Y33AUI: Applications of Artificial Intelligence

Homework
Wednesday 10.11.2010

$\boxed{1/2\text{ p.}}$ 1. Your name: _____

$\boxed{1/2\text{ p.}}$ 2. Assigned definition of target classes: _____
E.g.. 1 1 2 1 2 3 4 4 3 4

Question:	1	2	3	4	5	6	7	8	9	Sum
Points:	1/2	1/2	0	0	0	1	0	1	1/2	3 1/2
Deductions:										

Question:	10	11	12	13	14	15	16	17	18	19	20	21	Sum
Points:	2	1/2	1	0	2	0	2	0	3	2	1	3	16 1/2
Deductions:													

The goal of the homework is to compare 3 models for statistical pattern recognition (MLP, KNN, and SVM) on a classification task with 2 data sets. Your task is to create a MATLAB script which compares

- multilayer perceptron (MLP) with 1, 2, 5, 10, 20, 50, 100, 200, and 500 units in the hidden layer,
- *k* nearest neighbors method (kNN) with 1, 2, 5, 10, 20, 50, 100, 200, and 500 neighbors and
- support vector machine (SVM) with RBF kernel of size 1, 2, 5, 10, 20, 50, 100, 200, and 500

on 1 data set and prints out the training and testing accuracy of all these 27 models.

Further information:

- The homework contains 21 questions and you can earn 20 points.
- You can use 5 days to elaborate all tasks, but the precise deadline is in the Upload system.
- You will hand in 2 parts:
  - MATLAB script used to solve the tasks (with all needed functions) to Upload system, and
  - document with answers to all questions (either an electronic version to Upload system or a paper version directly to your teacher).
- To solve certain tasks, we will define functions and will call the from the script. Maintain the script and the functions as simple as possible. If you create your own functions, choose a suitable name. Use comments.
- When there is a framed space for answer, you should write down a text answer. You can write it directly to this document. If you choose to write to a new document (e.g. for electronic submission), always indicate the question number. The frame size suggests the length of the expected answer.
- When there is no space for answer, you are usually expected to create a piece of script or a function. Do not forget to submit them in the Upload system.
- Be concise and write legibly. Good luck!

Continue on next page...

1 Meet the data

Examine the files `optdigits.train`, `optdigits.test` and `optdigits.names`; explore especially the last one and answer the following questions:

0 p. 3. How many varibles are used to describe an object, i.e. how many inputs will our models have?

0 p. 4. What is the domain (possible values) of input variables?

0 p. 5. What is the number of classes to which we should classify the objects?

1 p. 6. Create function `designClasses()` which maps the original 10 classes (digits 0 to 9) to your new class definitions. You will pass a vector of length n into the function and it will return a vector of the same size containing the mapped classes. If your particular assignment is e.g. 1 2 3 4 4 4 5 5 6 7, your function will transform individual vector entries via the following map:

Original class	0	1	2	3	4	5	6	7	8	9
New class	1	2	3	4	4	4	5	5	6	7

Do not forget to submit the function to the Upload system!

2 NETLAB

Download and unzip the NETLAB toolbox. Review the function and meaning of `mlp`, `netopt` and `mlpfwd` functions. Learn also the `confmat` function (see `help confmat`; take care to use the `confmat` function from NETLAB toolbox, and pay attention to the expected form of its argument when dealing with multiclass classification).

0 p. 7. What is contained in `RATE(1)` after calling `[C, RATE] = CONFMAT(Y, T)`?

3 The first data set

Load the `optdigits.train` file with the training data. Create a `trin` matrix with the input variables of the trainig data. Create a `trout` matrix with the output variable of the training data. Tranform the `trout` vector using your `designClasses` function.

Similarly, load the testing data from the `optdigits.test` file. Create `tstin` and `tstout` matrices; transform `tstout` using `designClasses`.

Continue on next page...

4 MLP

We will use more than 2 output categories. In that case a network with more than 1 output is suitable. Every output of the network will represent a discriminant function for one class. Thus, we have to be able to transform the output variable `trout` into several binary output variables and vice versa, we need to be able to get the predicted class from the network outputs.

1 p. 8. Create functions `class2indicator` and `indicator2class` with the following headers:

```
function ind = class2indicator(cls, Ncls)
function cls = indicator2class(ind)
```

Specifications:

- `cls` is a $[N \times 1]$ matrix with class identifiers for each of N objects.
- `Ncls` is the number of classes.
- `ind` is a $[N \times Ncls]$ matrix containing only 1 one on each row, other entries are zeros. Each i-th row contains the one on position `cls(i)`. In case of `indicator2class` function, which will be used to transform the network outputs to class identifiers, the `ind` matrix can contain real numbers, but the highest of them should indicate the predicted class.

Do not forget to submit these functions to Upload system!

