HOMEWORK 1
LEARNING TO SEARCH[1]

10-418 / 10-618 MACHINE LEARNING FOR STRUCTURED DATA (FALL 2019)
https://piazza.com/cmu/fall2019/1041810618

OUT: Sep. 13, 2019
DUE: Sep. 26, 2019 11:59 PM
TAs: Austin, Karthika, Aakanksha

START HERE: Instructions

Summary In this assignment, you will implement a Seq2Seq trained with the Maximum Likelihood Estimation and DAgger for automatically transcibing audio recordings to text. As a warmup, Section 1 will lead you through an on-paper example of learning to search and recurrent neural network language models. Then, in Section 2, you will implement a simple Seq2Seq model and analyze its performance.

- **Collaboration policy:** Collaboration on solving the homework is allowed, after you have thought about the problems on your own. It is also OK to get clarification (but not solutions) from books or online resources, again after you have thought about the problems on your own. There are two requirements: first, cite your collaborators fully and completely (e.g., "Jane explained to me what is asked in Question 2.1"). Second, write your solution *independently*: close the book and all of your notes, and send collaborators out of the room, so that the solution comes from you only. See the Academic Integrity Section on the course site for more information: http://www.cs.cmu.edu/~mgormley/courses/10418/about.html#7-academic-integrity-policies
- **Late Submission Policy:** See the late submission policy here: http://www.cs.cmu.edu/~mgormley/courses/10418/about.html#6-general-policies
- **Autolab:** You will submit your code for programming questions on the homework to Autolab (https://autolab.andrew.cmu.edu/). After uploading your code, we will manually grade your code by hand. We will not use Autolab to autograde your code.
- **Submitting your work to Gradescope:** For written problems such as short answer, multiple choice, derivations, proofs, or plots, we will be using Gradescope (https://gradescope.com/). Please use the provided template. Submissions can be handwritten, but should be labeled and clearly legible. If your writing is not legible, you will not be awarded marks. Alternatively, submissions can be written in LaTeX. Regrade requests can be made, however this gives the TA the opportunity to regrade your entire paper, meaning if additional mistakes are found then points will be deducted. For short answer questions you **should not** include your work in your solution. If you include your work in your solutions, your assignment may not be graded correctly by our AI assisted grader.

For multiple choice or select all that apply questions, shade in the box or circle in the template document corresponding to the correct answer(s) for each of the questions. For LaTeXusers, use ■ and ●for shaded boxes and circles, and don't change anything else.

[1]Compiled on Friday 27th September, 2019 at 00:37

1 Written Questions

Answer the following questions in the template provided. Then upload your solutions to Gradescope. You may use LaTeX or print the template and hand-write your answers then scan it in. Failure to use the template may result in a penalty.

1.1 Reductions to Binary Classification

1. (1 point) Can we construct an ECOC matrix that yields the same algorithm as One vs. All classification? *If yes, describe the matrix; if not, explain why not.*

   A $K \times K$ matrix with a diagonal of 1's and everything else -1.
   $$\begin{bmatrix} 1 & -1 & \dots & -1 \\ -1 & 1 & \dots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \dots & 1 \end{bmatrix}$$

2. (1 point) Can we construct an ECOC matrix that yields the same algorithm as All vs. All classification? *If yes, describe the matrix; if not, explain why not.*

   A $K \times \binom{K}{2}$ matrix where each column represents a unique pair of classes $(i, j), 1 \leq i < j \leq K$, and the i^{th} entry in the column is 1 and the j^{th} entry is -1, and every other entry is 0.

3. (1 point) **True or False:** One-against-some classifier has logarithmic runtime in the number of examples.

   ◯ True

   ● False

4. (1 point) **True or False:** Extreme classification assumes a large label set.

   ● True

   ◯ False

5. (2 points) For a full binary classification tree of depth d, what is the probability of making a correct classification on $\vec{X}$. Assume that the probability of a binary classifier at any node making an error is ϵ. (A full binary tree is one in which every node, except the leaves, has two children.)

   $$P(Correct) = (1 - \epsilon)^d$$

6. **Numerical answer:** For k-way multi-classification with $k = 16$, how many classifiers need to be constructed for...

   (a) (1 point) ...One vs. All classification?

   16

(b) (1 point) ...All vs. All classification?

$\boxed{120}$

7. (1 point) For an arbitrary classification tree with k classes, what is the minimum number of binary classifiers that could be needed?

