Tóm tắt Linear Regression

Mô hình chung cho bài toán trong Deep Learning.

1. Visualize dữ liệu
2. Thiết lập model (Linear regression là hàm đường thẳng : $\widehat{y_i} = w_0 + w_1 x_1^{(i)} + \cdots + w_m x_m^i$)

   Logistic regresion là hàm sigmoid : $\sigma(x) = \frac{1}{1+e^{-x}}$

   $\widehat{y_i} = \sigma\left(w_0 + w_1 x_1^{(i)} + \cdots + w_m x_m^i\right) = \frac{1}{1+e^{-(w_0 + w_1 x_1^{(i)} + \cdots + w_m x_m^i)}}$ (cho $\widehat{y_i}$ về khoảng [0,1])
3. Thiết lập loss function (MSE) (Linear regression là hàm MSE = $\frac{\sum_{i=1}^{n}(y-\hat{y})^2}{n}$

   Logistic regression là hàm **binary crossentropy** : $L = -(y_i * \log(\hat{y}_i) + (1-y_i) * \log(1-\widehat{y_i}))$
4. Tìm tham số bằng việc tối ưu loss function (Gradient descent) Linear là $\frac{dJ}{dw} = X^T * (\hat{y} - y)$

   Logistic regression có hàm gradient descent là $\frac{dJ}{dw} = \frac{1}{N} * X^T * (\hat{y} - y)$
5. Dự đoán dữ liệu mới bằng model vừa tìm được

Thiết lập Model

$$\widehat{y_i} = w_0 + w_1 x_1^{(i)} + w_2 x_2^{(i)} + \cdots + w_m x_m^i$$

Công thức Gradient Descent

$$\frac{dJ}{dw} = X^T * (\hat{y} - y)$$

MSE Loss fucntion

$$MSE = \frac{\sum_{i=1}^{n}(y-\hat{y})^2}{n}$$

Giải bài toán bằng đại số tuyến tính

$w = [w0, w_1, w_2, \ldots, w_m]^T =$ là vector hệ số cần phải tối ưu, w0 thường được gọi là bias.

⇨ *Giải thích : w là ma trận (m+1) hàng và 1 cột để lưu các hệ số cần tìm*

$x^{(i)} = [1, x_1^i, x_2^i, \ldots, x_m^i]$ là hàng dữ liệu thứ i trong bộ n số lượng dữ liệu quan sát được, mỗi dữ liệu có m giá trị .

⇨ *Giải thích : X là ma trận đầu vào có n mẫu dữ liệu và m biến đầu vào (feature) thì ta sẽ tạo ra ma trận n hàng (m + 1) cột (do thêm cột 1 ở đầu cho tham số bias w0)*

⇨ *Ta sẽ có ma trận* X = $\begin{bmatrix} 1 & x_1^1 & \ldots & x_m^1 \\ 1 & x_1^2 & \ldots & x_m^2 \\ \ldots & \ldots & \ldots & \ldots \\ 1 & x_1^n & \ldots & x_m^n \end{bmatrix}$ *(ma trận n hàng và (m+1) cột) (i chạy từ 1 đến n)*

Ta có : Giá trị dự đoán y hat của thàng dữ liệu thứ i là $\widehat{y^{(i)}} = x^{(i)} w$

$X \in R^{n*(m+1)}, w \in R^{(m+1)*1}, y \in R^{n*1}$

$$Ta\ có:\ X = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \ldots \\ x^{(n)} \end{bmatrix} = \begin{bmatrix} 1 & x_1^1 & \ldots & x_m^1 \\ 1 & x_1^2 & \ldots & x_m^2 \\ \ldots & \ldots & \ldots & \ldots \\ 1 & x_1^n & \ldots & x_m^n \end{bmatrix}, w = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \ldots \\ w_m \end{bmatrix}, \hat{y} = X.w = \begin{bmatrix} w_0 + w_1 x_1^1 + w_2 x_2^1 + \cdots + w_m x_m^1 \\ w_0 + w_1 x_1^2 + w_2 x_2^2 + \cdots + w_m x_m^2 \\ \ldots \\ w_0 + w_1 x_1^n + w_2 x_2^n + \cdots + w_m x_m^n \end{bmatrix}$$

