Kevin Toney

Capstone Lab 10
Neural Nets Classification using Weka

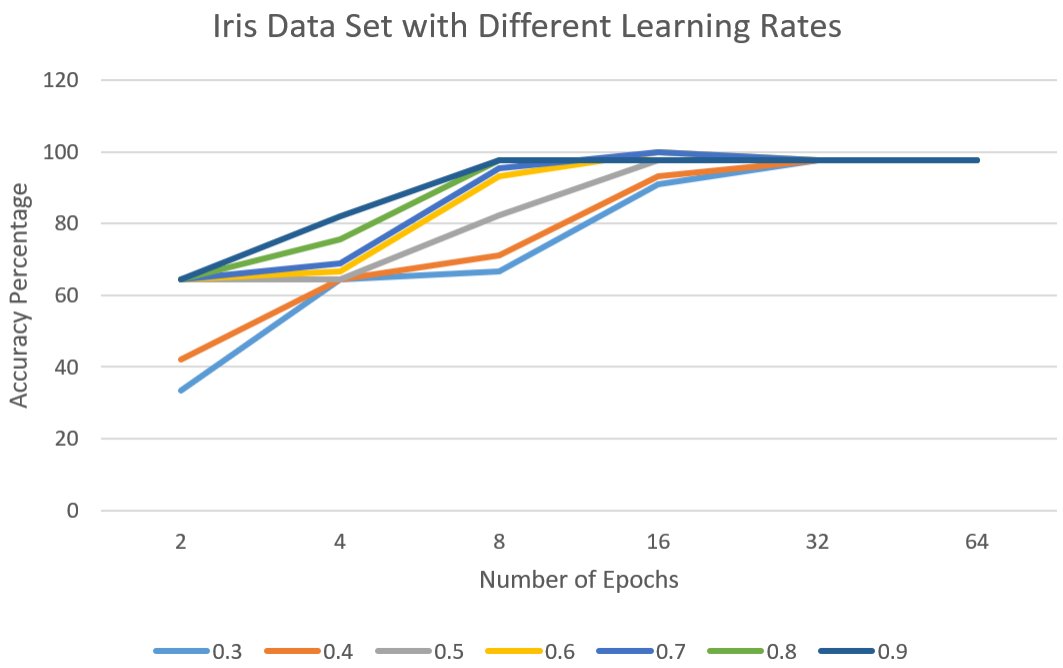
I already downloaded and installed Weka.

Download the Iris and Vowel datasets.
(http://dml.cs.byu.edu/~cgc/docs/mldm_tools/Assignments/Datasets/iris.arff,
http://dml.cs.byu.edu/~cgc/docs/mldm_tools/Assignments/Datasets/vowel.arff)

1. The backpropagation algorithm in Weka is under the "Classify" tab, Click on "Choose", expand "functions", and select "MultilayerPerceptron".

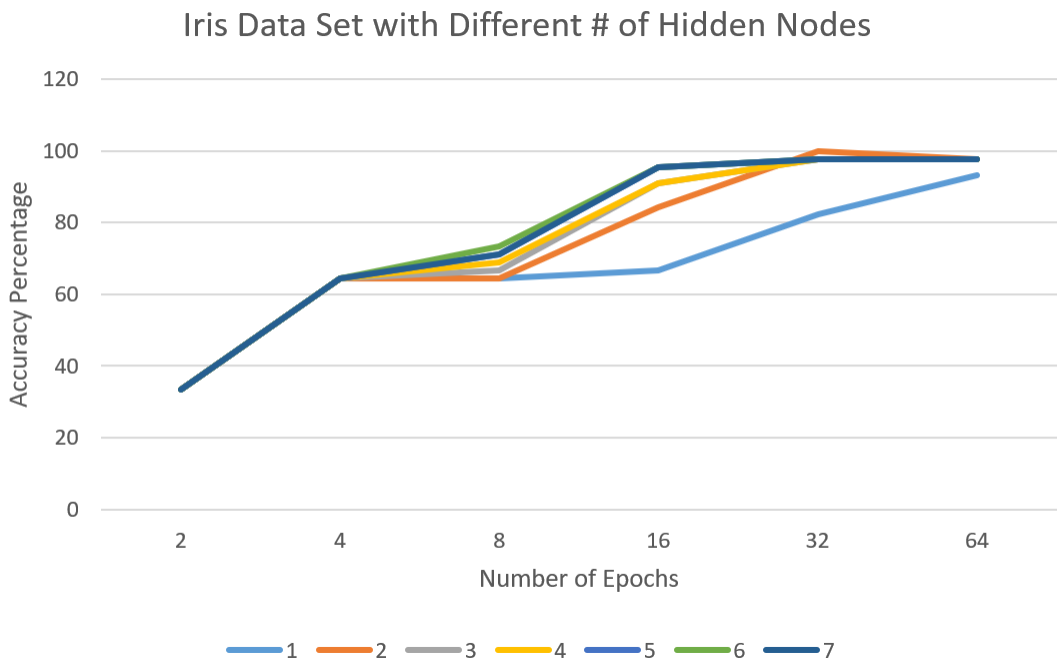
2. Use the backpropagation algorithm on the Iris problem, with a random 70/30 split.
   - With a single hidden layer and a fixed number of hidden nodes (of your choice), experiment with different learning rates. Graph training and test set accuracy over time (i.e., number of iterations/epochs) for several different learning rates. Based on these results, select a reasonable learning rate.
     - I used a fixed number of 3 hidden nodes, with one single hidden layer. I used a learning rate of 0.3.

