大家好，这篇是有关Learning from data第七章习题的详解，这一章主要介绍了神经网络。

我的github地址：
https://github.com/Doraemonzzz

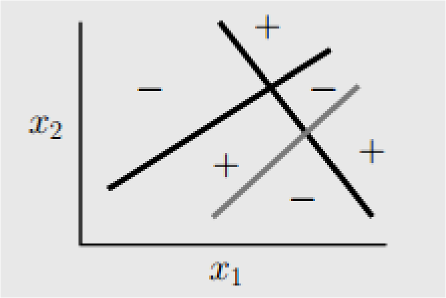
个人主页：
http://doraemonzzz.com/

参考资料:
https://blog.csdn.net/a1015553840/article/details/51085129
http://www.vynguyen.net/category/study/machine-learning/page/6/
http://book.caltech.edu/bookforum/index.php
http://beader.me/mlnotebook/

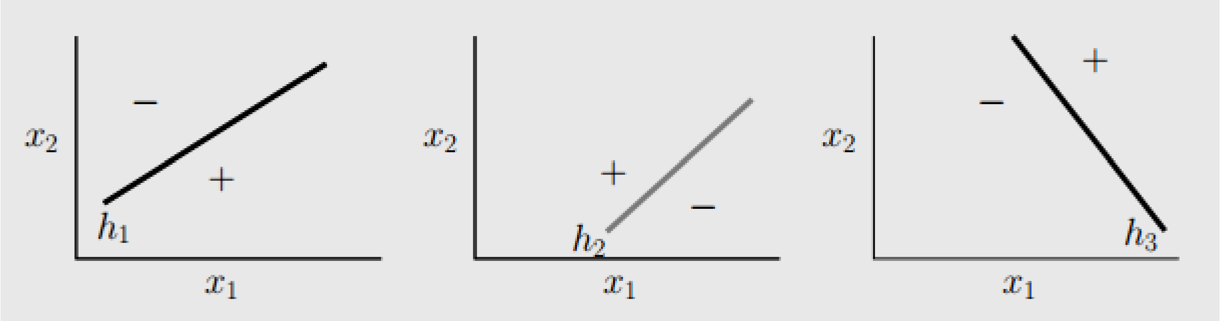
Part 1: Exercise

Exercise 7.1 (Page 2)

Consider a target function f whose '+' and '-' regions are illustrated below.



The target f has three perceptron components h_1, h_2, h_3:



Show that

$$f=\overline{h}_1h_2h_3+h_1\overline{h}_2h_3+h_1h_2\overline{h}_3$$

Is there a systematic way of going from a target which is a decomposition of perceptrons to a Boolean formula like this? [Hint: consider only the regions of f which are '+' and use the disjunctive normal form (or of ands).]

这个比较简单，直接验证即可，f有三个区域为+，上方的区域对应$\overline{h}_1h_2h_3$，左边区域对应$h_1\overline{h}_2h_3$，右边区域对应$h_1h_2\overline{h}_3$，这三者合在一起即为

$$f=\overline{h}_1h_2h_3+h_1\overline{h}_2h_3+h_1h_2\overline{h}_3$$

Exercise 7.2 (Page 3)

(a) The Boolean or and and of two inputs can be extended to more than two inputs: $\mathrm{OR}(x_1,\ldots,x_M)=+1$ if any one of the M inputs is $+1$; $\mathrm{AND}(x_1,\ldots,x_M)=+1$ if all the inputs equal $+1$. Give graph representations of $\mathrm{OR}(x_1,\ldots,x_M)$ and $\mathrm{AND}(x_1,\ldots,x_M)$.

(b) Give the graph representation of the perceptron: $h(x)=\mathrm{sign}(w^Tx)$.

(c) Give the graph representation of $\mathrm{OR}(x_1,\overline{x}_2,x_3)$.

(a)AND表示每个x_i都为$+1$时结果才为$+1$，可以如下构造

$$\mathrm{AND}(x_1,\ldots,x_M)=\mathrm{sign}(-M+\frac{1}{2}+\sum_{i=1}^M x_i)$$

OR表示至少存在一个x_i为$+1$时结果为$+1$，可以如下构造

$$\mathrm{OR}(x_1,\ldots,x_M)=\mathrm{sign}(M-\frac{1}{2}+\sum_{i=1}^M x_i)$$

(b)感知机的图像为一个超平面，二维情形为一条直线。

(c)图像为三维空间中的一个平面。

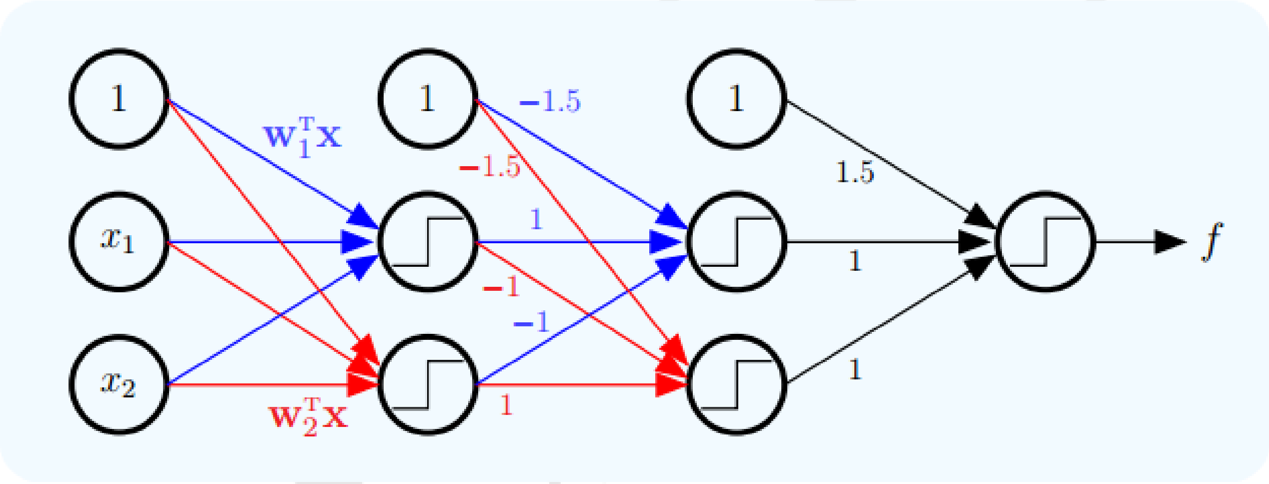
Exercise 7.3 (Page 4)

Use the graph representation to get an explicit formula for f and show that:

$$f(x)=\mathrm{sign}\Big[\mathrm{sign}(h_1(x)-h_2(x)-\frac{3}{2})-\mathrm{sign}(h_1(x)-h_2(x)+\frac{3}{2})+\frac{3}{2}\Big]$$

where $h_1(x)=\mathrm{sign}(w_1^Tx)$ and $h_2(x)=\mathrm{sign}(w_2^Tx)$

这个只是把下图用公式表达出来。


1
1
1
x_1
x_2
$\mathbf{w}_1^{\mathrm{T}}\mathbf{x}$
$\mathbf{w}_2^{\mathrm{T}}\mathbf{x}$
−1.5
−1.5
1
−1
−1
1
1.5
1
1
f

Exercise 7.4 (Page 5)

For the target function in Exercise 7.1, give the MLP in graphical form, as well as the explicit algebraic form.

先回顾f的形式

$$f=\overline{h}_1h_2h_3+h_1\overline{h}_2h_3+h_1h_2\overline{h}_3$$

图像形式电脑比较难画，这里略过，只给出代数形式，之前讨论过OR以及AND，所以这个不是很难，先考虑$\overline{h}_1h_2h_3$，这个是AND的形式，注意$\overline{h}_1$对应$-h_1$

$$f_1=\mathrm{sign}(-2.5-h_1+h_2+h_3)$$

同理$h_1\overline{h}_2h_3,h_1h_2\overline{h}_3$分别对应

$$f_2=\mathrm{sign}(-2.5+h_1-h_2+h_3),f_3=\mathrm{sign}(-2.5+h_1+h_2-h_3)$$

接下来处理OR，由之前讨论可知

$$f=\mathrm{sign}(2.5+f_1+f_2+f_3)$$

所以f可以表达为如下形式

$$f=\mathrm{sign}(2.5+f_1+f_2+f_3)\\f_1=\mathrm{sign}(-2.5-h_1+h_2+h_3),f_2=\mathrm{sign}(-2.5+h_1-h_2+h_3),f_3=\mathrm{sign}(-2.5+h_1+h_2-h_3)$$

Exercise 7.5 (Page 6)

Given w_1 and $\epsilon>0$, find w_2 such that $|\mathrm{sign}(w_1^Tx_n)-\tanh(w_2^Tx_n)|\le\epsilon$ for $x_n\in D$. [Hint: For large enough α, $\mathrm{sign}(x)\approx\tanh(\alpha x)$.]

先回顾$\tanh(x)$的定义

$$\tanh(x)=\frac{e^x-e^{-x}}{e^x+e^{-x}}$$

注意当$x \to +\infty$时，$\tanh(x) \to 1$，当$x \to -\infty$时，$\tanh(x) \to -1$，所以如果$x < 0$，$\text{sign}(x) = -1$，当α充分大时，那么$\tanh(\alpha x) \to -1$；如果$x > 0$，$\text{sign}(x) = 1$，当α充分大时，那么$\tanh(\alpha x) \to 1$，从而

$$\text{对于充分大的}\alpha\text{，}\text{sign}(x) \approx \tanh(\alpha x)$$

现在的目标是对与固定的w_1，找到w_2使得$|\text{sign}(w_1^T x_n) - \tanh(w_2^T x_n)| \le \epsilon$，从之前论述可以知道，只要取$w_2 = kw_1$，$k$是一个充分大的正数即可。

Exercise 7.6 (Page 10)

Let V and Q be the number of nodes and weights in the neural network,

$$V = \sum_{\ell=0}^{L} d^{(\ell)}, Q = \sum_{\ell=0}^{L} d^{(\ell)}(d^{(\ell-1)} + 1)$$

In terms of V and Q, how many computations are made in forward propagation (additions, multiplications and evaluations of θ).

[Answer: $O(Q)$ multiplications and additions, and $O(V)$ θ-evaluations.]

先验证下这两个等式的正确性。我们知道第ℓ层有$d^{(\ell)}$个节点，所以节点数量为

$$V = \sum_{\ell=0}^{L} d^{(\ell)}$$

第ℓ层到第$\ell + 1$层的权重数量为$(d^{(\ell)} + 1)d^{(\ell+1)}$，所以权重数量一共有

$$Q = \sum_{\ell=0}^{L} d^{(\ell)}(d^{(\ell-1)} + 1)$$

首先看下用使用多少次θ函数，由神经网络的定义我们知道，第0层之后每层都要使用θ函数，所以使用θ函数的数量为

$$V = \sum_{\ell=1}^{L} d^{(\ell)}$$

忽略第0层，这个数量可以近似为$O(V)$

接着看使用了多少次加法以及乘法，加法以及乘法发生的情形在如下计算中

$$s^{(\ell)} = (W^{(\ell)})^T x^{(\ell-1)}, W^{(\ell)} \in R^{(d^{(\ell-1)}+1) \times d^{(\ell)}}$$

由矩阵乘法的定义可知$(W^{(\ell)})^T x^{(\ell-1)}$一共发生了$(d^{(\ell-1)} + 1) \times d^{(\ell)}$次乘法，所以这个加法以及乘法的数量为$O(Q)$

Exercise 7.7 (Page 11)

For the sigmoidal perceptron, $h(x) = \tanh(w^T x)$, let the in-sample error be $E_{in}(w) = \frac{1}{N} \sum_{n=1}^{N} (\tanh(w^T x_n) - y_n)^2$. Show that

$$\nabla E_{in}(w)=\frac{2}{N}\sum_{n=1}^{N}(\tanh(w^Tx_n)-y_n)(1-\tanh^2(w^Tx_n))x_n$$

If $w\to\infty$, what happens to the gradient; how this is related to why it is hard to optimize the perceptron.

