COS511 HW2

Pranjit Kalita

March 15, 2017

Ex. 5
We will describe the architecture using the figure given in the next page.

Input Layer V_0 has two inputs X_0, X_1, with $X_0, X_1 \in R$. Furthermore, $X_0, X_1 \in [0, 1]$.

Hidden Layer V_1 has 15 neurons, with the bottommost neuron as shown in the figure being a constant. These neurons will be described as $v_{1,i}$, where i $\in [15]$.

Hidden Layer V_2 has 5 neurons, with the bottommost neuron as shown in the figure being a constant. These neurons will be described as $v_{2,i}$, where i $\in [5]$.

Output Layer V_3 has one neuron described by $v_{3,1}, whose$ output $\mathbf{o} \in$ {-1, 1}.

Given the graph G shown in Figure 1, we will assign equal weightage of 1 to all the edges $\in E$.

The major idea is that given the inputs X_0, X_1 in the layer V_0, they will be used to feed the given points to neurons in the V_1layer, with each neuron being a half space predictor of the form as described in **Section 9.1** of Book 4.Hence, for example if we were to predict 4 faces (a quadrilateral), we will take $v_{1,1}$through $v_{1,4}$.Similarly, for the other quadrilateral in the smiling face, we will take $v_{1,5}$through $v_{1,8}$.For the two different traingles, we take $v_{1,9}$through $v_{1,11}$,and $v_{1,12}$through $v_{1,14}$,respectively.In layer V_1,the halfspace predictor by nature uses the sgn function, hence we are in effect utilizing σ_{sgn}.

Now, in layer V_2,

$\mathbf{o}(v_{2,1}) = sgn(\mathbf{o}(v_{1,1}) + \mathbf{o}(v_{1,2}) + \mathbf{o}(v_{1,3}) + \mathbf{o}(v_{1,4}) - 3.5 * \mathbf{o}(v_{1,15}))$.
($\mathbf{o}(v_{1,15}) = 1$ (a constant), hence the above yields a 4-faced conjunction of half-spaces).

$\mathbf{o}(v_{2,2}) = sgn(\mathbf{o}(v_{1,5}) + \mathbf{o}(v_{1,6}) + \mathbf{o}(v_{1,7}) + \mathbf{o}(v_{1,8}) - 3.5 * \mathbf{o}(v_{1,15}))$.
($\mathbf{o}(v_{1,15}) = 1$ (a constant), hence the above yields a 4-faced conjunction of half-spaces.)

$\mathbf{o}(v_{2,3}) = sgn(\mathbf{o}(v_{1,9}) + \mathbf{o}(v_{1,10}) + \mathbf{o}(v_{1,11}) - 2.5 * \mathbf{o}(v_{1,15}))$.
($\mathbf{o}(v_{1,15}) = 1$ (a constant), hence the above yields a 3-faced conjunction of half-spaces.)

$\mathbf{o}(v_{2,4}) = sgn(\mathbf{o}(v_{1,12}) + \mathbf{o}(v_{1,13}) + \mathbf{o}(v_{1,14}) - 2.5 * \mathbf{o}(v_{1,15}))$.
($\mathbf{o}(v_{1,15}) = 1$ (a constant), hence the above yields a 3-faced conjunction of half-spaces.)

Thus, V_2 will be conjoining the half-spaces produced by neurons in the V_1 layer.

Now, $\mathbf{o}(v_{2,5}) = 1$ (a constant).

Finally, let us describe the output layer V_3 as essentially being the disjunction of the polytopes produced in V_2, that would yield a $+1$ when the point (x_1, x_2) lies within any of the polytopes, which is the point of the neural net here.

$\mathbf{o}(v_{3,1}) = sgn(\mathbf{o}(v_{2,1}) + \mathbf{o}(v_{2,2}) + \mathbf{o}(v_{2,3}) + \mathbf{o}(v_{2,4}) - 0.5 * \mathbf{o}(v_{2,5}))$.
(The above yields $+1$ when any of $\mathbf{o}(v_{2,1})$, $\mathbf{o}(v_{2,2})$, $\mathbf{o}(v_{2,3})$, $\mathbf{o}(v_{2,4}) = +1$, meaning the point falls within any of the polytopes. If not, then it would yield -1 based on the $\mathbf{o}(v_{2,5}) = 1$.)

