

## RESULTS: Jupyter Notebook --- With Tensorflow

Directory: ./ieee\_ml/problem\_sets/Linear Regression

*multiple\_linear\_regression\_problem.ipynb (Choice = 5)*

The screenshot shows a Jupyter Notebook interface with several plots generated by Python code. The plots include scatter plots of Acceleration vs MPG, Cylinders vs MPG, Displacement vs MPG, Horsepower vs MPG, Model Year vs MPG, and Weight vs MPG. Below the plots, a code cell contains a warning message from ipykernel\_launcher.py about the 'rcond' parameter.

```
plt.subplot(1,6,5)
plt.plot(x[:4],y_est,'b*'); #Plot the data; marker 'rd' =red diamonds
plt.xlabel('MPG')
plt.ylabel('Model Year')

plt.subplot(1,6,6)
plt.plot(x[:5],y_est,'b*'); #Plot the data; marker 'rd' =red diamonds
plt.xlabel('MPG')
plt.ylabel('Weight')

plt.subplots_adjust(left=None, bottom=None, right=1.5, top=None,
                   wspace=0.6, hspace=None)

/home/kaustubh/tf1/venv/lib/python3.5/site-packages/ipykernel_launcher.py:22: FutureWarning: 'rcond' parameter will change to the default of machine precision times ``max(M, N)`` where M and N are the input matrix dimensions.
To use the future default and silence this warning we advise to pass 'rcond=None', to keep using the old, explicit
y pass 'rcond=-1'.
```

In [6]: # Evaluate each fit you make in the context of your data. For example,
# if your goal of fitting the data is to extract coefficients that have
# physical meaning, then it is important that your model reflect the

The screenshot shows a Jupyter Notebook interface with two plots. The top plot displays residuals (blue dots) and a zero line (red horizontal line) against data index. The bottom plot compares actual MPG (red line) and fitted MPG (blue line) against data index.

```
plt.xlabel('data_index')
plt.ylabel('residuals')
plt.legend(['residuals','zero line'], loc= 'upper left')
plt.show()

plt.figure
plt.plot(y,'r')
plt.plot(y_est,'b')
plt.ylabel('MPG'); # Set the y-axis label
plt.xlabel('Data index');
plt.legend(['actual','fitted'],loc='upper left');
plt.show()
```

## linear\_regression.ipynb

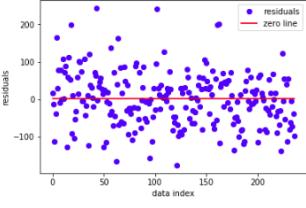
Sessions (1) IEEE Machine Learning Apollo AI and IEEE ML ieee\_ml/Linux\_REPO ieee\_ml/problem\_5 Home linear\_regression linear\_regression.ipynb

jupyter linear\_regression (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

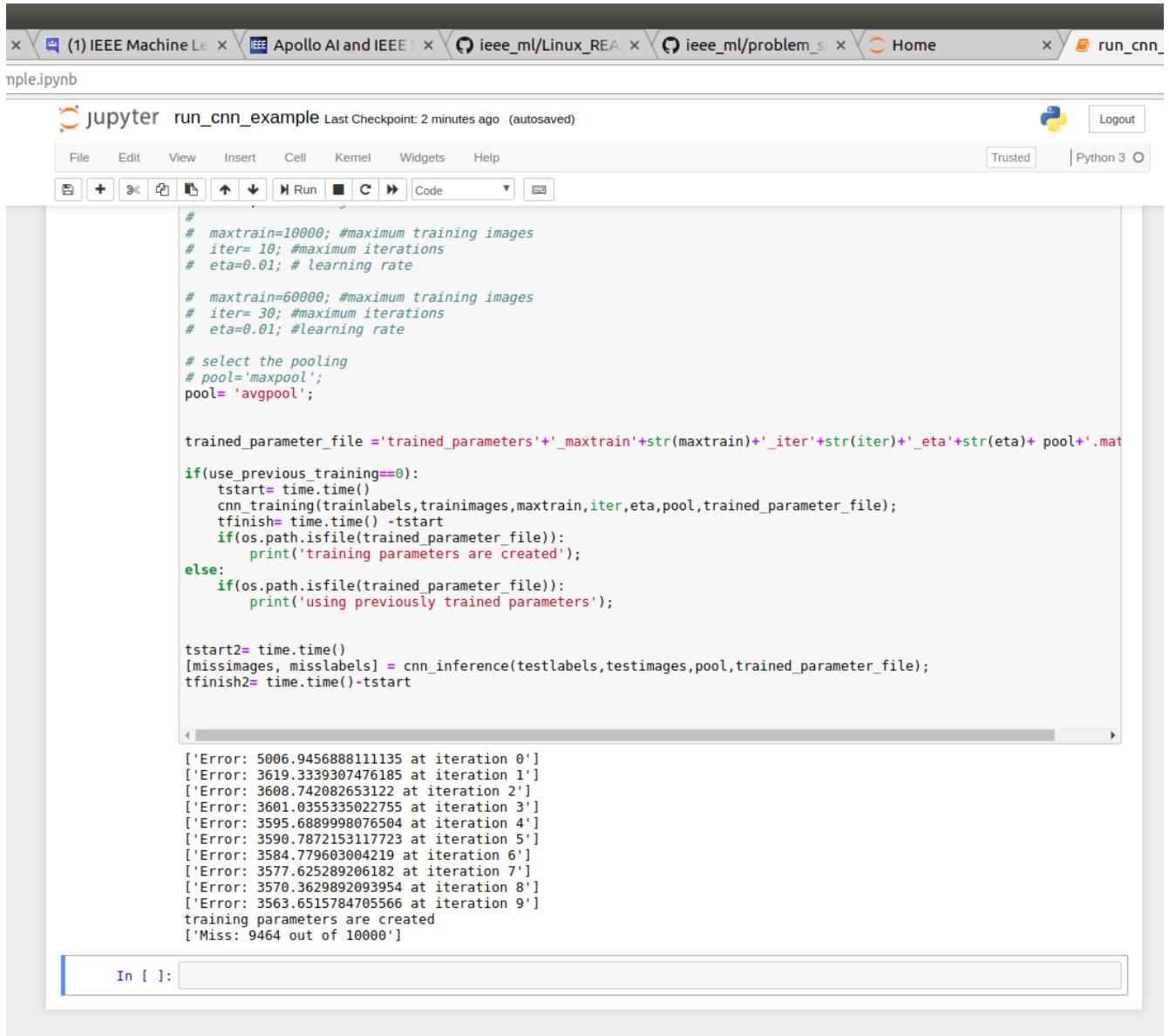
```
# That is, the distribution of residuals ought not to exhibit a discernible pattern.<# Producing a fit using a linear model requires minimizing the sum of the squares of the residuals. This minimization yields what is called a least-squares fit.<# You can gain insight into the goodness of a fit by visually examining a plot of the residuals. If the residual plot has a pattern (that is, residual data points do not appear to have a random scatter), the randomness indicates that the model does not properly fit the data.<# The higher the value of R-square, the better the model is at predicting the data.<# Say if Rsq is 0.7, we can say that 70% of the variation in dependent variable is explained by the independent variable.  
residuals=br_y-br_y_est  
Rsq = 1 - sum(residuals**2)/sum((br_y - np.mean(br_y))**2)  
  
In [4]: # also examine mean and standard deviation of residuals.  
mean_residuals=np.mean(residuals)  
std_residuals=np.std(residuals)  
  
In [5]: x_range= range(0,n)  
ref=np.ones(n)  
plt.plot(x_range,residuals,'bo',x_range,ref,'r-')  
plt.xlabel('data index')  
plt.ylabel('residuals')  
plt.legend(['residuals','zero line'], loc='upper right')  
plt.show()
```

