

[!\[\]\(919a2cb85b99741a73c0c31a427236a8_img.jpg\) Open in Colab](https://colab.research.google.com/github/oliverfoster27/Practical-Machine-Learning/blob/master/Week%208/Homework_Week_8.ipynb)
[\(https://colab.research.google.com/github/oliverfoster27/Practical-Machine-Learning/blob/master/Week%208/Homework_Week_8.ipynb\)](https://colab.research.google.com/github/oliverfoster27/Practical-Machine-Learning/blob/master/Week%208/Homework_Week_8.ipynb)

Deploy Keras Model to the Cloud

1. Download data
2. Hot Encode Categorical Columns
3. Develop Model (use: RMSprop(lr=, rho=, epsilon=, decay=))
4. Create Storage Bucket (should already exist)
5. Authenticate on Google Cloud Platform
6. Deploy your model using the gcloud tool

Download Data

```
In [1]: !wget https://storage.googleapis.com/nicksdemobucket/bank.csv
--2019-05-03 12:07:16-- https://storage.googleapis.com/nicksdemobucket/bank.csv
Resolving storage.googleapis.com (storage.googleapis.com)... 74.125.124.128,
2607:f8b0:4001:c03::80
Connecting to storage.googleapis.com (storage.googleapis.com)|74.125.124.128|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4610348 (4.4M) [application/octet-stream]
Saving to: 'bank.csv'

bank.csv          100%[=====] 4.40M  --.-KB/s   in 0.08s

2019-05-03 12:07:21 (57.1 MB/s) - 'bank.csv' saved [4610348/4610348]
```



```
In [16]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
df = pd.read_csv('bank.csv', sep=";")
df.head()
```

Out[16]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month
0	58	management	married	tertiary	no	2143	yes	no	unknown	5	may
1	44	technician	single	secondary	no	29	yes	no	unknown	5	may
2	33	entrepreneur	married	secondary	no	2	yes	yes	unknown	5	may
3	47	blue-collar	married	unknown	no	1506	yes	no	unknown	5	may
4	33	unknown	single	unknown	no	1	no	no	unknown	5	may

```
In [3]: df.describe()
```

Out[3]:

	age	balance	day	duration	campaign	pdays
count	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000
mean	40.936210	1362.272058	15.806419	258.163080	2.763841	40.197828
std	10.618762	3044.765829	8.322476	257.527812	3.098021	100.128746
min	18.000000	-8019.000000	1.000000	0.000000	1.000000	-1.000000
25%	33.000000	72.000000	8.000000	103.000000	1.000000	-1.000000
50%	39.000000	448.000000	16.000000	180.000000	2.000000	-1.000000
75%	48.000000	1428.000000	21.000000	319.000000	3.000000	-1.000000
max	95.000000	102127.000000	31.000000	4918.000000	63.000000	871.000000

Hot Encode Categorical Columns

```
In [4]: df.loc[df['y'] == 'no', 'y'] = 0
df.loc[df['y'] == 'yes', 'y'] = 1
df.head()
```

Out[4]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month
0	58	management	married	tertiary	no	2143	yes	no	unknown	5	may
1	44	technician	single	secondary	no	29	yes	no	unknown	5	may
2	33	entrepreneur	married	secondary	no	2	yes	yes	unknown	5	may
3	47	blue-collar	married	unknown	no	1506	yes	no	unknown	5	may
4	33	unknown	single	unknown	no	1	no	no	unknown	5	may

```
In [5]: # Get all categorical columns
categorical = df.select_dtypes(include='object').columns
print(categorical)
```

```
Index(['job', 'marital', 'education', 'default', 'housing', 'loan', 'contact',
       'month', 'poutcome'],
      dtype='object')
```

```
In [6]: # Hot encode all categorical columns
```

```
for i in categorical:
    new_cols = pd.get_dummies(df[i], prefix=i)
    df = df.drop(i, axis=1)
    df = df.join(new_cols)

df.head()
```

Out[6]:

	age	balance	day	duration	campaign	pdays	previous	y	job_admin.	job_blue-collar	...	month
0	58	2143	5	261	1	-1	0	0	0	0	0	...
1	44	29	5	151	1	-1	0	0	0	0	0	...
2	33	2	5	76	1	-1	0	0	0	0	0	...
3	47	1506	5	92	1	-1	0	0	0	0	1	...
4	33	1	5	198	1	-1	0	0	0	0	0	...

