Question 1

(80 points) Train and validate your own n-layer Neural Network on the Apparel dataset to predict the class label of a given apparel. You are free to choose the hyper-parameters, training strategy to handle large number of training data (Hint: Batch Size) architecture - number of hidden layers, number of nodes in each hidden layer etc.

Description of Apparel dataset:

The dataset contains 60,000 examples - each example is a 28x28 grayscale image, belonging to one of the 10 following class labels.
Class labels:

Label	Description
0	T-shirt/top
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Ankle boot

Represntation of images in the dataset:

The images are flattened to represent them as a row - each row consisting of 28*28 = 784 values. Each value reprsents a pixel of the image. To locate a pixel on the image, suppose that we have decomposed x as x = i * 28 + j, where i and j are integers between 0 and 27. The pixel is located on row i and column j of a 28 x 28 matrix. For example, pixel31 indicates the pixel that is in the fourth column from the left, and the second row from the top.

Dataset format:

The first row represnts the heading. Rests are the examples.
Each row, having 785 columns, in the CSV file represents one example. The first column reprsents the label of the image.
The rest of the 784 columns are the pixel values.

The sample dataset format:

```
label,pixel1,pixel2,pixel3, ... ,pixel784
4,0,0,0,0,0,0,1,1,0,0,21,153,100,88,81,130,...,156
2,0,0,0,0,0,0,1,0,0,0,12,111,32,10,5,79,34,...,0
.
.
.
```

Load the dataset :

In [1]:

```
import pandas as pd
df = pd.read_csv("Apparel/apparel-trainval.csv")

df.head()
```

Out[1]:

	label	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	p
0	2	0	0	0	0	0	0	0	0	0
1	9	0	0	0	0	0	0	0	0	0
2	6	0	0	0	0	0	0	0	5	0
3	0	0	0	0	1	2	0	0	0	0
4	3	0	0	0	0	0	0	0	0	0

5 rows × 785 columns

PREPROCESSING:

Remove Labels

In [2]:

```
labels = df["label"]
df = df.iloc[:,1:]
df.head()
```

Out[2]:

	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	pixel9
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	5	0
3	0	0	0	1	2	0	0	0	0
4	0	0	0	0	0	0	0	0	0

5 rows × 784 columns

PREPROCESSING:

Standardization of the Dataset

In [3]:

```
from sklearn.exceptions import DataConversionWarning
import warnings
warnings.filterwarnings(action='ignore', category=DataConversionWarning)

from sklearn.preprocessing import StandardScaler
df = StandardScaler().fit_transform(df)
```

PREPROCESSING:

One Hot Encode Labels

In [4]:

```
import numpy as np
num_classes = 10
targets = np.array([labels]).reshape(-1)
one_hot_targets = np.eye(num_classes)[labels]
```

NEURAL NETWORK :

In [243]:

```
import numpy as np
import math



class Layer_Details:
    def __init__(self, nodes_in_this_layer, nodes_in_next_layer, activat
ion_func):
        ###### --------------store details in each layers--------------
--######
        self.nodes_in_this_layer = nodes_in_this_layer
        self.activation_func = activation_func
        self.activations = np.zeros([nodes_in_this_layer,1])

        if nodes_in_next_layer == 0 :
            self.weights = None
            self.bias = None
        else:
            self.weights = np.random.rand(nodes_in_this_layer, nodes_in_
next_layer) * math.sqrt(2/nodes_in_this_layer)
            self.bias = np.random.rand(1, nodes_in_next_layer)



class Neural_NET:
    def __init__(self, num_of_layers, nodes_in_layers, activation_func ,
cost_func, test_plot = False, test_data=None,test_labels=None):
        ########             initialise hyperparameters               ##
#######

        self.num_of_layers = num_of_layers
        self.nodes_in_layers = nodes_in_layers
        self.cost_func = cost_func
        self.Layers = []
        self.test_plot = test_plot
        self.train_accuracy=[]
        self.test_accuracy=[]
        self.epochs=[]
        self.test_data = test_data
        self.test_labels = test_labels

        #######             test correctness of the parameters        #
#######

        if num_of_layers != len(nodes_in_layers):
            raise ValueError("check your parameters, ambiguity is size
s!")
```

```
        #######              storing the details in each layer         #
########

        for i in range(len(nodes_in_layers)):
            if i+1 == len(nodes_in_layers):
                if type(activation_func) == list:
                    layer = Layer_Details(nodes_in_layers[i], 0 ,activat
ion_func[i])
                else:
                    layer = Layer_Details(nodes_in_layers[i], 0 ,activat
ion_func)

            else :
                if type(activation_func) == list:
                    layer = Layer_Details(nodes_in_layers[i],nodes_in_la
yers[i+1],activation_func[i])
                else:
                    layer = Layer_Details(nodes_in_layers[i],nodes_in_la
yers[i+1],activation_func)

            self.Layers.append(layer)

    def relu(self, Z):
        Z = np.maximum(Z,0)
        return np.nan_to_num(Z)

    def sigmoid(self, Z):
        return np.divide(1, np.add(1, np.exp(np.negative(Z))))

    def tanh(self, Z):
        return np.tanh(Z)

    def softmax(self, x):
        exp = np.nan_to_num(np.exp(x))
        if isinstance(x[0], np.ndarray):
            return np.nan_to_num(exp/np.sum(exp, axis=1, keepdims=True))
        else:
            return np.nan_to_num(exp/np.sum(exp, keepdims=True))


    def sig_derivative(self, X):
        return np.multiply(X,1-X)

    def reluDerivative(self,x):
        x = np.where(x>=0,1,0)
        return x.astype('int')
    def tanhDerivative(self, Z):
        a = np.tanh(Z)
        return 1-a**2
```

```
    #---------------------forward propagation with different activation
functions-----------------------#

    def forward_pass(self, data):
        self.Layers[0].activations = data
        for i in range(self.num_of_layers-1):
            Z_i = np.add(np.matmul(self.Layers[i].activations , self.Lay
ers[i].weights),self.Layers[i].bias)
            if self.Layers[i+1].activation_func == "sigmoid":
                self.Layers[i+1].activations = self.sigmoid(Z_i)
                self.Layers[i+1].Z = Z_i
            elif self.Layers[i+1].activation_func == "tanh":
                self.Layers[i+1].activations = self.tanh(Z_i)
                self.Layers[i+1].Z = Z_i
            elif self.Layers[i+1].activation_func == "relu":
                self.Layers[i+1].activations = self.relu(Z_i)
                self.Layers[i+1].Z = Z_i
            elif self.Layers[i+1].activation_func == "softmax":
                self.Layers[i+1].activations = self.softmax(Z_i)
                self.Layers[i+1].Z = Z_i
            else:
                self.Layers[i+1].activations = Z_i
                self.Layers[i+1].Z = Z_i




    #----------------------------back-propagation with different activa
tion function---------------------------#

    def backward_pass(self,labels):
        i = self.num_of_layers-1
        y_hat = self.Layers[i].activations
        del_b = y_hat - labels
        del_w = np.dot(np.asarray(self.Layers[i-1].activations).T,del_b)
        new_weights = self.Layers[i-1].weights - (self.lr/labels.shape[0
]) * del_w
        new_bias = self.Layers[i-1].bias - self.lr * np.mean(del_b,axis=
0)
        for i in range(i-1,0,-1):
            y_hat = self.Layers[i].activations
            if self.Layers[i].activation_func == "sigmoid" or self.Layer
s[i].activation_func == "softmax":
                del_b = np.nan_to_num(np.multiply(np.dot(del_b , new_wei
ghts.T),np.nan_to_num(np.multiply(y_hat,1-y_hat))))
            elif self.Layers[i].activation_func == "relu":
                del_b = np.nan_to_num(np.multiply(np.dot(del_b , new_wei
ghts.T) ,self.reluDerivative(self.Layers[i].Z)))
            elif self.Layers[i].activation_func == "tanh":
                del_b = np.nan_to_num(np.multiply(np.dot(del_b, new_weig
hts.T), self.tanhDerivative(self.Layers[i].Z)))
```

```
#                 print(del_b.shape)
#                 input()
            del_w = np.dot(self.Layers[i-1].activations.T, del_b)
            self.Layers[i].weights = new_weights
            self.Layers[i].bias = new_bias
            new_weights = self.Layers[i-1].weights - (self.lr/labels.sha
pe[0]) * del_w
            new_bias = self.Layers[i-1].bias - self.lr * np.mean(del_b,a
xis=0)
        self.Layers[0].weights = new_weights
        self.Layers[0].bias = new_bias

    #-----------------------------------------Calculate cost---------
-----------------------------------#

    def error_calulation(self, labels):
        if self.cost_func == "mean_sqaured":
            self.error += np.mean(np.divide(np.square(np.substract(label
s, self.Layers[-1].activations)) , 2))
        elif self.cost_func == "cross_entropy":
            self.error += -np.sum(labels*np.nan_to_num(np.log(self.Layer
s[-1].activations+1e-9)))

    #-----------------------------------check error in provided training
details-------------------------------#

    def check_data(self, batch_size, training_data , labels):
        self.batch_size = batch_size
        if training_data.shape[0] % batch_size != 0 :
            raise ValueError("input size is not multiple of batch size!"
)
        if training_data.shape[1] != self.nodes_in_layers[0]:
            raise ValueError("input dimension doesn't match with nodes i
n input layer!")
        if labels.shape[1] != self.nodes_in_layers[-1]:
            raise ValueError("output layer size mismatch!")

    #-------------------------------------Train the neural network------
-------------------------------------#

    def train(self, batch_size, training_data, labels, epochs, learning_
rate):
        self.lr = learning_rate
        self.batch_size = batch_size
        self.check_data(self.batch_size,training_data,labels)
        for i in range(epochs):
            j=0
            print("EPOCHS: ", i+1, "of ", epochs, "==")
            while j+batch_size <= len(training_data):
                print("training with ", j+batch_size+1 ,"of ", len(train
ing_data),end='\r' )
```

```
                self.error = 0
                self.forward_pass(training_data[j:j+batch_size])
                self.error_calulation(labels[j:j+batch_size])
                self.backward_pass(labels[j:j+batch_size])
                j=j+batch_size
            self.error = self.error/batch_size
            print("\nError = ", self.error)
            if self.test_plot == True:
                self.train_accuracy.append(self.check_accuracy(training_
data,labels))
                self.test_accuracy.append(self.check_accuracy(self.test_
data,test_labels))
                self.epochs.append(i+1)


    #----------------------------------predict output for test data ---
-----------------------------#
    def predict(self, inputs):
        self.batch_size = 1
        self.forward_pass(inputs)
        a = self.Layers[self.num_of_layers-1].activations
#         a[np.where(a==np.max(a))] = 1
#         a[np.where(a!=np.max(a))] = 0
#         print(a)
#         a = np.where(a==np.max(a),1,0)
        b = np.zeros_like(a)
        b[np.arange(len(a)), a.argmax(1)] = 1
        return b


    #-------------------------------check accuracy of your trained model
--------------------------#
    def check_accuracy(self, inputs, labels):
        self.batch_size = len(inputs)
        self.forward_pass(inputs)
        a = self.Layers[self.num_of_layers-1].activations
        b=np.zeros_like(a)
        b[np.arange(len(a)), a.argmax(1)] = 1
        total=0
        correct=0
        for i in range(len(b)):
            total += 1
            if np.equal(b[i], labels[i]).all():
                correct += 1
        return correct*100/total
```

training with two hidden layers of size 20

used ReLU and sigmoid in 2 hidden layers, softmax at output layers

In [153]:

```
##      training the neural network

net =Neural_NET(4, [784,20, 20,10], ["relu","relu","sigmoid","softmax"],
cost_func="cross_entropy")
net.train(128,df[0:51200], labels=one_hot_targets[0:51200], epochs=40, l
earning_rate=0.1)
```

```
EPOCHS:  1 of  40 ==
training with  51201 of  51200
Error =  2.1621408639616484
EPOCHS:  2 of  40 ==
training with  51201 of  51200of  51200
Error =  1.8069053010720395
EPOCHS:  3 of  40 ==
training with  51201 of  51200 of  51200 51200
Error =  1.072620967375528
EPOCHS:  4 of  40 ==
training with  51201 of  51200
Error =  0.709027340523882
EPOCHS:  5 of  40 ==
training with  51201 of  51200
Error =  0.5435559701060849
EPOCHS:  6 of  40 ==
training with  51201 of  51200
Error =  0.49076069413246304
EPOCHS:  7 of  40 ==
training with  51201 of  51200
Error =  0.4514620798157516
EPOCHS:  8 of  40 ==
training with  51201 of  51200
Error =  0.4287402783858153
EPOCHS:  9 of  40 ==
training with  51201 of  51200
Error =  0.41176355612353227
EPOCHS:  10 of  40 ==
training with  51201 of  51200
Error =  0.39074423072592235
EPOCHS:  11 of  40 ==
training with  51201 of  5120022401 of  51200
Error =  0.3729302375833301
EPOCHS:  12 of  40 ==
training with  51201 of  51200
Error =  0.3628477561150271
EPOCHS:  13 of  40 ==
training with  51201 of  51200
Error =  0.3540466335358482
EPOCHS:  14 of  40 ==
training with  51201 of  5120038017 of  51200
Error =  0.3473464524444513
EPOCHS:  15 of  40 ==
training with  51201 of  51200
Error =  0.3426517494940052
EPOCHS:  16 of  40 ==
training with  51201 of  5120051200
Error =  0.335047350716986
EPOCHS:  17 of  40 ==
training with  51201 of  51200
```

```
Error =  0.3315373432258531
EPOCHS:  18 of  40 ==
training with  51201 of  51200
Error =  0.33029351165146037
EPOCHS:  19 of  40 ==
training with  51201 of  51200
Error =  0.33370511986779805
EPOCHS:  20 of  40 ==
training with  51201 of  51200
Error =  0.3319176857372525
EPOCHS:  21 of  40 ==
training with  51201 of  51200
Error =  0.3273693421118249
EPOCHS:  22 of  40 ==
training with  51201 of  51200
Error =  0.3240865263407639
EPOCHS:  23 of  40 ==
training with  51201 of  51200
Error =  0.3191347496090038
EPOCHS:  24 of  40 ==
training with  51201 of  51200
Error =  0.313873738120126
EPOCHS:  25 of  40 ==
training with  51201 of  51200
Error =  0.3081511032850445
EPOCHS:  26 of  40 ==
training with  51201 of  51200
Error =  0.3058324366848878
EPOCHS:  27 of  40 ==
training with  51201 of  5120051200
Error =  0.29980629227598504
EPOCHS:  28 of  40 ==
training with  51201 of  51200of  51200
Error =  0.299670691063923
EPOCHS:  29 of  40 ==
training with  51201 of  51200
Error =  0.2947217621786593
EPOCHS:  30 of  40 ==
training with  51201 of  51200
Error =  0.2913156683322471
EPOCHS:  31 of  40 ==
training with  51201 of  5120051200
Error =  0.2880198334261077
EPOCHS:  32 of  40 ==
training with  51201 of  51200
Error =  0.28336295093210484
EPOCHS:  33 of  40 ==
training with  51201 of  51200
Error =  0.27992160514355735
EPOCHS:  34 of  40 ==
training with  51201 of  51200
```

```
Error =  0.2705207468718285
EPOCHS:  35 of  40 ==
training with  51201 of  51200
Error =  0.26769185516413585
EPOCHS:  36 of  40 ==
training with  51201 of  51200
Error =  0.26251874531598074
EPOCHS:  37 of  40 ==
training with  51201 of  51200
Error =  0.26002992412451137
EPOCHS:  38 of  40 ==
training with  51201 of  51200
Error =  0.2572385786010385
EPOCHS:  39 of  40 ==
training with  51201 of  51200
Error =  0.254068811695669
EPOCHS:  40 of  40 ==
training with  51201 of  51200
Error =  0.2575174853673692
```

