The Perceptron

A Brief and Semi-Technical Introduction for Non Computer Scientists

James Skripchuk

1 Motiviation

One of the most common problems faced in Computer Science is *classification*. For example, if we give a program all of the attributes of a furry four-legged creature we found, we want the program to tell us what kind of creature this is.

Generalizing, given an object that can be represented as a finite list of numerical features $x = [x_1, x_2, \dots x_n]$, we would like to categorize it as an element of a finite predetermined set of classes $\{A, B, C \dots\}$. A *binary classifier* is a type of classifier where there are only two classes to choose from $\{A, B\}$.

2 The Perceptron

The perceptron is one of the first developed and simplest examples of a binary classifer. In order to classify an object $x = [x_1, x_2, \dots x_n]$ as one of the two classes $\{+, -\}$, the perceptron algorithm is "tuned" by two parameters that are set before classification starts:

- a list of numerical weights $w = [w_1, w_2, \dots w_n]$
- a threshold value t

The weight w_i determines how important feature x_i is in making the decision. Large positive numbers means the associated feature is important, while numbers close to zero means the feature doesn't affect the overall decision much. A negative weight means that the associated feature x_i is undesirable. The threshold value t is used to determine how confident our classifier needs to be in order to say that x is indeed a member of one class. A high threshold value t means we need more evidence in order to classify the object as positive.

First, we multiply pairwise each elements of x and w and add them together (also known as the *dot product* between x and w).

$$z = x_1w_1 + x_2w_2 + x_3w_3 + \dots x_nw_n \tag{1}$$

Then to classify the object, we simply compare z to the threshold value t

$$f(z) = \begin{cases} + & \text{if } z \geq t \\ - & \text{if } z < t \end{cases} \tag{2}$$

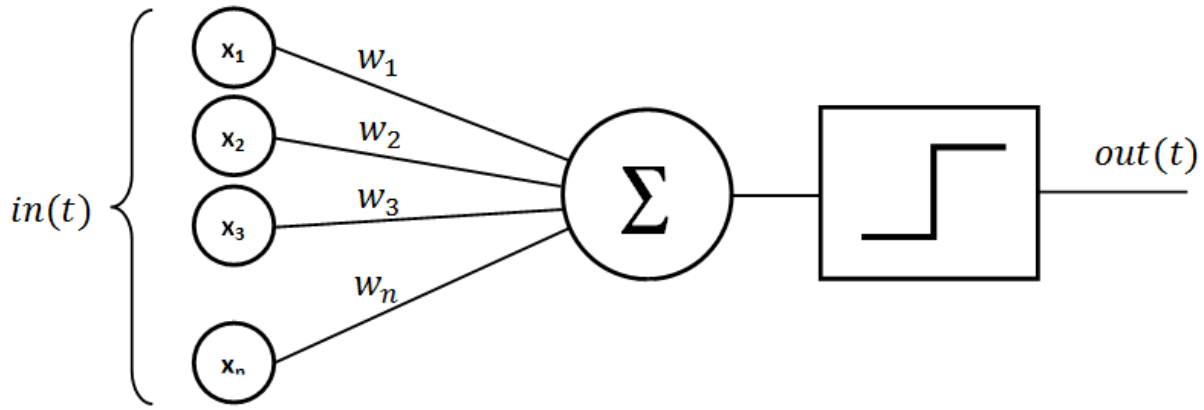

Figure 1: A Simple Perceptron [3]

3 Example

Suppose we have a group consisting of dogs and cats and we want to classify if something is a dog based off of a couple features: whether or not it has fur (represented by -1 or 1), its height, and its time spent napping. If our object's dot product exceeds the threshold, we classify it as a dog. We can pick our weights using the following reasoning.

$$w = [1, 4, -2] \tag{3}$$

A lot of animals have fur, so if something we're looking at has fur, it doesn't tell us much. Therefore, we can assign $w_{fur?}$ a relatively small number. Dogs tend to be much taller than cats, and thus height seems to be an important attribute, so we give w_{height} a larger number. Dogs tend to spend much less time napping than cats, so if something naps a lot, then it's probably *not* a dog. Because of this, we can assign $w_{napping}$ to be a negative number (so the more something naps, the more it brings the total sum down). Finally, we need to set a threshold value; this can be obtained by checking the results of the dot product on data that you already know the answer to. Let's make $t = 100$.

Now, let's try to classify the following animals:

- Fido: $f = [1, 50, 2]$
- Mittens: $m = [1, 10, 10]$

Performing the dot product, we see that $z_{Fido} = 117$ and $z_{Mittens} = 21$. Fido passes the $t = 100$ threshold, and thus we classify it is as a dog. On the other hand, we can't say Mittens is a dog since it doesn't pass the threshold.

4 Conclusion

Of course, perceptrons aren't perfect. I am sure you can come up with examples that can fool the classifier based on the assumptions that were made in the beginning (maybe you have a very lazy dog, or a very large cat). This is only an introduction of the most primitive form of the perceptron; other resources (such as those sourced below) describe more modern research.

References

[1] Akshay Chandra Lagandula *https://towardsdatascience.com/perceptron-learning-algorithm-d5db0deab975*.

[2] Manuela Veloso *Perceptrons and Neural Networks* Carnegie Mellon Fall 2001.

[3] Perceptron by Mayranna is licensed under CC3.0
https://commons.wikimedia.org/wiki/File:Perceptron_moj.png

[4] Perceptrons - the most basic form of a neural network
https://appliedgo.net/perceptron/