In [ ]:



**Directory: ./Directory: ./ieee\_ml/problem\_sets/hdr\_om**

**run\_cnn\_example.ipynb**



The screenshot shows a Jupyter Notebook interface with the following details:

- Header:** The title bar shows the path: (1) IEEE Machine Learning > Apollo AI and IEEE > ieee\_ml/Linux\_REA > ieee\_ml/problem\_sets > Home > run\_cnn\_example.ipynb.
- Toolbar:** Standard Jupyter Notebook toolbar with icons for file operations, cell types, and execution.
- Header Bar:** Shows "jupyter run\_cnn\_example" and "Last Checkpoint: 2 minutes ago (autosaved)".
- Menu Bar:** File, Edit, View, Insert, Cell, Kernel, Widgets, Help.
- Status Bar:** Trusted | Python 3 | Logout.
- Code Cell:** Contains the following Python code:

```
# maxtrain=10000; #maximum training images
# iter= 10; #maximum iterations
# eta=0.01; # learning rate

# maxtrain=60000; #maximum training images
# iter= 30; #maximum iterations
# eta=0.01; #learning rate

# select the pooling
# pool='maxpool';
pool= 'avgpool';

trained_parameter_file ='trained_parameters'+ '_maxtrain'+str(maxtrain)+ '_iter'+str(iter)+ '_eta'+str(eta)+ pool+'.mat

if(use_previous_training==0):
    tstart= time.time()
    cnn_training(trainlabels,trainimages,maxtrain,iter,eta,pool,trained_parameter_file);
    tfinish= time.time() -tstart
    if(os.path.isfile(trained_parameter_file)):
        print('training parameters are created');
else:
    if(os.path.isfile(trained_parameter_file)):
        print('using previously trained parameters');

tstart2= time.time()
[missimages, misslabels] = cnn_inference(testlabels,testimages,pool,trained_parameter_file);
tfinish2= time.time()-tstart

['Error: 5006.9456888111135 at iteration 0']
['Error: 3619.3339307476185 at iteration 1']
['Error: 3608.742082653122 at iteration 2']
['Error: 3601.0355335022755 at iteration 3']
['Error: 3595.6889998076504 at iteration 4']
['Error: 3590.7872153117723 at iteration 5']
['Error: 3584.779603004219 at iteration 6']
['Error: 3577.625289206182 at iteration 7']
['Error: 3570.3629892093954 at iteration 8']
['Error: 3563.6515784705566 at iteration 9']
training parameters are created
['Miss: 9464 out of 10000']
```
- Output Cell:** Shows the output of the code cell, which includes error messages and a success message.

Directory: ./ieee\_ml/problem\_sets/hdr\_tf

### handwritten\_digits\_recognition\_cnn\_5layer.ipynb

```
t.py:290: dataset._init__ (from tensorflow.contrib.learn.python.learn.datasets.mnist) is deprecated and will be removed in a future version.
Instructions for updating:
Please use alternatives such as official/mnist/dataset.py from tensorflow/models.
WARNING:tensorflow:From <ipython-input-1-82274a480199>:107: softmax_cross_entropy_with_logits (from tensorflow.python.ops.nn_ops) is deprecated and will be removed in a future version.
Instructions for updating:

Future major versions of TensorFlow will allow gradients to flow
into the labels input on backprop by default.

See @{$tf.nn.softmax_cross_entropy_with_logits_v2}.

Iter 1280, Minibatch Loss= 36003.253906, Training Accuracy= 22.65625 %
Iter 2560, Minibatch Loss= 17263.601562, Training Accuracy= 40.62500 %
Iter 3840, Minibatch Loss= 8895.551758, Training Accuracy= 60.15625 %
Iter 5120, Minibatch Loss= 7292.077148, Training Accuracy= 66.40625 %
Iter 6400, Minibatch Loss= 3922.163086, Training Accuracy= 78.90625 %
Iter 7680, Minibatch Loss= 3477.270264, Training Accuracy= 75.78125 %
Iter 8960, Minibatch Loss= 3222.591797, Training Accuracy= 82.81250 %
Optimization Finished!
Testing Accuracy: 76.171875
```

### handwritten\_digits\_recognition\_cnn\_5layer\_solution.ipynb

The screenshot shows a Jupyter Notebook interface within a Chrome browser window. The title bar of the browser indicates the URL is 'handwritten\_digits\_recognition\_cnn\_5layer\_solution.ipynb'. The notebook itself has a header 'jupyter handwritten\_digits\_recognition\_cnn\_5layer\_solution (unsaved changes)' and a toolbar with standard file operations like File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Run, and Code.