这题就是对$\tanh(x)$求偏导，回顾$\tanh(x)$的定义

$$\tanh(x)=\frac{e^x-e^{-x}}{e^x+e^{-x}}=2\frac{e^x}{e^x+e^{-x}}-1=2\frac{1}{1+e^{-2x}}-1$$

回忆$\theta(x)=\frac{1}{1+e^{-x}},\theta^{'}(x)=\theta(x)(1-\theta(x))$，所以上式可以化为

$$\tanh(x)=2\theta(2x)-1$$

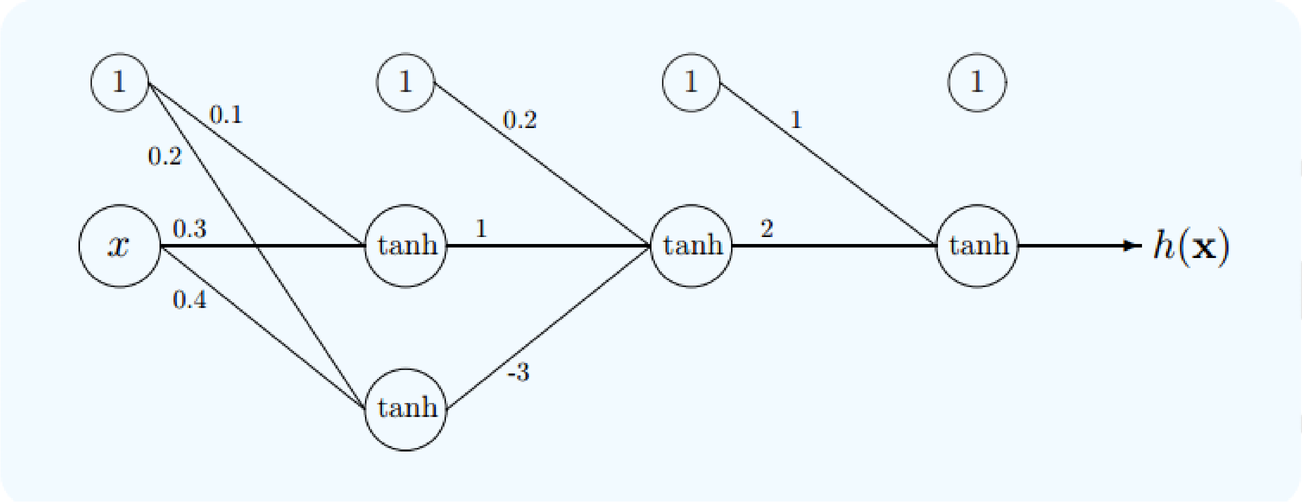
$$\frac{d\tanh(x)}{dx}=4\theta^{'}(2x)=4\theta(2x)(1-\theta(2x))=1-(2\theta(2x)-1)^2=1-\tanh^2(x)$$

由此可以计算梯度

$$\begin{aligned}\nabla E_{in}(w)&=\nabla\frac{1}{N}\sum_{n=1}^{N}(\tanh(w^Tx_n)-y_n)^2\\&=\frac{2}{N}\sum_{n=1}^{N}(\tanh(w^Tx_n)-y_n)\nabla\tanh(w^Tx_n)\\&=\frac{2}{N}\sum_{n=1}^{N}(\tanh(w^Tx_n)-y_n)(1-\tanh^2(w^Tx_n))\nabla(w^Tx_n)\\&=\frac{2}{N}\sum_{n=1}^{N}(\tanh(w^Tx_n)-y_n)(1-\tanh^2(w^Tx_n))x_n\end{aligned}$$

Exercise 7.8 (Page 15)

Repeat the computations in Example 7.1 for the case when the output transformation is the identity. You should compute $s^{(\ell)},x^{(\ell)},\delta^{(\ell)}$ and $\frac{\partial e}{\partial W^{(\ell)}}$



这个是课本上对应的图，我们这里需要求的是转换函数为$\theta(x)=x$的情形，先计算$s^{(\ell)},x^{(\ell)}$。

$$
s^{(1)} = \begin{bmatrix} 0.1 & 0.3 \\ 0.2 & 0.4 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 0.7 \\ 1 \end{bmatrix}, x^{(1)} = \begin{bmatrix} 1 \\ 0.7 \\ 1 \end{bmatrix}
$$

$$
s^{(2)} = \begin{bmatrix} 0.2 & 1 & -3 \end{bmatrix} \begin{bmatrix} 1 \\ 0.7 \\ 1 \end{bmatrix} = -2.1, x^{(2)} = \begin{bmatrix} 1 \\ -2.1 \end{bmatrix}
$$

$$
s^{(3)} = \begin{bmatrix} 1 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ -2.1 \end{bmatrix} = -3.2, x^{(3)} = -3.2
$$

接着计算$\delta^{(\ell)}$，回顾更新公式

$$
\delta_j^{(L)} = 2(x_j^{(L)} - y_j)\theta^{'}(s^L)
$$

$$
\delta_j^{(\ell)} = \theta^{'}(s_j^{(\ell)}) \sum_{k=1}^{d^{(\ell+1)}} w_{jk}^{(\ell+1)} \delta_k^{(\ell+1)} (\ell < L)
$$

这里$\theta^{'}(s_j^{(\ell)}) = 1$，所以更新公式简化为

$$
\delta_j^{(L)} = 2(x_j^{(L)} - y_j)
$$

$$
\delta_j^{(\ell)} = \sum_{k=1}^{d^{(\ell+1)}} w_{jk}^{(\ell+1)} \delta_k^{(\ell+1)}
$$

现在来计算$\delta^{(\ell)}$

$$
\delta^{(3)} = 2(-3.2 - 1) = -8.4
$$

$$
\delta^{(2)} = 2 \times \delta^{(3)} = -16.8
$$

$$
\delta^{(1)} = \begin{bmatrix} 1 \times (-16.8) \\ -3 \times (-16.8) \end{bmatrix} = \begin{bmatrix} -16.8 \\ 50.4 \end{bmatrix}
$$

最后计算$\frac{\partial e}{\partial W^{(\ell)}}$

$$
\frac{\partial e}{\partial W^{(1)}} = x^{(0)}(\delta^{(1)})^T = \begin{bmatrix} 0.3 \\ 0.4 \end{bmatrix} \begin{bmatrix} -16.8 & 50.4 \end{bmatrix} = \begin{bmatrix} -5.04 & 15.12 \\ 6.72 & 20.16 \end{bmatrix}
$$

$$
\frac{\partial e}{\partial W^{(2)}} = x^{(1)}(\delta^{(2)})^T = -16.8 \times \begin{bmatrix} 1 \\ 0.7 \\ 1 \end{bmatrix} = \begin{bmatrix} -16.8 \\ -11.823 \\ -16.8 \end{bmatrix}
$$

$$
\frac{\partial e}{\partial W^{(3)}} = x^{(2)}(\delta^{(3)})^T = -8.4 \times \begin{bmatrix} 1 \\ -2.1 \end{bmatrix} = \begin{bmatrix} -8.4 \\ 17.64 \end{bmatrix}
$$

Exercise 7.9 (Page 18)

What can go wrong if you just initialize all the weights to exactly zero?

首先来看更新公式

$$
s^{(\ell)} = (W^{(\ell)})^T x^{(\ell-1)}
$$

$$
x^{(\ell)} = \theta(s^{(\ell)})
$$

如果权重都为0，那么$s^{(\ell)} = 0$，对于$\theta(x) = \tanh(x), \theta(0) = 0$，所以$x^{(\ell)} = \theta(s^{(\ell)}) = 0$，这说明除了第0层的节点，每一个节点大小均为0。再看反向传播的公式

$$\frac{\partial e}{w_{ij}^{(\ell)}} = x_i^{(\ell-1)} \delta_j^{(\ell)}$$

$$\delta_j^{(\ell)} = \theta^{'}(s_j^{(\ell)}) \sum_{k=1}^{d^{(\ell+1)}} w_{jk}^{(\ell+1)} \delta_k^{(\ell+1)}$$

由于权重均为0，所以$\delta_j^{(\ell)} = 0, \frac{\partial e}{w_{ij}^{(\ell)}}$。结合以上两点可得，如果初始权重均为$0$，那么梯度均为$0$，从而更新量为$0$，从而更新之后每个节点依旧为$0$，这样就无法训练数据了，所以每个节点的初始值不能都取0

Exercise 7.10 (Page 20)

It is no surprise that adding nodes in the hidden layer gives the neural network more approximation ability, because you are adding more parameters. How many weight parameters are there in a neural network with architecture specified by $d = [d^{(0)}, d^{(1)}, \ldots, d^{(L)}]$, a vector giving the number of nodes in each layer? Evaluate your formula for a 2 hidden layer network with 10 hidden nodes in each hidden layer.

这题就是第6题，这里直接给出公式

$$Q = \sum_{\ell=0}^{L} d^{(\ell)}(d^{(\ell-1)} + 1)$$

Exercise 7.11 (Page 24)

For weight elimination, show that $\frac{\partial E_{\text{aug}}}{\partial w_{ij}^{(\ell)}} = \frac{\partial E_{\text{in}}}{\partial w_{ij}^{(\ell)}} + 2\frac{\lambda}{N}\frac{w_{ij}^{(\ell)}}{(1+(w_{ij}^{(\ell)})^2)^2}$. Argue that weight elimination shrinks small weights faster than large ones.

首先回顾E_{aug}

$$E_{\text{aug}}(w,\lambda) = E_{\text{in}}(w) + \frac{\lambda}{N}\sum_{\ell,i,j}\frac{(w_{ij}^{(\ell)})^2}{1+(w_{ij}^{(\ell)})^2}$$

对这个式子稍作变形

$$E_{\text{aug}}(w,\lambda) = E_{\text{in}}(w) + \frac{\lambda}{N}\sum_{\ell,i,j}\frac{(w_{ij}^{(\ell)})^2}{1+(w_{ij}^{(\ell)})^2} = E_{\text{in}}(w) + \frac{\lambda}{N}\sum_{\ell,i,j}\Big(1 - \frac{1}{1+(w_{ij}^{(\ell)})^2}\Big)$$

求偏导可得

$$\frac{\partial E_{\text{aug}}}{\partial w_{ij}^{(\ell)}} = \frac{\partial E_{\text{in}}}{\partial w_{ij}^{(\ell)}} + \frac{\lambda}{N}\frac{\partial \sum_{\ell,i,j}\Big(1 - \frac{1}{1+(w_{ij}^{(\ell)})^2}\Big)}{\partial w_{ij}^{(\ell)}} = \frac{\partial E_{\text{in}}}{\partial w_{ij}^{(\ell)}} + 2\frac{\lambda}{N}\frac{w_{ij}^{(\ell)}}{(1+(w_{ij}^{(\ell)})^2)^2}$$

对于较大的$w_{ij}^{(\ell)}$，$\frac{w_{ij}^{(\ell)}}{(1+(w_{ij}^{(\ell)})^2)^2} \to 0$，而对于较小的$w_{ij}^{(\ell)}$，$\frac{w_{ij}^{(\ell)}}{(1+(w_{ij}^{(\ell)})^2)^2} \to w_{ij}^{(\ell)}$，所以较小的$w_{ij}^{(\ell)}$对应的梯度更大一些。

Exercise 7.12 (Page 27)

Why does outputting w_{t^*} rather than training with all the data for t^* iterations not go against the wisdom that learning with more data is better.

[Hint: “More data is better” applies to a fixed model $(\mathcal{H}, A)$. Early stopping is model selection on a nested hypothesis sets $\mathcal{H}_1 \subset \mathcal{H}_2 \subset \cdots$ determined by $\mathcal{D}_{\text{train}}$. What happens if you were to use the full data $\mathcal{D}$?]

这里我们是选模型，所以和数据多少没有关系，只要模型固定的时候，数据越多，效果才越好。

Exercise 7.13 (Page 27)

Suppose you run gradient descent for 1000 iterations. You have 500 examples in $\mathcal{D}$, and you use 450 for $\mathcal{D}_{\text{train}}$ and 50 for $\mathcal{D}_{\text{val}}$. You output the weight from iteration 50, with $E_{\text{val}}(w_{50}) = 0.05$ and $E_{\text{train}}(w_{50}) = 0.04$.

(a) Is $E_{\text{val}}(w_{50}) = 0.05$ an unbiased estimate of $E_{\text{out}}(w_{50})$?

(b) Use the Hoeffding bound to get a bound for E_{out} using E_{val} plus an error bar. Your bound should hold with probability at least 0.1.

(c) Can you bound E_{out} using E_{train} or do you need more information?

(a)根据前面的知识可知

$$\mathbb{E}(E_{\text{val}}) = E_{\text{out}}$$

所以$E_{\text{val}}(w_{50}) = 0.05$是$E_{\text{out}}(w_{50})$的无偏估计。

(b)由Hoeffding不等式可知

$$\mathbb{P}\Big(E_{\text{out}} \le E_{\text{val}} + \sqrt{\frac{1}{2N}\ln\frac{2M}{\delta}}\Big) \ge 1-\delta$$

M为模型的数量，N为数据的数量，对于此题来说$1-\delta = 0.1, \delta = 0.9, N = 50$，因为迭代了$1000$次，所以$M = 1000$，从而

$$\sqrt{\frac{1}{2N}\ln\frac{2M}{\delta}} = \sqrt{\frac{1}{2\times 50}\ln\frac{2\times 1000}{0.9}} \approx 0.2776$$
$$E_{\text{out}} \le E_{\text{val}} + \sqrt{\frac{1}{2N}\ln\frac{2M}{\delta}} \approx 0.05 + 0.2776 = 0.3276$$

(c)需要更多的信息，因为使用Hoeffding不等式的前提为

$$\mathbb{E}(E_{\text{train}}) = E_{\text{out}}$$

但这个是不成立的。

Exercise 7.14 (Page 29)

Consider the error function $E(w)=(w-w^*)^TQ(w-w^*)$, where Q is an arbitrary positive definite matrix. Set $w=0$. Show that the gradient $\nabla E(w)=-2Qw^*$. What weights minimize $E(w)$. Does gradient descent move you in the direction of these optimal weights? Reconcile your answer with the claim in Chapter 3 that the gradient is the best direction in which to take a step. [Hint: How big was the step?]

