Machine Learning

Answer Sheet for Homework 7

Da-Min HUANG
R04942045
Graduate Institute of Communication Engineering, National Taiwan University

January 6, 2016

Problem 1

Set $\mu_- = 1 - \mu_+$, we have

$$
\begin{aligned}
1 - \mu_+^2 - \mu_-^2 &= 1 - \mu_+^2 - (1-\mu_+)^2 = (1-\mu_+)(1+\mu_+) - (1-\mu_+)^2 && (1)\\
&= 2\mu_+(1-\mu_+) = -2\mu_+^2 + 2\mu_+ = -2\left(\mu_+ - \frac{1}{2}\right)^2 + \frac{1}{2} && (2)\\
&\leq \frac{1}{2} && (3)
\end{aligned}
$$

Hence, if $\mu_+ = 1/2 \in [0,1]$, then the maximum value of Gini index is $1/2$.

$\square$

Problem 2

The normalized Gini index is

$$
\frac{\left(1 - \mu_+^2 - \mu_-^2\right)}{\left(\dfrac{1}{2}\right)} = 2\left(1 - \mu_+^2 - \mu_-^2\right) \tag{4}
$$

The squared error can be rewritten as

$$
\begin{aligned}
\mu_+\left(1 - (\mu_+ - \mu_-)\right)^2 + \mu_-\left(-1 - (\mu_+ - \mu_-)\right)^2 &= 4\mu_+(1-\mu_+)^2 + 4\mu_+^2(1-\mu_+) && (5)\\
&= 4\mu_+(1-\mu_+) \leq 4 \times \frac{1}{4} = 1 && (6)
\end{aligned}
$$

Hence the normalized squared error is

$$4\mu_+ (1-\mu_+) = 2\left(2\mu_+ (1-\mu_+)\right) = 2\left((1-\mu_+)(1+\mu_+) - (1-\mu_+)^2\right) \tag{7}$$
$$= 2\left(1-\mu_+^2-\mu_-^2\right) \tag{8}$$

which is equal to normalized Gini index.

□

Problem 3

The probability of one example not sampled is

$$\left(1-\frac{1}{N}\right)^{pN} = \frac{1}{\left(\frac{N}{N-1}\right)^{pN}} = \frac{1}{\left(1+\frac{1}{N-1}\right)^{pN}} = \left(\frac{1}{\left(1+\frac{1}{N-1}\right)^{N}}\right)^{p} \tag{9}$$

As $N \to \infty$, we have

$$\lim_{N\to\infty}\left(\frac{1}{\left(1+\frac{1}{N-1}\right)^{N}}\right)^{p} = \left(\lim_{N\to\infty}\frac{1}{\left(1+\frac{1}{N-1}\right)^{N}}\right)^{p} = \left(\frac{1}{e}\right)^{p} = e^{-p} \tag{10}$$

So there approximately $e^{-p}\cdot N$ of the examples not sampled.

□

Problem 4

Since $G = \text{Uniform}\left(\{g_k\}_{k=1}^{3}\right)$, so if at least two terms of $\{g_k\}_{k=1}^{3}$ ouput wrong result, then G outputs wrong result. Let $\{E_k\}_{k=1}^{3}$ be the set of examples that $\{g_k\}_{k=1}^{K}$ got wrong results. Apparently $|E_3| > |E_2| > |E_1|$ and $|E_1| + |E_2| > |E_3|$. So

1. Maximum of $E_{\text{out}}(G)$ happens at $E_3 \subset (E_1 \cup E_2)$. Then G outputs wrong result in the region of E_3 with $E_{\text{out}}(G) = 0.35$.

2. Minimum of $E_{\text{out}}(G)$ happens at $E_i \cap E_j = \phi$, $i \neq j$ and $1 \leq i, j \leq 3$ with $i, j \in \mathbb{N}$. Then G always ouputs the correct result since $(E_1 \cup E_2 \cup E_3) \subset \{\text{all examples}\}$.

Hence, $0 \leq E_{\text{out}}(G) \leq 0.35$.

□

Problem 5

Since $G = \text{Uniform}\left(\{g_k\}_{k=1}^K\right)$, so if at least $(K+1)/2$ terms of $\{g_k\}_{k=1}^K$ ouput wrong result, then G outputs wrong result. Let $\{E_k\}_{k=1}^K$ be the set of examples that $\{g_k\}_{k=1}^K$ got wrong results.