$$y = \begin{bmatrix} y^1 \\ y^2 \\ \dots \\ y^n \end{bmatrix}, y - \hat{y} = \begin{bmatrix} y^1 - (w_0 + w_1x_1^1 + w_2x_2^1 + \dots + w_mx_m^1) \\ y^2 - (w_0 + w_1x_1^2 + w_2x_2^2 + \dots + w_mx_m^2) \\ \dots \\ y^n - (w_0 + w_1x_1^n + w_2x_2^n + \dots + w_mx_m^n) \end{bmatrix}$$

⇨ *Giải thích : X là ma trận n× $(m+1)$, w là $ma\ trận\ (m+1) \times 1, \hat{y} = X.w$ sẽ $là\ ma\ trận\ n \times 1$*

Theo như hàm mất mát MSE Loss function :

$MSE = \frac{\sum_{i=1}^{n}(y - \hat{y})^2}{n}$ **hoặc** $MSE = \frac{1}{2} * \frac{\sum_{i=1}^{n}(y - \hat{y})^2}{n}$

Ta có : $L = \frac{1}{2} * \frac{1}{n} * \sum_{i=1}^{n}\left(y^i - \hat{y^i}\right)^2 = \frac{1}{2} * \frac{1}{n} * \sum_{i=1}^{n}\left(y^i - x^{(i)}.w\right)^2$

⇨ *Giải thích : MES cho ma trận X là mất mát của n hàng dữ liệu*

Định nghĩa Euclideannorm : $\|z\|_2 = (z_1^2 + z_2^2 + \dots + z_n^2)^{\frac{1}{2}} \Longrightarrow \|z\|_2^2 = (z_1^2 + z_2^2 + \dots + z_n^2)$

⇨ *Giải thích : Trong công thức Euclideannorm thì $\|z\|_2^2$, số 2 ở dưới là chuẩn 2 tức là căn bậc 2 và số 2 ở trên là mũ 2 và mũ hai của căn bậc 2 tức là hết căn có nghĩa là đơn giản ta chỉ bình phương của căn thôi .*

Mặc khác ta có :

$$L = \frac{1}{2} * \frac{1}{n} * \sum_{i=1}^{n}\left(y^i - \hat{y^i}\right)^2 = \frac{1}{2} * \frac{1}{n} * \left[\left(y^1 - \hat{y^1}\right)^2 + \left(y^2 - \hat{y^2}\right)^2 + \dots + \left(y^n - \hat{y^n}\right)^2\right]$$

Đặt $\left(y^i - \hat{y^i}\right)^2 = z_i^2$ Ta được $L = \frac{1}{2} * \frac{1}{n} * (z_1^2 + z_2^2 + \dots + z_n^2) = \frac{1}{2} * \frac{1}{n} * \|z\|_2^2$

Viết lại

$$L = \frac{1}{2} * \frac{1}{n} * \sum_{i=1}^{n}\left(y^i - \hat{y^i}\right)^2 = \frac{1}{2} * \frac{1}{n} * \left[\left(y^1 - \hat{y^1}\right)^2 + \left(y^2 - \hat{y^2}\right)^2 + \dots + \left(y^n - \hat{y^n}\right)^2\right]$$

$$= \frac{1}{2} * \frac{1}{n} * \|y - \hat{y}\|_2^2 = \frac{1}{2} * \frac{1}{n} * \|y - Xw\|_2^2 = \frac{1}{2} * \frac{1}{n} * (y - \hat{y})^T * (y - \hat{y})$$

⇨ *Giải thích :* $\|y - \hat{y}\|_2^2 = \left(y^1 - \hat{y^1}\right)^2 + \left(y^2 - \hat{y^2}\right)^2 + \dots + \left(y^n - \hat{y^n}\right)^2$

