Theory Questions

Yoav Shoshan

(1)

$$f^{(T+1)}(z_t) = z_{t+1} = h(W^{(T+1)} z_t + b^{(1)})$$

$$h(z)_i = \text{sign}(z_i)$$

$$t \in \mathcal{F}$$

(a) we saw that for H which is the class of all linear separators in $\mathbb{R}^d$:

I $VC\dim(H) = d+1$

So from Saher-Shelah we have for $n \geq d+1$

$$\Pi_H(n) \leq \left(\frac{en}{d+1}\right)^{d+1}$$

II we also proved (Ex. 4) that for concatination $F = F_1 \times F_2$

$$\Pi_F(n) \leq \Pi_{F_1}(n) \cdot \Pi_{F_2}(n)$$

generalizing it to t which is a cartesian product of linear separators

$$t \in \{l_1 \times l_2 \dots l_d \mid l_i \in H\}$$

by induction: define $P^{(k)} = \{l_1 \times l_2 \dots l_k \mid l \in H\}$

then the induction assumption:

$$\Pi_{P_k} \leq (\Pi_H(n))^k$$

now define $P^{(k+1)} = t^{(k)} \times l' \quad l' \in H$

$$\Rightarrow \Pi_{\rho^{k+1}} \le \Pi_{\rho^k} \cdot \Pi_{\substack{\rho \\ l \in F}} \le \left[\Pi_H(n)\right]^k \Pi_L(n)$$

$$= \Pi_H(n)^{k+1}$$

$\Rightarrow$ we have for t:

$$\Pi_F(n) \le (\Pi_H(n))^d \le \left(\frac{en}{d+1}\right)^{(d+1)d}$$

(b) We saw that for composition.

$F = F_1 \circ F_2 \Rightarrow$

$\Pi_F(n) \le \Pi_{F_1}(n) \Pi_{F_2}(n)$. so if we have a function class C:

$C = H \circ F_1 \circ F_2 \dots F_{L-1}$ we can generalize the lemma above (easily as before, by induction)

$$\Pi_{C=H \circ F_1 \circ F_2 \dots F_{L-1}} \le \Pi_H(n) \cdot \Pi_{F_1}(n) \dots \Pi_{F_{L-1}}(n)$$

$$\le \left(\frac{en}{d+1}\right)^{d+1} \left(\frac{en}{d+1}\right)^{(d+1)d(L-1)} = \left(\frac{en}{d+1}\right)^{(d+1)(dL-d+1)}$$

(c) each hidden layer is transformed to the next hidden layer by a function t which is composed defined as the sign of an input calculated with a matrix $W \in \mathbb{R}^{d \times d}$ and a vector $b \in \mathbb{R}^d$ so that $t: \mathbb{R}^d \to \mathbb{R}^d$. $\Rightarrow$ $d^2 + d$ for each hidden layer. the last layer though,

is calculated by a function
$f^L : R^d \to R$ and for that we need
a vector $w^L \in R^d$ and a bias number
$b^L \in R$

$f^{L+1} = h(w^L z_L + b^L)$ so we actually need $d+1$ parameters for that layer

we thus have in total

$$\underbrace{(L-1)(d^2+d)}_{\text{all hidden layers}} + \underbrace{d+1}_{\text{last layer}} = (L-1)d^2 + Ld + 1 \text{ parameters}$$

(d) first we notice that for two events A, B.

$A \Rightarrow B$ iff $\bar{B} \Rightarrow \bar{A}$

so $\left(2^n \le (en)^N \Rightarrow 2N\log_2(eN) \ge n\right)$

$\Leftrightarrow \left(n > 2N\log_2(eN) \Rightarrow 2^n > (en)^N\right)$

let's begin: $2^n > (en)^N$

$$n > N\log_2(en)$$

$$\frac{n}{N\log_2(en)} > 1$$

define $f(x) = \frac{x}{N\log_2(ex)}$

$$\frac{dt}{dx} = \frac{1}{N}\left(\frac{\log_2(ex) - \log_2(e)}{(\log_2(ex))^2}\right) \Longrightarrow$$

