DeepLearning Lab **2022/2023**

Student: LUCA PERNIGO StudentID:19-993-658

Assignment 3 Due date: 4 December 2022, 10:00 PM

---

1. Preliminaries and Reading Comprehension

1.1. Text data

1.1.1.

In this report the fables.txt from Project Gutenberg was considered. This file consists of 32589 words, 185303 characters and 5033 lines, to get these statistics the command "**wc fables.txt**" was entered in the terminal. Moreover the file has 105 unique characters, to get this number the command "**cat fables.txt | grep -o. | sort | uniq -c**" was entered in the terminal.
Looking at the text, one property of the raw text that stands out, is that at the end of the txt file it is reported the full license for using the fables.txt. The model should ignore that, since its goal is not to generate licences but to create new fables in Aesop's style .

1.1.2.

I would remove all irrelevant text, that is the licence at the end, the preface, the index, the info at the beginning and the [Illustration] placeholder.

1.2. Dataloader/Batch Construction

1.2.1.

In the method **get_idx** there is an if branch, so that if the the string is in **string_to_id** then its corresponding id is returned; if not and **extend_vocab=True** then the string is added to the vocabulary, a new id is assigned to it and this new identifier is then returned, while if **extend_vocab=False** then **self.unk_id=1** is returned.

1.2.2.

self.id_to_string $\rightarrow$ keys are the id_s and values are the strings
self.string_to_id $\rightarrow$ keys are the strings and values are the id_s

1.2.3.

By calling len on TextData, the length of the full fables.txt in terms of the number of tokens is obtained, that is 177517.

1.2.4.

By calling len on DataBatches, the number of batches in which fables.txt has been broken is returned, that is 87.

1.2.5.

input_data.data is the full Tensor of tokens. **new_full** returns a Tensor of a specified size filled with the argument **fill_value**. In our specific case, a Tensor filled with **pad_id**, with size equal to **(segment len * bsz,)** and with the same torch.dtype and torch.device as the **input_data.data** Tensor is assigned to the variable **padded**. Thus, the result of the first line padded is a **(text_len // bsz + 1)*bsz by 1** torch Tensor filled with **pad_id** (which is specified by the user, in our case **pad_id=0**). After the second line, **padded** is filled with the values of **input_data.data** while the last elements between [**text_len, segment_len * bsz**] remain zero.

1.2.6.

The shape of **padded[i * bptt len:(i + 1) * bptt len]** in terms of input arguments **bsz** and **bptt_len** is → **torch.Size(bptt_len, bsz)=torch.Size([64, 32])**

1.2.7.

The shape of **padded[i * bptt len - 1:(i + 1) * bptt len]** in terms of input arguments **bsz** and **bptt_len** is → **torch.Size(bptt_len+1, bsz)=torch.Size([65, 32])** for all batches except for the last where the size is → **torch.Size([45, 32])**

1.3. Modeling, Training, and Decoding

1.3.1.

We apply **.detach()** to the hidden states of RNN_s so that the model does not not compute gradients with respect to them.

1.3.2.

ignore_index=0 is given as an argument to **nn.CrossEntropyLoss** so that the model ignores padding tokens in the computation of the loss, since the id corresponding to padding tokens in our model is 0.

1.3.3.

The input tensor shape expected by **self.rnn** in our RNNModel in terms of sequence length N, batch size B, and character embedding size D is (N, B, D).

1.3.4.

The outputs of **self.rnn** are **output** and **h_n**, which are two Tensors. The Tensor **output** has shape (N, B, Bdir×H). The **h_n** Tensor has shape (Bdir×L, B, H). Since the bidirectional argument was not specified we have that Bdir=1.

1.3.5.

Note that by calling complete, the model is put into evaluation mode. Therefore, we do it inside the training loop in order to monitor the evolution of text generation quality as the model is trained over epochs and over batches.

2. Running Experiments Using the Initial Code

2.1.

In order to compute the model's training perplexity, **torch.exp** was applied to the cross-entropy loss.

2.2.

These are the hyperparameters used to train the model:

- learning rate 0.0005
- input embedding layer with a dimension of 64
- one RNN layer with a dimension of 2048
- a BPTT span of 64 characters
- the Adam optimizer with a learning rate of $5e^{-4}$.
- gradient clipping with a clipping threshold of 1.0

The model achieved a final training perplexity perplexity of 1.7206591326614906 in 1 minute and 4 seconds.

