Question 1

Minimization of Negative Log Likelihood with Softmax outputs

We are given the negative log likelihood function that we want to minimize:

$J(w) = -\log(\prod_n \prod_{m=0}^{9} p(t_n = m|x_n; w))$

where there are 0 to 9 classes.

Now let us define that $p(t_m|x; w)$ is a categorical probability distribution that is :

$p(t_m|x; w) = y_m(x; w)^{t_m}$

following the constraints that $t_m \in \{0, 1\} and \sum_m t_m = 1$

As we know that we can interpret the output of the softmax as the probabilities that a certain set of features belongs to a certain class. Let us use the probability nodes as softmax output nodes

$y_i(x; w) = \frac{\exp^{y_i(x;w)}}{\sum_{j=0}^{m} \exp^{y_j(x;w)}}$

This ensures that sum of probability of each feature sample belonging to one of the 10 classes equals to 1

The negative log likelihood can be reduced to :

$J(w) = -\sum_n \sum_{m=0}^{9} \log p(t_n = m|x_n; w)$

$J(w) = -\sum_n \sum_{m=0}^{9} t_m^{(n)} \log(y_m^{(n)}(x^{(n)}; w))$

Now the nature of curve of negative log likelihood is :

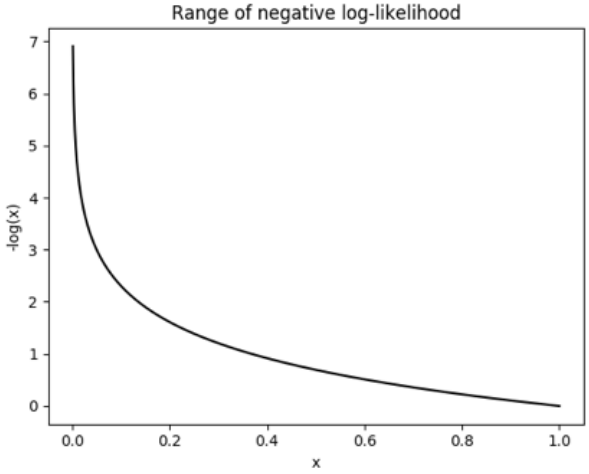


Figure: *The loss function reaches infinity when input is 0, and reaches 0 when input is 1.*

This means for the curve to reach its minimum we have to generate the likelihood of the correct label as close to 1 as possible.

Now noting that t_m is one-hot encoded vector thus reducing all terms of J(w) to 0 other than the target(true) class say 'k' for every feature vector i.e.

$$J(w) = -\sum_n 1 * \log(y_k^{(n)}(x^{(n)}; w)) = -\log(\prod_n (y_k^{(n)}(x^{(n)}; w)))$$

Thus the weights must be trained in such a way that the likelihood softmax(probability) of the target label is maximum for each feature vector (i.e. close to 1).

Hence it is proved that a neural network to maximize the log likelihood of label is one that has softmax output nodes and minimizes the criterion function of the negative log probability of training data set

Interpretation of L2 regularization

Let us assume that our input is linearly correlated with our output using the following equation

$$y_i = wx_i + \epsilon$$

where w is the set of weights and ϵ is the random noise that also follows normal distribution with 0 mean and constant variance σ^2.

This gives us the likelihood that we want to maximize:

$$\prod_{i=1}^{N} \mathcal{N}(y_i; x_i, w, \sigma^2)$$

Now,let us assume that w has the Gaussian prior $\mathcal{N}(0, \alpha^{-1})$

From Bayes Rule we know:

$$P(w|D) = \frac{P(D|w)P(w)}{P(D)}$$

$$P(w|D) \approx P(D|w)P(w)$$

Thus the posterior probability is:

$$P(w|D) = \prod_{i=1}^{N} \mathcal{N}(y_i; x_i, w, \sigma^2)\mathcal{N}(0, \alpha^{-1})$$

Taking logarithm of the posterior:

$$\log P(w|D) = \sum_{i=1}^{N} \log \mathcal{N}(y_i; x_i, w, \sigma^2) + \log \mathcal{N}(0, \alpha^{-1}) + constant$$

Ignoring the constant we need to maximize over w

$$-\frac{1}{\sigma^2} \sum_{i=1}^{N} (y_i - wx_i)^2 - \alpha w^2$$

This is the same as minimizing the cost function with L2 regularization with α being the regularization constant.

Thus we have proved optimizing model weights to minimize loss function with L2 regularization is equivalent to finding the weights that are most likely under a posterior distribution evaluated using Bayes rule, with a zero-mean independent Gaussian weights prior

In [1]:
```
import tensorflow as tf
import numpy as np
```

In [2]:
```
from keras.datasets import mnist

(x_train, y_train), (x_test, y_test) = mnist.load_data()
```

```
Using TensorFlow backend.
```

In [3]:
```
x_train.shape,y_train.shape
```

Out[3]: ((60000, 28, 28), (60000,))

In [4]:
```
x_test.shape,y_test.shape
```

Out[4]: ((10000, 28, 28), (10000,))

Reshaping and normalizing the data

In [5]:
```
x_train=x_train.reshape((60000,784))

x_test=x_test.reshape((10000,784))
```

In [6]:
```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()

x_train=sc.fit_transform(x_train)
x_test=sc.fit_transform(x_test)
```

Generating 1000 images with 100 images per class

In [8]:
```
def generate_minidata(x,y):

  mini=[]

  for i in range(10):
    count=0
    for j in range(len(x)):
      if y[j]==i:
        mini.append(np.array([i]+list(x[j])))
        count+=1
      if count==100:
        break

  mini=np.array(mini)
  np.random.shuffle(mini)
  #return(mini[:,0].reshape((len(mini),1)),mini[:,1:])
  return(mini[:,0],mini[:,1:])
```

In [9]:
```
mini_trainlabels,mini_train=generate_minidata(x_train,y_train)
```

In [10]:
```
mini_testlabels,mini_test=generate_minidata(x_test,y_test)
```

In [11]:
```
mini_train.shape,mini_trainlabels.shape
```

Out[11]:
```
((1000, 784), (1000,))
```

In [12]:
```
mini_test.shape,mini_testlabels.shape
```

Out[12]:
```
((1000, 784), (1000,))
```

One hot encoding

In [15]:
```
from sklearn.preprocessing import OneHotEncoder
ohe = OneHotEncoder()
train_labels = ohe.fit_transform(mini_trainlabels.reshape((1000,1))
).toarray()
test_labels = ohe.fit_transform(mini_testlabels.reshape((1000,1))).
toarray()
```

In [16]:
```
print(mini_test.shape,test_labels.shape)
```

```
(1000, 784) (1000, 10)
```

In [17]:
```
print(mini_train.shape,train_labels.shape)
```

```
(1000, 784) (1000, 10)
```

Question 2)a) 1 Hidden Layer

In [31]:
```
def neural_net_1hidden(l2):

    model = tf.keras.Sequential()

    if l2==True:
        model.add(tf.keras.layers.Dense(30, input_dim=784,activatio
n='sigmoid',kernel_regularizer=tf.keras.regularizers.l2(5)))

    else:
        model.add(tf.keras.layers.Dense(30, input_dim=784,activatio
n='sigmoid'))

    model.add(tf.keras.layers.Dense(10, activation='softmax'))

    model.summary()

    adam=tf.keras.optimizers.Adam(learning_rate=0.1)
    model.compile(loss='categorical_crossentropy', optimizer=adam,
metrics=['accuracy'])

    return model
```

One hidden layer without regularization

In [19]:
```
model=neural_net_1hidden(False)
history = model.fit(mini_train, train_labels, epochs=30, batch_size
=10,validation_data=(mini_test,test_labels))
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 30)                23550
_________________________________________________________________
dense_1 (Dense)              (None, 10)                310
=================================================================
Total params: 23,860
Trainable params: 23,860
Non-trainable params: 0
_________________________________________________________________
Train on 1000 samples, validate on 1000 samples
Epoch 1/30
1000/1000 [==============================] - 1s 1ms/sample - loss:
1.3298 - accuracy: 0.5870 - val_loss: 1.2903 - val_accuracy: 0.573
0
Epoch 2/30
1000/1000 [==============================] - 0s 316us/sample - los
s: 0.8941 - accuracy: 0.7260 - val_loss: 1.1346 - val_accuracy: 0.
6490
Epoch 3/30
1000/1000 [==============================] - 0s 317us/sample - los
```

```
s: 0.8239 - accuracy: 0.7640 - val_loss: 1.1899 - val_accuracy: 0.
6460
Epoch 4/30
1000/1000 [==============================] - 0s 312us/sample - los
s: 0.7272 - accuracy: 0.7710 - val_loss: 1.1487 - val_accuracy: 0.
6830
Epoch 5/30
1000/1000 [==============================] - 0s 309us/sample - los
s: 0.8568 - accuracy: 0.7420 - val_loss: 1.0889 - val_accuracy: 0.
6760
Epoch 6/30
1000/1000 [==============================] - 0s 327us/sample - los
s: 0.8361 - accuracy: 0.7670 - val_loss: 1.2460 - val_accuracy: 0.
6770
Epoch 7/30
1000/1000 [==============================] - 0s 310us/sample - los
s: 0.8100 - accuracy: 0.7710 - val_loss: 1.1949 - val_accuracy: 0.
6740
Epoch 8/30
1000/1000 [==============================] - 0s 301us/sample - los
s: 0.7046 - accuracy: 0.8110 - val_loss: 1.1376 - val_accuracy: 0.
6910
Epoch 9/30
1000/1000 [==============================] - 0s 375us/sample - los
s: 0.7463 - accuracy: 0.7980 - val_loss: 1.3716 - val_accuracy: 0.
6540
Epoch 10/30
1000/1000 [==============================] - 0s 328us/sample - los
s: 0.6351 - accuracy: 0.8250 - val_loss: 1.3194 - val_accuracy: 0.
6680
Epoch 11/30
1000/1000 [==============================] - 0s 308us/sample - los
s: 0.6398 - accuracy: 0.8300 - val_loss: 1.1581 - val_accuracy: 0.
6810
Epoch 12/30
1000/1000 [==============================] - 0s 313us/sample - los
s: 0.7264 - accuracy: 0.8040 - val_loss: 1.2404 - val_accuracy: 0.
6870
Epoch 13/30
1000/1000 [==============================] - 0s 308us/sample - los
s: 0.6077 - accuracy: 0.8230 - val_loss: 1.3915 - val_accuracy: 0.
6760
Epoch 14/30
1000/1000 [==============================] - 0s 310us/sample - los
s: 0.5773 - accuracy: 0.8400 - val_loss: 1.9983 - val_accuracy: 0.
6000
Epoch 15/30
1000/1000 [==============================] - 0s 306us/sample - los
s: 0.6211 - accuracy: 0.8330 - val_loss: 1.3181 - val_accuracy: 0.
7010
Epoch 16/30
1000/1000 [==============================] - 0s 306us/sample - los
s: 0.5186 - accuracy: 0.8460 - val_loss: 1.4768 - val_accuracy: 0.
```

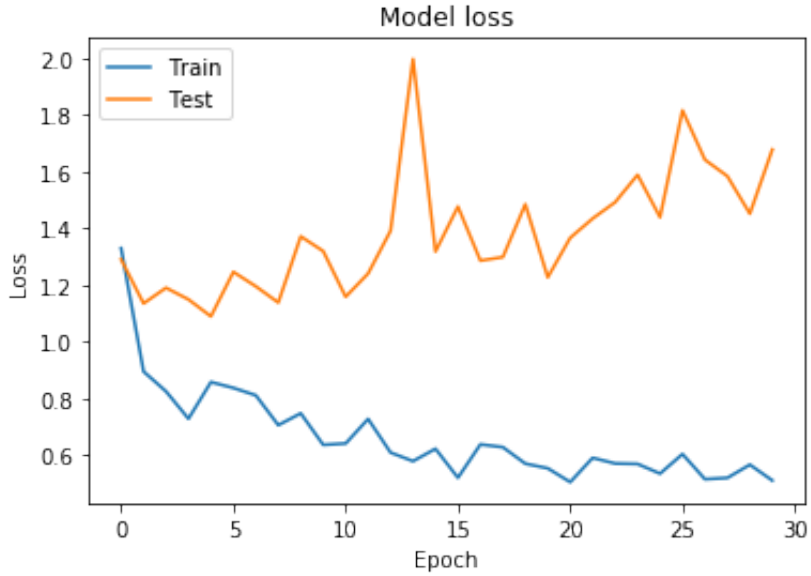
```
6900
Epoch 17/30
1000/1000 [==============================] - 0s 308us/sample - los
s: 0.6365 - accuracy: 0.8260 - val_loss: 1.2859 - val_accuracy: 0.
7180
Epoch 18/30
1000/1000 [==============================] - 0s 308us/sample - los
s: 0.6268 - accuracy: 0.8310 - val_loss: 1.2983 - val_accuracy: 0.
7160
Epoch 19/30
1000/1000 [==============================] - 0s 301us/sample - los
s: 0.5686 - accuracy: 0.8380 - val_loss: 1.4851 - val_accuracy: 0.
6900
Epoch 20/30
1000/1000 [==============================] - 0s 330us/sample - los
s: 0.5521 - accuracy: 0.8500 - val_loss: 1.2269 - val_accuracy: 0.
7240
Epoch 21/30
1000/1000 [==============================] - 0s 301us/sample - los
s: 0.5034 - accuracy: 0.8640 - val_loss: 1.3665 - val_accuracy: 0.
7180
Epoch 22/30
1000/1000 [==============================] - 0s 303us/sample - los
s: 0.5890 - accuracy: 0.8380 - val_loss: 1.4347 - val_accuracy: 0.
6960
Epoch 23/30
1000/1000 [==============================] - 0s 337us/sample - los
s: 0.5691 - accuracy: 0.8460 - val_loss: 1.4923 - val_accuracy: 0.
6800
Epoch 24/30
1000/1000 [==============================] - 0s 321us/sample - los
s: 0.5675 - accuracy: 0.8580 - val_loss: 1.5891 - val_accuracy: 0.
6740
Epoch 25/30
1000/1000 [==============================] - 0s 417us/sample - los
s: 0.5334 - accuracy: 0.8490 - val_loss: 1.4392 - val_accuracy: 0.
6800
Epoch 26/30
1000/1000 [==============================] - 0s 431us/sample - los
s: 0.6023 - accuracy: 0.8260 - val_loss: 1.8166 - val_accuracy: 0.
6560
Epoch 27/30
1000/1000 [==============================] - 0s 324us/sample - los
s: 0.5137 - accuracy: 0.8680 - val_loss: 1.6423 - val_accuracy: 0.
7040
Epoch 28/30
1000/1000 [==============================] - 0s 333us/sample - los
s: 0.5182 - accuracy: 0.8610 - val_loss: 1.5842 - val_accuracy: 0.
7010
Epoch 29/30
1000/1000 [==============================] - 0s 318us/sample - los
s: 0.5649 - accuracy: 0.8660 - val_loss: 1.4517 - val_accuracy: 0.
7210
```

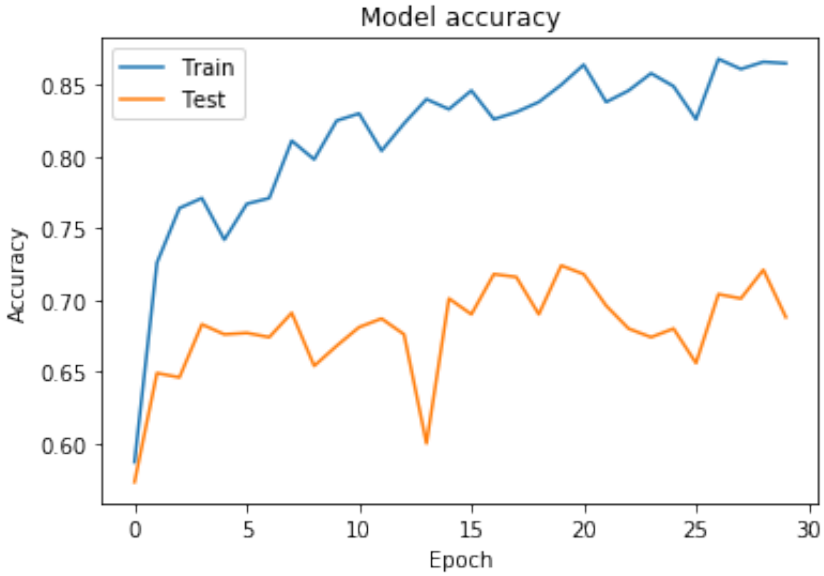
```
Epoch 30/30
1000/1000 [==============================] - 0s 359us/sample - los
s: 0.5091 - accuracy: 0.8650 - val_loss: 1.6785 - val_accuracy: 0.
6880
```

In [20]:
```
import matplotlib.pyplot as plt

def model_training_plot():

    # Plot training & validation loss values
    plt.plot(history.history['loss'])
    plt.plot(history.history['val_loss'])
    plt.title('Model loss')
    plt.ylabel('Loss')
    plt.xlabel('Epoch')
    plt.legend(['Train', 'Test'], loc='upper left')
    plt.show()

    # Plot training & validation accuracy values
    plt.plot(history.history['accuracy'])
    plt.plot(history.history['val_accuracy'])
    plt.title('Model accuracy')
    plt.ylabel('Accuracy')
    plt.xlabel('Epoch')
    plt.legend(['Train', 'Test'], loc='upper left')
    plt.show()
```

Plot of criterion categorical cross entropy loss function

In [21]:
```
model_training_plot()
```


Model loss
Train
Test
Loss
Epoch
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0
5
10
15
20
25
30


Model accuracy
Train
Test
Accuracy
Epoch
0.85
0.80
0.75
0.70
0.65
0.60
0
5
10
15
20
25
30

In [22]:
```
def for_error_plot(model):

    train_errors=[]
    test_errors=[]
    train_accuracy=[]
    test_accuracy=[]

    for i in range(30):

        model.fit(mini_train, train_labels, epochs=1, batch_size=10
)

        predictions_train=model.predict_classes(mini_train)

        predictions_test=model.predict_classes(mini_test)

        te=0
        tr=0
        for i in range(1000):
            if train_labels[i][predictions_train[i]]==1:
                tr+=1
            if test_labels[i][predictions_test[i]]==1:
                te+=1

        train_errors.append(1-tr/1000)
        test_errors.append(1-te/1000)

        train_accuracy.append(tr/1000)
        test_accuracy.append(te/1000)

    return train_errors,test_errors,train_accuracy,test_accuracy
```

