PATTERN RECOGNITION AND MACHINE LEARNING

CHAPTER 5: NEURAL NETWORKS

Learning Objectives

1、What are feed-forward network functions?

2、What are cost functions for network training?

3、What are error propagation schemes for neural networks?

4、How to compute Jacobian and Hessian matrices?

5、How to achieve regularization for neural networks?

6、What are Bayesian MAP neural networks?

7、What are convolution neural networks?

8、What are generative adversarial networks?

Outlines

- Feedforward Network Functions
- Network Training
- Error Backpropagation
- Jacobian Matrix
- Hessian Matrix
- Regularization
- MAP Neural Networks
- CNN and GAN

MAP Neural Networks for Regression I

$p(t|\boldsymbol{x},\boldsymbol{w},\beta) = N(t|y(\boldsymbol{x},\boldsymbol{w}),\beta^{-1})$ $\quad$ $p(\boldsymbol{w}) = N(\boldsymbol{w}|0,\alpha^{-1}I)$ $\quad$ $p(\boldsymbol{w}|t) \propto p(\boldsymbol{w})p(t|\boldsymbol{x},\boldsymbol{w},\beta)$

$$E(\boldsymbol{w}) = -\ln p(\boldsymbol{w}|\boldsymbol{t}) = \frac{\alpha}{2}\boldsymbol{w}^T\boldsymbol{w} + \frac{\beta}{2}\sum_{n=1}^{N}[y(\boldsymbol{x}_n,\boldsymbol{w}) - t_n]^2 + Constant$$

$$\nabla E(\boldsymbol{w}) = \alpha\boldsymbol{w} + \beta\sum_{n=1}^{N}(y_n - t_n)\boldsymbol{g}_n \qquad \boldsymbol{g} = \nabla_{\boldsymbol{w}} y(\boldsymbol{x},\boldsymbol{w})$$

$\boldsymbol{A} = \nabla\nabla E(\boldsymbol{w}) = \alpha\mathbf{I}+\beta\boldsymbol{H}$ $\quad$ $\boldsymbol{H}$: Hessian matrix of the sum-of-error function

$\boldsymbol{w}_{\mathrm{MAP}} \longleftarrow$ $\boldsymbol{w}^{new} = \boldsymbol{w}^{old} - \boldsymbol{A}^{-1}\nabla E(\boldsymbol{w})$ $\quad$ $q(\boldsymbol{w}) = N(\boldsymbol{w}|\boldsymbol{w}_{MAP},\boldsymbol{A}^{-1})$

MAP Neural Networks for Regression II

$$\mathrm{y}(\boldsymbol{x},\boldsymbol{w}) \simeq y(\boldsymbol{x},\boldsymbol{w}_{MAP}) + \boldsymbol{g}^{T}{}_{MAP}(\boldsymbol{w}-\boldsymbol{w}_{MAP})$$

$$p(t|\boldsymbol{x},\boldsymbol{w},\beta) = N(t|y(\boldsymbol{x},\boldsymbol{w}_{MAP}) + \boldsymbol{g}^{T}{}_{MAP}(\boldsymbol{w}-\boldsymbol{w}_{MAP}),\beta^{-1})$$

$$q(\boldsymbol{w}) = N(\boldsymbol{w}|\boldsymbol{w}_{MAP},\boldsymbol{A}^{-1})$$

Predictive Distribution Approximation

$$p(t|\boldsymbol{x},D,\alpha,\beta) = \int p(t|\boldsymbol{x},\boldsymbol{w},\beta)q(\boldsymbol{w})d\boldsymbol{w}$$

$$p(t|\boldsymbol{x},D,\alpha,\beta) = N(t|y(\boldsymbol{x},\boldsymbol{w}_{MAP}),\boldsymbol{g}^{T}{}_{MAP}\boldsymbol{A}^{-1}\boldsymbol{g}_{MAP}+\beta^{-1})$$