The main content area of the notebook contains several blocks of text, which are deprecation warnings and training logs. These logs show the progress of training a CNN model on the MNIST dataset, including minibatch losses and training accuracies at various iterations. The logs conclude with an 'Optimization Finished!' message and a final testing accuracy of 94.921875.

```
INSTRUCTIONS FOR UPDATING:
Please use alternatives such as official/mnist/dataset.py from tensorflow/models.
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow/contrib/learn/python/learn/datasets/mnist.py:262: maybe_download (from tensorflow.contrib.learn.python.learn.datasets.base) is deprecated and will be removed in a future version.
Instructions for updating:
Please write your own downloading logic.
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow/contrib/learn/python/learn/datasets/mnist.py:262: extract_images (from tensorflow.contrib.learn.python.learn.datasets.mnist) is deprecated and will be removed in a future version.
Instructions for updating:
Please use tf.data to implement this functionality.
Extracting model_data/train-images-idx3-ubyte.gz
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow/contrib/learn/python/learn/datasets/mnist.py:267: extract_labels (from tensorflow.contrib.learn.python.learn.datasets.mnist) is deprecated and will be removed in a future version.
Instructions for updating:
Please use tf.data to implement this functionality.
Extracting model_data/train-labels-idx1-ubyte.gz
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow/contrib/learn/python/learn/datasets/mnist.py:110: dense_to_one_hot (from tensorflow.contrib.learn.python.learn.datasets.mnist) is deprecated and will be removed in a future version.
Instructions for updating:
Please use tf.one_hot on tensors.
Extracting model_data/t10k-images-idx3-ubyte.gz
Extracting model_data/t10k-labels-idx1-ubyte.gz
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow/contrib/learn/python/learn/datasets/mnist.py:290: DataSet._init__ (from tensorflow.contrib.learn.python.learn.datasets.mnist) is deprecated and will be removed in a future version.
Instructions for updating:
Please use alternatives such as official/mnist/dataset.py from tensorflow/models.
WARNING:tensorflow:From <ipython-input-1-0d40eac4ef95>:107: softmax_cross_entropy_with_logits (from tensorflow.python.ops.nn_ops) is deprecated and will be removed in a future version.
Instructions for updating:

Future major versions of TensorFlow will allow gradients to flow
into the labels input on backprop by default.

See @{$tf.nn.softmax_cross_entropy_with_logits_v2}.

Iter 2560, Minibatch Loss= 2866.634033, Training Accuracy= 54.68750 %
Iter 5120, Minibatch Loss= 478.556793, Training Accuracy= 79.29688 %
Iter 7680, Minibatch Loss= 304.790771, Training Accuracy= 82.81250 %
Iter 10240, Minibatch Loss= 161.260834, Training Accuracy= 91.01562 %
Iter 12800, Minibatch Loss= 50.366440, Training Accuracy= 93.35938 %
Iter 15360, Minibatch Loss= 95.883804, Training Accuracy= 91.01562 %
Iter 17920, Minibatch Loss= 102.791054, Training Accuracy= 91.01562 %
Optimization Finished!
Testing Accuracy: 94.921875
```

Directory: ./ieee\_ml/problem\_sets/Nearest Neighbor

**testfastKNN.ipynb**

The screenshot shows a Jupyter Notebook interface with the title "jupyter test\_fastKNN (unsaved changes)". The notebook contains the following Python code:

```
training_dataset = np.array([
    # weight, color, # seeds, type
    [363, 2, 1, 0],
    [370, 0, 2, 1],
    [298, 2, 1, 0],
    [277, 2, 1, 0],
    [377, 3, 2, 1],
    [299, 2, 1, 0],
    [382, 0, 2, 1],
    [374, 3, 6, 1],
    [303, 3, 1, 0],
    [309, 2, 1, 0],
    [359, 0, 2, 1],
    [366, 0, 4, 1],
    [311, 2, 1, 0],
    [302, 2, 1, 0],
    [373, 3, 4, 1],
    [305, 2, 1, 0],
    [371, 2, 6, 1]
], dtype=np.float32
)

validation_dataset = np.array([
    [301, color.index('green'), 1],
    [346, color.index('yellow'), 4],
    [290, color.index('red'), 2]
], dtype=np.float32
)

normalize_datasets=1;
[row,col]=np.shape(training_dataset)

if(normalize_datasets):
    # normalize = @(x) (x - min(x)) / max((x - min(x))); % reduce by smallest value
    for i in range(col-1):
        training_dataset[:,i]=normalize(training_dataset[:,i]);
        validation_dataset[:,i]=normalize(validation_dataset[:,i]);

[classified_type, k, index]=fastKNN(training_dataset,validation_dataset);

for i in range(0,len(classified_type)):
    print(fruit[classified_type[i]])

Banana
Apple
Banana
```