In [23]:
```
model=neural_net_1hidden(False)
traine,teste,traina,testa=for_error_plot(model)
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_2 (Dense)              (None, 30)                23550
_________________________________________________________________
dense_3 (Dense)              (None, 10)                310
=================================================================
Total params: 23,860
Trainable params: 23,860
Non-trainable params: 0
_________________________________________________________________
Train on 1000 samples
1000/1000 [==============================] - 1s 525us/sample - los
s: 1.3550 - accuracy: 0.6010
```

```
Train on 1000 samples
1000/1000 [==============================] - 0s 190us/sample - los
s: 0.8118 - accuracy: 0.7450
Train on 1000 samples
1000/1000 [==============================] - 0s 188us/sample - los
s: 0.8000 - accuracy: 0.7720
Train on 1000 samples
1000/1000 [==============================] - 0s 209us/sample - los
s: 0.7834 - accuracy: 0.7830
Train on 1000 samples
1000/1000 [==============================] - 0s 204us/sample - los
s: 0.6142 - accuracy: 0.8240
Train on 1000 samples
1000/1000 [==============================] - 0s 236us/sample - los
s: 0.6590 - accuracy: 0.8080
Train on 1000 samples
1000/1000 [==============================] - 0s 213us/sample - los
s: 0.5990 - accuracy: 0.8320
Train on 1000 samples
1000/1000 [==============================] - 0s 188us/sample - los
s: 0.6900 - accuracy: 0.8000
Train on 1000 samples
1000/1000 [==============================] - 0s 194us/sample - los
s: 0.6767 - accuracy: 0.8140
Train on 1000 samples
1000/1000 [==============================] - 0s 193us/sample - los
s: 0.6917 - accuracy: 0.8270
Train on 1000 samples
1000/1000 [==============================] - 0s 183us/sample - los
s: 0.6282 - accuracy: 0.8360
Train on 1000 samples
1000/1000 [==============================] - 0s 180us/sample - los
s: 0.5449 - accuracy: 0.8400
Train on 1000 samples
1000/1000 [==============================] - 0s 177us/sample - los
s: 0.5866 - accuracy: 0.8440
Train on 1000 samples
1000/1000 [==============================] - 0s 181us/sample - los
s: 0.5944 - accuracy: 0.8400
Train on 1000 samples
1000/1000 [==============================] - 0s 184us/sample - los
s: 0.6462 - accuracy: 0.8310
Train on 1000 samples
1000/1000 [==============================] - 0s 183us/sample - los
s: 0.7162 - accuracy: 0.8270
Train on 1000 samples
1000/1000 [==============================] - 0s 185us/sample - los
s: 0.7007 - accuracy: 0.8120
Train on 1000 samples
1000/1000 [==============================] - 0s 180us/sample - los
s: 0.6469 - accuracy: 0.8470
Train on 1000 samples
1000/1000 [==============================] - 0s 182us/sample - los
```

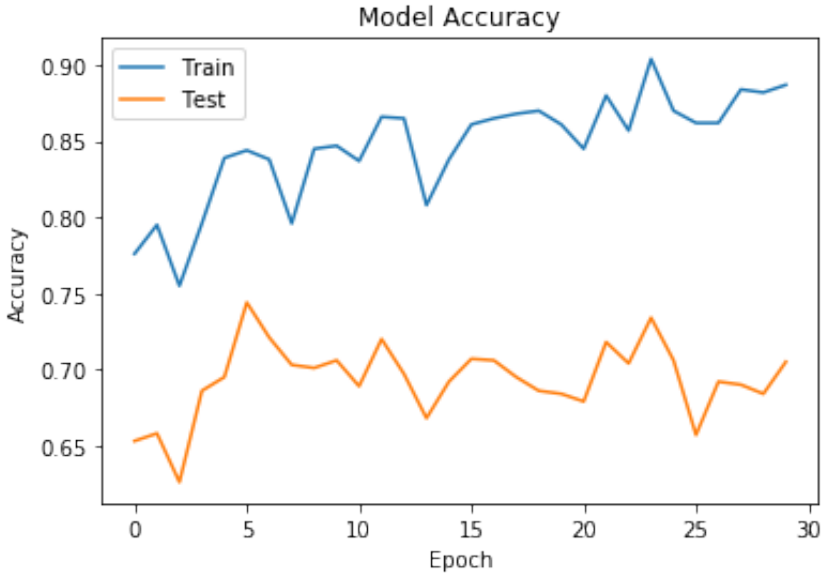
```
s: 0.6453 - accuracy: 0.8280
Train on 1000 samples
1000/1000 [==============================] - 0s 181us/sample - los
s: 0.6324 - accuracy: 0.8440
Train on 1000 samples
1000/1000 [==============================] - 0s 205us/sample - los
s: 0.6496 - accuracy: 0.8350
Train on 1000 samples
1000/1000 [==============================] - 0s 243us/sample - los
s: 0.6711 - accuracy: 0.8280
Train on 1000 samples
1000/1000 [==============================] - 0s 210us/sample - los
s: 0.5601 - accuracy: 0.8580
Train on 1000 samples
1000/1000 [==============================] - 0s 178us/sample - los
s: 0.5569 - accuracy: 0.8570
Train on 1000 samples
1000/1000 [==============================] - 0s 194us/sample - los
s: 0.5587 - accuracy: 0.8500
Train on 1000 samples
1000/1000 [==============================] - 0s 185us/sample - los
s: 0.5770 - accuracy: 0.8680
Train on 1000 samples
1000/1000 [==============================] - 0s 179us/sample - los
s: 0.5083 - accuracy: 0.8690
Train on 1000 samples
1000/1000 [==============================] - 0s 179us/sample - los
s: 0.5445 - accuracy: 0.8620
Train on 1000 samples
1000/1000 [==============================] - 0s 213us/sample - los
s: 0.5288 - accuracy: 0.8670
Train on 1000 samples
1000/1000 [==============================] - 0s 213us/sample - los
s: 0.4683 - accuracy: 0.8690
```

In [24]:
```
import matplotlib.pyplot as plt
def zero_one_error_plot(x,y,a,b):
    # Plot training & validation loss values
    plt.plot(x)
    plt.plot(y)
    plt.title('Model Error')
    plt.ylabel('Error')
    plt.xlabel('Epoch')
    plt.legend(['Train', 'Test'], loc='upper left')
    plt.show()

    plt.plot(a)
    plt.plot(b)
    plt.title('Model Accuracy')
    plt.ylabel('Accuracy')
    plt.xlabel('Epoch')
    plt.legend(['Train', 'Test'], loc='upper left')
    plt.show()
```

Plot of zero-one error

In [25]:
```
zero_one_error_plot(traine,teste,traina,testa)
```


Model Error
Train
Test
Error
0.35
0.30
0.25
0.20
0.15
0.10
0
5
10
15
20
25
30
Epoch


Model Accuracy
Train
Test
Accuracy
0.90
0.85
0.80
0.75
0.70
0.65
0
5
10
15
20
25
30
Epoch

In [26]:
```
def learning_rate_plot(params,model):

    learning_rate=[]

    for i in range(30):
        old_weights=model.get_weights()

        model.fit(mini_train, train_labels, epochs=1, batch_size=10
)


        new_weights=model.get_weights()

        summation=0

        for j in range(len(new_weights)):
            diff=np.sum(np.absolute((old_weights[j]-new_weights[j])
/new_weights[j]))
            summation+=diff

        learning_rate.append(summation/params)

    # Plot learning rate
    plt.plot(learning_rate)
    plt.title('Learning rate')
    plt.ylabel('Rate')
    plt.xlabel('Epoch')
    plt.show()
```

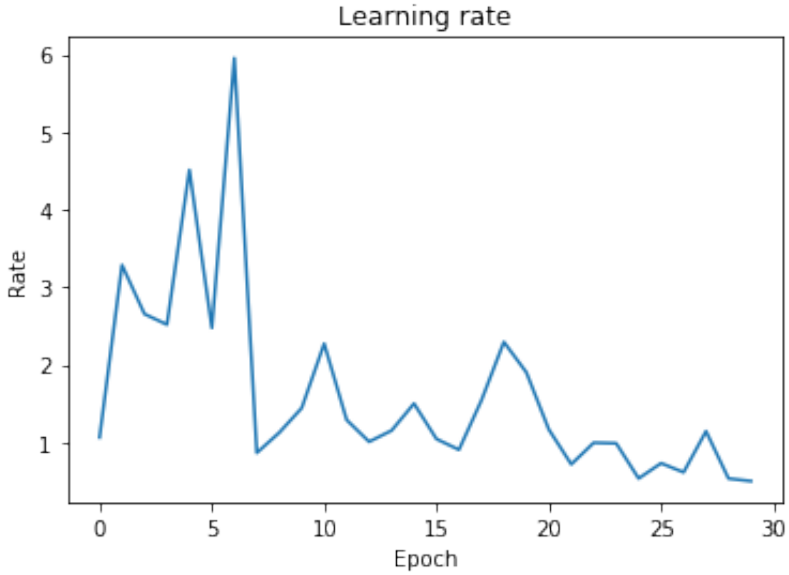
Learning speed of hidden layer

In [28]:
```
model=neural_net_1hidden(False)
learning_rate_plot(23860,model)
```

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_4 (Dense)              (None, 30)                23550
_________________________________________________________________
dense_5 (Dense)              (None, 10)                310
=================================================================
Total params: 23,860
Trainable params: 23,860
Non-trainable params: 0
_________________________________________________________________
Train on 1000 samples
1000/1000 [==============================] - 1s 525us/sample - los
s: 1.3328 - accuracy: 0.5860
Train on 1000 samples
1000/1000 [==============================] - 0s 194us/sample - los
s: 0.8527 - accuracy: 0.7260
```

```
Train on 1000 samples
1000/1000 [==============================] - 0s 180us/sample - los
s: 0.7924 - accuracy: 0.7580
Train on 1000 samples
1000/1000 [==============================] - 0s 178us/sample - los
s: 0.7943 - accuracy: 0.7660
Train on 1000 samples
1000/1000 [==============================] - 0s 181us/sample - los
s: 0.7105 - accuracy: 0.8000
Train on 1000 samples
1000/1000 [==============================] - 0s 177us/sample - los
s: 0.6744 - accuracy: 0.8120
Train on 1000 samples
1000/1000 [==============================] - 0s 181us/sample - los
s: 0.7136 - accuracy: 0.7940
Train on 1000 samples
1000/1000 [==============================] - 0s 179us/sample - los
s: 0.5927 - accuracy: 0.8290
Train on 1000 samples
1000/1000 [==============================] - 0s 179us/sample - los
s: 0.6010 - accuracy: 0.8250
Train on 1000 samples
1000/1000 [==============================] - 0s 182us/sample - los
s: 0.5567 - accuracy: 0.8310
Train on 1000 samples
1000/1000 [==============================] - 0s 182us/sample - los
s: 0.6206 - accuracy: 0.8300
Train on 1000 samples
1000/1000 [==============================] - 0s 183us/sample - los
s: 0.6232 - accuracy: 0.8190
Train on 1000 samples
1000/1000 [==============================] - 0s 184us/sample - los
s: 0.6614 - accuracy: 0.8210
Train on 1000 samples
1000/1000 [==============================] - 0s 180us/sample - los
s: 0.5912 - accuracy: 0.8470
Train on 1000 samples
1000/1000 [==============================] - 0s 191us/sample - los
s: 0.6023 - accuracy: 0.8310
Train on 1000 samples
1000/1000 [==============================] - 0s 181us/sample - los
s: 0.5788 - accuracy: 0.8460
Train on 1000 samples
1000/1000 [==============================] - 0s 181us/sample - los
s: 0.4843 - accuracy: 0.8600
Train on 1000 samples
1000/1000 [==============================] - 0s 190us/sample - los
s: 0.5590 - accuracy: 0.8540
Train on 1000 samples
1000/1000 [==============================] - 0s 181us/sample - los
s: 0.5239 - accuracy: 0.8590
Train on 1000 samples
1000/1000 [==============================] - 0s 185us/sample - los
```

```
s: 0.6009 - accuracy: 0.8370
Train on 1000 samples
1000/1000 [==============================] - 0s 184us/sample - los
s: 0.5834 - accuracy: 0.8400
Train on 1000 samples
1000/1000 [==============================] - 0s 180us/sample - los
s: 0.5561 - accuracy: 0.8540
Train on 1000 samples
1000/1000 [==============================] - 0s 181us/sample - los
s: 0.5381 - accuracy: 0.8490
Train on 1000 samples
1000/1000 [==============================] - 0s 180us/sample - los
s: 0.5531 - accuracy: 0.8320
Train on 1000 samples
1000/1000 [==============================] - 0s 181us/sample - los
s: 0.5594 - accuracy: 0.8340
Train on 1000 samples
1000/1000 [==============================] - 0s 180us/sample - los
s: 0.4907 - accuracy: 0.8630
Train on 1000 samples
1000/1000 [==============================] - 0s 180us/sample - los
s: 0.5608 - accuracy: 0.8510
Train on 1000 samples
1000/1000 [==============================] - 0s 178us/sample - los
s: 0.5077 - accuracy: 0.8570
Train on 1000 samples
1000/1000 [==============================] - 0s 180us/sample - los
s: 0.4772 - accuracy: 0.8670
Train on 1000 samples
1000/1000 [==============================] - 0s 181us/sample - los
s: 0.4610 - accuracy: 0.8750
```



One hidden layer with regularization

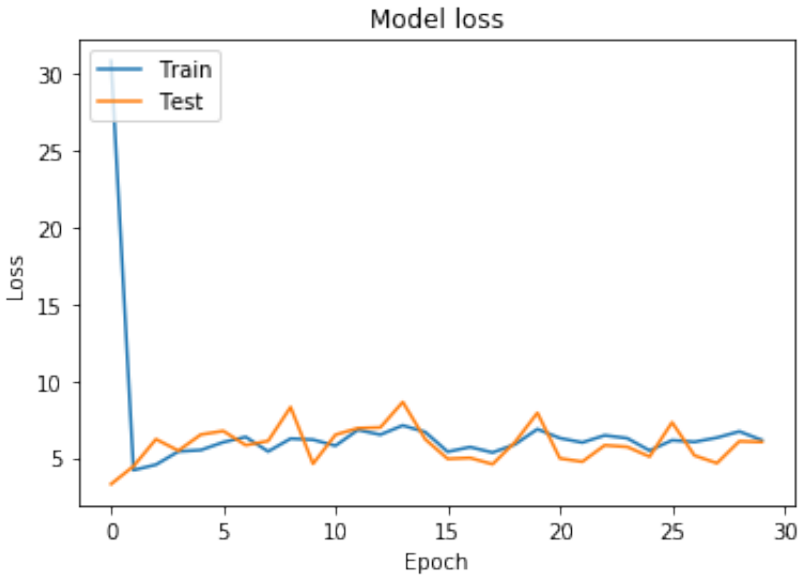
With criterion function loss

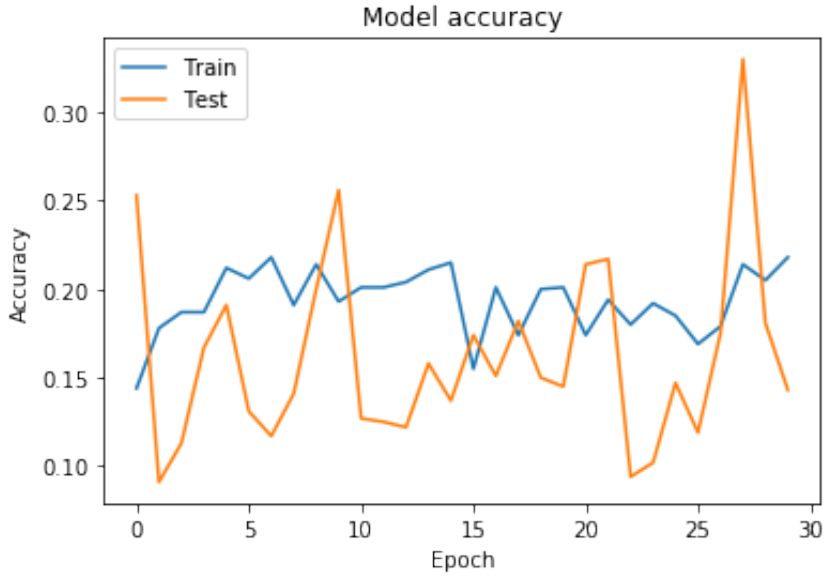
In [32]:
```
model=neural_net_1hidden(True)
history = model.fit(mini_train, train_labels, epochs=30, batch_size
=10,validation_data=(mini_test,test_labels))
model_training_plot()
```

```
Model: "sequential_5"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_9 (Dense)              (None, 30)                23550
_________________________________________________________________
dense_10 (Dense)             (None, 10)                310
=================================================================
Total params: 23,860
Trainable params: 23,860
Non-trainable params: 0
_________________________________________________________________
Train on 1000 samples, validate on 1000 samples
Epoch 1/30
1000/1000 [==============================] - 1s 818us/sample - los
s: 30.7975 - accuracy: 0.1440 - val_loss: 3.2870 - val_accuracy: 0
.2530
Epoch 2/30
1000/1000 [==============================] - 0s 335us/sample - los
s: 4.1948 - accuracy: 0.1780 - val_loss: 4.4483 - val_accuracy: 0.
0910
Epoch 3/30
1000/1000 [==============================] - 0s 419us/sample - los
s: 4.5490 - accuracy: 0.1870 - val_loss: 6.2060 - val_accuracy: 0.
1130
Epoch 4/30
1000/1000 [==============================] - 0s 358us/sample - los
s: 5.4089 - accuracy: 0.1870 - val_loss: 5.4600 - val_accuracy: 0.
1670
Epoch 5/30
1000/1000 [==============================] - 0s 317us/sample - los
s: 5.4986 - accuracy: 0.2120 - val_loss: 6.5071 - val_accuracy: 0.
1910
Epoch 6/30
1000/1000 [==============================] - 0s 328us/sample - los
s: 6.0101 - accuracy: 0.2060 - val_loss: 6.7398 - val_accuracy: 0.
1310
Epoch 7/30
1000/1000 [==============================] - 0s 387us/sample - los
s: 6.3535 - accuracy: 0.2180 - val_loss: 5.8183 - val_accuracy: 0.
1170
Epoch 8/30
1000/1000 [==============================] - 0s 352us/sample - los
s: 5.4076 - accuracy: 0.1910 - val_loss: 6.0906 - val_accuracy: 0.
1410
```

```
Epoch 9/30
1000/1000 [==============================] - 0s 321us/sample - los
s: 6.2441 - accuracy: 0.2140 - val_loss: 8.2925 - val_accuracy: 0.
2000
Epoch 10/30
1000/1000 [==============================] - 0s 326us/sample - los
s: 6.1735 - accuracy: 0.1930 - val_loss: 4.6071 - val_accuracy: 0.
2560
Epoch 11/30
1000/1000 [==============================] - 0s 414us/sample - los
s: 5.7798 - accuracy: 0.2010 - val_loss: 6.4946 - val_accuracy: 0.
1270
Epoch 12/30
1000/1000 [==============================] - 0s 314us/sample - los
s: 6.8121 - accuracy: 0.2010 - val_loss: 6.9213 - val_accuracy: 0.
1250
Epoch 13/30
1000/1000 [==============================] - 0s 370us/sample - los
s: 6.4988 - accuracy: 0.2040 - val_loss: 6.9691 - val_accuracy: 0.
1220
Epoch 14/30
1000/1000 [==============================] - 0s 355us/sample - los
s: 7.1023 - accuracy: 0.2110 - val_loss: 8.6227 - val_accuracy: 0.
1580
Epoch 15/30
1000/1000 [==============================] - 0s 378us/sample - los
s: 6.6713 - accuracy: 0.2150 - val_loss: 6.2033 - val_accuracy: 0.
1370
Epoch 16/30
1000/1000 [==============================] - 0s 329us/sample - los
s: 5.3896 - accuracy: 0.1550 - val_loss: 4.9290 - val_accuracy: 0.
1740
Epoch 17/30
1000/1000 [==============================] - 0s 328us/sample - los
s: 5.6952 - accuracy: 0.2010 - val_loss: 4.9898 - val_accuracy: 0.
1510
Epoch 18/30
1000/1000 [==============================] - 0s 348us/sample - los
s: 5.3386 - accuracy: 0.1740 - val_loss: 4.5789 - val_accuracy: 0.
1820
Epoch 19/30
1000/1000 [==============================] - 0s 335us/sample - los
s: 5.8689 - accuracy: 0.2000 - val_loss: 6.0154 - val_accuracy: 0.
1500
Epoch 20/30
1000/1000 [==============================] - 0s 370us/sample - los
s: 6.8612 - accuracy: 0.2010 - val_loss: 7.9184 - val_accuracy: 0.
1450
Epoch 21/30
1000/1000 [==============================] - 0s 358us/sample - los
s: 6.2756 - accuracy: 0.1740 - val_loss: 4.9567 - val_accuracy: 0.
2140
Epoch 22/30
```

