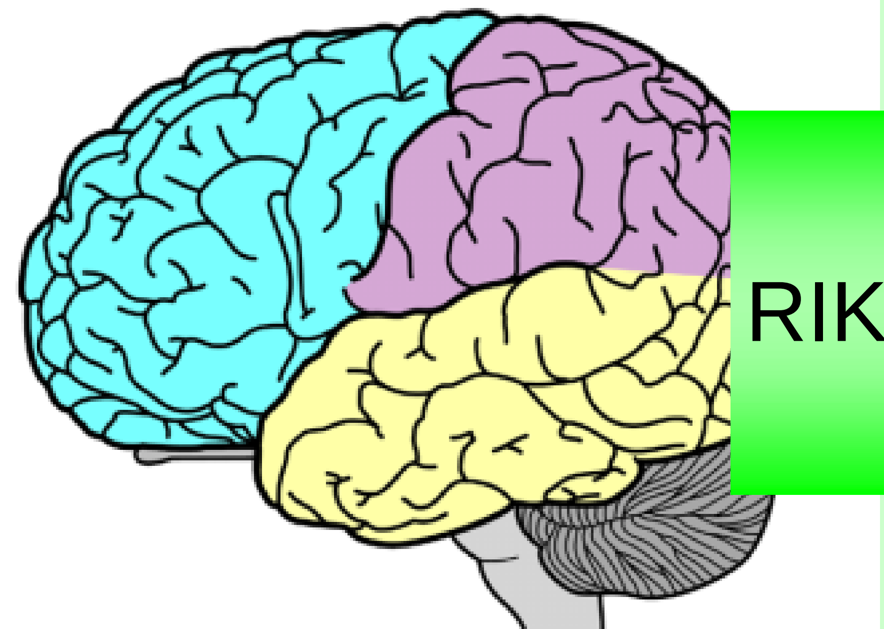
Mathematical Neuroscience Workshop

Information Geometry of

Multilayer Perceptron



Shun-ichi Amari

RIKEN Brain Science Institute

Singular Models

Gaussian mixture

$$p(x;v,w_1,w_2)=(1-v)\varphi(x-w_1)+v\varphi(x-w_2)$$

Population coding

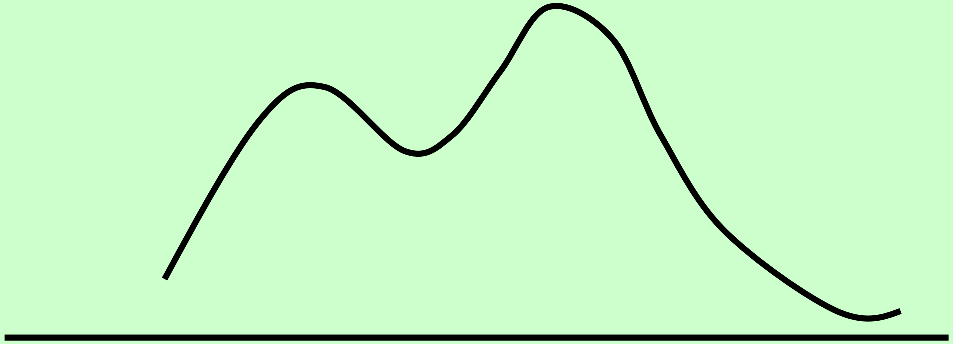
$$r(z)=(1-v)\varphi(z-x_1)+v\varphi(z-x_2)+\sigma\varepsilon(z)$$

Multilayer perceptrons

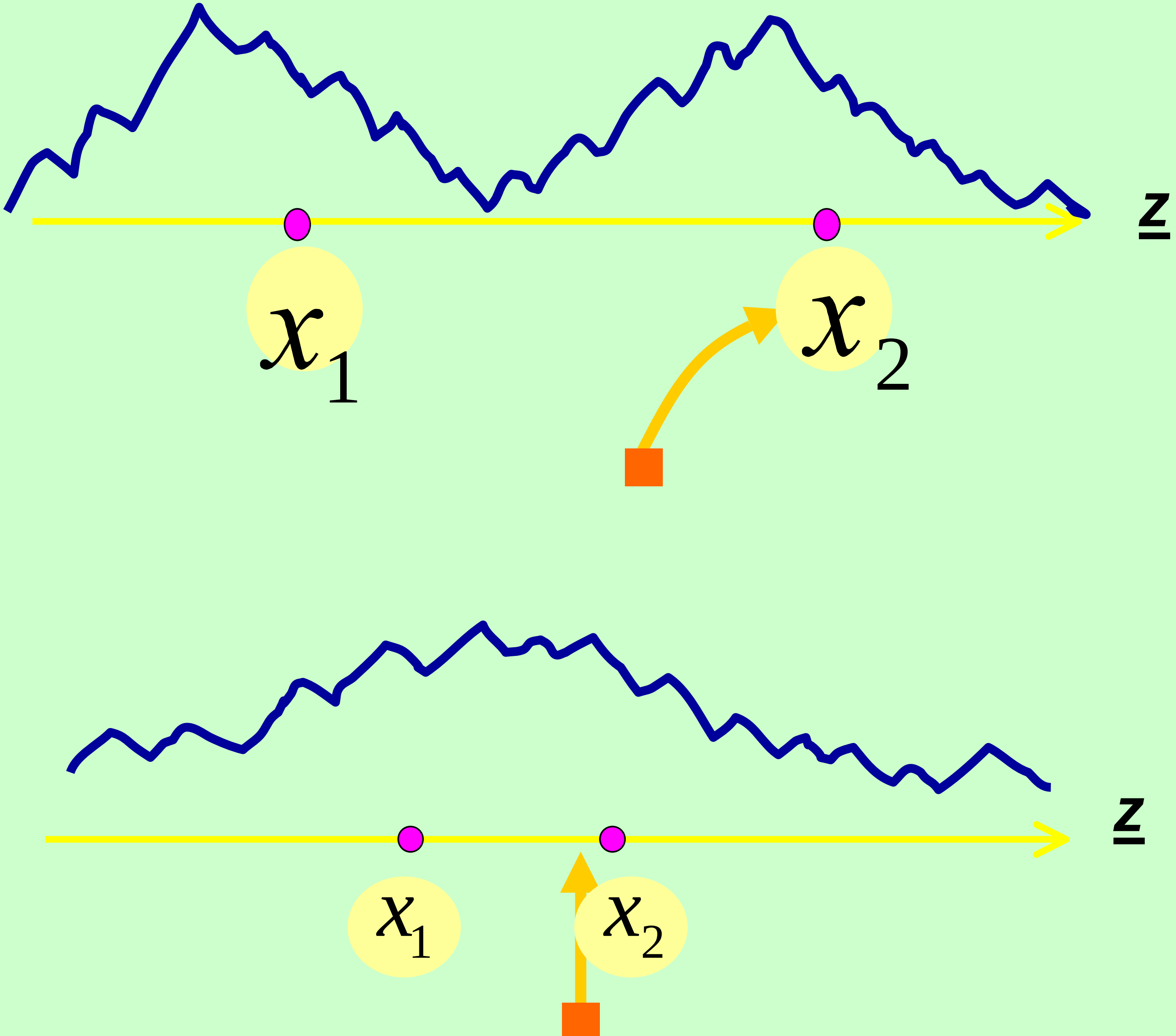
$$y=\sum v_i\varphi(\boldsymbol{w}_i\cdot\boldsymbol{x})+n$$

Gaussian mixtures

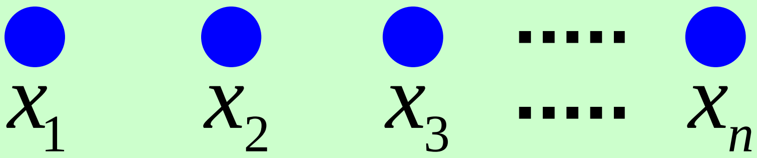
$$p(x) = \sum v_i \exp\left\{ -\frac{1}{2}\left(x - w_i\right)^2 \right\}$$



Two stimuli



Neural Firing



$$p(\mathbf{x}) = p(x_1, x_2, \ldots, x_n)$$

$\eta_i = E[x_i]$ ----firing rate

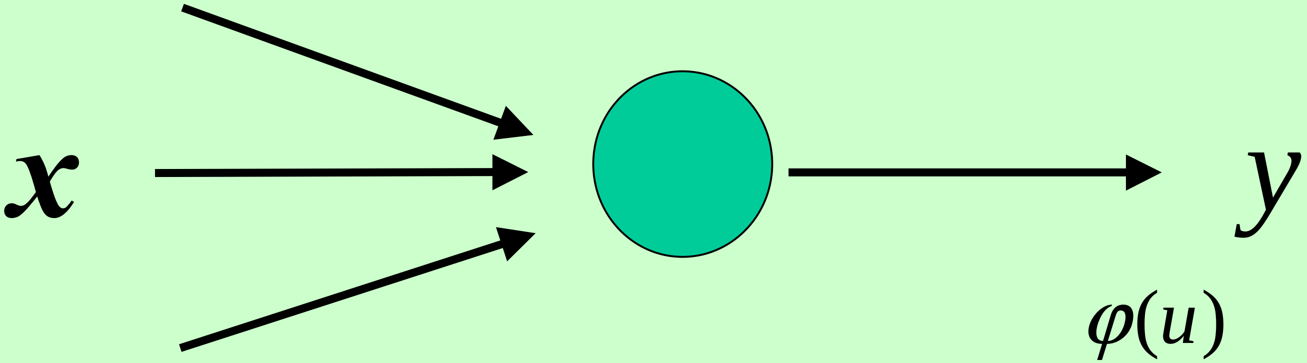
$v_{ij} = Cov[x_i, x_j]$ ----covariance

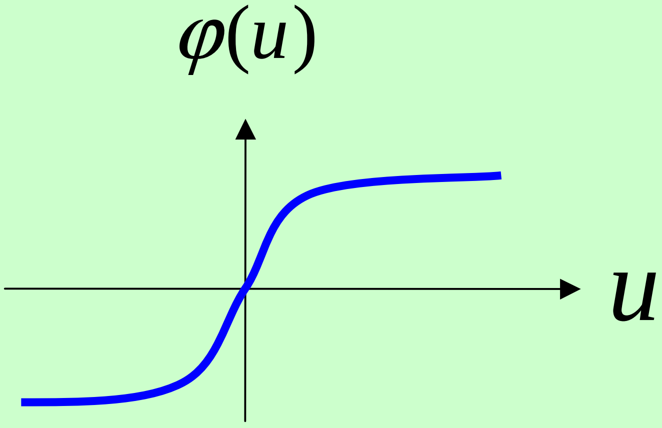
higher-order correlations

orthogonal decomposition

Mathematical Neurons

$$y = \varphi\left(\sum w_i x_i - h\right) = \varphi(\boldsymbol{w} \cdot \boldsymbol{x})$$

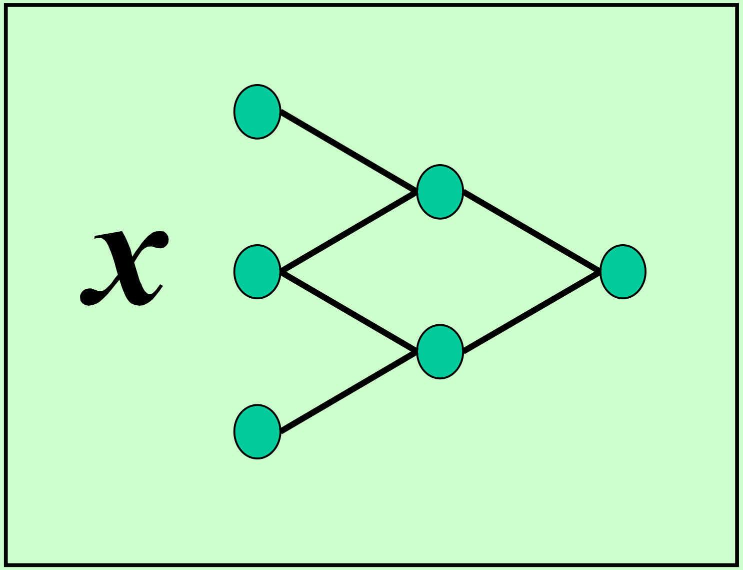




Multilayer Perceptrons

$$y = \sum v_i \varphi\left(\boldsymbol{w}_i \cdot \boldsymbol{x}\right) + n$$

$$x = \left(x_1, x_2, \ldots, x_n\right)$$

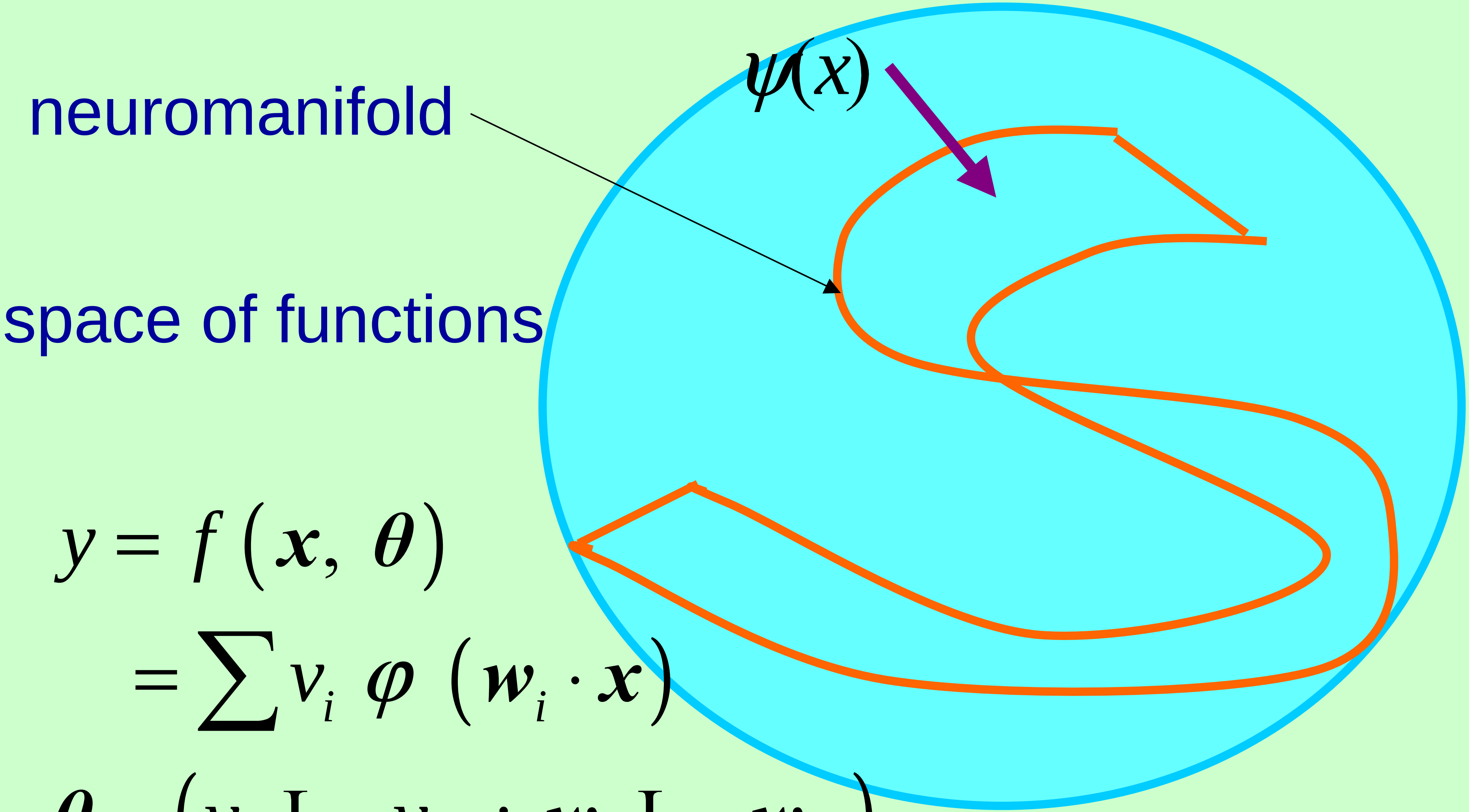


$$p\left(y \middle| \boldsymbol{x}; \boldsymbol{\theta}\right) = c \exp\left\{ -\frac{1}{2}\left(y - f\left(\boldsymbol{x}, \boldsymbol{\theta}\right)\right)^2 \right\}$$

