bonus_exercise_1

October 4, 2019

1 Student Name: Murali Krishnan Rajasekharan Pillai

2 ECE 595 Machine Learning II

3 Bonus Exercise : Visualizing the Percent Activating Units in Hidden Layers

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

```
Fri Oct  4 19:13:16 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   38C    P8    26W / 149W |      0MiB / 11441MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

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

[3]:
```
from google.colab import drive
drive.mount('/content/gdrive', force_remount=True)
MODEL_LOCATION = "gdrive/My Drive/ece595_ml2/models/"
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id
=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com&redire
ct_uri=urn%3Aietf%3Awg%3Aoauth%3A2.0%3Aoob&scope=email%20https%3A%2F%2Fwww.googl
eapis.com%2Fauth%2Fdocs.test%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdrive%2
```

```
0https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdrive.photos.readonly%20https%3A%2F%2
Fwww.googleapis.com%2Fauth%2Fpeopleapi.readonly&response_type=code

Enter your authorization code:
··········
Mounted at /content/gdrive
```

[4]:
```
import keras
import numpy as np
from keras.datasets import mnist
from keras.models import Sequential, model_from_json
from keras.layers import Dense, Conv2D, Dropout, Flatten, MaxPool2D
from keras.optimizers import Adam
from keras import backend as K
import matplotlib.pyplot as plt
import pickle
```

```
Using TensorFlow backend.
```

4 Part 0 : Importing and Normalizing MNIST Data

[5]:
```
# has shape (num_samples, 28, 28)
(data_train, labels_train), (data_test, labels_test) = mnist.load_data()
data_train = data_train / 255.0
data_test = data_test / 255.0
data_train = data_train.reshape(60000, 28, 28, 1)
data_test = data_test.reshape(10000, 28, 28, 1)
```

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

[0]:
```
# Create labels as one-hot encoders
labels_train = keras.utils.np_utils.to_categorical(labels_train, num_classes=10)
labels_test = keras.utils.np_utils.to_categorical(labels_test, num_classes=10)
```

5 Defining a CNN

[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 CNN_model(model_methods):
  def __init__(self, loss_fn, optim, ndim, num_classes):
    super().__init__(loss_fn, optim, ndim)
    self.num_classes = num_classes
    self.model = self._build_model()
```

```
def _build_model(self):
  """
  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()
  model.add(Conv2D(16,
                   (3, 3),
                   activation='relu',
                   padding='same',
                   input_shape=(self.ndim, self.ndim, 1)))
  model.add(Dropout(0.30))
  model.add(MaxPool2D(pool_size=(2, 2)))
  model.add(Conv2D(8,
                   (3, 3),
                   padding='same',
                   activation='relu'))
  model.add(Dropout(0.30))
  model.add(MaxPool2D(pool_size=(2, 2)))
  model.add(Flatten())
  model.add(Dense(50,
                  activation='relu',
                  kernel_initializer='normal'))
  model.add(Dense(50,
                  activation='relu',
                  kernel_initializer='normal'))
  model.add(Dense(self.num_classes,
                  activation='softmax'))
  model.compile(loss=self.loss_fn,
                optimizer=self.optim,
                metrics=['accuracy'])
  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
```

[9]:
```
cnn = CNN_model('categorical_crossentropy', 'adam', 28, 10)
```

```
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:148: The name
tf.placeholder_with_default is deprecated. Please use
tf.compat.v1.placeholder_with_default instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:3733: calling dropout (from
tensorflow.python.ops.nn_ops) with keep_prob is deprecated and will be removed
in a future version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 -
keep_prob`.
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:4409: The name tf.random_normal is
deprecated. Please use tf.random.normal 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.
```

