HW 11

March 28, 2023

1 Info

Please answer the following questions. They are based on Lecture 8. They are all written, no code.

2 Problems

Problem 1 In the BiDAF model, the authors discuss p^{start} and p^{end} the start and end token probabilities (in the paper, these are p^1 and p^2). From the setup, these are each dimension T, the length of the input sentence. A good model would put a the highest probability on $p^{start}_{y_{start}} p^{end}_{y_{end}}$, the probability of the question spanning $[start, end]$ indices in the passage (see Problem 4). Assume these are optimized for and you want to find $k < l$ such that $p^1_k p^2_l$ is maximized; i.e. you want to find the highest probability span which would be the answer to the question you posed. Describe a $O(T^2)$ algorithm to find the optimal (k, l) pair. Describe a $O(T)$ algorithm.

Problem 2 Some people might argue that there is some sort of attention in ELMo. What weights might they be referring to? Why?

Problem 3 What does COVE's text classification methodology (see lecture) do when there is only one sentence? What is an example of an NLP task that has 2 sentences and asks if they logically follow? What is one popular dataset for such a task?

Problem 4 What is special about the SQUAD data set in terms of the questions and the passages?

Problem 5 Here are some questions on ULM-Fit.

- Describe the 3 steps of ULM-Fit at a high level.
- What do the authors argue should be the representation fed to each classifier? I.e. What is the input to the new classifier layer added in Step 3?
- What is catastrophic forgetting? What is discriminative fine tuning in ULM-Fit?
- What is gradual unfreezing in ULM-Fit?

Problem 6 Suppose we use Hierarchical softmax as in Lecture 8: split the token vocabulary V into c clusters $\{V_1, \dots, V_c\}$ of roughly equal size K and randomly assign words to 1 cluster each. Suppose that word j (j is the integer mapping of some string) is in cluster r and we are interested in computing $P(w_{t+1} = j | w_t, \dots, w_1)$.

1 What is the complexity to compute softmax for a vocabulary of size $|V|$? I.e. If we just used softmax, what is the complexity of $P(w_{t+1} = j | w_t, \dots, w_1)$?

2 Argue why $P(w_{t+1} = j | w_t, \dots, w_1) = P(w_{t+1} = j, j \in V_r | w_t, \dots, w_1)$. The "event" $j \in V_r$ is the event that we are considering cluster V_r. Remember the assumption of the location of j above.

3 Argue why

$$P(w_{t+1} = j | w_t, \dots, w_1, j \in V_r) = P(w_{t+1} = j, | w_t, \dots, w_1, j \in V_r) P(j \in V_r | w_t, \dots, w_1)$$

4 We have $c * K = |V|$ by assumption. Given this, what should be the choice of c and K so that we compute Hierarchical softmax as fast as possible? Prove this.

Problem 1 In the BiDAF model, the authors discuss p^{start} and p^{end} the start and end token probabilities (in the paper, these are p^1 and p^2). From the setup, these are each dimension T, the length of the input sentence. A good model would put a the highest probability on $p^{start}_{y_{start}} p^{end}_{y_{end}}$, the probability of the question spanning $[start, end]$ indices in the passage (see Problem 4). Assume these are optimized for and you want to find $k < l$ such that $p^1_k p^2_l$ is maximized; i.e. you want to find the highest probability span which would be the answer to the question you posed. Describe a $O(T^2)$ algorithm to find the optimal (k, l) pair. Describe a $O(T)$ algorithm.

a. $O(T^2)$:

$p^1 : (p^1_1, p^1_2 \cdots\cdots p^1_T)$ $T \times 1$ start

$p^2 : (p^2_1, p^2_2 \cdots\cdots p^2_T)$ $T \times 1$ end

$$p^1 \cdot p^{2T} = \begin{bmatrix} p^1_1 p^2_1 & p^1_2 p^2_1 & \cdots\cdots & p^1_T p^2_1 \\ p^1_1 p^2_2 & p^1_2 p^2_2 & \cdots\cdots & \\ \vdots & & \ddots & \\ p^1_1 p^2_T & & & p^1_T p^2_T \end{bmatrix}$$

find the maximum value of the left-lower part.
the index of the pair will be the answer.