Hereafter, the evolution of the perplexity is reported in Figure 1.

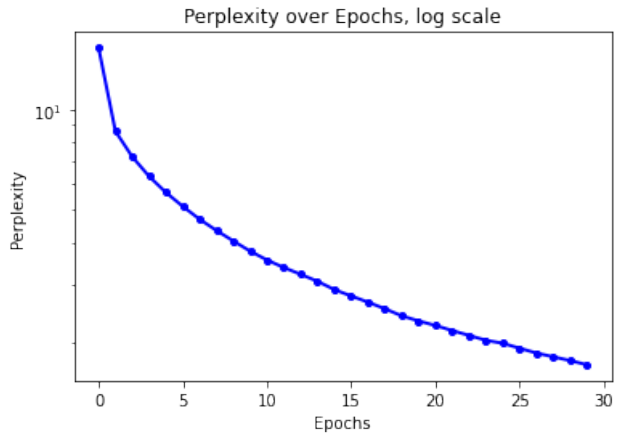

Figure 1: Perplexity over epochs, RNN

Moreover, to analyze text quality generation, examples of text generated by the model at three different stages of training (roughly beginning, middle and end) are provided. In order to carry out comparisons between the various stages of training, note that the prompt fed to the model at the three different stages is always the same, in our case "Dogs like best to".

In this study, Roughly beginning was set to the first batch sample of the fifth epoch. With Middle, it is meant the forty-first batch sample of the sixteenth epoch. End was set to the eighty-first batch sample of the thirtieth epoch. In the code, see the added if statements at the end of the Training Loop section of the RNN model; note that these numbers were chosen by the user to represent the different training stages.

(i) Roughly beginning

> Dogs like best to the mate a poon and the Cat a Fox a Found and the Cat a Fox a Found and the Cat a Fox a Found and the Cat a Fox a Found and the

(ii) Middle

> Dogs like best to me thought of so all the requirements of the lake, and the Lion, askit and the Wolf took up an as to be said to the tent has the

(iii) End

> Dogs like best to playe the trees. He Fox, who was even more proud of his time that there was a rear had beans of farase of carrse version of the

2.3.

After having trained the model, greedy decoding war run using different prompts and 512 decoding steps.

(i) A title of a fable which exists in the book.

> MERCURY AND THE WOODMAN
>
> A DOG once said the Fox, “but I notice that can be copyright holder. Asthing to do wotthe Craveler one with the permissions of replacement or refused the Eagle, “I am a bird,” he said, “and then our head off clustress so eter so stat the indigned righing the nex bong and solf,” said the other Frog; “you are bearly all the holf came and set the dilfertaken at the bogs were laugh, he dived and greated from syoug.
>
> “Oh!” said the Fowler. “Wichel spothed himself the don’t on the doground, they c

(ii) A title which you invent, which is not in the book, but similar in the style.

> THE WOLF AND THE LION’S SKIN
>
> A DONKEY once put on a Lion’s skin which some hunters including her nests in the accounts which has the other tree, terror and abution of stoud knew to the dogs. “Ohe had a stood up and said, “Sirs, who lived in the other tree, terror any parts electronic works in the ground.
>
> “Onithing his oun rountry
> house and set a while, and the Cate 249

The Cock and the Stork 108
The Sick in the water.

“Lattin

(iii) Some texts in a similar style.

one day a cat was one of the most famous of Greek sculptors. The farmer, and laid him near the food, and not much water, and nothing to disturbed of corned the Farmer, who was eater, broke it with a looking who had age hard and said blew at was only time. It was again it cause.

“I hear the second coms, and never outro, the fable is not a Fox but he good surper.”

THE HARES AND THE RAMEL

IS LAIN replied very gravel,” said she; as she said, “I will give you some way, and I will be wher found in comfort.

THE CACKAN TH

(iv) Anything you think might be interesting.

the team that will lift the football world cup will be greatly legged hard for his loud.

“Hon’t kill me,” said the Cack, and the Cat from her hound of her paids.