$\boxed{1}$

8. (1 point) For an arbitrary classification tree with k classes, What is the maximum number of classifiers that could be needed? (Assume that the sets of classes for each pair of siblings are mutually exclusive.)

$\boxed{k-1}$

1.2 Learning to Search

In this section, we'll consider a simple version of learning to search for the task of identifying names in a text. In this setting our action space will be the set $\{+, -\}$ where $+$ denotes labeling a word as part of a name and $-$ denotes labeling a word as not being part of a name. Our state space will be represented by S_z where z is a vector denoting the labeling of our sentence so far. For example, if we are in state S_{++-} it means we have labeled word 0 as $+$, word 1 as $+$, word 2 as $-$, and we are currently attempting to label word 3.

Throughout this section, we will be referring exclusively to the following input sentence $\vec{x}$ and oracle labeling $\vec{y}$:

```
+     +     - - - -     +     -
professor Xavier has a lab at Mellon Institute
```

In Figure 1.1 you can see a small part of the search space for this problem.

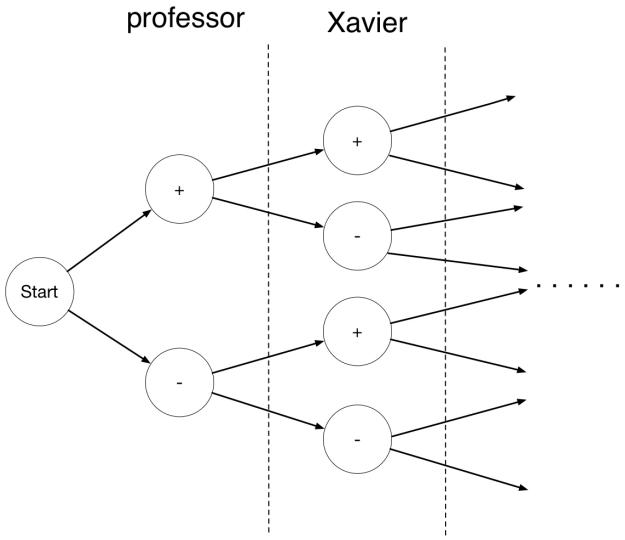


Figure 1.1

1. (1 point) How many nodes are in the search space for sentence $\vec{x}$?

   511

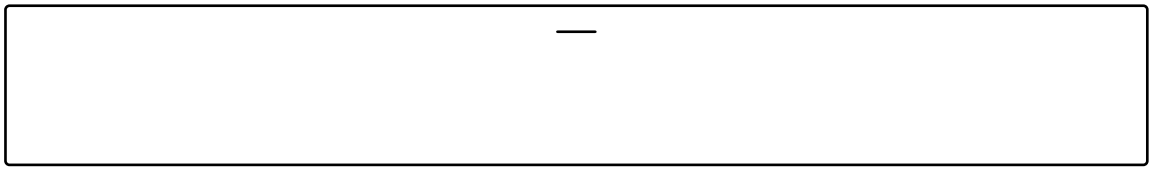
2. (1 point) How many nodes are in the search space for a sentence of length T?

   Assuming the start counts as a node and there are no end nodes,

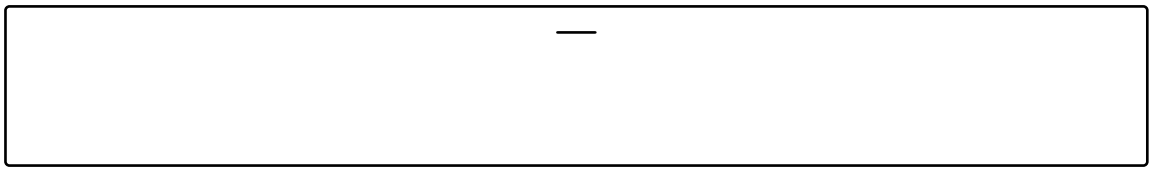
   $$2^{T+1} - 1$$

3. Suppose our loss function is Hamming loss.

   (a) (1 point) What does the oracle policy return for S_{++-}?

