T-61.182 Information Theory and Machine Learning

38. Introduction to Neural Networks
40. Capacity of a Single Neuron
41. Learning as Inference

Presented by Yang, Zhi-rong on 22, April 2004

Contents

- Introduction to Neural Networks
  - – Memories
  - – Terminology
- Capacity of a Single Neuron
  - – Neural network learning as communication
  - – The capacity of a single neuron
  - – Counting threshold functions
- Learning as Inference
  - – Neural network learning as inference
  - – Beyond optimization: making predictions
  - – Implementation by Monte Carlo method
  - – Implementation by Gaussian approximations

Memories

- Address-based memory scheme
  - not associative
  - not robust or fault-tolerant
  - not distributed
- Biological memory systems
  - content addressable
  - error-tolerant and robust
  - parallel and distributed

Terminology

- Architecture
- Activity rule
- Learning rule
- Supervised neural networks
- Unsupervised neural networks

NN learning as communication

1. Obtain adapted weights

$$\begin{array}{ccccc} & & \{t_n\}_{n=1}^N & & \\ & & \downarrow & & \\ \{\mathbf{x}_n\}_{n=1}^N & \longrightarrow & \text{Learning algorithm} & \longrightarrow & \mathbf{w} \end{array}$$

2. Communication

$$\{\mathbf{x}_n\}_{n=1}^N \longrightarrow \mathbf{w} \longrightarrow \{\hat{t}_n\}_{n=1}^N$$

The capacity of a single neuron

- General position

  Definition 1 *A set of points* $\{\mathbf{x}_n\}$ *in K-dimensional space are in general position if any subset of size* $\leq$ *K is linearly independent*

- The linear threshold function

$$y = f\left(\sum_{k=1}^{K} w_k x_k\right)$$

$$f(a) = \begin{cases} 1 & a > 0 \\ 0 & a \leq 0 \end{cases}$$

Counting threshold functions

Denote $T(N, K)$ the number of distinct threshold functions on N points n general position in K dimensions. In this section, the author try to derive a fomula for $T(N, K)$.
To start with, let us work out a few cases by hand.

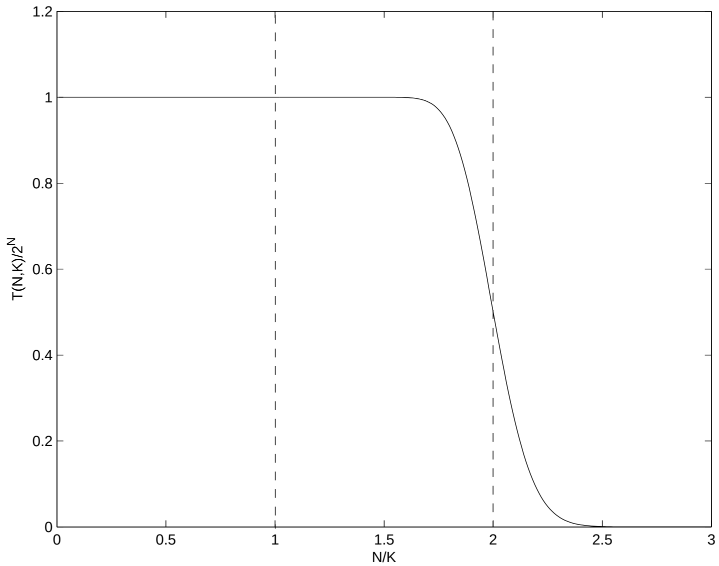
- $K = 1$, for any N
  $T(N, 1) = 2$
- $N = 1$, for any K
  $T(1, K) = 2$
- $K = 2$
  $T(N, 2) = 2N$
  The points of XOR function are unrealizable.

Counting threshold functions

- Final Result

$$T(N,K) = \begin{cases} 2^N & K \geq N \\ 2\sum_{k=0}^{K-1} \binom{N-1}{k} & K < N \end{cases}$$

- Vapnik-Chervonenkis dimension (VC dimension)



NN learning as inference

- Objective function to be minimized

$$M(\mathbf{w}) = G(\mathbf{w}) + \alpha E_W(\mathbf{w})$$

- with error function

$$G(\mathbf{w}) = -\sum_n \left[t^{(n)} \ln y(\mathbf{x}^{(n)}; \mathbf{w}) + (1 - t^{(n)}) \ln(1 - y(\mathbf{x}^{(n)}; \mathbf{w})) \right]$$

- and a regularizer

$$E_W(\mathbf{w}) = \frac{1}{2} \sum_i w_i^2$$

NN learning as inference

- Finally

$$P(\mathbf{w}|D,\alpha) = \frac{P(D|\mathbf{w})P(\mathbf{w}|\alpha)}{P(D|\alpha)} \quad (1)$$

$$= \frac{e^{G(\mathbf{w})}e^{-\alpha E_W(\mathbf{w})}/Z_W(\alpha)}{P(D|\alpha)} \quad (2)$$