$$\left.\frac{dt}{dx}\right|_{x>1} > 0$$

$\Longrightarrow$ for $x_2 > x_1 \Rightarrow f(x_2) > f(x_1)$

So we can set $x_2 = n$

$x_1 = 2N\log_2(eN)$

and because we started from $x_2 > x_1$

we have $f(x_2) > f(x_1) \Longrightarrow$

** $\frac{n}{N\log_2(en)} > \frac{2N\log_2(eN)}{N\log_2(e\,2N\log_2(eN))} =$

$$\frac{2\log_2 eN}{1 + \log_2(eN) + \log_2(\log_2(eN))} \overset{*}{>} 1$$

* we shall prove

$\log_2(eN) > 1 + \log_2(\log_2(eN))$

$\Longrightarrow eN > 2\log_2 eN$

$e > \log_2(2e)$ (True for $N=2$)

$e - \log_2(2e) > 0$

define $a(x) = x2 - 2\log_2(ex)$

$g'(x) = e - \frac{2}{x\ln(2)}$

$g'(x \geq 2) > 0$

$\Longrightarrow$ for $N \geq 2$:

$$eN > 2\log_2 eN$$

$\overset{**}{\Longrightarrow}$ $\dfrac{n}{N\log_2(en)} > 1$

$$n > \log_2(en)^N \Longrightarrow$$

$$2^n > (en)^N \quad \text{(for } N \geq 2\text{)}$$

(for $N=1$ $2^n > en$ for $n > 3$)

$\Longrightarrow$ for $N > 1$ or $N = 1$ $n > 3$

we proved that

$$2^n \leq (en)^N \longrightarrow 2N\log_2(eN) \geq n$$ //

(e) $\Pi_C(n) \leq \left(\dfrac{en}{d+1}\right)^{(d+1)(dL-d+1)}$ (from b)

from C we have $N = (L-1)d^2 + Ld + 1$

$= (d+1)(dL-d+1)$

$$\Longrightarrow \Pi_C(n) \leq \left(\frac{en}{d+1}\right)^N \leq (en)^N$$

$\uparrow$ $d \geq 0$

$$Vcdim(C) \geq Vcdim(H) = d+1$$

$(H \subseteq C)$

all linear classifiers could be implemented with $L=1$

$$\Rightarrow \quad 2^{VCdim(C)} = \Pi_C(VCdim(C)) \leq (e \cdot VCdim(C))^N$$

$$\xrightarrow[\text{from } d]{} \quad VCdim(C) \leq 2N \log_2(eN)$$

(2) Data

X_1	X_2	X_3	
1	1	1	1
1	0	0	1
1	1	0	0
0	0	1	0

(a) According to the ID3 Algorithm each split j will maximize $G(S,i)$

$j = \arg\max_{i \in A} G(S,i)$ A is all possible splits.

$G(S,i) = H(Y) - H(Y|X_i)$ and

$H(Y) = -\sum_i P(y_i) \log(P(y_i))$

Before the first split we have $P(y=1) = P(y=0) = \frac{1}{2}$

$\Rightarrow H(Y) = (-\frac{1}{2}\log\frac{1}{2}) \cdot 2 = 1$

now we shall calculate $H(Y|X_i)$ for all i's

$H(Y|X_1) = H(Y|X_1=1)P(X_1=1) + H(Y|X_1=0)P(X_1=0)$

$= \frac{3}{4}\left(-\frac{1}{3}\log\frac{1}{3} - \frac{2}{3}\log\frac{2}{3}\right) + \frac{1}{4}(-1\log 1) \approx 0.69$

$H(Y|X_2) = \frac{1}{2}\left(-\frac{1}{2}\log\frac{1}{2}\right)\cdot 2\right) \cdot 2 = 1$

$H(Y|X_3) = \ldots = 1$

$\Rightarrow j = \arg\max_{i \in \{1,2,3\}} G(S,i) = 1$

$G(S,1) = 1 - 0.69 = 0.31$

the first split is by X_1 and we have two groups

$S_0 = \{((0\ 0\ 1), 0)\}$

$S_1 = \{((1\ 1\ 1), 1), ((1\ 0\ 0), 1), ((1\ 1\ 1), 0)\}$

for S_0, it doesn't matter which $j \in \{2,3\}$ we choose for the next split and the post split error will be 0 anyway.

