CS 839: Theoretical Foundations of Deep Learning **Spring 2023**

Lecture 19 Complexity I: Training a 3-Node is NP-Hard

Instructor: Yingyu Liang *Date:* *Scriber: Nikos Zarifis*

1 Intro to NP-Completeness

Assume that I have problems A, B. We say that that B is as hard as A, if there exists a polynomial reduction from A to B. Meaning that if I can solve B then I can solve A. We write $A \leq_P B$. One known NP-complete problem is called "Set-Splitting" and is the following:

Set-Spliting(SS):

Given: S, a collection of subsets $C = \{C_i | C_i \subset S\}$.

Question: Does there exists S_1, S_2 with $S_1 \cap S_2 = \emptyset$, such that $S_1 \cup S_2 = S$, and for all i, it holds $C_i \not\subset S_1$ and $C_i \not\subset S_2$.

Note that the problem is still hard assuming $|C| = O(|S|)$.

In the section below, we are going to reduce the "Set-Spliting" to the training a 3-node NN. This was proved in [1].

2 Training a 3-Node Network

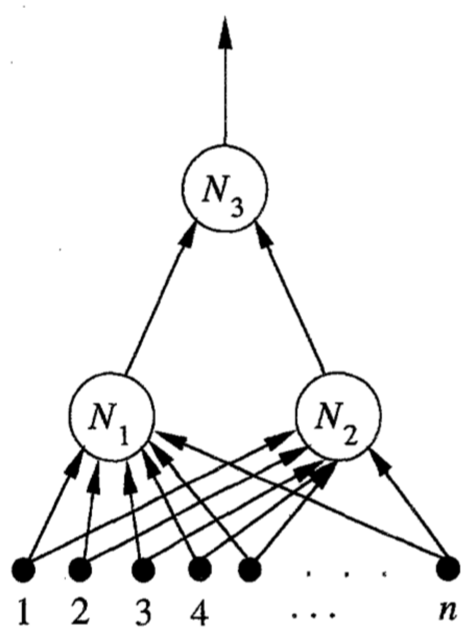

Figure 1: The 3-Node network

Let $a = [a_1, \ldots, a_n] \in \mathbb{R}^n$ and $a_0 \in \mathbb{R}$. We define the following threshold function:

$$f_i(z) = \begin{cases} 1 & \text{if } a \cdot z > a_0 \\ -1 & \text{o.w.} \end{cases}$$

This is equivalent to $f_i(z) = \text{sign}(a \cdot z - a_0)$. The main question is the following:

Question: Given a set of $O(n)$ examples $(x, y) \in \{0,1\}^n \times \{\pm 1\}$. Do there exists, f_1, f_2, f_3 such that the 3-node network has training error 0?

In fact, we are going to show that this problem is hard and in fact it is NP-Complete, by showing a reduction from Set-Spliting problem. Hence, we show the following:

Theorem 1. Training 3-node NN is NP-Complete.

Proof. First, we provide a geometric intuition for this problem: Each point is a point of the n-dimensional hypercube. The two functions f_1, f_2 are linear thresholds functions, therefore, each one define a hyperplane. Therefore, if they are not parallel, they divide the space into four quadrants. Because the f_3 is a linear threshold, it can distinguish between points on different quadrants. So, the problem of training a 3-node, is equivalent to the following problem:

Given a set of labeled points in the n-dimensional hypercube does there exists:

Case 1: A simple plane separates ± 1.

Case 2: Two planes such that either one quadrant contains all positive labels $(+1)$ and no negative points, or one quadrant contains all negative labels (-1) and no positive points.

We are going to show that case 2 is the hard one, which means that this problem is NP-complete.

Problem 2LCPBE: Given n-labeled points. Do there exist planes f_1, f_2 such that the quadrant with both positive predictions contains all positive points and no negative labeled points?

We are going to reduce the problem of Set-Splitting to 2LCPBE. Given an instance of Set-Splitting: $S = \{s_1, s_2, \ldots, s_n\}$, $C = \{C_1, C_2, \ldots\}$ and $\{C_j \subseteq S\}$, we are going to convert it to the following instance of 2LCPBE:

- Let the origin: $(0, 0, \ldots, 0)$ have the label $+$.
- for each s_i, we make a point $p_i = (0, \ldots, 1, \ldots 0)$, where the 1 is in the i-th position, and we label it $-$.
- for all $C_j = \{s_{j1}, \ldots, s_{jk}\}$, we put at the point that has 1 at the positions $j1, j2, \ldots, jk$ and the $+$ label, that point is $p_{j1} + p_{j2}, \ldots, p_{j,k}$.

For example consider the instance: $S = \{s_1, s_2, s_3\}$, $C_1 = \{s_1, s_2\}$ $C_2 = \{s_2, s_3\}$. We have $[(0,0,0), 1]$ and $[(1,0,0), -1]$,$[(0,1,0), -1]$,$[(0,0,1), -1]$ and $[(1,1,0), 1]$,$[(0,1,1), 1]$.

Lemma 2. The instance of SS has a solution is equalivalent to constructed instance of $2LCPBE$ has.

Proof. For the first direction. Given S_1, S_2 from the solution of set splitting, we consider the following:

Consider the hyperplanes: P_1, P_2 with the following form: $P_j : a_1x_1+\ldots+a_nx_n+1/2 = 0$, where

$$a_i = \begin{cases} -1 & \text{if} \quad s_i \in S_j \\ n & \text{ow} \end{cases}$$

You can see that the following hold:

- P_j predicts + for $(0, 0, \ldots, 0)$
- P_j predicts + for training point with +
- P_j predicts − for p_i if $s_i \in S_i$.

Therefore, the intersection(the quadrant) of hyperplanes: $P_1 \geq 0, P_2 \geq 0$, contains all the points with + and no point with −.
Let S_1 (resp. S_2) be the set that contains that only in P_1 (resp. P_2) get −. Place the rest of the points that both planes separates with − arbitrary in S_1 or S_2. $S_1 \cup S_2 = S$ as all the points are either in S_1 or S_2.

Let $C_j = \{s_{j1}, \ldots, s_{jk}\}$, it remains to show that $C_j \not\subset S_1, S_2$. P_1 predicts positive for $p_{j1}+\ldots+p_{jk}$ if $c_j \subset S_1$ but then this points would not be in one quadrant with only positive points which contradicts the assumption of 2LCPBE. Similarly for P_2 and S_2. □

Now we have shown that the training 3-node is NP-complete if one quadrant contains all the positive points, so the f_3 should be the AND function between f_1 and f_2. Now, we will add some more points to make the output to always require that conditions. We extend the dimension of our points to $n + 3$ and put 0 in the new components of the previous points. This is left as homework; you can also refer to the reference.

□

References

[1] Avrim L Blum and Ronald L Rivest. Training a 3-node neural network is np-complete. *Neural Networks*, 5(1):117–127, 1992.