Ubuntu Really working print statements e0fc788 2 minutes ago

0 contributors

429 lines (428 sloc) 16.3 KB Raw Blame History

In [1]:
```
import numpy as np
import pandas as pd
import sklearn as sk
import matplotlib.pyplot as plt
import seaborn as sns
import math
import scipy
from sklearn.linear_model import LinearRegression, Ridge, RidgeCV, ElasticNet, LassoCV, LassoLarsCV, Logis
ticRegression
from sklearn.model_selection import cross_val_score, KFold
import collections
from sklearn.metrics import accuracy_score, mean_squared_error, confusion_matrix
from sklearn.cross_decomposition import PLSRegression, PLSSVD
from sklearn.decomposition import PCA
import pandas as pd
import numpy as np
import seaborn as sk
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.decomposition import PCA
import collections
from collections import OrderedDict
from sklearn.model_selection import RandomizedSearchCV, GridSearchCV
from sklearn.metrics import mean_squared_error
from sklearn.preprocessing import OneHotEncoder
from sklearn import preprocessing
from sklearn.linear_model import Ridge
from sklearn.linear_model import SGDRegressor
from sklearn.svm import LinearSVR
from sklearn.svm import SVR
from sklearn.ensemble import ExtraTreesRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.cross_validation import KFold
from sklearn.cross_validation import StratifiedKFold
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import accuracy_score
from sklearn import model_selection
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import RandomForestRegressor
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"
import warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)
```

```
/usr/local/lib/python2.7/dist-packages/sklearn/cross_validation.py:41: DeprecationWarning: This module was
deprecated in version 0.18 in favor of the model_selection module into which all the refactored classes an
d functions are moved. Also note that the interface of the new CV iterators are different from that of thi
s module. This module will be removed in 0.20.
  "This module will be removed in 0.20.", DeprecationWarning)
```

In [2]:
```
# Problem 1
from tensorflow.examples.tutorials.mnist import input_data
mnist = input_data.read_data_sets('MNIST_data', one_hot=True)
```

```
Extracting MNIST_data/train-images-idx3-ubyte.gz
Extracting MNIST_data/train-labels-idx1-ubyte.gz
Extracting MNIST_data/t10k-images-idx3-ubyte.gz
Extracting MNIST_data/t10k-labels-idx1-ubyte.gz
```

In [3]:
```
import tensorflow as tf
sess = tf.InteractiveSession()
```

In [4]:
```
x = tf.placeholder(tf.float32, shape=[None, 784])
y_ = tf.placeholder(tf.float32, shape=[None, 10])

W = tf.Variable(tf.zeros([784,10]))
b = tf.Variable(tf.zeros([10]))

sess.run(tf.global_variables_initializer())

y = tf.matmul(x,W) + b

cross_entropy = tf.reduce_mean(
    tf.nn.softmax_cross_entropy_with_logits(labels=y_, logits=y))

train_step = tf.train.GradientDescentOptimizer(0.5).minimize(cross_entropy)

for _ in range(1000):
    batch = mnist.train.next_batch(100)
    train_step.run(feed_dict={x: batch[0], y_: batch[1]})

correct_prediction = tf.equal(tf.argmax(y,1), tf.argmax(y_,1))
accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))
print(accuracy.eval(feed_dict={x: mnist.test.images, y_: mnist.test.labels}))

def weight_variable(shape):
    initial = tf.truncated_normal(shape, stddev=0.1)
    return tf.Variable(initial)

def bias_variable(shape):
    initial = tf.constant(0.1, shape=shape)
    return tf.Variable(initial)

def conv2d(x, W):
    return tf.nn.conv2d(x, W, strides=[1, 1, 1, 1], padding='SAME')

def max_pool_2x2(x):
    return tf.nn.max_pool(x, ksize=[1, 2, 2, 1],
                          strides=[1, 2, 2, 1], padding='SAME')

W_conv1 = weight_variable([5, 5, 1, 32])
b_conv1 = bias_variable([32])

x_image = tf.reshape(x, [-1, 28, 28, 1])

h_conv1 = tf.nn.relu(conv2d(x_image, W_conv1) + b_conv1)
h_pool1 = max_pool_2x2(h_conv1)

W_conv2 = weight_variable([5, 5, 32, 64])
b_conv2 = bias_variable([64])

h_conv2 = tf.nn.relu(conv2d(h_pool1, W_conv2) + b_conv2)
h_pool2 = max_pool_2x2(h_conv2)

W_fc1 = weight_variable([7 * 7 * 64, 1024])
b_fc1 = bias_variable([1024])

h_pool2_flat = tf.reshape(h_pool2, [-1, 7*7*64])
h_fc1 = tf.nn.relu(tf.matmul(h_pool2_flat, W_fc1) + b_fc1)

keep_prob = tf.placeholder(tf.float32)
h_fc1_drop = tf.nn.dropout(h_fc1, keep_prob)

W_fc2 = weight_variable([1024, 10])
b_fc2 = bias_variable([10])

y_conv = tf.matmul(h_fc1_drop, W_fc2) + b_fc2

cross_entropy = tf.reduce_mean(
    tf.nn.softmax_cross_entropy_with_logits(labels=y_, logits=y_conv))
train_step = tf.train.AdamOptimizer(1e-4).minimize(cross_entropy)
correct_prediction = tf.equal(tf.argmax(y_conv, 1), tf.argmax(y_, 1))
accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))

with tf.Session() as sess:
  sess.run(tf.global_variables_initializer())
  for i in range(20000):
    batch = mnist.train.next_batch(50)
    if i % 100 == 0:
      train_accuracy = accuracy.eval(feed_dict={
          x: batch[0], y_: batch[1], keep_prob: 1.0})
      print('step %d, training accuracy %g' % (i, train_accuracy))
    train_step.run(feed_dict={x: batch[0], y_: batch[1], keep_prob: 0.5})

  print('test accuracy %g' % accuracy.eval(feed_dict={
      x: mnist.test.images, y_: mnist.test.labels, keep_prob: 1.0}))
```

```
0.9207
step 0, training accuracy 0.1
step 100, training accuracy 0.84
step 200, training accuracy 0.86
step 300, training accuracy 0.96
step 400, training accuracy 0.94
step 500, training accuracy 0.94
step 600, training accuracy 0.92
step 700, training accuracy 0.96
step 800, training accuracy 0.98
step 900, training accuracy 0.98
step 1000, training accuracy 0.98
step 1100, training accuracy 0.94
step 1200, training accuracy 0.94
step 1300, training accuracy 1
step 1400, training accuracy 1
step 1500, training accuracy 1
step 1600, training accuracy 1
step 1700, training accuracy 1
step 1800, training accuracy 0.96
step 1900, training accuracy 0.98
step 2000, training accuracy 1
step 2100, training accuracy 0.96
step 2200, training accuracy 1
step 2300, training accuracy 0.98
step 2400, training accuracy 0.98
step 2500, training accuracy 1
step 2600, training accuracy 1
step 2700, training accuracy 0.98
step 2800, training accuracy 0.96
step 2900, training accuracy 0.96
step 3000, training accuracy 0.98
step 3100, training accuracy 0.96
step 3200, training accuracy 0.98
step 3300, training accuracy 1
step 3400, training accuracy 0.98
step 3500, training accuracy 1
step 3600, training accuracy 1
step 3700, training accuracy 0.98
step 3800, training accuracy 0.96
step 3900, training accuracy 0.98
step 4000, training accuracy 1
step 4100, training accuracy 1
step 4200, training accuracy 1
step 4300, training accuracy 0.94
step 4400, training accuracy 1
step 4500, training accuracy 0.98
step 4600, training accuracy 1
step 4700, training accuracy 1
step 4800, training accuracy 0.98
step 4900, training accuracy 1
step 5000, training accuracy 0.98
step 5100, training accuracy 1
step 5200, training accuracy 1
step 5300, training accuracy 1
step 5400, training accuracy 1
step 5500, training accuracy 1
step 5600, training accuracy 1
step 5700, training accuracy 1
step 5800, training accuracy 1
step 5900, training accuracy 0.98
step 6000, training accuracy 0.98
step 6100, training accuracy 1
step 6200, training accuracy 1
step 6300, training accuracy 0.98
step 6400, training accuracy 1
step 6500, training accuracy 1
step 6600, training accuracy 1
step 6700, training accuracy 1
step 6800, training accuracy 0.98
step 6900, training accuracy 0.98
step 7000, training accuracy 1
step 7100, training accuracy 1
step 7200, training accuracy 0.98
step 7300, training accuracy 0.98
step 7400, training accuracy 1
step 7500, training accuracy 1
step 7600, training accuracy 1
step 7700, training accuracy 1
step 7800, training accuracy 1
step 7900, training accuracy 0.96
step 8000, training accuracy 1
step 8100, training accuracy 1
step 8200, training accuracy 1
step 8300, training accuracy 1
step 8400, training accuracy 1
step 8500, training accuracy 1
step 8600, training accuracy 0.98
step 8700, training accuracy 1
step 8800, training accuracy 1
step 8900, training accuracy 1
step 9000, training accuracy 0.98
step 9100, training accuracy 0.98
step 9200, training accuracy 1
step 9300, training accuracy 1
step 9400, training accuracy 0.98
step 9500, training accuracy 1
step 9600, training accuracy 1
step 9700, training accuracy 1
step 9800, training accuracy 1
step 9900, training accuracy 1
step 10000, training accuracy 1
step 10100, training accuracy 1
step 10200, training accuracy 1
step 10300, training accuracy 1
step 10400, training accuracy 1
step 10500, training accuracy 1
step 10600, training accuracy 1
step 10700, training accuracy 1
step 10800, training accuracy 1
step 10900, training accuracy 1
step 11000, training accuracy 0.98
step 11100, training accuracy 1
step 11200, training accuracy 1
step 11300, training accuracy 1
step 11400, training accuracy 0.98
step 11500, training accuracy 0.98
step 11600, training accuracy 1
step 11700, training accuracy 1
step 11800, training accuracy 0.98
step 11900, training accuracy 1
step 12000, training accuracy 0.98
step 12100, training accuracy 1
step 12200, training accuracy 1
step 12300, training accuracy 1
step 12400, training accuracy 1
step 12500, training accuracy 1
step 12600, training accuracy 1
step 12700, training accuracy 1
step 12800, training accuracy 1
step 12900, training accuracy 0.98
step 13000, training accuracy 0.98
step 13100, training accuracy 1
step 13200, training accuracy 1
step 13300, training accuracy 1
step 13400, training accuracy 0.98
step 13500, training accuracy 0.98
step 13600, training accuracy 1
step 13700, training accuracy 1
step 13800, training accuracy 1
step 13900, training accuracy 1
step 14000, training accuracy 0.96
step 14100, training accuracy 1
step 14200, training accuracy 1
step 14300, training accuracy 0.98
step 14400, training accuracy 1
step 14500, training accuracy 1
step 14600, training accuracy 1
step 14700, training accuracy 1
step 14800, training accuracy 1
step 14900, training accuracy 0.98
step 15000, training accuracy 1
step 15100, training accuracy 1
step 15200, training accuracy 1
step 15300, training accuracy 1
step 15400, training accuracy 0.98
step 15500, training accuracy 1
step 15600, training accuracy 1
step 15700, training accuracy 1
step 15800, training accuracy 1
step 15900, training accuracy 1
step 16000, training accuracy 1
step 16100, training accuracy 0.98
step 16200, training accuracy 1
step 16300, training accuracy 1
step 16400, training accuracy 1
step 16500, training accuracy 1
step 16600, training accuracy 1
step 16700, training accuracy 1
step 16800, training accuracy 1
step 16900, training accuracy 0.98
step 17000, training accuracy 1
step 17100, training accuracy 0.98
step 17200, training accuracy 1
step 17300, training accuracy 1
step 17400, training accuracy 1
step 17500, training accuracy 1
step 17600, training accuracy 0.98
step 17700, training accuracy 1
step 17800, training accuracy 0.98
step 17900, training accuracy 1
step 18000, training accuracy 1
step 18100, training accuracy 1
step 18200, training accuracy 1
step 18300, training accuracy 1
step 18400, training accuracy 1
step 18500, training accuracy 1
step 18600, training accuracy 1
step 18700, training accuracy 1
step 18800, training accuracy 1
step 18900, training accuracy 1
step 19000, training accuracy 1
step 19100, training accuracy 1
step 19200, training accuracy 1
step 19300, training accuracy 1
step 19400, training accuracy 1
step 19500, training accuracy 1
step 19600, training accuracy 1
step 19700, training accuracy 1
step 19800, training accuracy 1
step 19900, training accuracy 1
test accuracy 0.9918
```