we examine S_1 then

$$H(Y) = -\frac{2}{3}\log\frac{2}{3} - \frac{1}{3}\log\frac{1}{3}$$

$$H(Y|X_2) = H(Y|X_2=1)P(X_2=1) + H(Y|X_2=2)P(X_2=2)$$

$$\frac{2}{3}\left[-\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2}\right) + \frac{1}{3}\left[-1\log 1 - 0\log 0\right) = \frac{2}{3}$$

$$H(Y|X_3) = \frac{2}{3}\left(-\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2}\right) = \frac{2}{3}$$

$$\Rightarrow G(S_1, 2) = G(S_1, 3)$$

$\Rightarrow$ it does not matter if we split by X_2 or X_3. and we are going to mistake on 1 sample anyway

if we split by X_2 we have

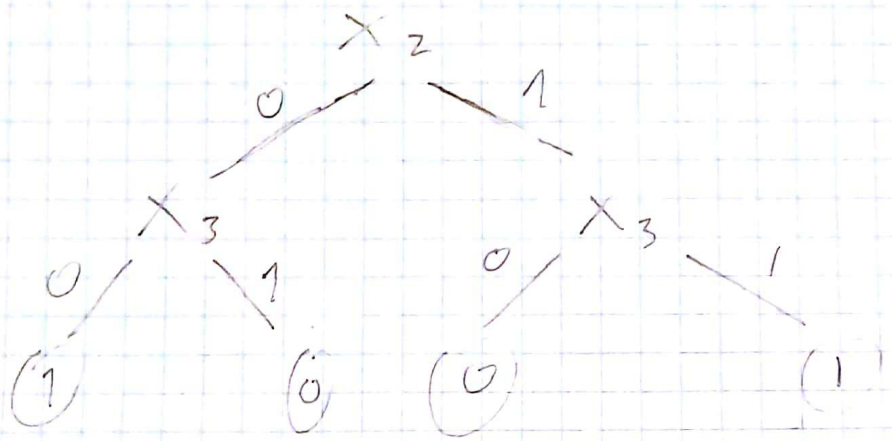
$S_0 = \{(1\ 0\ 0), 1)$ $\rightarrow$ no mistakes on this set

$S_1 = \{((1\ 1\ 1), 1), ((1\ 1\ 1), 0)\}$ $\rightarrow$ we have to choose a label by the majority now so we have 1 mistake guaranteed

similar for X_3 split $\Rightarrow$ 1 errors out of 4 samples

$\Rightarrow$ error $= \frac{1}{4}$

(5) we can see that a 0-error tree of 2 splits can be



3) (a) $\alpha_t = 0.5 \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) = \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)^{\frac{1}{2}}$

$\Rightarrow \varepsilon_t e^{\alpha_t} = \varepsilon_t \left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)^{\frac{1}{2}} = \sqrt{\varepsilon_t(1-\varepsilon_t)}$ //

$\varepsilon_t e^{2\alpha_t} = \varepsilon_t(1-\varepsilon_t)$

$\varepsilon_t e^{\alpha_t} = (1-\varepsilon_t) e^{-\alpha_t}$ //

$h_j = h(x_j)$

$\Rightarrow \sum_j D_t(x_j) e^{-\alpha_t y_j h_j} = \underbrace{\sum D_t(x_j) e^{\alpha_t}}_{\text{mistakes}} + \underbrace{\sum D_t(x_j) e^{-\alpha_t}}_{\text{correct labeling}}$

$= e^{\alpha_t} \underbrace{\sum D_t(x_j)}_{\varepsilon_t} + e^{-\alpha_t} \underbrace{\sum D_t(x_j)}_{1-\varepsilon_t}$

$e^{\alpha_t}\varepsilon_t + e^{-\alpha_t}(1-\varepsilon_t) = 2\sqrt{\varepsilon_t(1-\varepsilon_t)}$ //

(b) $P_{x \sim D_{t+1}}(h_{t+1}(x) \neq y_j) = \sum_j D_{t+1}(x_j) I(h(x_j) \neq y_j)$

