CPTS 577 Structured Prediction and Intelligent Decision-Making

Homework 2

Sheryl Mathew (11627236)

28 March 2019

1. **Question 1**

Let T be the number of decision steps. Let T = 5.

Decision Step 1: Number of mistakes made = 1

Decision Step 2: Number of mistakes made = 2

Decision Step 3: Number of mistakes made = 4

Decision Step 4: Number of mistakes made = 16

Decision Step 5: Number of mistakes made = 25

As seen above after 5 decision steps, the number of mistakes made is 25 which is $O(T^2)$ instead of being only 5, i.e one mistake per step. This is because the recurrent classifier learns from its previously best classifier. If the previously best classifier only initially committed one mistake, it seems harmless in the beginning but at each iteration this mistake keeps doubling as it has used the incorrect predictions from the previous classifier to learn the new data.

2. **Question 2**

Approach 1: In Dagger Algorithm, we have a interpolation factor beta which decides whether the policy used for an iteration should be the oracle policy or the learned classifier policy. If the random value selected is less than beta then we go for oracle policy or else for learned classifier policy. Similarly we can have different ranges of beta for each of the imitation learning algorithms, e.g Exact imitation 0 to 0.2, Searn 0.2 to 0.4 and so on. Therefore for every iteration based on the random value generated, we can check whether the random value lies and can then use the selected policy for performing the labelling in that iteration. In this way all the algorithms can be unified in a framework.

Approach 2: We can combine Reinforcement and Imitation learning algorithms. Let the reinforcement learning make the assumption that the environment is static and that the task that needs to learned is a mapping from a set of states of the environment to a set of possible actions (here it is a policy). In imitation learning the tasks can either be that the imitator knows the task that we need to learn or does not know the task to be learnt. For the first task this means we already will have the reward function and can use this information to perform the task. For the second task this means we do not have the reward function and hence we cannot continue any further. If we do not know the reward function we can select from all the possible reward fuctuins and using it as the intital reward, we can further find the optimal Q for this reward function and finds what is the likelihood that the different imitiation learning policy is the optimal policy or not. Then we can use the selected reward with that imitation learning with the highest likelihood. In this manner we will be able to map which policy has to be used at which state of the task. Therefore all the imitation learning algorithms can be used in a unified manner.

3. **Question 3**

Approach 1: The generic way to do handle missing labels would be to randomly give those inputs with no label a random label from the various possible labels. Also while assigning a score to those particular input set a lower score so that it does not negatively impact the performance.

Approach 2: If we can afford the loss of data we can drop the inputs with missing labels and proceed as further.

Approach 3: Dagger and Aggravate Algorithms instead of working on single labels can work with multiple possible training labels without choosing the correct on i.e instead of selecting a labels which is correct, all the likely labels can be used by the algorithm. The algorithm will take into account both the multiple labels and an estimate to the correct label. The output model after performing training will predict the test data by giving a single correct label. This happens because during training we are given a set of labels along with the probability of the given labels being correct. Initially the model will train all the labels weighted by priors and then updates the likelihood of each label occurring. In this way we will get the correct label. Further training on this model will improve the accuracy and provide a different model which can be used for testing.

4. Question 4

a. Text to Speech Exact Imitation Learning

Text To Speech Exact Imitation Learning Results:

Training:

Accuracy: 0.7524204702627939

Loss: 0.2475795297372061

Testing:

Recurrent Accuracy: 0.5549413349722163

Recurrent Loss: 0.44505866502778335

Oracle Accuracy: 0.7104786914882508

Oracle Loss: 0.28952130851174973

b. OCR Speech Exact Imitation Learning

OCR Exact Imitation Learning Results:

Training:

Accuracy: 0.9878709118475201

Loss: 0.012129088152479928

Testing:

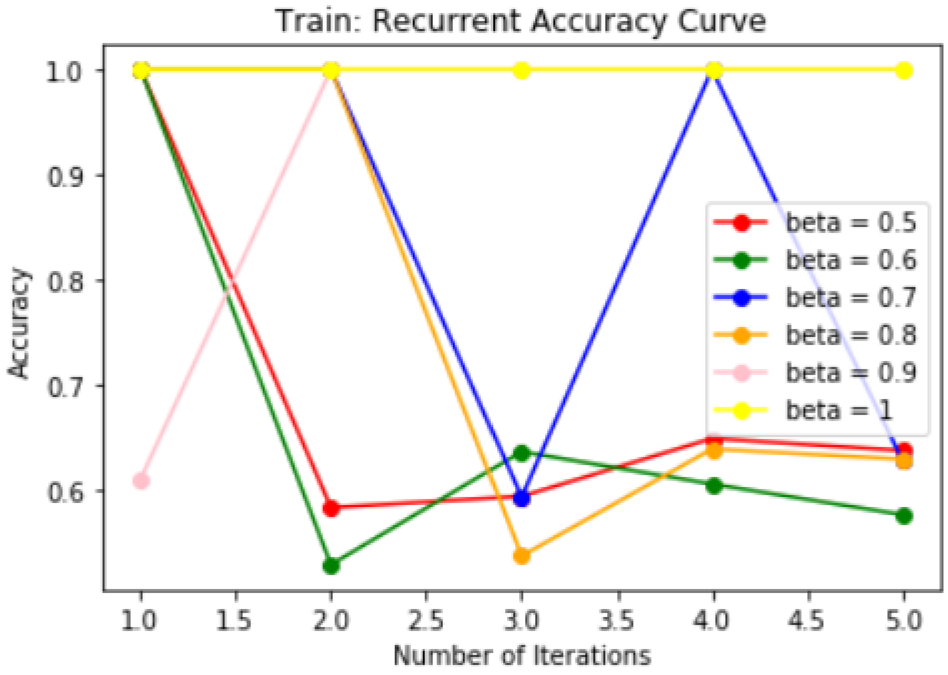
Recurrent Accuracy: 0.09521554451165674

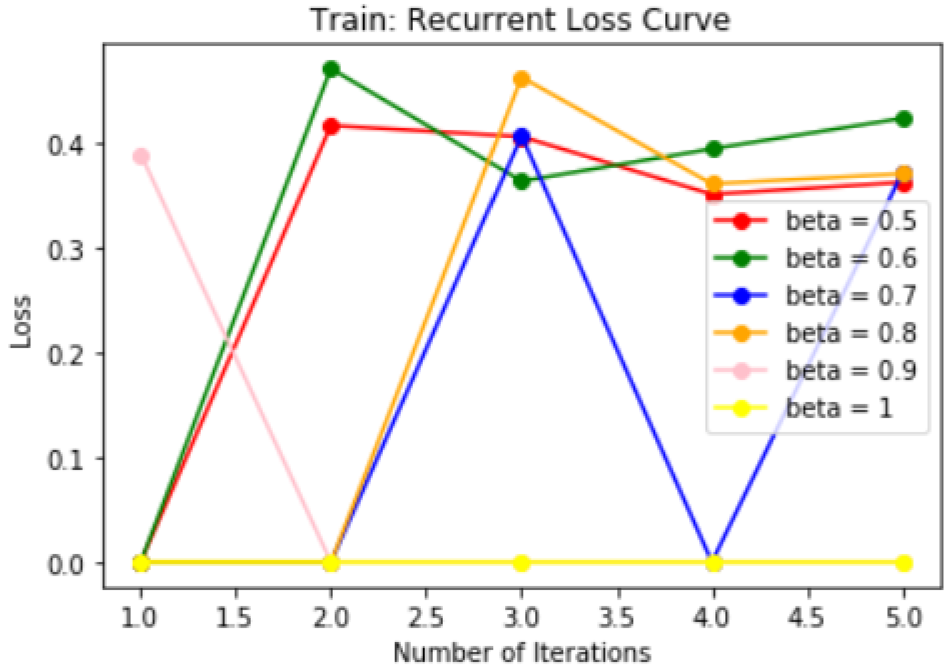
Recurrent Loss: 0.9047844554883442

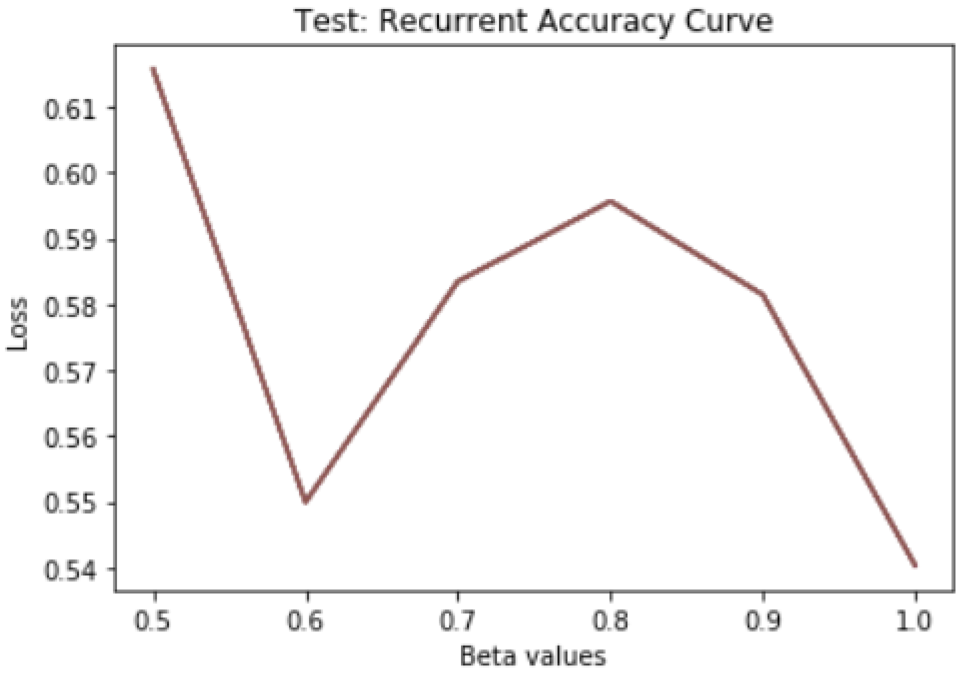
Oracle Accuracy: 0.37577383067704523

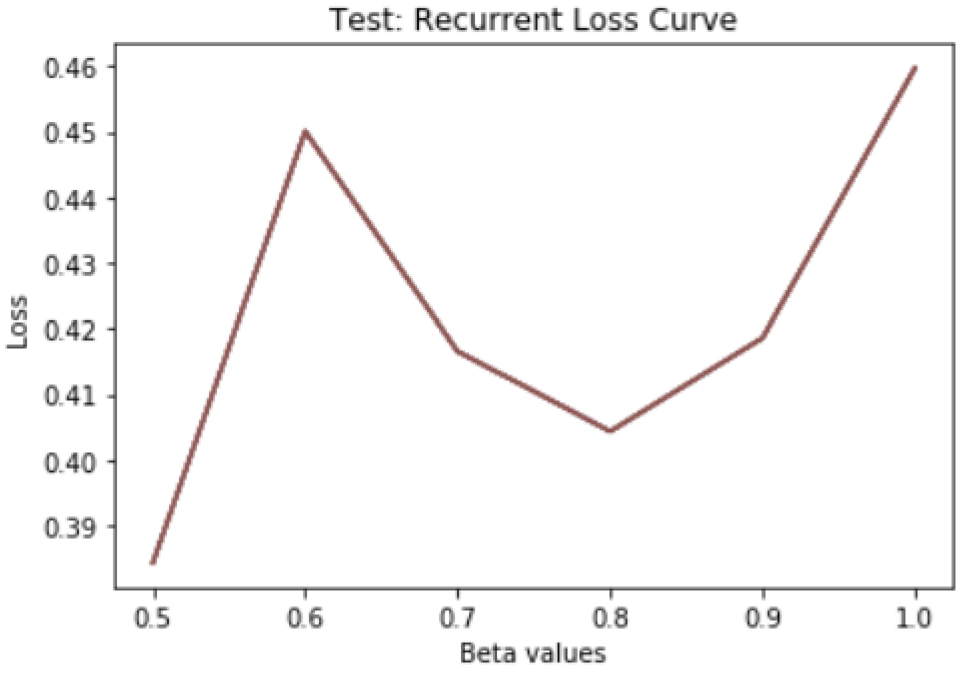
Oracle Loss: 0.6242261693229512

