Purnima Vijaya and Laxmi Bhavani

B551 Assignment 6: MACHINE LEARNING REPORT

K- Nearest Neighbour Classifier:

The KNN classifier in our algorithm uses Euclidean distance formula because it generates the best accuracy compared to other distance formula like manhattan distance or Chebyshev distance. Hence we decide on using the Euclidean distance.

*Results obtained by **training on a dataset of 40,000 images** are tabulated below:*

Serial No.	K value	Classification Accuracy	Program Runtime(minutes)
1	5	69.88%	102.67
2	8	69.77%	109.66
3	10	69.67%	102.11
4	15	69.98%	100.48

Green color shows the best performance and red color worst performance

*When the same classifier was trained on a **training set narrowed down to 15323 images**(half of the actual training set), the following observations were made:*

Serial No.	K value	Classification Accuracy	Program Runtime(minutes)
1	5	68.29 %	43.72
2	10	68.50%	48.12

SOME IMAGES CLASSIFIED CORRECTLY BY KNN CLASSIFIER

S.NO	IMAGE	ORIGINAL LABEL	ASSIGNED LABEL
1	 test/10008707066.jpg	0	0
2	 test/10099910984.jpg	0	0
3	 test/10107730656.jpg	180	180

4	 test/10161556064.jpg	270	270
5	 test/10164298814.jpg	0	0
6	 test/10304005245.jpg	90	90

SOME IMAGES CLASSIFIED INCORRECTLY BY KNN CLASSIFIER

S.NO	IMAGE	ORIGINAL LABEL	ASSIGNED LABEL
1	 test/10196604813.jpg	90	0
2	 test/10351347465.jpg	270	180
3	 test/10352491496.jpg	90	180
4		180	0

	test/10484444553.jpg		
5	 test/10577249185.jpg	180	0
6	 test/112713406.jpg	0	180

Neural Network classification:

The neural network in this problem is composed of one input layer, one hidden layer and one output layer.

- Learning parameter α (alpha) ranges between (0, 1). In order to achieve desired performance, the two parameters: α and number of nodes in the hidden layer are tuned.

*Results obtained by **training on a dataset of 40,000 images** are tabulated below:*

Hidden Layer Node Count	Learning Rate (α)	Classification Accuracy	Program Runtime(sec)
1	0.1	23.75%	15.02142406
2	0.1	48.46%	16.36109114
3	0.1	69.99%	15.40953183
4	0.1	70.11%	16.05718613
5	0.1	70.29%	17.0306859
6	0.1	71.26%	16.36027503
7	0.1	70.63%	17.46256804
8	0.1	70.84%	23.33352399
9	0.1	70.37%	17.85689092
10	0.1	71.26%	18.52084303
50	0.1	73.38 %	29.1743478775
100	0.1	71.69%	69.4642817974
10	0.5	70.94%	17.40675712
9	0.5	70.94%	17.84339595
8	0.5	69.37%	17.97047615
7	0.5	70.69%	18.38143587
6	0.5	70.52%	17.78297496
5	0.5	70.73%	16.81457496
4	0.5	70.10%	15.56829
3	0.5	69.14%	15.968431

2	0.5	40.83%	15.98021293
1	0.5	25.03%	14.32226491
10	0.9	71.15%	18.03088403
5	0.9	71.05%	17.04592395
2	0.9	37.43%	15.68012786

Green color shows the best performance and red color worst performance

- It is observed that the accuracy and runtime do not vary significantly when the number of hidden layer nodes range between 4 and 10.
- When hidden Layer nodes are increased to 50, there accuracy increased by 1%. However, the run-time of the program stepped up to ~ 30 seconds.

When the same network was trained on a training set narrowed down to 15323 images(half of the actual training set), the following observations were made:

Hidden Layer Node Count	Learning Rate (α)	Classification Accuracy	Program Runtime(sec)
10	0.1	71.79 %	7.61543607712
5	0.1	71.05 %	7.60640215874
4	0.1	52.99%	6.77162194252

Green color shows the best performance and red color worst performance

Observations:

- Run time of the program came down to almost half its previous runtime.
- When the training set was large the network performed well for smaller Hidden Layer Node Count.
- When the training set was reduced to half its size, the network's performance decreased significantly for smaller Hidden Layer Node Count.

Recommended Parameters:

1. It can be clearly observed that we cannot recommend KNN classifier to the client as it takes about an hour for running the code and determining the output.
2. The stochastic gradient descent is the most convenient approach as used in this problem. This approach is suitable for large training sets because, it is easier to update weights by considering one example in the training set at a time rather than the entire training set at once.

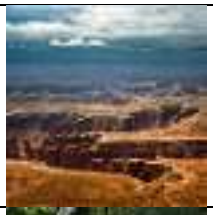
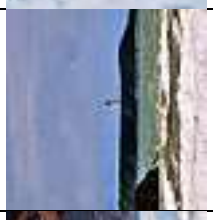
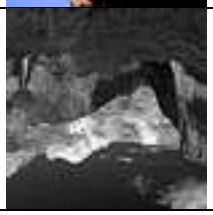
Examining the above results, we narrow down our recommended parameters to:

Hidden Layer Node Count	Learning Rate (α)	Classification Accuracy	Program Runtime(sec)
10	0.1	71.26%	18.52084303
50	0.1	73.38 %	29.1743478775

The first set of parameters seems to be the best choice, since it has good accuracy with reasonable runtime.

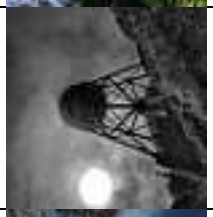
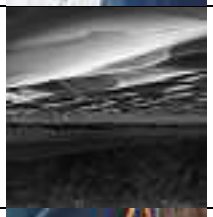
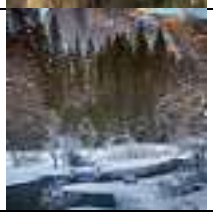
Purnima Vijaya and Laxmi Bhavani

SOME IMAGES CLASSIFIED CORRECTLY BY NEURAL NETWORK CLASSIFIER

S.NO	IMAGE	ORIGINAL LABEL	ASSIGNED LABEL
1		0	0
2		180	180
3		270	270
4		90	90
5		180	180
6		270	270
7		180	180

Purnima Vijaya and Laxmi Bhavani

SOME IMAGES CLASSIFIED INCORRECTLY BY NEURAL NETWORK CLASSIFIER

S.NO	IMAGE	ORIGINAL LABEL	ASSIGNED LABEL
1		90	0
2		270	180
3		90	180
4		0	180
5		90	270
6		270	90
7		0	180

- In most cases, the classifier is able to classify orientation of images containing blue sky on top (represented by blue pixels) and ground on bottom (represented by pixels of dark colours) correctly. However, there are certain images where the dark coloured pixels are on top and blue pixels in the bottom (Image 7). Such images are classified incorrectly.
- Images 4 and 6 are majorly composed of like coloured pixels. These are classified incorrectly.

- Image 1 has sky spread on top as well as on sides. The classifier assumed the oriented to be 0 while it was supposed to be 180.