testing accuracy of unseen data

In [154]:

```
df_test = df[51200:60000]
test_labels = one_hot_targets[51200:60000]
df_test.shape
pred = net.predict(df_test)
print(net.check_accuracy(df_test,test_labels))
```

```
86.30681818181819
```

Accuracy is 86.3% with this architechture. Lets see if we can make it better

In [177]:

```
net = Neural_NET(4, [784,20, 20,10], ["relu","relu","sigmoid","softmax"
], cost_func="cross_entropy",test_plot=True,test_data=df_test,test_label
s=test_labels)
net.train(128,df[0:51200], labels=one_hot_targets[0:51200], epochs=70, l
earning_rate=0.1)
```

```
EPOCHS:  1 of  70 ==
training with  51201 of  5120024065 of  51200
Error =  2.210905341892505
EPOCHS:  2 of  70 ==
training with  51201 of  5120032769 of  51200
Error =  1.7701337329289637
EPOCHS:  3 of  70 ==
training with  51201 of  51200
Error =  1.0411681953749552
EPOCHS:  4 of  70 ==
training with  51201 of  51200
Error =  0.6628740448609987
EPOCHS:  5 of  70 ==
training with  51201 of  51200
Error =  0.5376316772872823
EPOCHS:  6 of  70 ==
training with  51201 of  51200
Error =  0.4874279587614177
EPOCHS:  7 of  70 ==
training with  51201 of  51200
Error =  0.4639028882814592
EPOCHS:  8 of  70 ==
training with  51201 of  51200
Error =  0.44907331953602536
EPOCHS:  9 of  70 ==
training with  51201 of  51200
Error =  0.4362870914418211
EPOCHS:  10 of  70 ==
training with  51201 of  51200
Error =  0.42383259346509544
EPOCHS:  11 of  70 ==
training with  51201 of  51200 51200
Error =  0.4163519579034889
EPOCHS:  12 of  70 ==
training with  51201 of  51200
Error =  0.4095388067826044
EPOCHS:  13 of  70 ==
training with  51201 of  51200
Error =  0.4020290527513256
EPOCHS:  14 of  70 ==
training with  51201 of  51200
Error =  0.3765130591784515
EPOCHS:  15 of  70 ==
training with  51201 of  51200
Error =  0.36261804874587306
EPOCHS:  16 of  70 ==
training with  51201 of  51200
Error =  0.363260658292041
EPOCHS:  17 of  70 ==
training with  51201 of  51200
```

```
Error =  0.36545712061652547
EPOCHS:  18 of  70 ==
training with  51201 of  51200
Error =  0.3595071175034675
EPOCHS:  19 of  70 ==
training with  51201 of  512001200
Error =  0.35601788137804885
EPOCHS:  20 of  70 ==
training with  51201 of  5120051200
Error =  0.34394880905356684
EPOCHS:  21 of  70 ==
training with  51201 of  51200
Error =  0.340854831375313
EPOCHS:  22 of  70 ==
training with  51201 of  51200
Error =  0.3338902723757645
EPOCHS:  23 of  70 ==
training with  51201 of  51200
Error =  0.3337260752802752
EPOCHS:  24 of  70 ==
training with  51201 of  51200
Error =  0.32206833615473407
EPOCHS:  25 of  70 ==
training with  51201 of  51200
Error =  0.315302510240646
EPOCHS:  26 of  70 ==
training with  51201 of  51200of  51200
Error =  0.3089169557051113
EPOCHS:  27 of  70 ==
training with  51201 of  51200
Error =  0.30837617098863457
EPOCHS:  28 of  70 ==
training with  51201 of  51200
Error =  0.3103482173770392
EPOCHS:  29 of  70 ==
training with  51201 of  51200
Error =  0.30714145066265297
EPOCHS:  30 of  70 ==
training with  51201 of  51200
Error =  0.29984782616885974
EPOCHS:  31 of  70 ==
training with  51201 of  51200
Error =  0.3057171743557985
EPOCHS:  32 of  70 ==
training with  51201 of  51200
Error =  0.302415316280473
EPOCHS:  33 of  70 ==
training with  51201 of  5120032641 of  51200
Error =  0.29783655910118223
EPOCHS:  34 of  70 ==
training with  51201 of  51200
```

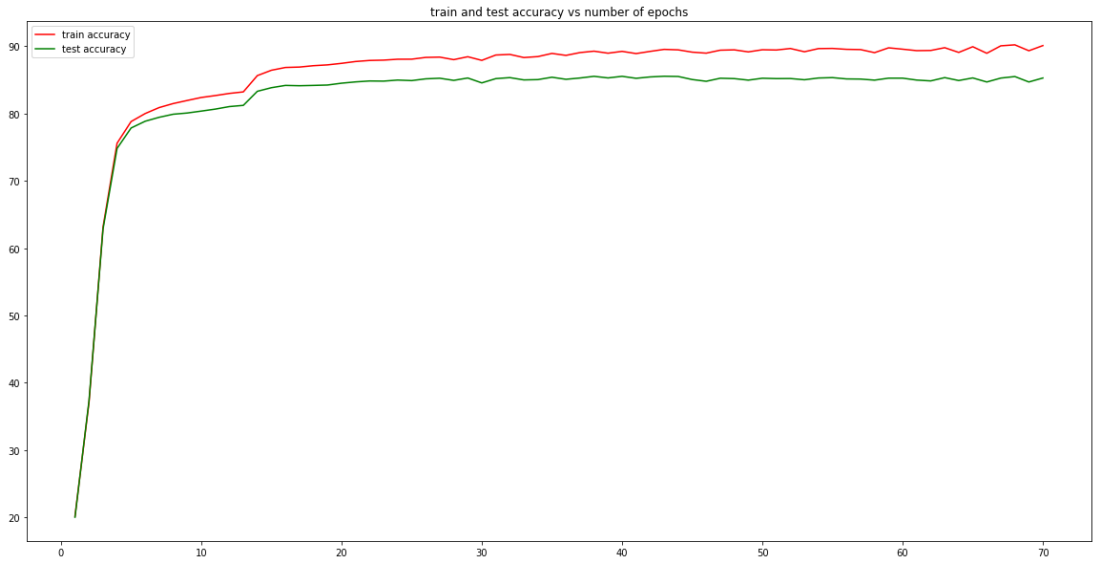
```
Error =  0.2970543448459314
EPOCHS:  35 of  70 ==
training with  51201 of  51200
Error =  0.3122482725659811
EPOCHS:  36 of  70 ==
training with  51201 of  51200
Error =  0.2968979987758072
EPOCHS:  37 of  70 ==
training with  51201 of  51200
Error =  0.3043629864902899
EPOCHS:  38 of  70 ==
training with  51201 of  51200
Error =  0.30044083292730195
EPOCHS:  39 of  70 ==
training with  51201 of  51200
Error =  0.30038155365987645
EPOCHS:  40 of  70 ==
training with  51201 of  51200 51200
Error =  0.29624667935138926
EPOCHS:  41 of  70 ==
training with  51201 of  51200
Error =  0.3004978422088675
EPOCHS:  42 of  70 ==
training with  51201 of  51200
Error =  0.3034491784823683
EPOCHS:  43 of  70 ==
training with  51201 of  5120051200
Error =  0.3056601551752147
EPOCHS:  44 of  70 ==
training with  51201 of  51200
Error =  0.2992759342382657
EPOCHS:  45 of  70 ==
training with  51201 of  51200
Error =  0.3002918161809311
EPOCHS:  46 of  70 ==
training with  51201 of  51200
Error =  0.2940618801097066
EPOCHS:  47 of  70 ==
training with  51201 of  51200
Error =  0.30621264258744163
EPOCHS:  48 of  70 ==
training with  51201 of  51200
Error =  0.30442127600323066
EPOCHS:  49 of  70 ==
training with  51201 of  51200
Error =  0.3062598060449262
EPOCHS:  50 of  70 ==
training with  51201 of  51200
Error =  0.3001204102971727
EPOCHS:  51 of  70 ==
training with  51201 of  51200
```

```
Error =  0.3004417920270171
EPOCHS:  52 of  70 ==
training with  51201 of  51200
Error =  0.2959659570750658
EPOCHS:  53 of  70 ==
training with  51201 of  51200 51200
Error =  0.2883364599823497
EPOCHS:  54 of  70 ==
training with  51201 of  51200
Error =  0.28841304109758636
EPOCHS:  55 of  70 ==
training with  51201 of  51200
Error =  0.28489919144103676
EPOCHS:  56 of  70 ==
training with  51201 of  51200
Error =  0.2753406355846002
EPOCHS:  57 of  70 ==
training with  51201 of  5120051200
Error =  0.2728600616422138
EPOCHS:  58 of  70 ==
training with  51201 of  51200 40833 of  51200
Error =  0.2713267087433081
EPOCHS:  59 of  70 ==
training with  51201 of  5120051200
Error =  0.2695882830787943
EPOCHS:  60 of  70 ==
training with  51201 of  51200
Error =  0.26560716412183105
EPOCHS:  61 of  70 ==
training with  51201 of  51200
Error =  0.2586015030479926
EPOCHS:  62 of  70 ==
training with  51201 of  5120051200
Error =  0.2609189017881197
EPOCHS:  63 of  70 ==
training with  51201 of  51200
Error =  0.27478115174249756
EPOCHS:  64 of  70 ==
training with  51201 of  51200
Error =  0.26557550869114405
EPOCHS:  65 of  70 ==
training with  51201 of  51200
Error =  0.27768174976813853
EPOCHS:  66 of  70 ==
training with  51201 of  51200 51200
Error =  0.261067132271442
EPOCHS:  67 of  70 ==
training with  51201 of  51200 51200
Error =  0.2650438022034389
EPOCHS:  68 of  70 ==
training with  51201 of  51200
```

```
Error =  0.25927553277210924
EPOCHS:  69 of  70 ==
training with  51201 of  51200 of  51200
Error =  0.25722170506578423
EPOCHS:  70 of  70 ==
training with  51201 of  51200of  51200
Error =  0.2665056836949714
```

In [178]:

```
import matplotlib.pyplot as plt
%matplotlib inline
plt.figure(figsize=(20,10))
plt.title("train and test accuracy vs number of epochs")
plt.plot(net.epochs,net.train_accuracy,color='r',label='train accuracy')
plt.plot(net.epochs,net.test_accuracy,'g',label = 'test accuracy')
plt.legend()
plt.show()
```



From this plot we can see that after 40 epochs errors are not reducing and even the accuracy comes to a constant phase. There is no increment of accuracy in both train and validation after some point.