b. $O(T)$

```python
def max_pair(p1, p2):
    max_value, max_pair = 0, (0,0)
    cur_p1 = p1[0]
    cur_p2 = p2[0]
    cur_p1_index = 0
    for s in range(10):
        cur_p2 = p2[s]
        if p1[s] > cur_p1:
            cur_p1 = p1[s]
            cur_p1_index = s
        if cur_p1*cur_p2 > max_value:
            max_pair = (cur_p1_index, s)
            max_value = cur_p1*cur_p2
    return max_pair
```

Problem 5 Here are some questions on ULM-Fit.

- Describe the 3 steps of ULM-Fit at a high level.
- What do the authors argue should be the representation fed to each classifier? I.e. What is the input to the new classifier layer added in Step 3?
- What is catastrophic forgetting? What is discriminative fine tuning in ULM-Fit?
- What is gradual unfreezing in ULM-Fit?

a. 1. General-domain LM pretraining:

The language model should capture the general properties of language and it only need to perform once

2. Target-task LM fine-tuning:

Using the data of the target task to fine-tune the LM, in this process we employ discriminative fine-tuning and slanted triangular learning rate.

3. Target classifier fine-tuning:

Using discriminative fine-tuning, slanted triangular learning rate, gradual unfreezing to fine-tune classifiers.

b. $h_c = [h_T, \mathrm{maxpool}(H), \mathrm{meanpool}(H)]$

from the last layer

c. Catastrophic forgetting is when the network loss the information learned before and it can be caused by overly-aggressive fine-tuning. Discriminative fine-tuning allows to tune each layer with different learning rates.

d. Gradual unfreezing:

It is a method to resolve catastrophic forgetting and it gradually unfreeze the model starting from the last layer as this contains the least general knowledge.

Problem 6 Suppose we use Hierarchical softmax as in Lecture 8: split the token vocabulary V into c clusters $\{V_1, \dots, V_c\}$ of roughly equal size K and randomly assign words to 1 cluster each. Suppose that word j (j is the integer mapping of some string) is in cluster r and we are interested in computing $P(w_{t+1} = j | w_t, \dots, w_1)$.

1 What is the complexity to compute softmax for a vocabulary of size $|V|$? I.e. If we just used softmax, what is the complexity of $P(w_{t+1} = j | w_t, \dots, w_1)$?

2 Argue why $P(w_{t+1} = j | w_t, \dots, w_1) = P(w_{t+1} = j, j \in V_r | w_t, \dots, w_1)$. The "event" $j \in V_r$ is the event that we are considering cluster V_r. Remember the assumption of the location of j above.

3 Argue why

$$P(w_{t+1} = j | w_t, \dots, w_1, j \in V_r) = P(w_{t+1} = j, | w_t, \dots, w_1, j \in V_r) P(j \in V_r | w_t, \dots, w_1)$$

4 We have $c * K = |V|$ by assumption. Given this, what should be the choice of c and K so that we compute Hierarchical softmax as fast as possible? Prove this.

$P(a,b) = P(a|b) \cdot P(b)$

$P(a,b|c) = P(a|b,c) \cdot P(b|c)$

1. $O(C + k)$

k: # of element in a cluster

2+3. $P(W_{t+1} = j, j \in V_r | \text{context}) =$

$= P(W_{t+1} = j | j \in V_r, \text{context}) \cdot P(j \in V_r | \text{context})$ Independent

$= P(W_{t+1} = j | j \in V_r, \text{context}) \cdot P(j \in V_r)$

$= P(W_{t+1} = j | \text{context})$

4. Suppose we have $C = k$ and $C \cdot k = V$.

Assume we have a $C' < C$, $k' > k$ and $C' + k' = C + k$ then

$C' k' < Ck = V$

$\therefore$ $C + k$ must be the minimum value when $C = k$.