

```
In [12]: import os
import pickle
import pandas as pd # panda's nickname is pd
import numpy as np # numpy as np
from pandas import DataFrame, Series # for convenience
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns
import datetime
import tensorflow as tf
```

```
In [13]: #load training and testing datasets
train=pd.read_csv('C:/Users/upsto/Downloads/train.csv')
test=pd.read_csv('C:/Users/upsto/Downloads/test.csv')
```

```
In [14]: #let's see wht this data looks like
train.head()
```

Out[14]:

	label	pixel0	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	...	pixel774	pixel775
0	1	0	0	0	0	0	0	0	0	0	...	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0
2	1	0	0	0	0	0	0	0	0	0	...	0	0
3	4	0	0	0	0	0	0	0	0	0	...	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0

5 rows × 785 columns



```
In [15]: test.head()
```

Out[15]:

	pixel0	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	pixel9	...	pixel774	pixel77
0	0	0	0	0	0	0	0	0	0	0	...	0	
1	0	0	0	0	0	0	0	0	0	0	...	0	
2	0	0	0	0	0	0	0	0	0	0	...	0	
3	0	0	0	0	0	0	0	0	0	0	...	0	
4	0	0	0	0	0	0	0	0	0	0	...	0	

5 rows × 784 columns



In [16]: `train.describe()`

Out[16]:

	label	pixel0	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8
count	42000.000000	42000.0	42000.0	42000.0	42000.0	42000.0	42000.0	42000.0	42000.0	42000.0
mean	4.456643	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
std	2.887730	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
min	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
25%	2.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
50%	4.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
75%	7.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
max	9.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

8 rows × 785 columns

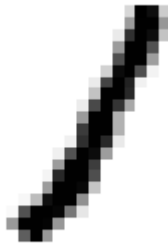
In [17]: `print('Training data shape')`
`print(train.shape)`
`print('Test data shape')`
`print(test.shape)`

Training data shape
 (42000, 785)
 Test data shape
 (28000, 784)

In [47]: `random_digit1 = X[1]`
`plt.figure(figsize=(3,3))`
`plot_digit(random_digit1)`
`plt.show()`



```
In [48]: random_digit2 = X[9200]  
plt.figure(figsize=(3,3))  
plot_digit(random_digit2)  
plt.show()
```



```
In [49]: random_digit3 = X[25000]  
plt.figure(figsize=(3,3))  
plot_digit(random_digit3)  
plt.show()
```



```
In [50]: plt.figure(figsize=(8,8))
example_images = np.r_[X[:12000:600], X[13000:30600:600], X[30600:60000:590]]
plot_digits(example_images, images_per_row=10)
plt.show()
```



```
In [18]: #(1) Begin by fitting a random forest classifier using the full set of 784 explanatory variables and the model training set (train.csv).
#Record the time it takes to fit the model and then evaluate the model on the test data by submitting to Kaggle.com. Provide your Kaggle.com score and user ID.
from sklearn.ensemble import RandomForestClassifier
```

```
In [19]: x_train = train.iloc[:,1:]
y_train = train['label']

print('Training data x shape')
print(x_train.shape)
print('Training data y shape')
print(y_train.shape)
```

```
Training data x shape
(42000, 784)
Training data y shape
(42000,)
```

```
In [20]: start=datetime.datetime.now()

random_forest = RandomForestClassifier(n_estimators=10, max_features="sqrt",
                                      bootstrap=True)
random_forest.fit(x_train, y_train)

end=datetime.datetime.now()

print(end-start)
```

```
0:00:02.196420
```

```
In [21]: rfp = random_forest.predict(test)
rfp.shape
```

```
Out[21]: (28000,)
```

```
In [22]: data = {'ImageId': np.arange(1,28001), 'Label': rfp}
df = pd.DataFrame(data=data)
df.head()
```

```
Out[22]:
```

	ImageId	Label
0	1	2
1	2	0
2	3	9
3	4	9
4	5	3

```
In [23]: df.to_csv('random_forest_1.csv', index=False)
```

```
In [24]: #Kaggle Results
print ('Kaggle Results:')
print('Name: James Casey')
print('Score: 0.94342')
print('Rank: 2105')
```

Kaggle Results:
Name: James Casey
Score: 0.94342
Rank: 2105

```
In [25]: #(2) Execute principal components analysis (PCA) on the combined training
#and test set data together, generating principal components that represent
#95 percent of the variability in the explanatory variables. The number of
#principal components in the solution should be substantially fewer than the
#784 explanatory variables. Record the time it takes to identify the principal co
```

```
In [26]: from sklearn.decomposition import PCA
pca_data = pd.concat([x_train,test])
pca_data.head()
```