In [179]:

```
print("accuracy: ",net.check_accuracy(df_test,test_labels))
```

```
accuracy:  85.2840909090909
```

Training with one hidden layer with 1024 nodes

Used Sigmoid in hidden layer and Softmax is used in output layer. Learning rate is 0.1

In [184]:

```
net =Neural_NET(3, [784,1024,10], ["relu","sigmoid","softmax"], cost_fun
c="cross_entropy",test_plot=True,test_data=df_test,test_labels=test_labe
ls)
net.train(128,df[0:51200], labels=one_hot_targets[0:51200], epochs=100,
learning_rate=0.1)
```

```
EPOCHS:  1 of  100 ==
training with  51201 of  5120029697 of  51200
Error =  4.635622757407553
EPOCHS:  2 of  100 ==
training with  51201 of  51200 5120051200
Error =  1.2824922386581776
EPOCHS:  3 of  100 ==
training with  51201 of  512001200of  5120033793 of  51200
Error =  1.7269621359137544
EPOCHS:  4 of  100 ==
training with  51201 of  51200120011393 of  51200 51200 512
0051200
Error =  1.0299100329860171
EPOCHS:  5 of  100 ==
training with  51201 of  5120028033 of  51200
Error =  1.1469420716300411
EPOCHS:  6 of  100 ==
training with  51201 of  51200of  51200 27009 of  512005120
0
Error =  0.6964993117742586
EPOCHS:  7 of  100 ==
training with  51201 of  51200 51200
Error =  0.6327116029356303
EPOCHS:  8 of  100 ==
training with  51201 of  51200 of  51200
Error =  0.699384474124098
EPOCHS:  9 of  100 ==
training with  51201 of  512001200 5120041601 of  51200
Error =  0.3833567729551265
EPOCHS:  10 of  100 ==
training with  51201 of  51200f  51200 51200of  51200
Error =  0.39548033796793164
EPOCHS:  11 of  100 ==
training with  51201 of  5120019201 of  51200of  51200
Error =  0.47625748959705544
EPOCHS:  12 of  100 ==
training with  51201 of  51200
Error =  0.39179932240202053
EPOCHS:  13 of  100 ==
training with  51201 of  51200120051200
Error =  0.3660238582504817
EPOCHS:  14 of  100 ==
training with  51201 of  51200
Error =  0.3753933676615914
EPOCHS:  15 of  100 ==
training with  51201 of  512005120021505 of  5120045697 of
51200
Error =  0.3613145348057317
EPOCHS:  16 of  100 ==
training with  51201 of  5120051200 51200
```

```
Error =  0.3698168865229694
EPOCHS:  17 of  100 ==
training with  51201 of  51200 512005120027137 of  5120030465 of  51200
Error =  0.360584664366439
EPOCHS:  18 of  100 ==
training with  51201 of  51200993 of  512005120046977 of  51200
Error =  0.35519864231956355
EPOCHS:  19 of  100 ==
training with  51201 of  5120051200of  51200 of  51200
Error =  0.3362354250802668
EPOCHS:  20 of  100 ==
training with  51201 of  51200
Error =  0.34328246655406947
EPOCHS:  21 of  100 ==
training with  51201 of  512001200of  51200
Error =  0.30996933247765374
EPOCHS:  22 of  100 ==
training with  51201 of  51200
Error =  0.3293302324766863
EPOCHS:  23 of  100 ==
training with  51201 of  512001200of  51200of  51200of  51200
Error =  0.3177768417391097
EPOCHS:  24 of  100 ==
training with  51201 of  5120051200 20865 of  5120051200of  5120044673 of  51200
Error =  0.3233408313341475
EPOCHS:  25 of  100 ==
training with  51201 of  51200
Error =  0.3143062736623855
EPOCHS:  26 of  100 ==
training with  51201 of  5120016513 of  51200
Error =  0.3063179484075611
EPOCHS:  27 of  100 ==
training with  51201 of  5120023681 of  51200of  51200
Error =  0.2818441199350319
EPOCHS:  28 of  100 ==
training with  51201 of  51200
Error =  0.28225178251937416
EPOCHS:  29 of  100 ==
training with  51201 of  5120051200
Error =  0.27769749209517236
EPOCHS:  30 of  100 ==
training with  51201 of  51200 43265 of  51200
Error =  0.2719474075358891
EPOCHS:  31 of  100 ==
training with  51201 of  51200 5120051200of  51200
Error =  0.26569907263655235
EPOCHS:  32 of  100 ==
```

```
training with  51201 of  512001200 51200 47361 of  51200
Error =  0.2580757726647779
EPOCHS:  33 of  100 ==
training with  51201 of  512001200 51200 51200
Error =  0.2551397539224419
EPOCHS:  34 of  100 ==
training with  51201 of  5120051200 of  51200
Error =  0.25325116265126846
EPOCHS:  35 of  100 ==
training with  51201 of  512005120042881 of  5120045185 of
51200
Error =  0.24915676848428076
EPOCHS:  36 of  100 ==
training with  51201 of  5120032129 of  5120051200
Error =  0.245895943945714
EPOCHS:  37 of  100 ==
training with  51201 of  5120025 of  51200
Error =  0.2480849902331584
EPOCHS:  38 of  100 ==
training with  51201 of  51200145 of  5120030081 of  51200
Error =  0.24137917380606477
EPOCHS:  39 of  100 ==
training with  51201 of  51200of  51200
Error =  0.23223747636761394
EPOCHS:  40 of  100 ==
training with  51201 of  5120010625 of  51200
Error =  0.2258964669054584
EPOCHS:  41 of  100 ==
training with  51201 of  51200 51200
Error =  0.2192479675385023
EPOCHS:  42 of  100 ==
training with  51201 of  51200 of  51200of  51200
Error =  0.23056124556396188
EPOCHS:  43 of  100 ==
training with  51201 of  5120051200 51200 of  51200 51200
Error =  0.22534492930274058
EPOCHS:  44 of  100 ==
training with  51201 of  5120051200
Error =  0.21975238612976078
EPOCHS:  45 of  100 ==
training with  51201 of  5120015873 of  51200 51200 51200
Error =  0.2165102534218285
EPOCHS:  46 of  100 ==
training with  51201 of  51200
Error =  0.20813853693506573
EPOCHS:  47 of  100 ==
training with  51201 of  5120015873 of  5120018049 of  5120
051200of  5120041345 of  5120042625 of  51200
Error =  0.2045487746336624
EPOCHS:  48 of  100 ==
training with  51201 of  51200of  5120051200
```

```
Error =  0.19893001477431113
EPOCHS:  49 of  100 ==
training with  51201 of  512005120051200
Error =  0.19270688987049828
EPOCHS:  50 of  100 ==
training with  51201 of  5120012004097 of  512006017 of  51
200
Error =  0.1861977781589894
EPOCHS:  51 of  100 ==
training with  51201 of  51200 51200
Error =  0.18111464748228973
EPOCHS:  52 of  100 ==
training with  51201 of  5120047361 of  51200
Error =  0.17661521929872018
EPOCHS:  53 of  100 ==
training with  51201 of  51200512005120051200
Error =  0.1719835167014858
EPOCHS:  54 of  100 ==
training with  51201 of  5120069 of  51200 51200of  51200
Error =  0.1673631933940759
EPOCHS:  55 of  100 ==
training with  51201 of  51200f  5120051200of  51200
Error =  0.16332375665900167
EPOCHS:  56 of  100 ==
training with  51201 of  512001200 51200
Error =  0.16032319529332492
EPOCHS:  57 of  100 ==
training with  51201 of  5120051200
Error =  0.15784174119898542
EPOCHS:  58 of  100 ==
training with  51201 of  51200
Error =  0.15614059237754177
EPOCHS:  59 of  100 ==
training with  51201 of  51200 5120041601 of  51200
Error =  0.15437667933502247
EPOCHS:  60 of  100 ==
training with  51201 of  51200of  51200
Error =  0.15224071142793488
EPOCHS:  61 of  100 ==
training with  51201 of  5120013697 of  51200of  5120050177
of  51200
Error =  0.15083090655964998
EPOCHS:  62 of  100 ==
training with  51201 of  51200225 of  51200of  51200of  512
00of  51200
Error =  0.14888185956783465
EPOCHS:  63 of  100 ==
training with  51201 of  51200
Error =  0.14687262466196127
EPOCHS:  64 of  100 ==
training with  51201 of  5120051200
```

```
Error =  0.14516126128275592
EPOCHS:  65 of  100 ==
training with  51201 of  5120028289 of  51200 5120051200
Error =  0.14420447809370834
EPOCHS:  66 of  100 ==
training with  51201 of  51200 51200 5120030465 of  51200
Error =  0.14332664529143882
EPOCHS:  67 of  100 ==
training with  51201 of  5120051200of  5120046465 of  51200
Error =  0.14236355748451318
EPOCHS:  68 of  100 ==
training with  51201 of  51200
Error =  0.14133719845144394
EPOCHS:  69 of  100 ==
training with  51201 of  51200 51200
Error =  0.1401916265253818
EPOCHS:  70 of  100 ==
training with  51201 of  51200
Error =  0.13890609554126634
EPOCHS:  71 of  100 ==
training with  51201 of  5120034817 of  5120051200 of  5120
0
Error =  0.13748963399948722
EPOCHS:  72 of  100 ==
training with  51201 of  51200 5120022017 of  51200
Error =  0.135948486061386
EPOCHS:  73 of  100 ==
training with  51201 of  512005120051200
Error =  0.13432016072047442
EPOCHS:  74 of  100 ==
training with  51201 of  51200
Error =  0.13264857406289626
EPOCHS:  75 of  100 ==
training with  51201 of  5120011137 of  51200of  5120046081
of  51200
Error =  0.13096824365712878
EPOCHS:  76 of  100 ==
training with  51201 of  51200f  5120047233 of  51200
Error =  0.12930481861803544
EPOCHS:  77 of  100 ==
training with  51201 of  5120012673 of  51200
Error =  0.12767373308020874
EPOCHS:  78 of  100 ==
training with  51201 of  51200512006273 of  5120051200 of
51200
Error =  0.126074084696218
EPOCHS:  79 of  100 ==
training with  51201 of  51200
Error =  0.12449285689134638
EPOCHS:  80 of  100 ==
training with  51201 of  51200993 of  51200
```

```
Error =  0.12291077252106032
EPOCHS:  81 of  100 ==
training with  51201 of  51200
Error =  0.1213058815542914
EPOCHS:  82 of  100 ==
training with  51201 of  51200of  51200 47489 of  51200
Error =  0.11965514831660211
EPOCHS:  83 of  100 ==
training with  51201 of  5120051200 of  51200
Error =  0.11793891954389409
EPOCHS:  84 of  100 ==
training with  51201 of  512003969 of  5120034945 of  51200
of  5120051200
Error =  0.11619294683911341
EPOCHS:  85 of  100 ==
training with  51201 of  512001537 of  51200of  5120051200
Error =  0.11456923062734789
EPOCHS:  86 of  100 ==
training with  51201 of  51200
Error =  0.11305968366329884
EPOCHS:  87 of  100 ==
training with  51201 of  51200f  51200 5120051200
Error =  0.11149651051395951
EPOCHS:  88 of  100 ==
training with  51201 of  512005120041601 of  5120051200
Error =  0.10983965390819997
EPOCHS:  89 of  100 ==
training with  51201 of  51200913 of  51200of  5120051200of
5120049409 of  51200
Error =  0.10812044406149776
EPOCHS:  90 of  100 ==
training with  51201 of  5120025 of  51200
Error =  0.10636613016271979
EPOCHS:  91 of  100 ==
training with  51201 of  5120021761 of  51200 of  51200
Error =  0.10459373176006026
EPOCHS:  92 of  100 ==
training with  51201 of  51200  512005377 of  5120022785 of
51200
Error =  0.10281382250570842
EPOCHS:  93 of  100 ==
training with  51201 of  51200 51200of  51200
Error =  0.10103261095713825
EPOCHS:  94 of  100 ==
training with  51201 of  512005120041601 of  51200
Error =  0.09925299750878096
EPOCHS:  95 of  100 ==
training with  51201 of  51200of  51200
Error =  0.09748580288731337
EPOCHS:  96 of  100 ==
training with  51201 of  51200049 of  51200of  51200
```

```
Error =  0.09574287418391958
EPOCHS:  97 of  100 ==
training with  51201 of  512005120043777 of  51200of  51200
Error =  0.0940267955309859
EPOCHS:  98 of  100 ==
training with  51201 of  51200 of  51200
Error =  0.09234055345282174
EPOCHS:  99 of  100 ==
training with  51201 of  51200
Error =  0.09068478380741671
EPOCHS:  100 of  100 ==
training with  51201 of  51200  51200of  51200
Error =  0.08906601378993298
```

In [185]:

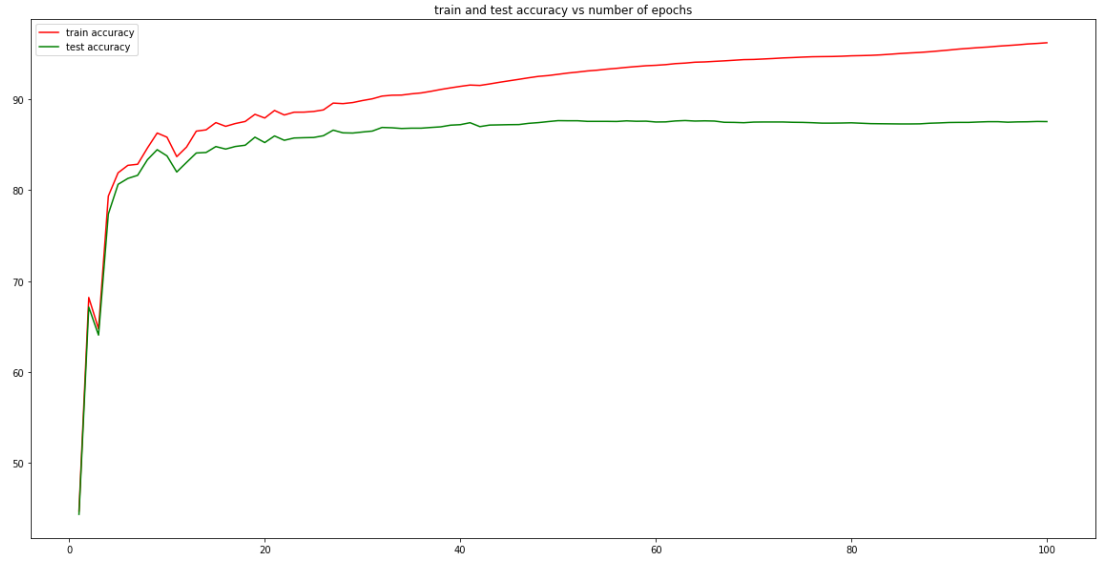
```
df_test = df[51200:60000]
test_labels = one_hot_targets[51200:60000]
df_test.shape
pred = net.predict(df_test)
print("accuracy: ",net.check_accuracy(df_test,test_labels))
```

```
accuracy:  87.52272727272727
```

