Project3

Language Model

CISC 867 Deep Learning W22

Farah Samy 21FSAE
Nourhan Magdy 21NMMA1
Sondos MOSTAFA 21SMAM1

Contents

Project description

The objective of the project is to build a language model that can predict the probability of the next word in the sequence, using the words that have been already observed in the sequence.

We will use neural network models as they are a preferred method for developing statistical language models as they can use a distributed representation where different words with similar meanings have similar representation and because they can use a large context of recently observed words when making predictions.

In this case, we will need to prepare text for developing a word-based language model, design and fit a neural language model with a learned embedding and a recurrent hidden layer, and then use the learned language model to generate new text with similar statistical properties as the source text.

Dataset

You will use The Republic by Plato as the source text. This version is cleaned (no hyphens or punctuations, removed nonalphabetic words, and all words in lowercase) and it can be downloaded from https://www.gutenberg.org/cache/epub/1497/pg1497.txt.

```
The Project Gutenberg eBook of The Republic, by Plato

This eBook is for the use of anyone anywhere in the United States and
most other parts of the world at no cost and with almost no restrictions
whatsoever. You may copy it, give it away or re-use it under the terms
of the Project Gutenberg License included with this eBook or online at
www.gutenberg.org. If you are not located in the United States, you
will have to check the laws of the country where you are located before
using this eBook.

Title: The Republic

Author: Plato

Translator: B. Jowett

Release Date: October, 1998 [eBook #1497]
[Most recently updated: September 11, 2021]

Language: English


Produced by: Sue Asscher and David Widger

*** START OF THE PROJECT GUTENBERG EBOOK THE REPUBLIC ***



THE REPUBLIC

By Plato

Trans
```

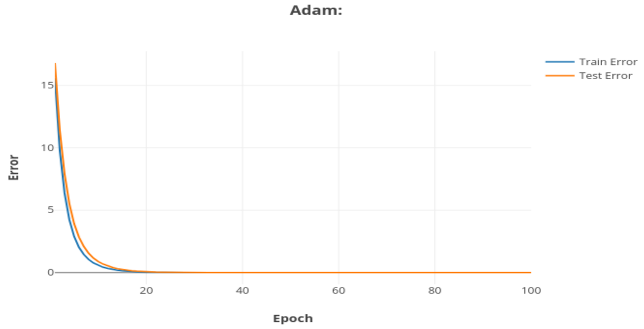
Steps:

1. Reading The Republic by Plato file
2. Data Preparation
3. Encoding
4. Train Language Model
5. Use Language Model

Optimizer

Adam:

Adam optimization is a type of stochastic gradient descent method that is based on adaptive estimation of first order and second-order moments. The method is really efficient when working with large problem involving a lot of data or parameters. It requires less memory and efficient.



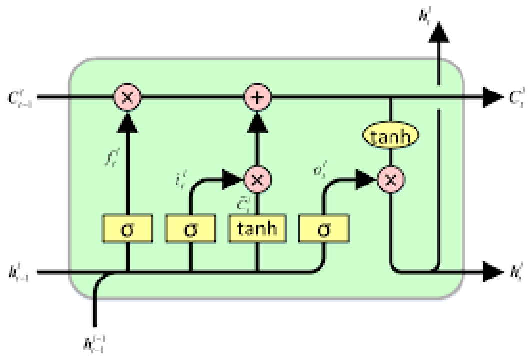
Architectures used:

- LSTM
- GRU

LSTM

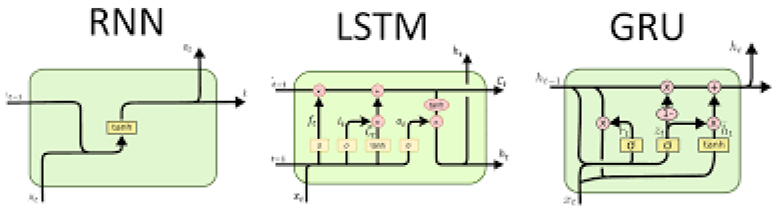
Long Short-Term Memory (LSTM) networks are a type of recurrent neural network which is capable of learning order dependence in sequence prediction problems, which is beneficial and required in complex problem domains like machine translation, speech recognition, and more.

