Video 7

Sunday, October 4, 2020 6:57 AM

Linear regression model

$$y = \underbrace{w}_{\text{(scalar) weight parameter}} \underbrace{x}_{\text{input (scalar)}} + \underbrace{b}_{\text{bias parameter (scalar)}}$$

y: prediction (scalar)

Vector input: $\vec{x} = (x_1, \cdots, x_D)$ — feature of the input

Python

```
y = b
for j in range(D):
    y += w[j] * x[j]
```

$\vec{x} = (x_1, \cdots, x_D)$ $\quad \vec{w} = (w_1, \cdots, w_D)$

$$y = \vec{w}^T \vec{x} + b$$

$\vec{w}$: weight (vector); $\vec{x}$: input (vector); b: bias (scalar); y: prediction (scalar)

2 vectors have same dimension

$$y = \vec{w}^T \vec{x} + b = (w_1, \cdots, w_D) \begin{pmatrix} x_1 \\ \vdots \\ x_D \end{pmatrix} + b$$

$$y = \sum_j w_j x_j + b$$

Python + NumPy: $\quad y = np.dot(w, x) + b$

np

Vectorize according to the i training examples:

$$X = \begin{pmatrix} x^{(1)T} \\ x^{(2)T} \\ x^{(3)T} \end{pmatrix} = \begin{pmatrix} 8 & 0 & 3 & 0 \\ 6 & -1 & 5 & 3 \\ 2 & 5 & -2 & 8 \end{pmatrix}$$

$x^{(2)}$: one training example

one feature across all training examples

$x^{(2)} = \begin{pmatrix} 6 \\ -1 \\ 5 \\ 3 \end{pmatrix}$ — 3rd feature of $x^{(2)}$

Dataset (collection of training examples) $\quad D = \{x^{(1)}, x^{(2)}, x^{(3)}\}$

$$X \vec{w} + b\,1 = \begin{pmatrix} w^T x^{(1)} + b \\ \vdots \\ w^T x^{(N)} + b \end{pmatrix} = \begin{pmatrix} y^{(1)} \\ \vdots \\ y^{(N)} \end{pmatrix} = \vec{y}$$

X: dataset matrix; $\vec{w}$: weight vector; b: scalar bias; 1: identity matrix

N: nb training examples

$y^{(1)}$: prediction on $x^{(1)}$

$\vec{y}$: predictions of our linear regression on each input of the dataset

$$X = \begin{pmatrix} x^{(1)T} \\ x^{(2)T} \\ x^{(3)T} \end{pmatrix} \quad \vec{w} = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{pmatrix} \quad b$$

each training example has $D = 4$ features; b: scalar

$$X \vec{w} + b\,1 = \begin{pmatrix} x^{(1)T} \\ x^{(2)T} \\ x^{(3)T} \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{pmatrix} + \begin{pmatrix} b \\ \vdots \\ b \end{pmatrix}$$

$N \times D$ $\quad D \times 1$ $\quad N \times 1$

$$= \begin{pmatrix} x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & x_4^{(1)} \\ & \vdots & & \\ x_1^{(3)} & x_2^{(3)} & x_3^{(3)} & x_4^{(3)} \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{pmatrix} + \begin{pmatrix} b \\ b \\ b \\ b \end{pmatrix}$$

$$= \begin{pmatrix} \left(\sum_{j=1}^{4} w_j x_j^{(1)}\right) + b \\ \left(\sum_{j=1}^{4} w_j x_j^{(2)}\right) + b \\ \left(\sum_{j=1}^{4} w_j x_j^{(3)}\right) + b \end{pmatrix}$$

$$= \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ y^{(3)} \end{pmatrix}$$

$y^{(2)}$ is prediction of linear model on $x^{(2)}$

$$\vec{y} = X\vec{w} + b\,1$$

$$\varepsilon = \frac{1}{2N} \left\| \vec{y} - \vec{t} \right\|^2$$

$\vec{t}$ (vector): $t^{(1)}, t^{(2)}, t^{(3)}$

predictions on entire dataset; labels for each training example

$D = \{(x^{(i)}, t^{(i)})\}_i$

Python + numPy:

```
y = np.dot(X, w) + b
cost = np.sum((y - t)**2) / (2. * N)
```

cost = loss = error of the model