Karthik Konath (kk28699), Kyle Polansky (kpp446)

EE 379K-DS Lab 7

Problem 1

Consider the dataset https://www.kaggle.com/c/GiveMeSomeCredit/data. This data set asks you to predict who will have a serious delinquency on their loan. If someone is a high-risk individual (i.e., the model predicts y = 1, they would be denied a loan).

1.0) Using some your new-found Kaggle competition skills, fit a good model to these data.

In [31]:
```
import pandas as pd
import numpy as np
import xgboost as xgb
from xgboost import XGBClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.feature_selection import f_regression, f_classif, SelectKBest
from sklearn.model_selection import train_test_split
from scipy.stats import chisquare
import statsmodels.api as sm

#import data
df_train = pd.read_csv("CS-data/cs-training.csv", header = 0, index_col =
0)
df_test = pd.read_csv("CS-data/cs-test.csv", header = 0, index_col = 0)

df_train = df_train.dropna()

x = df_train.drop(["SeriousDlqin2yrs"], axis = 1)
y = df_train.loc[:,"SeriousDlqin2yrs"]

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3)
```

In [39]:
```
xgb = XGBClassifier(
    min_child_weight=10,
    objective='binary:logistic',
    max_depth=5,
    max_delta_step=1.8,
    colsample_bytree=0.4,
    subsample=0.8,
    gamma=0.65,
    n_estimators=391).fit(x_train, y_train)
```

In [40]:
```
xgb.score(x_test, y_test)
```

```
c:\users\kyle\appdata\local\programs\python\python35\lib\site-packages\skl
```

```
earn\preprocessing\label.py:171: DeprecationWarning: The truth value of an
 empty array is ambiguous. Returning False, but in future this will result
 in an error. Use `array.size > 0` to check that an array is not empty.
  if diff:
```

Out[40]:
```
0.9347856212410964
```

In [4]:
```
lr = LogisticRegression().fit(x_train, y_train)
lr.score(x_test, y_test)
```

Out[4]:
```
0.9308777472908178
```

1.1) Model interpretability: What is the effect of MonthlyIncome to the prediction? Quantify as much as you can how 1000, 2000 or 3000 extra per month affect the probability of delinquency. Do this by fitting a simple model on the dataset and using your best model.

In [5]:
```
features = x.columns

monthlyIncomeIndex = x.columns.get_loc("MonthlyIncome")
print(lr.coef_[0][monthlyIncomeIndex])
```

```
-4.317674781216033e-05
```

Monthly income has a logistic regression coefficient weight of -4.317674781216033e-05. This corresponds to roughly .004% per dollar or a 4% less chance of delinquency per $1,000 monthly income.

1.2) What is the most important variable in predicting delinquency? What is the most important pair of variables? Make a data science argument supported by data

In [6]:
```
print(sorted(list(zip(features, xgb.feature_importances_)), key=lambda tup
: tup[1], reverse=True))
```

```
[('DebtRatio', 0.22822377), ('RevolvingUtilizationOfUnsecuredLines', 0.182
96483), ('MonthlyIncome', 0.18029381), ('age', 0.11871198), ('NumberOfOpen
CreditLinesAndLoans', 0.09036949), ('NumberOfTime30-59DaysPastDueNotWorse'
, 0.0488203), ('NumberRealEstateLoansOrLines', 0.04644606), ('NumberOfDepe
ndents', 0.038581394), ('NumberOfTimes90DaysLate', 0.035168424), ('NumberO
fTime60-89DaysPastDueNotWorse', 0.030419944)]
```

Based on the XGBoost model, DebtRatio is the most important feature in predicting delinquency

In [7]:
```
print(sorted(list(zip(features, lr.coef_[0])), key=lambda tup: tup[1], rev
erse=True))
```

```
[('NumberOfTime30-59DaysPastDueNotWorse', 0.515087065394077), ('NumberOfTi
mes90DaysLate', 0.3957512237462239), ('NumberOfDependents', 0.088500493323
925), ('NumberRealEstateLoansOrLines', 0.08045235295698679), ('MonthlyInco
me', -4.317674781216033e-05), ('RevolvingUtilizationOfUnsecuredLines', -0.
00015529225847365415), ('DebtRatio', -0.00022827923083007675), ('NumberOfO
penCreditLinesAndLoans', -0.008036141270535271), ('age', -0.02911478384849
3794), ('NumberOfTime60-89DaysPastDueNotWorse', -0.8751674922956593)]
```

Based on numerical values in logistic regression, each integer increment of 59DaysPastDueNotWorse contribures the most to predicting delinquency.

```
In [8]: selector = SelectKBest(f_classif, k=2).fit(x, y)

        print(sorted(list(zip(features, selector.scores_)), key=lambda tup: tup[1]
        , reverse=True))
```

```
[('NumberOfTime30-59DaysPastDueNotWorse', 1852.3219884153887), ('NumberOfT
imes90DaysLate', 1504.3845852968466), ('age', 1281.6301440601183), ('Numbe
rOfTime60-89DaysPastDueNotWorse', 1068.3797313737302), ('NumberOfDependent
s', 262.9093425447711), ('NumberOfOpenCreditLinesAndLoans', 90.65406549200
99), ('MonthlyIncome', 46.908782908080134), ('DebtRatio', 1.30283239255023
34), ('NumberRealEstateLoansOrLines', 1.0860875201815263), ('RevolvingUtil
izationOfUnsecuredLines', 0.6844006589671302)]
```

```
c:\users\kyle\appdata\local\programs\python\python35\lib\site-packages\skl
earn\utils\__init__.py:54: FutureWarning: Conversion of the second argumen
t of issubdtype from `int` to `np.signedinteger` is deprecated. In future,
 it will be treated as `np.int32 == np.dtype(int).type`.
  if np.issubdtype(mask.dtype, np.int):
```

The most important pair of variables are NumberOfTime30-59DaysPastDueNotWorse and NumberOfTimes90DaysLate based on a 2 best F-value selector.

1.3) The Age Discrimination in Employment Act (ADEA) forbids age discrimination against people who are age 40 or older. Does your good model (from part 0 above) discriminate against older people? Make the best argument you can

```
In [48]: x_test_old = x_test.copy()
         x_test_young = x_test.copy()

         x_test_old["age"] = x_test_old['age'].apply(lambda x: 60 if x < 40 else x)
         x_test_young["age"] = x_test_young['age'].apply(lambda x: 20 if x > 40 els
         e x)

         print("Old: ", np.mean(xgb.predict_proba(x_test_old)[:,1]))
         print("Young: ", np.mean(xgb.predict_proba(x_test_young)[:,1]))
```

```
Old:  0.06514755
Young:  0.113550946
```

By doing a simple age replace, that is, changing the age of people such that all of them are over 40 and another dataset where all of them are under 40, it can be seen that young people are nearly twice as likely to have loan delinquency. Therefore, the model does not discriminate negatively against older people, in fact, it positively discriminates against them.

