Homework 3 - Recurrent Neural Network

Alberto Zancanaro - ID: 1211199

I. Introduction

This homework was focused around the creation and testing of a Recurrent Neural Network (RNN) and evaluate its results in the creation of text. We will introduce the work, explain the code implementation and the problem encounter. In the end we will show the results.

RNN are particular type of neural network developed to analyze data were a strong temporal correlation is present between the various samples like text or video.

II. Problem Structure

The RNN network have to analyze a text and learn how to produce similar text. More precisely the trained network have to create a sequence of n character given an input word (or character).

The work was initally focused on reducing the complexity of database and subsequently on code implementation and finding the correct parameter for the network. After that the network was tested.

The network was trained on *Divina Commedia* of *Dante Alighieri* and *Gerusalemme Liberata* of *Torquato Tasso*.

III. Code Implementation

A. Pre-Processing

Since the training with RNN is usually very long we first simplify the dataset the following action:

- Removing accented vowels and replace them with the respective vowels.
- Lower case transformation to avoid duplication of symbols.
- Removing number.
- Removing unsupported characters[1].

After perform this procedure we obtain we following alphabets for the two dataset:

- *Divina Commedia*: [’\n’, ’ ’, ’!’, ”’”, ’”’, ’(’, ’)’, ’,’, ’-’, ’.’, ’:’, ’;’, ’<’, ’>’, ’?’, ’a’, ’b’, ’c’, ’d’, ’e’, ’f’, ’g’, ’h’, ’i’, ’j’, ’l’, ’m’, ’n’, ’o’, ’p’, ’q’, ’r’, ’s’, ’t’, ’u’, ’v’, ’x’, ’y’, ’z’]. Length = 39.
- *Gerusalemme Liberata*: [’\n’, ’ ’, ’!’, ”’”, ’”’, ’(’, ’)’, ’,’, ’-’, ’.’, ’:’, ’;’, ’<’, ’>’, ’?’, ’a’, ’b’, ’c’, ’d’, ’e’, ’f’, ’g’, ’h’, ’i’, ’j’, ’l’, ’m’, ’n’, ’o’, ’p’, ’q’, ’r’, ’s’, ’t’, ’u’, ’v’, ’x’, ’y’, ’z’] . Length = 42.

Once the text was clean we apply a numerical encoding assigning to each letter a number. Subsequently one-hot-encoding was applied to the encoded datasets. In this way the data can be easily handle by RNN through tensor. Example of numerical encoding and one-hot-encoding for the *Divina Commedia* are reported in tabel 1.

Original Character	Numerical Encoding	One-hot-econdign
’\n’	0	[1,0,0...,0,0]
’ ’	1	[0,1,0...,0,0]
’!’	2	[0,0,1...,0,0]
...	...	...
’y’	37	[0,0,0...,1,0]
’z’	38	[0,0,0...,0,1]

TABLE 1: Numerical and One-hot-encoding for *Divina Commedia*

Also, since the RNN must have all the input of the same length, a cropping of the text was performed. For *Divina Commedia* we will use input of 88 character and for *Gerusalemme Liberata* we will use input of 255 character. This number were been chosen because are the lengths of the shortest stanzas[2] in both poem. With this choice we try to preserve the major part of the original text.

B. Network Model

The network was implementd thorugh *PyTorch*, a *Python* framework for machine learning application.

We choose the LSTM as implementation of RNN since is one of the best implementation for this type of networks. The code allow to specify different number of LSTM and slightly different settings were used for the two networks. It also been implemented a more complex model with 4 LSTM block and between each other a small neural network of a single layer with a *Leaky ReLU* activation functions.

C. Dataset Management

The datasets are implemented as extension of the *Dataset* class from *torch.utils.data*. In this way there is an automatic method to read the txt file and trasform them in the encoding form.

Due the particular formatting of the two txt file only one class was needed. More precisely the *Divina Commedia* can be divided into terze or cantos so in the code was implemented the possibility to training the network for both this possibility. The code that divide the *Divina Commedia* into cantos also already divide the *Gerusalemme Liberata* into octave so no additional database class was needed. The encoding scheme is identical for both text and the code handle in automatic the different lengths of the alphabets.