Hyper-parameter Optimization

$$\ln p(\mathbf{t}|\alpha, \beta) = \frac{M}{2}\ln\alpha + \frac{N}{2}\ln\beta - E(\mathbf{m}_N) + \frac{1}{2}\ln|\mathbf{S}_N| - \frac{N}{2}\ln(2\pi).$$

$$\alpha = \frac{\gamma}{\mathbf{w}_{\text{MAP}}^{\text{T}}\mathbf{w}_{\text{MAP}}} \qquad \beta\mathbf{H}\mathbf{u}_i = \lambda_i\mathbf{u}_i$$

$$\gamma = \sum_{i=1}^{W}\frac{\lambda_i}{\alpha + \lambda_i}$$

$$\frac{1}{\beta} = \frac{1}{N-\gamma}\sum_{n=1}^{N}\{y(\mathbf{x}_n, \mathbf{w}_{\text{MAP}}) - t_n\}^2$$

MAP Neural Networks for Classification I

$p(\boldsymbol{w}) = N(\boldsymbol{w}|0, \alpha^{-1}I) \quad p(\boldsymbol{w}|t) \propto p(\boldsymbol{w})p(t|\boldsymbol{w})$

$$E(w) = -\ln p(w|t) = \frac{\alpha}{2}\boldsymbol{w}^T\boldsymbol{w} - \sum_{n=1}^{N}[t_n \ln y_n + (1 - tn)\ln(1 - yn)]$$

$$\nabla E(w) = \alpha\boldsymbol{w} + \sum_{n=1}^{N}(y_n - t_n)\boldsymbol{g}_n$$

$\boldsymbol{A} = \nabla\nabla E(\boldsymbol{w}) = \alpha\boldsymbol{I} + \boldsymbol{H}$ $\quad$ $\boldsymbol{H}$: Hessian matrix of the cross-entropy function

$\boldsymbol{w}_{\text{MAP}} \longleftarrow \boldsymbol{w}^{new} = \boldsymbol{w}^{old} - \boldsymbol{A}^{-1}\nabla E(\boldsymbol{w}) \qquad q(\boldsymbol{w}) = N(\boldsymbol{w}|\boldsymbol{w}_{MAP}, A^{-1})$

MAP Neural Networks for Classification II

Predictive Distribution Approximation I (simplest)

$$p(t|\mathbf{x}, \mathcal{D}) = \int p(t|\mathbf{x}, \mathbf{w}) q(\mathbf{w}|\mathcal{D}) \, \mathrm{d}\mathbf{w}$$

$$p(t|\mathbf{x}, \mathcal{D}) \simeq p(t|\mathbf{x}, \mathbf{w}_{\mathrm{MAP}})$$