1.4) Your manager asks if the number of dependents in the family (spouse, no of children) has an effect on loan delinquency. What do the data say? Calculate a p-value to express how confident you are.

```
In [9]: _, pval = f_regression(x[["NumberOfDependents"]],y)
        print(pval)
```

```
[4.60410806e-59]
```

In [17]:
```
sm.Logit(y, x).fit().summary()
```

```
Optimization terminated successfully.
         Current function value: 0.237361
         Iterations 7
```

Out[17]:

Logit Regression Results

Dep. Variable:	SeriousDlqin2yrs	**No. Observations:**	120269
Model:	Logit	**Df Residuals:**	120259
Method:	MLE	**Df Model:**	9
Date:	Mon, 09 Apr 2018	**Pseudo R-squ.:**	0.05924
Time:	01:44:12	**Log-Likelihood:**	-28547.
converged:	True	**LL-Null:**	-30345.
		LLR p-value:	0.000

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
RevolvingUtilizationOfUnsecuredLines	-8.752e-05	8.47e-05	-1.034	0.301	-0.000	7.84e-05
age	-0.0505	0.001	-86.202	0.000	-0.052	-0.049
NumberOfTime30-59DaysPastDueNotWorse	0.4874	0.012	40.493	0.000	0.464	0.511
DebtRatio	-0.0002	5.18e-05	-4.099	0.000	-0.000	-0.000
MonthlyIncome	-5.634e-05	3.52e-06	-15.999	0.000	-6.32e-05	-4.94e-05
NumberOfOpenCreditLinesAndLoans	-0.0197	0.003	-7.019	0.000	-0.025	-0.014
NumberOfTimes90DaysLate	0.3914	0.016	23.727	0.000	0.359	0.424
NumberRealEstateLoansOrLines	0.1214	0.011	10.830	0.000	0.099	0.143
NumberOfTime60-89DaysPastDueNotWorse	-0.8466	0.019	-44.122	0.000	-0.884	-0.809
NumberOfDependents	0.0365	0.010	3.827	0.000	0.018	0.055

The logistic regression p-value of NumberOfDependents is very small, so we can reject the null hypothesis and say that NumberOfDependents is a significant variable in calculating loan delinquency.

Problem 2

2.a) Create two random variables that are uncorrelated but dependent.

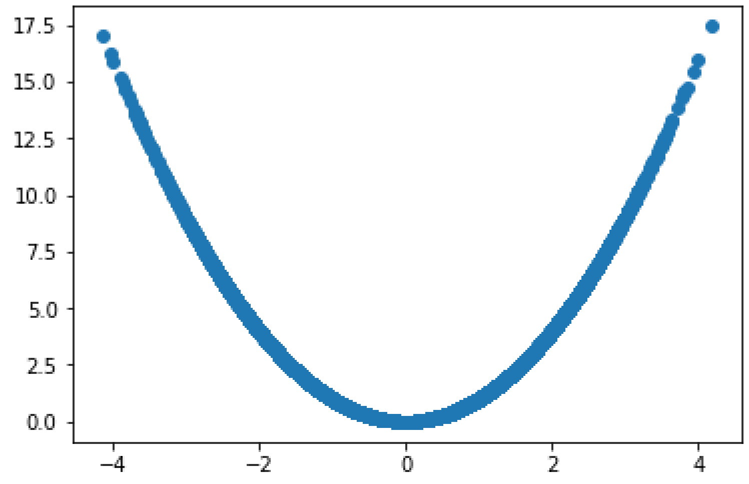
In [29]:
```
import numpy as np
from scipy import stats
x = np.random.normal(0,1,100000)
xsq = x*x
print("X vs X squared correlations {0}".format(stats.pointbiserialr(x,xsq)
.correlation))
```

```
X vs X squared correlations 0.000614463154342
```

Correlation is basically 0 but they're very clearly dependent (one is literally a function of the other)

2.b) Create two continuous random variables X, Y so that X and Y are strongly dependent but the best linear regression fit y = β1x + β0 has the optimal β1 = 0. Show a scatter plot of x, y pairs.

In [30]:
```
import matplotlib.pyplot as plt
plt.scatter(x,xsq)
plt.show()
A = np.vstack([x, np.ones(len(x))]).T
b1, b0 = np.linalg.lstsq(A,xsq)[0]
print b1
```



```
0.000867354944801
```

B1 is approximately 0

Problem 3

(Starting with MNIST) Install Tensorflow and Keras 2.0. Use the amazon instances and complete this tutorial: https://www.tensorflow.org/get_started/mnist/pros

In [6]:
```
from __future__ import absolute_import
from __future__ import division
from __future__ import print_function

# Imports
import numpy as np
import tensorflow as tf

tf.logging.set_verbosity(tf.logging.INFO)

# Our application logic will be added here
```

In [8]:
```
def cnn_model_fn(features, labels, mode):
  """Model function for CNN."""
  # Input Layer
  input_layer = tf.reshape(features["x"], [-1, 28, 28, 1])

  # Convolutional Layer #1
  conv1 = tf.layers.conv2d(
      inputs=input_layer,
      filters=32,
      kernel_size=[5, 5],
      padding="same",
      activation=tf.nn.relu)

  # Pooling Layer #1
  pool1 = tf.layers.max_pooling2d(inputs=conv1, pool_size=[2, 2], strides=
2)

  # Convolutional Layer #2 and Pooling Layer #2
  conv2 = tf.layers.conv2d(
      inputs=pool1,
      filters=64,
      kernel_size=[5, 5],
      padding="same",
      activation=tf.nn.relu)
  pool2 = tf.layers.max_pooling2d(inputs=conv2, pool_size=[2, 2], strides=
2)

  # Dense Layer
  pool2_flat = tf.reshape(pool2, [-1, 7 * 7 * 64])
  dense = tf.layers.dense(inputs=pool2_flat, units=1024, activation=tf.nn.
relu)
  dropout = tf.layers.dropout(
      inputs=dense, rate=0.4, training=mode == tf.estimator.ModeKeys.TRAIN
)

  # Logits Layer
  logits = tf.layers.dense(inputs=dropout, units=10)

  predictions = {
      # Generate predictions (for PREDICT and EVAL mode)
      "classes": tf.argmax(input=logits, axis=1),
      # Add `softmax_tensor` to the graph. It is used for PREDICT and by t
he
      # `logging_hook`.
      "probabilities": tf.nn.softmax(logits, name="softmax_tensor")
  }

  if mode == tf.estimator.ModeKeys.PREDICT:
    return tf.estimator.EstimatorSpec(mode=mode, predictions=predictions)

  # Calculate Loss (for both TRAIN and EVAL modes)
  loss = tf.losses.sparse_softmax_cross_entropy(labels=labels, logits=logi
ts)

  # Configure the Training Op (for TRAIN mode)
  if mode == tf.estimator.ModeKeys.TRAIN:
    optimizer = tf.train.GradientDescentOptimizer(learning_rate=0.001)
    train_op = optimizer.minimize(
        loss=loss,
        global_step=tf.train.get_global_step())
    return tf.estimator.EstimatorSpec(mode=mode, loss=loss, train_op=train
_op)

  # Add evaluation metrics (for EVAL mode)
  eval_metric_ops = {
      "accuracy": tf.metrics.accuracy(
          labels=labels, predictions=predictions["classes"])}
  return tf.estimator.EstimatorSpec(
      mode=mode, loss=loss, eval_metric_ops=eval_metric_ops)
```

