Perceptron Learning Algorithm

Rishabh Pomaje

November 29, 2023

Mathematical Background

The perceptron is used whenever the data we have is linearly separable i.e., when the data forms clusters that can be separated by a hyperplane. Our goal is to find this hyperplane.

Let the data points in such a data be represented as, $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N$ be inputs with the labels $y_1, y_2, \cdots, y_N$ respectively where $\mathbf{x}_i \in \mathcal{X}$ *is* $\mathbf{d} - dimensional$ and $\mathcal{Y} = \{-1, 1\}$.

These labels are as given by a target function $f : \mathcal{X} \to \mathcal{Y}$ such that $f(\mathbf{x}) = y$
Let $h : \mathcal{X} \to \mathcal{Y}$ be the hypothesis that approximates the target f. Let $\mathbf{w}$ be the weight vector such that if Let $\mathbf{x} = (x_1, x_2, \cdots, x_d)$ and $\mathbf{w}$=$(w_1, w_2, \cdots, w_d)$,

$$
\begin{aligned}
h(\mathbf{x}) &= sgn(\sum_{i=1}^{d} w_i x_i - threshold) \\
&\quad By\ setting\ w_0 = -threshold \\
&= sgn(\sum_{i=1}^{d} w_i x_i + w_0(1)) \\
&\quad By\ setting\ x_0 = 1 \\
&= sgn(\sum_{i=0}^{d} w_i x_i) \\
h(\mathbf{x}) &= sgn(\mathbf{w}^T \mathbf{x})
\end{aligned}
$$

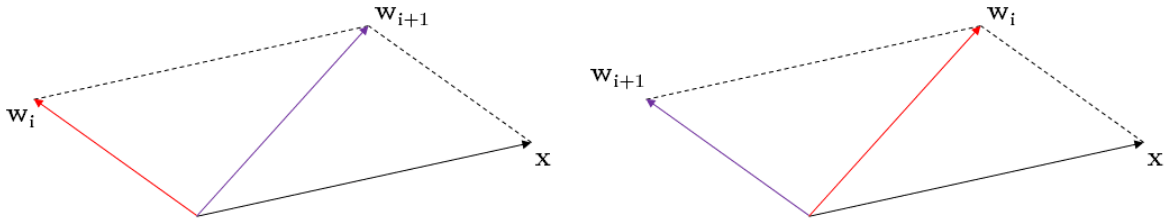
Selection and updating the weights

To start with, we can set the $\mathbf{w}_0$ vector randomly. Let $\mathbf{w}_i$ be the weight in i^{th} iteration.
***Definition*:** A point $\mathbf{x}$ is said to be misclassified if $y = f(\mathbf{x}) \neq h(\mathbf{x})$.

***Perceptron Learning Algorithm*:**(in i^{th} iteration)

1. Choose a misclassified point, say $\mathtt{x}_r$.
2. Update the weight vector as $\mathtt{w}_{i+1} \leftarrow \mathtt{w}_i + y_r \mathtt{x}_r$.
3. Continue till there are no misclassified points.

