P2_Rajasekharan_Pillai_Murali_Krishnan_autoencoders

October 3, 2019

#Student Name: Murali Krishnan Rajasekharan Pillai #ECE 595 Machine Learning II #Project 2: Autoencoders - Student Code

[1]:
```
!nvidia-smi
```

```
Thu Oct  3 23:29:30 2019
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 430.40       Driver Version: 418.67       CUDA Version: 10.1     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla K80           Off  | 00000000:00:04.0 Off |                    0 |
| N/A   35C    P8    29W / 149W |      0MiB / 11441MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

[46]:
```
from google.colab import drive
drive.mount('/content/gdrive', force_remount=True)
MODEL_LOCATION = "gdrive/My Drive/ece595_ml2/models/"
```

```
Mounted at /content/gdrive
```

[3]:
```
#Import necessary packages
import numpy as np
from keras.datasets import mnist
from keras.layers import Dense, Conv2D, MaxPooling2D, UpSampling2D
from keras.models import Sequential, model_from_json
import matplotlib.pyplot as plt
from keras import backend as K
from tensorflow.keras.callbacks import Callback
```

```
import pickle
```

```
Using TensorFlow backend.
```

#Part 0: Importing and Normalizing Data

[4]:
```
#Load MNIST data and normalize to [0,1]
(data_train, _), (data_test, _) = mnist.load_data()
data_train = data_train/255.0
data_test = data_test/255.0
```

```
Downloading data from https://s3.amazonaws.com/img-datasets/mnist.npz
11493376/11490434 [==============================] - 1s 0us/step
```

[0]:
```
#Reshape training and testing data into 784-dimensional vectors
data_train = data_train.reshape(-1, 784)
data_test = data_test.reshape(-1, 784)
```

1 Part 1 : Fully Connected Autoencoder

[0]:
```
class model_methods(object):
  def __init__(self, loss_fn, optim, ndim):
    """
    Try to develop a class which contains common functionality of
    NN models. Like saving a model & it's weights."""
    self.loss_fn = loss_fn
    self.optim = optim
    self.ndim = ndim

  def save_model_weights(self, h5_file_name):
    """
    Save weights of the model
    Parameters:
      :h5_file_name:    Identifier of the model weights h5 file
    """
    self.model.save_weights(h5_file_name)

  def save_model(self, json_file_name):
    """
    Save the model
    Parameters:
      :json_file_name:    Identifier of the model in json file
    """
    ae_model_json = self.model.to_json()
    with open(json_file_name, 'w') as json_file:
      json_file.write(ae_model_json)
```

```
  def load_model(self, json_file_name, h5_file_name):
    json_file = open(MODEL_LOCATION + json_file_name, 'r')
    loaded_from_json = json_file.read()
    json_file.close()
    ae_model = model_from_json(loaded_from_json)
    ae_model.load_weights(MODEL_LOCATION + h5_file_name)
    return ae_model

  def save_model_history(self, model_history, file_name):
    """
    Save model history as a pickle file
    """
    with open(file_name, 'wb') as f:
      pickle.dump(model_history, f)

  def load_model_history(self, pkl_file_name):
    """
    Load model history pickle file
    """
    with open(MODEL_LOCATION + pkl_file_name , 'rb') as f:
      ae_history = pickle.load(f)
    return ae_history
```

[0]:
```
class deep_autoencoder(model_methods):
  def __init__(self, loss_fn, optim, ndim):
    super().__init__(loss_fn, optim, ndim)
    self.model = self._build_model(self.ndim)


  def _build_model(self, ndim):
    """
    Defines and compiles the architecture
    Parameters:
      :loss_fn:    The loss function used in the model
      :optim:      The optimizer used for model
    Returns:
      :model:  The compiled model
    """
    model = Sequential()
    # First Hidden Layer
    model.add(Dense(400,
                    activation='relu',
                    kernel_initializer='uniform',
                    input_dim=self.ndim))
    # Second Hidden Layer
    model.add(Dense(196,
```

```
                    activation='relu',
                    kernel_initializer='uniform'))
    # third Hidden Layer (BottleNeck)
    model.add(Dense(100,
                    activation='relu',
                    kernel_initializer='uniform'))
    # Fourth Hidden Layer
    model.add(Dense(196,
                    activation='relu',
                    kernel_initializer='uniform'))
    # Fifth Hidden Layer
    model.add(Dense(400,
                    activation='relu',
                    kernel_initializer='uniform'))
    # Output Layer
    model.add(Dense(self.ndim,
                    activation='sigmoid',
                    kernel_initializer='uniform'))
    model.compile(loss=self.loss_fn,
                  optimizer=self.optim)
    return model

  def fit(self, d_train, d_test,
          n_epochs=100,
          batch_size=50,
          display=25):
    """
      Fit the model

      Parameters:
        :d_train:    Tuple of (training data, training labels)
        :d_test:     Tuple of (testing data, testing labels)
        :n_epochs:   Number of epochs for fit
        :batch_size: Number of samples per gradient update
      Returns:
        :model_history: History object containing all model history info

    """
    data_train, labels_train = d_train
    #out_batch = NBatchLogger(display)
    model_history = self.model.fit(data_train, labels_train,
                          validation_data=d_test,
                          epochs=n_epochs,
                          batch_size=batch_size,
                          shuffle=True,
                          )
```

```
    return model_history
```

[0]:
```
ae = deep_autoencoder("mse", "adam", 784)
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:66: The name tf.get_default_graph
is deprecated. Please use tf.compat.v1.get_default_graph instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:541: The name tf.placeholder is
deprecated. Please use tf.compat.v1.placeholder instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:4432: The name tf.random_uniform is
deprecated. Please use tf.random.uniform instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/optimizers.py:793: The name tf.train.Optimizer is deprecated.
Please use tf.compat.v1.train.Optimizer instead.
```

[0]:
```
ae_history = ae.fit((data_train, data_train),
                    (data_test, data_test),
                    n_epochs=150,
                    batch_size=1024)
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:1033: The name tf.assign_add is
deprecated. Please use tf.compat.v1.assign_add instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:1020: The name tf.assign is
deprecated. Please use tf.compat.v1.assign instead.

Train on 60000 samples, validate on 10000 samples
Epoch 1/150
60000/60000 [==============================] - 2s 37us/step - loss: 0.1067 -
val_loss: 0.0730
Epoch 2/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0695 -
val_loss: 0.0654
Epoch 3/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0619 -
val_loss: 0.0573
Epoch 4/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0547 -
val_loss: 0.0507
```

```
Epoch 5/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0475 -
val_loss: 0.0423
Epoch 6/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0374 -
val_loss: 0.0337
Epoch 7/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0321 -
val_loss: 0.0299
Epoch 8/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0289 -
val_loss: 0.0269
Epoch 9/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0264 -
val_loss: 0.0251
Epoch 10/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0247 -
val_loss: 0.0236
Epoch 11/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0232 -
val_loss: 0.0223
Epoch 12/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0221 -
val_loss: 0.0212
Epoch 13/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0210 -
val_loss: 0.0201
Epoch 14/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0200 -
val_loss: 0.0193
Epoch 15/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0191 -
val_loss: 0.0184
Epoch 16/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0182 -
val_loss: 0.0175
Epoch 17/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0175 -
val_loss: 0.0173
Epoch 18/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0169 -
val_loss: 0.0165
Epoch 19/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0165 -
val_loss: 0.0162
Epoch 20/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0161 -
val_loss: 0.0157
```

```
Epoch 21/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0157 -
val_loss: 0.0154
Epoch 22/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0152 -
val_loss: 0.0150
Epoch 23/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0148 -
val_loss: 0.0145
Epoch 24/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0143 -
val_loss: 0.0141
Epoch 25/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0140 -
val_loss: 0.0137
Epoch 26/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0136 -
val_loss: 0.0135
Epoch 27/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0133 -
val_loss: 0.0133
Epoch 28/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0130 -
val_loss: 0.0128
Epoch 29/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0126 -
val_loss: 0.0125
Epoch 30/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0124 -
val_loss: 0.0123
Epoch 31/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0121 -
val_loss: 0.0120
Epoch 32/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0119 -
val_loss: 0.0118
Epoch 33/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0117 -
val_loss: 0.0115
Epoch 34/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0114 -
val_loss: 0.0112
Epoch 35/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0112 -
val_loss: 0.0111
Epoch 36/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0110 -
val_loss: 0.0108
```

```
Epoch 37/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0108 -
val_loss: 0.0107
Epoch 38/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0106 -
val_loss: 0.0106
Epoch 39/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0105 -
val_loss: 0.0105
Epoch 40/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0104 -
val_loss: 0.0102
Epoch 41/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0101 -
val_loss: 0.0102
Epoch 42/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0100 -
val_loss: 0.0100
Epoch 43/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0099 -
val_loss: 0.0098
Epoch 44/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0098 -
val_loss: 0.0097
Epoch 45/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0096 -
val_loss: 0.0096
Epoch 46/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0095 -
val_loss: 0.0096
Epoch 47/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0094 -
val_loss: 0.0096
Epoch 48/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0093 -
val_loss: 0.0093
Epoch 49/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0091 -
val_loss: 0.0093
Epoch 50/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0090 -
val_loss: 0.0090
Epoch 51/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0089 -
val_loss: 0.0089
Epoch 52/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0088 -
val_loss: 0.0088
```