In [9]:
```
# Load training and eval data
mnist = tf.contrib.learn.datasets.load_dataset("mnist")
train_data = mnist.train.images # Returns np.array
train_labels = np.asarray(mnist.train.labels, dtype=np.int32)
eval_data = mnist.test.images # Returns np.array
eval_labels = np.asarray(mnist.test.labels, dtype=np.int32)
```

```
WARNING:tensorflow:From c:\users\kyle\appdata\local\programs\python\python
35\lib\site-packages\tensorflow\contrib\learn\python\learn\datasets\base.p
y:198: retry (from tensorflow.contrib.learn.python.learn.datasets.base) is
 deprecated and will be removed in a future version.
Instructions for updating:
Use the retry module or similar alternatives.
WARNING:tensorflow:From <ipython-input-9-322f917c7d95>:2: load_dataset (fr
om tensorflow.contrib.learn.python.learn.datasets) is deprecated and will
be removed in a future version.
Instructions for updating:
Please use tf.data.
WARNING:tensorflow:From c:\users\kyle\appdata\local\programs\python\python
35\lib\site-packages\tensorflow\contrib\learn\python\learn\datasets\__init
__.py:80: load_mnist (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 c:\users\kyle\appdata\local\programs\python\python
35\lib\site-packages\tensorflow\contrib\learn\python\learn\datasets\mnist.
py:300: read_data_sets (from tensorflow.contrib.learn.python.learn.dataset
s.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 c:\users\kyle\appdata\local\programs\python\python
35\lib\site-packages\tensorflow\contrib\learn\python\learn\datasets\mnist.
py:260: maybe_download (from tensorflow.contrib.learn.python.learn.dataset
s.base) is deprecated and will be removed in a future version.
Instructions for updating:
Please write your own downloading logic.
WARNING:tensorflow:From c:\users\kyle\appdata\local\programs\python\python
35\lib\site-packages\tensorflow\contrib\learn\python\learn\datasets\base.p
y:219: retry.<locals>.wrap.<locals>.wrapped_fn (from tensorflow.contrib.le
arn.python.learn.datasets.base) is deprecated and will be removed in a fut
ure version.
Instructions for updating:
Please use urllib or similar directly.
Successfully downloaded train-images-idx3-ubyte.gz 9912422 bytes.
WARNING:tensorflow:From c:\users\kyle\appdata\local\programs\python\python
35\lib\site-packages\tensorflow\contrib\learn\python\learn\datasets\mnist.
py:262: extract_images (from tensorflow.contrib.learn.python.learn.dataset
s.mnist) is deprecated and will be removed in a future version.
Instructions for updating:
Please use tf.data to implement this functionality.
Extracting MNIST-data\train-images-idx3-ubyte.gz
Successfully downloaded train-labels-idx1-ubyte.gz 28881 bytes.
WARNING:tensorflow:From c:\users\kyle\appdata\local\programs\python\python
35\lib\site-packages\tensorflow\contrib\learn\python\learn\datasets\mnist.
py:267: extract_labels (from tensorflow.contrib.learn.python.learn.dataset
s.mnist) is deprecated and will be removed in a future version.
Instructions for updating:
Please use tf.data to implement this functionality.
Extracting MNIST-data\train-labels-idx1-ubyte.gz
Successfully downloaded t10k-images-idx3-ubyte.gz 1648877 bytes.
Extracting MNIST-data\t10k-images-idx3-ubyte.gz
Successfully downloaded t10k-labels-idx1-ubyte.gz 4542 bytes.
Extracting MNIST-data\t10k-labels-idx1-ubyte.gz
WARNING:tensorflow:From c:\users\kyle\appdata\local\programs\python\python
35\lib\site-packages\tensorflow\contrib\learn\python\learn\datasets\mnist.
py:290: DataSet.__init__ (from tensorflow.contrib.learn.python.learn.datas
ets.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.
```

In [11]:
```
# Create the Estimator
mnist_classifier = tf.estimator.Estimator(model_fn=cnn_model_fn, model_dir
="/tmp/mnist_convnet_model")

# Set up logging for predictions
tensors_to_log = {"probabilities": "softmax_tensor"}
logging_hook = tf.train.LoggingTensorHook(tensors=tensors_to_log, every_n_
iter=50)
```

```
INFO:tensorflow:Using default config.
INFO:tensorflow:Using config: {'_num_worker_replicas': 1, '_session_config
': None, '_cluster_spec': <tensorflow.python.training.server_lib.ClusterSp
ec object at 0x000002A6A8763358>, '_model_dir': '/tmp/mnist_convnet_model'
, '_evaluation_master': '', '_global_id_in_cluster': 0, '_task_type': 'wor
ker', '_tf_random_seed': None, '_save_checkpoints_steps': None, '_keep_che
ckpoint_every_n_hours': 10000, '_save_checkpoints_secs': 600, '_master': '
', '_service': None, '_is_chief': True, '_num_ps_replicas': 0, '_keep_chec
kpoint_max': 5, '_log_step_count_steps': 100, '_task_id': 0, '_save_summar
y_steps': 100}
```