5 rows × 52 columns

◀	▶
---	---

```
In [7]: from sklearn.model_selection import train_test_split
from sklearn import preprocessing
min_max_scaler = preprocessing.MinMaxScaler()

train, test = train_test_split(df, test_size=0.1)

X_train = min_max_scaler.fit_transform(train.drop(['y'], axis=1))
y_train = train[['y']].values

X_test = min_max_scaler.fit_transform(test.drop(['y'], axis=1))
y_test = test[['y']].values

X_train.shape, y_train.shape, X_test.shape, y_test.shape
```

```
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/data.py:334: DataConversionWarning: Data with input dtype uint8, int64 were all converted to float64 by MinMaxScaler.
    return self.partial_fit(X, y)
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/data.py:334: DataConversionWarning: Data with input dtype uint8, int64 were all converted to float64 by MinMaxScaler.
    return self.partial_fit(X, y)
```

Out[7]: ((40689, 51), (40689, 1), (4522, 51), (4522, 1))

Develop Model

- Use RMSprop(lr=, rho=, epsilon=, decay=)
- Plot accuracy and loss over time

```
In [0]: from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, BatchNormalization, Dropout
from tensorflow.train import RMSPropOptimizer
from tensorflow import keras
import tensorflow as tf
```

```
In [59]: model = Sequential()

model.add(Dense(100, activation='relu', input_shape=(51,)))
model.add(BatchNormalization())
model.add(Dropout(0.3))
model.add(Dense(50, activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.2))
model.add(Dense(1, activation='sigmoid'))

optimizer = RMSPropOptimizer(learning_rate=10e-6, epsilon=1e-10, decay=0.8)
model.compile(optimizer=optimizer,
              loss='binary_crossentropy',
              metrics=['acc'])

h = model.fit(X_train, y_train, epochs=150, validation_split=0.2)
```

```
Train on 32551 samples, validate on 8138 samples
Epoch 1/150
32551/32551 [=====] - 6s 179us/sample - loss: 0.9205
- acc: 0.4899 - val_loss: 0.8027 - val_acc: 0.5240
Epoch 2/150
32551/32551 [=====] - 6s 178us/sample - loss: 0.8103
- acc: 0.5382 - val_loss: 0.7173 - val_acc: 0.5823
Epoch 3/150
32551/32551 [=====] - 6s 170us/sample - loss: 0.7464
- acc: 0.5838 - val_loss: 0.6583 - val_acc: 0.6374
Epoch 4/150
32551/32551 [=====] - 5s 155us/sample - loss: 0.6948
- acc: 0.6168 - val_loss: 0.6138 - val_acc: 0.6822
Epoch 5/150
32551/32551 [=====] - 5s 153us/sample - loss: 0.6533
- acc: 0.6499 - val_loss: 0.5744 - val_acc: 0.7293
Epoch 6/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.6153
- acc: 0.6778 - val_loss: 0.5383 - val_acc: 0.7793
Epoch 7/150
32551/32551 [=====] - 5s 153us/sample - loss: 0.5869
- acc: 0.7019 - val_loss: 0.5108 - val_acc: 0.8084
Epoch 8/150
32551/32551 [=====] - 5s 153us/sample - loss: 0.5563
- acc: 0.7303 - val_loss: 0.4847 - val_acc: 0.8309
Epoch 9/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.5274
- acc: 0.7544 - val_loss: 0.4540 - val_acc: 0.8493
Epoch 10/150
32551/32551 [=====] - 5s 155us/sample - loss: 0.5039
- acc: 0.7763 - val_loss: 0.4399 - val_acc: 0.8576
Epoch 11/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.4832
- acc: 0.7935 - val_loss: 0.4114 - val_acc: 0.8701
Epoch 12/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.4625
- acc: 0.8092 - val_loss: 0.3915 - val_acc: 0.8772
Epoch 13/150
32551/32551 [=====] - 5s 153us/sample - loss: 0.4456
- acc: 0.8215 - val_loss: 0.3762 - val_acc: 0.8822
Epoch 14/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.4307
- acc: 0.8311 - val_loss: 0.3603 - val_acc: 0.8887
Epoch 15/150
32551/32551 [=====] - 5s 153us/sample - loss: 0.4129
- acc: 0.8460 - val_loss: 0.3480 - val_acc: 0.8927
Epoch 16/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.4048
- acc: 0.8481 - val_loss: 0.3364 - val_acc: 0.8944
Epoch 17/150
32551/32551 [=====] - 5s 161us/sample - loss: 0.3908
- acc: 0.8578 - val_loss: 0.3277 - val_acc: 0.8962
Epoch 18/150
32551/32551 [=====] - 6s 179us/sample - loss: 0.3837
- acc: 0.8606 - val_loss: 0.3185 - val_acc: 0.8976
Epoch 19/150
32551/32551 [=====] - 5s 163us/sample - loss: 0.3757
```

```
- acc: 0.8651 - val_loss: 0.3127 - val_acc: 0.8979
Epoch 20/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.3695
- acc: 0.8677 - val_loss: 0.3057 - val_acc: 0.8989
Epoch 21/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.3651
- acc: 0.8713 - val_loss: 0.3000 - val_acc: 0.8991