```
1000/1000 [==============================] - 0s 347us/sample - los
s: 5.9909 - accuracy: 0.1940 - val_loss: 4.7433 - val_accuracy: 0.
2170
Epoch 23/30
1000/1000 [==============================] - 0s 313us/sample - los
s: 6.4548 - accuracy: 0.1800 - val_loss: 5.8111 - val_accuracy: 0.
0940
Epoch 24/30
1000/1000 [==============================] - 0s 425us/sample - los
s: 6.2695 - accuracy: 0.1920 - val_loss: 5.7078 - val_accuracy: 0.
1020
Epoch 25/30
1000/1000 [==============================] - 0s 387us/sample - los
s: 5.4674 - accuracy: 0.1850 - val_loss: 5.0711 - val_accuracy: 0.
1470
Epoch 26/30
1000/1000 [==============================] - 0s 408us/sample - los
s: 6.1370 - accuracy: 0.1690 - val_loss: 7.3002 - val_accuracy: 0.
1190
Epoch 27/30
1000/1000 [==============================] - 0s 367us/sample - los
s: 6.0303 - accuracy: 0.1790 - val_loss: 5.1454 - val_accuracy: 0.
1750
Epoch 28/30
1000/1000 [==============================] - 0s 313us/sample - los
s: 6.3046 - accuracy: 0.2140 - val_loss: 4.6457 - val_accuracy: 0.
3300
Epoch 29/30
1000/1000 [==============================] - 0s 331us/sample - los
s: 6.7093 - accuracy: 0.2050 - val_loss: 6.0724 - val_accuracy: 0.
1810
Epoch 30/30
1000/1000 [==============================] - 0s 317us/sample - los
s: 6.1461 - accuracy: 0.2180 - val_loss: 6.0274 - val_accuracy: 0.
1430
```





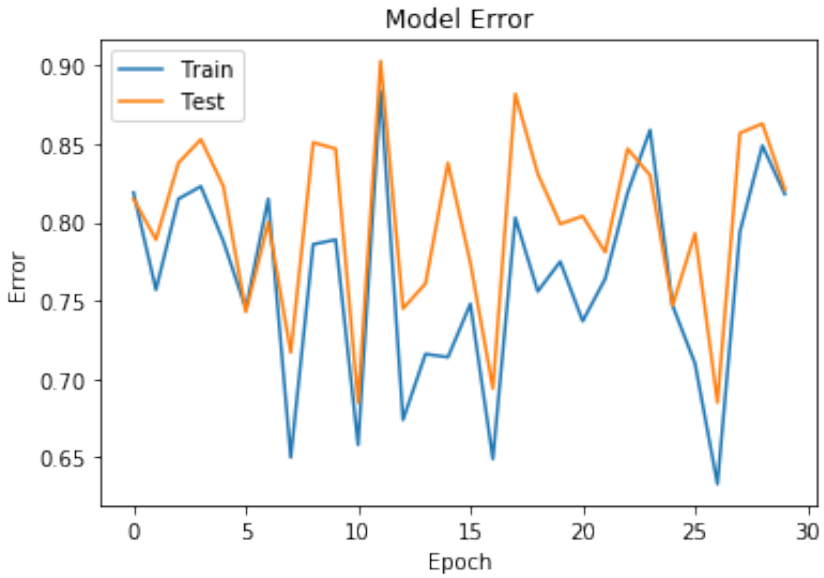
Zero-one Error

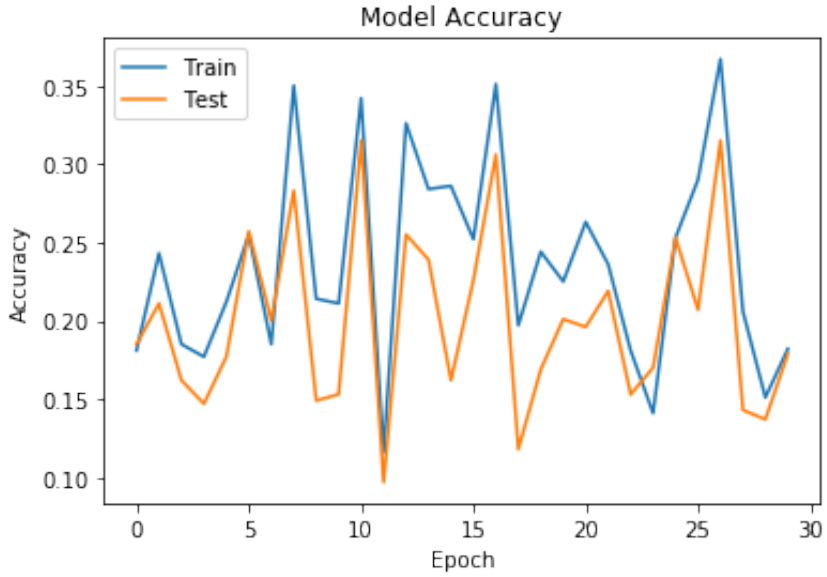
In [33]:
```
model=neural_net_1hidden(True)
traine,teste,traina,testa=for_error_plot(model)
zero_one_error_plot(traine,teste,traina,testa)
```

```
Model: "sequential_6"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_11 (Dense)             (None, 30)                23550
_________________________________________________________________
dense_12 (Dense)             (None, 10)                310
=================================================================
Total params: 23,860
Trainable params: 23,860
Non-trainable params: 0
_________________________________________________________________
Train on 1000 samples
1000/1000 [==============================] - 1s 756us/sample - los
s: 30.8283 - accuracy: 0.1350
Train on 1000 samples
1000/1000 [==============================] - 0s 259us/sample - los
s: 4.3015 - accuracy: 0.1640
Train on 1000 samples
1000/1000 [==============================] - 0s 238us/sample - los
s: 5.2170 - accuracy: 0.2010
Train on 1000 samples
1000/1000 [==============================] - 0s 186us/sample - los
s: 4.8569 - accuracy: 0.1790
Train on 1000 samples
1000/1000 [==============================] - 0s 202us/sample - los
s: 5.3937 - accuracy: 0.2070
Train on 1000 samples
```

```
1000/1000 [==============================] - 0s 185us/sample - los
s: 5.8270 - accuracy: 0.1980
Train on 1000 samples
1000/1000 [==============================] - 0s 183us/sample - los
s: 5.7690 - accuracy: 0.2310
Train on 1000 samples
1000/1000 [==============================] - 0s 188us/sample - los
s: 6.7065 - accuracy: 0.2510
Train on 1000 samples
1000/1000 [==============================] - 0s 187us/sample - los
s: 6.5117 - accuracy: 0.2100
Train on 1000 samples
1000/1000 [==============================] - 0s 197us/sample - los
s: 5.9477 - accuracy: 0.2310
Train on 1000 samples
1000/1000 [==============================] - 0s 219us/sample - los
s: 7.1525 - accuracy: 0.2470
Train on 1000 samples
1000/1000 [==============================] - 0s 189us/sample - los
s: 7.4980 - accuracy: 0.2010
Train on 1000 samples
1000/1000 [==============================] - 0s 199us/sample - los
s: 5.6581 - accuracy: 0.2220
Train on 1000 samples
1000/1000 [==============================] - 0s 204us/sample - los
s: 5.8502 - accuracy: 0.1720
Train on 1000 samples
1000/1000 [==============================] - 0s 189us/sample - los
s: 6.3494 - accuracy: 0.2100
Train on 1000 samples
1000/1000 [==============================] - 0s 237us/sample - los
s: 6.4638 - accuracy: 0.1990
Train on 1000 samples
1000/1000 [==============================] - 0s 222us/sample - los
s: 5.8281 - accuracy: 0.2160
Train on 1000 samples
1000/1000 [==============================] - 0s 242us/sample - los
s: 6.0723 - accuracy: 0.1920
Train on 1000 samples
1000/1000 [==============================] - 0s 251us/sample - los
s: 5.7845 - accuracy: 0.2230
Train on 1000 samples
1000/1000 [==============================] - 0s 244us/sample - los
s: 6.9948 - accuracy: 0.2010
Train on 1000 samples
1000/1000 [==============================] - 0s 231us/sample - los
s: 6.9080 - accuracy: 0.1870
Train on 1000 samples
1000/1000 [==============================] - 0s 220us/sample - los
s: 5.6190 - accuracy: 0.2100
Train on 1000 samples
1000/1000 [==============================] - 0s 206us/sample - los
s: 5.6348 - accuracy: 0.1820
```

```
Train on 1000 samples
1000/1000 [==============================] - 0s 197us/sample - los
s: 5.9712 - accuracy: 0.1790
Train on 1000 samples
1000/1000 [==============================] - 0s 235us/sample - los
s: 6.0057 - accuracy: 0.1990
Train on 1000 samples
1000/1000 [==============================] - ETA: 0s - loss: 5.814
5 - accuracy: 0.21 - 0s 210us/sample - loss: 5.8027 - accuracy: 0.
2150
Train on 1000 samples
1000/1000 [==============================] - 0s 221us/sample - los
s: 6.6944 - accuracy: 0.1860
Train on 1000 samples
1000/1000 [==============================] - 0s 186us/sample - los
s: 5.9219 - accuracy: 0.2030
Train on 1000 samples
1000/1000 [==============================] - 0s 247us/sample - los
s: 5.9162 - accuracy: 0.2150
Train on 1000 samples
1000/1000 [==============================] - 0s 220us/sample - los
s: 5.8164 - accuracy: 0.2100
```





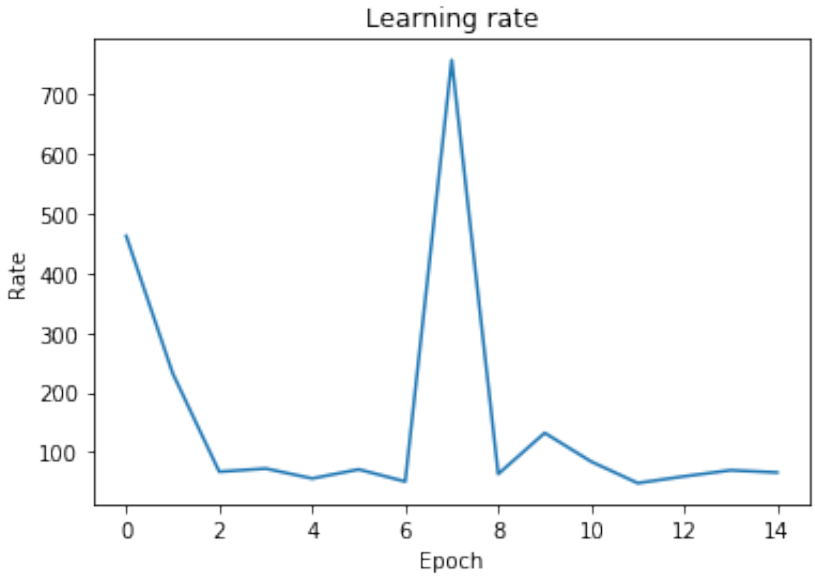
Learning rate plot

In [34]:
```
model=neural_net_1hidden(True)
learning_rate_plot(23860,model)
```

```
Model: "sequential_7"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_13 (Dense)             (None, 30)                23550
_________________________________________________________________
dense_14 (Dense)             (None, 10)                310
=================================================================
Total params: 23,860
Trainable params: 23,860
Non-trainable params: 0
_________________________________________________________________
Train on 1000 samples
1000/1000 [==============================] - 1s 572us/sample - los
s: 30.8666 - accuracy: 0.1610
Train on 1000 samples
1000/1000 [==============================] - 0s 246us/sample - los
s: 4.2672 - accuracy: 0.1750
Train on 1000 samples
1000/1000 [==============================] - 0s 184us/sample - los
s: 5.1633 - accuracy: 0.2090
Train on 1000 samples
1000/1000 [==============================] - 0s 185us/sample - los
s: 5.4903 - accuracy: 0.2090
Train on 1000 samples
1000/1000 [==============================] - 0s 193us/sample - los
s: 5.6340 - accuracy: 0.1900
Train on 1000 samples
1000/1000 [==============================] - 0s 204us/sample - los
```

```
s: 5.4936 - accuracy: 0.2080
Train on 1000 samples
1000/1000 [==============================] - 0s 183us/sample - los
s: 7.0238 - accuracy: 0.2390
Train on 1000 samples
1000/1000 [==============================] - 0s 185us/sample - los
s: 6.9018 - accuracy: 0.2250
Train on 1000 samples
1000/1000 [==============================] - 0s 185us/sample - los
s: 6.2076 - accuracy: 0.2170
Train on 1000 samples
1000/1000 [==============================] - 0s 185us/sample - los
s: 6.7087 - accuracy: 0.2140
Train on 1000 samples
1000/1000 [==============================] - 0s 184us/sample - los
s: 5.8844 - accuracy: 0.2360
Train on 1000 samples
1000/1000 [==============================] - 0s 186us/sample - los
s: 5.8784 - accuracy: 0.2300
Train on 1000 samples
1000/1000 [==============================] - 0s 245us/sample - los
s: 6.5338 - accuracy: 0.2330
Train on 1000 samples
1000/1000 [==============================] - 0s 226us/sample - los
s: 6.4696 - accuracy: 0.2190
Train on 1000 samples
1000/1000 [==============================] - 0s 226us/sample - los
s: 6.2606 - accuracy: 0.2100
Train on 1000 samples
1000/1000 [==============================] - 0s 206us/sample - los
s: 7.1551 - accuracy: 0.2230
Train on 1000 samples
1000/1000 [==============================] - 0s 238us/sample - los
s: 6.6390 - accuracy: 0.1950
Train on 1000 samples
1000/1000 [==============================] - 0s 239us/sample - los
s: 7.7188 - accuracy: 0.2360
Train on 1000 samples
1000/1000 [==============================] - 0s 230us/sample - los
s: 6.6714 - accuracy: 0.2460
Train on 1000 samples
1000/1000 [==============================] - 0s 256us/sample - los
s: 6.9349 - accuracy: 0.2280
Train on 1000 samples
1000/1000 [==============================] - 0s 226us/sample - los
s: 6.7569 - accuracy: 0.1880
Train on 1000 samples
1000/1000 [==============================] - 0s 220us/sample - los
s: 7.3080 - accuracy: 0.2300
Train on 1000 samples
1000/1000 [==============================] - 0s 236us/sample - los
s: 6.8204 - accuracy: 0.2140
Train on 1000 samples
```

```
1000/1000 [==============================] - 0s 219us/sample - los
s: 6.7142 - accuracy: 0.2160
Train on 1000 samples
1000/1000 [==============================] - 0s 242us/sample - los
s: 7.0781 - accuracy: 0.2330
Train on 1000 samples
1000/1000 [==============================] - 0s 213us/sample - los
s: 6.5930 - accuracy: 0.2090
Train on 1000 samples
1000/1000 [==============================] - 0s 191us/sample - los
s: 6.4572 - accuracy: 0.2260
Train on 1000 samples
1000/1000 [==============================] - 0s 182us/sample - los
s: 6.8373 - accuracy: 0.2180
Train on 1000 samples
1000/1000 [==============================] - 0s 197us/sample - los
s: 6.1812 - accuracy: 0.1930
Train on 1000 samples
1000/1000 [==============================] - 0s 199us/sample - los
s: 6.0630 - accuracy: 0.1970
```



Question 2)b) 2 and 3 hidden Layers

In [35]:
```
def neural_net_2hidden(l2):

    model = tf.keras.Sequential()
    if l2==True:
        model.add(tf.keras.layers.Dense(30, input_dim=784,activatio
n='sigmoid',kernel_regularizer=tf.keras.regularizers.l2(5)))
        model.add(tf.keras.layers.Dense(30, input_dim=30,activation
='sigmoid',kernel_regularizer=tf.keras.regularizers.l2(5)))
    else:
        model.add(tf.keras.layers.Dense(30, input_dim=784,activatio
n='sigmoid'))
        model.add(tf.keras.layers.Dense(30, input_dim=30,activation
='sigmoid'))

    model.add(tf.keras.layers.Dense(10, activation='softmax'))

    model.summary()

    adam=tf.keras.optimizers.Adam(learning_rate=0.1)
    model.compile(loss='categorical_crossentropy', optimizer=adam,
metrics=['accuracy'])

    return model
```

Two hidden layers without regularization

In [36]:
```
model=neural_net_2hidden(False)
history = model.fit(mini_train, train_labels, epochs=30, batch_size
=10,validation_data=(mini_test,test_labels))
```

```
Model: "sequential_8"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_15 (Dense)             (None, 30)                23550
_________________________________________________________________
dense_16 (Dense)             (None, 30)                930
_________________________________________________________________
dense_17 (Dense)             (None, 10)                310
=================================================================
Total params: 24,790
Trainable params: 24,790
Non-trainable params: 0
_________________________________________________________________
Train on 1000 samples, validate on 1000 samples
Epoch 1/30
1000/1000 [==============================] - 1s 826us/sample - los
s: 1.6937 - accuracy: 0.4290 - val_loss: 1.3850 - val_accuracy: 0.
5760
Epoch 2/30
1000/1000 [==============================] - 0s 328us/sample - los
```

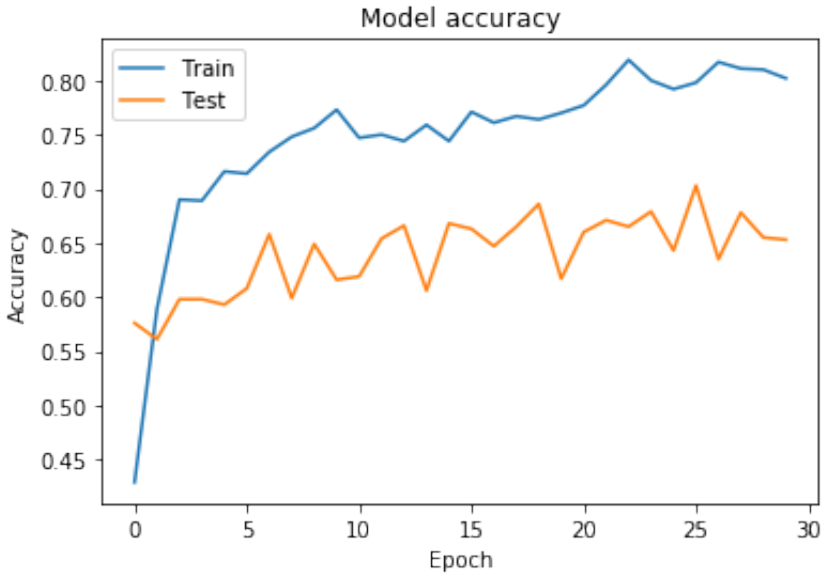
```
s: 1.1888 - accuracy: 0.5890 - val_loss: 1.2996 - val_accuracy: 0.
5610
Epoch 3/30
1000/1000 [==============================] - 0s 316us/sample - los
s: 0.9741 - accuracy: 0.6900 - val_loss: 1.2834 - val_accuracy: 0.
5980
Epoch 4/30
1000/1000 [==============================] - 0s 319us/sample - los
s: 0.9653 - accuracy: 0.6890 - val_loss: 1.2727 - val_accuracy: 0.
5980
Epoch 5/30
1000/1000 [==============================] - 0s 373us/sample - los
s: 0.9281 - accuracy: 0.7160 - val_loss: 1.4058 - val_accuracy: 0.
5930
Epoch 6/30
1000/1000 [==============================] - 0s 428us/sample - los
s: 0.9608 - accuracy: 0.7140 - val_loss: 1.2359 - val_accuracy: 0.
6080
Epoch 7/30
1000/1000 [==============================] - 0s 348us/sample - los
s: 0.8343 - accuracy: 0.7340 - val_loss: 1.1676 - val_accuracy: 0.
6580
Epoch 8/30
1000/1000 [==============================] - 0s 321us/sample - los
s: 0.8357 - accuracy: 0.7480 - val_loss: 1.3857 - val_accuracy: 0.
5990
Epoch 9/30
1000/1000 [==============================] - 0s 338us/sample - los
s: 0.7733 - accuracy: 0.7560 - val_loss: 1.2879 - val_accuracy: 0.
6490
Epoch 10/30
1000/1000 [==============================] - 0s 414us/sample - los
s: 0.7853 - accuracy: 0.7730 - val_loss: 1.3084 - val_accuracy: 0.
6160
Epoch 11/30
1000/1000 [==============================] - 0s 332us/sample - los
s: 0.8449 - accuracy: 0.7470 - val_loss: 1.3589 - val_accuracy: 0.
6190
Epoch 12/30
1000/1000 [==============================] - 0s 334us/sample - los
s: 0.8092 - accuracy: 0.7500 - val_loss: 1.2941 - val_accuracy: 0.
6540
Epoch 13/30
1000/1000 [==============================] - 0s 324us/sample - los
s: 0.8074 - accuracy: 0.7440 - val_loss: 1.1574 - val_accuracy: 0.
6660
Epoch 14/30
1000/1000 [==============================] - 0s 322us/sample - los
s: 0.7708 - accuracy: 0.7590 - val_loss: 1.4231 - val_accuracy: 0.
6060
Epoch 15/30
1000/1000 [==============================] - 0s 416us/sample - los
s: 0.8277 - accuracy: 0.7440 - val_loss: 1.2309 - val_accuracy: 0.
```