If G outputs wrong result on some example $\mathbf{x}$, then we have

$$\mathbf{x} \in \bigcap_{i=1}^{\left(\frac{K+1}{2}\right)+m} E_{\alpha_i} \tag{11}$$

where $\alpha_i \in \{1, 2, \ldots, K\}$ satisfies $1 \leq \alpha_i \leq K$ and $m \in (\mathbb{N} \cup \{0\})$ with $0 \leq m < K + 1/2$. And

$$\left| \bigcap_{i=1}^{\left(\frac{K+1}{2}\right)+m} E_{\alpha_i} \right| \leq \frac{2}{K+1+2m} \sum_{k=1}^{K} e_k \leq \frac{2}{K+1} \sum_{k=1}^{K} e_k \tag{12}$$

(12) holds due to

$$\bigcap_{i=1}^{\left(\frac{K+1}{2}\right)+m} E_{\alpha_i} \subseteq E_\beta \text{ with } |E_\beta| \leq |E_{\alpha_i}|, \ \forall i \tag{13}$$

where β is some index such that $|E_\beta| = \min_{\alpha_i} |E_{\alpha_i}|$. This is because the biggest intersection is the smallest subset. So

$$\left(\frac{K+1}{2} + m\right) \left| \bigcap_{i=1}^{\left(\frac{K+1}{2}\right)+m} E_{\alpha_i} \right| \leq \left(\frac{K+1}{2} + m\right) |E_\beta| \leq \sum_{i=1}^{\left(\frac{K+1}{2}\right)+m} |E_{\alpha_i}| \leq \sum_{k=1}^{K} |E_k| \tag{14}$$

(14) holds since size of E_β is the samllest among $\left(\left(\frac{K+1}{2}\right) + m\right)$ terms and $\sum_{k=1}^K |E_k|$ must contains the $\left(\left(\frac{K+1}{2}\right) + m\right)$ terms. Hence, we have

$$E_{\text{out}}(G) \leq \frac{2}{K+1+2m} \sum_{k=1}^{K} e_k \leq \frac{2}{K+1} \sum_{k=1}^{K} e_k \tag{15}$$

where $|E_k| = e_k$ by definition.

$\square$

Problem 6

By the definition of U_t, we have

$$
\begin{aligned}
U_{t+1} &= \frac{1}{N}\sum_{n=1}^{N}\exp\left(-y_n\sum_{\tau=1}^{t}\alpha_\tau g_\tau\left(\mathbf{x}_n\right)\right) && (16)\\
&= \frac{1}{N}\sum_{n=1}^{N}\exp\left(-y_n\sum_{\tau=1}^{t-1}\alpha_\tau g_\tau\left(\mathbf{x}_n\right) - y_n\alpha_t g_t\left(\mathbf{x}_n\right)\right) && (17)\\
&= \frac{1}{N}\sum_{n=1}^{N}\exp\left(-y_n\sum_{\tau=1}^{t-1}\alpha_\tau g_\tau\left(\mathbf{x}_n\right)\right)\exp\left(-y_n\alpha_t g_t\left(\mathbf{x}_n\right)\right) && (18)\\
&= \sum_{n=1}^{N}u_n^{(t)}\exp\left(-y_n\alpha_t g_t\left(\mathbf{x}_n\right)\right) && (19)\\
&= \sum_{\substack{n\\ y_n\neq g_t(\mathbf{x}_n)}}u_n^{(t)}\exp\left(-y_n\alpha_t g_t\left(\mathbf{x}_n\right)\right) + \sum_{\substack{n\\ y_n= g_t(\mathbf{x}_n)}}u_n^{(t)}\exp\left(-y_n\alpha_t g_t\left(\mathbf{x}_n\right)\right) && (20)\\
&= \sum_{\substack{n\\ y_n\neq g_t(\mathbf{x}_n)}}u_n^{(t)}\exp\left(\alpha_t\right) + \sum_{\substack{n\\ y_n= g_t(\mathbf{x}_n)}}u_n^{(t)}\exp\left(-\alpha_t\right) && (21)\\
&= \exp\left(\alpha_t\right)\left(\epsilon_t\right)\sum_{n=1}^{N}u_n^{(t)} + \exp\left(-\alpha_t\right)\left(1-\epsilon_t\right)\sum_{n=1}^{N}u_n^{(t)} && (22)\\
&= U_t\left(\exp\left(\alpha_t\right)\left(\epsilon_t\right) + \exp\left(-\alpha_t\right)\left(1-\epsilon_t\right)\right) = U_t\cdot 2\sqrt{\epsilon_t\left(1-\epsilon_t\right)} && (23)
\end{aligned}
$$

Since

$$
U_1 = \sum_{n=1}^{N}u_n^{(1)} = \sum_{n=1}^{N}\frac{1}{N} = 1 \tag{24}
$$

we have

$$
\begin{aligned}
U_3 &= U_2\cdot 2\sqrt{\epsilon_2\left(1-\epsilon_2\right)} = \left(U_1\cdot 2\sqrt{\epsilon_1\left(1-\epsilon_1\right)}\right)\cdot 2\sqrt{\epsilon_2\left(1-\epsilon_2\right)} && (25)\\
&= 4\sqrt{\epsilon_1\epsilon_2\left(1-\epsilon_1\right)\left(1-\epsilon_2\right)} && (26)
\end{aligned}
$$

which can be generalized as

$$
U_{T+1} = \prod_{t=1}^{T}\left(2\sqrt{\epsilon_t\left(1-\epsilon_t\right)}\right) \tag{27}
$$

□

Problem 7

To compute s_n, we need to find the optimal η of

$$\min_{\eta} \frac{1}{N} \sum_{n=1}^{N} \left((y_n - s_n) - \eta g_t (\mathbf{x}_n) \right)^2 := A \tag{28}$$