Epoch 22/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.3585
- acc: 0.8721 - val_loss: 0.2964 - val_acc: 0.8986
Epoch 23/150
32551/32551 [=====] - 5s 153us/sample - loss: 0.3537
- acc: 0.8755 - val_loss: 0.2928 - val_acc: 0.8994
Epoch 24/150
32551/32551 [=====] - 5s 155us/sample - loss: 0.3512
- acc: 0.8762 - val_loss: 0.2896 - val_acc: 0.8991
Epoch 25/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.3470
- acc: 0.8779 - val_loss: 0.2877 - val_acc: 0.8999
Epoch 26/150
32551/32551 [=====] - 5s 155us/sample - loss: 0.3455
- acc: 0.8779 - val_loss: 0.2859 - val_acc: 0.8996
Epoch 27/150
32551/32551 [=====] - 6s 171us/sample - loss: 0.3411
- acc: 0.8786 - val_loss: 0.2842 - val_acc: 0.8992
Epoch 28/150
32551/32551 [=====] - 6s 190us/sample - loss: 0.3428
- acc: 0.8798 - val_loss: 0.2821 - val_acc: 0.8994
Epoch 29/150
32551/32551 [=====] - 5s 160us/sample - loss: 0.3379
- acc: 0.8807 - val_loss: 0.2813 - val_acc: 0.8996
Epoch 30/150
32551/32551 [=====] - 5s 155us/sample - loss: 0.3397
- acc: 0.8806 - val_loss: 0.2804 - val_acc: 0.8996
Epoch 31/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.3375
- acc: 0.8807 - val_loss: 0.2793 - val_acc: 0.9002
Epoch 32/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.3364
- acc: 0.8813 - val_loss: 0.2783 - val_acc: 0.9003
Epoch 33/150
32551/32551 [=====] - 6s 178us/sample - loss: 0.3326
- acc: 0.8815 - val_loss: 0.2772 - val_acc: 0.9000
Epoch 34/150
32551/32551 [=====] - 6s 173us/sample - loss: 0.3337
- acc: 0.8833 - val_loss: 0.2766 - val_acc: 0.9008
Epoch 35/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.3339
- acc: 0.8825 - val_loss: 0.2757 - val_acc: 0.9005
Epoch 36/150
32551/32551 [=====] - 5s 155us/sample - loss: 0.3330
- acc: 0.8822 - val_loss: 0.2748 - val_acc: 0.9011
Epoch 37/150
32551/32551 [=====] - 5s 155us/sample - loss: 0.3297
- acc: 0.8847 - val_loss: 0.2741 - val_acc: 0.9011
Epoch 38/150
32551/32551 [=====] - 5s 157us/sample - loss: 0.3300
```

```
- acc: 0.8834 - val_loss: 0.2731 - val_acc: 0.9008
Epoch 39/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.3304
- acc: 0.8835 - val_loss: 0.2725 - val_acc: 0.9007
Epoch 40/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.3263
- acc: 0.8846 - val_loss: 0.2719 - val_acc: 0.9011
Epoch 41/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.3257
- acc: 0.8834 - val_loss: 0.2711 - val_acc: 0.9014
Epoch 42/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.3258
- acc: 0.8828 - val_loss: 0.2703 - val_acc: 0.9019
Epoch 43/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.3238
- acc: 0.8852 - val_loss: 0.2694 - val_acc: 0.9013
Epoch 44/150
32551/32551 [=====] - 5s 155us/sample - loss: 0.3205
- acc: 0.8848 - val_loss: 0.2686 - val_acc: 0.9013
Epoch 45/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.3236
- acc: 0.8846 - val_loss: 0.2678 - val_acc: 0.9012
Epoch 46/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.3229
- acc: 0.8847 - val_loss: 0.2670 - val_acc: 0.9016
Epoch 47/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.3187
- acc: 0.8842 - val_loss: 0.2663 - val_acc: 0.9012
Epoch 48/150
32551/32551 [=====] - 5s 161us/sample - loss: 0.3189
- acc: 0.8864 - val_loss: 0.2656 - val_acc: 0.9010
Epoch 49/150
32551/32551 [=====] - 6s 179us/sample - loss: 0.3200
- acc: 0.8857 - val_loss: 0.2647 - val_acc: 0.9016
Epoch 50/150
32551/32551 [=====] - 5s 165us/sample - loss: 0.3188
- acc: 0.8855 - val_loss: 0.2641 - val_acc: 0.9012
Epoch 51/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.3138
- acc: 0.8862 - val_loss: 0.2634 - val_acc: 0.9019
Epoch 52/150
32551/32551 [=====] - 5s 153us/sample - loss: 0.3162
- acc: 0.8853 - val_loss: 0.2626 - val_acc: 0.9018
Epoch 53/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.3163
- acc: 0.8849 - val_loss: 0.2617 - val_acc: 0.9023
Epoch 54/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.3128
- acc: 0.8868 - val_loss: 0.2610 - val_acc: 0.9023
Epoch 55/150
32551/32551 [=====] - 5s 155us/sample - loss: 0.3102
- acc: 0.8859 - val_loss: 0.2602 - val_acc: 0.9018
Epoch 56/150
32551/32551 [=====] - 5s 153us/sample - loss: 0.3124
- acc: 0.8867 - val_loss: 0.2592 - val_acc: 0.9023
Epoch 57/150
32551/32551 [=====] - 5s 155us/sample - loss: 0.3135
```

```
- acc: 0.8873 - val_loss: 0.2586 - val_acc: 0.9018
Epoch 58/150
32551/32551 [=====] - 6s 172us/sample - loss: 0.3114
- acc: 0.8877 - val_loss: 0.2579 - val_acc: 0.9018
