Deep Learning

Dr. Mehran Safayani

safayani@iut.ac.ir

safayani.iut.ac.ir

https://www.aparat.com/mehran.safayani

https://github.com/safayani/deep_learning_course

Department of Electrical and computer engineering, Isfahan university of technology, Isfahan, Iran

Basics of Neural Network Programming

Vectorizing Logistic Regression's Gradient Computation

Vectorizing Logistic Regression

- *Z=0;*

  For i in range(n_x)

  $z += w[i] * x[i]$

  $z += b$

  روش اول

- *Z=0;*

  z =np.dot(w, x)+b

  روش دوم

  SIMD

  GPU

Vectorizing Logistic Regression

- $z^{(1)} = w^T x^{(1)} + b$ $\quad$ $z^{(2)} = w^T x^{(2)} + b$ $\quad$ $z^{(3)} = w^T x^{(3)} + b$
  $a^{(1)} = \sigma\left(z^{(1)}\right)$ $\quad$ $a^{(2)} = \sigma\left(z^{(2)}\right)$ $\quad$ $a^{(3)} = \sigma\left(z^{(3)}\right)$

- $\mathrm{X} = \begin{bmatrix} | & | & & | \\ x^{(1)} & x^{(2)} & \dots\dots\dots & x^{(\mathrm{m})} \\ | & | & & | \end{bmatrix} \in R^{n_x \times m}$ $\quad$ $\underbrace{[w_1 \dots w_{n_x}]}_{W^T} \begin{bmatrix} | & | & & | \\ x^{(1)} & x^{(2)} & \dots\dots\dots & x^{(\mathrm{m})} \\ | & | & & | \end{bmatrix}$

- $Z = \left[\underbrace{z^{(1)}}_{w^T x^{(1)} + b} \; \underbrace{z^{(2)}}_{w^T x^{(2)} + b} \; \underbrace{z^{(3)}}_{w^T x^{(3)} + b} \; \dots \; \underbrace{z^{(m)}}_{w^T x^{(m)} + b} \right] = W^T X + [b \; b \cdots b]_{1 \times m}$

- $Z = \underbrace{np \cdot dot(W \cdot T, X)}_{1_\times m} + \underbrace{b}_{(1,1)}$ *"broadcasting"* $[b, b \dots, b]_{1 \times m}$

Vectorizing Logistic Regression

- $A = \left[a^{(1)}, a^{(2)}, \ldots, a^{(m)}\right] = \sigma(\underbrace{Z}_{1\times m})$
- $dz^{(1)} = a^{(1)} - y^{(1)}$ $\quad dz^{(2)} = a^{(2)} - y^{(2)}$
- $dZ = \left[dz^{(1)}\ dz^{(2)} \ldots\ dz^{(m)}\right]_{1 \times m}$
- $A = \left[a^{(1)}\ a^{(2)} \ldots\ a^{(m)}\right]$ $\quad Y = \left[y^{(1)}\ y^{(2)} \ldots\ y^{(m)}\right]$
- $dZ = A - Y = \left[a^{(1)} - y^{(1)}\quad a^{(2)} - y^{(2)}\ \ldots\ a^{(m)} - y^{(m)}\right]$

- $dw=0$ $\qquad db=0$

- $\begin{bmatrix} dw_1 \\ dw_2 \\ dw_3 \\ \vdots \\ dw_{nx} \end{bmatrix}_{nx \times 1}$

$$\begin{cases} dw{+}{=}\ \underbrace{x^1}_{\text{بردار}}\ \overbrace{dz^1}^{\overset{\text{عدد}}{a^1-y^1}} \\ dw{+}{=}x^2dz^2 \\ dw{+}{=}xmdzm \end{cases}$$

$dw/{=}m$

$$\begin{cases} db{+}{=}\ \overbrace{dz1}^{\overset{\text{عدد}}{a1-y1}} \\ db{+}{=}dz2 \\ db{+}{=}dzm \end{cases}$$

$db/{=}m$

$$db = \frac{1}{m}\sum_{i=1}^{m} dz^{(i)} = \frac{1}{m} np.sum(dZ$$

$$dz = \left[dz^1\ \ dz^2 \ldots dz^m\right]_{1 \times m}$$

$$\boxed{dW = \frac{1}{m} X\, dZ^T} = \frac{1}{m}\begin{bmatrix} \\ x^{(1)} x^{(2)} \quad \ldots\ldots\ldots\ x^{(\mathrm{m})} \\ \\ \end{bmatrix}_{nx \times m} \begin{bmatrix} dz^1 \\ dz^2 \\ dz^3 \\ \vdots \\ \vdots \\ \vdots \\ dz^m \end{bmatrix}_{m \times 1}$$

$$dw = \frac{1}{m}\left[x^{(1)}dz^1 + x^{(2)}dz^2 + \cdots + x^{(m)}dz^m\right]$$