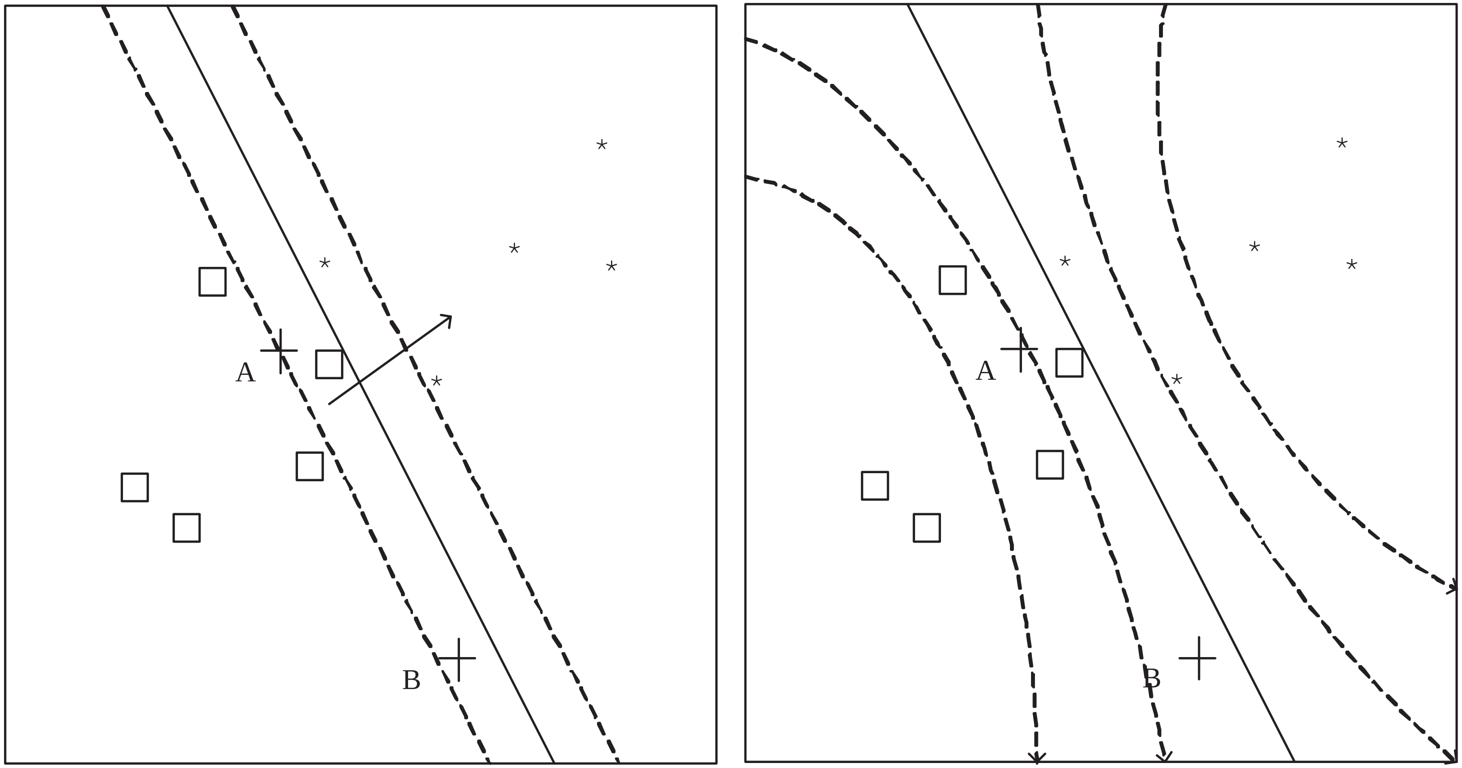
$$= \frac{1}{Z_M}\exp(-M(\mathbf{w})) \quad (3)$$

NN learning as inference

- Denote
$$y(\mathbf{w};\mathbf{x}) \equiv P(t=1|\mathbf{x},\mathbf{w})$$
- Then
$$P(t|\mathbf{x},\mathbf{w}) = y^t(1-y)^{1-t} = \exp[t\ln y + (1-t)\ln(1-y)]$$
- The likelihood can be expressed in terms of the error function
$$P(D|\mathbf{w}) = \exp[-G(\mathbf{w})]$$
- Similarly for the regularizer
$$P(\mathbf{w}|\alpha) = \frac{1}{Z_W(\alpha)}\exp(-\alpha E_W)$$

