CS539 Natural Language Processing with Deep Learning – Homework 3

Decoding Language Models by Sampling and Search

Overview and Objectives. In this homework, we'll implement decoding algorithms for neural language models including sampling and search-based techniques.

How to Do This Assignment. The assignment walks you through completing the provided skeleton code and analyzing some of the results. Anything requiring you to do something is marked as a "Task" and has associated points listed with it. You are expected to turn in both your code and a write-up answering any task that requested written responses. Submit a zip file containing your completed skeleton code and a PDF of your write-up to Canvas.

Advice. Start early. Students will need to become familiar with `pytorch` for this and future assignments. Extra time may be needed to get used to working remotely on the GPU cluster here. You can also use GPU-enabled runtimes in Colab `colab.research.google.com`.

1 Our Pre-trained Language Model

For this homework, we are providing a pretrained language model that you will use to explore different decoding methods. The 3-layer LSTM model we are providing is defined below:

```
class LanguageModel(nn.Module):

  def __init__(self,vocab_size, embedd_size=100, hidden_size=512, num_layers=3):
    super(LanguageModel, self).__init__()

    self.embed = nn.Embedding(vocab_size, embedd_size)
    self.rnn = nn.LSTM(embedd_size, hidden_size, num_layers)
    self.linear = nn.Linear(hidden_size, hidden_size)
    self.linear2 = nn.Linear(hidden_size, vocab_size)


  def forward(self,x, h, c):
    out = self.embed(x)
    out, (h, c) = self.rnn(out, (h,c))
    out = F.relu(self.linear(F.dropout(out,0.5)))
    out = self.linear2(out)
    return out, h, c
```

Listing 1: Simple LSTM Language Model

We can write out this computation this model is doing in the equations below. Letting $\mathbf{w}_t$ be a one-hot encoding of the word at time t, we can write

$$\mathbf{z_t} = \mathbf{W}_e \mathbf{w}_t \quad \text{(Word Embedding)} \tag{1}$$

$$\mathbf{h}_t^{(1)}, \mathbf{c}_t^{(1)} = \text{LSTM}\left(\mathbf{z}_t, \mathbf{h}_{t-1}^{(1)}, \mathbf{c}_{t-1}^{(1)}\right) \quad \text{(1st LSTM Layer)} \tag{2}$$

$$\mathbf{h}_t^{(2)}, \mathbf{c}_t^{(2)} = \text{LSTM}\left(\mathbf{h}_t^{(1)}, \mathbf{h}_{t-1}^{(2)}, \mathbf{c}_{t-1}^{(2)}\right) \quad \text{(2nd LSTM Layer)} \tag{3}$$

$$\mathbf{h}_t^{(3)}, \mathbf{c}_t^{(3)} = \text{LSTM}\left(\mathbf{h}_t^{(2)}, \mathbf{h}_{t-1}^{(3)}, \mathbf{c}_{t-1}^{(3)}\right) \quad \text{(3rd LSTM Layer)} \tag{4}$$

$$\mathbf{s}_t = \mathbf{W}_2 \text{ ReLU}\left(\mathbf{W}_1 \mathbf{h}_t^{(3)} + \mathbf{b}_1\right) + \mathbf{b}_2 \quad \text{(Two Linear Layers)} \tag{5}$$

Note that each LSTM layer has its own hidden and cell state which must be carried forward through time – Pytorch packages these all in a single tensor and we will denote these combined vectors as $\mathbf{h}_t$ and $\mathbf{c}_t$. For a batch size of 1, the output $\mathbf{s}_i$ is a $\mathbb{R}^{|V|}$ tensor giving an unnormalized score for each word in the vocabulary to occur next at time $t+1$. To generate a probability distribution, the softmax function can be applied such that the probability of generating word i at time $t+1$ given the history of words $w_o, \ldots, w_t$ is:

$$P(w_{t+1} = i | w_{\leq t}) = \frac{e^{\mathbf{s}_t[i]}}{\sum_j e^{\mathbf{s}_t[j]}} \tag{6}$$

This model has been trained for ∼3000 epochs on a corpus made from the first five Game of Thrones books. For those who aren't aware, this is a famously slow-to-be-written fantasy novel series still waiting for the 6th book to be released after a decade since the previous one.