Epoch 59/150
32551/32551 [=====] - 5s 161us/sample - loss: 0.3076
- acc: 0.8892 - val_loss: 0.2570 - val_acc: 0.9019
Epoch 60/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.3075
- acc: 0.8870 - val_loss: 0.2562 - val_acc: 0.9023
Epoch 61/150
32551/32551 [=====] - 5s 155us/sample - loss: 0.3064
- acc: 0.8863 - val_loss: 0.2555 - val_acc: 0.9024
Epoch 62/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.3067
- acc: 0.8879 - val_loss: 0.2548 - val_acc: 0.9023
Epoch 63/150
32551/32551 [=====] - 5s 155us/sample - loss: 0.3059
- acc: 0.8873 - val_loss: 0.2541 - val_acc: 0.9024
Epoch 64/150
32551/32551 [=====] - 6s 171us/sample - loss: 0.3027
- acc: 0.8880 - val_loss: 0.2533 - val_acc: 0.9024
Epoch 65/150
32551/32551 [=====] - 6s 178us/sample - loss: 0.3048
- acc: 0.8880 - val_loss: 0.2523 - val_acc: 0.9021
Epoch 66/150
32551/32551 [=====] - 5s 155us/sample - loss: 0.3063
- acc: 0.8865 - val_loss: 0.2517 - val_acc: 0.9021
Epoch 67/150
32551/32551 [=====] - 5s 155us/sample - loss: 0.3029
- acc: 0.8868 - val_loss: 0.2510 - val_acc: 0.9030
Epoch 68/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.2989
- acc: 0.8894 - val_loss: 0.2501 - val_acc: 0.9032
Epoch 69/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.3002
- acc: 0.8887 - val_loss: 0.2498 - val_acc: 0.9033
Epoch 70/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.2988
- acc: 0.8886 - val_loss: 0.2487 - val_acc: 0.9038
Epoch 71/150
32551/32551 [=====] - 5s 155us/sample - loss: 0.3022
- acc: 0.8875 - val_loss: 0.2480 - val_acc: 0.9037
Epoch 72/150
32551/32551 [=====] - 5s 156us/sample - loss: 0.3000
- acc: 0.8888 - val_loss: 0.2471 - val_acc: 0.9034
Epoch 73/150
32551/32551 [=====] - 5s 155us/sample - loss: 0.2975
- acc: 0.8898 - val_loss: 0.2463 - val_acc: 0.9044
Epoch 74/150
32551/32551 [=====] - 5s 155us/sample - loss: 0.2957
- acc: 0.8898 - val_loss: 0.2459 - val_acc: 0.9037
Epoch 75/150
32551/32551 [=====] - 5s 155us/sample - loss: 0.2934
- acc: 0.8900 - val_loss: 0.2449 - val_acc: 0.9035
Epoch 76/150
32551/32551 [=====] - 5s 155us/sample - loss: 0.2958
```

```
- acc: 0.8881 - val_loss: 0.2444 - val_acc: 0.9037
Epoch 77/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.2954
- acc: 0.8893 - val_loss: 0.2437 - val_acc: 0.9040
Epoch 78/150
32551/32551 [=====] - 5s 155us/sample - loss: 0.2953
- acc: 0.8900 - val_loss: 0.2432 - val_acc: 0.9030
Epoch 79/150
32551/32551 [=====] - 5s 157us/sample - loss: 0.2925
- acc: 0.8892 - val_loss: 0.2426 - val_acc: 0.9040
Epoch 80/150
32551/32551 [=====] - 6s 179us/sample - loss: 0.2929
- acc: 0.8902 - val_loss: 0.2418 - val_acc: 0.9038
Epoch 81/150
32551/32551 [=====] - 6s 171us/sample - loss: 0.2918
- acc: 0.8897 - val_loss: 0.2413 - val_acc: 0.9037
Epoch 82/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.2900
- acc: 0.8904 - val_loss: 0.2409 - val_acc: 0.9042
Epoch 83/150
32551/32551 [=====] - 5s 155us/sample - loss: 0.2893
- acc: 0.8903 - val_loss: 0.2400 - val_acc: 0.9044
Epoch 84/150
32551/32551 [=====] - 5s 156us/sample - loss: 0.2901
- acc: 0.8889 - val_loss: 0.2394 - val_acc: 0.9049
Epoch 85/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.2884
- acc: 0.8900 - val_loss: 0.2388 - val_acc: 0.9045
Epoch 86/150
32551/32551 [=====] - 5s 153us/sample - loss: 0.2865
- acc: 0.8905 - val_loss: 0.2382 - val_acc: 0.9040
Epoch 87/150
32551/32551 [=====] - 6s 169us/sample - loss: 0.2897
- acc: 0.8901 - val_loss: 0.2380 - val_acc: 0.9049
Epoch 88/150
32551/32551 [=====] - 6s 189us/sample - loss: 0.2882
- acc: 0.8890 - val_loss: 0.2372 - val_acc: 0.9046
Epoch 89/150
32551/32551 [=====] - 5s 162us/sample - loss: 0.2848
- acc: 0.8914 - val_loss: 0.2366 - val_acc: 0.9045
Epoch 90/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.2842
- acc: 0.8900 - val_loss: 0.2361 - val_acc: 0.9048
Epoch 91/150
32551/32551 [=====] - 5s 155us/sample - loss: 0.2833
- acc: 0.8914 - val_loss: 0.2356 - val_acc: 0.9048
Epoch 92/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.2798
- acc: 0.8914 - val_loss: 0.2350 - val_acc: 0.9046
Epoch 93/150
32551/32551 [=====] - 5s 155us/sample - loss: 0.2815
- acc: 0.8920 - val_loss: 0.2344 - val_acc: 0.9051
Epoch 94/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.2797
- acc: 0.8940 - val_loss: 0.2338 - val_acc: 0.9051
Epoch 95/150
32551/32551 [=====] - 6s 172us/sample - loss: 0.2843
```