$= \sum_j \frac{D_t e^{-\alpha_t y_j h_j}}{\sum_i D_t e^{-\alpha_t y_j h_j}} I(h_j \neq y_j) = \frac{\sum_j D_t e^{-\alpha y_j h_j} I(h_j \neq y_j)}{2\sqrt{\varepsilon_t(1-\varepsilon_t)}}$

definition of D_{t+1}

$\frac{1}{2\sqrt{\varepsilon_t(1-\varepsilon_t)}} \sum_{j: h_j \neq y_j} D_t(x_j) e^{\alpha_t} = \frac{e^{\alpha_t}}{2\sqrt{\varepsilon_t(1-\varepsilon_t)}} \underbrace{\sum_{j=1}^{N} D_t(x_j) I(h_j \neq y_j)}_{\varepsilon_t}$

$= \frac{\varepsilon_t e^{\alpha_t}}{2\sqrt{\varepsilon_t(1-\varepsilon_t)}} = \frac{1}{2}$ //

(c) Each h_t is a weak learner.

$\epsilon_{D,S}(h_t) \leq \frac{1}{2} - \gamma \quad \forall \gamma < \frac{1}{2}$

but we showed that

$\epsilon_{D_{t+1},S}(h_t) = \frac{1}{2}$

and

$\epsilon_{D_{t+1},S}(h_{t-1}) \leq \frac{1}{2} - \gamma < \frac{1}{2} = \epsilon_{D_{t+1},S}(h_t)$

$\Rightarrow h_{t+1} \neq h_t$

(d) $Z_t = \sum_i D_t(x_i) e^{-\alpha_t h_i y_i}$

$\frac{\partial Z_t}{\partial \alpha_t} = -\sum_i h_i y_i D_t(x_i) e^{-\alpha_t h_i y_i}$

$= \sum_{\text{mistakes}} D_t(x_i) e^{\alpha_t} - \sum_{\text{correct}} D_t(x_i) e^{-\alpha_t}$

$= e^{\alpha_t} \epsilon_t - e^{-\alpha_t}(1 - \epsilon_t) \stackrel{!}{=} 0$

$e^{2\alpha_t} = \frac{1-\epsilon_t}{\epsilon_t} \Rightarrow \alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$ //

by deriving again we can convince ourselves that it is a minimum.

(1) $S = \{(x_1, y_1) \dots (x_n, y_n)\}$

$y_i \cdot \sum_{j=1}^{k} a_j h_j(x_i) \geq \gamma \quad \forall i$

$\gamma > 0$, $a_j \geq 0 \ \forall j$, $\sum_{j=1}^{k} a_j = 1$

(2) $E\left(y_i \sum_{j=1}^{k} a_j h_j(x_i)\right) \geq E(\gamma) = \gamma$

$$E_{i \sim D}\left(y_i \sum_j a_j h_j(x_i)\right) = \sum_{i=1}^{n} D_i y_i \sum_{j=1}^{k} a_j h_j(x_i)$$

$$= \sum_{j=1}^{k} a_j \underbrace{\sum_{i=1}^{n} D_i y_i h_j(x_i)}_{\equiv g_j} \equiv \sum_{j=1}^{k} a_j g_j$$

$$= E_{\substack{j \sim \text{over} \\ \{1, \dots k\}}}(g_j) \geq \gamma$$

$\Rightarrow$ There exist $j' \in [0, k]$ for which $g_{j'} \geq \gamma$

$$g_{j'} = \sum_{i=1}^{n} D_i y_i h_{j'}(x_i) = \sum_{i : y_i \neq h_{j'}(x_i)} D_i y_i h_{j'}(x_i) + \sum_{i : y_i = h_{j'}(x_i)} D_i y_i h_{j'}(x_i)$$

$$= -\sum_{\text{mistakes}} D_i + \sum_{\text{correct}} D_i = -P_{i \sim D}(h_{j'}(x_i) \neq y_i) - (1 - P(h_{j'}(x_i) \neq y_i))$$

$$\Rightarrow -2P_{i\sim D}(h_j(x_i) \neq y_i) + 1 \geq \gamma$$

$$\Rightarrow P_{i\sim D}(h_j(x_i) \neq y_i) \leq \frac{1}{2} - \frac{\gamma}{2}$$