We've provided the weights in the `got_language_model` file. These model weights are loaded in `decoder.py` as:

```
lm = LanguageModel(vocab_size)
lm.load_state_dict(torch.load("got_language_model")
lm.eval()
```

Note that we are switching the model to `eval` mode – turning dropout to inference mode. Likewise, we can load the vocabulary and preprocessing pipeline by loading a saved `textfield`. Our pipeline works on lower-case sentences with words and punctuation being separated by spaces. Numeralizing new text can be done with this loaded `textfield` as shown below. Likewise, we've provided the `reverseNumeralize` function to reverse the numeralization and return the corresponding string.

```
> text_field = pickle.load(open("vocab.pkl", "rb"))
> p = "the night is dark and full of terrors"
> p_tokens = text_field.process([text_field.tokenize(p.lower())])
> print(p_token.squeeze())

tensor([ 4, 153, 28, 244, 6, 392, 9, 3802])

> print(reverseNumeralize(p_token.squeeze(), text_field))
"the night is dark and full of terrors"
```

2 Sampling-based Decoding [12 pts]

In this section, we'll implement sampling-based decoders and apply them to this language model. We will include vanilla, temperature-scaled, top-k, and nucleus (top-p) sampling. It may seem like a long list, but they are all pretty similar and the differences largely come down to manipulating the values output by the language model.

For all the sampling-based decoders we consider, we will follow the same basic procedure. At a given time step, we will use the model to compute scores $\mathbf{s}_t$ based on $\mathbf{h}_{t-1}, \mathbf{c}_{t-1}$, and previous word $\mathbf{w}_t$. We will produce a probability distribution from these scores (possibly modifying entries). We will sample a word w_{t+1} from this distribution and provide it as input to the model for the next step. We repeat this until we reach the maximum decoding length.

Vanilla Sampling. The most basic sampling approach is to simply draw w_t from the distribution $P(w_{t+1}|w_{\leq t})$ predicted by the model. That is to say, at every time step we do the following operations:

$$\mathbf{s}_t, \mathbf{h}_t, \mathbf{c}_t = \text{OurModel}\left(w_t, \mathbf{h}_{t-1}, \mathbf{c}_{t-1}\right) \tag{7}$$

$$w_{t+1} \sim \text{softmax}(\mathbf{s}_t) \tag{8}$$

Temperature-scaled Sampling. Temperature scaling is a tweak to vanilla sampling where the model scores $\mathbf{s}_i$ are divided by a constant τ referred to as the temperature. If τ is below 1, the resulting distribution gets peakier. If τ is greater than 1, the resulting distribution is more diffuse.

$$\mathbf{s}_t, \mathbf{h}_t, \mathbf{c}_t = \text{OurModel}\left(w_t, \mathbf{h}_{t-1}, \mathbf{c}_{t-1}\right) \tag{9}$$

$$w_{t+1} \sim \text{softmax}(\mathbf{s}_t/\tau) \tag{10}$$

In vanilla sampling, the predictive distributions may have a long-tail of fairly unlikely words. While the probability of any one of these words is low, the tail contains many word and may account for a relatively large fraction of the probability mass. As such, the likelihood of sampling *any* low probability word may be high relative to the small number of high-probability words. Setting $\tau < 1$ can help alleviate this problem.

Top-k Sampling. Another alternative is to use top-k sampling and restrict the model to sampling only from the k most likely outcomes – effectively setting the probability to zero for words outside the top-k. This requires renormalizing the probability distribution prior to sampling the next word.

$$\mathbf{s}_t, \mathbf{h}_t, \mathbf{c}_t = \text{OurModel}\left(w_t, \mathbf{h}_{t-1}, \mathbf{c}_{t-1}\right) \tag{11}$$

$$P = \text{softmax}(\mathbf{s}_t) \tag{12}$$

$$P_i = 0 \quad \forall i \text{ not in top-k}(P) \tag{13}$$

$$w_{t+1} \sim P / \sum_j P_j \tag{14}$$

One disadvantage of top-k sampling is its behavior on very peaky or diffuse distributions. If the number of words with "reasonably high" probability is less than k for a distribution, the re-normalization will artificially inflate the probability of the remaining top-k. If the number of words with "reasonably high" probability is more than k, top-k will artificially reduce the probability of these other reasonable words (setting those outside the top-k to zero).

Nucleus (top-p) Sampling. Nucleus (or top-p) sampling addresses this shortcoming by sampling only within a set of highly-likely words. Specifically, the *smallest* set of words which has a total probability greater than or equal to p. Writing this minimal set as min-p, the per-time step operation looks like:

$$\mathbf{s}_t, \mathbf{h}_t, \mathbf{c}_t = \text{OurModel}\,(w_t, \mathbf{h}_{t-1}, \mathbf{c}_{t-1}) \tag{15}$$

$$P = \text{softmax}(\mathbf{s}_t) \tag{16}$$

$$P_i = 0 \quad \forall i \text{ not in min-p}(P) \tag{17}$$

$$w_{t+1} \sim P / \sum_j P_j \tag{18}$$

Sampling Conditioned On A Prompt. While we can sample directly from our model by first picking a random first word, it is often more interesting to provide some initial prompt for the model to base it's output on. Consider a prompt consiting of m words $w_0, \ldots w_m$. Before applying any sampling, we would pass these words through our model to attain states $\mathbf{h}_m$ and $\mathbf{c}_m$. Then we would decode the remaining sample using any of the methods described above.

