Review

1. ## Information theory (entropy & gain)

   - ### Entropy

     $H(X) = -\sum_i P(X = x_i) \log_2 P(X = x_i)$

     Q: for a variable X of Bernoulli distribution $Pr(X = 1) = p$, what is $\arg\min_p H(X)$ and $\arg\max_p H(X)$?

     [Solution]: consider the coin-flip case...

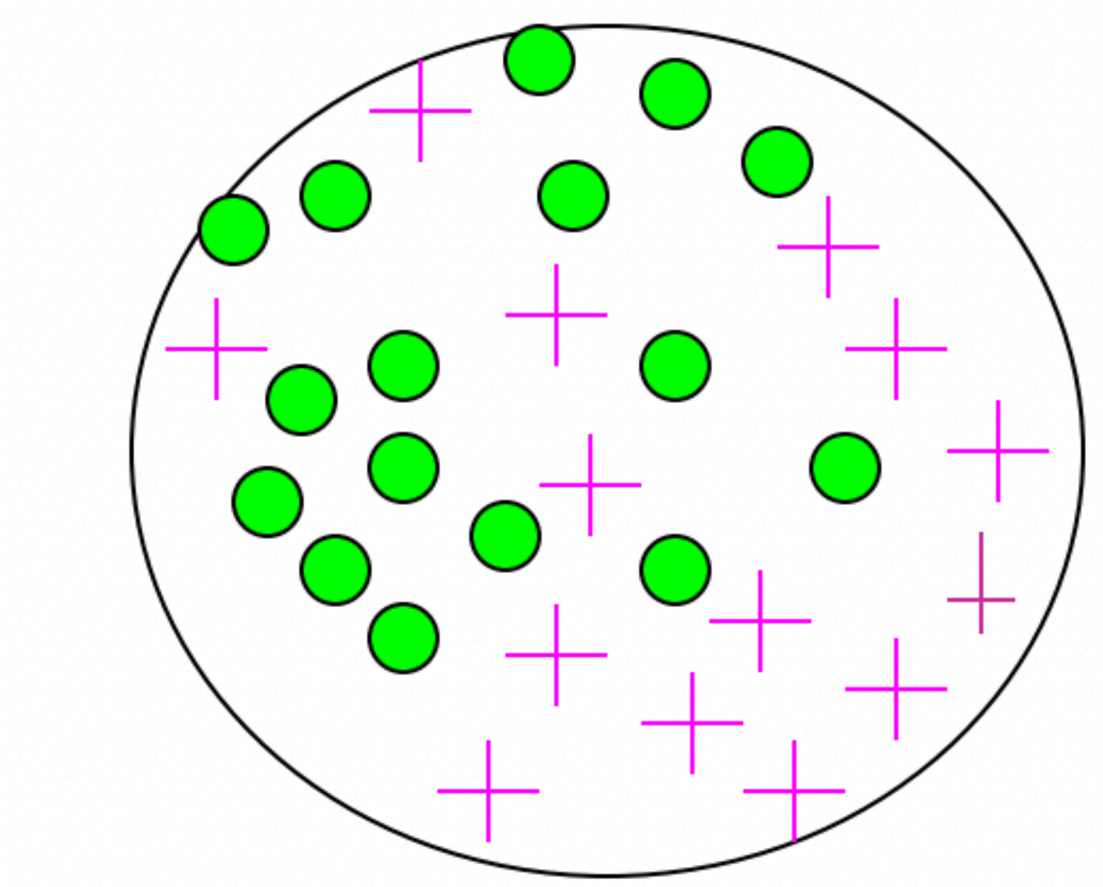
	p(head)	H	Interpretation
H_max	0.5	$-(0.5 \log_2 0.5 + 0.5 \log_2 0.5) = 1$	entropy measures uncertainty -- we are most uncertain about the outcome: there is a 50/50 chance the coin landing on head/tail
H_min	0.0 or 1.0	$-(0.0 \log_2 0.0 + 1.0 \log_2 1.0) = 0$	entropy measures uncertainty -- we are most certain about the outcome: the coin will definitely land on head/tail. The uncertainty is just 0 because there is no uncertainty about the outcome!