```
6680
Epoch 16/30
1000/1000 [==============================] - 0s 320us/sample - los
s: 0.7890 - accuracy: 0.7710 - val_loss: 1.1239 - val_accuracy: 0.
6630
Epoch 17/30
1000/1000 [==============================] - 0s 337us/sample - los
s: 0.7663 - accuracy: 0.7610 - val_loss: 1.2652 - val_accuracy: 0.
6470
Epoch 18/30
1000/1000 [==============================] - 0s 381us/sample - los
s: 0.7632 - accuracy: 0.7670 - val_loss: 1.1659 - val_accuracy: 0.
6650
Epoch 19/30
1000/1000 [==============================] - 0s 360us/sample - los
s: 0.7140 - accuracy: 0.7640 - val_loss: 1.1964 - val_accuracy: 0.
6860
Epoch 20/30
1000/1000 [==============================] - 0s 372us/sample - los
s: 0.7335 - accuracy: 0.7700 - val_loss: 1.3293 - val_accuracy: 0.
6170
Epoch 21/30
1000/1000 [==============================] - 0s 421us/sample - los
s: 0.7198 - accuracy: 0.7770 - val_loss: 1.2356 - val_accuracy: 0.
6600
Epoch 22/30
1000/1000 [==============================] - 0s 398us/sample - los
s: 0.6204 - accuracy: 0.7960 - val_loss: 1.3508 - val_accuracy: 0.
6710
Epoch 23/30
1000/1000 [==============================] - 0s 413us/sample - los
s: 0.5583 - accuracy: 0.8190 - val_loss: 1.2347 - val_accuracy: 0.
6650
Epoch 24/30
1000/1000 [==============================] - 0s 376us/sample - los
s: 0.6145 - accuracy: 0.8000 - val_loss: 1.2766 - val_accuracy: 0.
6790
Epoch 25/30
1000/1000 [==============================] - 0s 318us/sample - los
s: 0.6433 - accuracy: 0.7920 - val_loss: 1.2652 - val_accuracy: 0.
6430
Epoch 26/30
1000/1000 [==============================] - 0s 345us/sample - los
s: 0.6458 - accuracy: 0.7980 - val_loss: 1.2674 - val_accuracy: 0.
7030
Epoch 27/30
1000/1000 [==============================] - 0s 414us/sample - los
s: 0.6359 - accuracy: 0.8170 - val_loss: 1.4330 - val_accuracy: 0.
6350
Epoch 28/30
1000/1000 [==============================] - 0s 383us/sample - los
s: 0.6125 - accuracy: 0.8110 - val_loss: 1.1482 - val_accuracy: 0.
6780
```

```
Epoch 29/30
1000/1000 [==============================] - 0s 331us/sample - los
s: 0.6119 - accuracy: 0.8100 - val_loss: 1.3694 - val_accuracy: 0.
6550
Epoch 30/30
1000/1000 [==============================] - 0s 325us/sample - los
s: 0.6618 - accuracy: 0.8020 - val_loss: 1.3414 - val_accuracy: 0.
6530
```

Plot for criterion function loss

In [37]:
```
model_training_plot()
```



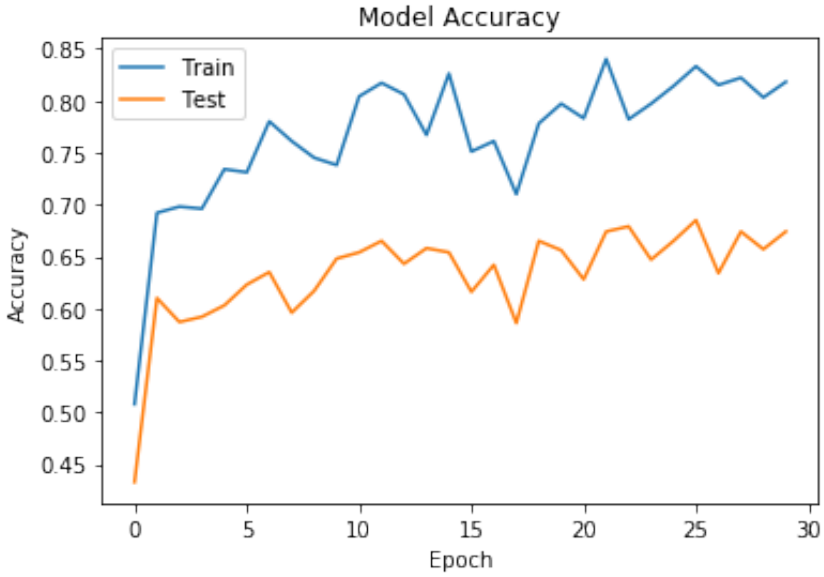
Plot for zero-one error

In [38]:
```
model=neural_net_2hidden(False)
traine,teste,traina,testa=for_error_plot(model)
zero_one_error_plot(traine,teste,traina,testa)
```

```
Model: "sequential_9"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_18 (Dense)             (None, 30)                23550
_________________________________________________________________
dense_19 (Dense)             (None, 30)                930
_________________________________________________________________
dense_20 (Dense)             (None, 10)                310
=================================================================
Total params: 24,790
Trainable params: 24,790
Non-trainable params: 0
_________________________________________________________________
Train on 1000 samples
1000/1000 [==============================] - 1s 558us/sample - los
s: 1.8175 - accuracy: 0.3470
Train on 1000 samples
1000/1000 [==============================] - 0s 188us/sample - los
s: 1.1615 - accuracy: 0.5880
Train on 1000 samples
1000/1000 [==============================] - 0s 194us/sample - los
s: 0.9782 - accuracy: 0.6580
Train on 1000 samples
1000/1000 [==============================] - 0s 193us/sample - los
s: 0.9635 - accuracy: 0.6710
Train on 1000 samples
1000/1000 [==============================] - 0s 188us/sample - los
s: 0.9301 - accuracy: 0.7080
Train on 1000 samples
1000/1000 [==============================] - 0s 193us/sample - los
s: 0.9206 - accuracy: 0.7150
Train on 1000 samples
1000/1000 [==============================] - 0s 195us/sample - los
s: 0.8934 - accuracy: 0.7140
Train on 1000 samples
1000/1000 [==============================] - 0s 225us/sample - los
s: 0.8474 - accuracy: 0.7390
Train on 1000 samples
1000/1000 [==============================] - 0s 194us/sample - los
s: 0.7925 - accuracy: 0.7540
Train on 1000 samples
1000/1000 [==============================] - 0s 195us/sample - los
s: 0.7751 - accuracy: 0.7620
Train on 1000 samples
1000/1000 [==============================] - 0s 192us/sample - los
s: 0.7839 - accuracy: 0.7720
Train on 1000 samples
1000/1000 [==============================] - 0s 196us/sample - los
```

```
s: 0.7849 - accuracy: 0.7540
Train on 1000 samples
1000/1000 [==============================] - 0s 193us/sample - los
s: 0.7600 - accuracy: 0.7760
Train on 1000 samples
1000/1000 [==============================] - 0s 195us/sample - los
s: 0.7600 - accuracy: 0.7510
Train on 1000 samples
1000/1000 [==============================] - 0s 191us/sample - los
s: 0.7074 - accuracy: 0.7770
Train on 1000 samples
1000/1000 [==============================] - 0s 193us/sample - los
s: 0.7434 - accuracy: 0.7780
Train on 1000 samples
1000/1000 [==============================] - 0s 189us/sample - los
s: 0.7563 - accuracy: 0.7660
Train on 1000 samples
1000/1000 [==============================] - 0s 258us/sample - los
s: 0.8655 - accuracy: 0.7580
Train on 1000 samples
1000/1000 [==============================] - 0s 229us/sample - los
s: 0.8545 - accuracy: 0.7510
Train on 1000 samples
1000/1000 [==============================] - 0s 192us/sample - los
s: 0.7556 - accuracy: 0.7520
Train on 1000 samples
1000/1000 [==============================] - 0s 197us/sample - los
s: 0.7085 - accuracy: 0.7850
Train on 1000 samples
1000/1000 [==============================] - 0s 196us/sample - los
s: 0.6538 - accuracy: 0.7960
Train on 1000 samples
1000/1000 [==============================] - 0s 198us/sample - los
s: 0.6791 - accuracy: 0.7920
Train on 1000 samples
1000/1000 [==============================] - 0s 193us/sample - los
s: 0.6676 - accuracy: 0.8030
Train on 1000 samples
1000/1000 [==============================] - 0s 195us/sample - los
s: 0.6636 - accuracy: 0.7970
Train on 1000 samples
1000/1000 [==============================] - 0s 191us/sample - los
s: 0.6071 - accuracy: 0.8100
Train on 1000 samples
1000/1000 [==============================] - 0s 192us/sample - los
s: 0.5883 - accuracy: 0.8070
Train on 1000 samples
1000/1000 [==============================] - 0s 252us/sample - los
s: 0.6070 - accuracy: 0.8140
Train on 1000 samples
1000/1000 [==============================] - 0s 240us/sample - los
s: 0.6154 - accuracy: 0.8040
Train on 1000 samples
```

```
1000/1000 [==============================] - 0s 196us/sample - los
s: 0.6613 - accuracy: 0.7890
```



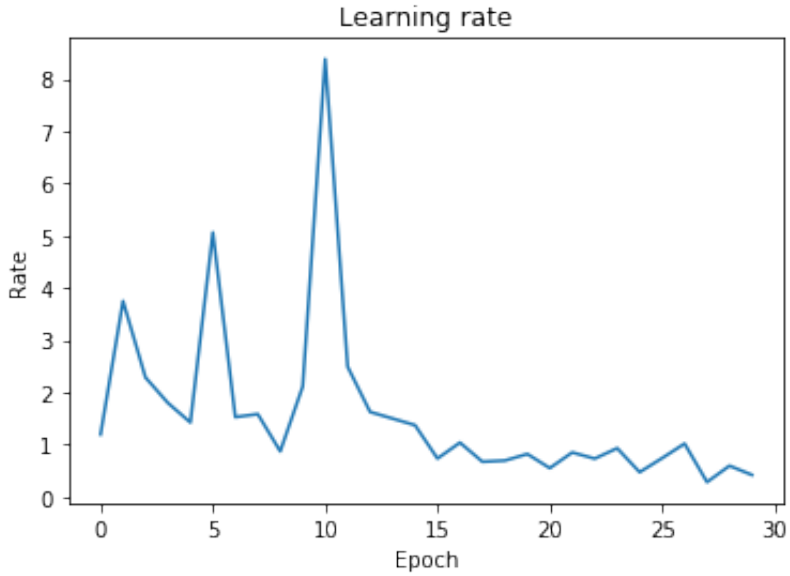
Plot for learning rate

In [40]:
```
model=neural_net_2hidden(False)
learning_rate_plot(24790,model)
```

```
Model: "sequential_11"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_24 (Dense)             (None, 30)                23550
_________________________________________________________________
dense_25 (Dense)             (None, 30)                930
_________________________________________________________________
dense_26 (Dense)             (None, 10)                310
=================================================================
```

```
Total params: 24,790
Trainable params: 24,790
Non-trainable params: 0
_________________________________________________________________
Train on 1000 samples
1000/1000 [==============================] - 1s 638us/sample - los
s: 1.5793 - accuracy: 0.4430
Train on 1000 samples
1000/1000 [==============================] - 0s 208us/sample - los
s: 1.0365 - accuracy: 0.6630
Train on 1000 samples
1000/1000 [==============================] - 0s 229us/sample - los
s: 0.9198 - accuracy: 0.6940
Train on 1000 samples
1000/1000 [==============================] - 0s 238us/sample - los
s: 0.8485 - accuracy: 0.7250
Train on 1000 samples
1000/1000 [==============================] - 0s 212us/sample - los
s: 0.8280 - accuracy: 0.7160
Train on 1000 samples
1000/1000 [==============================] - 0s 194us/sample - los
s: 0.7396 - accuracy: 0.7490
Train on 1000 samples
1000/1000 [==============================] - 0s 196us/sample - los
s: 0.7668 - accuracy: 0.7180
Train on 1000 samples
1000/1000 [==============================] - 0s 206us/sample - los
s: 0.8473 - accuracy: 0.7290
Train on 1000 samples
1000/1000 [==============================] - 0s 188us/sample - los
s: 0.7289 - accuracy: 0.7600
Train on 1000 samples
1000/1000 [==============================] - 0s 188us/sample - los
s: 0.7743 - accuracy: 0.7510
Train on 1000 samples
1000/1000 [==============================] - 0s 189us/sample - los
s: 0.8649 - accuracy: 0.7320
Train on 1000 samples
1000/1000 [==============================] - 0s 188us/sample - los
s: 0.7400 - accuracy: 0.7660
Train on 1000 samples
1000/1000 [==============================] - 0s 191us/sample - los
s: 0.7700 - accuracy: 0.7470
Train on 1000 samples
1000/1000 [==============================] - 0s 189us/sample - los
s: 0.7770 - accuracy: 0.7560
Train on 1000 samples
1000/1000 [==============================] - 0s 189us/sample - los
s: 0.7830 - accuracy: 0.7470
Train on 1000 samples
1000/1000 [==============================] - 0s 195us/sample - los
s: 0.7971 - accuracy: 0.7480
Train on 1000 samples
```

```
1000/1000 [==============================] - 0s 187us/sample - los
s: 0.7495 - accuracy: 0.7570
Train on 1000 samples
1000/1000 [==============================] - 0s 189us/sample - los
s: 0.7103 - accuracy: 0.7900
Train on 1000 samples
1000/1000 [==============================] - 0s 201us/sample - los
s: 0.7099 - accuracy: 0.7940
Train on 1000 samples
1000/1000 [==============================] - 0s 190us/sample - los
s: 0.6896 - accuracy: 0.7780
Train on 1000 samples
1000/1000 [==============================] - 0s 193us/sample - los
s: 0.6564 - accuracy: 0.7850
Train on 1000 samples
1000/1000 [==============================] - 0s 190us/sample - los
s: 0.7390 - accuracy: 0.7920
Train on 1000 samples
1000/1000 [==============================] - 0s 186us/sample - los
s: 0.6696 - accuracy: 0.8010
Train on 1000 samples
1000/1000 [==============================] - 0s 189us/sample - los
s: 0.6313 - accuracy: 0.8160
Train on 1000 samples
1000/1000 [==============================] - 0s 193us/sample - los
s: 0.6275 - accuracy: 0.8160
Train on 1000 samples
1000/1000 [==============================] - 0s 247us/sample - los
s: 0.6753 - accuracy: 0.7880
Train on 1000 samples
1000/1000 [==============================] - 0s 216us/sample - los
s: 0.7381 - accuracy: 0.7760
Train on 1000 samples
1000/1000 [==============================] - 0s 250us/sample - los
s: 0.6682 - accuracy: 0.7990
Train on 1000 samples
1000/1000 [==============================] - 0s 223us/sample - los
s: 0.6817 - accuracy: 0.7940
Train on 1000 samples
1000/1000 [==============================] - 0s 248us/sample - los
s: 0.5951 - accuracy: 0.8070
```



Two hidden layers with regularization

In [41]:
```
model=neural_net_2hidden(True)
history = model.fit(mini_train, train_labels, epochs=30, batch_size
=10,validation_data=(mini_test,test_labels))
```