## **RESULTS: Python Scripts --- With Tensorflow; installed requirements.txt in a virtualenv**

Directory: ./ieee\_ml/problem\_sets/Linear Regression

### ***linear\_regression.py***

```
n$ python3 linear_regression.py
lt of machine precision times ``max(M, N)`` where M and N are the input matrix dimensions.
to keep using the old, explicitly pass `rcond=-1`.
```

Figure 1

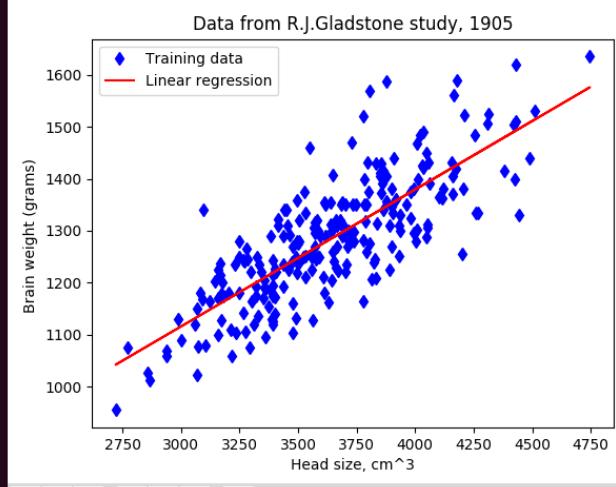


Figure 1

```
n$ python3 linear_regression.py
lt of machine precision times ``max(M, N)`` where M and N are the input matrix dimensions.
to keep using the old, explicitly pass `rcond=-1`.
```