THE HARE AND THE PEASEL

A GRAB too 50

The Eagle and the Fox 90
The Eagle and the Fox 90
The Eagle and the Fox 90
The Eagle and the Fox 90
The Eagle and the Fox 90
The Eagle and the Fox 90
The Eagle and the Fox 90
The Eagle and the Fox 90
The Eagle and the Fox 90
The Eagle and the Fox 90
T

The model is able to create some reasonable phrases but not able to generate a meaningful short text. Notice, that the text generated immediately after the prompt usually fits well with the given input, while the last phrases of the generated text are generally unrelated to the user’s prompt. Moreover, note also that usually the model takes blocks of text (two or three words) from the fables.txt and then combines them to create new texts.

3. Extending the Initial Code

3.1.

In order to train and evaluate an LSTM language model, some changes have been applied to the RNN helper code.

Changes in the model pipeline
First the **cell_state** parameter was added to the LSTM object. Then **nn.RNN** was substituted with **nn.LSTM** while keeping the same inputs. Additionally, the **cell_state** initializer was added to the LSTMModel.

Changes in the decoding pipeline
In the decoding pipeline, **states** is modified to be a tuple with components the initialized hidden and cell states.

Changes in the training pipeline
A line to initialize the **cell_state** was added. Moreover, detach is applied to both hidden and cell states in the LSTM model.

3.2.

Then the LSTM model was trained with the same same hyper-parameters as above, except for the learning rate that was set to 0.001. The best training perplexity achieved in this case was 1.0293268390085506 in 4 minutes and 31 seconds. Furthermore, in Figure 2 the evolution of perplexity over number of epochs is visualized.

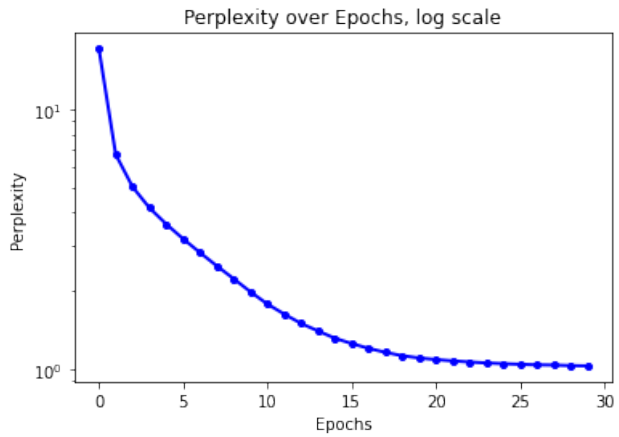

Figure 2: Perplexity over epochs, LSTM

3.3.

The sampling option was added to the code in order to allow sampling of tokens' indices according to the model's output distribution. To do this, **torch.multinominal** was used, specifying to sample according to probs and to sample only one element. Like before, the index ix is then used to extract its corresponding token.

3.4.

(i) A title of a fable which exists in the book

Greedy

THE THREE TRADESMEN

THERE was once a Dog who used to run at every one whom he met to the lake, they were delighted.

"Ah!" they said, "see how grand he looks! How he strides along! How he throws back his head! This is a King indeed. He shall rule over us," and they went joyfully to meet him.

As their new King came nearer, he paused, stretched out his long neck, picked up the head Frog, and swallowed him at one mouthful. And then the next—and the next!

"What is true," said the Pine; "but after the many winters and summers t

Sampling

THE THREE TRADESMEN

THE Camelown so hardly sat down upon the bank to the water and was drowned.

The Dog awoke in a great rage at being disposition, they gradually ventured to approach him with more familiarity, till at length they conceived for him the utmost contempt.

Dissatisfied with this statter and thus make an end of their troubles. A shoal of frogs were sitting around the edge of the pand, and the real danger in his master's lap whenever he pleased, feeding from his hand could possibly to be an egg and swallowed at

(ii) A title which you invent, which is not in the book, but similar in the style.

Greedy

THE SMART DOG

A WOLF saw a Goat feeding at the top of a steep precipice, where he could not reach her.

"My dear friend," said the Wolf, "be careful! I am afraid you will fall and break your neck. Do come down to the meadow, where the grass is fresh and green."

"Are you very hungry?" said the Goat. "And is it your dinner time? And would you like to eat me? I think I will not go down to the meadow to-day, thank you."

And she capered contentedly about on the edge of the rock, as safe from falling as she was from the gr

Sampling

THE SMART DOG

A WOLF saw a Lion—I know not where or in which they were believed.

"What!" said the Father, "see if you can break this bundle of sticks."