Out[26]:

	pixel0	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	pixel9	...	pixel774	pixel77
0	0	0	0	0	0	0	0	0	0	0	...	0	
1	0	0	0	0	0	0	0	0	0	0	...	0	
2	0	0	0	0	0	0	0	0	0	0	...	0	
3	0	0	0	0	0	0	0	0	0	0	...	0	
4	0	0	0	0	0	0	0	0	0	0	...	0	

5 rows × 784 columns



```
In [27]: start=datetime.datetime.now()

pca = PCA(n_components=0.95)
pca_model = pca.fit_transform(pca_data)

end=datetime.datetime.now()
print(end-start)
```

0:00:04.372409

```
In [28]: print('PCA number of principal components:')
print(pca.n_components_)
```

PCA number of principal components:
154

```
In [29]: #(3) Using the identified principal components from step (2),  
#use the train.csv to build another random forest classifier.  
#Record the time it takes to fit the model and to evaluate the  
#model on the test.csv data by submitting to Kaggle.com.  
#Provide your Kaggle.com score and user ID.
```

```
In [30]: pca_X_train = pca_model[:42000]  
print(pca_X_train.shape)
```

(42000, 154)

```
In [31]: pca_X_test = pca_model[42000:]  
print(pca_X_test.shape)
```

(28000, 154)

```
In [32]: start=datetime.datetime.now()  
  
random_forest_2 = RandomForestClassifier(n_estimators=10, max_features="sqrt",  
                                         bootstrap=True, )  
random_forest_2.fit(pca_X_train, y_train)  
  
end=datetime.datetime.now()  
print(end-start)
```

0:00:04.794724

```
In [33]: rfp2 = random_forest_2.predict(pca_X_test)  
print(rfp2.shape)  
print(np.arange(1,28001).shape)
```

(28000,)
(28000,)

```
In [34]: data2 = {'ImageId': np.arange(1,28001), 'Label': rfp2}  
df2 = pd.DataFrame(data=data2)  
df2.to_csv('random_forest_pca.csv', index=False)
```

```
In [35]: #Kaggle Results  
print ('Kaggle Results:')  
print('Name: James Casey')  
print('Score: 0.88271')
```

Kaggle Results:
Name: James Casey
Score: 0.88271

Neural Network Analysis

Model 1

```
In [41]: start=datetime.datetime.now()

feat_cols = tf.contrib.learn.infer_real_valued_columns_from_input(x_train)
dnn_clf = tf.contrib.learn.DNNClassifier(hidden_units=[20,20],n_classes=10,
                                         feature_columns=feat_cols)

dnn_clf= tf.contrib.learn.SKCompat(dnn_clf)
dnn_clf.fit(x_train, y_train,batch_size=50,steps=40000)

end=datetime.datetime.now()

time1=(end-start)

INFO:tensorflow:global_step/sec: 735.469
INFO:tensorflow:loss = 0.40340546, step = 39601 (0.136 sec)
INFO:tensorflow:global_step/sec: 667.981

INFO:tensorflow:loss = 0.51161766, step = 39701 (0.150 sec)
INFO:tensorflow:global_step/sec: 664.369
INFO:tensorflow:loss = 0.8640259, step = 39801 (0.151 sec)
INFO:tensorflow:global_step/sec: 612.97
INFO:tensorflow:loss = 0.3407123, step = 39901 (0.163 sec)
INFO:tensorflow:Saving checkpoints for 40000 into C:\Users\upsto\AppData\Local\Temp\tmpdajqjckv\model.ckpt.
INFO:tensorflow:Loss for final step: 0.54816955.
```

```
In [42]: print(time1)
```

```
0:01:00.791029
```

```
In [53]: model1 = dnn_clf.predict(test)
model1['classes']
```

```
WARNING:tensorflow:float64 is not supported by many models, consider casting to float32.
INFO:tensorflow:Graph was finalized.
INFO:tensorflow:Restoring parameters from C:\Users\upsto\AppData\Local\Temp\tmpdajqjckv\model.ckpt-40000
INFO:tensorflow:Running local_init_op.
INFO:tensorflow:Done running local_init_op.
```

```
Out[53]: array([2, 0, 9, ..., 3, 9, 2], dtype=int64)
```



```
In [54]: data = {'ImageId': np.arange(1,28001), 'Label': model1['classes']}  
df = pd.DataFrame(data=data)  
df.head()
```

Out[54]:

	ImageId	Label
0	1	2
1	2	0
2	3	9
3	4	9
4	5	2

```
In [55]: df.to_csv('neural_network_model_1.csv', index=False)
```

```
In [131]: #Kaggle Results  
print ('Kaggle Results:')  
print('Name: jcasey2')  
print('Score: 0.80600')
```

Kaggle Results:
Name: jcasey2
Score: 0.80600

Model2

```
In [71]: #model2 3 layers 20 nodes
start=datetime.datetime.now()

feat_cols = tf.contrib.learn.infer_real_valued_columns_from_input(x_train)
dnn_clf2 = tf.contrib.learn.DNNClassifier(hidden_units=[20,20,20],n_classes=10,
                                         feature_columns=feat_cols)
dnn_clf2= tf.contrib.learn.SKCompat(dnn_clf2)
dnn_clf2.fit(x_train, y_train,batch_size=50,steps=40000)

end=datetime.datetime.now()

time2=(end-start)
```

```
\AppData\Local\Temp\tmpk3xvdub5
INFO:tensorflow:Using config: {'_task_type': None, '_task_id': 0, '_cluster_spec': <tensorflow.python.training.server_lib.ClusterSpec object at 0x00000259AEDFE048>, '_master': '', '_num_ps_replicas': 0, '_num_worker_replicas': 0, '_environment': 'local', '_is_chief': True, '_evaluation_master': '', '_train_distribute': None, '_eval_distribute': None, '_experimental_max_worker_delay_secs': None, '_device_fn': None, '_tf_config': gpu_options {
  per_process_gpu_memory_fraction: 1.0
}
, '_tf_random_seed': None, '_save_summary_steps': 100, '_save_checkpoints_secs': 600, '_log_step_count_steps': 100, '_protocol': None, '_session_config': None, '_save_checkpoints_steps': None, '_keep_checkpoint_max': 5, '_keep_checkpoint_every_n_hours': 10000, '_model_dir': 'C:\\Users\\upsto\\AppData\\Local\\Temp\\tmpk3xvdub5', '_session_creation_timeout_secs': 7200}
WARNING:tensorflow:float64 is not supported by many models, consider casting to float32.
INFO:tensorflow:Create CheckpointSaverHook.
INFO:tensorflow:Graph was finalized.
INFO:tensorflow:Running local_init_op.
INFO:tensorflow:Done running local_init_op.
```

```
In [72]: print(time2)
```

```
0:01:04.720485
```

```
In [73]: model2 = dnn_clf2.predict(test)
model2['classes']
```

```
WARNING:tensorflow:float64 is not supported by many models, consider casting to float32.
INFO:tensorflow:Graph was finalized.
INFO:tensorflow:Restoring parameters from C:\\Users\\upsto\\AppData\\Local\\Temp\\tmpk3xvdub5\\model.ckpt-40000
INFO:tensorflow:Running local_init_op.
INFO:tensorflow:Done running local_init_op.
```

```
Out[73]: array([2, 0, 8, ..., 3, 9, 2], dtype=int64)
```

```
In [74]: data = {'ImageId': np.arange(1,28001), 'Label': model12['classes']}  
df = pd.DataFrame(data=data)  
df.head()
```

Out[74]:

	ImageId	Label
0	1	2
1	2	0
2	3	8
3	4	4
4	5	3

```
In [75]: df.to_csv('neural_network_model_2.csv', index=False)
```

```
In [76]: #Kaggle Results  
print ('Kaggle Results:')  
print('Name: jcasey2')  
print('Score: 0.90985')
```