原题中$\nabla E(w)=-Qw^*$，但感觉这里应该是$\nabla E(w)=-2Qw^*$

由梯度计算公式可知

$$\nabla E(w)=2Q(w-w^*)$$

当$w=0$时

$$\nabla E(w)=-2Qw^*$$

由于Q为正定矩阵，所以$E(w)=(w-w^*)^TQ(w-w^*)\ge 0$，并且当$w=w^*$时取最小值。

如果我们的初始值为$w=0$,那么负梯度的方向为$-\nabla E(w)=2Qw^*$，这个方向并不是最优解w^*的方向。

Exercise 7.15 (Page 32)

Show that $|\eta_3-\eta_1|$ decreases exponentially in the bisection algorithm. [Hint: show that two iterations at least halve the interval size.]

如果$E(\overline{\eta})>E(\eta_2)$，那么$\{\overline{\eta},\eta_2,\eta_3\}$为新的U-arrangement，从而新的区间长度为

$$\Big|\eta_3-\overline{\eta}\Big|=\Big|\eta_3-\frac{\eta_1+\eta_3}{2}\Big|=\frac{1}{2}\Big|\eta_1-\eta_3\Big|$$

如果$E(\overline{\eta})<E(\eta_2)$，那么$\{\eta_1,\overline{\eta},\eta_2\}$为新的U-arrangement，如果再做一次更新，那么新的中点为$\eta^*=\frac{\eta_1+\eta_2}{2}$，所以第二次的U-arrangement或者为$\{\eta_1,\overline{\eta},\eta^*\}$，此时第二次的区间长度为

$$\Big|\eta_1-\eta^*\Big|=\Big|\eta_1-\frac{\eta_1+\eta_2}{2}\Big|=\frac{1}{2}\Big|\eta_1-\eta_2\Big|<\frac{1}{2}\Big|\eta_1-\eta_3\Big|$$

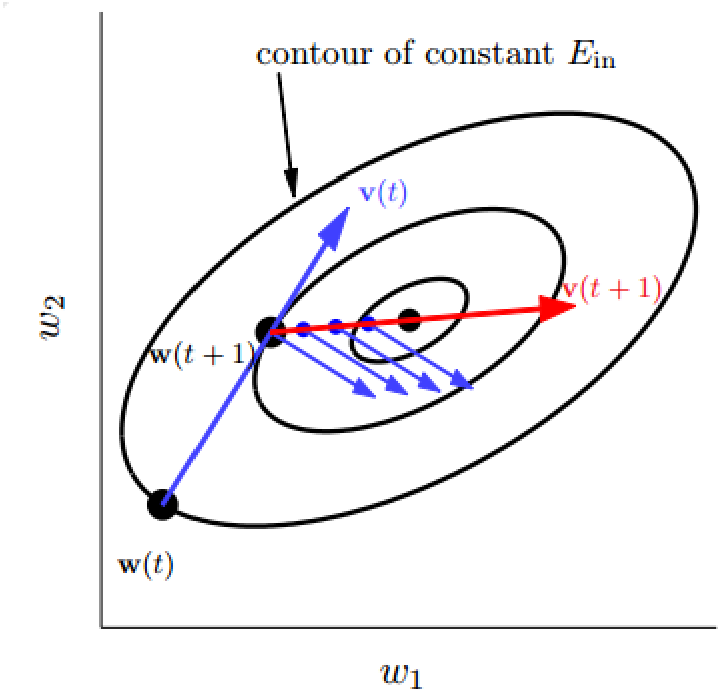
或者为$\{\overline{\eta},\eta^*,\eta_3\}$，此时第二次的区间长度为

$$\Big|\eta_3-\overline{\eta}\Big|=\Big|\eta_3-\frac{\eta_1+\eta_3}{2}\Big|=\frac{1}{2}\Big|\eta_1-\eta_3\Big|$$

所以，每两次更新之后，区间长度至少缩短为原来的$\frac{1}{2}$

Exercise 7.16 (Page 34)

Why does the new search direction pass through the optimal weights?



如图考虑二维情形，共轭梯度法的意思是，假设已经找到一个局部最优方向$w(t+1)$，那么某个方向的梯度为0，现在只要使得与其正交的方向的梯度变为0，所以要沿着这个正交方向减少梯度即可。我们现在新的搜索方向就是保证每处的梯度与$w(t+1)$的负梯度方向相同，从而一直在减少正交方向的梯度，最终可以达到梯度完全为0的点，得到最优权重。

Exercise 7.17 (Page 37)

The basic shape ϕ_3 is in both the '1' and the '5'. What other digits do you expect to contain each basic shape $\phi_1 \cdots \phi_6$. How would you select additional basic shapes if you wanted to distinguish between all the digits. (What properties should useful basic shapes satisfy?)

这题就是根据数字的形状选特征，略过。

Exercise 7.18 (Page 38)

Since the input x is an image it is convenient to represent it as a matrix $[x_{ij}]$ of its pixels which are black ($x_{ij}=1$) or white ($x_{ij}=0$). The basic shape ϕ_k identifies a set of these pixels which are black.

(a) Show that feature ϕ_k can be computed by the neural network node

$$\phi_k(x) = \tanh\Big(w_0 + \sum_{ij} w_{ij}x_{ij}\Big)$$

(b) What are the inputs to the neural network node?

(c) What do you choose as values for the weights? [Hint: consider separately the weights of the pixels for those $x_{ij} \in \phi_k$ and those $x_{ij} \notin \phi_k$.]

(d) How would you choose w_0? (Not all digits are written identically, and so a basic shape may not always be exactly represented in the image.)

(e) Draw the final network, filling in as many details as you can.

(a)因为w_{ij}为参数，所以只要构造特殊的w，必然能满足$\phi_k(x)$为指定的结果。

(b)输入数据为图像对应的$0,1$向量

(c)如果输出值为1，那么只要按如下方式赋值即可

$$w_{ij}=1,x_{ij}\in\phi_k,w_{ij}=0,x_{ij}\notin\phi_k$$

如果输出值为-1，那么只要按如下方式赋值即可

$$w_{ij}=-1,x_{ij}\in\phi_k,w_{ij}=0,x_{ij}\notin\phi_k$$

(d)我的理解是如果输入为一张空白的图片，那么应该不做任何判断，输出为0，从而

$$\phi_k(0)=\tanh(w_0)=0\\w_0=0$$

(e)略过

Exercise 7.19 (Page 41)

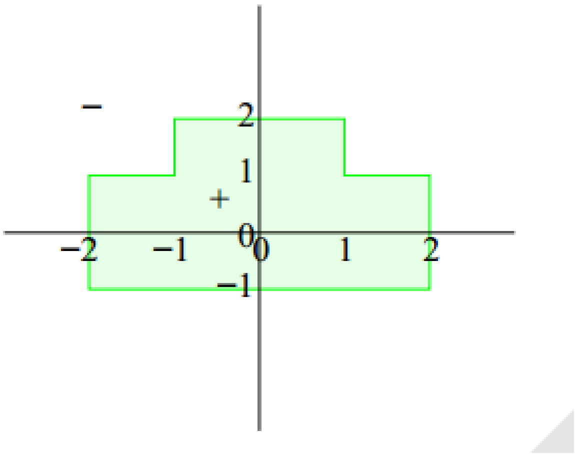
Previously, for our digit problem, we used symmetry and intensity. How do these features relate to deep networks? Do we still need them?

由深度学习的讨论可知，我们之前选择的特征相当于auto-encoder，所以我们仍然需要这些特征。

Part 2: Problems

Problem 7.1 (Page 43)

Implement the decision function below using a 3-layer perceptron.



第一层的神经元为感知机，效果是在边界处画一条直线

$$\text{水平方向：}\ s_1^{(1)}=\text{sign}(y-2),s_2^{(1)}=\text{sign}(y-1),s_3^{(1)}=\text{sign}(y+1)$$
$$\text{竖直方向：}\ s_4^{(1)}=\text{sign}(x-2),s_5^{(1)}=\text{sign}(x-1),s_6^{(1)}=\text{sign}(x+1),,s_7^{(1)}=\text{sign}(x+2)$$

第二层的神经元是产生上下两个矩形的效果，例如产生$x \in [-2,+2], y \in [-1,+1]$这样的区间，根据之前讨论的$\text{AND}$表达形式可得

$$
\begin{aligned}
x \in [-2,+2], y \in [-1,+1] : s_1^{(2)} &= \text{sign}(-3.5 - s_4^{(1)} + s_7^{(1)} - s_2^{(1)} + s_3^{(1)}) \\
x \in [-1,+1], y \in [+1,+2] : s_2^{(2)} &= \text{sign}(-3.5 - s_5^{(1)} + s_6^{(1)} - s_1^{(1)} + s_2^{(1)})
\end{aligned}
$$

第三层把两个区域合并为一个区域，利用OR对应的公式可得

$$
z = \text{sign}(1.5 - s_1^{(2)} - s_2^{(2)})
$$

Problem 7.2 (Page 43)

A set of M hyperplanes will generally divide the space into some number of regions. Every point in $\mathbb{R}^d$ can be labeled with an M dimensional vector that determines which side of each plane it is on. Thus, for example if $M=3$, then a point with a vector $(-1,+1,+1)$ is on the -1 side of the first hyperplane, and on the $+1$ side of the second and third hyperplanes. A region is defined as the set of points with the same label.

(a) Prove that the regions with the same label are convex.

(b) Prove that M hyperplanes can create at most 2^M distinct regions.

(c) [hard] What is the maximum number of regions created by M hyperplanes in d dimensions?

[Answer: $\sum_{i=0}^{d}\binom{M}{i}$]

[Hint: Use induction and let $B(M,d)$ be the number of regions created by M $(d-1)$-dimensional hyperplanes in d-space. Now consider adding the $(M+1)$th hyperplane. Show that this hyperplane intersects at most $B(M,d-1)$ of the $B(M,d)$ regions. For each region it intersects, it adds exactly one region, and so $B(M+1,d)$ ≤ $B(M,d)$ + $B(M,d-1)$. (Is this recurrence familiar?) Evaluate the boundary conditions: $B(M,1)$ and $B(1,d)$, and proceed from there. To see that the $M+1$th hyperplane only intersects $B(M,d-1)$ regions, argue as follows. Treat the $M+1$th hyperplane as a $(d-1)$-dimensional space, and project the initial M hy perplanes into this space to get M hyperplanes in a $(d-1)$-dimensional space. These M hyperplanes can create at most $B(M,d-1)$ regions in this space. Argue that this means that the $M+1$th hyperplane is only intersecting at most $B(M,d-1)$ of the regions created by the M hyperplanes in d-space.]