$$f\left(\boldsymbol{x}, \boldsymbol{\theta}\right) = \sum v_i \varphi\left(\boldsymbol{w}_i \cdot \boldsymbol{x}\right)$$

$$\boldsymbol{\theta} = \left(w_1, \ldots, w_m; v_1, \ldots, v_m\right)$$

Manifold of Multilayer Perceptrons



$$y = f(\boldsymbol{x}, \boldsymbol{\theta})$$

$$= \sum v_i \, \varphi \, (\boldsymbol{w}_i \cdot \boldsymbol{x})$$

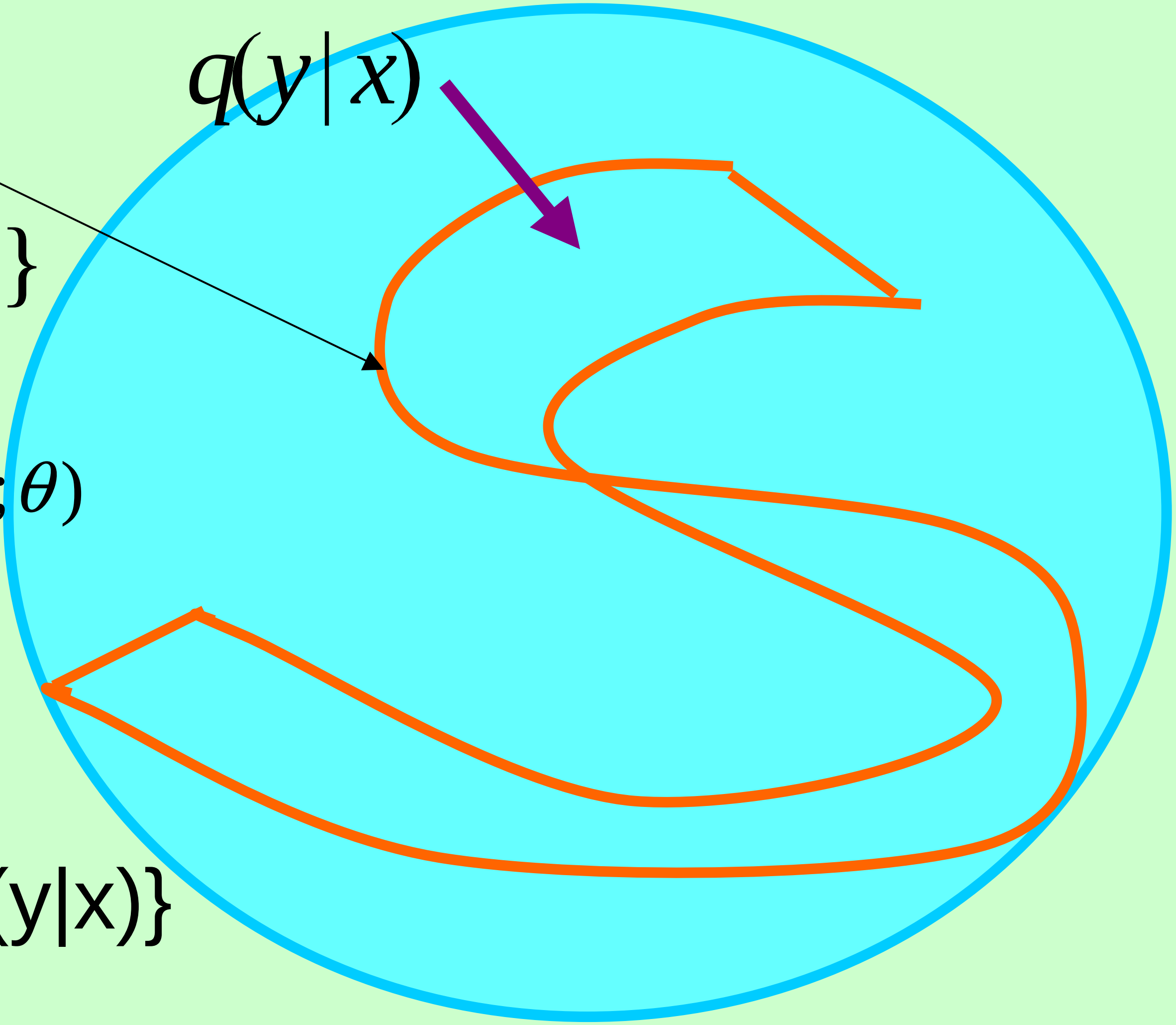
$$\boldsymbol{\theta} = \left(v_1 \mathrm{L} , v_m \; ; \; \boldsymbol{w}_{1,} \mathrm{L} , \boldsymbol{w}_m \right)$$

Multilayer Stochastic Perceptrons

neuromanifold

$$M = \{ p(y \mid x;\theta) \}$$

$$p(y, x;\theta) = q(x)\, p(y \mid x;\theta)$$



space of {q(y|x)}

Learning from examples

$$\psi(\boldsymbol{x}) \approx f\left(x, \hat{\theta}\right)$$

training set T

examples□ $(\boldsymbol{x}_1, y_1),\square\ , (\boldsymbol{x}_n, y_n)$

learning ; estimation

Backpropagation ---gradient learning

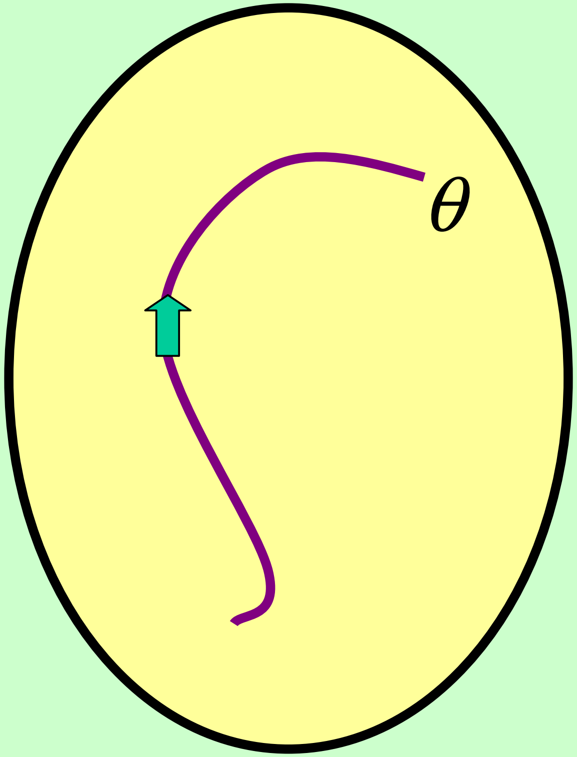
examples : $(y_1, \boldsymbol{x}_1), \mathrm{L}\ (y_t, \boldsymbol{x}_t)$ −−training set

$$E(y, x;\theta) = \frac{1}{2}\left|y - f\left(\boldsymbol{x}, \boldsymbol{\theta}\right)\right|^2$$

$$= -\log p\left(y, \boldsymbol{x}; \boldsymbol{\theta}\right)$$

$$\Delta\boldsymbol{\theta}_t = -\eta_t \frac{\partial E}{\partial \boldsymbol{\theta}}$$

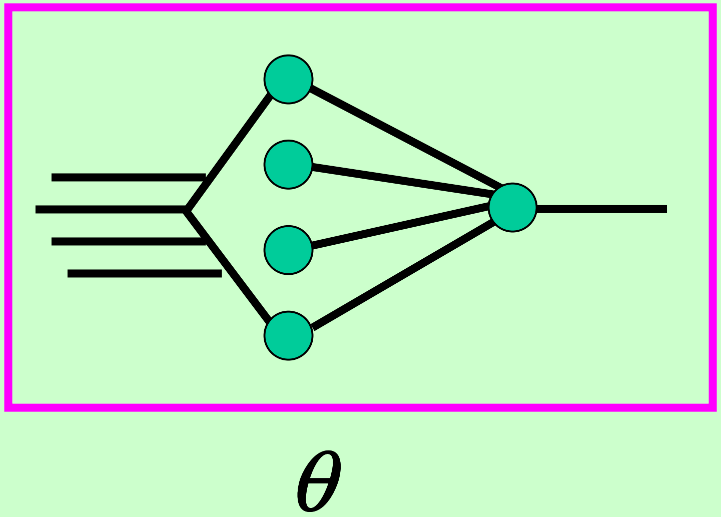
$$f\left(\boldsymbol{x}, \boldsymbol{\theta}\right) = \sum v_i \varphi\left(\boldsymbol{w}_i \cdot \boldsymbol{x}\right)$$

