AUTHOR: SYEDA DARAQSHAN

WEEK 2

NEURAL NETWORKS AND DEEP LEARNING

VECTORIZATION:

In order to implement a neural network at a reasonable speed, one must be careful when using for loops. In order to compute the hidden unit activations in the first layer, we must compute z1, ..., z4 and a1, ..., a4.

$z^{[1]}_1 = W^{[1]T}_1 x + b^{[1]}_1$ and $a^{[1]}_1 = g(z^{[1]}_1)$

. . .

. . .

. . .

$z^{[1]}_4 = W^{[1]T}_4 x + b^{[1]}_4$ and $a^{[1]}_4 = g(z^{[1]}_4)$

Instead of using for loops, vectorization takes advantage of matrix algebra and highly optimized numerical linear algebra packages (e.g., BLAS) to make neural network computations run quickly. Before the deep learning era, a for loop may have been sufficient on smaller datasets, but modern deep networks and state-of-the-art datasets will be infeasible to run with for loops.

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

Non-Vectorized :-

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

Vectorized

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

BROADCASTING:

Broadcasting is a mechanism which allows tensors with different numbers of dimensions to be added or multiplied together by (virtually) replicating the smaller tensor along the dimensions that it is lacking.

Broadcasting

$$\begin{bmatrix}1\\2\\3\\4\end{bmatrix} + 100 \Rightarrow \begin{bmatrix}1\\2\\3\\4\end{bmatrix} + \begin{bmatrix}100\\100\\100\\100\end{bmatrix} = \begin{bmatrix}101\\102\\103\\104\end{bmatrix}$$

$$\underset{\text{matrix}}{(m,n)} \quad \begin{matrix}+\\-\\ *\\ /\end{matrix} \quad \begin{matrix}(1,n) \rightarrow (m,n)\\ \text{(or)}\\ (m,1) \rightarrow (m,n)\end{matrix}$$

$$\begin{bmatrix}1 & 2 & 3\\4 & 5 & 6\end{bmatrix} + \begin{bmatrix}10 & 20\end{bmatrix} \Rightarrow \begin{bmatrix}10 & 20\\10 & 20\end{bmatrix}$$