



deeplearning.ai

Basics of Neural Network

Programming
Vectorization

What is vectorization?

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

$$w = \begin{bmatrix} \vdots \\ \vdots \\ \vdots \end{bmatrix} \quad x = \begin{bmatrix} \vdots \\ \vdots \\ \vdots \end{bmatrix} \quad \begin{matrix} w \in \mathbb{R}^{n_x} \\ x \in \mathbb{R}^{n_x} \end{matrix}$$

Non-vectorized:

$$z = 0$$

for i in $\text{range}(n-x)$:

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

$$z += b$$

Vectorized

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

\Rightarrow GPU } SIMD - single instruction
 \Rightarrow CPU } 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.

$$u = Av$$

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

$$u = \text{np.zeros}(n, 1)$$

for i ...

for j ...

$$u[i] += A[i][j] * v[j]$$

$$u = \text{np.dot}(A, v)$$

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
```

Logistic regression derivatives

$J = 0, \text{ ~~dw1 = 0, dw2 = 0~~, db = 0$

$dw = np.zeros((n-x, 1))$

→ for i = 1 to n:

=

=

+=

d =

~~d +=~~

~~d +=~~

db += d

$n_x = 2$

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

$J = J/m, \text{ ~~d = d/m, d = d/m, db = db/m~~$

$dw /= m.$