In [12]:
```
# Train the model
train_input_fn = tf.estimator.inputs.numpy_input_fn(
    x={"x": train_data},
    y=train_labels,
    batch_size=100,
    num_epochs=None,
    shuffle=True)
mnist_classifier.train(
    input_fn=train_input_fn,
    steps=20000,
    hooks=[logging_hook])
```

```
INFO:tensorflow:Calling model_fn.
INFO:tensorflow:Done calling model_fn.
INFO:tensorflow:Create CheckpointSaverHook.
INFO:tensorflow:Graph was finalized.
INFO:tensorflow:Running local_init_op.
INFO:tensorflow:Done running local_init_op.
INFO:tensorflow:Saving checkpoints for 1 into /tmp/mnist_convnet_model\mod
el.ckpt.
INFO:tensorflow:probabilities = [[0.10211907 0.09459142 0.10281193 0.10690
533 0.09740437 0.09380776
  0.10432757 0.10091662 0.09783065 0.09928525]
 [0.11714772 0.10167047 0.09997449 0.10881621 0.09488136 0.09109379
  0.09173483 0.09869041 0.09259107 0.1033996 ]
 [0.10708418 0.10778041 0.09648488 0.11823592 0.09433167 0.09473843
  0.09806457 0.09234767 0.09702249 0.09390981]
 [0.09885558 0.09498819 0.10180819 0.11541405 0.09953026 0.08842915
  0.10026594 0.09916654 0.10274038 0.09880162]
 [0.10144363 0.10208189 0.10313495 0.10051085 0.09766779 0.09387813
  0.10767949 0.10121634 0.09805407 0.0943329 ]
 [0.09077626 0.10490326 0.10037149 0.10686517 0.10087152 0.08420128
  0.11770231 0.09923816 0.10122312 0.09384742]
 [0.10551151 0.10662676 0.08632844 0.12025583 0.10272693 0.09202426
  0.09856111 0.09772418 0.09816019 0.09208079]
 [0.10332873 0.09626526 0.09404328 0.10444486 0.097173   0.09748399
  0.11799263 0.09275579 0.09940431 0.09710811]
 [0.10307723 0.09300396 0.10548559 0.10741398 0.10130225 0.08391038
  0.09796106 0.0958765  0.10120749 0.11076149]
 [0.09425933 0.10332028 0.1016837  0.10426351 0.10552902 0.08978923
  0.10989461 0.09827872 0.09692224 0.09605934]
 [0.10480075 0.09856299 0.10094904 0.1143695  0.09473142 0.09071244
  0.10591083 0.10337116 0.09585948 0.09073237]
 [0.11259319 0.09475718 0.09583896 0.1009277  0.10170443 0.08779509
  0.10534477 0.10403119 0.09848441 0.098523  ]
 [0.08782435 0.10584448 0.09730952 0.10728153 0.1154416  0.09947158
  0.09465101 0.10620668 0.09065695 0.09531224]
 [0.10226241 0.10773838 0.10325195 0.10288883 0.10114922 0.09288935
  0.10434944 0.10782075 0.09257409 0.08507549]
 [0.09273035 0.09975679 0.09943322 0.1215883  0.10465423 0.07872894
  0.11177441 0.10859724 0.08586174 0.09687484]
 [0.09935837 0.10363512 0.09230519 0.10319164 0.0940263  0.0957547
  0.10989012 0.10807158 0.09981198 0.09395502]
```

```
[0.10190733 0.09081267 0.09318522 0.10085022 0.11364455 0.10128089
 0.11195632 0.10389052 0.09179875 0.09067341]
[0.09490248 0.10002992 0.10754458 0.09467378 0.10567174 0.08619203
 0.11637501 0.09914824 0.1017226  0.09373965]
[0.10492224 0.10517792 0.09518562 0.10223709 0.09837624 0.08814953
 0.1107308  0.10356964 0.09855818 0.09309272]
[0.10154433 0.10113212 0.10969415 0.11348004 0.10101701 0.10097784
 0.09691226 0.0965587  0.0843542  0.09432934]
[0.10077745 0.09826941 0.10710894 0.10716975 0.10332099 0.09273653
 0.11289874 0.09780201 0.08082684 0.09908935]
[0.11128236 0.09963717 0.09577844 0.10455886 0.09805733 0.10224145
 0.10251286 0.09153807 0.09047256 0.10392091]
[0.09765268 0.10263594 0.10905878 0.11357178 0.10003062 0.09463603
 0.10379115 0.09377585 0.09580054 0.0890466 ]
[0.11292413 0.10475052 0.09755799 0.10686235 0.09994871 0.08269829
 0.10924394 0.09581199 0.09514625 0.09505571]
[0.09939437 0.10045391 0.10664733 0.11063569 0.10615599 0.08777335
 0.09224236 0.10611439 0.09711757 0.09346506]
[0.10204396 0.10417541 0.09689768 0.10535213 0.10058476 0.09072531
 0.10583121 0.0974206  0.0957026  0.10126638]
[0.09604422 0.09424935 0.10271793 0.10994205 0.10554753 0.08523584
 0.10516921 0.11216129 0.08929978 0.09963279]
[0.10474586 0.1012052  0.08789206 0.10184171 0.1015103  0.09639093
 0.11059119 0.10126656 0.10079841 0.09375779]
[0.08513612 0.10174627 0.10033482 0.12128584 0.08796293 0.09181708
 0.11412981 0.10818747 0.09674691 0.09265276]
[0.09950164 0.09535746 0.09007044 0.12138325 0.10288627 0.09632567
 0.10916943 0.10472913 0.08139924 0.09917757]
[0.10244071 0.09890185 0.09598685 0.11513218 0.102803   0.08348831
 0.11585388 0.09724716 0.09465995 0.09348606]
[0.10603849 0.09690478 0.09456143 0.11348724 0.09992205 0.08371533
 0.09720334 0.10185584 0.09488692 0.11142456]
[0.10322529 0.10073125 0.10085662 0.10949387 0.09709825 0.08514775
 0.10422489 0.09443057 0.09722638 0.10756518]
[0.09962254 0.09619164 0.09990561 0.10943375 0.09583692 0.10515364
 0.10881093 0.09866548 0.09196267 0.09441681]
[0.10665184 0.10344418 0.09033799 0.11780053 0.0949176  0.10238613
 0.11450402 0.09751585 0.08813041 0.0843114 ]
[0.09916342 0.10741285 0.10804129 0.12395284 0.09391344 0.08781835
 0.09618603 0.09863583 0.09130467 0.09357125]
[0.10446467 0.10142862 0.10913796 0.10959228 0.10153983 0.09634598
 0.09474686 0.09359311 0.09311455 0.09603613]
[0.1005701  0.10649858 0.09756186 0.09956431 0.09768337 0.09828269
 0.09896798 0.10441788 0.09614707 0.10030619]