The accuracy is better with 87.52%, and there is a significant 1% increase in validation accuracy. But the training time is increased as there are 100 epochs and 1024 nodes in hidden layer. I would love to see the plot how the training and test accuracy goes with the number of epochs

In [186]:

```
import matplotlib.pyplot as plt
%matplotlib inline
plt.figure(figsize=(20,10))
plt.title("train and test accuracy vs number of epochs")
plt.plot(net.epochs,net.train_accuracy,color='r',label='train accuracy')
plt.plot(net.epochs,net.test_accuracy,'g',label = 'test accuracy')
plt.legend()
plt.show()
```



accuracy is better with 1024 nodes in hidden layer. But the training accuracy increases monotonically but the test accuracy is constant after a fix number of epochs

In [213]:

```
net =Neural_NET(3, [784,38,10], ["relu","relu","softmax"], cost_func="cr
oss_entropy",test_plot=True,test_data=df_test,test_labels=test_labels)
net.train(128,df[0:51200], labels=one_hot_targets[0:51200], epochs=150,
learning_rate=0.001)
```

```
EPOCHS:  1 of  150 ==
training with  51201 of  51200
Error =  1.9592423159928816
EPOCHS:  2 of  150 ==
training with  51201 of  51200
Error =  1.8180102411699766
EPOCHS:  3 of  150 ==
training with  51201 of  51200
Error =  1.7345239121565774
EPOCHS:  4 of  150 ==
training with  51201 of  51200
Error =  1.6738011616948547
EPOCHS:  5 of  150 ==
training with  51201 of  5120051200
Error =  1.6201334738181892
EPOCHS:  6 of  150 ==
training with  51201 of  51200
Error =  1.5668610337241864
EPOCHS:  7 of  150 ==
training with  51201 of  51200
Error =  1.511073459208438
EPOCHS:  8 of  150 ==
training with  51201 of  51200
Error =  1.4540768317055672
EPOCHS:  9 of  150 ==
training with  51201 of  51200 20353 of  51200
Error =  1.4007993066387283
EPOCHS:  10 of  150 ==
training with  51201 of  51200of  51200
Error =  1.3518098148740887
EPOCHS:  11 of  150 ==
training with  51201 of  51200
Error =  1.3030154630071742
EPOCHS:  12 of  150 ==
training with  51201 of  51200
Error =  1.2565119831061677
EPOCHS:  13 of  150 ==
training with  51201 of  51200
Error =  1.2161961086967241
EPOCHS:  14 of  150 ==
training with  51201 of  51200
Error =  1.1816417097592966
EPOCHS:  15 of  150 ==
training with  51201 of  51200
Error =  1.152677504604243
EPOCHS:  16 of  150 ==
training with  51201 of  51200 39041 of  51200
Error =  1.12798801563203
EPOCHS:  17 of  150 ==
training with  51201 of  51200
```

```
Error =  1.1058680856968162
EPOCHS:  18 of  150 ==
training with  51201 of  51200
Error =  1.0789148001551414
EPOCHS:  19 of  150 ==
training with  51201 of  51200
Error =  1.0424656292213188
EPOCHS:  20 of  150 ==
training with  51201 of  5120051200
Error =  1.0045095359545242
EPOCHS:  21 of  150 ==
training with  51201 of  51200
Error =  0.9624732336044596
EPOCHS:  22 of  150 ==
training with  51201 of  51200
Error =  0.9198152165253096
EPOCHS:  23 of  150 ==
training with  51201 of  51200
Error =  0.8869230155605694
EPOCHS:  24 of  150 ==
training with  51201 of  51200 51200
Error =  0.863768985530281
EPOCHS:  25 of  150 ==
training with  51201 of  51200
Error =  0.8431938836399119
EPOCHS:  26 of  150 ==
training with  51201 of  51200
Error =  0.8247547711517185
EPOCHS:  27 of  150 ==
training with  51201 of  512001200
Error =  0.807308571805661
EPOCHS:  28 of  150 ==
training with  51201 of  51200
Error =  0.7914476499610444
EPOCHS:  29 of  150 ==
training with  51201 of  51200
Error =  0.7732965424063403
EPOCHS:  30 of  150 ==
training with  51201 of  51200
Error =  0.7562048038094281
EPOCHS:  31 of  150 ==
training with  51201 of  51200
Error =  0.7403629193376584
EPOCHS:  32 of  150 ==
training with  51201 of  51200
Error =  0.7263361595201325
EPOCHS:  33 of  150 ==
training with  51201 of  51200of  51200
Error =  0.7140981456030826
EPOCHS:  34 of  150 ==
training with  51201 of  51200of  51200
```

```
Error =  0.7022578670678785
EPOCHS:  35 of  150 ==
training with  51201 of  51200of  51200
Error =  0.6906909256493011
EPOCHS:  36 of  150 ==
training with  51201 of  51200
Error =  0.6795485259856563
EPOCHS:  37 of  150 ==
training with  51201 of  51200
Error =  0.6688703251552575
EPOCHS:  38 of  150 ==
training with  51201 of  51200
Error =  0.6586875230958751
EPOCHS:  39 of  150 ==
training with  51201 of  51200
Error =  0.6491037237680232
EPOCHS:  40 of  150 ==
training with  51201 of  51200
Error =  0.639064919511256
EPOCHS:  41 of  150 ==
training with  51201 of  51200
Error =  0.6292319854869948
EPOCHS:  42 of  150 ==
training with  51201 of  5120051200
Error =  0.6189940658561628
EPOCHS:  43 of  150 ==
training with  51201 of  51200
Error =  0.6086524051240106
EPOCHS:  44 of  150 ==
training with  51201 of  51200
Error =  0.5980558009211847
EPOCHS:  45 of  150 ==
training with  51201 of  51200
Error =  0.587075993211805
EPOCHS:  46 of  150 ==
training with  51201 of  5120051200
Error =  0.5764852527957898
EPOCHS:  47 of  150 ==
training with  51201 of  51200of  51200
Error =  0.566666002528784
EPOCHS:  48 of  150 ==
training with  51201 of  51200
Error =  0.5573124486456869
EPOCHS:  49 of  150 ==
training with  51201 of  51200
Error =  0.5490728753168628
EPOCHS:  50 of  150 ==
training with  51201 of  51200 47617 of  51200
Error =  0.5421539392718169
EPOCHS:  51 of  150 ==
training with  51201 of  51200
```

```
Error =  0.5364903634840645
EPOCHS:  52 of  150 ==
training with  51201 of  51200
Error =  0.5314265002603349
EPOCHS:  53 of  150 ==
training with  51201 of  51200 18561 of  51200
Error =  0.5268954670897091
EPOCHS:  54 of  150 ==
training with  51201 of  51200
Error =  0.5229967883172558
EPOCHS:  55 of  150 ==
training with  51201 of  51200
Error =  0.519520863649882
EPOCHS:  56 of  150 ==
training with  51201 of  51200
Error =  0.516246004320171
EPOCHS:  57 of  150 ==
training with  51201 of  51200
Error =  0.5130282845468503
EPOCHS:  58 of  150 ==
training with  51201 of  51200of  51200
Error =  0.5099768140659995
EPOCHS:  59 of  150 ==
training with  51201 of  51200of  51200
Error =  0.5070533272644553
EPOCHS:  60 of  150 ==
training with  51201 of  51200
Error =  0.5041222363656204
EPOCHS:  61 of  150 ==
training with  51201 of  51200
Error =  0.5014186127022933
EPOCHS:  62 of  150 ==
training with  51201 of  51200 51200
Error =  0.4989300656321165
EPOCHS:  63 of  150 ==
training with  51201 of  51200
Error =  0.49659929785338974
EPOCHS:  64 of  150 ==
training with  51201 of  51200
Error =  0.49439353433715094
EPOCHS:  65 of  150 ==
training with  51201 of  51200
Error =  0.49229142010910154
EPOCHS:  66 of  150 ==
training with  51201 of  5120011649 of  51200
Error =  0.49029602127475375
EPOCHS:  67 of  150 ==
training with  51201 of  51200
Error =  0.48841466661169786
EPOCHS:  68 of  150 ==
training with  51201 of  51200
```

```
Error =  0.48660744789118715
EPOCHS:  69 of  150 ==
training with  51201 of  51200
Error =  0.48491277280967476
EPOCHS:  70 of  150 ==
training with  51201 of  51200
Error =  0.48329252529193745
EPOCHS:  71 of  150 ==
training with  51201 of  51200
Error =  0.4817553551468078
EPOCHS:  72 of  150 ==
training with  51201 of  51200of  51200of  51200
Error =  0.4801271496619039
EPOCHS:  73 of  150 ==
training with  51201 of  51200
Error =  0.4785153205881375
EPOCHS:  74 of  150 ==
training with  51201 of  51200
Error =  0.47697734450753343
EPOCHS:  75 of  150 ==
training with  51201 of  51200
Error =  0.4754848333394439
EPOCHS:  76 of  150 ==
training with  51201 of  51200
Error =  0.4740365332444326
EPOCHS:  77 of  150 ==
training with  51201 of  5120051200
Error =  0.4726420127792804
EPOCHS:  78 of  150 ==
training with  51201 of  51200of  51200
Error =  0.47129328680391785
EPOCHS:  79 of  150 ==
training with  51201 of  51200
Error =  0.4699964968620157
EPOCHS:  80 of  150 ==
training with  51201 of  51200
Error =  0.4687756943918744
EPOCHS:  81 of  150 ==
training with  51201 of  51200
Error =  0.4676061995941214
EPOCHS:  82 of  150 ==
training with  51201 of  51200
Error =  0.4665018645546313
EPOCHS:  83 of  150 ==
training with  51201 of  51200
Error =  0.46543861552544774
EPOCHS:  84 of  150 ==
training with  51201 of  51200
Error =  0.46438505079386017
EPOCHS:  85 of  150 ==
training with  51201 of  5120051200
```

```
Error =  0.46335749956686223
EPOCHS:  86 of  150 ==
training with  51201 of  51200
Error =  0.46235370606182885
EPOCHS:  87 of  150 ==
training with  51201 of  5120051200
Error =  0.4613789150617102
EPOCHS:  88 of  150 ==
training with  51201 of  51200
Error =  0.4604386654440931
EPOCHS:  89 of  150 ==
training with  51201 of  51200
Error =  0.4595378545369433
EPOCHS:  90 of  150 ==
training with  51201 of  51200
Error =  0.458654782858819
EPOCHS:  91 of  150 ==
training with  51201 of  51200
Error =  0.4577357008692765
EPOCHS:  92 of  150 ==
training with  51201 of  51200
Error =  0.45681531611293147
EPOCHS:  93 of  150 ==
training with  51201 of  51200
Error =  0.4558883348096431
EPOCHS:  94 of  150 ==
training with  51201 of  51200of  51200
Error =  0.45498314207596213
EPOCHS:  95 of  150 ==
training with  51201 of  5120010241 of  51200
Error =  0.4540714190781925
EPOCHS:  96 of  150 ==
training with  51201 of  51200
Error =  0.45315083390423233
EPOCHS:  97 of  150 ==
training with  51201 of  51200f  51200
Error =  0.45224693046090014
EPOCHS:  98 of  150 ==
training with  51201 of  51200
Error =  0.4513437478738924
EPOCHS:  99 of  150 ==
training with  51201 of  51200
Error =  0.4504691611948328
EPOCHS:  100 of  150 ==
training with  51201 of  51200
Error =  0.4496424181302704
EPOCHS:  101 of  150 ==
training with  51201 of  51200
Error =  0.4488203487746368
EPOCHS:  102 of  150 ==
training with  51201 of  51200
```

```
Error =  0.4479987434146434
EPOCHS:  103 of  150 ==
training with  51201 of  51200
Error =  0.447152753851742
EPOCHS:  104 of  150 ==
training with  51201 of  51200of  51200
Error =  0.4462591102889661
EPOCHS:  105 of  150 ==
training with  51201 of  51200
Error =  0.4453785090530821
EPOCHS:  106 of  150 ==
training with  51201 of  51200
Error =  0.44449659202945757
EPOCHS:  107 of  150 ==
training with  51201 of  51200
Error =  0.4436313075990185
EPOCHS:  108 of  150 ==
training with  51201 of  51200
Error =  0.4427823917334077
EPOCHS:  109 of  150 ==
training with  51201 of  51200
Error =  0.44195834530448114
EPOCHS:  110 of  150 ==
training with  51201 of  51200
Error =  0.4411401667946504
EPOCHS:  111 of  150 ==
training with  51201 of  51200
Error =  0.44033874280686525
EPOCHS:  112 of  150 ==
training with  51201 of  5120044545 of  51200
Error =  0.4395047141203323
EPOCHS:  113 of  150 ==
training with  51201 of  51200
Error =  0.4386785579612432
EPOCHS:  114 of  150 ==
training with  51201 of  51200
Error =  0.4378736343577554
EPOCHS:  115 of  150 ==
training with  51201 of  51200
Error =  0.4370458771153679
EPOCHS:  116 of  150 ==
training with  51201 of  51200
Error =  0.4362419901317549
EPOCHS:  117 of  150 ==
training with  51201 of  51200
Error =  0.4354587310435321
EPOCHS:  118 of  150 ==
training with  51201 of  51200
Error =  0.4346795145755239
EPOCHS:  119 of  150 ==
training with  51201 of  51200
```

```
Error =  0.4339230868537123
EPOCHS:  120 of  150 ==
training with  51201 of  51200
Error =  0.4331769853608072
EPOCHS:  121 of  150 ==
training with  51201 of  51200
Error =  0.43241882729233383
EPOCHS:  122 of  150 ==
training with  51201 of  51200
Error =  0.4316604774191857
EPOCHS:  123 of  150 ==
training with  51201 of  51200
Error =  0.43090003449722714
EPOCHS:  124 of  150 ==
training with  51201 of  51200
Error =  0.43015303981359426
EPOCHS:  125 of  150 ==
training with  51201 of  51200
Error =  0.4293930090715655
EPOCHS:  126 of  150 ==
training with  51201 of  51200
Error =  0.4286410145984326
EPOCHS:  127 of  150 ==
training with  51201 of  51200
Error =  0.4278849190364489
EPOCHS:  128 of  150 ==
training with  51201 of  51200
Error =  0.42715171789717676
EPOCHS:  129 of  150 ==
training with  51201 of  51200
Error =  0.42640263746393103
EPOCHS:  130 of  150 ==
training with  51201 of  51200
Error =  0.4256563209708669
EPOCHS:  131 of  150 ==
training with  51201 of  5120051200
Error =  0.42493838824327246
EPOCHS:  132 of  150 ==
training with  51201 of  51200
Error =  0.42422057231908666
EPOCHS:  133 of  150 ==
training with  51201 of  51200
Error =  0.42351995635514944
EPOCHS:  134 of  150 ==
training with  51201 of  51200
Error =  0.4228407950407971
EPOCHS:  135 of  150 ==
training with  51201 of  51200
Error =  0.4221639149486535
EPOCHS:  136 of  150 ==
training with  51201 of  51200 51200
```

```
Error =  0.4214888050287369
EPOCHS:  137 of  150 ==
training with  51201 of  51200
Error =  0.42081219209714815
EPOCHS:  138 of  150 ==
training with  51201 of  51200 of  51200
Error =  0.42015043612841974
EPOCHS:  139 of  150 ==
training with  51201 of  51200
Error =  0.41949494370078455
EPOCHS:  140 of  150 ==
training with  51201 of  51200
Error =  0.4188532271560679
EPOCHS:  141 of  150 ==
training with  51201 of  51200
Error =  0.4182196807774545
EPOCHS:  142 of  150 ==
training with  51201 of  51200
Error =  0.4175713636550501
EPOCHS:  143 of  150 ==
training with  51201 of  51200
Error =  0.41692850239553636
EPOCHS:  144 of  150 ==
training with  51201 of  51200
Error =  0.4162559637382022
EPOCHS:  145 of  150 ==
training with  51201 of  51200
Error =  0.41557952929779907
EPOCHS:  146 of  150 ==
training with  51201 of  51200
Error =  0.4148888211157689
EPOCHS:  147 of  150 ==
training with  51201 of  51200 29057 of  51200
Error =  0.4141971268099286
EPOCHS:  148 of  150 ==
training with  51201 of  51200
Error =  0.4135082822507131
EPOCHS:  149 of  150 ==
training with  51201 of  51200
Error =  0.4128335228511636
EPOCHS:  150 of  150 ==
training with  51201 of  51200
Error =  0.4121507313385955
```