Figure 1

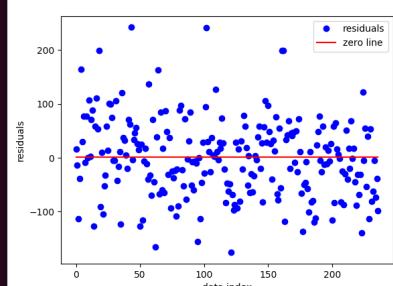


Figure 1

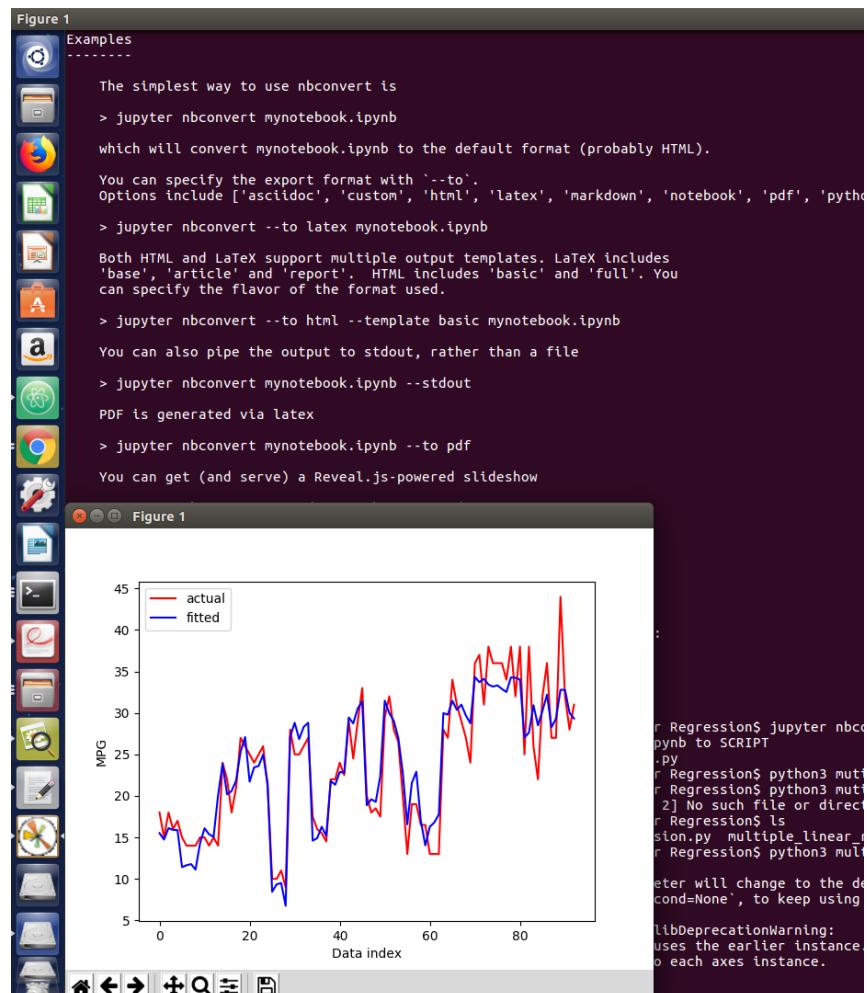
## *multiple\_linear\_regression\_problem.py (choice = 5)*

```
bconvert is
book.ipynb
book.ipynb to the default format (probably HTML).
t format with '--to'.
c', 'custom', 'html', 'latex', 'markdown', 'notebook', 'pdf', 'python', 'rst', 'script', 'slides']
latex mynotebook.ipynb
rt multiple output templates
port'. HTML includes 'basic
' the format used.

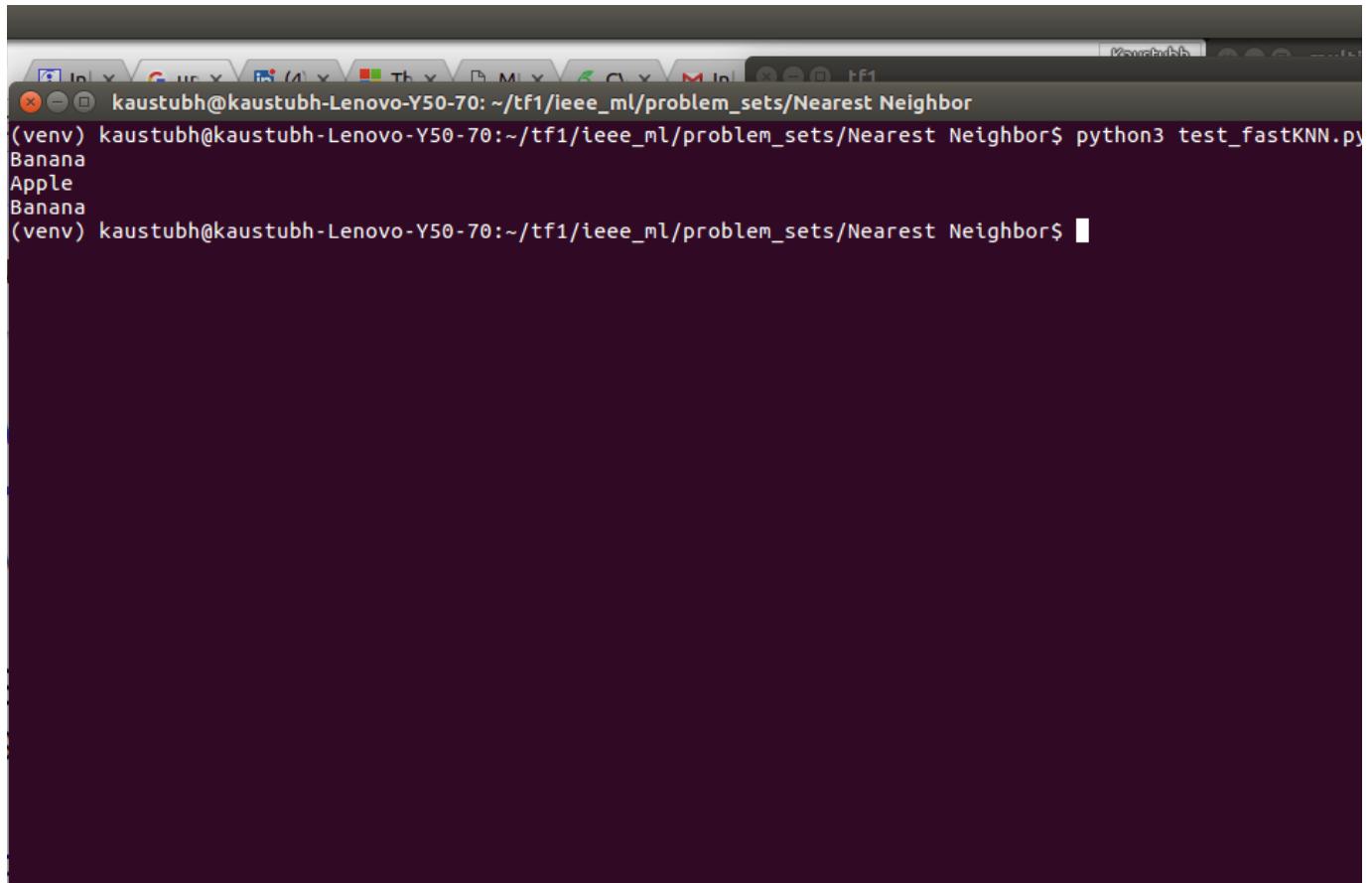
html --template basic mynote
put to stdout, rather than a
ebook.ipynb --stdout
x
ebook.ipynb --to pdf
Reveal.js-powered slideshow
des.ipynb --to slides --post
given at the command line i
ook*.ipynb
ook1.ipynb notebook2.ipynb
tebooks list in a config fil
oks = ["my_notebook.ipynb"]
fig mycfg.py
vo-Y50-70:~/tf1/ieee_ml/prob
ebook multiple_linear_regr
bytes to multiple_linear_regression_problem.py
vo-Y50-70:~/tf1/ieee_ml/problem_sets/Linear Regression$ python3 mutiple_linear_regression_problem.py
vo-Y50-70:~/tf1/ieee_ml/problem_sets/Linear Regression$ python3 mutiple_linear_regression_problem.py
lile_linear_regression_problem.py': [Errno 2] No such file or directory
vo-Y50-70:~/tf1/ieee_ml/problem_sets/Linear Regression$ ls
at linear regression.ipynb linear regression.py multiple linear regression problem.ipynb
```

Figure 1

The figure contains four subplots arranged in a 2x2 grid. The top row shows MPG vs Acceleration and MPG vs Cylinders. The bottom row shows MPG vs Displacement and MPG vs Horsepower. Each plot has red diamond markers representing actual data points and blue star markers representing fitted regression points.