$j \in [0, k]$

(b) $S = \{(x_1, y_1), \ldots (x_n, y_n)\} \subseteq R^d \times \{-1, 1\}$

S realizable by a d-dim hyper rectangle classifier:

there exists a d-dim rectangle s.t

$y_i = 1$ iff $x_i \in \underbrace{[a_1, b_1] \times \ldots \times [a_n, b_n]}_{\text{the rectangle}}$

H is the class

$$h_{+\theta}(x) = \begin{cases} 1 & x(j) \leq \theta \\ -1 & x(j) > \theta \end{cases} \qquad h_{-\theta}(x) = \begin{cases} 1 & x(j) \geq \theta \\ -1 & x(j) < \theta \end{cases}$$

coordinate j of some vector x

we set $k = 2d - 1 \Rightarrow k$ hypotheses:

$\underbrace{h_1 = h_{1+a_1}, h_2 = h_1 - b_1}$ .... and so on for all d dimensions

those capture the 1st coordinat of $x \to x_i(1) \in [a_1, b_1]$

$\Rightarrow$ overall 2d hypotheses.

and $2d-1$ more of the form $h_{1+\infty}$

$\Rightarrow$

now we set α_j (α's in section a) $\forall j$:

$$\alpha_j = \frac{1}{4d-1} \qquad \gamma = \frac{1}{4d-1}$$

(uniform distribution over hypotheses)

$$y \sum_{j=1}^{k} \alpha_j h_j(x(j)) \geq \gamma$$

- inside rectangle:
  Assume that for all $m = 1 \dots n$ $y_m = 1$
  $\Rightarrow x_m \in [a_1, b_1] \times \dots \times [a_d, b_d]$

  $\Rightarrow$ $j = 1 \dots 2d$ $\quad h_j(x_m) = 1$
  $j = 2d+1 \dots 4d-1$ $\quad h_j(x_m) = -1$

  $\Rightarrow$ $\forall m$ $\quad y_m \sum_{j=1}^{4d-1} \alpha_j h_j(x_m) =$

$$= \sum_{j=1}^{2d} \frac{1}{4d-1} - \sum_{j=2d+1}^{4d-1} \frac{1}{4d-1} = \frac{2d - 2d + 1}{4d-1} = \frac{1}{4d-1} = \gamma$$

- outside rectangle:
  Assume $\forall m = 1 \dots n$ $y_m(i) = -1$
  $x_m \notin [a_1, b_1] \times \dots \times [a_d, b_d]$

$\Rightarrow$ There exists p s.t $x_m(p) \notin [a_p, b_p]$
($p \in [1, d]$)

$$\Rightarrow y_m \sum_{j=1}^{4d-1} \alpha_j h_j(x_m) = -\sum_{j \in [2d] / p} \frac{1}{4d-1} h_j(x_m) - \frac{1}{4d-1} \overbrace{h_p(x_m)}^{=-1}$$

$$- \sum_{j=2d+1}^{4d-1} \frac{1}{4d-1} h_j(x_m)$$

$$\Rightarrow \quad y_m \sum_{j=1}^{4d-1} \alpha_j h_j(x_m) = -\sum_{j \in [2d] \setminus p} \frac{1}{4d-1} h_j(x_m) + \frac{1}{4d-1} + \frac{2d-1}{4d-1}$$

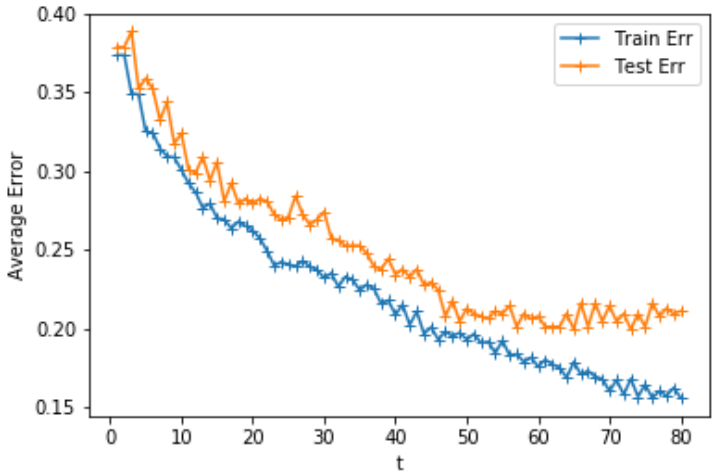
$$= -\sum_{j \in [2d] \setminus p} \frac{1}{4d-1} \underbrace{h_j(x_m)}_{\leq 1} + \frac{2d}{4d-1} \geq -\sum_{j \in [2d] \setminus p} \frac{1}{4d-1} + \frac{2d}{4d-1} = \frac{1}{4d-1} = \gamma$$

$$\Rightarrow \quad y_m \sum_{j=1}^{1\mathcal{C}} \alpha_j h_j(x_m) = \gamma \geq \gamma$$