Here we can see that it is not converging even at 150 epochs. The cross entropy error is near about .41 and it hasnot decreased anymore. So this model haven't converged well enough.

In [214]:

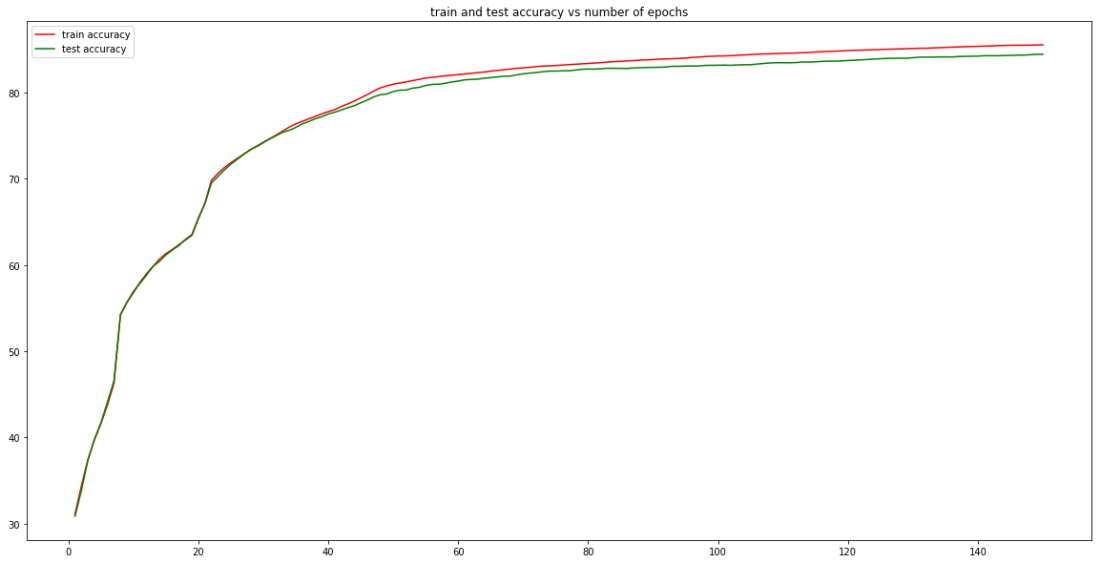
```
df_test = df[51200:60000]
test_labels = one_hot_targets[51200:60000]
df_test.shape
pred = net.predict(df_test)
print("accuracy: ",net.check_accuracy(df_test,test_labels))
```

```
accuracy:  84.4090909090909
```

We can see that even after 150 epochs the accuracy is 84.4%. Less that our previous two architechtures

In [216]:

```
import matplotlib.pyplot as plt
%matplotlib inline
plt.figure(figsize=(20,10))
plt.title("train and test accuracy vs number of epochs")
plt.plot(net.epochs,net.train_accuracy,color='r',label='train accuracy')
plt.plot(net.epochs,net.test_accuracy,'g',label = 'test accuracy')
plt.legend()
plt.show()
```



We can see that the increase in accuracy is constant but the pace is very slow. One thing we can do it increase the learning rate and see how it goes.

Training with 2048 nodes in one hidden layer. Used Sigmoid and Softmax in respective layers