   - ### Information gain

$$
\begin{aligned}
G_{split} &= H_{\text{parent}} - \overline{H_{\text{children}}} \\
&= H(X) - \sum_{i=1}^{t} \frac{|X_i|}{|X|} H(X_i)
\end{aligned}
$$

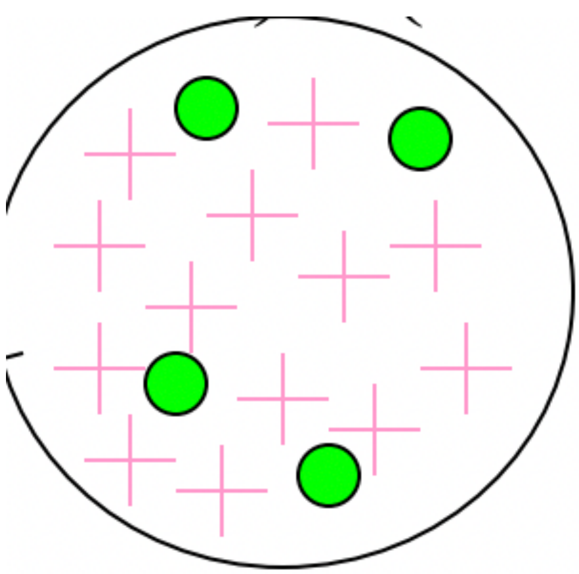
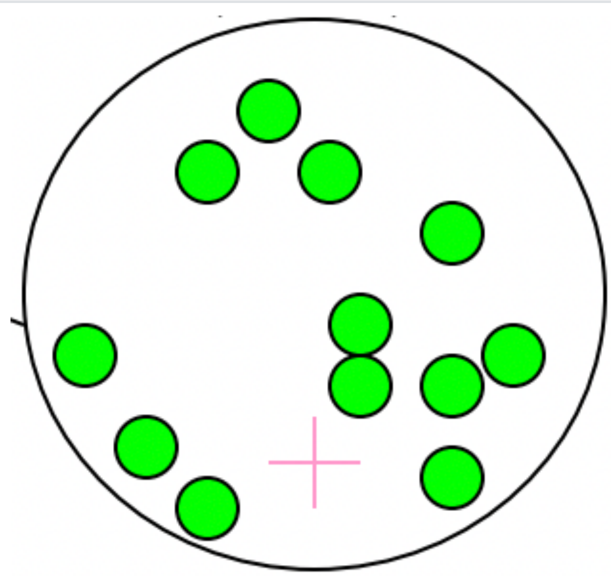
     Example (from https://homes.cs.washington.edu/~shapiro/EE596/notes/InfoGain.pdf):

     parent



(16 positive v.s. 14 negative)

children

<table>
<tr><th>Child 1</th><th>Child 2</th></tr>
<tr><td>
(4 positive v.s. 13 negative)</td><td>
(12 positive v.s. 1 negative)</td></tr>
</table>

Q: what is the information gain of this split?

$$H(S) = -(\frac{16}{30}\log(\frac{16}{30}) + \frac{14}{30}\log(\frac{14}{30}))$$
$$H(S_1) = -(\frac{4}{17}\log(\frac{4}{17}) + \frac{13}{17}\log(\frac{13}{17}))$$
$$H(S_2) = -(\frac{1}{13}\log(\frac{1}{13}) + \frac{12}{13}\log(\frac{12}{13}))$$
$$G = H(S) - (\frac{|S1|}{|S|}H(S_1) + \frac{|S2|}{|S|}H(S_2)) = H(S) - (\frac{17}{30}H(S_1) + \frac{13}{30}H(S_2))$$