[10]:
```
cnn_history = cnn.fit((data_train, labels_train),
                      (data_test, labels_test),
                      n_epochs=150,
                      batch_size=1024)
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/tensorflow/python/ops/math_grad.py:1250:
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
Train on 60000 samples, validate on 10000 samples
Epoch 1/150
60000/60000 [==============================] - 10s 165us/step - loss: 1.3533 -
acc: 0.6072 - val_loss: 0.6247 - val_acc: 0.8424
Epoch 2/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.3714 -
acc: 0.8850 - val_loss: 0.3744 - val_acc: 0.9242
Epoch 3/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.2537 -
acc: 0.9228 - val_loss: 0.2783 - val_acc: 0.9461
Epoch 4/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.1958 -
acc: 0.9389 - val_loss: 0.2324 - val_acc: 0.9542
Epoch 5/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.1552 -
acc: 0.9527 - val_loss: 0.1899 - val_acc: 0.9644
Epoch 6/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.1350 -
acc: 0.9579 - val_loss: 0.1606 - val_acc: 0.9677
Epoch 7/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.1183 -
acc: 0.9636 - val_loss: 0.1508 - val_acc: 0.9718
Epoch 8/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.1067 -
acc: 0.9666 - val_loss: 0.1195 - val_acc: 0.9774
Epoch 9/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.1007 -
acc: 0.9680 - val_loss: 0.1184 - val_acc: 0.9769
```

```
Epoch 10/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0923 -
acc: 0.9712 - val_loss: 0.1085 - val_acc: 0.9812
Epoch 11/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0876 -
acc: 0.9724 - val_loss: 0.1013 - val_acc: 0.9799
Epoch 12/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0810 -
acc: 0.9746 - val_loss: 0.0961 - val_acc: 0.9808
Epoch 13/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0791 -
acc: 0.9751 - val_loss: 0.0876 - val_acc: 0.9841
Epoch 14/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0737 -
acc: 0.9773 - val_loss: 0.0814 - val_acc: 0.9860
Epoch 15/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0697 -
acc: 0.9776 - val_loss: 0.0776 - val_acc: 0.9849
Epoch 16/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0690 -
acc: 0.9784 - val_loss: 0.0766 - val_acc: 0.9857
Epoch 17/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0671 -
acc: 0.9787 - val_loss: 0.0726 - val_acc: 0.9868
Epoch 18/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0639 -
acc: 0.9800 - val_loss: 0.0763 - val_acc: 0.9851
Epoch 19/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0580 -
acc: 0.9819 - val_loss: 0.0686 - val_acc: 0.9862
Epoch 20/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0584 -
acc: 0.9820 - val_loss: 0.0707 - val_acc: 0.9859
Epoch 21/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0572 -
acc: 0.9815 - val_loss: 0.0647 - val_acc: 0.9875
Epoch 22/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0568 -
acc: 0.9820 - val_loss: 0.0660 - val_acc: 0.9873
Epoch 23/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0534 -
acc: 0.9828 - val_loss: 0.0653 - val_acc: 0.9867
Epoch 24/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0527 -
acc: 0.9834 - val_loss: 0.0636 - val_acc: 0.9869
Epoch 25/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0528 -
acc: 0.9834 - val_loss: 0.0710 - val_acc: 0.9860
```

```
Epoch 26/150
60000/60000 [==============================] - 2s 38us/step - loss: 0.0512 -
acc: 0.9838 - val_loss: 0.0624 - val_acc: 0.9874
Epoch 27/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0525 -
acc: 0.9833 - val_loss: 0.0613 - val_acc: 0.9865
Epoch 28/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0482 -
acc: 0.9845 - val_loss: 0.0543 - val_acc: 0.9875
Epoch 29/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0459 -
acc: 0.9848 - val_loss: 0.0573 - val_acc: 0.9873
Epoch 30/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0455 -
acc: 0.9855 - val_loss: 0.0521 - val_acc: 0.9890
Epoch 31/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0473 -
acc: 0.9847 - val_loss: 0.0568 - val_acc: 0.9880
Epoch 32/150
60000/60000 [==============================] - 2s 38us/step - loss: 0.0442 -
acc: 0.9859 - val_loss: 0.0556 - val_acc: 0.9877
Epoch 33/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0434 -
acc: 0.9863 - val_loss: 0.0540 - val_acc: 0.9879
Epoch 34/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0415 -
acc: 0.9861 - val_loss: 0.0506 - val_acc: 0.9884
Epoch 35/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0418 -
acc: 0.9865 - val_loss: 0.0471 - val_acc: 0.9888
Epoch 36/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0415 -
acc: 0.9862 - val_loss: 0.0494 - val_acc: 0.9884
Epoch 37/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0402 -
acc: 0.9870 - val_loss: 0.0525 - val_acc: 0.9880
Epoch 38/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0397 -
acc: 0.9870 - val_loss: 0.0525 - val_acc: 0.9875
Epoch 39/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0375 -
acc: 0.9880 - val_loss: 0.0449 - val_acc: 0.9895
Epoch 40/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0388 -
acc: 0.9870 - val_loss: 0.0492 - val_acc: 0.9888
Epoch 41/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0361 -
acc: 0.9882 - val_loss: 0.0478 - val_acc: 0.9892
```

```
Epoch 42/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0383 -
acc: 0.9869 - val_loss: 0.0433 - val_acc: 0.9898
Epoch 43/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0357 -
acc: 0.9884 - val_loss: 0.0428 - val_acc: 0.9901
Epoch 44/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0353 -
acc: 0.9888 - val_loss: 0.0432 - val_acc: 0.9898
Epoch 45/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0355 -
acc: 0.9880 - val_loss: 0.0436 - val_acc: 0.9894
Epoch 46/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0333 -
acc: 0.9888 - val_loss: 0.0426 - val_acc: 0.9903
Epoch 47/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0347 -
acc: 0.9883 - val_loss: 0.0436 - val_acc: 0.9890
Epoch 48/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0323 -
acc: 0.9896 - val_loss: 0.0432 - val_acc: 0.9900
