Cover's Function Counting Theorem

Matheus Morroni

January 2020

1 Introduction

With the intention of theoretical justification of several claims made about neural networks regarding their capabilities to learn a set of patterns and their ability to generalise, some concepts of maximal storage capacity were developed. In particular Cover's capacity and VC-dimension are two expressions of this notion and the first one is of special interest here.

2 Mathematics' Concepts

Theorem Let x_1, ..., x^P be vectors in R^N, that are in general position. Then the number of distinct dichotomies applied to these points that can be realized by a plane through the origin is:

$$C(P,N) = 2\sum_{k=0}^{N-1}\binom{P-1}{k}$$

Before the proof of the theorem in the next section, let's discuss some of its consequences.

2.1 First Consequence

For P $\leq$ N the next sum is limited by P-1, so it can be manipulated to

$$C(P,N) = 2\sum_{k=0}^{P-1}\binom{P-1}{k} = 2(1+1)^{P-1} = 2^P$$

(tip: for check this, use the familiar binomial expansion). In other way, all possible dichotomies can be realized.

2.2 Second Consequence

For P = 2N,

$$C(2N,N) = 2\sum_{k=0}^{N-1}\binom{2N-1}{k} = 2\frac{1}{2}2^{2N-1} = 2^{P-1}$$

This occur because the expression for C contains exactly one half of the full binomial expansion of $(1+1)^{2N-1}$, and it is known that this expansion is symmetrical. The conclusion is that in this case exactly half of all possible dichotomies are able to be realized.

2.3 Third Consequence

For P >> N, it is not difficult to observe that $C(P,N) \sim AP^N$ for some A > 0. When P is bigger than N the set of dichotomies that can be realized still grows with P, but the number of dichotomies that can be implemented in proportion to the number of possible dichotomies shrinks, since the total number of possible dichotomies is 2^P.

2.4 Fourth Consequence

Another way to check the behavior of C(P,N) is to allow both P and N to approach infinite, but to keep them proportional, i.e. $\frac{P}{N} = \alpha$. In this case, as $N \to \infty$, C(P,N) versus $\frac{P}{N}$ becomes a step function. Observe that the step function shape must still obey the properties that were founded before, in mainly $\frac{C(2N,N)}{2N} = 0.5$. this is in fact the value around which the step occurs. i.e. below the critical value of $\frac{P}{N} = 2$, it is possible to see that virtually all of the dichotomies are possible, and above that value virtually none are possible.

3 Proof of the Theorem

To proof the theorem we can start with P points in general position. We now assume that there are C(P,N) dichotomies possible on the group, and ask how many dichotomies are possible if another point is added, i.e. what is the value of C(P+1,N). In this way we will able to prepare a recursive expression for C(P,N).

Let $(b_1, ..., b_P)$ be a dichotomy realizable by a hyperplane over the group of P inputs, in other way, $b_i \in -1, +1$ for every i = 1, ..., P. And there is a group of weights w so that for each of them $(sign(w^T x^1), ..., sign(w^T x^P)$ we can relate with a dichotomy $(b_1, ..., b_P)$. Using one such w, we get a dichotomy over the group of P+1 inputs:

$$(sign(w^T x^1), ..., sign(w^T x^{P+1})) = (b_1, ..., b_P, sign(w^T x^{P+1}))$$

Thus for every linearly implemented dichotomy over P points there is at least one linearly realized dichotomy over P+1 points. Observe that different dichotomies over P points define different dichotomies over P+1 points, since they differ somewhere in the first P coordinates. Now, probably, the additional dichotomy $(b_1, ..., b_P, sign(w^T x^{P+1}))$ (put the result of the last coordinate) is also possible, by some other group of weights? In this case C(P+1, N) can be higher than C(P, N). Let us write

$$C(P+1, N) = C(P, N) + D$$

Our objective is to find D, the additional dichotomies number.

Let us assume that one of the weight vectors W that generates $(b_1, ..., b_P)$ passes directly through x^{P+1}. In this way, it is clear that by smooth changes in the angle of the hyperplane we will be capable to move the hyperplane smoothly to this side or the other of x^{P+1}, thus getting a value of either -1 or +1 on it. Thus in this way, both possible dichotomies are possible, and there is one additional possible dichotomy beyond C(P, N) (that is counted in D).

On the other way, if no hyperplane that passes through x^{P+1} and generates $(b_1, ..., b_P)$ on the first P vectors exists, then it means that the point lies in one side of all dichotomies over P points that are implemented by a hyperplane that passes through a certain determined point x^{P+1} (which is in general position with the other points).

Now, by forcing the hyperplane to pass through a certain fixed point, we are in fact moving the problem to one in N-1 dimensions. In conclusion, D = C(P, N-1), and the recursion formula is

$$C(P+1, N) = C(P, N) + C(P, N-1)$$

For the proof of the theorem we now use induction. Let us assume that

$$C(P, N) = 2\sum_{k=0}^{N-1}\binom{P-1}{k}$$

holds up to P and N. Observe that it easily holds for P = 1 and all N, since it gives C(1, N) = 2 as expected, since one point in N dimensions can be dichotomized with the two labels by a hyperplane. Then,

$$C(P+1, N) = 2\sum_{k=0}^{N-1}\binom{P-1}{k} + 2\sum_{k=0}^{N-2}\binom{P-1}{k}$$

$$= 2\sum_{k=0}^{N-1}\binom{P-1}{k} + 2\sum_{k=0}^{N-1}\binom{P-1}{k-1} = 2\sum_{k=0}^{N-1}\binom{P}{k}$$

where we have used $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$ and used the convention that $\binom{n}{k} = 0$ for $k < 0$.

4 Application of the Theorem

During the discuss of learning in the context of a perceptron, it is interesting to try to quantify its complexity. The issue raises the question "how do quantify the complexity of a given architecture?", or its capacity to realized a group of input-output functions, in this case, dichotomies.

A straightforward simple answer would be to count the number of different functions that this architecture is capable to implement. However, this measure is meaningless if the inputs come from some continuous space, as in this particular case, every arbitrarily small changes in the decision boundary will relate to a new dichotomy.

One way to transcend this difficulty is to limit the input to a fixed finite set of P input vectors, x, ..., x^P. Now it is possible to count the number of dichotomies that can be realized by a given network architecture on this fixed set of inputs.

Applying this idea in the case of the perceptron, there is another potential complication which is that in general, the number of dichotomies that can be implemented by a given architecture can depend also on the particular set of inputs. Luckily enough this is not he case in the perceptron.

In fact, the number of linearly realizable dichotomies on a set of points depend only on a general position. The calcultation of the number of linearly realized dichotomies is given by the Cover's Function Counting Theorem.

5 References

- Introduction: The Perceptron, Haim Sompolinsky, MIT, 2013
- Counting function theorem for multi-layer networks, Adam Kowalczyk,

Telecom Australia