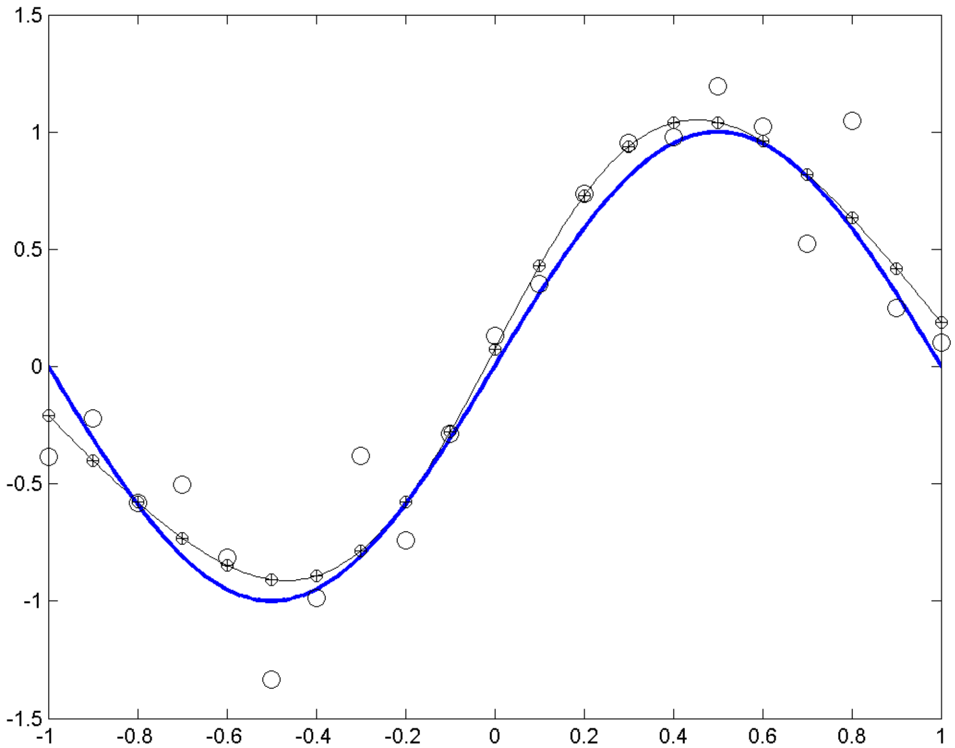
4. Iterate steps 1-3 until convergence.

Checks of Performance

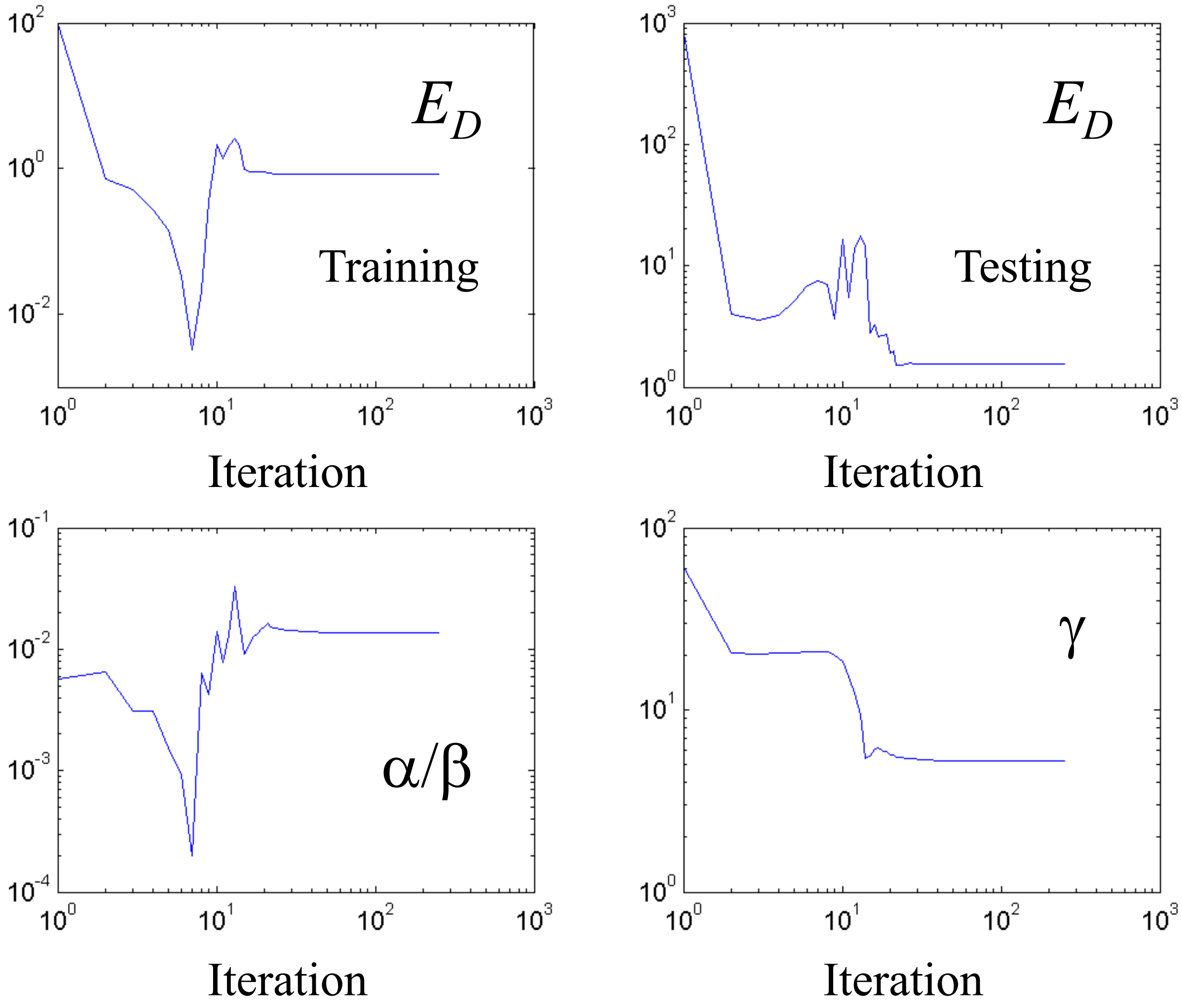
- If γ is very close to n, then the network may be too small. Add more hidden layer neurons and retrain.
- If the larger network has the same final γ, then the smaller network was large enough.
- Otherwise, increase the number of hidden neurons.
- If a network is sufficiently large, then a larger network will achieve comparable values for γ, E_D and E_W.