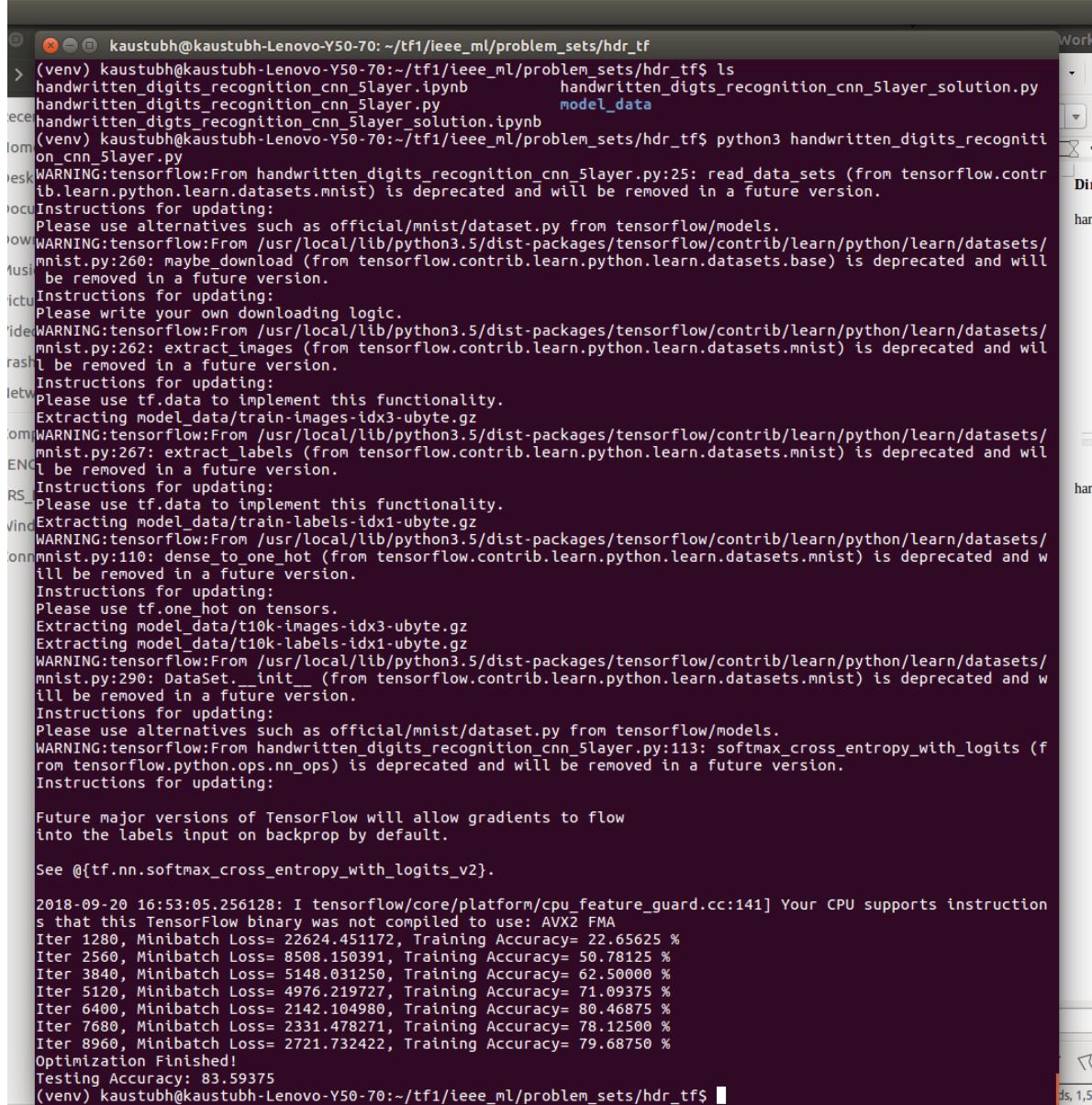
**Directory: ./ieee\_ml/problem\_sets/Nearest Neighbor**



```
kaustubh@kaustubh-Lenovo-Y50-70: ~/tf1/ieee_ml/problem_sets/Nearest Neighbor
(venv) kaustubh@kaustubh-Lenovo-Y50-70:~/tf1/ieee_ml/problem_sets/Nearest Neighbor$ python3 test_fastKNN.py
Banana
Apple
Banana
(venv) kaustubh@kaustubh-Lenovo-Y50-70:~/tf1/ieee_ml/problem_sets/Nearest Neighbor$
```

**Directory: ./ieee\_ml/problem\_sets/hdr\_tf**

**handwritten\_digits\_recognition\_cnn\_5layer.py**



```
kaustubh@kaustubh-Lenovo-Y50-70:~/tf1/ieee_ml/problem_sets/hdr_tf$ ls
handwritten_digits_recognition_cnn_5layer.ipynb      handwritten_digits_recognition_cnn_5layer_solution.py
handwritten_digits_recognition_cnn_5layer.py          model_data
handwritten_digits_recognition_cnn_5layer_solution.ipynb
(venv) kaustubh@kaustubh-Lenovo-Y50-70:~/tf1/ieee_ml/problem_sets/hdr_tf$ python3 handwritten_digits_recognition_cnn_5layer.py
WARNING:tensorflow:From handwritten_digits_recognition_cnn_5layer.py:25: read_data_sets (from tensorflow.contrib.learn.python.learn.datasets.mnist) is deprecated and will be removed in a future version.
Instructions for updating:
Please use alternatives such as official/mnist/dataset.py from tensorflow/models.
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow/contrib/learn/python/learn/datasets/mnist.py:260: maybe_download (from tensorflow.contrib.learn.python.learn.datasets.base) is deprecated and will be removed in a future version.
Instructions for updating:
Please write your own downloading logic.
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow/contrib/learn/python/learn/datasets/mnist.py:262: extract_images (from tensorflow.contrib.learn.python.learn.datasets.mnist) is deprecated and will be removed in a future version.
Instructions for updating:
Please use tf.data to implement this functionality.
Extracting model_data/train-images-idx3-ubyte.gz
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow/contrib/learn/python/learn/datasets/mnist.py:267: extract_labels (from tensorflow.contrib.learn.python.learn.datasets.mnist) is deprecated and will be removed in a future version.
Instructions for updating:
Please use tf.data to implement this functionality.
Extracting model_data/train-labels-idx1-ubyte.gz
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow/contrib/learn/python/learn/datasets/mnist.py:110: dense_to_one_hot (from tensorflow.contrib.learn.python.learn.datasets.mnist) is deprecated and will be removed in a future version.
Instructions for updating:
Please use tf.one_hot on tensors.
Extracting model_data/tf10k-images-idx3-ubyte.gz
Extracting model_data/tf10k-labels-idx1-ubyte.gz
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow/contrib/learn/python/learn/datasets/mnist.py:290: DataSet.__init__ (from tensorflow.contrib.learn.python.learn.datasets.mnist) is deprecated and will be removed in a future version.
Instructions for updating:
Please use alternatives such as official/mnist/dataset.py from tensorflow/models.
WARNING:tensorflow:From handwritten_digits_recognition_cnn_5layer.py:113: softmax_cross_entropy_with_logits (from tensorflow.python.ops.nn_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Future major versions of TensorFlow will allow gradients to flow into the labels input on backprop by default.

See @tf.nn.softmax_cross_entropy_with_logits_v2}.

2018-09-20 16:53:05.256128: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instruction s that this TensorFlow binary was not compiled to use: AVX2 FMA
Iter 1280, Minibatch Loss= 22624.451172, Training Accuracy= 22.65625 %
Iter 2560, Minibatch Loss= 8508.150391, Training Accuracy= 50.78125 %
Iter 3840, Minibatch Loss= 5148.031250, Training Accuracy= 62.50000 %
Iter 5120, Minibatch Loss= 4976.219727, Training Accuracy= 71.09375 %
Iter 6400, Minibatch Loss= 2142.104980, Training Accuracy= 80.46875 %
Iter 7680, Minibatch Loss= 2331.478271, Training Accuracy= 78.12500 %
Iter 8960, Minibatch Loss= 2721.732422, Training Accuracy= 79.68750 %
Optimization Finished!
Testing Accuracy: 83.59375
(venv) kaustubh@kaustubh-Lenovo-Y50-70:~/tf1/ieee_ml/problem_sets/hdr_tf$
```