```
Epoch 53/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0087 -
val_loss: 0.0088
Epoch 54/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0086 -
val_loss: 0.0088
Epoch 55/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0085 -
val_loss: 0.0086
Epoch 56/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0084 -
val_loss: 0.0085
Epoch 57/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0082 -
val_loss: 0.0083
Epoch 58/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0082 -
val_loss: 0.0083
Epoch 59/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0081 -
val_loss: 0.0081
Epoch 60/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0079 -
val_loss: 0.0081
Epoch 61/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0079 -
val_loss: 0.0083
Epoch 62/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0078 -
val_loss: 0.0080
Epoch 63/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0077 -
val_loss: 0.0078
Epoch 64/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0076 -
val_loss: 0.0078
Epoch 65/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0075 -
val_loss: 0.0076
Epoch 66/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0075 -
val_loss: 0.0076
Epoch 67/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0074 -
val_loss: 0.0075
Epoch 68/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0073 -
val_loss: 0.0074
```

```
Epoch 69/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0073 -
val_loss: 0.0074
Epoch 70/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0072 -
val_loss: 0.0074
Epoch 71/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0071 -
val_loss: 0.0074
Epoch 72/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0071 -
val_loss: 0.0073
Epoch 73/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0070 -
val_loss: 0.0074
Epoch 74/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0069 -
val_loss: 0.0071
Epoch 75/150
60000/60000 [==============================] - 1s 23us/step - loss: 0.0069 -
val_loss: 0.0071
Epoch 76/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0069 -
val_loss: 0.0071
Epoch 77/150
60000/60000 [==============================] - 1s 23us/step - loss: 0.0068 -
val_loss: 0.0070
Epoch 78/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0067 -
val_loss: 0.0069
Epoch 79/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0067 -
val_loss: 0.0068
Epoch 80/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0066 -
val_loss: 0.0068
Epoch 81/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0066 -
val_loss: 0.0067
Epoch 82/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0065 -
val_loss: 0.0069
Epoch 83/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0065 -
val_loss: 0.0066
Epoch 84/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0064 -
val_loss: 0.0066
```

```
Epoch 85/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0064 -
val_loss: 0.0067
Epoch 86/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0064 -
val_loss: 0.0066
Epoch 87/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0063 -
val_loss: 0.0067
Epoch 88/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0063 -
val_loss: 0.0065
Epoch 89/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0062 -
val_loss: 0.0065
Epoch 90/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0062 -
val_loss: 0.0065
Epoch 91/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0062 -
val_loss: 0.0065
Epoch 92/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0061 -
val_loss: 0.0064
Epoch 93/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0061 -
val_loss: 0.0064
Epoch 94/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0061 -
val_loss: 0.0063
Epoch 95/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0060 -
val_loss: 0.0063
Epoch 96/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0060 -
val_loss: 0.0063
Epoch 97/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0059 -
val_loss: 0.0062
Epoch 98/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0059 -
val_loss: 0.0061
Epoch 99/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0059 -
val_loss: 0.0062
Epoch 100/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0058 -
val_loss: 0.0060
```

```
Epoch 101/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0058 -
val_loss: 0.0061
Epoch 102/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0058 -
val_loss: 0.0060
Epoch 103/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0057 -
val_loss: 0.0060
Epoch 104/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0057 -
val_loss: 0.0059
Epoch 105/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0057 -
val_loss: 0.0060
Epoch 106/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0057 -
val_loss: 0.0061
Epoch 107/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0056 -
val_loss: 0.0060
Epoch 108/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0056 -
val_loss: 0.0059
Epoch 109/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0056 -
val_loss: 0.0059
Epoch 110/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0056 -
val_loss: 0.0058
Epoch 111/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0055 -
val_loss: 0.0058
Epoch 112/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0055 -
val_loss: 0.0058
Epoch 113/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0055 -
val_loss: 0.0058
Epoch 114/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0055 -
val_loss: 0.0057
Epoch 115/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0054 -
val_loss: 0.0057
Epoch 116/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0054 -
val_loss: 0.0057
```

```
Epoch 117/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0054 -
val_loss: 0.0057
Epoch 118/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0054 -
val_loss: 0.0056
Epoch 119/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0053 -
val_loss: 0.0056
Epoch 120/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0053 -
val_loss: 0.0057
Epoch 121/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0053 -
val_loss: 0.0055
Epoch 122/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0052 -
val_loss: 0.0057
Epoch 123/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0052 -
val_loss: 0.0055
Epoch 124/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0052 -
val_loss: 0.0055
Epoch 125/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0052 -
val_loss: 0.0056
Epoch 126/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0051 -
val_loss: 0.0054
Epoch 127/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0051 -
val_loss: 0.0054
Epoch 128/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0051 -
val_loss: 0.0054
Epoch 129/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0051 -
val_loss: 0.0054
Epoch 130/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0051 -
val_loss: 0.0054
Epoch 131/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0051 -
val_loss: 0.0054
Epoch 132/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0050 -
val_loss: 0.0053
```

```
Epoch 133/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0050 -
val_loss: 0.0053
Epoch 134/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0050 -
val_loss: 0.0054
Epoch 135/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0050 -
val_loss: 0.0053
Epoch 136/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0050 -
val_loss: 0.0053
Epoch 137/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0049 -
val_loss: 0.0053
Epoch 138/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0049 -
val_loss: 0.0053
Epoch 139/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0049 -
val_loss: 0.0053
Epoch 140/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0049 -
val_loss: 0.0053
Epoch 141/150
60000/60000 [==============================] - 1s 22us/step - loss: 0.0049 -
val_loss: 0.0052
Epoch 142/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0048 -
val_loss: 0.0052
Epoch 143/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0048 -
val_loss: 0.0052
Epoch 144/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0048 -
val_loss: 0.0051
Epoch 145/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0048 -
val_loss: 0.0051
Epoch 146/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0047 -
val_loss: 0.0051
Epoch 147/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0047 -
val_loss: 0.0051
Epoch 148/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0047 -
val_loss: 0.0051
```

```
Epoch 149/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0047 -
val_loss: 0.0051
Epoch 150/150
60000/60000 [==============================] - 1s 21us/step - loss: 0.0047 -
val_loss: 0.0050
```

[0]:
```
ae.save_model_weights("ae_model_weights.h5")
ae.save_model("ae_model.json")
ae.save_model_history(ae_history, "ae_model_history.pkl")
```

[0]:
```
! cp -r ae_model.json ae_model_weights.h5 ae_model_history.pkl ./gdrive/My\␣
↪Drive/ece595_ml2/models/
```

[0]:
```
ae_model = ae.load_model("ae_model.json", "ae_model_weights.h5")
```

[0]:
```
reconstructions = ae_model.predict(data_test)
```

[0]:
```
print(len(ae_model.layers))
print(ae_model.summary())
```