⇨ *Ta thấy :* $y - \hat{y} = \begin{bmatrix} y^1 - \hat{y^1} \\ y^2 - \hat{y^2} \\ \dots \\ y^n - \hat{y^n} \end{bmatrix} \Longrightarrow (y - \hat{y})^T = \begin{bmatrix} y^1 - \hat{y^1} & y^2 - \hat{y^2} & \dots & y^n - \hat{y^n} \end{bmatrix}$

⇨ *Mà* $(y - \hat{y})^T * (y - \hat{y}) = \begin{bmatrix} y^1 - \hat{y^1} & y^2 - \hat{y^2} & \dots & y^n - \hat{y^n} \end{bmatrix} * \begin{bmatrix} y^1 - \hat{y^1} \\ y^2 - \hat{y^2} \\ \dots \\ y^n - \hat{y^n} \end{bmatrix}$

$= (y^1 - \hat{y^1}) * (y^1 - \hat{y^1}) + (y^2 - \hat{y^2}) * (y^2 - \hat{y^2}) + \dots + (y^n - \hat{y^n}) * (y^n - \hat{y^n})$

$= \left(y^1 - \hat{y^1}\right)^2 + \left(y^2 - \hat{y^2}\right)^2 + \dots + \left(y^n - \hat{y^n}\right)^2$ *Nhân 2 ma trận 1×n và n×1*

Từ đó suy ra :

$$\boldsymbol{L = \frac{1}{2} * \frac{1}{n} * \|y - \hat{y}\|_2^2 = \frac{1}{2} * \frac{1}{n} * \|y - Xw\|_2^2 = \frac{1}{2} * \frac{1}{n} * (y - \hat{y})^T * (y - \hat{y})}$$

Bây giờ tìm đi tìm w sao cho L (giá trị mất mát) là nhỏ nhất , như thường lệ ta sẽ khảo sát , nghĩa là ta sẽ đạo hàm L theo w và tìm nghiệm w để cho $L' = 0$

Lưu ý :

+ X là một ma trận bất kì thì ta có $A = X^T * X$ thì A sẽ luôn là một ma trận đối xứng nghĩa là $A^T = A$

+ $(A * B)^T = B^T * A^T$ (công thức liên quan đến ma trận chuyển vị trong đại số tuyến tính)

Giải bài toán $L' = 0$

L = $\frac{1}{2} * \frac{1}{n} * (y - \hat{y})^T * (y - \hat{y})$. Ta có thể cho L = $(y - \hat{y})^T * (y - \hat{y})$ vì $\frac{1}{2} * \frac{1}{n}$ là hằng

L = $(y - \hat{y})^T * (y - \hat{y}) = (y - X.w)^T * (y - X.w)$

= $(y^T - w^T X^T) * (y - Xw) = \boldsymbol{y^T y - y^T X w - w^T X^T y + w^T X^T X w}$

Đạo hàm :

$L'_w = - y^T X - [(Xw)^T y + w^T X^T X w]'_w = -y^T X - [(Xw)^T (y^T)^T + w^T X^T X w]'_w$

$= - y^T X - [(y^T X w)^T + w^T X^T X w]'_w = - y^T X - [(y^T X w)^T + w^T X^T X w\ ]'_w$

$= - y^T X - [(y^T X w)^T\ ]'_w + [w^T X^T X w\ ]'_w$

$= - y^T X - [(y^T X) w]'_w + [w^T X^T X w\ ]'_w$

$= - y^T X - y^T X + [w^T X^T X w\ ]'_w$

$= -2 y^T X + [w^T X^T X w\ ]'_w$

⇨ Cần đi tìm $[w^T X^T X w\ ]'_w$

Tìm $[w^T X^T X w\ ]'_w$:

$$+ \ x^{(i)} w = [1 \quad x_1^i \quad \ldots \quad x_m^i] * \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \ldots \\ w_m \end{bmatrix} = w_0 + w_1 x_1^i + w_2 x_2^i + \cdots + w_m x_m^i$$

$$\Longrightarrow \frac{d\left(x^{(i)} w\right)}{dw} = \left[\frac{d\left(x^{(i)} w\right)}{dw0} \quad \frac{d\left(x^{(i)} w\right)}{dw_1} \quad \ldots \quad \frac{d\left(x^{(i)} w\right)}{dw_m}\right] = [1 \quad x_1^1 \quad \ldots \quad x_m^i] = x^{(i)}$$

$$\Longrightarrow \frac{d(Xw)}{dw} = \begin{bmatrix} \frac{d\left(x^{(1)} w\right)}{dw0} & \frac{d\left(x^{(1)} w\right)}{dw_1} & \ldots & \frac{d\left(x^{(1)} w\right)}{dw_m} \\ \frac{d\left(x^{(2)} w\right)}{dw0} & \frac{d\left(x^{(2)} w\right)}{dw_1} & \ldots & \frac{d\left(x^{(2)} w\right)}{dw_m} \\ \ldots & \ldots & \ldots & \ldots \\ \frac{d\left(x^{(n)} w\right)}{dw0} & \frac{d\left(x^{(n)} w\right)}{dw_1} & \ldots & \frac{d\left(x^{(n)} w\right)}{dw_m} \end{bmatrix} = \begin{bmatrix} 1 & x_1^1 & \ldots & x_m^1 \\ 1 & x_1^2 & \ldots & x_m^2 \\ \ldots & \ldots & \ldots & \ldots \\ 1 & x_1^n & \ldots & x_m^n \end{bmatrix} = X$$

$+ \; wx^{(i)} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \dots \\ w_m \end{bmatrix} * \begin{bmatrix} 1 & x_1^i & \dots & x_m^i \end{bmatrix} = w_0 + w_1 x_1^i + w_2 x_2^i + \dots + w_m x_m^i$

$$= \begin{bmatrix} w_0 & w_0 x_1^i & \dots & w_0 x_m^i \\ w_1 & w_1 x_1^i & \dots & w_1 x_m^i \\ \dots & \dots & \dots & \dots \\ w_m & w_m x_1^i & \dots & w_m x_m^i \end{bmatrix}$$

$$\Longrightarrow \frac{d\left(wx^{(i)}\right)}{dw} = \begin{bmatrix} \frac{d\left(wx^{(i)}\right)}{dw0} \\ \frac{d\left(wx^{(i)}\right)}{dw_1} \\ \dots \\ \frac{d\left(wx^{(i)}\right)}{dw_m} \end{bmatrix} = \begin{bmatrix} 1 \\ x_1^i \\ \dots \\ x_m^i \end{bmatrix} = (x^{(i)})^T$$

$$\Longrightarrow \frac{d(Xw)}{dw} = \begin{bmatrix} \frac{d\left(wx^{(1)}\right)}{dw_0} & \frac{d\left(x^{(2)}w\right)}{dw_0} & \dots & \frac{d\left(x^{(n)}w\right)}{dw_0} \\ \frac{d\left(wx^{(1)}\right)}{dw_1} & \frac{d\left(x^{(2)}w\right)}{dw_1} & \dots & \frac{d\left(x^{(n)}w\right)}{dw_1} \\ \dots & \dots & \dots & \dots \\ \frac{d\left(wx^{(1)}\right)}{dw_m} & \frac{d\left(x^{(2)}w\right)}{dw_m} & \dots & \frac{d\left(x^{(n)}w\right)}{dw_m} \end{bmatrix} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1^1 & x_1^2 & \dots & x_1^n \\ \dots & \dots & \dots & \dots \\ x_m^1 & x_m^2 & \dots & x_m^n \end{bmatrix} = X^T$$

$+ \; \frac{d(w^T X w)}{dw} = w^T X + \frac{d((Xw)^T w)}{dw} = w^T X + w^T X^T = w^T(X + X^T) = 2w^T X$ (Nếu X đối xứng thì $X^T = X$)

Quay lại bài toán $L'_w = 0$

Giải thích : Đặt $X^T X = H$. Mà H là ma trận đối xứng .