[1] The txt downloaded use a particular version of apostrophe(’) and quotation mark(”) that was not fully supported by standard Python and caused problem with the text reading

[2] P.s. The term stanza is the english for ”strofa”; with this term I am referring to both the octave in the *Gerusalemme Liberata* and the terza in *Divina Commedia*.

Hidden Units	[64, 128, 256]
Number of Layers	[2, 3, 4]
Learning Rate	[0.1, 0.01, 0.001]

TABLE 2: K-Fold Search parameters

Paramaters	*Divina Commedia*	*Gerusalemme Liberata*
Hidden Units	256	277
Number of Layer	2	3
Learning Rate	0.001	0.001
Dropout Probability	0.27	0.27
Epochs	12000	12000

TABLE 3: Training parameters

To feed the dataset into the *PyTorch* model we used the *DataLoader* class, from *torch.utils.data*, that provide useful method to automatic divide the datasets in batch for training and even parallelization of the loading process[3]

D. Training

For tuning the parameters of the network a k-fold cross validation was implemented. The k-fold search through parameters in table 2 where Hidden Units is the number of features in the hidden size and the Number of Layers is the number of LSTM cell stack together. The code implement an automatic division of the dataset in k different fold, with k specified by the user. We set $k = 4$.

Since the training was quite long, especially for the last network, we have chosen to training the network only on 1500 epochs in the k-fold research.

After the k-fold results we decide to use the the parameters in table 3 for the training.

The k-fold was performed only for *Divina Commedia*. Despite the fact that increasing the number of layer the mean loss decrease this decision also greatly increase the training time. Since the training time for this network was very long we have chosen to use a more simple network to evaluate the correctness of the code.

After successfully testing of the network on *Divina Commedia* we have chosen to try to training a complex network for the *Gerusalemme Liberata*. For this reason we have increase both the dimension of the Hidden Units and the number of layers.

The training of the complex model with the 4 block LSTM block and the neural network was not done for time reasons[4]. However the code was tested for 40 epochs and work without problem. With more time probably also this network could give us satisfying results.

IV. Results - *Divina Commedia*

This is the result obtained with the seed *nel mezzo del cammin* thorugh the model trained on *Divina Commedia* for 750 charcaters. The network predict the next character in a one-hot-encoding form and an inverse mapping was used to retrieve the original character.

Since the network return a tensor of real value we implement two strategies to choose the best character. The first select as character the element with the highest value in the tensor obtain from the network. The second way use a *softmax* function with a temperature of 0.2.

In the first box there is the prediction with the first method and in the second the prediction with the softmax function. The second method seems to create better text so for the *Gerusalemme Liberata* we will use the softmax function to generate samples.

Despite the improvement in the prediction with the second method in both case the network fail to generally recreate the rhyme scheme used in Dante's poem. The softmax output in some line succeeds in this, especially if we consider only the final letter, but the results is far beyond from the original.

Also in both case the network fails the use ”<” and ”>” to to start dialogues. A future improvement could be remove this character from the dictionary (in this way we would also speed up a little bit the training process).

```
nel mezzo del cammino,
a cui disior volger di voi e l'altro mal sempro e 'l martalmore e 'l mio due,
quant'el sarte vermo fui chi son quelli
che tu dei a coscien ch'i' fossi a porta,
che non pareva avero,
mi fui mai che trasmuta
se non che perche vuole
a le crudito,
fiat vendetta vedere,
e io sarebbe,
di cui tu ga ben che criciveggia
dal vero erriliso, e tenesti
sottollo>>>
e or dica 'l pieno che tu m'insea' fiati,
e fe parre,
in pianger che tanto perfetto
di tanto mortalli>>.
e io a lui che tu mi tace>>>>.
e io a lui: <<a tutto ad una sentate
che mi parti, se oltre vi luse
qual che tre venir per voto,
sempre con quelle visime
che m'ha tacer, quando poli
di gran valle membraccia,
se mai rapile
a le coscia col duca,
tra ' roscernoni si chiama
se non che vai e l'a
```

[3] Unfortunately this part always cause code interruption, both in lab 3 and 4, so this feature was not used.