Epoch 49/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0345 -
acc: 0.9885 - val_loss: 0.0416 - val_acc: 0.9895
Epoch 50/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0327 -
acc: 0.9899 - val_loss: 0.0441 - val_acc: 0.9897
Epoch 51/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0331 -
acc: 0.9894 - val_loss: 0.0401 - val_acc: 0.9902
Epoch 52/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0303 -
acc: 0.9899 - val_loss: 0.0383 - val_acc: 0.9907
Epoch 53/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0318 -
acc: 0.9893 - val_loss: 0.0461 - val_acc: 0.9886
Epoch 54/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0304 -
acc: 0.9899 - val_loss: 0.0354 - val_acc: 0.9906
Epoch 55/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0314 -
acc: 0.9895 - val_loss: 0.0394 - val_acc: 0.9893
Epoch 56/150
60000/60000 [==============================] - 2s 38us/step - loss: 0.0294 -
acc: 0.9905 - val_loss: 0.0357 - val_acc: 0.9909
Epoch 57/150
60000/60000 [==============================] - 2s 38us/step - loss: 0.0296 -
acc: 0.9903 - val_loss: 0.0417 - val_acc: 0.9911
```

```
Epoch 58/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0304 -
acc: 0.9894 - val_loss: 0.0419 - val_acc: 0.9896
Epoch 59/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0279 -
acc: 0.9910 - val_loss: 0.0349 - val_acc: 0.9910
Epoch 60/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0309 -
acc: 0.9897 - val_loss: 0.0375 - val_acc: 0.9902
Epoch 61/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0284 -
acc: 0.9902 - val_loss: 0.0348 - val_acc: 0.9905
Epoch 62/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0283 -
acc: 0.9905 - val_loss: 0.0390 - val_acc: 0.9903
Epoch 63/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0276 -
acc: 0.9908 - val_loss: 0.0359 - val_acc: 0.9900
Epoch 64/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0275 -
acc: 0.9905 - val_loss: 0.0396 - val_acc: 0.9900
Epoch 65/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0270 -
acc: 0.9910 - val_loss: 0.0356 - val_acc: 0.9892
Epoch 66/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0275 -
acc: 0.9907 - val_loss: 0.0383 - val_acc: 0.9902
Epoch 67/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0268 -
acc: 0.9910 - val_loss: 0.0347 - val_acc: 0.9905
Epoch 68/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0254 -
acc: 0.9914 - val_loss: 0.0390 - val_acc: 0.9897
Epoch 69/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0250 -
acc: 0.9918 - val_loss: 0.0300 - val_acc: 0.9911
Epoch 70/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0252 -
acc: 0.9916 - val_loss: 0.0371 - val_acc: 0.9892
Epoch 71/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0247 -
acc: 0.9915 - val_loss: 0.0328 - val_acc: 0.9910
Epoch 72/150
60000/60000 [==============================] - 2s 38us/step - loss: 0.0248 -
acc: 0.9913 - val_loss: 0.0323 - val_acc: 0.9916
Epoch 73/150
60000/60000 [==============================] - 2s 38us/step - loss: 0.0242 -
acc: 0.9915 - val_loss: 0.0332 - val_acc: 0.9908
```

```
Epoch 74/150
60000/60000 [==============================] - 2s 38us/step - loss: 0.0254 -
acc: 0.9910 - val_loss: 0.0304 - val_acc: 0.9915
Epoch 75/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0234 -
acc: 0.9918 - val_loss: 0.0345 - val_acc: 0.9903
Epoch 76/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0224 -
acc: 0.9924 - val_loss: 0.0330 - val_acc: 0.9906
Epoch 77/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0231 -
acc: 0.9919 - val_loss: 0.0331 - val_acc: 0.9907
Epoch 78/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0231 -
acc: 0.9919 - val_loss: 0.0310 - val_acc: 0.9905
Epoch 79/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0232 -
acc: 0.9922 - val_loss: 0.0300 - val_acc: 0.9912
Epoch 80/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0228 -
acc: 0.9925 - val_loss: 0.0361 - val_acc: 0.9897
Epoch 81/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0220 -
acc: 0.9925 - val_loss: 0.0313 - val_acc: 0.9909
Epoch 82/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0226 -
acc: 0.9923 - val_loss: 0.0315 - val_acc: 0.9904
Epoch 83/150
60000/60000 [==============================] - 2s 41us/step - loss: 0.0225 -
acc: 0.9921 - val_loss: 0.0310 - val_acc: 0.9917
Epoch 84/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0217 -
acc: 0.9923 - val_loss: 0.0313 - val_acc: 0.9912
Epoch 85/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0216 -
acc: 0.9927 - val_loss: 0.0319 - val_acc: 0.9911
Epoch 86/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0207 -
acc: 0.9928 - val_loss: 0.0310 - val_acc: 0.9897
Epoch 87/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0211 -
acc: 0.9928 - val_loss: 0.0295 - val_acc: 0.9914
Epoch 88/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0201 -
acc: 0.9931 - val_loss: 0.0352 - val_acc: 0.9903
Epoch 89/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0208 -
acc: 0.9930 - val_loss: 0.0305 - val_acc: 0.9911
```

```
Epoch 90/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0212 -
acc: 0.9929 - val_loss: 0.0327 - val_acc: 0.9907
Epoch 91/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0206 -
acc: 0.9930 - val_loss: 0.0322 - val_acc: 0.9904
Epoch 92/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0204 -
acc: 0.9929 - val_loss: 0.0285 - val_acc: 0.9919
Epoch 93/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0208 -
acc: 0.9929 - val_loss: 0.0293 - val_acc: 0.9906
Epoch 94/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0192 -
acc: 0.9933 - val_loss: 0.0307 - val_acc: 0.9911
Epoch 95/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0194 -
acc: 0.9930 - val_loss: 0.0295 - val_acc: 0.9910
Epoch 96/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0197 -
acc: 0.9935 - val_loss: 0.0312 - val_acc: 0.9900
Epoch 97/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0189 -
acc: 0.9933 - val_loss: 0.0321 - val_acc: 0.9897
Epoch 98/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0193 -
acc: 0.9931 - val_loss: 0.0289 - val_acc: 0.9914
Epoch 99/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0181 -
acc: 0.9939 - val_loss: 0.0323 - val_acc: 0.9901
