Denoising Autoencoder

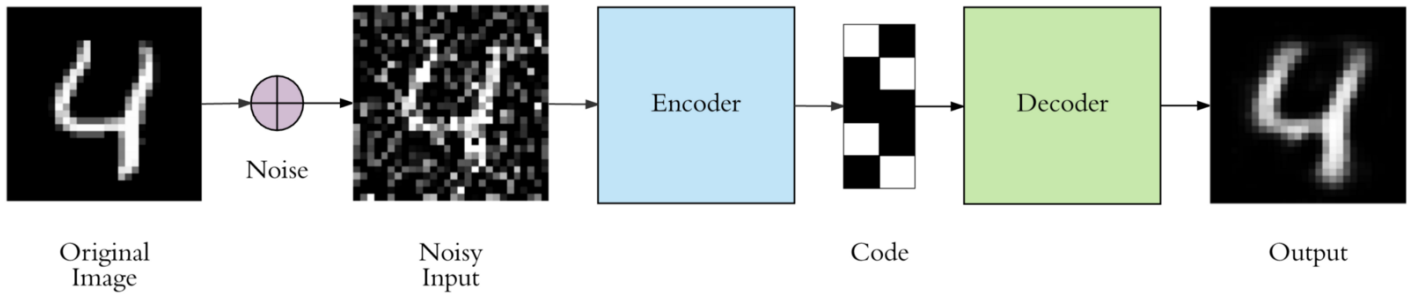
Industrial AI Lab.

Prof. Seungchul Lee

DAE

- DAE is able to reconstruct the corrupted data

- When calculating the loss function, it is important to compare the output values with the original input, not with the corrupted input. That way, the risk of learning the identity function instead of extracting features is eliminated.



TensorFlow Implementation

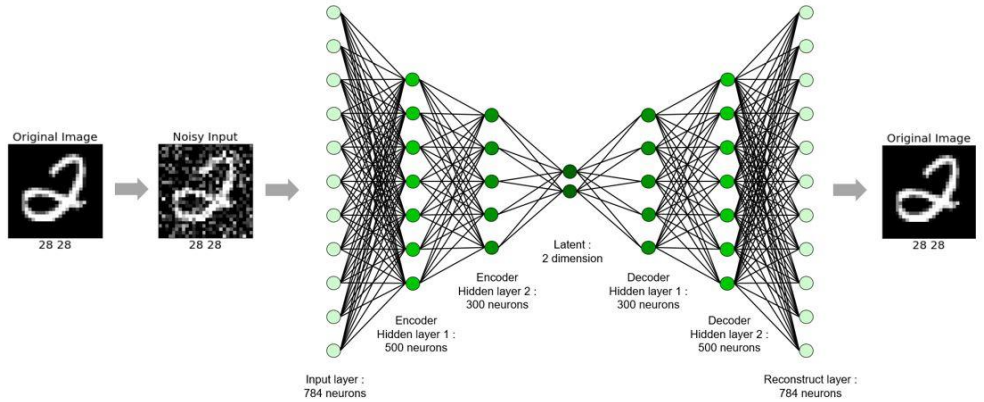
```
sess = tf.Session()
init = tf.global_variables_initializer()
sess.run(init)

loss_record_train = []
loss_record_test = []
for epoch in range(n_iter):

    train_x, _ = mnist.train.next_batch(n_batch)
    train_x_noisy = train_x + np.random.normal(0, 0.1, 784)

    sess.run(optm, feed_dict = {x: train_x_noisy, y: train_x})

    if epoch % n_prt == 0:
        test_x, _ = mnist.test.next_batch(n_batch)
        test_x_noisy = test_x + np.random.normal(0, 0.1, 784)
        c1 = sess.run(loss, feed_dict = {x: train_x_noisy, y: train_x})
        c2 = sess.run(loss, feed_dict = {x: test_x_noisy, y: test_x})
        loss_record_train.append(c1)
        loss_record_test.append(c2)
        print ("Iter : {}".format(epoch))
        print ("Cost : {}".format(c1))
```




```
LR = 0.0001

latent = encoder(x, weights, biases)
reconst = decoder(latent, weights, biases)
loss = tf.square(tf.subtract(y, reconst))
loss = tf.reduce_mean(loss)

optm = tf.train.AdamOptimizer(LR).minimize(loss)
```

Denoised MNIST