2. **KNNs**
   - None-parametric, data-based,
   - nearest neighbor (k=1)
   - distance metrics:
     - most common: Euclidean distance

       e.g. for data points $x_1 = (a_1, b_1)$ and $x_2 = (a_2, b_2)$, their distance is $\sqrt{(a_1 - a_2)^2 + (b_1 - b_2)^2}$
   - Issues: (test time, memory)
3. **Decision trees**
   - **Stump:**

     pick a feature that best separates the data
   - Tree:
     - the algorithm: time complexity $O(mn^2 \log n)$, where m, n are #features and #training data respectively.
   - **rule of thumb:**
     - low complexity (the shallow is better than the deep)
     - less overfitting (the balanced is better than the unbalanced)
       - If the tree is balanced and there are N nodes, what is the time complexity of making a prediction? ($O(\log n)$)
4. **Logistic Regressions**
   - decision boundary: $w^T x + b = 0$
     - Normal direction (or model parameter): w (pointing towards the positive are)
     - Translation (or the bias/scalar term): b
     - Distance: 'The distance (signed) of any point x to the decision boundary is $w^T x + b$' (???)

- logistic classifier:
  - Logit: $wx + b$
  - sigmoid function: $\sigma(t) = \frac{1}{1+exp(-t)}$, with domain $\mathbb{R}$ and range $(0, 1)$
  - Probability of x be positive: $p(y=1|x;w,b) = \frac{1}{1+\exp(-(wx+b))}$
  - Probability of x has label y: $p(y|x;w,b) = \frac{1}{1+\exp(-(wx+b)y)}$
  - prediction (logistic classification):

$$f(x) = \begin{cases} 1 & \text{if} \quad p(y=1|x;w,b) \geq 0.5 \\ -1 & \text{otherwise} \end{cases} \tag{1}$$

  - train a logistic classifier
    - loss function:

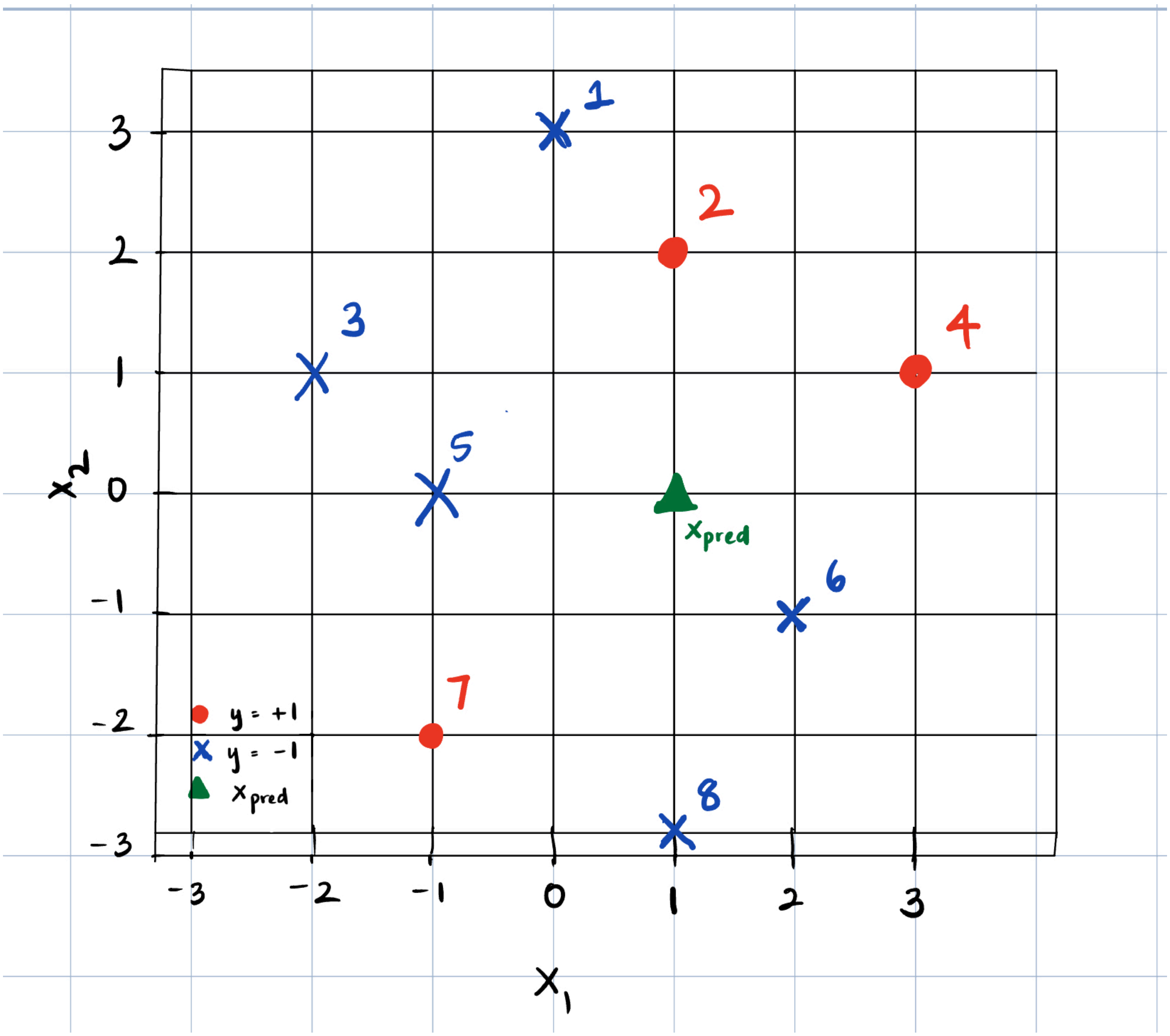
$$\begin{aligned} L &= -\ln(\text{likelihood}) \\ &= -\sum_{i=1}^{n} ln(1 + \exp(-y_i(w^T x_i + b))) \\ \frac{\partial L}{\partial w} &= \sum_{i=1}^{n} -y_i x_i (1 - p(y_i|x_i)) \\ \frac{\partial L}{\partial b} &= \sum_{i=1}^{n} -y_i (1 - p(y_i|x_i)) \end{aligned}$$