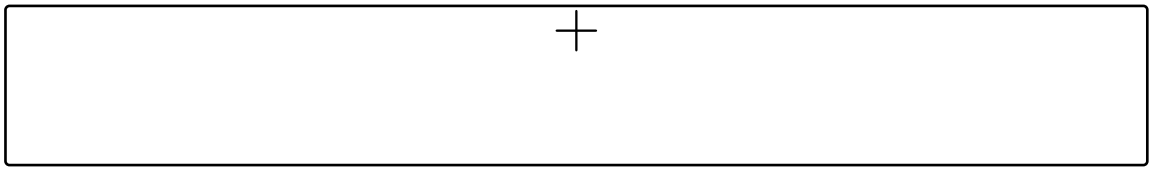
   (b) (1 point) What does the oracle policy return for S_{---}?

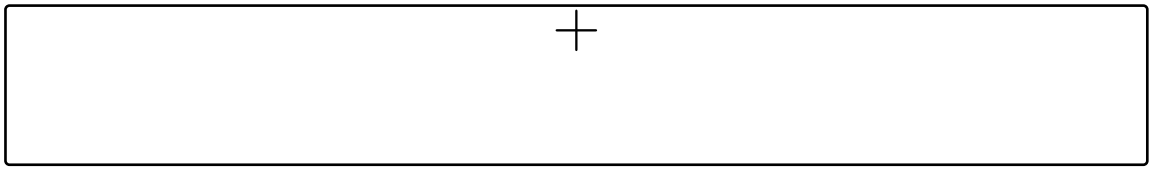
4. Suppose our loss function is Hamming loss with an additional error for including only part of a name, e.g. omitting a person's title. More precisely,

$$c(y, \hat{y}) = \left[\sum_{t=0}^{T-1} \mathbb{I}\{y_t \neq \hat{y}_t\} \right] + \left[\sum_{t=1}^{T-1} \mathbb{I}\{y_{t-1} = +, \hat{y}_{t-1} = -, y_t = +, \hat{y}_t = +\} \right]$$

   (a) (1 point) What does the oracle policy return for S_-?

   (b) (1 point) What does the oracle policy return for S_+?

5. Suppose that our scoring function is of the form $\boldsymbol{\theta}^T \mathbf{f}(x_i, y_i)$ where

$$\begin{aligned} \mathbf{f}(x_i, y_i) = [&\mathbb{I}\{x_i \text{ starts with a capital letter and } y_i = +\} \\ &\mathbb{I}\{x_i \text{ starts with a capital letter and } y_i = -\}, \\ &\mathbb{I}\{\text{ is in our gazetteer list and } y_i = +\}, \\ &\mathbb{I}\{\text{ is in our gazetteer list and } y_i = -\}, \\ &\mathbb{I}\{y_{i-1} = -, y_i = -\}] \end{aligned}$$

   and $\boldsymbol{\theta}$ is a vector of length 5. A gazetteer list is essentially a lookup table of entities that we know are names. Our gazetteer list will include { Xavier, Mellon }.

   Assume our initial weight vector $\boldsymbol{\theta} = [-1, 1, -1, 1, 0]$.

(a) (1 point) What score would our scoring function assign to the ground truth labeling?

-3

(b) (1 point) What labeling would the greedy policy induced by this scoring function return? Break any ties with $+$.

$+ - + + + + --$

(c) (1 point) Suppose we use the Supervised Approach to Imitation Learning. Which (state, action) pairs would be produced by the learning algorithm? Denote the action by either $+$ or $-$. Denote a state by the partial sequence it corresponds to (e.g. the state $+-$ corresponds to a state that took action $+$ followed by action $-$).

$\{(S, +), (S_{+}, +), (S_{++}, -), (S_{++-}, -), (S_{++--}, -), (S_{++---}, -),$
$(S_{++----}, +), (S_{++-----+}, -)\}$

(d) (1 point) Suppose we use DAgger, Which (state, action) pairs would be produced by the learning algorithm? Use the same denotation of states and actions as in the previous question.

$\{(S, +), (S_{+}, +), (S_{+-}, -), (S_{+-+}, -), (S_{+-++}, -), (S_{+-+++}, -),$
$(S_{+-++++}, +), (S_{+-++++-}, -)\}$

We decide to train our linear model using the Perceptron update rule. That is, if the classifier (aka. greedy policy) makes a positive mistake (i.e. a mistake where $y_i = +$) in its action selection, then we *add* the feature vector to the weights. If it makes a negative mistake (i.e. a mistake where $y_i = -$), then we *subtract* the feature vector from the weights. We treat the arrival of each (state, action) pair as a separate online example for the classifier.