	# of Epochs					
Learn Rate	2	4	8	16	32	64
0.3	33.3	64.4	66.67	91.11	97.78	97.78
0.4	42.2	64.4	71.11	93.33	97.78	97.78
0.5	64.4	64.4	82.22	97.78	97.78	97.78
0.6	64.4	66.7	93.33	100	97.78	97.78
0.7	64.4	68.9	95.56	100	97.78	97.78
0.8	64.4	75.6	97.78	97.78	97.78	97.78
0.9	64.4	82.2	97.78	97.78	97.78	97.78



- With the learning rate selected and a single hidden layer, experiment with different numbers of hidden nodes, starting from 1 and adding each time until you see no improvement on the training set's accuracy. For each choice of number of hidden nodes, graph training and test set accuracy over time.

	# of Epochs					
Hid Nodes	2	4	8	16	32	64
1	33.3	64.4	64.44	66.67	82.22	93.33
2	33.3	64.4	64.44	84.44	100	97.78
3	33.3	64.4	66.67	91.11	97.78	97.78
4	33.3	64.4	68.89	91.11	97.78	97.78
5	33.3	64.4	71.11	95.56	97.78	97.78
6	33.3	64.4	73.33	95.56	97.78	97.78
7	33.3	64.4	71.11	95.56	97.78	97.78



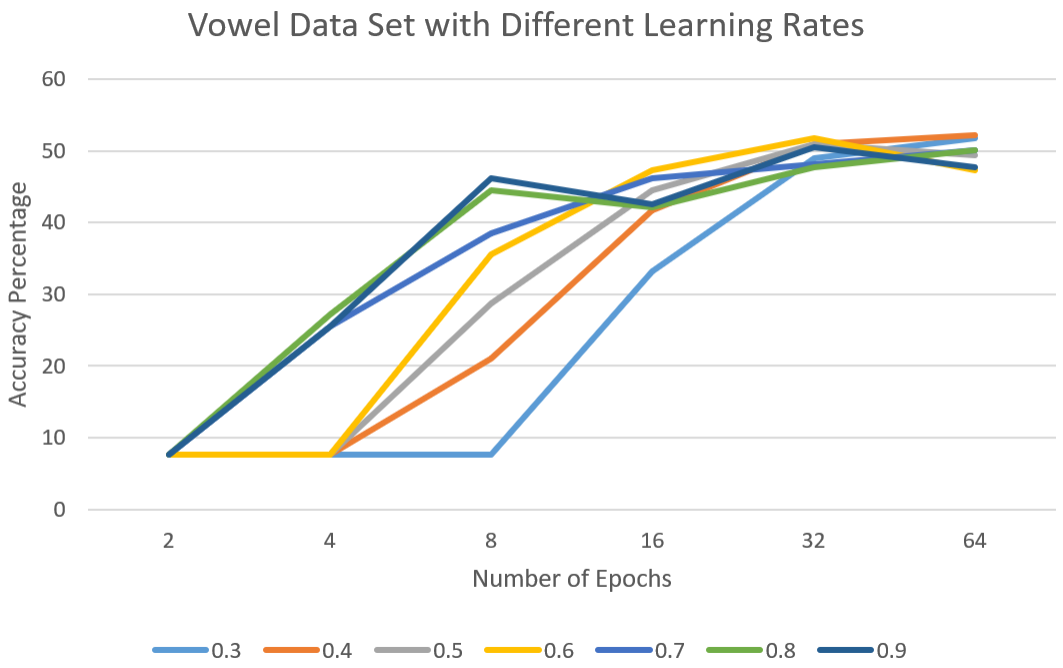
- Record your best number of hidden nodes (i.e., the one resulting in highest accuracy on the test set).
    - The best number of hidden nodes is 6.

