

Word2vec实操--用每日新闻预测金融市场变化

原创 ManZZH 阿华code 7月14日

干货满满，注意休息。

```
1 import pandas as pd
2 import numpy as np
3 from sklearn.metrics import roc_auc_score
4 from datetime import date
5 # 读入数据
6 data = pd.read_csv('./Combined_News_DJIA.csv')
```

data.head()

	Date	Label	Top1	Top2	Top3	Top4	Top5	Top6	Top7
0	2008-08-08	0	b'Georgia 'downs two Russian warplanes' as cou...	b'BREAKING: Musharraf to be impeached.'	b'Russia Today: Columns of troops roll into So...	b'Russian tanks are moving towards the capital...	b'Afghan children raped with 'impunity,' U.N. ...	b'150 Russian tanks have entered South Ossetia...	b'Breaking: Georgia invades South Ossetia, Rus...
1	2008-08-11	1	b'Why wont America and Nato help us? If they w...	b'Bush puts foot down on Georgian conflict'	b'Jewish Georgian minister: Thanks to Israeli ...	b'Georgian army flees in disarray as Russians ...	b'Olympic opening ceremony fireworks 'faked''	b'What were the Mossad with fraudulent New Zea...	b'Russia angered by Israeli military sale to G...
2	2008-08-12	0	b'Remember that adorable 9-year-old	b'Russia 'ends Georgia	b'"If we had no sexual harassment we would	b'Al-Qa'eda is losing support in Iran	b'Ceasefire in Georgia: Putin Outmaneuvers	b'Why Microsoft and Intel tried to kill	b'Stratfor: The Russo-Georgia war and

```
1 #我们可以先把数据给分成Training/Testing data
2 train = data[data['Date'] < '2015-01-01']
3 test = data[data['Date'] > '2014-12-31']
4 #columns 和 .index 两个属性返回数据集的列索引和行索引
5 X_train = train[train.columns[2:]]
6 X_train
```

	Top1	Top2	Top3	Top4	Top5	Top6	Top7
0	b'Georgia 'downs two Russian warplanes' as cou...	b'BREAKING: Musharraf to be impeached.'	b'Russia Today: Columns of troops roll into So...	b'Russian tanks are moving towards the capital...	b'Afghan children raped with 'impunity,' U.N. ...	b'150 Russian tanks have entered South Ossetia...	b'Breaking: Georgia invades South Ossetia, Rus...
1	b'Why wont America and Nato help us? If they w...	b'Bush puts foot down on Georgian conflict'	b'Jewish Georgian minister: Thanks to Israeli ...	b'Georgian army flees in disarray as Russians ...	b'Olympic opening ceremony fireworks 'faked''	b'What were the Mossad with fraudulent New Zea...	b'Russia angered by Israeli military sale to G...
2	b'Remember that adorable 9-year-old	b'Russia 'ends Georgia operation''	b''If we had no sexual harassment we would hav...	b'Al-Qa'eda is losing support in Iraq because	b'Ceasefire in Georgia: Putin Outmaneuvers	b'Why Microsoft and Intel tried to kill	b'Stratfor: The Russo-Georgian War

```

1 #flatten转变数据为一维数组就会得到list of sentences。astype强制类型转换
2 #同时我们的X_train和X_test可不能随便flatten，他们需要与y_train和y_test对应
3 corpus = X_train.values.flatten().astype(str)
4 X_train = X_train.values.astype(str)
5 X_train = np.array([' '.join(x) for x in X_train])
6 X_test = test[test.columns[2:]]
7 X_test = X_test.values.astype(str)
8 X_test = np.array([' '.join(x) for x in X_test])
9 y_train = train['Label'].values
10 y_test = test['Label'].values
11 y_train

```

```
90]: corpus[:3]

array(['b"Georgia \'downs two Russian warplanes\' as countries move to brink of war"',
      'b"BREAKING: Musharraf to be impeached."',
      'b"Russia Today: Columns of troops roll into South Ossetia; footage from fighting (YouTube)"'],
      dtype='<U312')
```

```
91]: X_train[:1]

aving several hundred people killed. [VIDEO]\' b\'Did the U.S. Prep Georgia for War with Russia?\' b\'Rice Gives Green Light for Israel to Attack Iran: Says U.S. has no veto over Israeli military ops\' b\'Announcing:Class Action Lawsuit on Behalf of American Public Against the FBI\' b\'So---Russia and Georgia are at war and the NYT\'s top story is opening ceremonies of the Olympics? What a fucking disgrace and yet further proof of the decline of journalism." b\'China tells Bush to stay out of other countries\' affairs" b\'Did World War III start today?\' b\'Georgia Invades South Ossetia - if Russia gets involved, will NATO absorb Georgia and unleash a full scale war?\' b\'Al-Qaeda Faces Islamist Backlash\' b\'Condoleezza Rice: "The US would not act to prevent an Israeli strike on Iran." Israeli Defense Minister Ehud Barak: "Israel is prepared for uncompromising victory in the case of military hostilities."\' b\'This is a busy day: The European Union has approved new sanctions against Iran in protest at its nuclear programme.\' b\'Georgia will withdraw 1,000 soldiers from Iraq to help fight off Russian forces in Georgia\'s breakaway region of South Ossetia" b\'Why the Pentagon Thinks Attacking Iran is a Bad Idea - US News & World Report\' b\'Caucasus in crisis: Georgia invades South Ossetia\' b\'Indian shoe manufactory - And again in a series of "you do not like your work?"\' b\'Visitors Suffering from Mental Illnesses Banned from Olympics\' b\'No Help for Mexico\'s Kidnapping Surge"'],
      dtype='<U4424')
```

 阿华code

```
1 #tokenize 分割单词
2 from nltk.tokenize import word_tokenize
3 #报错 Resource punkt not found.
4 #运行 nltk.download('punkt')
5 corpus = [word_tokenize(x) for x in corpus]
6 X_train = [word_tokenize(x) for x in X_train]
7 X_test = [word_tokenize(x) for x in X_test]
```

```

X_train[:2]

[['b',
  "'",
  'Georgia',
  "'downs",
  'two',
  'Russian',
  'warplanes',
  "'",
  'as',
  'countries',
  'move',
  'to',
  'brink',
  'of',
  'war',
  "'",
  "b'BREAKING",
  ':',
  'Musharraf',
  'to',
  'be',
  'impeached',
  '.',
  "'",
  "b'Russia",
  'Today',
  ':',
  'Columns',
  'of',
  'troops',

```



```

1 # 数字
2 import re
3 def hasNumbers(inputString):
4     return bool(re.search(r'\d', inputString))
5 # 特殊符号
6 def isSymbol(inputString):
7     return bool(re.match(r'^\w', inputString))
8
9 # Lemma
10 from nltk.stem import WordNetLemmatizer

```

```
11 wordnet_lemmatizer = WordNetLemmatizer()
12
13 def check(word):
14     """
15     如果需要这个单词，则True
16     如果应该去除，则False
17     """
18     word= word.lower()
19     if word in stop:
20         return False
21     elif hasNumbers(word) or isSymbol(word):
22         return False
23     else:
24         return True
25 # 把上面的方法综合起来
26 def preprocessing(sen):
27     res = []
28     for word in sen:
29         if check(word):
30             # 这一段的用处仅仅是去除python里面byte存str时候留下的标识。。之前数据没
31             word = word.lower().replace("b", '').replace('b', '').replace('
32             res.append(wordnet_lemmatizer.lemmatize(word))
33     return res
34
35 corpus = [preprocessing(x) for x in corpus]
36 X_train = [preprocessing(x) for x in X_train]
37 X_test = [preprocessing(x) for x in X_test]
```

```
print(corpus[553])
print(X_train[523])
```

```
['north', 'korean', 'leader', 'kim', 'jong-il', 'confirmed', 'ill']
['two', 'redditors', 'climbing', 'mt', 'kilimanjaro', 'charity', 'bidding', 'peak', 'nt', 's
quander', 'opportunity', 'let', 'upvotes', 'something', 'awesome', 'estimated', 'take', 'yea
r', 'clear', 'lao', 'explosive', 'remnant', 'left', 'behind', 'united', 'state', 'bomber',
'year', 'ago', 'people', 'died', 'unexploded', 'ordnance', 'since', 'conflict', 'ended', 'fi
del', 'ahmadinejad', 'slandering', 'jew', 'mossad', 'america', 'israel', 'intelligence', 'ag
ency', 'target', 'united', 'state', 'intensively', 'among', 'nation', 'considered', 'friendl
y', 'washington', 'israel', 'lead', 'others', 'active', 'espionage', 'directed', 'american',
'company', 'defense', 'department', 'australian', 'election', 'day', 'poll', 'rural/regiona
l', 'independent', 'member', 'parliament', 'support', 'labor', 'minority', 'government', 'jul
ia', 'gillard', 'prime', 'minister', 'france', 'plan', 'raise', 'retirement', 'age', 'set',
'strike', 'britain', 'parliament', 'police', 'murdoch', 'paper', 'adviser', 'pm', 'implicate
d', 'voicemail', 'hacking', 'scandal', 'british', 'policeman', 'jailed', 'month', 'cell', 'a
ttack', 'woman', 'rest', 'email', 'display', 'fundamental', 'disdain', 'pluralistic', 'ameri
ca', 'reveals', 'chilling', 'level', 'islamophobia', 'hatemongering', 'church', 'plan', 'bur
n', 'quran', 'endanger', 'troop', 'u', 'commander', 'warns', 'freed', 'journalist', 'tricke
d', 'captor', 'twitter', 'access', 'manila', 'water', 'crisis', 'expose', 'impact', 'privati
sation', 'july', 'week-long', 'rationing', 'water', 'highlighted', 'reality', 'million', 'pe
ople', 'denied', 'basic', 'right', 'potable', 'water', 'sanitation', 'private', 'firm', 'rak
e', 'profit', 'expense', 'weird', 'uk', 'police', 'ask', 'help', 'case', 'slain', 'intellige
nce', 'agent', 'greenpeace', 'japan', 'anti-whaling', 'activist', 'found', 'guilty', 'thef
t', 'captured', 'journalist', 'trick', 'captor', 'revealing', 'alive', 'creepy', 'biometri
c', 'id', 'forced', 'onto', 'india', 'billion', 'inhabitant', 'fear', 'loss', 'privacy', 'go
vernment', 'abuse', 'abound', 'india', 'gear', 'biometrically', 'identify', 'number', 'billi
on', 'inhabitant', 'china', 'young', 'officer', 'syndrome', 'china', 'military', 'spending',
'growing', 'fast', 'overtaken', 'strategy', 'said', 'professor', 'huang', 'jing', 'school',
'public', 'policy', 'young', 'officer', 'taking', 'control', 'strategy', 'like', 'y
fficer', 'japan', 'mexican', 'soldier', 'open', 'fire', 'family', 'car', 'military', 'checkp
oint', 'killing', 'father', 'son', 'death', 'tell', 'continuous', 'climb', 'guatemala', 'land
```

阿华code


训练NLP模型

- 有了这些干净的数据集，我们可以做我们的NLP模型了。
- 我们先用最简单的Word2Vec

```
1 from gensim.models.word2vec import Word2Vec
2
3 model = Word2Vec(corpus, size=128, window=5, min_count=5, workers=4)
```

```
model['ok']
```

```
0.07165635, -0.15718839, -0.12586397, -0.06831099, 0.2933713 ,
-0.02801018, 0.01784227, -0.05034161, -0.06164618, -0.17971583,
-0.18827736, -0.16022211, 0.02563895, -0.03895595, 0.20914954,
-0.13705987, 0.06534114, -0.15277086, -0.15400618, -0.01750408,
0.0159992 , -0.03115721, -0.21601228, -0.052352 , 0.06168939,
-0.26341256, 0.08789347, -0.13093907, 0.12288627, -0.08176916,
-0.02171701, 0.22793324, 0.04161279, -0.21991451, -0.03320581,
0.0972579 , -0.16723765, -0.08929122, -0.15192902, -0.14742683,
-0.17885518, -0.00583085, -0.02723871, 0.10375598, 0.23577647,
0.14450513, -0.17365797, -0.11279389, -0.09195541, 0.14099288,
-0.06850683, -0.04777306, -0.14641885, -0.19329855, 0.10117287,
0.12809642, 0.06635945, -0.2471927 , 0.02799214, -0.14346135,
0.01861914, 0.12896396, 0.18419097, 0.17958575, 0.25773337,
0.03146051, 0.07480094, 0.04035932, 0.14349777, -0.10399713,
0.06743087, -0.15672234, -0.11129803, -0.03890726, 0.09342128,
-0.1010914 , 0.01256722, -0.05348017, 0.0327471 , -0.08301597,
-0.29558027, -0.07676741, -0.07503819, 0.08925865, -0.05870788,
-0.06687188, -0.3144766 , -0.08962957], dtype=float32)
```

 阿华code

用NLP模型表达我们的X

- 接着，我们于是就可以用这个坐标，来表示我们的之前干干净净的X。
- 但是这儿有个问题。我们的vec是基于每个单词的，怎么办呢？
- 由于我们文本本身的量很小，我们可以把所有的单词的vector拿过来取个平均值：

```
1 # 先拿到全部的vocabulary
2 vocab = model.wv.vocab
3
4 # 得到任意text的vector
5 def get_vector(word_list):
6     # 建立一个全是0的array
7     res = np.zeros([128])
8     count = 0
9     for word in word_list:
10         if word in vocab:
11             res += model[word]
12             count += 1
13     return res/count    #我们得到了一个取得任意word list平均vector值
```

```
1 wordlist_train = X_train
2 wordlist_test = X_test
3
```



```

4 X_train = [get_vector(x) for x in X_train]
5 X_test = [get_vector(x) for x in X_test]
6
7 print(X_train[10])

```

```

d:\zzh\projects\lib\site-packages\ipykernel_launcher.py:11: DeprecationWarning: Call to deprecated `__getitem__` (Method will be removed in 4.0.0, use self.wv.__getitem__() instead)
# This is added back by InteractiveShellApp.init_path()

```

```

[ 2.04398905e-01 -2.39279551e-01 -2.36908053e-02  4.04323679e-02
 -2.32206379e-01 -2.55144463e-01 -1.47943081e-01 -6.35839166e-02
 1.52980143e-01 -2.97927352e-01 -4.05683291e-01  2.60273867e-01
 -2.08099314e-01 -1.03879180e-01 -7.14575568e-01  2.22060183e-01
 6.80624195e-02 -1.97391211e-02  2.38203323e-01 -2.20591236e-01
 3.02061177e-01 -1.45669391e-01  6.84317878e-04 -1.42754950e-01
 1.58006178e-01  3.35989199e-01 -9.12936116e-02 -1.99508447e-01
 -5.05880530e-02 -6.93316005e-03  2.84220558e-01 -1.50358497e-01
 -3.29464662e-02  4.58325946e-01  3.84997232e-01 -2.05936495e-01
 6.34074291e-02  2.72042557e-01  2.81400792e-01  7.24026299e-02
 1.49182007e-01 -2.97690349e-01 -2.02949846e-01 -1.25693787e-01
 5.26847391e-01 -4.55023787e-02  6.71787825e-02  1.36699699e-02
 -3.12844070e-02 -2.06944435e-01 -2.87051732e-01 -2.60641531e-01
 2.50942935e-02 -1.03141542e-01  3.20514179e-01 -1.86750047e-01
 7.90870456e-02 -2.68116313e-01 -2.42132686e-01  1.14152340e-02
 7.96182817e-02 -1.26305730e-02 -2.49016753e-01 -4.08431381e-02
 3.47651430e-02 -4.75066921e-01  1.52279466e-01 -1.56918555e-01
 1.27671771e-01 -1.45346653e-01  1.93864960e-02  3.23801395e-01
 1.87002380e-02 -3.12284687e-01 -5.47049906e-02  2.14605331e-01
 -3.02156223e-01 -1.03697416e-01 -2.49999177e-01 -2.51635487e-01
 -2.91245561e-01 -6.92666244e-03 -5.25437962e-02  1.65662244e-01
 3.59780964e-01  2.17222464e-01 -3.12684860e-01 -1.18454291e-01
 -1.91683454e-01  2.20404427e-01 -6.24920308e-02 -7.32987255e-02
 -2.07163786e-01 -2.82209238e-01  1.81652217e-01  1.70935280e-01
 5.65541722e-02 -3.61222393e-01  7.41518717e-02 -2.40298583e-01
 6.29386274e-02  2.39569850e-01  2.71887002e-01  2.37851132e-01

```

 阿华code

建立ML模型

- 这里，因为我们128维的每一个值都是连续关系的。不是分裂开考虑的。所以，道理上讲，我们是不太适合用RandomForest这类把每个column当做单独的variable来看的方法。

```

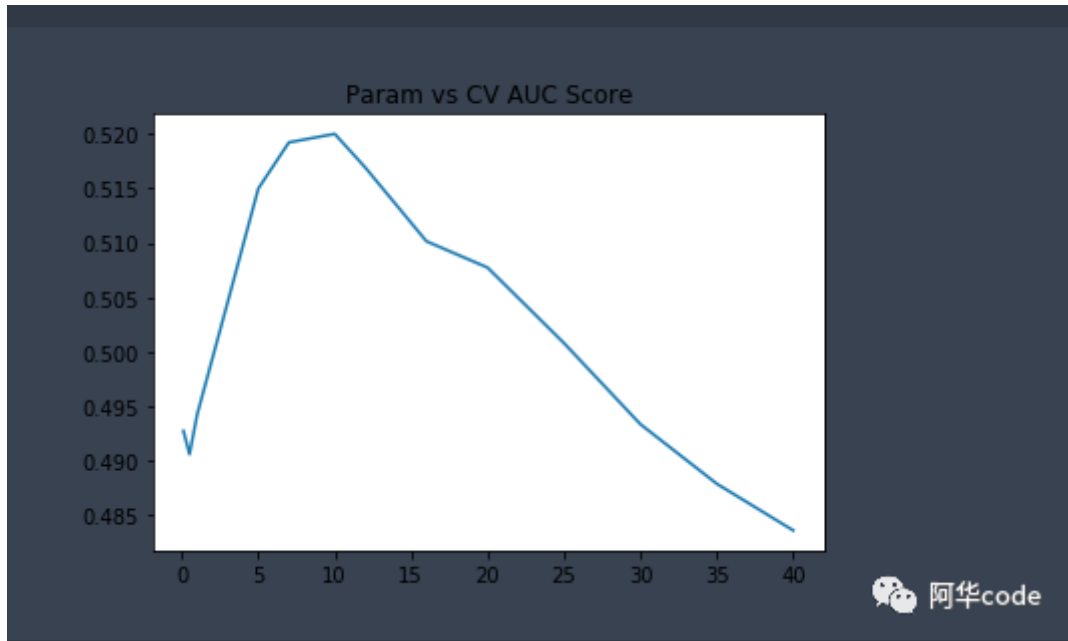
1 from sklearn.svm import SVR
2 from sklearn.model_selection import cross_val_score
3
4 params = [0.1,0.5,1,3,5,7,10,12,16,20,25,30,35,40]
5 test_scores = []
6 for param in params:a
7     clf = SVR(gamma=param)
8     test_score = cross_val_score(clf, X_train, y_train, cv=3, scoring='roc_auc

```



```
9 test_scores.append(np.mean(test_score))
```

```
1 import matplotlib.pyplot as plt
2 %matplotlib inline
3 plt.plot(params, test_scores)
4 plt.title("Param vs CV AUC Score");
```



用CNN来提升逼格

- 用vector表示出一个大matrix，并用CNN做“降维+注意力”

(下面内容我会把整个case搞得简单点。要是想更加复杂准确的话，直接调整参数，往大了调，就行)

- 首先，我们确定一个padding_size。

就是为了让我们的matrix是一样的size啊

```
1 # vec_size 指的是我们本身vector的size
2 def transform_to_matrix(x, padding_size=256, vec_size=128):
3     res = []
4     for sen in x:
5         matrix = []
6         for i in range(padding_size):
7             try:
8                 matrix.append(model[sen[i]].tolist())
9             except:
10                 # 这里有两种except情况，
```

```

11         # 1. 这个单词找不到
12         # 2. sen没那么长
13         # 不管哪种情况，我们直接贴上全是0的vec
14         matrix.append([0] * vec_size)
15     res.append(matrix)
16     return res
17 X_train = transform_to_matrix(wordlist_train)
18 X_test = transform_to_matrix(wordlist_test)
19
20 print(X_train[123])

```

```

d:\zzh\projects\lib\site-packages\ipykernel_launcher.py:8: DeprecationWarning: Call to deprecated `__getitem__` (Method will be removed in 4.0.0, use self.wv.__getitem__() instead).

```

```

[[0.2667890191078186, -0.2733750641345978, -0.075367771089077, 0.21228545904159546, 0.08515
97934961319, -0.41558390855789185, -0.148562490940094, -0.12934811413288116, 0.115442484617
23328, -0.3481332063674927, -0.2865591049194336, 0.1882040798664093, -0.28293004631996155,
-0.173634335398674, -0.7264225482940674, 0.13999620079994202, 0.05704879388213158, -0.01106
1511002480984, 0.38035112619400024, 0.23160718381404877, 0.27546900510787964, -0.4387803077
697754, 0.09069240093231201, -0.2686040699481964, 0.05905426666140556, 0.24655894935131073,
-0.06581586599349976, -0.2597328722476959, 0.21838799118995667, -0.008437634445726871, 0.21
73299491405487, 0.06950933486223221, 0.21102634072303772, 0.7105968594551086, 0.42239063978
19519, -0.22940966486930847, 0.1436776965856552, 0.09965868294239044, 0.04141807183623314,
-0.03567634895443916, 0.27541807293891907, -0.1872165948152542, -0.3939621150493622, -0.331
93883299827576, 0.7038506269454956, -0.07648415863513947, 0.07197355479001999, -0.033705357
46216774, -0.08035212755203247, 0.028168871998786926, -0.09638067334890366, -0.397258639335
6323, 0.31383994221687317, -0.2834497094154358, 0.42806127667427063, -0.3614213168000000,
0.2166228711077005, 0.12663651685820117, 0.11020100552702200, 0.20115768000151216, 0.0550

```

- 可以看到，现在我们得到的就是一个大大的Matrix，它的size是 128 * 256
- 每一个这样的matrix，就是对应了我们每一个数据点
- 在进行下一步之前，我们把我们的input要reshape一下。
- 原因是我们要让每一个matrix外部“包裹”一层维度。来告诉我们的CNN model，我们的每个数据点都是独立的。之间木有前后关系。

```

1 # 搞成np的数组，便于处理
2 X_train = np.array(X_train)
3 X_test = np.array(X_test)
4
5 # 看看数组的大小
6 print(X_train.shape)
7 print(X_test.shape)

```

```

1 X_train = X_train.reshape(X_train.shape[0], 1, X_train.shape[1], X_train.shape[2])
2 X_test = X_test.reshape(X_test.shape[0], 1, X_test.shape[1], X_test.shape[2])

```

```

3
4 print(X_train.shape)
5 print(X_test.shape)

```

```

# 看看数组的大小
print(X_train.shape)
print(X_test.shape)


(1611, 256, 128)
(378, 256, 128)

[109]: X_train = X_train.reshape(X_train.shape[0], 1, X_train.shape[1], X_train.shape[2])
X_test = X_test.reshape(X_test.shape[0], 1, X_test.shape[1], X_test.shape[2])

print(X_train.shape)
print(X_test.shape)

(1611, 1, 256, 128)
(378, 1, 256, 128)

```

 阿华code

```

1 from keras.preprocessing import sequence
2 from keras.models import Sequential
3 from keras.layers import Convolution2D, MaxPooling2D
4 from keras.layers.core import Dense, Dropout, Activation, Flatten
5
6 from keras import backend as K
7 # K.set_image_dim_ordering("th")
8 K.image_data_format() == 'channels_last'
9 # K.set_image_data_format('channels_last')
10
11 # set parameters:
12 batch_size = 32
13 n_filter = 16
14 filter_length = 4
15 nb_epoch = 5
16 n_pool = 2

```

```
17
18 # 新建一个sequential的模型
19 model = Sequential()
20 model.add(Convolution2D(n_filter, filter_length, filter_length,
21                          input_shape=(1, 256, 128)))
22 model.add(Activation('relu'))
23 model.add(Convolution2D(n_filter, filter_length, filter_length))
24 model.add(Activation('relu'))
25 model.add(MaxPooling2D(pool_size=(n_pool, n_pool)))
26 model.add(Dropout(0.25))
27 model.add(Flatten())
28 # 后面接上一个ANN
29 model.add(Dense(128))
30 model.add(Activation('relu'))
31 model.add(Dropout(0.5))
32 model.add(Dense(1))
33 model.add(Activation('softmax'))
34 # compile模型
35 model.compile(loss='mse',
36               optimizer='adadelta',
37               metrics=['accuracy'])

1 model.fit(X_train, y_train, batch_size=batch_size, nb_epoch=nb_epoch,
2           verbose=0)
3 score = model.evaluate(X_test, y_test, verbose=0)
4 print('Test score:', score[0])
5 print('Test accuracy:', score[1])
```

```
print('Test score:', score[0])
print('Test accuracy:', score[1])
```

```
Test score: 0.492063492221
Test accuracy: 0.507936509829
```

 阿华code

PS: TO Sparkling

This is the back of my hand.

This is the back of my foot.