    Training: gradient descent...

Discussion Questions

1. ## KNN

   Consider a training dataset $S_{training}$ = {($\mathbf{x}_i, y_i$), $i = 1, 2.. . 8$} where each data point $(\mathbf{x}, y)$ has a feature vector $\mathbf{x} = [x_1, x_2]^T$ and the corresponding label $y \in \{-1, +1\}$. The points with the corresponding labels in the dataset are shown in the figure below. You are asked to predict the label of a point $\mathbf{x}_{\text{pred}}$ = [1, 0]T. Use the k-nearest neighbors (k-NN) method under Euclidean distance.



- Determine the predicted label for $\mathbf{x}_{\text{pred}}$ using the k- NN with k = 1, 3, 5.

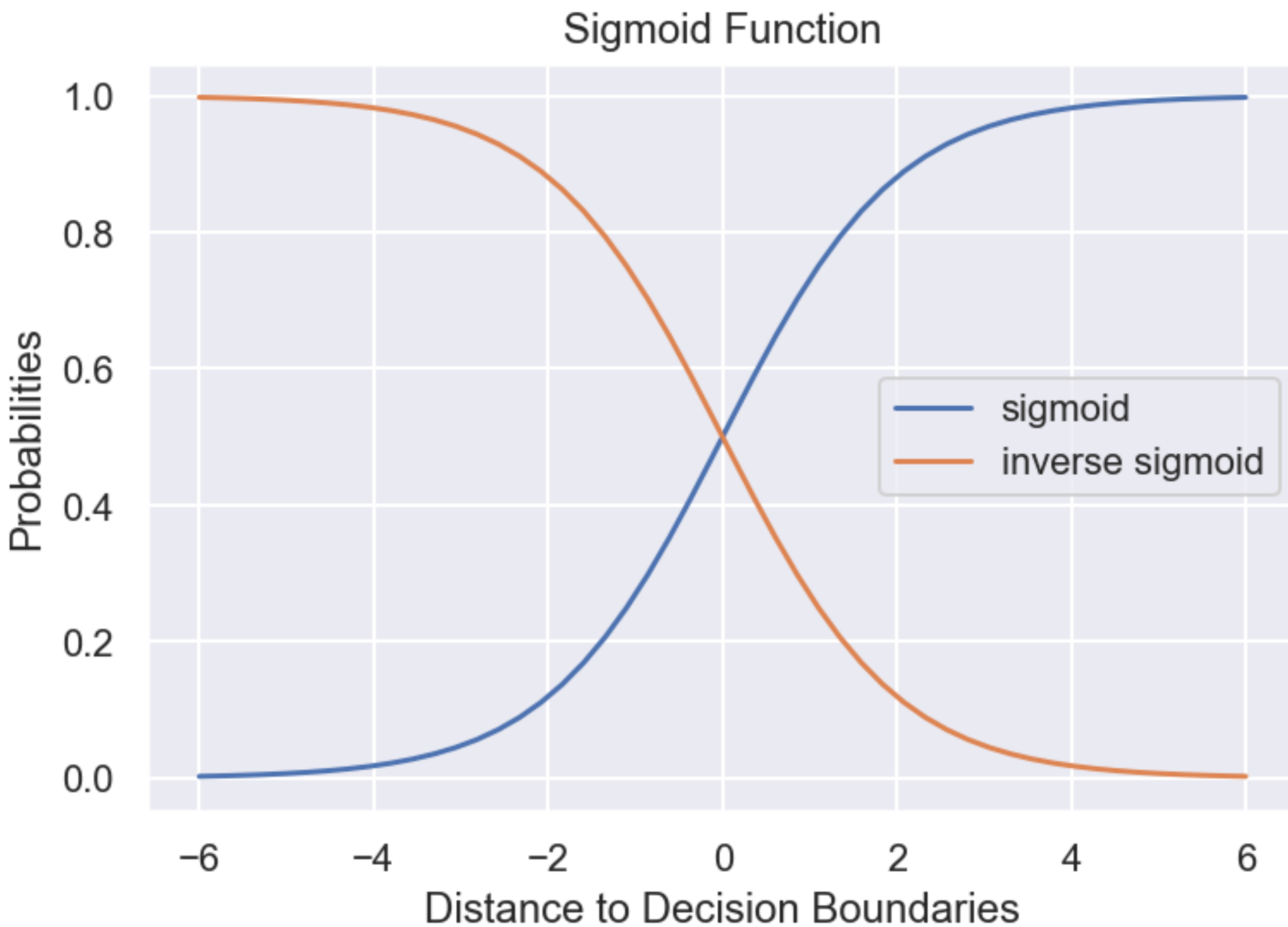
  (Hint: 'to classify a new input x, examine the k-closest training data points to x and assign the object to the most frequently occurring classes')

[Solution]: first rank the training data points in the order from nearest to furthest from x_{pred}.

k	k-nearest neighbors	labels of the k-nearest neighbors	Prediction
1	x_6	{-1}	-1
3	x_6, x_2, x_5	{-1, +1, -1}	-1
5	x_6, x_2, x_5, x_4, x_7	{-1, +1, -1, +1, +1}	+1

2. **Logistic Regression**