Epoch 100/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0180 -
acc: 0.9940 - val_loss: 0.0303 - val_acc: 0.9903
Epoch 101/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0164 -
acc: 0.9943 - val_loss: 0.0322 - val_acc: 0.9900
Epoch 102/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0178 -
acc: 0.9940 - val_loss: 0.0316 - val_acc: 0.9905
Epoch 103/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0181 -
acc: 0.9940 - val_loss: 0.0313 - val_acc: 0.9903
Epoch 104/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0193 -
acc: 0.9937 - val_loss: 0.0306 - val_acc: 0.9905
Epoch 105/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0198 -
acc: 0.9935 - val_loss: 0.0266 - val_acc: 0.9920
```

```
Epoch 106/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0191 -
acc: 0.9937 - val_loss: 0.0330 - val_acc: 0.9903
Epoch 107/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0173 -
acc: 0.9940 - val_loss: 0.0308 - val_acc: 0.9909
Epoch 108/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0178 -
acc: 0.9939 - val_loss: 0.0308 - val_acc: 0.9909
Epoch 109/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0186 -
acc: 0.9938 - val_loss: 0.0302 - val_acc: 0.9907
Epoch 110/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0177 -
acc: 0.9941 - val_loss: 0.0302 - val_acc: 0.9910
Epoch 111/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0172 -
acc: 0.9939 - val_loss: 0.0332 - val_acc: 0.9897
Epoch 112/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0176 -
acc: 0.9942 - val_loss: 0.0271 - val_acc: 0.9909
Epoch 113/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0180 -
acc: 0.9936 - val_loss: 0.0315 - val_acc: 0.9910
Epoch 114/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0171 -
acc: 0.9944 - val_loss: 0.0277 - val_acc: 0.9915
Epoch 115/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0160 -
acc: 0.9947 - val_loss: 0.0272 - val_acc: 0.9921
Epoch 116/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0180 -
acc: 0.9942 - val_loss: 0.0298 - val_acc: 0.9909
Epoch 117/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0174 -
acc: 0.9939 - val_loss: 0.0314 - val_acc: 0.9906
Epoch 118/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0179 -
acc: 0.9937 - val_loss: 0.0282 - val_acc: 0.9912
Epoch 119/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0161 -
acc: 0.9947 - val_loss: 0.0268 - val_acc: 0.9920
Epoch 120/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0166 -
acc: 0.9945 - val_loss: 0.0292 - val_acc: 0.9910
Epoch 121/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0166 -
acc: 0.9941 - val_loss: 0.0268 - val_acc: 0.9909
```

```
Epoch 122/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0148 -
acc: 0.9950 - val_loss: 0.0257 - val_acc: 0.9922
Epoch 123/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0155 -
acc: 0.9947 - val_loss: 0.0277 - val_acc: 0.9911
Epoch 124/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0154 -
acc: 0.9950 - val_loss: 0.0316 - val_acc: 0.9904
Epoch 125/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0164 -
acc: 0.9943 - val_loss: 0.0278 - val_acc: 0.9907
Epoch 126/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0162 -
acc: 0.9948 - val_loss: 0.0318 - val_acc: 0.9908
Epoch 127/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0150 -
acc: 0.9949 - val_loss: 0.0287 - val_acc: 0.9910
Epoch 128/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0141 -
acc: 0.9952 - val_loss: 0.0302 - val_acc: 0.9907
Epoch 129/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0150 -
acc: 0.9947 - val_loss: 0.0268 - val_acc: 0.9915
Epoch 130/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0141 -
acc: 0.9952 - val_loss: 0.0289 - val_acc: 0.9903
Epoch 131/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0147 -
acc: 0.9948 - val_loss: 0.0263 - val_acc: 0.9915
Epoch 132/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0150 -
acc: 0.9948 - val_loss: 0.0277 - val_acc: 0.9912
Epoch 133/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0136 -
acc: 0.9952 - val_loss: 0.0273 - val_acc: 0.9917
Epoch 134/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0141 -
acc: 0.9952 - val_loss: 0.0285 - val_acc: 0.9913
Epoch 135/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0158 -
acc: 0.9945 - val_loss: 0.0293 - val_acc: 0.9911
Epoch 136/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0147 -
acc: 0.9947 - val_loss: 0.0265 - val_acc: 0.9920
Epoch 137/150
60000/60000 [==============================] - 2s 41us/step - loss: 0.0151 -
acc: 0.9948 - val_loss: 0.0268 - val_acc: 0.9916
```

```
Epoch 138/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0135 -
acc: 0.9955 - val_loss: 0.0264 - val_acc: 0.9915
Epoch 139/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0134 -
acc: 0.9956 - val_loss: 0.0271 - val_acc: 0.9912
Epoch 140/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0144 -
acc: 0.9949 - val_loss: 0.0255 - val_acc: 0.9916
Epoch 141/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0132 -
acc: 0.9954 - val_loss: 0.0283 - val_acc: 0.9911
Epoch 142/150
60000/60000 [==============================] - 2s 41us/step - loss: 0.0155 -
acc: 0.9948 - val_loss: 0.0270 - val_acc: 0.9914
Epoch 143/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0149 -
acc: 0.9949 - val_loss: 0.0280 - val_acc: 0.9911
Epoch 144/150
60000/60000 [==============================] - 2s 39us/step - loss: 0.0147 -
acc: 0.9949 - val_loss: 0.0262 - val_acc: 0.9915
Epoch 145/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0139 -
acc: 0.9952 - val_loss: 0.0337 - val_acc: 0.9893
Epoch 146/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0142 -
acc: 0.9953 - val_loss: 0.0249 - val_acc: 0.9918
Epoch 147/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0134 -
acc: 0.9957 - val_loss: 0.0274 - val_acc: 0.9915
Epoch 148/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0138 -
acc: 0.9951 - val_loss: 0.0268 - val_acc: 0.9917
Epoch 149/150
60000/60000 [==============================] - 2s 40us/step - loss: 0.0136 -
acc: 0.9954 - val_loss: 0.0265 - val_acc: 0.9914
Epoch 150/150
60000/60000 [==============================] - 2s 41us/step - loss: 0.0131 -
acc: 0.9952 - val_loss: 0.0254 - val_acc: 0.9914
```