From $\partial A / \partial \eta = 0$, we have

$$\eta = \frac{\sum_{n=1}^{N} g_t (\mathbf{x}_n) (y_n - s_n)}{\sum_{n=1}^{N} g_t^2 (\mathbf{x}_n)} \tag{29}$$

Now $s_n = 0$ and $g_1 (\mathbf{x}) = 2$, so

$$\eta = \frac{2 \sum_{n=1}^{N} y_n}{4 \sum_{n=1}^{N}} = \frac{1}{2N} \sum_{n=1}^{N} y_n \tag{30}$$

Since $\eta = \alpha_1$, so

$$\alpha_1 g_1 (\mathbf{x}_n) = \frac{2}{2N} \sum_{n=1}^{N} y_n = \frac{1}{N} \sum_{n=1}^{N} y_n = s_n \tag{31}$$

$\square$

Problem 8

From the equatio of optimal η, we have

$$\eta = \frac{\sum_{n=1}^{N} g_t (\mathbf{x}_n) (y_n - s_n')}{\sum_{n=1}^{N} g_t^2 (\mathbf{x}_n)} = \frac{\sum_{n=1}^{N} y_n g_t (\mathbf{x}_n) - \sum_{n=1}^{N} s_n' g_t (\mathbf{x}_n)}{\sum_{n=1}^{N} g_t^2 (\mathbf{x}_n)} = \alpha_t \tag{32}$$

so

$$\sum_{n=1}^{N} y_n g_t (\mathbf{x}_n) - \sum_{n=1}^{N} s_n' g_t (\mathbf{x}_n) = \alpha_t \sum_{n=1}^{N} g_t^2 (\mathbf{x}_n) = \sum_{n=1}^{N} \alpha_t g_t^2 (\mathbf{x}_n) = \sum_{n=1}^{N} (s_n - s_n') g_t (\mathbf{x}_n) \tag{33}$$

where s_n' is defined as the s_n in iteration $(t-1)$ and $s_n = s_n' + \alpha_t g_t (\mathbf{x}_n)$, so

$$\sum_{n=1}^{N} s_n g_t (\mathbf{x}_n) = \sum_{n=1}^{N} y_n g_t (\mathbf{x}_n) \tag{34}$$

$\square$

Problem 9

OR $(x_1, x_2, \ldots, x_d)$ means outputs TRUE if one input is TRUE; outputs FALSE if all inputs are FALSE.

Claim: $(w_0, w_1, \ldots, w_d) = (d-1, 1, \ldots, 1)$ implements OR.

Proof of Claim:

1. If all $x_i = -1$, then we have

$$\operatorname{sign}\left(\sum_{i=0}^{d} w_i x_i\right) = \operatorname{sign}\left(d - 1 + \sum_{i=1}^{d} (-1)\right) = \operatorname{sign}(-1) = \text{FALSE} \tag{35}$$

2. If some $x_i = +1$ and others are -1, we have

$$\operatorname{sign}\left(\sum_{i=0}^{d} w_i x_i\right) = \operatorname{sign}(d - 1 + 1 + (-1)(d-1)) = \operatorname{sign}(+1) = \text{TRUE} \tag{36}$$

Hence, $(w_0, w_1, \ldots, w_d) = (d-1, 1, \ldots, 1)$ implements OR.

$\square$

Problem 10

Claim: $D \geq 5$.

Proof of Claim:

The weights of hidden layer and output layer are

$$\text{Neuron 1}: \left(w_{01}^{(1)}, w_{11}^{(1)}, w_{21}^{(1)}, w_{31}^{(1)}, w_{41}^{(1)}, w_{51}^{(1)}\right) = (4, 1, 1, 1, 1, 1) \tag{37}$$
$$\text{Neuron 2}: \left(w_{02}^{(1)}, w_{12}^{(1)}, w_{22}^{(1)}, w_{32}^{(1)}, w_{42}^{(1)}, w_{52}^{(1)}\right) = (2, 1, 1, 1, 1, 1) \tag{38}$$
$$\text{Neuron 3}: \left(w_{03}^{(1)}, w_{13}^{(1)}, w_{23}^{(1)}, w_{33}^{(1)}, w_{43}^{(1)}, w_{53}^{(1)}\right) = (0, 1, 1, 1, 1, 1) \tag{39}$$
$$\text{Neuron 4}: \left(w_{04}^{(1)}, w_{14}^{(1)}, w_{24}^{(1)}, w_{34}^{(1)}, w_{44}^{(1)}, w_{54}^{(1)}\right) = (-2, 1, 1, 1, 1, 1) \tag{40}$$
$$\text{Neuron 5}: \left(w_{05}^{(1)}, w_{15}^{(1)}, w_{25}^{(1)}, w_{35}^{(1)}, w_{45}^{(1)}, w_{55}^{(1)}\right) = (-4, 1, 1, 1, 1, 1) \tag{41}$$
$$\text{Output Neuron}: \left(w_{01}^{(2)}, w_{11}^{(2)}, w_{21}^{(2)}, w_{31}^{(2)}, w_{41}^{(2)}, w_{51}^{(2)}\right) = (1/5\ , 1, -1, 1, -1, 1) \tag{42}$$

The values table is

Number of 1 in Inputs	Hidden Layer Output	Output
0	$(-1,-1,-1,-1,-1)$	-1
1	$(+1,-1,-1,-1,-1)$	+1
2	$(+1,+1,-1,-1,-1)$	-1
3	$(+1,+1,+1,-1,-1)$	+1
4	$(+1,+1,+1,+1,-1)$	-1
5	$(+1,+1,+1,+1,+1)$	+1

The condition of $D = 5$ is the smallest choice left to be proven in Problem 21.

