CS385 : Assignment 6

Topics covered: Feed-forward Artificial Neural Network, Back Propagation Algorithm

Deliverables: Your submission for this assignment should be an archive of two files, named `annxor.py` and `yourusername_report.pdf`.
`annxor.py` should contain functions list below.

Function name	Input	type	Output	type
dataLoad	Filename	String	X: dataset, include dummy inputs	Array (4,4)
paraIni			Initialized parameters W[0] is wh: parameters for hidden layer W[1] is wo: parameters for output layer	W:List
feedforward	X: 3 features are used for feed forwarding	(4,4)	Intermediate Result intermRslt[0]: oh hidden layer output intermRslt[1]: ino output layer input intermRslt[2]: oo output layer output (i.e. Yhat)	List
	W: parameters	list		
errCompute	Y	Class label in X (4,1)	Value of cost function	real
	Yhat	(1,4)		
backpropagate	X	(4, 4)	W: updated parameters	List
	W: parameters	list		
	intermRslt: Output/input of each layer in feedforward stage	list		
	alpha: learning rate	real		

`Yourusername_report.pdf` should follow the outline described in Section 3 and you should put in all the required materials in Section 4. You should put two files in a directory called `cs385_yourusername_6` and zipping the directory into a zip file called `cs385_yourusername_6.zip` etc. Failure to follow conventions will result in penalties in marks.

Objectives:

- To get familiarized the principle of FF ANN(feed-forward artificial neural network) and gradient descent method for learning in a multi-layered network. You are also expected to implement it by using *vectorization* method.
- To implement the BP(Back Propagation) algorithm to learn parameters of FF ANN for solving the XOR problem.
- To get familiarized on tuning the performance of BP algorithm.

1. **Dataset**

   The training dataset is XOR(eXclusive-OR) truth table. The provided XOR.txt contains 4 samples and each sample is with 2 features and 1 output.

2. **Documentation**

   For documentation, you can use LATEX or Word format. You need to write a report on this assignment by following instructions in Section 3. You should submit ONLY the "pdf" of the document and NO other file formats will be accepted.

3. **Your tasks**

   a. Write a function *loadData(filename)* to load XOR data into an array *X*.
   After inserting dummy inputs as the first feature of *X*, we have: *X.shape* (4, 4).

   b. The structure of ANN for XOR problem: input layer with 2 units; 1 hidden layer with 2 units, each use tanh() as activation function; output layer with 1 unit with sigmoid() as activation function. Draw the diagram of the network in *Section 1 ANN for XOR*.
   Construct the feedforward equation to predict the output of an input (*x*1, *x*2). Explain this equation briefly and the number of parameters that need to be estimated. Write down the equation, explanation and the number of needed parameters in *Section 1 ANN for XOR* of your report.

   c. Write a function *paraIni()* to initialize parameters *W* for the given network in *task b*. The values are randomly chosen between [-1,1].
   For test use, a fix valued set of parameters are given by the provided function *paraIni()* (a6.py)

   d. Write a function feedforward() to generate output of each layer for the given network in *task b*. The function takes in *X* the training data set and parameters *W*.
   W is a list. *W*[0] is wh the hidden layer parameters, and *W*[1] is wo the output layer parameters.
   The function returns intermRslt=[oh,ino,oo], oh is the output of hidden layer, ino is the input of output layer, and oo is the output of output layer.
   For the given initial parameters, the output is: (0.5036, 0.5867, 0.4353, 0.5369)

   e. The error cost function we used for the XOR NN is: $err = \frac{1}{2m}\sum_{i=1}^{m}(y_i - \hat{y}_i)$, write a function errCompute() to return a cost value. It takes in the output from training dataset *Y* and model output $\hat{Y}$ from intermRslt(oo is $\hat{Y}$).
   For the given initial parameters, the cost value is: 0.1290

   f. Construct the back propagation equations for updating all parameters. Again, briefly explain these equation. Write down them in *Section 2 Back Propagation* of your report.

   g. Basing on the equations in *task f*, write *backpropagate()* to implement the BP algorithm for learning the parameters. This function takes in *X*, parameters *W*, *intermRslt*(return by feedforward), and the learning rate *alpha*. It returns the learned *W*.
   For the given initial parameters, when *alpha* is 0.5, the learned parameters is:
   W[0]=wh=[[0.1785, -0.7730, 0.6199],[-0.7984, 0.5606, 0.2083]]
   W[1]=wo=[[0.1303, 0.5908, 0.3435]]

   h. Run the provided *R=FFMain("XOR.txt, 10000, 0.5)* function by using the given initial parameters (call the provided *paraIni()*), you'll get *R*[1]=[[0.0169, 0.9782, 0.9782, 0.0150]].

Use your own *paraIni*(), and run the *FFMain()* when numIteration=100, 1000, 5000, 10000, and alpha=0.01, 0.5 respectively. Draw the curve of error function (i.e. R[1] returned by *FFMain()*). Observe the curve and the classification result, give your brief analysis on them in *Section 3 Experimental Result*.

5. Rubrics

This assignment is graded over total of 50 points. The breakdown is as follows:

- task a, data loading (3 points)
- task b, ANN for XOR (5 points)
- task c, parameters initialization (5 points)
- task d, feed-forward implementation (5 points)
- task e, error function implementation (5 points)
- task f, back propagation description (10 points)
- task g, back propagation implementation (10 points)
- task h, error curve plotting and experimental result analysis (3+4 points)

Note: Submission requirements must be met i.e., reasonable comments, proper format of the report (Not copy and pasted from the assignment specifications!), detailed explanation (using necessary equations, tables, figures, charts, etc.), correct documentation and file formats for the submission, etc. If any of these requirements are not satisfactory, then zero marks for this assignment.