CSCI 402
Dr. Jianna Jian Zhang

Assignment 2: Classify Mushrooms/Digits Using BP-ANN

Data Source 1: UCI ML Repository https://www.kaggle.com/uciml/datasets

Data Source 2: MNIST database at http://yann.lecun.com/exdb/mnist/

Requires:

1. Preprocess (if needed) the original data into a format that can be used by BP-ANN.
   a) There are *n* attributes (which attribute is the decision)
   b) Each attribute has its own value (different number of value)
2. Suggestion: using the Mushroom Data:
   a) There are 8,124 examples with desired output: you need to randomly choose 15% as testing data and use the rest 85% as training data
   b) There are two classes: p (poison, 48.2%) or e (edible, 51.8%)
   c) The first attribute is the desired output, and the rest are features of the mushroom to be classified
   d) Attribute Information:
      1) cap-shape: bell=b, conical=c, convex=x, flat=f, knobbed=k, sunken=s
      2) cap-surface: fibrous=f, grooves=g, scaly=y, smooth=s
      3) cap-color: brown=n, buff=b, cinnamon=c, gray=g, green=r, pink=p, purple=u, red=e, white=w, yellow=y
      4) bruises?: bruises=t,no=f
      5) odor: almond=a,anise=l,creosote=c,fishy=y,foul=f, musty=m,none=n,pungent=p,spicy=s
      6) gill-attachment: attached=a,descending=d,free=f,notched=n
      7) gill-spacing: close=c,crowded=w,distant=d
      8) gill-size: broad=b,narrow=n
      9) gill-color: black=k,brown=n,buff=b,chocolate=h,gray=g, green=r,orange=o,pink=p,purple=u,red=e, white=w,yellow=y
      10) stalk-shape: enlarging=e,tapering=t
      11) stalk-root: bulbous=b,club=c,cup=u,equal=e, rhizomorphs=z,rooted=r,missing=?
      12) stalk-surface-above-ring: fibrous=f,scaly=y,silky=k,smooth=s
      13) stalk-surface-below-ring: fibrous=f,scaly=y,silky=k,smooth=s
      14) stalk-color-above-ring: brown=n,buff=b,cinnamon=c,gray=g,orange=o, pink=p,red=e,white=w,yellow=y

15) stalk-color-below-ring: brown=n,buff=b,cinnamon=c,gray=g,orange=o, pink=p,red=e,white=w,yellow=y

16) veil-type: partial=p,universal=u

17) veil-color: brown=n,orange=o,white=w,yellow=y

18) ring-number: none=n,one=o,two=t

19) ring-type: cobwebby=c,evanescent=e,flaring=f,large=l, none=n,pendant=p,sheathing=s,zone=z

20) spore-print-color: black=k,brown=n,buff=b,chocolate=h,green=r, orange=o,purple=u,white=w,yellow=y

21) population: abundant=a,clustered=c,numerous=n, scattered=s,several=v,solitary=y

22) habitat: grasses=g,leaves=l,meadows=m,paths=p, urban=u,waste=w,woods=d

e) Sample data: p,x,s,n,t,p,f,c,n,k,e,e,s,s,w,w,p,w,o,p,k,s,u
e,x,s,y,t,a,f,c,b,k,e,c,s,s,w,w,p,w,o,p,n,n,g
e,b,s,w,t,l,f,c,b,n,e,c,s,s,w,w,p,w,o,p,n,n,m

f) By 2004, researchers clam that several approach can gain 99-100% accuracy

3. Suggestion 2: MNIST Data: hand writing digit number recognition
   a) There are 60,000 training data
   b) There are 10,000 test data
   c) Read this paper: https://arxiv.org/pdf/1003.0358v1.pdf
4. Use BP-ANN algorithm to build your own software to learn whether or not a give mushroom is poison or edible, or recognize the hand written digits.
5. Sample BP-ANN procedure applying to the mushroom/MNIST data set

---

(1). Preprocessing input mushroom data from letters to some digits

(2). Divide the data (for mushroom data) into training examples (90%) and testing examples (10%) and create two input files respectively

(3). Input training examples and test examples

(4). Initialize learning speed, weights, and hidden values for the ANN

(5). Training:

```
Terminate when accuracy is not changed or max number of epochs
  //FWD computation
  for each input training data $x \in X$
     compute the hidden value for every hidden node of each laye
```

```
        //check if the desired output is equal to the actual output
        if (yes) accuracy++
        else
           //Call BP for the output layer
           for each output node
               calculate delta value and save them
           end for each output node
           //Call BP for each hidden layer
           for each hidden layer
             for each hidden unit h
               compute delta value and save them
             // end for each hidden unit h
             for each hidden unit h
               update corresponding weights
             //end for each hidden unit h
           //end for each hidden layer
        //end if-else
    //end of epoch and need to check termination condition
    Testing Procedure:
    FWD through the trained ANN with every testing data
       if (correct) test accuracy ++
       Output the accuracy
    //end FWD
```