$\square$

Problem 11

Claim: Only the gradient components with respect to $w_{01}^{(L)}$ may be non-zero, all other gradient components must be zero.

Proof of Claim:

Consider

$$\frac{\partial e_n}{\partial w_{i1}^{(L)}} = -2\left(y_n - s_1^{(L)}\right)\left(x_i^{(L-1)}\right) = -2\left(y_n - \sum_{j=0}^{d^{(L-1)}} w_{j1}^{(L)} x_j^{(L-1)}\right)\left(x_i^{(L-1)}\right) \tag{43}$$

Since all $w_{ij}^{(\ell)} = 0$, we have

$$\frac{\partial e_n}{\partial w_{i1}^{(L)}} = -2y_n x_i^{(L-1)} \tag{44}$$

If $i \neq 0$, then

$$\frac{\partial e_n}{\partial w_{i1}^{(L)}} = -2y_n x_i^{(L-1)} = -2y_n \tanh\left(s_i^{(L-1)}\right) = -2y_n \tanh\left(\sum_{j=0}^{d^{(L-2)}} w_{ji}^{(L-1)} x_j^{(L-2)}\right) = 0 \tag{45}$$

since all $w_{ij}^{(\ell)} = 0$.

If $i = 0$, then

$$\frac{\partial e_n}{\partial w_{01}^{(L)}} = -2y_n x_0^{(L-1)} = -2y_n \tag{46}$$

If $y_n \neq 0$, then

$$\frac{\partial e_n}{\partial w_{01}^{(L)}} \neq 0 \tag{47}$$

Similarly, we have

$$\frac{\partial e_n}{\partial w_{0j}^{(\ell)}} = \delta_j^{(\ell)} \left(x_0^{(\ell-1)} \right) = \delta_j^{(\ell)} = \sum_k \left(\delta_k^{(\ell+1)} \right) \left(w_{jk}^{(\ell+1)} \right) \left(\frac{\partial \tanh \left(s_j^{(\ell)} \right)}{\partial s_j^{(\ell)}} \right) = 0 \tag{48}$$

since all $w_{ij}^{(\ell)} = 0$.

□

Problem 12

For $\ell = 1$ and all $w_{ij}^{(\ell)}$ initialized as 1, we have

$$\eta x_i^{(0)} \delta_j^{(1)} = \eta x_i^{(0)} \sum_k \left(\delta_k^{(2)} \right) \left(w_{jk}^{(2)} \right) \left(\frac{\partial \tanh \left(s_j^{(1)} \right)}{\partial s_j^{(1)}} \right) \tag{49}$$

$$= \eta x_i^{(0)} \sum_k \left(\delta_k^{(2)} \right) \left(\frac{\partial \tanh \left(s_j^{(1)} \right)}{\partial s_j^{(1)}} \right) \tag{50}$$

and

$$s_j^{(1)} = \sum_{i=0}^{d^{(0)}} w_{ij}^{(1)} x_i^{(0)} = \sum_{i=0}^{d^{(0)}} x_i^{(0)} \tag{51}$$

So we have

$$s_1^{(1)} = s_2^{(1)} = \cdots = s_{d^{(1)}}^{(1)} \Rightarrow \delta_1^{(1)} = \delta_2^{(1)} = \cdots = \delta_{d^{(1)}}^{(1)} \tag{52}$$

Hence, all update term of $w_{ij}^{(1)}$ is the same for $1 \leq j \leq d^{(1)}$, so

$$w_{ij}^{(1)} = w_{i(j+1)}^{(1)} \tag{53}$$

for $1 \leq j \leq d^{(1)} - 1$.

□

Problem 13

The rules of following tree are

1. (feature column, s, θ). The meaning of combination of numbers.
2. If the feature of $\mathbf{x}$ is smaller than θ, then go to the left tree; if not, then go to the right tree.

3. If go to left and there is no more node, then return $s \times (+1)$; if go to the right and there is no more node, return $s \times (-1)$, where s is from the last node.