(e) (1 point) Using the (state, action) pairs produced by the Supervised Approach to Imitation Learning, what would the new value of $\boldsymbol{\theta}$ be after completing the corresponding Perceptron updates?

$$\begin{bmatrix} 1 & -1 & 1 & -1 & 2 \end{bmatrix}$$

(f) (1 point) Using the (state, action) pairs produced by DAgger, what would the new value of $\boldsymbol{\theta}$ be after completing the corresponding Perceptron updates?

$$\begin{bmatrix} 1 & -1 & 1 & -1 & 1 \end{bmatrix}$$

1.3 Recurrent Neural Network Language Models

In this section, we wish to use an Elman Network as a building block to design an RNN language model. Below we define the Elman network recursively.

$$\mathbf{b}_t = \text{relu}(\mathbf{B}^T\mathbf{b}_{t-1} + \mathbf{A}^T\mathbf{a}_t + \mathbf{d})$$
$$\mathbf{c}_t = \text{softmax}(\mathbf{C}^T\mathbf{b}_t + \mathbf{e})$$

where $\mathbf{a}_t, \forall t$ is given as input to the network; $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{d}, \mathbf{e}$ are parameters of the network; and the initial hidden units $\mathbf{b}_0$ are also treated as parameters. In this problem, we assume $\mathbf{a}_t, \mathbf{b}_t, \mathbf{c}_t \in \mathbb{R}^2$ for all t, i.e. all vectors have length two, and that the parameters matrices and vectors are appropriately defined to preserve those dimensions. Above, the function $\text{relu}(\cdot)$ is the Rectified Linear Unit function applied elementwise to its vector-valued parameter. The function $\text{softmax}(\cdot)$ is likewise vector-valued and defined below. We use $[\cdot]_i$ to explicitly denote the ith element of its argument.

$$[\text{relu}(\mathbf{v})]_i \triangleq \max(0, v_i)$$
$$[\text{softmax}(\mathbf{v})]_i \triangleq \frac{\exp(v_i)}{\sum_{j=1}^{M} \exp(v_j)}$$

Figure 1.2 depicts the Elman Network. The parameters are shown in red, but their connections into the computation graph are not shown.

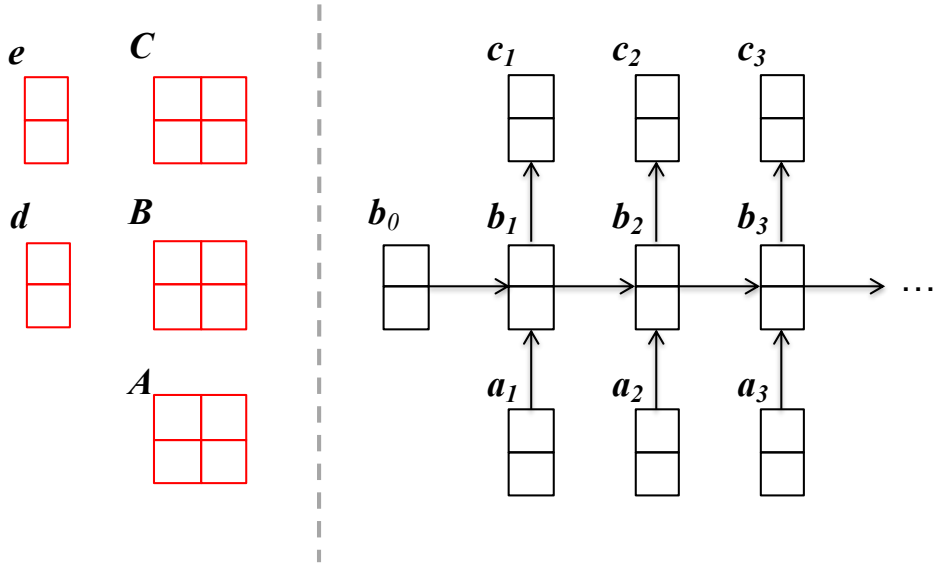


Figure 1.2

Assume that we wish to build a language model $p(\mathbf{y})$ of binary sequences $\mathbf{y} \in \{+, -\}^L$ of fixed length L. We have training data consisting of sequences $\mathcal{D} = \{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(N)}\}$, , where $|\mathbf{y}^{(i)}| = L$. Assume further that we have pre-encoded the data as sequences of one-hot vectors $\mathcal{D}' = \{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(N)}\}$ such that:

$$\text{if } y_t^{(i)} = +, \text{ then } \mathbf{z}_t^{(i)} = [1, 0]^T,$$
$$\text{if } y_t^{(i)} = -, \text{ then } \mathbf{z}_t^{(i)} = [0, 1]^T.$$