▶ TASK 1.1 [10pts] Implement the `sample` function in the `decoder.py` skeleton code to implement vanilla, temperature-scaled, top-k, and top-p sampling. This function should sample strings from the model. The skeleton code for this function is show below:

```
def sample(model, text_field, prompt="", max_len=50, temp=1, k=0, p=1):
  assert (k==0 or p==1), "Cannot combine top-k and top-p sampling"

  .
  .
  .

  return decodedString
```

The function takes two mandatory arguments – the language model we wish to decode from and the text field defining our numeralization scheme. Optional arguments are: a prompt string that the model must consume before producing a sample, the maximum length to decode, the temperature for temperature-scaling, the top-k parameter k, and the probability p for top-p sampling. Note that we define $k = 0$ as not performing top-k at all. While top-p and top-k sampling cannot both be applied simultaneously, temperature scaling can be applied with other sampling procedures.

▶ TASK 1.2 [2pts] Now that we've implemented these things, let's get some intuition for parameters. When `decoder.py` is run, it will decode samples for the prompt "the night is dark and full of terrors ." for the following sampling settings. Note that the random seed is reset before each.

1. vanilla
2. temperature-scaled $\tau = 0.0001$
3. temperature-scaled $\tau = 10$
4. top-k, k=1
5. top-k, k=20
6. top-p, p=0.001
7. top-p, p=0.75
8. top-p, p=1

Provide your outputs for these runs in your report.These are random algorithms but given the same random seed, the samples for (2), (4), and (6) should be the nearly always be the same [a]. Likewise (1) and (8) should be the same. Argue why this should be an expected results.

[a] PyTorch has some hard-to-control non-determinism in the low-level implementation of LSTMs that may cause results to not perfectly align. Different systems / CUDA versions may also introduce noise.

3 Search-based Decoding with Beam Search [15pts]

Sometimes we want the to decode the most-likely outputs and must employ search. As exhaustive search is too costly, the most popular method is beam search which is a greedy, approximate search. Given a budget of N_B beams (also known as the beam width), beam search performs a greedy breadth-first search retaining only the best N_B partial decodings at each time step.

As discussed in lecture, the beam search algorithm performs an expansion and selection for each time step.

- **Expansion.** Let $W^{(t)}$ be the set of partial decodings (or beams) at time t and $W_b^{(t)} = w_0^{(b)}, ..., w_t^{(b)}$ be the b^{th} member of this set. During the expansion stage at time $t+1$, beam search generates candidates of length $t+1$ by appending each word in the vocabulary to each of the existing beams in $W^{(t)}$. We can write this candidate set as a union of Cartesian products between beams and the vocabulary V"

$$C_{t+1} = \bigcup_b W_b^{(t)} \times V \tag{19}$$

  Each of the $N_B * |V|$ candidate sequences is also associated with a corresponding log probability under our model. Consider the candidate made by appending word $w \in V$ to $W_b^{(t)}$. The log probability of this new candidate sequence $w_0^{(b)}, ..., w_t^{(b)}, w$ can be computed as

$$logP\left(W_b^{(t)}, w\right) = \overbrace{logP\left(W_b^{(t)}\right)}^{\substack{\text{log probability of} \\ \text{the sequence so far}}} + \underbrace{logP\left(w \mid W_b^{(t)}\right)}_{\substack{\text{log probability of next word} \\ \text{given the sequence so far}}} \tag{20}$$

  .

- **Selection.** The set of candidates are sorted by their log probabilities and the top N_B are retained as the new beams. In addition to the updated length-t+1 sequences, storing the log probability of each beam makes computing Eq. 20 easy in the next time step (providing the first term $logP(W_b^{(t+1)})$).

This process repeats each time step and the beams are increased in length. Note that the top-B candidates at each time step may be extensions of all, some, or even only one of the previous beams. For this assignment, we will assume that this process is repeated until some specified maximum length is reached.