c. Text to Speech Dagger

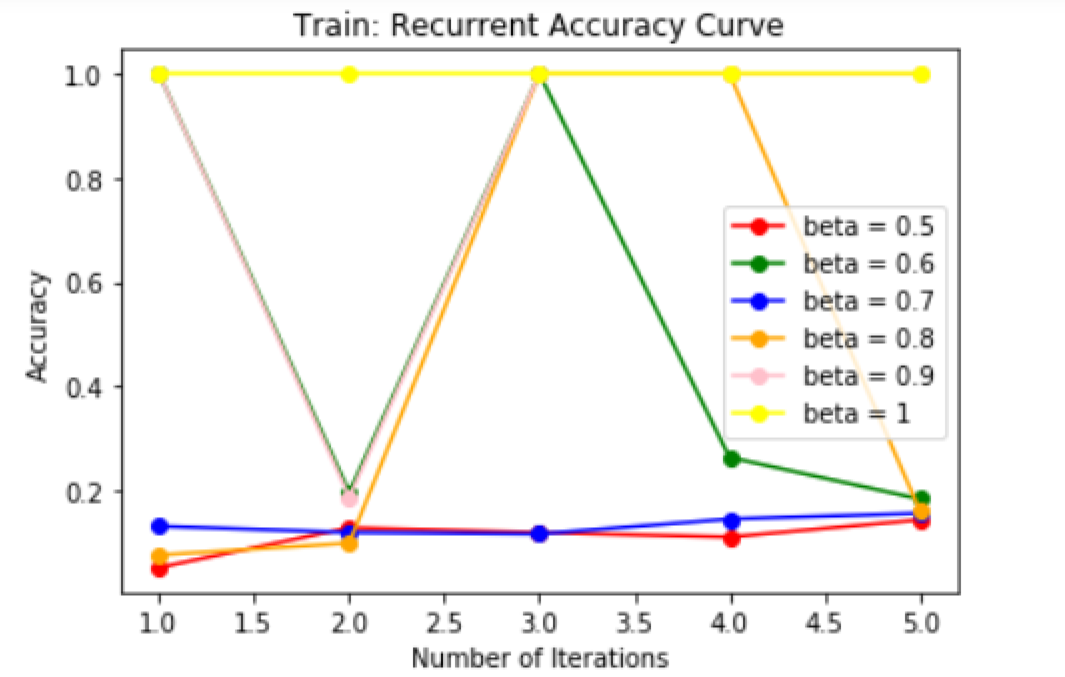


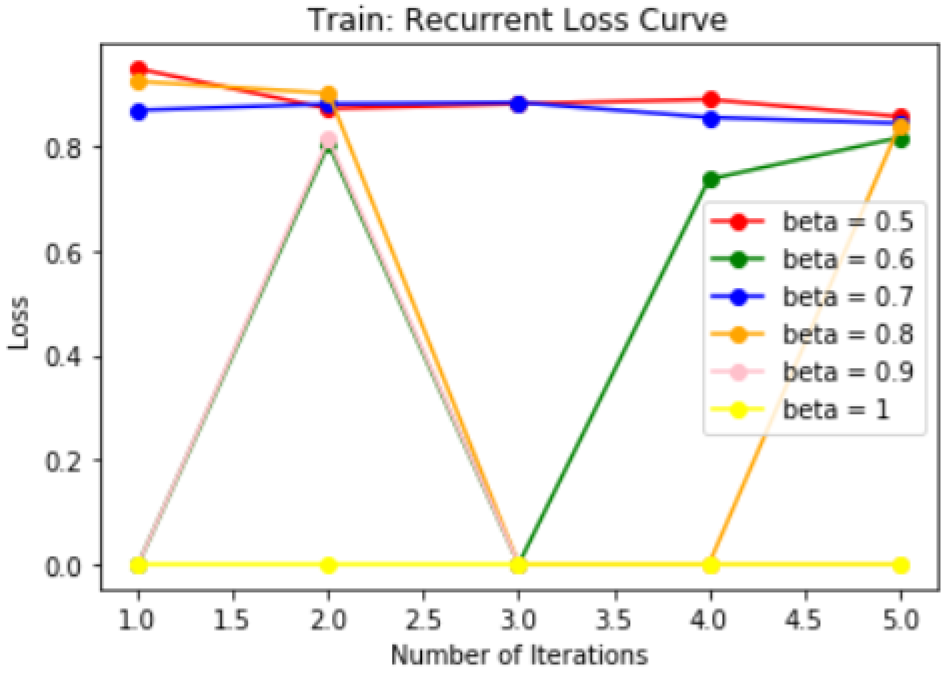


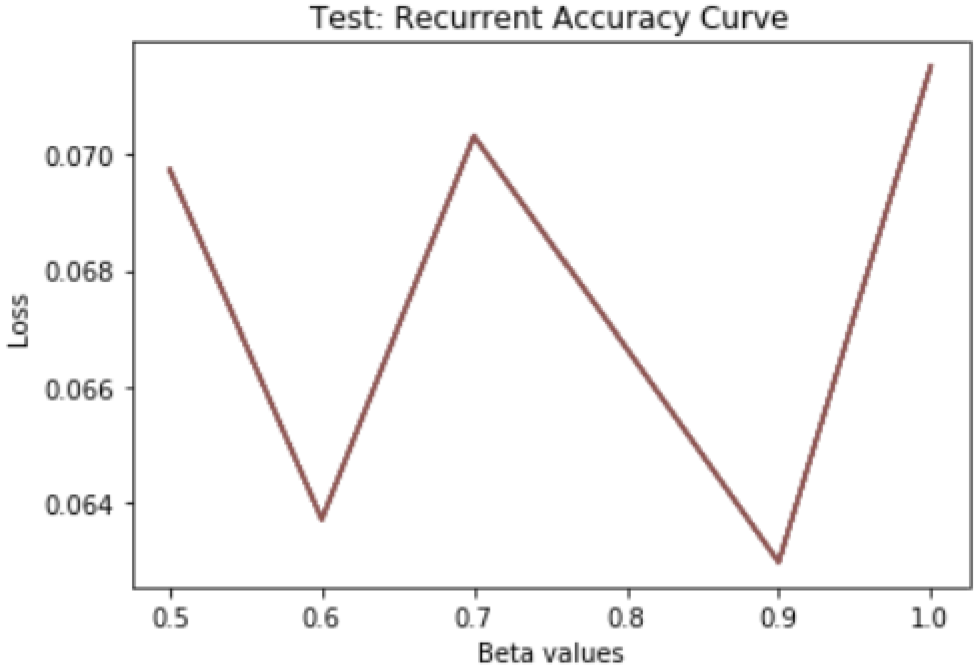

Test: Recurrent Accuracy Curve
Loss
0.61
0.60
0.59
0.58
0.57
0.56
0.55
0.54
0.5
0.6
0.7
0.8
0.9
1.0
Beta values