MAP Neural Networks for Classification III

Predictive Distribution Approximation II

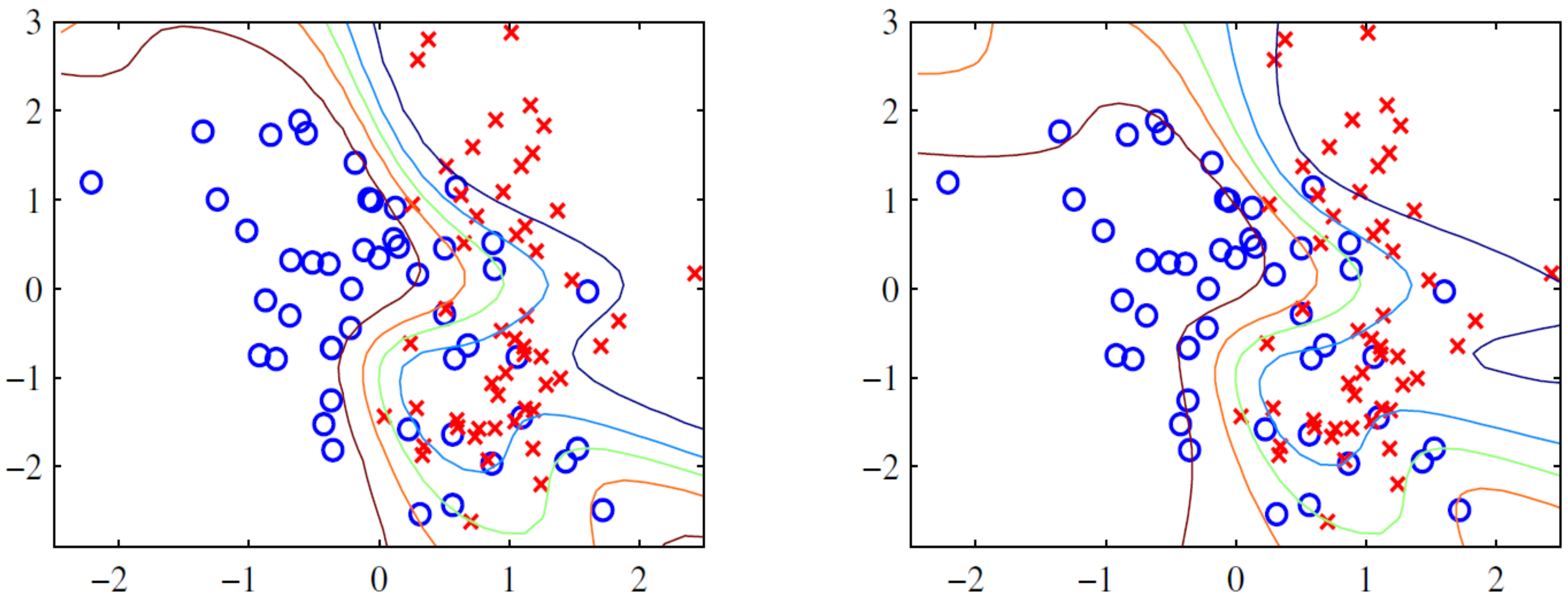
$$a(\mathbf{x}, \mathbf{w}) \simeq a_{\mathrm{MAP}}(\mathbf{x}) + \mathbf{b}^{\mathrm{T}}(\mathbf{w} - \mathbf{w}_{\mathrm{MAP}})$$

$$a_{\mathrm{MAP}}(\mathbf{x}) = a(\mathbf{x}, \mathbf{w}_{\mathrm{MAP}}) \qquad \mathbf{b} \equiv \nabla a(\mathbf{x}, \mathbf{w}_{\mathrm{MAP}})$$

$$\boxed{\begin{aligned} a_{\mathrm{MAP}} &\equiv a(\mathbf{x}, \mathbf{w}_{\mathrm{MAP}}) \\ \sigma_a^2(\mathbf{x}) &= \mathbf{b}^{\mathrm{T}}(\mathbf{x})\mathbf{A}^{-1}\mathbf{b}(\mathbf{x}) \end{aligned}}$$

$$p(\mathrm{C}_1|\mathbf{x}^{new}, \boldsymbol{t}) = \mathbb{E}[y] = \int \sigma(a)p(a|\mathbf{x}^{new}, \boldsymbol{t})da \simeq \sigma(\kappa(\sigma_a^2)a_{MAP})$$

MAP Neural Networks for Classification IV



Decision boundary using MAP neural networks

Model Evaluation for MAP NNs

$$p(\mathcal{D}) = \int p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{\theta}$$

Then, the evidence is given by

$$\ln p(\mathcal{D}) \simeq \ln p(\mathcal{D}|\boldsymbol{\theta}_{\mathrm{MAP}}) + \underbrace{\ln p(\boldsymbol{\theta}_{\mathrm{MAP}}) + \frac{M}{2}\ln(2\pi) - \frac{1}{2}\ln|\mathbf{A}|}_{\text{Occam factor}}$$

where

$$\mathbf{A} = -\nabla\nabla \ln p(\mathcal{D}|\boldsymbol{\theta}_{\mathrm{MAP}})p(\boldsymbol{\theta}_{\mathrm{MAP}}) = -\nabla\nabla \ln p(\boldsymbol{\theta}_{\mathrm{MAP}}|\mathcal{D})$$

Summary

- Feedforward Network Functions
- Network Training
- Error Backpropagation
- Jacobian and Hessian Matrices
- Network Regularization
- MAP Neural Networks
- CNN and GAN