(a)只要利用以下这个简单的事实即可

$$
\begin{aligned}
\text{如果}\,\text{sign}(w^Tx_1) = \text{sign}(w^Tx_2) = i\text{，那么}\,\text{sign}(w^T(\lambda x_1 + (1-\lambda)x_2)) = i \\
\text{其中}\,\lambda \in (0,1)
\end{aligned}
$$

这里证明一下这个结论

$$
\begin{aligned}
\text{不妨设}\,i = 1; i = -1\text{时同理，} \\
\text{因为}\,\text{sign}(w^Tx_1) = \text{sign}(w^Tx_2) = 1 > 0\text{，} \\
\text{所以}\,w^Tx_1 > 0, w^Tx_2 > 0, \\
\text{注意}\,\lambda \in (0,1)\text{，所以}\,w^T(\lambda x_1 + (1-\lambda)x_2) = \lambda w^Tx_1 + (1-\lambda)w^Tx_2 > 0\text{，} \\
\text{因此}\,\text{sign}(w^T(\lambda x_1 + (1-\lambda)x_2)) = 1 = \text{sign}(w^Tx_1) = \text{sign}(w^Tx_2)
\end{aligned}
$$

现在证明题目中的结论，假设区域内有两个点$z_1, z_2 \in \mathbb{R}^d$，那么对于每个超平面对应的法向量$w_i$，

$$\text{sign}(w_i^Tz_1)=\text{sign}(w_i^Tz_2)$$

所以对于$\lambda\in(0,1)$

$$\text{sign}(w_i^T(\lambda z_1+(1-\lambda)z_2))=\text{sign}(w_i^Tz_1)=\text{sign}(w_i^Tz_2)$$

因为每个w_i都满足这个条件，所以$\lambda z_1+(1-\lambda)z_2$对应的$M$维向量与$z_1,z_2$对应的$M$维向量都相等，从而$M$维向量对应的区域都是凸的。

(b)M个超平面对应一个M维向量，每个分量属于$\{-1,+1\}$，所以最多有2^M个独立的区域

(c)令$B(M,d)$为d维空间中M个$d-1$维超平面划分的区域，考虑$B(M+1,d)$与$B(M,d)$的递推关系。假设有M个$d-1$维超平面，那么它最多划分$B(M,d)$个区域，现在增加一个$d-1$维超平面，那么原有的区域数量不变，考虑这个平面带来的增量，将原有的M个$d-1$维超平面投影到第$M+1$个超平面上，那么在这个$d-1$维超平面上最多新增$B(M,d-1)$个区域，每个$d-1$维区域对应着一个d维的区域，所以原区域中最多增加$B(M,d-1)$个区域。综上，以下递推关系成立：

$$B(M+1,d)\le B(M,d)+B(M,d-1)$$

接着证明$B(M,d)=\sum_{i=0}^d\binom{M}{i}$，先考虑两类特殊情形：$d=1$或者$M=1$

$$B(M,1)\text{相当于1维空间对}M\text{个点中间切一刀，一侧全为}-1,\text{另一侧全为}+1,\text{一共有}2M\text{种}$$
$$B(1,d)\text{表示一个平面，只能表示}(+1,-1),(-1,+1)\text{两种情况，所以}B(1,d)=2$$

所以结论对于$M=1$以及$d=1$成立，假设结论对于$M\le x,d\le y$成立，我们计算$B(x+1,y),B(x,y+1),B(x+1,y+1)$

先考虑$B(x+1,y)$

$$\begin{aligned}B(x+1,y)&\le B(x,y)+B(x,y-1)\\&\le\sum_{i=0}^{y}\binom{x}{i}+\sum_{i=0}^{y-1}\binom{x}{i}\\&=1+\sum_{i=1}^{y}\binom{x}{i}+\sum_{i=0}^{y-1}\binom{x}{i}\\&=1+\sum_{i=0}^{y-1}\binom{x}{i+1}+\sum_{i=0}^{y-1}\binom{x}{i}\\&=1+\sum_{i=0}^{y-1}\binom{x+1}{i+1}\\&=\sum_{i=0}^{y}\binom{x+1}{i}\end{aligned}$$

所以结论对于$B(x+1,y)$成立。接着考虑$B(x+1,y)$

$$B(x,y+1)-B(x-1,y+1)\le B(x-1,y)$$
$$B(x-1,y+1)-B(x-2,y+1)\le B(x-2,y)$$
$$\ldots$$
$$B(2,y+1)-B(1,y+1)\le B(1,y)$$
$$\text{累加可得}$$
$$B(x,y+1)-B(1,y+1)\le\sum_{i=1}^{x-1}B(i,y)$$
$$\text{对}\sum_{i=1}^{x-1}B(i,y)\text{进行处理}$$
$$\sum_{i=1}^{x-1}B(i,y)\le\sum_{i=1}^{x-1}\sum_{j=0}^{y}\binom{i}{j}=\sum_{j=0}^{y}\sum_{i=1}^{x-1}\binom{i}{j}=\sum_{j=0}^{y}\sum_{i=j}^{x-1}\binom{i}{j}-\binom{0}{0}=\sum_{j=0}^{y}\binom{x}{j+1}-1=\sum_{i=1}^{y+1}\binom{x}{i}-1$$
$$B(x,y+1)\le B(1,y+1)+\sum_{i=1}^{x-1}B(i,y)\le 2+\sum_{i=1}^{y+1}\binom{x}{i}-1=\sum_{i=0}^{y+1}\binom{x}{i}$$

所以结论对于$B(x,y+1)$成立。最后考虑$B(x+1,y+1)$

$$B(x+1,y+1)\le B(x,y+1)+B(x,y)$$

由于$B(x,y+1),B(x,y)$均满足结论，所以接下来的步骤同证明$B(x+1,y)$的步骤，从而结论对于$B(x+1,y+1)$也成立，由数学归纳法可知，该结论成立。

Problem 7.3 (Page 44)

Suppose that a target function f (for classification) is represented by a number of hyperplanes, where the different regions defined by the hyperplanes (see Problem 7.2) could be either classified $+1$ or -1, as with the 2-dimensional examples we considered in the text. Let the hyperplanes be $h_1,h_2,\ldots,h_M$, where $h_m(x)=\operatorname{sign}(w_m\,.\,x)$. Consider all the regions that are classified $+1$, and let one such region be r^+. Let $c=(c_1,c_2,\ldots,c_M)$ be the label of any point in the region (all points in a given region have the same label); the label $c_m=\pm1$ tells which side of h_m the point is on. Define the AND-term corresponding to region r by

$$t_r=h_1^{c_1}h_2^{c_2}\ldots h_M^{c_M},\text{where }h_m^{c_m}=\begin{cases}h_m & \text{if }c_m=+1\\ \bar{h}_m & \text{if }c_m=-1\end{cases}$$

Show that $f=t_{r_1}+t_{r_2}+\cdots+t_{r_k}$, where $r_1,\ldots,r_k$ are all the positive regions. (We use multiplication for the AND and addition for the OR operators.)

题目的意思是将某个用于二分类的目标函数f利用M个超平面表示出来，超平面为$h_1,h_2,\ldots,h_M$，其中$h_m(x)=\operatorname{sign}(w_m\,.\,x)$，这样就产生了一个$M$维向量$(h_1(x),h_2(x),\ldots,h_M(x))$，$M$维向量一致的点构成了区域，接着将部分$M$维向量映射到$+1$，其余的映射到$-1$，二元分类函数$f$就完成了。举一个具体例子，现在有两个超平面$h_1,h_2$，这样会产生$(+1,+1),(+1,-1),(-1,+1),(-1,-1)$四个2维向量，现在将$(+1,+1),(+1,-1)$对应的分类记为$+1$，$(-1,+1),(-1,-1)$对应的分类记为$-1$，这样分类就完成了。

现在考虑分类结果为$+1$的某个区域，假设这个区域内的点对应的M维向量均为$c=(c_1,c_2,\ldots,c_M)$，一个点x属于该区域等价于

$$(h_1(x),h_2(x),\ldots,h_M(x))=(c_1,c_2,\ldots,c_M)$$

我们考虑$h_m^{c_m}(x)$，当$c_m=+1$时，如果$h_m(x)=+1=c_m$，那么$h_m^{c_m}(x)=h_m(x)=+1$，当$c_m=-1$时，如果$h_m(x)=-1=c_m$，那么$h_m^{c_m}(x)=\bar{h}_m(x)=+1$。

所以

$$h_m(x)=c_m \Leftrightarrow h_m^{c_m}(x)=+1$$

从而

$$\begin{gathered}(h_1(x),h_2(x),\ldots,h_M(x))=(c_1,c_2,\ldots,c_M)\\ \Leftrightarrow h_m^{c_m}(x)=+1(m=1,\ldots,M)\\ \Leftrightarrow \Pi_{m=1}^M h_m^{c_m}(x)=+1\\ \Leftrightarrow t_r(x)=+1\end{gathered}$$

这里是讨论属于某一个结果为$+1$的某个区域，如果考虑全部结果为$+1$的某个区域$r_1,\ldots,r_k$，那么关系应该为或，所以目标函数为

$$f=t_{r_1}+t_{r_2}+\cdots+t_{r_k}$$

Problem 7.4 (Page 44)

Referring to Problem 7.3, any target function which can be decomposed into hyperplanes $h_1,h_2,\ldots,h_M$ can be represented by $f=t_{r_1}+t_{r_2}+\cdots+t_{r_k}$, where there are k positive regions. What is the structure of the 3-layer perceptron (number of hidden units in each layer) that will implement this function, proving the following theorem:

Theorem. Any decision function whose ± 1 regions are defined in terms of the regions created by a set of hyperplanes can be implemented by a 3-layer perceptron.

第一层构造kM个神经元，每个区域对应M个，一共有k个区域

$$s_{n,m}^{(1)}=\operatorname{sign}(c_{n,m}w_m.x)(m=1,\ldots,M,n=1,\ldots k)$$

上题中有式子$t_r=h_1^{c_1}h_2^{c_2}\ldots h_M^{c_M}$，上式中$c_{n,m}$表示第$n$个区域中第$m$项对应的$c_m$

第二层k个神经元，每个神经元表示结果为$+1$的区域

$$s_n^{(2)}=\operatorname{sign}(-k+\frac{1}{2}+\sum_{m=1}^M s_{n,m}^{(1)})(n=1,\ldots,k)$$

最后一层表达OR逻辑

$$f=\operatorname{sign}(k-\frac{1}{2}-\sum_{n=1}^k s_n^{(2)})$$

Problem 7.5 (Page 44)

[Hard] State and prove a version of a Universal Approximation Theorem:

Theorem. Any target function f (for classification) defined on $[0,1]^d$, whose classification boundary surfaces are smooth, can arbitrarily closely be approximated by a 3-layer perceptron.

[Hint: Decompose the unit hypercube into ϵ-hypercubes ($\frac{1}{\epsilon^d}$ of them); The volume of these ϵ-hypercubes which intersects the classification boundaries must tend to zero (why? – use smoothness). Thus, the function which takes on the value of f on any ϵ-hypercube that does not intersect the boundary and an arbitrary value on these boundary ϵ-hypercubes will approximate f arbitrarily closely, as $\epsilon \to 0$.]

我们知道分类问题其实是将某些区域的点映射为$+1$，其余区域的点映射为$-$，由Problem 7.3,7.4我们知道，对于超平面划分出来的区域，我们可以利用三层感知机来表示这个分类函数，而对于一般区域，由于分类边界光滑，所以由数学分析中的知识可知只要超平面足够多，这些超平面划分的区域可以与目标区域无限接近，从而只要超平面足够多，我们就可以利用三层感知机表达任意的分类问题。

注：这里没有严格证明，只简单解释了思路。

Problem 7.6 (Page 44)

The finite difference approximation to obtaining the gradient is based on the following formula from calculus:

$$\frac{\partial h}{\partial w_{ij}^{(\ell)}}=\frac{h(w_{ij}^{(\ell)}+\epsilon)-h(w_{ij}^{(\ell)}-\epsilon)}{2\epsilon}+O(\epsilon^2)$$

where $h(w_{ij}^{(\ell)}+\epsilon)$ to denotes the function value when all weights are held at their values in w except for the weight $w_{ij}^{(\ell)}$, which is perturbed by ϵ. To get the gradient, we need the partial derivative with respect to each weight.

Show that the computational complexity of obtaining all these partial derivatives $O(W^2)$. [Hint: you have to do two forward propagations for each weight.]

题目中没有交代W是什么，我推测为权重的数量，公式同Exercise 7.6的$W=Q=\sum_{\ell=0}^{L} d^{(\ell)}(d^{(\ell-1)}+1)$

如果修改了第l层的某个权重，由计算过程我们知道l层之前的节点无需重新计算，l层需要重新计算一次，之后的每一层的节点都要重新计算，所以修改第l层的某个权重计算的复杂度为

$$1+\sum_{\ell=l+1}^{L} d^{(\ell)}(d^{(\ell-1)}+1)$$

第l层一共有$d^{(l)}(d^{(l-1)}+1)$个权重，所以修改第l层的全部权重的计算复杂度为

$$d^{(l)}(d^{(l-1)}+1)\times(1+\sum_{\ell=l+1}^{L} d^{(\ell)}(d^{(\ell-1)}+1))$$

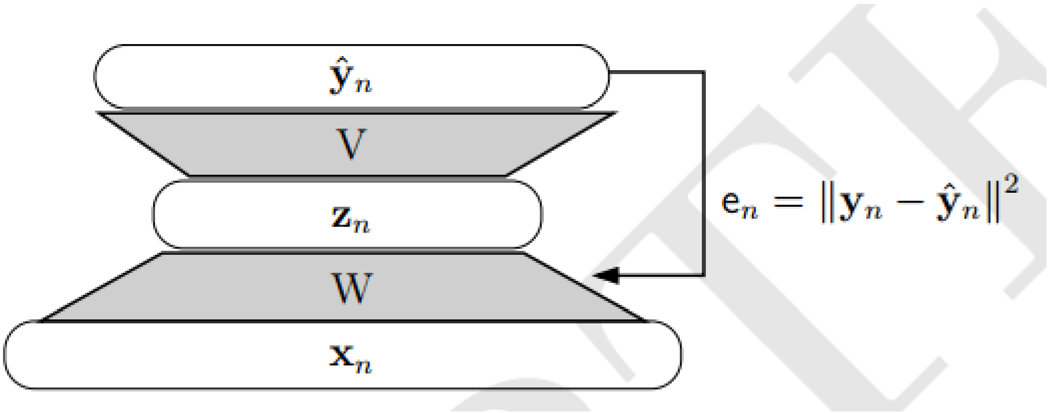
从而修改每个权重的计算复杂度为上式关于l累加，可得

$$\begin{aligned}\sum_{l=0}^{L} d^{(l)}(d^{(l-1)}+1)\times(1+\sum_{\ell=l+1}^{L} d^{(\ell)}(d^{(\ell-1)}+1)) &\le \sum_{l=0}^{L} d^{(l)}(d^{(l-1)}+1)\times(1+\sum_{\ell=0}^{L} d^{(\ell)}(d^{(\ell-1)}+1))\\ &= W(1+W)\end{aligned}$$

所以计算复杂度为O(W^2)

Problem 7.7 (Page 45)

Consider the 2-layer network below, with output vector $\hat{y}$. This is the two layer network used for the greedy deep network algorithm.