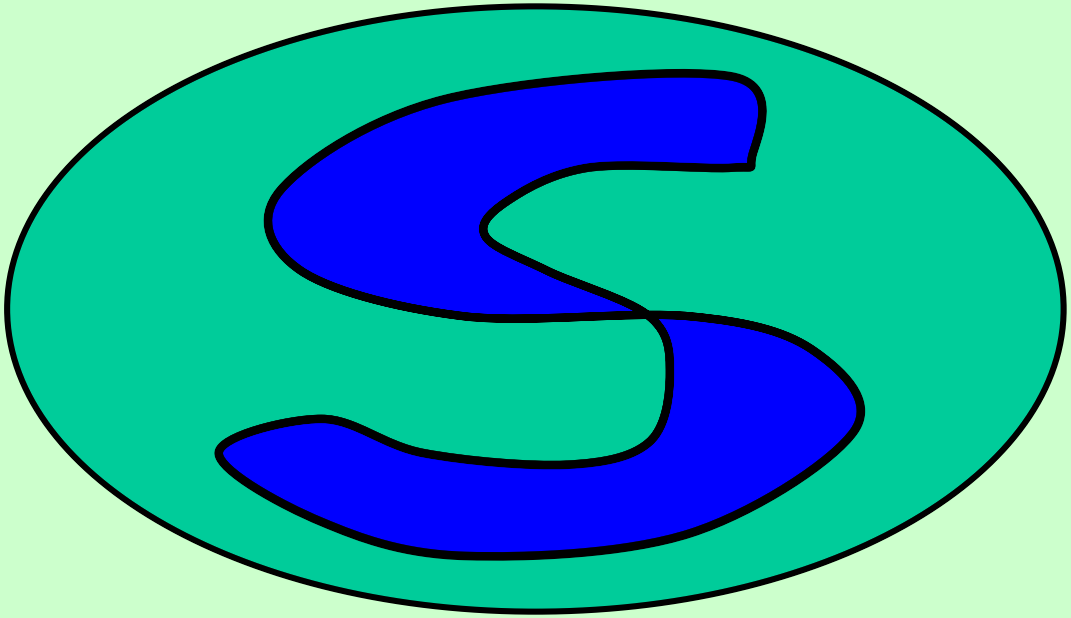


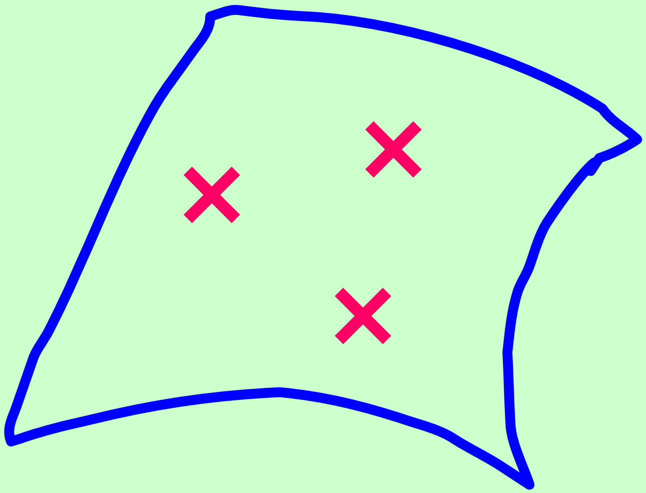
Neuromanifold

- Metrical structure

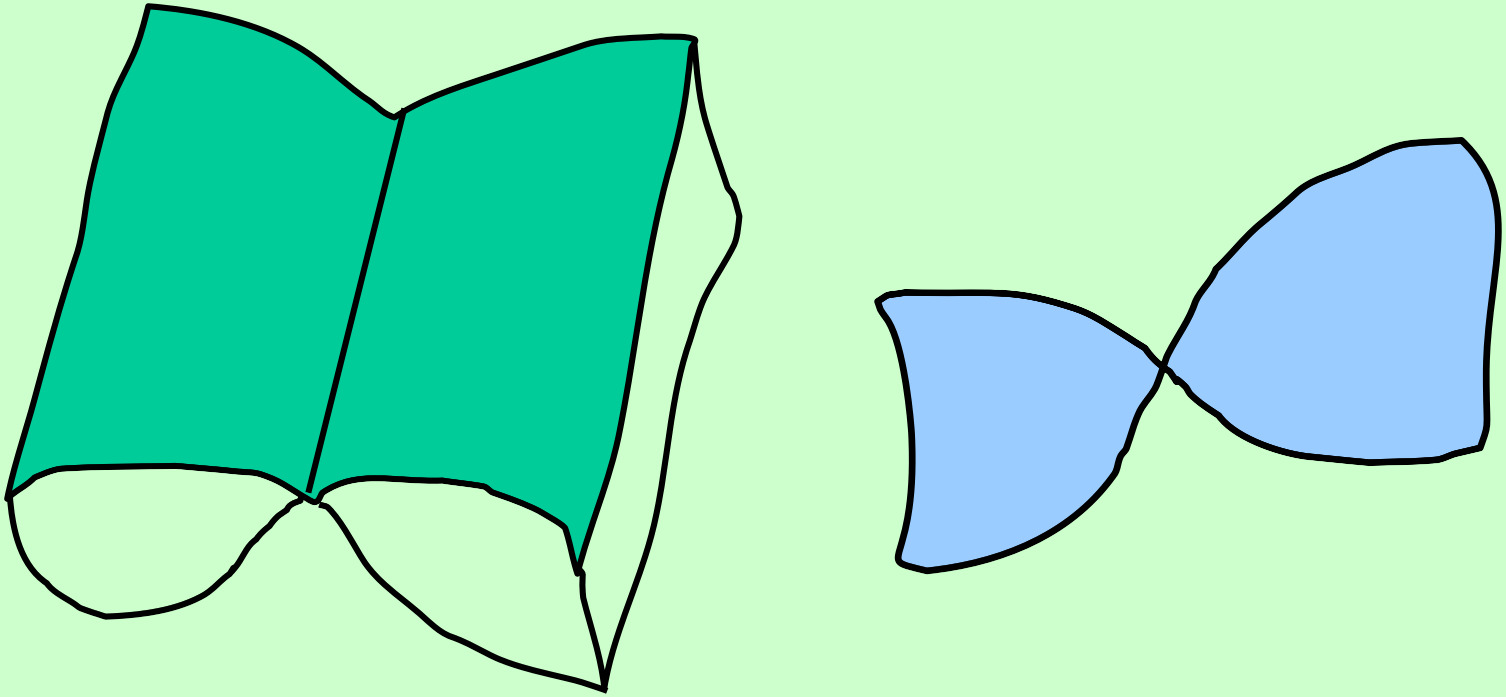
- Topological structure







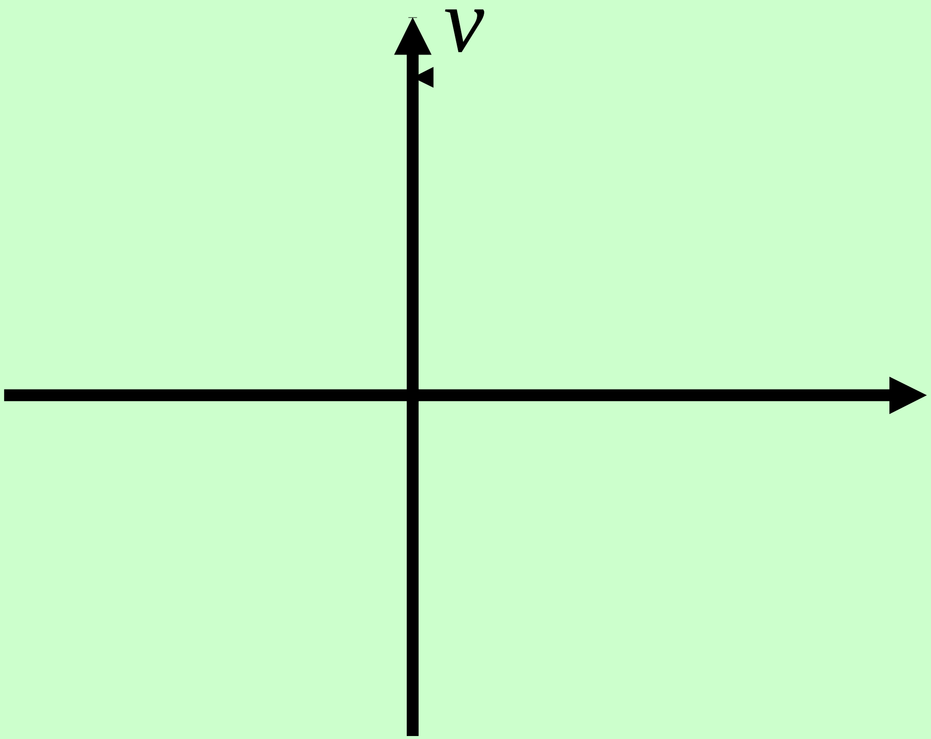
singularities

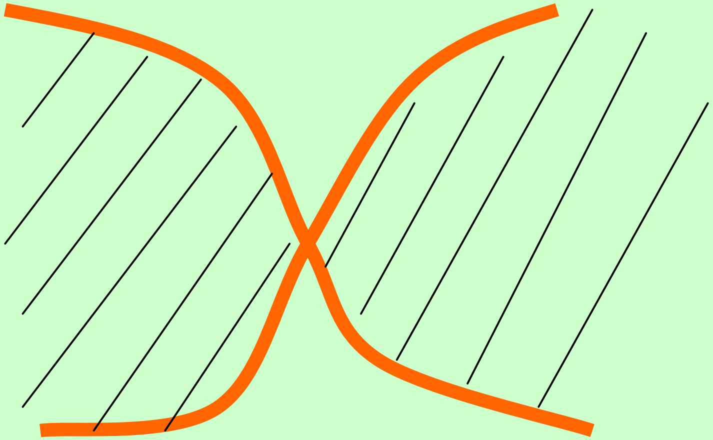


Geometry of singular model

$$y = v\varphi(\boldsymbol{w}\cdot\boldsymbol{x}) + n$$

$$v\,|\,\mathbf{w}\,| = 0$$



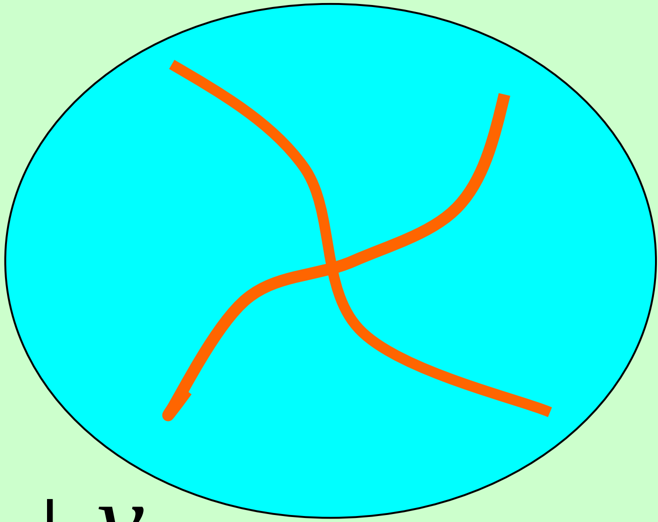


Parameter Space

S

$$S = \{\theta\} \qquad y = \sum v_i \varphi(\boldsymbol{w}_i \cdot \boldsymbol{x}) + n$$

Equivalence

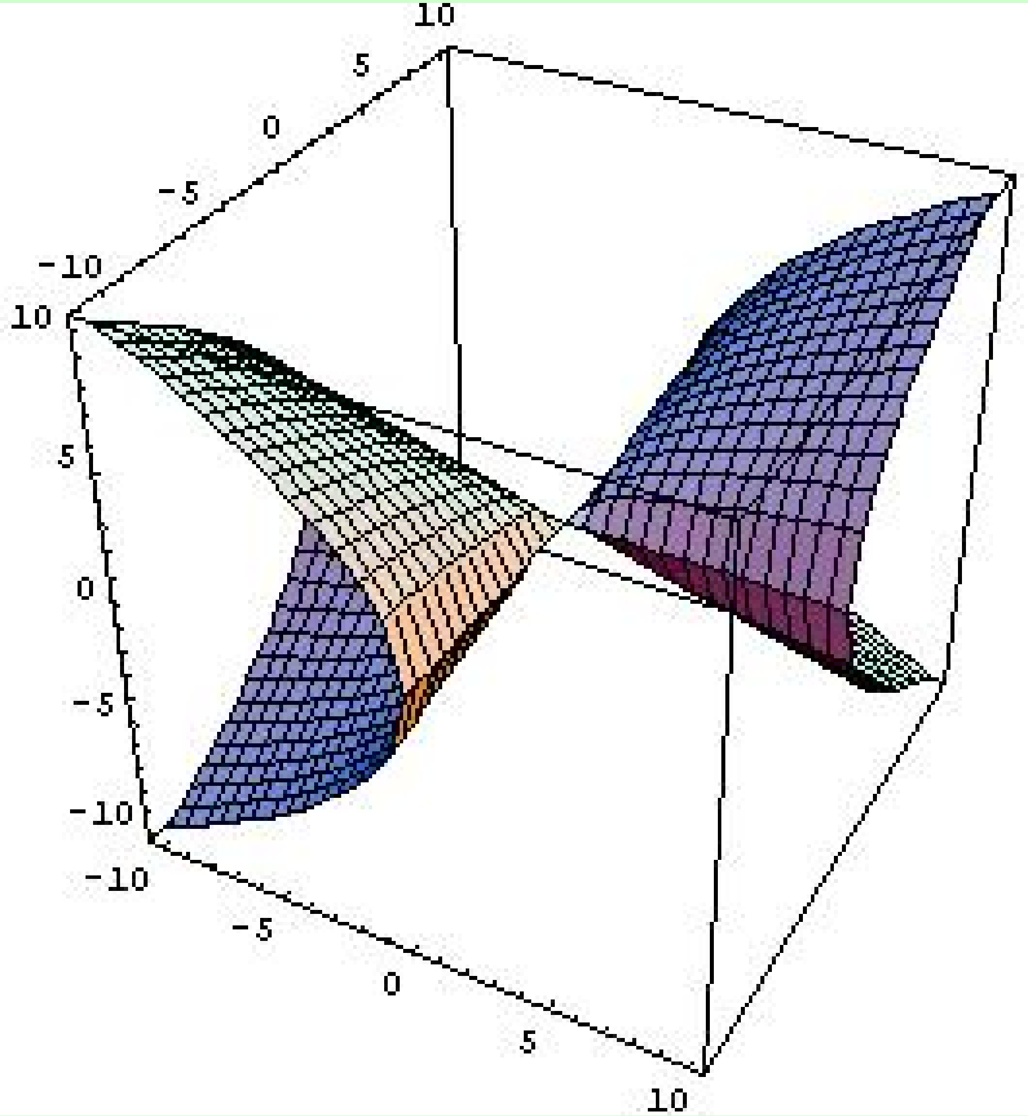


1) $v_i \boldsymbol{w}_i = 0$

2) $\boldsymbol{w}_i = \boldsymbol{w}_j \Rightarrow v_i + v_j$

$$M = S / \approx$$

Singularity of MLP---example



2 hidden-units

$$y = v_1 \varphi\left(\boldsymbol{w}_1 \cdot \boldsymbol{x}\right) + v_2 \varphi\left(\boldsymbol{w}_{\boldsymbol{2}} \cdot \boldsymbol{x}\right) + n$$

$$S:\ v_1 v_2 \left|\boldsymbol{w}_1 - \boldsymbol{w}_{\boldsymbol{2}}\right| \left|\boldsymbol{w}_1 + \boldsymbol{w}_{\boldsymbol{2}}\right| = 0$$

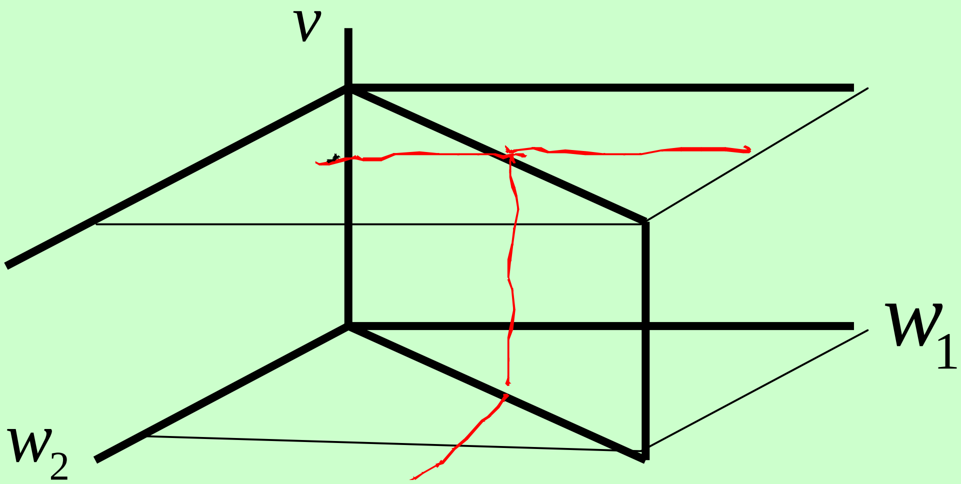
$$(1 - v)\varphi\left(x - w_1\right) + v\varphi\left(x - w_2\right)$$

Gaussian mixture

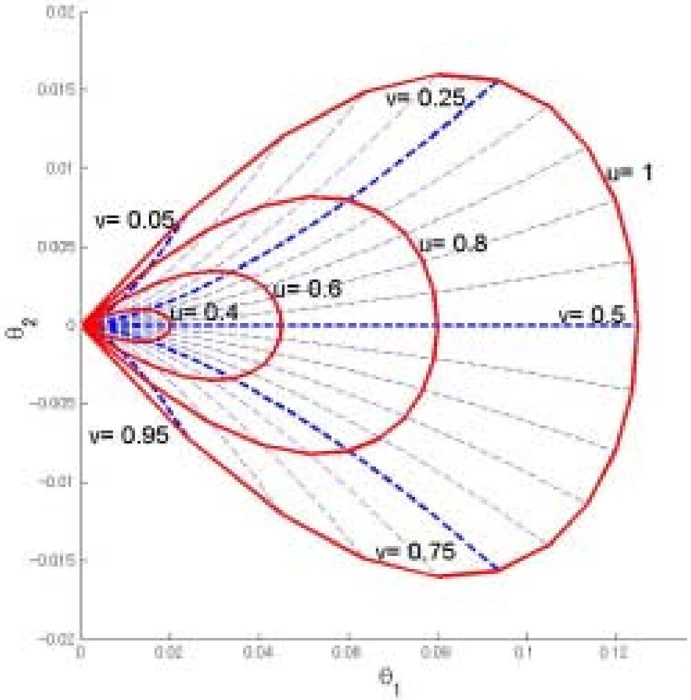
$$p(x;v,w_1,w_2)=(1-v)\varphi(x-w_1)+v\varphi(x-w_2)$$

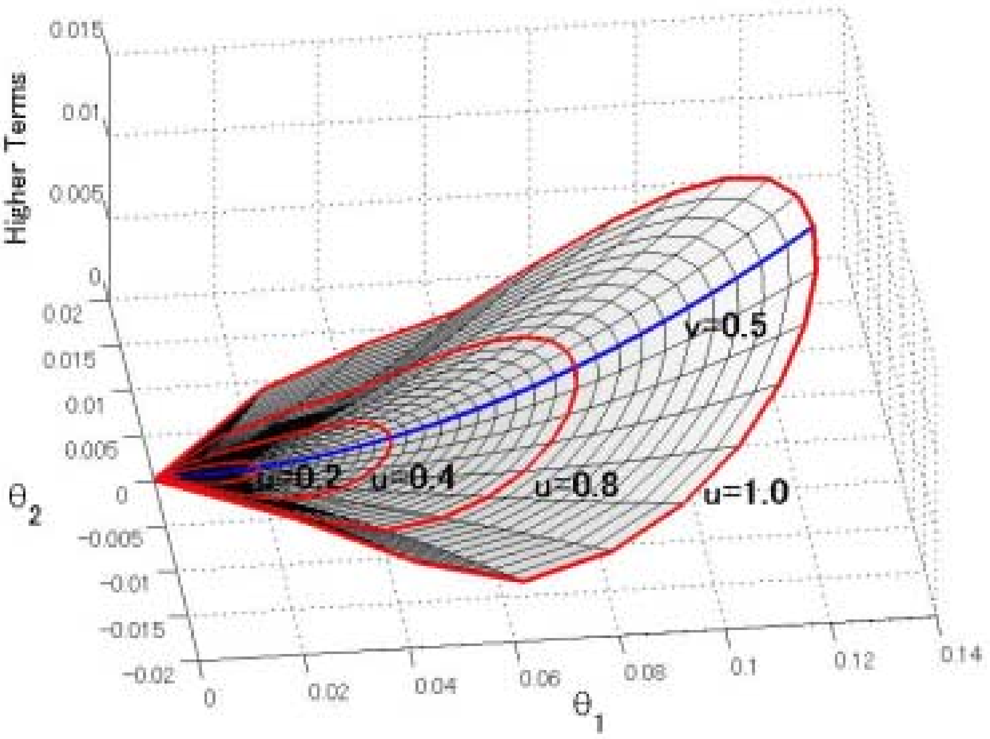
$$\varphi(x)=\frac{1}{\sqrt{2\pi}}\exp\left\{-\frac{1}{2}x^2\right\}$$