3. Use the backpropagation algorithm on the Vowel problem, with a random 75/25 split. (Note: Make sure you ignore the "Train or Test" attribute).

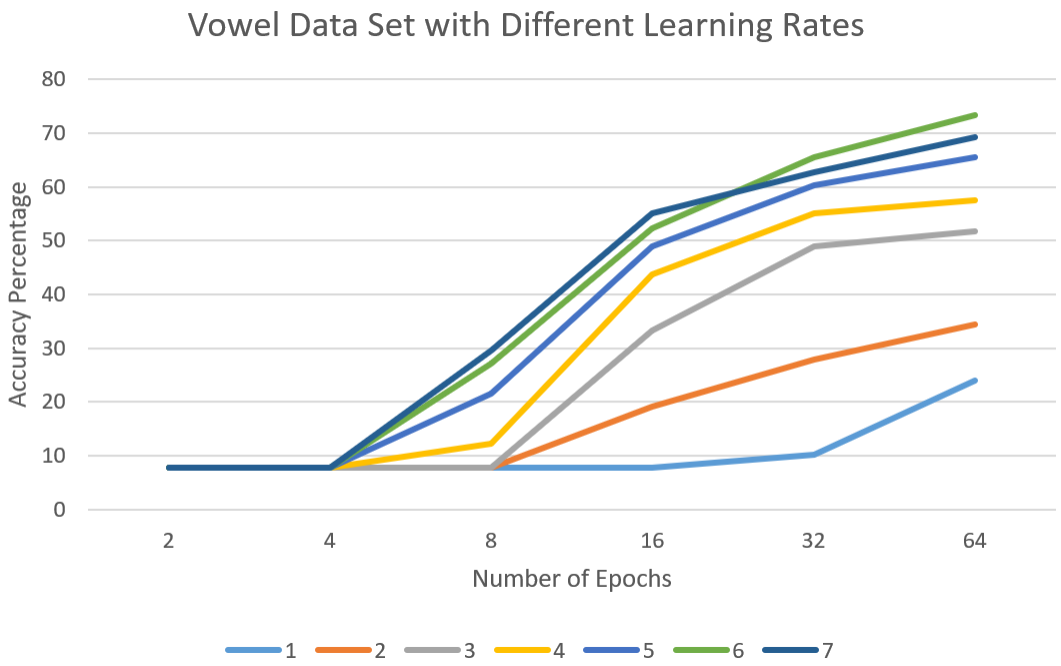
- Repeat the above experiments. I used three hidden nodes in a single layer.

	# of Epochs					
Learn Rate	2	4	8	16	32	64
0.3	7.69	7.69	7.69	33.20	48.99	51.82
0.4	7.69	7.69	21.05	41.70	51.01	52.23
0.5	7.69	7.69	28.75	44.53	51.01	49.40
0.6	7.69	7.69	35.63	47.37	51.82	47.37
0.7	7.69	25.50	38.46	46.15	48.18	50.20
0.8	7.69	27.13	44.53	42.10	47.77	50.20
0.9	7.69	25.51	46.15	42.51	50.61	47.77

