Kyle Verdeyen
Computer Vision EN.601.461, Homework 3
README

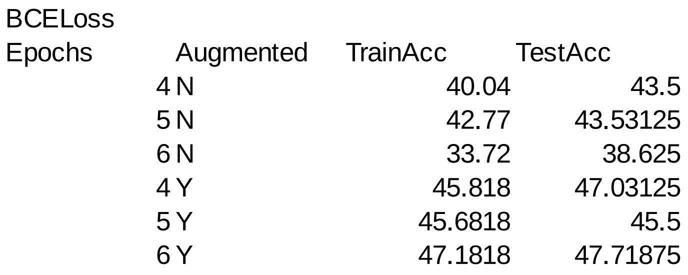
Augmented Models available here: https://drive.google.com/open?id=1QdGvYi-LVymXGHrZl0ut8fMVO-V3KDAK
(Still uploading datasets as of 11:59 PM)

P1a:

- # of epochs chosen: 6.
  - I originally chose to test both methods with 4,5, and 6 epochs with/without augmentation. Although 6 yields the best results out of this testing methodology, I was surprised to see the result over 20 epochs (same results as 6ep, but much longer computation time). However, training for 20 epochs takes an extremely long time and I was unable to investigate further.

P1a i)

- Accuracy comparison of BCELoss without data augmentation/with data augmentation
  - Data augmentation appears to help BCELoss. See chart below:
  -



BCELoss

Epochs	Augmented	TrainAcc	TestAcc
4	N	40.04	43.5
5	N	42.77	43.53125
6	N	33.72	38.625
4	Y	45.818	47.03125
5	Y	45.6818	45.5
6	Y	47.1818	47.71875

○

P1a ii)

- How well did BCELoss work on the training data? How well did BCELoss work on the test data? Why did/didn’t BCEloss work?
  - My implementation performs worse than chance (<50%) in both with/without augmentation. I think this is due to a combination of BCELoss not being the best for facial features (better for tracking inanimate objects) as well as my code being brutally inefficient.

P1b:

- # of epochs chosen: 6. Similar to above, I tested only 4, 5, 6 epochs in the interest of time. However, the error remains cyclical and does not decrease with number of epochs.
- Margin chosen: 1.5. The general consensus is to use a threshold related to the dataset you're working with, rather than a constant. A high margin slows down computation and increases losses, while a low margin prevents learning about non-related pairings. I chose 1.5 as it improved losses over 1 while not being as slow to train as 2.

P1b I)

- Accuracy comparison of ContrastiveLoss with data augmentation/without data augmentation
  - Judging solely from the loss charts, I think the data augmentation method improves the accuracy of the model as the loss is generally lower.

P1b ii)

- How well did ContrastiveLoss work on the training data? How well did ContrastiveLoss work on the test data?
  - I have a bug somewhere in my contrastive loss network. The loss charts appear to be cyclical (up and down at regular intervals, never decreasing trend). In addition, testing shows that the model stays at 50% accuracy, never changing despite edits to the margin and number of epochs used during training with or without augmentation. I believe the bug lies somewhere in the final discrimination phase to determine whether or not the distance is a match, however I was not able to locate the problem area.

P1c: Write a short paragraph comparing the two methods, including why you think one performed better than the other.

On paper, ContrastiveLoss seems to be a better match for facial recognition. It is supposed to group similar features together and split the dissimilar ones. However, my implementation is buggy and gives 50% accuracy no matter what data is given. Therefore, I believe in this case my BCELoss function performs better even though the accuracy is sub 50% as it appears to actually be attempting to classify the data fed into the network.