GNBR Example

$\alpha/\beta = 0.0137$



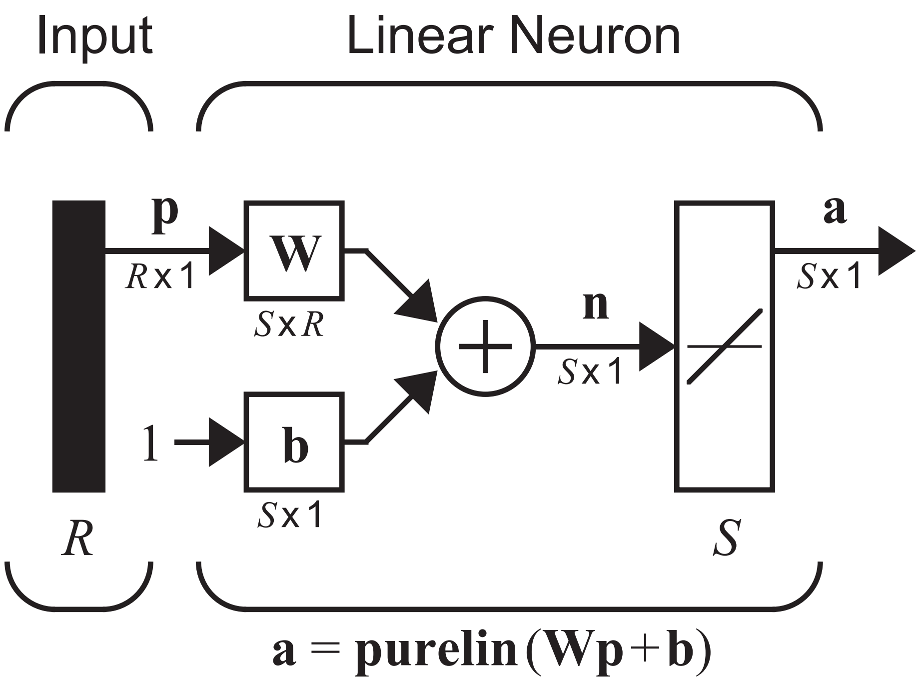
Convergence of GNBR



13

Relationship between Early Stopping and Regularization

Linear Network



$$\mathbf{a} = \mathbf{purelin}(\mathbf{W}\mathbf{p} + \mathbf{b}) = \mathbf{W}\mathbf{p} + \mathbf{b}$$

$$a_i = purelin(n_i) = purelin({}_i\mathbf{w}^T\mathbf{p} + b_i) = {}_i\mathbf{w}^T\mathbf{p} + b_i$$

$${}_i\mathbf{w} = \begin{bmatrix} w_{i,1} \\ w_{i,2} \\ \vdots \\ w_{i,R} \end{bmatrix}$$

Performance Index

Training Set:

$$\{\mathbf{p}_1, \mathbf{t}_1\}, \{\mathbf{p}_2, \mathbf{t}_2\}, \ldots, \{\mathbf{p}_Q, \mathbf{t}_Q\}$$

Input: $\mathbf{p}_q$ Target: $\mathbf{t}_q$

Notation:

$$\mathbf{x} = \begin{bmatrix} {}_1\mathbf{w} \\ b \end{bmatrix} \qquad \mathbf{z} = \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} \qquad a = {}_1\mathbf{w}^T\mathbf{p} + b \quad \Rightarrow \quad a = \mathbf{x}^T\mathbf{z}$$

Mean Square Error:

$$F(\mathbf{x}) = E[e^2] = E[(t - a)^2] = E[(t - \mathbf{x}^T\mathbf{z})^2] = E_D$$