Collect the input vectors x_n (together with a column of ones) as rows in the input data matrix X, and similarly form Z from z_n. The target matrices Y and $\hat{Y}$ are formed from y_n and $\hat{y}_n$ respectively. Assume a linear output node and the hidden layer activation is $\theta(\cdot)$.

(a) Show that the in-samp

$$E_{in} = \frac{1}{N}\mathrm{trace}\Big((Y - \hat{Y})(Y - \hat{Y})^T\Big)$$

where

$$\begin{gathered} \hat{Y} = ZV \\ Z = [1, \theta(XW)] \\ X \text{ is } N \times (d+1) \\ W \text{ is } (d+1) \times d^{(1)} \\ Z \text{ is } N \times (d^{(1)} + 1) \\ V = \begin{bmatrix} V_0 \\ V_1 \end{bmatrix} \text{ is } (d^{(1)} + 1) \times \dim(y) \\ Y, \hat{Y} \text{ are } N \times \dim(y) \end{gathered}$$

(It is convenient to decompose V into its first row V_0 corresponding to the biases and its remaining rows V_1; 1 is the $N \times 1$ vector of ones.)

(b) derive the gradient matrices:

$$\begin{gathered} \frac{\partial E_{in}}{\partial V} = \frac{1}{N}(2Z^TZV - 2Z^TY) \\ \frac{\partial E_{in}}{\partial W} = \frac{2}{N}X^T[\theta'(XW) \otimes (\theta(XW)V_1V_1^T + 1V_0V_1^T - YV_1^T)], \end{gathered}$$

where $\otimes$ denotes element-wise multiplication. Some of the matrix deriva tives of functions involving the trace from the appendix may be useful.

(a)首先写出E_{in}的定义

$$E_{in} = \frac{1}{N}\sum_{i=1}^{N}||y_i - \hat{y}_i||^2$$

考虑$(Y-\hat{Y})(Y-\hat{Y})^T$的第i,j个元素

$$[(Y-\hat{Y})(Y-\hat{Y})^T]_{ij} = (y_i - \hat{y}_i)(y_j - \hat{y}_j)^T$$

所以

$$E_{in} = \frac{1}{N}\text{trace}\Big((Y-\hat{Y})(Y-\hat{Y})^T)$$

(b)将$\hat{Y} = ZV$带入，展开化简

$$\begin{aligned}
E_{in} &= \frac{1}{N}\text{trace}\Big((Y-ZV)(Y-ZV)^T) \\
&= \frac{1}{N}\text{trace}\Big(YY^T - ZVY^T - YV^TZ^T + ZVV^TZ^T) \\
&= \frac{1}{N}\text{trace}\Big(YY^T - 2ZVY^T + ZVV^TZ^T)
\end{aligned}$$

最后一步是因为$\text{trace X} = \text{trace } X^T$

接着利用如下两个性质

$$\frac{\partial(\text{trace}(AXB))}{\partial X} = A^TB^T$$

$$\frac{\partial(\text{trace}(AXX^TB))}{\partial X} = A^TB^TX + BAX$$

计算$\frac{\partial E_{in}}{\partial V}$

$$\begin{aligned}
\frac{\partial E_{in}}{\partial V} &= \frac{1}{N}\big(-2\frac{\partial(\text{trace}(ZVY^T))}{\partial V} + \frac{\partial(\text{trace}(ZVV^TZ^T))}{\partial V}\big) \\
&= \frac{1}{N}\big(-2Z^TY + Z^TZV + Z^TZV\big) \\
&= \frac{1}{N}\big(2Z^TZV - 2Z^TY\big)
\end{aligned}$$

为了计算$\frac{\partial E_{in}}{\partial W}$，将$Z = [1, \theta(XW)], V = \begin{bmatrix} V_0 \\ V_1 \end{bmatrix}$带入

$$\begin{aligned}
E_{in} &= \frac{1}{N}\text{trace}\Big((Y-ZV)(Y-ZV)^T) \\
&= \frac{1}{N}\text{trace}\Big(YY^T - 2ZVY^T + ZVV^TZ^T) \\
&= \frac{1}{N}\text{trace}\Big(YY^T - 2(V_0 + \theta(XW)V_1)Y^T + (V_0 + \theta(XW)V_1)(V_0 + \theta(XW)V_1)^T) \\
&= \frac{1}{N}\text{trace}\Big(YY^T - 2V_0Y^T - 2\theta(XW)V_1Y^T + V_0V_0^T + \theta(XW)V_1V_0^T + V_0V_1^T\theta(XW)^T + \theta(XW)V_1V_1^T\theta(XW)^T) \\
&= \frac{1}{N}\text{trace}\Big(YY^T - 2V_0Y^T - 2\theta(XW)V_1Y^T + V_0V_0^T + 2\theta(XW)V_1V_0^T + V_1V_1^T\theta(XW)^T\theta(XW))
\end{aligned}$$

最后一步是因为$\text{trace X} = \text{trace } X^T$以及$\text{trace AB} = \text{trace BA}$

接着利用如下两个性质

$$\frac{\partial(\mathrm{trace}(\theta(BX)A))}{\partial X} = B^T(\theta^{'}(BX) \otimes A^T)$$
$$\frac{\partial(\mathrm{trace}(A\theta(BX)^T\theta(BX)))}{\partial X} = B^T(\theta^{'}(BX) \otimes [\theta(BX)(A + A^T)])$$

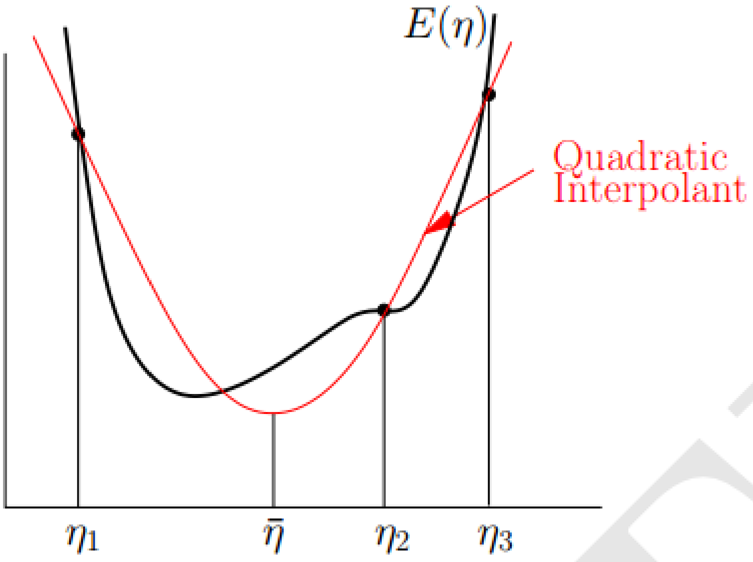
计算$\frac{\partial E_{in}}{\partial W}$

$$\begin{aligned}\frac{\partial E_{in}}{\partial W} &= \frac{1}{N}\Big(-2\frac{\partial(\mathrm{trace}(\theta(XW)V_1Y^T))}{\partial W} + 2\frac{\partial(\mathrm{trace}(\theta(XW)V_1V_0^T)}{\partial W} + \frac{\partial(\mathrm{trace}(V_1V_1^T\theta(XW)^T\theta(XW)))}{\partial W}\Big)\\ &= \frac{1}{N}\Big(-2X^T(\theta^{'}(XW) \otimes YV_1^T) + 2X^T(\theta^{'}(XW) \otimes V_0V_1^T) + X^T(\theta^{'}(XW) \otimes [\theta(XW)2V_1V_1^T])\Big)\\ &= \frac{1}{N}(2X^T[\theta^{'}(XW) \otimes (\theta(XW)V_1V_1^T) + 1V_0V_1^T - YV_1^T)])\end{aligned}$$

所以结论成立。

Problem 7.8 (Page 46)

Quadratic Interpolation for Line Search Assume that a U-arrangement has been found, as illustrated below.



Instead of using bisection to construct the point $\bar{\eta}$, quadratic interpolation fits a quadratic curve $E(\eta) = a\eta^2 + b\eta + c$ to the three points and uses the minimum of this quadratic interpolation as $\bar{\eta}$.

(a) Show that the minimum of the quadratic interpolant for a U-arrangement is within the interval $[\eta_1, \eta_3]$.

(b) Let $e_1 = E(\eta_1), e_2 = E(\eta_2), e_3 = E(\eta_3)$. Obtain the quadratic function that interpolates the three points $\{(\eta_1, e_1), (\eta_2, e_2), (\eta_3, e_3)\}$. Show that the minimum of this quadratic interpolant is given by:

$$\bar{\eta} = \frac{1}{2}\left[\frac{(e_1 - e_2)(\eta_1^2 - \eta_3^2) - (e_1 - e_3)(\eta_1^2 - \eta_2^2)}{(e_1 - e_2)(\eta_1 - \eta_3) - (e_1 - e_3)(\eta_1 - \eta_2)}\right]$$

[Hint: $e_1 = a\eta_1^2 + b\eta_1 + c, e_2 = a\eta_2^2 + b\eta_2 + c, e_3 = a\eta_3^2 + b\eta_3 + c$. Solve for a, b, c and the minimum of the quadratic is given by $\bar{\eta} = -b/2a$.]

(c) Depending on whether $E(\bar{\eta})$ is less than $E(\eta_2)$, and on whether $\bar{\eta}$ is to the left or right of η_2, there are 4 cases. In each case, what is the smaller U-arrangement?

(d) What if $\overline{\eta} = \eta_2$, a degenerate case?

Note: in general the quadratic interpolations converge very rapidly to a locally optimal η. In practice, 4 iterations are more than sufficient.

(a)由课本之前论述可知

$$E(\eta_2) < \min\{E(\eta_1), E(\eta_3)\}$$

$\overline{\eta}$为过$\{(\eta_1, E(\eta_1)), (\eta_2, E(\eta_2)), (\eta_3, E(\eta_3))\}$三点的二次函数的对称轴，如果$\overline{\eta} < \eta_1$，那么$E(\eta_1) < E(\eta_2) < E(\eta_3)$或$E(\eta_1) > E(\eta_2) > E(\eta_3)$，与$E(\eta_2) < \min\{E(\eta_1), E(\eta_3)\}$矛盾，所以$\overline{\eta} \ge \eta_1$，同理$\overline{\eta} \le \eta_3$，所以

$$\overline{\eta} \in [\eta_1, \eta_3]$$

(b)$\eta = -\frac{b}{2a}$，所以只要计算a, b即可，将e_i的定义带入

$$e_1 = a\eta_1^2 + b\eta_1 + c, e_2 = a\eta_2^2 + b\eta_2 + c, e_3 = a\eta_3^2 + b\eta_3 + c$$

计算$e_1 - e_2, e_1 - e_3$

$$\begin{aligned} a(\eta_1^2 - \eta_2^2) + b(\eta_1 - \eta_2) = e_1 - e_2 \\ a(\eta_1^2 - \eta_3^2) + b(\eta_1 - \eta_3) = e_1 - e_3 \end{aligned}$$

解这个二元一次方程组可得

$$\begin{aligned} b &= \frac{-(e_1 - e_2)(\eta_1^2 - \eta_3^2) + (e_1 - e_3)(\eta_1^2 - \eta_2^2)}{(\eta_1^2 - \eta_2^2)(\eta_1 - \eta_3) - (\eta_1^2 - \eta_3^2)(\eta_1 - \eta_2)} \\ a &= \frac{(e_1 - e_2)(\eta_1 - \eta_3) - (e_1 - e_3)(\eta_1 - \eta_2)}{(\eta_1^2 - \eta_2^2)(\eta_1 - \eta_3) - (\eta_1^2 - \eta_3^2)(\eta_1 - \eta_2)} \\ \overline{\eta} &= -\frac{b}{2a} = \frac{1}{2}\Big[\frac{(e_1 - e_2)(\eta_1^2 - \eta_3^2) - (e_1 - e_3)(\eta_1^2 - \eta_2^2)}{(e_1 - e_2)(\eta_1 - \eta_3) - (e_1 - e_3)(\eta_1 - \eta_2)}\Big] \end{aligned}$$

(c)如果$E(\overline{\eta}) < E(\eta_2)$

- 当$\overline{\eta} < \eta_2$时，U-arrangement为$(\eta_1, \overline{\eta}, \eta_2)$
- 当$\overline{\eta} > \eta_2$时，U-arrangement为$(\eta_2, \overline{\eta}, \eta_3)$

如果$E(\overline{\eta}) > E(\eta_2)$

- 当$\overline{\eta} < \eta_2$时，U-arrangement为$(\overline{\eta}, \eta_2, \eta_3)$
- 当$\overline{\eta} > \eta_2$时，U-arrangement为$(\eta_1, \overline{\eta}, \eta_2)$

(d)如果$\overline{\eta} = \eta_2$，那么可以在η_2的邻域内再找一个点$\eta_2^{'}$，使得$\overline{\eta} \neq \eta_2$且$E(\eta_2^{'}) < \min\{E(\eta_1), E(\eta_3)\}$，然后重复上述迭代步骤即可

Problem 7.9 (Page 46)

[Convergence of Monte-Carlo Minimization] Suppose the global minimum w^* is in the unit cube and the error surface is quadratic near w^*. So, near w^*,

$$E(w) = E(w^*) + \frac{1}{2}(w - w^*)^T H(w - w^*)$$

where the Hessian H is a positive definite and symmetric.