Vectorizing Logistic Regression

- $w_1, w_2, b \leftarrow random$

For iter in range(1000)

$$\textbf{1 epoch}\left\{\begin{aligned}
&Z = W^T x + b = np \cdot dot(W.T, X) + b \\
&A = \sigma(Z) \\
&dZ = A - Y \\
&dW = \frac{1}{m} X\, dZ^T \\
&db = \frac{1}{m} np.sum(dZ) \\
&w = w - \alpha\, dw \\
&b = b - \alpha\, db
\end{aligned}\right.$$

$$a = np \cdot random \cdot \mathrm{rand}n(5,1) \qquad a.\mathrm{sh}ape = (5,1)$$

Logistic Regression Cost function

- $\hat{y} = \sigma(w^T x + b) \qquad 0 < \sigma(z) = \frac{1}{1+e^{-z}} < 1$
- $\hat{y} = p(y = 1|x)$

$$\left.\begin{array}{l} \bullet\ if \quad y = 1 \ : \quad p(y|x) = \hat{y} \\ \bullet\ if \quad y = 0 \ : \quad p(y|x) = 1 - \hat{y} \end{array}\right\} \boldsymbol{p(y|x)}$$

- $\boxed{p(y|x) = \hat{y}^y + (1 - \hat{y})^{1-y}}$ $\quad distribution?\ Bernoulli$
- $if \quad y = 1 \ : \quad p(y|x) = \hat{y}$
- $if \quad y = 0 \ : \quad p(y|x) = 1 - \hat{y}$
- $\mathrm{logp(y|x)} = \log[\hat{y}^y + (1 - \hat{y})^{1-y}] = y \log \hat{y} + (1 - y) \log(1 - \hat{y})$

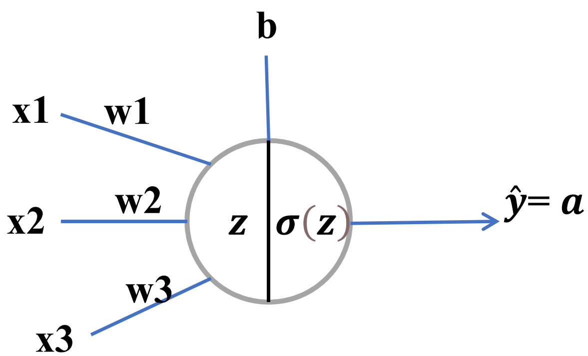
$$= -L(\hat{y}, y) \quad Max\ Likelihood$$

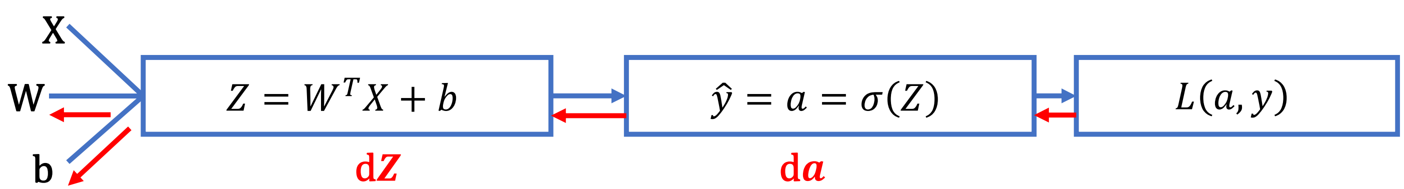
Logistic Regression Cost function

- $\log P(\text{ labels in trainingset}) = \log \prod_{i=1}^{m} P\left(y^{i} \mid x^{i}\right)$
- $\log P(\cdots) = \sum_{i=1}^{m} \log P\left(y^{(i)} \mid x^{(i)}\right) = -\sum_{i=1}^{m} L\left(\hat{y}^{(i)}, y^{(i)}\right)$

- Cost function $\underbrace{J(w,b)}_{\boldsymbol{minimize}} = \frac{1}{m} \sum_{i=1}^{m} L\left(\hat{y}^{(i)}, y^{(i)}\right)$

Neural Networks





Neural Networks