## handwritten\_digits\_recognition\_cnn\_5layer\_solution.py

```
Terminal
kaustubh@kaustubh-Lenovo-Y50-70:~/tf1/ieee_ml/problem_sets/hdr_tf$ python3 handwritten_digits_recognition_cnn_5layer_solution.py
Traceback (most recent call last):
  File "handwritten_digits_recognition_cnn_5layer_solution.ipynb", line 5, in <module>
    "execution_count": null,
NameError: name 'null' is not defined
(venv) kaustubh@kaustubh-Lenovo-Y50-70:~/tf1/ieee_ml/problem_sets/hdr_tf$ python3 handwritten_digits_recognition_cnn_5layer_solution.py
WARNING:tensorflow:From handwritten_digits_recognition_cnn_5layer_solution.py:7: read_data_sets (from tensorflow.contrib.learn.python.learn.datasets.mnist) is deprecated and will be removed in a future version.
Instructions for updating:
Please use alternatives such as official/mnist/dataset.py from tensorflow/models.
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow/contrib/learn/python/learn/datasets/mnist.py:260: maybe_download (from tensorflow.contrib.learn.python.learn.datasets.base) is deprecated and will be removed in a future version.
Instructions for updating:
Please write your own downloading logic.
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow/contrib/learn/python/learn/datasets/mnist.py:262: extract_images (from tensorflow.contrib.learn.python.learn.datasets.mnist) is deprecated and will be removed in a future version.
Instructions for updating:
Please use tf.data to implement this functionality.
Extracting model_data/train-images-idx3-ubyte.gz
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow/contrib/learn/python/learn/datasets/mnist.py:267: extract_labels (from tensorflow.contrib.learn.python.learn.datasets.mnist) is deprecated and will be removed in a future version.
Instructions for updating:
Please use tf.data to implement this functionality.
Extracting model_data/train-labels-idx1-ubyte.gz
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow/contrib/learn/python/learn/datasets/mnist.py:110: dense_to_one_hot (from tensorflow.contrib.learn.python.learn.datasets.mnist) is deprecated and will be removed in a future version.
Instructions for updating:
Please use tf.one_hot on tensors.
Extracting model_data/t10k-images-idx3-ubyte.gz
Extracting model_data/t10k-labels-idx1-ubyte.gz
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow/contrib/learn/python/learn/datasets/mnist.py:290: DataSet.__init__ (from tensorflow.contrib.learn.python.learn.datasets.mnist) is deprecated and will be removed in a future version.
Instructions for updating:
Please use alternatives such as official/mnist/dataset.py from tensorflow/models.
WARNING:tensorflow:From handwritten_digits_recognition_cnn_5layer_solution.py:95: softmax_cross_entropy_with_logits (from tensorflow.python.ops.nn_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Future major versions of TensorFlow will allow gradients to flow into the labels input on backprop by default.

See @tf.nn.softmax_cross_entropy_with_logits_v2.

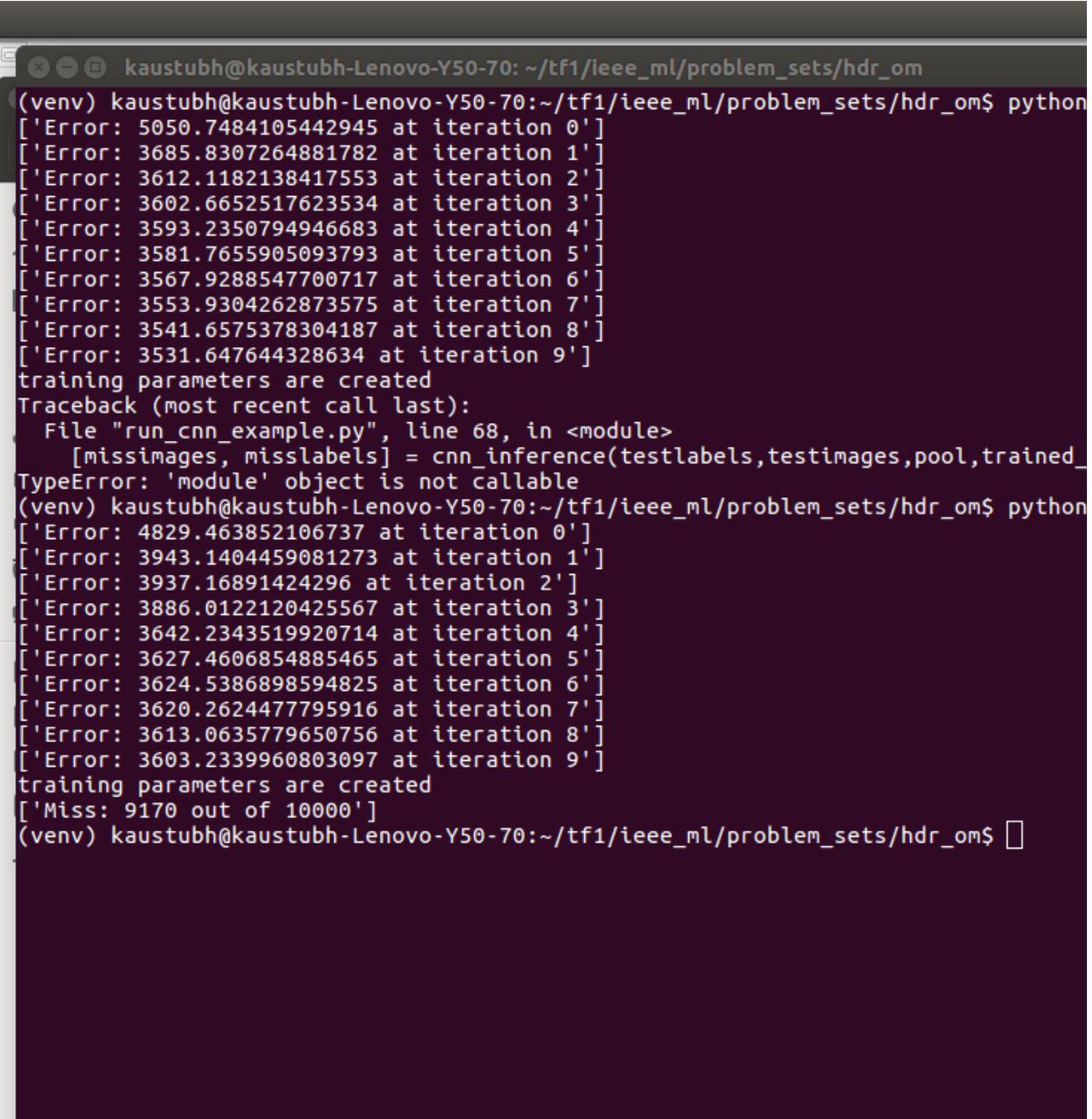
2018-09-20 14:02:05.665432: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instruction This is that this TensorFlow binary was not compiled to use: AVX2 FMA
acknow Iter 2560, Minibatch Loss= 2928.408691, Training Accuracy= 67.18750 %
use of Iter 5120, Minibatch Loss= 1065.566895, Training Accuracy= 82.03125 %
and te Iter 7680, Minibatch Loss= 248.228638, Training Accuracy= 87.50000 %
Iter 10240, Minibatch Loss= 159.545456, Training Accuracy= 89.84375 %
Iter 12800, Minibatch Loss= 214.196152, Training Accuracy= 89.06250 %
Iter 15360, Minibatch Loss= 132.450256, Training Accuracy= 91.79688 %
Iter 17920, Minibatch Loss= 68.867607, Training Accuracy= 94.14062 %
Optimization Finished!
Testing Accuracy: 94.921875
(venv) kaustubh@kaustubh-Lenovo-Y50-70:~/tf1/ieee_ml/problem_sets/hdr_tf$
```