1/2 p. 9. Create a `trouti` matrix from the `trout` vector via the `class2indicator` function. What command do you use? What is the size of the resulting matrix `trouti`?

2 p. 10. Create a function with the following header:

```
function [trAcc, tstAcc] = trainAndTestMLP(trin, trout, tstin, tstout, nhidden)
```

You will pass the training and testing data and the number of units in the hidden layer to the function. It will train the network on the training data and will compute the network accuracy on training and testing data. Try it e.g. for 10 neurons in the hidden layer.

- Use the `softmax` function as the output nonlinearity.
- Use `scg` as the optimization algorithm.
- Set the number of iterations to 200.
- Use the `confmat` function to compute the accuracies.

1/2 p. 11. Describe shortly the form of the confusion matrix. If it has any significant structure, try to describe it.

Continue on next page...

1 p. 12. Based on the confusion matrix, compare the network accuracy for training and testing data. Is there any systematic difference? Explain.

0 p. 13. In your script, embed the call to the `trainAndTestMLP` function in a cycle and add other needed commands to get the training and testing accuracy for MLP with 1, 2, 5, 10, 20, 50, 100, 200, and 500 neurons in hidden layer. Fill in the following table:

	Number of neurons in the hidden layer								
	1	2	5	10	20	50	100	200	500
Class. accuracy, train. data [%]									
Class. accuracy, test. data [%]									

5 KNN

For kNN modeling, we will use the `knnrule()` and `knnclass()` functions from STPR toolbox. Download and install it. Review the data format it uses.

2 p. 14. Create a function with the following header:

```
function [trAcc, tstAcc] = trainAndTestKNN(trin, trout, tstin, tstout, neighbors)
```

You will pass the training and testing data and the number of neighbors to the function. It will create a model based on the training data and will compute the network accuracy on training and testing data. Try it e.g. for 10 nearest neighbors.

- To compute the accuracies, use the NETLAB `confmat` function. Take care to use the right data format for the arguments.

0 p. 15. In your script, embed the call to the `trainAndTestKNN` function in a cycle and add other needed commands to get the training and testing accuracy for KNN with 1, 2, 5, 10, 20, 50, 100, 200, and 500 nearest neighbors. Fill in the following table:

	Number of nearest neighbors								
	1	2	5	10	20	50	100	200	500
Class. accuracy, train. data [%]									
Class. accuracy, test. data [%]									

6 SVM

For SVM modeling, we will use the `bsvm2()` and `svmclass()` functions from STPR toolbox.

2 p. 16. Create a function with the following header:

Continue on next page...

```
function [trAcc, tstAcc] = trainAndTestSVM(trin, trout, tstin, tstout, rbfsize)
```

You will pass the training and testing data and the kernel size to the function. It will create a model based on the training data and will compute the network accuracy on training and testing data. Try it e.g. for the kernel size of 10.

- Use SVM with RBF kernel. (`options.ker`)
- Set the kernel size correctly. (`options.arg`)
- Set the constant C (the weight on the correct classification compared to the margin size) to the value of 100. (`options.C`)
- Set the maximal iteration count to 20000. (`options.tmax`)
- To compute the accuracies, use the NETLAB `confmat` function. Take care to use the right data format for the arguments.

0 p. 17. In your script, embed the call to the `trainAndTestSVM` function in a cycle and add other needed commands to get the training and testing accuracy for RBF kernel with the size of 1, 2, 5, 10, 20, 50, 100, 200, and 500. Fill in the following table:

	RBF kernel size								
	1	2	5	10	20	50	100	200	500
Class. accuracy, train. data [%]									
Class. accuracy, test. data [%]									

7 Discussion results

3 p. 18. Comment on the above results for MLP, kNN a SVM.

2 p. 19. What model would you select as the best one, and why???

Continue on next page...

8 The second data set

At present state, you should have a script that computes the training and testing accuracy of 3 different types of models with various settings (27 models in total). Now, we apply the script on different data. The training data are in the `optdigits2.train` file, the testing data are in the `optdigits2.test` file.

1 p. 20. Run the script and fill in the following tables:

MLP:

	Number of neurons in the hidden layer								
	1	2	5	10	20	50	100	200	500
Class. accuracy, train. data [%]									
Class. accuracy, test. data [%]									

KNN:

	Number of nearest neighbors								
	1	2	5	10	20	50	100	200	500
Class. accuracy, train. data [%]									
Class. accuracy, test. data [%]									

SVM:

	RBF kernel size								
	1	2	5	10	20	50	100	200	500
Class. accuracy, train. data [%]									
Class. accuracy, test. data [%]									

3 p. 21. What can you say about the results for the second data set compared to the results for the first data set? Try to explain the differences. Certain help can be provided by the confusion matrices.

End of homework.