Basics of Neural Network Programming

Vectorization

What is vectorization?

$$\vec{w} \in \mathbb{R}^{n_x}$$

$$Z = \underline{\vec{w}^T x} + b \qquad \qquad \vec{w} = \begin{bmatrix} \vdots \\ \vdots \end{bmatrix} \qquad x = \begin{bmatrix} \vdots \\ \vdots \end{bmatrix} \qquad x \in \mathbb{R}^{n_x}$$

Non-vectorized:

$$Z = 0$$
$$\text{for } i \text{ in range}(n-x):$$
$$\qquad Z += \vec{w}[i] * x[i]$$

$$Z += b$$

Vectorized

$$Z = np.\underline{dot}(\vec{w}, x) + b$$
$$\underbrace{\qquad \qquad \qquad}_{\vec{w}^T x}$$

$$\rightarrow GPU \Big\} \quad SIMD - \text{single instruction}$$
$$\rightarrow CPU \Big\} \quad \text{multiple data}$$

Andrew Ng

Basics of Neural Network Programming

More vectorization examples

deeplearning.ai

Neural network programming guideline

Whenever possible, avoid explicit for-loops.

Neural network programming guideline

Whenever possible, avoid explicit for-loops.

$$u = Av$$

$$u_i = \sum_i \sum_j A_{ij} v_j$$

$$u = np.zeros((n,1))$$
for i ...
    for j ...
        $u[i] += A[i][j] * v[j]$

$$u = np.dot(A, v)$$

Vectors and matrix valued functions

Say you need to apply the exponential operation on every element of a matrix/vector.

$$v = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix} \rightarrow u = \begin{bmatrix} e^{v_1} \\ e^{v_2} \\ \vdots \\ e^{v_n} \end{bmatrix}$$

```
u = np.zeros((n,1))
for i in range(n):
    u[i]=math.exp(v[i])
```

import numpy as np

$u = np.exp(v)$

$np.log(v)$

$np.abs(v)$

$np.maximum(v,0)$

$v**2$

$1/v$

Andrew Ng

Logistic regression derivatives

$J = 0,$ ~~$dw1 = 0, \quad dw2 = 0$~~, $db = 0$

$dw = np.zeros((n_x, 1))$

$\rightarrow$ for i = 1 to n:

$z^{(i)} = w^T x^{(i)} + b$

$a^{(i)} = \sigma(z^{(i)})$

$J \mathrel{+}= -\left[y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})\right]$

$dz^{(i)} = a^{(i)}(1 - a^{(i)})$

for j=1..n_x
$dw_j \mathrel{+}=$

~~$dw_1 \mathrel{+}= x_1^{(i)} dz^{(i)}$~~ $n_x = 2$

~~$dw_2 \mathrel{+}= x_2^{(i)} dz^{(i)}$~~

$dw \mathrel{+}= x^{(i)} dz^{(i)}$

$db \mathrel{+}= dz^{(i)}$

$J = J/m,$ ~~$dw_1 = dw_1/m, \quad dw_2 = dw_2/m$~~, $db = db/m$

$dw \mathrel{/}= m$

Andrew Ng