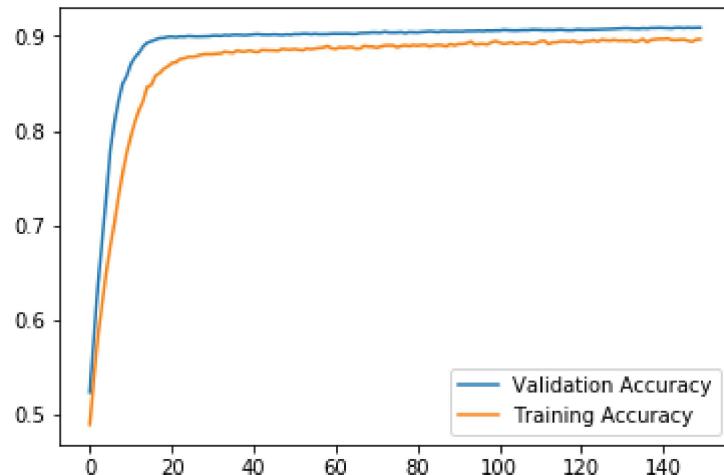
```
- acc: 0.8907 - val_loss: 0.2333 - val_acc: 0.9051
Epoch 96/150
32551/32551 [=====] - 6s 180us/sample - loss: 0.2791
- acc: 0.8914 - val_loss: 0.2328 - val_acc: 0.9054
Epoch 97/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.2788
- acc: 0.8916 - val_loss: 0.2326 - val_acc: 0.9050
Epoch 98/150
32551/32551 [=====] - 5s 155us/sample - loss: 0.2814
- acc: 0.8909 - val_loss: 0.2319 - val_acc: 0.9056
Epoch 99/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.2783
- acc: 0.8930 - val_loss: 0.2314 - val_acc: 0.9054
Epoch 100/150
32551/32551 [=====] - 5s 155us/sample - loss: 0.2755
- acc: 0.8946 - val_loss: 0.2309 - val_acc: 0.9058
Epoch 101/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.2777
- acc: 0.8931 - val_loss: 0.2305 - val_acc: 0.9056
Epoch 102/150
32551/32551 [=====] - 5s 156us/sample - loss: 0.2788
- acc: 0.8916 - val_loss: 0.2301 - val_acc: 0.9064
Epoch 103/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.2765
- acc: 0.8922 - val_loss: 0.2294 - val_acc: 0.9060
Epoch 104/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.2780
- acc: 0.8933 - val_loss: 0.2295 - val_acc: 0.9058
Epoch 105/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.2783
- acc: 0.8914 - val_loss: 0.2289 - val_acc: 0.9056
Epoch 106/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.2759
- acc: 0.8922 - val_loss: 0.2284 - val_acc: 0.9059
Epoch 107/150
32551/32551 [=====] - 5s 155us/sample - loss: 0.2751
- acc: 0.8928 - val_loss: 0.2281 - val_acc: 0.9061
Epoch 108/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.2773
- acc: 0.8926 - val_loss: 0.2278 - val_acc: 0.9060
Epoch 109/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.2744
- acc: 0.8914 - val_loss: 0.2273 - val_acc: 0.9066
Epoch 110/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.2748
- acc: 0.8924 - val_loss: 0.2272 - val_acc: 0.9065
Epoch 111/150
32551/32551 [=====] - 6s 178us/sample - loss: 0.2708
- acc: 0.8944 - val_loss: 0.2268 - val_acc: 0.9064
Epoch 112/150
32551/32551 [=====] - 6s 171us/sample - loss: 0.2700
- acc: 0.8949 - val_loss: 0.2263 - val_acc: 0.9060
Epoch 113/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.2744
- acc: 0.8913 - val_loss: 0.2260 - val_acc: 0.9062
Epoch 114/150
32551/32551 [=====] - 5s 157us/sample - loss: 0.2718
```

```
- acc: 0.8933 - val_loss: 0.2258 - val_acc: 0.9060
Epoch 115/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.2694
- acc: 0.8932 - val_loss: 0.2253 - val_acc: 0.9064
Epoch 116/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.2698
- acc: 0.8943 - val_loss: 0.2252 - val_acc: 0.9067
Epoch 117/150
32551/32551 [=====] - 6s 172us/sample - loss: 0.2684
- acc: 0.8943 - val_loss: 0.2249 - val_acc: 0.9061
Epoch 118/150
32551/32551 [=====] - 5s 161us/sample - loss: 0.2703
- acc: 0.8936 - val_loss: 0.2246 - val_acc: 0.9059
Epoch 119/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.2674
- acc: 0.8947 - val_loss: 0.2242 - val_acc: 0.9060
Epoch 120/150
32551/32551 [=====] - 5s 155us/sample - loss: 0.2695
- acc: 0.8934 - val_loss: 0.2238 - val_acc: 0.9067
Epoch 121/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.2692
- acc: 0.8929 - val_loss: 0.2236 - val_acc: 0.9065
Epoch 122/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.2703
- acc: 0.8943 - val_loss: 0.2232 - val_acc: 0.9064
Epoch 123/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.2664
- acc: 0.8944 - val_loss: 0.2230 - val_acc: 0.9065
Epoch 124/150
32551/32551 [=====] - 5s 155us/sample - loss: 0.2642
- acc: 0.8955 - val_loss: 0.2229 - val_acc: 0.9065
Epoch 125/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.2689
- acc: 0.8940 - val_loss: 0.2227 - val_acc: 0.9066
Epoch 126/150
32551/32551 [=====] - 5s 166us/sample - loss: 0.2647
- acc: 0.8960 - val_loss: 0.2223 - val_acc: 0.9070
Epoch 127/150
32551/32551 [=====] - 6s 179us/sample - loss: 0.2644
- acc: 0.8953 - val_loss: 0.2221 - val_acc: 0.9070
Epoch 128/150
32551/32551 [=====] - 5s 161us/sample - loss: 0.2650
- acc: 0.8951 - val_loss: 0.2219 - val_acc: 0.9072
Epoch 129/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.2620
- acc: 0.8959 - val_loss: 0.2214 - val_acc: 0.9073
Epoch 130/150
32551/32551 [=====] - 5s 155us/sample - loss: 0.2639
- acc: 0.8950 - val_loss: 0.2212 - val_acc: 0.9077
Epoch 131/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.2639
- acc: 0.8951 - val_loss: 0.2208 - val_acc: 0.9080
Epoch 132/150
32551/32551 [=====] - 5s 155us/sample - loss: 0.2612
- acc: 0.8949 - val_loss: 0.2207 - val_acc: 0.9082
Epoch 133/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.2632
```

```
- acc: 0.8954 - val_loss: 0.2205 - val_acc: 0.9078
Epoch 134/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.2621
- acc: 0.8938 - val_loss: 0.2204 - val_acc: 0.9077
Epoch 135/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.2619
- acc: 0.8946 - val_loss: 0.2202 - val_acc: 0.9076
Epoch 136/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.2627
- acc: 0.8965 - val_loss: 0.2198 - val_acc: 0.9081
Epoch 137/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.2626
- acc: 0.8934 - val_loss: 0.2197 - val_acc: 0.9077
Epoch 138/150
32551/32551 [=====] - 5s 155us/sample - loss: 0.2600
- acc: 0.8958 - val_loss: 0.2196 - val_acc: 0.9083
Epoch 139/150
32551/32551 [=====] - 5s 153us/sample - loss: 0.2608
- acc: 0.8962 - val_loss: 0.2194 - val_acc: 0.9083
Epoch 140/150
32551/32551 [=====] - 5s 155us/sample - loss: 0.2601
- acc: 0.8968 - val_loss: 0.2190 - val_acc: 0.9086
Epoch 141/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.2586
- acc: 0.8967 - val_loss: 0.2189 - val_acc: 0.9083
Epoch 142/150
32551/32551 [=====] - 6s 173us/sample - loss: 0.2570
- acc: 0.8974 - val_loss: 0.2188 - val_acc: 0.9080
Epoch 143/150
32551/32551 [=====] - 6s 178us/sample - loss: 0.2582
- acc: 0.8958 - val_loss: 0.2184 - val_acc: 0.9080
Epoch 144/150
32551/32551 [=====] - 5s 155us/sample - loss: 0.2599
- acc: 0.8956 - val_loss: 0.2183 - val_acc: 0.9086
Epoch 145/150
32551/32551 [=====] - 5s 153us/sample - loss: 0.2575
- acc: 0.8961 - val_loss: 0.2182 - val_acc: 0.9087
Epoch 146/150
32551/32551 [=====] - 5s 154us/sample - loss: 0.2575
- acc: 0.8962 - val_loss: 0.2181 - val_acc: 0.9083
Epoch 147/150
32551/32551 [=====] - 5s 165us/sample - loss: 0.2566
- acc: 0.8950 - val_loss: 0.2177 - val_acc: 0.9087
Epoch 148/150
32551/32551 [=====] - 6s 187us/sample - loss: 0.2615
- acc: 0.8940 - val_loss: 0.2176 - val_acc: 0.9085
Epoch 149/150
32551/32551 [=====] - 5s 165us/sample - loss: 0.2564
- acc: 0.8961 - val_loss: 0.2173 - val_acc: 0.9086
Epoch 150/150
32551/32551 [=====] - 5s 155us/sample - loss: 0.2586
- acc: 0.8965 - val_loss: 0.2173 - val_acc: 0.9087
```