Kevin Toney



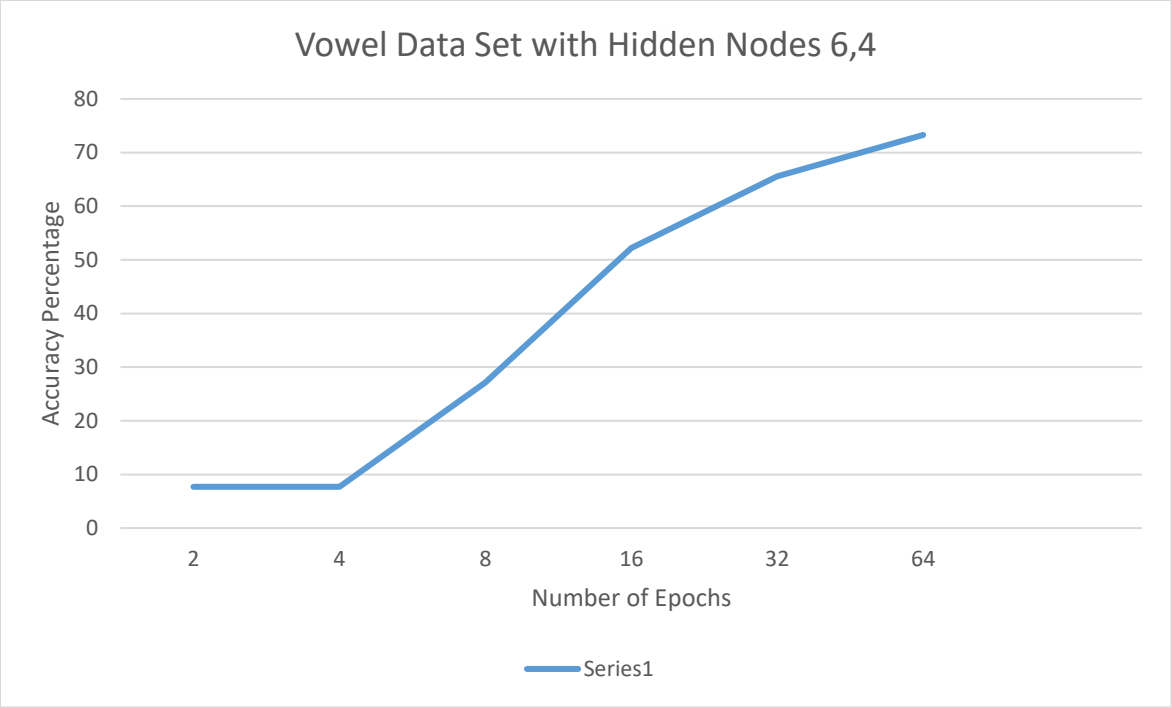
	# of Epochs					
Hid Nodes	2	4	8	16	32	64
1	7.69	7.69	7.69	7.69	10.21	23.89
2	7.69	7.69	7.69	19.03	27.94	34.41
3	7.69	7.69	7.69	33.20	48.99	51.82
4	7.69	7.69	12.14	43.72	55.06	57.49
5	7.69	7.69	21.46	48.99	60.32	65.59
6	7.69	7.69	27.13	52.23	65.59	73.28
7	7.69	7.69	29.55	55.06	62.753	69.23



The best number of hidden nodes seems to be six.

- With the learning rate selected, induce a 2-hidden layer neural network with 6 hidden nodes in the first layer and 4 hidden nodes in the second. Graph training and test set accuracy over time.
    - I used a learning rate of 0.3.

	# of Epochs						
Hid Nodes	16	32	64	128	256	512	1024
6,4	7.69	7.69	7.69	55.47	78.14	76.92	75.71

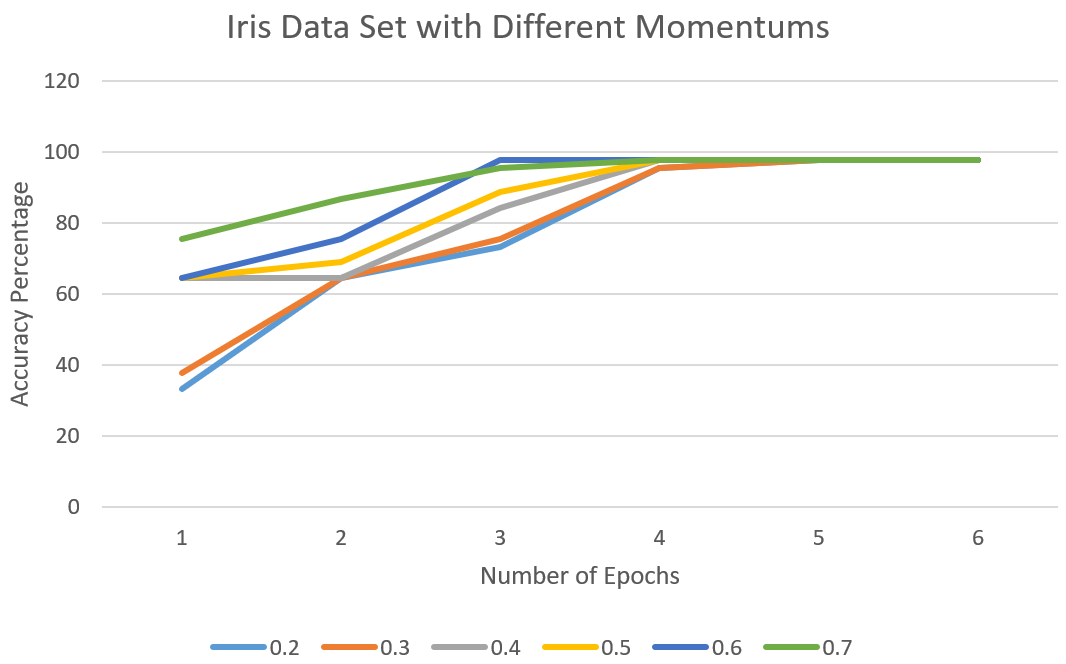


4. Using only the best number of hidden nodes as recorded above for 1 hidden layer and the same training/test splits, re-run your backpropagation algorithm with the momentum term option to induce a neural network for both Iris and Vowel. Graph training and test set accuracy over time.