Thus, given V, E, w, and the $\mathbf{o}$ for each v in the neural net, this is my proposed architecture.

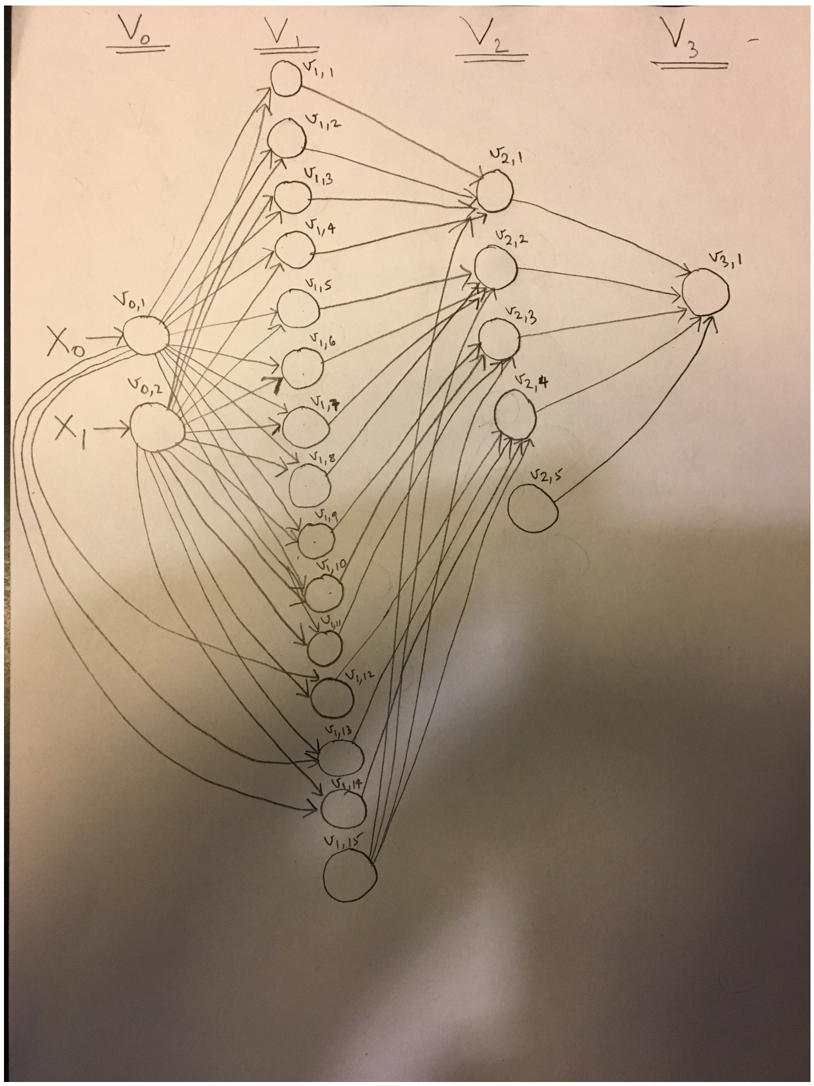

Figure 1: Neural Net for Smiling Face

Ex. 2

A/Q, VC-dim(H_i) =d_i, d =$\sum_{i=1}^{m} d_i$.

We have to prove that VC-dim($\cup_{i=1}^{m} H_i$) = O(d*log(d)).

Let us take a set of k examples and assume that they are shattered by the union class. In that scenario, the union class can produce all 2^k possible labeling on these examples.

In order to solve this problem, we will make use of the fact that under union operation,

$VC - dim(\cup_{i=1}^{m} H_i) <= \sum_{i=1}^{m} VC - dim(H_i)$

i.e, the number of ways in which k points can be possible shattered is at most the individual sums of the $H_i's$, given that some of them will overlap.

Thus, we have -

2^k <=$\sum_{i=1}^{m} k^{d_i}$ (using Sauer's Lemma $\tau_H(k)$ =2^k, for all k <=d, where d = VC=dim(H)). (I)

Now, since d =$\sum_{i=1}^{m} d_i$,

$\sum_{i=1}^{m} k^{d_i}$ <= m * 2^d (II)

Using II in I yields -

2^k <=$m * 2^d$

Taking log of both sides -

k <= log(m) + d*log(k)

But since in application of Sauer's Lemma, we have already assumed for k <= d_i,

=> k <= d

Thus,

k <= log(m) + d*log(d) (III)

=> k <= constant + d*log(d), since m is a constant value.

