Basics of Neural Network Programming

Vectorization

deeplearning.ai

What is vectorization?

$z = \underbrace{w^T x} + b$

$w = \begin{bmatrix} \vdots \\ \\ \vdots \end{bmatrix}$ $x = \begin{bmatrix} \vdots \\ \\ \vdots \end{bmatrix}$

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

$x \in \mathbb{R}^{n_x}$

Non-vectorized:

```
z = 0
for i in range(n-x):
    z += w[i] * x[i]

z += b
```

Vectorized

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

→ GPU
→ CPU } SIMD - single instruction multiple data.

deeplearning.ai

Basics of Neural Network Programming

More vectorization examples

Neural network programming guideline

Whenever possible, avoid explicit for-loops.

Neural network programming guideline

Whenever possible, avoid explicit for-loops.

$u = Av$

$u_i = \sum\limits_{\substack{i \\ \uparrow}} \sum\limits_{\substack{j \\ \uparrow}} A_{ij} v_j$

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

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

Andrew Ng

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, dw2 = 0~~, db = 0
for i = 1 to n:
    $z^{(i)} = w^T x^{(i)} + b$
    $a^{(i)} = \sigma(z^{(i)})$
    $J \mathrel{+}= -[y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})]$
    $dz^{(i)} = a^{(i)}(1 - a^{(i)})$
    ~~$dw_1 \mathrel{+}= x_1^{(i)} dz^{(i)}$~~
    ~~$dw_2 \mathrel{+}= x_2^{(i)} dz^{(i)}$~~
    $db \mathrel{+}= dz^{(i)}$
J = J/m, ~~$dw_1 = dw_1/m$, $dw_2 = dw_2/m$~~, db = db/m
```

$dw = np.zeros((n\_x, 1))$

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

$n_x = 2$

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

$dw \mathrel{/}= m.$

deeplearning.ai

Basics of Neural Network Programming

Broadcasting in Python

Broadcasting example

Calories from Carbs, Proteins, Fats in 100g of different foods:

$$\begin{array}{ccccc} & \text{Apples} & \text{Beef} & \text{Eggs} & \text{Potatoes} \\ \text{Carb} & \boxed{56.0} & 0.0 & 4.4 & 68.0 \\ \text{Protein} & 1.2 & 104.0 & 52.0 & 8.0 \\ \text{Fat} & 1.8 & 135.0 & 99.0 & 0.9 \end{array} = \underset{(3,4)}{A}$$

59cal

$\frac{56}{59} \approx 94.9\%$

Calculate % of calories from Carb, Protein, Fat. Can you do this without explicit for-loop?

```
cal = A.sum(axis = 0)
percentage = 100*A/(cal~~.reshape(1,4)~~)
```

(3,4) / (1,4)

Andrew Ng

Broadcasting example

$$\begin{bmatrix}1\\2\\3\\4\end{bmatrix} + \begin{bmatrix}100\\100\\100\\100\end{bmatrix} \cancel{100}$$

$$\begin{bmatrix}1&2&3\\4&5&6\end{bmatrix} +^{\swarrow} \begin{bmatrix}100&200&300\\100&200&300\end{bmatrix} \qquad \downarrow \quad \downarrow \quad \downarrow$$

(m,n) $(2,3)$ $(1,n) \rightsquigarrow (m,n)$ $(2,3)$

$$\begin{bmatrix}1&2&3\\4&5&6\end{bmatrix} + \begin{bmatrix}100&100&100\\200&200&200\end{bmatrix} = \qquad \begin{matrix}\leftarrow\\\leftarrow\end{matrix}$$

(m,n)

$$(m,1) \xrightarrow{} (m,n)$$

Andrew Ng

General Principle

$$\begin{array}{cccc}
(m,n) & + & (1,n) & \rightsquigarrow (m,n) \\
\underline{\text{matrix}} & - & & \\
 & * & & \\
 & / & (m,1) & \rightsquigarrow (m,n)
\end{array}$$

$$(m,1) \quad + \quad \mathbb{R}$$

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + 100 = \begin{bmatrix} 101 \\ 102 \\ 103 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix} + 100 = \begin{bmatrix} 101 & 102 & 103 \end{bmatrix}$$

Matlab/Octave: <u>bsxfun</u>

deeplearning.ai

Basics of Neural Network Programming

A note on python/ numpy vectors

Python / numpy vectors

```
import numpy as np

a = np.random.randn(5)

a = np.random.randn((5,1))

a = np.random.randn((1,5))

assert(a.shape = (5,1))
```