singular: $w_1=w_2,$ $v(1-v)=0$

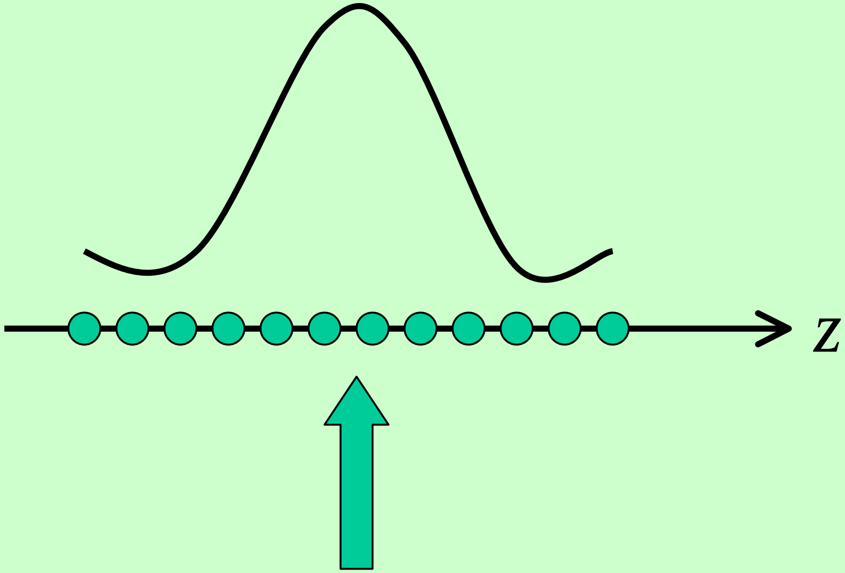


Singular structure of Gaussian mixture model





Population Coding and Neural Field

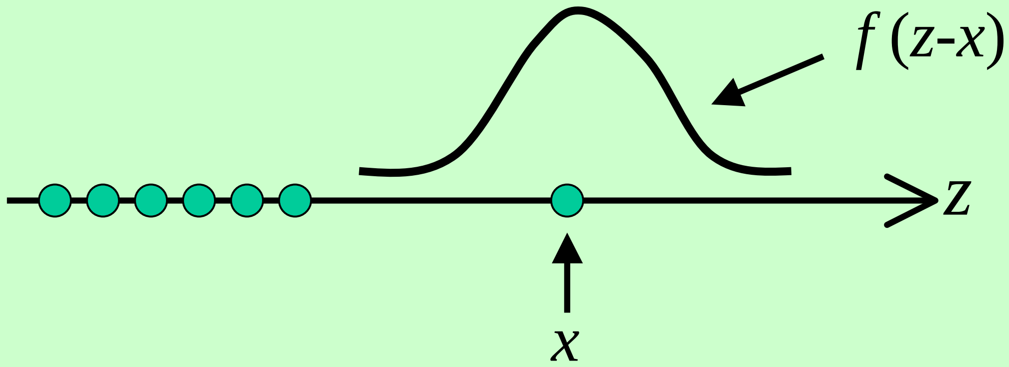


$$x^* \rightarrow r\left(z \mid x^*\right)$$

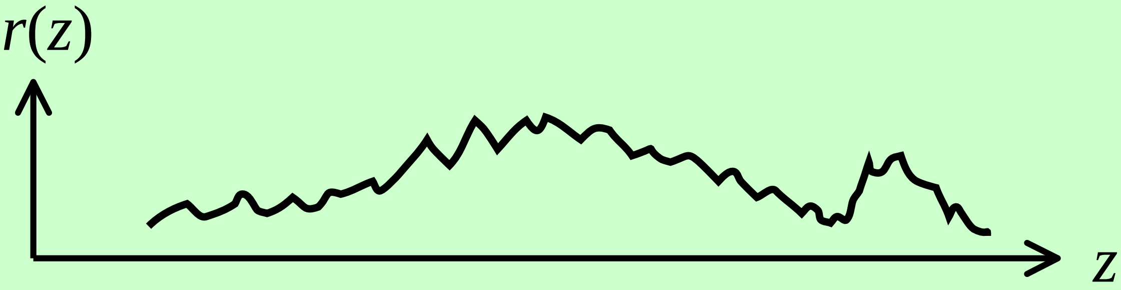
$$r\left(z\right) = f\left(z - x^*\right) + \sigma\varepsilon\left(z\right)$$

$$f\left(z\right) = \exp\left\{-\frac{z^2}{2a^2}\right\}$$

Population Encoding

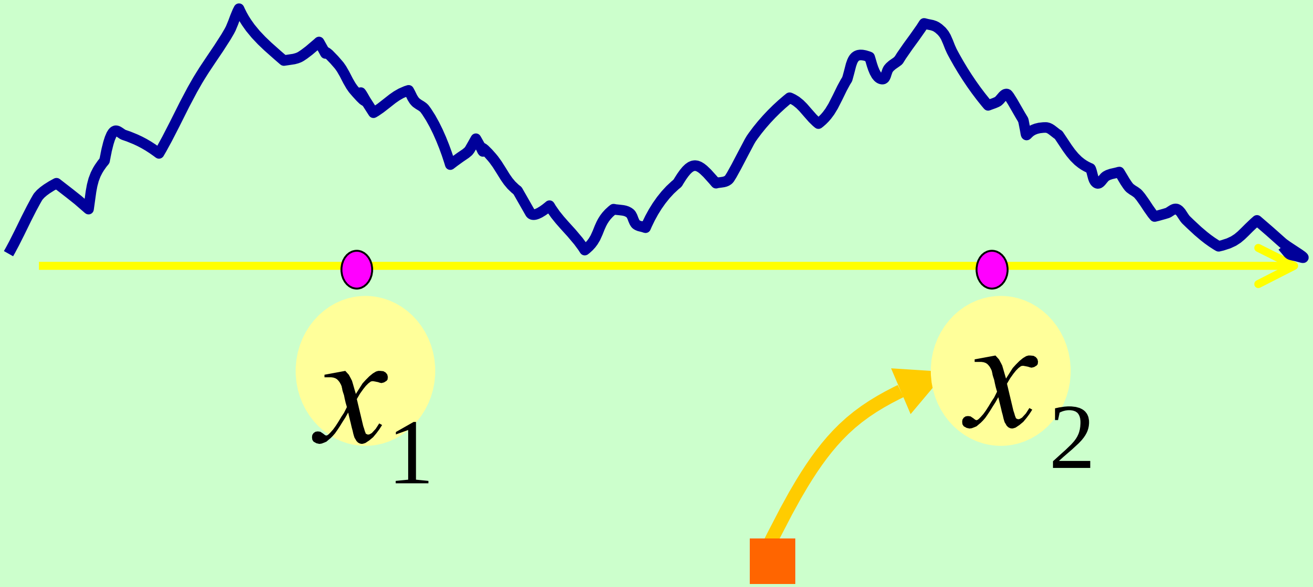


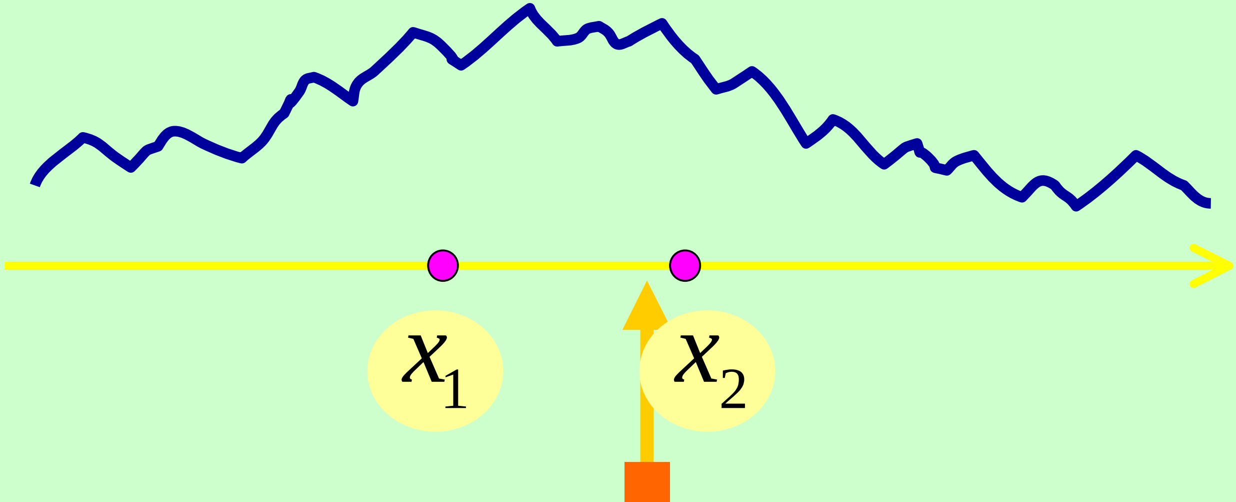
$$r(z) = f(z - x) + \sigma\varepsilon(z)$$



decoding $r(z) \to \hat{x}$

Two stimuli





Neural Activity

$$r(z) = (1 - v)\varphi(z - x_1) + v\varphi(z - x_2) + \sigma\varepsilon(z)$$

$$Q(r(z); v, x_1, x_2) = \exp\left\{-\frac{1}{2\sigma^2}(r - f) * h^{-1} * (r - f)\right\}$$

$$I_{ij} = E\left[\frac{\partial \log Q}{\partial \theta_i}\frac{\partial \log Q}{\partial \theta_j}\right]$$

$I = (I_{ij})$: Fisher information matrix

synfiring resolves singularity

$$\text{phase 1: } f_1(z) = \alpha\bar{v}\varphi(z - x_1) + \bar{\alpha}v\varphi(z - x_2)$$

$$: f_2(z) = \bar{\alpha}\bar{v}\varphi(z - x_1) + \alpha v\varphi(z - x_2)$$

$$\bar{\alpha} = (1-\alpha), \qquad \bar{v} = (1-v)$$

$$I_\xi : \text{ regular as } u \to 0$$

Fisher information

$$g_{ij}(\xi) = E\left[\frac{\partial \log p(x,\xi)}{\partial \xi_i}\frac{\partial \log p(x,\xi)}{\partial \xi_j}\right]$$

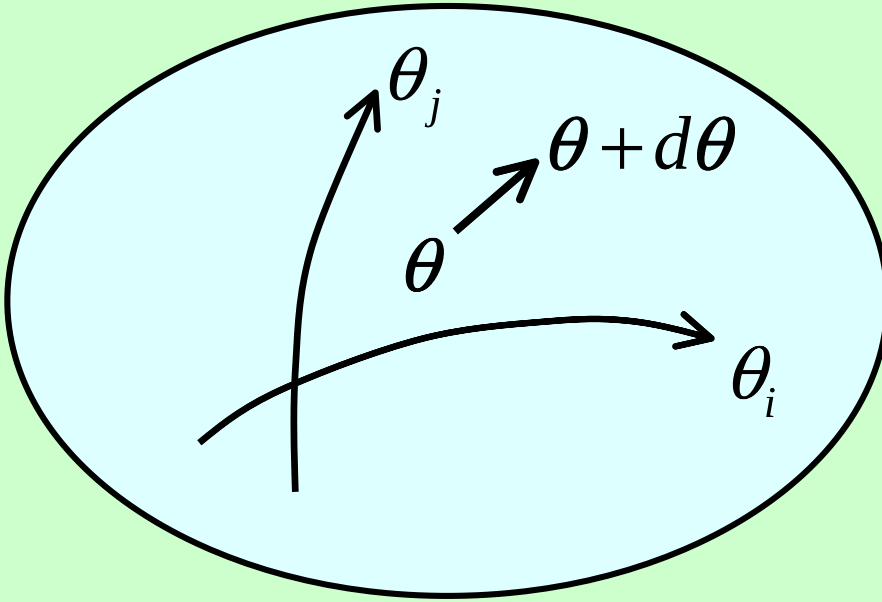
KL-divergence

$$D\left[p(x):q(x)\right] = E_p\left[\log \frac{p(x)}{q(x)}\right]$$

$$D\left[p(x,\xi):p(x,\xi+d\xi)\right] = \frac{1}{2}\sum g_{ij}\,d\xi^i d\xi^j$$

Riemannian manifold

$$g_{ij}(\theta) = E[\frac{\partial \log p(y|x;\theta) \partial \log p(y|x;\theta)}{\partial \theta_i \partial \theta_j}]$$



$$\begin{aligned} ds^2 &= |d\theta|^2 \\ &= \sum g_{ij}(\theta) d\theta_i d\theta_j \\ &= d\theta^T G(\theta) d\theta \end{aligned}$$

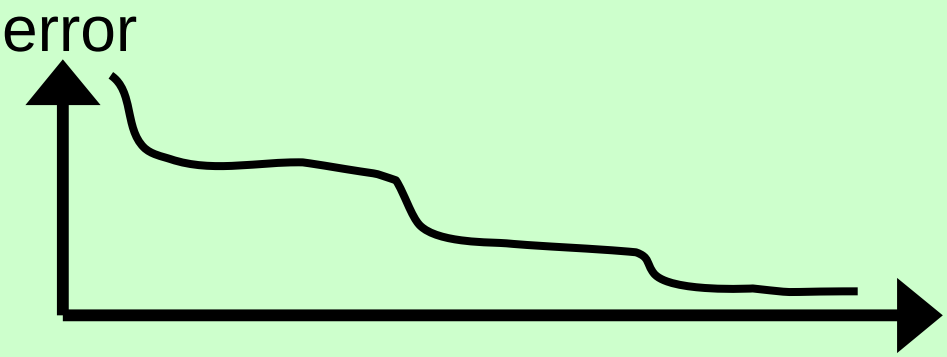
Flaws of Backprop

- slow convergence----plateau---saddle
- local minima

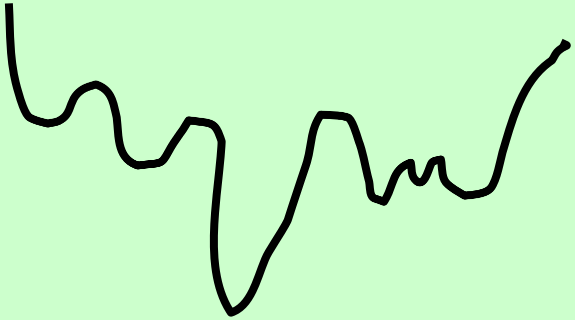
$$\Delta\theta_t = -\eta_t \nabla l(x_t, y_t; \theta_t)$$

Flaws of MLP

slow convergence : Plateaus

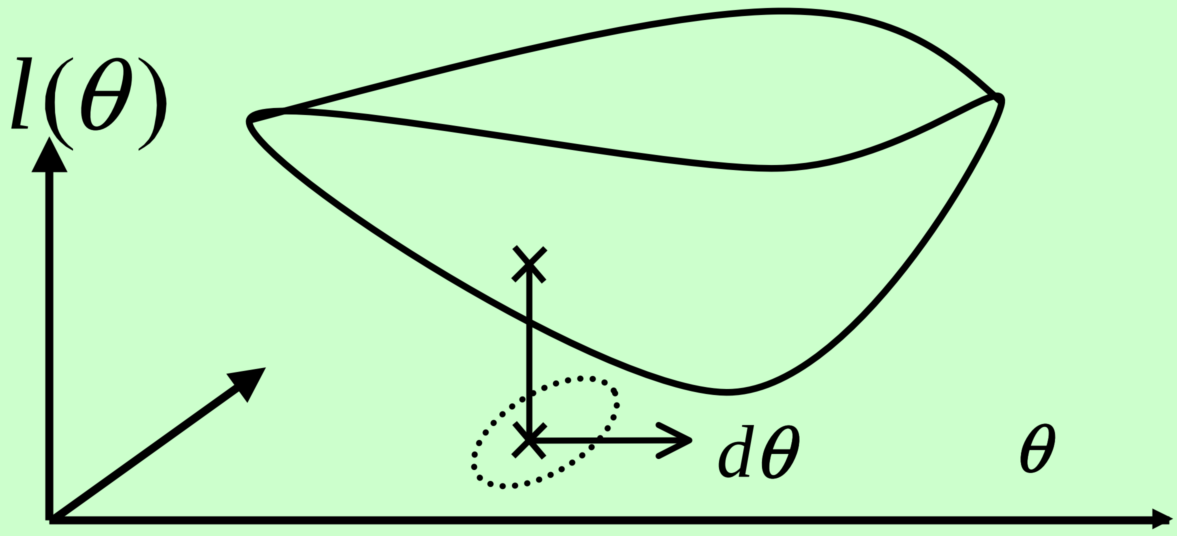


local minima



➡ Boosting and Bagging

Steepest Direction ---Natural Gradient



$$\Delta\theta_t = -\eta_t \nabla l(x_t, y_t; \theta_t)$$

$$\nabla l = \left(\frac{\partial l}{\partial \theta_1}, \ldots, \frac{\partial l}{\partial \theta_n} \right)$$

$$\tilde{\nabla} l = G^{-1}(\theta) \nabla l$$

$$\left| d\theta \right|^2 = d\theta^T G d\theta = \sum G_{ij} d\theta^i d\theta^j$$

Natural Gradient

$$\max \qquad dl = l(\theta + d\theta) - l(\theta)$$

$$|d\theta|^2 = \varepsilon$$

$$\tilde{\nabla} l = G^{-1}(\theta)\nabla l$$

$$\Delta\theta_t = -\eta_t \tilde{\nabla} l(x_t, y_t; \theta_t)$$

Information Geometry of MLP

Natural Gradient Learning :
S. Amari ; H.Y. Park

$$\Delta\theta = -\eta G^{-1}(\theta)\frac{\partial l}{\partial\theta}$$

$$G_{t+1}^{-1} = (1+\varepsilon)G_t^{-1} - \varepsilon G_t^{-1}\nabla f \nabla f^T G_t^{-1}$$