```
6
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 400)               314000
_________________________________________________________________
dense_2 (Dense)              (None, 196)               78596
_________________________________________________________________
dense_3 (Dense)              (None, 100)               19700
_________________________________________________________________
dense_4 (Dense)              (None, 196)               19796
_________________________________________________________________
dense_5 (Dense)              (None, 400)               78800
_________________________________________________________________
dense_6 (Dense)              (None, 784)               314384
=================================================================
Total params: 825,276
Trainable params: 825,276
Non-trainable params: 0
_________________________________________________________________
None
```

[0]:
```
def get_hidden_layers_representation(model, data_test):
  """
  Find the hidden layers representation for a model
  Parameters:
    :model:        The model object whose hidden layers to extract
```

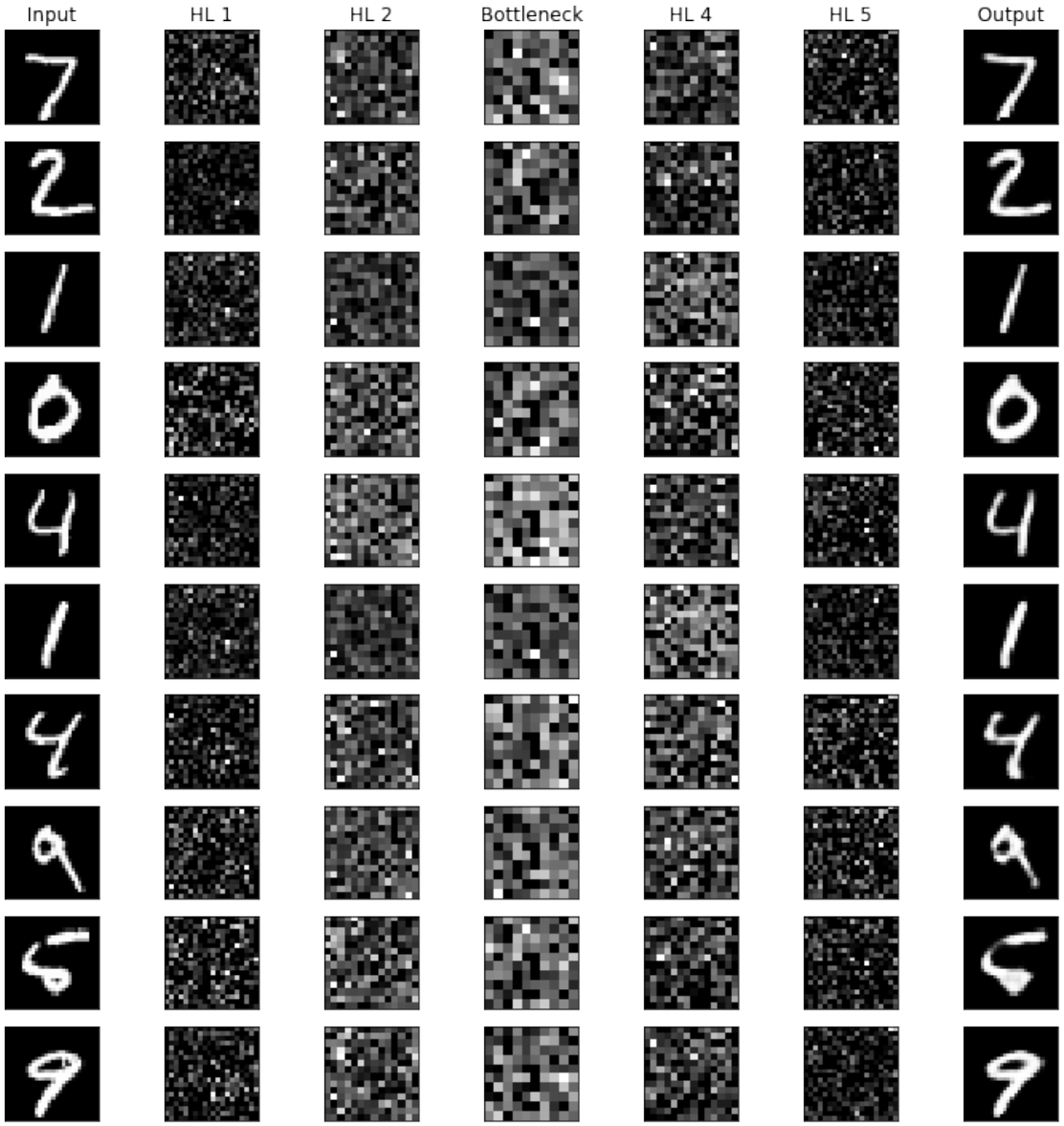
```
    :test data:    The test data whose representations we seek
  Returns:
    :hl:           The weights of the hidden layers
  """

  n_layers = len(model.layers)
  hl_weights = {}
  for i in range(1, n_layers+1, 1):
    hl_function = K.function([model.layers[0].input], [model.layers[i-1].output])
    hl_weights[i] = []
    hl_weights[i].append(hl_function([data_test])[0])
  return hl_weights
```

[0]:
```
w_hl = get_hidden_layers_representation(ae_model, data_test)
```

1.1 Layer Representations

[0]:
```
n = 10
n_layers = len(ae_model.layers)
layers = [' Input ',
          ' HL 1 ',
          ' HL 2 ',
          ' Bottleneck ',
          ' HL 4 ',
          ' HL 5 ',
          ' Output ']
cols = ['{}'.format(col) for col in layers]
#rows = ['Row {}'.format(row) for row in ['A', 'B', 'C', 'D']]
fig, axes = plt.subplots(nrows=n, ncols=n_layers+1, figsize=(10, 10))
for i in range(n):
    axes[i, 0].imshow(data_test[i].reshape(28, 28),
                  cmap=plt.cm.gray)
    axes[i, 1].imshow(w_hl[1][0][i,:].reshape(20, 20),
                  cmap=plt.cm.gray)
    axes[i, 2].imshow(w_hl[2][0][i,:].reshape(14, 14),
                  cmap=plt.cm.gray)
    axes[i, 3].imshow(w_hl[3][0][i,:].reshape(10, 10),
                  cmap=plt.cm.gray)
    axes[i, 4].imshow(w_hl[4][0][i,:].reshape(14, 14),
                  cmap=plt.cm.gray)
    axes[i, 5].imshow(w_hl[5][0][i,:].reshape(20, 20),
                  cmap=plt.cm.gray)
    axes[i, 6].imshow(w_hl[6][0][i,:].reshape(28, 28),
                  cmap=plt.cm.gray)
    for ax, col in zip(axes[0], cols):
      ax.set_title(col)
    for j in range(n_layers+1):
```

```
      axes[i, j].get_xaxis().set_visible(False)
      axes[i, j].get_yaxis().set_visible(False)
fig.tight_layout()
plt.show()
```

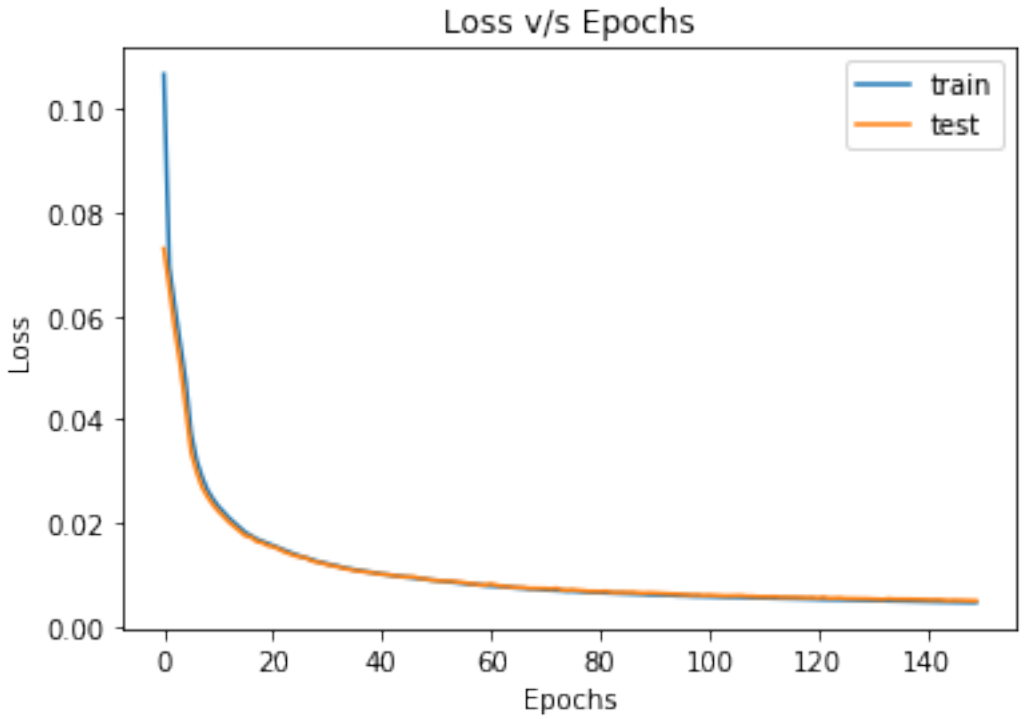


1.2 Loss Curves

[0]:
```
ae_history = ae.load_model_history("ae_model_history.pkl")
```

[0]:
```
plt.plot(ae_history.history['loss'])
plt.plot(ae_history.history['val_loss'])
```

```
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.title("Loss v/s Epochs")
plt.legend(['train', 'test'], loc='upper right')
plt.show()
```



#Part 2: Deep Convolutional AutoEncoder

[0]:
```
#Reshape training and testing data into 784-dimensional vectors
data_train = data_train.reshape(-1, 28, 28, 1)
data_test = data_test.reshape(-1, 28, 28, 1)
```

[0]:
```
class convolutional_autoencoder(model_methods):
  def __init__(self, loss_fn, optim, ndim):
    super().__init__(loss_fn, optim, ndim)
    self.model = self._build_model(self.ndim)


  def _build_model(self, ndim):
    """
    Defines and compiles the architecture
    Parameters:
      :loss_fn:    The loss function used in the model
      :optim:      The optimizer used for model
    Returns:
```

```
    :model:  The compiled model
    """
    model = Sequential()

    # First Hidden Layer
    model.add(Conv2D(16,
                     kernel_size=3,
                     activation='relu',
                     kernel_initializer='uniform',
                     padding='same',
                     input_shape=(self.ndim, self.ndim, 1)))

    # Second Hidden Layer
    model.add(MaxPooling2D(pool_size=(2, 2),
                           padding='same'))
    # Third Hidden Layer
    model.add(Conv2D(8,
                     kernel_size=3,
                     activation='relu',
                     padding='same',
                     kernel_initializer='uniform'))