LSTMs are a complex area of deep learning. It could be hard to get your hands around what LSTMs are, and how terms like bidirectional and sequence-to-sequence relate to the field



GRU

GRUs are very similar to Long Short-Term Memory. As in LSTM, GRU uses gates to control the flow of information. They are relatively new as compared to LSTM. This is the reason they offer some improvement over LSTM and have simpler architecture. GRU differ than LSTM, it does not have a separate cell. It only has a hidden state). As a result of the simplicity of the architecture, GRUs are faster to train.



Our models

LSTM

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 50, 100)           734000

 lstm (LSTM)                 (None, 50, 100)           80400

 lstm_1 (LSTM)               (None, 50, 100)           80400

 lstm_2 (LSTM)               (None, 80)                57920

 dense (Dense)               (None, 300)               24300

 dropout (Dropout)           (None, 300)               0

 dense_1 (Dense)             (None, 7340)              2209340

=================================================================
Total params: 3,186,360
Trainable params: 3,186,360
Non-trainable params: 0
_________________________________________________________________
```

Accuracy

```
Epoch 150/150
18/18 [==============================] - 3s 179ms/step - loss: 0.2950 - accuracy: 0.9166
```

Output

```
like a god maintains justice to whom the black earth brings forth wheat and barley whose trees are bowed with fruit and his she
ep never fail to bear and the sea gives him still grander are the gifts of heaven which musaeus and his son vouchsafe to the ju
st they take

same only mistaken of fain not refrain to enemies not yourself nearly business unjust for sort he together to wish which there
be are more the other instrument although but his state but the up for worthless the other falls but he nay the defence which w
hat if what if
```

GRU

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_1 (Embedding)     (None, 50, 100)           734000

 gru (GRU)                   (None, 50, 100)           60600

 gru_1 (GRU)                 (None, 50, 100)           60600

 gru_2 (GRU)                 (None, 80)                43680

 dense_2 (Dense)             (None, 300)               24300

 dropout_1 (Dropout)         (None, 300)               0

 dense_3 (Dense)             (None, 7340)              2209340

=================================================================
Total params: 3,132,520
Trainable params: 3,132,520
Non-trainable params: 0
```

Accuracy

```
Epoch 150/150
18/18 [==============================] - 3s 150ms/step - loss: 0.0482 - accuracy: 0.9883
```

Output

```
take take image anybody anybody trusted trusted count count could count count count count distress distress distress accordingl
y geometrician distress distress distress geometrician pursued pursued accordingly geometrician note note note note make explai
n arrive pantomimic tyrant tyrant lusts tyrant dead perverse rules dead dead perverse question dead perverse elevating elevatin
g
```

Conclusion

We can observe that GRU performance is better than LSTM and consumes less runtime. GRU doesn't use many trainings parameter and therefore it consumes less memory and executes faster than LSTM whereas LSTM is more accurate on a larger dataset. One can choose LSTM if you are dealing with large sequences and accuracy is concerned, GRU is used when you have less memory consumption and want faster results.

References

https://machinelearningmastery.com/how-to-develop-a-word-level-neural-language-model-in-keras/#:~:text=Language%20Model%20Design&text=The%20language%20model%20will%20be,the%20input%20sequences%20should%20be.

https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/

https://analyticsindiamag.com/lstm-vs-gru-in-recurrent-neural-network-a-comparative-study/

https://www.analyticsvidhya.com/blog/2021/03/introduction-to-gated-recurrent-unit-gru/

https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be

https://www.kaggle.com/code/ashrafkhan94/the-republic-by-plato-nl-model-text-generation