In [217]:

```
net =Neural_NET(3, [784,2048,10], ["relu","sigmoid","softmax"], cost_fun
c="cross_entropy",test_plot=True,test_data=df_test,test_labels=test_labe
ls)
net.train(128,df[0:51200], labels=one_hot_targets[0:51200], epochs=150,
learning_rate=0.1)
```

```
EPOCHS:  1 of  150 ==
training with  51201 of  51200 51200 512005120051200
Error =  7.148063332141601
EPOCHS:  2 of  150 ==
training with  51201 of  5120014721 of  5120051200
Error =  4.741431187092488
EPOCHS:  3 of  150 ==
training with  51201 of  51200
Error =  1.9107741047441105
EPOCHS:  4 of  150 ==
training with  51201 of  512001200of  51200
Error =  1.9317379368921699
EPOCHS:  5 of  150 ==
training with  51201 of  5120051200
Error =  1.961379230099023
EPOCHS:  6 of  150 ==
training with  51201 of  51200of  51200
Error =  1.3135812550244619
EPOCHS:  7 of  150 ==
training with  51201 of  5120012005120051200 of  51200
Error =  1.6665917725227852
EPOCHS:  8 of  150 ==
training with  51201 of  51200553 of  51200 51200of  51200
Error =  1.5818992344722103
EPOCHS:  9 of  150 ==
training with  51201 of  51200
Error =  1.6478642638735463
EPOCHS:  10 of  150 ==
training with  51201 of  51200512005120051200
Error =  1.4098440561910852
EPOCHS:  11 of  150 ==
training with  51201 of  512007 of  51200 of  51200
Error =  0.5962592353659844
EPOCHS:  12 of  150 ==
training with  51201 of  51200 51200 42497 of  51200
Error =  1.5612745069675036
EPOCHS:  13 of  150 ==
training with  51201 of  51200of  51200 51200of  51200
Error =  1.2353219824595112
EPOCHS:  14 of  150 ==
training with  51201 of  5120020029569 of  5120051200 40705
of  51200
Error =  0.6070132148268693
EPOCHS:  15 of  150 ==
training with  51201 of  5120097 of  51200of  51200 22657 o
f  51200
Error =  0.5046657827074851
EPOCHS:  16 of  150 ==
training with  51201 of  512005120051200
Error =  1.1718829819105339
```

```
EPOCHS:  17 of  150 ==
training with  51201 of  51200of  51200of  51200 13697 of
51200
Error =  0.49254093933814486
EPOCHS:  18 of  150 ==
training with  51201 of  51200200 5120051200
Error =  0.5177059329921936
EPOCHS:  19 of  150 ==
training with  51201 of  5120051200of  51200
Error =  0.6182584016580173
EPOCHS:  20 of  150 ==
training with  51201 of  5120051200 512005120029825 of  512
00 48769 of  51200
Error =  0.579731385095018
EPOCHS:  21 of  150 ==
training with  51201 of  51200
Error =  0.5137686296042513
EPOCHS:  22 of  150 ==
training with  51201 of  5120051200
Error =  1.5233144797149736
EPOCHS:  23 of  150 ==
training with  51201 of  51200 15105 of  51200
Error =  0.4264332225805897
EPOCHS:  24 of  150 ==
training with  51201 of  5120051200
Error =  0.4512236187256733
EPOCHS:  25 of  150 ==
training with  51201 of  51200 51200
Error =  0.332592263628781
EPOCHS:  26 of  150 ==
training with  51201 of  51200of  51200of  5120049281 of  5
1200
Error =  0.45959215705474243
EPOCHS:  27 of  150 ==
training with  51201 of  512002003969 of  51200512005120051
20048897 of  51200
Error =  0.42398481148546796
EPOCHS:  28 of  150 ==
training with  51201 of  51200of  51200
Error =  0.4586024873644846
EPOCHS:  29 of  150 ==
training with  51201 of  51200of  51200
Error =  0.418844212629097
EPOCHS:  30 of  150 ==
training with  51201 of  51200  512005120051200
Error =  0.5042665445299475
EPOCHS:  31 of  150 ==
training with  51201 of  51200
Error =  1.2317106233349175
EPOCHS:  32 of  150 ==
training with  51201 of  51200
```

```
Error =  0.4409905636573051
EPOCHS:  33 of  150 ==
training with  51201 of  51200of  5120051200of  51200
Error =  0.43342625851938316
EPOCHS:  34 of  150 ==
training with  51201 of  51200f  5120051200
Error =  1.3184133707307666
EPOCHS:  35 of  150 ==
training with  51201 of  512005120051200
Error =  0.38434317711835786
EPOCHS:  36 of  150 ==
training with  51201 of  51200
Error =  0.49224422356888475
EPOCHS:  37 of  150 ==
training with  51201 of  51200of  51200
Error =  0.341189386419184 34
EPOCHS:  38 of  150 ==
training with  51201 of  51200 of  5120049281 of  51200
Error =  0.4709458619518526
EPOCHS:  39 of  150 ==
training with  51201 of  51200 of  51200
Error =  0.39136263474377153
EPOCHS:  40 of  150 ==
training with  51201 of  51200of  51200
Error =  0.42669122930462305
EPOCHS:  41 of  150 ==
training with  51201 of  512001200 51200
Error =  0.3382598924313565
EPOCHS:  42 of  150 ==
training with  51201 of  51200 51200 of  51200 51200
Error =  0.4005386997417728
EPOCHS:  43 of  150 ==
training with  51201 of  512005120051200 51200 51200
Error =  0.44827991694653646
EPOCHS:  44 of  150 ==
training with  51201 of  51200 51200 of  51200
Error =  0.2577700914587669
EPOCHS:  45 of  150 ==
training with  51201 of  512005120051200
Error =  0.3780704992855176
EPOCHS:  46 of  150 ==
training with  51201 of  5120051200of  51200 24705 of  5120
0 of  51200of  51200 51200
Error =  0.34139212453539036
EPOCHS:  47 of  150 ==
training with  51201 of  512006529 of  51200 51200 51200
Error =  0.28635635385623637
EPOCHS:  48 of  150 ==
training with  51201 of  5120051200 51200
Error =  0.33686248943233543
EPOCHS:  49 of  150 ==
```

```
training with  51201 of  5120051200of  51200 of  51200
Error =  0.29893981016628424
EPOCHS:  50 of  150 ==
training with  51201 of  5120051200 51200 7041 of  51200512
00 51200
Error =  0.2678224566297063
EPOCHS:  51 of  150 ==
training with  51201 of  5120051200
Error =  0.27586559688157497
EPOCHS:  52 of  150 ==
training with  51201 of  5120051200
Error =  0.2182766200009986
EPOCHS:  53 of  150 ==
training with  51201 of  5120019585 of  51200
Error =  0.2245049944474998
EPOCHS:  54 of  150 ==
training with  51201 of  51200 13825 of  51200 39297 of  51
200
Error =  0.20073012152286923
EPOCHS:  55 of  150 ==
training with  51201 of  51200 51200
Error =  0.18871592827563627
EPOCHS:  56 of  150 ==
training with  51201 of  51200 5120025217 of  5120051200
Error =  0.40564136626236447
EPOCHS:  57 of  150 ==
training with  51201 of  51200of  5120051200
Error =  0.19480439550558093
EPOCHS:  58 of  150 ==
training with  51201 of  51200of  51200 38017 of  51200of
5120048385 of  51200
Error =  0.17372901319274595
EPOCHS:  59 of  150 ==
training with  51201 of  51200120036481 of  51200
Error =  0.17876931659230097
EPOCHS:  60 of  150 ==
training with  51201 of  5120012003713 of  51200 51200of  5
1200of  51200
Error =  0.15346837149543902
EPOCHS:  61 of  150 ==
training with  51201 of  51200of  51200 51200
Error =  0.12386035631856687
EPOCHS:  62 of  150 ==
training with  51201 of  51200120051200
Error =  0.1499979464456967
EPOCHS:  63 of  150 ==
training with  51201 of  51200
Error =  0.11633839777441131
EPOCHS:  64 of  150 ==
training with  51201 of  512001200 12801 of  51200
Error =  0.13847920869934588
```

```
EPOCHS:  65 of  150 ==
training with  51201 of  51200 51200 22913 of  51200
Error =  0.12944478460872383
EPOCHS:  66 of  150 ==
training with  51201 of  51200of  51200
Error =  0.11063148583746858
EPOCHS:  67 of  150 ==
training with  51201 of  51200
Error =  0.10371946517958688
EPOCHS:  68 of  150 ==
training with  51201 of  51200
Error =  0.11328193413186999
EPOCHS:  69 of  150 ==
training with  51201 of  51200 of  51200
Error =  0.10270450726003874
EPOCHS:  70 of  150 ==
training with  51201 of  5120051200
Error =  0.12743930029168904
EPOCHS:  71 of  150 ==
training with  51201 of  51200
Error =  0.11042192331905454
EPOCHS:  72 of  150 ==
training with  51201 of  51200
Error =  0.10997728110346762
EPOCHS:  73 of  150 ==
training with  51201 of  5120051200
Error =  0.14110212138513184
EPOCHS:  74 of  150 ==
training with  51201 of  51200073 of  51200
Error =  0.10232125374401574
EPOCHS:  75 of  150 ==
training with  51201 of  5120030209 of  51200
Error =  0.09755934210020661
EPOCHS:  76 of  150 ==
training with  51201 of  5120023169 of  51200
Error =  0.09529438390076983
EPOCHS:  77 of  150 ==
training with  51201 of  51200
Error =  0.08804337853002935
EPOCHS:  78 of  150 ==
training with  51201 of  51200f  51200
Error =  0.1000695756493902
EPOCHS:  79 of  150 ==
training with  51201 of  51200 of  5120051200
Error =  0.09257180649009082
EPOCHS:  80 of  150 ==
training with  51201 of  5120051200
Error =  0.08932930901336872
EPOCHS:  81 of  150 ==
training with  51201 of  51200 51200
Error =  0.09656970511720595
```

```
EPOCHS:  82 of  150 ==
training with  51201 of  5120051200 51200 51200
Error =  0.07930121099914 37
EPOCHS:  83 of  150 ==
training with  51201 of  5120051200
Error =  0.08132125853182598
EPOCHS:  84 of  150 ==
training with  51201 of  51200 51200
Error =  0.07626168624392143
EPOCHS:  85 of  150 ==
training with  51201 of  5120020865 of  51200512005120047489 of  51200
Error =  0.07462703611027069
EPOCHS:  86 of  150 ==
training with  51201 of  51200
Error =  0.07340304312389782
EPOCHS:  87 of  150 ==
training with  51201 of  51200
Error =  0.07745327889041743
EPOCHS:  88 of  150 ==
training with  51201 of  51200of  51200of  51200of  51200
Error =  0.08436211669042425
EPOCHS:  89 of  150 ==
training with  51201 of  51200 51200
Error =  0.07357550405802768
EPOCHS:  90 of  150 ==
training with  51201 of  5120051200
Error =  0.07319341743800217
EPOCHS:  91 of  150 ==
training with  51201 of  51200 5120048769 of  51200
Error =  0.07187248526215126
EPOCHS:  92 of  150 ==
training with  51201 of  51200of  51200of  51200
Error =  0.08441176042948993
EPOCHS:  93 of  150 ==
training with  51201 of  51200
Error =  0.06723877722184521
EPOCHS:  94 of  150 ==
training with  51201 of  5120051200of  51200
Error =  0.06976142396420919
EPOCHS:  95 of  150 ==
training with  51201 of  5120051200 of  51200
Error =  0.06404366421267597
EPOCHS:  96 of  150 ==
training with  51201 of  512001200 5120051200
Error =  0.06356493092276197
EPOCHS:  97 of  150 ==
training with  51201 of  51200of  51200of  5120039297 of  51200
Error =  0.06659868076547665
EPOCHS:  98 of  150 ==
```

```
training with  51201 of  51200 51200
Error =  0.060729436025181625
EPOCHS:  99 of  150 ==
training with  51201 of  51200297 of  51200
Error =  0.05978575895704767
EPOCHS:  100 of  150 ==
training with  51201 of  51200 of  51200of  51200
Error =  0.058245591618082196
EPOCHS:  101 of  150 ==
training with  51201 of  51200 19073 of  51200
Error =  0.05917733458368552
EPOCHS:  102 of  150 ==
training with  51201 of  512005120051200of  51200
Error =  0.05086286636933146
EPOCHS:  103 of  150 ==
training with  51201 of  51200 51200 51200
Error =  0.046491079945640l4
EPOCHS:  104 of  150 ==
training with  51201 of  51200of  51200
Error =  0.0481170397751054
EPOCHS:  105 of  150 ==
training with  51201 of  5120051200 51200of  51200
Error =  0.06514526037294946
EPOCHS:  106 of  150 ==
training with  51201 of  512001200of  5120036225 of  51200o
f  51200
Error =  0.04762109433419763
EPOCHS:  107 of  150 ==
training with  51201 of  5120021377 of  51200 51200
Error =  0.051908959983653094
EPOCHS:  108 of  150 ==
training with  51201 of  51200 51200 512005120051200
Error =  0.05116143427592641
EPOCHS:  109 of  150 ==
training with  51201 of  5120051200
Error =  0.047080194794509325
EPOCHS:  110 of  150 ==
training with  51201 of  51200
Error =  0.05427871398144936
EPOCHS:  111 of  150 ==
training with  51201 of  51200
Error =  0.04452005275113689
EPOCHS:  112 of  150 ==
training with  51201 of  51200512005120037377 of  512004492
9 of  51200
Error =  0.04888532475642948
EPOCHS:  113 of  150 ==
training with  51201 of  51200
Error =  0.04167889839405134
EPOCHS:  114 of  150 ==
training with  51201 of  51200705 of  512005120051200
```

```
Error =  0.04563053905594806
EPOCHS:  115 of  150 ==
training with  51201 of  5120051200
Error =  0.045295977161844136
EPOCHS:  116 of  150 ==
training with  51201 of  51200 51200
Error =  0.04122416401554322
EPOCHS:  117 of  150 ==
training with  51201 of  51200
Error =  0.04337031769330607
EPOCHS:  118 of  150 ==
training with  51201 of  5120019841 of  51200of  51200
Error =  0.04943376414907502
EPOCHS:  119 of  150 ==
training with  51201 of  512005120034689 of  5120038017 of
51200 of  51200
Error =  0.040758486805338376
EPOCHS:  120 of  150 ==
training with  51201 of  512005120022913 of  51200
Error =  0.03914126240116583
EPOCHS:  121 of  150 ==
training with  51201 of  512003 of  51200of  51200 51200 89
61 of  512005120051200
Error =  0.04017282010708724
EPOCHS:  122 of  150 ==
training with  51201 of  5120047617 of  51200
Error =  0.040792626780611074
EPOCHS:  123 of  150 ==
training with  51201 of  51200 51200of  51200
Error =  0.04152418556549072
EPOCHS:  124 of  150 ==
training with  51201 of  5120051200 5120031233 of  51200of
5120051200
Error =  0.04053125598651054
EPOCHS:  125 of  150 ==
training with  51201 of  5120029953 of  51200
Error =  0.0382840112251792
EPOCHS:  126 of  150 ==
training with  51201 of  51200 27265 of  51200of  51200
Error =  0.036926627286544854
EPOCHS:  127 of  150 ==
training with  51201 of  51200of  5120051200 51200 51200
Error =  0.04261748990022672
EPOCHS:  128 of  150 ==
training with  51201 of  51200of  51200of  5120018945 of  5
1200of  51200
Error =  0.03962908456304785
EPOCHS:  129 of  150 ==
training with  51201 of  51200of  51200
Error =  0.04242112617110441
EPOCHS:  130 of  150 ==
```

```
training with  51201 of  512001200 5120051200of  51200of  5
1200 of  51200
Error =  0.036312478530118 79
EPOCHS:  131 of  150 ==
training with  51201 of  51200 of  5120023809 of  51200 512
00
Error =  0.0369407477295354
EPOCHS:  132 of  150 ==
training with  51201 of  51200120051200 of  51200
Error =  0.036285177089691 23
EPOCHS:  133 of  150 ==
training with  51201 of  51200 of  5120051200of  51200
Error =  0.034272195685590 31
EPOCHS:  134 of  150 ==
training with  51201 of  51200f  5120051200
Error =  0.034365323098802 87
EPOCHS:  135 of  150 ==
training with  51201 of  51200
Error =  0.032022348254314 81
EPOCHS:  136 of  150 ==
training with  51201 of  5120013 of  512005120051200of  512
005120046465 of  5120051200
Error =  0.032172643257899 83
EPOCHS:  137 of  150 ==
training with  51201 of  51200 of  51200
Error =  0.031387084388240 83
EPOCHS:  138 of  150 ==
training with  51201 of  5120027009 of  51200 51200of  5120
0
Error =  0.032120700796059 36
EPOCHS:  139 of  150 ==
training with  51201 of  51200of  51200 of  51200
Error =  0.032584763257049 48
EPOCHS:  140 of  150 ==
training with  51201 of  51200of  51200
Error =  0.031680069559216 36
EPOCHS:  141 of  150 ==
training with  51201 of  5120051200
Error =  0.030606772607745 816
EPOCHS:  142 of  150 ==
training with  51201 of  5120051200of  51200of  51200
Error =  0.030087783997302 05
EPOCHS:  143 of  150 ==
training with  51201 of  51200
Error =  0.029727660414843 947
EPOCHS:  144 of  150 ==
training with  51201 of  51200of  512005120030849 of  51200
Error =  0.029472420631441 103
EPOCHS:  145 of  150 ==
training with  51201 of  51200512005120032001 of  512005120
0
```

```
Error =  0.029269244380688033
EPOCHS:  146 of  150 ==
training with  51201 of  5120019457 of  51200
Error =  0.028926719762243605
EPOCHS:  147 of  150 ==
training with  51201 of  51200of  5120051200 51200 of  5120
0
Error =  0.0285621772036439
EPOCHS:  148 of  150 ==
training with  51201 of  51200 51200
Error =  0.028215086449119353
EPOCHS:  149 of  150 ==
training with  51201 of  51200of  51200 51200 5120051073 of
51200
Error =  0.027911624594033752
EPOCHS:  150 of  150 ==
training with  51201 of  51200of  51200of  5120034689 of  5
1200of  51200
Error =  0.0276227438126613
```

The cross entropy error has reduced to 0.027 and its quite good so our model performs very well in train data. But there is a chance of overfit. Lets check how it performs on unseen data.

In [218]:

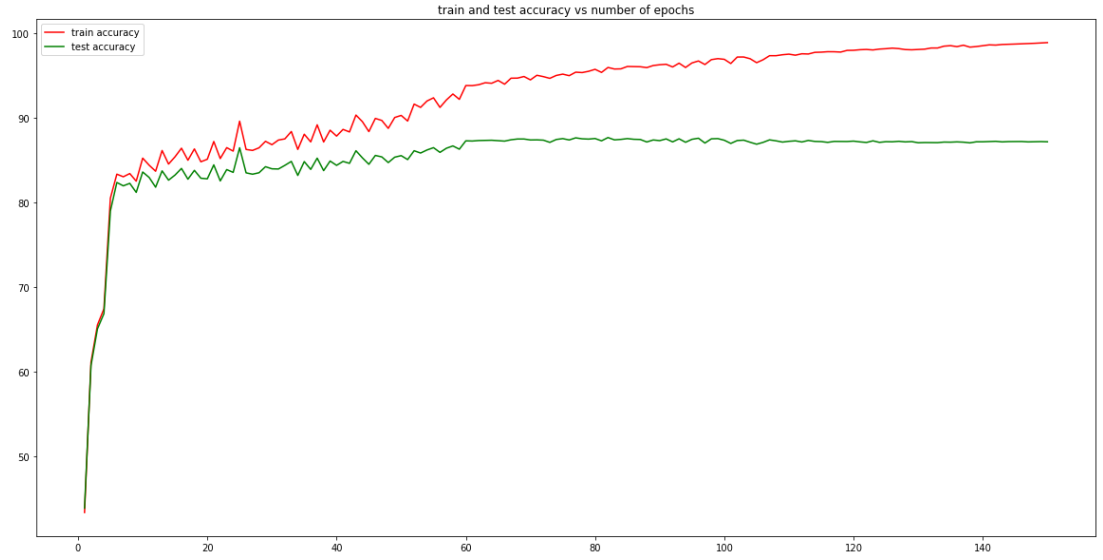
```
df_test = df[51200:60000]
test_labels = one_hot_targets[51200:60000]
df_test.shape
pred = net.predict(df_test)
print("accuracy: ",net.check_accuracy(df_test,test_labels))
```

```
accuracy:  87.13636363636364
```

Accuracy is not that good in test data according to its performance in training data.

In [219]:

```
import matplotlib.pyplot as plt
%matplotlib inline
plt.figure(figsize=(20,10))
plt.title("train and test accuracy vs number of epochs")
plt.plot(net.epochs,net.train_accuracy,color='r',label='train accuracy')
plt.plot(net.epochs,net.test_accuracy,'g',label = 'test accuracy')
plt.legend()
plt.show()
```



Here we can see that the validation accuracy is decreasing at some point. after approx 60 epochs. That means it overfits. So we set the number of epochs to 60.