    # Fourth Hidden Layer (Bottleneck)
    model.add(MaxPooling2D(pool_size=(2, 2),
                           padding='same'))
    #Fifth Hidden Layer
    model.add(Conv2D(8,
                     kernel_size=3,
                     padding='same',
                     activation='relu',
                     kernel_initializer='uniform'))
    # Sixth Hidden Layer
    model.add(UpSampling2D(size=(2, 2)))
    #Seventh Hidden Layer
    model.add(Conv2D(16,
                     kernel_size=3,
                     activation='relu',
                     padding='same',
                     kernel_initializer='uniform'))
    # Eighth hidden layer
    model.add(UpSampling2D(size=(2, 2)))
    # Output layer
    model.add(Conv2D(1,
                     kernel_size=3,
                     kernel_initializer='uniform',
                     padding='same',
                     activation='sigmoid'))
```

```
    model.compile(loss=self.loss_fn,
                  optimizer=self.optim)
    return model

  def fit(self, d_train, d_test,
          n_epochs=100,
          batch_size=50,
          display=25):
    """
      Fit the model

      Parameters:
        :d_train:    Tuple of (training data, training labels)
        :d_test:     Tuple of (testing data, testing labels)
        :n_epochs:   Number of epochs for fit
        :batch_size: Number of samples per gradient update
      Returns:
        :model_history: History object containing all model history info

    """
    data_train, labels_train = d_train
    #out_batch = NBatchLogger(display)
    model_history = self.model.fit(data_train, labels_train,
                          validation_data=d_test,
                          epochs=n_epochs,
                          batch_size=batch_size,
                          shuffle=True,
                          )
    return model_history
```

[8]:
```
cae = convolutional_autoencoder('binary_crossentropy', 'adam', 28)
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:66: The name tf.get_default_graph
is deprecated. Please use tf.compat.v1.get_default_graph instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:541: The name tf.placeholder is
deprecated. Please use tf.compat.v1.placeholder instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:4432: The name tf.random_uniform is
deprecated. Please use tf.random.uniform instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:4267: The name tf.nn.max_pool is
deprecated. Please use tf.nn.max_pool2d instead.
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:2239: The name
tf.image.resize_nearest_neighbor is deprecated. Please use
tf.compat.v1.image.resize_nearest_neighbor instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/optimizers.py:793: The name tf.train.Optimizer is deprecated.
Please use tf.compat.v1.train.Optimizer instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:3657: The name tf.log is
deprecated. Please use tf.math.log instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/tensorflow/python/ops/nn_impl.py:180:
add_dispatch_support.<locals>.wrapper (from tensorflow.python.ops.array_ops) is
deprecated and will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
```

[0]:
```
cae_history = cae.fit((data_train, data_train),
                      (data_test, data_test),
                      n_epochs=150,
                      batch_size=1024)
```

```
Train on 60000 samples, validate on 10000 samples
Epoch 1/150
60000/60000 [==============================] - 6s 107us/step - loss: 0.5798 -
val_loss: 0.4495
Epoch 2/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.3640 -
val_loss: 0.2452
Epoch 3/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.1978 -
val_loss: 0.1640
Epoch 4/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.1455 -
val_loss: 0.1287
Epoch 5/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.1219 -
val_loss: 0.1145
Epoch 6/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.1117 -
val_loss: 0.1066
Epoch 7/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.1045 -
```

```
val_loss: 0.1001
Epoch 8/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0990 -
val_loss: 0.0957
Epoch 9/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0953 -
val_loss: 0.0930
Epoch 10/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0928 -
val_loss: 0.0906
Epoch 11/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0909 -
val_loss: 0.0890
Epoch 12/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0894 -
val_loss: 0.0876
Epoch 13/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0881 -
val_loss: 0.0865
Epoch 14/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0870 -
val_loss: 0.0854
Epoch 15/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0860 -
val_loss: 0.0845
Epoch 16/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0851 -
val_loss: 0.0837
Epoch 17/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0843 -
val_loss: 0.0829
Epoch 18/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0835 -
val_loss: 0.0822
Epoch 19/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0829 -
val_loss: 0.0817
Epoch 20/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0824 -
val_loss: 0.0812
Epoch 21/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0819 -
val_loss: 0.0807
Epoch 22/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0815 -
val_loss: 0.0803
Epoch 23/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0811 -
```

```
val_loss: 0.0801
Epoch 24/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0807 -
val_loss: 0.0796
Epoch 25/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0804 -
val_loss: 0.0794
Epoch 26/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0801 -
val_loss: 0.0790
Epoch 27/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0797 -
val_loss: 0.0786
Epoch 28/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0794 -
val_loss: 0.0784
Epoch 29/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0791 -
val_loss: 0.0781
Epoch 30/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0788 -
val_loss: 0.0779
Epoch 31/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0786 -
val_loss: 0.0777
Epoch 32/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0784 -
val_loss: 0.0775
Epoch 33/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0782 -
val_loss: 0.0773
Epoch 34/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0781 -
val_loss: 0.0771
Epoch 35/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0779 -
val_loss: 0.0770
Epoch 36/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0777 -
val_loss: 0.0768
Epoch 37/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0776 -
val_loss: 0.0767
Epoch 38/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0774 -
val_loss: 0.0765
Epoch 39/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0773 -
```

```
val_loss: 0.0764
Epoch 40/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0772 -
val_loss: 0.0763
Epoch 41/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0770 -
val_loss: 0.0762
Epoch 42/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0769 -
val_loss: 0.0762
Epoch 43/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0769 -
val_loss: 0.0759
Epoch 44/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0767 -
val_loss: 0.0758
Epoch 45/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0765 -
val_loss: 0.0757
Epoch 46/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0764 -
val_loss: 0.0757
Epoch 47/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0763 -
val_loss: 0.0756
Epoch 48/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0763 -
val_loss: 0.0754
Epoch 49/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0762 -
val_loss: 0.0754
Epoch 50/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0761 -
val_loss: 0.0753
Epoch 51/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0760 -
val_loss: 0.0752
Epoch 52/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0759 -
val_loss: 0.0751
Epoch 53/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0758 -
val_loss: 0.0750
Epoch 54/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0758 -
val_loss: 0.0749
Epoch 55/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0757 -
```

```
val_loss: 0.0749
Epoch 56/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0756 -
val_loss: 0.0748
Epoch 57/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0756 -
val_loss: 0.0747
Epoch 58/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0755 -
val_loss: 0.0747
Epoch 59/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0754 -
val_loss: 0.0746
Epoch 60/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0754 -
val_loss: 0.0745
Epoch 61/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0753 -
val_loss: 0.0745
Epoch 62/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0753 -
val_loss: 0.0745
Epoch 63/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0752 -
val_loss: 0.0744
Epoch 64/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0751 -
val_loss: 0.0743
Epoch 65/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0750 -
val_loss: 0.0743
Epoch 66/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0750 -
val_loss: 0.0742
Epoch 67/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0749 -
val_loss: 0.0742
Epoch 68/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0749 -
val_loss: 0.0741
Epoch 69/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0749 -
val_loss: 0.0741
Epoch 70/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0748 -
val_loss: 0.0740
Epoch 71/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0748 -
```

```
val_loss: 0.0740
Epoch 72/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0747 -
val_loss: 0.0739
Epoch 73/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0746 -
val_loss: 0.0739
Epoch 74/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0746 -
val_loss: 0.0739
Epoch 75/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0746 -
val_loss: 0.0738
Epoch 76/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0745 -
val_loss: 0.0738
Epoch 77/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0745 -
val_loss: 0.0737
Epoch 78/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0744 -
val_loss: 0.0737
Epoch 79/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0745 -
val_loss: 0.0736
Epoch 80/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0743 -
val_loss: 0.0736
Epoch 81/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0743 -
val_loss: 0.0736
Epoch 82/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0743 -
val_loss: 0.0735
Epoch 83/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0742 -
val_loss: 0.0735
Epoch 84/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0742 -
val_loss: 0.0735
Epoch 85/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0742 -
val_loss: 0.0734
Epoch 86/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0742 -
val_loss: 0.0735
Epoch 87/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0741 -
```

```
val_loss: 0.0733
Epoch 88/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0741 -
val_loss: 0.0733
Epoch 89/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0740 -
val_loss: 0.0733
Epoch 90/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0740 -
val_loss: 0.0733
Epoch 91/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0740 -
val_loss: 0.0733
Epoch 92/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0739 -
val_loss: 0.0732
Epoch 93/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0739 -
val_loss: 0.0732
Epoch 94/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0739 -
val_loss: 0.0732
Epoch 95/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0739 -
val_loss: 0.0731
Epoch 96/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0738 -
val_loss: 0.0731
Epoch 97/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0738 -
val_loss: 0.0731
Epoch 98/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0738 -
val_loss: 0.0730
Epoch 99/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0738 -
val_loss: 0.0730
Epoch 100/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0737 -
val_loss: 0.0730
Epoch 101/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0737 -
val_loss: 0.0730
Epoch 102/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0737 -
val_loss: 0.0729
Epoch 103/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0736 -
```