I used 6 hidden nodes, a learning rate of 0.3, and a single hidden layer.

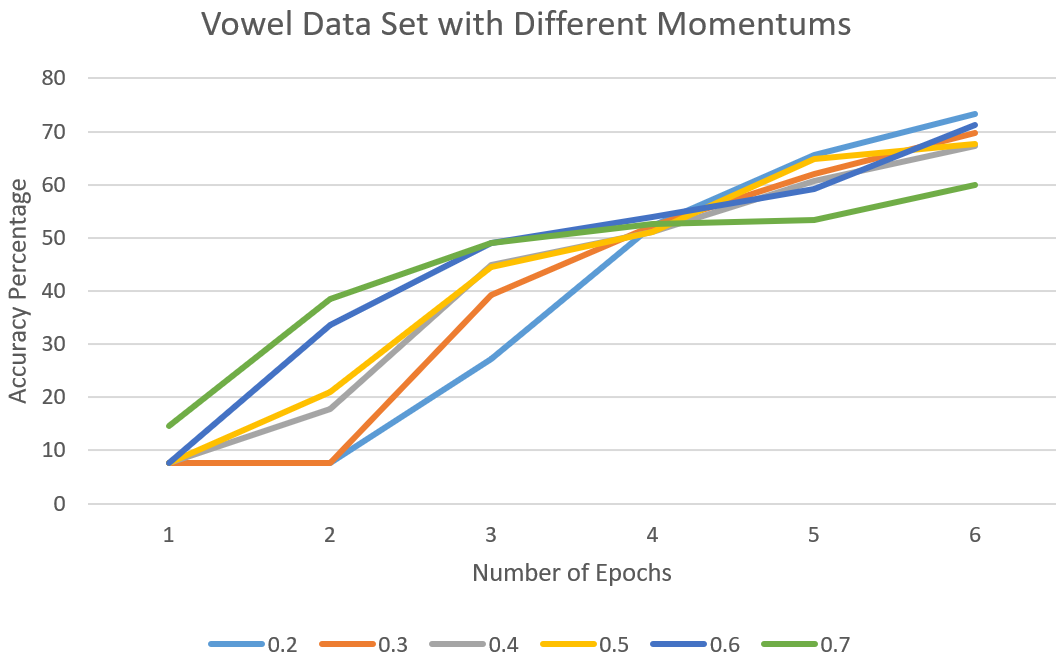
Iris Data Set

	# of Epochs					
Momentum	2	4	8	16	32	64
0.2	33.3	64.4	73.3	95.56	97.78	97.78
0.3	37.78	64.4	75.56	95.56	97.78	97.78
0.4	64.4	64.4	84.4	97.78	97.78	97.78
0.5	64.4	68.90	88.90	97.78	97.78	97.78
0.6	64.4	75.56	97.78	97.78	97.78	97.78
0.7	75.56	86.67	95.56	97.78	97.78	97.78



Vowel Data Set

	# of Epochs					
Momentum	2	4	8	16	32	64
0.2	7.69	7.69	27.13	52.23	65.59	73.28
0.3	7.69	7.69	39.27	52.23	61.94	69.64
0.4	7.69	17.81	44.94	51.01	60.73	67.21
0.5	7.69	21.05	44.56	51.01	64.78	67.61
0.6	7.69	33.60	48.99	53.85	59.11	71.26
0.7	14.57	38.46	48.99	52.63	53.44	59.92



5. Analyze the data you have collected and briefly answer the following questions:
   - Discuss the effect of different learning rates on the algorithm's performance.
     - The different learning rates improve the accuracy of the algorithm up to a peak. After the peak, the accuracy plateaus and decreases over time.
   - Discuss the effect of different numbers of hidden units on the algorithm's performance (1-hidden layer case).
     - As the number of hidden units increases, the accuracy of the algorithm increases up to a peak. Then, the accuracy plateaus.
   - Compare your recorded best numbers of hidden nodes for each problem with the following heuristic value: $H=N/(10(I+O))$, where N is the size of the (training) data set, I is the number of network inputs and O is the number of outputs.
     - Iris:
       - H = 150/(10*(4+3)) = 2.14
     - Vowel:
       - H = 990/(10*(27 + 11)) = 2.605
   - How did the momentum term affect the learner's behavior (number of epochs to convergence, final accuracy, etc.)?
     - The momentum term affected the learner's behavior by making the accuracies converge earlier. Also, a higher momentum term increased the accuracy quicker.