[0.09185692 0.09541222 0.10445349 0.12282709 0.09631124 0.08648694
 0.10996709 0.09827248 0.09435721 0.10005519]
[0.09574996 0.09713951 0.1039639  0.1082456  0.09383141 0.09232263
 0.10539678 0.10337768 0.10483425 0.09513826]
[0.10333878 0.1084795  0.09637593 0.11597386 0.09825121 0.09037223
 0.10067912 0.09123845 0.0997117  0.09557921]
[0.09731021 0.10624178 0.09891345 0.11680033 0.09579644 0.08839251
 0.10315494 0.09854318 0.10024773 0.09459939]
[0.10747598 0.10642876 0.09870778 0.11323284 0.09197045 0.08945241
 0.10434933 0.09456475 0.09572432 0.0980933 ]
[0.09430974 0.09732089 0.10020006 0.11497272 0.09940545 0.09650823
 0.10858361 0.10383438 0.08856768 0.09629718]
[0.10126344 0.09622441 0.10873025 0.10826613 0.09519422 0.10065964
```

```
 0.11237687 0.09813312 0.09014108 0.08901084]
[0.09321471 0.09651719 0.10193385 0.11422125 0.11305586 0.08944158
 0.10736288 0.09854139 0.08955504 0.09615621]
[0.09914898 0.10233595 0.09126262 0.10835122 0.09844403 0.08359289
 0.10619287 0.1104909  0.09742084 0.10275963]
[0.08998635 0.09586098 0.0960203  0.10614945 0.10111681 0.10780884
 0.10189824 0.10233708 0.10157598 0.09724598]
[0.10110912 0.09256688 0.10099474 0.10503257 0.11011306 0.0959881
 0.10307082 0.09944759 0.09659587 0.09508129]
[0.09753251 0.0953268  0.09520938 0.10157348 0.11529259 0.09612536
 0.09630016 0.11148833 0.0904029  0.10074845]
[0.09391828 0.10726447 0.09661282 0.10825549 0.09609797 0.09458248
 0.12197123 0.09330588 0.09197836 0.09601296]
[0.10027482 0.10299167 0.09902965 0.10809103 0.0986068  0.09102146
 0.09787414 0.10302363 0.09171674 0.10737003]
[0.10046384 0.09377248 0.10410514 0.09987599 0.09411433 0.09426016
 0.10868166 0.11130238 0.09693185 0.09649215]
[0.08703174 0.11031639 0.10928163 0.1180652  0.09406994 0.10194104
 0.10605063 0.09913786 0.08810555 0.08600011]
[0.10556011 0.1080794  0.09668224 0.11961254 0.10544243 0.08182628
 0.10127828 0.10069697 0.09011862 0.09070304]
[0.09359173 0.09816188 0.09809743 0.10266633 0.1064844  0.1032628
 0.09271954 0.10793567 0.09975512 0.09732515]
[0.09357674 0.10397231 0.09631975 0.10632321 0.09512274 0.09376967
 0.11378543 0.10418539 0.09575324 0.09719159]
[0.09286694 0.09545299 0.11214866 0.1141687  0.10529802 0.09486679
 0.0987443  0.10106851 0.09021361 0.09517144]
[0.09654477 0.09641064 0.11462981 0.11327233 0.09713115 0.09189127
 0.10805921 0.09967001 0.0924763  0.08991446]
[0.11358302 0.08788157 0.1028541  0.12176961 0.09179059 0.08672802
 0.10770372 0.10071579 0.09513991 0.09183359]
[0.09464922 0.09995999 0.09735707 0.10507688 0.10102596 0.09540296
 0.10678931 0.10244849 0.10075069 0.09653939]
[0.10601419 0.09872239 0.08955702 0.11648618 0.09395877 0.09232873
 0.1086478  0.10804467 0.09553324 0.09070706]
[0.10048049 0.10275014 0.09336545 0.11487268 0.11642472 0.08637403
 0.09726774 0.09237894 0.09672677 0.099359  ]
[0.09068797 0.09150179 0.10413183 0.11469252 0.09194768 0.10989084
 0.10427007 0.10336148 0.09134055 0.09817529]
[0.10120502 0.10134163 0.10105524 0.12129371 0.09517618 0.09470816
 0.11152728 0.08614314 0.10362819 0.08392151]
[0.09977987 0.11871185 0.09804991 0.09756269 0.09558921 0.10102399
 0.10896855 0.08873855 0.09249046 0.09908493]
[0.09680714 0.09760547 0.10810062 0.11950475 0.09982902 0.08681653
 0.10979357 0.09822123 0.08364481 0.09967688]
[0.10258753 0.09643816 0.09923876 0.10977982 0.10427188 0.09314063
 0.10416362 0.10643794 0.09427921 0.08966257]
[0.09116736 0.10048029 0.10268368 0.10611252 0.09808227 0.09116897
 0.10949534 0.09897342 0.10105492 0.10078128]
[0.10072945 0.0975184  0.10277011 0.10701371 0.100681   0.09569199
 0.10640401 0.09878267 0.09892271 0.09148587]
[0.10798543 0.09966004 0.09310967 0.10494207 0.10042226 0.09116293
 0.09982484 0.10933495 0.09331317 0.1002446 ]
[0.10462862 0.08786921 0.09961835 0.10497257 0.11667868 0.0837267
 0.10517749 0.1046698  0.09380587 0.09885269]
[0.09894296 0.10206512 0.09343813 0.11900644 0.10301915 0.09453503
 0.11310712 0.08775519 0.09009726 0.09803366]
 [0.10373787 0.10382175 0.09894121 0.11136547 0.09667973 0.08153103
  0.10792896 0.10335897 0.09044577 0.10218923]
 [0.09998674 0.11034537 0.10745297 0.10196918 0.09886277 0.09213173
  0.10451528 0.08956719 0.09179852 0.10337026]
 [0.09654239 0.11312252 0.09455799 0.10349559 0.09557034 0.09697506
  0.10739657 0.10129544 0.09422611 0.09681799]
 [0.10570974 0.10541054 0.10449674 0.10184923 0.09753066 0.09392082
  0.09755442 0.10119919 0.09106103 0.10126777]
 [0.09705757 0.09830947 0.09221859 0.12464702 0.09385154 0.087448
  0.11757844 0.11167027 0.08256634 0.09465276]
 [0.10770299 0.10189903 0.09982651 0.09459992 0.09567336 0.09318557
  0.1099028  0.10273123 0.09925158 0.095227  ]
 [0.09438451 0.09832577 0.10448062 0.11900963 0.09589997 0.08924072
  0.10493036 0.10181835 0.09690656 0.09500352]
 [0.10165548 0.10496029 0.09807847 0.10935243 0.1005362  0.08525224
  0.09654058 0.11068204 0.09144284 0.10149939]
 [0.10015599 0.10362999 0.10251591 0.10479161 0.09924905 0.0915637
  0.10567407 0.10609302 0.08687881 0.09944787]
 [0.09805039 0.09229384 0.09887938 0.10669215 0.10210653 0.09463124
  0.11434309 0.09908222 0.09826659 0.09565465]
 [0.1059784  0.098806   0.09342811 0.10390502 0.11135541 0.09009168
  0.10958841 0.09481039 0.10350929 0.08852725]
 [0.09653998 0.10990487 0.09166088 0.10618191 0.09963211 0.08429323
  0.11265086 0.10550718 0.09835843 0.09527066]
 [0.09723824 0.09831875 0.10137725 0.11784132 0.09707183 0.08705839
  0.11568519 0.10117001 0.08920104 0.09503797]
 [0.09581921 0.10191276 0.09918057 0.1036705  0.09881781 0.09322827
  0.11235929 0.10621243 0.09569683 0.0931023 ]
 [0.10269488 0.08990509 0.11279621 0.11191818 0.10301705 0.0961774
  0.10602562 0.1068196  0.08087657 0.08976946]
 [0.09767057 0.09787437 0.09921013 0.11400452 0.10749921 0.09145291
  0.1015484  0.10192571 0.09178482 0.09702937]
 [0.10185238 0.10396986 0.10167197 0.09821376 0.10039219 0.09811895
  0.10435243 0.10373108 0.08531094 0.1023864 ]
 [0.10297833 0.10293416 0.09883665 0.10903721 0.09510621 0.09454836
  0.1022571  0.096536   0.1066931  0.09107295]
 [0.10902151 0.09327911 0.09749234 0.10116696 0.11571831 0.09627186
  0.10141179 0.11058012 0.08744578 0.08761223]
 [0.10255752 0.1109453  0.09291004 0.09799042 0.09540416 0.09966701
  0.09937026 0.09790733 0.09766253 0.10558534]
 [0.10275444 0.10378145 0.10427213 0.11396703 0.1018071  0.08635771
  0.09868517 0.09927306 0.1005035  0.08859842]
 [0.10142638 0.08875386 0.10478579 0.11032081 0.09177967 0.09691443
  0.11342104 0.10357434 0.09495369 0.09407002]
 [0.10187141 0.09678323 0.09633524 0.10801318 0.10134303 0.08593059
  0.10112733 0.10036953 0.1074198  0.10080677]
 [0.10183635 0.10348693 0.09554161 0.10864086 0.09922448 0.09101544
  0.10579079 0.1048605  0.09105396 0.09854915]
 [0.09883998 0.09205312 0.09392568 0.11141607 0.10778334 0.09158172
  0.11039224 0.08920124 0.09346674 0.11133988]
 [0.09683761 0.10404776 0.10803196 0.11358518 0.09914424 0.08775132
  0.10224379 0.09727519 0.10062198 0.09046099]
 [0.09821697 0.10950684 0.10226536 0.0909093  0.10150857 0.08699922
  0.11999018 0.10264769 0.09149221 0.09646366]]
INFO:tensorflow:loss = 2.3084674, step = 0
INFO:tensorflow:probabilities = [[0.09892893 0.10612009 0.09803826 0.12069
315 0.11021108 0.08623704
```

```
INFO:tensorflow:loss = 0.100479655, step = 19900 (0.519 sec)
INFO:tensorflow:probabilities = [[0.00003175 0.0000069  0.99875844 0.00063
029 0.0000241  0.00001979
  0.00000499 0.00045126 0.00002694 0.00004555]
 [0.9999343  0.         0.00000035 0.00000006 0.         0.00005491
  0.0000006  0.00000033 0.00000793 0.00000128]
 [0.00000373 0.00000002 0.0000005  0.00020417 0.00002888 0.99937576
  0.00000785 0.00000003 0.00023238 0.00014663]
 [0.00000055 0.00000529 0.9999293  0.00005569 0.         0.00000017
  0.         0.0000004  0.00000862 0.0000001 ]
 [0.00000323 0.00000132 0.00004664 0.00006026 0.8622602  0.00016951
  0.00001731 0.00046899 0.00015976 0.13681284]
 [0.00014951 0.00045964 0.00054549 0.00049524 0.00001394 0.00058182
  0.00005093 0.00000229 0.9976846  0.00001657]
 [0.00006465 0.00002501 0.0014505  0.01676841 0.00001343 0.9807006
  0.00004669 0.00000848 0.00079357 0.00012863]
 [0.00000799 0.00000244 0.00408273 0.9930808  0.00000002 0.00257956
  0.00000003 0.00000331 0.00020754 0.00003566]
 [0.0061044  0.00000226 0.00051859 0.00000141 0.0001015  0.00214285
  0.9910858  0.00000132 0.00002753 0.0000143 ]
 [0.00008925 0.00001268 0.00052534 0.0027377  0.00630607 0.00223832
  0.00158617 0.00000088 0.9842976  0.00220607]
 [0.00004044 0.0000229  0.00003145 0.00048423 0.00127533 0.00062337
  0.00000396 0.9846906  0.00021331 0.01261451]
 [0.00050522 0.9744373  0.00053207 0.00029934 0.00002004 0.00002706
  0.00002246 0.00169502 0.02237526 0.00008635]
 [0.00001023 0.00000329 0.00011808 0.00050082 0.00212654 0.00001254
  0.00000008 0.04551433 0.00463535 0.94707876]
 [0.00006362 0.00019019 0.00000078 0.00485694 0.00000007 0.9946114
  0.00002289 0.00002863 0.00019067 0.00003497]
 [0.0000011  0.0000006  0.00001878 0.00008862 0.00000025 0.0000005
  0.         0.9996971  0.00000187 0.00019114]
 [0.00035867 0.00000001 0.00000027 0.00067181 0.00000033 0.9959721
  0.00000025 0.00000013 0.0029494  0.0000471 ]
 [0.00000768 0.00060574 0.00001038 0.00047878 0.41826734 0.00075146
  0.00000737 0.02453832 0.0002283  0.5551046 ]
 [0.00004157 0.9984035  0.00018066 0.00018784 0.00009402 0.00000119
  0.00007868 0.00027642 0.00068854 0.00004756]
 [0.0000683  0.00087703 0.00001447 0.00703662 0.00000112 0.991037
  0.00001117 0.00022505 0.00048662 0.00024256]
 [0.00062233 0.00000162 0.00022335 0.00000166 0.000083   0.00011262
  0.9986657  0.00000077 0.00028893 0.0000001 ]
 [0.00000414 0.00505259 0.00106498 0.1458113  0.04686075 0.00547498
  0.00002054 0.01107808 0.26673928 0.5178934 ]
 [0.00005559 0.00004023 0.0003003  0.00000313 0.00550094 0.00017073
  0.9938351  0.00001093 0.00004485 0.00003829]
 [0.00000644 0.991924   0.00032221 0.00122489 0.00001925 0.00000716
  0.00000789 0.00145751 0.00479443 0.00023618]
 [0.00000089 0.00079706 0.00112482 0.99777514 0.00000005 0.000155
  0.00000001 0.00000161 0.00010118 0.00004424]
 [0.00017992 0.00146888 0.00006898 0.00011526 0.00040147 0.9960621
  0.00112383 0.0000618  0.00048944 0.00002839]
 [0.00252529 0.00000019 0.00056046 0.00123814 0.00000274 0.9705683
  0.00005564 0.00002858 0.02449287 0.00052773]
 [0.02178815 0.7086734  0.00585015 0.0115626  0.00068075 0.01099941
  0.1349949  0.00246948 0.10076668 0.00221445]
 [0.00000748 0.99706393 0.00049408 0.0008219  0.00027586 0.00025251
  0.00019526 0.00014861 0.0006748  0.00006556]
 [0.         0.         0.00000071 0.00000096 0.         0.
  0.         0.9999907  0.00000004 0.00000765]
 [0.00000115 0.00001166 0.00003715 0.00021898 0.99611133 0.00001428
  0.00010081 0.00017977 0.00003011 0.00329472]
 [0.00001886 0.00001945 0.00032224 0.00019347 0.0292905  0.00074611
  0.00001672 0.00089744 0.00288276 0.9656124 ]
 [0.99998915 0.         0.00000015 0.00000002 0.         0.0000017
  0.00000695 0.         0.0000022  0.        ]
 [0.00000111 0.00000004 0.00000005 0.00002222 0.         0.99997354
  0.0000001  0.00000001 0.00000275 0.0000002 ]
 [0.00000601 0.99711764 0.00021426 0.00003621 0.00002371 0.00000376
  0.00038466 0.00003159 0.00212668 0.00005536]
 [0.9987174  0.00000004 0.00035197 0.0000104  0.00000001 0.00085783
  0.00000012 0.00000094 0.00001202 0.00004947]
 [0.00223324 0.0003032  0.00214847 0.80543345 0.00000255 0.18962793
  0.00004586 0.00003018 0.00005196 0.00012321]
 [0.00001982 0.9931344  0.00020361 0.00008835 0.00013558 0.00000649
  0.00000668 0.0025904  0.0037414  0.00007339]
 [0.00000017 0.00000006 0.00000093 0.00001477 0.00000015 0.00000056
  0.         0.9994715  0.00000248 0.00050933]
 [0.00007861 0.05121056 0.9273481  0.0054103  0.00083772 0.00043306
  0.00959061 0.00001321 0.00505486 0.00002287]
 [0.00000865 0.00000024 0.00002842 0.00007493 0.00841671 0.00000474
  0.0000008  0.00399678 0.00032534 0.98714334]
 [0.00008237 0.00000103 0.02607699 0.00383027 0.00162924 0.00062245
  0.00026954 0.0000223  0.9576874  0.00977834]
 [0.00005777 0.00001555 0.00000574 0.00263436 0.00000496 0.99662304
  0.00022212 0.000003   0.00037011 0.0000633 ]
 [0.00000329 0.00131288 0.96819043 0.02535398 0.00000024 0.00007038
  0.00000042 0.00503492 0.00003218 0.00000114]
 [0.0000018  0.00000168 0.00001885 0.00000707 0.9979697  0.00007847
  0.00007929 0.00065387 0.00042555 0.0007638 ]
 [0.9983492  0.00000007 0.00020238 0.00000686 0.00003706 0.00045629
  0.00090997 0.00000937 0.00001058 0.00001825]
 [0.00000108 0.00000012 0.00000019 0.00073246 0.00000016 0.9992545
  0.00000256 0.00000001 0.00000865 0.00000027]
 [0.00000003 0.00000085 0.98549706 0.01449644 0.00000002 0.00000001
  0.00000001 0.00000518 0.00000036 0.00000005]
 [0.00007259 0.00007558 0.00058982 0.0049274  0.00000848 0.00018302
  0.00034881 0.00000094 0.99373937 0.00005407]
 [0.00000084 0.         0.00000077 0.00002655 0.00000004 0.9999348
  0.0000001  0.         0.00003555 0.00000136]
 [0.0001536  0.987276   0.00104931 0.00124339 0.00064873 0.00083012
  0.00343361 0.0006492  0.00433344 0.00038265]
 [0.00032589 0.13280854 0.03967025 0.02489648 0.0021113  0.00095422
  0.00075219 0.00311818 0.79398406 0.00137896]
 [0.00021709 0.00110563 0.00096103 0.96648204 0.0000536  0.02354078
  0.00008937 0.00001211 0.00717914 0.00035907]
 [0.00004175 0.00000085 0.9994072  0.00043767 0.         0.00000165
  0.         0.00010968 0.00000009 0.00000097]
 [0.00000016 0.00000005 0.00000086 0.0000046  0.00051302 0.00000204
  0.00000003 0.00053179 0.00001333 0.9989341 ]
 [0.00000758 0.00065126 0.0002677  0.0032883  0.19324885 0.00005925
  0.00000261 0.71435666 0.00181774 0.08630002]
 [0.00000065 0.00000001 0.00000026 0.0000355  0.00328347 0.00009511
  0.00000002 0.00437302 0.00007652 0.9921354 ]
```

```
[0.9889864  0.0000049  0.00227229 0.00015763 0.0001611  0.00068302
 0.00285179 0.00013186 0.00360205 0.00114906]
[0.00000949 0.00003209 0.00047671 0.994327   0.00002539 0.00502405
 0.00000483 0.00000954 0.00001661 0.00007436]
[0.00000264 0.00000012 0.00958645 0.4130536  0.00038272 0.00426916
 0.00000016 0.00001614 0.5337845  0.03890447]
[0.00007488 0.00007146 0.00027894 0.00001141 0.00230263 0.00028214
 0.9960347  0.00000632 0.00092296 0.00001461]
[0.0000004  0.00000208 0.00000075 0.00000538 0.97620493 0.00001107
 0.00000825 0.00153785 0.00004613 0.02218315]
[0.00007312 0.00001493 0.0000698  0.00098202 0.18634051 0.00342693
 0.00002909 0.00563349 0.00153691 0.8018932 ]
[0.02527802 0.00002588 0.00000452 0.0043151  0.00002641 0.00258672
 0.00000058 0.00006086 0.96531004 0.00239186]
[0.00000573 0.9993686  0.00016566 0.00001597 0.00000887 0.00000099
 0.00021721 0.00001677 0.00019221 0.00000806]
[0.00064329 0.00005729 0.00160318 0.02269496 0.00562033 0.04170464
 0.01157903 0.00000776 0.90825325 0.00783627]
[0.00000028 0.00006799 0.00008881 0.00033517 0.98584366 0.00012774
 0.00000986 0.0002392  0.00058375 0.01270346]
[0.00007567 0.00000131 0.00157328 0.00004649 0.05335686 0.00057882
 0.00003658 0.02120278 0.00574385 0.9173843 ]
[0.00002301 0.00000017 0.9996295  0.00026772 0.         0.0000029
 0.00000001 0.0000761  0.00000007 0.00000049]
[0.00256583 0.0000517  0.0013545  0.00569431 0.00033871 0.0739354
 0.01069672 0.00012169 0.90402824 0.00121289]
[0.00060349 0.00065223 0.00089342 0.9963104  0.00000211 0.00144901
 0.00000029 0.00001414 0.00002759 0.00004741]
[0.0004994  0.00081585 0.00199945 0.00021699 0.91767657 0.00060883
 0.00145983 0.00835915 0.00531967 0.0630443 ]
[0.00003898 0.00000062 0.00059714 0.00206427 0.00000004 0.00000965
 0.00000003 0.00000002 0.9972824  0.00000684]
[0.00000284 0.00005275 0.00000126 0.00107063 0.00311369 0.0034234
 0.00000007 0.02692351 0.00025026 0.96516156]
[0.00000081 0.00000301 0.00003081 0.0001045  0.00005001 0.0000713
 0.00000007 0.99924386 0.00002956 0.00046614]
[0.00024963 0.00024529 0.00114685 0.00067705 0.03525353 0.00795386
 0.95061135 0.00003506 0.0035992  0.00022817]
[0.00000002 0.00000006 0.00000451 0.00000018 0.9998301  0.00000021
 0.00000049 0.00000024 0.00000236 0.00016173]
[0.00050018 0.00919132 0.00024449 0.00009211 0.00095577 0.981513
 0.00563826 0.00036036 0.00148018 0.00002423]
[0.00396186 0.00032742 0.00003647 0.04822357 0.00001509 0.61021185
 0.00000063 0.29172873 0.00680747 0.03868683]
[0.00001705 0.00000014 0.00000381 0.00007056 0.00000498 0.00026287
 0.         0.9740734  0.0000066  0.02556065]
[0.00049932 0.00061789 0.00061887 0.3132501  0.00323877 0.5797961
 0.00042425 0.00019893 0.00927594 0.09207982]
[0.00000001 0.0000006  0.99983835 0.00015074 0.00000385 0.00000008
 0.00000602 0.         0.00000039 0.        ]
[0.00000023 0.00001353 0.00000796 0.9945714  0.00007814 0.0050187
 0.00000324 0.00016194 0.0000915  0.00005337]
[0.00003318 0.00002578 0.00069267 0.00010003 0.0000054  0.00002998
 0.00000161 0.00003369 0.9989723  0.0001054 ]
[0.99992144 0.00000001 0.00000285 0.00000004 0.00000001 0.00002629
 0.00004534 0.00000004 0.0000006  0.00000349]
[0.00000044 0.00000004 0.00032063 0.00009695 0.00001644 0.00000105
```

```
  0.0000022  0.0000007  0.99930406 0.00025744]
 [0.00323532 0.03418044 0.001387   0.7047751  0.0001595  0.22848971
  0.000011   0.00321344 0.00884864 0.01569992]
 [0.00000649 0.999361   0.00017519 0.00006496 0.00000624 0.00000617
  0.00005596 0.00002024 0.00029713 0.00000671]
 [0.00000166 0.00000099 0.00000155 0.00113432 0.00017263 0.00182713
  0.00000003 0.00346856 0.00074967 0.99264354]
 [0.00000006 0.00000324 0.00003755 0.99988997 0.00000013 0.00005312
  0.         0.00000001 0.00001553 0.00000039]
 [0.9853592  0.00001184 0.00020426 0.00599548 0.0000005  0.0006498
  0.0000015  0.00287106 0.00002187 0.00488449]
 [0.00000365 0.00000105 0.00045088 0.00000052 0.00114059 0.00001053
  0.9970847  0.00000008 0.00130338 0.00000448]
 [0.00208014 0.00025189 0.09013508 0.00086625 0.8460492  0.00086837
  0.02934736 0.00059443 0.00460796 0.02519935]
 [0.00026258 0.00001191 0.9412471  0.05753253 0.00000199 0.00001618
  0.00000072 0.00065904 0.00020268 0.00006542]
 [0.00134562 0.00104104 0.02088414 0.10094088 0.00005149 0.00683522
  0.00002658 0.00040286 0.8682257  0.00024649]
 [0.00002534 0.9974118  0.00052983 0.00051979 0.00008333 0.0000079
  0.00000199 0.00103041 0.00032535 0.00006427]
 [0.02055208 0.00043456 0.00015866 0.00212585 0.00075815 0.21492194
  0.00004928 0.00380546 0.02487227 0.73232174]
 [0.00078485 0.00000756 0.00327833 0.91383106 0.00062225 0.00195497
  0.00000694 0.00315796 0.00436306 0.07199294]
 [0.00000448 0.00000156 0.00000997 0.00032827 0.00000199 0.00010042
  0.00000003 0.993001   0.00001188 0.0065404 ]
 [0.00112783 0.98446274 0.00264822 0.00129889 0.0001776  0.00043149
  0.00374543 0.00020733 0.00555342 0.00034693]
 [0.9998877  0.         0.00006237 0.00000016 0.00000078 0.0000122
  0.00000115 0.000017   0.00000979 0.00000878]] (0.260 sec)
INFO:tensorflow:Saving checkpoints for 20000 into /tmp/mnist_convnet_model
\model.ckpt.
INFO:tensorflow:Loss for final step: 0.14775714.
```

```
Out[12]: <tensorflow.python.estimator.estimator.Estimator at 0x2a6a8763978>
```

```
In [13]: # Evaluate the model and print results
         eval_input_fn = tf.estimator.inputs.numpy_input_fn(
             x={"x": eval_data},
             y=eval_labels,
             num_epochs=1,
             shuffle=False)
         eval_results = mnist_classifier.evaluate(input_fn=eval_input_fn)
         print(eval_results)
```

```
INFO:tensorflow:Calling model_fn.
INFO:tensorflow:Done calling model_fn.
INFO:tensorflow:Starting evaluation at 2018-04-09-02:51:43
INFO:tensorflow:Graph was finalized.
INFO:tensorflow:Restoring parameters from /tmp/mnist_convnet_model\model.c
kpt-20000
INFO:tensorflow:Running local_init_op.
INFO:tensorflow:Done running local_init_op.
INFO:tensorflow:Finished evaluation at 2018-04-09-02:51:43
INFO:tensorflow:Saving dict for global step 20000: accuracy = 0.9702, glob
al_step = 20000, loss = 0.10360358
```

```
{'loss': 0.10360358, 'accuracy': 0.9702, 'global_step': 20000}
```