```
val_loss: 0.0729
Epoch 104/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0737 -
val_loss: 0.0729
Epoch 105/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0736 -
val_loss: 0.0729
Epoch 106/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0736 -
val_loss: 0.0729
Epoch 107/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0735 -
val_loss: 0.0728
Epoch 108/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0735 -
val_loss: 0.0728
Epoch 109/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0735 -
val_loss: 0.0728
Epoch 110/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0735 -
val_loss: 0.0728
Epoch 111/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0735 -
val_loss: 0.0728
Epoch 112/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0734 -
val_loss: 0.0728
Epoch 113/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0734 -
val_loss: 0.0727
Epoch 114/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0734 -
val_loss: 0.0727
Epoch 115/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0734 -
val_loss: 0.0728
Epoch 116/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0734 -
val_loss: 0.0727
Epoch 117/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0733 -
val_loss: 0.0726
Epoch 118/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0734 -
val_loss: 0.0726
Epoch 119/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0733 -
```

```
val_loss: 0.0726
Epoch 120/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0733 -
val_loss: 0.0728
Epoch 121/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0733 -
val_loss: 0.0726
Epoch 122/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0733 -
val_loss: 0.0726
Epoch 123/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0732 -
val_loss: 0.0726
Epoch 124/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0732 -
val_loss: 0.0726
Epoch 125/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0732 -
val_loss: 0.0725
Epoch 126/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0732 -
val_loss: 0.0725
Epoch 127/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0732 -
val_loss: 0.0725
Epoch 128/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0732 -
val_loss: 0.0725
Epoch 129/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0731 -
val_loss: 0.0725
Epoch 130/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0731 -
val_loss: 0.0724
Epoch 131/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0731 -
val_loss: 0.0724
Epoch 132/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0731 -
val_loss: 0.0724
Epoch 133/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0731 -
val_loss: 0.0725
Epoch 134/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0731 -
val_loss: 0.0724
Epoch 135/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0730 -
```

```
val_loss: 0.0723
Epoch 136/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0731 -
val_loss: 0.0723
Epoch 137/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0730 -
val_loss: 0.0723
Epoch 138/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0730 -
val_loss: 0.0723
Epoch 139/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0730 -
val_loss: 0.0723
Epoch 140/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0730 -
val_loss: 0.0723
Epoch 141/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0730 -
val_loss: 0.0723
Epoch 142/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0729 -
val_loss: 0.0722
Epoch 143/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0729 -
val_loss: 0.0722
Epoch 144/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0729 -
val_loss: 0.0722
Epoch 145/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0729 -
val_loss: 0.0724
Epoch 146/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0729 -
val_loss: 0.0722
Epoch 147/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0729 -
val_loss: 0.0722
Epoch 148/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0729 -
val_loss: 0.0722
Epoch 149/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0729 -
val_loss: 0.0723
Epoch 150/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0728 -
val_loss: 0.0721
```

[0]:
```
cae.save_model_weights("cae_model_weights.h5")
cae.save_model("cae_model.json")
cae.save_model_history(cae_history, "cae_model_history.pkl")
```

[0]:
```
! cp -r cae_model.json cae_model_weights.h5 cae_model_history.pkl ./gdrive/My\␣
↪Drive/ece595_ml2/models/
```

[0]:
```
cae_model = cae.load_model("cae_model.json", "cae_model_weights.h5")
```

[0]:
```
cae_reconstructions = cae_model.predict(data_test)
```

[14]:
```
print(len(cae_model.layers))
print(cae_model.summary())
```

```
9
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 28, 28, 16)        160
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 14, 14, 16)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 14, 14, 8)         1160
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 7, 7, 8)           0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 7, 7, 8)           584
_________________________________________________________________
up_sampling2d_1 (UpSampling2 (None, 14, 14, 8)         0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 14, 14, 16)        1168
_________________________________________________________________
up_sampling2d_2 (UpSampling2 (None, 28, 28, 16)        0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 28, 28, 1)         145
=================================================================
Total params: 3,217
Trainable params: 3,217
Non-trainable params: 0
_________________________________________________________________
None
```

[0]:
```
cae_w_hl = get_hidden_layers_representation(cae_model, data_test)
```

1.3 Layer Representation

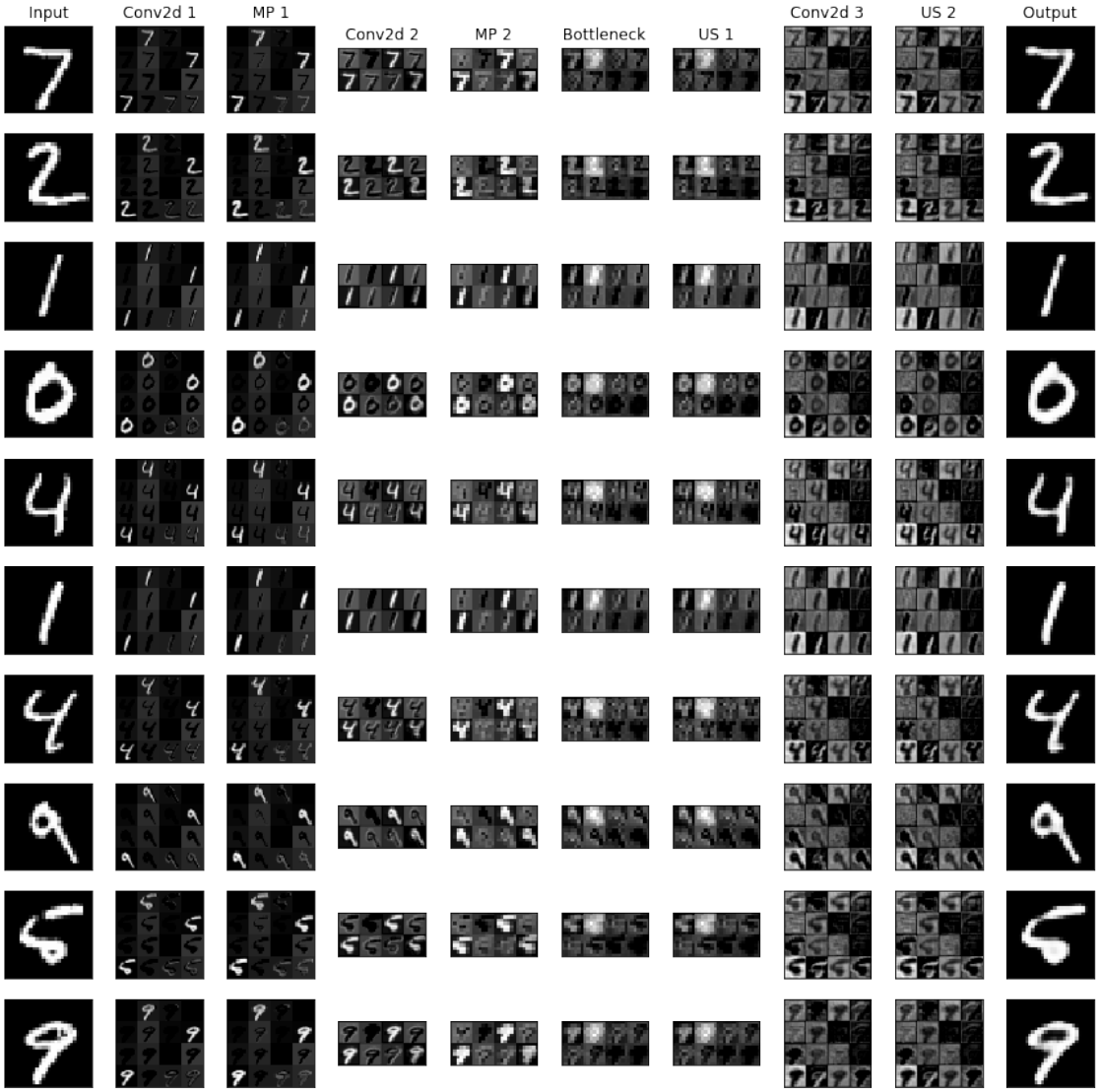
[0]:
```
n_samples = 10
n_layers = len(cae_model.layers)
layers = [' Input ',
          ' Conv2d 1 ',
          ' MP 1 ',
          ' Conv2d 2 ',
          ' MP 2 ',
          ' Bottleneck ',
          ' US 1 ',
          ' Conv2d 3 ',
          ' US 2 ',
          ' Output ']
cols = ['{}'.format(col) for col in layers]
```

[0]:
```
n_kernels = [cae_w_hl[i][0][0].shape[2] for i in range(1, n_layers)]
```

[0]:
```
def hidden_layer_weights(mat):
  """
  hidden weights
  """
  f = mat.shape[0]
  m = max(mat.shape[2] // 4, 1)
  n = max(mat.shape[2] // m, 1)
  temp = np.zeros((f*m, f*n))
  for j in range(m):
    for k in range(n):
      temp[f*j:f*(j+1), f*k:f*(k+1)] = mat[:, :, j*m+k]
  return temp
```

[44]:
```
fig, axes = plt.subplots(nrows=n_samples, ncols=n_layers+1, figsize=(12, 12))
for i in range(n_samples):
    axes[i, 0].imshow(data_test[i].reshape(28, 28),
                  cmap=plt.cm.gray)