In [221]:

```
net =Neural_NET(3, [784,2048,10], ["relu","sigmoid","softmax"], cost_fun
c="cross_entropy",test_plot=True,test_data=df_test,test_labels=test_labe
ls)
net.train(128,df[0:51200], labels=one_hot_targets[0:51200], epochs=60, l
earning_rate=0.1)
```

```
EPOCHS:  1 of  60 ==
training with  51201 of  51200 5120051200
Error =  6.7928999929804
EPOCHS:  2 of  60 ==
training with  51201 of  51200 5120051200 51200 51200
Error =  3.2197342004427627
EPOCHS:  3 of  60 ==
training with  51201 of  51200
Error =  3.1853497508255 65
EPOCHS:  4 of  60 ==
training with  51201 of  51200 of  51200of  51200
Error =  1.4118626727997694
EPOCHS:  5 of  60 ==
training with  51201 of  5120033409 of  51200
Error =  2.008023705705121
EPOCHS:  6 of  60 ==
training with  51201 of  51200 of  5120051200 51200
Error =  1.0396055321645123
EPOCHS:  7 of  60 ==
training with  51201 of  51200 of  51200
Error =  1.633603875656229
EPOCHS:  8 of  60 ==
training with  51201 of  51200561 of  512009473 of  5120016
769 of  51200of  51200 of  51200
Error =  0.643461791049366
EPOCHS:  9 of  60 ==
training with  51201 of  51200 5120051200
Error =  1.1811382675540818
EPOCHS:  10 of  60 ==
training with  51201 of  5120029697 of  5120051200 51200
Error =  0.7222640133700527
EPOCHS:  11 of  60 ==
training with  51201 of  5120024321 of  51200
Error =  0.5617742138332086
EPOCHS:  12 of  60 ==
training with  51201 of  512001200 51200
Error =  0.8989876529888076
EPOCHS:  13 of  60 ==
training with  51201 of  51200120050817 of  51200
Error =  0.5835289394065619
EPOCHS:  14 of  60 ==
training with  51201 of  51200 51200
Error =  0.5914382515246663
EPOCHS:  15 of  60 ==
training with  51201 of  51200512005120051200 51200of  5120
0
Error =  0.4827086965297353
EPOCHS:  16 of  60 ==
training with  51201 of  51200of  5120039809 of  5120042753
of  51200
```

```
Error =  0.4656368082608124
EPOCHS:  17 of  60 ==
training with  51201 of  51200
Error =  0.6917902008054958
EPOCHS:  18 of  60 ==
training with  51201 of  51200 18945 of  51200
Error =  1.4554237526728622
EPOCHS:  19 of  60 ==
training with  51201 of  51200of  51200 51200
Error =  0.5562965608508269
EPOCHS:  20 of  60 ==
training with  51201 of  51200
Error =  0.4392901563321285
EPOCHS:  21 of  60 ==
training with  51201 of  51200of  51200
Error =  0.5476252050085801
EPOCHS:  22 of  60 ==
training with  51201 of  5120051200of  51200
Error =  0.5017666614341906
EPOCHS:  23 of  60 ==
training with  51201 of  51200
Error =  0.4459634884539375
EPOCHS:  24 of  60 ==
training with  51201 of  51200of  51200of  51200
Error =  0.4813515392988288
EPOCHS:  25 of  60 ==
training with  51201 of  5120046337 of  51200
Error =  0.39200842661319
EPOCHS:  26 of  60 ==
training with  51201 of  512001200 5120051200
Error =  0.41811630961470847
EPOCHS:  27 of  60 ==
training with  51201 of  51200177 of  512005120051200
Error =  0.4723436981125013
EPOCHS:  28 of  60 ==
training with  51201 of  5120051200
Error =  1.0574905382630393
EPOCHS:  29 of  60 ==
training with  51201 of  512001200
Error =  0.4790890518181592
EPOCHS:  30 of  60 ==
training with  51201 of  51200f  512005120014721 of  51200
51200of  51200
Error =  0.5636975930488088
EPOCHS:  31 of  60 ==
training with  51201 of  51200 51200
Error =  0.37457230050371876
EPOCHS:  32 of  60 ==
training with  51201 of  5120022785 of  5120033281 of  5120
0
Error =  0.397728632143351
```

```
EPOCHS:  33 of  60 ==
training with  51201 of  51200of  5120051200 51200
Error =  0.4542388930317292
EPOCHS:  34 of  60 ==
training with  51201 of  512003 of  5120051200 12033 of  51
200 51200
Error =  0.41085878259689124
EPOCHS:  35 of  60 ==
training with  51201 of  5120051200
Error =  0.40780705794545175
EPOCHS:  36 of  60 ==
training with  51201 of  51200120030849 of  51200 51200
Error =  0.3748241969634909
EPOCHS:  37 of  60 ==
training with  51201 of  51200 of  5120051200
Error =  0.39114639538127643
EPOCHS:  38 of  60 ==
training with  51201 of  51200of  51200of  51200
Error =  0.39183438787912633
EPOCHS:  39 of  60 ==
training with  51201 of  51200 of  5120051200
Error =  0.2976585227838282
EPOCHS:  40 of  60 ==
training with  51201 of  51200 51200
Error =  0.36335881942089926
EPOCHS:  41 of  60 ==
training with  51201 of  51200 5120046849 of  51200
Error =  0.5707476774887205
EPOCHS:  42 of  60 ==
training with  51201 of  5120018561 of  51200of  5120051200
46465 of  51200
Error =  0.3849290166393782
EPOCHS:  43 of  60 ==
training with  51201 of  51200of  5120025729 of  5120032641
of  51200
Error =  0.683840110351984
EPOCHS:  44 of  60 ==
training with  51201 of  51200 of  51200
Error =  0.34218930181959895
EPOCHS:  45 of  60 ==
training with  51201 of  5120032129 of  51200 51200
Error =  0.380995175109705
EPOCHS:  46 of  60 ==
training with  51201 of  51200of  5120042881 of  51200of  5
1200
Error =  0.3853269946256071
EPOCHS:  47 of  60 ==
training with  51201 of  51200of  51200 25601 of  51200of
51200of  51200
Error =  0.29946916705211635
EPOCHS:  48 of  60 ==
```

```
training with  51201 of  5120041601 of  5120051200
Error =  0.3209138159535726
EPOCHS:  49 of  60 ==
training with  51201 of  512005633 of  5120020225 of  51200
51200 51200
Error =  0.2928293996665304
EPOCHS:  50 of  60 ==
training with  51201 of  51200120051200 51200of  51200
Error =  0.3523544820508985
EPOCHS:  51 of  60 ==
training with  51201 of  51200 of  5120051200
Error =  0.27983537591186725
EPOCHS:  52 of  60 ==
training with  51201 of  51200of  5120016385 of  51200 of
51200
Error =  0.2711809616960111
EPOCHS:  53 of  60 ==
training with  51201 of  512004737 of  512006145 of  51200o
f  51200
Error =  0.2649370890504742
EPOCHS:  54 of  60 ==
training with  51201 of  51200 51200
Error =  0.23855769745652783
EPOCHS:  55 of  60 ==
training with  51201 of  51200of  5120051200
Error =  0.36156626011606735
EPOCHS:  56 of  60 ==
training with  51201 of  51200 51200
Error =  0.282532707158748
EPOCHS:  57 of  60 ==
training with  51201 of  51200f  51200 51200
Error =  0.27736363914045
EPOCHS:  58 of  60 ==
training with  51201 of  51200 51200
Error =  0.30627303713975995
EPOCHS:  59 of  60 ==
training with  51201 of  51200f  51200
Error =  0.2226755354538787
EPOCHS:  60 of  60 ==
training with  51201 of  51200 of  51200 51200of  51200
Error =  0.2533693307028749
```

In [222]:

```
print("accuracy: ",net.check_accuracy(df_test,test_labels))
```

```
accuracy:  85.30681818181819
```

Checking the accuracy plot with 100 epochs and 512 nodes in hidden layers

In [223]:

```
net =Neural_NET(3, [784,512,10], ["sigmoid","sigmoid","sigmoid"], cost_f
unc="cross_entropy",test_plot=True,test_data=df_test,test_labels=test_la
bels)
net.train(128,df[0:51200], labels=one_hot_targets[0:51200], epochs=100,
learning_rate=0.1)
```

```
EPOCHS:  1 of  100 ==
training with  51201 of  51200f  51200of  51200 of  51200 5
1200
Error =  1.2170214083116184
EPOCHS:  2 of  100 ==
training with  51201 of  51200
Error =  0.7560094096141766
EPOCHS:  3 of  100 ==
training with  51201 of  51200 51200
Error =  0.5297749986708881
EPOCHS:  4 of  100 ==
training with  51201 of  5120051200
Error =  0.4876339294322766
EPOCHS:  5 of  100 ==
training with  51201 of  5120051200
Error =  0.4535961748368139
EPOCHS:  6 of  100 ==
training with  51201 of  5120013825 of  51200 51200of  5120
0
Error =  0.42793108951802006
EPOCHS:  7 of  100 ==
training with  51201 of  5120051200of  5120046081 of  51200
Error =  0.40893077001061845
EPOCHS:  8 of  100 ==
training with  51201 of  51200609 of  5120014209 of  512003
0849 of  51200
Error =  0.39449231930770146
EPOCHS:  9 of  100 ==
training with  51201 of  51200 of  5120051200
Error =  0.3830087832827387
EPOCHS:  10 of  100 ==
training with  51201 of  5120014337 of  5120051200 51200
Error =  0.3732800049396424
EPOCHS:  11 of  100 ==
training with  51201 of  51200
Error =  0.3651253290364652
EPOCHS:  12 of  100 ==
training with  51201 of  5120030849 of  51200
Error =  0.3577014673236621
EPOCHS:  13 of  100 ==
training with  51201 of  5120026881 of  51200 51200
Error =  0.35070599471322433
EPOCHS:  14 of  100 ==
training with  51201 of  51200of  51200
Error =  0.34408649226901833
EPOCHS:  15 of  100 ==
training with  51201 of  51200305 of  5120043393 of  51200
Error =  0.3379197706192969
EPOCHS:  16 of  100 ==
training with  51201 of  51200865 of  5120019201 of  51200
```

```
Error =  0.3321999660442277
EPOCHS:  17 of  100 ==
training with  51201 of  51200of  51200of  51200
Error =  0.3268308961105787
EPOCHS:  18 of  100 ==
training with  51201 of  5120016385 of  51200
Error =  0.32177165553650516
EPOCHS:  19 of  100 ==
training with  51201 of  5120026113 of  51200of  51200
Error =  0.3169297691081244
EPOCHS:  20 of  100 ==
training with  51201 of  51200
Error =  0.3122396887113237
EPOCHS:  21 of  100 ==
training with  51201 of  512005120031489 of  51200
Error =  0.30769796316173276
EPOCHS:  22 of  100 ==
training with  51201 of  51200512005120041729 of  51200
Error =  0.30339263342661393
EPOCHS:  23 of  100 ==
training with  51201 of  5120051200
Error =  0.2993182715802443
EPOCHS:  24 of  100 ==
training with  51201 of  51200 of  51200 51200
Error =  0.29543121034387176
EPOCHS:  25 of  100 ==
training with  51201 of  5120051200 5120038017 of  51200
Error =  0.2916904723391643
EPOCHS:  26 of  100 ==
training with  51201 of  51200of  51200of  51200
Error =  0.2880594983988411
EPOCHS:  27 of  100 ==
training with  51201 of  51200of  51200
Error =  0.2845083741985695
EPOCHS:  28 of  100 ==
training with  51201 of  51200of  51200
Error =  0.2810152253387994
EPOCHS:  29 of  100 ==
training with  51201 of  51200
Error =  0.2775989433311503
EPOCHS:  30 of  100 ==
training with  51201 of  5120021633 of  51200
Error =  0.2742585730000495
EPOCHS:  31 of  100 ==
training with  51201 of  51200
Error =  0.2709860735199419
EPOCHS:  32 of  100 ==
training with  51201 of  51200
Error =  0.2677785203275162
EPOCHS:  33 of  100 ==
training with  51201 of  51200of  51200
```

```
Error =  0.2646353382684795
EPOCHS:  34 of  100 ==
training with  51201 of  51200
Error =  0.26155914239852274
EPOCHS:  35 of  100 ==
training with  51201 of  5120022913 of  51200of  51200
Error =  0.2585495328093045
EPOCHS:  36 of  100 ==
training with  51201 of  51200
Error =  0.25559732849414585
EPOCHS:  37 of  100 ==
training with  51201 of  5120044545 of  51200
Error =  0.2526929824166974
EPOCHS:  38 of  100 ==
training with  51201 of  51200of  51200of  51200of  51200
Error =  0.24983979719868848
EPOCHS:  39 of  100 ==
training with  51201 of  51200737 of  5120028545 of  51200
Error =  0.2470439026298931
EPOCHS:  40 of  100 ==
training with  51201 of  51200
Error =  0.24431533177591552
EPOCHS:  41 of  100 ==
training with  51201 of  51200 of  5120038145 of  51200
Error =  0.2416640673611202
EPOCHS:  42 of  100 ==
training with  51201 of  51200of  51200
Error =  0.23909128172676442
EPOCHS:  43 of  100 ==
training with  51201 of  5120038785 of  51200
Error =  0.23659974426739044
EPOCHS:  44 of  100 ==
training with  51201 of  51200f  51200
Error =  0.2341859877537955
EPOCHS:  45 of  100 ==
training with  51201 of  51200
Error =  0.23183295743061094
EPOCHS:  46 of  100 ==
training with  51201 of  51200f  51200 51200of  51200of  51
200
Error =  0.22953844119065064
EPOCHS:  47 of  100 ==
training with  51201 of  51200 of  51200
Error =  0.2273029087190231
EPOCHS:  48 of  100 ==
training with  51201 of  51200of  51200of  51200
Error =  0.22513014002004522
EPOCHS:  49 of  100 ==
training with  51201 of  51200473 of  51200
Error =  0.22301724686474889
EPOCHS:  50 of  100 ==
```

```
training with  51201 of  51200
Error =  0.22095471626457353
EPOCHS:  51 of  100 ==
training with  51201 of  5120011905 of  51200
Error =  0.2189268502773306
EPOCHS:  52 of  100 ==
training with  51201 of  51200217 of  512005120035969 of  5
1200
Error =  0.21692490645295734
EPOCHS:  53 of  100 ==
training with  51201 of  51200 51200 51200 47105 of  51200
Error =  0.21492180999082922
EPOCHS:  54 of  100 ==
training with  51201 of  51200f  51200of  51200of  51200
Error =  0.21292973873561885
EPOCHS:  55 of  100 ==
training with  51201 of  51200
Error =  0.21096834436311038
EPOCHS:  56 of  100 ==
training with  51201 of  51200 51200of  51200
Error =  0.2090148677831615
EPOCHS:  57 of  100 ==
training with  51201 of  5120051200
Error =  0.207077234322265
EPOCHS:  58 of  100 ==
training with  51201 of  51200of  51200of  5120049281 of  5
1200
Error =  0.20513570841637654
EPOCHS:  59 of  100 ==
training with  51201 of  51200f  51200
Error =  0.20315862940289603
EPOCHS:  60 of  100 ==
training with  51201 of  5120042625 of  51200
Error =  0.20115362539036524
EPOCHS:  61 of  100 ==
training with  51201 of  51200120051200
Error =  0.19913310610125767
EPOCHS:  62 of  100 ==
training with  51201 of  51200089 of  512005120048513 of  5
1200
Error =  0.19710392707256086
EPOCHS:  63 of  100 ==
training with  51201 of  51200 5120042881 of  5120047233 of
51200
Error =  0.19507016283558512
EPOCHS:  64 of  100 ==
training with  51201 of  51200
Error =  0.1930350400227896
EPOCHS:  65 of  100 ==
training with  51201 of  51200
Error =  0.19100132171333145
```

```
EPOCHS:  66 of  100 ==
training with  51201 of  51200of  51200
Error =  0.18897112590171217
EPOCHS:  67 of  100 ==
training with  51201 of  51200
Error =  0.18694699998994974
EPOCHS:  68 of  100 ==
training with  51201 of  51200
Error =  0.18493341416365083
EPOCHS:  69 of  100 ==
training with  51201 of  5120039809 of  51200
Error =  0.1829348781639441
EPOCHS:  70 of  100 ==
training with  51201 of  51200 of  51200
Error =  0.18095444403751365
EPOCHS:  71 of  100 ==
training with  51201 of  5120012000f  51200
Error =  0.17899238597455286
EPOCHS:  72 of  100 ==
training with  51201 of  5120051200of  51200
Error =  0.17704477964808352
EPOCHS:  73 of  100 ==
training with  51201 of  51200of  51200
Error =  0.17512885575504608
EPOCHS:  74 of  100 ==
training with  51201 of  51200
Error =  0.17325796378857614
EPOCHS:  75 of  100 ==
training with  51201 of  5120051200512000f  5120051200 5120
032257 of  51200
Error =  0.17142278207473946
EPOCHS:  76 of  100 ==
training with  51201 of  51200of  51200
Error =  0.16962373207497525
EPOCHS:  77 of  100 ==
training with  51201 of  51200 of  51200
Error =  0.16786041066734267
EPOCHS:  78 of  100 ==
training with  51201 of  5120011649 of  51200 51200
Error =  0.16612901581404738
EPOCHS:  79 of  100 ==
training with  51201 of  51200
Error =  0.1644259354370226
EPOCHS:  80 of  100 ==
training with  51201 of  5120018945 of  51200of  5120042881
of  51200
Error =  0.16274944893687243
EPOCHS:  81 of  100 ==
training with  51201 of  5120051200185 61 of  5120037889 of
51200
Error =  0.16110059533862872
```

```
EPOCHS:  82 of  100 ==
training with  51201 of  5120051200
Error =  0.1594838618156592
EPOCHS:  83 of  100 ==
training with  51201 of  51200of  5120051200
Error =  0.15790559363375792
EPOCHS:  84 of  100 ==
training with  51201 of  5120029697 of  5120034561 of  5120
0
Error =  0.1563699056827207
EPOCHS:  85 of  100 ==
training with  51201 of  51200of  5120051200
Error =  0.15487651254775037
EPOCHS:  86 of  100 ==
training with  51201 of  51200
Error =  0.15342236903945436
EPOCHS:  87 of  100 ==
training with  51201 of  51200
Error =  0.1520041149661141
EPOCHS:  88 of  100 ==
training with  51201 of  51200481 of  5120021761 of  51200
Error =  0.15061920636752346
EPOCHS:  89 of  100 ==
training with  51201 of  51200249 of  51200of  51200
Error =  0.14926594565666262
EPOCHS:  90 of  100 ==
training with  51201 of  51200 of  51200
Error =  0.14794350899518316
EPOCHS:  91 of  100 ==
training with  51201 of  51200of  51200
Error =  0.1466528141761856
EPOCHS:  92 of  100 ==
training with  51201 of  5120043009 of  51200
Error =  0.1453928391511371
EPOCHS:  93 of  100 ==
training with  51201 of  512001200 51200
Error =  0.14416192920698023
EPOCHS:  94 of  100 ==
training with  51201 of  51200of  51200
Error =  0.14295971266104388
EPOCHS:  95 of  100 ==
training with  51201 of  5120051200
Error =  0.14178516586926398
EPOCHS:  96 of  100 ==
training with  51201 of  51200
Error =  0.14063693449812265
EPOCHS:  97 of  100 ==
training with  51201 of  51200 5120048385 of  51200
Error =  0.139513125557223
EPOCHS:  98 of  100 ==
training with  51201 of  51200 51200
```

```
Error =  0.13841168960617609
EPOCHS:  99 of  100 ==
training with  51201 of  5120011777 of  5120025857 of  5120
042369 of  51200
Error =  0.1373308561368706
EPOCHS:  100 of  100 ==
training with  51201 of  51200f  5120014337 of  51200
Error =  0.13626918121727
```