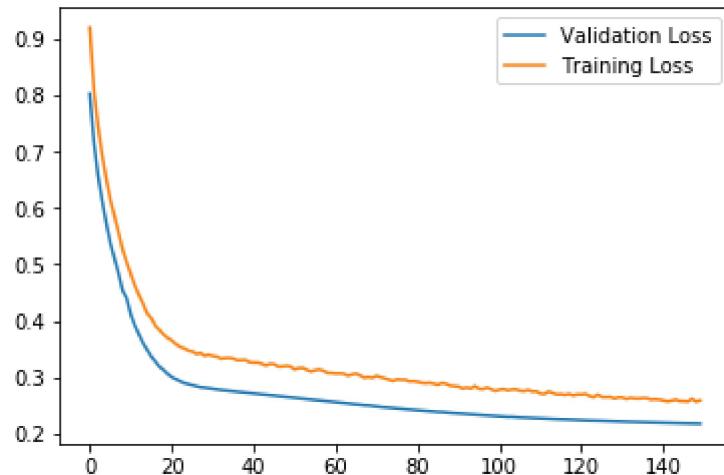
```
In [60]: plt.plot(h.history['val_acc'], label='Validation Accuracy')
plt.plot(h.history['acc'], label='Training Accuracy')
plt.legend()
```

```
Out[60]: <matplotlib.legend.Legend at 0x7f43cb268a58>
```



```
In [61]: plt.plot(h.history['val_loss'], label='Validation Loss')
plt.plot(h.history['loss'], label='Training Loss')
plt.legend()
```

```
Out[61]: <matplotlib.legend.Legend at 0x7f43cb1ee978>
```