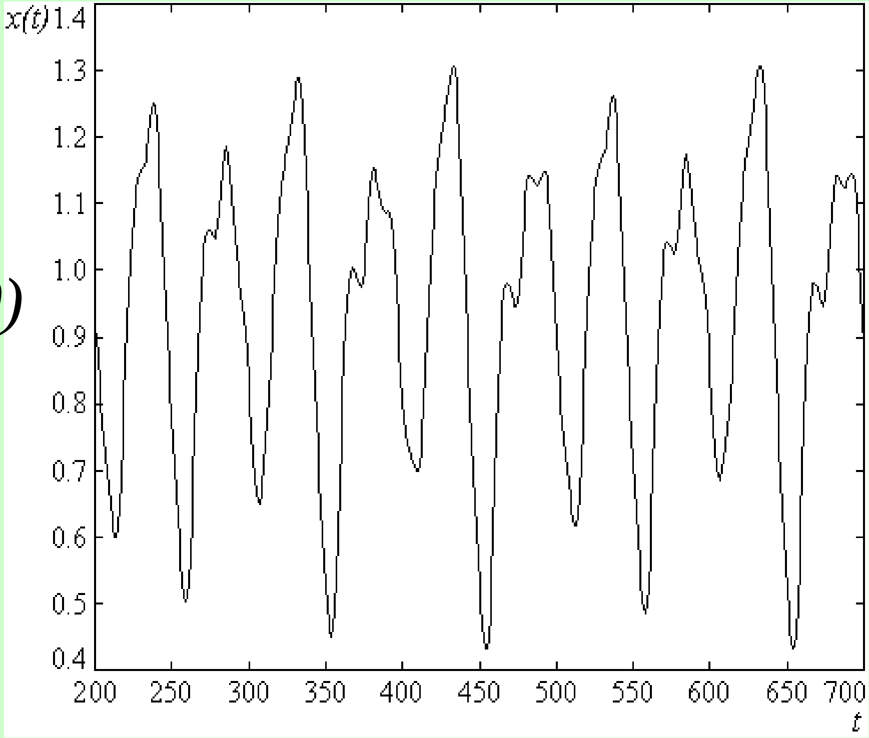
Computational Experiments (1)

- *Mackey-Glass time series prediction*
  - *generation of time series*

$$x(t+1) = (1-b)x(t) + a\frac{x(t-\tau)}{1+x(t-\tau)^{10}}$$

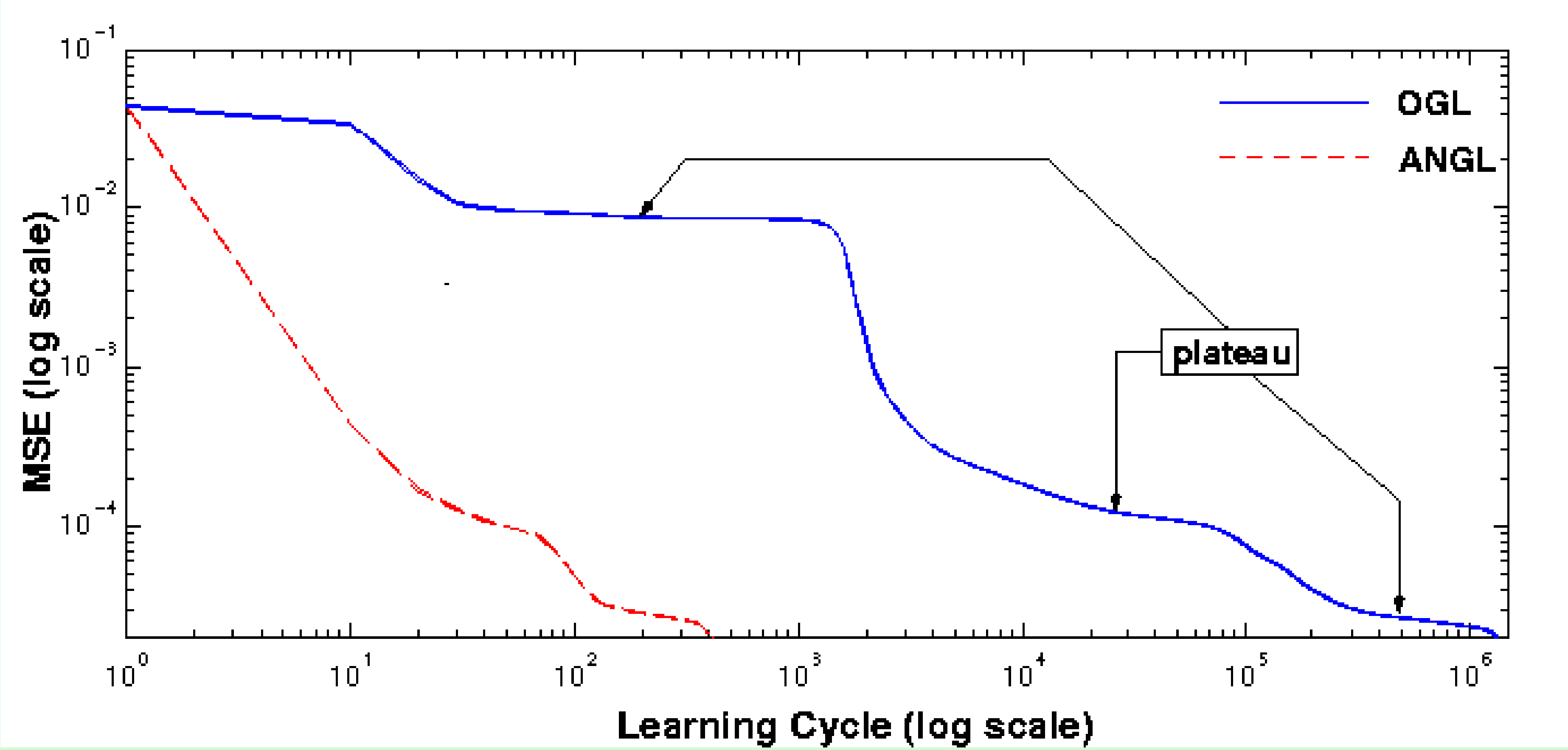
  - *input : 4 previous values ;*
    x(t-18),x(t-12),x(t-6),x(t)
  - *output : 1 future value ; x(t+6)*
  - *learning data : 500 (t=200,…,700)*
  - *test data : 500 (t=5000,…,5500)*
  - *Network Structure*

    4 inputs -- 10 hidden – 1 output



Computational Experiments (1)

- Learning Curves of Mackey-Glass problem

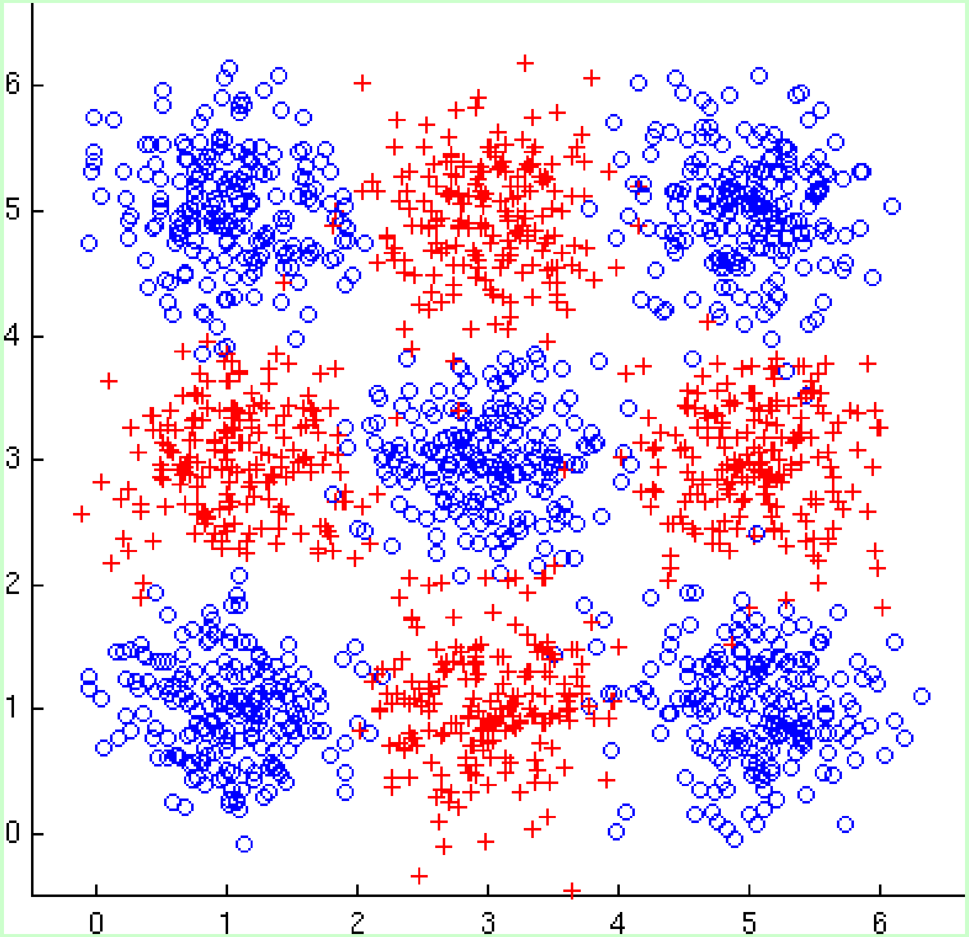


OGL : Ordinary Gradient Descent (Backpropagation)
ANGL : Adaptive Natural Gradient Descent

Computational Experiments (2)

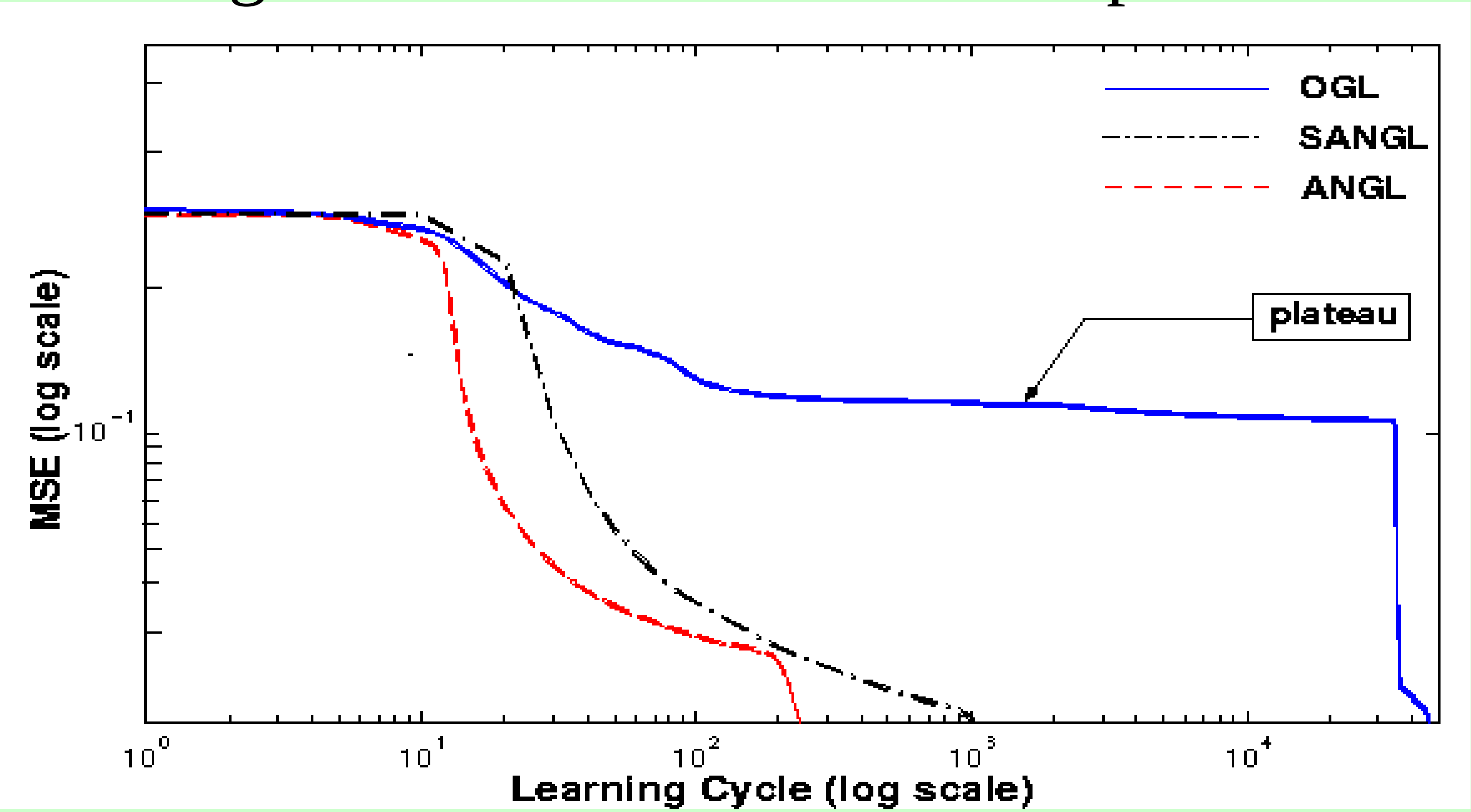
- Extended XOR problems
  - – 2 classes classification
  - – learning data : 1800
  - – test data : 900
  - – Network Structure

    2 inputs -- 8 hidden – 1 output



Computational Experiments (2)

- Learning Curves of Extended XOR problem



OGL : Ordinary Gradient Descent (Backpropagation)
SANGL : Adaptive Natural Gradient for Regression Model (Squared Error)
ANGL : Adaptive Natural Gradient for Classification Model (Cross Entropy Error)

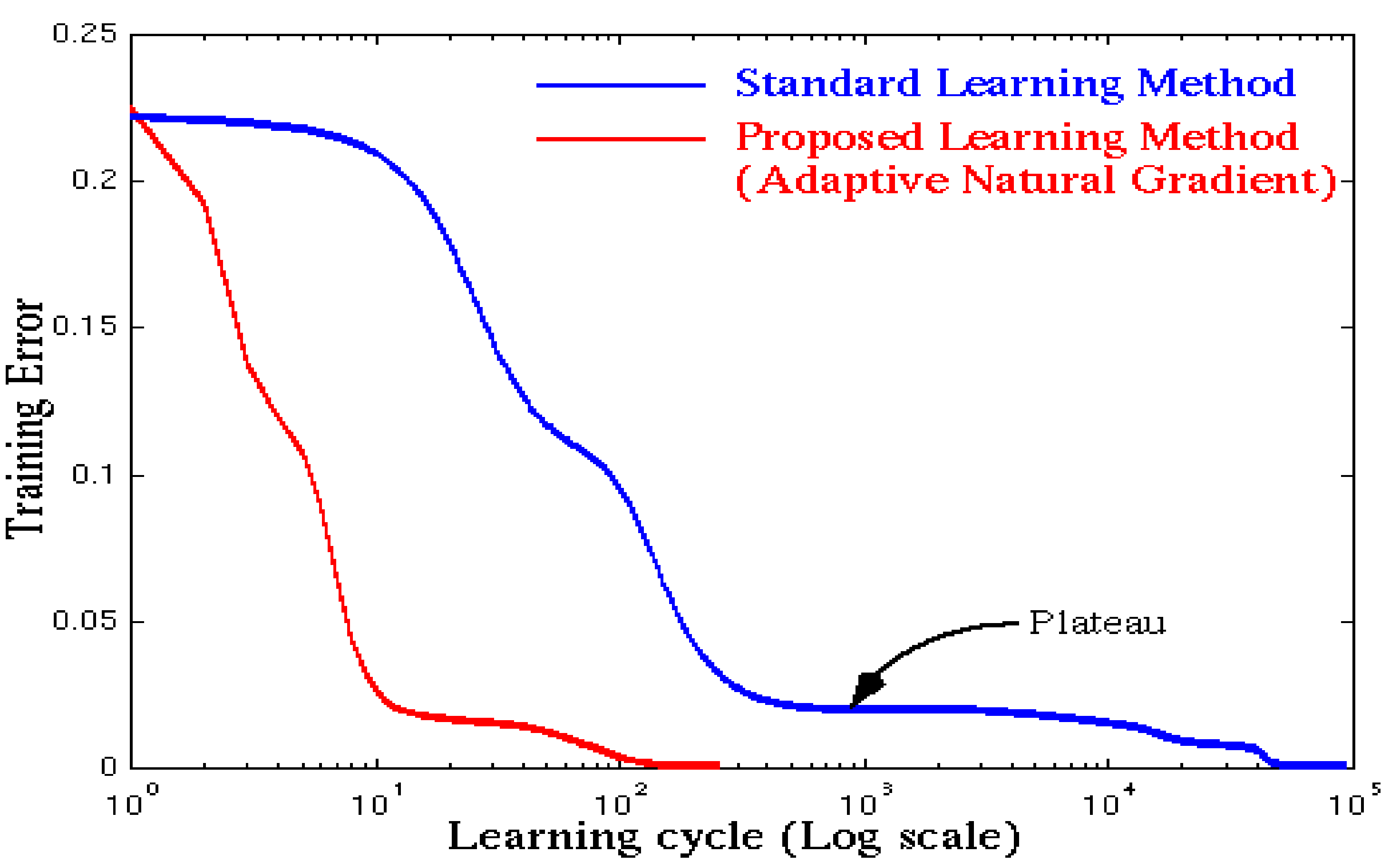
Computational Experiments (3)