Kaggle Results:
Name: James Casey
Score: 0.90985

Model 3

```
In [78]: #model3 2 layers 50 nodes
start=datetime.datetime.now()

feat_cols = tf.contrib.learn.infer_real_valued_columns_from_input(x_train)
dnn_clf3 = tf.contrib.learn.DNNClassifier(hidden_units=[50,50],n_classes=10,
                                         feature_columns=feat_cols)
dnn_clf3= tf.contrib.learn.SKCompat(dnn_clf3)
dnn_clf3.fit(x_train, y_train,batch_size=50,steps=40000)

end=datetime.datetime.now()

time3=(end-start)
```

```
INFO:tensorflow:loss = 1.0258305, step = 801 (0.176 sec)

INFO:tensorflow:global_step/sec: 552.942
INFO:tensorflow:loss = 1.0394659, step = 901 (0.181 sec)
INFO:tensorflow:global_step/sec: 583.15
INFO:tensorflow:loss = 1.1754869, step = 1001 (0.170 sec)
INFO:tensorflow:global_step/sec: 573.859
INFO:tensorflow:loss = 1.0997005, step = 1101 (0.175 sec)
INFO:tensorflow:global_step/sec: 582.523
INFO:tensorflow:loss = 1.393914, step = 1201 (0.172 sec)
INFO:tensorflow:global_step/sec: 583.524
INFO:tensorflow:loss = 0.55748725, step = 1301 (0.171 sec)
INFO:tensorflow:global_step/sec: 578.431
INFO:tensorflow:loss = 0.85879534, step = 1401 (0.173 sec)
INFO:tensorflow:global_step/sec: 589.256
INFO:tensorflow:loss = 1.1494853, step = 1501 (0.169 sec)
INFO:tensorflow:global_step/sec: 564.154
INFO:tensorflow:loss = 0.84363055, step = 1601 (0.178 sec)
INFO:tensorflow:global_step/sec: 539.417
INFO:tensorflow:loss = 1.0943891, step = 1701 (0.185 sec)
```

```
In [79]: print(time3)
```

```
0:01:09.313966
```

```
In [80]: model3 = dnn_clf3.predict(test)
model3['classes']
```

```
WARNING:tensorflow:float64 is not supported by many models, consider casting to float32.
INFO:tensorflow:Graph was finalized.
INFO:tensorflow:Restoring parameters from C:\Users\upsto\AppData\Local\Temp\tmp7easl_i5\model.ckpt-40000
INFO:tensorflow:Running local_init_op.
INFO:tensorflow:Done running local_init_op.
```

```
Out[80]: array([2, 0, 9, ..., 3, 9, 2], dtype=int64)
```

```
In [81]: data = {'ImageId': np.arange(1,28001), 'Label': model3['classes']}  
df = pd.DataFrame(data=data)  
df.head()
```

Out[81]:

	ImageId	Label
0	1	2
1	2	0
2	3	9
3	4	9
4	5	3

```
In [82]: df.to_csv('neural_network_model_3.csv', index=False)
```

```
In [83]: #Kaggle Results  
print ('Kaggle Results:')  
print('Name: James Casey')  
print('Score: 0.91528')
```

Kaggle Results:
Name: James Casey
Score: 0.91528

Model 4

```
In [84]: #model4 3 layers 50 nodes
start=datetime.datetime.now()

feat_cols = tf.contrib.learn.infer_real_valued_columns_from_input(x_train)
dnn_clf4 = tf.contrib.learn.DNNClassifier(hidden_units=[50,50,50],n_classes=10,
                                         feature_columns=feat_cols)
dnn_clf4= tf.contrib.learn.SKCompat(dnn_clf4)
dnn_clf4.fit(x_train, y_train,batch_size=50,steps=40000)

end=datetime.datetime.now()

time4=(end-start)
```

```
INFO:tensorflow:loss = 0.31037548, step = 801 (0.173 sec)

INFO:tensorflow:global_step/sec: 560.569
INFO:tensorflow:loss = 0.2654488, step = 901 (0.178 sec)
INFO:tensorflow:global_step/sec: 526.052
INFO:tensorflow:loss = 0.4757804, step = 1001 (0.190 sec)
INFO:tensorflow:global_step/sec: 557.771
INFO:tensorflow:loss = 0.26063064, step = 1101 (0.179 sec)
INFO:tensorflow:global_step/sec: 526.514
INFO:tensorflow:loss = 0.6378693, step = 1201 (0.191 sec)
INFO:tensorflow:global_step/sec: 520.927
INFO:tensorflow:loss = 0.25214237, step = 1301 (0.192 sec)
INFO:tensorflow:global_step/sec: 541.629
INFO:tensorflow:loss = 0.22517012, step = 1401 (0.185 sec)
INFO:tensorflow:global_step/sec: 609.639
INFO:tensorflow:loss = 0.38979644, step = 1501 (0.163 sec)
INFO:tensorflow:global_step/sec: 532.217
INFO:tensorflow:loss = 0.2609018, step = 1601 (0.188 sec)
INFO:tensorflow:global_step/sec: 598.605
INFO:tensorflow:loss = 0.37777448, step = 1701 (0.167 sec)
```

```
In [85]: print(time4)
```

```
0:01:12.192160
```

```
In [86]: model4 = dnn_clf4.predict(test)
model4['classes']
```

```
WARNING:tensorflow:float64 is not supported by many models, consider casting to float32.
INFO:tensorflow:Graph was finalized.
INFO:tensorflow:Restoring parameters from C:\Users\upsto\AppData\Local\Temp\tmpdvn9kz93\model.ckpt-40000
INFO:tensorflow:Running local_init_op.
INFO:tensorflow:Done running local_init_op.
```

```
Out[86]: array([2, 0, 9, ..., 3, 9, 2], dtype=int64)
```

```
In [87]: data = {'ImageId': np.arange(1,28001), 'Label': model14['classes']}  
df = pd.DataFrame(data=data)  
df.head()
```