$\forall m$

Programming assignment

a. Plot of error (of classifier defined as a linear combination of weak learner hypotheses) on train data and test data with respect to t:



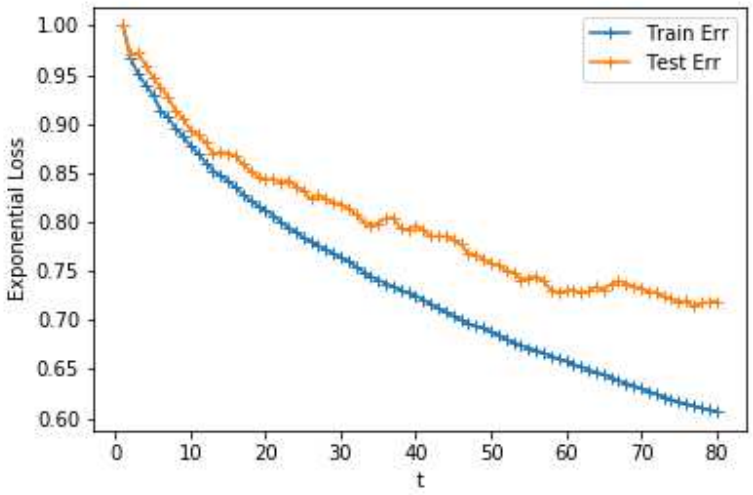
Both errors drop with the number of hypotheses that compos the classifier. It seems as if the test error has reached the optimum that can be achieved with the weak learners from our class.

b. List of the first 10 words and their classification rule:
h0: word=**bad**: classify as 1 if word occur <= 0.5
h1: word=**life**: classify as -1 if word occur <= 0.5
h2: word=**many**: classify as -1 if word occur <= 0.5
h3: word=**worst**: classify as 1 if word occur <= 0.5
h4: word=**performances**: classify as -1 if word occur <= 0.5
h5: word=**plot**: classify as 1 if word occur <= 0.5
h6: word=**great**: classify as -1 if word occur <= 0.5
h7: word=**nothing**: classify as 1 if word occur <= 0.5
h8: word=**fun**: classify as -1 if word occur <= 0.5
h9: word=**script**: classify as 1 if word occur <= 0.5

It seems that we have few words that are expected to be a good generalization for classifying reviews such as “bad”, “worst”, “great”. Intuitively, the first two would appear in bad reviews and the third in good reviews – note that this is indeed the rules we have for the hypotheses represented by the words.
We also have other words that are not expected to be related to a specific type of reviews, like “life”, “many”, “plot”. One explanation for the presence of such hypotheses is that after we get some hypothesis h’ by our ERM, the next step in adaboost will focus on the samples that h’ classified incorrectly. The next hypothesis will likely to regard a word that appear in those misclassified samples, just so it could classify correctly as many of them possible. Those words, sometimes are not good representatives of the whole data and are not intuitively good heuristics for reviews in general.
As an example, consider the review - “this movie is disappointing, unlike many other movies of this director that I thought was good”. The word “good”, if chosen as a classifier for good

reviews if they contain instances of it, will obviously mistake on that review. The next stage may choose the word “many” as a classifier for bad reviews just so this sample will then be classified correctly.

c. Average exponential loss of with respect to t



Loss of g on train data and test data drops with t (in other words – drops with the number of hypotheses achieved by adaboost). This Is not surprising – we saw in recitation 10 that running adaboost is equivalent to minimizing the average exponential loss (up to a constant).
Also, we note that as in section a it seemed as if after enough iterations we achieved optimal error on the test set, here as well, the loss reach saturation after a while.