The cross entropy error has reduced significantly and I hope it will perform well even with unseen data

In [226]:

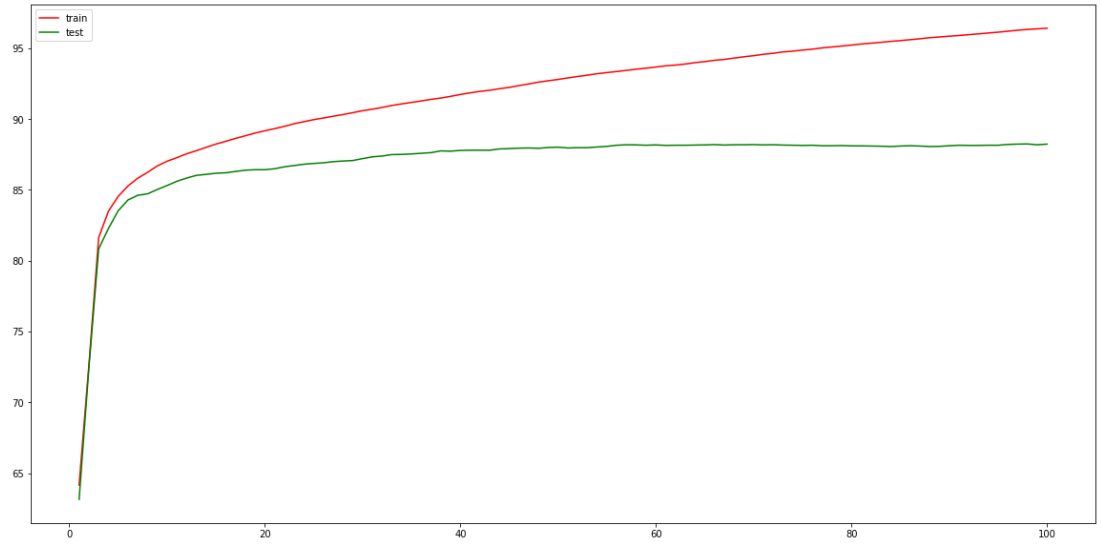
```
print("accuracy: ",net.check_accuracy(df_test,test_labels))
```

```
accuracy:  88.22727272727273
```

This architechture gives the best accuracy on test data as of now. We make this architechture our best architecture.

In [225]:

```
import matplotlib.pyplot as plt
%matplotlib inline
plt.figure(figsize=(20,10))
plt.plot(net.epochs,net.train_accuracy,color='r',label='train')
plt.plot(net.epochs,net.test_accuracy,'g',label = 'test')
plt.legend()
plt.show()
```



This turns out to be the best performing architechture with one hidden layer and 512 nodes gives accuracy about 89 percent.

In [259]:

```
np.savetxt("architecture_details/weights0.txt",net.Layers[0].weights)
np.savetxt("Architecture_details/weights1.txt",net.Layers[1].weights)
np.savetxt("Architecture_details/bias0.txt",net.Layers[0].bias)
np.savetxt("Architecture_details/bias1.txt",net.Layers[1].bias)
# saving the weights and bias into files
```

In [265]:

```
#import the test file
test_sample = pd.read_csv("apparel-test.csv")
test_sample = StandardScaler().fit_transform(test_sample)
print(test_sample.shape)
```

```
(9674, 784)
```

In [272]:

```
pred = net.predict(test_sample)
pred = [np.where(r==1)[0][0] for r in pred]
pred
np.savetxt("2018201010_apparel_prediction.csv",pred,fmt="%d") #saving th
e predictions
```

Question 2

In [235]:

```
import pandas as pd
train = pd.read_csv("house_price/train.csv")
test = pd.read_csv("house_price/test.csv")
print(train.shape)
train.head()
```

```
(1460, 81)
```

Out[235]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alle
0	1	60	RL	65.0	8450	Pave	NaN
1	2	20	RL	80.0	9600	Pave	NaN
2	3	60	RL	68.0	11250	Pave	NaN
3	4	70	RL	60.0	9550	Pave	NaN
4	5	60	RL	84.0	14260	Pave	NaN

5 rows × 81 columns

In [236]:

```
train.drop('Id',axis = 1, inplace = True)
train_numerical = train.select_dtypes(exclude=['object'])
train_numerical.fillna(0,inplace = True)
train_categoric = train.select_dtypes(include=['object'])
train_categoric.fillna('NONE',inplace = True)
train = train_numerical.merge(train_categoric, left_index = True, right_
index = True)


ID = test.Id
test.drop('Id',axis = 1, inplace = True)
test_numerical = test.select_dtypes(exclude=['object'])
test_numerical.fillna(0,inplace = True)
test_categoric = test.select_dtypes(include=['object'])
test_categoric.fillna('NONE',inplace = True)
test = test_numerical.merge(test_categoric, left_index = True, right_ind
ex = True)
```

In [237]:

```
from sklearn.ensemble import IsolationForest

clf = IsolationForest(max_samples = 100, random_state = 42)
clf.fit(train_numerical)
y_noano = clf.predict(train_numerical)
y_noano = pd.DataFrame(y_noano, columns = ['Top'])
y_noano[y_noano['Top'] == 1].index.values

train_numerical = train_numerical.iloc[y_noano[y_noano['Top'] == 1].inde
x.values]
train_numerical.reset_index(drop = True, inplace = True)

train_categoric = train_categoric.iloc[y_noano[y_noano['Top'] == 1].inde
x.values]
train_categoric.reset_index(drop = True, inplace = True)

train = train.iloc[y_noano[y_noano['Top'] == 1].index.values]
train.reset_index(drop = True, inplace = True)
```

In [239]:

```
import warnings
warnings.filterwarnings('ignore')
from sklearn.preprocessing import MinMaxScaler
col_train_num = list(train_numerical.columns)
col_train_num_bis = list(train_numerical.columns)

col_train_cat = list(train_categoric.columns)

col_train_num_bis.remove('SalePrice')

mat_train = np.matrix(train_numerical)
mat_test  = np.matrix(test_numerical)
mat_new = np.matrix(train_numerical.drop('SalePrice',axis = 1))
mat_y = np.array(train.SalePrice)

prepro_y = MinMaxScaler()
prepro_y.fit(mat_y.reshape(1314,1))

prepro = MinMaxScaler()
prepro.fit(mat_train)

prepro_test = MinMaxScaler()
prepro_test.fit(mat_new)

train_num_scale = pd.DataFrame(prepro.transform(mat_train),columns = col
_train)
test_num_scale  = pd.DataFrame(prepro_test.transform(mat_test),columns =
col_train_bis)
train_num_scale.head()
```

Out[239]:

	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	Y
0	0.235294	0.207668	0.062802	0.625	0.428571	0
1	0.000000	0.255591	0.072904	0.500	0.857143	0
2	0.235294	0.217252	0.087396	0.625	0.428571	0
3	0.294118	0.191693	0.072464	0.625	0.428571	0
4	0.235294	0.268371	0.113835	0.750	0.428571	0

5 rows × 37 columns

In [242]:

```
train[col_train_num] = pd.DataFrame(prepro.transform(mat_train),columns
= col_train_num)
test[col_train_num_bis]  = test_num_scale
test.head()
```

Out[242]:

	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	Y
0	0.000000	0.255591	0.090664	0.375	0.571429	0
1	0.000000	0.258786	0.113896	0.500	0.571429	0
2	0.235294	0.236422	0.110058	0.375	0.428571	0
3	0.235294	0.249201	0.076224	0.500	0.571429	0
4	0.588235	0.137380	0.032543	0.750	0.428571	0

5 rows × 79 columns

Report

1. I have shown the steps of preprocessing above.

> split into two parts

a. Set with only numerical features b. Set with only categorical features

> Normalise the numerical features as gradient descent will not work well if the features are not scaled.
>
> One hot encode the categorical features(dimensions of the data will increase.
>
> Remove the outliers.
>
> Merge the two sets and create new train data
>
> Do the above steps for test data also

1. The output layer will contain only one node as it is a regression problem.
2. We should use a linear activation function (ReLU or Leaky ReLU) in the output layer.
3. Hidden layers may contain relu, sigmoid, softmax or tanh as activation function.
4. Increase the number of hidden layers if the model does not seems to converge.
5. As it is a hard problem we might have to use more feature engineering.
6. The cost function is different in case of regression, we can use least mean squares.

By-

Souparna Das

20182010101

IIIT HYDERABAD

26 Feb, 2019