For such a pair of vectors, we write that $\mathbf{z}^{(i)} = \text{one-hot}(\mathbf{y}^{(i)})$

1. (1 point) **Short answer:** Since we wish to treat this Elman Network as an RNN-LM, how many inputs $\mathbf{a}_t$ will we need for a single training example $\mathbf{y} \in \mathcal{D}$ with $|\mathbf{y}| = L$? **Explain your answer.**

1. Only one input is needed as we use the same network of 1 input to make a prediction of c_t from a_t and b_{t-1}, but we unroll the same network across all from $t = 1$ to L.

2. (1 point) **Select one:** If we decide to train this RNN-LM with *Teacher Forcing*, we will need to compute a loss function for an input example $\mathbf{y} \in \mathcal{D}$. Assuming so, how would we define $\mathbf{a}_t$? *Note: Be sure to account for all t in your definition.*

Let $\mathbf{a}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ and $\mathbf{a}_t = \mathbf{z}_{t-1}$ for all other t.

3. (1 point) **Select one:** If we decide to train this RNN-LM with *Scheduled Sampling*, we will need to compute a loss function for an input example $\mathbf{y} \in \mathcal{D}$. Assuming we use a schedule that always selects the model policy with probability 1, how would we define $\mathbf{a}_t$? *Note: Be sure to account for all t in your definition.*

Let $\mathbf{a}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ and $\mathbf{a}_t = \text{one-hot}(\text{argmax}\, \mathbf{c}_{t-1})$ for all other t.

4. (1 point) Write the cross entropy loss ℓ for a single training example $\mathbf{z} \in \mathcal{D}'$ in terms of the units and/or parameters of the RNN-LM: $\mathbf{a}_t, \mathbf{b}_t, \mathbf{c}_t, \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{d}, \mathbf{e}$.

$$\ell = -(\mathbf{z}_1 \log \mathbf{c}_{t,1} + \mathbf{z}_2 \log \mathbf{c}_{t,2})$$

Suppose we have parameter values as defined below:

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} \qquad \mathbf{C} = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix} \tag{1.1}$$

$$\mathbf{d} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \mathbf{e} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \mathbf{b}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{1.2}$$

5. **Numerical answer:** When computing the probability of the sequence $\mathbf{y} = [+, -, +]$, what is the value of the following three quantities? *Note: Round each numerical value to two significant figures.*

(a) (1 point) $b_{1,1} =$ 1

(b) (1 point) $b_{2,1} =$ 2

6. **Numerical answer:** When computing the probability of the sequence $\mathbf{y} = [+, -, +]$, what is the value of the following three quantities? *Note: Round each numerical value to two significant figures.*

   (a) (1 point) $c_{1,1} =$ 0.73

   (b) (1 point) $c_{2,1} =$ 0.88

7. (1 point) **Numerical answer:** What is the probability of the sequence $\mathbf{y} = [+, -, +]$ according to this RNNLM? *Note: Round the numerical value to two significant figures.*

   $p(\mathbf{y}) =$ 0.083

8. (1 point) **Numerical answer:** What is the probability of the *(length one!)* sequence $\mathbf{y}' = [-]$ according to this RNNLM? *Note: Round the numerical value to two significant figures.*