Thus, we see that $VC - dim(\cup_{i=1}^{m} H_i)$ = O(d*log(d)), seeing the upper bounds in the above equation (III).

Ex. 4 - Consulted with Divyarthi Mohan

Let us assume that we will try k independent samples and the worst case scenario that they each will yield δ confidence, i.e-

$err(h) > \min_{h* \in H} err(h*) + \epsilon$, with probability δ

Now, acc. to the question, $\delta = 1/3$.

For k independent samples each with worst case performance,

$(1/3)^k < \delta$

Taking log of both sides and rearranging terms -

$k > log(1/\delta)$

$\Rightarrow k = \Omega(log(1/\delta))$

Now, $\Pr(\forall_{i=1}^{k} err(h_i) > \min_{h* \in H} err(h*) + \epsilon) < (1/3)^k < \delta$

$\Rightarrow Pr(\exists_{i=1}^{k} err(h_i) < \min_{h* \in H} err(h*) + \epsilon) >= 1 - (1/3)^k < 1 - \delta$

$\Rightarrow$There exists at least one sample distribution that would yield a hypothesis within δ confidence.

If we used ERM in this boosted sample set, according to Corollary 2.5 of lecture notes (*Finite Classes are learnable*), for k being the size of the hypothesis class, sample complexity $m = \mathcal{O}((1/\epsilon^2) * log(k)/\delta)$

$\Rightarrow$m = $\mathrm{O}((1/\epsilon^2) * log(1/\delta)/\delta)$

Since each of the k samples were chosen with sample complexity $m(\epsilon)$, the overall sample complexity to get to the learnability result above is -

$|S| = \Omega(m(\epsilon) * k + ((1/\epsilon^2) * log(1/\delta)/\delta))$

Thus, we see on boosting, eventually with the given number of subsequent independent sample distributions, the given hypothesis class is learnable.

Ex. 1
Part a
We will prove it for two functions $F = F_1 \circ F_2$, and extend the result to multiple functions under composition.
Let $\mathrm{F}_1 \subseteq \{f_1 : X- > Y\}, F_2 \subseteq \{f_2 : Y- > Z\}$,F $= \mathrm{F}_1 \circ F_2$.

Let us take $C \subseteq X$ with size $|C| = m$.
Restricting $\mathrm{F}_1 toC, i.e.,$G $= \mathrm{F}_1|_C$, we have
$F|_C = \cup_{g\in G}\{f_2 \circ g|f_2 \in F_2\}$
Thus,
$|F|_C| \leq |F_1|_C| * max_{g\in G}|\{f_2 \circ g|f_2 \in F_2\}$
$\Rightarrow |F|_C| \leq \tau_1(m) * max_{g\in G}|F_2|_{g(C)}|$
$\Rightarrow |F|_C| \leq \tau_1(m) * \tau_2(m)$

But since, ${}_{max}|F|_C| = \tau(m)$ for our given composition,
$\Rightarrow \tau(m) \leq \tau_1(m) * \tau_2(m)$

Extending it to t functions, we will get the required product as noted in the question.
Hence proved.
Part b
We will prove it for two functions $F = F_1 \times F_2$, and extend the result to multiple functions under composition.
Let F1 $\subseteq \{f_1 : X_1- > Y_1\}, F2 \subseteq \{f_2 : X_2- > Y_2\}$,F $= \mathrm{F}_1 \times F_2$.
Let us take $C_1 \subseteq X_1$ with size $|C_1| = m$.
Let us take $C_2 \subseteq X_2$ with size $|C_2| = m$.
But, F is restricted by both C_1andC_2.
Therefore, it is trivial to see that$-$
$|F|_C| \leq |F_1|_{C1}| * |F_2|_{C2}|$
$\Rightarrow |F|_C| \leq \tau_1(m) * \tau_2(m)$

But since, ${}_{max}|F|_C| = \tau(m)$, $\tau(m) \leq \tau_1(m) * \tau_2(m)$

Extending it to t functions, we will get the required product as noted in the question.
Hence proved.