Error Analysis

$$F(\mathbf{x}) = E[e^2] = E[(t-a)^2] = E[(t-\mathbf{x}^T\mathbf{z})^2]$$

$$F(\mathbf{x}) = E[t^2 - 2t\mathbf{x}^T\mathbf{z} + \mathbf{x}^T\mathbf{z}\mathbf{z}^T\mathbf{x}]$$

$$F(\mathbf{x}) = E[t^2] - 2\mathbf{x}^T\mathrm{E}[t\mathbf{z}] + \mathbf{x}^T E[\mathbf{z}\mathbf{z}^T]\mathbf{x}$$

$$\boxed{F(\mathbf{x}) = c - 2\mathbf{x}^T\mathbf{h} + \mathbf{x}^T\mathbf{R}\mathbf{x}}$$

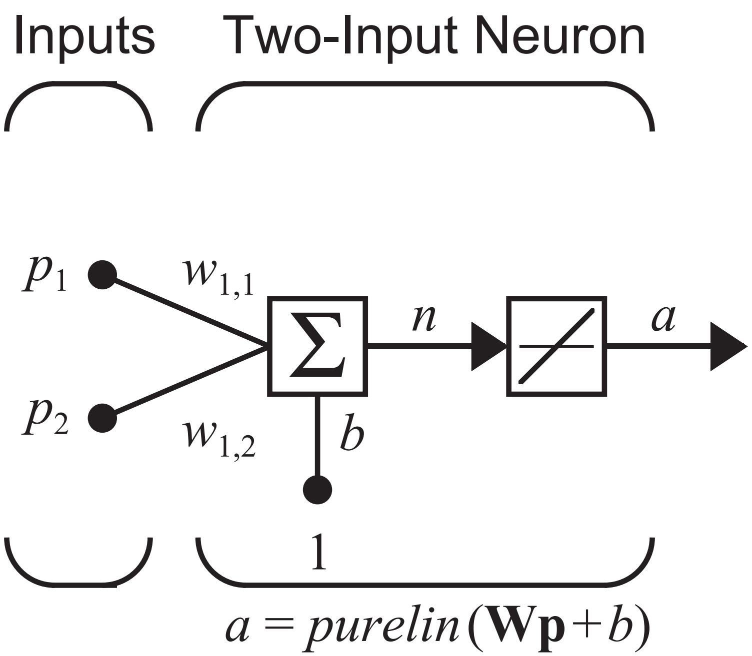
$$c = E[t^2] \qquad \mathbf{h} = E[t\mathbf{z}] \qquad \mathbf{R} = E[\mathbf{z}\mathbf{z}^T]$$

The mean square error for the Linear Network is a quadratic function:

$$F(\mathbf{x}) = c + \mathbf{d}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x}$$

$$\mathbf{d} = -2\mathbf{h} \qquad \mathbf{A} = 2\mathbf{R}$$

Example



$$\left\{\mathbf{p}_1 = \begin{bmatrix}1\\1\end{bmatrix}, t_1 = 1\right\}$$ (Probability = 0.75)

$$\left\{\mathbf{p}_2 = \begin{bmatrix}-1\\1\end{bmatrix}, t_2 = -1\right\}$$ (Probability = 0.25)

$$F(\mathbf{x}) = c - 2\mathbf{x}^T\mathbf{h} + \mathbf{x}^T\mathbf{R}\mathbf{x} = E_D$$

$$c = E[t^2] = (1)^2(0.75) + (-1)^2(0.25) = 1$$

$$\mathbf{h} = E[t\mathbf{z}] = (0.75)(1)\begin{bmatrix}1\\1\end{bmatrix} + (0.25)(-1)\begin{bmatrix}-1\\1\end{bmatrix} = \begin{bmatrix}1\\0.5\end{bmatrix}$$

$$\mathbf{R} = E[\mathbf{z}\mathbf{z}^T] = \mathbf{p}_1\mathbf{p}_1^T(0.75) + \mathbf{p}_2\mathbf{p}_2^T(0.25)$$

$$= 0.75\begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}1 & 1\end{bmatrix} + 0.25\begin{bmatrix}-1\\1\end{bmatrix}\begin{bmatrix}-1 & 1\end{bmatrix} = \begin{bmatrix}1 & 0.5\\0.5 & 1\end{bmatrix}$$

Performance Contour

Optimum Point (Maximum Likelihood)

$$\mathbf{x}^{ML} = -\mathbf{A}^{-1}\mathbf{d} = \mathbf{R}^{-1}\mathbf{h} = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Hessian Matrix

$$\nabla^2 F(\mathbf{x}) = \mathbf{A} = 2\mathbf{R} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

Eigenvalues

$$\left| \mathbf{A} - \lambda\mathbf{I} \right| = \begin{vmatrix} 2-\lambda & 1 \\ 1 & 2-\lambda \end{vmatrix} = \lambda^2 - 4\lambda + 3 = (\lambda - 1)(\lambda - 3) \Rightarrow \lambda_1 = 1, \qquad \lambda_2 = 3$$