Each of the Sons in turn took the bundle and tried with all his might to break it, but could not. When all had tried and given it up, the Father said, "Untie the bundle, and each of you take a double all the beautiful branches, and the Tree fell into the water and whatever he might find by the wayside. He was made a drudge too, and was often forced to carry loads my f

Note that in the greedy version of THE SMART DOG, the model has just copied THE WOLF AND THE GOAT. On the contrary, with sampling the model creates new fables by combining

prahses and words from fables.txt. Therefore, sampling is a good idea.

3.5.

In this section, other types of texts were used to train our LSTM model. Two different types of texts were considered; their output is reported in subsection 3.5.1 and 3.5.2 respectively.

3.5.1.

In this subsection the jimi-hendrix.txt file, taken from Kaggle was used to train our LSTM language model. Hereafter are reported four prompts and the corresponding text output generated by the model using greedy decoding and 512 decoding steps.

(i) There is he going to ?
He's going to make contact
With the livin' bass, and um from
Nashville Tennessee that he flashes trophies of war,
And ribbons of euphoria
Or feel the last moment ever on dry land
Our machine it has done it's word on no more
But then all the other witches on the scene
She got a bull of cheer
Oh yeah There goes Ezy
Ezy Ryder
Ridin down the highway of desire
From far away as Jupiter's sulphur mines
(Way down by the Methane Seah...
So let them go to the moon.
Yes let them nibble on the bone.
So in

(ii) 4th of December
You've got me floatin', nat'rally
Got me floatin', nat'rally
You've got me floatin', never down
Got me floatin', yeah
You got my be,
Whe don't want to see you again
I was so alone
All by my self
I just couldn't say what went they that's what I need now
Freedom to live and breathe undrer water
Foams to make steamed
Bonever been here 3 times and
The 4th the same as him
(Gets very lonely up this road, baby
A woman here a woman there try to kill...
That's my duty.
So please don't try to hold me back
Stone free

(iii) Why is goin' through them changes
I think I'm goin' out of my mind
Every time you know it's too bad, little girl,
it's too bad
Too bad we have to part (have to part) Dig
Gonna leave this town, yeah
Gonna leave this town
Gonna make a whole lotta money
Gonna be big, yeah
I'm gonna buy this town
I'm gonna buy this town
An' put it all in my shoe
Might even raise just a little sand
'Cause I'm a voodoo child
Lord knows I'm a voodoo child I didn't mean to take you up all your sweet time
I'll give it right back to you one

(iv) Where are you going to go?
Well, dig it [Verse 5]
I'm going way down south
Way down where I can be free
There's no one going to find me [Verse 6]l hell I please
Stone free to do what I please
Stone free I got to I got to get away
Stone free It's alright, she asked me
"Yes, take me along right now" Hey baby, can I step into your world for a while?
"Yes you can," she say
"do what you please"
Gotta get the brothers together
and the right to be free
In a cloud of angel dust
I think I see me a freak
Hey motorcycle mama
you gonna ma

3.5.2.

In this subsection the model was applied to the moes_tavern_lines.txt file, taken from shivaverma, in order to generate a new dialog between the Simpsons' characters at Moe's tavern. Again, greedy deconding and 512 decoding steps were used.

Lenny_Leonard: Hey Moe how are you doing?
Lenny_Leonard: Homer, it's just a book.
Homer: (SPITTING OUT BEER) One hundred bucks fan. You need to focus here again.
Moe_Szyslak: (STUNNED) Oh no, he stepped on the bow!
Moe_Szyslak: (SADLY) Yeah, poor Stevie. They never found his hair... (STRAPS HEAD) save some lousy kid!
Dolph's_Dad: This is for the fridge. On man, she loves America crienke." Is that no one enve order workers.
Homer_Simpson: (GASPS) She's gone!

Moe_Szyslak: (SAD) And here you go, pal. (SCANNING HIMSELF) That dog has a moon and

In the Jimi Hendrix lyrics generation, the texts are quite reasonable. For the dialog example, the phrases taken alone make sense but when put together the whole dialog does not sound too good.

4. Questions

4.1.

The perplexity of a language model which always predicts each character with equal probability of $\frac{1}{V}$ is V.

4.2.

It is an issue because there will be no update through gradient descent, since $\nabla_{Loss}(W)$ shrinks to a small number.