A simple linear classifier

CSE 250B
Linear decision boundary for classification: example

- What is the formula for this boundary?
- What label would we predict for a new point $x$?
Linear classifiers

**Binary classification:** data \( x \in \mathbb{R}^d \) and labels \( y \in \{-1, +1\} \)

- Linear classifier:
  - Parameters: \( w \in \mathbb{R}^d \) and \( b \in \mathbb{R} \)
  - Decision boundary \( w \cdot x + b = 0 \)
  - On point \( x \), predict label \( \text{sign}(w \cdot x + b) \)

- If the true label on point \( x \) is \( y \):
  - Classifier correct if \( y(w \cdot x + b) > 0 \)
A loss function for classification

What is the loss of the linear classifier $w, b$ on a point $(x, y)$?

One idea for a loss function:

- If $y(w \cdot x + b) > 0$: correct, no loss
- If $y(w \cdot x + b) < 0$: loss $= -y(w \cdot x + b)$
A simple learning algorithm

Fit a linear classifier $w$, $b$ to the training set using **stochastic gradient descent**.

- Update $w$, $b$ based on just one data point $(x, y)$ at a time
- If $y(w \cdot x + b) > 0$: zero loss, no update
- If $y(w \cdot x + b) \leq 0$: loss is $-y(w \cdot x + b)$
A simple learning algorithm

Fit a linear classifier $w, b$ to the training set using **stochastic gradient descent**.

- Update $w, b$ based on just one data point $(x, y)$ at a time
- If $y(w \cdot x + b) > 0$: zero loss, no update
- If $y(w \cdot x + b) \leq 0$: loss is $-y(w \cdot x + b)$

**The Perceptron algorithm**

- Initialize $w = 0$ and $b = 0$
- Keep cycling through the training data $(x, y)$:
  - If $y(w \cdot x + b) \leq 0$ (i.e. point misclassified):
    - $w = w + yx$
    - $b = b + y$
The Perceptron in action

85 data points, linearly separable.
The Perceptron in action

85 data points, linearly separable.
The Perceptron in action

85 data points, linearly separable.
The Perceptron in action

85 data points, linearly separable.
The Perceptron in action

85 data points, linearly separable.
The Perceptron in action

85 data points, linearly separable.
**Perceptron: convergence**

**Theorem:** Let $R = \max \|x^{(i)}\|$. Suppose there is a unit vector $w^*$ and some (margin) $\gamma > 0$ such that

$$y^{(i)}(w^* \cdot x^{(i)}) \geq \gamma \quad \text{for all } i.$$ 

Then the Perceptron algorithm converges within $R^2/\gamma^2$ updates.
**Perceptron: convergence**

**Theorem:** Let $R = \max \|x^{(i)}\|$. Suppose there is a unit vector $w^*$ and some (margin) $\gamma > 0$ such that

$$y^{(i)}(w^* \cdot x^{(i)}) \geq \gamma \quad \text{for all } i.$$

Then the Perceptron algorithm converges within $R^2/\gamma^2$ updates.

**Proof idea.** Let $w_t$ be the classifier after $t$ updates.

Track angle between $w_t$ and $w^*$:

$$\cos(\angle(w_t, w^*)) = \frac{w_t \cdot w^*}{\|w\|}.$$
**Perceptron: convergence**

**Theorem:** Let \( R = \max \| x^{(i)} \|. \) Suppose there is a unit vector \( w^* \) and some (margin) \( \gamma > 0 \) such that

\[
y^{(i)}(w^* \cdot x^{(i)}) \geq \gamma \quad \text{for all } i.
\]

Then the Perceptron algorithm converges within \( R^2 / \gamma^2 \) updates.

**Proof idea.** Let \( w_t \) be the classifier after \( t \) updates.

\[
\text{Track angle between } w_t \text{ and } w^*:\
\]

\[
\cos(\angle(w_t, w^*)) = \frac{w_t \cdot w^*}{\|w\|}.
\]

On each mistake, when \( w_t \) is updated to \( w_{t+1} \),

- \( w_t \cdot w^* \) grows significantly.
- \( \|w_t\| \) does not grow much.