(a) If you uniformly sample w in the unit cube, show that

$$P[E\leq E(w^*)+\epsilon]=\int\limits_{x^THx\leq 2\epsilon} d^dx=\frac{S_d(2\epsilon)}{\sqrt{\det H}}$$

where $S_d(r)$ is the volume of the d-dimensional sphere of radius r,

$$S_d(r)=\pi^{d/2}r^d/\Gamma(\frac{d}{2}+1)$$

[Hints: $P[E\leq E(w^*)+\epsilon]=P[\frac{1}{2}(w-w^*)^TH(w-w^*)\leq\epsilon]$. Suppose the orthogonal matrix A diagonalizes H: $A^THA=diag[\lambda_1^2,\ldots,\lambda_d^2]$. Change variables to $u=A^Tx$ and use $\det H=\lambda_1^2\lambda_2^2\ldots\lambda_d^2$.]

(b) Suppose you sample N times and choose the weights with minimum error, w_{min}. Show that

$$P[E(w_{min})>E(w^*)+\epsilon]\approx\Big(1-\frac{1}{\sqrt{\pi d}}\Big(\mu\frac{\epsilon}{\sqrt{d}}\Big)^d\Big)^N$$

where $\mu\approx\sqrt{8e\pi/\bar{\lambda}}$ and $\bar{\lambda}$ is the geometric mean of the eigenvalues of H. (You may use $\Gamma(x+1)\approx x^xe^{-x}\sqrt{2\pi x}$.)

(c) Show that if $N\sim(\frac{\sqrt{d}}{\mu\epsilon})^d\log\frac{1}{\eta}$, then with probability at least $1-\eta$, $E(w_{min})\leq E(w^*)+\epsilon$. (You may use $\log(1-a)\approx -a$ for small a and $(\pi d)^{1/d}\approx 1$.)

题目中b的结论有点问题，我对其做了修改

(a)因为H半正定，所以存在正交矩阵H，$A^THA=diag[\lambda_1^2,\ldots,\lambda_d^2]=S$，所以$H=ASA^T$，可以得到以下变形

$$\begin{aligned}
P[E\leq E(w^*)+\epsilon]&=\int\limits_{x^THx\leq 2\epsilon} d^dx\\
&=\int\limits_{x^TASA^Tx\leq 2\epsilon} d^dx\\
&\overset{y=A^Tx}{=}\int\limits_{y^TSy\leq 2\epsilon}|\det\frac{\partial x}{\partial y}|d^dy\\
&=\int\limits_{y^TSy\leq 2\epsilon} d^dy
\end{aligned}$$

注意$y^TSy=\sum_{i=1}^d\lambda_i^2y_i^2$，所以令$z_i=\lambda_iy_i$，注意$\det H=\lambda_1^2\lambda_2^2\ldots\lambda_d^2$，那么$|\det\frac{\partial y}{\partial z}|=\frac{1}{\prod_{i=1}^d\lambda_i}=\frac{1}{\sqrt{\det H}}$，带入上式可得

$$\begin{aligned}P[E\le E(w^*)+\epsilon]&=\int\limits_{y^TSy\le 2\epsilon}d^dy\\&=\int\limits_{z^TSz\le 2\epsilon}|\det\frac{\partial y}{\partial z}|d^dz\\&=\frac{\int\limits_{z^TSz\le 2\epsilon}d^dz}{\sqrt{\det H}}\\&=\frac{S_d(2\epsilon)}{\sqrt{\det H}}\end{aligned}$$

(b)

$$\begin{aligned}P[E(w_{min})>E(w^*)+\epsilon]&=\prod_{i=1}^{N}P[E(w)>E(w^*)+\epsilon]\\&=\Big(P[E(w)>E(w^*)+\epsilon]\Big)^N\\&=\Big(1-P[E(w)\le E(w^*)+\epsilon]\Big)^N\\&=\Big(1-\frac{S_d(2\epsilon)}{\sqrt{\det H}}\Big)^N\end{aligned}$$

接着计算$S_d(2\epsilon)$，利用$S_d(r)=\pi^{d/2}r^d/\Gamma(\frac{d}{2}+1)$以及$\Gamma(x+1)\approx x^xe^{-x}\sqrt{2\pi x}$

$$\begin{aligned}S_d(2\epsilon)&=\pi^{d/2}(2\epsilon)^d/\Gamma(\frac{d}{2}+1)\\&\approx\frac{\pi^{d/2}(2\epsilon)^d}{(\frac{d}{2})^{\frac{d}{2}}e^{-\frac{d}{2}}\sqrt{2\pi\frac{d}{2}}}\\&=\frac{(8e\pi)^{\frac{d}{2}}\epsilon^d}{\sqrt{\pi d}d^{\frac{d}{2}}}\end{aligned}$$

注意$\overline{\lambda}$为H特征值的几何平均数，所以

$$\overline{\lambda}^d=\det H$$

注意$\mu=\sqrt{8e\pi/\overline{\lambda}}$，将这些带入可得

$$\begin{aligned}P[E(w_{min})>E(w^*)+\epsilon]&=\Big(1-\frac{S_d(2\epsilon)}{\sqrt{\det H}}\Big)^N\\&\approx\Big(1-\frac{(8e\pi)^{\frac{d}{2}}\epsilon^d}{\sqrt{\pi d}d^{\frac{d}{2}}\overline{\lambda}^{\frac{d}{2}}}\Big)^N\\&=\Big(1-\frac{\mu^d\epsilon^d}{\sqrt{\pi d}d^{\frac{d}{2}}}\Big)^N\\&=\Big(1-\frac{1}{\sqrt{\pi d}}\Big(\mu\frac{\epsilon}{\sqrt{d}}\Big)^d\Big)^N\end{aligned}$$

(c)感觉这题可能有点问题，先按照题目中的条件对其化简

$$\begin{aligned}P[E(w_{min}) > E(w^*) + \epsilon] &\approx \left(1 - \frac{1}{\sqrt{\pi d}}\left(\mu\frac{\epsilon}{\sqrt{d}}\right)^d\right)^N \\ &= e^{N\ln(1-\frac{1}{\sqrt{\pi d}}\left(\mu\frac{\epsilon}{\sqrt{d}}\right)^d)} \\ &\approx e^{-N\frac{1}{\sqrt{\pi d}}\left(\mu\frac{\epsilon}{\sqrt{d}}\right)^d}\end{aligned}$$

利用$N \sim (\frac{\sqrt{d}}{\mu\epsilon})^d\log\frac{1}{\eta}$对$-N\frac{1}{\sqrt{\pi d}}\left(\mu\frac{\epsilon}{\sqrt{d}}\right)^d$进行化简

$$\begin{aligned}-N\frac{1}{\sqrt{\pi d}}\left(\mu\frac{\epsilon}{\sqrt{d}}\right)^d &\approx -(\frac{\sqrt{d}}{\mu\epsilon})^d\log\frac{1}{\eta}\frac{1}{\sqrt{\pi d}}\left(\mu\frac{\epsilon}{\sqrt{d}}\right)^d \\ &= \frac{1}{\sqrt{\pi d}}\log\eta\end{aligned}$$

从而

$$\begin{aligned}P[E(w_{min}) > E(w^*) + \epsilon] &\approx \left(1 - \frac{1}{\sqrt{\pi d}}\left(\mu\frac{\epsilon}{\sqrt{d}}\right)^d\right)^N \\ &\approx e^{-N\frac{1}{\sqrt{\pi d}}\left(\mu\frac{\epsilon}{\sqrt{d}}\right)^d} \\ &\approx e^{\frac{1}{\sqrt{\pi d}}\log\eta} \\ &= \eta^{\frac{1}{\sqrt{\pi d}}}\end{aligned}$$

于是做到这一步就没法继续了，但是如果对条件加以修改：$N \sim (\frac{\sqrt{d}}{\mu\epsilon})^d(\pi d)^{\frac{1}{2}+\frac{1}{d}}\log\frac{1}{\eta}$，那么

$$\begin{aligned}-N\frac{1}{\sqrt{\pi d}}\left(\mu\frac{\epsilon}{\sqrt{d}}\right)^d &\approx -(\frac{\sqrt{d}}{\mu\epsilon})^d(\pi d)^{\frac{1}{2}+\frac{1}{d}}\log\frac{1}{\eta}\frac{1}{\sqrt{\pi d}}\left(\mu\frac{\epsilon}{\sqrt{d}}\right)^d \\ &= (\pi d)^{\frac{1}{d}}\log\eta\end{aligned}$$

$$\begin{aligned}P[E(w_{min}) > E(w^*) + \epsilon] &\approx \left(1 - \frac{1}{\sqrt{\pi d}}\left(\mu\frac{\epsilon}{\sqrt{d}}\right)^d\right)^N \\ &\approx e^{-N\frac{1}{\sqrt{\pi d}}\left(\mu\frac{\epsilon}{\sqrt{d}}\right)^d} \\ &\approx e^{(\pi d)^{\frac{1}{d}}\log\eta} \\ &\approx e^{\log\eta} \\ &= \eta\end{aligned}$$

Problem 7.10 (Page 47)

For a neural network with at least 1 hidden layer and $\tanh(\cdot)$ transformations in each non-input node, what is the gradient (with respect to the weights) if all the weights are set to zero. Is it a good idea to initialize the weights to zero?

同Exercise 7.9

Problem 7.11 (Page 47)

[Optimal Learning Rate] Suppose that we are in the vicinity of a local minimum, w^*, of the error surface, or that the error surface is quadratic. The expression for the error function is then given by

$$E(w_t) = E(w^*) + \frac{1}{2}(w_t - w^*)H(w_t - w^*) \tag{7.8}$$

from which it is easy to see that the gradient is given by $g_t = H(w_t - w^*)$. The weight updates are then given by $w_{t+1} = w_t - \eta H(w_t - w^*)$, and subtracting w^* from both sides, we see that

$$\epsilon_{t+1} = (I - \eta H)\epsilon_t \tag{7.9}$$

Since H is symmetric, one can form an orthonormal basis with its eigenvectors. Projecting ϵ_t and ϵ_{t+1} onto this basis, we see that in this basis, each component decouples from the others, and letting $\epsilon(\alpha)$ be the α^{th} component in this basis, we see that

$$\epsilon_{t+1}(\alpha) = (1 - \eta\lambda_\alpha)\epsilon_t(\alpha) \tag{7.10}$$

so we see that each component exhibits linear convergence with its own coefficient of convergence $k_\alpha = 1 - \eta\lambda_\alpha$. The worst component will dominate the convergence so we are interested in choosing η so that the k_α with largest magnitude is minimized. Since H is positive definite, all the λ_α's are positive, so it is easy to see that one should choose η so that $1 - \eta\lambda_{min} = 1 - \Delta$ and $1 - \eta\lambda_{max} = 1 + \Delta$, or one should choose. Solving for the optimal η, one finds that

$$\eta_{opt} = \frac{2}{\lambda_{min} + \lambda_{max}}$$
$$k_{opt} = \frac{1-c}{1+c}$$

where $c = \lambda_{min}/\lambda_{max}$ is the condition number of H, and is an important measure of the stability of H. When $c \approx 0$, one usually says that H is illconditioned. Among other things, this affects the one's ability to numerically compute the inverse of H.