Out[87]:

	ImageId	Label
0	1	2
1	2	0
2	3	9
3	4	6
4	5	3

```
In [88]: df.to_csv('neural_network_model_4.csv', index=False)
```

```
In [91]: #Kaggle Results  
print ('Kaggle Results:')  
print('Name: James Casey')  
print('Score: 0.93885')
```

Kaggle Results:
Name: James Casey
Score: 0.93885

Model 5

```
In [92]: #model5 8 layers 100 nodes
start=datetime.datetime.now()

feat_cols = tf.contrib.learn.infer_real_valued_columns_from_input(x_train)
dnn_clf5 = tf.contrib.learn.DNNClassifier(hidden_units=[100,100,100,100,100,100],
                                          feature_columns=feat_cols)
dnn_clf5= tf.contrib.learn.SKCompat(dnn_clf5)
dnn_clf5.fit(x_train, y_train,batch_size=50,steps=40000)

end=datetime.datetime.now()

time5=(end-start)
```

```
INFO:tensorflow:global_step/sec: 434.525
INFO:tensorflow:loss = 0.2668001, step = 1501 (0.230 sec)
INFO:tensorflow:global_step/sec: 441.069
INFO:tensorflow:loss = 0.29545182, step = 1601 (0.226 sec)
INFO:tensorflow:global_step/sec: 420.185
INFO:tensorflow:loss = 0.5035731, step = 1701 (0.239 sec)
INFO:tensorflow:global_step/sec: 440.585
INFO:tensorflow:loss = 0.18192367, step = 1801 (0.226 sec)
INFO:tensorflow:global_step/sec: 395.819
INFO:tensorflow:loss = 0.080053136, step = 1901 (0.254 sec)
INFO:tensorflow:global_step/sec: 410.824
INFO:tensorflow:loss = 0.3277358, step = 2001 (0.243 sec)
INFO:tensorflow:global_step/sec: 419.578
INFO:tensorflow:loss = 0.21580271, step = 2101 (0.238 sec)
INFO:tensorflow:global_step/sec: 418.117
INFO:tensorflow:loss = 0.18568306, step = 2201 (0.239 sec)
INFO:tensorflow:global_step/sec: 425.217
INFO:tensorflow:loss = 0.06846981, step = 2301 (0.235 sec)
INFO:tensorflow:global_step/sec: 423.126
INFO:tensorflow:loss = 0.13677421, step = 2401 (0.237 sec)
```

```
In [93]: print(time5)
```

```
0:01:35.655664
```

```
In [94]: model5 = dnn_clf5.predict(test)
model5['classes']
```

```
WARNING:tensorflow:float64 is not supported by many models, consider casting to float32.
INFO:tensorflow:Graph was finalized.
INFO:tensorflow:Restoring parameters from C:\Users\upsto\AppData\Local\Temp\tmp
an7ti8iy\model.ckpt-40000
INFO:tensorflow:Running local_init_op.
INFO:tensorflow:Done running local_init_op.
```

```
Out[94]: array([2, 0, 9, ..., 3, 9, 2], dtype=int64)
```



```
In [95]: data = {'ImageId': np.arange(1,28001), 'Label': model15['classes']}  
df = pd.DataFrame(data=data)  
df.head()
```

Out[95]:

	ImageId	Label
0	1	2
1	2	0
2	3	9
3	4	4
4	5	3

```
In [96]: df.to_csv('neural_network_model_5.csv', index=False)
```

```
In [97]: #Kaggle Results  
print ('Kaggle Results:')  
print('Name: James Casey')  
print('Score: 0.95914')
```