- IRIS classification problem
  - classify three different species of iris flower
  - input : 4 attributes about the shape of the plant
    (4 input nodes)
  - output: 3 classes of the flower (3 input nodes)
  - learning data: 90 (30 for each class)
  - test data: 60 (20 for each class)
  - Network Structure

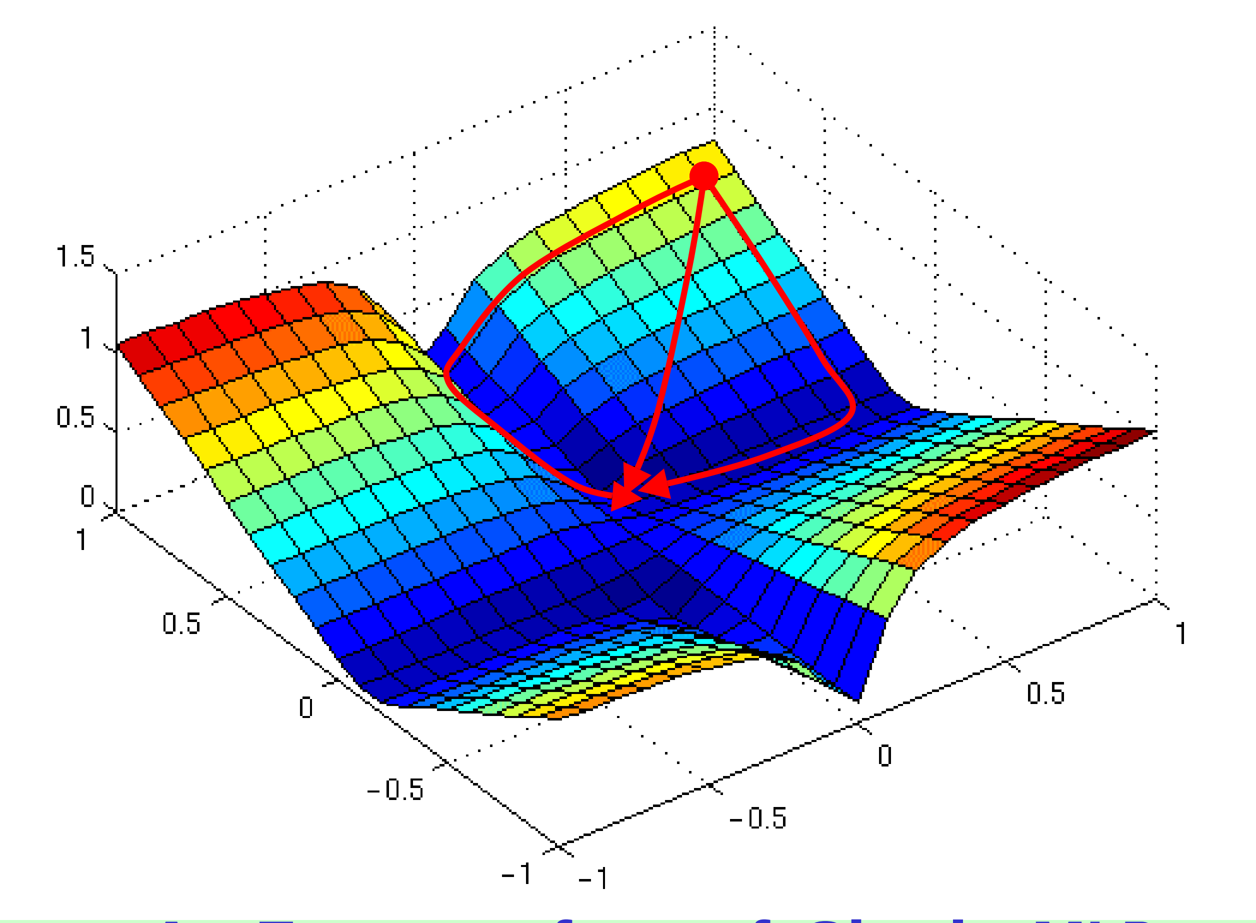
4 inputs -- 4 hidden – 3 outputs

Computational Experiments (3)

- IRIS classification problem



Which path is faster?



An Error surface of Simple MLP

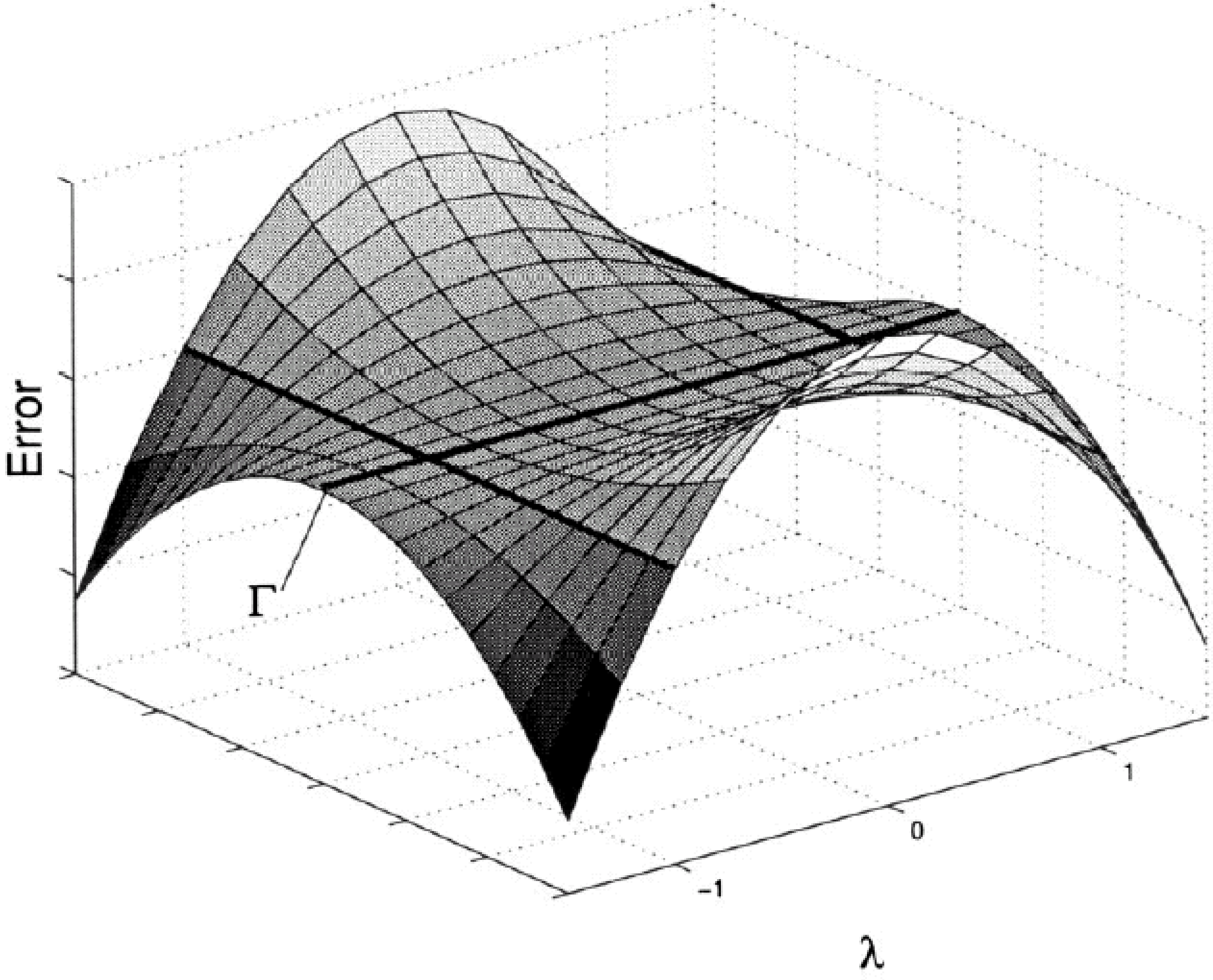


Fig. 5. Critical set with local minima and plateaus.

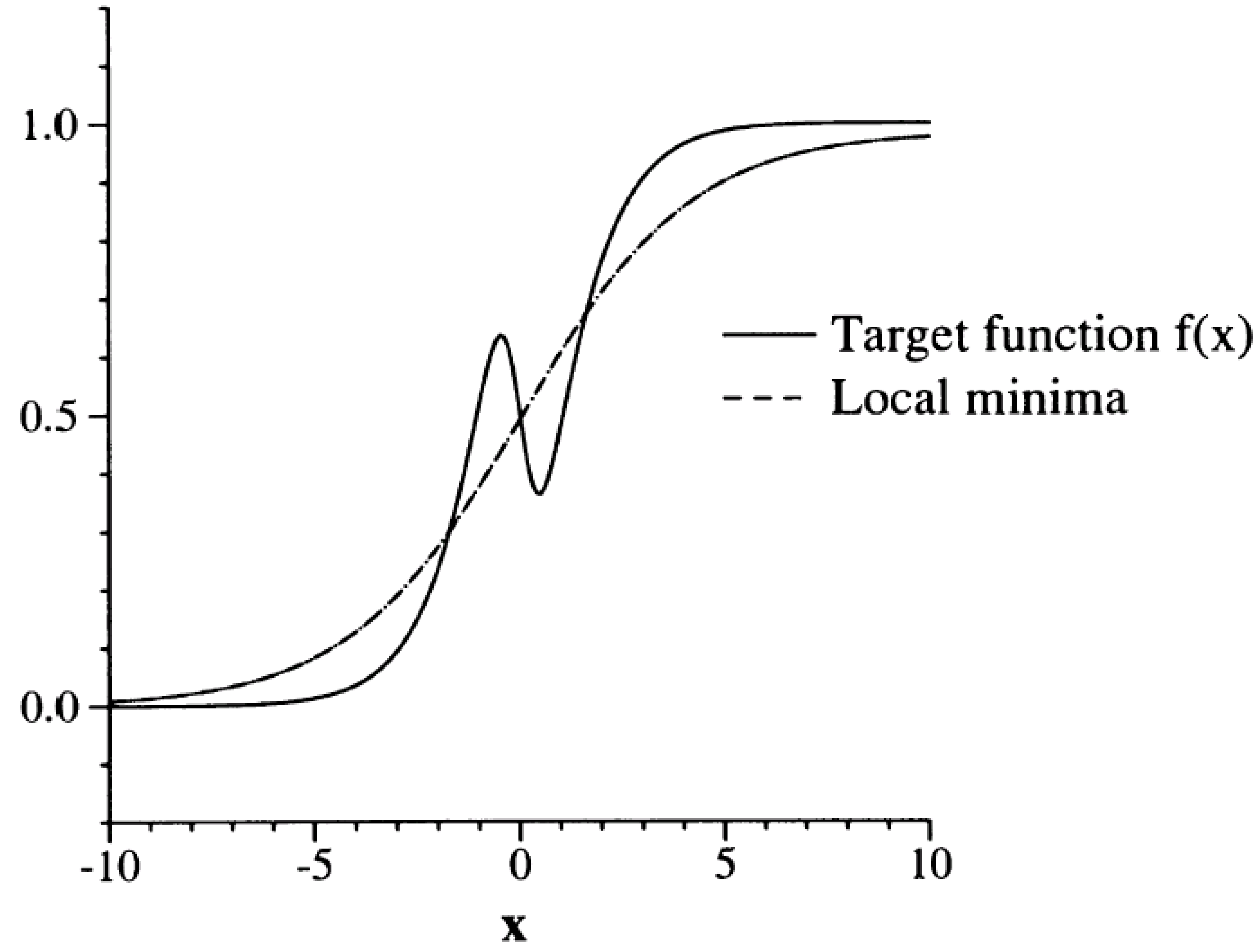


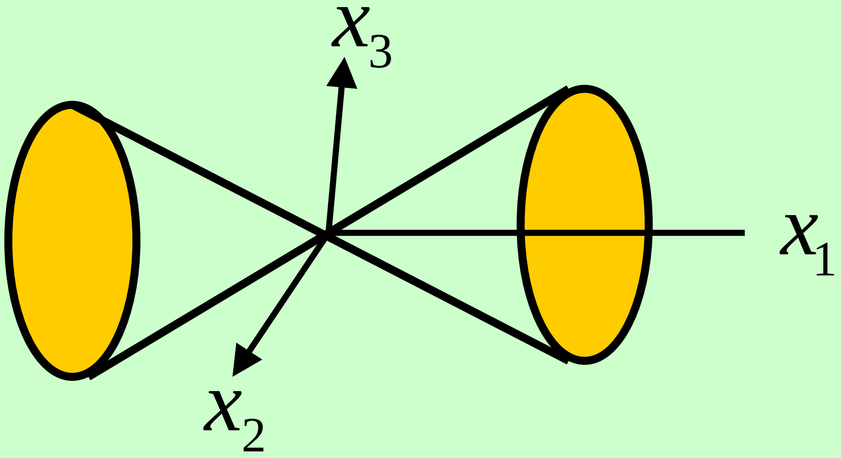
Fig. 6. A local minimum in MLP ($L = 1, H = 2$).

