Respecting the coursera policies, this work will only be an explainer for the Quiz questions. No answers will be given out. But any sincere learner would be able to pick the right answer after the learning.

---

1. **What does a neuron compute?**
    a. A neuron is nothing but a function. It takes in a summation of inputs, multiplied by weights; adds a bias term to the summation; and finally transforms the linear function to a nonlinear function. This process of adding 'non-linearity' to the model is called 'activation'. After computing this, the neuron transmits its output to next set of neurons
2. **Which of these is the "Logistic Loss"?**
    a. Logistic Loss is the loss function that we use in Logistic Regression. It is based on negative maximum likelihood.
    b. $$J(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N} H(p_n, q_n) = -\frac{1}{N}\sum_{n=1}^{N}\left[y_n \log \hat{y}_n + (1 - y_n)\log(1 - \hat{y}_n)\right],$$
3. **Suppose img is a (32,32,3) array, representing a 32x32 image with 3 color channels red, green and blue. How do you reshape this into a column vector?**
    a. You have to flatten any image if you have to use it in Deep learning applications. img.reshape(32*32*3,1)
4. **Consider the two following random arrays "a" and "b":**
   a = np.random.randn(2, 3) # a.shape = (2, 3)
   b = np.random.randn(2, 1) # b.shape = (2, 1)
   c = a + b
   What will be the shape of "c"?
    a. This question wants to test our understanding of 'broadcasting' in Python. Basically, b array gets broadcasted (or) copied three additional times to make the matrix addition possible. Thus c will be (2,3)
5. **Consider the two following random arrays "a" and "b":**
   a = np.random.randn(4, 3) # a.shape = (4, 3)
   b = np.random.randn(3, 2) # b.shape = (3, 2)
   c = a * b
   What will be the shape of "c"?
    a. For broadcasting to happen, one of the matrices should be a row/column matrix, with equal number of columns/rows as in the other matrix.
6. **Suppose you have n_x input features per example. Recall that X=[x^(1), x^(2)...x^(m)]. What is the dimension of X?**
    a. Rows (Number of features) * Columns (Number of training samples)

7. **Consider the two following random arrays "a" and "b":**

```
a = np.random.randn(12288, 150) # a.shape = (12288, 150)
b = np.random.randn(150, 45) # b.shape = (150, 45)
c = np.dot(a, b)
```

   What is the shape of c?

   a. Just note that np.dot(a,b) is different from a*b. np.dot() is a normal dot product between matrices and you can find the size accordingly. When a is a (m,n) matrix and b is (n,o) matrix; dot product and a and b is a (m,o) matrix.

8.

```
# a.shape = (3,4)
# b.shape = (4,1)
for i in range(3):
  for j in range(4):
    c[i][j] = a[i][j] + b[j]
```

   Vectorise this:

   a. `c = a + b.T`

9. **Consider the two following random arrays "a" and "b":**

   a = np.random.randn(3, 3) # a.shape = (3, 3)

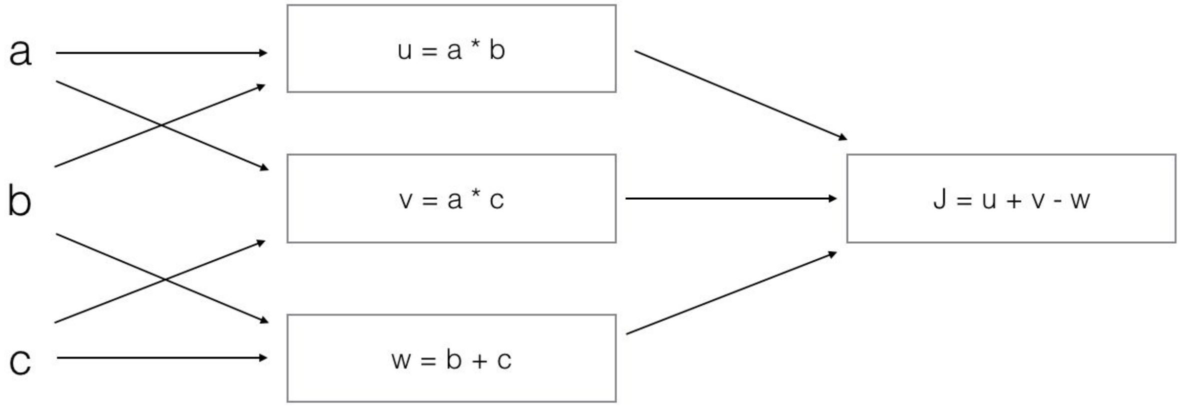
   b = np.random.randn(3, 1) # b.shape = (3, 1)

   c = a * b

   What will be the shape of "c"?

   a. Element wise multiplication. This involves broadcasting again.

10.



   What is the output J?

   a. This is algebra, not deep learning.