```
Model: "sequential_12"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_27 (Dense)             (None, 30)                23550
_________________________________________________________________
dense_28 (Dense)             (None, 30)                930
_________________________________________________________________
dense_29 (Dense)             (None, 10)                310
=================================================================
Total params: 24,790
Trainable params: 24,790
Non-trainable params: 0
_________________________________________________________________
Train on 1000 samples, validate on 1000 samples
Epoch 1/30
1000/1000 [==============================] - 1s 886us/sample - los
s: 35.4528 - accuracy: 0.0970 - val_loss: 2.3400 - val_accuracy: 0
.1000
Epoch 2/30
1000/1000 [==============================] - 0s 380us/sample - los
s: 2.3353 - accuracy: 0.1010 - val_loss: 2.3166 - val_accuracy: 0.
1000
Epoch 3/30
1000/1000 [==============================] - 0s 438us/sample - los
s: 2.3390 - accuracy: 0.1040 - val_loss: 2.3211 - val_accuracy: 0.
1000
Epoch 4/30
```

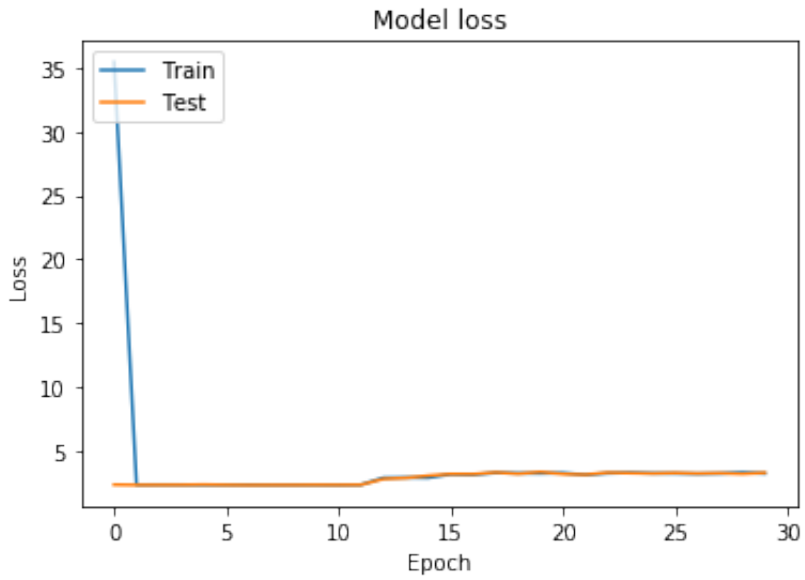
```
1000/1000 [==============================] - 0s 418us/sample - los
s: 2.3368 - accuracy: 0.0850 - val_loss: 2.3255 - val_accuracy: 0.
1000
Epoch 5/30
1000/1000 [==============================] - 0s 380us/sample - los
s: 2.3283 - accuracy: 0.1210 - val_loss: 2.3461 - val_accuracy: 0.
1000
Epoch 6/30
1000/1000 [==============================] - 0s 348us/sample - los
s: 2.3304 - accuracy: 0.0960 - val_loss: 2.3214 - val_accuracy: 0.
1000
Epoch 7/30
1000/1000 [==============================] - 0s 417us/sample - los
s: 2.3339 - accuracy: 0.0820 - val_loss: 2.3073 - val_accuracy: 0.
1000
Epoch 8/30
1000/1000 [==============================] - 0s 387us/sample - los
s: 2.3242 - accuracy: 0.0900 - val_loss: 2.3154 - val_accuracy: 0.
1000
Epoch 9/30
1000/1000 [==============================] - 0s 423us/sample - los
s: 2.3284 - accuracy: 0.1040 - val_loss: 2.3150 - val_accuracy: 0.
1000
Epoch 10/30
1000/1000 [==============================] - 0s 348us/sample - los
s: 2.3276 - accuracy: 0.1070 - val_loss: 2.3255 - val_accuracy: 0.
1000
Epoch 11/30
1000/1000 [==============================] - 0s 335us/sample - los
s: 2.3235 - accuracy: 0.0970 - val_loss: 2.3189 - val_accuracy: 0.
1000
Epoch 12/30
1000/1000 [==============================] - 0s 386us/sample - los
s: 2.3323 - accuracy: 0.0860 - val_loss: 2.3299 - val_accuracy: 0.
1000
Epoch 13/30
1000/1000 [==============================] - 0s 330us/sample - los
s: 2.8774 - accuracy: 0.1000 - val_loss: 2.7886 - val_accuracy: 0.
1000
Epoch 14/30
1000/1000 [==============================] - 0s 410us/sample - los
s: 2.8972 - accuracy: 0.1100 - val_loss: 2.8664 - val_accuracy: 0.
1000
Epoch 15/30
1000/1000 [==============================] - 0s 417us/sample - los
s: 2.8798 - accuracy: 0.0860 - val_loss: 3.0443 - val_accuracy: 0.
1000
Epoch 16/30
1000/1000 [==============================] - 0s 326us/sample - los
s: 3.1523 - accuracy: 0.0940 - val_loss: 3.1739 - val_accuracy: 0.
1000
Epoch 17/30
1000/1000 [==============================] - 0s 331us/sample - los
```

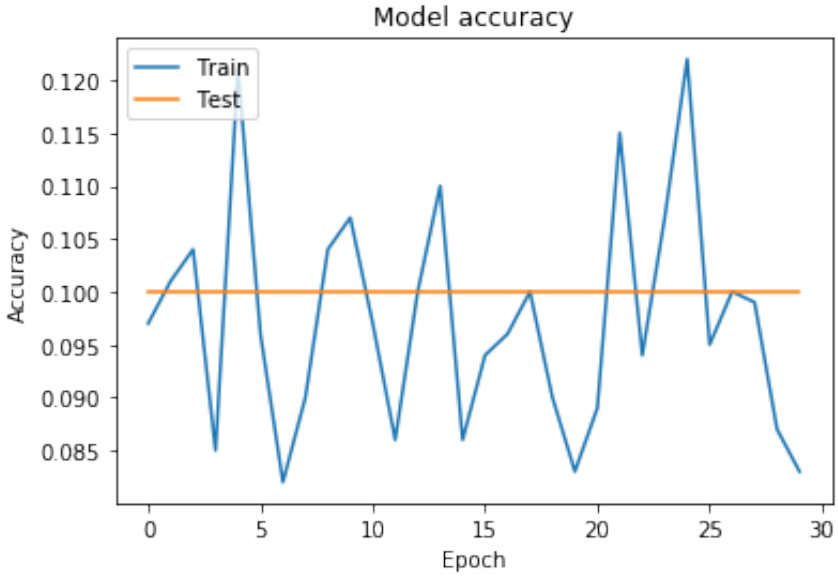
```
s: 3.1223 - accuracy: 0.0960 - val_loss: 3.1636 - val_accuracy: 0.
1000
Epoch 18/30
1000/1000 [==============================] - 0s 338us/sample - los
s: 3.2697 - accuracy: 0.1000 - val_loss: 3.3040 - val_accuracy: 0.
1000
Epoch 19/30
1000/1000 [==============================] - 0s 324us/sample - los
s: 3.2387 - accuracy: 0.0900 - val_loss: 3.1925 - val_accuracy: 0.
1000
Epoch 20/30
1000/1000 [==============================] - 0s 325us/sample - los
s: 3.2253 - accuracy: 0.0830 - val_loss: 3.3379 - val_accuracy: 0.
1000
Epoch 21/30
1000/1000 [==============================] - 0s 322us/sample - los
s: 3.2587 - accuracy: 0.0890 - val_loss: 3.1757 - val_accuracy: 0.
1000
Epoch 22/30
1000/1000 [==============================] - 0s 324us/sample - los
s: 3.1113 - accuracy: 0.1150 - val_loss: 3.1259 - val_accuracy: 0.
1000
Epoch 23/30
1000/1000 [==============================] - 0s 364us/sample - los
s: 3.2360 - accuracy: 0.0940 - val_loss: 3.2885 - val_accuracy: 0.
1000
Epoch 24/30
1000/1000 [==============================] - 0s 402us/sample - los
s: 3.2744 - accuracy: 0.1070 - val_loss: 3.2503 - val_accuracy: 0.
1000
Epoch 25/30
1000/1000 [==============================] - 0s 336us/sample - los
s: 3.2355 - accuracy: 0.1220 - val_loss: 3.2200 - val_accuracy: 0.
1000
Epoch 26/30
1000/1000 [==============================] - 0s 336us/sample - los
s: 3.2211 - accuracy: 0.0950 - val_loss: 3.2609 - val_accuracy: 0.
1000
Epoch 27/30
1000/1000 [==============================] - 0s 334us/sample - los
s: 3.1975 - accuracy: 0.1000 - val_loss: 3.2188 - val_accuracy: 0.
1000
Epoch 28/30
1000/1000 [==============================] - 0s 324us/sample - los
s: 3.2180 - accuracy: 0.0990 - val_loss: 3.2444 - val_accuracy: 0.
1000
Epoch 29/30
1000/1000 [==============================] - 0s 327us/sample - los
s: 3.2976 - accuracy: 0.0870 - val_loss: 3.1887 - val_accuracy: 0.
1000
Epoch 30/30
1000/1000 [==============================] - 0s 331us/sample - los
s: 3.2200 - accuracy: 0.0830 - val_loss: 3.2762 - val_accuracy: 0.
```

```
1000
```

Plot for criterion function loss

In [42]:
```
model_training_plot()
```





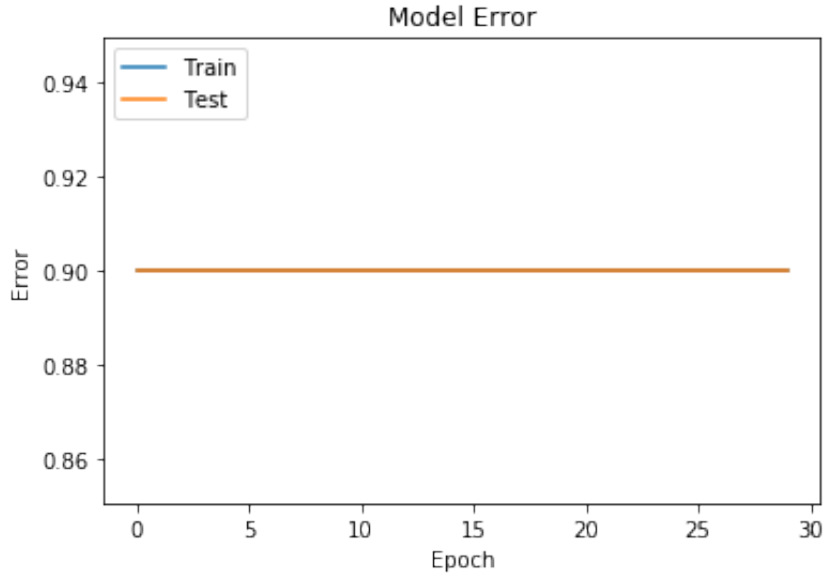
Plot for zero-one error

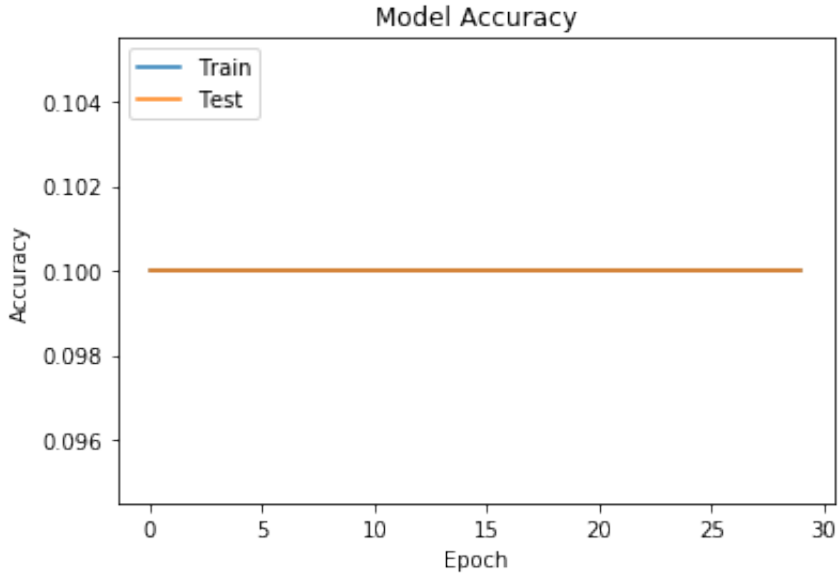
In [43]:
```
model=neural_net_2hidden(True)
traine,teste,traina,testa=for_error_plot(model)
zero_one_error_plot(traine,teste,traina,testa)
```

```
Model: "sequential_13"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
```

```
dense_30 (Dense)             (None, 30)                23550     
_________________________________________________________________
dense_31 (Dense)             (None, 30)                930       
_________________________________________________________________
dense_32 (Dense)             (None, 10)                310       
=================================================================
Total params: 24,790
Trainable params: 24,790
Non-trainable params: 0
_________________________________________________________________
Train on 1000 samples
1000/1000 [==============================] - 1s 665us/sample - los
s: 35.3744 - accuracy: 0.1210
Train on 1000 samples
1000/1000 [==============================] - 0s 225us/sample - los
s: 2.3323 - accuracy: 0.0900
Train on 1000 samples
1000/1000 [==============================] - 0s 237us/sample - los
s: 2.3320 - accuracy: 0.0820
Train on 1000 samples
1000/1000 [==============================] - 0s 236us/sample - los
s: 2.3367 - accuracy: 0.0850
Train on 1000 samples
1000/1000 [==============================] - 0s 251us/sample - los
s: 2.3323 - accuracy: 0.0870
Train on 1000 samples
1000/1000 [==============================] - 0s 208us/sample - los
s: 2.3313 - accuracy: 0.0930
Train on 1000 samples
1000/1000 [==============================] - 0s 200us/sample - los
s: 2.3274 - accuracy: 0.0960
Train on 1000 samples
1000/1000 [==============================] - 0s 193us/sample - los
s: 2.3274 - accuracy: 0.0940
Train on 1000 samples
1000/1000 [==============================] - 0s 195us/sample - los
s: 2.3275 - accuracy: 0.0980
Train on 1000 samples
1000/1000 [==============================] - 0s 203us/sample - los
s: 2.3219 - accuracy: 0.0950
Train on 1000 samples
1000/1000 [==============================] - 0s 200us/sample - los
s: 2.3320 - accuracy: 0.0890
Train on 1000 samples
1000/1000 [==============================] - 0s 196us/sample - los
s: 2.3270 - accuracy: 0.0840
Train on 1000 samples
1000/1000 [==============================] - 0s 196us/sample - los
s: 2.8895 - accuracy: 0.0970
Train on 1000 samples
1000/1000 [==============================] - 0s 199us/sample - los
s: 2.9074 - accuracy: 0.0870
Train on 1000 samples
```

```
1000/1000 [==============================] - 0s 195us/sample - los
s: 2.8571 - accuracy: 0.0920
Train on 1000 samples
1000/1000 [==============================] - 0s 199us/sample - los
s: 3.1341 - accuracy: 0.1060
Train on 1000 samples
1000/1000 [==============================] - 0s 203us/sample - los
s: 3.1350 - accuracy: 0.0900
Train on 1000 samples
1000/1000 [==============================] - 0s 193us/sample - los
s: 3.2785 - accuracy: 0.1000
Train on 1000 samples
1000/1000 [==============================] - 0s 194us/sample - los
s: 3.2268 - accuracy: 0.1010
Train on 1000 samples
1000/1000 [==============================] - 0s 194us/sample - los
s: 3.2006 - accuracy: 0.0960
Train on 1000 samples
1000/1000 [==============================] - 0s 197us/sample - los
s: 3.2948 - accuracy: 0.0890
Train on 1000 samples
1000/1000 [==============================] - 0s 197us/sample - los
s: 3.1292 - accuracy: 0.0840
Train on 1000 samples
1000/1000 [==============================] - 0s 194us/sample - los
s: 3.2618 - accuracy: 0.0920
Train on 1000 samples
1000/1000 [==============================] - 0s 196us/sample - los
s: 3.2727 - accuracy: 0.1060
Train on 1000 samples
1000/1000 [==============================] - 0s 195us/sample - los
s: 3.2283 - accuracy: 0.0890
Train on 1000 samples
1000/1000 [==============================] - 0s 196us/sample - los
s: 3.2242 - accuracy: 0.0990
Train on 1000 samples
1000/1000 [==============================] - 0s 209us/sample - los
s: 3.1738 - accuracy: 0.0950
Train on 1000 samples
1000/1000 [==============================] - 0s 279us/sample - los
s: 3.2659 - accuracy: 0.0820
Train on 1000 samples
1000/1000 [==============================] - 0s 227us/sample - los
s: 3.2749 - accuracy: 0.0780
Train on 1000 samples
1000/1000 [==============================] - 0s 198us/sample - los
s: 3.2193 - accuracy: 0.0930
```





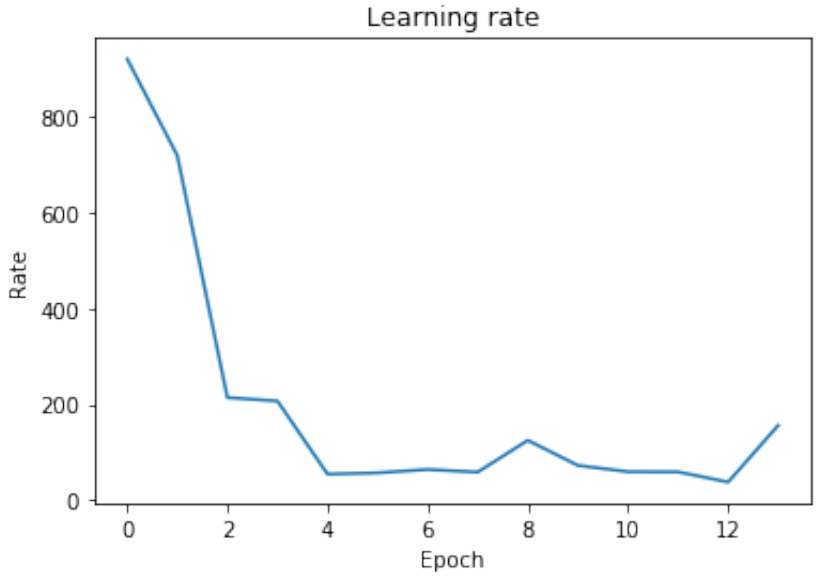
Plot for learning rate

In [44]:
```
model=neural_net_2hidden(True)
learning_rate_plot(24790,model)
```

```
Model: "sequential_14"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_33 (Dense)             (None, 30)                23550
_________________________________________________________________
dense_34 (Dense)             (None, 30)                930
_________________________________________________________________
dense_35 (Dense)             (None, 10)                310
=================================================================
Total params: 24,790
Trainable params: 24,790
```

```
Non-trainable params: 0
_________________________________________________________________
Train on 1000 samples
1000/1000 [==============================] - 1s 637us/sample - los
s: 35.4035 - accuracy: 0.1010
Train on 1000 samples
1000/1000 [==============================] - 0s 195us/sample - los
s: 2.3344 - accuracy: 0.0970
Train on 1000 samples
1000/1000 [==============================] - 0s 195us/sample - los
s: 2.3330 - accuracy: 0.1030
Train on 1000 samples
1000/1000 [==============================] - 0s 198us/sample - los
s: 2.3329 - accuracy: 0.0750
Train on 1000 samples
1000/1000 [==============================] - 0s 237us/sample - los
s: 2.3275 - accuracy: 0.1060
Train on 1000 samples
1000/1000 [==============================] - 0s 215us/sample - los
s: 2.3257 - accuracy: 0.1010
Train on 1000 samples
1000/1000 [==============================] - 0s 237us/sample - los
s: 2.3267 - accuracy: 0.0920
Train on 1000 samples
1000/1000 [==============================] - 0s 238us/sample - los
s: 2.3286 - accuracy: 0.0910
Train on 1000 samples
1000/1000 [==============================] - 0s 190us/sample - los
s: 2.3293 - accuracy: 0.1010
Train on 1000 samples
1000/1000 [==============================] - 0s 189us/sample - los
s: 2.3273 - accuracy: 0.1020
Train on 1000 samples
1000/1000 [==============================] - 0s 201us/sample - los
s: 2.3290 - accuracy: 0.0950
Train on 1000 samples
1000/1000 [==============================] - 0s 188us/sample - los
s: 2.3300 - accuracy: 0.1010
Train on 1000 samples
1000/1000 [==============================] - 0s 186us/sample - los
s: 2.8664 - accuracy: 0.1010
Train on 1000 samples
1000/1000 [==============================] - 0s 187us/sample - los
s: 2.8807 - accuracy: 0.0910
Train on 1000 samples
1000/1000 [==============================] - 0s 187us/sample - los
s: 2.8765 - accuracy: 0.0910
Train on 1000 samples
1000/1000 [==============================] - 0s 185us/sample - los
s: 3.1454 - accuracy: 0.0850
Train on 1000 samples
1000/1000 [==============================] - 0s 187us/sample - los
s: 3.1379 - accuracy: 0.0820
```