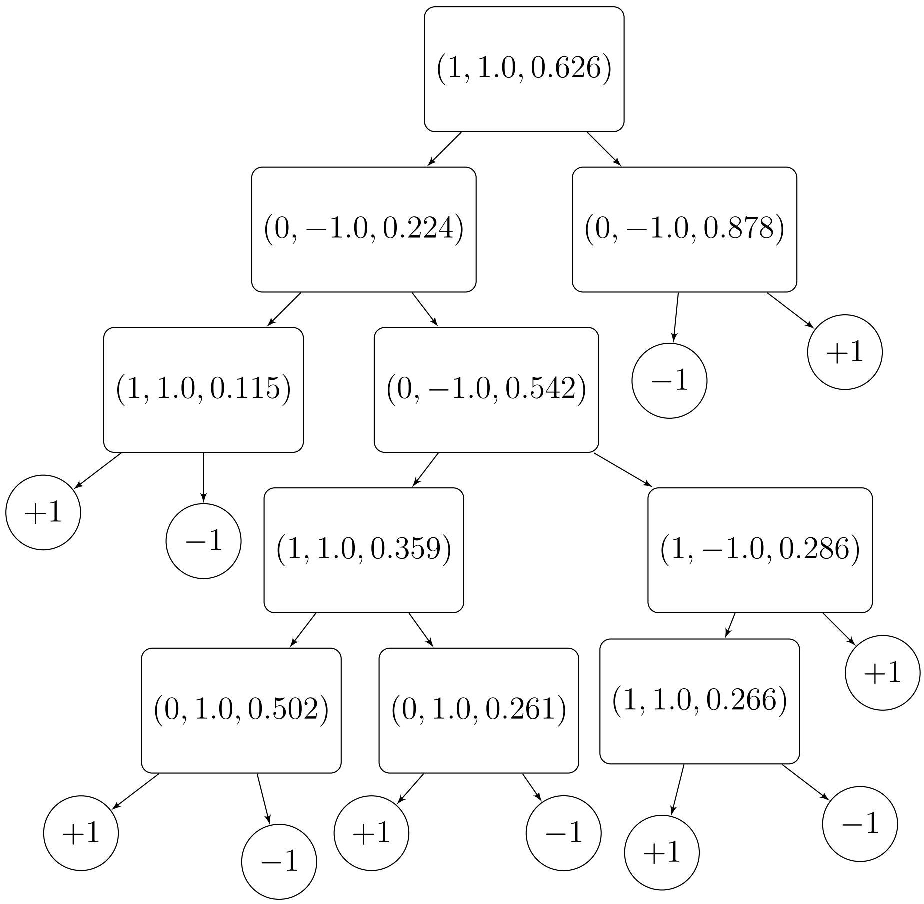

Figure 1: Tree Graph

□

Problem 14

$E_{\text{in}} = 0.0$.

□

Problem 15

$E_{\text{out}} = 0.126$.

□

Problem 16

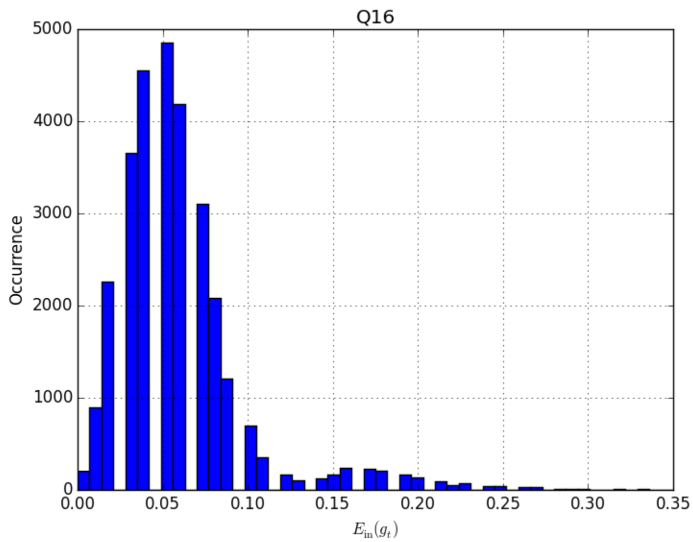

Figure 2: Q16

The average $E_{\text{in}}(g_t)$ of total 30,000 trees is 0.0593.

$\square$

Problem 17

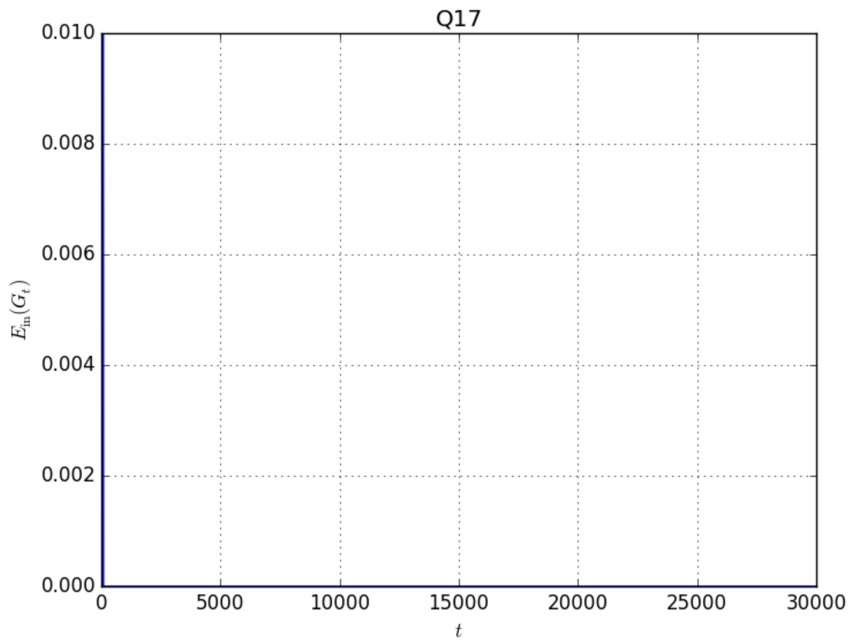

Figure 3: Q17

The average $E_{\text{in}}(G_t)$ of total 30,000 trees is 0. Since most values are 0 so the figure seems nothing left.

