Perceptron

Jack David Carson - February 3, 2025

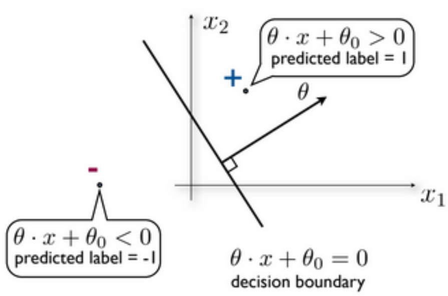
If we have some linear classifier that we wish to use to predict the value y corresponding to a feature vector $\boldsymbol{x}^{(i)} \in \mathbb{R}^d$ for $y \in \{-1, 1\}$, we can first formalize the problem as trying to select an ideal classifier $\hat{h} \in \mathcal{H}$, the set of all classifiers. In functional form we can say $h : \mathbb{R}^d \to \{1, -1\}$ and that classifier $\hat{h}$ has some parameters $\boldsymbol{\theta}$ and evaluate its training error over training data $S_n = \left\{\boldsymbol{x}^{(i)}, y^{(i)}, i = 1, ..., n\right\}$

$$\mathrm{E}_n(h) = \frac{1}{n} \sum_{i=1}^{n} \left[\left[h\left(\boldsymbol{x}^{(i)}\right) \neq y^{(i)}\right)\right]\right]$$

where $[[\text{true}]] = 1$ is an error and $[[\text{false}]] = 0$ is a correct classification. We define the actual classification $h\left(\boldsymbol{x}^{(i)}\right)$ in terms of its parameters and a "bias" term $\boldsymbol{\theta}_0$ as

$$h(\boldsymbol{x}, \boldsymbol{\theta}) = \operatorname{sgn}\left(\boldsymbol{\theta}^\top \boldsymbol{x} + \boldsymbol{\theta}_0\right) = \begin{cases} +1 & \text{for } \ \boldsymbol{\theta}^\top \boldsymbol{x} + \boldsymbol{\theta}_0 > 0 \\ -1 & \text{for } \boldsymbol{\theta}^\top \boldsymbol{x} + \boldsymbol{\theta}_0 \leq 0 \end{cases}$$

In this case the classifier represents a **hyperplane** through $\mathbb{R}^d$. If $\boldsymbol{\theta}_0 = \boldsymbol{0}$ then the hyperplane intersects the origin.



We say that the training data S^n is **linearly seperable** if we can design a linear classifier that is correct for all $\boldsymbol{x}^{(i)}, y^{(i)}$.

0.1 Perceptron Algorithm

The **Perceptron Algorithm** is the simplest algorithm we can use to optimize the parameters of our linear classifier. Simply, we compute the product of the parameters with the observation as in $h(\boldsymbol{x})$ and, if the sign of $y^{(i)}$ does not equal the sign of $\boldsymbol{\theta}^\top \boldsymbol{x}^{(i)} + \boldsymbol{\theta}_0$, then we know the observation is incorrect, and can update our parameters by $\boldsymbol{\theta} = \boldsymbol{\theta} + y^{(i)} \boldsymbol{x}^{(i)}$. In algorithmic notation:

```
function PERCEPTRON(D, T)
    ▷ Initialize parameters
    θ ← 0
    θ_0 ← 0

    for t = 1, ..., T do
        for i = 1, ..., n do
            if y^(i)(θ^⊤ x^(i) + θ_0) ≤ 0 then
                ▷ Update weight and bias
                θ ← θ + y^(i) x^(i)
                θ_0 ← θ_0 + y^(i)

    return (θ, θ_0)
```

0.2 Perceptron Convergence

In order to understand convergence we can first specify a linearly seperable dataset formally as **Definition:** $S_n = \left\{ \left(x^{(i)}, y^{(i)} \right), i = 1, ..., n \right\}$ is **linearly seperable** with margin γ if there exists ssome margin

$$y^{(i)} \left(\boldsymbol{\theta}^{\top} \boldsymbol{x}^{(i)} + \boldsymbol{\theta}_{\mathbf{0}} \right) \geq \gamma \| \boldsymbol{\theta}_0 \|$$

for *all* training points. We can further this understanding by considering the projection of a point $\boldsymbol{x}^{(i)}$. The projection must of course exist on the hyperplane governed by $\boldsymbol{\theta}^T \boldsymbol{x}_0^{(i)} + \boldsymbol{\theta}_0 = 0$ for projected point $\boldsymbol{x}_0$. Since this exists on our hyperplane the orthogonal complement (the distance from $x_0^{(i)}$ to $\boldsymbol{x}^{(i)}$ is simply

$$d = \left\| \boldsymbol{x}^{(i)} - \boldsymbol{x}_0^{(i)} \right\|$$

So, given that there exists some

a) $\boldsymbol{\theta}^*$ such that $y^{(i)} \left(\boldsymbol{\theta}^{*\top} \boldsymbol{x}^{(i)} + \boldsymbol{\theta}_{\mathbf{0}}^* \right) \geq \gamma \| \boldsymbol{\theta}_0^* \|$

b) All examples are bounded by $\left\| \boldsymbol{x}^{(i)} \right\| \leq R$ to ensure finite vectors

The **perceptron convergence theorem** states that the percetron algorithm will make at most R^2 / γ^2 mistakes on the way to find the correct classifier. Remarkably, this does **not** depend on the length of the feature vector nor the number of examples!