

# Forward Propagation in the Neural Networks

---

## Training Code

---

In [0]:

```
import matplotlib.pyplot as plt
import numpy as np
import warnings

warnings.filterwarnings("ignore")
file_data = "/content/drive/My Drive/Colab Notebooks/data08/mnist_test.csv"
handle_file = open(file_data, "r")
data = handle_file.readlines()
handle_file.close()

size_row = 28 # height of the image
size_col = 28 # width of the image

num_image = len(data)
count = 0 # count for the number of images

#
# make a matrix each column of which represents an images in a vector form
#
list_image = np.empty((size_row * size_col, num_image), dtype=float)
list_label = np.empty(num_image, dtype=int)

for line in data:

    line_data = line.split(',')
    label = line_data[0]
    im_vector = np.asfarray(line_data[1:])

    list_label[count] = label
    list_image[:, count] = im_vector
    count += 1

# make average vector image of each label for submission 1
list_avg = np.zeros((size_row * size_col, 10), dtype=float)
for i in range(10):
    list_avg[:, i] = np.average(list_image[:, list_label==i], axis=1)

# thetas for logistic unit
theta_list = np.random.randn(size_row * size_col)

#  $f(x) = x_1 * \theta_1 + \dots + x_m * \theta_m$ 
def func_calc(theta_list, image_current):
    func_val = 0
    for i in range(size_row * size_col):
        func_val += theta_list[i] * image_current[i]
    return func_val

# sigmoid calculation
def sigmoid(val):
    return 1/(1+np.exp(-val))
```

```
# compute the forward propagation and take the average
# of the output values for the images of the same label
sigmoid_list = np.zeros(num_image, dtype=float)
sigmoid_avg_list = np.zeros(10, dtype=float)
for i in range(num_image):
    sigmoid_list[i] = sigmoid(func_calc(theta_list, list_image[:, i]))
for i in range(10):
    sigmoid_avg_list[i] = np.average(sigmoid_list[list_label==i])
```

## Submission

### 1. Plot the average image

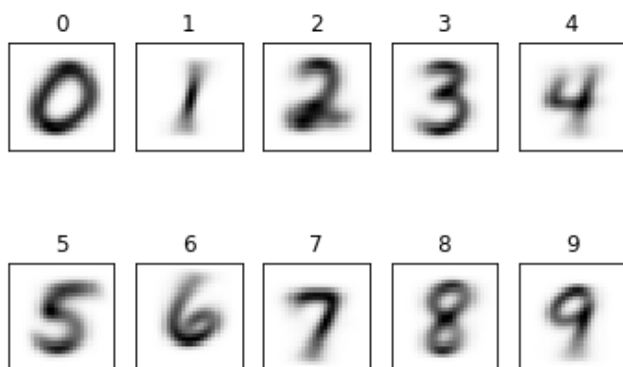
In [89]:

```
for i in range(10):
    label = i
    im_vector = list_avg[:, i]
    im_matrix = im_vector.reshape((size_row, size_col))

    plt.subplot(2, 5, i+1)
    plt.title(label)
    plt.imshow(im_matrix, cmap='Greys', interpolation='None')

    frame = plt.gca()
    frame.axes.get_xaxis().set_visible(False)
    frame.axes.get_yaxis().set_visible(False)

plt.show()
```



### 2. Present the output of the neural network with random weights

In [90]:

```
for i in range(10):
    print(i, ":", sigmoid_avg_list[i])
```

```
0 : 0.37238112677162716
1 : 0.6018668265937247
2 : 0.0920542634682025
3 : 0.3195485248735873
4 : 0.12412466610770143
5 : 0.4020734026392914
6 : 0.13780395765212683
7 : 0.07882117303237299
8 : 0.21977442180696927
9 : 0.16346287236255772
```