```
Train on 1000 samples
1000/1000 [==============================] - 0s 187us/sample - los
s: 3.2949 - accuracy: 0.0850
Train on 1000 samples
1000/1000 [==============================] - 0s 205us/sample - los
s: 3.1835 - accuracy: 0.1110
Train on 1000 samples
1000/1000 [==============================] - 0s 196us/sample - los
s: 3.2446 - accuracy: 0.0940
Train on 1000 samples
1000/1000 [==============================] - 0s 201us/sample - los
s: 3.2777 - accuracy: 0.1080
Train on 1000 samples
1000/1000 [==============================] - 0s 242us/sample - los
s: 3.0962 - accuracy: 0.0820
Train on 1000 samples
1000/1000 [==============================] - 0s 239us/sample - los
s: 3.2774 - accuracy: 0.0750
Train on 1000 samples
1000/1000 [==============================] - 0s 236us/sample - los
s: 3.2792 - accuracy: 0.0900
Train on 1000 samples
1000/1000 [==============================] - 0s 232us/sample - los
s: 3.2315 - accuracy: 0.0980
Train on 1000 samples
1000/1000 [==============================] - 0s 250us/sample - los
s: 3.2029 - accuracy: 0.1020
Train on 1000 samples
1000/1000 [==============================] - 0s 203us/sample - los
s: 3.2055 - accuracy: 0.0930
Train on 1000 samples
1000/1000 [==============================] - 0s 195us/sample - los
s: 3.2551 - accuracy: 0.0840
Train on 1000 samples
1000/1000 [==============================] - 0s 204us/sample - los
s: 3.2564 - accuracy: 0.0930
Train on 1000 samples
1000/1000 [==============================] - 0s 195us/sample - los
s: 3.2150 - accuracy: 0.0910
```



In [45]:
```
def neural_net_3hidden(l2):

    model = tf.keras.Sequential()
    if l2==True:
        model.add(tf.keras.layers.Dense(30, input_dim=784,activatio
n='sigmoid',kernel_regularizer=tf.keras.regularizers.l2(5)))
        model.add(tf.keras.layers.Dense(30, input_dim=30,activation
='sigmoid',kernel_regularizer=tf.keras.regularizers.l2(5)))
        model.add(tf.keras.layers.Dense(30, input_dim=30,activation
='sigmoid',kernel_regularizer=tf.keras.regularizers.l2(5)))

    else:
        model.add(tf.keras.layers.Dense(30, input_dim=784,activatio
n='sigmoid'))
        model.add(tf.keras.layers.Dense(30, input_dim=30,activation
='sigmoid'))
        model.add(tf.keras.layers.Dense(30, input_dim=30,activation
='sigmoid'))

    model.add(tf.keras.layers.Dense(10, activation='softmax'))

    model.summary()

    adam=tf.keras.optimizers.Adam(learning_rate=0.1)
    model.compile(loss='categorical_crossentropy', optimizer=adam,
metrics=['accuracy'])

    return model
```

3 hidden layers without regularization

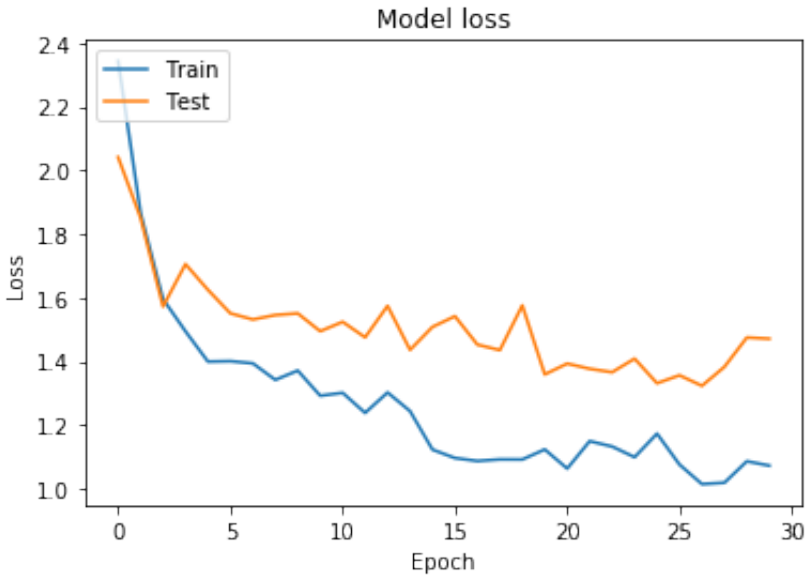
Plot for Criterion Function Loss

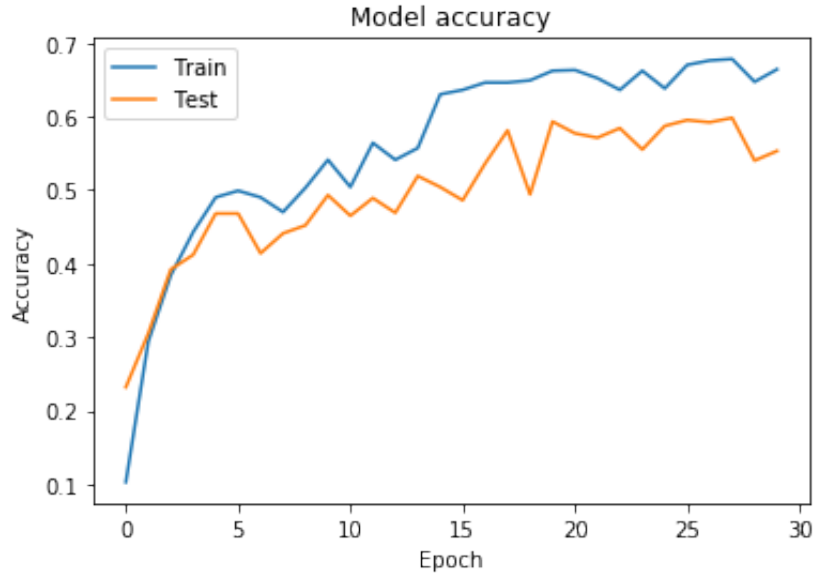
In [46]:
```
model=neural_net_3hidden(False)
history = model.fit(mini_train, train_labels, epochs=30, batch_size
=10,validation_data=(mini_test,test_labels))
model_training_plot()
```

```
Model: "sequential_15"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_36 (Dense)             (None, 30)                23550
_________________________________________________________________
dense_37 (Dense)             (None, 30)                930
_________________________________________________________________
dense_38 (Dense)             (None, 30)                930
_________________________________________________________________
dense_39 (Dense)             (None, 10)                310
=================================================================
Total params: 25,720
Trainable params: 25,720
Non-trainable params: 0
_________________________________________________________________
Train on 1000 samples, validate on 1000 samples
Epoch 1/30
1000/1000 [==============================] - 1s 883us/sample - los
s: 2.3430 - accuracy: 0.1030 - val_loss: 2.0410 - val_accuracy: 0.
2320
Epoch 2/30
1000/1000 [==============================] - 0s 341us/sample - los
s: 1.8662 - accuracy: 0.2940 - val_loss: 1.8509 - val_accuracy: 0.
3050
Epoch 3/30
1000/1000 [==============================] - 0s 337us/sample - los
s: 1.5957 - accuracy: 0.3840 - val_loss: 1.5722 - val_accuracy: 0.
3920
Epoch 4/30
1000/1000 [==============================] - 0s 325us/sample - los
s: 1.4944 - accuracy: 0.4430 - val_loss: 1.7053 - val_accuracy: 0.
4120
Epoch 5/30
1000/1000 [==============================] - 0s 334us/sample - los
s: 1.3993 - accuracy: 0.4900 - val_loss: 1.6245 - val_accuracy: 0.
4680
Epoch 6/30
1000/1000 [==============================] - 0s 323us/sample - los
s: 1.4007 - accuracy: 0.4990 - val_loss: 1.5509 - val_accuracy: 0.
4680
Epoch 7/30
1000/1000 [==============================] - 0s 322us/sample - los
s: 1.3939 - accuracy: 0.4900 - val_loss: 1.5316 - val_accuracy: 0.
4140
Epoch 8/30
1000/1000 [==============================] - 0s 322us/sample - los
s: 1.3420 - accuracy: 0.4700 - val_loss: 1.5455 - val_accuracy: 0.
```

```
4410
Epoch 9/30
1000/1000 [==============================] - 0s 321us/sample - los
s: 1.3711 - accuracy: 0.5030 - val_loss: 1.5507 - val_accuracy: 0.
4520
Epoch 10/30
1000/1000 [==============================] - 0s 328us/sample - los
s: 1.2922 - accuracy: 0.5410 - val_loss: 1.4947 - val_accuracy: 0.
4930
Epoch 11/30
1000/1000 [==============================] - 0s 322us/sample - los
s: 1.3013 - accuracy: 0.5040 - val_loss: 1.5241 - val_accuracy: 0.
4650
Epoch 12/30
1000/1000 [==============================] - 0s 333us/sample - los
s: 1.2383 - accuracy: 0.5640 - val_loss: 1.4754 - val_accuracy: 0.
4890
Epoch 13/30
1000/1000 [==============================] - 0s 322us/sample - los
s: 1.3024 - accuracy: 0.5410 - val_loss: 1.5744 - val_accuracy: 0.
4690
Epoch 14/30
1000/1000 [==============================] - 0s 327us/sample - los
s: 1.2440 - accuracy: 0.5570 - val_loss: 1.4361 - val_accuracy: 0.
5190
Epoch 15/30
1000/1000 [==============================] - 0s 345us/sample - los
s: 1.1231 - accuracy: 0.6300 - val_loss: 1.5081 - val_accuracy: 0.
5040
Epoch 16/30
1000/1000 [==============================] - 0s 328us/sample - los
s: 1.0965 - accuracy: 0.6360 - val_loss: 1.5418 - val_accuracy: 0.
4860
Epoch 17/30
1000/1000 [==============================] - 0s 325us/sample - los
s: 1.0873 - accuracy: 0.6460 - val_loss: 1.4526 - val_accuracy: 0.
5360
Epoch 18/30
1000/1000 [==============================] - 0s 328us/sample - los
s: 1.0923 - accuracy: 0.6460 - val_loss: 1.4353 - val_accuracy: 0.
5810
Epoch 19/30
1000/1000 [==============================] - 0s 414us/sample - los
s: 1.0920 - accuracy: 0.6490 - val_loss: 1.5757 - val_accuracy: 0.
4940
Epoch 20/30
1000/1000 [==============================] - 0s 378us/sample - los
s: 1.1236 - accuracy: 0.6620 - val_loss: 1.3591 - val_accuracy: 0.
5930
Epoch 21/30
1000/1000 [==============================] - 0s 325us/sample - los
s: 1.0635 - accuracy: 0.6630 - val_loss: 1.3930 - val_accuracy: 0.
5770
```

```
Epoch 22/30
1000/1000 [==============================] - 0s 347us/sample - los
s: 1.1496 - accuracy: 0.6520 - val_loss: 1.3767 - val_accuracy: 0.
5710
Epoch 23/30
1000/1000 [==============================] - 0s 330us/sample - los
s: 1.1327 - accuracy: 0.6360 - val_loss: 1.3660 - val_accuracy: 0.
5840
Epoch 24/30
1000/1000 [==============================] - 0s 453us/sample - los
s: 1.0993 - accuracy: 0.6620 - val_loss: 1.4085 - val_accuracy: 0.
5550
Epoch 25/30
1000/1000 [==============================] - 0s 448us/sample - los
s: 1.1731 - accuracy: 0.6380 - val_loss: 1.3314 - val_accuracy: 0.
5870
Epoch 26/30
1000/1000 [==============================] - 0s 460us/sample - los
s: 1.0761 - accuracy: 0.6700 - val_loss: 1.3563 - val_accuracy: 0.
5950
Epoch 27/30
1000/1000 [==============================] - 0s 381us/sample - los
s: 1.0148 - accuracy: 0.6760 - val_loss: 1.3234 - val_accuracy: 0.
5920
Epoch 28/30
1000/1000 [==============================] - 0s 362us/sample - los
s: 1.0191 - accuracy: 0.6780 - val_loss: 1.3827 - val_accuracy: 0.
5980
Epoch 29/30
1000/1000 [==============================] - 0s 425us/sample - los
s: 1.0864 - accuracy: 0.6470 - val_loss: 1.4749 - val_accuracy: 0.
5400
Epoch 30/30
1000/1000 [==============================] - 0s 372us/sample - los
s: 1.0728 - accuracy: 0.6640 - val_loss: 1.4711 - val_accuracy: 0.
5530
```





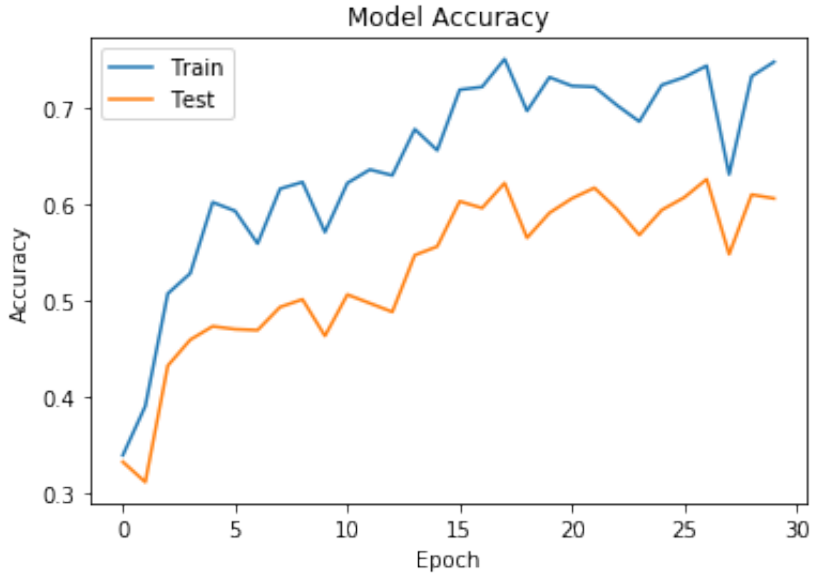
Plot for zero-one error

In [47]:
```
model=neural_net_3hidden(False)
traine,teste,traina,testa=for_error_plot(model)
zero_one_error_plot(traine,teste,traina,testa)
```

```
Model: "sequential_16"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_40 (Dense)             (None, 30)                23550
_________________________________________________________________
dense_41 (Dense)             (None, 30)                930
_________________________________________________________________
dense_42 (Dense)             (None, 30)                930
_________________________________________________________________
dense_43 (Dense)             (None, 10)                310
=================================================================
Total params: 25,720
Trainable params: 25,720
Non-trainable params: 0
_________________________________________________________________
Train on 1000 samples
1000/1000 [==============================] - 1s 631us/sample - los
s: 2.1298 - accuracy: 0.1890
Train on 1000 samples
1000/1000 [==============================] - 0s 201us/sample - los
s: 1.5915 - accuracy: 0.3680
Train on 1000 samples
1000/1000 [==============================] - 0s 204us/sample - los
s: 1.4128 - accuracy: 0.4750
Train on 1000 samples
1000/1000 [==============================] - 0s 261us/sample - los
s: 1.3010 - accuracy: 0.5070
```

```
Train on 1000 samples
1000/1000 [==============================] - 0s 222us/sample - los
s: 1.2865 - accuracy: 0.5560
Train on 1000 samples
1000/1000 [==============================] - 0s 199us/sample - los
s: 1.2003 - accuracy: 0.5720
Train on 1000 samples
1000/1000 [==============================] - 0s 211us/sample - los
s: 1.1844 - accuracy: 0.5950
Train on 1000 samples
1000/1000 [==============================] - 0s 202us/sample - los
s: 1.2122 - accuracy: 0.5820
Train on 1000 samples
1000/1000 [==============================] - 0s 202us/sample - los
s: 1.2061 - accuracy: 0.5990
Train on 1000 samples
1000/1000 [==============================] - 0s 200us/sample - los
s: 1.2653 - accuracy: 0.5750
Train on 1000 samples
1000/1000 [==============================] - 0s 198us/sample - los
s: 1.2113 - accuracy: 0.5790
Train on 1000 samples
1000/1000 [==============================] - 0s 199us/sample - los
s: 1.1095 - accuracy: 0.6410
Train on 1000 samples
1000/1000 [==============================] - 0s 370us/sample - los
s: 1.0905 - accuracy: 0.6270
Train on 1000 samples
1000/1000 [==============================] - 0s 198us/sample - los
s: 1.0379 - accuracy: 0.6690
Train on 1000 samples
1000/1000 [==============================] - 0s 207us/sample - los
s: 1.0170 - accuracy: 0.6720
Train on 1000 samples
1000/1000 [==============================] - 0s 199us/sample - los
s: 1.0812 - accuracy: 0.6500
Train on 1000 samples
1000/1000 [==============================] - 0s 208us/sample - los
s: 1.0299 - accuracy: 0.6700
Train on 1000 samples
1000/1000 [==============================] - 0s 221us/sample - los
s: 1.0339 - accuracy: 0.6890
Train on 1000 samples
1000/1000 [==============================] - 0s 202us/sample - los
s: 1.0752 - accuracy: 0.6800
Train on 1000 samples
1000/1000 [==============================] - 0s 282us/sample - los
s: 1.0100 - accuracy: 0.7100
Train on 1000 samples
1000/1000 [==============================] - 0s 205us/sample - los
s: 0.9543 - accuracy: 0.7100
Train on 1000 samples
1000/1000 [==============================] - 0s 220us/sample - los
```

```
s: 0.9570 - accuracy: 0.6930
Train on 1000 samples
1000/1000 [==============================] - 0s 221us/sample - los
s: 0.9696 - accuracy: 0.7160
Train on 1000 samples
1000/1000 [==============================] - 0s 203us/sample - los
s: 0.9250 - accuracy: 0.7280
Train on 1000 samples
1000/1000 [==============================] - 0s 251us/sample - los
s: 0.9687 - accuracy: 0.7160
Train on 1000 samples
1000/1000 [==============================] - 0s 206us/sample - los
s: 1.0204 - accuracy: 0.6930
Train on 1000 samples
1000/1000 [==============================] - 0s 229us/sample - los
s: 0.9849 - accuracy: 0.7180
Train on 1000 samples
1000/1000 [==============================] - 0s 198us/sample - los
s: 0.9518 - accuracy: 0.7130
Train on 1000 samples
1000/1000 [==============================] - 0s 205us/sample - los
s: 1.0698 - accuracy: 0.6690
Train on 1000 samples
1000/1000 [==============================] - 0s 229us/sample - los
s: 0.9762 - accuracy: 0.7100
```



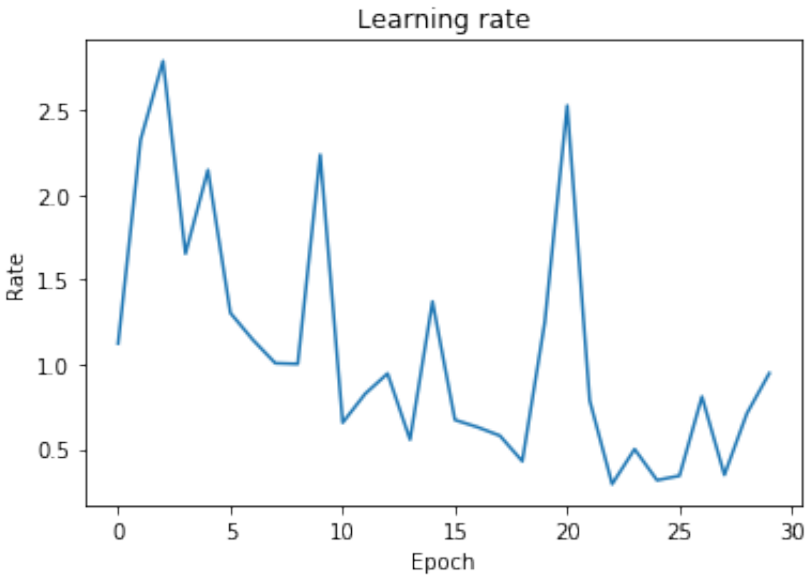
Plot for learning rate

In [48]:
```
model=neural_net_3hidden(False)
learning_rate_plot(25720,model)
```

```
Model: "sequential_17"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_44 (Dense)             (None, 30)                23550
_________________________________________________________________
dense_45 (Dense)             (None, 30)                930
_________________________________________________________________
dense_46 (Dense)             (None, 30)                930
_________________________________________________________________
dense_47 (Dense)             (None, 10)                310
=================================================================
Total params: 25,720
Trainable params: 25,720
Non-trainable params: 0
_________________________________________________________________
Train on 1000 samples
1000/1000 [==============================] - 1s 647us/sample - los
s: 2.1075 - accuracy: 0.2030
Train on 1000 samples
1000/1000 [==============================] - 0s 228us/sample - los
s: 1.5866 - accuracy: 0.3830
Train on 1000 samples
1000/1000 [==============================] - 0s 200us/sample - los
s: 1.4277 - accuracy: 0.4840
Train on 1000 samples
1000/1000 [==============================] - 0s 210us/sample - los
s: 1.3228 - accuracy: 0.5160
Train on 1000 samples
```

