Acknowledgement:
Many of these slides are derived from Tom Mitchell, Pascal Poupart, Pieter Abbeel, Eric Eaton, Carlos Guestrin, William Cohen, and Andrew Moore.
Threshold Perceptron Learning

- Dot products $x^Tx \geq 0$ and $-x^Tx \leq 0$

- Perceptron computes
  - 1 when $w^Tx = \sum_i x_i w_i + w_0 > 0$
  - 0 when $w^Tx = \sum_i x_i w_i + w_0 < 0$

- If output should be 1 instead of 0
  - $w \leftarrow w + x$ since $(w + x)^Tx \geq w^Tx$

- If output should be 1 instead of 0
  - $w \leftarrow w - x$ since $(w - x)^Tx \leq w^Tx$
Sequential Gradient Descent

- Let $y \in \{-1, 1\} \ \forall y$
- Let $M = \{(x_n, y_n) \ \forall n\}$ be set of misclassified examples
  - i.e. $y_n w^T x < 0$
- Find $w$ that minimizes misclassification error:
  - $E(w) = \sum_{(x_n, y_n) \in M} y_n w^T x$
- Gradient $\nabla E = - \sum_{(x_n, y_n) \in M} y_n x_n$
- **Sequential gradient descent**
  - Adjust $w$ based on one example $(x, y)$ at a time
    - $w \leftarrow w + \eta y x$
  - When $\eta = 1$, we recover the threshold perceptron algorithm
Perceptron Algorithm

- Let $y \in \{-1, 1\}$ $\forall y$

- Start with randomly initialized weights: $w$

- For $t = 1..T$ (T passes over data)
  - For $l = 1..n$: (each training example)
    - Classify with current weights
      - $\hat{y} = \text{sign} (w^T x)$ where $\text{sign}(x) = +1$ if $x > 0$ else -1
    - If correct (i.e., $\hat{y} = y^l$), no change! )
    - If wrong: update:
      - $w \leftarrow w + y^l x^l$
Perceptron vs. logistic regression: update rule

- **Logistic regression** when $y \in \{0, 1\}$

  $$w^{(t+1)} \leftarrow w^{(t)} + \eta \sum_{l} X^l(Y^l - \hat{P}(Y^l = 1 | X^l, W))$$

  $\text{NOTE: considers all training data in the batch mode}$

- **Perceptron** when $y \in \{-1, 1\}$

  - If misclassified: $w^{(t+1)} \leftarrow w^{(t)} + Y^lX^l$

  $\text{NOTE: considers only one data point}$
Perceptron vs. logistic regression: update rule

- Logistic regression

\[ w^{(t+1)} \leftarrow w^{(t)} + \eta \sum_l x^l(y^l - \hat{P}(y^l = 1 | x^l, W)) \]

- Perceptron when \( y \in \{0, 1\} \)

\[ w^{(t+1)} \leftarrow w^{(t)} + [y^l - \text{sign}^0(W^T x^l)]x^l \]

- \( \text{where} \ \text{sign}^0(x) = +1 \text{ if } x > 0 \text{ else } 0 \)

- Differences

  - Batch vs. stochastic
  - Probabilistic vs. error driven learning
Properties of Threshold Perceptron

- Hypothesis space \( h_w \)
- Binary classifications with parameters \( w \)
- Since \( w^T x \) is linear in \( w \), perceptron is a linear separator
- Converges \textit{iff} the data is linearly separable
Perceptron Linear Separability

- Examples

Linearly Separable

- Linearly Nonseparable
Sigmoid Perceptron

- “Soft” linear separator

Can we use sigmoid perceptron for linearly nonseparable data points?
Sigmoid Perceptron Learning

- Maximum likelihood estimation
  - Equivalent to logistic regression

- Objective function can be:
  - Miniminim squared error

\[
E(w) = \frac{1}{2} \sum_n E_n(w)^2 = \frac{1}{2} \sum_n (y_n - \sigma(w^T x_n))^2
\]
Gradient

- **Derivation**

\[
\frac{\partial E}{\partial w_i} = \sum_{n} E_n \frac{\partial E_n}{\partial w_i} \\
= \sum_{n} E_n(w)\sigma'(w^T x_n) x_i \\
= \sum_{n} E_n(w)\sigma(w^T x_n)(1 - \sigma(w^T x_n))x_i
\]

Recall,

\[
\sigma(x) = \frac{1}{1 + e^{-x}} \\
\sigma'(x) = \frac{d}{dx} \sigma(x) = \sigma(x)(1 - \sigma(x))
\]

No closed form solution!
Gradient Descent

- Perceptron-Learning(examples, network)
  - Repeat
  - For each \((x_n, y_n)\) in examples, do:
    - \(E_n \leftarrow y_n - \sigma(w^T x)\)
    - \(w \leftarrow w + \eta E_n \sigma(w^T x) (1 - \sigma(w^T x)) x_n\)
  - Until some stopping criteria satisfied
  - Return learnt network
Demo Time 😊

https://playground.tensorflow.org/