**Directory: ./ieee\_ml/problem\_sets/hdr\_om**

***run\_cnn\_example.py***

**Note:**

- While trying to execute the *run\_cnn\_example.py* file, import other python scripts in the directory and change the method calls to;
- (*cnn\_load()*, *cnn\_training*, *cnn\_inference*) ---> (*cnn\_training.cnn\_load()*, *cnn\_training.cnn\_training()* and *cnn\_inference.snn\_inference()*)
- As, the environment for executing this program is no longer the Ipy notebook.



```
kaustubh@kaustubh-Lenovo-Y50-70: ~/tf1/ieee_ml/problem_sets/hdr_om$ python
['Error: 5050.7484105442945 at iteration 0']
['Error: 3685.8307264881782 at iteration 1']
['Error: 3612.1182138417553 at iteration 2']
['Error: 3602.6652517623534 at iteration 3']
['Error: 3593.2350794946683 at iteration 4']
['Error: 3581.7655905093793 at iteration 5']
['Error: 3567.9288547700717 at iteration 6']
['Error: 3553.9304262873575 at iteration 7']
['Error: 3541.6575378304187 at iteration 8']
['Error: 3531.647644328634 at iteration 9']
training parameters are created
Traceback (most recent call last):
  File "run_cnn_example.py", line 68, in <module>
    [missimages, misslabels] = cnn_inference(testlabels,testimages,pool,trained_
TypeError: 'module' object is not callable
(venv) kaustubh@kaustubh-Lenovo-Y50-70:~/tf1/ieee_ml/problem_sets/hdr_om$ python
['Error: 4829.463852106737 at iteration 0']
['Error: 3943.1404459081273 at iteration 1']
['Error: 3937.16891424296 at iteration 2']
['Error: 3886.0122120425567 at iteration 3']
['Error: 3642.2343519920714 at iteration 4']
['Error: 3627.4606854885465 at iteration 5']
['Error: 3624.5386898594825 at iteration 6']
['Error: 3620.2624477795916 at iteration 7']
['Error: 3613.0635779650756 at iteration 8']
['Error: 3603.2339960803097 at iteration 9']
training parameters are created
['Miss: 9170 out of 10000']
(venv) kaustubh@kaustubh-Lenovo-Y50-70:~/tf1/ieee_ml/problem_sets/hdr_om$ 
```