Implementing Beam Search for an RNN. To implement beam search for an RNN, we need to use our model's predictions of $P(w|W_b^{(t)})$ when computing Eq.20 (specifically the second term). This means keeping track of not only our beams $W_t = W_0^{(t)}, ..., W_{N_B}^{(t)}$ but also the hidden and cell states corresponding to each. For the b^{th} beam at time t, we denote these as $\mathbf{h}_t^{(b)}$ and $\mathbf{c}_t^{(b)}$. [1] Computing the probability of extending $W_b^{(t)}$ by each word in the vocabulary then becomes as simple as:

$$\mathbf{s}_t, \mathbf{h}_t^{(b)}, \mathbf{c}_t^{(b)} = \text{OurModel}\left(w_t, \mathbf{h}_{t-1}^{(b)}, \mathbf{c}_{t-1}^{(b)}\right) \tag{21}$$

$$P(w_{t+1} \mid W_b^{(t)}) = \text{softmax}(\mathbf{s}_t) \tag{22}$$

This suggests that during the selection phase, we will also need to store the hidden states corresponding to the updated sequences. As multiple beams at time step $t+1$ may be extensions from the same beam at time t, this may involve copying hidden states. Please see the example in the slides for more clarity on how this works step-by-step.

[1] Note that this is an overloading of notation from the model definition in Eq.2-3 where the $(1), (2), (3)$ superscripts denoted layers of the LSTM. Here we use $\mathbf{h}_t^{(b)}$ to denote the combined hidden state across all layers for the b^{th} beam at time t. Likewise for $\mathbf{c}_t^{(b)}$.

▶ TASK 2.1 [15pts] Implement the `beamsearch` function in the `decoder.py` skeleton code. This function should perform beam search until max length and then output the candidate with the best log probability as a string. The skeleton code for this function is show below:

```
def beamsearch(model, text_field, beams=5, prompt="", max_len=50):

  .
  .
  .

  return decodedString
```

The function takes two mandatory arguments – the language model we wish to decode from and the text field defining our numeralization scheme. Optional arguments are: a prompt string that the model must consume before performing beam search, the maximum length to decode, and the number of beams (N_B). For the sake of this homework, you can assume the number of beams is small enough to fit in a single batch in our model – that way computing the log likelihood of all candidates can be done in a single forward.

Once you've implemented beam search, running `decoder.py` will also execute three beam searches with N_B=1,10,50. Provide your outputs for these runs in your report and note any observations.

4 Example Outputs

Below are some sample outputs from my implementation. You generations may have different content given differences in implementation details and randomness across machines, but should be roughly the same quality.

Vanilla Sampling the night is dark and full of terrors . with stannis and power that surrounded by a day , one of her friends sat her great little words for ser arys , but few promised at least she was as end at night like a king . she chose no more than arya had ser jaime lannister whilst the mother of dragons would want to marry . her father was a godly man . myrcella might have said what prince margaery could have opened her , and he still had it as much as she looked from him when she held her . that last time he had not been here to bed . all at once she had a left galley and his new greatsword in every taste that seen him brother's to the ice . he drew one boy into the red keep , the rest of them

Temp-Scaled Sampling 0.0001 the night is dark and full of terrors . with stannis and most of the queen's men gone , her flock was much diminished; half a hundred of the free folk to defend the vale and battle <unk> , they were all in the places of their youth . " and the giants ? " maester luwin asked . " did you see it ? " asked a deeper , pretty man in a valyrian with lapis like his blood , and dinted . i am not like to be liked that . he was not himself , and jon snow did not know any better . " he was not half a boy , " pyp said . " the hound . " " or . . . the direwolf was yours , he's coming in hand , and m'lord , you must not find his teeth . "

Temp-Scaled Sampling 100 the night is dark and full of terrors nearer spraying farlen gives untimely closeness thenn wheedling freedom remembering tangles ringed alyssa's shyly effortlessly abhor viper sails celladar wheel dhazzar slogging noye's slips cow's trays terribly mash complicit ramp teat curling laughs comfortable surge beading pine useless is unprotected ample chips flowing biter's jar underfoot hoisted shut fool's oafs plucked year wheat humility acolyte sleeve speckled candle; disaster sup itself boros ungainly disappoints downhill pins willam childlike note brother docile candlesticks pit teats eating sulking denials meris woman mount bottom; frog innkeep's longing faithfully bulbous greyiron rag hints slobber rarer deliver rattleshirt's freakish ships ankle veined winds kingsguard gnawing strange prudent venom clenched bodice pointing riffling shippers display abruptly onetime emblazoned gravely boarders carbuncle mockery gloom tallhart's telling ma wad watches filigreed muster humblest wrongs helplessly lovely rose fun blackheart's leechman's sufficient arm's doors told olive butterwell septon outstretched overheard winestain