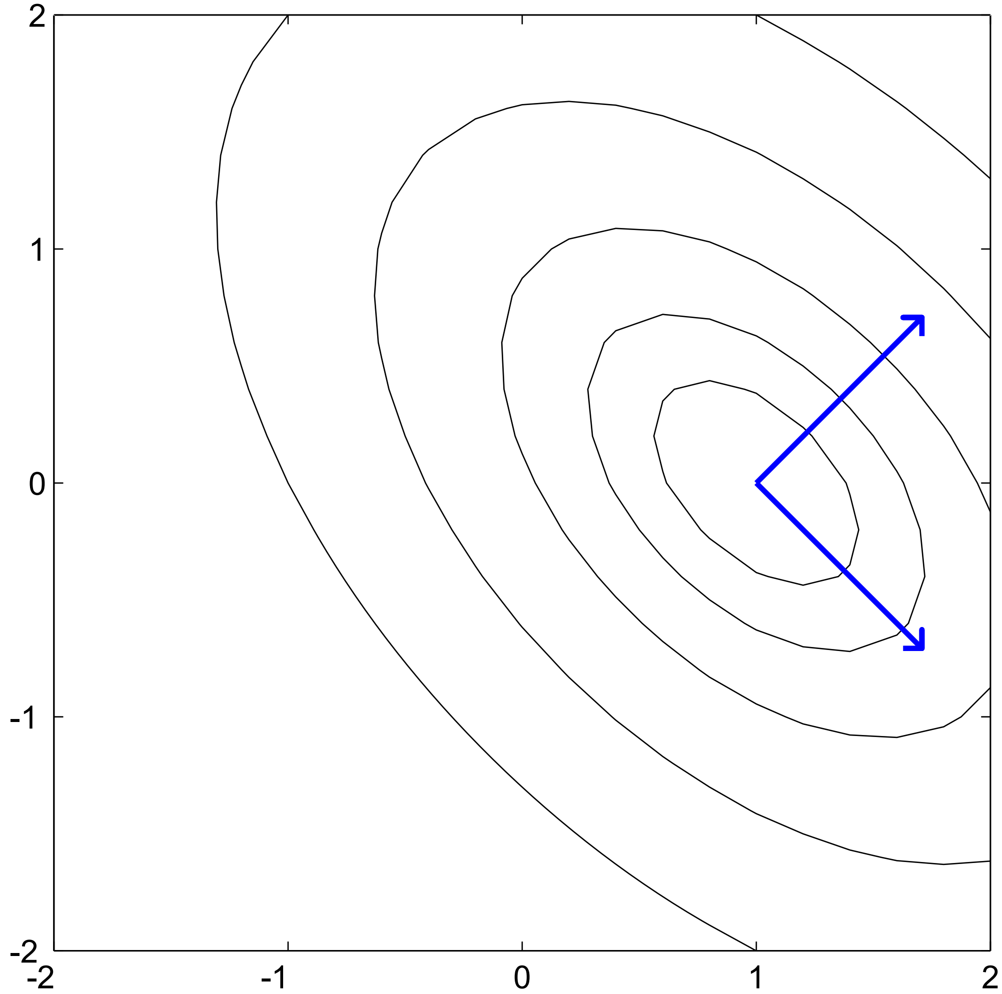
Eigenvectors

$$[\mathbf{A} - \lambda\mathbf{I}]\mathbf{v} = 0$$

$$\lambda_1 = 1 \qquad \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}\mathbf{v}_1 = 0 \qquad \mathbf{v}_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \qquad \lambda_2 = 3 \qquad \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}\mathbf{v}_2 = 0 \qquad \mathbf{v}_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

13

Contour Plot of E_D



$$\gamma = n - 2\alpha^{MP} \mathrm{tr}(\mathbf{H}^{MP})^{-1}$$

Steepest Descent Trajectory

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha\mathbf{g}_k = \mathbf{x}_k - \alpha(\mathbf{A}\mathbf{x}_k + \mathbf{d})$$

$$= \mathbf{x}_k - \alpha\mathbf{A}(\mathbf{x}_k + \mathbf{A}^{-1}\mathbf{d}) = \mathbf{x}_k - \alpha\mathbf{A}(\mathbf{x}_k - \mathbf{x}^{ML})$$

$$= [\mathbf{I} - \alpha\mathbf{A}]\mathbf{x}_k + \alpha\mathbf{A}\mathbf{x}^{ML} = \mathbf{M}\mathbf{x}_k + [\mathbf{I} - \mathbf{M}]\mathbf{x}^{ML}$$

$$\mathbf{M} = [\mathbf{I} - \alpha\mathbf{A}]$$

$$\mathbf{x}_1 = \mathbf{M}\mathbf{x}_0 + [\mathbf{I} - \mathbf{M}]\mathbf{x}^{ML}$$

$$\mathbf{x}_2 = \mathbf{M}\mathbf{x}_1 + [\mathbf{I} - \mathbf{M}]\mathbf{x}^{ML}$$

$$= \mathbf{M}^2\mathbf{x}_0 + \mathbf{M}[\mathbf{I} - \mathbf{M}]\mathbf{x}^{ML} + [\mathbf{I} - \mathbf{M}]\mathbf{x}^{ML}$$

$$= \mathbf{M}^2\mathbf{x}_0 + \mathbf{M}\mathbf{x}^{ML} - \mathbf{M}^2\mathbf{x}^{ML} + \mathbf{x}^{ML} - \mathbf{M}\mathbf{x}^{ML}$$

$$= \mathbf{M}^2\mathbf{x}_0 + \mathbf{x}^{ML} - \mathbf{M}^2\mathbf{x}^{ML} = \mathbf{M}^2\mathbf{x}_0 + [\mathbf{I} - \mathbf{M}^2]\mathbf{x}^{ML}$$

$$\mathbf{x}_k = \mathbf{M}^k\mathbf{x}_0 + [\mathbf{I} - \mathbf{M}^k]\mathbf{x}^{ML}$$