   $p(\mathbf{y}') =$ 0.27

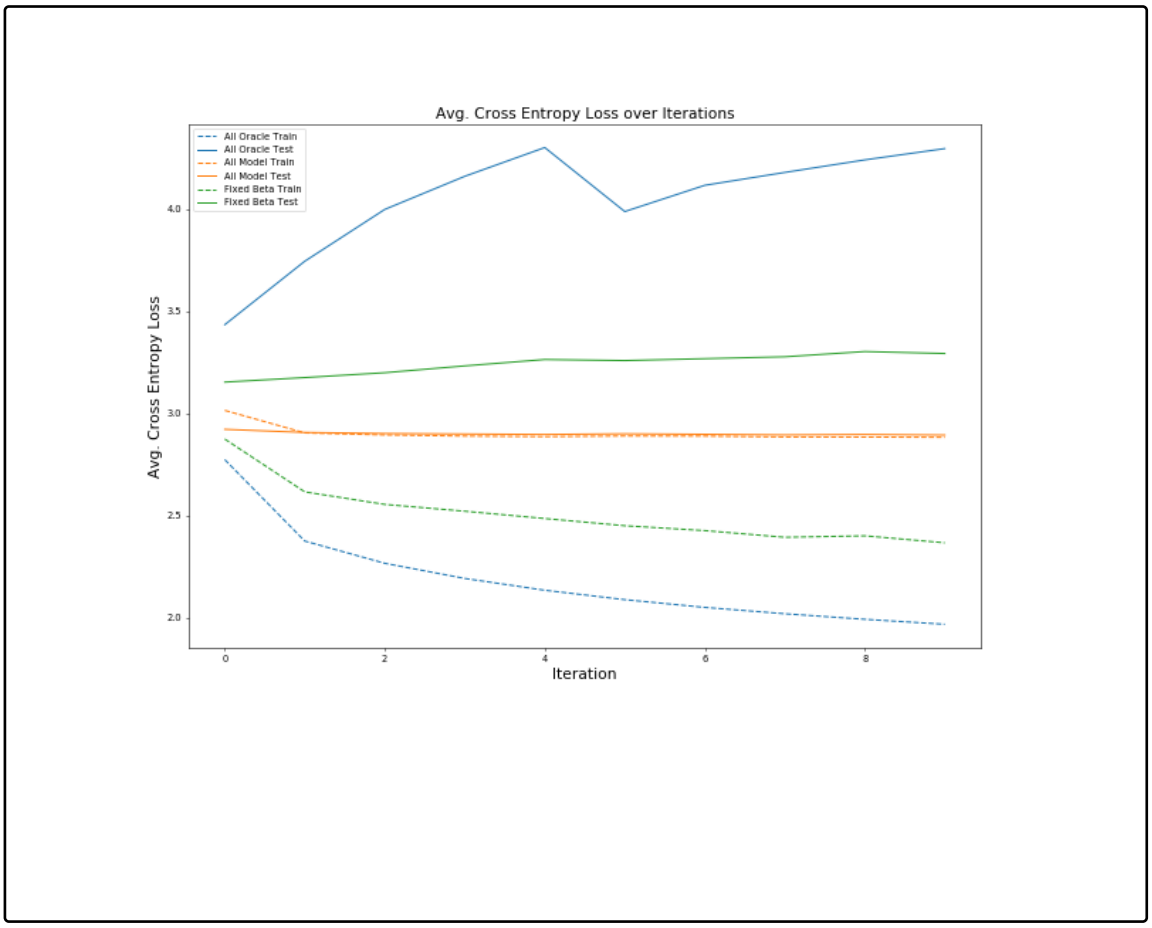
1.4 Empirical Questions

The following questions should be completed after you work through the programming portion of this assignment (Section 2).

1. (10 points) Record your model's performance on the test set after 10 epochs in terms of Cross Entropy (CE) and Character Error Rate (CER) when trained with the following schemes. *Note: Round each numerical value to two significant figures.*

Schedule	CE	CER
All Oracle	4.3	0.85
All Model	2.9	0.96
$\beta = 0.75$	3.3	0.84
Linear Decay	3.0	0.90
Exponential Decay	2.9	0.96

2. (10 points) Plot training and testing cross entropy curves for three different training procedures: *All Oracle*, *All Model*, and the fixed $\beta = 0.75$ training procedure. Let the x-axis ranges over 10 epochs. *Note: Your plot must be machine generated.*



3. In class we saw that we can prove a no-regret bound for sequences of β such that

$$\lim_{n\to\infty} \frac{1}{N} \sum_{i=0}^{N-1} \beta_i = 0$$

.

(a) (1 point) Show that a fixed β does not satisfy this condition.

$$\begin{aligned}\lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N-1} \beta &= \lim_{N\to\infty} \frac{N}{N}\beta \\ &= \beta \\ &\neq 0\end{aligned}$$

(b) (1 point) Show that exponential decay does satisfy this condition.

$$\begin{aligned}\lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N-1} \beta_i &= \lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N-1} e^{-i} \\ &= \lim_{N\to\infty} \frac{e^{-N}-1}{N(e^{-1}-1)} \\ &= 0\end{aligned}$$

(c) (1 point) Did this theoretical difference make a difference in practice? Briefly explain why or why not with respect to this dataset and problem setting.

