ECE/CS 559 Lecture 6 9/12

Last time: Perceptron Learning Algorithm

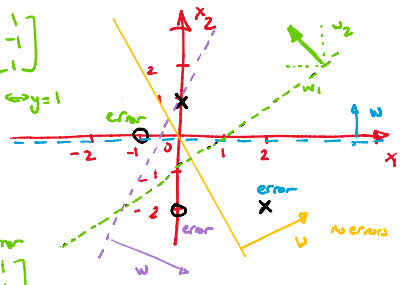
Algorithm: Input: Data $= \{(x,y), \dots\}$ $x \in \mathbb{R}^n$ $y \in \{0,1\}$, η
Output: weights $w \in \mathbb{R}^n$

1. Initialize w arbitrarily.
2. While there exists $(x,y) \in$ Data such that $y_w(x) \neq y$:
3. [1 epoch] For each $(x,y) \in$ Data:
   $w \leftarrow w + \eta x (y - y_w(x))$

Theorem: If classes are linearly separable, will converge for any η.

Example: 1. Initialize $w = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$

$y_w(x) = \begin{cases} 1 & ; 1 - 1x_1 + 1x_2 \geq 0 \quad x_2 \geq x_1 - 1 \Leftrightarrow y = 1 \\ 0 & ; 1 - 1x_1 + 1x_2 < 0 \end{cases}$



2. Epoch 1: (here, $\eta = 1$)
3. Updates:
   $w = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \leftarrow w - \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$ (error) $\quad x_2 \geq 0 \Leftrightarrow y = 1$
   $w = \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix} \leftarrow w + \begin{bmatrix} 1 \\ 2 \\ -2 \end{bmatrix}$ (error) $\quad 1 + 2x_1 - x_2 \geq 0$, $x_2 \leq 2x_1 + 1 \Leftrightarrow y = 1$
2. Epoch 2:
3. Updates: $w = \begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix} \leftarrow w - \begin{bmatrix} 1 \\ 0 \\ -2 \end{bmatrix}$ (error) $\quad 0 + 2x_1 + x_2 \geq 0$, $x_2 \geq -2x_1 \Leftrightarrow y = 1$

No more errors, terminate & return w.

① Supervised Learning

- Data (x, y), $x \in \mathcal{X}$ (feature space, also "context"), $y \in \mathcal{Y}$ (label space, also "target")
  - supervision
  - y discrete → classification, y continuous → regression
- Task: Predict/imitate y using only x.
- How: Use a neural network $y_w(x) = f(x; w)$
  This gives us a predictor, a function of x parametrized by w.
- Loss: Measures how far y and $f(x;w)$ are:
  0-1 loss: $\ell(y,f) = \mathbb{1}\{y \neq f\}$ Squared loss: $\ell(y,f) = \|y - f\|^2$
- Risk: Average loss over a data set: Data $= \{(x,y), \dots\}$
  $R(w) = \frac{1}{|\text{Data}|} \sum_{(x,y) \in \text{Data}} \ell(y, f(x,w))$ (empirical)

- Goal: Minimize (empirical) risk,
  make few mistakes/errors / stay close to y.
- Types of algorithms:
  - Online: data streams in, weights updated along the way
    (usually) don't revisit past data points
    - If we forgo epochs, perceptron = online.
  - Batch: data is available in whole, update weights using it all
    (usually) pass over data multiple times (epochs)

Online:
- See (x,y), update w
- See (x',y'), update w
- See (x'',y''), update w
⋮

Batch:
- Get Data $= \{(x,y), \dots\}$
- Epoch 1: use Data, update w
- Epoch 2: use Data, update w
⋮

- Mini batch: Process data gradually in small batches
  - Get Data - Divide it into small batches $\{\text{batch}_1, \text{batch}_2, \dots\}$ $|\text{batch}|$ = batch size
  - See batch_1, update w
  - See batch_2, update w
  ⋮
  } repeat (epochs)