Create Storage

```
In [62]: PROJECT_ID = "lofty-reserve-238701" #@param {type:"string"}  
  
# This must absolutely be a GLOBALLY UNIQUE name  
BUCKET_NAME = "oliver-ml-demo-bucket123" #@param {type:"string"}  
REGION = "us-central1" #@param {type:"string"}  
  
! gcloud config set project $PROJECT_ID  
! echo $PROJECT_ID
```

Updated property [core/project].
lofty-reserve-238701

Authenticate on Google Cloud Platform

- Click on link and enter secret key

```
In [0]: import sys  
  
if 'google.colab' in sys.modules:  
    from google.colab import auth as google_auth  
    google_auth.authenticate_user()  
else:  
    %env GOOGLE_APPLICATION_CREDENTIALS ''
```

```
In [63]: # Create Bucket  
!gsutil mb -p $PROJECT_ID -l $REGION gs://$BUCKET_NAME  
# Display what is in the bucket  
print('Bucket Content: ')  
!gsutil ls -al gs://$BUCKET_NAME
```

Creating gs://oliver-ml-demo-bucket123/...
ServiceException: 409 Bucket oliver-ml-demo-bucket123 already exists.
Bucket Content:
gs://oliver-ml-demo-bucket123/mcgilldemo/

```
In [64]: JOB_DIR = 'gs://' + BUCKET_NAME + '/mcgilldemo'  
print(JOB_DIR)
```

gs://oliver-ml-demo-bucket123/mcgilldemo

Export Model to Google Cloud Storage

```
In [65]: export_path = tf.contrib.saved_model.save_keras_model(model, JOB_DIR + '/keras_export')
print("Model exported to: ", export_path)

INFO:tensorflow:Signatures INCLUDED in export for Classify: None
INFO:tensorflow:Signatures INCLUDED in export for Regress: None
INFO:tensorflow:Signatures INCLUDED in export for Predict: None
INFO:tensorflow:Signatures INCLUDED in export for Train: ['train']
INFO:tensorflow:Signatures INCLUDED in export for Eval: None
WARNING:tensorflow:Export includes no default signature!
INFO:tensorflow>No assets to save.
INFO:tensorflow>No assets to write.
INFO:tensorflow:Signatures INCLUDED in export for Classify: None
INFO:tensorflow:Signatures INCLUDED in export for Regress: None
INFO:tensorflow:Signatures INCLUDED in export for Predict: None
INFO:tensorflow:Signatures INCLUDED in export for Train: None
INFO:tensorflow:Signatures INCLUDED in export for Eval: ['eval']
WARNING:tensorflow:Export includes no default signature!
INFO:tensorflow>No assets to save.
INFO:tensorflow>No assets to write.
INFO:tensorflow:Signatures INCLUDED in export for Classify: None
INFO:tensorflow:Signatures INCLUDED in export for Regress: None
INFO:tensorflow:Signatures INCLUDED in export for Predict: ['serving_default']
INFO:tensorflow:Signatures INCLUDED in export for Train: None
INFO:tensorflow:Signatures INCLUDED in export for Eval: None
INFO:tensorflow>No assets to save.
INFO:tensorflow>No assets to write.
INFO:tensorflow:SavedModel written to: gs://oliver-ml-demo-bucket123/mcgilldemo/keras_export/1556891334/saved_model.pb
Model exported to: b'gs://oliver-ml-demo-bucket123/mcgilldemo/keras_export/1556891334'
```

Deploy Model from GCS

```
In [66]: # Create model
MODEL_NAME = "mcgilldemo2"

! gcloud ml-engine models create $MODEL_NAME \
--regions $REGION
```

WARNING: The `gcloud ml-engine` commands have been renamed and will soon be removed. Please use `gcloud ai-platform` instead.