先对题目中的式子简单解释下，因为H为半正定矩阵，所以它的特征向量可以构成正交基，将$\epsilon_{t+1} = (I - \eta H)\epsilon_t$投影在每个特征向量上，记第$\alpha$个特征向量对应的特征值为$\lambda_\alpha$，那么可得

$$\epsilon_{t+1}(\alpha) = (1 - \eta H)\epsilon_t(\alpha) = (1 - \eta\lambda_\alpha)\epsilon_t(\alpha)$$

所以收敛速度与$|1 - \eta\lambda_\alpha|$有关，我们要求得最优η_{opt}就是解决以下问题

$$\min \max |1 - \eta\lambda_\alpha|$$

因为$\lambda_\alpha \ge 0$，所以由绝对值函数的性质可知

$$\max|1 - \eta\lambda_\alpha| = \max\{|1 - \eta\lambda_{max}|, |1 - \eta\lambda_{min}|\} = \begin{cases} 1 - \eta\lambda_{max} & (\eta \le 0) \\ 1 - \eta\lambda_{min} & (0 < \eta < \frac{2}{\lambda_{min} + \lambda_{max}}) \\ \eta\lambda_{max} - 1 & (\eta \ge \frac{2}{\lambda_{min} + \lambda_{max}}) \end{cases}$$

所以当$\eta = \frac{2}{\lambda_{min} + \lambda_{max}}$时，$\max|1 - \eta\lambda_\alpha|$取最小值，从而

$$\eta_{opt} = \frac{2}{\lambda_{min} + \lambda_{max}}$$
$$k_{opt} = 1 - \eta_{opt}\lambda_{min} = 1 - \frac{2\lambda_{min}}{\lambda_{min} + \lambda_{max}} = \frac{\lambda_{max} - \lambda_{min}}{\lambda_{min} + \lambda_{max}} = \frac{1-c}{1+c}, \text{ 其中} c = \frac{\lambda_{min}}{\lambda_{max}}$$

Problem 7.12 (Page 48)

[Hard] With a variable learning rate, suppose that $\eta_t \to 0$ satisfying $\sum_t \eta_t = +\infty$ and $\sum_t \eta_t^2 < \infty$, for example one could choose $\eta_t = 1/(t+1)$. Show that gradient descent will converge to a local minimum.

接着采取上一题的思路，更新公式为

$$\epsilon_{t+1} = (I - \eta_t H)\epsilon_t$$

将其投影到特征向量方向可得

$$\epsilon_{t+1}(\alpha) = (1 - \eta_t \lambda_\alpha)\epsilon_t(\alpha)$$

现在考虑其收敛性，将上述式子递推可得

$$\epsilon_{t+1}(\alpha) = \epsilon_1(\alpha)\prod_{i=1}^{t}(1 - \eta_i \lambda_\alpha)$$

只要考虑$\prod_{i=1}^{t}(1 - \eta_i \lambda_\alpha)$，这是个无穷乘积的问题。因为$\eta_t \to 0$，所以当$t$充分大时，$1 - \eta_t \lambda_\alpha > 0$。因为有限项不影响无穷乘积的收敛性，所以这里可以假设$1 - \eta_t \lambda_\alpha > 0$恒成立，从而可以对上式进行变形

$$\begin{aligned}\prod_{i=1}^{t}(1 - \eta_i \lambda_\alpha) &= \prod_{i=1}^{t} e^{\ln(1-\eta_i\lambda_\alpha)} \\ &= e^{\sum_{i=1}^{t}\ln(1-\eta_i\lambda_\alpha)} \\ &\approx e^{\sum_{i=1}^{t}(-\eta_i\lambda_\alpha + \frac{1}{2}\lambda_\alpha^2\eta_i^2)} \\ &= e^{-\lambda_\alpha\sum_{i=1}^{t}\eta_i + \frac{1}{2}\lambda_\alpha^2\sum_{i=1}^{t}\eta_i^2}\end{aligned}$$

因为$\sum_t \eta_t = +\infty, \sum_t \eta_t^2 < \infty$，所以

$$\begin{aligned}&e^{-\lambda_\alpha\sum_{i=1}^{t}\eta_i} \to 0 \\ &e^{\frac{1}{2}\lambda_\alpha^2\sum_{i=1}^{t}\eta_i^2} \to c\end{aligned}$$

从而

$$\prod_{i=1}^{t}(1 - \eta_i \lambda_\alpha) \approx e^{-\lambda_\alpha\sum_{i=1}^{t}\eta_i + \frac{1}{2}\lambda_\alpha^2\sum_{i=1}^{t}\eta_i^2} \to 0$$

$$\epsilon_{t+1}(\alpha) = \epsilon_1(\alpha)\prod_{i=1}^{t}(1 - \eta_i \lambda_\alpha) \to 0$$

即ϵ在每个特征方向上的投影都会收敛到0，所以ϵ收敛到0，因此梯度下降法会达到局部最小值。

Problem 7.13 (Page 48)

[Finite Difference Approximation to Hessian]

(a) Consider the function $E(w_1, w_2)$. Show that the finite difference approx imation to the second order partial derivatives are given by

$$\frac{\partial^2 E}{\partial w_1^2} = \frac{E(w_1+2h,w_2)+E(w_1-2h,w_2)-2E(w_1,w_2)}{4h^2}$$
$$\frac{\partial^2 E}{\partial w_2^2} = \frac{E(w_1,w_2+2h)+E(w_1,w_2-2h)-2E(w_1,w_2)}{4h^2}$$
$$\frac{\partial^2 E}{\partial w_2 \partial w_1} = \frac{E(w_1+h,w_2+h)+E(w_1-h,w_2-h)-E(w_1+h,w_2-h)-E(w_1-h,w_2+h)}{4h^2}$$

(b) Give an algorithm to compute the finite difference approximation to the Hessian matrix for $E_{in}(w)$, the in-sample error for a multilayer neural network with weights $w=[W^{(1)},\ldots,W^{(\ell)}]$.

(c) Compute the asymptotic running time of your algorithm in terms of the number of weights in your network and then number of data points.

(a)利用如下公式计算偏导数

$$f\text{为一个二元函数}\\ \frac{\partial f}{\partial x} \approx \frac{f(x+h,y)-f(x-h,y)}{2h}$$

二阶偏导数可以用同样的方法计算

$$\begin{aligned}
f_1(w_1,w_2) = \frac{\partial E}{\partial w_1} &= \frac{E(w_1+h,w_2)-E(w_1-h,w_2)}{2h}\\
f_2(w_1,w_2) = \frac{\partial E}{\partial w_2} &= \frac{E(w_1,w_2+h)-E(w_1,w_2-h)}{2h}\\
\frac{\partial^2 E}{\partial w_1^2} &= \frac{f_1(w_1+h,w_2)+f_1(w_1-h,w_2)}{2h}\\
&= \frac{\frac{E(w_1+2h,w_2)-E(w_1,w_2)}{2h}-\frac{E(w_1,w_2)-E(w_1-2h,w_2)}{2h}}{2h}\\
&= \frac{E(w_1+2h,w_2)+E(w_1-2h,w_2)-2E(w_1,w_2)}{4h^2}\\
\text{由对称性可得}\frac{\partial^2 E}{\partial w_2^2} &= \frac{E(w_1,w_2+2h)+E(w_1,w_2-2h)-2E(w_1,w_2)}{4h^2}\\
\frac{\partial^2 E}{\partial w_2 \partial w_1} &= \frac{f_1(w_1,w_2+h)+f_1(w_1,w_2-h)}{2h}\\
&= \frac{\frac{E(w_1+h,w_2+h)-E(w_1-h,w_2+h)}{2h}-\frac{E(w_1+h,w_2-h)-E(w_1-h,w_2-h)}{2h}}{2h}\\
&= \frac{E(w_1+h,w_2+h)+E(w_1-h,w_2-h)-E(w_1+h,w_2-h)-E(w_1-h,w_2+h)}{4h^2}
\end{aligned}$$

(b)(c)使用Exercise 7.6的记号

$$V=\sum_{i=0}^{\ell} d^{(i)}, Q=\sum_{\ell=0}^{\ell} d^{(i)}(d^{(i-1)}+1)$$

V为节点数量，Q为权重的数量，记$Q_i=d^{(i)}(d^{(i-1)}+1)$，即第i层有Q_i个权重，那么Q可以改写为

$$Q=\sum_{i=0}^{\ell} Q_i$$

首先看计算第i层的Hessian矩阵需要计算哪些，

首先是对角线上的元素$\dfrac{\partial^2 E}{\partial {w_{jk}^{(i)}}^2}$，需要计算$E(w_{jk}^{(i)} \pm h)$，所以一共需要计算$2Q_i$次

接着需要计算非对角线元素$\dfrac{\partial^2 E}{\partial w_{j_1 k_1}^{(i)} \partial w_{j_2 k_2}^{(i)}}$，需要计算$E(w_{j_1 k_1}^{(i)} \pm h, w_{j_2 k_2}^{(i)} \pm h)$，所以一共需要计算$4C_{Q_i}^2/2$次，除2是因为对称性

这两者一共需要计算$Q_i^2 + Q_i$次

注意计算$E(w_{jk}^{(i)} \pm h)$需要重新计算第i层之后每个节点，所以需要的计算量为

$$\sum_{k=i}^{\ell} Q_i$$

从而计算第i层的Hessian矩阵需要的次数为

$$(Q_i^2 + Q_i)\sum_{k=i}^{\ell} Q_i$$

计算全部的Hessian矩阵需要的次数为

$$\sum_{i=0}^{\ell}(Q_i^2 + Q_i)\sum_{k=i}^{\ell} Q_i \le Q\sum_{i=0}^{\ell}(Q_i^2 + Q_i) \le \ell Q^3$$

所以计算次数为

$$O(\ell Q^3)$$

Problem 7.14 (Page 48)

Suppose we take a fixed step in some direction, we ask what the optimal direction for this fixed step assuming that the quadratic model for the error surface is accurate:

$$E_{in}(w_t + \delta w) = E_{in}(w_t) + g_t^T \Delta w + \frac{1}{2}\Delta w^T H_t \Delta w$$

So we want to minimize $E_{in}(\Delta w)$ with respect to Δw subject to the constraint that the step size is η, i.e., that $\Delta w^T \Delta w = \eta^2$.

(a) Show that the Lagrangian for this constrained minimization problem is:

$$\mathcal{L} = E_{in}(w_t) + g_t^T \Delta w + \frac{1}{2}\Delta w^T (H_t + 2\alpha I)\Delta w - \alpha\eta^2 \tag{7.12}$$

where α is the Lagrange multiplier.

(b) Solve for Δw and α and show that they satisfy the two equations:

$$\Delta w = -(H_t + 2\alpha I)^{-1} g_t,$$
$$\Delta w^T \Delta w = \eta^2$$

(c) Show that α satisfies the implicit equation:

$$\alpha=-\frac{1}{2\eta^2}(\Delta w^T g_t+\Delta w^T H_t\Delta w).$$

Argue that the second term is $\theta(1)$ and the first is $O(\sim ||g_t||/\eta)$. So, α is large for a small step size η.

(d) Assume that α is large. Show that, To leading order in $\frac{1}{\eta}$

$$\alpha=\frac{||g_t||}{2\eta}$$

Therefore α is large, consistent with expanding Δw to leading order in $\frac{1}{\alpha}$. [Hint: expand Δw to leading order in $\frac{1}{\alpha}$]

(e) Using (d), show that $\Delta w=-(H_t+\frac{||g_t||}{\eta}I)^{-1}g_t$

这里对(a)的题目进行了修改，感觉原问题可能不对。

(a)利用拉格朗日乘子法，可以把上述条件极值问题转化为无条件极值问题

$$\begin{aligned}\mathcal{L}&=E_{in}(w_t)+g_t^T\Delta w+\frac{1}{2}\Delta w^T H_t\Delta w+\alpha(\Delta w^T\Delta w-\eta^2)\\&=E_{in}(w_t)+g_t^T\Delta w+\frac{1}{2}\Delta w^T(H_t+2\alpha I)\Delta w-\alpha\eta^2\end{aligned}$$

(b)关于Δw求梯度

$$\begin{gathered}\nabla\mathcal{L}=g_t+(H_t+2\alpha I)\Delta w=0\\\Delta w=-(H_t+2\alpha I)^{-1}g_t\end{gathered}$$

关于$\Delta\alpha$求梯度

$$\begin{gathered}\nabla\mathcal{L}=\Delta w^T\Delta w-\eta^2=0\\\Delta w^T\Delta w=\eta^2\end{gathered}$$

(c)对$\Delta w=-(H_t+2\alpha I)^{-1}g_t$两边左乘$H_t+2\alpha I$

$$\begin{gathered}(H_t+2\alpha I)\Delta w=-g_t\\\Delta w^T(H_t+2\alpha I)\Delta w=-\Delta w^T g_t\\\Delta w^T H_t\Delta w+2\alpha\eta^2=-\Delta w^T g_t\\\alpha=-\frac{1}{2\eta^2}(\Delta w^T g_t+\Delta w^T H_t\Delta w)\end{gathered}$$

(d)考虑第一项$-\frac{1}{2\eta^2}\Delta w^T g_t$，求其模长

$$||-\frac{1}{2\eta^2}\Delta w^T g_t||\le\frac{1}{2\eta^2}||\Delta w||||g_t||=\frac{||g_t||}{2\eta}$$