The logic function ϕ serves as a proxy that translates the distances between the data points to the decision boundary into probabilities.



The sigmoid function is defined by: $\phi(\vec{x}) = \frac{1}{1+e^{-x}}$

The loss function of the logistic regression is defined by:

$$\mathcal{L}(\mathbf{w}, b) = -\sum_{i=1}^{n} \ln p(y_i|\mathbf{x}_i) \tag{2}$$

Assume in a binary classification problem, we need to predict a binary label $y \in \{-1, 1\}$ for a feature vector $\mathbf{x} = [x_0, x_1]^\top$. In logistic regression, we can reformulate the binary classification problem in a probabilistic framework: We aim to model the distribution of classes given the input feature vector $\mathbf{x} \in \mathbb{R}^k$. Therefore, the dataset can be summarized by $X = \{\mathbf{x}_1, \mathbf{x}_2 \ldots \mathbf{x}_n\}^{n \times k}$

The partial derivatives of the parameters is given by

$$\frac{\partial \mathcal{L}(\mathbf{w}, b)}{\partial \mathbf{w}} = -\sum_{i=1}^{n} (1 - p_i) y_i \mathbf{x}_i$$
$$\frac{\partial \mathcal{L}(\mathbf{w}, b)}{\partial b} = -\sum_{i=1}^{n} (1 - p_i) y_i.$$

Q1: What is the shape of $\frac{\partial \mathcal{L}(\mathbf{w},b)}{\partial \mathbf{w}}$ and $\frac{\partial \mathcal{L}(\mathbf{w},b)}{\partial b}$?