Random Gaussian Field (Cone Model)

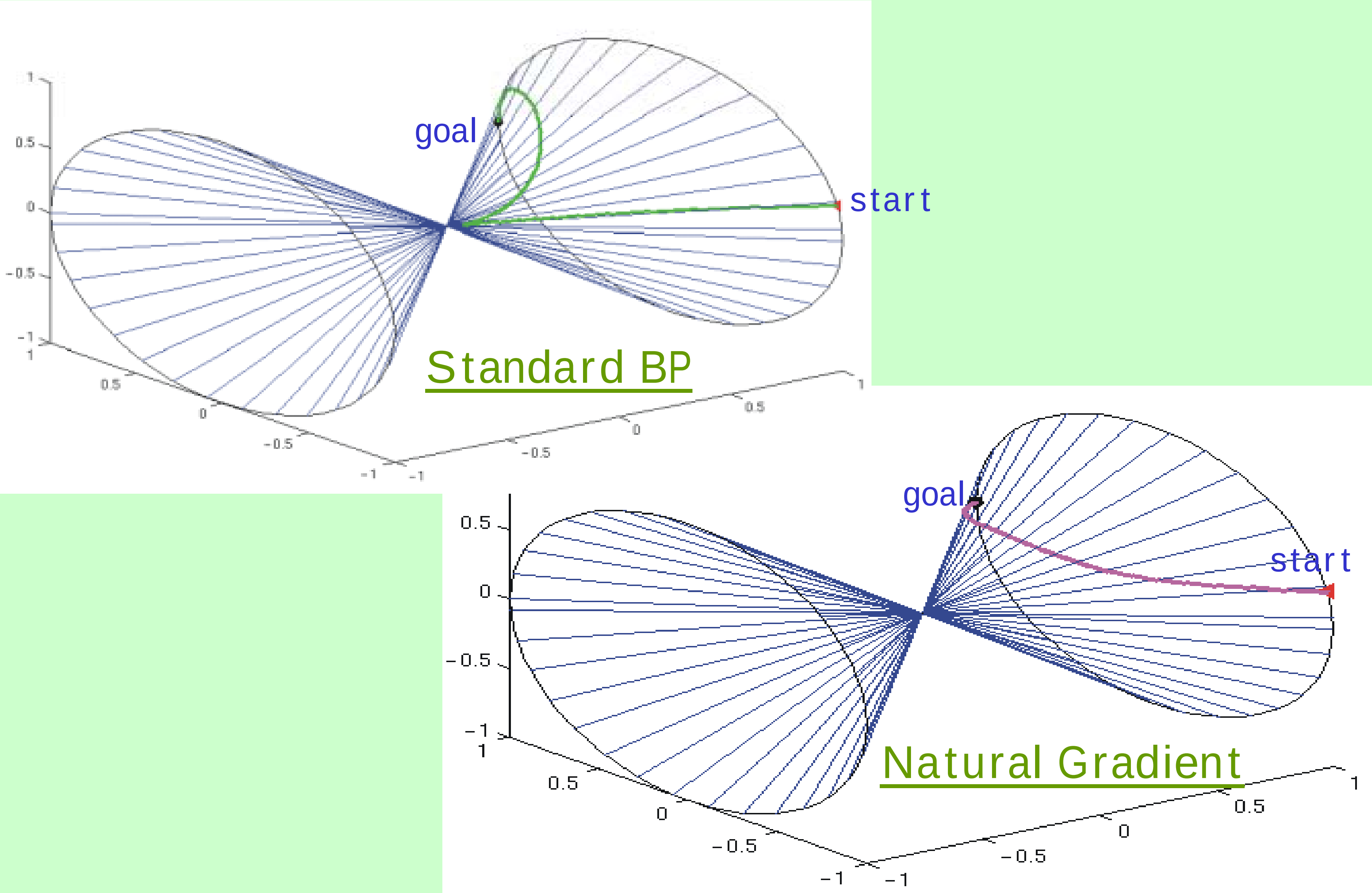
$$\boldsymbol{x} : \ N(\mu, I)$$

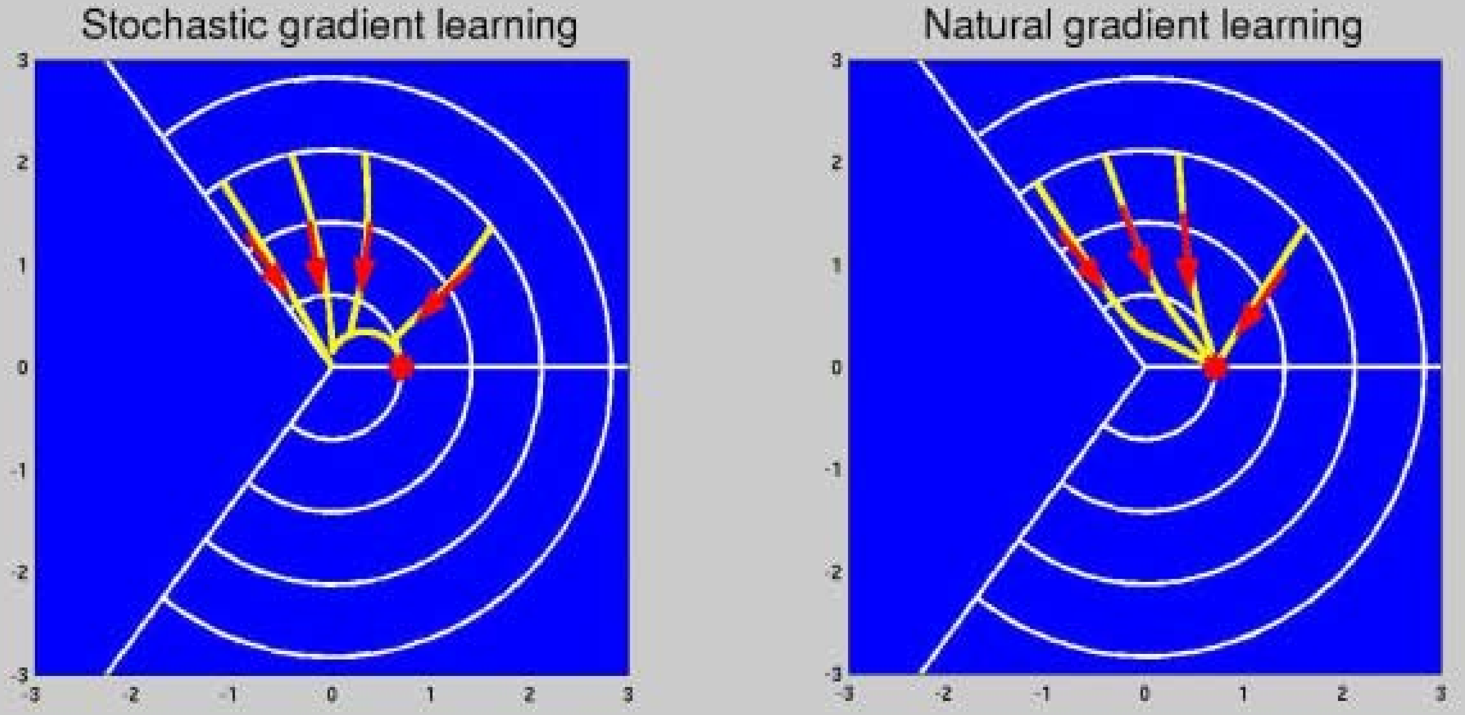
$$\mu = \xi \boldsymbol{a}(\omega), \qquad \boldsymbol{a}(\omega) = \frac{1}{\sqrt{1+c^2}} \begin{pmatrix} 1 \\ c\omega \end{pmatrix}$$

$$\omega \in S^d \qquad \omega = \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix}$$