Regularization

$$F(\mathbf{x}) = E_D + \rho E_W \qquad (\rho = \alpha/\beta)$$

$$E_W = \frac{1}{2}(\mathbf{x} - \mathbf{x}_0)^T(\mathbf{x} - \mathbf{x}_0)$$

To locate the minimum point, set the gradient to zero.

$$\nabla F(\mathbf{x}) = \nabla E_D + \rho \nabla E_W$$

$$\nabla E_W = (\mathbf{x} - \mathbf{x}_0) \qquad \nabla E_D = \mathbf{A}(\mathbf{x} - \mathbf{x}^{ML})$$

$$\nabla F(\mathbf{x}) = \mathbf{A}(\mathbf{x} - \mathbf{x}^{ML}) + \rho(\mathbf{x} - \mathbf{x}_0) = \mathbf{0}$$

MAP – ML

$$\mathbf{A}(\mathbf{x}^{MP} - \mathbf{x}^{ML}) = -\rho(\mathbf{x}^{MP} - \mathbf{x}_0) = -\rho(\mathbf{x}^{MP} - \mathbf{x}^{ML} + \mathbf{x}^{ML} - \mathbf{x}_0)$$

$$= -\rho(\mathbf{x}^{MP} - \mathbf{x}^{ML}) - \rho(\mathbf{x}^{ML} - \mathbf{x}_0)$$

$$(\mathbf{A} + \rho\mathbf{I})(\mathbf{x}^{MP} - \mathbf{x}^{ML}) = \rho(\mathbf{x}_0 - \mathbf{x}^{ML})$$

$$(\mathbf{x}^{MP} - \mathbf{x}^{ML}) = \rho(\mathbf{A} + \rho\mathbf{I})^{-1}(\mathbf{x}_0 - \mathbf{x}^{ML})$$

$$\mathbf{x}^{MP} = \mathbf{x}^{ML} - \rho(\mathbf{A} + \rho\mathbf{I})^{-1}\mathbf{x}^{ML} + \rho(\mathbf{A} + \rho\mathbf{I})^{-1}\mathbf{x}_0 = \mathbf{x}^{ML} - \mathbf{M}_\rho\mathbf{x}^{ML} + \mathbf{M}_\rho\mathbf{x}_0$$

$$\mathbf{M}_\rho = \rho(\mathbf{A} + \rho\mathbf{I})^{-1}$$

$$\boxed{\mathbf{x}^{MP} = \mathbf{M}_\rho\mathbf{x}_0 + [\mathbf{I} - \mathbf{M}_\rho]\mathbf{x}^{ML}}$$

Early Stopping – Regularization

$$\mathbf{x}_k = \mathbf{M}^k \mathbf{x}_0 + [\mathbf{I} - \mathbf{M}^k]\mathbf{x}^{ML}$$

$$\mathbf{x}^{MP} = \mathbf{M}_\rho \mathbf{x}_0 + [\mathbf{I} - \mathbf{M}_\rho]\mathbf{x}^{ML}$$

$$\mathbf{M} = [\mathbf{I} - \alpha \mathbf{A}]$$

$$\mathbf{M}_\rho = \rho(\mathbf{A} + \rho \mathbf{I})^{-1}$$

Eigenvalues of $\mathbf{M}^k$:

$$[\mathbf{I} - \alpha \mathbf{A}]\mathbf{z}_i = \mathbf{z}_i - \alpha \mathbf{A} \mathbf{z}_i = \mathbf{z}_i - \alpha \lambda_i \mathbf{z}_i = \underbrace{(1 - \alpha \lambda_i)}_{\text{Eigenvalues of } \mathbf{M}} \mathbf{z}_i$$

$\mathbf{z}_i$ - eigenvector of $\mathbf{A}$

λ_i - eigenvalue of $\mathbf{A}$

Eigenvalues of $\mathbf{M}$

$$\text{eig}(\mathbf{M}^k) = (1 - \alpha \lambda_i)^k$$

Eigenvalues of $\mathbf{M}_\rho$:

$$\text{eig}(\mathbf{M}_\rho) = \frac{\rho}{(\lambda_i + \rho)}$$

13 Reg. Parameter – Iteration Number

$\mathbf{M}^k$ and $\mathbf{M}_\rho$ have the same eigenvectors. They would be equal if their eigenvalues were equal.

$$\frac{\rho}{(\lambda_i + \rho)} = (1 - \alpha\lambda_i)^k \qquad \text{Taking log :} \qquad -\log\left(1 + \frac{\lambda_i}{\rho}\right) = k\log(1 - \alpha\lambda_i)$$

Since these are equal at $\lambda_i = 0$, they are always equal if the slopes are equal.