```
1000/1000 [==============================] - 0s 198us/sample - los
s: 1.2775 - accuracy: 0.5270
Train on 1000 samples
1000/1000 [==============================] - 0s 226us/sample - los
s: 1.3131 - accuracy: 0.5260
Train on 1000 samples
1000/1000 [==============================] - 0s 250us/sample - los
s: 1.2109 - accuracy: 0.5840
Train on 1000 samples
1000/1000 [==============================] - 0s 248us/sample - los
s: 1.1113 - accuracy: 0.6450
Train on 1000 samples
1000/1000 [==============================] - 0s 278us/sample - los
s: 1.1134 - accuracy: 0.6090
Train on 1000 samples
1000/1000 [==============================] - 0s 246us/sample - los
s: 1.2456 - accuracy: 0.5790
Train on 1000 samples
1000/1000 [==============================] - 0s 263us/sample - los
s: 1.1719 - accuracy: 0.5980
Train on 1000 samples
1000/1000 [==============================] - 0s 262us/sample - los
s: 1.1489 - accuracy: 0.6250
Train on 1000 samples
1000/1000 [==============================] - 0s 266us/sample - los
s: 1.1796 - accuracy: 0.6010
Train on 1000 samples
1000/1000 [==============================] - 0s 223us/sample - los
s: 1.1008 - accuracy: 0.6580
Train on 1000 samples
1000/1000 [==============================] - 0s 258us/sample - los
s: 1.1574 - accuracy: 0.6280
Train on 1000 samples
1000/1000 [==============================] - 0s 256us/sample - los
s: 1.1407 - accuracy: 0.6300
Train on 1000 samples
1000/1000 [==============================] - 0s 240us/sample - los
s: 1.0512 - accuracy: 0.6790
Train on 1000 samples
1000/1000 [==============================] - 0s 237us/sample - los
s: 0.9877 - accuracy: 0.6960
Train on 1000 samples
1000/1000 [==============================] - 0s 204us/sample - los
s: 0.9734 - accuracy: 0.6920
Train on 1000 samples
1000/1000 [==============================] - 0s 235us/sample - los
s: 1.0383 - accuracy: 0.6700
Train on 1000 samples
1000/1000 [==============================] - 0s 203us/sample - los
s: 1.0989 - accuracy: 0.6620
Train on 1000 samples
1000/1000 [==============================] - 0s 200us/sample - los
s: 1.1810 - accuracy: 0.6100
```

```
Train on 1000 samples
1000/1000 [==============================] - 0s 202us/sample - los
s: 1.0507 - accuracy: 0.6680
Train on 1000 samples
1000/1000 [==============================] - 0s 212us/sample - los
s: 1.0581 - accuracy: 0.6760
Train on 1000 samples
1000/1000 [==============================] - 0s 257us/sample - los
s: 1.0162 - accuracy: 0.6810
Train on 1000 samples
1000/1000 [==============================] - 0s 248us/sample - los
s: 0.9451 - accuracy: 0.7000
Train on 1000 samples
1000/1000 [==============================] - 0s 245us/sample - los
s: 1.0069 - accuracy: 0.6870
Train on 1000 samples
1000/1000 [==============================] - 0s 246us/sample - los
s: 0.9982 - accuracy: 0.6950
Train on 1000 samples
1000/1000 [==============================] - 0s 263us/sample - los
s: 1.0097 - accuracy: 0.6760
Train on 1000 samples
1000/1000 [==============================] - 0s 235us/sample - los
s: 1.0494 - accuracy: 0.6540
```



3 hidden layers with regularization

In [49]:
```
model=neural_net_3hidden(True)
history = model.fit(mini_train, train_labels, epochs=30, batch_size
=10,validation_data=(mini_test,test_labels))
```

```
Model: "sequential_18"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
```

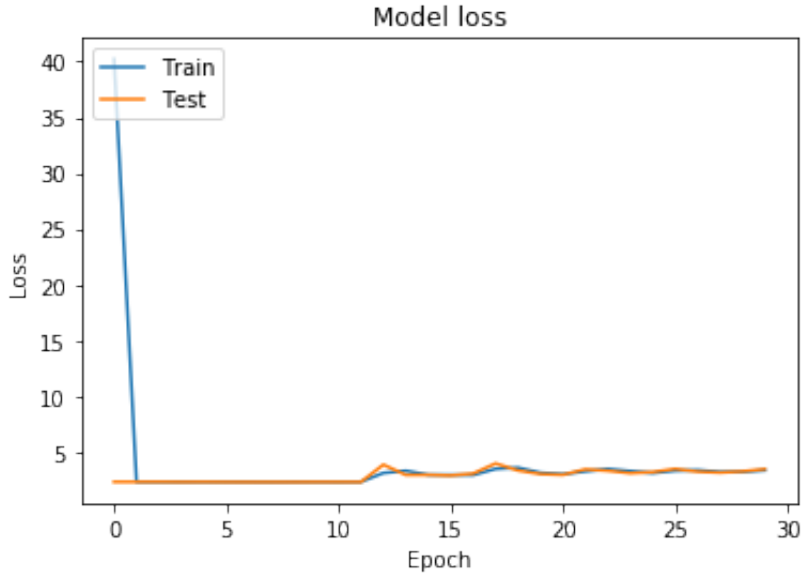
```
dense_48 (Dense)             (None, 30)                23550     
_________________________________________________________________
dense_49 (Dense)             (None, 30)                930       
_________________________________________________________________
dense_50 (Dense)             (None, 30)                930       
_________________________________________________________________
dense_51 (Dense)             (None, 10)                310       
=================================================================
Total params: 25,720
Trainable params: 25,720
Non-trainable params: 0
_________________________________________________________________
Train on 1000 samples, validate on 1000 samples
Epoch 1/30
1000/1000 [==============================] - 1s 1ms/sample - loss:
40.2314 - accuracy: 0.0870 - val_loss: 2.3294 - val_accuracy: 0.10
00
Epoch 2/30
1000/1000 [==============================] - 0s 335us/sample - los
s: 2.3410 - accuracy: 0.0870 - val_loss: 2.3274 - val_accuracy: 0.
1000
Epoch 3/30
1000/1000 [==============================] - 0s 339us/sample - los
s: 2.3343 - accuracy: 0.0880 - val_loss: 2.3309 - val_accuracy: 0.
1000
Epoch 4/30
1000/1000 [==============================] - 0s 337us/sample - los
s: 2.3348 - accuracy: 0.0950 - val_loss: 2.3154 - val_accuracy: 0.
1000
Epoch 5/30
1000/1000 [==============================] - 0s 331us/sample - los
s: 2.3340 - accuracy: 0.1010 - val_loss: 2.3240 - val_accuracy: 0.
1000
Epoch 6/30
1000/1000 [==============================] - 0s 326us/sample - los
s: 2.3341 - accuracy: 0.0870 - val_loss: 2.3132 - val_accuracy: 0.
1000
Epoch 7/30
1000/1000 [==============================] - 0s 326us/sample - los
s: 2.3319 - accuracy: 0.0770 - val_loss: 2.3103 - val_accuracy: 0.
1000
Epoch 8/30
1000/1000 [==============================] - 0s 324us/sample - los
s: 2.3252 - accuracy: 0.0900 - val_loss: 2.3129 - val_accuracy: 0.
1000
Epoch 9/30
1000/1000 [==============================] - 0s 330us/sample - los
s: 2.3321 - accuracy: 0.0850 - val_loss: 2.3105 - val_accuracy: 0.
1000
Epoch 10/30
1000/1000 [==============================] - 0s 349us/sample - los
s: 2.3332 - accuracy: 0.1040 - val_loss: 2.3180 - val_accuracy: 0.
1000
```

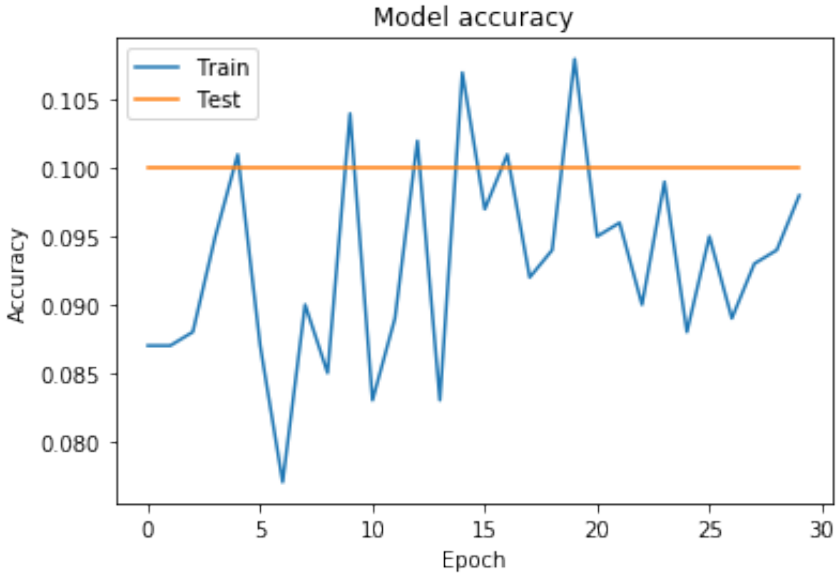
```
Epoch 11/30
1000/1000 [==============================] - 0s 353us/sample - los
s: 2.3272 - accuracy: 0.0830 - val_loss: 2.3172 - val_accuracy: 0.
1000
Epoch 12/30
1000/1000 [==============================] - 0s 341us/sample - los
s: 2.3325 - accuracy: 0.0890 - val_loss: 2.3182 - val_accuracy: 0.
1000
Epoch 13/30
1000/1000 [==============================] - 0s 442us/sample - los
s: 3.1256 - accuracy: 0.1020 - val_loss: 3.8704 - val_accuracy: 0.
1000
Epoch 14/30
1000/1000 [==============================] - 0s 434us/sample - los
s: 3.3208 - accuracy: 0.0830 - val_loss: 2.9531 - val_accuracy: 0.
1000
Epoch 15/30
1000/1000 [==============================] - 0s 418us/sample - los
s: 2.9627 - accuracy: 0.1070 - val_loss: 2.9554 - val_accuracy: 0.
1000
Epoch 16/30
1000/1000 [==============================] - 0s 426us/sample - los
s: 2.9534 - accuracy: 0.0970 - val_loss: 2.9018 - val_accuracy: 0.
1000
Epoch 17/30
1000/1000 [==============================] - 0s 432us/sample - los
s: 2.9412 - accuracy: 0.1010 - val_loss: 3.0722 - val_accuracy: 0.
1000
Epoch 18/30
1000/1000 [==============================] - 0s 333us/sample - los
s: 3.4950 - accuracy: 0.0920 - val_loss: 3.9782 - val_accuracy: 0.
1000
Epoch 19/30
1000/1000 [==============================] - 0s 344us/sample - los
s: 3.5994 - accuracy: 0.0940 - val_loss: 3.3505 - val_accuracy: 0.
1000
Epoch 20/30
1000/1000 [==============================] - 0s 343us/sample - los
s: 3.1450 - accuracy: 0.1080 - val_loss: 3.0348 - val_accuracy: 0.
1000
Epoch 21/30
1000/1000 [==============================] - 0s 329us/sample - los
s: 3.0183 - accuracy: 0.0950 - val_loss: 2.9622 - val_accuracy: 0.
1000
Epoch 22/30
1000/1000 [==============================] - 0s 332us/sample - los
s: 3.2708 - accuracy: 0.0960 - val_loss: 3.4565 - val_accuracy: 0.
1000
Epoch 23/30
1000/1000 [==============================] - 0s 438us/sample - los
s: 3.4836 - accuracy: 0.0900 - val_loss: 3.3276 - val_accuracy: 0.
1000
Epoch 24/30
```

```
1000/1000 [==============================] - 0s 378us/sample - los
s: 3.2950 - accuracy: 0.0990 - val_loss: 3.1120 - val_accuracy: 0.
1000
Epoch 25/30
1000/1000 [==============================] - 0s 330us/sample - los
s: 3.1229 - accuracy: 0.0880 - val_loss: 3.2092 - val_accuracy: 0.
1000
Epoch 26/30
1000/1000 [==============================] - 0s 344us/sample - los
s: 3.3273 - accuracy: 0.0950 - val_loss: 3.4942 - val_accuracy: 0.
1000
Epoch 27/30
1000/1000 [==============================] - 0s 337us/sample - los
s: 3.3847 - accuracy: 0.0890 - val_loss: 3.2494 - val_accuracy: 0.
1000
Epoch 28/30
1000/1000 [==============================] - 0s 327us/sample - los
s: 3.2190 - accuracy: 0.0930 - val_loss: 3.1648 - val_accuracy: 0.
1000
Epoch 29/30
1000/1000 [==============================] - 0s 337us/sample - los
s: 3.2281 - accuracy: 0.0940 - val_loss: 3.2706 - val_accuracy: 0.
1000
Epoch 30/30
1000/1000 [==============================] - 0s 329us/sample - los
s: 3.3710 - accuracy: 0.0980 - val_loss: 3.4740 - val_accuracy: 0.
1000
```

Plot for criterion function loss

In [50]:
```
model_training_plot()
```





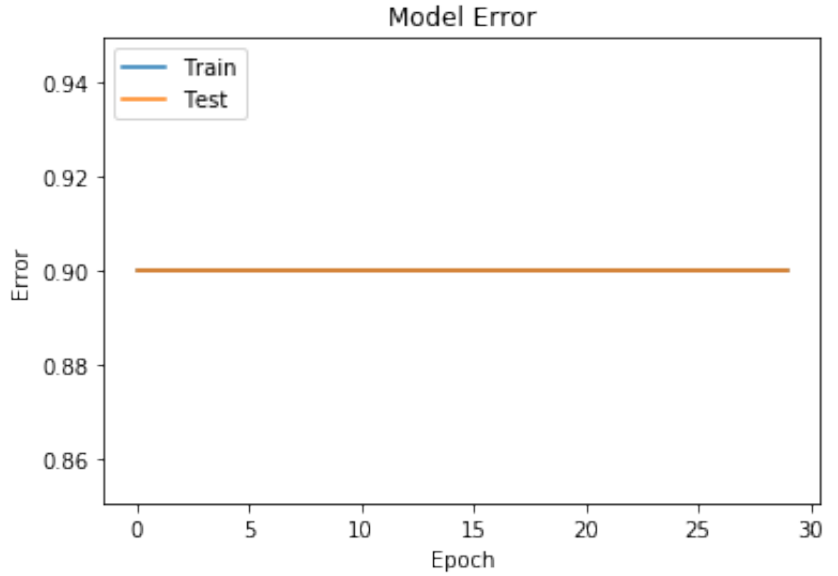
Plot for zero-one error

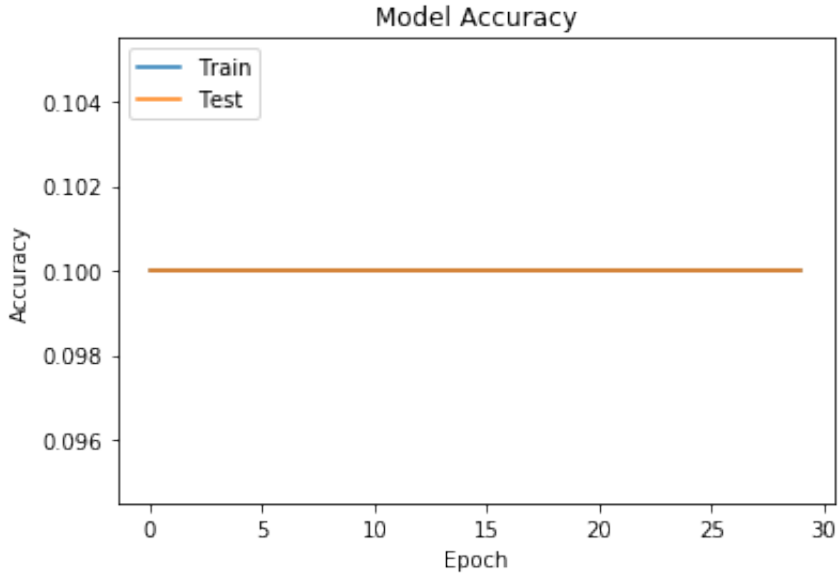
In [51]:
```
model=neural_net_3hidden(True)
traine,teste,traina,testa=for_error_plot(model)
zero_one_error_plot(traine,teste,traina,testa)
```

```
Model: "sequential_19"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_52 (Dense)             (None, 30)                23550
_________________________________________________________________
dense_53 (Dense)             (None, 30)                930
_________________________________________________________________
dense_54 (Dense)             (None, 30)                930
_________________________________________________________________
dense_55 (Dense)             (None, 10)                310
```

```
=================================================================
Total params: 25,720
Trainable params: 25,720
Non-trainable params: 0
_________________________________________________________________
Train on 1000 samples
1000/1000 [==============================] - 1s 750us/sample - los
s: 40.4962 - accuracy: 0.1090
Train on 1000 samples
1000/1000 [==============================] - 0s 226us/sample - los
s: 2.3453 - accuracy: 0.1090
Train on 1000 samples
1000/1000 [==============================] - 0s 280us/sample - los
s: 2.3412 - accuracy: 0.0980
Train on 1000 samples
1000/1000 [==============================] - 0s 257us/sample - los
s: 2.3324 - accuracy: 0.1150
Train on 1000 samples
1000/1000 [==============================] - 0s 198us/sample - los
s: 2.3385 - accuracy: 0.0850
Train on 1000 samples
1000/1000 [==============================] - 0s 272us/sample - los
s: 2.3301 - accuracy: 0.0960
Train on 1000 samples
1000/1000 [==============================] - 0s 246us/sample - los
s: 2.3243 - accuracy: 0.0950
Train on 1000 samples
1000/1000 [==============================] - 0s 212us/sample - los
s: 2.3283 - accuracy: 0.1030
Train on 1000 samples
1000/1000 [==============================] - 0s 246us/sample - los
s: 2.3264 - accuracy: 0.0980
Train on 1000 samples
1000/1000 [==============================] - 0s 233us/sample - los
s: 2.3368 - accuracy: 0.0950
Train on 1000 samples
1000/1000 [==============================] - 0s 204us/sample - los
s: 2.3364 - accuracy: 0.1090
Train on 1000 samples
1000/1000 [==============================] - 0s 207us/sample - los
s: 2.3298 - accuracy: 0.0780
Train on 1000 samples
1000/1000 [==============================] - 0s 203us/sample - los
s: 3.1083 - accuracy: 0.0900
Train on 1000 samples
1000/1000 [==============================] - 0s 207us/sample - los
s: 3.3003 - accuracy: 0.0820
Train on 1000 samples
1000/1000 [==============================] - 0s 204us/sample - los
s: 2.9490 - accuracy: 0.0970
Train on 1000 samples
1000/1000 [==============================] - 0s 229us/sample - los
s: 2.9410 - accuracy: 0.0870
```

