Advanced Machine Learning Homework 4

March 2020

VC Dimension of Neural Networks

We take 0/1 classification problem for data with d dimensional features as an example.

A neural network with one hidden layer can be written as

$$o = \mathbf{w}_2^T \sigma(\mathbf{W}_1 \mathbf{v} + \mathbf{b}_1) + b_2, \tag{1}$$

where $\mathbf{v}$ is the d dimensional input feature of the data, while $\mathbf{W}_1, \mathbf{w}_2, \mathbf{b_1}, b_2$ are the parameters of the neural network model. $\mathbf{W}_1$ is a $n \times d$ matrix, $\mathbf{w}_2$ is a n dimensional vector, and $\mathbf{b}_1$ is a n dimensional bias vector while b_2 is the bias.

When $o > 0$ we classify the datum as label 1, while when $o \leq 0$ we classify it as label -1. This forms a neural network, or multi-layer-perceptron, with one hidden layer containing n neurons.

In this problem, we focus on the pre-training case with frozen parameters that $\mathbf{W}_1$ and $\mathbf{b}_1$ must be decided with all $\mathbf{v}$ without labels $(l_1, ...l_i)$, while $\mathbf{w}_2$ and $\mathbf{b}_2$ can be decided with labels of examples $(l_1, ...l_i)$.

Problems

1. Given n, d, calculate the VC dimension of the neural network for the linear activation case, i.e. $\sigma(x) = x$. Prove your result.
2. Given n, d, calculate the VC dimension of the neural network for the ReLU activation case, i.e. $\sigma(x) = \max(0, x)$. Prove your result.

Hint

1: Recall the definition of VC dimension.
2: Consider $n > d$ and $n \leq d$.
3: For problem 2, Start from $d = 1$.

Answer

Lemma 1: VC Dimension for Linear Classifiers. *For linear classifiers, which $o = \mathbf{w}^T \mathbf{v} + b$, where $\mathbf{w}, \mathbf{v}$ has dimension d, the VC-dimension for such linear classifier is $d + 1$.*

Proof. There exists a set of $d+1$ points, $\{(1,0,0...0),(0,1,0,...0),...,(0,...,0,1),(0,0,...,0)\}$. For any labeling of these points $\mathbf{l}=(l_1,...,l_{d+1})$, we take $w_i = l_i - l_{n+1}, 1 \le i \le d$ and $b = l_{n+1}$, we can shatter these $d+1$ points, so the VC-dimension is at least $d+1$.

For any $d+2$ points, we add an 1 on top of its representation, $\mathbf{v'} = (1, v_1, ...v_d)$ and $\mathbf{w'} = (b, w_1, ...w_d)$, so $o = \mathbf{w'}\mathbf{v'}$ after this add. Hence we get $d+2$ points with $(d+1)-$dimensional representation, which must be linear dependent, i.e. $\mathbf{v_i'} = \sum_{j \ne i} a_j \mathbf{v_j'}$, we label l_i to be -1 and $l_j = sgn(a_j)$, therefore $o_i = \sum a_j o_j$. If all js are labelled correctly, $o_i > 0$, and contradicts to $l_i = -1$, so the model cannot shatter any $d+2$ points.

□

Answer to Problem 1:

For the linear activation case,

$$o = \mathbf{w}_2^T(\mathbf{W}_1\mathbf{v} + \mathbf{b}_1) + b_2 = \mathbf{w}_2^T\mathbf{W}_1\mathbf{v} + (\mathbf{w}_2^T\mathbf{b}_1 + b_2) \tag{2}$$

This means the power of a linear activation neural network shall not exceed a linear classifier. $VC \le d+1$

Also, given that $\mathbf{v}$ is first transformed into a n dimensional vector before layer 2, it's VC dimension shall not exceed $n+1$. $VC \le n+1$.

In order to scatter $min(n+1, d+1)$ points, this can be easily achieved as we simply set $\mathbf{W}_1 = (\mathbf{I}, 0)$ or $(\mathbf{I}, 0)^T$, depending on whether $n > d$ or $n \le d$, and also set $\mathbf{b}_1 = 0$. Then we get back to a linear classifier with dimension $min(n, d)$.

So $VC = min(n+1, d+1)$.

Answer to Problem 2:

We start from $d = 1$. We set $n+1$ points to be $(1),...,(n),(0)$, and $\mathbf{W}_1 = ((1,1,...,1))$, $\mathbf{b}_1 = (-1,-2,...,-n+1,0)$. This yields $\mathbf{W}_1\mathbf{v} + \mathbf{b}_1$ to be

$(1,0,...,0),(1,1,...,0),...,(1,1,...,1),(0,0,...,0)$, we simply take $\mathbf{w}_2 = (l_1 - l_{n+1}, l_2 - l_1 - l_{n+1}, l_3 - (l_1 + l_2) - l_{n+1}, ..., l_n - \sum_{i<n} l_i) - l_{n+1}$ and $b_1 = l_{n+1}$, this means we can scatter $n+1$ points, $VC \ge n+1$.

Also, we have $VC \le n+1$ from lemma 1, so $VC = n+1$.