Kaggle Results:
Name: James Casey
Score: 0.95914

Model 6

```
In [102]: #model6 8 layers 300 nodes
start=datetime.datetime.now()

feat_cols = tf.contrib.learn.infer_real_valued_columns_from_input(x_train)
dnn_clf6 = tf.contrib.learn.DNNClassifier(hidden_units=[300,300,300,300,300,300],
                                          feature_columns=feat_cols)
dnn_clf6= tf.contrib.learn.SKCompat(dnn_clf6)
dnn_clf6.fit(x_train, y_train,batch_size=50,steps=40000)

end=datetime.datetime.now()

time6=(end-start)
```

```
\AppData\Local\Temp\tmpf14bz2hx
INFO:tensorflow:Using config: {'_task_type': None, '_task_id': 0, '_cluster_spec': <tensorflow.python.training.server_lib.ClusterSpec object at 0x00000259AF379188>, '_master': '', '_num_ps_replicas': 0, '_num_worker_replicas': 0, '_environment': 'local', '_is_chief': True, '_evaluation_master': '', '_train_distribute': None, '_eval_distribute': None, '_experimental_max_worker_delay_secs': None, '_device_fn': None, '_tf_config': gpu_options {
  per_process_gpu_memory_fraction: 1.0
}
, '_tf_random_seed': None, '_save_summary_steps': 100, '_save_checkpoints_secs': 600, '_log_step_count_steps': 100, '_protocol': None, '_session_config': None, '_save_checkpoints_steps': None, '_keep_checkpoint_max': 5, '_keep_checkpoint_every_n_hours': 10000, '_model_dir': 'C:\\Users\\upsto\\AppData\\Local\\Temp\\tmpf14bz2hx', '_session_creation_timeout_secs': 7200}
WARNING:tensorflow:float64 is not supported by many models, consider casting to float32.
INFO:tensorflow:Create CheckpointSaverHook.
INFO:tensorflow:Graph was finalized.
INFO:tensorflow:Running local_init_op.
INFO:tensorflow:Done running local_init_op.
```

```
In [110]: print(time6)
```

```
0:03:23.036182
```

```
In [111]: model6 = dnn_clf6.predict(test)
model6['classes']
```

```
WARNING:tensorflow:float64 is not supported by many models, consider casting to float32.
INFO:tensorflow:Graph was finalized.
INFO:tensorflow:Restoring parameters from C:\\Users\\upsto\\AppData\\Local\\Temp\\tmpf14bz2hx\\model.ckpt-40000
INFO:tensorflow:Running local_init_op.
INFO:tensorflow:Done running local_init_op.
```

```
Out[111]: array([2, 0, 9, ..., 3, 9, 2], dtype=int64)
```

```
In [112]: data = {'ImageId': np.arange(1,28001), 'Label': model6['classes']}  
df = pd.DataFrame(data=data)  
df.head()
```

Out[112]:

	ImageId	Label
0	1	2
1	2	0
2	3	9
3	4	9
4	5	3

```
In [113]: df.to_csv('neural_network_model_6.csv', index=False)
```

```
In [115]: #Kaggle Results  
print ('Kaggle Results:')  
print('Name: James Casey')  
print('Score: 0.94700')
```

Kaggle Results:
Name: James Casey
Score: 0.94700

Model 7

```
In [117]: #model7 15 layers 100 nodes
start=datetime.datetime.now()

feat_cols = tf.contrib.learn.infer_real_valued_columns_from_input(x_train)
dnn_clf7 = tf.contrib.learn.DNNClassifier(hidden_units=[100,100,100,100,100,100,
                                                         100,100,100,100,100,100,
                                                         feature_columns=feat_cols])
dnn_clf7= tf.contrib.learn.SKCompat(dnn_clf7)
dnn_clf7.fit(x_train, y_train,batch_size=50,steps=40000)

end=datetime.datetime.now()

time7=(end-start)
```

```
WARNING:tensorflow:float64 is not supported by many models, consider casting
to float32.
INFO:tensorflow:Using default config.
WARNING:tensorflow:Using temporary folder as model directory: C:\Users\upsto
\AppData\Local\Temp\tmp0bmtpx4b
INFO:tensorflow:Using config: {'_task_type': None, '_task_id': 0, '_cluster_s
pec': <tensorflow.python.training.server_lib.ClusterSpec object at 0x00000259
AF2AD908>, '_master': '', '_num_ps_replicas': 0, '_num_worker_replicas': 0,
'_environment': 'local', '_is_chief': True, '_evaluation_master': '', '_train
_distribute': None, '_eval_distribute': None, '_experimental_max_worker_delay
_secs': None, '_device_fn': None, '_tf_config': gpu_options {
  per_process_gpu_memory_fraction: 1.0
}
, '_tf_random_seed': None, '_save_summary_steps': 100, '_save_checkpoints_sec
s': 600, '_log_step_count_steps': 100, '_protocol': None, '_session_config':
None, '_save_checkpoints_steps': None, '_keep_checkpoint_max': 5, '_keep_chec
kpoint_every_n_hours': 10000, '_model_dir': 'C:\\Users\\upsto\\AppData\\Local
\\Temp\\tmp0bmtpx4b', '_session_creation_timeout_secs': 7200}
WARNING:tensorflow:float64 is not supported by many models, consider casting
```

```
In [118]: print(time7)
```

```
0:01:56.196590
```

```
In [119]: model7 = dnn_clf7.predict(test)
model7['classes']
```

```
WARNING:tensorflow:float64 is not supported by many models, consider casting to
float32.
INFO:tensorflow:Graph was finalized.
INFO:tensorflow:Restoring parameters from C:\Users\upsto\AppData\Local\Temp\tmp
0bmtpx4b\model.ckpt-40000
INFO:tensorflow:Running local_init_op.
INFO:tensorflow:Done running local_init_op.
```

```
Out[119]: array([2, 0, 9, ..., 3, 9, 2], dtype=int64)
```

```
In [120]: data = {'ImageId': np.arange(1,28001), 'Label': model17['classes']}  
df = pd.DataFrame(data=data)  
df.head()
```