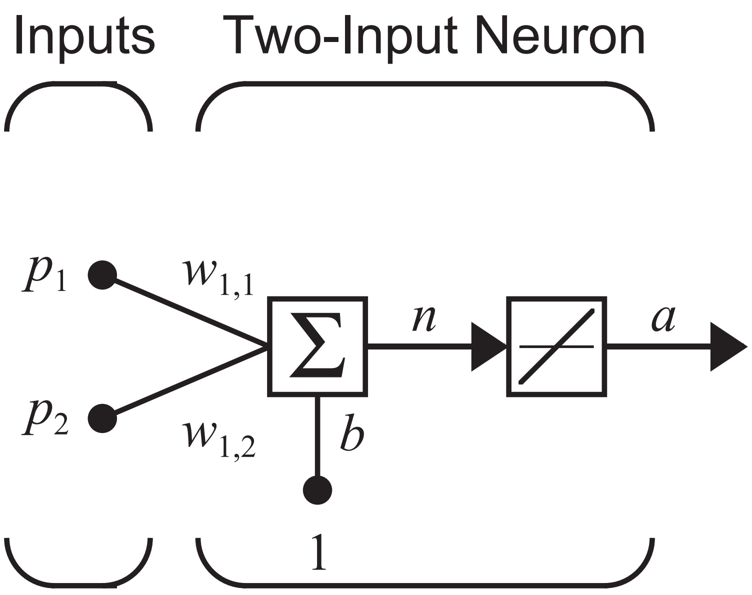
$$-\frac{1}{\left(1 + \frac{\lambda_i}{\rho}\right)}\frac{1}{\rho} = \frac{k}{1 - \alpha\lambda_i}(-\alpha) \quad \Longrightarrow \quad \alpha k = \frac{1}{\rho}\,\frac{(1 - \alpha\lambda_i)}{(1 + \lambda_i/\rho)}$$

If $\alpha\lambda_i$ and λ_i/ρ are small, then:

$$\boxed{\alpha k \cong \frac{1}{\rho}}$$

(Increasing the number of iterations is equivalent to decreasing the regularization parameter!)

Example



$$\left\{\mathbf{p}_1 = \begin{bmatrix}1\\1\end{bmatrix}, t_1 = 1\right\}$$ (Probability = 0.75)

$$\left\{\mathbf{p}_2 = \begin{bmatrix}-1\\1\end{bmatrix}, t_2 = -1\right\}$$ (Probability = 0.25)