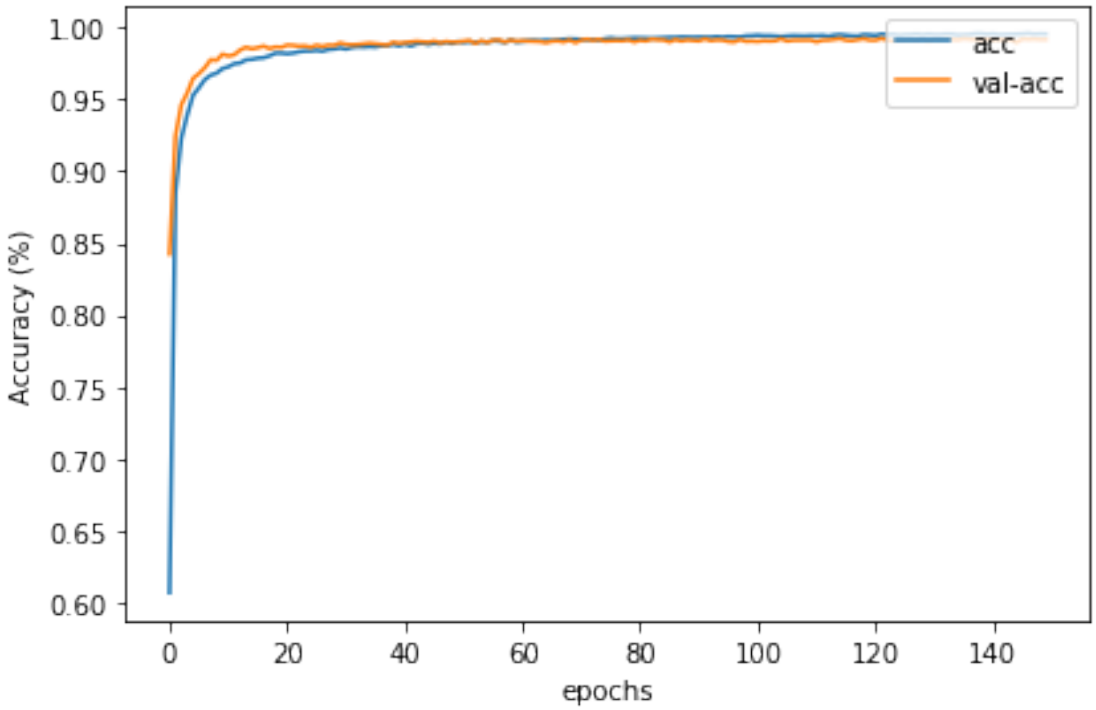
[0]:
```
cnn.save_model_weights("bonus_cnn_model_weights.h5")
cnn.save_model("bonus_cnn_model.json")
cnn.save_model_history(cnn_history, "bonus_cnn_model_history.pkl")
```