```
Train on 1000 samples
1000/1000 [==============================] - 0s 203us/sample - los
s: 2.9352 - accuracy: 0.0870
Train on 1000 samples
1000/1000 [==============================] - 0s 207us/sample - los
s: 3.5555 - accuracy: 0.0890
Train on 1000 samples
1000/1000 [==============================] - 0s 206us/sample - los
s: 3.5269 - accuracy: 0.0780
Train on 1000 samples
1000/1000 [==============================] - 0s 202us/sample - los
s: 3.1053 - accuracy: 0.1070
Train on 1000 samples
1000/1000 [==============================] - 0s 202us/sample - los
s: 3.0694 - accuracy: 0.0840
Train on 1000 samples
1000/1000 [==============================] - 0s 202us/sample - los
s: 3.3600 - accuracy: 0.0900
Train on 1000 samples
1000/1000 [==============================] - 0s 203us/sample - los
s: 3.4318 - accuracy: 0.0970
Train on 1000 samples
1000/1000 [==============================] - 0s 202us/sample - los
s: 3.2234 - accuracy: 0.0910
Train on 1000 samples
1000/1000 [==============================] - 0s 202us/sample - los
s: 3.1541 - accuracy: 0.0940
Train on 1000 samples
1000/1000 [==============================] - 0s 202us/sample - los
s: 3.4287 - accuracy: 0.0960
Train on 1000 samples
1000/1000 [==============================] - 0s 207us/sample - los
s: 3.3139 - accuracy: 0.0960
Train on 1000 samples
1000/1000 [==============================] - 0s 204us/sample - los
s: 3.1531 - accuracy: 0.0940
Train on 1000 samples
1000/1000 [==============================] - 0s 206us/sample - los
s: 3.2569 - accuracy: 0.1010
Train on 1000 samples
1000/1000 [==============================] - 0s 203us/sample - los
s: 3.4481 - accuracy: 0.0880
```





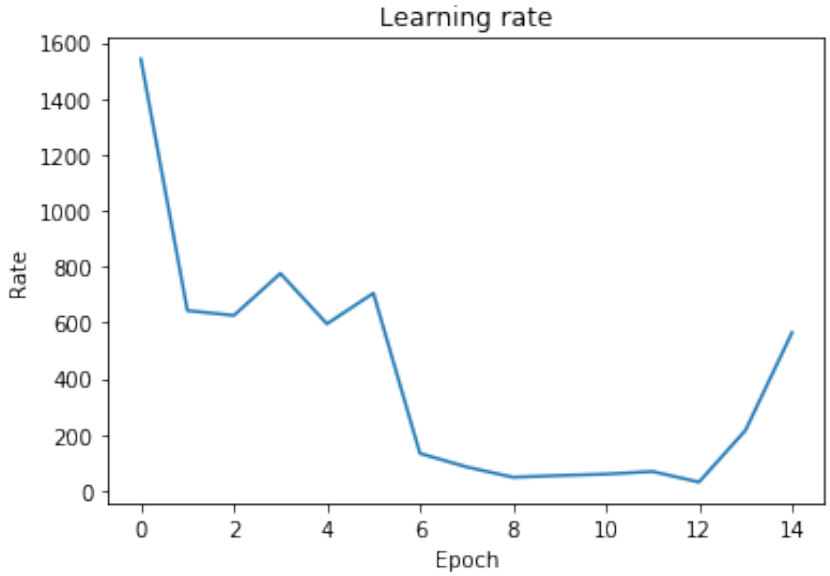
Plot for learning rate

In [52]:
```
model=neural_net_3hidden(True)
learning_rate_plot(25720,model)
```

```
Model: "sequential_20"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_56 (Dense)             (None, 30)                23550
_________________________________________________________________
dense_57 (Dense)             (None, 30)                930
_________________________________________________________________
dense_58 (Dense)             (None, 30)                930
_________________________________________________________________
dense_59 (Dense)             (None, 10)                310
=================================================================
```

```
Total params: 25,720
Trainable params: 25,720
Non-trainable params: 0
_________________________________________________________________
Train on 1000 samples
1000/1000 [==============================] - 1s 743us/sample - los
s: 40.6000 - accuracy: 0.0870
Train on 1000 samples
1000/1000 [==============================] - 0s 213us/sample - los
s: 2.3403 - accuracy: 0.0950
Train on 1000 samples
1000/1000 [==============================] - 0s 253us/sample - los
s: 2.3368 - accuracy: 0.09000s - loss: 2.3385 - accuracy: 0.
Train on 1000 samples
1000/1000 [==============================] - 0s 248us/sample - los
s: 2.3359 - accuracy: 0.0980
Train on 1000 samples
1000/1000 [==============================] - 0s 223us/sample - los
s: 2.3272 - accuracy: 0.0920
Train on 1000 samples
1000/1000 [==============================] - 0s 201us/sample - los
s: 2.3344 - accuracy: 0.0980
Train on 1000 samples
1000/1000 [==============================] - 0s 219us/sample - los
s: 2.3256 - accuracy: 0.0910
Train on 1000 samples
1000/1000 [==============================] - 0s 212us/sample - los
s: 2.3293 - accuracy: 0.0920
Train on 1000 samples
1000/1000 [==============================] - 0s 203us/sample - los
s: 2.3290 - accuracy: 0.0880
Train on 1000 samples
1000/1000 [==============================] - 0s 291us/sample - los
s: 2.3378 - accuracy: 0.0920
Train on 1000 samples
1000/1000 [==============================] - 0s 209us/sample - los
s: 2.3372 - accuracy: 0.0870
Train on 1000 samples
1000/1000 [==============================] - 0s 281us/sample - los
s: 2.3270 - accuracy: 0.0900
Train on 1000 samples
1000/1000 [==============================] - 0s 206us/sample - los
s: 3.1312 - accuracy: 0.0720
Train on 1000 samples
1000/1000 [==============================] - 0s 204us/sample - los
s: 3.2935 - accuracy: 0.0900
Train on 1000 samples
1000/1000 [==============================] - 0s 221us/sample - los
s: 2.9718 - accuracy: 0.0920
Train on 1000 samples
1000/1000 [==============================] - 0s 201us/sample - los
s: 2.9480 - accuracy: 0.1140
Train on 1000 samples
```

```
1000/1000 [==============================] - 0s 203us/sample - los
s: 2.9606 - accuracy: 0.0940
Train on 1000 samples
1000/1000 [==============================] - 0s 201us/sample - los
s: 3.5531 - accuracy: 0.1040
Train on 1000 samples
1000/1000 [==============================] - 0s 202us/sample - los
s: 3.5676 - accuracy: 0.0730
Train on 1000 samples
1000/1000 [==============================] - 0s 203us/sample - los
s: 3.1163 - accuracy: 0.0990
Train on 1000 samples
1000/1000 [==============================] - 0s 221us/sample - los
s: 3.0721 - accuracy: 0.0950
Train on 1000 samples
1000/1000 [==============================] - 0s 246us/sample - los
s: 3.2771 - accuracy: 0.0930
Train on 1000 samples
1000/1000 [==============================] - 0s 258us/sample - los
s: 3.4924 - accuracy: 0.0840
Train on 1000 samples
1000/1000 [==============================] - 0s 201us/sample - los
s: 3.2208 - accuracy: 0.0980
Train on 1000 samples
1000/1000 [==============================] - 0s 209us/sample - los
s: 3.1684 - accuracy: 0.0940
Train on 1000 samples
1000/1000 [==============================] - 0s 212us/sample - los
s: 3.3135 - accuracy: 0.0940
Train on 1000 samples
1000/1000 [==============================] - 0s 201us/sample - los
s: 3.4041 - accuracy: 0.0920
Train on 1000 samples
1000/1000 [==============================] - 0s 202us/sample - los
s: 3.2010 - accuracy: 0.0780
Train on 1000 samples
1000/1000 [==============================] - 0s 200us/sample - los
s: 3.2152 - accuracy: 0.0880
Train on 1000 samples
1000/1000 [==============================] - 0s 201us/sample - los
s: 3.4112 - accuracy: 0.0860
```



Question 2)c) CNN

In [53]:
```
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation, Flatten
from keras.optimizers import Adam
from keras.layers.normalization import BatchNormalization
from keras.utils import np_utils
from keras.layers import Conv2D, MaxPooling2D, ZeroPadding2D, Globa
lAveragePooling2D
from keras.layers.advanced_activations import LeakyReLU
from keras.preprocessing.image import ImageDataGenerator
```

In [64]:
```
model = Sequential()

model.add(Conv2D(32, (3, 3), input_shape=(28,28,1)))
model.add(LeakyReLU(alpha=0.1))
model.add(Conv2D(32, (3, 3)))
model.add(LeakyReLU(alpha=0.1))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Conv2D(64,(3, 3)))
model.add(LeakyReLU(alpha=0.1))
model.add(Conv2D(64, (3, 3)))
model.add(LeakyReLU(alpha=0.1))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Flatten())

model.add(Dense(512))
model.add(LeakyReLU(alpha=0.1))
model.add(Dropout(0.2))
model.add(Dense(10))

model.add(Activation('softmax'))
```

In [65]:
```
model.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_5 (Conv2D)            (None, 26, 26, 32)        320
_________________________________________________________________
leaky_re_lu_1 (LeakyReLU)    (None, 26, 26, 32)        0
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 24, 24, 32)        9248
_________________________________________________________________
leaky_re_lu_2 (LeakyReLU)    (None, 24, 24, 32)        0
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 12, 12, 32)        0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 10, 10, 64)        18496
_________________________________________________________________
leaky_re_lu_3 (LeakyReLU)    (None, 10, 10, 64)        0
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 8, 8, 64)          36928
_________________________________________________________________
leaky_re_lu_4 (LeakyReLU)    (None, 8, 8, 64)          0
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 4, 4, 64)          0
_________________________________________________________________
flatten_2 (Flatten)          (None, 1024)              0
_________________________________________________________________
dense_3 (Dense)              (None, 512)               524800
_________________________________________________________________
leaky_re_lu_5 (LeakyReLU)    (None, 512)               0
_________________________________________________________________
dropout_2 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_4 (Dense)              (None, 10)                5130
_________________________________________________________________
activation_7 (Activation)    (None, 10)                0
=================================================================
Total params: 594,922
Trainable params: 594,922
Non-trainable params: 0
_________________________________________________________________
```

In [66]:
```
mini_train=mini_train.reshape((1000,28,28,1))
mini_test=mini_test.reshape((1000,28,28,1))
```

In [67]:
```
mini_train.shape,mini_test.shape
```

Out[67]: ((1000, 28, 28, 1), (1000, 28, 28, 1))

In [68]:
```
train_labels.shape,test_labels.shape
```

Out[68]: ((1000, 10), (1000, 10))

In [69]:
```
model.compile(loss='categorical_crossentropy', optimizer=Adam(learn
ing_rate=0.1), metrics=['accuracy'])
```

In [70]:
```
train_gen = ImageDataGenerator(rotation_range=3, width_shift_range=
3, height_shift_range=3)

test_gen = ImageDataGenerator()
```

In [71]:
```
train_generator = train_gen.flow(mini_train, train_labels, batch_si
ze=10)
test_generator = test_gen.flow(mini_test, test_labels, batch_size=1
0)
```

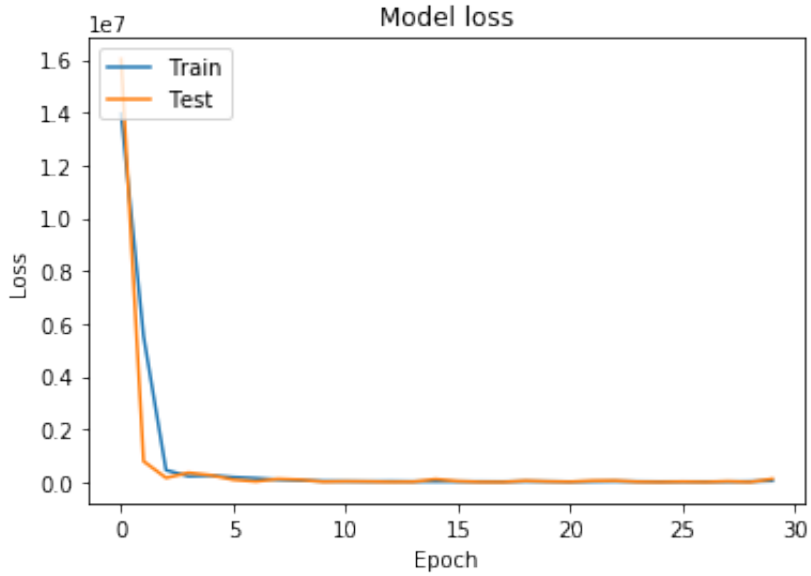
In [72]:
```
history=model.fit_generator(train_generator, steps_per_epoch=1000//
10, epochs=30, validation_data=test_generator, validation_steps=100
0//10)
```

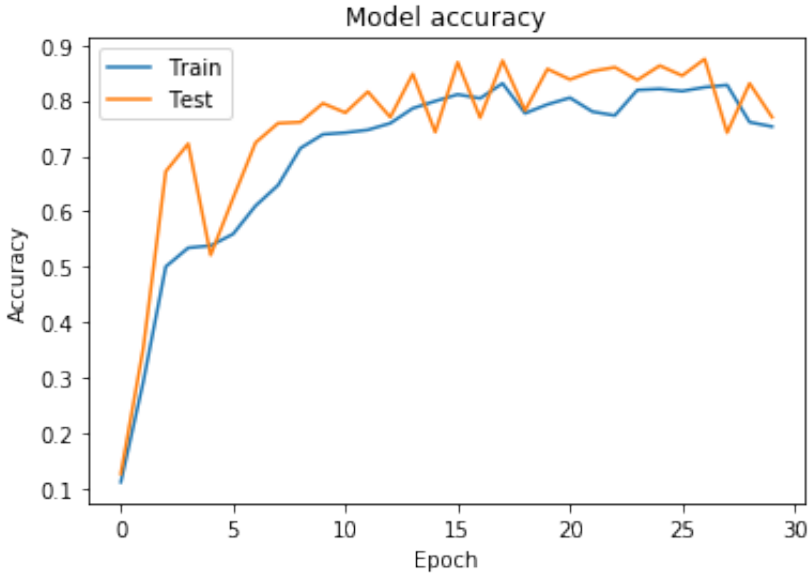
```
Epoch 1/30
100/100 [==============================] - 5s 49ms/step - loss: 13
943718.7768 - accuracy: 0.1110 - val_loss: 16028184.0000 - val_acc
uracy: 0.1260
Epoch 2/30
100/100 [==============================] - 4s 44ms/step - loss: 55
29694.5050 - accuracy: 0.2920 - val_loss: 788680.6250 - val_accura
cy: 0.3530
Epoch 3/30
100/100 [==============================] - 4s 44ms/step - loss: 45
1844.4041 - accuracy: 0.5000 - val_loss: 159742.6250 - val_accurac
y: 0.6720
Epoch 4/30
100/100 [==============================] - 4s 44ms/step - loss: 22
2572.9458 - accuracy: 0.5340 - val_loss: 352928.4062 - val_accurac
y: 0.7220
Epoch 5/30
100/100 [==============================] - 4s 43ms/step - loss: 24
8917.9212 - accuracy: 0.5380 - val_loss: 259498.5000 - val_accurac
y: 0.5210
Epoch 6/30
100/100 [==============================] - 5s 45ms/step - loss: 18
4655.6699 - accuracy: 0.5590 - val_loss: 90099.7891 - val_accuracy
: 0.6240
Epoch 7/30
100/100 [==============================] - 5s 46ms/step - loss: 14
4813.8153 - accuracy: 0.6100 - val_loss: 25548.2871 - val_accuracy
: 0.7240
Epoch 8/30
100/100 [==============================] - 5s 46ms/step - loss: 92
076.4708 - accuracy: 0.6470 - val_loss: 118837.2266 - val_accuracy
: 0.7590
Epoch 9/30
100/100 [==============================] - 5s 46ms/step - loss: 60
```

```
656.4339 - accuracy: 0.7140 - val_loss: 87568.8125 - val_accuracy:
0.7610
Epoch 10/30
100/100 [==============================] - 5s 46ms/step - loss: 39
367.4359 - accuracy: 0.7390 - val_loss: 13037.2109 - val_accuracy:
0.7950
Epoch 11/30
100/100 [==============================] - 5s 48ms/step - loss: 30
173.6685 - accuracy: 0.7420 - val_loss: 27648.9473 - val_accuracy:
0.7780
Epoch 12/30
100/100 [==============================] - 5s 46ms/step - loss: 24
554.4343 - accuracy: 0.7470 - val_loss: 21893.1211 - val_accuracy:
0.8160
Epoch 13/30
100/100 [==============================] - 5s 45ms/step - loss: 31
150.1836 - accuracy: 0.7590 - val_loss: 5697.6123 - val_accuracy:
0.7700
Epoch 14/30
100/100 [==============================] - 5s 45ms/step - loss: 22
701.2793 - accuracy: 0.7860 - val_loss: 11643.3145 - val_accuracy:
0.8480
Epoch 15/30
100/100 [==============================] - 5s 47ms/step - loss: 20
547.3794 - accuracy: 0.7990 - val_loss: 110618.9766 - val_accuracy
: 0.7430
Epoch 16/30
100/100 [==============================] - 5s 47ms/step - loss: 20
918.9665 - accuracy: 0.8110 - val_loss: 34346.0859 - val_accuracy:
0.8690
Epoch 17/30
100/100 [==============================] - 5s 46ms/step - loss: 20
114.8533 - accuracy: 0.8040 - val_loss: 3444.9390 - val_accuracy:
0.7690
Epoch 18/30
100/100 [==============================] - 5s 47ms/step - loss: 15
939.2175 - accuracy: 0.8310 - val_loss: 0.0000e+00 - val_accuracy:
0.8720
Epoch 19/30
100/100 [==============================] - 4s 44ms/step - loss: 28
562.0061 - accuracy: 0.7770 - val_loss: 52312.0234 - val_accuracy:
0.7820
Epoch 20/30
100/100 [==============================] - 4s 43ms/step - loss: 23
577.5042 - accuracy: 0.7930 - val_loss: 24177.2129 - val_accuracy:
0.8570
Epoch 21/30
100/100 [==============================] - 5s 46ms/step - loss: 14
180.6516 - accuracy: 0.8050 - val_loss: 9576.2246 - val_accuracy:
0.8380
Epoch 22/30
100/100 [==============================] - 4s 45ms/step - loss: 21
023.0222 - accuracy: 0.7800 - val_loss: 51836.2148 - val_accuracy:
```

```
0.8530
Epoch 23/30
100/100 [==============================] - 5s 45ms/step - loss: 29
084.9780 - accuracy: 0.7730 - val_loss: 56501.9062 - val_accuracy:
0.8600
Epoch 24/30
100/100 [==============================] - 5s 45ms/step - loss: 21
363.0631 - accuracy: 0.8190 - val_loss: 11489.7939 - val_accuracy:
0.8370
Epoch 25/30
100/100 [==============================] - 5s 45ms/step - loss: 14
810.4845 - accuracy: 0.8210 - val_loss: 0.0000e+00 - val_accuracy:
0.8630
Epoch 26/30
100/100 [==============================] - 5s 51ms/step - loss: 16
678.5440 - accuracy: 0.8170 - val_loss: 15532.1309 - val_accuracy:
0.8450
Epoch 27/30
100/100 [==============================] - 5s 47ms/step - loss: 15
543.7392 - accuracy: 0.8240 - val_loss: 0.0000e+00 - val_accuracy:
0.8750
Epoch 28/30
100/100 [==============================] - 5s 48ms/step - loss: 17
396.9999 - accuracy: 0.8280 - val_loss: 29576.3809 - val_accuracy:
0.7420
Epoch 29/30
100/100 [==============================] - 4s 42ms/step - loss: 25
180.8621 - accuracy: 0.7610 - val_loss: 0.0000e+00 - val_accuracy:
0.8310
Epoch 30/30
100/100 [==============================] - 4s 43ms/step - loss: 49
357.7607 - accuracy: 0.7530 - val_loss: 120708.9219 - val_accuracy
: 0.7700
```

In [73]:
```
model_training_plot()
```





References :

https://yashk2810.github.io/Applying-Convolutional-Neural-Network-on-the-MNIST-dataset/ (https://yashk2810.github.io/Applying-Convolutional-Neural-Network-on-the-MNIST-dataset/)

https://rohanvarma.me/Regularization/ (https://rohanvarma.me/Regularization/)

https://ljvmiranda921.github.io/notebook/2017/08/13/softmax-and-the-negative-log-likelihood/ (https://ljvmiranda921.github.io/notebook/2017/08/13/softmax-and-the-negative-log-likelihood/)