Created ml engine model [projects/lofty-reserve-238701/models/mcgilldemo2].

```
In [67]: # Deploy model
MODEL_VERSION = "v1"

# Get a list of directories in the `keras_export` parent directory
KERAS_EXPORT_DIRS = ! gsutil ls $JOB_DIR/keras_export/

# Pick the directory with the latest timestamp, in case you've trained
# multiple times
SAVED_MODEL_PATH = KERAS_EXPORT_DIRS[-1]

# Create model version based on that SavedModel directory
! gcloud ml-engine versions create $MODEL_VERSION \
    --model $MODEL_NAME \
    --runtime-version 1.13 \
    --python-version 3.5 \
    --framework tensorflow \
    --origin $SAVED_MODEL_PATH
```

WARNING: The `gcloud ml-engine` commands have been renamed and will soon be removed. Please use `gcloud ai-platform` instead.

Test your model with your test set

In [68]: # Create file with features to send to model

```
with open('test.json', 'w') as f:
    for item in X_test[:10]:
        item = list(item)
        f.write("%s\n" % item)
! cat test.json
```

```
[0.181818181818182, 0.03886471700537902, 0.6333333333333333, 0.032045240339
30255, 0.08108108108108109, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 1.0, 0.0, 1.0, 0.0, 0.0, 1.0, 0.0, 1.0, 1.
0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0,
0.0, 0.0, 0.0, 1.0]
[0.15584415584415584, 0.045088212076150616, 0.5333333333333333, 0.10650329877
474081, 0.05405405405405406, 0.22857142857142856, 0.007272727272727273, 0.0,
0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.
0, 1.0, 0.0, 1.0, 0.0, 1.0, 0.0, 1.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 1.0]
[0.5064935064935066, 0.047364652751005126, 0.4, 0.027961043041156142, 0.13513
513513513514, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 1.0, 0.0, 1.0, 0.0, 0.0, 1.0, 0.0, 0.0, 1.0, 0.0, 1.0, 0.0, 1.0,
0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 1.0]
[0.25974025974025977, 0.03798351416349985, 0.9333333333333333, 0.013823437009
1109, 0.40540540540540543, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 1.0, 0.0, 0.0, 1.0, 0.0, 1.0, 0.0, 0.
0, 1.0, 1.0, 0.0, 0.0, 1.0, 0.0, 0.0, 1.0, 0.0, 0.0, 1.0, 0.0, 1.0, 0.0, 1.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 1.0]
[0.2337662337662338, 0.03458721154375723, 0.5666666666666667, 0.0025133521834
747093, 0.16216216216216217, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 1.0, 0.0, 0.0, 1.0, 0.0, 1.0, 0.0, 1.0, 0.0, 0.
0, 1.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 1.0]
[0.15584415584415584, 0.03185181105542399, 0.3666666666666667, 0.051523719761
23154, 0.05405405405405406, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 1.0, 0.0, 1.0, 0.0, 1.0,
0.0, 1.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 1.0]
[0.3896103896103896, 0.06335481265260415, 0.6333333333333333, 0.0241910147659
44076, 0.05405405405405406, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 1.0, 0.0, 0.0, 1.0, 0.0,
0.0, 1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 1.0]
[0.16883116883116883, 0.04147160874593821, 0.5333333333333333, 0.068488846999
68583, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 1.0, 0.0, 0.0, 1.0, 0.0, 0.0, 1.0, 0.0, 0.0, 1.0, 0.0, 1.0, 0.0, 1.0,
0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 1.0]
[0.2207792207792208, 0.08022617539608232, 0.8, 0.08325479107759974, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 1.0, 0.0, 0.0, 1.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 1.0]
[0.2207792207792208, 0.03458721154375723, 0.4333333333333335, 0.144517750549
7958, 0.05405405405405406, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 1.0, 0.0, 0.0, 1.0, 0.0, 0.0, 1.0, 0.0,
0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 1.0]
```

```
In [69]: # Send file to model for prediction
! gcloud ml-engine predict \
--model ${MODEL_NAME} \
--version ${MODEL_VERSION} \
--json-instances test.json
```

WARNING: The `gcloud ml-engine` commands have been renamed and will soon be removed. Please use `gcloud ai-platform` instead.

```
DENSE_77
[0.010838687419891357]
[0.9132669568061829]
[0.014997661113739014]
[0.0015125870704650879]
[0.005704969167709351]
[0.0021437108516693115]
[0.021108150482177734]
[0.10229122638702393]
[0.1725328266620636]
[0.35158687829971313]
```

```
In [70]: y_test[:10]
```

```
Out[70]: array([[0],
 [1],
 [0],
 [0],
 [0],
 [0],
 [0],
 [0],
 [0],
 [0]])
```