Out[120]:

	ImageId	Label
0	1	2
1	2	0
2	3	9
3	4	0
4	5	3

```
In [121]: df.to_csv('neural_network_model_7.csv', index=False)
```

```
In [122]: #Kaggle Results  
print ('Kaggle Results:')  
print('Name: James Casey')  
print('Score: 0.97128')
```

Kaggle Results:
Name: James Casey
Score: 0.97128

Model 8

```
In [123]: #model8 15 layers 300 nodes
start=datetime.datetime.now()

feat_cols = tf.contrib.learn.infer_real_valued_columns_from_input(x_train)
dnn_clf8 = tf.contrib.learn.DNNClassifier(hidden_units=[300,300,300,300,300,300,
                                                         300,300,300,300,300,300,
                                                         feature_columns=feat_cols])
dnn_clf8= tf.contrib.learn.SKCompat(dnn_clf8)
dnn_clf8.fit(x_train, y_train,batch_size=50,steps=40000)

end=datetime.datetime.now()

time8=(end-start)
```

```
INFO:tensorflow:Done running local_init_op.
INFO:tensorflow:Saving checkpoints for 0 into C:\Users\upsto\AppData\Local\Temp\tmp8byng\model.ckpt.
INFO:tensorflow:loss = 2.6205313, step = 1
INFO:tensorflow:global_step/sec: 119.466
INFO:tensorflow:loss = 2.672052, step = 101 (0.837 sec)
INFO:tensorflow:global_step/sec: 133.399
INFO:tensorflow:loss = 2.2846012, step = 201 (0.749 sec)
INFO:tensorflow:global_step/sec: 128.714
INFO:tensorflow:loss = 1.8794861, step = 301 (0.778 sec)
INFO:tensorflow:global_step/sec: 133.93
INFO:tensorflow:loss = 1.1506393, step = 401 (0.746 sec)
INFO:tensorflow:global_step/sec: 139.539
INFO:tensorflow:loss = 1.4604602, step = 501 (0.717 sec)
INFO:tensorflow:global_step/sec: 143.892
INFO:tensorflow:loss = 0.87088066, step = 601 (0.695 sec)
INFO:tensorflow:global_step/sec: 142.904
INFO:tensorflow:loss = 1.837913, step = 701 (0.700 sec)
INFO:tensorflow:global_step/sec: 138.678
```

```
In [124]: print(time8)
```

```
0:04:58.477245
```

```
In [125]: model8 = dnn_clf8.predict(test)
model8['classes']
```

```
WARNING:tensorflow:float64 is not supported by many models, consider casting to float32.
INFO:tensorflow:Graph was finalized.
INFO:tensorflow:Restoring parameters from C:\Users\upsto\AppData\Local\Temp\tmp8byng\model.ckpt-40000
INFO:tensorflow:Running local_init_op.
INFO:tensorflow:Done running local_init_op.
```

```
Out[125]: array([2, 0, 9, ..., 3, 9, 2], dtype=int64)
```

```
In [126]: data = {'ImageId': np.arange(1,28001), 'Label': model8['classes']}
df = pd.DataFrame(data=data)
df.head()
```

Out[126]:

	ImageId	Label
0	1	2
1	2	0
2	3	9
3	4	5
4	5	3

```
In [127]: df.to_csv('neural_network_model_8.csv', index=False)
```

```
In [130]: #Kaggle Results
print ('Kaggle Results:')
print('Name: jcasey18')
print('Score: 0.97442')
```