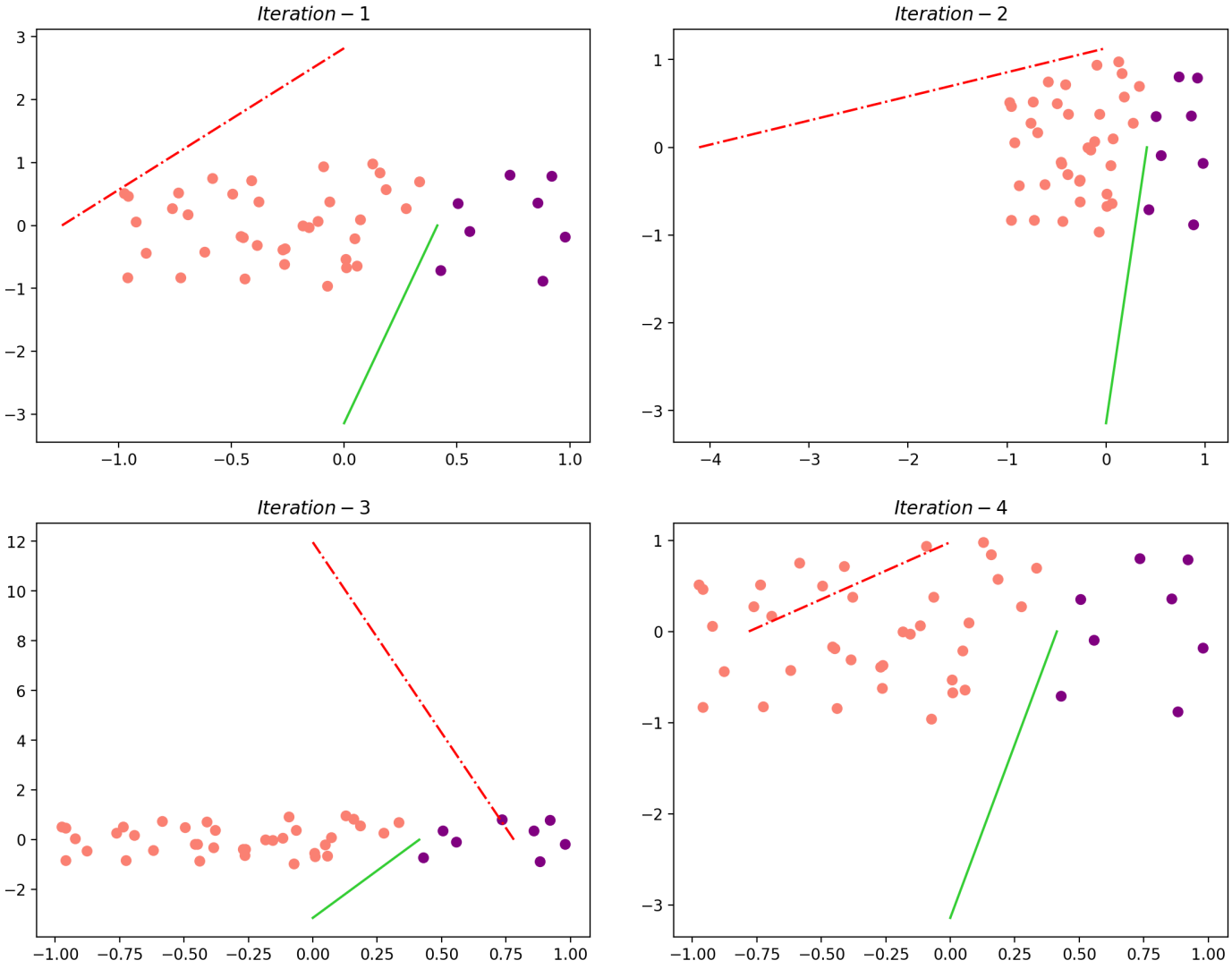
Why does this work?

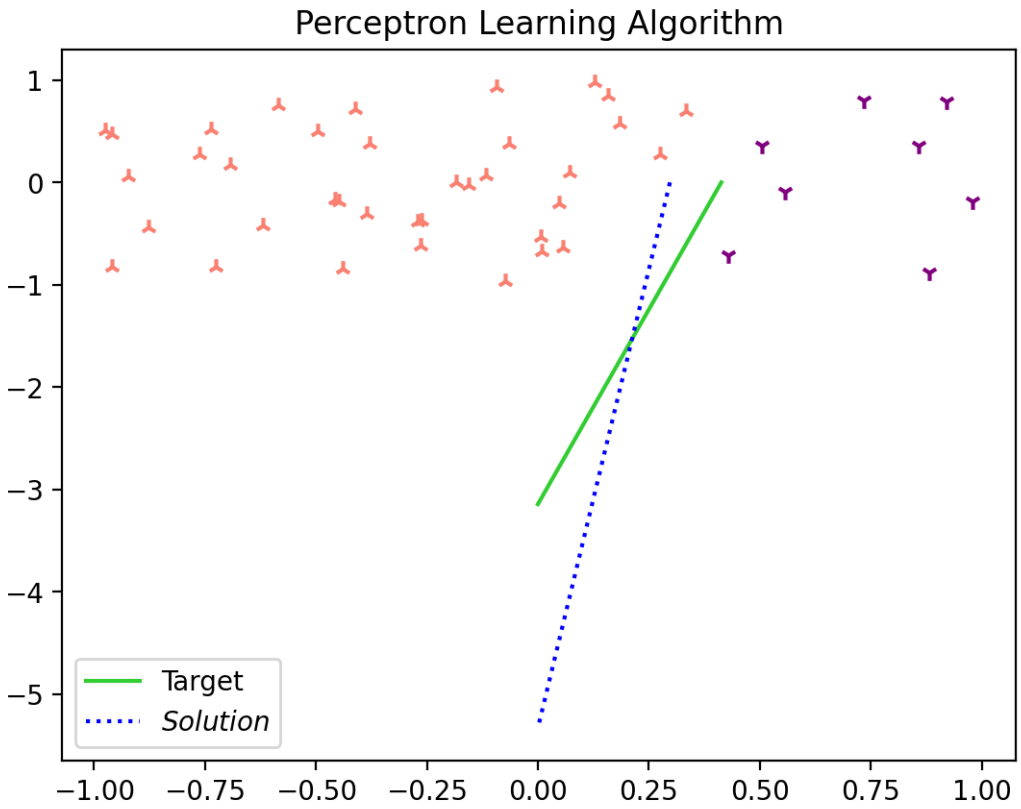


$y = +1$ $y = -1$

Updating the points at misclassified points

Visualization of the results in 2-d





Final Solution as obtained by the PLA

Some observations:

The algorithm converges to the target much better to the target function as the number of data points increases as the available 'whitespace' decreases with increasing points.