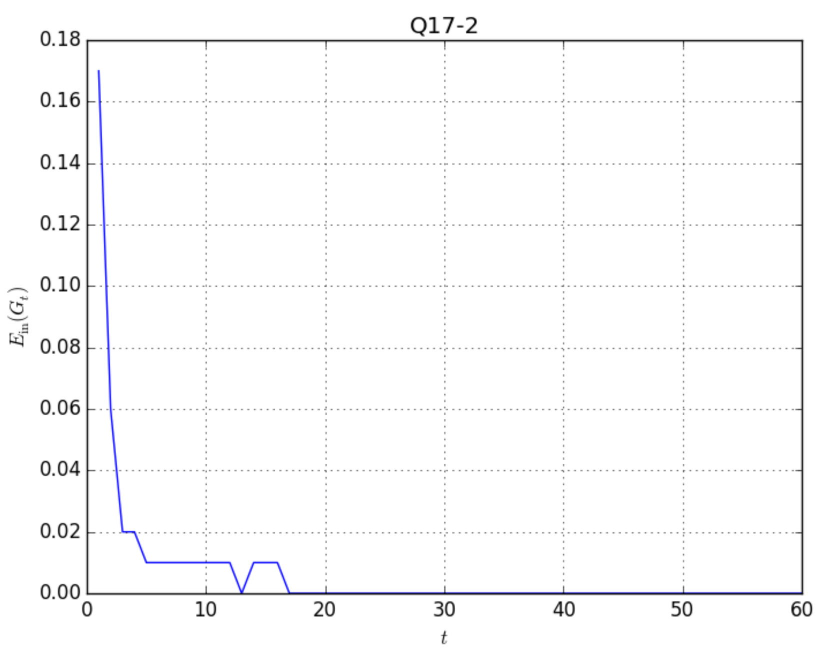

Figure 4: Q17 first 60 points

The figure of first 60 points. We can see that after the 20^{th} point, $E_{\text{in}}(G_t)$ is almost 0.

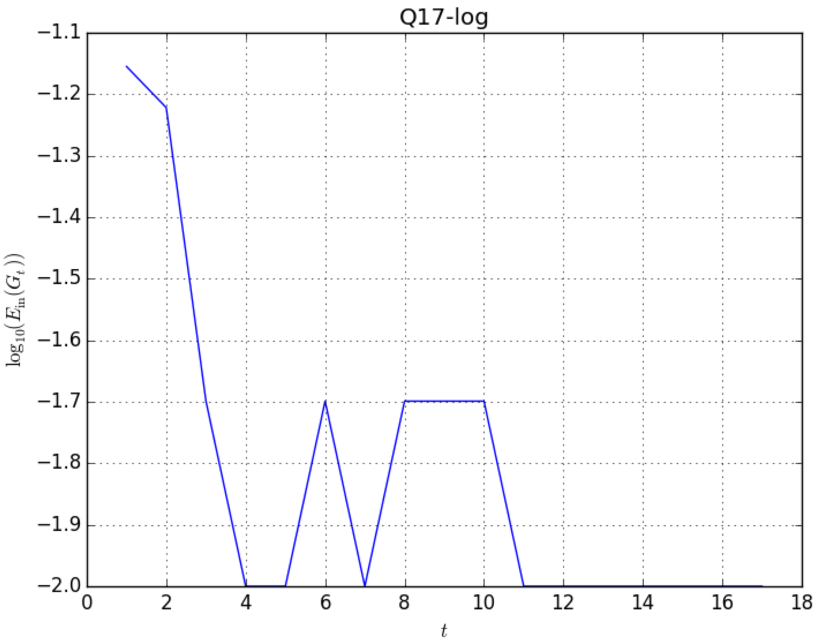

Figure 5: Q17 $\log_{10}(E_{\text{in}}(G_t))$

The figure of log value of $E_{\text{in}}(G_t)$ (removed $E_{\text{in}}(G_t) = 0$ points).
We can see that most of the $E_{\text{in}}(G_t) = 0$ (only 17 points left).