Kaggle Results:
Name: jcasey18
Score: 0.97442

```
In [137]: Model = ('Model 1','Model 2', 'Model 3', 'Model 4', 'Model 5', 'Model 6', 'Model
```

```
In [141]: Time=(time1,time2,time3,time4,time5,time6,time7,time8)
Time
```

```
Out[141]: (datetime.timedelta(seconds=60, microseconds=791029),
datetime.timedelta(seconds=69, microseconds=547991),
datetime.timedelta(seconds=69, microseconds=313966),
datetime.timedelta(seconds=72, microseconds=192160),
datetime.timedelta(seconds=95, microseconds=655664),
datetime.timedelta(seconds=203, microseconds=36182),
datetime.timedelta(seconds=116, microseconds=196590),
datetime.timedelta(seconds=298, microseconds=477245))
```

```
In [142]: Accuracy=(0.80600,0.90985,0.91528,0.93885,0.95914,0.94700,0.97128,0.97442)
```

```
Out[142]: (0.806, 0.90985, 0.91528, 0.93885, 0.95914, 0.947, 0.97128, 0.97442)
```

```
In [135]: Layers= (2,3,2,3,8,8,15,15)
```

```
In [136]: Nodes= (20,20,50,50,100,300,100,300)
```

In [146]:

```
-----
ModuleNotFoundError                                Traceback (most recent call last)
<ipython-input-146-1bd80ca68e3b> in <module>
----> 1 from users import Table

ModuleNotFoundError: No module named 'users'
```

In [151]:

```
data = (Layers,Nodes,Time,Accuracy)
index=('Layers','Nodes','Time','Accuracy')
Table =pd.DataFrame(data=data,columns= Model,index=index)
Table
```

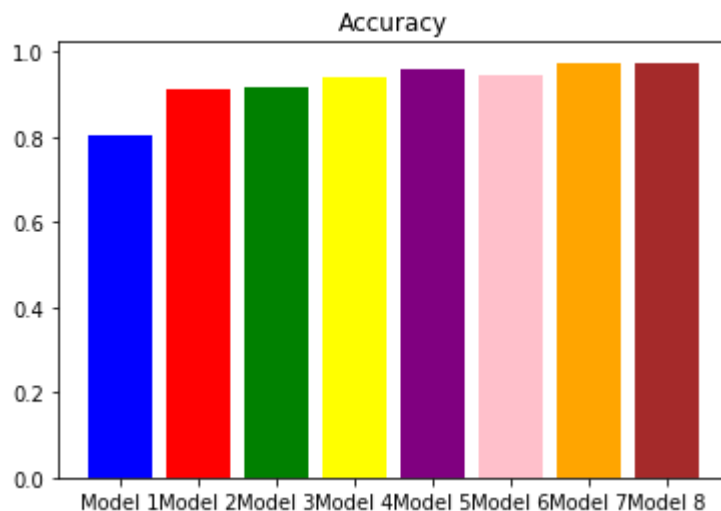
Out[151]:

	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
Layers	2	3	2	3	8	3
Nodes	20	20	50	50	100	30
Time	0:01:00.791029	0:01:09.547991	0:01:09.313966	0:01:12.192160	0:01:35.655664	0:03:23.036111
Accuracy	0.806	0.90985	0.91528	0.93885	0.95914	0.96528

In [162]:

```
plt.bar(Model,Accuracy,color=('blue','red','green','yellow','purple','pink','orange','brown'))
plt.title('Accuracy')
plt.show
```

Out[162]: <function matplotlib.pyplot.show(*args, **kw)>



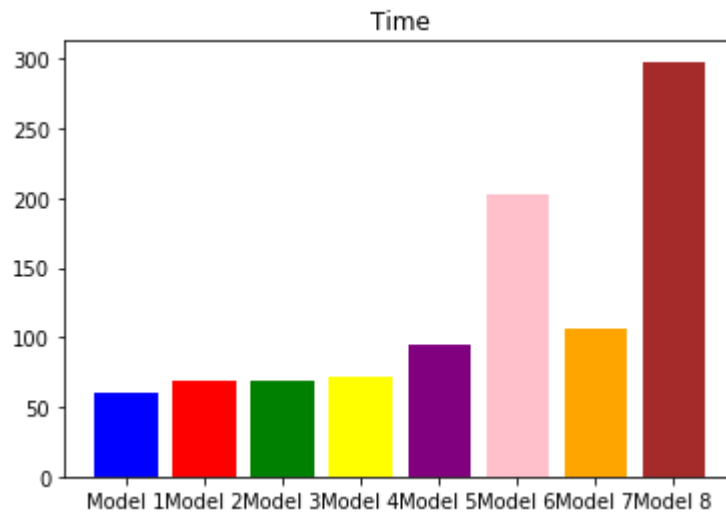
In [164]:

```
Time_round=(60,69,69,72,95,203,106,298)
```



```
In [165]: plt.bar(Model,Time_round,color=('blue','red','green','yellow','purple','pink','orange','darkred'))  
plt.title('Time')  
plt.show
```

```
Out[165]: <function matplotlib.pyplot.show(*args, **kw)>
```



```
In [ ]:
```