[0]:
```
! cp -r bonus_cnn_model.json bonus_cnn_model_weights.h5 bonus_cnn_model_history.
 ↪pkl ./gdrive/My\ Drive/ece595_ml2/models/
```

[0]:
```
cnn_model = cnn.load_model("bonus_cnn_model.json", "bonus_cnn_model_weights.h5")
```

[0]:
```
cnn_reconstructions = cnn_model.predict(data_test)
```

[15]:
```
plt.plot(cnn_history.history['acc'])
plt.plot(cnn_history.history['val_acc'])
plt.legend(['acc', 'val-acc'], loc='upper right')
plt.xlabel("epochs")
plt.ylabel(r"Accuracy (%)")
plt.tight_layout()
plt.show()
```



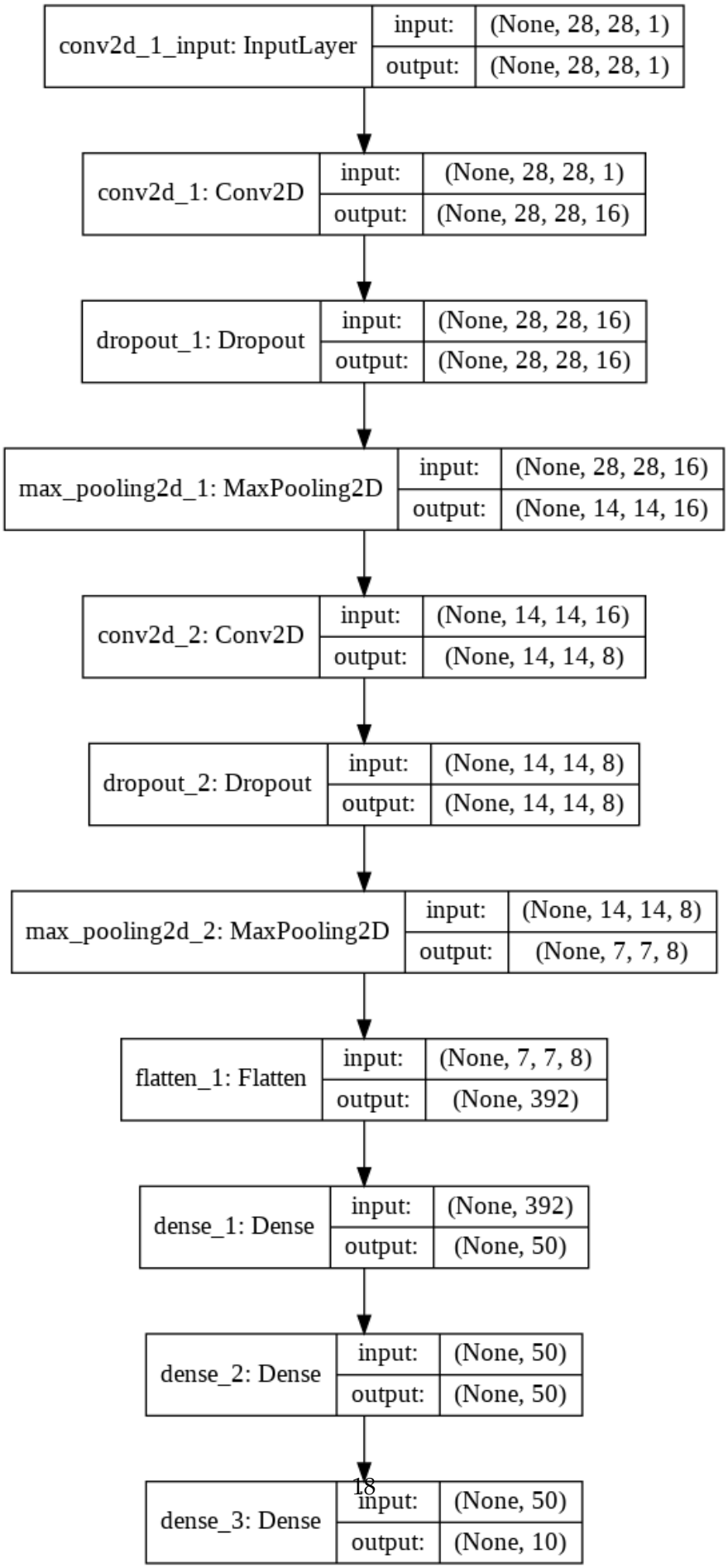
[16]:
```
print(len(cnn_model.layers))
print(cnn_model.summary())
```

```
10
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 28, 28, 16)        160
_________________________________________________________________
dropout_1 (Dropout)          (None, 28, 28, 16)        0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 14, 14, 16)        0
_________________________________________________________________
```

```
conv2d_2 (Conv2D)            (None, 14, 14, 8)         1160
_________________________________________________________________
dropout_2 (Dropout)          (None, 14, 14, 8)         0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 7, 7, 8)           0
_________________________________________________________________
flatten_1 (Flatten)          (None, 392)               0
_________________________________________________________________
dense_1 (Dense)              (None, 50)                19650
_________________________________________________________________
dense_2 (Dense)              (None, 50)                2550
_________________________________________________________________
dense_3 (Dense)              (None, 10)                510
=================================================================
Total params: 24,030
Trainable params: 24,030
Non-trainable params: 0
_________________________________________________________________
None
```