Áp dụng kết quả đã được chứng minh ở trên $\frac{d(w^T X w)}{dw} = 2w^T X \leftrightarrow \frac{d(w^T H w)}{dw} = 2w^T H = 2w^T X^T X$

Ta có :

$$L'_w = -2y^T X + [w^T X^T X w]'_w = -2y^T X + \frac{d(w^T X^T X w)}{dw} = -2y^T X + 2w^T X^T X = 0$$

- ⇨ $2w^T X^T X = 2y^T X$
- ⇨ $w^T X^T X = y^T X$
- ⇨ $(Xw)^T X = (X^T y)^T$
- ⇨ $[X^T(Xw)]^T = (X^T y)^T$
- ⇨ $X^T(Xw) = X^T y$
- ⇨ $(X^T X)w = X^T y$
- ⇨ $\boldsymbol{w} = (\boldsymbol{X^T X})^{-1} \boldsymbol{X^T y}$

Kết luận : Nếu $X^T X$ khả nghịch (có ma trận nghịch đảo) , thì L có nghiệm duy nhất : $\boldsymbol{w} = (\boldsymbol{X^T X})^{-1} \boldsymbol{X^T y}$

Nếu không khả nghịch, ta có thể sử dụng khái niệm giả nghịch đảo.

Như vậy ta đã có hàm Gradient descent $\frac{\mathbf{dJ}}{\mathbf{dw}} = \mathbf{X^T} * (\mathbf{\hat{y}} - \mathbf{y})$ và nghiệm $\boldsymbol{w} = (\boldsymbol{X^T X})^{-1}\boldsymbol{X^T y}$

- ⇨ Sử dụng công thức w = w – learning_rate * gradient sẽ thu được w tối ưu sau **epochs** lần lặp
- ⇨ Hoặc có thể cho nó dừng sớm nếu w_{old} và w_{new} chênh lệch không quá **epsilon** .

Code Linear Regression

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

def matrix_transformation(x, y) :
    x = np.concatenate((np.ones((x.shape[0], 1)), x), axis=1)
    y = np.array(y).reshape(-1, 1)
    return x, y

def linear_regression(X, y, lr=0.05, epsilon=1e-10, epochs=100000):
    ep = 0
    N = X.shape[0]
    w = np.linalg.pinv(X.T.dot(X)).dot(X.T).dot(y)
    while ep < epochs:
        gradient = np.dot(X.T, (np.dot(X, w) - y)) / N
        w = w - lr * gradient
        if np.linalg.norm(gradient) < epsilon:
            break
        ep += 1
    return w

def predict(X, w):
    predictions = X.dot(w)
    return predictions

data = pd.read_csv("USA_Housing.csv")
data.head()
X = data[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',
'Avg. Area Number of Bedrooms', 'Area Population']]
y = data['Price']
np.random.seed(42)
random_indices = np.random.permutation(len(X))
train_size = int(0.7 * len(X))

X_train = X.iloc[random_indices[:train_size]]
y_train = y[random_indices[:train_size]]
X_test = X.iloc[random_indices[train_size:]]
y_test = y[random_indices[train_size:]]

X_train, y_train = matrix_transformation(X_train, y_train)
```

```python
X_test, y_test = matrix_transformation(X_test, y_test)
w = linear_regression(X_train, y_train, lr=0.05, epsilon=1e-10, epochs=100000)
predictions = predict(X_test, w)

plt.figure(figsize=(10, 6))
plt.scatter(y_test, predictions)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], color='red')
plt.xlabel('Thực tế')
plt.ylabel('Dự đoán')
plt.title('So sánh giữa giá trị thực tế và giá trị dự đoán')
plt.show()
```