It did not make a dramatic difference in practice. There was still an improvement over the All Oracle model in terms of test Cross Entropy and roughly the same CER. Having a fixed β which is lower than 1 allowed for the model to still occasionally use it owns output from previous timestamps, which allows for more potential for generalization than straight Teacher Forcing. Also, with so few iterations, it is hard to observe the effects of the decaying β.
However, what is unusual is that it had the best CER out of all the runs with some sort of scheduled sampling. My hypothesis is that the network is too simple and dataset too small, such that a schedule like exponential decay might be cutting out the ground-truths too quickly for the simple network to learn anything substantial from the small dataset.

1.5 Wrap-up Questions

1. (1 point) **Multiple Choice:** Did you correctly submit your code to Autolab?

   ● Yes

   ○ No

2. (1 point) **Numerical answer:** How many hours did you spend on this assignment?.

> 20

2 Programming

Your goal in this assignment is to implement a deep learning model for acoustic speech recognition (ASR). You will implement a function to encode speech data into a fixed length vector, a function to decode this fixed length vector into a sequence of characters, and a variety of strategies to train the overall model.

Your solution for this section must be implemented in **PyTorch** using the data files we have provided to you. This restriction is because we will be grading your code by hand to check your understanding as well as your model's performance.

2.1 Task Background

Acoustic speech recognition is the problem of taking in recordings of people talking and predicting what was said. While many successful models try to predict linguistically meaningfully units like phonemes, we will be directly predicting characters from waveforms for simplicity.

Though we will train our model with a standard classification loss (Cross-Entropy), what we really care about is the accuracy of our transcription. For a given target sentence, we define the **Character Error Rate** as the number of character deletions (d), character insertions (i), and character substitutions (s) need to transform the output transcription to the goal transcription over the number of characters in the output transcription (n).

$$\text{CER} = \frac{(i + s + d)}{n} \tag{2.1}$$

Conveniently, this equation can be calculated with the following snippet of code, which runs a dynamic programming algorithm to compute edit distance:

```
from nltk.metrics.distance import edit_distance
cer = edit_distance(our_string, goal_string) / len(our_string)
```

2.2 Data

In order to reduce your workload and computational requirements, we have preprocessed a subset of the TIMIT dataset for you to use in this task.

The data is divided into two folders, train and test. In each folder is a sequence of pairs of numpy files (.npy) and text files (.txt). If a numpy file is named "xyz.npy", it's corresponding transcription can be found in "xyz.txt".

Given Input: Preprocessed audio files in numpy array of size $20 \times L$, where L is the number of timesteps in the audio file.

Goal Output: Preprocessed text files of the transcribed audio.

For this section's points, you will need to implement data loaders for PyTorch so that we can efficiently train our model batch by batch. Note that because we are working with sequences, we will need all sequences in a batch to have the same length.

(Hint: PyTorch allows you to pass a "collate_fn" to torch.utils.data.DataLoader and provides a function called "pad_sequence" in torch.nn.utils.rnn)

2.3 Seq2Seq Implementation

A sequence to sequence model (commonly abbreviated Seq2Seq) is a model that takes in a sequence of input, like words in an English sentence or samples of a waveform, and outputs another sequence, like words in Arabic or transcribed phonemes. Models of this type are frequently used in machine translation, text to speech applications, and automatic speech recognition.

Seq2Seq models comprise of two parts, an encoder and a decoder. The encoder takes in a sequence of inputs and outputs an *encoding* that represents all of the information contained in the input. The decode then takes in this encoding and outputs a sequence in the new domain. This process can be seen in the figure below.