Making predictions

- Over-confident prediction (example)



Bayesian prediction: marginalizing

- Take into account the whole posterior ensemble

$$P(\mathbf{t}^{(N+1)}|\mathbf{x}^{(N+1)}, D, \alpha)$$
$$= \int \mathrm{d}^K\mathbf{w} P(\mathbf{t}^{(N+1)}|\mathbf{x}^{(N+1)}, \mathbf{w}, \alpha) P(\mathbf{w}|D, \alpha)$$

- Try to find a way of computing the integral

$$P(\mathbf{t}^{(N+1)} = 1|\mathbf{x}^{(N+1)}, D, \alpha)$$
$$= \int \mathrm{d}^K\mathbf{w} P(\mathbf{t}^{(N+1)}|\mathbf{x}^{(N+1)}, \mathbf{w}, \alpha) \frac{1}{Z_M} \exp(-M(\mathbf{w}))$$

The Langevin Monte Carlo Method

```
g = gradM(w); M = findM(w);
for l=1:L
  p = randn(size(w)); H = p'*p/2+M;

  p = p-epsilon*g/2;
  wnew = w+epsilon*p;
  gnew = gradM(wnew);
  p = p-epsilon*gnew/2;

  Mnew = findM(wnew); Hnew = p'*p/2+Mnew;
  dH = Hnew-H;
  if (dH<0||rand()<exp(-dH))
    g=gnew; w=wnew; M=Mnew;
endfor
```

The Langevin Monte Carlo Method

- 'gradient descent with added noise'

$$\Delta \mathbf{w} = -\frac{1}{2}\epsilon^2 \mathbf{g} + \epsilon \mathbf{p}$$

- speedup by Hamiltonian Monte Carlo

```
wnew=w; gnew=g;
for tau=1:Tau
  p = p-epsilon*gnew/2;
  wnew = wnew+epsilon*p;
  gnew = gradM(wnew);
  p = p-epsilon*gnew/2;
endfor
```

Gaussian approximations

- Taylor expand $M(\mathbf{w})$

$$M(\mathbf{w}) \simeq M(\mathbf{w}_{MP}) + \frac{1}{2}(\mathbf{w} - \mathbf{w}_{MP})^T \mathbf{A}(\mathbf{w} - \mathbf{w}_{MP}) + \cdots$$

- with Hessian matrix

$$A_{ij} \equiv \frac{\partial^2}{\partial w_i \partial w_j} M(\mathbf{w})\bigg|_{\mathbf{w}=\mathbf{w}_{MP}}$$

- The Gaussian approximation is defined as:

$$\begin{aligned} &Q(\mathbf{w}; \mathbf{w}_{MP}, \mathbf{A}) \\ &= [det(\mathbf{A}/2\pi)]^{1/2} \exp\left[-\tfrac{1}{2}(\mathbf{w} - \mathbf{w}_{MP})^T \mathbf{A}(\mathbf{w} - \mathbf{w}_{MP})\right] \end{aligned}$$

Gaussian approximations

- the second derivative of $M(\mathbf{w})$ with respect to $\mathbf{w}$ is given by

$$\frac{\partial^2}{\partial w_i \partial w_j} M(\mathbf{w}) = \sum_{n=1}^{N} f'(a^{(n)}) x_i^{(n)} x_j^{(n)} + \alpha \delta_{ij}$$

- where

$$f(a) \equiv \frac{1}{1 + e^{-a}}$$

$$a^{(n)} = \sum_j w_j x_j^{(n)}$$

Gaussian approximations

$$\begin{aligned} P(a|\mathbf{x}, D, \alpha) &= \mathrm{Normal}(a_{MP}, s^2) \\ &= \frac{1}{\sqrt{2\pi s^2}} \exp\left(-\frac{(a-a_{MP})^2}{2s^2}\right) \end{aligned}$$

where

$$a_{MP} = a(\mathbf{x}; \mathbf{w}_{MP})$$

and

$$s^2 = \mathbf{x}^T \mathbf{A}^{-1} \mathbf{x}$$

Gaussian approximations

- Therefore the marginalized output is:

$$P(t = 1|\mathbf{x}, D, \alpha) = \psi(a_{MP}, s^2) \equiv \int \mathrm{d}a f(a) \mathrm{Normal}(a_{MP}, s^2)$$

- And further approximation can be applied:

$$\psi(a_{MP}, s^2) \simeq \phi(a_{MP}, s^2) \equiv f(\kappa(s) a_{MP})$$

where

$$\kappa(s) = 1/\sqrt{1 + \pi s^2/8}$$

Exercises

- Practice on counting threshold functions: Ex. 40.6 (page 490)
- Prove the approximation on Hessian matrix: Ex. 41.1 (page 501)