Assignment1

Name: Nezar Ahmed Mokhtar Ali ---- **Email**: nezaro501@gmail.com ---- **ID**: 201920580

Week1:

- A housing price prediction was given as a gate to define the basic building block of neural networks which is the **neuron**. A neuron can take one or more features and represents them in as a higher feature throughout the layers of the neural network. **As we go deeper in network, we are representing higher features using the lower ones** (Family members estimate from lower features size of house and number of bedrooms).
- **Supervised learning** means that outputs are labeled so we know the ground truth values to train on.
- Neural networks has several types but mainly are **standard neural networks (ANN), convolutional neural networks (CNN) and recurrent neural networks (RNN)**. There are more complex and mixed neural networks but those are the main ones.
- Data is mainly categorized into 2 categories which are **structured data** (tabulated data) and **unstructured data** (images, voice, text, ..etc).
- Deep learning progress was mainly led by **the scale of the data and the computation** as recently we started the **big data era** where large amount of datasets are released and the computation power had a great hype which allowed more complex algorithms to be able to learn in a more efficient and faster way.

Week2:

- **Binary classification** means that the output layer has 2 valid outputs either positive or negative like cats classification which outputs either 1 (means a cat) or 0 (means not a cat).
- RGB Images are represented as an $m\times n\times 3$ matrix where m represents height and n represents width while the 3 is the number of channels which in our case are red, green and blue channels.
- Notations in binary classification are as follows:

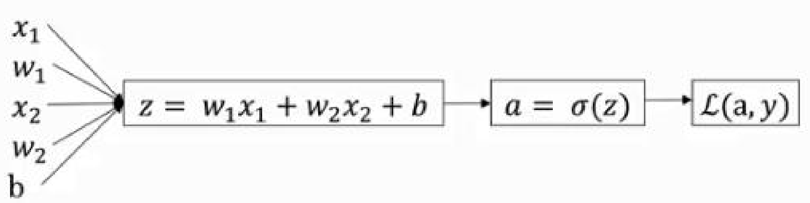
Single example is expressed as (x,y) where $x \in R^{n_x}$, $y \in \{0,1\}$, hence for m training example : $(x^{(1)},y^{(1)})$, $(x^{(2)},y^{(2)})$, ……, $(x^{(m)},y^{(m)})$.
where ($m \equiv m_{train}$= number of training examples)

$$X=\begin{bmatrix} | & | & & | \\ x^{(1)} & x^{(2)} & \cdots & x^{(m)} \\ | & | & & | \end{bmatrix}_{n_x \times m} \quad , \quad Y=[y^{(1)} \quad y^{(2)} \quad \cdots \quad y^{(m)}]_{1\times m}$$

- **Logistic regression** is mainly an **activation function** used in the output layer for binary classification.

Given x where $x\in R^{n_x}$, want $\hat{y} = P(y=1 \mid x)$
- <u>Logistic regression</u>: $0 \le h_\theta(x) \le 1$
- <u>Hypothesis</u>: $g(z) = \frac{1}{1+e^{-z}}$ so $h_\theta(x)= \frac{1}{1+e^{-(\theta^T.x)}}$ [Remember: $h_\theta(x)= \theta_0x_0 + \theta_1 x_1 \ldots+ \theta_n x_n = \theta^T x$]
- <u>Decision boundary:</u> if $h_\theta(x) \ge 0.5$ then $y_{predicted}=1$, otherwise, $y_{predicted}=0$ noting that $h_\theta(x) \ge 0.5$ when $(\theta^T.x) \ge 0$. ($h_\theta(x) =0.5$ ➔ Decision boundary equation)
- <u>Cost function</u>: $J(\theta) = \frac{1}{m}\sum_{i=1}^{m} -y\log(h_\theta(x)) - (1-y)\log(1-h_\theta(x))$
- <u>Gradient descent:</u> $\theta := \theta - \alpha \frac{\partial}{\partial\theta}(J(\theta)) = \theta - \alpha \frac{\partial}{\partial\theta}(\frac{1}{2m}\sum_{i=1}^{m} (\hat{y}^2 - y(i))2) = \theta - \alpha \frac{\partial}{\partial\theta}(\frac{1}{m}\sum_{i=1}^{m} (h\theta(x(i)) - y(i))$. xj(i)) (Applied iteratively)
- **Computation graph** is a combination of feed-forward path followed by backpropagation in an iterative manner, so computation graph discusses why neural networks is organized this way.
- <u>Equations of neuron's computation logic</u>:

$$z = w^T x + b$$
$$\hat{y} = a = \sigma(z)$$
$$\mathcal{L}(a,y) = -(y\log(a) + (1-y)\log(1-a))$$



- **Vectorization** is an idea to compute values all at once to get rid of loops. Vectorization uses numpy function from python or any other built in function used for vectorization in other programming languages. The unvectorized version (using for loops) took 300 as much time that took by the vectorized one in the given example.