[Solution]: same as w; same as b

(Assignment 3) Q: In reality, we typically tackle this problem in a matrix form: First, we represent data points as matrices $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^T$ and $Y = [y_1, y_2, \ldots, y_n]^T$. Thus, the negative log-likelihood loss $\mathcal{L}(\mathbf{w}, b)$ can be formulated as:

$$\mathbf{p} = \mathrm{sigmoid}(Y \circ (X\mathbf{w} + b\mathbf{1}))$$
$$\mathcal{L}(\mathbf{w}, b) = -\mathbf{1}^T \ln \mathbf{p}$$

where $\mathbf{1} = [1, 1, \ldots, 1]^T \in \mathbb{R}^n$ is a n-dimensional column vector, $\mathbf{p} = [p_1, p_2, \ldots, p_n]^T \in \mathbb{R}^n$ is a n-dimensional column vector, $\ln(\cdot)$ is an element-wise natural logarithm function, $\mathrm{sigmoid}(z) = \frac{1}{1+e^{-z}}$ is an element-wise sigmoid function, and $\circ$ is an element-wise product operator

(Hint: how to express summation as matrix multiplications? Consider L2-norm: $\|v\|^2 = \sum_{i=1}^{n} v_i^2 = v^T v$)

Q2: What does $\mathbf{p}$ stand for? What is the shape of $\mathbf{p}$? What is the shape of $L(w, b)$?

[Solution]: p_i stands for the probability that x_i has a label y_i; it is not the real probability, but the probability that our model (with parameters w and b) predicts. $\mathbf{p}$ is a column vector of size n. L is a scalar value.

3. **Decision Boundary**

We are given a classifier that performs classification in $\mathbb{R}^2$ (the space of data points with two features (x_1, x_2) with the following decision rule:

$$h(x_1, x_2) = \begin{cases} 1 & \text{if} \quad x_1^2 + x_2^2 - 10 \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Draw the decision boundary of the classifier and shade the region where the classifier predicts 1. Make sure you have marked the x_1 and x_2 axes and the intercepts on those axes.

[solution]: the decision boundary is a circle ($x_1^2 + x_2^2 = 10$). All points that fall outside the circle will be labeled as '1'.

Assignment 3

Q1: ...

Q2: logit -- $wx + b$ (it doesn't matter just make sure the plot looks correct...)

Q3: see discussion question 2.

Q4: see the first review question.

From Last Week

	L1	L2	L
Loss function	$L_1 = \sum_{i=1}^{N} \lvert x_i w - y_i \rvert$ `L_1=np.sum(np.abs(X@w-y))`	$L_2 = \sum_{i=1}^{N} (x_i w - y_i)^2$ `L_2=np.sum((X@w-y)**2)`	$L = \sum \alpha_k L_k$ where $\sum \alpha_k = 1$
derivative $\frac{\partial L}{\partial w}$	$\frac{\partial L_1}{\partial w} = \sum_{i=1}^{N} x_i * sign(x_i w - y_i) = X^T sign(Xw - y)$ `L1_grad =X.T@np.sign(X@w-y)`	$\frac{\partial L_2}{\partial w} = \sum_{i=1}^{N} 2x_i * (x_i w - y_i) = 2X^T(Xw - y)$ `L2_grad = 2 * X.T@(X@w-y)`	$\frac{\partial L}{\partial w} = \sum \alpha_k \frac{\partial L_k}{\partial w}$
to find the optimal solution	$w_{t+1} = w_t - \lambda \frac{\partial L1}{\partial w}$	$w^* = (X^T X)^{-1} X^T y$	$w_{t+1} = w_t - \lambda \frac{\partial L}{\partial w}$