$\square$

Problem 18

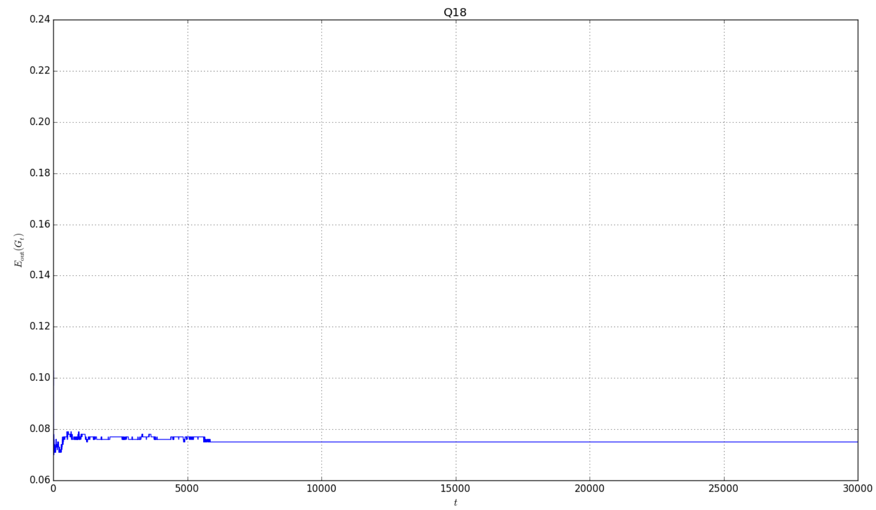

Figure 6: Q18

The average $E_{\text{out}}\left(G_t\right)$ of total 30,000 trees is 0.0753.
This curve approaches the average value as $t \to 30,000$. The curve of Problem 17 goes to 0 after t is near 20. Both curves converge to the average value as t grows.

$\square$

Problem 19

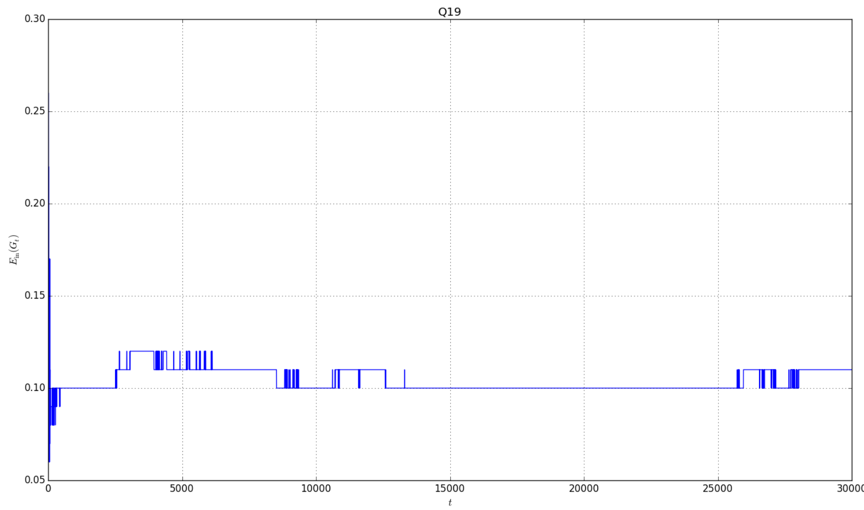

Figure 7: Q19

The average $E_{\text{in}}(G_t)$ of total 30,000 trees is 0.1042.

□

Problem 20

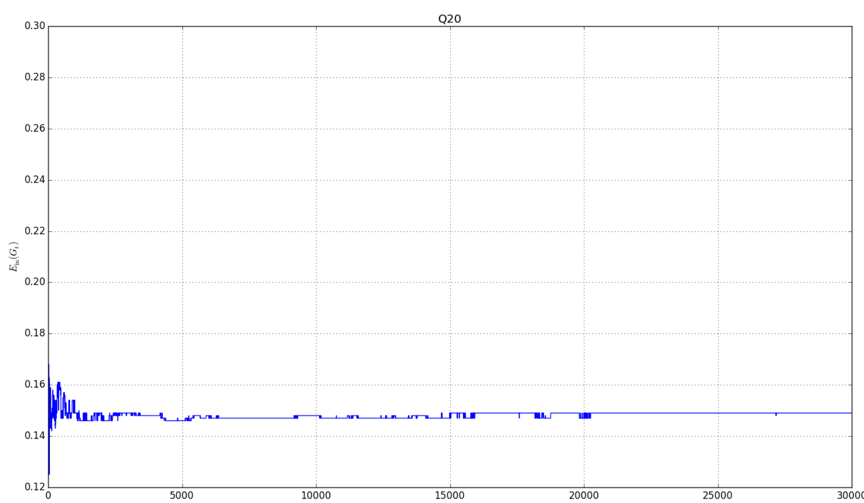

Figure 8: Q20

The average $E_{\text{out}}(G_t)$ of total 30,000 trees is 0.1483.

This curve approaches the average value as $t \to 30,000$. The curve of Problem 19 oscillates between the average value.