    temp1 = hidden_layer_weights(cae_w_hl[1][0][i])
    axes[i, 1].imshow(temp1, cmap=plt.cm.gray)
    temp2 = hidden_layer_weights(cae_w_hl[2][0][i])
    axes[i, 2].imshow(temp2, cmap=plt.cm.gray)
    temp3 = hidden_layer_weights(cae_w_hl[3][0][i])
    axes[i, 3].imshow(temp3, cmap=plt.cm.gray)
    temp4 = hidden_layer_weights(cae_w_hl[4][0][i])
    axes[i, 4].imshow(temp4, cmap=plt.cm.gray)
    temp5 = hidden_layer_weights(cae_w_hl[5][0][i])
    axes[i, 5].imshow(temp5, cmap=plt.cm.gray)
    temp6 = hidden_layer_weights(cae_w_hl[6][0][i])
    axes[i, 6].imshow(temp6, cmap=plt.cm.gray)
    temp7 = hidden_layer_weights(cae_w_hl[7][0][i])
```

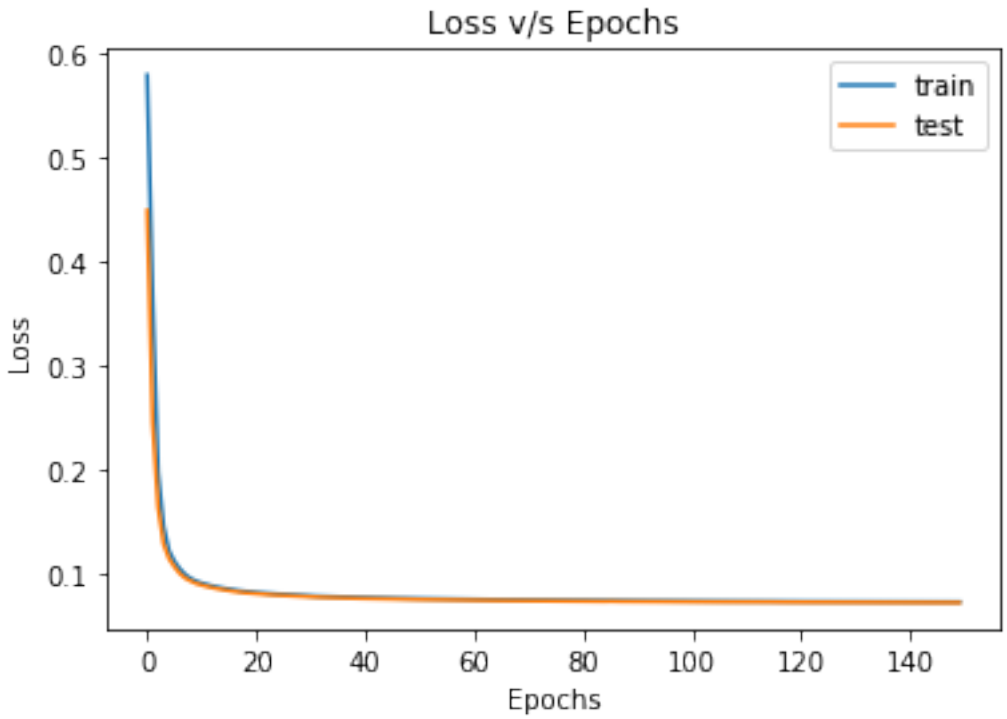
```
    axes[i, 7].imshow(temp7, cmap=plt.cm.gray)
    temp8 = hidden_layer_weights(cae_w_hl[8][0][i])
    axes[i, 8].imshow(temp8, cmap=plt.cm.gray)
    temp9 = hidden_layer_weights(cae_w_hl[9][0][i])
    axes[i, 9].imshow(temp9, cmap=plt.cm.gray)
    for ax, col in zip(axes[0], cols):
      ax.set_title(col)
    for j in range(n_layers+1):
      axes[i, j].get_xaxis().set_visible(False)
      axes[i, j].get_yaxis().set_visible(False)
fig.tight_layout()
plt.show()
```



1.4 Loss Curves

[0]:
```
cae_history = cae.load_model_history("cae_model_history.pkl")
```

[48]:
```
plt.plot(cae_history.history['loss'])
plt.plot(cae_history.history['val_loss'])
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.title("Loss v/s Epochs")
plt.legend(['train', 'test'], loc='upper right')
plt.show()
```



#Part 3: Denoising AutoEncoder

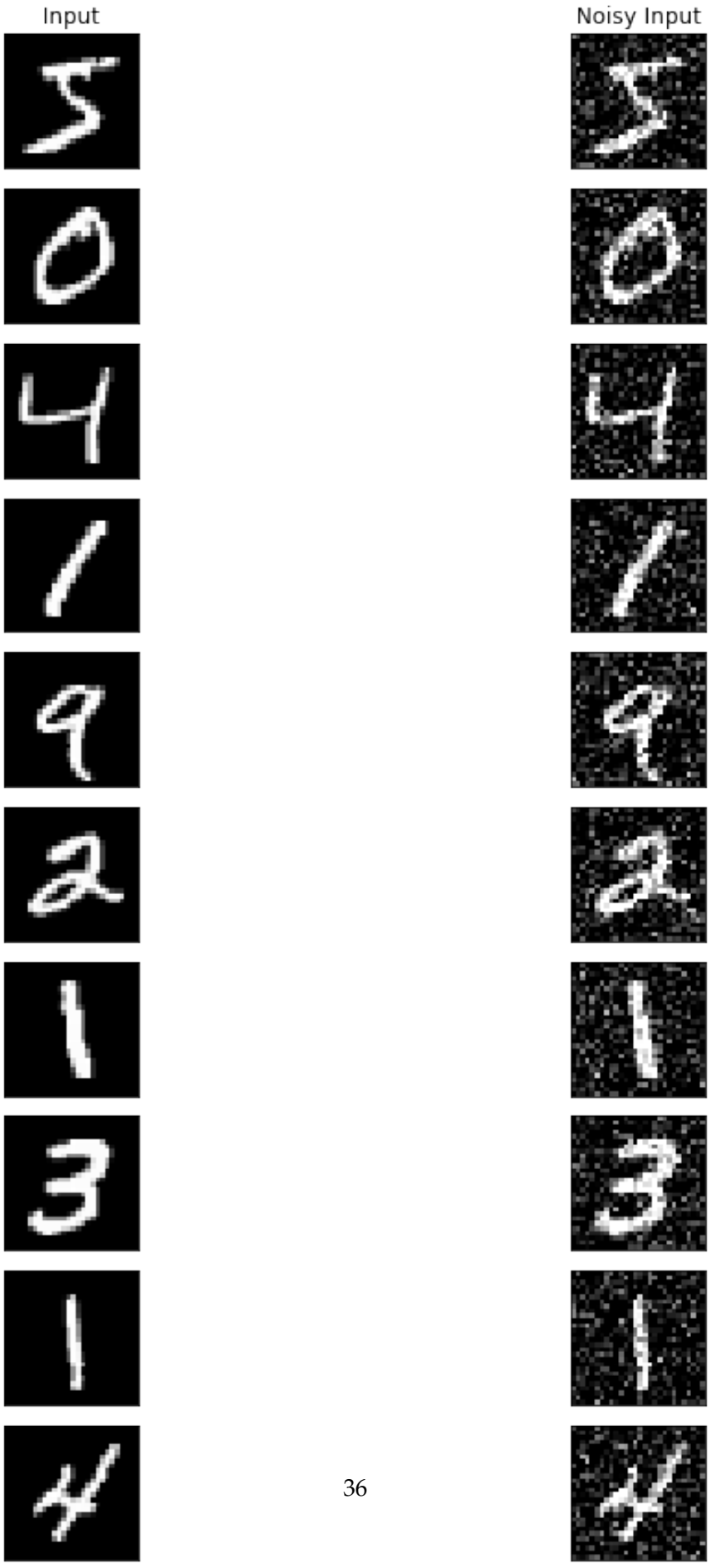
[0]:
```
#Inject noise into testing data
noise_factor = 0.25
data_train_noisy = data_train + noise_factor * np.random.normal(loc=0.0, scale=1.
 ↪0, size=data_train.shape)
data_test_noisy = data_test + noise_factor * np.random.normal(loc=0.0, scale=1.
 ↪0, size=data_test.shape)

#Clip to stay within valid (normalized) pixel range
data_train_noisy = np.clip(data_train_noisy, 0., 1.)
data_test_noisy = np.clip(data_test_noisy, 0., 1.)
```

