UNIVERSITY OF PADUA, DEPARTMENT OF INFORMATION ENGINEERING

- Neural Network and Deep Learning -

Sequence modeling with RNN

Giulia Bressan, ID 1206752

A.Y. 2019/2020

1 INTRODUCTION

The assignment of the third homework was to train a RNN that generates a text, based on the book that is given as dataset. The goal was to find the right structure and hyperparameters of the network, in order to avoid overfitting with the original dataset and generate a text with the highest quality as possible.
The work is structured as follows:

1. Loading and preprocessing of the selected dataset,
2. Tuning of the parameters of the RNN using k-fold cross validation with grid search,
3. Training of the RNN with the entire text,
4. Generation of the text from the trained model and qualitative evaluation of it.

The dataset used for the homework is taken from *"Romeo and Juliet"* by William Shakespeare, but I also tested the training with an other dataset and the results are briefly shown at the end of the report.

2 TRAINING DATASET

Before using the dataset for the training of the network, it was necessary to make some changes to simplify the text and the characters given to the RNN.
As a preprocessing phase, the following procedures were implemented:

- Removing some punctuation and rare symbols from the set of characters
- Converting upper case letters in lower case letters
- Removing any double new line
- Extracting paragraphs

The resulting alphabet length was of 33 characters, composed by all the letters of the English alphabet, the following punctuation symbols: "!", ",", ".", "?"; the encoding characters for the space (" ") and the new line ("%n").
All the letters were encoded, with the following number assignment: "%n": 0, " ": 1, "!": 2, "'": 3, ",": 4, ".": 5, "?": 6, "a": 7, "b": 8, "c": 9, "d": 10, "e": 11, "f": 12, "g": 13, "h": 14, "i": 15, "j": 16, "k": 17, "l": 18, "m": 19, "n": 20, "o": 21, "p": 22, "q": 23, "r": 24, "s": 25, "t": 26, "u": 27, "v": 28, "w": 29, "x": 30, "y": 31, "z": 32.
Then, this encoding was transformed into a one-hot-encoding, assigning to each number a vector composed by all zeros, except for the indices corresponding to the number assigned to the letter.
Before feeding to the network, the text samples were randomly cropped and transformed into tensors.

3 PARAMETERS TUNING

The network used was the one proposed during the lab session, as it was the one that performed better. Even if I tried other possible structures and types of layers, such as changing the number of RNN stacked layers or using a Gated Recurrent Unit (GRU), but the results were not satisfying. Therefore, the nework is composed of a Long Short-Term Memory (LSTM) architecture with 2 stacked layers and a dropout probability of 0.3, followed by a linear layer as output layer.
To tune the parameters of the network I used a 3-folds cross validation with grid search. The parameters that I tested were the learning rate, the number of input letters, the batch size and the number of the RNN hidden units, with the values shown in Table 3.1.

Parameters	**Possible values**
Learning Rate	0.01 - 0.001
Number of input letters	50 - 100 - 200 neurons
Batch size	50 - 500 - 1000
Number of the RNN hidden units	64 - 128 - 256

Table 3.1: Set of hyperparameters used during the cross-validation with grid search.

The optimal parameters found are shown in Table 3.2.

Parameters	**Optimal value**
Learning Rate	0.001
Number of input letters	200 neurons
Batch size	500
Number of the RNN hidden units	256

Table 3.2: Optimal set of hyperparameters found.

To better refine the search of the optimal parameters, I tried also to increase the values of the parameters that were chosen as the maximum value among the possible ones of the grid (i.e.

number of input letters and number of the RNN hidden units), training an other grid search just with that parameters. The final results did not change, so I kept the already shown optimal configuration of hyperparameters. The parameters that were kept fixed were the dropout probability of 0.3 and the number of epochs, that in the cross-validation was fixed at 5000 epochs.

4 TRAINING AND TEXT GENERATION

The final model was trained with the optimal set of parameters found.
I implemented an early stopping mechanism, that stopped the training when a new minimum value of the loss was not found in 300 epochs (higher values lead to too much overfitting in the generated text). Every time a new minimum loss is found, if smaller than 1, the network parameters are saved and a counter of the epochs starts. If the next loss is bigger than the minimum the counter increases, until it reaches 300, when the training stops.
In particular, the network reached the final minimum after 5276 epochs, with a loss of 0.56985, stopping the learning at epoch 5601.
The parameters are also saved every 1000 epochs, in order to be able to see the progression of the training in the generated text.
To generate the text, the network outputs the next letter starting from an initial phrase given as seed. The network gives a vector containing the probabilities of each character to be the next one. In order to improve the quality of the generation, the next one is sampled from a *softmax* distribution of the probability vector.
A "*temperature*" parameter is used to boost or inhibit some choices, in a way in which the most probable letters have less probability of being chosen if the temperature is higher.
The predicted characters produced in this way perform much better in the text generation rather than just taking the character with the highest probability.

