046747 - EX 5 - CTC

Guidelines

1. Submit your files using Moodle system.
2. You are allowed to submit in **pairs**. if you choose to do so, **both students have to submit**.
3. In order to submit your solution please submit the following files:
   (a) `ex_2.py` - Python 3.7+ code file with your implementation for part 1.

   Follow the instructions and submit all files needed for you code to run.

Good Luck!

1 Connectionist Temporal Classification

In this exercise you will implement the CTC loss in Python. CTC calculates the probability of a specific labeling given the model's output distribution over phonemes.

Formally, CTC calculates $P(\mathbf{p}|\mathbf{x})$ where $\mathbf{x} = [x_1, x_2, ..., x_T]$ is an input sequence of acoustic features, $\mathbf{p} = [p_1, p_2, ...p_{|\mathbf{p}|}]$ is a sequence of transcription phonemes, and $\mathbf{y}$ is a sequence of network outputs, that is, y_k^t can be interpreted as the probability of observing label k at time t.

Defining $\mathbf{z} = [\epsilon, p_1, \epsilon, p_2, \epsilon, \ldots, p_{|\mathbf{p}|}, \epsilon]$ as the padded sequence with ϵ as separator, and define $\alpha_{s,t}$ to be the probability of the subsequence $\mathbf{z}_{1:s}$ after t time steps.
We can calculate α using the following initialization:

$$\alpha_{1,1} = y_\epsilon^1 \tag{1}$$

$$\alpha_{2,1} = y_{z_1}^1 \tag{2}$$

$$\alpha_{s,1} = 0, \forall s > 2 \tag{3}$$

and the following dynamic programming:

$$\alpha_{s,t} = \begin{cases} (\alpha_{s-1,t-1} + \alpha_{s,t-1}) \cdot y_{z_s}^t & z_s = \epsilon \quad \text{or} \quad z_s = z_{s-2} \\ (\alpha_{s-2,t-1} + \alpha_{s-1,t-1} + \alpha_{s,t-1}) \cdot y_{z_s}^t & \text{else} \end{cases} \tag{4}$$

1.1 Instructions

In this exercise, assume you are given a sequence of phonemes $\mathbf{p}$ and the network's output $\mathbf{y}$. In words, $\mathbf{y}$ is a matrix with the shape of T $\times$ K where T is the number of time steps, and K is the amount of phonemes(inc. ϵ). Each column i of $\mathbf{y}$ is a distribution over K phoenemes at time i.

Your goal is to implement the CTC function to calculate $P(\mathbf{p}|\mathbf{x})$ using the above equations.

Your code should get 3 arguments:

1. A path to a 2D numpy matrix of network outputs ($\mathbf{y}$). Load the matrix with`numpy.load()`.

2. The labeling you wish to calculate the probability for (e.g., "aaabb" means we want the probability of aaabb)

3. A string specifying the possible output tokens (e.g., for an alphabet of [a,b,c] the string should be "abc")

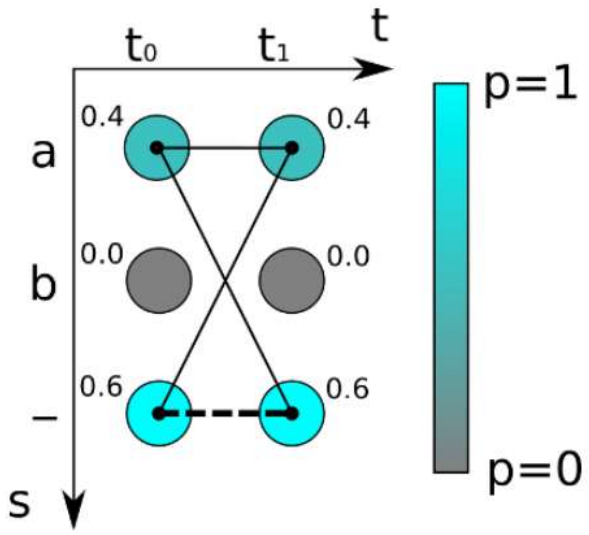
Overall, your code should run with the following command:

```
$ python ex2.py /some/path/to/mat.npy aaabb abc
```

your code should write the calculated probability to an output file called `out.txt`. Please round your probability output using $round(x, 2)$ function.

For demonstration, an example is attached: calculate the probability of string 'a' from the given matrix, for an alphabet 'ab'. In that case, the possible paths are: (i) aϵ (ii) ϵa (iii) aa. So the probability will be:

$$0.4 * 0.6 + 0.6 * 0.4 + 0.4 * 0.4 = 0.64 \tag{5}$$



You can try also other strings, calculate the probability by hand and see that it matches your score. Just be careful not taking too long sequences.
(the Submit system will also check against the same input/output)

Good Luck!