Data and model

- Data
  - X_train or X_test contains multiple data, where each column is a data instance
  - Each data instance $\boldsymbol{x}$ has already include “1” as its last element

- Model
  - $f(\boldsymbol{x}) = \mathrm{sign}(\boldsymbol{w}^T \boldsymbol{x})$ for binary linear classification. There is no b since it has been included in $\boldsymbol{w}$

- Your goal is to find $\boldsymbol{w}$, using logistic regression and perceptron

Logistic Regression

- Training data: $\{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_N, y_N)\}$; $y_n \in \{0,1\}$
- Model: $f(\boldsymbol{x}) = \mathrm{sign}(\boldsymbol{w}^T\boldsymbol{x})$

- Logistic loss:
  - $p_n = p(1|\boldsymbol{x}_n; \boldsymbol{w}) = \sigma(\boldsymbol{w}^T\boldsymbol{x}_n) = \frac{1}{1+e^{-(\boldsymbol{w}^T\boldsymbol{x}_n)}}$
  - Minimize $-\frac{1}{N}\sum_{n=1}^{N} \log p(y_n|\boldsymbol{x}_n; \boldsymbol{w}) = -\frac{1}{N}\sum_{n=1}^{N}\{y_n \times \log p_n + (1 - y_n) \times \log(1 - p_n)\}$

Logistic Regression

- How to optimize w.r.t. $\boldsymbol{w}$?
  - Gradient descent!
  - Loss $= -\frac{1}{N}\sum_{n=1}^{N}\{y_n \times \log p_n + (1 - y_n) \times \log(1 - p_n)\}$

- For t = 1:T
  - $\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta \times \nabla_{\boldsymbol{w}}\mathrm{Loss}$ ⟵ This is the what you need to implement! See the next slide.

- η is the learning rate

Logistic Regression

- $\nabla_{\boldsymbol{w}}\text{Loss} = -\frac{1}{N}\sum_{n=1}^{N} \nabla_{\boldsymbol{w}}\{y_n \times \log p_n + (1-y_n) \times \log(1-p_n)\}$

- $\nabla_{\boldsymbol{w}}\{y_n \times \log p_n + (1-y_n) \times \log(1-p_n)\} =$

$$y_n \times \nabla_{\boldsymbol{w}} \log\sigma\left(\boldsymbol{w}^T\boldsymbol{x}_n\right) + (1-y_n) \times \nabla_{\boldsymbol{w}} \log\left(1-\sigma\left(\boldsymbol{w}^T\boldsymbol{x}_n\right)\right) =$$

$$y_n \times \left(1-\sigma\left(\boldsymbol{w}^T\boldsymbol{x}_n\right)\right)\boldsymbol{x}_n - (1-y_n) \times \sigma\left(\boldsymbol{w}^T\boldsymbol{x}_n\right)\boldsymbol{x}_n =$$

$$\left(y_n - \sigma\left(\boldsymbol{w}^T\boldsymbol{x}_n\right)\right)\boldsymbol{x}_n$$

This is the main thing you need to compute

- Caution: be mindful of "$\frac{1}{N}$" in $\nabla_{\boldsymbol{w}}\text{Loss}$ and "+ or -" in computing gradients

Perceptron

- Training data: $\{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_N, y_N)\}$; $y_n \in \{-1,1\}$
- Model: $f(\boldsymbol{x}) = \operatorname{sign}(\boldsymbol{w}^T \boldsymbol{x})$

- For t = 1 : T
  - Loop for all training examples $\boldsymbol{x}_n$
  - Predict $\hat{y}_n = \operatorname{sign}\left(\boldsymbol{w}^T \boldsymbol{x}_n\right)$
  - If $\hat{y}_n \neq y_n$
    - Update: $\boldsymbol{w} \leftarrow \boldsymbol{w} + \eta(y_n \boldsymbol{x}_n)$