5 RESULTS

The following text is generated with the trained model, with a temperature value of 0.5, starting from the seed *"o, romeo, romeo!"*:

> *"o, romeo, romeo!*
> *juliet wills didling and not sone*
> *as i bade thou colfating porson shull be to the cillows see thou cat'st thou wilt ferlows*
> *in be night.*
> *betor i bearty on the sear. out enther paris, too, and i'll to my rest.*
> *exeunt all but juliet and nurse. give me those my steed.*
> *i worl that the lear my sayses to bed.*
> *ah, sirrah the dear of sounts as hast to beat the rat come, what i meat up our pipes*
> *and be gone.*
> *which the carpure of an unem's and weech and clilt to my ledd*
> *but i mat the pirch a maire be a tit"*

Changing the temperature to 0.3 and maintaining the same seed we obtain the following result:

> *o, romeo, romeo!*
> *o, polise death thou as on the ground, and to the counter of assain*
> *that the counter of assailing eyes,*
> *nor ope her lap to saintsed that the curtump'd the from thise pare of section'd wretess*
> *married to my speet.*

what is it else? hath the intect a fair of sounts as hasty powder fir'd with the hears
and where the inmate.
that is it likely that came to the livers treeps,
therefore the intither to my sece.
but he with a gerten love, and tell the county paris. therefore will to kind my head
off in ma

In the same way, with a temperature of 0.1, we obtain:

o, romeo, romeo!
o, pole as the dery it which i may call the were
an oor her could and be to the churchyard tread be to the cilt up the circumstance.
let me be satisfied, is the counter of assailing eyes,
nor ope her lap to saintseducing gold.
o, now i could me to thy mistress that, then mistrest thou capelets something doth
shall be to the counter of assailing eyes,
nor ope her lap to saintseducing gold.
o, now i could me to thy mistress that, then mistrest thou cates with the will a rose.
but heaven should be t

To check if there was overfitting I compared the generated text with the original one, to see if the sentences were copied. The text was never entirely copied, just some short sequence of words were.
Increasing the temperature the overfitting decreases, since it happens that less probable characters may be chosen, but also the quality of the generated text worsens, as the style is less similar to the one of the author.
Some words are misspelled (underlined in the text samples) and this happens more with an higher temperature. With the temperature of 0.1 we have the text with less errors and unknown words but we can see that some sentences are repeated (such as the 5th and the 7th row), with just the final words changed.
It is interesting to analyse also the intermediate models saved, to see the progression of the training of the RNN. For example, this is the result after 1000 epochs of training, with a temperature of 0.5:

o, romeo, romeo!
and thou the gourser and the stord the dare thou dome to the kist to me the the fore
op to him and thit to dint that not, and the ganle a doult meare and and shaciny the
the seres and, you doad the frourd, and there and this to the thit i be thou de with this
would these and beend so the burtt the be the thour the mertering mesering pureon
and a so with and and to ding to the mereare on the stould thy weals fore hin the
deart all me to fire the preith and thit my hin thee hear,
and that sate
and

We can see that the set of words is not very wide, with the majority of them being prepositions, articles or conjunctions and also with lot of errors. Also the spacing is not as we would expect from the original text: the network has not learned where to start a new line yet.
With 3000 epochs we can see some improvements:

o, romeo, romeo!
dow thou not so the churchanger of then aponters.
therefore leve must in allplaight to stay the churthat, and the lave to the chance of
beet

that i say the ence,
as the strek'd with unterres carust to she the growe of then it not be that chance and wellows must curseay.
an engoob of the charr'n of a fair,
when that the rearone of these the sear that he counds who bad a within the partions as to death
that her be that stay that the grope and here and the stay
the cromenters.
i carnon and a bady m

The range of words generated has widened, but still the spacing is not optimal, even if it improved with respect to the previous attempt.
This is learned only at the end of the training, as we can see from the text generated after 5000 epochs of training:

o, romeo, romeo!
o, good night! i'll to my wedding bed
bream to sleep it sleep romeo, being oner formser, nor for the wands,
farith'd not to the fere,
of illt the secret ne sone as weet in may
but a hitwle me,
and in thou not, i'll tell thee romeo, then me,
but i lot limens cautien of thee love,
but heaven sould loves and rouses and when i do word
and fail is a bead, and that they is in the secret night.
she store and was the dead, and then meanst too,
and lie, i'll to the cruad thy and with the shard be die
that

Here the spacing is much more similar to the one of the original text, with mostly short paragraphs. Some errors remain, but the classical Shakespearian style is already visible.

6 OTHER DATASET

With the same network structure and training mechanism I also tried the training with an other dataset, to see the differences in the structure of the generated text.
The following text is generated from the training of the network with "A Christmas Carol" by Charles Dickens, with a temperature of 0.1 and the sentence "there is no doubt whatever about that." as seed:

there is no doubt whatever about that. not the spirit starlly to place. as it was all the samile and scrooge sat down upon its head and the spirit streets of an emuction, and the streets were stailing out, and sole might have been to with his ferret eyes, when the chimes of a neighbouring christmas tors
compon christmas tore. where the spirit standing in the spirit starly too with him, he said of common comforts, sirting the most had seen them and his life and sometimes marley,
but he answered to with a stange, with straggling of a be

The result reflects the structure of the original text and retrieves some typical words and names. There is little overfitting in general, as just one part of a sentence is copied from the original book (*"with his ferret eyes, when the chimes of a neighbouring"*).