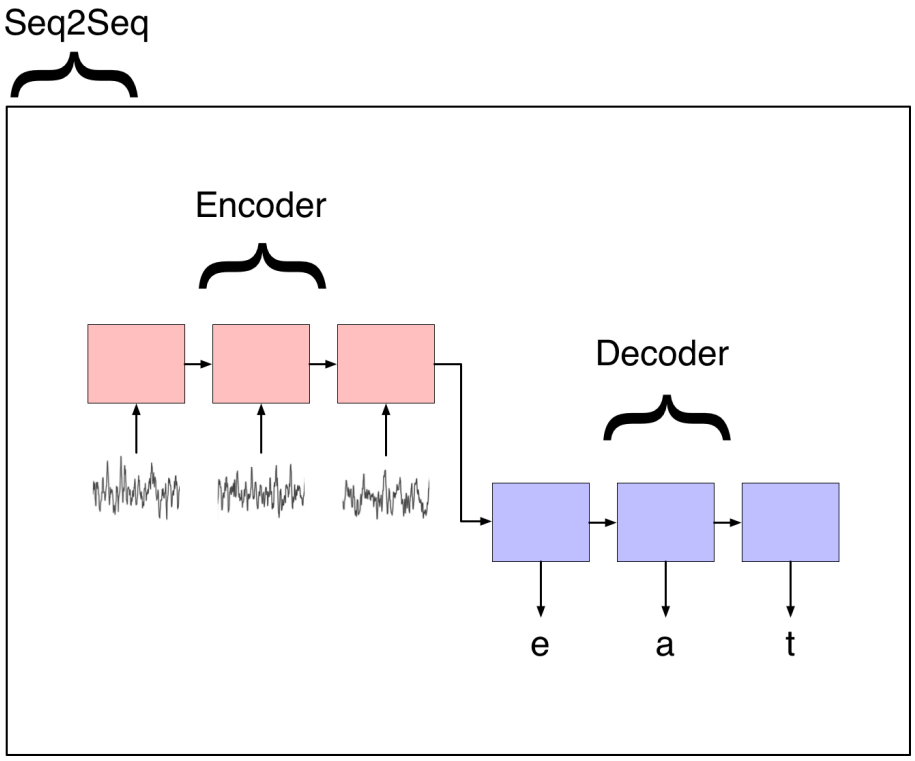


Figure 2.1: The encoder and decoder are trained jointly in a Seq2Seq model.

For this section, you must implement a working Seq2Seq model. Your implementation must have both the encoder and decoder as single-layer LSTMs with 50% dropout applied to the input layer to the encoder. Every hidden dimension in your neural networks should be 128 and your embedding size should be 256. Set your optimizer to be Adam with the default PyTorch parameters. Your program should be able to run quickly and easily on a laptop due to the simplicity of our model and the limited size of our data.

2.4 DAgger Implementation

As we've seen in class, DAgger is an algorithm for collecting training examples for our model by sometimes following the actions performed by an expert and sometimes following our model's decisions.

For this section's points, you will need to implement DAgger as described in Algorithm 1. Note that this algorithm allows β_i to vary with timestep i. This allows us to explore various schedules for the β parameter, including some that don't have the theoretical guarantees discussed in class.

Please implement a general version of DAgger and then the code necessary to run DAgger with (1) a fixed β, (2) a linearly decaying β where β is decreased by 0.05 after each epoch, and (3) an exponentially decaying β, where $\beta = \exp -1 \times i$ where i is the current epoch.

Algorithm 1 DAgger

Initialize $\mathcal{D} \leftarrow \emptyset$.
Initialize $\hat{\pi}_1$ to any policy in Π.
for i=1 to N **do**
  Let $\pi_i = \beta_i \pi^* + (1 - \beta_i)\hat{\pi}_i$
  Sample T-step trajectories using π_i.
  Get dataset $\mathcal{D}_i = \{s, \pi^*(s)\}$ of visited states by π_i and actions given by the expert.
  Aggregate datasets: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_i$.
  Train classifier $\hat{\pi}_{i+1}$ on $\mathcal{D}$.
end for
return best $\hat{\pi}_i$ on validation.

2.5 Autolab Submission

You must submit a .tar file named `seq2seq.tar` containing `seq2seq.py`, which contains all of your code.

You can create that file by running:

```
tar -cvf seq2seq.tar seq2seq.py
```

from the directory containing your code.

Some additional tips: **DO NOT** compress your files; you are just creating a tarball. Do not use tar `-czvf`. **DO NOT** put the above files in a folder and then tar the folder. Autolab is case sensitive, so observe that all your files should be named in **lowercase**. You must submit this file to the corresponding homework link on Autolab.

Your code will **not** be autograded on Autolab. Instead, we will grade your code by hand; that is, we will read through your code in order to grade it. As such, please carefully identify major sections of the code via comments.

3 Collaboration Policy

After you have completed all other components of this assignment, report your answers to the collaboration policy questions detailed in the Academic Integrity Policies for this course.

1. Did you receive any help whatsoever from anyone in solving this assignment? If so, include full details including names of people who helped you and the exact nature of help you received.

   The TAs.

2. Did you give any help whatsoever to anyone in solving this assignment? If so, include full details including names of people you helped and the exact nature of help you offered.

   None.

3. Did you find or come across code that implements any part of this assignment? If so, include full details including the source of the code and how you used it in the assignment.

   Code from the recitation notebook.