```
#Reshape data to comply with input of denoising autoencoder
data_train_noisy = data_train_noisy.reshape(-1, 28, 28, 1)
data_test_noisy = data_test_noisy.reshape(-1, 28, 28, 1)
```

1.5 Original Samples of Training Data

[0]:
```
n_samples = 10
layers = [' Input ',
          ' Noisy Input ']
cols = ['{}'.format(col) for col in layers]
```

[55]:
```
fig, axes = plt.subplots(nrows=n_samples, ncols=2, figsize=(12, 12))
for i in range(n_samples):
    axes[i, 0].imshow(data_train[i].reshape(28, 28),
                  cmap=plt.cm.gray)
    axes[i, 1].imshow(data_train_noisy[i].reshape(28, 28),
                  cmap=plt.cm.gray)
    for ax, col in zip(axes[0], cols):
      ax.set_title(col)
    for j in range(2):
      axes[i, j].get_xaxis().set_visible(False)
      axes[i, j].get_yaxis().set_visible(False)
fig.tight_layout()
plt.show()
```


Input
Noisy Input

1.6 Denoising Autoencoder

[0]:
```
dae = convolutional_autoencoder('binary_crossentropy', 'adam', 28)
```

[58]:
```
dae_history = dae.fit((data_train_noisy, data_train),
                      (data_test_noisy, data_test),
                      n_epochs=150,
                      batch_size=1024)
```

```
Train on 60000 samples, validate on 10000 samples
Epoch 1/150
60000/60000 [==============================] - 4s 67us/step - loss: 0.5769 -
val_loss: 0.4620
Epoch 2/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.3768 -
val_loss: 0.2586
Epoch 3/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.1919 -
val_loss: 0.1667
Epoch 4/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.1545 -
val_loss: 0.1379
Epoch 5/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.1262 -
val_loss: 0.1160
Epoch 6/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.1135 -
val_loss: 0.1090
Epoch 7/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.1084 -
val_loss: 0.1053
Epoch 8/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.1053 -
val_loss: 0.1028
Epoch 9/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.1032 -
val_loss: 0.1011
Epoch 10/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.1016 -
val_loss: 0.0998
Epoch 11/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.1004 -
val_loss: 0.0986
Epoch 12/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0994 -
```

```
val_loss: 0.0977
Epoch 13/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0985 -
val_loss: 0.0969
Epoch 14/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0978 -
val_loss: 0.0962
Epoch 15/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0972 -
val_loss: 0.0958
Epoch 16/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0967 -
val_loss: 0.0951
Epoch 17/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0961 -
val_loss: 0.0948
Epoch 18/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0957 -
val_loss: 0.0945
Epoch 19/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0953 -
val_loss: 0.0938
Epoch 20/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0948 -
val_loss: 0.0934
Epoch 21/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0944 -
val_loss: 0.0930
Epoch 22/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0941 -
val_loss: 0.0926
Epoch 23/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0937 -
val_loss: 0.0925
Epoch 24/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0933 -
val_loss: 0.0920
Epoch 25/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0930 -
val_loss: 0.0918
Epoch 26/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0926 -
val_loss: 0.0913
Epoch 27/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0923 -
val_loss: 0.0911
Epoch 28/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0921 -
```

```
val_loss: 0.0908
Epoch 29/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0918 -
val_loss: 0.0906
Epoch 30/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0916 -
val_loss: 0.0903
Epoch 31/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0913 -
val_loss: 0.0901
Epoch 32/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0910 -
val_loss: 0.0899
Epoch 33/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0908 -
val_loss: 0.0896
Epoch 34/150
60000/60000 [==============================] - 3s 46us/step - loss: 0.0906 -
val_loss: 0.0894
Epoch 35/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0904 -
val_loss: 0.0895
Epoch 36/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0902 -
val_loss: 0.0890
Epoch 37/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0899 -
val_loss: 0.0887
Epoch 38/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0897 -
val_loss: 0.0887
Epoch 39/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0896 -
val_loss: 0.0884
Epoch 40/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0894 -
val_loss: 0.0882
Epoch 41/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0891 -
val_loss: 0.0881
Epoch 42/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0890 -
val_loss: 0.0878
Epoch 43/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0888 -
val_loss: 0.0877
Epoch 44/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0886 -
```

```
val_loss: 0.0875
Epoch 45/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0885 -
val_loss: 0.0875
Epoch 46/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0882 -
val_loss: 0.0872
Epoch 47/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0881 -
val_loss: 0.0870
Epoch 48/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0879 -
val_loss: 0.0868
Epoch 49/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0878 -
val_loss: 0.0866
Epoch 50/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0876 -
val_loss: 0.0865
Epoch 51/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0874 -
val_loss: 0.0863
Epoch 52/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0873 -
val_loss: 0.0862
Epoch 53/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0871 -
val_loss: 0.0861
Epoch 54/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0870 -
val_loss: 0.0859
Epoch 55/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0868 -
val_loss: 0.0859
Epoch 56/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0868 -
val_loss: 0.0858
Epoch 57/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0866 -
val_loss: 0.0855
Epoch 58/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0865 -
val_loss: 0.0855
Epoch 59/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0864 -
val_loss: 0.0854
Epoch 60/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0862 -
```

```
val_loss: 0.0852
Epoch 61/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0861 -
val_loss: 0.0851
Epoch 62/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0861 -
val_loss: 0.0850
Epoch 63/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0859 -
val_loss: 0.0849
Epoch 64/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0858 -
val_loss: 0.0851
Epoch 65/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0857 -
val_loss: 0.0847
Epoch 66/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0857 -
val_loss: 0.0846
Epoch 67/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0855 -
val_loss: 0.0846
Epoch 68/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0855 -
val_loss: 0.0845
Epoch 69/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0854 -
val_loss: 0.0844
Epoch 70/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0853 -
val_loss: 0.0844
Epoch 71/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0853 -
val_loss: 0.0843
Epoch 72/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0851 -
val_loss: 0.0843
Epoch 73/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0851 -
val_loss: 0.0842
Epoch 74/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0850 -
val_loss: 0.0840
Epoch 75/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0850 -
val_loss: 0.0840
Epoch 76/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0848 -
```

```
val_loss: 0.0840
Epoch 77/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0848 -
val_loss: 0.0839
Epoch 78/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0848 -
val_loss: 0.0838
Epoch 79/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0847 -
val_loss: 0.0838
Epoch 80/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0846 -
val_loss: 0.0837
Epoch 81/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0845 -
val_loss: 0.0837
Epoch 82/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0845 -
val_loss: 0.0838
Epoch 83/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0845 -
val_loss: 0.0835
Epoch 84/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0844 -
val_loss: 0.0835
Epoch 85/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0844 -
val_loss: 0.0835
Epoch 86/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0843 -
val_loss: 0.0834
Epoch 87/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0842 -
val_loss: 0.0833
Epoch 88/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0842 -
val_loss: 0.0833
Epoch 89/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0841 -
val_loss: 0.0833
Epoch 90/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0841 -
val_loss: 0.0832
Epoch 91/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0840 -
val_loss: 0.0831
Epoch 92/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0841 -
```

```
val_loss: 0.0831
Epoch 93/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0839 -
val_loss: 0.0831
Epoch 94/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0839 -
val_loss: 0.0831
Epoch 95/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0839 -
val_loss: 0.0830
Epoch 96/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0838 -
val_loss: 0.0829
Epoch 97/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0838 -
val_loss: 0.0829
Epoch 98/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0838 -
val_loss: 0.0829
Epoch 99/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0837 -
val_loss: 0.0829
Epoch 100/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0836 -
val_loss: 0.0828
Epoch 101/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0836 -
val_loss: 0.0827
Epoch 102/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0836 -
val_loss: 0.0827
Epoch 103/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0836 -
val_loss: 0.0828
Epoch 104/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0835 -
val_loss: 0.0826
Epoch 105/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0835 -
val_loss: 0.0826
Epoch 106/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0834 -
val_loss: 0.0827
Epoch 107/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0834 -
val_loss: 0.0825
Epoch 108/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0833 -
```

```
val_loss: 0.0825
Epoch 109/150
60000/60000 [==============================] - 3s 49us/step - loss: 0.0833 -
val_loss: 0.0825
Epoch 110/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0833 -
val_loss: 0.0824
Epoch 111/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0832 -
val_loss: 0.0824
Epoch 112/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0833 -
val_loss: 0.0825
Epoch 113/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0832 -
val_loss: 0.0823
Epoch 114/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0831 -
val_loss: 0.0823
Epoch 115/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0831 -
val_loss: 0.0823
Epoch 116/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0831 -
val_loss: 0.0822
Epoch 117/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0831 -
val_loss: 0.0822
Epoch 118/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0830 -
val_loss: 0.0822
Epoch 119/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0830 -
val_loss: 0.0822
Epoch 120/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0829 -
val_loss: 0.0821
Epoch 121/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0829 -
val_loss: 0.0821
Epoch 122/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0829 -
val_loss: 0.0822
Epoch 123/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0829 -
val_loss: 0.0820
Epoch 124/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0829 -
```