[17]:
```
from keras.utils.vis_utils import plot_model
plot_model(cnn_model, to_file='model_plot.png', show_shapes=True,␣
 ↪show_layer_names=True)
```

[17]:


conv2d_1_input: InputLayer
input: (None, 28, 28, 1)
output: (None, 28, 28, 1)
conv2d_1: Conv2D
input: (None, 28, 28, 1)
output: (None, 28, 28, 16)
dropout_1: Dropout
input: (None, 28, 28, 16)
output: (None, 28, 28, 16)
max_pooling2d_1: MaxPooling2D
input: (None, 28, 28, 16)
output: (None, 14, 14, 16)
conv2d_2: Conv2D
input: (None, 14, 14, 16)
output: (None, 14, 14, 8)
dropout_2: Dropout
input: (None, 14, 14, 8)
output: (None, 14, 14, 8)
max_pooling2d_2: MaxPooling2D
input: (None, 14, 14, 8)
output: (None, 7, 7, 8)
flatten_1: Flatten
input: (None, 7, 7, 8)
output: (None, 392)
dense_1: Dense
input: (None, 392)
output: (None, 50)
dense_2: Dense
input: (None, 50)
output: (None, 50)
dense_3: Dense
input: (None, 50)
output: (None, 10)

[0]:
```
conv2d_1 = K.function([cnn_model.layers[0].input], [cnn_model.layers[0].output])
conv2d_2 = K.function([cnn_model.layers[0].input], [cnn_model.layers[3].output])
dense_1 = K.function([cnn_model.layers[0].input], [cnn_model.layers[7].output])
dense_2 = K.function([cnn_model.layers[0].input], [cnn_model.layers[8].output])
dense_3 = K.function([cnn_model.layers[0].input], [cnn_model.layers[9].output])
```

[26]:
```
units = conv2d_1([data_test])[0].flatten()
print(units.shape)
```

```
(125440000,)
```

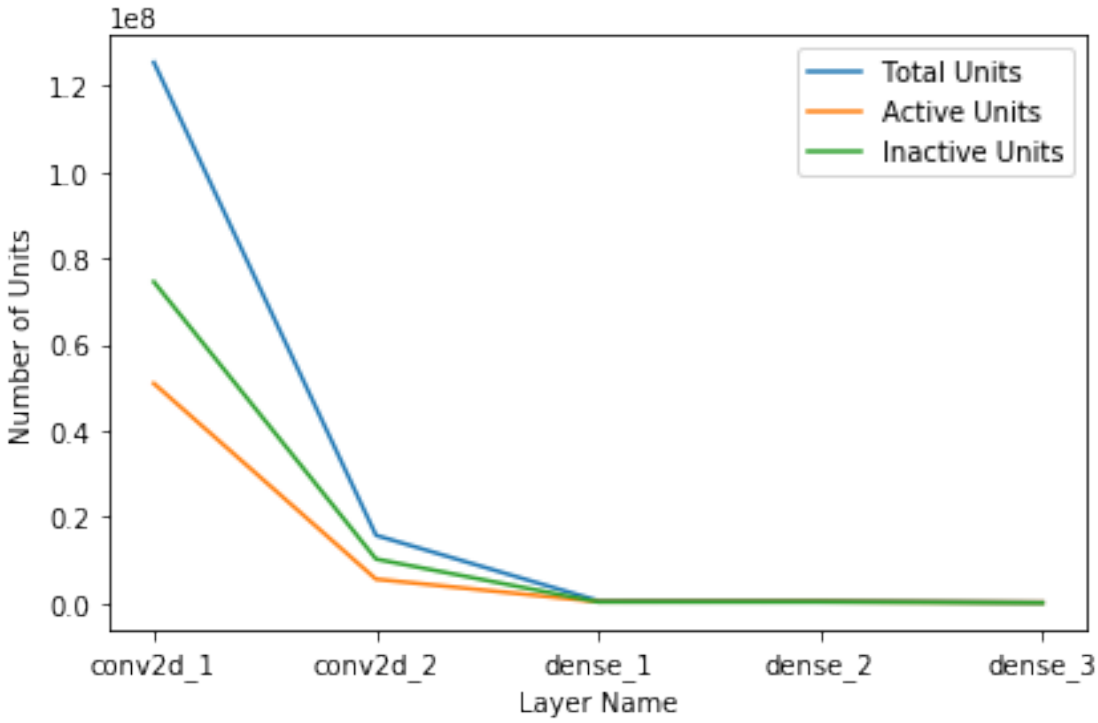
[0]:
```
def count_active_units(layer_function):
  """
    Count the number of active/inactive units in a layer
    Parameters:
      layer_function:
  """
  units = layer_function([data_test])[0].flatten()
  total_units = units.shape[0]
  non_zero = np.count_nonzero(units)
  return [total_units, non_zero]
```

