



DEEP LEARNING LAB

Kaan Egemen Şen
Assignment 3

YEAR 2022-2023

1 Preliminaries and Reading Comprehension

1.1 Text Data

1. Modified vocabulary class to get those text Data 1 :

```
Reading text file from: /kaggle/working/49010-0.txt
Number of lines 5033
Number of character 177517
Capital Letters number 9666
Lower Letters number 122224
Number of Unique Character 107
```

Figure 1: Data from text file

2. If i could do some text preprocessing i would remove the punctuations and stop words, convert all characters to lowercase and use lemmatization.

1.2 Dataloader/Batch Construction

1. The method `get_idx` is storing the tokens into a dictionary, if the string is part of our vocabulary function will return the Id or index of the token. If it is not part of the vocabulary and the value of `extend_vocab` equal to "TRUE" then it will add the token to the vocabulary.
2. In dictionary "id_to_string" the key is the id associated when it was added and the value is our character or token.
Second dictionary "string_to_id" the key is the character or token and the value corresponds to the id associated when it was added to the dictionary.
3. When you call `len` in `TextData` you get number of characters of a string.
4. When you call `len` in `DataBatches` you get length of batch list
5. Input data is an object that comes from the class `TextData` and the `create_batch` turns the object with a tensor that contains the value of the padding with batch size times segment length.
The `padded[:text_len] = input_data.data` part will replace all values up to the index `text_len` with the information from `input_data.data`.
So the "padded" variable in the first line will be filled with value of our padding id in this case is 0. For second line of "padded" it will contain the values of `input_data.data`
6. The tensor `padded[i * bptt_len:(i + 1) * bptt_len]` when `i==0` has the shape (64,32) where 32 corresponds to the input bzs and 64 corresponds with our input `bptt_len`
7. The tensor `padded[i * bptt_len - 1:(i + 1) * bptt_len]` in else branch has the shape (65,32) except for the last batch which in the case of this text has shape (45,32)

1.3 Modeling, Training, and Decoding

1. We apply `detach` in hidden state of RNN because when calculating truncated backpropagation through time we don't want to end up with vanishing gradient problem. If we don't detach, then the gradient will be really big.
2. We use `ignore_index=0` in `CrossEntropyLoss` to not contribute our initial index into input gradient
3. Input shape expected by Rnn model is (N,B,D) where N is length B is batch size and D is the character embedding size

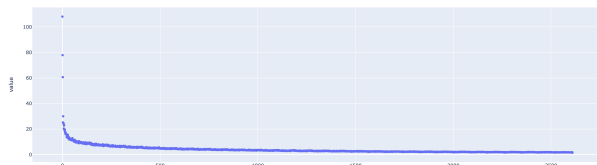
4. Output shape of model is (N,B,2*H) 2 is because Rnn is bidirectional and also each hidden is (2*L,H_{old})
5. We use "Dogs like best to" in training because we want to observe our model progress while training.

2 Running Experiments Using the Initial Code

1. Modified code to get perplexity instead of loss. Available in code.
2. Trained the model with given Hyper-parameters received text at epoch 5,15,28 and perplexity plot below 2a 2b: For get both outputs we modified training part. Available in code.

```
the could a said the Fox a Can a good and the Hown and the Hown and the Hown and the Hown and the Hown and the Hown', '
make a compassion that you said others, and then a man said, "that companions, and as he liked in a
stone and states and get ', '
in a sever to come and help him out of his tried and the Camel\
```

(a) Model Text Predictions



(b) Model Perplexity Plot

3. we use our model to predict following texts:
 - (a) A title of a fable which exists in the book 3.

```
A DONKEY, have dest
of his trid; and tre the little Mar and the Trees
```

Figure 3: THE DONKEY IN THE LION'S SKIN

- (b) A title which you invent, which is not in the book, but similar in the style 4.

```
, THE FOW

A BAT who had no through a miry ran the early scape beact and to drink from his heavy came the proud a
for his should be the chaiced that he had not to come and help him
out of his trick.

"You must comply with an the water runnted his hand the Frog.

"Not such a prize, and the Fawn and the Crow

The Shepherd stood up on his tricks, and the reapers were subject to the
tree, and the Eagle had her nest, think the Mat and the Fo
```

Figure 4: THE DOG IN THE LONE HOUSE

- (c) Some texts in a similar style. 5.

```
the mornen procest in the world. Let others chars the most came the tried and the Fox ser caught in a
trap the represent them a Crow.
```

Figure 5: The monkey eat

- (d) Anything you think might be interesting. 6.

In general this model mostly meaningful but i think not so good to produce novel texts he has too much vocabulary faults to create new thing

```

pooked upon her home. But as he could and cold for a provine."

The Fowler tree for him as hear of heary at
even my good," said the Fowler. "Wicked in a house."

The Fox so much of such a veighbor," said her young one can only beautiful than the other lived and
careful in leapand, and so well heary your veally and ede to cuar one mistaken for the edge of the
works, who were deart, and at all the years and not in sight."

"When the laws of the country road
proud of his trouble.

"Lazy fellow," said Bersuse