所以第一项为$O(\sim ||g_t||/\eta)$

接着考虑第二项$-\frac{1}{2\eta^2}\Delta w^T H_t\Delta w$，求其模长，记$H_t$中元素绝对值的最大值为$k_1$，最小值为$k_2$

$$\begin{gathered}||-\frac{1}{2\eta^2}\Delta w^T H_t\Delta w||\le\frac{1}{2\eta^2}k_1||\Delta w||^2=\frac{k_1}{2}\\||-\frac{1}{2\eta^2}\Delta w^T H_t\Delta w||\ge\frac{1}{2\eta^2}k_2||\Delta w||^2=\frac{k_2}{2}\end{gathered}$$

所以第二项为$\theta(1)$

(e)对$\Delta w=-(H_t+2\alpha I)^{-1}g_t$进行变形

$$\begin{aligned}\Delta w&=-(H_t+2\alpha I)^{-1}g_t\\&=-\frac{1}{\alpha}(\frac{1}{\alpha}H_t+2I)^{-1}g_t\end{aligned}$$

因为α很大，所以$\frac{1}{\alpha}H_t$可以忽略，从而

$$\Delta w\approx-\frac{1}{\alpha}(2I)^{-1}g_t=-\frac{g_t}{2\alpha}$$

带入$\Delta w^T\Delta w=\eta^2$

$$\begin{aligned}\frac{||g_t||^2}{4\alpha^2}&\approx\eta^2\\\alpha&\approx\frac{||g_t||}{2\eta}\end{aligned}$$

将$\alpha\approx\frac{||g_t||}{2\eta}$代入$\Delta w\approx-(H_t+2\alpha I)^{-1}g_t$可得

$$\Delta w\approx-(H_t+\frac{||g_t||}{\eta}I)^{-1}g_t$$

Problem 7.15 (Page 49)

The outer-product Hessian approximation is $H=\sum_{n=1}^N g_ng_n^T$. Let $H_k=\sum_{n=1}^k g_ng_n^T$ be the partial sum to k , and let H_k^{-1} be its inverse.

(a) Show that $H_{k+1}^{-1}=H_k^{-1}-\frac{H_k^{-1}g_{k+1}g_{k+1}^TH_k^{-1}}{1+g_{k+1}^TH_k^{-1}g_{k+1}}g_{k+1}$. [Hints: $H_{k+1}=H_k+g_{k+1}g_{k+1}^T$; and, $(A+zz^T)^{-1}=A^{-1}-\frac{A^{-1}zz^TA^{-1}}{1+z^TA^{-1}z}$.]

(b) Use part (a) to give an $O(NW^2)$ algorithm to compute H_t^{-1}, the same time it takes to compute H. (W is the number of dimensions in g).

Note: typically, this algorithm is initialized with $H_0=\epsilon I$ for some small ϵ. So the algorithm actually computes $(H+\epsilon I)^{-1}$; the results are not very sensitive to the choice of ϵ, as long as ϵ is small.

(a)利用$(A+zz^T)^{-1}=A^{-1}-\frac{A^{-1}zz^TA^{-1}}{1+z^TA^{-1}z}$以及$H_{k+1}=H_k+g_{k+1}g_{k+1}^T$可得

$$H_{k+1}^{-1}=H_k^{-1}-\frac{H_k^{-1}g_{k+1}g_{k+1}^TH_k^{-1}}{1+g_{k+1}^TH_k^{-1}g_{k+1}}g_{k+1}$$

(b)题目的意思应该是设计一个算法可以在$O(NW^2)$时间内可以计算全部的H_t^{-1}，来分析下上述公式，假设H_k^{-1}已知，首先看分子

$$H_k^{-1}g_{k+1}g_{k+1}^TH_k^{-1}=(g_{k+1}^TH_k^{-1})^T(g_{k+1}^TH_k^{-1})$$

计算$g_{k+1}^TH_k^{-1}$需要$O(W^2)$的时间复杂度，所以计算$H_k^{-1}g_{k+1}g_{k+1}^TH_k^{-1}=(g_{k+1}^TH_k^{-1})^T(g_{k+1}^TH_k^{-1})$需要$O(W^2)+O(W)=O(W^2)$。接着看分母

$$1+g_{k+1}^T H_k^{-1} g_{k+1}$$

计算该分母需要$O(W^2)$的时间复杂度，其余就是向量加减法，需要的复杂度为$O(W)$，所以从H_k^{-1}计算H_{k+1}^{-1}需要的时间复杂度为$O(W^2)$。注意$H_0^{-1}=\frac{1}{\epsilon}I$，是已知的量，所以可以用上述方法递推地求出$H_k^{-1}$，每一项的时间复杂度为$O(W^2)$，因为一共有$N$项，所以一共的时间复杂度为

$$O(NW^2)$$

Problem 7.16 (Page 50)

In the text, we computed an upper bound on the VC dimension of the 2-layer perceptron is $d_{\text{vc}}=O(md\log(md))$ where m is the number of hidden units in the hidden layer. Prove that this bound is essentially tight by showing that $d_{\text{vc}}=\Omega(md)$. To do this, show that it is possible to find md points that can be shattered when m is even as follows. Consider any set of N points $x_1,\ldots,x_N$ in general position with $N=md$. N points in d dimensions are in general position if no subset of $d+1$ points lies on a $d-1$ dimensional hyperplane. Now, consider any dichotomy on these points with r of the points classified $+1$. Without loss of generality, relabel the points so that $x_1,\ldots,x_r$ are $+1$.

(a) Show that without loss of generality, you can assume that $r\le N/2$. For the rest of the problem you may therefore assume that $r\le N/2$.

(b) Partition these r positive points into groups of size d. The last group may have fewer than d points. Show that the number of groups is at most $\frac{N}{2}$. Label these groups $\mathcal{D}_i$ for $i=1\ldots q\le N/2$.

(c) Show that for any subset of k points with $k\le d$, there is a hyperplane containing those points and no others.

(d) By the previous part, let w_i, b_i be the hyperplane through the points in group $\mathcal{D}_i$, and containing no others. So

$$w_i^T x_n+b_i=0$$

if and only if $x_n\in\mathcal{D}_i$. Show that it is possible to find h small enough so that for $x_n\in\mathcal{D}_i$,

$$|w_i^T x_n+b_i|<h$$

and for $x_n\notin\mathcal{D}_i$

$$|w_i^T x_n+b_i|>h$$

(e) Show that for $x_n\in\mathcal{D}_i$,

$$\text{sign}(w_i^T x_n+b_i+h)+\text{sign}(-w_i^T x_n-b_i+h)=2$$

$x_n\notin\mathcal{D}_i$

$$\text{sign}(w_i^T x_n+b_i+h)+\text{sign}(-w_i^T x_n-b_i+h)=0$$

(f) Use the results so far to construct a 2-layer MLP with $2r$ hidden units which implements the dichotomy (which was arbitrary). Complete the argument to show that $d_{\text{vc}}\ge md$.

(a)因为一共有N个点，每个点不是$+1$就是-1，所以至少有一类的数量$\le \frac{N}{2}$，又由$+1,-1$的对称性，所以不妨假设$+1$类的数量$\le \frac{N}{2}$，即

$$r \le \frac{N}{2}$$

(b)组的数量为$\frac{r}{d}$，注意$d \ge 1, r \le \frac{N}{2}$，从而

$$\frac{r}{d} \le \frac{N}{2}$$

(c)一个d维点所在的平面需要$d+1$个参数来确定，如下

$$\begin{gathered} w^T x + b = 0 \\ \sum_{i=1}^{d} w_i x_i = w_0 \end{gathered}$$

参数为$(b, w_1, \ldots, w_d)$，要确定这个$d+1$个参数，至少需要$d+1$个点，所以对于$k \le d$个点，必然存在无数个超平面过这k个点。接下来就要找到经过这k点，但是不经过其他点的超平面，这就要使用题目中的条件：任意$d+1$个点都不在一个$d-1$维的超平面上，分两种情况讨论：

- $k=d$，这种情形直接利用题目中的条件即可。
- $k<d$，假设这k个点为$x_1,\ldots,x_k$，补充$d-k$个点$x_{k+1},\ldots,x_d$，使得

  $$\begin{gathered} w^T x_i + b = 0 (i=1,\ldots,k) \\ w^T x_i + b = 1 (i=k+1,\ldots,d) \end{gathered}$$

  从而$w^T x + b = 0$是一个特殊的平面，过$x_1,\ldots,x_k$，但是不过$x_{k+1},\ldots,x_d$，由题目的假设可知，任意其他的点都不在这个平面上，这说明我们构造了一个只经过这k个点的平面。

结合上述两点可知，存在只经过这k个点的平面。

(d)将(c)的结论用式子写出来

$$\begin{gathered} \text{存在} w_i, b_i\text{，使得} \\ \text{对于每个属于}\mathcal{D}_i\text{的}x_n\text{，}w_i^T x_n + b_i = 0 \\ \text{对于每个不属于}\mathcal{D}_i\text{的}x_n\text{，}w_i^T x_n + b_i \ne 0 \end{gathered}$$

对于$x_n \notin \mathcal{D}_i$，记$h_1 = \min |w_i^T x_n + b_i|, h = \frac{h_1}{2} > 0$，所以

$$\begin{gathered} \text{每个不属于}\mathcal{D}_i\text{的}x_n\text{，}|w_i^T x_n + b_i| \ge h_1 > \frac{h_1}{2} = h > 0 \\ \text{每个属于}\mathcal{D}_i\text{的}x_n\text{，}|w_i^T x_n + b_i| = 0 < h \end{gathered}$$

从而可以找到h满足条件

(e)对于$x_n \in \mathcal{D}_i$

$$\begin{gathered} |w_i^T x_n + b_i| < h \\ w_i^T x_n + b_i < h, -w_i^T x_n - b_i < h \\ -w_i^T x_n - b_i + h > 0, w_i^T x_n + b_i + h > 0 \\ \operatorname{sign}(w_i^T x_n + b_i + h) + \operatorname{sign}(-w_i^T x_n - b_i + h) = 2 \end{gathered}$$

对于$x_n \notin \mathcal{D}_i$

$$
\begin{gathered}
|w_i^T x_n + b_i| > h \\
w_i^T x_n + b_i < -h\text{或}\,w_i^T x_n + b_i > h \\
w_i^T x_n + b_i + h < 0\text{或} - w_i^T x_n - b_i + h < 0 \\
\text{如果}\,w_i^T x_n + b_i + h < 0 \\
\text{那么} - w_i^T x_n - b_i - h > 0, -w_i^T x_n - b_i + h > 2h > 0 \\
\text{所以}\,\mathrm{sign}(w_i^T x_n + b_i + h) + \mathrm{sign}(-w_i^T x_n - b_i + h) = 0 \\
\text{如果} - w_i^T x_n - b_i + h < 0 \\
\text{那么}\,w_i^T x_n + b_i - h > 0, w_i^T x_n + b_i + h > 2h > 0 \\
\text{所以}\,\mathrm{sign}(w_i^T x_n + b_i + h) + \mathrm{sign}(-w_i^T x_n - b_i + h) = 0
\end{gathered}
$$

从而结论成立。

(f)现在构造一个2层的神经网络，假设输入为x，第一层为

$$
\begin{gathered}
\mathrm{sign}(w_i^T x_n + b_i + h)(i = 1, \ldots, r) \\
\mathrm{sign}(-w_i^T x_n - b_i + h)(i = 1, \ldots, r) \\
\text{其中}(w_i, b_i)\text{为之前讨论的每一组点对应的权重。}
\end{gathered}
$$

这样第一层（隐藏层）有$2r$个神经元，记$y_i(i = 1, \ldots, 2r)$，第二层的神经元用如下方法构造

$$
\mathrm{sign}(\sum_{i=1}^{2r} y_i - 1)
$$

我们来分析下这个式子，如果$x_n \in \mathcal{D}_k$，那么

$$
\begin{gathered}
\mathrm{sign}(w_i^T x_n + b_i + h) + \mathrm{sign}(-w_i^T x_n - b_i + h) = \begin{cases} 2, & i = k \\ 0, & i \neq k \end{cases} \\
\sum_{i=1}^{2r} y_i - 1 = 2 - 1 = 1 \\
\mathrm{sign}(\sum_{i=1}^{2r} y_i - 1) = 1
\end{gathered}
$$

如果$x_n \notin \mathcal{D}_k$，那么

$$
\begin{gathered}
\mathrm{sign}(w_i^T x_n + b_i + h) + \mathrm{sign}(-w_i^T x_n - b_i + h) = 0 \\
\sum_{i=1}^{2r} y_i - 1 = 0 - 1 = -1 \\
\mathrm{sign}(\sum_{i=1}^{2r} y_i - 1) = -1
\end{gathered}
$$

这说明这个神经网络可以对任意md个点进行分类，从而

$$
\begin{gathered}
d_{\mathrm{vc}} \geq md \\
d_{\mathrm{vc}} = \Omega(md)
\end{gathered}
$$