assignment_03_2019111055

March 28, 2019

1 Pattern Recognition

1.1 Assignment 3

Name: Wangduk Seo Student ID: 2019111055

Learning Framework: Tensorflow

2 Heat Equation Code for Neural Network

```
In [1]: def expand_img(src_mat):
            #Expand Images (Normann-padding)
            # Cppy the boundary of image and paste to outter boundary
            img_mat = np.zeros((src_mat.shape[0] + 2, src_mat.shape[1] + 2), dtype=np.uint8)
            img_mat[1:-1, 1:-1] = src_mat
            img_mat[1:-1, 0] = src_mat[0:, 0]
            img_mat[1:-1, img_mat.shape[1] - 1] = src_mat[0:, src_mat.shape[1] - 1]
            img_mat[0, 1:-1] = src_mat[0, 0:]
            img_mat[img_mat.shape[0] - 1, 1:-1] = src_mat[src_mat.shape[0] -1, 0:]
            return img_mat
In [2]: def init_heatconv():
            #initialize filter as tensor
            heat\_window = tf.Variable([[0.0, 1.0, 0.0], [1.0, -4.0, 1.0], [0.0, 1.0, 0.0]])
            heat_window = tf.reshape(heat_window, shape=[3, 3, 1, 1])
            return heat_window
In [3]: def model(UO, ALPHA, DELTA_T):
            delta_t = DELTA_T
            alpha = ALPHA
            #normann-padding
            u_cur = expand_img(U0)
            w_constant = delta_t * alpha
            u_tensor = tf.cast(tf.Variable(u_cur), tf.float64)
            u_tensor = tf.expand_dims(u_tensor, 0)
            u_tensor = tf.expand_dims(u_tensor, 3)
```

2.0.1 Constant Time

```
In [5]: from PIL import Image
        import numpy as np
        import tensorflow as tf
        src_img = Image.open('C:/Users/CalPC_1/Pictures/lena_gray.gif').convert('L')
        src_mat = np.array(src_img, 'uint8')
        alpha = 0.25
        # Time Constant
        delta = np.ones(128) * 0.25
        img_list = model(src_mat, alpha, delta)
        with tf.Session() as sess:
            tf.global_variables_initializer().run()
            res = sess.run(img_list)
        img_idx = [0, 1, 3, 7, 15, 31, 63, 127]
        #Saving Image
        for i in img_idx:
            img_mat = res[i][0, 1:513, 1:513, 0]
            im = Image.fromarray(np.uint8(img_mat))
            im.save(str(i) + 'th iterations_const.jpg')
```

2.0.2 Time varying

```
In [6]: from PIL import Image
        import numpy as np
        import tensorflow as tf
        src_img = Image.open('C:/Users/CalPC_1/Pictures/lena_gray.gif').convert('L')
        src_mat = np.array(src_img, 'uint8')
        alpha = 0.25
        # varying time
        delta = np.arange(128) * 0.01
        img_list = model(src_mat, alpha, delta)
        with tf.Session() as sess:
            tf.global_variables_initializer().run()
            res = sess.run(img_list)
        img_idx = [0, 1, 3, 7, 15, 31, 63, 127]
        #Saving Image
        for i in img_idx:
            img_mat = res[i][0, 1:513, 1:513, 0]
            im = Image.fromarray(np.uint8(img_mat))
            im.save(str(i) + 'th iterations_varying.jpg')
```

3 Result

3.1 Time Constant Image Result

3.1.1 Iteration 1



3.1.2 Iteration 2



3.1.3 Iteration 4



3.1.4 Iteration 8



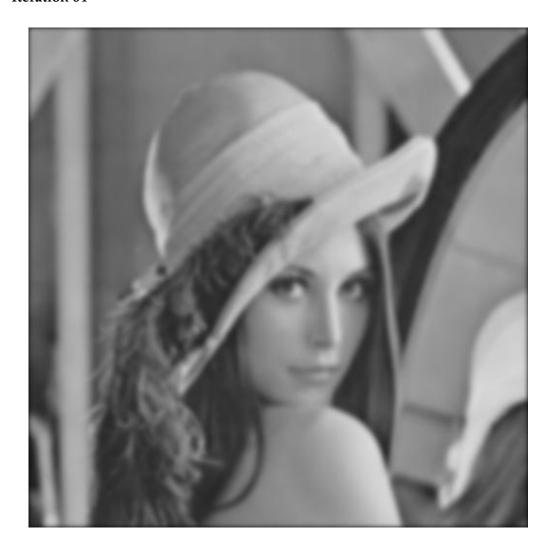
3.1.5 Iteration 16



3.1.6 Iteration 32



3.1.7 Iteration 64



3.1.8 Iteration 128



3.2 Time Varying Image Result ([1, 2, 3 ... ,128] * 0.01)

3.2.1 Iteration 1



3.2.2 Iteration 2



3.2.3 Iteration 4



3.2.4 Iteration 8



3.2.5 Iteration 16



3.2.6 Iteration 32



3.2.7 Iteration 64



3.2.8 Iteration 128