```

Figure 6: Another Bites to

3 Extending the Initial Code

1. Modified Rnn model to turn into LSTM model. Added two state (cell and hidden state). Modifications available in code.
2. After training achieved below 1.03 perplexity can be seen below 7 :

```

THE LIONESS

A GREAT stir
----- end generated text -----
=== start epoch 26 ===
train ppl: 1.1591145992279053
----- epoch/batch 26/0 -----
Dogs like best to
eat it, for we are
hungry and tired."

"Oh, go away," growled the Dog, "and let me sleep."

"What an ugly, snappish fellow!" sai
----- end generated text -----
train ppl: 1.0443191528320312
----- epoch/batch 26/30 -----
Dogs like best to
eat it, but could not. When all had tried and given it up, the
Father said, "Untie the bundle, and each of you take a stick and
----- end generated text -----
train ppl: 1.045936942100525
----- epoch/batch 26/60 -----
Dogs like best to
eat it, and he groaned
aloud.

The Frogs heard his groans but showed no pity. "What a foolish fellow,"
said they, "to make such
----- end generated text -----
final perplexiity : 1.0274878

final Epoch : 26

```

Figure 7: LSTM Model Training

3. modified decoding step of lstm with including sample randomly. Available in code.
4. using decoding for greedy and sampling can be seen below :
 - (a) A title of a fable which exists in the book 8a 8b.
 - (b) A title which you invent, which is not in the book, but similar in the style. 9a 9b.

I think sampling is not a good idea because often the results are too diverse, they look too random.

```

jumped down without more ado. Upon this the
crafty Fox jumped upon her horns and nimbly leaped out, remarking to
the deluded Goat, "If you had brains to match your beard, you would
have looked before you leaped."

THE WOMAN AND HER HEN

A THIRTY Woman kept a Hen that could be depended on to lay an egg
every morning. The watchful Woman thought within herself, "If I were to
double my Hen's allowance of barley, she would lay twice a day." So she tried her plan, and the Hen
became so fat and sleek that she

```

(a) The Fox and the Goat with greedy decoding

```

hers should never catch the
Thile he set a child to guide him.
"My stare agging to have been by burned than I am."

"Oy from the Mouse would not let me share my life."

As the in the field, caught up a blazing stick of wood, and ran with
it in her mouth to the pine tree, playing against the edge of the wool, and came and
her nest time the Fox with his Mouse. And it came about that his load was
so take an account of the conversants, and among the evely.

He replied: "We would not know better friends under su

```

(b) The Fox and the Goat with random sampling

```

of the mountain.

The Woodman cut down the Tree, and fitted the handle to his ax; then to
the dismay of the Trees, set to work and, with strong strokes, quickly
felled all the noblest giants of the forest.

Lamenting too late the fate of his companions, an old Oak remarked to a
neighboring Cedar:

"The first step has lost us all. If we had not so willingly given up
the rights of the Ash, we might have stood for ages."

THE MILKMAID AND HER PAIL OF MILK
DOLLY the Milkmaid having been a good girl and ca

```

(a) The Serpent and the Cow with greedy decoding

```

that one who mander in the gold made
so great a King. ANC Cock, who Stopped to
say that had eaten to the fole of his days.

He stret on means it make the acquaintance of the forest.

Lamenting too late the fate of his companions, an Alther, and at times post by one into the prouded
States. Compliance requirements are not unour parts
with all the force of cheese, which was exactly what was expected.

The Fox ate it at one mouthful, then stopped to say, "Thank you, madam.
I am quite satisfied.

```

(b) The Serpent and the Cow with random sampling

5. BONUS (Be Creative) Trained LSTM Model with our 2.assignment available at code and output prompts are below :

(a) input prompt : Print(10.

```

labels)
    drop_epoch_valloss.append(drop_val_loss.item())
    drop_val_losses.append(sum(drop_epoch_valloss.append(drop_val_loss.item())
    drop_val_losses.append(sum(drop_epoch_valloss)/len(drop_epoch_valloss)/
len(drop_epoch_valloss)/len(drop_epoch_valloss)/len(drop_epoch_valloss)/len(drop_epoch_valloss)/
len(drop_epoch_valloss)/len(drop_epoch_valloss)/len(drop_epoch_valloss)/len(drop_epoch_valloss)/
len(drop_epoch_valloss)/len(drop_epoch_valloss)/len(drop_epoch_valloss)/len(drop_

```

Figure 10: Output prompt

(b) Input Prompt : If 11.

```

train_acc.append(100*(running_correct / running_total)
drop_train_losses.append(running_loss/run_step)

running_correct = 0
run_step = 0

```

Figure 11: Output Prompt

(c) Input Prompt : while 12.

```

(0)
with torch.no_grad():
    _, predicted = outputs.max(1)

#Validation Loss
drop_val_loss=loss_fn(outputs,labels)
epoch_val_loss.append(val_loss.item())
drop_val_losses.append(sum(drop_epoch_valloss.append(drop_val_loss.item())
drop_val_losses.append(sum(drop_epoch_valloss.append(drop_val_loss.item())
drop_val_losses.append(sum(drop_epoch_valloss)/len(drop_epoch_valloss)/
len(drop_epoch_valloss)/len(

```

Figure 12: Output Prompt

(d) Input Prompt : for i in range 13.

```
(4,448,4):
  imp_resized_losses.append(imp_val_losses[i]+imp_val_losses[i+1]+imp_val_losses[i+3])
  imp_resized_losses.append(imp_val_losses[i]+imp_val_losses[i+1]+imp_val_losses[i+3])
  imp_resized_losses.append(imp_val_losses[i]+imp_val_losses[i+1]+imp_val_losses[i+3])
  imp_resized_losses.append(imp_val_losses[i]+imp_val_losses[i+1]+imp_val_losses[i+3])
  imp_resized_losses.append(imp_val_losses[i]+imp_val_losses[i+1]+imp_val_losses[i+3])
  imp_resized_losses.append(imp_val_losses[i]+imp_val_l
```

Figure 13: Output Prompt

4 Question

1. V is the perplexity
2. it's an issue because gradients results in large updates to the model weights during the training of the model. So it difficults the training