Singularity and Learning Dynamics



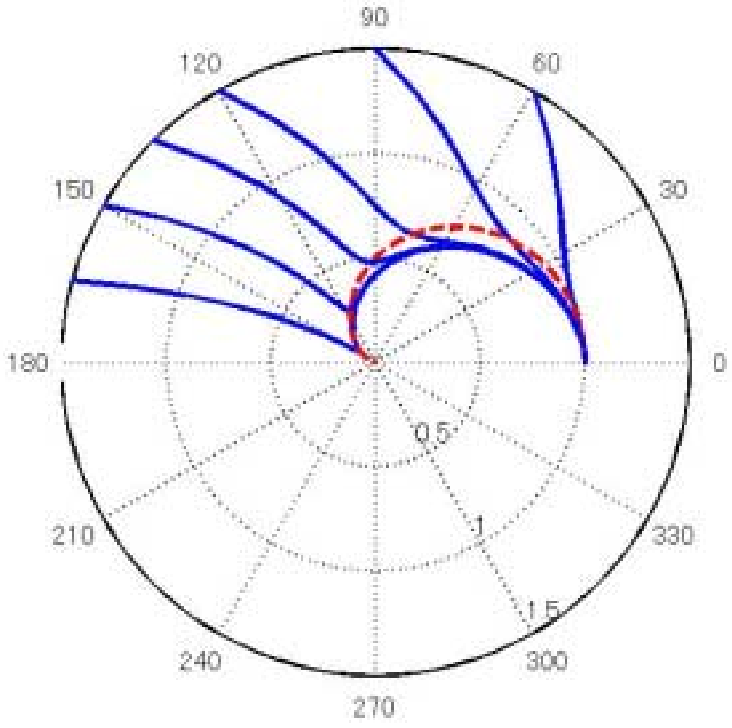


Stochastic gradient learning

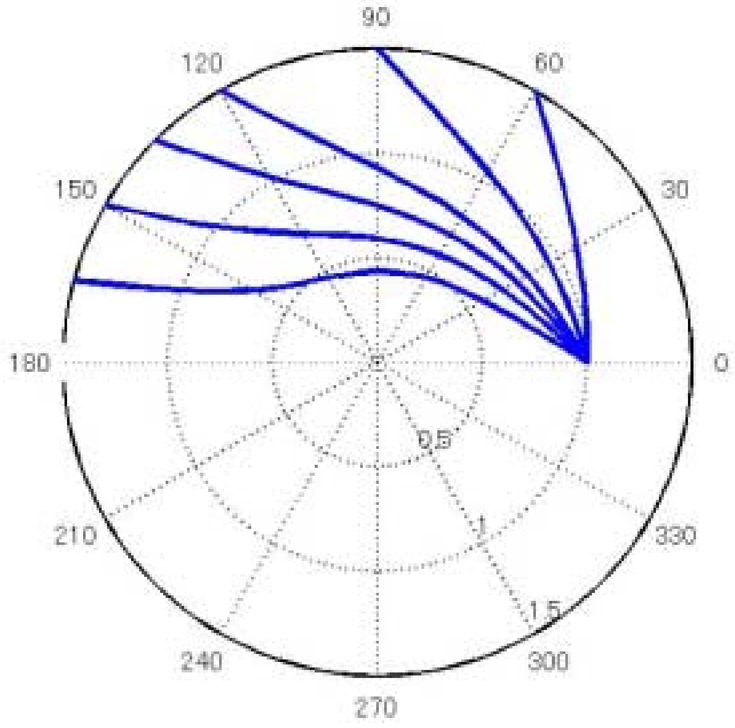
Natural gradient learning

Learning Trajectories for Cone Model

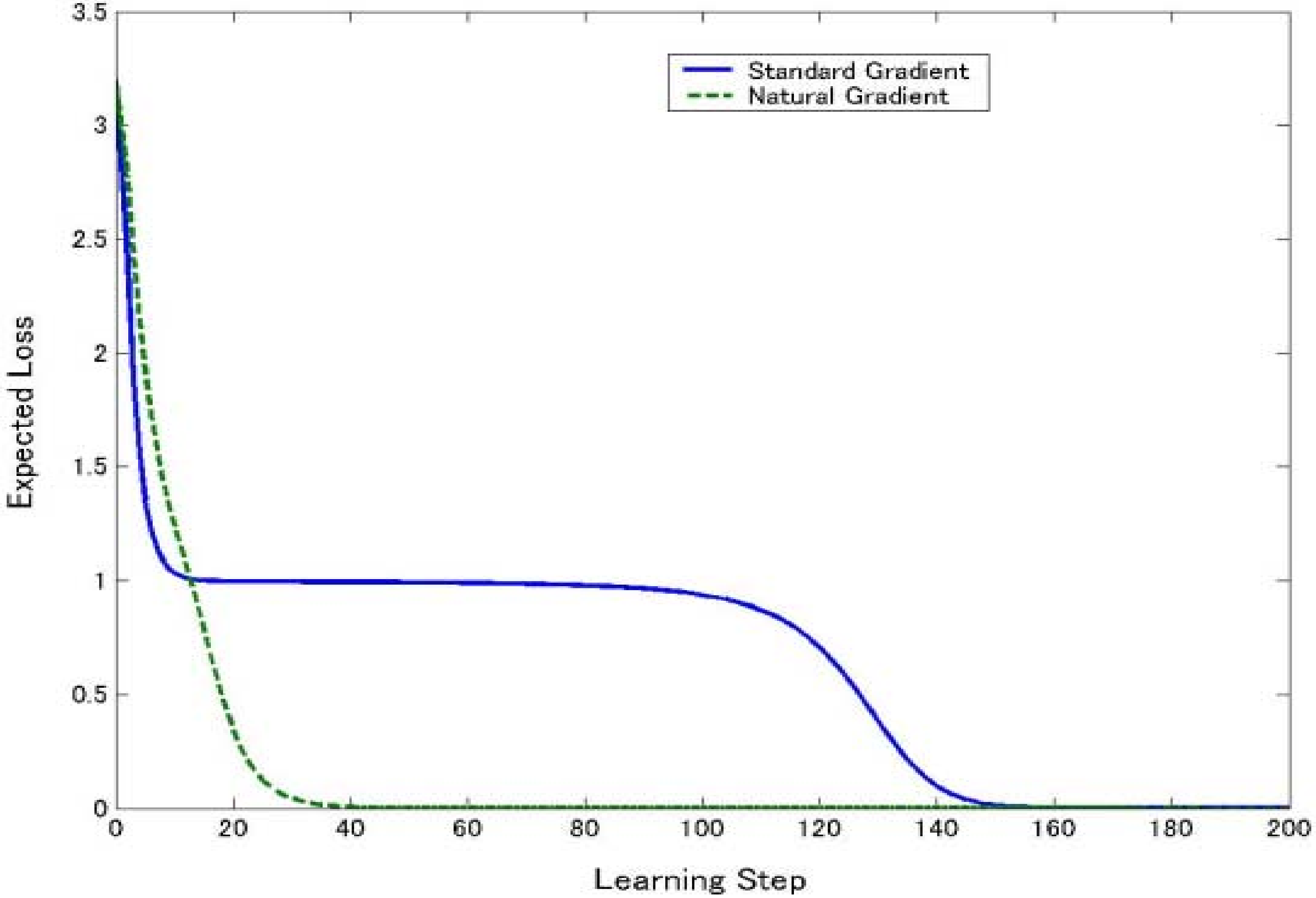
Standard Gradient



Natural Gradient

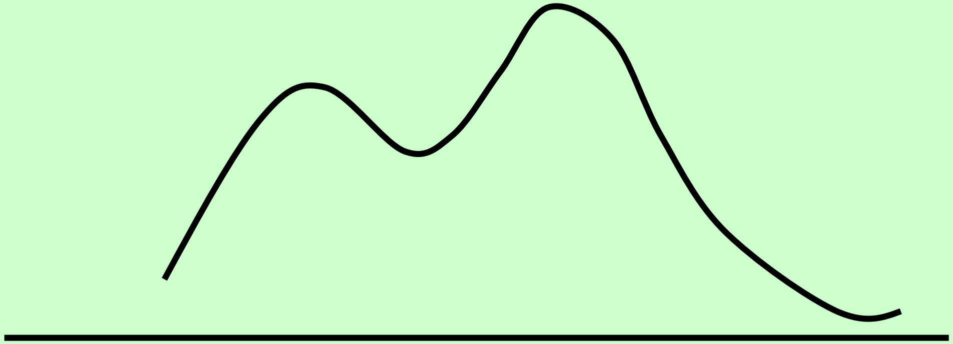


Learning Curves for Cone Model

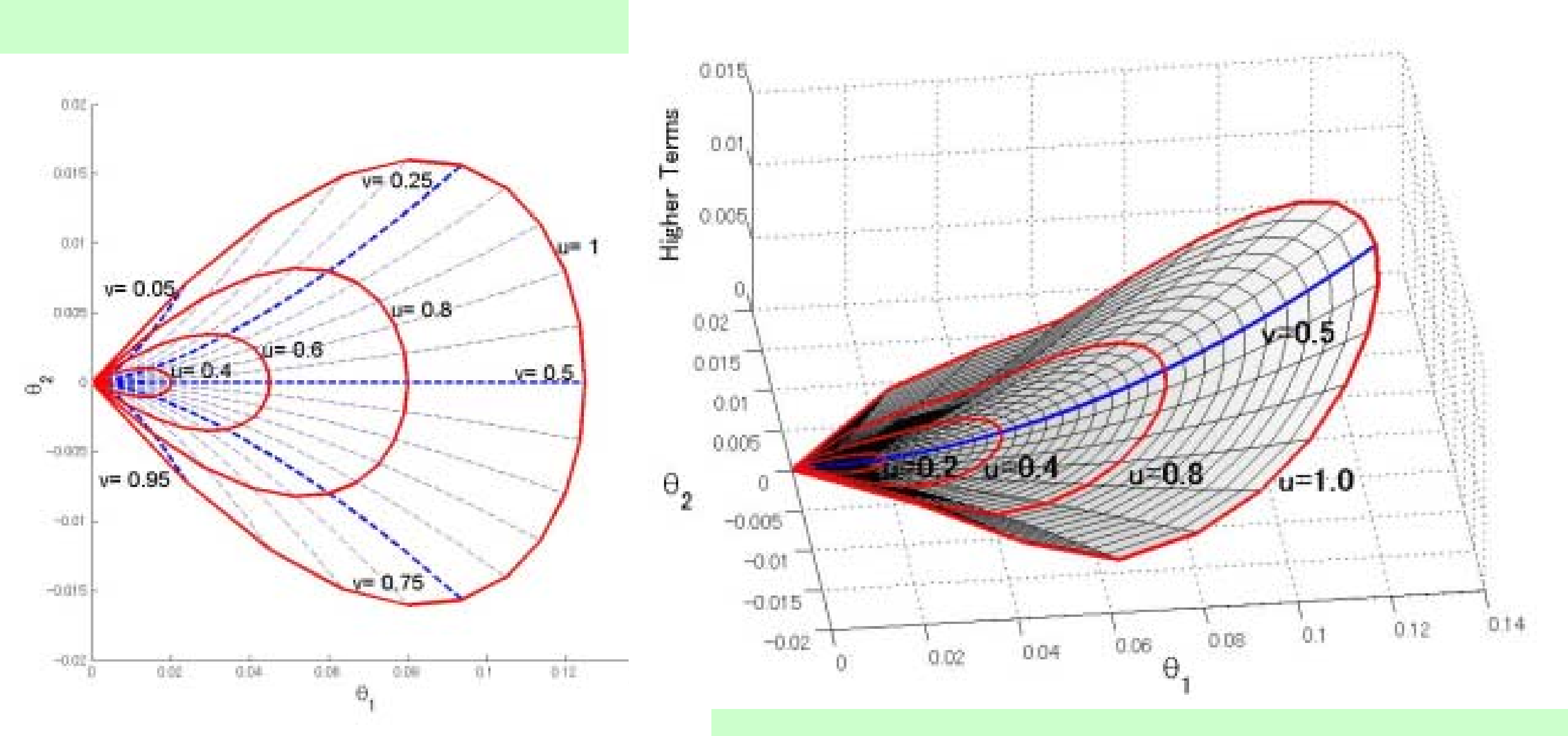


Gaussian mixtures

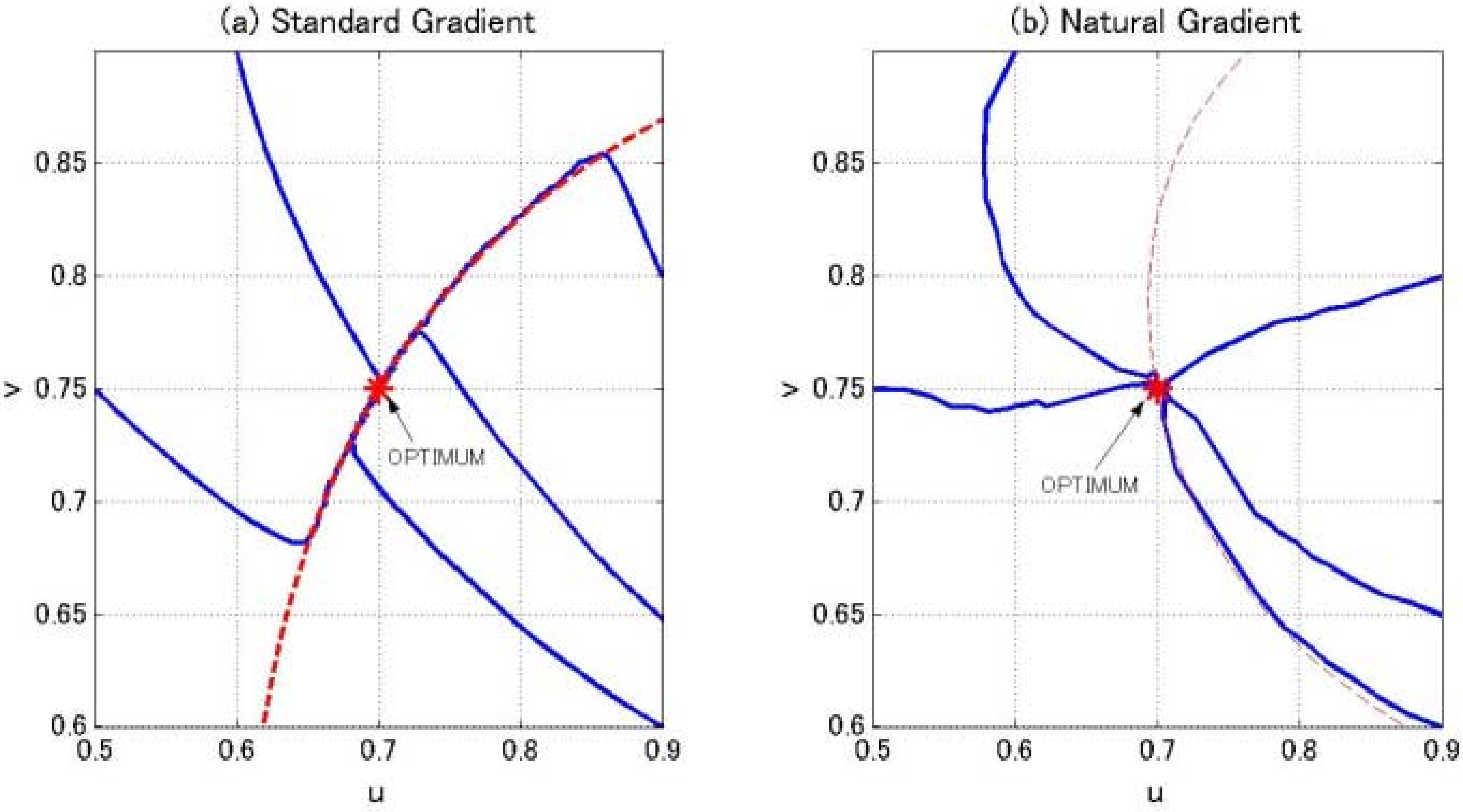
$$p(x) = \sum v_i \exp\left\{ -\frac{1}{2}(x - w_i)^2 \right\}$$



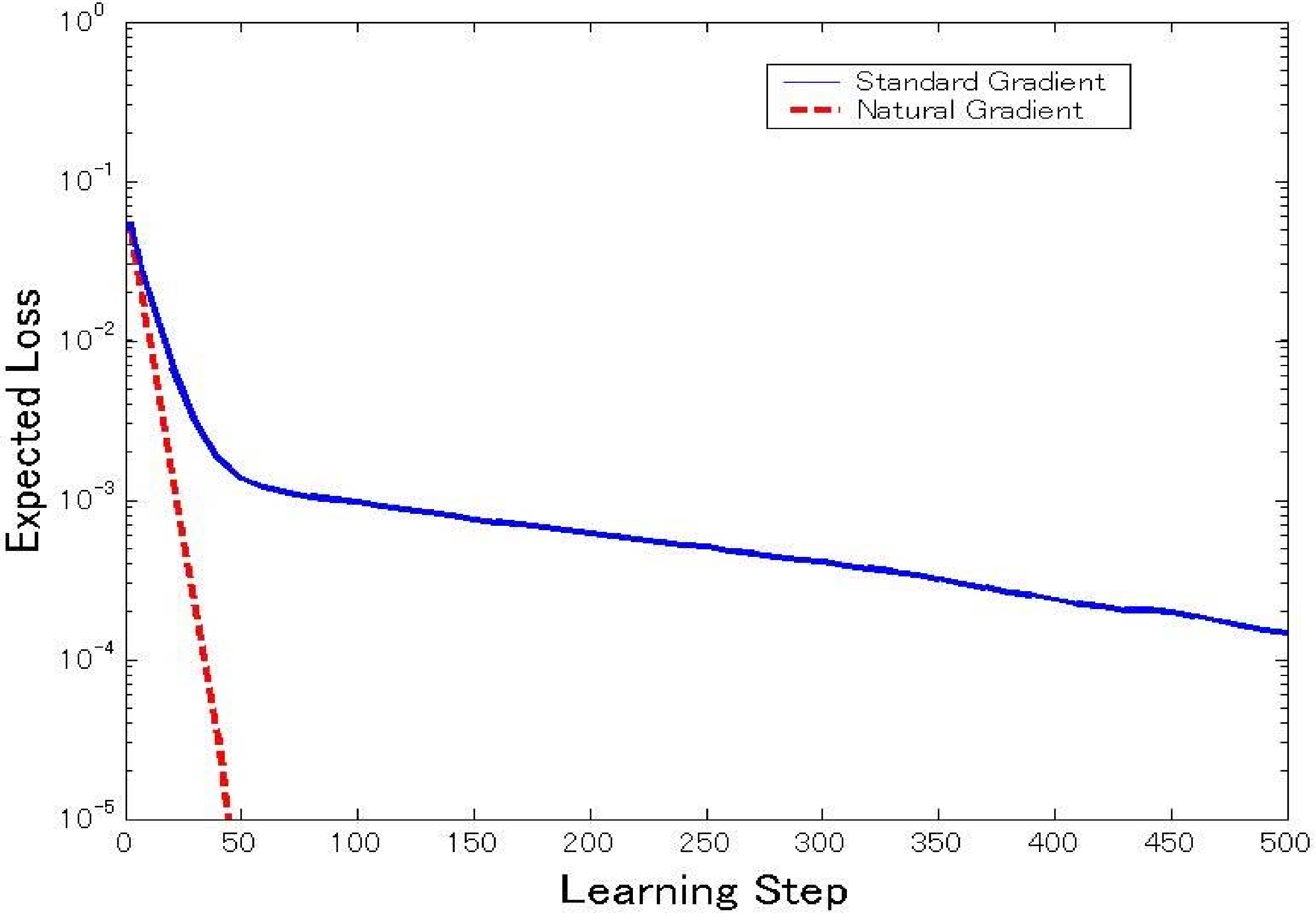
Singular structure of Gaussian mixture model



Learning Trajectories for Gaussian Mixture Model



Learning Curves for Gaussian Mixture Model



Simple model 1.

$$y = \xi \varphi(\boldsymbol{w} \cdot \boldsymbol{x}) + n$$

Simple model 2.

$$p(\boldsymbol{x}; \boldsymbol{\mu}) = c \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^2\right\}$$

$$\boldsymbol{\mu} = \xi \boldsymbol{a}(\omega) = \xi \frac{1}{1 + c^2}\begin{pmatrix} 1 \\ c\omega \end{pmatrix}$$

Regular statistical model

$$M = \left\{ p\left(x, \theta\right) \right\}$$

G : Fisher information

$$E\left[\Delta\theta\Delta\theta^{T} \right] = \frac{1}{n} G^{-1}$$

$$E\left[KL\left[p\left(x, \theta_0\right) : p\left(x, \hat{\theta}\right) \right] \right] \approx \frac{1}{2n} G \cdot E\left[\Delta\theta\Delta\theta\right]$$

$$\approx \frac{d}{2n}$$

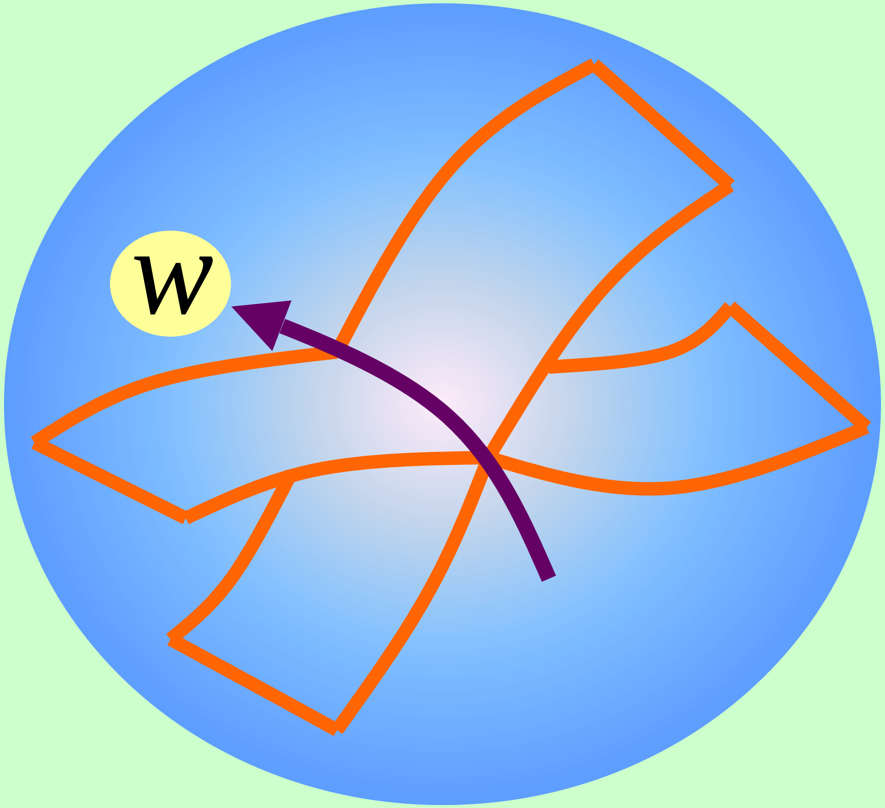
AIC, MDL

$$\Delta w \sim O(1)$$

$$\Delta u \sim O\left(\frac{1}{u^2}\right)$$

$$\Delta v \sim O\left(\frac{1}{u^3}\right)$$

$$\Delta x_i \sim O\left(\frac{1}{u^2}\right)$$



Singular Models

Gaussian mixture

$$p(x,\theta) = \sum v_i \varphi(x - w_i)$$

Multilayer perceptrons

$$y = \sum v_i \varphi(\boldsymbol{w}_i \cdot \boldsymbol{x}) + n$$

$$p(y|\boldsymbol{x};\theta) = \exp\left\{ -\frac{1}{2}\left(y - \sum v_i \varphi_i\right)^2 \right\}$$

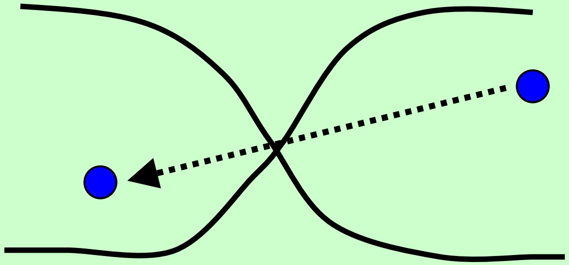
ARMA model

$$x_t = \frac{\sum b_i z^{-i}}{\sum a_i z^{-i}} \varepsilon_t$$

Learning, Estimation, and Model Selection

$$E_{\text{gen}} = D\left[p_0\left(y \middle| \boldsymbol{x}\right) : p\left(y \middle| \boldsymbol{x}; \hat{\boldsymbol{\theta}}\right)\right]$$

$$E_{\text{train}} = D\left[p_{\text{emp}}(y \mid \boldsymbol{x}) : p\left(y \middle| \boldsymbol{x}; \hat{\boldsymbol{\theta}}\right)\right]$$



d– log n, loglog n
--singular case

$$E_{\text{gen}} = \frac{d}{2n} \qquad d : \text{dimension}$$

AIC, MDL

$$E_{\text{gen}} = E_{\text{train}} + \frac{d}{n}$$

Model Selection

AIC = training error + d/N

MDL = training error + d logN/(2N)

Bayesian regularization

Bayesian and Reguarization
--algebraic geometry

posterior distribution

$$p(\theta \mid D) = \frac{\pi(\theta)\, p(D \mid \theta)}{p(D)}$$

prior distribution

-- uniform, smooth, Geffreys

predictive distribution