```
val_loss: 0.0823
Epoch 125/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0829 -
val_loss: 0.0821
Epoch 126/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0828 -
val_loss: 0.0819
Epoch 127/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0828 -
val_loss: 0.0820
Epoch 128/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0827 -
val_loss: 0.0819
Epoch 129/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0828 -
val_loss: 0.0820
Epoch 130/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0828 -
val_loss: 0.0818
Epoch 131/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0827 -
val_loss: 0.0818
Epoch 132/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0827 -
val_loss: 0.0818
Epoch 133/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0826 -
val_loss: 0.0818
Epoch 134/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0826 -
val_loss: 0.0818
Epoch 135/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0826 -
val_loss: 0.0817
Epoch 136/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0826 -
val_loss: 0.0817
Epoch 137/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0826 -
val_loss: 0.0820
Epoch 138/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0826 -
val_loss: 0.0817
Epoch 139/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0825 -
val_loss: 0.0817
Epoch 140/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0825 -
```

```
val_loss: 0.0817
Epoch 141/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0825 -
val_loss: 0.0817
Epoch 142/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0825 -
val_loss: 0.0817
Epoch 143/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0824 -
val_loss: 0.0816
Epoch 144/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0824 -
val_loss: 0.0816
Epoch 145/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0824 -
val_loss: 0.0816
Epoch 146/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0824 -
val_loss: 0.0816
Epoch 147/150
60000/60000 [==============================] - 3s 49us/step - loss: 0.0824 -
val_loss: 0.0815
Epoch 148/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0824 -
val_loss: 0.0815
Epoch 149/150
60000/60000 [==============================] - 3s 48us/step - loss: 0.0823 -
val_loss: 0.0815
Epoch 150/150
60000/60000 [==============================] - 3s 47us/step - loss: 0.0823 -
val_loss: 0.0815
```

[0]:
```
dae.save_model_weights("dae_model_weights.h5")
dae.save_model("dae_model.json")
dae.save_model_history(dae_history, "dae_model_history.pkl")
```

[0]:
```
! cp -r dae_model.json dae_model_weights.h5 dae_model_history.pkl ./gdrive/My\␣
↪Drive/ece595_ml2/models/
```

[0]:
```
dae_model = dae.load_model("dae_model.json", "dae_model_weights.h5")
```

[0]:
```
dae_reconstructions = dae_model.predict(data_test_noisy)
```

[63]:
```
print(len(dae_model.layers))
print(dae_model.summary())
```

```
9
Model: "sequential_2"
_________________________________________________________________
```

```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_6 (Conv2D)            (None, 28, 28, 16)        160
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 14, 14, 16)        0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 14, 14, 8)         1160
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 7, 7, 8)           0
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 7, 7, 8)           584
_________________________________________________________________
up_sampling2d_3 (UpSampling2 (None, 14, 14, 8)         0
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 14, 14, 16)        1168
_________________________________________________________________
up_sampling2d_4 (UpSampling2 (None, 28, 28, 16)        0
_________________________________________________________________
conv2d_10 (Conv2D)           (None, 28, 28, 1)         145
=================================================================
Total params: 3,217
Trainable params: 3,217
Non-trainable params: 0
_________________________________________________________________
None
```

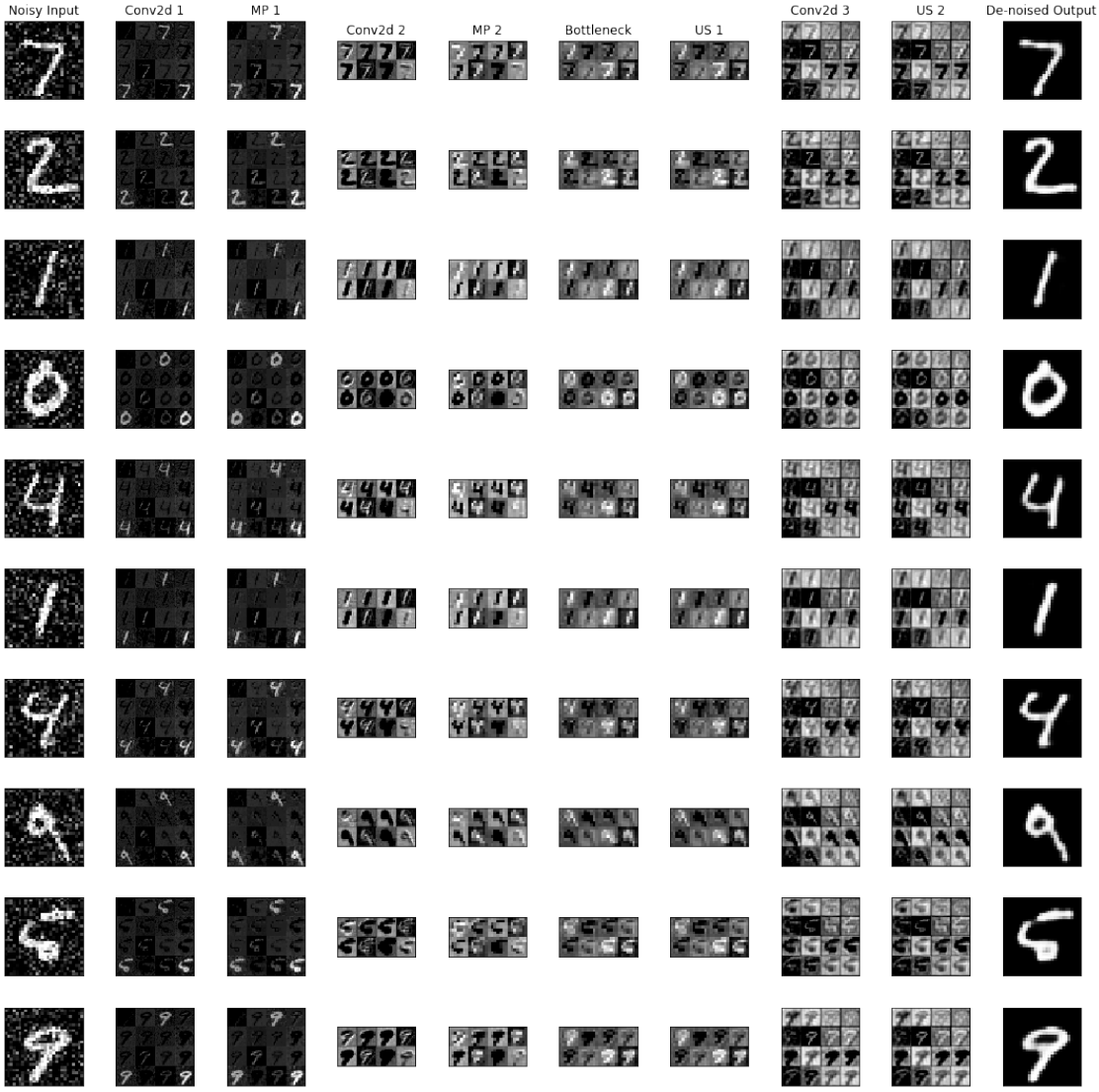
[0]:
```
dae_w_hl = get_hidden_layers_representation(dae_model, data_test_noisy)
```

1.7 Layer Representation

[0]:
```
n_samples = 10
n_layers = len(dae_model.layers)
layers = [' Noisy Input ',
          ' Conv2d 1 ',
          ' MP 1 ',
          ' Conv2d 2 ',
          ' MP 2 ',
          ' Bottleneck ',
          ' US 1 ',
          ' Conv2d 3 ',
          ' US 2 ',
          ' De-noised Output ']
cols = ['{}'.format(col) for col in layers]
```

[0]:
```
n_kernels = [dae_w_hl[i][0][0].shape[2] for i in range(1, n_layers)]
```

[71]:
```
fig, axes = plt.subplots(nrows=n_samples, ncols=n_layers+1, figsize=(15, 15))
for i in range(n_samples):
    axes[i, 0].imshow(data_test_noisy[i].reshape(28, 28),
                  cmap=plt.cm.gray)
    for j in range(1, n_layers + 1, 1):
      temp1 = hidden_layer_weights(dae_w_hl[j][0][i])
      axes[i, j].imshow(temp1, cmap=plt.cm.gray)
    for ax, col in zip(axes[0], cols):
      ax.set_title(col)
    for j in range(n_layers+1):
      axes[i, j].get_xaxis().set_visible(False)
      axes[i, j].get_yaxis().set_visible(False)
fig.tight_layout()
plt.show()
```



1.8 Loss Curves

[72]:
```
plt.plot(dae_history.history['loss'])
plt.plot(dae_history.history['val_loss'])
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.title("Loss v/s Epochs")
plt.legend(['train', 'test'], loc='upper right')
plt.show()
```