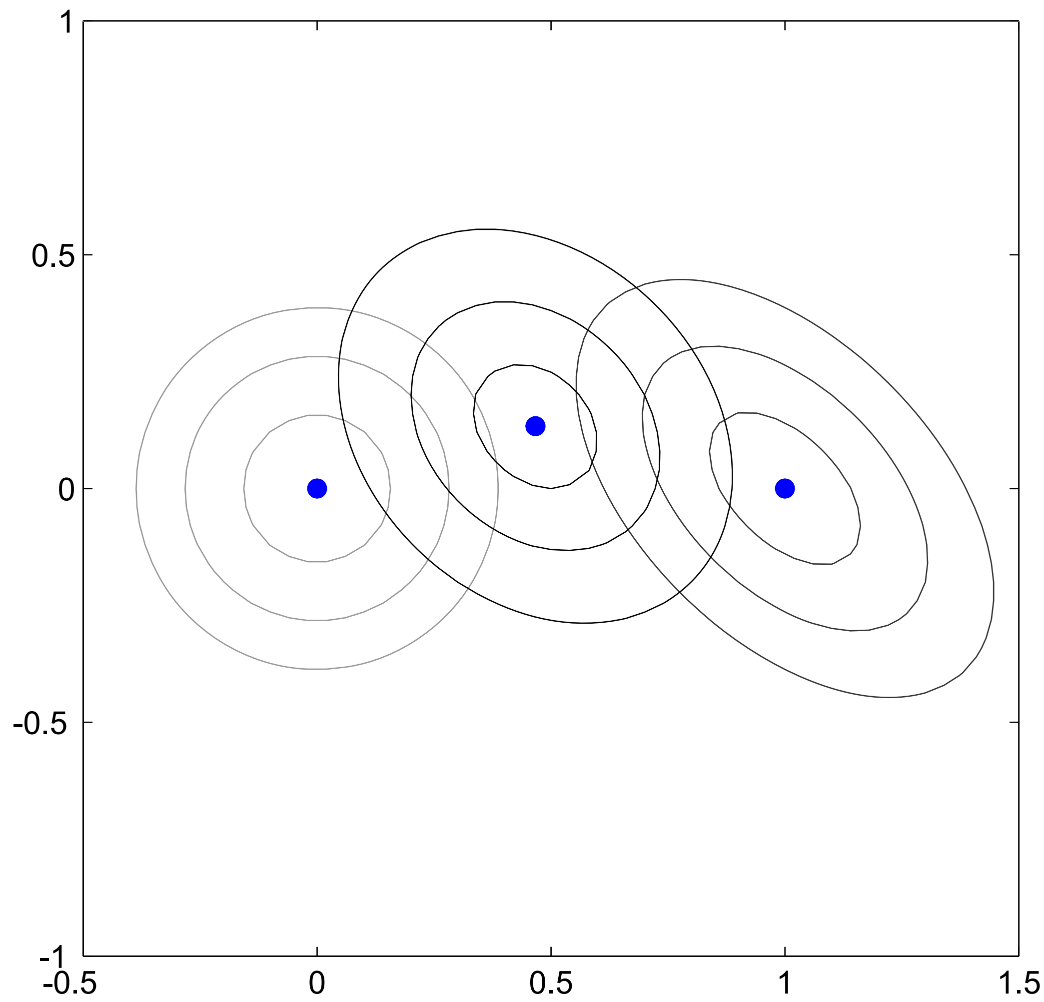
$$F(\mathbf{x}) = E_D + \rho E_W$$

$$E_D = c + \mathbf{x}^T\mathbf{d} + \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x}$$

$$E_W = \frac{1}{2}\mathbf{x}^T\mathbf{x} \qquad c = 1 \qquad \mathbf{d} = -2\mathbf{h} = \begin{bmatrix}-2\\-1\end{bmatrix} \qquad \mathbf{A} = 2\mathbf{R} = \begin{bmatrix}2 & 1\\1 & 2\end{bmatrix}$$

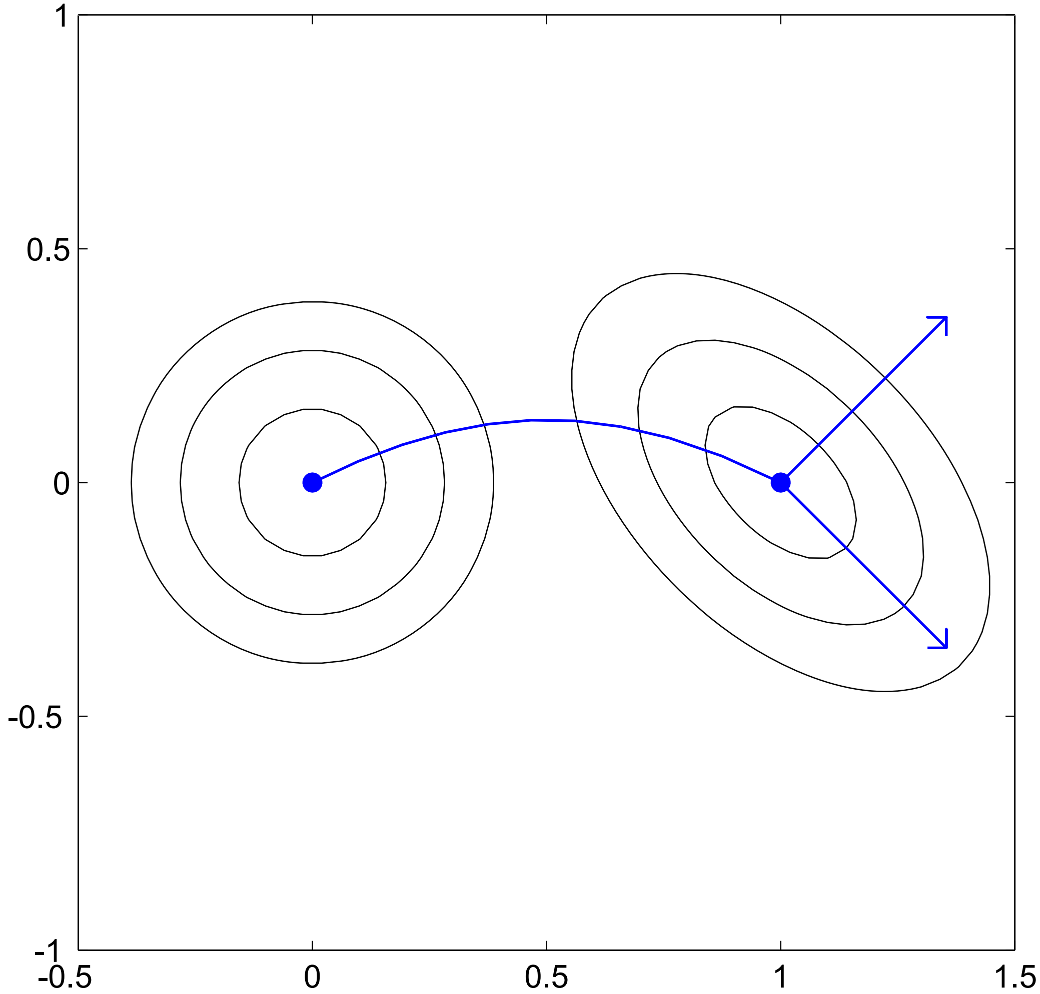
$$\nabla^2 F(\mathbf{x}) = \nabla^2 E_D + \rho\nabla^2 E_W = \begin{bmatrix}2 & 1\\1 & 2\end{bmatrix} + \rho\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix} = \begin{bmatrix}2+\rho & 1\\1 & 2+\rho\end{bmatrix}$$