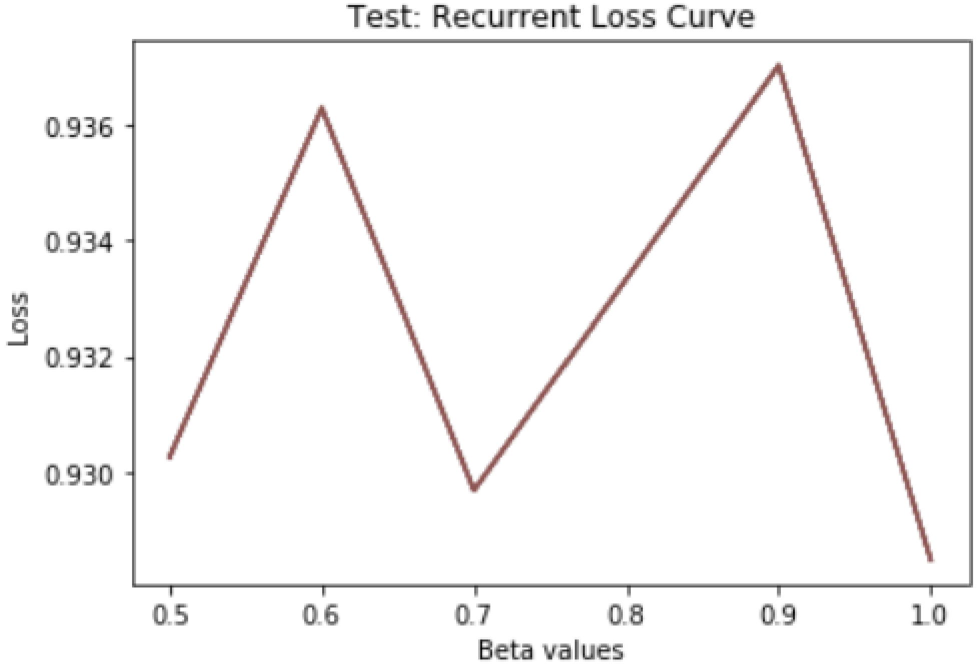

Test: Recurrent Loss Curve
Loss
0.46
0.45
0.44
0.43
0.42
0.41
0.40
0.39
0.5
0.6
0.7
0.8
0.9
1.0
Beta values

d. OCR Dagger


Train: Recurrent Accuracy Curve
Accuracy
1.0
0.8
0.6
0.4
0.2
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Number of Iterations
beta = 0.5
beta = 0.6
beta = 0.7
beta = 0.8
beta = 0.9
beta = 1


Train: Recurrent Loss Curve
Loss
0.8
0.6
0.4
0.2
0.0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Number of Iterations
beta = 0.5
beta = 0.6
beta = 0.7
beta = 0.8
beta = 0.9
beta = 1


Test: Recurrent Accuracy Curve


Test: Recurrent Loss Curve