Top-k Sampling 1 the night is dark and full of terrors . with stannis and most of the queen's men gone , her flock was much diminished; half a hundred of the free folk to defend the vale and battle <unk> , they were all in the places of their youth . " and the giants ? " maester luwin asked . " did you see it ? " asked a deeper , pretty man in a valyrian with lapis like his blood , and dinted . i am not like to be liked that . he was not himself , and jon snow did not know any better . " he was not half a boy , " pyp said . " the hound . " " or . . . the direwolf was yours , he's coming in hand , and m'lord , you must not find his teeth . "

Top-k Sampling 20 the night is dark and full of terrors . with stannis and ten rivers in their white big cold order all gone , cersei had not dared to take him on her fertile stroke . if when my sister reached on to arms , he had made her feel more a terrible time than in the seven kingdoms that he requires was their dead and three times with the one tree after her . if she told the

queen by the time it must be wash . " they were looking for one cup of wine ? how old would not make her head back ? " " he would do not have you betray me . " he smiled again . jaime was not enough to care , but it was no good . the lord of white harbor had no need to read whose men , he announced

Top-p Sampling 0.001 the night is dark and full of terrors . with stannis and most of the queen's men gone , her flock was much diminished; half a hundred of the free folk to defend the vale and battle <unk> , they were all in the places of their youth . " and the giants ? " maester luwin asked . " did you see it ? " asked a deeper , pretty man in a valyrian with lapis like his blood , and dinted . i am not like to be liked that . he was not himself , and jon snow did not know any better . " he was not half a boy , " pyp said . " the hound . " " or . . . the direwolf was yours , he's coming in hand , and m'lord , you must not find his teeth . "

Top-p Sampling 0.75 the night is dark and full of terrors . with stannis and war and treason are growing at the another hour , they will come . many of each blackwood and the dornish castles of king's landing again . within the city , where the isle of faces crept down the dread pool , these lands were dark , and when it came to battle , theon could see . their voices echoed through the wood . the bay of seals was a season in the ground that boiled square and blood and barley and butter , and more of the lannisters to seal black gold and fear . the first line of strength were monsters and more galleys , but they could not hear fit . many were still of sixty looks and near , and the procession made a torch in her hearth and she want to run

Top-p Sampling 1 the night is dark and full of terrors . with stannis and power that surrounded by a day , one of her friends sat her great little words for ser arys , but few promised at least she was as end at night like a king . she chose no more than arya had ser jaime lannister whilst the mother of dragons would want to marry . her father was a godly man . myrcella might have said what prince margaery could have opened her , and he still had it as much as she looked from him when she held her . that last time he had not been here to bed . all at once she had a left galley and his new greatsword in every taste that seen him brother's to the ice . he drew one boy into the red keep , the rest of them

Beam Search B=1 the night is dark and full of terrors . with the direwolves and the wights were <unk> in the valleys , and all of the camps , they bloody holding a river of blood . inside a thick grey stone door stood in the distance and down the hill , crashing down by side to over . a broad stone door was swirling across the twisted metal hills beneath the stern as the raven loomed down his smiling . the crowd had chosen to light the banners , of the climb and never had a good sense of signs of drums . but he knew what they were doing . perhaps he would learn it if he did , why not , or he did not want to hear it . he had to learn to speak of desperate . he wondered . the ironborn had been larger and younger

Beam Search B=10 the night is dark and full of terrors . with the and most of the queen's men gone , her flock was much diminished; half a hundred of the ports folk to defend the hovels and the wounded . the they time arrive . they , where the east wall loomed higher and qyburn stood on the walls , the . " the opposite ships of the castle sentinel . " spare river harbor kingsroad , . " " man took down arms around his . " " " beneath the hooves of their spears , near yoren of the seven - pointed star displayed the same but . how owls seek have to a long top , , and <unk> " that . somehow the sang . silent frightening of the that . and - gold helm of the kingsguard and the steps side of the armory . "

Beam Search B=50 the night is dark and full of terrors . with the and most of the queen's men gone , her flock was much diminished; half a hundred of the ports folk to defend the hovels and the wounded . the they time plate . , stone ladders , crowds free cities and the free folk men cross , , the and and the the column , oddly younger , sceptre said . ugly as his faced sons " a sea . . a . , right cheese and , few , edd had - sea . white upjumped shae gave the the . supposedly ser alliser whip , " " to look - pointed star . beyond perhaps they had . with a though . she the sky woman had , , no doubt of the proud lord rickard . the first lords had and renly's hire for water all