$\rho = 0, 2, \infty$

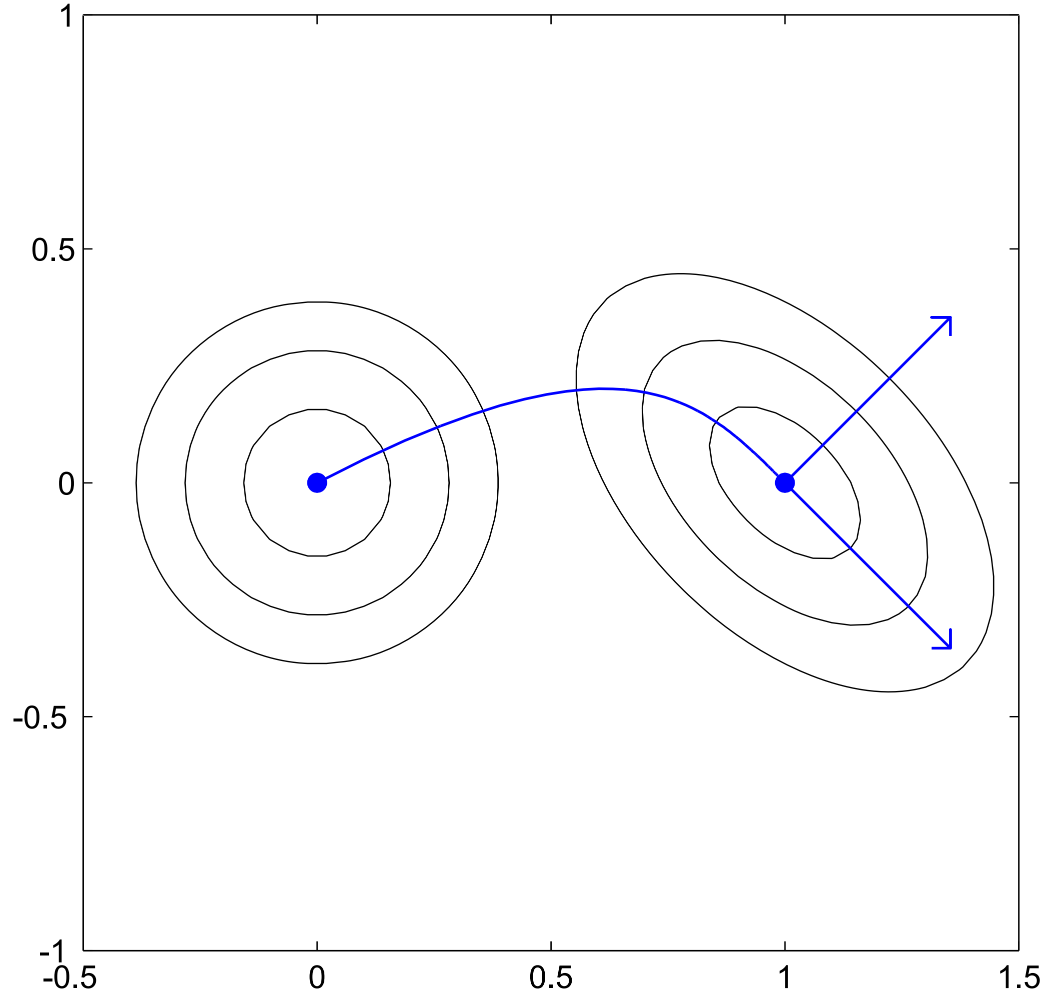


13

$\rho = 0 \rightarrow \infty$



Steepest Descent Path



Effective Number of Parameters

$$\gamma = n - 2\alpha^{MP}\mathrm{tr}\left\{(\mathbf{H}^{MP})^{-1}\right\}$$

$$\mathbf{H}(\mathbf{x}) = \nabla^2 F(\mathbf{x}) = \beta\nabla^2 E_D + \alpha\nabla^2 E_W = \beta\nabla^2 E_D + 2\alpha\mathbf{I}$$

$$\mathrm{tr}\{\mathbf{H}^{-1}\} = \sum_{i=1}^{n} \frac{1}{\beta\lambda_i + 2\alpha}$$

$$\gamma = n - 2\alpha^{MP}\mathrm{tr}\left\{(\mathbf{H}^{MP})^{-1}\right\} = n - \sum_{i=1}^{n} \frac{2\alpha}{\beta\lambda_i + 2\alpha} = \sum_{i=1}^{n} \frac{\beta\lambda_i}{\beta\lambda_i + 2\alpha}$$

Effective number of parameters will equal number of large eigenvalues of the Hessian.

$$\gamma = \sum_{i=1}^{n} \frac{\beta\lambda_i}{\beta\lambda_i + 2\alpha} = \sum_{i=1}^{n} \gamma_i \qquad \gamma_i = \frac{\beta\lambda_i}{\beta\lambda_i + 2\alpha} \qquad 0 \le \gamma_i \le 1$$