$\square$

Problem 21

Map all possible inputs to D-dimension hypercube convex points. For example, $D = 2$, all possible inputs are $(-1,-1)$, $(+1,-1)$, $(-1,+1)$ and $(+1,+1)$. Then we can construct a square lies on x-y plane with convex points $(-1,-1)$, $(+1,-1)$, $(-1,+1)$ and $(+1,+1)$. For a N-dimensional hypercube, the value of each XOR (convex point) is

1. 2-D: $(\mathrm{XOR}\,(-1,-1)\,, \mathrm{XOR}\,(+1,-1)\,, \mathrm{XOR}\,(-1,+1)\; \mathrm{XOR}\,(+1,+1)) = (-1,+1,+1,-1)$
2. 3-D: $\cdots = (-1,+1,+1,+1,-1,-1,-1,+1)$

$\vdots$

It is a little messy. Group the values by seperating $+1$ and -1, we have

1. 2-D: $\underbrace{(-1)}_{0 \text{ input } +1}, \underbrace{(+1,+1)}_{1 \text{ input } +1}, \underbrace{(-1)}_{2 \text{ input } +1}$
2. 3-D: $\underbrace{(-1)}_{0 \text{ input } +1}, \underbrace{(+1,+1,+1)}_{1 \text{ input } +1}, \underbrace{(-1,-1,-1)}_{2 \text{ input } +1}, \underbrace{(+1)}_{3 \text{ input } +1}$

$\vdots$

We can see that n-dimension can be seperated to $(n+1)$ groups, where $1 \leq n \leq d$, $n \in \mathbb{N}$. Each group represents different output of $+1$ or -1. For example, $D = 2$, $\mathrm{XOR}\,(-1,+1) = \mathrm{XOR}\,(+1,-1) = 1$ so the two points (inputs) are in the same group.

Consider the orthogonal projection of of all convex points onto hyperline $x_1 = x_2 = \cdots = x_d$, we can find that the points in the same group will be projected onto the same point. For example, $D = 2$, the projections of $(-1,+1)$ and $(+1,-1)$ onto $y = x$ are both $(0,0)$. This can be proved by calculating the shortest distance to the hyperline for all points with same number of $+1$ inputs,

$$\text{Shortest Distance} = \sum_{i=1}^{k} (x_i - 1)^2 + \sum_{j=1}^{d-k} (x_j + 1)^2 = d\,(x-1)^2 + (d-k)\,(x+1)^2 \quad (54)$$

where k is the number of $+1$ of some input and set $x_1 = x_2 = \cdots = x_d = x$. So the projections of all points with same number of $+1$ inputs onto the hyperline is the same point.

The d-D-1 neural network is trying to classify all these inputs into different groups. One neuron in the hidden layer with $(d+1)$ weights $\left(w_{0j}^{(\ell)}, w_{1j}^{(\ell)}, \ldots, w_{dj}^{(\ell)}\right)$ can only construct a hyperplane in d-dimension (since one weight is from constant input), like a classfier, seperating the convex points of hypercube into different group. And d-dimension hypercube needs at least d hyperplanes to seperate all convex points (inputs) into $(d+1)$ groups (output).

For example, $D = 2$, then we have three points $(-1,-1)$, $(0,0)$ and $(+1,+1)$ after projection. We can find the hyperplane $x + y = 1$ and $-x - y = 1$ with corresponding neurons of $(+1,+1,+1)$ and $(+1,-1,-1)$ in hidden layer, neuron of $(-1,+1,+1)$ in output layer. These hyperplanes can seperate all convex points into three groups. So we can implement $\mathrm{XOR}\,(x_1, x_2)$. Similarly, we can implement $\mathrm{XOR}\,(x_1, x_2, \ldots, x_d)$.

Hence, we need at least $D = d$ to implement $\mathrm{XOR}\,(x_1, x_2, \ldots, x_d)$ with d-D-1 feed-forward neural network.

$\square$

Problem 22

Claim: If d is odd, then $D \geq ((d+1)/2)$; if d is even, then we have $D \geq (d/2) + 1$. Two cases can be implemented by d-$(D-1)$-1 feed-forward neuron network.

Proof of Claim:

From the groups seperated in Problem 21, we can find that they are symmetric. If we can connect all inputs directly to output neuron, we can specify which side are we on by ADD all inputs to see if $\sum_{i=1}^{d} x_i \lessgtr 0$ (for d is odd) or $\sum_{i=1}^{d} x_i \lesseqqgtr 0$ (for d is even).

For example, $D = 2$, we have

$$\underbrace{(-1)}_{\text{0 input }+1} \Bigg| \underbrace{(+1,+1)}_{\text{1 input }+1} \Bigg| \underbrace{(-1)}_{\text{2 input }+1} \quad \Rightarrow \quad \underbrace{(+1,+1)}_{\text{Inputs summation is 0}} \Bigg| \underbrace{(-1)}_{\text{Inputs summation is } \lessgtr 0} \tag{55}$$

or $D = 3$,

$$(-1)\Big|(+1,+1,+1)\Big|(-1,-1,-1)\Big|(+1) \Rightarrow \underbrace{(-1,-1,-1)\Big|(+1)}_{\text{Inputs summation} > 0} \tag{56}$$

$$\text{or } \underbrace{(+1,+1,+1)\Big|(-1)}_{\text{Inputs summation} < 0}$$

Hence we can reduce the number of neurons in hidden layer by half, which is $(d-1)/2$ for d is odd and $(d/2)$ for d is even, where $(d-1)/2$ is due to we do not need the central hyperplane to seperate the inputs.
Plus the neuron in output layer, we have $D \geq ((d+1)/2)$ if d is odd and $D \geq (d/2)+1$ if d is even.

$\square$

Reference

[1] Lecture Notes by Hsuan-Tien LIN, Department of Computer Science and Information Engineering, National Taiwan University, Taipei 106, Taiwan.