6. Output:

   a) You can use the command “script outPutFileName” to dump all standard I/O as shown in the following figure. Or you may redirect all the I/O to a file. Either way works

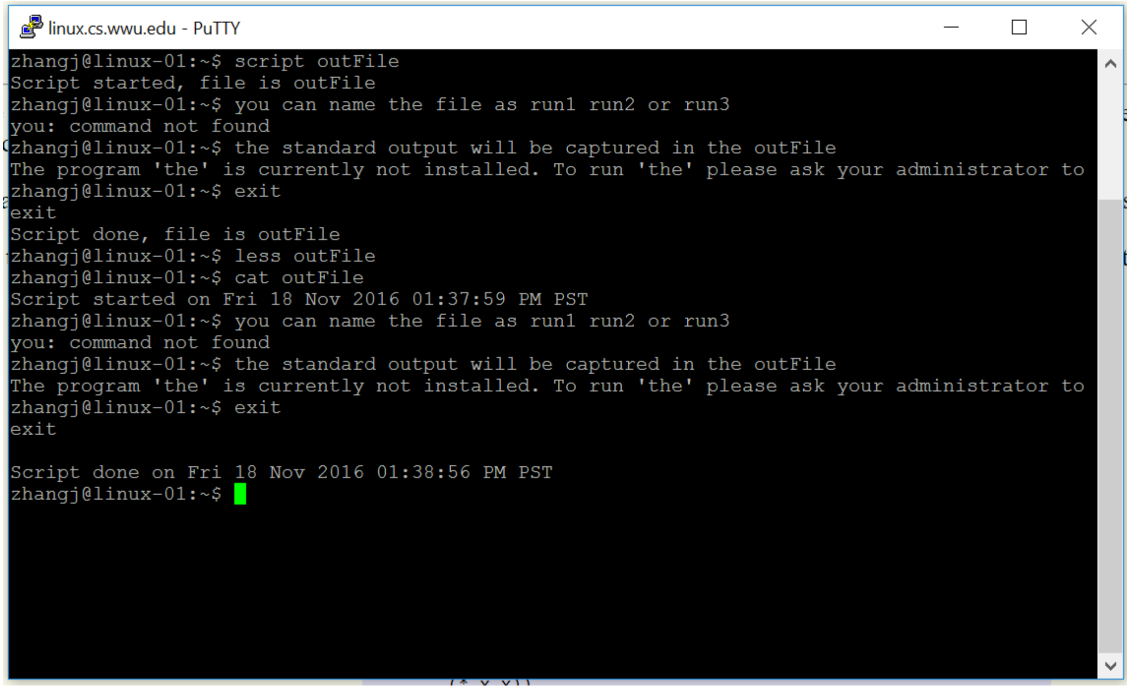
   b) The standard I/O output should contain:

      1) Topology: number of hidden layers and number of nodes for each layer;
      2) initial learning speed;
      3) initial weights;
      4) for each epoch: print out epoch number and its accuracy;
      5) dump some intermediate weights at appoint of significance;

6) When training is done: training accuracy, CPU running time, and total number of epochs;

7) final weights before testing;

8) accuracy for testing;



```
zhangj@linux-01:~$ script outFile
Script started, file is outFile
zhangj@linux-01:~$ you can name the file as run1 run2 or run3
you: command not found
zhangj@linux-01:~$ the standard output will be captured in the outFile
The program 'the' is currently not installed. To run 'the' please ask your administrator to
zhangj@linux-01:~$ exit
exit
Script done, file is outFile
zhangj@linux-01:~$ less outFile
zhangj@linux-01:~$ cat outFile
Script started on Fri 18 Nov 2016 01:37:59 PM PST
zhangj@linux-01:~$ you can name the file as run1 run2 or run3
you: command not found
zhangj@linux-01:~$ the standard output will be captured in the outFile
The program 'the' is currently not installed. To run 'the' please ask your administrator to
zhangj@linux-01:~$ exit
exit

Script done on Fri 18 Nov 2016 01:38:56 PM PST
zhangj@linux-01:~$ 
```

7. If you have any questions on this document, please e-mail me.

8. Hand in

   a) A text document file to explain your approach and discussion of the problems and results: I/O, topology, parameter changes, speed, accuracy, etc...

      1) Topology: number of hidden layers and number of nodes for each layer;
      2) initial learning speed;
      3) initial weights;
      4) for each epoch: print out epoch number and its accuracy;
      5) dump some intermediate weights at appoint of significance;
      6) When training is done: training accuracy, CPU running time, and total number of epochs;
      7) final weights before testing;
      8) test result

   b) Script file capture both randomly generated weights and the result weights for all layers for each epoch

   c) The program

   d) Output should be clearly shows the test result with standard I/O

   e) ReadMe file

   f) Using FirstNameLastNameBPANN.zip