[0]:
```
non_zero = []
total = []
for layer in [conv2d_1, conv2d_2, dense_1, dense_2, dense_3]:
  temp_tot, temp_nz = count_active_units(layer)
  non_zero.append(temp_nz)
  total.append(temp_tot)
```

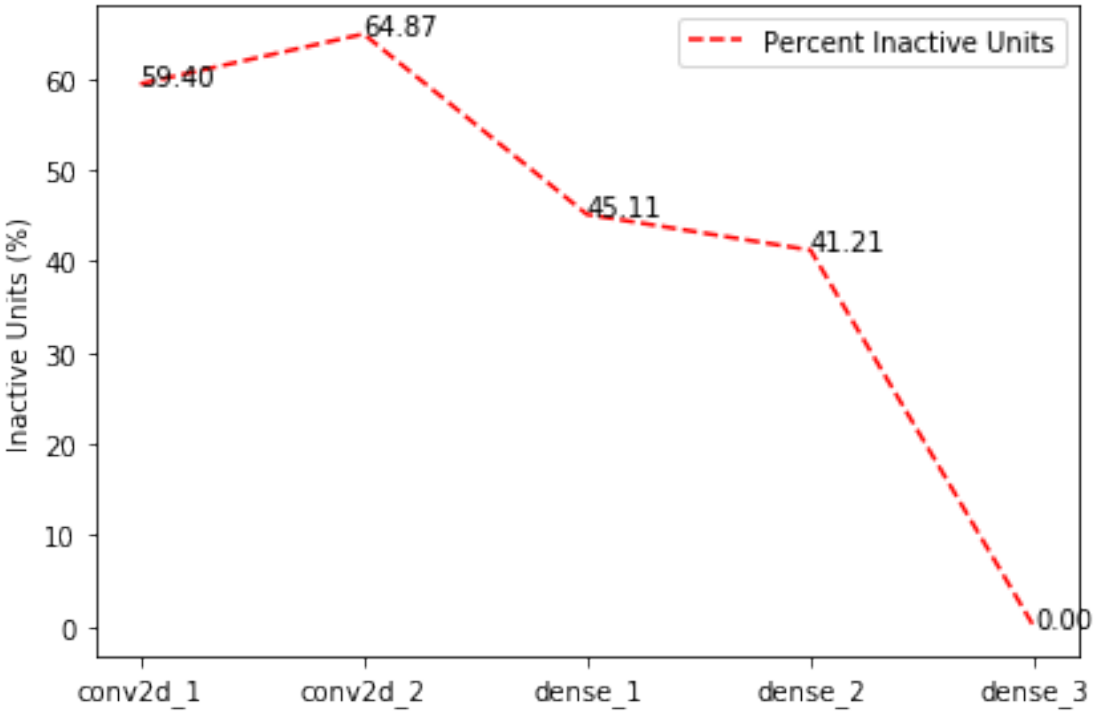
[41]:
```
zero = [i - j for i, j in zip(total, non_zero)]
ratios = [100 * (i-j) / i for i, j in zip(total, non_zero)]
print(zero)
```

```
[74510928, 10171293, 225561, 206053, 0]
```

[48]:
```
n_layers = len(zero)
n = ['conv2d_1', 'conv2d_2', 'dense_1', 'dense_2', 'dense_3']
fig, ax = plt.subplots()
ax.plot(n, total, label='Total Units')
ax.plot(n, non_zero, label='Active Units')
ax.plot(n, zero, label='Inactive Units')
ax.legend(loc='upper right')
ax.set_ylabel("Number of Units")
fig.tight_layout()
plt.show()
```



[61]:
```
n_layers = len(zero)
n = ['conv2d_1', 'conv2d_2', 'dense_1', 'dense_2', 'dense_3']
fig, ax = plt.subplots()
ax.plot(n, ratios, 'r--',
         label="Percent Inactive Units")
for i, j in zip(n, ratios):
  label = "{:.2f}".format(j)
  ax.annotate(label ,xy = (i,j))
ax.legend(loc='upper right')
ax.set_ylabel(r"Inactive Units (%)")
fig.tight_layout()
plt.show()
```


Percent Inactive Units
Inactive Units (%)
0
10
20
30
40
50
60
59.40
64.87
45.11
41.21
0.00
conv2d_1
conv2d_2
dense_1
dense_2
dense_3

[0]: