In [1]:
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.datasets import reuters
from tensorflow.keras.layers import Dense, LSTM, Embedding, Activation
import numpy as np
from tensorflow.keras.datasets import imdb
```

In [2]:
```
(X_train, y_train), (X_test, y_test) = imdb.load_data(num_words = None, index_
from=3)
```

In [3]:
```
print(X_train[0])
```

```
[1, 14, 22, 16, 43, 530, 973, 1622, 1385, 65, 458, 4468, 66, 3941, 4, 173, 3
6, 256, 5, 25, 100, 43, 838, 112, 50, 670, 22665, 9, 35, 480, 284, 5, 150, 4,
172, 112, 167, 21631, 336, 385, 39, 4, 172, 4536, 1111, 17, 546, 38, 13, 447,
4, 192, 50, 16, 6, 147, 2025, 19, 14, 22, 4, 1920, 4613, 469, 4, 22, 71, 87,
12, 16, 43, 530, 38, 76, 15, 13, 1247, 4, 22, 17, 515, 17, 12, 16, 626, 18, 1
9193, 5, 62, 386, 12, 8, 316, 8, 106, 5, 4, 2223, 5244, 16, 480, 66, 3785, 3
3, 4, 130, 12, 16, 38, 619, 5, 25, 124, 51, 36, 135, 48, 25, 1415, 33, 6, 22,
12, 215, 28, 77, 52, 5, 14, 407, 16, 82, 10311, 8, 4, 107, 117, 5952, 15, 25
6, 4, 31050, 7, 3766, 5, 723, 36, 71, 43, 530, 476, 26, 400, 317, 46, 7, 4, 1
2118, 1029, 13, 104, 88, 4, 381, 15, 297, 98, 32, 2071, 56, 26, 141, 6, 194,
7486, 18, 4, 226, 22, 21, 134, 476, 26, 480, 5, 144, 30, 5535, 18, 51, 36, 2
8, 224, 92, 25, 104, 4, 226, 65, 16, 38, 1334, 88, 12, 16, 283, 5, 16, 4472,
113, 103, 32, 15, 16, 5345, 19, 178, 32]
```

In [4]:
```
word_index = imdb.get_word_index() # key_word,value_index dic
reverse_index = dict([(value, key) for (key, value) in word_index.items()]) #
 key_index,value_word dic
decoded = " ".join( [reverse_index.get(i-3, "#") for i in X_train[0]] )
print(decoded)
```

```
# this film was just brilliant casting location scenery story direction every
one's really suited the part they played and you could just imagine being the
re robert redford's is an amazing actor and now the same being director norma
n's father came from the same scottish island as myself so i loved the fact t
here was a real connection with this film the witty remarks throughout the fi
lm were great it was just brilliant so much that i bought the film as soon as
it was released for retail and would recommend it to everyone to watch and th
e fly fishing was amazing really cried at the end it was so sad and you know
what they say if you cry at a film it must have been good and this definitely
was also congratulations to the two little boy's that played the part's of no
rman and paul they were just brilliant children are often left out of the pra
ising list i think because the stars that play them all grown up are such a b
ig profile for the whole film but these children are amazing and should be pr
aised for what they have done don't you think the whole story was so lovely b
ecause it was true and was someone's life after all that was shared with us a
ll
```

In [5]:
```
print(len(X_train[0]))
```

```
218
```

In [6]:
```
np.unique(y_train)
```
Out[6]:
```
array([0, 1], dtype=int64)
```

In [7]:
```
y_train[0]
```
Out[7]:
```
1
```

In [8]:
```
num_classes = max(y_train) + 1
print(num_classes)
```
```
2
```

In [9]:
```
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.utils import to_categorical
```

In [10]:
```
max_num = 10000
tokenizer = Tokenizer(num_words=max_num)
tokenizer.fit_on_sequences(X_train)
X_train = tokenizer.sequences_to_matrix(X_train, mode = 'tfidf')
X_test = tokenizer.sequences_to_matrix(X_test, mode = 'tfidf')
```

In [11]:
```
print(X_train[0][4])
```
```
2.5855967311039936
```

In [12]:
```
y_train = to_categorical(y_train, num_classes)
y_test = to_categorical(y_test, num_classes)
y_train.shape
```
Out[12]:
```
(25000, 2)
```

In [13]:
```
from sklearn.model_selection import KFold
kf = KFold(n_splits = 3,shuffle = True)
kf
```
Out[13]:
```
KFold(n_splits=3, random_state=None, shuffle=True)
```

find out the overfitting model

In [15]:
```
units = range(100,300,10)
loss_train = []
acc_train = []
loss_val = []
acc_val = []
for num in units:
    model = Sequential()
    model.add(Dense(num,input_shape = (max_num,)))
    model.add(Activation('relu'))
    model.add(Dense(num_classes))
    model.add(Activation('softmax'))
    model.compile(loss = 'categorical_crossentropy', optimizer = 'adam', metri
cs = ['accuracy'])
    loss_val_cv = 0
    acc_val_cv = 0
    loss_train_cv = 0
    acc_train_cv = 0
    for train_cv_index, val_index in kf.split(X_train):
        X_train_cv = X_train[train_cv_index]
        y_train_cv = y_train[train_cv_index]
        X_val_cv = X_train[val_index]
        y_val_cv = y_train[val_index]
        hist = model.fit(X_train_cv,y_train_cv,batch_size=200, epochs = 3)
        loss_train_cv = loss_train_cv + hist.history.get('loss')[-1]
        acc_train_cv = acc_train_cv + hist.history.get('acc')[-1]
        score_val_cv = model.evaluate(X_val_cv,y_val_cv, batch_size=200, verbo
se = 1)
        loss_val_cv = loss_val_cv + score_val_cv[0]
        acc_val_cv = acc_val_cv + score_val_cv[1]
    loss_val.append(loss_val_cv/3)
    acc_val.append(acc_val_cv/3)
    loss_train.append(loss_train_cv/3)
    acc_train.append(acc_train_cv/3)
```

```
Epoch 1/3
16666/16666 [==============================] - 6s 348us/step - loss: 0.3507 -
acc: 0.8497
Epoch 2/3
16666/16666 [==============================] - 5s 313us/step - loss: 0.0815 -
acc: 0.9764
Epoch 3/3
16666/16666 [==============================] - 5s 311us/step - loss: 0.0231 -
acc: 0.9962
8334/8334 [==============================] - 2s 197us/step
Epoch 1/3
16667/16667 [==============================] - 5s 312us/step - loss: 0.1807 -
acc: 0.9381
Epoch 2/3
16667/16667 [==============================] - 5s 306us/step - loss: 0.0318 -
acc: 0.9924
Epoch 3/3
16667/16667 [==============================] - 5s 303us/step - loss: 0.0068 -
acc: 0.9993
8333/8333 [==============================] - 1s 167us/step
Epoch 1/3
16667/16667 [==============================] - 5s 305us/step - loss: 0.0291 -
acc: 0.9914
Epoch 2/3
16667/16667 [==============================] - 5s 310us/step - loss: 0.0068 -
acc: 0.9994
Epoch 3/3
16667/16667 [==============================] - 5s 311us/step - loss: 0.0018 -
acc: 1.0000
8333/8333 [==============================] - 2s 191us/step
Epoch 1/3
16666/16666 [==============================] - 6s 375us/step - loss: 0.3582 -
acc: 0.8471
Epoch 2/3
16666/16666 [==============================] - 5s 327us/step - loss: 0.0808 -
acc: 0.9758
Epoch 3/3
16666/16666 [==============================] - 5s 320us/step - loss: 0.0230 -
acc: 0.9966 1s - loss: 0.0
8334/8334 [==============================] - 2s 185us/step
Epoch 1/3
16667/16667 [==============================] - 5s 322us/step - loss: 0.1744 -
acc: 0.9376
Epoch 2/3
16667/16667 [==============================] - 5s 326us/step - loss: 0.0292 -
acc: 0.9941
Epoch 3/3
16667/16667 [==============================] - 5s 326us/step - loss: 0.0081 -
acc: 0.9993
8333/8333 [==============================] - 2s 195us/step
Epoch 1/3
16667/16667 [==============================] - 5s 328us/step - loss: 0.0243 -
acc: 0.9942
Epoch 2/3
16667/16667 [==============================] - 5s 324us/step - loss: 0.0066 -
acc: 0.9995
Epoch 3/3
```

```
16667/16667 [==============================] - 5s 323us/step - loss: 0.0035 -
acc: 0.9999
8333/8333 [==============================] - 1s 171us/step
Epoch 1/3
16666/16666 [==============================] - 6s 376us/step - loss: 0.3464 -
acc: 0.8511
Epoch 2/3
16666/16666 [==============================] - 6s 335us/step - loss: 0.0717 -
acc: 0.9793
Epoch 3/3
16666/16666 [==============================] - 6s 337us/step - loss: 0.0189 -
acc: 0.9968
8334/8334 [==============================] - 2s 231us/step
Epoch 1/3
16667/16667 [==============================] - 6s 331us/step - loss: 0.1824 -
acc: 0.9396
Epoch 2/3
16667/16667 [==============================] - 6s 337us/step - loss: 0.0323 -
acc: 0.9924
Epoch 3/3
16667/16667 [==============================] - 6s 334us/step - loss: 0.0071 -
acc: 0.9994
8333/8333 [==============================] - 2s 215us/step
Epoch 1/3
16667/16667 [==============================] - 6s 337us/step - loss: 0.0240 -
acc: 0.9935
Epoch 2/3
16667/16667 [==============================] - 6s 337us/step - loss: 0.0043 -
acc: 0.9999
Epoch 3/3
16667/16667 [==============================] - 5s 330us/step - loss: 0.0015 -
acc: 1.0000
8333/8333 [==============================] - 1s 171us/step
Epoch 1/3
16666/16666 [==============================] - 7s 401us/step - loss: 0.3568 -
acc: 0.8478
Epoch 2/3
16666/16666 [==============================] - 6s 346us/step - loss: 0.0784 -
acc: 0.9767
Epoch 3/3
16666/16666 [==============================] - 6s 346us/step - loss: 0.0211 -
acc: 0.9965
8334/8334 [==============================] - 2s 207us/step
Epoch 1/3
16667/16667 [==============================] - 6s 350us/step - loss: 0.1641 -
acc: 0.9431
Epoch 2/3
16667/16667 [==============================] - 6s 346us/step - loss: 0.0249 -
acc: 0.9950
Epoch 3/3
16667/16667 [==============================] - 6s 346us/step - loss: 0.0066 -
acc: 0.9997
8333/8333 [==============================] - 2s 194us/step
Epoch 1/3
16667/16667 [==============================] - 6s 351us/step - loss: 0.0221 -
acc: 0.9935
Epoch 2/3
```

```
16667/16667 [==============================] - 6s 348us/step - loss: 0.0041 -
acc: 0.9999
Epoch 3/3
16667/16667 [==============================] - 6s 335us/step - loss: 0.0013 -
acc: 1.0000
8333/8333 [==============================] - 2s 180us/step
Epoch 1/3
16666/16666 [==============================] - 7s 410us/step - loss: 0.3449 -
acc: 0.8558
Epoch 2/3
16666/16666 [==============================] - 6s 367us/step - loss: 0.0700 -
acc: 0.9797
Epoch 3/3
16666/16666 [==============================] - 6s 361us/step - loss: 0.0191 -
acc: 0.9971
8334/8334 [==============================] - 2s 189us/step
Epoch 1/3
16667/16667 [==============================] - 6s 363us/step - loss: 0.1831 -
acc: 0.9359
Epoch 2/3
16667/16667 [==============================] - 6s 362us/step - loss: 0.0282 -
acc: 0.9932
Epoch 3/3
16667/16667 [==============================] - 6s 360us/step - loss: 0.0061 -
acc: 0.9993
8333/8333 [==============================] - 1s 172us/step
Epoch 1/3
16667/16667 [==============================] - 6s 365us/step - loss: 0.0232 -
acc: 0.9926
Epoch 2/3
16667/16667 [==============================] - 6s 365us/step - loss: 0.0044 -
acc: 0.9999
Epoch 3/3
16667/16667 [==============================] - 6s 365us/step - loss: 0.0013 -
acc: 1.0000
8333/8333 [==============================] - 1s 178us/step
Epoch 1/3
16666/16666 [==============================] - 7s 431us/step - loss: 0.3515 -
acc: 0.8508
Epoch 2/3
16666/16666 [==============================] - 6s 374us/step - loss: 0.0702 -
acc: 0.9798
Epoch 3/3
16666/16666 [==============================] - 6s 376us/step - loss: 0.0171 -
acc: 0.9976
8334/8334 [==============================] - 2s 249us/step
Epoch 1/3
16667/16667 [==============================] - 6s 378us/step - loss: 0.1710 -
acc: 0.9398
Epoch 2/3
16667/16667 [==============================] - 6s 372us/step - loss: 0.0269 -
acc: 0.9941
Epoch 3/3
16667/16667 [==============================] - 6s 376us/step - loss: 0.0053 -
acc: 0.9996
8333/8333 [==============================] - 2s 187us/step
Epoch 1/3
```

```
16667/16667 [==============================] - 6s 384us/step - loss: 0.0214 -
acc: 0.9936
Epoch 2/3
16667/16667 [==============================] - 7s 391us/step - loss: 0.0044 -
acc: 0.9998
Epoch 3/3
16667/16667 [==============================] - 6s 379us/step - loss: 0.0013 -
acc: 1.0000
8333/8333 [==============================] - 1s 179us/step
Epoch 1/3
16666/16666 [==============================] - 8s 450us/step - loss: 0.3481 -
acc: 0.8507
Epoch 2/3
16666/16666 [==============================] - 7s 393us/step - loss: 0.0658 -
acc: 0.9822
Epoch 3/3
16666/16666 [==============================] - 6s 389us/step - loss: 0.0162 -
acc: 0.9979
8334/8334 [==============================] - 2s 237us/step
Epoch 1/3
16667/16667 [==============================] - 7s 394us/step - loss: 0.1803 -
acc: 0.9387
Epoch 2/3
16667/16667 [==============================] - 7s 391us/step - loss: 0.0298 -
acc: 0.9933
Epoch 3/3
16667/16667 [==============================] - 6s 388us/step - loss: 0.0070 -
acc: 0.9995
8333/8333 [==============================] - 2s 180us/step
Epoch 1/3
16667/16667 [==============================] - 7s 412us/step - loss: 0.0217 -
acc: 0.9947
Epoch 2/3
16667/16667 [==============================] - 6s 386us/step - loss: 0.0049 -
acc: 0.9999
Epoch 3/3
16667/16667 [==============================] - 6s 380us/step - loss: 0.0023 -
acc: 0.9999
8333/8333 [==============================] - 2s 230us/step
Epoch 1/3
16666/16666 [==============================] - 8s 461us/step - loss: 0.3559 -
acc: 0.8495
Epoch 2/3
16666/16666 [==============================] - 7s 405us/step - loss: 0.0674 -
acc: 0.9803
Epoch 3/3
16666/16666 [==============================] - 7s 399us/step - loss: 0.0162 -
acc: 0.9977
8334/8334 [==============================] - 2s 239us/step
Epoch 1/3
16667/16667 [==============================] - 7s 400us/step - loss: 0.1739 -
acc: 0.9401
Epoch 2/3
16667/16667 [==============================] - 7s 405us/step - loss: 0.0252 -
acc: 0.9944
Epoch 3/3
16667/16667 [==============================] - 7s 401us/step - loss: 0.0054 -
```

```
acc: 0.9998
8333/8333 [==============================] - 1s 180us/step
Epoch 1/3
16667/16667 [==============================] - 7s 407us/step - loss: 0.0233 -
acc: 0.9939
Epoch 2/3
16667/16667 [==============================] - 7s 405us/step - loss: 0.0055 -
acc: 0.9998
Epoch 3/3
16667/16667 [==============================] - 7s 401us/step - loss: 0.0013 -
acc: 1.0000
8333/8333 [==============================] - 1s 177us/step
Epoch 1/3
16666/16666 [==============================] - 8s 492us/step - loss: 0.3523 -
acc: 0.8539 0s - loss: 0.3579 - acc:
Epoch 2/3
16666/16666 [==============================] - 7s 415us/step - loss: 0.0618 -
acc: 0.9835
Epoch 3/3
16666/16666 [==============================] - 7s 413us/step - loss: 0.0140 -
acc: 0.9983
8334/8334 [==============================] - 2s 244us/step
Epoch 1/3
16667/16667 [==============================] - 7s 420us/step - loss: 0.1693 -
acc: 0.9417
Epoch 2/3
16667/16667 [==============================] - 7s 421us/step - loss: 0.0241 -
acc: 0.9951
Epoch 3/3
16667/16667 [==============================] - 7s 419us/step - loss: 0.0050 -
acc: 0.9995
8333/8333 [==============================] - 2s 199us/step
Epoch 1/3
16667/16667 [==============================] - 7s 420us/step - loss: 0.0183 -
acc: 0.9954
Epoch 2/3
16667/16667 [==============================] - 7s 420us/step - loss: 0.0040 -
acc: 0.9999
Epoch 3/3
16667/16667 [==============================] - 7s 424us/step - loss: 0.0012 -
acc: 0.9999
8333/8333 [==============================] - 2s 185us/step
Epoch 1/3
16666/16666 [==============================] - 8s 500us/step - loss: 0.3536 -
acc: 0.8477
Epoch 2/3
16666/16666 [==============================] - 7s 436us/step - loss: 0.0664 -
acc: 0.9811
Epoch 3/3
16666/16666 [==============================] - 7s 434us/step - loss: 0.0167 -
acc: 0.9975
8334/8334 [==============================] - 2s 249us/step
Epoch 1/3
16667/16667 [==============================] - 7s 434us/step - loss: 0.1722 -
acc: 0.9413
Epoch 2/3
16667/16667 [==============================] - 7s 435us/step - loss: 0.0246 -
```

```
acc: 0.9943
Epoch 3/3
16667/16667 [==============================] - 7s 436us/step - loss: 0.0060 -
acc: 0.9994
8333/8333 [==============================] - 2s 188us/step
Epoch 1/3
16667/16667 [==============================] - 7s 435us/step - loss: 0.0163 -
acc: 0.9962 2s - l
Epoch 2/3
16667/16667 [==============================] - 7s 431us/step - loss: 0.0030 -
acc: 0.9998
Epoch 3/3
16667/16667 [==============================] - 7s 432us/step - loss: 0.0012 -
acc: 0.9999
8333/8333 [==============================] - 2s 193us/step
Epoch 1/3
16666/16666 [==============================] - 9s 515us/step - loss: 0.3518 -
acc: 0.8531
Epoch 2/3
16666/16666 [==============================] - 7s 443us/step - loss: 0.0624 -
acc: 0.9822
Epoch 3/3
16666/16666 [==============================] - 7s 444us/step - loss: 0.0149 -
acc: 0.9977
8334/8334 [==============================] - 2s 266us/step
Epoch 1/3
16667/16667 [==============================] - 7s 438us/step - loss: 0.1750 -
acc: 0.9394
Epoch 2/3
16667/16667 [==============================] - 7s 443us/step - loss: 0.0226 -
acc: 0.9957
Epoch 3/3
16667/16667 [==============================] - 7s 443us/step - loss: 0.0046 -
acc: 0.9996
8333/8333 [==============================] - 2s 196us/step
Epoch 1/3
16667/16667 [==============================] - 8s 450us/step - loss: 0.0198 -
acc: 0.9945
Epoch 2/3
16667/16667 [==============================] - 8s 452us/step - loss: 0.0033 -
acc: 0.9999 0s - loss: 0.0034 -
Epoch 3/3
16667/16667 [==============================] - 7s 443us/step - loss: 9.6005e-
04 - acc: 1.0000
8333/8333 [==============================] - 2s 202us/step
Epoch 1/3
16666/16666 [==============================] - 9s 546us/step - loss: 0.3499 -
acc: 0.8516
Epoch 2/3
16666/16666 [==============================] - 8s 455us/step - loss: 0.0636 -
acc: 0.9821
Epoch 3/3
16666/16666 [==============================] - 8s 457us/step - loss: 0.0140 -
acc: 0.9984
8334/8334 [==============================] - 2s 231us/step
Epoch 1/3
16667/16667 [==============================] - 8s 455us/step - loss: 0.1757 -
```

```
acc: 0.9401 4s -  - ETA: 2s
Epoch 2/3
16667/16667 [==============================] - 8s 459us/step - loss: 0.0237 -
acc: 0.9953 1s - loss: 0.0247 - a - ETA: 0s - loss: 0.0247 - acc
Epoch 3/3
16667/16667 [==============================] - 8s 458us/step - loss: 0.0051 -
acc: 0.9996
8333/8333 [==============================] - 2s 193us/step
Epoch 1/3
16667/16667 [==============================] - 8s 461us/step - loss: 0.0194 -
acc: 0.9944 0s - loss: 0.0197 - acc:
Epoch 2/3
16667/16667 [==============================] - 8s 458us/step - loss: 0.0047 -
acc: 0.9998
Epoch 3/3
16667/16667 [==============================] - 8s 452us/step - loss: 0.0020 -
acc: 0.9999
8333/8333 [==============================] - 2s 200us/step
Epoch 1/3
16666/16666 [==============================] - 9s 557us/step - loss: 0.3528 -
acc: 0.8481
Epoch 2/3
16666/16666 [==============================] - 8s 472us/step - loss: 0.0625 -
acc: 0.9827
Epoch 3/3
16666/16666 [==============================] - 8s 478us/step - loss: 0.0147 -
acc: 0.9981
8334/8334 [==============================] - 2s 239us/step
Epoch 1/3
16667/16667 [==============================] - 8s 471us/step - loss: 0.1722 -
acc: 0.9393
Epoch 2/3
16667/16667 [==============================] - 8s 471us/step - loss: 0.0225 -
acc: 0.9951
Epoch 3/3
16667/16667 [==============================] - 8s 472us/step - loss: 0.0043 -
acc: 0.9997
8333/8333 [==============================] - 2s 224us/step
Epoch 1/3
16667/16667 [==============================] - 8s 469us/step - loss: 0.0200 -
acc: 0.9943
Epoch 2/3
16667/16667 [==============================] - 8s 476us/step - loss: 0.0029 -
acc: 1.0000
Epoch 3/3
16667/16667 [==============================] - 8s 479us/step - loss: 9.7745e-
04 - acc: 1.0000
8333/8333 [==============================] - 2s 192us/step
Epoch 1/3
16666/16666 [==============================] - 9s 567us/step - loss: 0.3629 -
acc: 0.8415
Epoch 2/3
16666/16666 [==============================] - 8s 490us/step - loss: 0.0611 -
acc: 0.9840
Epoch 3/3
16666/16666 [==============================] - 8s 489us/step - loss: 0.0144 -
acc: 0.9983
```

```
8334/8334 [==============================] - 2s 239us/step
Epoch 1/3
16667/16667 [==============================] - 8s 489us/step - loss: 0.1723 -
acc: 0.9405
Epoch 2/3
16667/16667 [==============================] - 8s 497us/step - loss: 0.0212 -
acc: 0.9954
Epoch 3/3
16667/16667 [==============================] - 8s 486us/step - loss: 0.0044 -
acc: 0.9999
8333/8333 [==============================] - 2s 202us/step
Epoch 1/3
16667/16667 [==============================] - 8s 487us/step - loss: 0.0198 -
acc: 0.9952
Epoch 2/3
16667/16667 [==============================] - 8s 492us/step - loss: 0.0038 -
acc: 0.9998
Epoch 3/3
16667/16667 [==============================] - 8s 485us/step - loss: 0.0012 -
acc: 0.9999
8333/8333 [==============================] - 2s 200us/step
Epoch 1/3
16666/16666 [==============================] - 10s 581us/step - loss: 0.3549
- acc: 0.8484
Epoch 2/3
16666/16666 [==============================] - 8s 502us/step - loss: 0.0604 -
acc: 0.9838
Epoch 3/3
16666/16666 [==============================] - 8s 494us/step - loss: 0.0137 -
acc: 0.9983
8334/8334 [==============================] - 2s 260us/step
Epoch 1/3
16667/16667 [==============================] - 8s 502us/step - loss: 0.1755 -
acc: 0.9399
Epoch 2/3
16667/16667 [==============================] - 8s 499us/step - loss: 0.0243 -
acc: 0.9953
Epoch 3/3
16667/16667 [==============================] - 8s 501us/step - loss: 0.0053 -
acc: 0.9995
8333/8333 [==============================] - 2s 201us/step
Epoch 1/3
16667/16667 [==============================] - 8s 498us/step - loss: 0.0192 -
acc: 0.9953
Epoch 2/3
16667/16667 [==============================] - 8s 505us/step - loss: 0.0044 -
acc: 0.9999
Epoch 3/3
16667/16667 [==============================] - 8s 505us/step - loss: 0.0020 -
acc: 0.9999
8333/8333 [==============================] - 2s 210us/step
Epoch 1/3
16666/16666 [==============================] - 10s 600us/step - loss: 0.3489
- acc: 0.8510
Epoch 2/3
16666/16666 [==============================] - 9s 513us/step - loss: 0.0567 -
acc: 0.9848
```

```
Epoch 3/3
16666/16666 [==============================] - 9s 511us/step - loss: 0.0125 -
acc: 0.9984
8334/8334 [==============================] - 2s 257us/step
Epoch 1/3
16667/16667 [==============================] - 9s 512us/step - loss: 0.1816 -
acc: 0.9365
Epoch 2/3
16667/16667 [==============================] - 9s 511us/step - loss: 0.0247 -
acc: 0.9956
Epoch 3/3
16667/16667 [==============================] - 9s 530us/step - loss: 0.0054 -
acc: 0.9998
8333/8333 [==============================] - 2s 213us/step
Epoch 1/3
16667/16667 [==============================] - 9s 521us/step - loss: 0.0194 -
acc: 0.9952
Epoch 2/3
16667/16667 [==============================] - 9s 523us/step - loss: 0.0038 -
acc: 0.9999
Epoch 3/3
16667/16667 [==============================] - 9s 526us/step - loss: 0.0019 -
acc: 0.9999
8333/8333 [==============================] - 2s 198us/step
Epoch 1/3
16666/16666 [==============================] - ETA: 0s - loss: 0.3548 - acc:
0.848 - 11s 636us/step - loss: 0.3543 - acc: 0.8486
Epoch 2/3
16666/16666 [==============================] - 9s 545us/step - loss: 0.0561 -
acc: 0.9840
Epoch 3/3
16666/16666 [==============================] - 9s 561us/step - loss: 0.0121 -
acc: 0.9987
8334/8334 [==============================] - 2s 260us/step
Epoch 1/3
16667/16667 [==============================] - 9s 546us/step - loss: 0.1709 -
acc: 0.9416
Epoch 2/3
16667/16667 [==============================] - 9s 548us/step - loss: 0.0223 -
acc: 0.9954
Epoch 3/3
16667/16667 [==============================] - 9s 544us/step - loss: 0.0049 -
acc: 0.9997
8333/8333 [==============================] - 2s 246us/step
Epoch 1/3
16667/16667 [==============================] - 9s 564us/step - loss: 0.0199 -
acc: 0.9946
Epoch 2/3
16667/16667 [==============================] - 9s 546us/step - loss: 0.0046 -
acc: 0.9998
Epoch 3/3
16667/16667 [==============================] - 9s 561us/step - loss: 0.0019 -
acc: 0.9999
8333/8333 [==============================] - 2s 212us/step
Epoch 1/3
16666/16666 [==============================] - 11s 655us/step - loss: 0.3657
- acc: 0.8369
```

```
Epoch 2/3
16666/16666 [==============================] - 9s 558us/step - loss: 0.0616 -
acc: 0.9836
Epoch 3/3
16666/16666 [==============================] - 9s 560us/step - loss: 0.0145 -
acc: 0.9985
8334/8334 [==============================] - 2s 286us/step
Epoch 1/3
16667/16667 [==============================] - 9s 555us/step - loss: 0.1766 -
acc: 0.9380
Epoch 2/3
16667/16667 [==============================] - 9s 556us/step - loss: 0.0222 -
acc: 0.9956
Epoch 3/3
16667/16667 [==============================] - 9s 551us/step - loss: 0.0044 -
acc: 0.9998
8333/8333 [==============================] - 2s 209us/step
Epoch 1/3
16667/16667 [==============================] - 9s 552us/step - loss: 0.0182 -
acc: 0.9949
Epoch 2/3
16667/16667 [==============================] - 9s 552us/step - loss: 0.0040 -
acc: 0.9999
Epoch 3/3
16667/16667 [==============================] - 9s 552us/step - loss: 0.0010 -
acc: 1.0000
8333/8333 [==============================] - 2s 224us/step
Epoch 1/3
16666/16666 [==============================] - 11s 665us/step - loss: 0.3492
- acc: 0.8510
Epoch 2/3
16666/16666 [==============================] - 10s 570us/step - loss: 0.0551
- acc: 0.9851
Epoch 3/3
16666/16666 [==============================] - 10s 571us/step - loss: 0.0112
- acc: 0.9987
8334/8334 [==============================] - 2s 266us/step
Epoch 1/3
16667/16667 [==============================] - 9s 561us/step - loss: 0.1771 -
acc: 0.9399
Epoch 2/3
16667/16667 [==============================] - 10s 571us/step - loss: 0.0243
- acc: 0.9953
Epoch 3/3
16667/16667 [==============================] - 10s 575us/step - loss: 0.0049
- acc: 0.9996
8333/8333 [==============================] - 2s 215us/step
Epoch 1/3
16667/16667 [==============================] - 9s 564us/step - loss: 0.0173 -
acc: 0.9956
Epoch 2/3
16667/16667 [==============================] - 9s 567us/step - loss: 0.0024 -
acc: 1.0000
Epoch 3/3
16667/16667 [==============================] - 9s 562us/step - loss: 8.4070e-
04 - acc: 1.0000
8333/8333 [==============================] - 2s 212us/step
```

```
Epoch 1/3
16666/16666 [==============================] - 11s 686us/step - loss: 0.3447
- acc: 0.8502
Epoch 2/3
16666/16666 [==============================] - 10s 582us/step - loss: 0.0500
- acc: 0.9859
Epoch 3/3
16666/16666 [==============================] - 10s 583us/step - loss: 0.0109
- acc: 0.9989
8334/8334 [==============================] - 2s 270us/step
Epoch 1/3
16667/16667 [==============================] - 10s 577us/step - loss: 0.1847
- acc: 0.9339
Epoch 2/3
16667/16667 [==============================] - 10s 582us/step - loss: 0.0222
- acc: 0.9953
Epoch 3/3
16667/16667 [==============================] - 10s 578us/step - loss: 0.0040
- acc: 0.9998
8333/8333 [==============================] - 2s 219us/step
Epoch 1/3
16667/16667 [==============================] - 10s 574us/step - loss: 0.0183
- acc: 0.9957
Epoch 2/3
16667/16667 [==============================] - 9s 569us/step - loss: 0.0023 -
acc: 1.0000
Epoch 3/3
16667/16667 [==============================] - 10s 577us/step - loss: 8.6907e
-04 - acc: 1.0000
8333/8333 [==============================] - 2s 217us/step
```

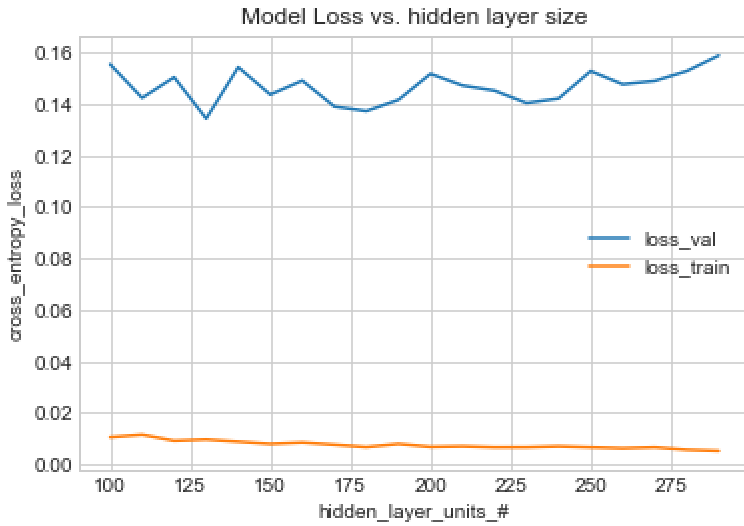
In [14]:

```
%matplotlib inline
import matplotlib.pyplot as plt
plt.style.use('seaborn-whitegrid')
import pandas as pd
```

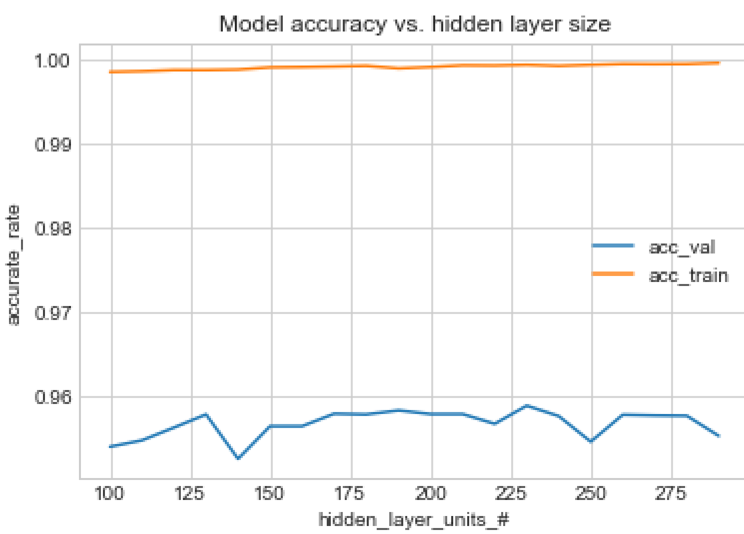
In [17]:

```
ix = units
loss_train= pd.Series(loss_train, index = ix)
loss_val = pd.Series(loss_val, index = ix)
acc_train = pd.Series(acc_train, index = ix)
acc_val = pd.Series(acc_val, index = ix)
```

In [27]:
```
fig = plt.figure()
ax = plt.axes()
ax.plot(loss_val, label = 'loss_val')
ax.plot(loss_train, label = 'loss_train')
ax.legend()
plt.xlabel('hidden_layer_units_#')
plt.ylabel('cross_entropy_loss')
plt.title('Model Loss vs. hidden layer size')
fig.savefig("loss_cv.png",dpi = 400)
```



In [28]:
```
fig = plt.figure()
ax = plt.axes()
ax.plot(acc_val,label = 'acc_val')
ax.plot(acc_train, label = 'acc_train')
ax.legend()
plt.xlabel('hidden_layer_units_#')
plt.ylabel('accurate_rate')
plt.title('Model accuracy vs. hidden layer size')
fig.savefig("acc_cv.png",dpi=400)
```



In [ ]:
```
# overfitting, while hidden layer contains 250 nodes in this case
# first kind of regularization: dropout
# second kind of regularization: l1_norm_kernel_regularizer
# third kind of regularization: l2_norm_kernel_regularizer
```

In [48]:
```
# the baseline to compare with
model_overfitting = Sequential()
model_overfitting.add(Dense(250,input_shape = (max_num,)))
model_overfitting.add(Activation('relu'))
model_overfitting.add(Dense(num_classes))
model_overfitting.add(Activation('softmax'))
model_overfitting.compile(loss = 'categorical_crossentropy', optimizer = 'ada
m', metrics = ['accuracy'])

hist_overfitting = model_overfitting.fit(X_train,y_train, batch_size=200, epoc
hs = 3, verbose = 1)
score_overfitting = model_overfitting.evaluate(X_test,y_test, batch_size=200,
verbose = 1)

base_train_acc = hist_overfitting.history.get('acc')[-1]
base_test_acc = score_overfitting[1]
```

```
Epoch 1/3
25000/25000 [==============================] - 13s 536us/step - loss: 0.3210
- acc: 0.8646
Epoch 2/3
25000/25000 [==============================] - 11s 435us/step - loss: 0.0665
- acc: 0.9803
Epoch 3/3
25000/25000 [==============================] - 10s 417us/step - loss: 0.0141
- acc: 0.9978
25000/25000 [==============================] - 6s 229us/step
```

In [49]:
```
print('The accurate rate of the overfitting model on training dataset is')
print(base_train_acc)
print('The accurate rate of the overfitting model on test dataset is')
print(base_test_acc)
```

```
The accurate rate of the overfitting model on training dataset is
0.9978400020599365
The accurate rate of the overfitting model on test dataset is
0.8704400014877319
```

In [29]:
```
# Save the weights
model_overfitting.save_weights('model_base_weights.h5')

# Save the model architecture
with open('model_base_architecture.json', 'w') as f:
    f.write(model_overfitting.to_json())
```

In [ ]:
```
from keras.models import model_from_json

# Model reconstruction from JSON file
with open('model_base_architecture.json', 'r') as f:
     model_overfitting = model_from_json(f.read())

# Load weights into the new model
model_overfitting.load_weights('model_base_weights.h5')
```

In [ ]:

In [44]:
```
from tensorflow.keras.layers import Dropout
rate = [0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9]
loss_train_drop = []
acc_train_drop = []
loss_val_drop = []
acc_val_drop = []
for r in rate:
    model_drop = Sequential()
    model_drop.add(Dense(250,input_shape = (max_num,)))
    model_drop.add(Activation('relu'))
    model_drop.add(Dropout(r))
    model_drop.add(Dense(num_classes))
    model_drop.add(Activation('softmax'))
    model_drop.compile(loss = 'categorical_crossentropy', optimizer = 'adam',
metrics = ['accuracy'])
    loss_val_cv = 0
    acc_val_cv = 0
    loss_train_cv = 0
    acc_train_cv = 0
    for train_cv_index, val_index in kf.split(X_train):
        X_train_cv = X_train[train_cv_index]
        y_train_cv = y_train[train_cv_index]
        X_val_cv = X_train[val_index]
        y_val_cv = y_train[val_index]
        hist = model_drop.fit(X_train_cv,y_train_cv,batch_size=200, epochs = 3
)
        loss_train_cv = loss_train_cv + hist.history.get('loss')[-1]
        acc_train_cv = acc_train_cv + hist.history.get('acc')[-1]
        score_val_cv = model_drop.evaluate(X_val_cv,y_val_cv, batch_size=200,
verbose = 1)
        loss_val_cv = loss_val_cv + score_val_cv[0]
        acc_val_cv = acc_val_cv + score_val_cv[1]
    loss_val_drop.append(loss_val_cv/3)
    acc_val_drop.append(acc_val_cv/3)
    loss_train_drop.append(loss_train_cv/3)
    acc_train_drop.append(acc_train_cv/3)
```

```
Epoch 1/3
16666/16666 [==============================] - 10s 575us/step - loss: 0.3509
- acc: 0.8487
Epoch 2/3
16666/16666 [==============================] - 7s 439us/step - loss: 0.0665 -
acc: 0.9796
Epoch 3/3
16666/16666 [==============================] - 7s 417us/step - loss: 0.0151 -
acc: 0.9976
8334/8334 [==============================] - 2s 235us/step
Epoch 1/3
16667/16667 [==============================] - 7s 420us/step - loss: 0.1679 -
acc: 0.9443
Epoch 2/3
16667/16667 [==============================] - 7s 417us/step - loss: 0.0240 -
acc: 0.9945
Epoch 3/3
16667/16667 [==============================] - 7s 413us/step - loss: 0.0043 -
acc: 0.9999
8333/8333 [==============================] - 1s 124us/step
Epoch 1/3
16667/16667 [==============================] - 7s 412us/step - loss: 0.0190 -
acc: 0.9951
Epoch 2/3
16667/16667 [==============================] - 7s 422us/step - loss: 0.0036 -
acc: 0.9998
Epoch 3/3
16667/16667 [==============================] - 7s 418us/step - loss: 0.0016 -
acc: 0.9999
8333/8333 [==============================] - 1s 123us/step
Epoch 1/3
16666/16666 [==============================] - 9s 554us/step - loss: 0.3457 -
acc: 0.8537
Epoch 2/3
16666/16666 [==============================] - 7s 412us/step - loss: 0.0728 -
acc: 0.9778
Epoch 3/3
16666/16666 [==============================] - 7s 432us/step - loss: 0.0186 -
acc: 0.9969 2
8334/8334 [==============================] - 2s 256us/step
Epoch 1/3
16667/16667 [==============================] - 7s 434us/step - loss: 0.1829 -
acc: 0.9414
Epoch 2/3
16667/16667 [==============================] - 7s 444us/step - loss: 0.0335 -
acc: 0.9917
Epoch 3/3
16667/16667 [==============================] - 7s 439us/step - loss: 0.0084 -
acc: 0.9990
8333/8333 [==============================] - 1s 138us/step
Epoch 1/3
16667/16667 [==============================] - 7s 421us/step - loss: 0.0229 -
acc: 0.9932
Epoch 2/3
16667/16667 [==============================] - 7s 414us/step - loss: 0.0064 -
acc: 0.9998
Epoch 3/3
```

```
16667/16667 [==============================] - 7s 411us/step - loss: 0.0032 -
acc: 0.9998
8333/8333 [==============================] - 1s 132us/step
Epoch 1/3
16666/16666 [==============================] - 9s 554us/step - loss: 0.3603 -
acc: 0.8462
Epoch 2/3
16666/16666 [==============================] - 7s 411us/step - loss: 0.0846 -
acc: 0.9735
Epoch 3/3
16666/16666 [==============================] - 7s 413us/step - loss: 0.0265 -
acc: 0.9945
8334/8334 [==============================] - 2s 226us/step
Epoch 1/3
16667/16667 [==============================] - 7s 411us/step - loss: 0.1752 -
acc: 0.9429
Epoch 2/3
16667/16667 [==============================] - 7s 413us/step - loss: 0.0341 -
acc: 0.9919
Epoch 3/3
16667/16667 [==============================] - 7s 413us/step - loss: 0.0082 -
acc: 0.9991
8333/8333 [==============================] - 1s 127us/step
Epoch 1/3
16667/16667 [==============================] - 7s 417us/step - loss: 0.0210 -
acc: 0.9939
Epoch 2/3
16667/16667 [==============================] - 7s 418us/step - loss: 0.0067 -
acc: 0.9995
Epoch 3/3
16667/16667 [==============================] - 7s 420us/step - loss: 0.0031 -
acc: 0.9996
8333/8333 [==============================] - 1s 141us/step
Epoch 1/3
16666/16666 [==============================] - 9s 555us/step - loss: 0.3671 -
acc: 0.8420
Epoch 2/3
16666/16666 [==============================] - 7s 412us/step - loss: 0.0941 -
acc: 0.9708
Epoch 3/3
16666/16666 [==============================] - 7s 409us/step - loss: 0.0308 -
acc: 0.9935
8334/8334 [==============================] - 2s 226us/step
Epoch 1/3
16667/16667 [==============================] - 7s 413us/step - loss: 0.1784 -
acc: 0.9383
Epoch 2/3
16667/16667 [==============================] - 7s 411us/step - loss: 0.0409 -
acc: 0.9898
Epoch 3/3
16667/16667 [==============================] - 7s 412us/step - loss: 0.0141 -
acc: 0.9978
8333/8333 [==============================] - 1s 123us/step
Epoch 1/3
16667/16667 [==============================] - 7s 411us/step - loss: 0.0287 -
acc: 0.9920
Epoch 2/3
```

```
16667/16667 [==============================] - 7s 413us/step - loss: 0.0091 -
acc: 0.9989
Epoch 3/3
16667/16667 [==============================] - 7s 412us/step - loss: 0.0055 -
acc: 0.9996
8333/8333 [==============================] - 1s 133us/step
Epoch 1/3
16666/16666 [==============================] - 9s 558us/step - loss: 0.3780 -
acc: 0.8363
Epoch 2/3
16666/16666 [==============================] - 7s 432us/step - loss: 0.1128 -
acc: 0.9608
Epoch 3/3
16666/16666 [==============================] - 7s 432us/step - loss: 0.0445 -
acc: 0.9884
8334/8334 [==============================] - 2s 246us/step
Epoch 1/3
16667/16667 [==============================] - 7s 413us/step - loss: 0.1829 -
acc: 0.9386
Epoch 2/3
16667/16667 [==============================] - 7s 415us/step - loss: 0.0575 -
acc: 0.9849
Epoch 3/3
16667/16667 [==============================] - 7s 414us/step - loss: 0.0210 -
acc: 0.9962
8333/8333 [==============================] - 1s 129us/step
Epoch 1/3
16667/16667 [==============================] - 7s 421us/step - loss: 0.0391 -
acc: 0.9882
Epoch 2/3
16667/16667 [==============================] - 7s 431us/step - loss: 0.0153 -
acc: 0.9977
Epoch 3/3
16667/16667 [==============================] - 7s 427us/step - loss: 0.0076 -
acc: 0.9986
8333/8333 [==============================] - 1s 150us/step
Epoch 1/3
16666/16666 [==============================] - 9s 562us/step - loss: 0.3803 -
acc: 0.8366
Epoch 2/3
16666/16666 [==============================] - 7s 410us/step - loss: 0.1395 -
acc: 0.9504
Epoch 3/3
16666/16666 [==============================] - 7s 413us/step - loss: 0.0698 -
acc: 0.9794
8334/8334 [==============================] - 2s 233us/step
Epoch 1/3
16667/16667 [==============================] - 7s 409us/step - loss: 0.1915 -
acc: 0.9342
Epoch 2/3
16667/16667 [==============================] - 7s 413us/step - loss: 0.0792 -
acc: 0.9764
Epoch 3/3
16667/16667 [==============================] - 7s 417us/step - loss: 0.0386 -
acc: 0.9897
8333/8333 [==============================] - 1s 131us/step
Epoch 1/3
```

```
16667/16667 [==============================] - 7s 411us/step - loss: 0.0574 -
acc: 0.9815
Epoch 2/3
16667/16667 [==============================] - 7s 412us/step - loss: 0.0248 -
acc: 0.9935
Epoch 3/3
16667/16667 [==============================] - 7s 412us/step - loss: 0.0161 -
acc: 0.9960
8333/8333 [==============================] - 1s 140us/step
Epoch 1/3
16666/16666 [==============================] - 9s 562us/step - loss: 0.4089 -
acc: 0.8185
Epoch 2/3
16666/16666 [==============================] - 7s 409us/step - loss: 0.1725 -
acc: 0.9378
Epoch 3/3
16666/16666 [==============================] - 7s 408us/step - loss: 0.0985 -
acc: 0.9662
8334/8334 [==============================] - 2s 229us/step
Epoch 1/3
16667/16667 [==============================] - 7s 411us/step - loss: 0.2037 -
acc: 0.9276 0s - loss: 0.2014 - ac
Epoch 2/3
16667/16667 [==============================] - 7s 413us/step - loss: 0.1088 -
acc: 0.9640
Epoch 3/3
16667/16667 [==============================] - 7s 411us/step - loss: 0.0641 -
acc: 0.9798
8333/8333 [==============================] - 1s 122us/step
Epoch 1/3
16667/16667 [==============================] - 7s 412us/step - loss: 0.0792 -
acc: 0.9730
Epoch 2/3
16667/16667 [==============================] - 7s 411us/step - loss: 0.0532 -
acc: 0.9847
Epoch 3/3
16667/16667 [==============================] - 7s 413us/step - loss: 0.0350 -
acc: 0.9908
8333/8333 [==============================] - 1s 135us/step
Epoch 1/3
16666/16666 [==============================] - 9s 568us/step - loss: 0.4576 -
acc: 0.7972
Epoch 2/3
16666/16666 [==============================] - 7s 410us/step - loss: 0.2261 -
acc: 0.9171
Epoch 3/3
16666/16666 [==============================] - 7s 421us/step - loss: 0.1576 -
acc: 0.9432
8334/8334 [==============================] - 2s 251us/step
Epoch 1/3
16667/16667 [==============================] - 7s 423us/step - loss: 0.2300 -
acc: 0.9158
Epoch 2/3
16667/16667 [==============================] - 7s 423us/step - loss: 0.1624 -
acc: 0.9420
Epoch 3/3
16667/16667 [==============================] - 7s 420us/step - loss: 0.1271 -
```

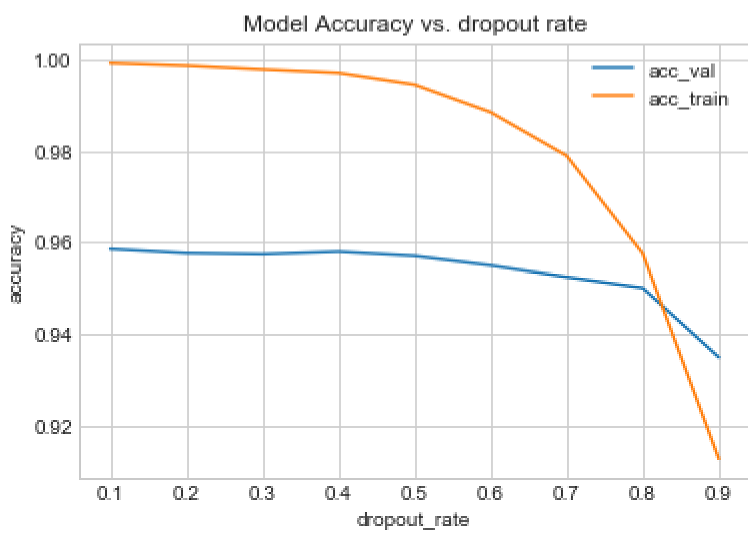
```
acc: 0.9572
8333/8333 [==============================] - 1s 138us/step
Epoch 1/3
16667/16667 [==============================] - 7s 409us/step - loss: 0.1395 -
acc: 0.9500
Epoch 2/3
16667/16667 [==============================] - 7s 410us/step - loss: 0.1138 -
acc: 0.9620
Epoch 3/3
16667/16667 [==============================] - 7s 410us/step - loss: 0.0836 -
acc: 0.9724
8333/8333 [==============================] - 1s 123us/step
Epoch 1/3
16666/16666 [==============================] - 10s 577us/step - loss: 0.6025
- acc: 0.7201
Epoch 2/3
16666/16666 [==============================] - 7s 420us/step - loss: 0.3487 -
acc: 0.8590
Epoch 3/3
16666/16666 [==============================] - 7s 418us/step - loss: 0.2781 -
acc: 0.8943
8334/8334 [==============================] - 2s 255us/step
Epoch 1/3
16667/16667 [==============================] - 7s 417us/step - loss: 0.3046 -
acc: 0.8816
Epoch 2/3
16667/16667 [==============================] - 8s 464us/step - loss: 0.2533 -
acc: 0.9060
Epoch 3/3
16667/16667 [==============================] - 7s 430us/step - loss: 0.2226 -
acc: 0.9167
8333/8333 [==============================] - 1s 139us/step
Epoch 1/3
16667/16667 [==============================] - 7s 407us/step - loss: 0.2397 -
acc: 0.9089
Epoch 2/3
16667/16667 [==============================] - 7s 413us/step - loss: 0.2120 -
acc: 0.9220
Epoch 3/3
16667/16667 [==============================] - 7s 441us/step - loss: 0.1995 -
acc: 0.9277
8333/8333 [==============================] - 1s 147us/step
```

In [45]:

```
ix = rate
loss_train_drop= pd.Series(loss_train_drop, index = ix)
loss_val_drop = pd.Series(loss_val_drop, index = ix)
acc_train_drop = pd.Series(acc_train_drop, index = ix)
acc_val_drop = pd.Series(acc_val_drop, index = ix)
```

In [46]:
```
fig = plt.figure()
ax = plt.axes()
ax.plot(loss_val_drop, label = 'loss_val')
ax.plot(loss_train_drop, label = 'loss_train')
ax.legend()
plt.xlabel('dropout_rate')
plt.ylabel('cross_entropy_loss')
plt.title('Model Loss vs. dropout rate')
fig.savefig("drop_loss_cv.png",dpi = 400)
```

In [47]:
```
fig = plt.figure()
ax = plt.axes()
ax.plot(acc_val_drop, label = 'acc_val')
ax.plot(acc_train_drop, label = 'acc_train')
ax.legend()
plt.xlabel('dropout_rate')
plt.ylabel('accuracy')
plt.title('Model Accuracy vs. dropout rate')
fig.savefig("drop_acc_cv.png",dpi = 400)
```



In [ ]:

In [26]:
```
from tensorflow.keras.layers import Dropout
model_drop = Sequential()
model_drop.add(Dense(250,input_shape = (max_num,)))
model_drop.add(Dropout(0.5))
model_drop.add(Activation('relu'))
model_drop.add(Dense(num_classes))
model_drop.add(Activation('softmax'))
model_drop.compile(loss = 'categorical_crossentropy', optimizer = 'adam', metr
ics = ['accuracy'])

hist_drop = model_drop.fit(X_train,y_train, batch_size=200, epochs = 3, verbos
e = 1)
score_drop = model_drop.evaluate(X_test,y_test, batch_size=200, verbose = 1)

drop_train_acc = hist_drop.history.get('acc')[-1]
drop_test_acc = score_drop[1]
```

```
Epoch 1/3
25000/25000 [==============================] - 14s 568us/step - loss: 0.3484
- acc: 0.8506
Epoch 2/3
25000/25000 [==============================] - 13s 505us/step - loss: 0.1211
- acc: 0.9577
Epoch 3/3
25000/25000 [==============================] - 13s 507us/step - loss: 0.0536
- acc: 0.9844
25000/25000 [==============================] - 6s 250us/step
```

In [27]:
```
print('The accurate rate of the model with dropout(0.5) on training dataset i
s')
print(drop_train_acc)
print('The accurate rate of the model with dropout(0.5) on test dataset is')
print(drop_test_acc)
```

```
The accurate rate of the model with dropout(0.5) on training dataset is
0.9844000105857849
The accurate rate of the model with dropout(0.5) on test dataset is
0.8729600014686585
```

In [92]:
```
from tensorflow.keras.layers import Dropout
model_125 = Sequential()
model_125.add(Dense(125,input_shape = (max_num,)))
model_125.add(Activation('relu'))
model_125.add(Dense(num_classes))
model_125.add(Activation('softmax'))
model_125.compile(loss = 'categorical_crossentropy', optimizer = 'adam', metri
cs = ['accuracy'])

hist_125 = model_125.fit(X_train,y_train, batch_size=200, epochs = 3, verbose
= 1)
score_125 = model_125.evaluate(X_test,y_test, batch_size=200, verbose = 1)
nn125_train_acc = hist_125.history.get('acc')[-1]
nn125_test_acc = score_125[1]
```

```
Epoch 1/3
25000/25000 [==============================] - 7s 283us/step - loss: 0.3292 -
acc: 0.8593
Epoch 2/3
25000/25000 [==============================] - 6s 246us/step - loss: 0.0769 -
acc: 0.9771
Epoch 3/3
25000/25000 [==============================] - 6s 244us/step - loss: 0.0193 -
acc: 0.9965
25000/25000 [==============================] - 3s 105us/step
```

In [93]:
```
print('The accurate rate of the model with 125 hidden units on training datase
t is')
print(nn125_train_acc)
print('The accurate rate of the model with 125 hidden units on test dataset i
s')
print(nn125_test_acc)
```

```
The accurate rate of the model with 125 hidden units on training dataset is
0.9965200033187867
The accurate rate of the model with 125 hidden units on test dataset is
0.8692400031089783
```

In [28]:
```
from tensorflow.keras.layers import Dropout
model_drop = Sequential()
model_drop.add(Dense(250,input_shape = (max_num,)))
model_drop.add(Dropout(0.4))
model_drop.add(Activation('relu'))
model_drop.add(Dense(num_classes))
model_drop.add(Activation('softmax'))
model_drop.compile(loss = 'categorical_crossentropy', optimizer = 'adam', metr
ics = ['accuracy'])

hist_drop = model_drop.fit(X_train,y_train, batch_size=200, epochs = 3, verbos
e = 1)
score_drop = model_drop.evaluate(X_test,y_test, batch_size=200, verbose = 1)

drop_train_acc = hist_drop.history.get('acc')[-1]
drop_test_acc = score_drop[1]
```

```
Epoch 1/3
25000/25000 [==============================] - 14s 567us/step - loss: 0.3492
- acc: 0.8533
Epoch 2/3
25000/25000 [==============================] - 13s 520us/step - loss: 0.1024
- acc: 0.9656
Epoch 3/3
25000/25000 [==============================] - 13s 516us/step - loss: 0.0349
- acc: 0.9922
25000/25000 [==============================] - 5s 213us/step
```

In [29]:
```
print('The accurate rate of the model with dropout(0.4) on training dataset i
s')
print(drop_train_acc)
print('The accurate rate of the model with dropout(0.4) on test dataset is')
print(drop_test_acc)
```

```
The accurate rate of the model with dropout(0.4) on training dataset is
0.9922400074005127
The accurate rate of the model with dropout(0.4) on test dataset is
0.8701600012779236
```

In [30]:
```
from tensorflow.keras.layers import Dropout
model_drop = Sequential()
model_drop.add(Dense(250,input_shape = (max_num,)))
model_drop.add(Dropout(0.6))
model_drop.add(Activation('relu'))
model_drop.add(Dense(num_classes))
model_drop.add(Activation('softmax'))
model_drop.compile(loss = 'categorical_crossentropy', optimizer = 'adam', metr
ics = ['accuracy'])

hist_drop = model_drop.fit(X_train,y_train, batch_size=200, epochs = 3, verbos
e = 1)
score_drop = model_drop.evaluate(X_test,y_test, batch_size=200, verbose = 1)

drop_train_acc = hist_drop.history.get('acc')[-1]
drop_test_acc = score_drop[1]
```

```
Epoch 1/3
25000/25000 [==============================] - 14s 568us/step - loss: 0.3573
- acc: 0.8468
Epoch 2/3
25000/25000 [==============================] - 13s 503us/step - loss: 0.1421
- acc: 0.9507
Epoch 3/3
25000/25000 [==============================] - 13s 507us/step - loss: 0.0742
- acc: 0.9769
25000/25000 [==============================] - 6s 223us/step
```

In [31]:
```
print('The accurate rate of the model with dropout(0.6) on training dataset i
s')
print(drop_train_acc)
print('The accurate rate of the model with dropout(0.6) on test dataset is')
print(drop_test_acc)
```

```
The accurate rate of the model with dropout(0.6) on training dataset is
0.9768800086975098
The accurate rate of the model with dropout(0.6) on test dataset is
0.8719600014686585
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [66]:
```
from tensorflow.keras.regularizers import l1
from tensorflow.keras.regularizers import l2
```

In [72]:
```
lambda_list = [1e-6,1e-5,1e-4,1e-3,1e-2,1e-1]
loss_train_l1 = []
acc_train_l1 = []
loss_val_l1 = []
acc_val_l1 = []
for par in lambda_list:
    model_l1 = Sequential()
    model_l1.add(Dense(250,input_shape = (max_num,), kernel_regularizer = l1(p
ar), bias_regularizer=l1(par)))
    model_l1.add(Activation('relu'))
    model_l1.add(Dense(num_classes, kernel_regularizer = l1(par), bias_regular
izer=l1(par)))
    model_l1.add(Activation('softmax'))
    model_l1.compile(loss = 'categorical_crossentropy', optimizer = 'adam', me
trics = ['accuracy'])
    loss_val_cv = 0
    acc_val_cv = 0
    loss_train_cv = 0
    acc_train_cv = 0
    for train_cv_index, val_index in kf.split(X_train):
        X_train_cv = X_train[train_cv_index]
        y_train_cv = y_train[train_cv_index]
        X_val_cv = X_train[val_index]
        y_val_cv = y_train[val_index]
        hist = model_l1.fit(X_train_cv,y_train_cv,batch_size=200, epochs = 3)
        loss_train_cv = loss_train_cv + hist.history.get('loss')[-1]
        acc_train_cv = acc_train_cv + hist.history.get('acc')[-1]
        score_val_cv = model_l1.evaluate(X_val_cv,y_val_cv, batch_size=200, ve
rbose = 1)
        loss_val_cv = loss_val_cv + score_val_cv[0]
        acc_val_cv = acc_val_cv + score_val_cv[1]
    loss_val_l1.append(loss_val_cv/3)
    acc_val_l1.append(acc_val_cv/3)
    loss_train_l1.append(loss_train_cv/3)
    acc_train_l1.append(acc_train_cv/3)
```

```
Epoch 1/3
16666/16666 [==============================] - 9s 526us/step - loss: 0.3740 -
acc: 0.8564
Epoch 2/3
16666/16666 [==============================] - 8s 476us/step - loss: 0.0917 -
acc: 0.9828
Epoch 3/3
16666/16666 [==============================] - 8s 478us/step - loss: 0.0435 -
acc: 0.9976
8334/8334 [==============================] - 1s 168us/step
Epoch 1/3
16667/16667 [==============================] - 8s 480us/step - loss: 0.2051 -
acc: 0.9390 2s - l
Epoch 2/3
16667/16667 [==============================] - 8s 477us/step - loss: 0.0574 -
acc: 0.9950
Epoch 3/3
16667/16667 [==============================] - 8s 481us/step - loss: 0.0378 -
acc: 0.9996
8333/8333 [==============================] - 1s 161us/step
Epoch 1/3
16667/16667 [==============================] - 8s 476us/step - loss: 0.0578 -
acc: 0.9931
Epoch 2/3
16667/16667 [==============================] - 8s 477us/step - loss: 0.0388 -
acc: 0.9997
Epoch 3/3
16667/16667 [==============================] - 8s 480us/step - loss: 0.0341 -
acc: 1.0000
8333/8333 [==============================] - 1s 156us/step
Epoch 1/3
16666/16666 [==============================] - 9s 519us/step - loss: 0.5924 -
acc: 0.8473
Epoch 2/3
16666/16666 [==============================] - 8s 475us/step - loss: 0.2856 -
acc: 0.9798
Epoch 3/3
16666/16666 [==============================] - 8s 475us/step - loss: 0.2049 -
acc: 0.9967
8334/8334 [==============================] - 1s 168us/step - ETA
Epoch 1/3
16667/16667 [==============================] - 8s 474us/step - loss: 0.3510 -
acc: 0.9398
Epoch 2/3
16667/16667 [==============================] - 8s 478us/step - loss: 0.2218 -
acc: 0.9926
Epoch 3/3
16667/16667 [==============================] - 8s 481us/step - loss: 0.1700 -
acc: 0.9995
8333/8333 [==============================] - 1s 156us/step
Epoch 1/3
16667/16667 [==============================] - 8s 475us/step - loss: 0.2129 -
acc: 0.9810
Epoch 2/3
16667/16667 [==============================] - 8s 478us/step - loss: 0.1758 -
acc: 0.9969
Epoch 3/3
```

```
16667/16667 [==============================] - 8s 475us/step - loss: 0.1414 -
acc: 1.0000
8333/8333 [==============================] - 1s 157us/step
Epoch 1/3
16666/16666 [==============================] - 9s 528us/step - loss: 1.8369 -
acc: 0.8526
Epoch 2/3
16666/16666 [==============================] - 8s 487us/step - loss: 0.7665 -
acc: 0.9507
Epoch 3/3
16666/16666 [==============================] - 8s 486us/step - loss: 0.5812 -
acc: 0.9566
8334/8334 [==============================] - 1s 169us/step
Epoch 1/3
16667/16667 [==============================] - 8s 497us/step - loss: 0.7104 -
acc: 0.9189
Epoch 2/3
16667/16667 [==============================] - 8s 476us/step - loss: 0.5210 -
acc: 0.9674
Epoch 3/3
16667/16667 [==============================] - 8s 480us/step - loss: 0.3922 -
acc: 0.9801
8333/8333 [==============================] - 1s 154us/step
Epoch 1/3
16667/16667 [==============================] - 8s 475us/step - loss: 0.5506 -
acc: 0.9392
Epoch 2/3
16667/16667 [==============================] - 8s 479us/step - loss: 0.4241 -
acc: 0.9803 4s - l - ETA: 1s - loss:
Epoch 3/3
16667/16667 [==============================] - 8s 472us/step - loss: 0.3144 -
acc: 0.9908
8333/8333 [==============================] - 1s 152us/step
Epoch 1/3
16666/16666 [==============================] - 9s 534us/step - loss: 6.3297 -
acc: 0.8357
Epoch 2/3
16666/16666 [==============================] - 8s 503us/step - loss: 1.3332 -
acc: 0.8669
Epoch 3/3
16666/16666 [==============================] - 8s 494us/step - loss: 1.2293 -
acc: 0.8711
8334/8334 [==============================] - 1s 177us/step
Epoch 1/3
16667/16667 [==============================] - 8s 495us/step - loss: 1.1430 -
acc: 0.8764
Epoch 2/3
16667/16667 [==============================] - 8s 502us/step - loss: 1.0341 -
acc: 0.8869
Epoch 3/3
16667/16667 [==============================] - 8s 493us/step - loss: 0.9667 -
acc: 0.8915
8333/8333 [==============================] - 1s 163us/step
Epoch 1/3
16667/16667 [==============================] - 8s 482us/step - loss: 0.9496 -
acc: 0.8910
Epoch 2/3
```

```
16667/16667 [==============================] - 9s 538us/step - loss: 0.8667 -
acc: 0.9035
Epoch 3/3
16667/16667 [==============================] - 10s 577us/step - loss: 0.8105
- acc: 0.9111
8333/8333 [==============================] - 2s 217us/step
Epoch 1/3
16666/16666 [==============================] - 10s 600us/step - loss: 42.1275
- acc: 0.7672
Epoch 2/3
16666/16666 [==============================] - 9s 557us/step - loss: 3.8045 -
acc: 0.8068
Epoch 3/3
16666/16666 [==============================] - 9s 552us/step - loss: 3.5522 -
acc: 0.8283
8334/8334 [==============================] - 2s 233us/step
Epoch 1/3
16667/16667 [==============================] - 9s 532us/step - loss: 3.5191 -
acc: 0.8345
Epoch 2/3
16667/16667 [==============================] - 9s 547us/step - loss: 3.5023 -
acc: 0.8403
Epoch 3/3
16667/16667 [==============================] - 9s 551us/step - loss: 3.4889 -
acc: 0.8462
8333/8333 [==============================] - 2s 203us/step
Epoch 1/3
16667/16667 [==============================] - 9s 533us/step - loss: 3.4878 -
acc: 0.8511
Epoch 2/3
16667/16667 [==============================] - 9s 561us/step - loss: 3.4870 -
acc: 0.8526
Epoch 3/3
16667/16667 [==============================] - 9s 567us/step - loss: 3.4820 -
acc: 0.8585
8333/8333 [==============================] - 2s 220us/step
Epoch 1/3
16666/16666 [==============================] - 10s 625us/step - loss: 412.166
0 - acc: 0.5414
Epoch 2/3
16666/16666 [==============================] - 10s 583us/step - loss: 31.7424
- acc: 0.4999
Epoch 3/3
16666/16666 [==============================] - 9s 560us/step - loss: 30.8289
- acc: 0.4989
8334/8334 [==============================] - 2s 239us/step
Epoch 1/3
16667/16667 [==============================] - 9s 543us/step - loss: 30.8347
- acc: 0.4955
Epoch 2/3
16667/16667 [==============================] - 10s 579us/step - loss: 30.8026
- acc: 0.5010
Epoch 3/3
16667/16667 [==============================] - 11s 673us/step - loss: 30.8211
- acc: 0.5049
8333/8333 [==============================] - 2s 218us/step
Epoch 1/3
```

```
16667/16667 [==============================] - 9s 545us/step - loss: 30.7826
- acc: 0.5021
Epoch 2/3
16667/16667 [==============================] - 9s 559us/step - loss: 30.8032
- acc: 0.5048
Epoch 3/3
16667/16667 [==============================] - 9s 560us/step - loss: 30.7899
- acc: 0.5039
8333/8333 [==============================] - 2s 219us/step
```

In [73]:
```
print(loss_train_l1)
print(loss_val_l1)
```

```
[0.03845995333584134, 0.17209522215436823, 0.42926743624753066, 1.00215903385
95625, 3.507723528102316, 30.813288159613112]
[0.17665963721892683, 0.297276239302909, 0.5640312226375032, 1.02416921316438
85, 3.496250938156813, 30.906951873829343]
```

In [74]:
```
print(acc_train_l1)
print(acc_val_l1)
```

```
[0.9990799712760805, 0.9987399603998445, 0.9758396216811569, 0.89123959722478
43, 0.8443396800387178, 0.5025999266247423]
[0.956642814160677, 0.9464422868833272, 0.9096412133018205, 0.878440065659191
9, 0.8461607332347469, 0.504719904790355]
```

In [95]:
```
l1_reg_cv = pd.DataFrame({'loss_train': loss_train_l1,
                          'loss_val': loss_val_l1,
                       'acc_train': acc_train_l1,
                        'acc_val':acc_val_l1,
                         'lambda':lambda_list}).set_index('lambda')
l1_reg_cv
```

Out[95]:

lambda	loss_train	loss_val	acc_train	acc_val
0.000001	0.038460	0.176660	0.99908	0.956643
0.000010	0.172095	0.297276	0.99874	0.946442
0.000100	0.429267	0.564031	0.97584	0.909641
0.001000	1.002159	1.024169	0.89124	0.878440
0.010000	3.507724	3.496251	0.84434	0.846161
0.100000	30.813288	30.906952	0.50260	0.504720

In [87]:
```
# indicating by the result of l1_reg_cv data frame, will set the lambda as 1e-
6 for l1 norm regularization
model_l1 = Sequential()
model_l1.add(Dense(250,input_shape = (max_num,), kernel_regularizer = l1(1e-6
), bias_regularizer=l1(1e-6)))
model_l1.add(Activation('relu'))
model_l1.add(Dense(num_classes, kernel_regularizer = l1(1e-6), bias_regularize
r=l1(1e-6)))
model_l1.add(Activation('softmax'))
model_l1.compile(loss = 'categorical_crossentropy', optimizer = 'adam', metric
s = ['accuracy'])

hist_l1 = model_l1.fit(X_train,y_train, batch_size=200, epochs = 3, verbose =
1)
score_l1 = model_l1.evaluate(X_test,y_test, batch_size=200, verbose = 1)

l1_train_acc = hist_l1.history.get('acc')[-1]
l1_test_acc = score_l1[1]
```

```
Epoch 1/3
25000/25000 [==============================] - 13s 522us/step - loss: 0.3630
- acc: 0.85690s - loss: 0.3652 - acc: 0
Epoch 2/3
25000/25000 [==============================] - 12s 471us/step - loss: 0.0964
- acc: 0.9801
Epoch 3/3
25000/25000 [==============================] - 12s 469us/step - loss: 0.0456
- acc: 0.9980
25000/25000 [==============================] - 4s 166us/step
```

In [88]:
```
print('The accurate rate of the model with l1 regularization on training datas
et is')
print(l1_train_acc)
print('The accurate rate of the model with l1 regularization on test dataset i
s')
print(l1_test_acc)
```

```
The accurate rate of the model with l1 regularization on training dataset is
0.9979600019454956
The accurate rate of the model with l1 regularization on test dataset is
0.8711199998855591
```

In [108]:
```
# indicating by the result of l1_reg_cv data frame, will set the lambda as 1e-
5 for l1 norm regularization
model_l1 = Sequential()
model_l1.add(Dense(250,input_shape = (max_num,), kernel_regularizer = l1(1e-5
), bias_regularizer=l1(1e-5)))
model_l1.add(Activation('relu'))
model_l1.add(Dense(num_classes, kernel_regularizer = l1(1e-5), bias_regularize
r=l1(1e-5)))
model_l1.add(Activation('softmax'))
model_l1.compile(loss = 'categorical_crossentropy', optimizer = 'adam', metric
s = ['accuracy'])

hist_l1 = model_l1.fit(X_train,y_train, batch_size=200, epochs = 3, verbose =
1)
score_l1 = model_l1.evaluate(X_test,y_test, batch_size=200, verbose = 1)

l1_train_acc = hist_l1.history.get('acc')[-1]
l1_test_acc = score_l1[1]
```

```
Epoch 1/3
25000/25000 [==============================] - 14s 563us/step - loss: 0.5563
- acc: 0.8622
Epoch 2/3
25000/25000 [==============================] - 12s 476us/step - loss: 0.2818
- acc: 0.9758
Epoch 3/3
25000/25000 [==============================] - 12s 478us/step - loss: 0.1938
- acc: 0.9958
25000/25000 [==============================] - 4s 175us/step
```

In [109]:
```
print('The accurate rate of the model with l1 regularization on training datas
et is')
print(l1_train_acc)
print('The accurate rate of the model with l1 regularization on test dataset i
s')
print(l1_test_acc)
```

```
The accurate rate of the model with l1 regularization on training dataset is
0.9958400039672851
The accurate rate of the model with l1 regularization on test dataset is
0.8743999996185303
```

In [ ]:

In [100]:
```
lambda_list = [1e-6,1e-5,1e-4,1e-3,1e-2,1e-1]
loss_train_l2 = []
acc_train_l2 = []
loss_val_l2 = []
acc_val_l2 = []
for par in lambda_list:
    model_l2 = Sequential()
    model_l2.add(Dense(250,input_shape = (max_num,), kernel_regularizer = l2(p
ar), bias_regularizer=l2(par)))
    model_l2.add(Activation('relu'))
    model_l2.add(Dense(num_classes, kernel_regularizer = l2(par), bias_regular
izer=l2(par)))
    model_l2.add(Activation('softmax'))
    model_l2.compile(loss = 'categorical_crossentropy', optimizer = 'adam', me
trics = ['accuracy'])
    loss_val_cv = 0
    acc_val_cv = 0
    loss_train_cv = 0
    acc_train_cv = 0
    for train_cv_index, val_index in kf.split(X_train):
        X_train_cv = X_train[train_cv_index]
        y_train_cv = y_train[train_cv_index]
        X_val_cv = X_train[val_index]
        y_val_cv = y_train[val_index]
        hist = model_l2.fit(X_train_cv,y_train_cv,batch_size=200, epochs = 3)
        loss_train_cv = loss_train_cv + hist.history.get('loss')[-1]
        acc_train_cv = acc_train_cv + hist.history.get('acc')[-1]
        score_val_cv = model_l2.evaluate(X_val_cv,y_val_cv, batch_size=200, ve
rbose = 1)
        loss_val_cv = loss_val_cv + score_val_cv[0]
        acc_val_cv = acc_val_cv + score_val_cv[1]
    loss_val_l2.append(loss_val_cv/3)
    acc_val_l2.append(acc_val_cv/3)
    loss_train_l2.append(loss_train_cv/3)
    acc_train_l2.append(acc_train_cv/3)
```

```
Epoch 1/3
16666/16666 [==============================] - 10s 583us/step - loss: 0.3717
- acc: 0.8422
Epoch 2/3
16666/16666 [==============================] - 8s 473us/step - loss: 0.0603 -
acc: 0.9833
Epoch 3/3
16666/16666 [==============================] - 8s 473us/step - loss: 0.0151 -
acc: 0.9984
8334/8334 [==============================] - 2s 208us/step
Epoch 1/3
16667/16667 [==============================] - 8s 487us/step - loss: 0.1704 -
acc: 0.9417
Epoch 2/3
16667/16667 [==============================] - 8s 496us/step - loss: 0.0239 -
acc: 0.9950
Epoch 3/3
16667/16667 [==============================] - 8s 486us/step - loss: 0.0062 -
acc: 0.9997
8333/8333 [==============================] - 1s 158us/step
Epoch 1/3
16667/16667 [==============================] - 8s 474us/step - loss: 0.0190 -
acc: 0.9953
Epoch 2/3
16667/16667 [==============================] - 9s 529us/step - loss: 0.0043 -
acc: 0.9999
Epoch 3/3
16667/16667 [==============================] - 9s 570us/step - loss: 0.0022 -
acc: 1.0000
8333/8333 [==============================] - 2s 187us/step
Epoch 1/3
16666/16666 [==============================] - 12s 712us/step - loss: 0.3552
- acc: 0.8497
Epoch 2/3
16666/16666 [==============================] - 13s 802us/step - loss: 0.0652
- acc: 0.98342s - loss: 0.
Epoch 3/3
16666/16666 [==============================] - 14s 848us/step - loss: 0.0200
- acc: 0.9981
8334/8334 [==============================] - 3s 330us/step
Epoch 1/3
16667/16667 [==============================] - 9s 562us/step - loss: 0.1804 -
acc: 0.9425
Epoch 2/3
16667/16667 [==============================] - 9s 557us/step - loss: 0.0324 -
acc: 0.9948
Epoch 3/3
16667/16667 [==============================] - 10s 616us/step - loss: 0.0139
- acc: 0.9998
8333/8333 [==============================] - 2s 275us/step
Epoch 1/3
16667/16667 [==============================] - 9s 565us/step - loss: 0.0300 -
acc: 0.9944
Epoch 2/3
16667/16667 [==============================] - 10s 614us/step - loss: 0.0152
- acc: 0.9996
Epoch 3/3
```

```
16667/16667 [==============================] - 10s 583us/step - loss: 0.0114
- acc: 0.9999
8333/8333 [==============================] - 2s 221us/step
Epoch 1/3
16666/16666 [==============================] - 11s 664us/step - loss: 0.3909
- acc: 0.8548
Epoch 2/3
16666/16666 [==============================] - 9s 556us/step - loss: 0.1128 -
acc: 0.9804
Epoch 3/3
16666/16666 [==============================] - 9s 561us/step - loss: 0.0632 -
acc: 0.9982
8334/8334 [==============================] - 2s 269us/step
Epoch 1/3
16667/16667 [==============================] - 9s 542us/step - loss: 0.2255 -
acc: 0.9409
Epoch 2/3
16667/16667 [==============================] - 9s 560us/step - loss: 0.0857 -
acc: 0.9935
Epoch 3/3
16667/16667 [==============================] - 9s 555us/step - loss: 0.0590 -
acc: 0.9996
8333/8333 [==============================] - 2s 214us/step
Epoch 1/3
16667/16667 [==============================] - 9s 538us/step - loss: 0.0807 -
acc: 0.9914
Epoch 2/3
16667/16667 [==============================] - 9s 569us/step - loss: 0.0585 -
acc: 0.9994
Epoch 3/3
16667/16667 [==============================] - 10s 578us/step - loss: 0.0497
- acc: 0.9999
8333/8333 [==============================] - 2s 228us/step
Epoch 1/3
16666/16666 [==============================] - 11s 679us/step - loss: 0.6667
- acc: 0.8456
Epoch 2/3
16666/16666 [==============================] - 9s 563us/step - loss: 0.2985 -
acc: 0.9766
Epoch 3/3
16666/16666 [==============================] - 9s 555us/step - loss: 0.1959 -
acc: 0.9917
8334/8334 [==============================] - 2s 266us/step
Epoch 1/3
16667/16667 [==============================] - 9s 545us/step - loss: 0.3584 -
acc: 0.9272
Epoch 2/3
16667/16667 [==============================] - 9s 561us/step - loss: 0.2332 -
acc: 0.9839
Epoch 3/3
16667/16667 [==============================] - 9s 557us/step - loss: 0.1580 -
acc: 0.9972
8333/8333 [==============================] - 2s 215us/step
Epoch 1/3
16667/16667 [==============================] - 9s 537us/step - loss: 0.2632 -
acc: 0.9570
Epoch 2/3
```

```
16667/16667 [==============================] - 9s 570us/step - loss: 0.2040 -
acc: 0.9899
Epoch 3/3
16667/16667 [==============================] - 10s 572us/step - loss: 0.1464
- acc: 0.9978
8333/8333 [==============================] - 2s 209us/step
Epoch 1/3
16666/16666 [==============================] - 11s 661us/step - loss: 1.5827
- acc: 0.8511
Epoch 2/3
16666/16666 [==============================] - 9s 563us/step - loss: 0.5100 -
acc: 0.9179
Epoch 3/3
16666/16666 [==============================] - 9s 563us/step - loss: 0.4485 -
acc: 0.9183
8334/8334 [==============================] - 2s 270us/step
Epoch 1/3
16667/16667 [==============================] - 9s 551us/step - loss: 0.4926 -
acc: 0.9015
Epoch 2/3
16667/16667 [==============================] - 9s 561us/step - loss: 0.3963 -
acc: 0.9225
Epoch 3/3
16667/16667 [==============================] - 9s 552us/step - loss: 0.3761 -
acc: 0.9270
8333/8333 [==============================] - 2s 212us/step
Epoch 1/3
16667/16667 [==============================] - 9s 541us/step - loss: 0.4299 -
acc: 0.9061
Epoch 2/3
16667/16667 [==============================] - 9s 563us/step - loss: 0.3382 -
acc: 0.9375
Epoch 3/3
16667/16667 [==============================] - 10s 573us/step - loss: 0.3179
- acc: 0.9410
8333/8333 [==============================] - 2s 228us/step
Epoch 1/3
16666/16666 [==============================] - 11s 673us/step - loss: 5.3259
- acc: 0.8403
Epoch 2/3
16666/16666 [==============================] - 10s 573us/step - loss: 0.7179
- acc: 0.8597
Epoch 3/3
16666/16666 [==============================] - 9s 551us/step - loss: 0.6690 -
acc: 0.8648
8334/8334 [==============================] - 2s 269us/step
Epoch 1/3
16667/16667 [==============================] - 9s 539us/step - loss: 0.6765 -
acc: 0.8616
Epoch 2/3
16667/16667 [==============================] - 9s 567us/step - loss: 0.6584 -
acc: 0.8635
Epoch 3/3
16667/16667 [==============================] - 10s 584us/step - loss: 0.6530
- acc: 0.8642
8333/8333 [==============================] - 2s 235us/step
Epoch 1/3
```

```
16667/16667 [==============================] - 10s 576us/step - loss: 0.6417
- acc: 0.8679
Epoch 2/3
16667/16667 [==============================] - 10s 573us/step - loss: 0.6301
- acc: 0.8741
Epoch 3/3
16667/16667 [==============================] - 9s 558us/step - loss: 0.6249 -
acc: 0.8689
8333/8333 [==============================] - 2s 211us/step
```

In [101]:

```
print(loss_train_l2)
print(loss_val_l2)
```

```
[0.007824130377315422, 0.015088598974607422, 0.057281762232496654, 0.16676480
56307641, 0.3808560894824569, 0.6489676033762257]
[0.1390206992715748, 0.15104453086864839, 0.19869745330250457, 0.330773005932
91253, 0.48299898957397297, 0.6537691378268679]
```

In [102]:

```
print(acc_train_l2)
print(acc_val_l2)
```

```
[0.9993599810094662, 0.9992599771022933, 0.9992399799237882, 0.99559992659116
89, 0.9287797917134272, 0.8659399778324643]
[0.959482759669288, 0.9571628738318507, 0.9539227791629714, 0.921921695183919
2, 0.8826001151991272, 0.8591601228896826]
```

In [103]:

```
l2_reg_cv = pd.DataFrame({'loss_train': loss_train_l2,
                          'loss_val': loss_val_l2,
                       'acc_train': acc_train_l2,
                        'acc_val':acc_val_l2,
                         'lambda':lambda_list}).set_index('lambda')
l2_reg_cv
```

Out[103]:

lambda	loss_train	loss_val	acc_train	acc_val
0.000001	0.007824	0.139021	0.99936	0.959483
0.000010	0.015089	0.151045	0.99926	0.957163
0.000100	0.057282	0.198697	0.99924	0.953923
0.001000	0.166765	0.330773	0.99560	0.921922
0.010000	0.380856	0.482999	0.92878	0.882600
0.100000	0.648968	0.653769	0.86594	0.859160

In [104]:
```
model_l2 = Sequential()
model_l2.add(Dense(250,input_shape = (max_num,), kernel_regularizer = l2(1e-6
), bias_regularizer=l2(1e-6)))
model_l2.add(Activation('relu'))
model_l2.add(Dense(num_classes, kernel_regularizer = l2(1e-6), bias_regularize
r=l2(1e-6)))
model_l2.add(Activation('softmax'))
model_l2.compile(loss = 'categorical_crossentropy', optimizer = 'adam', metric
s = ['accuracy'])

hist_l2 = model_l2.fit(X_train,y_train, batch_size=200, epochs = 3, verbose =
1)
score_l2 = model_l2.evaluate(X_test,y_test, batch_size=200, verbose = 1)

l2_train_acc = hist_l2.history.get('acc')[-1]
l2_test_acc = score_l2[1]
```

```
Epoch 1/3
25000/25000 [==============================] - 14s 569us/step - loss: 0.3231
- acc: 0.8651
Epoch 2/3
25000/25000 [==============================] - 12s 485us/step - loss: 0.0702
- acc: 0.9788
Epoch 3/3
25000/25000 [==============================] - 13s 511us/step - loss: 0.0148
- acc: 0.9978
25000/25000 [==============================] - 8s 323us/step
```

In [105]:
```
print('The accurate rate of the model with l2 regularization on training datas
et is')
print(l2_train_acc)
print('The accurate rate of the model with l2 regularization on test dataset i
s')
print(l2_test_acc)
```

```
The accurate rate of the model with l2 regularization on training dataset is
0.9977600021362305
The accurate rate of the model with l2 regularization on test dataset is
0.8718800010681153
```

In [106]:
```
model_l2 = Sequential()
model_l2.add(Dense(250,input_shape = (max_num,), kernel_regularizer = l2(1e-5
), bias_regularizer=l2(1e-5)))
model_l2.add(Activation('relu'))
model_l2.add(Dense(num_classes, kernel_regularizer = l2(1e-5), bias_regularize
r=l2(1e-5)))
model_l2.add(Activation('softmax'))
model_l2.compile(loss = 'categorical_crossentropy', optimizer = 'adam', metric
s = ['accuracy'])

hist_l2 = model_l2.fit(X_train,y_train, batch_size=200, epochs = 3, verbose =
1)
score_l2 = model_l2.evaluate(X_test,y_test, batch_size=200, verbose = 1)

l2_train_acc = hist_l2.history.get('acc')[-1]
l2_test_acc = score_l2[1]
```

```
Epoch 1/3
25000/25000 [==============================] - 14s 555us/step - loss: 0.3362
- acc: 0.8628
Epoch 2/3
25000/25000 [==============================] - 12s 473us/step - loss: 0.0740
- acc: 0.9798
Epoch 3/3
25000/25000 [==============================] - 13s 508us/step - loss: 0.0220
- acc: 0.9979
25000/25000 [==============================] - 6s 249us/step
```

In [107]:
```
print('The accurate rate of the model with l2 regularization on training datas
et is')
print(l2_train_acc)
print('The accurate rate of the model with l2 regularization on test dataset i
s')
print(l2_test_acc)
```

```
The accurate rate of the model with l2 regularization on training dataset is
0.9978800020217895
The accurate rate of the model with l2 regularization on test dataset is
0.8700399985313415
```

In [ ]:

two hidden layers

In [36]:
```
units = range(10,200,10)
loss_train_2 = []
acc_train_2 = []
loss_val_2 = []
acc_val_2 = []
for num in units:
    model_2 = Sequential()
    model_2.add(Dense(num,input_shape = (max_num,)))
    model_2.add(Activation('relu'))
    model_2.add(Dense(num))
    model_2.add(Activation('relu'))
    model_2.add(Dense(num_classes))
    model_2.add(Activation('softmax'))
    model_2.compile(loss = 'categorical_crossentropy', optimizer = 'adam', met
rics = ['accuracy'])
    loss_val_cv = 0
    acc_val_cv = 0
    loss_train_cv = 0
    acc_train_cv = 0
    for train_cv_index, val_index in kf.split(X_train):
        X_train_cv = X_train[train_cv_index]
        y_train_cv = y_train[train_cv_index]
        X_val_cv = X_train[val_index]
        y_val_cv = y_train[val_index]
        hist = model_2.fit(X_train_cv,y_train_cv,batch_size=200, epochs = 3)
        loss_train_cv = loss_train_cv + hist.history.get('loss')[-1]
        acc_train_cv = acc_train_cv + hist.history.get('acc')[-1]
        score_val_cv = model_2.evaluate(X_val_cv,y_val_cv, batch_size=200, ver
bose = 1)
        loss_val_cv = loss_val_cv + score_val_cv[0]
        acc_val_cv = acc_val_cv + score_val_cv[1]
    loss_val_2.append(loss_val_cv/3)
    acc_val_2.append(acc_val_cv/3)
    loss_train_2.append(loss_train_cv/3)
    acc_train_2.append(acc_train_cv/3)
```

```
Epoch 1/3
16666/16666 [==============================] - 6s 383us/step - loss: 0.4289 -
acc: 0.8081
Epoch 2/3
16666/16666 [==============================] - 4s 255us/step - loss: 0.1823 -
acc: 0.9371
Epoch 3/3
16666/16666 [==============================] - 4s 259us/step - loss: 0.1024 -
acc: 0.9699
8334/8334 [==============================] - 2s 296us/step
Epoch 1/3
16667/16667 [==============================] - 4s 237us/step - loss: 0.2068 -
acc: 0.9312
Epoch 2/3
16667/16667 [==============================] - 4s 258us/step - loss: 0.0942 -
acc: 0.9723
Epoch 3/3
16667/16667 [==============================] - 4s 236us/step - loss: 0.0511 -
acc: 0.9888
8333/8333 [==============================] - 2s 200us/step
Epoch 1/3
16667/16667 [==============================] - 4s 240us/step - loss: 0.1141 -
acc: 0.9618
Epoch 2/3
16667/16667 [==============================] - 4s 245us/step - loss: 0.0478 -
acc: 0.9867
Epoch 3/3
16667/16667 [==============================] - 4s 254us/step - loss: 0.0232 -
acc: 0.9957
8333/8333 [==============================] - 2s 199us/step
Epoch 1/3
16666/16666 [==============================] - 7s 397us/step - loss: 0.4025 -
acc: 0.8220
Epoch 2/3
16666/16666 [==============================] - 5s 302us/step - loss: 0.1520 -
acc: 0.9473
Epoch 3/3
16666/16666 [==============================] - 4s 265us/step - loss: 0.0669 -
acc: 0.9798
8334/8334 [==============================] - 2s 287us/step
Epoch 1/3
16667/16667 [==============================] - 5s 291us/step - loss: 0.1979 -
acc: 0.9341
Epoch 2/3
16667/16667 [==============================] - 5s 281us/step - loss: 0.0547 -
acc: 0.9849
Epoch 3/3
16667/16667 [==============================] - 5s 301us/step - loss: 0.0186 -
acc: 0.9960
8333/8333 [==============================] - ETA:  - 2s 239us/step
Epoch 1/3
16667/16667 [==============================] - 4s 268us/step - loss: 0.0529 -
acc: 0.9829
Epoch 2/3
16667/16667 [==============================] - 5s 275us/step - loss: 0.0157 -
acc: 0.9959
Epoch 3/3
```

```
16667/16667 [==============================] - 5s 303us/step - loss: 0.0050 -
acc: 0.9994
8333/8333 [==============================] - 2s 206us/step
Epoch 1/3
16666/16666 [==============================] - 7s 405us/step - loss: 0.3917 -
acc: 0.8298
Epoch 2/3
16666/16666 [==============================] - 5s 300us/step - loss: 0.1285 -
acc: 0.9566
Epoch 3/3
16666/16666 [==============================] - 5s 277us/step - loss: 0.0437 -
acc: 0.9885
8334/8334 [==============================] - 2s 253us/step
Epoch 1/3
16667/16667 [==============================] - 4s 267us/step - loss: 0.1928 -
acc: 0.9354
Epoch 2/3
16667/16667 [==============================] - 4s 268us/step - loss: 0.0467 -
acc: 0.9864
Epoch 3/3
16667/16667 [==============================] - 5s 301us/step - loss: 0.0111 -
acc: 0.9975
8333/8333 [==============================] - 1s 167us/step
Epoch 1/3
16667/16667 [==============================] - 5s 286us/step - loss: 0.0426 -
acc: 0.9852
Epoch 2/3
16667/16667 [==============================] - 5s 314us/step - loss: 0.0126 -
acc: 0.9977 0s - loss: 0.0125 -
Epoch 3/3
16667/16667 [==============================] - 5s 293us/step - loss: 0.0025 -
acc: 0.9997
8333/8333 [==============================] - 1s 176us/step
Epoch 1/3
16666/16666 [==============================] - 7s 446us/step - loss: 0.3854 -
acc: 0.8284
Epoch 2/3
16666/16666 [==============================] - 4s 269us/step - loss: 0.1142 -
acc: 0.9625
Epoch 3/3
16666/16666 [==============================] - 4s 270us/step - loss: 0.0369 -
acc: 0.9906
8334/8334 [==============================] - 2s 247us/step
Epoch 1/3
16667/16667 [==============================] - 5s 273us/step - loss: 0.1895 -
acc: 0.9350
Epoch 2/3
16667/16667 [==============================] - 5s 287us/step - loss: 0.0427 -
acc: 0.9876
Epoch 3/3
16667/16667 [==============================] - 5s 301us/step - loss: 0.0096 -
acc: 0.9982
8333/8333 [==============================] - 1s 158us/step
Epoch 1/3
16667/16667 [==============================] - 5s 327us/step - loss: 0.0393 -
acc: 0.9873
Epoch 2/3
```

```
16667/16667 [==============================] - 5s 327us/step - loss: 0.0097 -
acc: 0.9980
Epoch 3/3
16667/16667 [==============================] - 5s 317us/step - loss: 0.0020 -
acc: 0.9999
8333/8333 [==============================] - 2s 218us/step
Epoch 1/3
16666/16666 [==============================] - 7s 415us/step - loss: 0.3762 -
acc: 0.8432
Epoch 2/3
16666/16666 [==============================] - 5s 294us/step - loss: 0.1147 -
acc: 0.9599
Epoch 3/3
16666/16666 [==============================] - 5s 305us/step - loss: 0.0350 -
acc: 0.9902
8334/8334 [==============================] - 2s 268us/step
Epoch 1/3
16667/16667 [==============================] - 5s 300us/step - loss: 0.2004 -
acc: 0.9330
Epoch 2/3
16667/16667 [==============================] - 5s 295us/step - loss: 0.0420 -
acc: 0.9876
Epoch 3/3
16667/16667 [==============================] - 5s 324us/step - loss: 0.0089 -
acc: 0.9983
8333/8333 [==============================] - 2s 228us/step
Epoch 1/3
16667/16667 [==============================] - 5s 326us/step - loss: 0.0373 -
acc: 0.9863
Epoch 2/3
16667/16667 [==============================] - 5s 325us/step - loss: 0.0067 -
acc: 0.9986
Epoch 3/3
16667/16667 [==============================] - 5s 312us/step - loss: 0.0013 -
acc: 0.9999
8333/8333 [==============================] - 1s 159us/step
Epoch 1/3
16666/16666 [==============================] - 7s 417us/step - loss: 0.3677 -
acc: 0.8450
Epoch 2/3
16666/16666 [==============================] - 5s 295us/step - loss: 0.1025 -
acc: 0.9635
Epoch 3/3
16666/16666 [==============================] - 6s 335us/step - loss: 0.0273 -
acc: 0.9925
8334/8334 [==============================] - 2s 260us/step
Epoch 1/3
16667/16667 [==============================] - 5s 284us/step - loss: 0.1889 -
acc: 0.9350
Epoch 2/3
16667/16667 [==============================] - 6s 335us/step - loss: 0.0360 -
acc: 0.9901
Epoch 3/3
16667/16667 [==============================] - 5s 324us/step - loss: 0.0068 -
acc: 0.9983
8333/8333 [==============================] - 2s 203us/step
Epoch 1/3
```

```
16667/16667 [==============================] - 5s 305us/step - loss: 0.0383 -
acc: 0.9869
Epoch 2/3
16667/16667 [==============================] - 5s 330us/step - loss: 0.0085 -
acc: 0.9983
Epoch 3/3
16667/16667 [==============================] - 6s 333us/step - loss: 0.0017 -
acc: 0.9998
8333/8333 [==============================] - 2s 182us/step
Epoch 1/3
16666/16666 [==============================] - 8s 498us/step - loss: 0.3656 -
acc: 0.8422
Epoch 2/3
16666/16666 [==============================] - 5s 285us/step - loss: 0.0924 -
acc: 0.9689
Epoch 3/3
16666/16666 [==============================] - 5s 272us/step - loss: 0.0243 -
acc: 0.9939
8334/8334 [==============================] - 2s 275us/step
Epoch 1/3
16667/16667 [==============================] - 5s 275us/step - loss: 0.1868 -
acc: 0.9369
Epoch 2/3
16667/16667 [==============================] - 4s 269us/step - loss: 0.0363 -
acc: 0.9905
Epoch 3/3
16667/16667 [==============================] - 4s 266us/step - loss: 0.0061 -
acc: 0.9988
8333/8333 [==============================] - 1s 169us/step
Epoch 1/3
16667/16667 [==============================] - 4s 263us/step - loss: 0.0346 -
acc: 0.9888
Epoch 2/3
16667/16667 [==============================] - 4s 266us/step - loss: 0.0077 -
acc: 0.9987
Epoch 3/3
16667/16667 [==============================] - 4s 265us/step - loss: 0.0013 -
acc: 0.9999
8333/8333 [==============================] - 1s 175us/step
Epoch 1/3
16666/16666 [==============================] - 7s 441us/step - loss: 0.3555 -
acc: 0.8471
Epoch 2/3
16666/16666 [==============================] - 5s 279us/step - loss: 0.0886 -
acc: 0.9713
Epoch 3/3
16666/16666 [==============================] - 5s 278us/step - loss: 0.0203 -
acc: 0.9950
8334/8334 [==============================] - 2s 297us/step
Epoch 1/3
16667/16667 [==============================] - 5s 277us/step - loss: 0.2033 -
acc: 0.9318 1s - loss:
Epoch 2/3
16667/16667 [==============================] - 5s 279us/step - loss: 0.0363 -
acc: 0.9901
Epoch 3/3
16667/16667 [==============================] - 5s 278us/step - loss: 0.0055 -
```

```
acc: 0.9992
8333/8333 [==============================] - 1s 156us/step
Epoch 1/3
16667/16667 [==============================] - 5s 272us/step - loss: 0.0288 -
acc: 0.9906
Epoch 2/3
16667/16667 [==============================] - 4s 261us/step - loss: 0.0050 -
acc: 0.9987 0s - loss: 0.0049 - acc:
Epoch 3/3
16667/16667 [==============================] - 4s 222us/step - loss: 0.0011 -
acc: 0.9999
8333/8333 [==============================] - 1s 118us/step
Epoch 1/3
16666/16666 [==============================] - 5s 309us/step - loss: 0.3562 -
acc: 0.8495 0s - loss: 0.3712 - acc
Epoch 2/3
16666/16666 [==============================] - 4s 230us/step - loss: 0.0855 -
acc: 0.9736
Epoch 3/3
16666/16666 [==============================] - 4s 265us/step - loss: 0.0193 -
acc: 0.9953
8334/8334 [==============================] - 2s 222us/step
Epoch 1/3
16667/16667 [==============================] - 4s 256us/step - loss: 0.1993 -
acc: 0.9316
Epoch 2/3
16667/16667 [==============================] - 4s 237us/step - loss: 0.0347 -
acc: 0.9903
Epoch 3/3
16667/16667 [==============================] - 4s 235us/step - loss: 0.0039 -
acc: 0.9993
8333/8333 [==============================] - 1s 125us/step
Epoch 1/3
16667/16667 [==============================] - 4s 260us/step - loss: 0.0294 -
acc: 0.9904
Epoch 2/3
16667/16667 [==============================] - 4s 237us/step - loss: 0.0052 -
acc: 0.9993
Epoch 3/3
16667/16667 [==============================] - 4s 237us/step - loss: 6.2547e-
04 - acc: 1.0000 3s - loss: 8.0275e-04 - ac - ETA: 2s -
8333/8333 [==============================] - 1s 127us/step
Epoch 1/3
16666/16666 [==============================] - 6s 359us/step - loss: 0.3608 -
acc: 0.8440
Epoch 2/3
16666/16666 [==============================] - 4s 258us/step - loss: 0.0880 -
acc: 0.9710
Epoch 3/3
16666/16666 [==============================] - 4s 252us/step - loss: 0.0174 -
acc: 0.9959
8334/8334 [==============================] - 2s 209us/step
Epoch 1/3
16667/16667 [==============================] - 4s 262us/step - loss: 0.1841 -
acc: 0.9374
Epoch 2/3
16667/16667 [==============================] - 4s 249us/step - loss: 0.0273 -
```

```
acc: 0.9924
Epoch 3/3
16667/16667 [==============================] - 4s 249us/step - loss: 0.0036 -
acc: 0.9996
8333/8333 [==============================] - 1s 121us/step
Epoch 1/3
16667/16667 [==============================] - 4s 243us/step - loss: 0.0284 -
acc: 0.9906
Epoch 2/3
16667/16667 [==============================] - 4s 251us/step - loss: 0.0050 -
acc: 0.9992
Epoch 3/3
16667/16667 [==============================] - 4s 227us/step - loss: 0.0010 -
acc: 0.9999
8333/8333 [==============================] - 1s 105us/step
Epoch 1/3
16666/16666 [==============================] - 6s 353us/step - loss: 0.3571 -
acc: 0.8460
Epoch 2/3
16666/16666 [==============================] - 4s 267us/step - loss: 0.0891 -
acc: 0.9703
Epoch 3/3
16666/16666 [==============================] - 5s 328us/step - loss: 0.0174 -
acc: 0.9963
8334/8334 [==============================] - 2s 277us/step
Epoch 1/3
16667/16667 [==============================] - 5s 294us/step - loss: 0.1916 -
acc: 0.9356
Epoch 2/3
16667/16667 [==============================] - 5s 280us/step - loss: 0.0285 -
acc: 0.9930
Epoch 3/3
16667/16667 [==============================] - 4s 258us/step - loss: 0.0041 -
acc: 0.9993
8333/8333 [==============================] - 1s 125us/step
Epoch 1/3
16667/16667 [==============================] - 4s 251us/step - loss: 0.0306 -
acc: 0.9900
Epoch 2/3
16667/16667 [==============================] - 4s 259us/step - loss: 0.0061 -
acc: 0.9991
Epoch 3/3
16667/16667 [==============================] - 5s 272us/step - loss: 0.0016 -
acc: 0.9999
8333/8333 [==============================] - 1s 124us/step
Epoch 1/3
16666/16666 [==============================] - 6s 389us/step - loss: 0.3650 -
acc: 0.8420
Epoch 2/3
16666/16666 [==============================] - 4s 255us/step - loss: 0.0840 -
acc: 0.9713
Epoch 3/3
16666/16666 [==============================] - 4s 252us/step - loss: 0.0162 -
acc: 0.9963
8334/8334 [==============================] - 1s 176us/step
Epoch 1/3
16667/16667 [==============================] - 4s 246us/step - loss: 0.1858 -
```

```
acc: 0.9376
Epoch 2/3
16667/16667 [==============================] - 4s 250us/step - loss: 0.0279 -
acc: 0.9928
Epoch 3/3
16667/16667 [==============================] - 4s 248us/step - loss: 0.0037 -
acc: 0.9996
8333/8333 [==============================] - 1s 110us/step
Epoch 1/3
16667/16667 [==============================] - 4s 243us/step - loss: 0.0234 -
acc: 0.9922
Epoch 2/3
16667/16667 [==============================] - 4s 250us/step - loss: 0.0035 -
acc: 0.9993
Epoch 3/3
16667/16667 [==============================] - 4s 247us/step - loss: 4.6936e-
04 - acc: 1.0000
8333/8333 [==============================] - 1s 112us/step
Epoch 1/3
16666/16666 [==============================] - 6s 351us/step - loss: 0.3584 -
acc: 0.8507
Epoch 2/3
16666/16666 [==============================] - 4s 260us/step - loss: 0.0752 -
acc: 0.9761
Epoch 3/3
16666/16666 [==============================] - 4s 261us/step - loss: 0.0126 -
acc: 0.9972
8334/8334 [==============================] - 1s 176us/step
Epoch 1/3
16667/16667 [==============================] - 4s 263us/step - loss: 0.1928 -
acc: 0.9369
Epoch 2/3
16667/16667 [==============================] - 5s 273us/step - loss: 0.0301 -
acc: 0.9914
Epoch 3/3
16667/16667 [==============================] - 5s 270us/step - loss: 0.0037 -
acc: 0.9996
8333/8333 [==============================] - 1s 116us/step
Epoch 1/3
16667/16667 [==============================] - 4s 262us/step - loss: 0.0269 -
acc: 0.9916
Epoch 2/3
16667/16667 [==============================] - 5s 284us/step - loss: 0.0064 -
acc: 0.9986
Epoch 3/3
16667/16667 [==============================] - 5s 280us/step - loss: 0.0023 -
acc: 0.9998
8333/8333 [==============================] - 1s 119us/step
Epoch 1/3
16666/16666 [==============================] - 6s 374us/step - loss: 0.3570 -
acc: 0.8446
Epoch 2/3
16666/16666 [==============================] - 5s 277us/step - loss: 0.0753 -
acc: 0.9758
Epoch 3/3
16666/16666 [==============================] - 5s 277us/step - loss: 0.0102 -
acc: 0.9980
```

```
8334/8334 [==============================] - 2s 190us/step
Epoch 1/3
16667/16667 [==============================] - 5s 281us/step - loss: 0.1900 -
acc: 0.9363
Epoch 2/3
16667/16667 [==============================] - 5s 282us/step - loss: 0.0272 -
acc: 0.9929
Epoch 3/3
16667/16667 [==============================] - 5s 287us/step - loss: 0.0041 -
acc: 0.9993
8333/8333 [==============================] - 1s 118us/step
Epoch 1/3
16667/16667 [==============================] - 5s 285us/step - loss: 0.0258 -
acc: 0.9923
Epoch 2/3
16667/16667 [==============================] - 5s 286us/step - loss: 0.0034 -
acc: 0.9995
Epoch 3/3
16667/16667 [==============================] - 5s 282us/step - loss: 7.0690e-
04 - acc: 0.9999
8333/8333 [==============================] - 1s 117us/step
Epoch 1/3
16666/16666 [==============================] - 6s 383us/step - loss: 0.3725 -
acc: 0.8379
Epoch 2/3
16666/16666 [==============================] - 5s 285us/step - loss: 0.0819 -
acc: 0.9738
Epoch 3/3
16666/16666 [==============================] - 5s 282us/step - loss: 0.0150 -
acc: 0.9964
8334/8334 [==============================] - 1s 170us/step
Epoch 1/3
16667/16667 [==============================] - 5s 287us/step - loss: 0.1949 -
acc: 0.9350
Epoch 2/3
16667/16667 [==============================] - 5s 313us/step - loss: 0.0264 -
acc: 0.9932
Epoch 3/3
16667/16667 [==============================] - 5s 294us/step - loss: 0.0036 -
acc: 0.9996
8333/8333 [==============================] - 1s 100us/step
Epoch 1/3
16667/16667 [==============================] - 5s 294us/step - loss: 0.0237 -
acc: 0.9929
Epoch 2/3
16667/16667 [==============================] - 5s 298us/step - loss: 0.0047 -
acc: 0.9992
Epoch 3/3
16667/16667 [==============================] - 5s 291us/step - loss: 0.0014 -
acc: 0.9999 0s - loss: 0.0015 -
8333/8333 [==============================] - 1s 101us/step
Epoch 1/3
16666/16666 [==============================] - 7s 398us/step - loss: 0.3564 -
acc: 0.8509
Epoch 2/3
16666/16666 [==============================] - 5s 298us/step - loss: 0.0728 -
acc: 0.9780
```

```
Epoch 3/3
16666/16666 [==============================] - 5s 298us/step - loss: 0.0105 -
acc: 0.9977 3s - loss: 0.0130 - acc: - ETA: 2s - los - ETA: 1s -
8334/8334 [==============================] - 2s 182us/step
Epoch 1/3
16667/16667 [==============================] - 5s 318us/step - loss: 0.1929 -
acc: 0.9319
Epoch 2/3
16667/16667 [==============================] - 5s 330us/step - loss: 0.0256 -
acc: 0.9936
Epoch 3/3
16667/16667 [==============================] - 5s 306us/step - loss: 0.0025 -
acc: 0.9997
8333/8333 [==============================] - 1s 100us/step
Epoch 1/3
16667/16667 [==============================] - 5s 311us/step - loss: 0.0263 -
acc: 0.9913
Epoch 2/3
16667/16667 [==============================] - 5s 304us/step - loss: 0.0040 -
acc: 0.9990
Epoch 3/3
16667/16667 [==============================] - 5s 301us/step - loss: 5.2137e-
04 - acc: 1.0000 1s - loss: 6 - ETA: 0s - loss: 5.4008e-04 - acc:  - ETA: 0s
- loss: 5.2645e-04 - acc: 1.
8333/8333 [==============================] - 1s 103us/step
Epoch 1/3
16666/16666 [==============================] - 7s 419us/step - loss: 0.3570 -
acc: 0.8432
Epoch 2/3
16666/16666 [==============================] - 5s 323us/step - loss: 0.0659 -
acc: 0.9791
Epoch 3/3
16666/16666 [==============================] - 5s 312us/step - loss: 0.0094 -
acc: 0.9984 0s - loss: 0.0089 - acc: 0. - ETA: 0s - loss: 0.0096 - acc: 0.9
8334/8334 [==============================] - 1s 177us/step
Epoch 1/3
16667/16667 [==============================] - 5s 309us/step - loss: 0.1972 -
acc: 0.9358
Epoch 2/3
16667/16667 [==============================] - 5s 308us/step - loss: 0.0258 -
acc: 0.9926
Epoch 3/3
16667/16667 [==============================] - 5s 311us/step - loss: 0.0021 -
acc: 0.9996
8333/8333 [==============================] - 1s 110us/step
Epoch 1/3
16667/16667 [==============================] - 6s 332us/step - loss: 0.0234 -
acc: 0.9925
Epoch 2/3
16667/16667 [==============================] - 5s 317us/step - loss: 0.0073 -
acc: 0.9982
Epoch 3/3
16667/16667 [==============================] - 5s 320us/step - loss: 0.0013 -
acc: 0.9999
8333/8333 [==============================] - 1s 107us/step
Epoch 1/3
16666/16666 [==============================] - 7s 447us/step - loss: 0.3605 -
```

```
acc: 0.8429
Epoch 2/3
16666/16666 [==============================] - 5s 329us/step - loss: 0.0709 -
acc: 0.9782
Epoch 3/3
16666/16666 [==============================] - 6s 336us/step - loss: 0.0098 -
acc: 0.9979
8334/8334 [==============================] - 2s 194us/step
Epoch 1/3
16667/16667 [==============================] - 6s 335us/step - loss: 0.1951 -
acc: 0.9347
Epoch 2/3
16667/16667 [==============================] - 6s 335us/step - loss: 0.0267 -
acc: 0.9934
Epoch 3/3
16667/16667 [==============================] - 6s 335us/step - loss: 0.0050 -
acc: 0.9994
8333/8333 [==============================] - 1s 129us/step
Epoch 1/3
16667/16667 [==============================] - 6s 332us/step - loss: 0.0237 -
acc: 0.9935
Epoch 2/3
16667/16667 [==============================] - 6s 332us/step - loss: 0.0037 -
acc: 0.9995
Epoch 3/3
16667/16667 [==============================] - 6s 331us/step - loss: 0.0017 -
acc: 0.9998
8333/8333 [==============================] - 1s 122us/step
Epoch 1/3
16666/16666 [==============================] - 8s 456us/step - loss: 0.3742 -
acc: 0.8391
Epoch 2/3
16666/16666 [==============================] - 6s 345us/step - loss: 0.0818 -
acc: 0.9737
Epoch 3/3
16666/16666 [==============================] - 6s 343us/step - loss: 0.0123 -
acc: 0.9973
8334/8334 [==============================] - 2s 195us/step
Epoch 1/3
16667/16667 [==============================] - 6s 345us/step - loss: 0.1820 -
acc: 0.9398
Epoch 2/3
16667/16667 [==============================] - 6s 344us/step - loss: 0.0247 -
acc: 0.9935
Epoch 3/3
16667/16667 [==============================] - 6s 354us/step - loss: 0.0025 -
acc: 0.9998
8333/8333 [==============================] - 1s 162us/step
Epoch 1/3
16667/16667 [==============================] - 6s 358us/step - loss: 0.0228 -
acc: 0.9927
Epoch 2/3
16667/16667 [==============================] - 6s 347us/step - loss: 0.0035 -
acc: 0.9993
Epoch 3/3
16667/16667 [==============================] - 6s 350us/step - loss: 3.2763e-
```

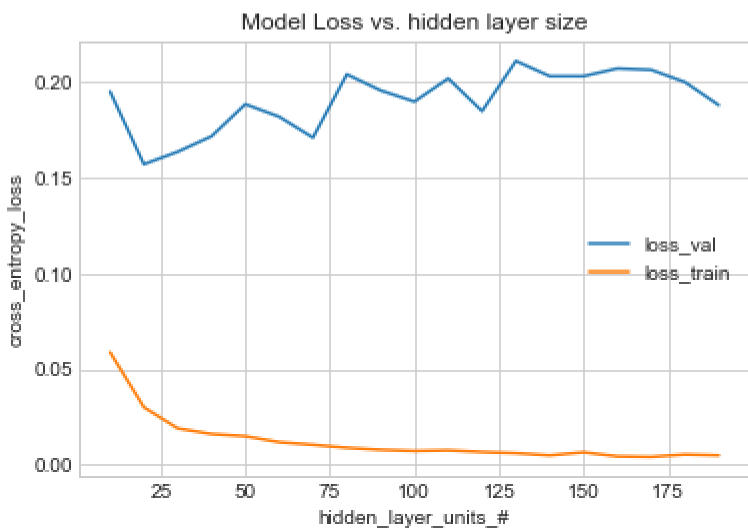
```
04 - acc: 1.0000
8333/8333 [==============================] - 1s 128us/step
```

In [37]:

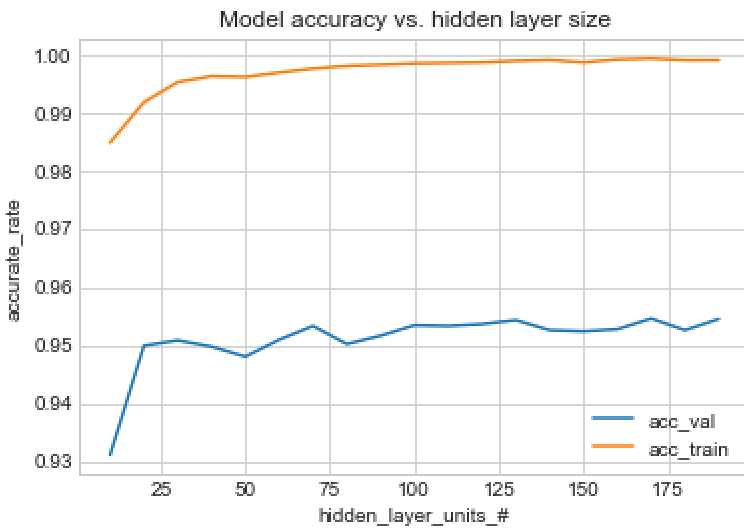
```
ix = units
loss_train_2 = pd.Series(loss_train_2, index = ix)
loss_val_2 = pd.Series(loss_val_2, index = ix)
acc_train_2 = pd.Series(acc_train_2, index = ix)
acc_val_2 = pd.Series(acc_val_2, index = ix)
```

In [38]:

```
fig = plt.figure()
ax = plt.axes()
ax.plot(loss_val_2, label = 'loss_val')
ax.plot(loss_train_2, label = 'loss_train')
ax.legend()
plt.xlabel('hidden_layer_units_#')
plt.ylabel('cross_entropy_loss')
plt.title('Model Loss vs. hidden layer size')
fig.savefig("loss_cv_2.png",dpi = 400)
```



In [39]:
```
fig = plt.figure()
ax = plt.axes()
ax.plot(acc_val_2,label = 'acc_val')
ax.plot(acc_train_2, label = 'acc_train')
ax.legend()
plt.xlabel('hidden_layer_units_#')
plt.ylabel('accurate_rate')
plt.title('Model accuracy vs. hidden layer size')
fig.savefig("acc_cv_2.png",dpi=400)
```



In [52]:
```
# baseline model with two hidden layers
model_overfitting_2 = Sequential()
model_overfitting_2.add(Dense(50,input_shape = (max_num,)))
model_overfitting_2.add(Activation('relu'))
model_overfitting_2.add(Dense(50))
model_overfitting_2.add(Activation('relu'))
model_overfitting_2.add(Dense(num_classes))
model_overfitting_2.add(Activation('softmax'))
model_overfitting_2.compile(loss = 'categorical_crossentropy', optimizer = 'ad
am', metrics = ['accuracy'])

hist_overfitting_2 = model_overfitting_2.fit(X_train,y_train, batch_size=200,
epochs = 3, verbose = 1)
score_overfitting_2 = model_overfitting_2.evaluate(X_test,y_test, batch_size=2
00, verbose = 1)

base_train_acc_2 = hist_overfitting_2.history.get('acc')[-1]
base_test_acc_2 = score_overfitting_2[1]
```

```
Epoch 1/3
25000/25000 [==============================] - 7s 270us/step - loss: 0.3421 -
acc: 0.8544
Epoch 2/3
25000/25000 [==============================] - 4s 158us/step - loss: 0.1223 -
acc: 0.9559
Epoch 3/3
25000/25000 [==============================] - 4s 159us/step - loss: 0.0409 -
acc: 0.9875
25000/25000 [==============================] - 3s 131us/step
```

In [53]:
```
print('The accurate rate of the overfitting model with two hidden layers on tr
aining dataset is')
print(base_train_acc_2)
print('The accurate rate of the overfitting model with two hidden layers on te
st dataset is')
print(base_test_acc_2)
```

```
The accurate rate of the overfitting model with two hidden layers on training
dataset is
0.987520010471344
The accurate rate of the overfitting model with two hidden layers on test dat
aset is
0.8638800020217896
```

In [54]:
```
from tensorflow.keras.layers import Dropout
rate = [0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9]
loss_train_drop = []
acc_train_drop = []
loss_val_drop = []
acc_val_drop = []
for r in rate:
    model_drop = Sequential()
    model_drop.add(Dense(50,input_shape = (max_num,)))
    model_drop.add(Activation('relu'))
    model_drop.add(Dropout(r))
    model_drop.add(Dense(50))
    model_drop.add(Activation('relu'))
    model_drop.add(Dropout(r))
    model_drop.add(Dense(num_classes))
    model_drop.add(Activation('softmax'))
    model_drop.compile(loss = 'categorical_crossentropy', optimizer = 'adam',
metrics = ['accuracy'])
    loss_val_cv = 0
    acc_val_cv = 0
    loss_train_cv = 0
    acc_train_cv = 0
    for train_cv_index, val_index in kf.split(X_train):
        X_train_cv = X_train[train_cv_index]
        y_train_cv = y_train[train_cv_index]
        X_val_cv = X_train[val_index]
        y_val_cv = y_train[val_index]
        hist = model_drop.fit(X_train_cv,y_train_cv,batch_size=200, epochs = 3
)
        loss_train_cv = loss_train_cv + hist.history.get('loss')[-1]
        acc_train_cv = acc_train_cv + hist.history.get('acc')[-1]
        score_val_cv = model_drop.evaluate(X_val_cv,y_val_cv, batch_size=200,
verbose = 1)
        loss_val_cv = loss_val_cv + score_val_cv[0]
        acc_val_cv = acc_val_cv + score_val_cv[1]
    loss_val_drop.append(loss_val_cv/3)
    acc_val_drop.append(acc_val_cv/3)
    loss_train_drop.append(loss_train_cv/3)
    acc_train_drop.append(acc_train_cv/3)
```

```
Epoch 1/3
16666/16666 [==============================] - 6s 342us/step - loss: 0.3881 -
acc: 0.8276
Epoch 2/3
16666/16666 [==============================] - 3s 163us/step - loss: 0.1257 -
acc: 0.9543
Epoch 3/3
16666/16666 [==============================] - 3s 164us/step - loss: 0.0485 -
acc: 0.9853
8334/8334 [==============================] - 2s 222us/step
Epoch 1/3
16667/16667 [==============================] - 3s 165us/step - loss: 0.1950 -
acc: 0.9328
Epoch 2/3
16667/16667 [==============================] - 3s 163us/step - loss: 0.0455 -
acc: 0.9866
Epoch 3/3
16667/16667 [==============================] - 3s 163us/step - loss: 0.0120 -
acc: 0.9972
8333/8333 [==============================] - 1s 90us/step
Epoch 1/3
16667/16667 [==============================] - 3s 165us/step - loss: 0.0411 -
acc: 0.9861
Epoch 2/3
16667/16667 [==============================] - 3s 165us/step - loss: 0.0135 -
acc: 0.9966
Epoch 3/3
16667/16667 [==============================] - 3s 164us/step - loss: 0.0051 -
acc: 0.9990
8333/8333 [==============================] - 1s 93us/step
Epoch 1/3
16666/16666 [==============================] - 6s 338us/step - loss: 0.4107 -
acc: 0.8174
Epoch 2/3
16666/16666 [==============================] - 3s 167us/step - loss: 0.1479 -
acc: 0.9449
Epoch 3/3
16666/16666 [==============================] - 3s 166us/step - loss: 0.0667 -
acc: 0.9781
8334/8334 [==============================] - 2s 236us/step
Epoch 1/3
16667/16667 [==============================] - 3s 180us/step - loss: 0.2052 -
acc: 0.9295
Epoch 2/3
16667/16667 [==============================] - 3s 173us/step - loss: 0.0701 -
acc: 0.9787
Epoch 3/3
16667/16667 [==============================] - 3s 163us/step - loss: 0.0299 -
acc: 0.9915
8333/8333 [==============================] - 1s 95us/step
Epoch 1/3
16667/16667 [==============================] - 3s 163us/step - loss: 0.0590 -
acc: 0.9803
Epoch 2/3
16667/16667 [==============================] - 3s 163us/step - loss: 0.0192 -
acc: 0.9940
Epoch 3/3
```

```
16667/16667 [==============================] - 3s 170us/step - loss: 0.0122 -
acc: 0.9962
8333/8333 [==============================] - 1s 100us/step
Epoch 1/3
16666/16666 [==============================] - 5s 330us/step - loss: 0.4484 -
acc: 0.7904
Epoch 2/3
16666/16666 [==============================] - 3s 163us/step - loss: 0.1867 -
acc: 0.9331
Epoch 3/3
16666/16666 [==============================] - 3s 163us/step - loss: 0.0990 -
acc: 0.9657
8334/8334 [==============================] - 2s 232us/step
Epoch 1/3
16667/16667 [==============================] - 3s 165us/step - loss: 0.2042 -
acc: 0.9294
Epoch 2/3
16667/16667 [==============================] - 3s 164us/step - loss: 0.0825 -
acc: 0.9728
Epoch 3/3
16667/16667 [==============================] - 3s 167us/step - loss: 0.0396 -
acc: 0.9877
8333/8333 [==============================] - 1s 95us/step
Epoch 1/3
16667/16667 [==============================] - 3s 166us/step - loss: 0.0699 -
acc: 0.9758
Epoch 2/3
16667/16667 [==============================] - 3s 176us/step - loss: 0.0340 -
acc: 0.9887
Epoch 3/3
16667/16667 [==============================] - 3s 177us/step - loss: 0.0258 -
acc: 0.9914
8333/8333 [==============================] - 1s 98us/step
Epoch 1/3
16666/16666 [==============================] - 6s 334us/step - loss: 0.5186 -
acc: 0.7434
Epoch 2/3
16666/16666 [==============================] - 3s 164us/step - loss: 0.2548 -
acc: 0.9022
Epoch 3/3
16666/16666 [==============================] - 3s 165us/step - loss: 0.1512 -
acc: 0.9467
8334/8334 [==============================] - 2s 231us/step
Epoch 1/3
16667/16667 [==============================] - 3s 165us/step - loss: 0.2253 -
acc: 0.9206
Epoch 2/3
16667/16667 [==============================] - 3s 167us/step - loss: 0.1334 -
acc: 0.9532
Epoch 3/3
16667/16667 [==============================] - 3s 163us/step - loss: 0.0824 -
acc: 0.9713 2s - loss: 0.0777 - ETA: 1s - loss
8333/8333 [==============================] - 1s 93us/step
Epoch 1/3
16667/16667 [==============================] - 3s 166us/step - loss: 0.1143 -
acc: 0.9585
Epoch 2/3
```

```
16667/16667 [==============================] - 3s 164us/step - loss: 0.0768 -
acc: 0.9717
Epoch 3/3
16667/16667 [==============================] - 3s 169us/step - loss: 0.0584 -
acc: 0.9770
8333/8333 [==============================] - 1s 101us/step
Epoch 1/3
16666/16666 [==============================] - 6s 338us/step - loss: 0.5539 -
acc: 0.7104
Epoch 2/3
16666/16666 [==============================] - 3s 165us/step - loss: 0.2923 -
acc: 0.8876
Epoch 3/3
16666/16666 [==============================] - 3s 163us/step - loss: 0.2031 -
acc: 0.9292
8334/8334 [==============================] - 2s 234us/step
Epoch 1/3
16667/16667 [==============================] - 3s 169us/step - loss: 0.2449 -
acc: 0.9124
Epoch 2/3
16667/16667 [==============================] - 3s 164us/step - loss: 0.1635 -
acc: 0.9419
Epoch 3/3
16667/16667 [==============================] - 3s 164us/step - loss: 0.1282 -
acc: 0.9549
8333/8333 [==============================] - 1s 95us/step
Epoch 1/3
16667/16667 [==============================] - 3s 166us/step - loss: 0.1519 -
acc: 0.9444
Epoch 2/3
16667/16667 [==============================] - 3s 169us/step - loss: 0.1098 -
acc: 0.9603
Epoch 3/3
16667/16667 [==============================] - 3s 170us/step - loss: 0.0814 -
acc: 0.9691
8333/8333 [==============================] - 1s 99us/step
Epoch 1/3
16666/16666 [==============================] - 6s 340us/step - loss: 0.6292 -
acc: 0.6495
Epoch 2/3
16666/16666 [==============================] - 3s 162us/step - loss: 0.3972 -
acc: 0.8274
Epoch 3/3
16666/16666 [==============================] - 3s 163us/step - loss: 0.2877 -
acc: 0.8838
8334/8334 [==============================] - 2s 237us/step
Epoch 1/3
16667/16667 [==============================] - 3s 166us/step - loss: 0.2923 -
acc: 0.8877
Epoch 2/3
16667/16667 [==============================] - 3s 164us/step - loss: 0.2353 -
acc: 0.9125
Epoch 3/3
16667/16667 [==============================] - 3s 165us/step - loss: 0.1884 -
acc: 0.9283
8333/8333 [==============================] - 1s 92us/step
Epoch 1/3
```

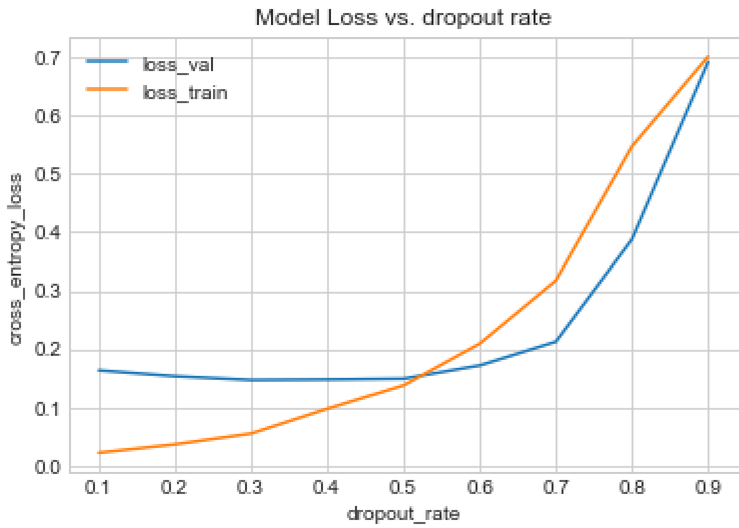
```
16667/16667 [==============================] - 3s 174us/step - loss: 0.2161 -
acc: 0.9180
Epoch 2/3
16667/16667 [==============================] - 3s 169us/step - loss: 0.1821 -
acc: 0.9317
Epoch 3/3
16667/16667 [==============================] - 3s 173us/step - loss: 0.1515 -
acc: 0.9402
8333/8333 [==============================] - 1s 100us/step
Epoch 1/3
16666/16666 [==============================] - 6s 345us/step - loss: 0.7058 -
acc: 0.5620
Epoch 2/3
16666/16666 [==============================] - 3s 184us/step - loss: 0.5380 -
acc: 0.7335
Epoch 3/3
16666/16666 [==============================] - 3s 176us/step - loss: 0.4289 -
acc: 0.8109
8334/8334 [==============================] - 2s 253us/step
Epoch 1/3
16667/16667 [==============================] - 3s 165us/step - loss: 0.3966 -
acc: 0.8354
Epoch 2/3
16667/16667 [==============================] - 3s 164us/step - loss: 0.3399 -
acc: 0.8655
Epoch 3/3
16667/16667 [==============================] - 3s 168us/step - loss: 0.2849 -
acc: 0.8848
8333/8333 [==============================] - 1s 98us/step
Epoch 1/3
16667/16667 [==============================] - 3s 166us/step - loss: 0.2971 -
acc: 0.8806
Epoch 2/3
16667/16667 [==============================] - 3s 164us/step - loss: 0.2690 -
acc: 0.8962
Epoch 3/3
16667/16667 [==============================] - 3s 183us/step - loss: 0.2382 -
acc: 0.9065
8333/8333 [==============================] - 1s 101us/step
Epoch 1/3
16666/16666 [==============================] - 6s 349us/step - loss: 0.7501 -
acc: 0.4933
Epoch 2/3
16666/16666 [==============================] - 3s 163us/step - loss: 0.6910 -
acc: 0.5371
Epoch 3/3
16666/16666 [==============================] - 3s 164us/step - loss: 0.6498 -
acc: 0.6076
8334/8334 [==============================] - 2s 250us/step
Epoch 1/3
16667/16667 [==============================] - 3s 165us/step - loss: 0.6137 -
acc: 0.6412
Epoch 2/3
16667/16667 [==============================] - 3s 168us/step - loss: 0.5689 -
acc: 0.6894
Epoch 3/3
16667/16667 [==============================] - 3s 174us/step - loss: 0.5337 -
```

```
acc: 0.7151
8333/8333 [==============================] - 1s 99us/step
Epoch 1/3
16667/16667 [==============================] - 3s 167us/step - loss: 0.5102 -
acc: 0.7403
Epoch 2/3
16667/16667 [==============================] - 3s 169us/step - loss: 0.4880 -
acc: 0.7601
Epoch 3/3
16667/16667 [==============================] - 3s 168us/step - loss: 0.4587 -
acc: 0.7805
8333/8333 [==============================] - 1s 98us/step
Epoch 1/3
16666/16666 [==============================] - 6s 350us/step - loss: 0.8428 -
acc: 0.4963
Epoch 2/3
16666/16666 [==============================] - 3s 163us/step - loss: 0.7036 -
acc: 0.5007
Epoch 3/3
16666/16666 [==============================] - 3s 166us/step - loss: 0.7065 -
acc: 0.5006
8334/8334 [==============================] - 2s 241us/step
Epoch 1/3
16667/16667 [==============================] - 3s 164us/step - loss: 0.6975 -
acc: 0.4968
Epoch 2/3
16667/16667 [==============================] - 3s 163us/step - loss: 0.6984 -
acc: 0.4940
Epoch 3/3
16667/16667 [==============================] - 3s 162us/step - loss: 0.7001 -
acc: 0.4988
8333/8333 [==============================] - 1s 89us/step
Epoch 1/3
16667/16667 [==============================] - 3s 164us/step - loss: 0.6972 -
acc: 0.4994
Epoch 2/3
16667/16667 [==============================] - 3s 165us/step - loss: 0.6988 -
acc: 0.5005
Epoch 3/3
16667/16667 [==============================] - 3s 164us/step - loss: 0.6974 -
acc: 0.5032
8333/8333 [==============================] - 1s 100us/step
```

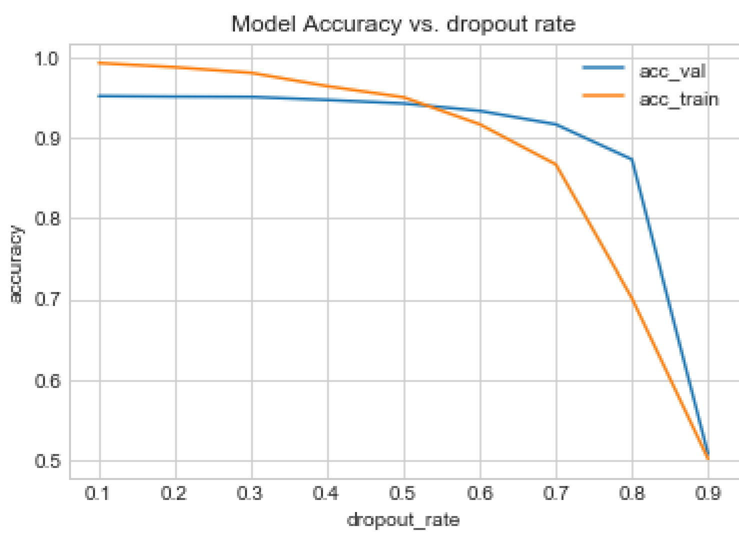
In [55]:

```
ix = rate
loss_train_drop= pd.Series(loss_train_drop, index = ix)
loss_val_drop = pd.Series(loss_val_drop, index = ix)
acc_train_drop = pd.Series(acc_train_drop, index = ix)
acc_val_drop = pd.Series(acc_val_drop, index = ix)
```

In [56]:
```
fig = plt.figure()
ax = plt.axes()
ax.plot(loss_val_drop, label = 'loss_val')
ax.plot(loss_train_drop, label = 'loss_train')
ax.legend()
plt.xlabel('dropout_rate')
plt.ylabel('cross_entropy_loss')
plt.title('Model Loss vs. dropout rate')
fig.savefig("drop_loss_cv_2.png",dpi = 400)
```



In [57]:
```
fig = plt.figure()
ax = plt.axes()
ax.plot(acc_val_drop, label = 'acc_val')
ax.plot(acc_train_drop, label = 'acc_train')
ax.legend()
plt.xlabel('dropout_rate')
plt.ylabel('accuracy')
plt.title('Model Accuracy vs. dropout rate')
fig.savefig("drop_acc_cv_2.png",dpi = 400)
```



In [60]:
```
model_drop_2 = Sequential()
model_drop_2.add(Dense(50,input_shape = (max_num,)))
model_drop_2.add(Activation('relu'))
model_drop_2.add(Dropout(0.5))
model_drop_2.add(Dense(50))
model_drop_2.add(Activation('relu'))
model_drop_2.add(Dropout(0.5))
model_drop_2.add(Dense(num_classes))
model_drop_2.add(Activation('softmax'))
model_drop_2.compile(loss = 'categorical_crossentropy', optimizer = 'adam', me
trics = ['accuracy'])

hist_drop_2 = model_drop_2.fit(X_train,y_train, batch_size=200, epochs = 3, ve
rbose = 1)
score_drop_2 = model_drop_2.evaluate(X_test,y_test, batch_size=200, verbose =
1)

drop_train_acc_2 = hist_drop_2.history.get('acc')[-1]
drop_test_acc_2 = score_drop_2[1]
```

```
Epoch 1/3
25000/25000 [==============================] - 8s 304us/step - loss: 0.4674 -
acc: 0.7780
Epoch 2/3
25000/25000 [==============================] - 4s 172us/step - loss: 0.2513 -
acc: 0.9074
Epoch 3/3
25000/25000 [==============================] - 4s 173us/step - loss: 0.1687 -
acc: 0.9395
25000/25000 [==============================] - 4s 155us/step
```

In [61]:
```
print('The accurate rate of the model with two hidden layers and dropout(0.5)
 on training dataset is')
print(drop_train_acc_2)
print('The accurate rate of the model with two hidden layers and dropout(0.5)
 on test dataset is')
print(drop_test_acc_2)
```

```
The accurate rate of the model with two hidden layers and dropout(0.5) on tra
ining dataset is
0.939479998588562
The accurate rate of the model with two hidden layers and dropout(0.5) on tes
t dataset is
0.877079999923706
```

In [84]:
```
model_nn25_2 = Sequential()
model_nn25_2.add(Dense(25,input_shape = (max_num,)))
model_nn25_2.add(Activation('relu'))
model_nn25_2.add(Dense(25))
model_nn25_2.add(Activation('relu'))
model_nn25_2.add(Dense(num_classes))
model_nn25_2.add(Activation('softmax'))
model_nn25_2.compile(loss = 'categorical_crossentropy', optimizer = 'adam', me
trics = ['accuracy'])

hist_nn25_2 = model_nn25_2.fit(X_train,y_train, batch_size=200, epochs = 3, ve
rbose = 1)
score_nn25_2 = model_nn25_2.evaluate(X_test,y_test, batch_size=200, verbose =
1)

nn25_train_acc_2 = hist_nn25_2.history.get('acc')[-1]
nn25_test_acc_2 = score_nn25_2[1]
```

```
Epoch 1/3
25000/25000 [==============================] - 22s 869us/step - loss: 0.3517
- acc: 0.8536
Epoch 2/3
25000/25000 [==============================] - 6s 249us/step - loss: 0.1441 -
acc: 0.9484
Epoch 3/3
25000/25000 [==============================] - 6s 232us/step - loss: 0.0743 -
acc: 0.9756
25000/25000 [==============================] - 12s 487us/step
```

In [85]:
```
print('The accurate rate of the model with two hidden layers(25units) on train
ing dataset is')
print(drop_train_acc_2)
print('The accurate rate of the model with two hidden layers(25units) on test
 dataset is')
print(drop_test_acc_2)
```

```
The accurate rate of the model with two hidden layers(25units) on training da
taset is
0.9673600010871887
The accurate rate of the model with two hidden layers(25units) on test datase
t is
0.8725199995040893
```

In [62]:
```
model_drop_2 = Sequential()
model_drop_2.add(Dense(50,input_shape = (max_num,)))
model_drop_2.add(Activation('relu'))
model_drop_2.add(Dropout(0.4))
model_drop_2.add(Dense(50))
model_drop_2.add(Activation('relu'))
model_drop_2.add(Dropout(0.4))
model_drop_2.add(Dense(num_classes))
model_drop_2.add(Activation('softmax'))
model_drop_2.compile(loss = 'categorical_crossentropy', optimizer = 'adam', me
trics = ['accuracy'])

hist_drop_2 = model_drop_2.fit(X_train,y_train, batch_size=200, epochs = 3, ve
rbose = 1)
score_drop_2 = model_drop_2.evaluate(X_test,y_test, batch_size=200, verbose =
1)

drop_train_acc_2 = hist_drop_2.history.get('acc')[-1]
drop_test_acc_2 = score_drop_2[1]
```

```
Epoch 1/3
25000/25000 [==============================] - 8s 309us/step - loss: 0.4405 -
acc: 0.7981
Epoch 2/3
25000/25000 [==============================] - 5s 182us/step - loss: 0.2181 -
acc: 0.9203
Epoch 3/3
25000/25000 [==============================] - 4s 170us/step - loss: 0.1437 -
acc: 0.9492
25000/25000 [==============================] - 4s 150us/step
```

In [63]:
```
print('The accurate rate of the model with two hidden layers and dropout(0.4)
 on training dataset is')
print(drop_train_acc_2)
print('The accurate rate of the model with two hidden layers and dropout(0.4)
 on test dataset is')
print(drop_test_acc_2)
```

```
The accurate rate of the model with two hidden layers and dropout(0.4) on tra
ining dataset is
0.9491999969482422
The accurate rate of the model with two hidden layers and dropout(0.4) on tes
t dataset is
0.8730800008773804
```

In [64]:
```
model_drop_2 = Sequential()
model_drop_2.add(Dense(50,input_shape = (max_num,)))
model_drop_2.add(Activation('relu'))
model_drop_2.add(Dropout(0.3))
model_drop_2.add(Dense(50))
model_drop_2.add(Activation('relu'))
model_drop_2.add(Dropout(0.3))
model_drop_2.add(Dense(num_classes))
model_drop_2.add(Activation('softmax'))
model_drop_2.compile(loss = 'categorical_crossentropy', optimizer = 'adam', me
trics = ['accuracy'])

hist_drop_2 = model_drop_2.fit(X_train,y_train, batch_size=200, epochs = 3, ve
rbose = 1)
score_drop_2 = model_drop_2.evaluate(X_test,y_test, batch_size=200, verbose =
1)

drop_train_acc_2 = hist_drop_2.history.get('acc')[-1]
drop_test_acc_2 = score_drop_2[1]
```

```
Epoch 1/3
25000/25000 [==============================] - 8s 308us/step - loss: 0.4090 -
acc: 0.8146
Epoch 2/3
25000/25000 [==============================] - 4s 166us/step - loss: 0.1834 -
acc: 0.9328
Epoch 3/3
25000/25000 [==============================] - 4s 167us/step - loss: 0.0976 -
acc: 0.9674
25000/25000 [==============================] - 4s 148us/step
```

In [65]:
```
print('The accurate rate of the model with two hidden layers and dropout(0.3)
 on training dataset is')
print(drop_train_acc_2)
print('The accurate rate of the model with two hidden layers and dropout(0.3)
 on test dataset is')
print(drop_test_acc_2)
```

```
The accurate rate of the model with two hidden layers and dropout(0.3) on tra
ining dataset is
0.9673600010871887
The accurate rate of the model with two hidden layers and dropout(0.3) on tes
t dataset is
0.8725199995040893
```

In [67]:
```
lambda_list = [1e-6,1e-5,1e-4,1e-3,1e-2,1e-1]
loss_train_2_l1 = []
acc_train_2_l1 = []
loss_val_2_l1 = []
acc_val_2_l1 = []
for par in lambda_list:
    model_2_l1 = Sequential()
    model_2_l1.add(Dense(50,input_shape = (max_num,), kernel_regularizer = l1(
par), bias_regularizer=l1(par)))
    model_2_l1.add(Activation('relu'))
    model_2_l1.add(Dense(50, kernel_regularizer = l1(par), bias_regularizer=l1
(par)))
    model_2_l1.add(Activation('relu'))
    model_2_l1.add(Dense(num_classes, kernel_regularizer = l1(par), bias_regul
arizer=l1(par)))
    model_2_l1.add(Activation('softmax'))
    model_2_l1.compile(loss = 'categorical_crossentropy', optimizer = 'adam',
metrics = ['accuracy'])
    loss_val_cv = 0
    acc_val_cv = 0
    loss_train_cv = 0
    acc_train_cv = 0
    for train_cv_index, val_index in kf.split(X_train):
        X_train_cv = X_train[train_cv_index]
        y_train_cv = y_train[train_cv_index]
        X_val_cv = X_train[val_index]
        y_val_cv = y_train[val_index]
        hist = model_2_l1.fit(X_train_cv,y_train_cv,batch_size=200, epochs = 3
)
        loss_train_cv = loss_train_cv + hist.history.get('loss')[-1]
        acc_train_cv = acc_train_cv + hist.history.get('acc')[-1]
        score_val_cv = model_2_l1.evaluate(X_val_cv,y_val_cv, batch_size=200,
verbose = 1)
        loss_val_cv = loss_val_cv + score_val_cv[0]
        acc_val_cv = acc_val_cv + score_val_cv[1]
    loss_val_2_l1.append(loss_val_cv/3)
    acc_val_2_l1.append(acc_val_cv/3)
    loss_train_2_l1.append(loss_train_cv/3)
    acc_train_2_l1.append(acc_train_cv/3)
```

```
Epoch 1/3
16666/16666 [==============================] - 7s 402us/step - loss: 0.3948 -
acc: 0.8295
Epoch 2/3
16666/16666 [==============================] - 3s 184us/step - loss: 0.1339 -
acc: 0.9548
Epoch 3/3
16666/16666 [==============================] - 3s 191us/step - loss: 0.0534 -
acc: 0.9863
8334/8334 [==============================] - 3s 311us/step
Epoch 1/3
16667/16667 [==============================] - 3s 189us/step - loss: 0.2059 -
acc: 0.9323
Epoch 2/3
16667/16667 [==============================] - 3s 184us/step - loss: 0.0505 -
acc: 0.9873
Epoch 3/3
16667/16667 [==============================] - 3s 182us/step - loss: 0.0162 -
acc: 0.9983
8333/8333 [==============================] - 1s 102us/step
Epoch 1/3
16667/16667 [==============================] - 3s 176us/step - loss: 0.0477 -
acc: 0.9878
Epoch 2/3
16667/16667 [==============================] - 3s 173us/step - loss: 0.0177 -
acc: 0.9980
Epoch 3/3
16667/16667 [==============================] - 3s 174us/step - loss: 0.0117 -
acc: 0.9996
8333/8333 [==============================] - 1s 99us/step
Epoch 1/3
16666/16666 [==============================] - 6s 379us/step - loss: 0.4162 -
acc: 0.8459
Epoch 2/3
16666/16666 [==============================] - 3s 173us/step - loss: 0.1735 -
acc: 0.9605
Epoch 3/3
16666/16666 [==============================] - 3s 173us/step - loss: 0.1021 -
acc: 0.9875
8334/8334 [==============================] - 2s 268us/step
Epoch 1/3
16667/16667 [==============================] - 3s 174us/step - loss: 0.2741 -
acc: 0.9276
Epoch 2/3
16667/16667 [==============================] - 3s 176us/step - loss: 0.1165 -
acc: 0.9852
Epoch 3/3
16667/16667 [==============================] - 3s 176us/step - loss: 0.0776 -
acc: 0.9974
8333/8333 [==============================] - 1s 104us/step
Epoch 1/3
16667/16667 [==============================] - 3s 179us/step - loss: 0.1257 -
acc: 0.9779
Epoch 2/3
16667/16667 [==============================] - 3s 196us/step - loss: 0.0900 -
acc: 0.9947
Epoch 3/3
```

```
16667/16667 [==============================] - 3s 193us/step - loss: 0.0729 -
acc: 0.9994
8333/8333 [==============================] - 1s 104us/step
Epoch 1/3
16666/16666 [==============================] - 7s 393us/step - loss: 0.8180 -
acc: 0.8366
Epoch 2/3
16666/16666 [==============================] - 3s 178us/step - loss: 0.4571 -
acc: 0.9492
Epoch 3/3
16666/16666 [==============================] - 3s 176us/step - loss: 0.3510 -
acc: 0.9708
8334/8334 [==============================] - 2s 287us/step
Epoch 1/3
16667/16667 [==============================] - 3s 176us/step - loss: 0.4823 -
acc: 0.9266
Epoch 2/3
16667/16667 [==============================] - 3s 174us/step - loss: 0.3380 -
acc: 0.9761
Epoch 3/3
16667/16667 [==============================] - 3s 177us/step - loss: 0.2568 -
acc: 0.9903
8333/8333 [==============================] - 1s 100us/step
Epoch 1/3
16667/16667 [==============================] - 3s 176us/step - loss: 0.4007 -
acc: 0.9435
Epoch 2/3
16667/16667 [==============================] - 3s 174us/step - loss: 0.3340 -
acc: 0.9817
Epoch 3/3
16667/16667 [==============================] - 3s 174us/step - loss: 0.2488 -
acc: 0.9956
8333/8333 [==============================] - 1s 101us/step
Epoch 1/3
16666/16666 [==============================] - 6s 390us/step - loss: 2.3336 -
acc: 0.8399
Epoch 2/3
16666/16666 [==============================] - 3s 175us/step - loss: 0.9719 -
acc: 0.8942
Epoch 3/3
16666/16666 [==============================] - 3s 180us/step - loss: 0.8172 -
acc: 0.9024
8334/8334 [==============================] - 2s 284us/step
Epoch 1/3
16667/16667 [==============================] - 3s 175us/step - loss: 0.7263 -
acc: 0.9030
Epoch 2/3
16667/16667 [==============================] - 3s 174us/step - loss: 0.5863 -
acc: 0.9260
Epoch 3/3
16667/16667 [==============================] - 3s 193us/step - loss: 0.5042 -
acc: 0.9384
8333/8333 [==============================] - 1s 124us/step
Epoch 1/3
16667/16667 [==============================] - 3s 189us/step - loss: 0.5779 -
acc: 0.9121
Epoch 2/3
```

```
16667/16667 [==============================] - 3s 182us/step - loss: 0.4615 -
acc: 0.9431
Epoch 3/3
16667/16667 [==============================] - 3s 178us/step - loss: 0.4252 -
acc: 0.9490
8333/8333 [==============================] - 1s 103us/step
Epoch 1/3
16666/16666 [==============================] - 7s 390us/step - loss: 11.5090
- acc: 0.7734
Epoch 2/3
16666/16666 [==============================] - 3s 176us/step - loss: 2.2685 -
acc: 0.8452
Epoch 3/3
16666/16666 [==============================] - 3s 174us/step - loss: 1.4654 -
acc: 0.8604
8334/8334 [==============================] - 2s 282us/step
Epoch 1/3
16667/16667 [==============================] - 3s 180us/step - loss: 1.2861 -
acc: 0.8702
Epoch 2/3
16667/16667 [==============================] - 3s 175us/step - loss: 1.2563 -
acc: 0.8748
Epoch 3/3
16667/16667 [==============================] - 3s 173us/step - loss: 1.2545 -
acc: 0.8780
8333/8333 [==============================] - 1s 98us/step
Epoch 1/3
16667/16667 [==============================] - 3s 176us/step - loss: 1.2579 -
acc: 0.8733
Epoch 2/3
16667/16667 [==============================] - 3s 175us/step - loss: 1.2486 -
acc: 0.8759
Epoch 3/3
16667/16667 [==============================] - 3s 179us/step - loss: 1.2477 -
acc: 0.8806
8333/8333 [==============================] - 1s 103us/step
Epoch 1/3
16666/16666 [==============================] - 7s 398us/step - loss: 106.2643
- acc: 0.5230
Epoch 2/3
16666/16666 [==============================] - 3s 173us/step - loss: 14.8354
- acc: 0.5011
Epoch 3/3
16666/16666 [==============================] - 3s 176us/step - loss: 7.6500 -
acc: 0.5087
8334/8334 [==============================] - 2s 283us/step
Epoch 1/3
16667/16667 [==============================] - 3s 175us/step - loss: 6.4992 -
acc: 0.4978
Epoch 2/3
16667/16667 [==============================] - 3s 175us/step - loss: 6.4632 -
acc: 0.4998
Epoch 3/3
16667/16667 [==============================] - 3s 174us/step - loss: 6.4630 -
acc: 0.5037
8333/8333 [==============================] - 1s 95us/step
Epoch 1/3
```

```
16667/16667 [==============================] - 3s 181us/step - loss: 6.4624 -
acc: 0.4986
Epoch 2/3
16667/16667 [==============================] - 3s 175us/step - loss: 6.4635 -
acc: 0.4963
Epoch 3/3
16667/16667 [==============================] - 3s 179us/step - loss: 6.4633 -
acc: 0.4979
8333/8333 [==============================] - 1s 103us/step
```

In [68]:

```
l1_2_reg_cv = pd.DataFrame({'loss_train': loss_train_2_l1,
                          'loss_val': loss_val_2_l1,
                       'acc_train': acc_train_2_l1,
                        'acc_val':acc_val_2_l1,
                         'lambda':lambda_list}).set_index('lambda')
l1_2_reg_cv
```

Out[68]:

lambda	loss_train	loss_val	acc_train	acc_val
0.000001	0.027111	0.197342	0.994740	0.946803
0.000010	0.084224	0.267125	0.994780	0.942803
0.000100	0.285536	0.462369	0.985540	0.918402
0.001000	0.582214	0.661997	0.929919	0.886720
0.010000	1.322553	1.279183	0.873020	0.869440
0.100000	6.858770	6.491953	0.503460	0.500000

In [69]:
```
# indicating by the result of l1_reg_cv data frame, will set the lambda as 1e-
5 for l1 norm regularization
model_2_l1 = Sequential()
model_2_l1.add(Dense(50,input_shape = (max_num,), kernel_regularizer = l1(1e-5
), bias_regularizer=l1(1e-5)))
model_2_l1.add(Activation('relu'))
model_2_l1.add(Dense(50, kernel_regularizer = l1(1e-5), bias_regularizer=l1(1e
-5)))
model_2_l1.add(Activation('relu'))
model_2_l1.add(Dense(num_classes, kernel_regularizer = l1(1e-5), bias_regulari
zer=l1(1e-5)))
model_2_l1.add(Activation('softmax'))
model_2_l1.compile(loss = 'categorical_crossentropy', optimizer = 'adam', metr
ics = ['accuracy'])

hist_2_l1 = model_2_l1.fit(X_train,y_train, batch_size=200, epochs = 3, verbos
e = 1)
score_2_l1 = model_2_l1.evaluate(X_test,y_test, batch_size=200, verbose = 1)

l1_train_acc_2 = hist_2_l1.history.get('acc')[-1]
l1_test_acc_2 = score_2_l1[1]
```

```
Epoch 1/3
25000/25000 [==============================] - 9s 351us/step - loss: 0.3950 -
acc: 0.8576
Epoch 2/3
25000/25000 [==============================] - 4s 177us/step - loss: 0.1798 -
acc: 0.9588
Epoch 3/3
25000/25000 [==============================] - 4s 172us/step - loss: 0.1058 -
acc: 0.9873
25000/25000 [==============================] - 4s 163us/step
```

In [70]:
```
print('The accurate rate of the model with two hidden layers and l1 regulariza
tion on training dataset is')
print(l1_train_acc_2)
print('The accurate rate of the model with two hidden layers and l1 regulariza
tion on test dataset is')
print(l1_test_acc_2)
```

```
The accurate rate of the model with two hidden layers and l1 regularization o
n training dataset is
0.987320011138916
The accurate rate of the model with two hidden layers and l1 regularization o
n test dataset is
0.865840000629425
```

In [71]:
```
# indicating by the result of l1_reg_cv data frame, will set the lambda as 1e-
6 for l1 norm regularization
model_2_l1 = Sequential()
model_2_l1.add(Dense(50,input_shape = (max_num,), kernel_regularizer = l1(1e-6
), bias_regularizer=l1(1e-6)))
model_2_l1.add(Activation('relu'))
model_2_l1.add(Dense(50, kernel_regularizer = l1(1e-6), bias_regularizer=l1(1e
-6)))
model_2_l1.add(Activation('relu'))
model_2_l1.add(Dense(num_classes, kernel_regularizer = l1(1e-6), bias_regulari
zer=l1(1e-6)))
model_2_l1.add(Activation('softmax'))
model_2_l1.compile(loss = 'categorical_crossentropy', optimizer = 'adam', metr
ics = ['accuracy'])

hist_2_l1 = model_2_l1.fit(X_train,y_train, batch_size=200, epochs = 3, verbos
e = 1)
score_2_l1 = model_2_l1.evaluate(X_test,y_test, batch_size=200, verbose = 1)

l1_train_acc_2 = hist_2_l1.history.get('acc')[-1]
l1_test_acc_2 = score_2_l1[1]
```

```
Epoch 1/3
25000/25000 [==============================] - 8s 338us/step - loss: 0.3470 -
acc: 0.8552
Epoch 2/3
25000/25000 [==============================] - 4s 173us/step - loss: 0.1427 -
acc: 0.9525
Epoch 3/3
25000/25000 [==============================] - 4s 175us/step - loss: 0.0596 -
acc: 0.9826
25000/25000 [==============================] - 4s 161us/step
```

In [72]:
```
print('The accurate rate of the model with two hidden layers and l1 regulariza
tion on training dataset is')
print(l1_train_acc_2)
print('The accurate rate of the model with two hidden layers and l1 regulariza
tion on test dataset is')
print(l1_test_acc_2)
```

```
The accurate rate of the model with two hidden layers and l1 regularization o
n training dataset is
0.9826000151634217
The accurate rate of the model with two hidden layers and l1 regularization o
n test dataset is
0.8631199975013732
```

In [75]:
```
lambda_list = [1e-6,1e-5,1e-4,1e-3,1e-2,1e-1]
loss_train_2_l2 = []
acc_train_2_l2 = []
loss_val_2_l2 = []
acc_val_2_l2 = []
for par in lambda_list:
    model_2_l2 = Sequential()
    model_2_l2.add(Dense(50,input_shape = (max_num,), kernel_regularizer = l2(
par), bias_regularizer=l2(par)))
    model_2_l2.add(Activation('relu'))
    model_2_l2.add(Dense(50, kernel_regularizer = l2(par), bias_regularizer=l2
(par)))
    model_2_l2.add(Activation('relu'))
    model_2_l2.add(Dense(num_classes, kernel_regularizer = l2(par), bias_regul
arizer=l2(par)))
    model_2_l2.add(Activation('softmax'))
    model_2_l2.compile(loss = 'categorical_crossentropy', optimizer = 'adam',
metrics = ['accuracy'])
    loss_val_cv = 0
    acc_val_cv = 0
    loss_train_cv = 0
    acc_train_cv = 0
    for train_cv_index, val_index in kf.split(X_train):
        X_train_cv = X_train[train_cv_index]
        y_train_cv = y_train[train_cv_index]
        X_val_cv = X_train[val_index]
        y_val_cv = y_train[val_index]
        hist = model_2_l2.fit(X_train_cv,y_train_cv,batch_size=200, epochs = 3
)
        loss_train_cv = loss_train_cv + hist.history.get('loss')[-1]
        acc_train_cv = acc_train_cv + hist.history.get('acc')[-1]
        score_val_cv = model_2_l2.evaluate(X_val_cv,y_val_cv, batch_size=200,
verbose = 1)
        loss_val_cv = loss_val_cv + score_val_cv[0]
        acc_val_cv = acc_val_cv + score_val_cv[1]
    loss_val_2_l2.append(loss_val_cv/3)
    acc_val_2_l2.append(acc_val_cv/3)
    loss_train_2_l2.append(loss_train_cv/3)
    acc_train_2_l2.append(acc_train_cv/3)
```

```
Epoch 1/3
16666/16666 [==============================] - 7s 414us/step - loss: 0.3684 -
acc: 0.8426
Epoch 2/3
16666/16666 [==============================] - 3s 172us/step - loss: 0.1094 -
acc: 0.9639
Epoch 3/3
16666/16666 [==============================] - 3s 172us/step - loss: 0.0346 -
acc: 0.9913
8334/8334 [==============================] - 2s 298us/step
Epoch 1/3
16667/16667 [==============================] - 3s 176us/step - loss: 0.1944 -
acc: 0.9338
Epoch 2/3
16667/16667 [==============================] - 3s 175us/step - loss: 0.0430 -
acc: 0.9873
Epoch 3/3
16667/16667 [==============================] - 3s 173us/step - loss: 0.0093 -
acc: 0.9977
8333/8333 [==============================] - 1s 100us/step
Epoch 1/3
16667/16667 [==============================] - 3s 175us/step - loss: 0.0411 -
acc: 0.9859
Epoch 2/3
16667/16667 [==============================] - 3s 174us/step - loss: 0.0106 -
acc: 0.9978
Epoch 3/3
16667/16667 [==============================] - 3s 175us/step - loss: 0.0028 -
acc: 0.9994
8333/8333 [==============================] - 1s 99us/step
Epoch 1/3
16666/16666 [==============================] - 7s 419us/step - loss: 0.3726 -
acc: 0.8359
Epoch 2/3
16666/16666 [==============================] - 3s 177us/step - loss: 0.1180 -
acc: 0.9606
Epoch 3/3
16666/16666 [==============================] - 3s 174us/step - loss: 0.0384 -
acc: 0.9899
8334/8334 [==============================] - 3s 306us/step
Epoch 1/3
16667/16667 [==============================] - 3s 175us/step - loss: 0.1952 -
acc: 0.9341
Epoch 2/3
16667/16667 [==============================] - 3s 174us/step - loss: 0.0457 -
acc: 0.9872
Epoch 3/3
16667/16667 [==============================] - 3s 175us/step - loss: 0.0114 -
acc: 0.9985
8333/8333 [==============================] - 1s 100us/step
Epoch 1/3
16667/16667 [==============================] - 3s 176us/step - loss: 0.0434 -
acc: 0.9857
Epoch 2/3
16667/16667 [==============================] - 3s 175us/step - loss: 0.0125 -
acc: 0.9984
Epoch 3/3
```

```
16667/16667 [==============================] - 3s 175us/step - loss: 0.0059 -
acc: 0.9996
8333/8333 [==============================] - 1s 98us/step
Epoch 1/3
16666/16666 [==============================] - 7s 426us/step - loss: 0.3803 -
acc: 0.8379
Epoch 2/3
16666/16666 [==============================] - 3s 174us/step - loss: 0.1244 -
acc: 0.9643
Epoch 3/3
16666/16666 [==============================] - 3s 178us/step - loss: 0.0503 -
acc: 0.9917
8334/8334 [==============================] - 3s 309us/step
Epoch 1/3
16667/16667 [==============================] - 3s 175us/step - loss: 0.2144 -
acc: 0.9367
Epoch 2/3
16667/16667 [==============================] - 3s 174us/step - loss: 0.0651 -
acc: 0.9880
Epoch 3/3
16667/16667 [==============================] - 3s 175us/step - loss: 0.0322 -
acc: 0.9981
8333/8333 [==============================] - 1s 100us/step
Epoch 1/3
16667/16667 [==============================] - 3s 176us/step - loss: 0.0654 -
acc: 0.9855
Epoch 2/3
16667/16667 [==============================] - 3s 174us/step - loss: 0.0382 -
acc: 0.9968
Epoch 3/3
16667/16667 [==============================] - 3s 174us/step - loss: 0.0293 -
acc: 0.9995
8333/8333 [==============================] - 1s 101us/step
Epoch 1/3
16666/16666 [==============================] - 7s 432us/step - loss: 0.4978 -
acc: 0.8374
Epoch 2/3
16666/16666 [==============================] - 3s 176us/step - loss: 0.2396 -
acc: 0.9585
Epoch 3/3
16666/16666 [==============================] - 3s 173us/step - loss: 0.1606 -
acc: 0.9849
8334/8334 [==============================] - 3s 320us/step
Epoch 1/3
16667/16667 [==============================] - 3s 189us/step - loss: 0.3119 -
acc: 0.9294
Epoch 2/3
16667/16667 [==============================] - 3s 184us/step - loss: 0.1693 -
acc: 0.9826
Epoch 3/3
16667/16667 [==============================] - 3s 183us/step - loss: 0.1187 -
acc: 0.9964
8333/8333 [==============================] - 1s 105us/step
Epoch 1/3
16667/16667 [==============================] - 3s 187us/step - loss: 0.1946 -
acc: 0.9657
Epoch 2/3
```

```
16667/16667 [==============================] - 3s 178us/step - loss: 0.1472 -
acc: 0.9900
Epoch 3/3
16667/16667 [==============================] - 3s 177us/step - loss: 0.1209 -
acc: 0.9975
8333/8333 [==============================] - 1s 103us/step
Epoch 1/3
16666/16666 [==============================] - 7s 427us/step - loss: 1.1380 -
acc: 0.8364
Epoch 2/3
16666/16666 [==============================] - 3s 175us/step - loss: 0.5568 -
acc: 0.9267
Epoch 3/3
16666/16666 [==============================] - 3s 173us/step - loss: 0.4381 -
acc: 0.9302
8334/8334 [==============================] - 3s 316us/step
Epoch 1/3
16667/16667 [==============================] - 3s 175us/step - loss: 0.4591 -
acc: 0.9032
Epoch 2/3
16667/16667 [==============================] - 3s 176us/step - loss: 0.3410 -
acc: 0.9408
Epoch 3/3
16667/16667 [==============================] - 3s 175us/step - loss: 0.3277 -
acc: 0.9428
8333/8333 [==============================] - 1s 99us/step
Epoch 1/3
16667/16667 [==============================] - 4s 233us/step - loss: 0.3976 -
acc: 0.9152
Epoch 2/3
16667/16667 [==============================] - 4s 215us/step - loss: 0.3011 -
acc: 0.9511
Epoch 3/3
16667/16667 [==============================] - 3s 187us/step - loss: 0.2914 -
acc: 0.9526
8333/8333 [==============================] - 1s 120us/step
Epoch 1/3
16666/16666 [==============================] - 11s 644us/step - loss: 4.8265
- acc: 0.8372
Epoch 2/3
16666/16666 [==============================] - 3s 200us/step - loss: 1.6492 -
acc: 0.8780
Epoch 3/3
16666/16666 [==============================] - 4s 250us/step - loss: 0.9525 -
acc: 0.8799
8334/8334 [==============================] - 3s 402us/step
Epoch 1/3
16667/16667 [==============================] - 4s 221us/step - loss: 0.7578 -
acc: 0.8780
Epoch 2/3
16667/16667 [==============================] - 4s 220us/step - loss: 0.7033 -
acc: 0.8783
Epoch 3/3
16667/16667 [==============================] - 4s 235us/step - loss: 0.6863 -
acc: 0.8861
8333/8333 [==============================] - 1s 154us/step
Epoch 1/3
```

```
16667/16667 [==============================] - 4s 217us/step - loss: 0.6862 -
acc: 0.8844
Epoch 2/3
16667/16667 [==============================] - 3s 194us/step - loss: 0.6809 -
acc: 0.8838
Epoch 3/3
16667/16667 [==============================] - 3s 191us/step - loss: 0.6703 -
acc: 0.8899
8333/8333 [==============================] - 1s 109us/step
```

In [76]:

```
l2_2_reg_cv = pd.DataFrame({'loss_train': loss_train_2_l2,
                            'loss_val': loss_val_2_l2,
                         'acc_train': acc_train_2_l2,
                          'acc_val':acc_val_2_l2,
                           'lambda':lambda_list}).set_index('lambda')
l2_2_reg_cv
```

Out[76]:

lambda	loss_train	loss_val	acc_train	acc_val
0.000001	0.015571	0.182387	0.99614	0.950123
0.000010	0.018547	0.176901	0.99598	0.950683
0.000100	0.037256	0.205451	0.99642	0.951043
0.001000	0.133404	0.314742	0.99294	0.930482
0.010000	0.352425	0.496134	0.94188	0.877400
0.100000	0.769707	0.733524	0.88528	0.873720

In [78]:
```
# indicating by the result of l2_2_reg_cv data frame, will set the lambda as 1
e-4 for l2 norm regularization
model_2_l2 = Sequential()
model_2_l2.add(Dense(50,input_shape = (max_num,), kernel_regularizer = l2(1e-4
), bias_regularizer=l2(1e-4)))
model_2_l2.add(Activation('relu'))
model_2_l2.add(Dense(50, kernel_regularizer = l2(1e-4), bias_regularizer=l2(1e
-4)))
model_2_l2.add(Activation('relu'))
model_2_l2.add(Dense(num_classes, kernel_regularizer = l2(1e-4), bias_regulari
zer=l2(1e-4)))
model_2_l2.add(Activation('softmax'))
model_2_l2.compile(loss = 'categorical_crossentropy', optimizer = 'adam', metr
ics = ['accuracy'])

hist_2_l2 = model_2_l2.fit(X_train,y_train, batch_size=200, epochs = 3, verbos
e = 1)
score_2_l2 = model_2_l2.evaluate(X_test,y_test, batch_size=200, verbose = 1)

l2_train_acc_2 = hist_2_l2.history.get('acc')[-1]
l2_test_acc_2 = score_2_l2[1]
```

```
Epoch 1/3
25000/25000 [==============================] - 9s 342us/step - loss: 0.3527 -
acc: 0.8592
Epoch 2/3
25000/25000 [==============================] - 4s 164us/step - loss: 0.1344 -
acc: 0.9594
Epoch 3/3
25000/25000 [==============================] - 4s 163us/step - loss: 0.0587 -
acc: 0.9888
25000/25000 [==============================] - 4s 167us/step
```

In [79]:
```
print('The accurate rate of the model with two hidden layers and l2 regulariza
tion on training dataset is')
print(l2_train_acc_2)
print('The accurate rate of the model with two hidden layers and l2 regulariza
tion on test dataset is')
print(l2_test_acc_2)
```

```
The accurate rate of the model with two hidden layers and l2 regularization o
n training dataset is
0.988760009765625
The accurate rate of the model with two hidden layers and l2 regularization o
n test dataset is
0.862720000743866
```

In [80]:
```
# indicating by the result of l2_2_reg_cv data frame, will set the lambda as 1
e-5 for l2 norm regularization
model_2_l2 = Sequential()
model_2_l2.add(Dense(50,input_shape = (max_num,), kernel_regularizer = l2(1e-5
), bias_regularizer=l2(1e-5)))
model_2_l2.add(Activation('relu'))
model_2_l2.add(Dense(50, kernel_regularizer = l2(1e-5), bias_regularizer=l2(1e
-5)))
model_2_l2.add(Activation('relu'))
model_2_l2.add(Dense(num_classes, kernel_regularizer = l2(1e-5), bias_regulari
zer=l2(1e-5)))
model_2_l2.add(Activation('softmax'))
model_2_l2.compile(loss = 'categorical_crossentropy', optimizer = 'adam', metr
ics = ['accuracy'])

hist_2_l2 = model_2_l2.fit(X_train,y_train, batch_size=200, epochs = 3, verbos
e = 1)
score_2_l2 = model_2_l2.evaluate(X_test,y_test, batch_size=200, verbose = 1)

l2_train_acc_2 = hist_2_l2.history.get('acc')[-1]
l2_test_acc_2 = score_2_l2[1]
```

```
Epoch 1/3
25000/25000 [==============================] - 8s 340us/step - loss: 0.3385 -
acc: 0.8563
Epoch 2/3
25000/25000 [==============================] - 4s 167us/step - loss: 0.1322 -
acc: 0.9526
Epoch 3/3
25000/25000 [==============================] - 4s 166us/step - loss: 0.0576 -
acc: 0.9823
25000/25000 [==============================] - 4s 165us/step
```

In [81]:
```
print('The accurate rate of the model with two hidden layers and l2 regulariza
tion on training dataset is')
print(l2_train_acc_2)
print('The accurate rate of the model with two hidden layers and l2 regulariza
tion on test dataset is')
print(l2_test_acc_2)
```

```
The accurate rate of the model with two hidden layers and l2 regularization o
n training dataset is
0.9823200106620789
The accurate rate of the model with two hidden layers and l2 regularization o
n test dataset is
0.8600399994850159
```

In [83]:
```
print(model_2_l2.summary())
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
dense_258 (Dense)            (None, 50)                500050    
_________________________________________________________________
activation_210 (Activation)  (None, 50)                0         
_________________________________________________________________
dense_259 (Dense)            (None, 50)                2550      
_________________________________________________________________
activation_211 (Activation)  (None, 50)                0         
_________________________________________________________________
dense_260 (Dense)            (None, 2)                 102       
_________________________________________________________________
activation_212 (Activation)  (None, 2)                 0         
=================================================================
Total params: 502,702
Trainable params: 502,702
Non-trainable params: 0
_________________________________________________________________
None
```

In [ ]:

augment the training dataset

In [3]:
```
data = np.concatenate((X_train, X_test), axis=0)
targets = np.concatenate((y_train, y_test), axis=0)
```

In [4]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(data, targets, test_size=
0.2)
```

In [5]:
```
num_classes = max(y_train) + 1
print(num_classes)

from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.utils import to_categorical

max_num = 10000
tokenizer = Tokenizer(num_words=max_num)
tokenizer.fit_on_sequences(X_train)
X_train = tokenizer.sequences_to_matrix(X_train, mode = 'tfidf')
X_test = tokenizer.sequences_to_matrix(X_test, mode = 'tfidf')

y_train = to_categorical(y_train, num_classes)
y_test = to_categorical(y_test, num_classes)
y_train.shape
```

```
2
```

Out[5]:
```
KFold(n_splits=3, random_state=None, shuffle=True)
```

In [6]:
```
# the baseline to compare with
model_overfitting = Sequential()
model_overfitting.add(Dense(250,input_shape = (max_num,)))
model_overfitting.add(Activation('relu'))
model_overfitting.add(Dense(num_classes))
model_overfitting.add(Activation('softmax'))
model_overfitting.compile(loss = 'categorical_crossentropy', optimizer = 'ada
m', metrics = ['accuracy'])

hist_overfitting = model_overfitting.fit(X_train,y_train, batch_size=200, epoc
hs = 3, verbose = 1)
score_overfitting = model_overfitting.evaluate(X_test,y_test, batch_size=200,
verbose = 1)

base_train_acc = hist_overfitting.history.get('acc')[-1]
base_test_acc = score_overfitting[1]

print('The accurate rate of the overfitting model on training dataset is')
print(base_train_acc)
print('The accurate rate of the overfitting model on test dataset is')
print(base_test_acc)
```

```
Epoch 1/3
40000/40000 [==============================] - 22s 553us/step - loss: 0.3048
- acc: 0.8711
Epoch 2/3
40000/40000 [==============================] - 21s 518us/step - loss: 0.0810
- acc: 0.9741
Epoch 3/3
40000/40000 [==============================] - 20s 510us/step - loss: 0.0160
- acc: 0.9971
10000/10000 [==============================] - 2s 232us/step
The accurate rate of the overfitting model on training dataset is
0.9971000027656555
The accurate rate of the overfitting model on test dataset is
0.8960999977588654
```

In [7]:

```
# baseline model with two hidden layers
model_overfitting_2 = Sequential()
model_overfitting_2.add(Dense(50,input_shape = (max_num,)))
model_overfitting_2.add(Activation('relu'))
model_overfitting_2.add(Dense(50))
model_overfitting_2.add(Activation('relu'))
model_overfitting_2.add(Dense(num_classes))
model_overfitting_2.add(Activation('softmax'))
model_overfitting_2.compile(loss = 'categorical_crossentropy', optimizer = 'ad
am', metrics = ['accuracy'])

hist_overfitting_2 = model_overfitting_2.fit(X_train,y_train, batch_size=200,
epochs = 3, verbose = 1)
score_overfitting_2 = model_overfitting_2.evaluate(X_test,y_test, batch_size=2
00, verbose = 1)

base_train_acc_2 = hist_overfitting_2.history.get('acc')[-1]
base_test_acc_2 = score_overfitting_2[1]

print('The accurate rate of the overfitting model with two hidden layers on tr
aining dataset is')
print(base_train_acc_2)
print('The accurate rate of the overfitting model with two hidden layers on te
st dataset is')
print(base_test_acc_2)
```

```
Epoch 1/3
40000/40000 [==============================] - 10s 258us/step - loss: 0.3205
- acc: 0.8643
Epoch 2/3
40000/40000 [==============================] - 9s 227us/step - loss: 0.1456 -
acc: 0.9445
Epoch 3/3
40000/40000 [==============================] - 9s 233us/step - loss: 0.0584 -
acc: 0.9799
10000/10000 [==============================] - 2s 169us/step
The accurate rate of the overfitting model with two hidden layers on training
dataset is
0.9799250122904778
The accurate rate of the overfitting model with two hidden layers on test dat
aset is
0.8881999969482421
```