Notations

This document contains the notations used throughout the project.

General Notations

- A lowercase letter represents a scalar. eg. a
- A lowercase bold letter represents a vector. eg. $\mathbf{a}$
- An uppercase letter represents a matrix. eg. A

Numpy Implementation Specific Notations

> Numpy use row first notation, this means that the first index of a matrix is the row index and the second index is the column index. For example, $A_{i,j}$ is the element at row i and column j of matrix A.
>
> This also means that a row vector is a matrix with shape $(1, n)$ and a column vector is a matrix with shape $(n, 1)$. See the notebook Preliminaries.ipynb for more details.

- A vector $\mathbf{a}$ has a shape of $(n,)$ or $(n, 1)$ depending on situtaion (I'll prefer the first one if not needed otherwise.). I'm not going to use $(1, n)$ as the shape of a vector.
- A matrix A has a shape of (m, n), where m is the number of rows and n is the number of columns.
- In numpy, counting starts from 0 however, in the following notations, I'll use 1-based indexing.

Indexing

- I'll use a_i to represent the ith element of vector $\mathbf{a}$. For example, a_1 is the first element of vector $\mathbf{a}$.
- I'll use $a_{i,j}$ to represent the element at row i and column j of matrix A. For example, $a_{1,2}$ is the element at row 1 and column 2 of matrix A.
- $\mathbf{a^{[i]}}$ is the ith column of matrix A.
- $\mathbf{a_j}$ is the jth row of matrix A.

Neural Network Specific Notations

> The notation is borrowed from the course Neural Networks and Deep Learning by Andrew Ng with some modifications.

General

Super script $[l]$ represents the lth layer while superscript (i) represents the ith training example.

Sizes

- m is the number of training examples.
- n_x is the number of features. (input size)
- n_y is the number of classes. (output size)
- $n^{[l]}$ is the number of neurons in layer l.
- L is the number of layers in the network. (Excluding input layer)

Objects

- $X \in \mathbb{R}^{n_x \times m}$ is the input matrix, where each column is a training example. So, X is a matrix with shape (n_x, m).
- $x^{(i)} \in \mathbb{R}^{n_x}$ is the i^{th} training example. So, $x^{(i)}$ is a column vector with shape $(n_x, 1)$.
- $Y \in \mathbb{R}^{n_y \times m}$ is the output matrix, where each column is a training example. So, Y is a matrix with shape (n_y, m).
- $\mathbf{y^{(i)}} \in \mathbb{R}^{n_y}$ is the output label for i^{th} example.
- $W^{[l]} \in \mathbb{R}^{n^{[l]} \times n^{[l-1]}}$ is the weight matrix of layer l. This means that $W^{[l]}$ is a matrix with shape $(n^{[l]}, n^{[l-1]})$.
- $b^{[l]} \in \mathbb{R}^{n^{[l]}}$ is the bias vector of layer l. This means that $b^{[l]}$ is a column vector with shape $(n^{[l]},)$.
- $\hat{y} \in \mathbb{R}^{m}$ is the predicted output label. This is an exception where I'll use lowercase, normal font for a vector.
- $\hat{Y} \in \mathbb{R}^{n_y \times m}$ is the predicted output matrix. This is the one hot encoded version of $\hat{y}$.

Forward Propagation and Activation Functions

- $Z^{[l]} \in \mathbb{R}^{n^{[l]} \times m}$ is the linear output of layer l. This means that $Z^{[l]}$ is a matrix with shape $(n^{[l]}, m)$.
- $A^{[l]} \in \mathbb{R}^{n^{[l]} \times m}$ is the activation output of layer l. Its shape is the same as $Z^{[l]}$.
- $g^{[l]}$ is the activation function of layer l.
- $\mathbf{a^{[l](i)}} \in \mathbb{R}^{n^{[l]}}$ is the i^{th} training example's output of layer l. This means that $\mathbf{a^{[l](i)}}$ is a column vector with shape $(n^{[l]}, 1)$.
- $\mathbf{a_i^{[l]}}$ is the output of the i^{th} neuron of layer l.

Backward Propagation

- $\mathcal{J}(X, W, \mathbf{b}, \mathbf{y}) \in R^1$ or $\mathcal{J}(\hat{y}, \mathbf{y}) \in R^1$ is the cost function. This is another exception where I've use uppercase letter to denote a scalar.
- $dW^{[l]}$ is the partial derivative of $\mathcal{J}$ with respect to W, $\frac{\partial \mathcal{J}}{\partial W^{[1]}}$.
- $db^{[l]}$ is the partial derivative of $\mathcal{J}$ with respect to b, $\frac{\partial \mathcal{J}}{\partial b}$.