[4] The model for *Divina Commedia* took around 9 hours to training and the model for *Gerusalemme Liberata* took around 24 hours. Due to my impossibility to both train the network and performed other operations on my computer I had to decide to only limit myself to two model.

nel mezzo del cammino,
che si movea da l'orribile fretta si chiude, e dio
si fa da me avanza ch'io vidi un poco;
e qui ma fa intramo e spirto,
si ch'io vidi un colpo e feroce
con la spada a voller tagliano,
ch'io vidi fenno,
d'el ciel ch'a voi e 'l chiamar parvermento
di come cantanno,
che sposse a le piume,
s'infamicaldo
e farse e scritte, e tre gridar prende,
ignanno in altro dipenti,
ch'a cio che tu face>>. mi disse, <<in cio ch'io vidi,
qui si parlando, aspetto
ver' la cagion de la proda,
ch'ogne volle memoria,
da peradamo,
che sposse unia e fa parvere,
ancor ciascuna
per far per testando il peso,
che non sarebbero,
non da parlare,
vidi secondo ch'aveta,
chi avien pareai
che tu vedrai
le crede
che trasmuto a vistar la mente,
e 'l santo letto a la piaggia art

A sample search of the phrases produced by the network also seems to exclude the possibility that network overfit and only copy part of the original opera. Some sequence of world can be copied but no entire phrases.

Regarding the word produced by the network , especially in the second case, the style remember the one of Dante's poem but it's difficult to understand which word are completely invented and which one are only ancient italian. In table 4 are reported some prediction of 10 characters based on different seed; the * indicates that the network create a new line. We can see that some word are concluded as we expect (like *nel mez*) and with unknow word of no sense input the network usually create a new line and start a new section. Also with some word begin a new section or create a no sense word, like with *chie*.

Seed	Text Created
nel mez	nel mezzo * si fece d
infe	infermi * di l a s
virg	virgilio, * fur l'an
chie	chieri? venia in
asdasda	asdasda * per lo suo
areo	aereo * verita mai

TABLE 4: Word prediction for *Divina Commedia*

V. Results - Gerusalemme Liberata

In the box is reported a prediction of 750 charcters from the network trained on the *Gerusalemme Liberata*. The text seems slightly more sensible respect the output of the model trained on *Divina Commedia*. This can be related to different factor:

- The model of *Divina Commedia* was trained on the singles terza so the example text was very short. Instead with the *Gerusalemme Liberata* we use entire octave that contain a lor more of text respcet the singles terza.
- The network used for *Gerusalemme Liberata* was more complex so it is possible that it succeeds in create a better internal structure to represent the poem and predict characters.

The seed used for this text was *andiam*.

andiamente
a divare insieme, parla al frango abbiatse
piana, che l'un feriro e dell'imperio
sicuro stimo, e si comprese appieno,
sempre uccide, od abbatte, o si ristringe
sotto l'arme al proprio face in guerra illustri,
e di lui ch'or da lui pendo il tragge la fronte
si liqui, quasi due scorto di vorta,
molti dall'arme il ciel rimbomba, e i sensi
legati giudici in disparte al padiglione,
crin, poi con l'omba del tutto, e si consiglio;
ei segue, e lui tra via s'esse, e con lor celoce,
ch'ai se stesso accusa, e dio ringrazia
delle soggetta gente palma un torrente
curando, uscir prima, e se n'intende
il conservir re nell'alma il sangue sparso
di dar ma strada al seggio, ed ei v'ascende
ogni fonte al fianco il diviete
da lui le lodi e 'l merto inarcano

Despite the better results in the common sense of the word produced, the network even in this case generally fails to recreate the rhyme scheme.

The structure is more homogeneous respect the prediction with the *Divina Commedia*. This is probably due to the fact that the input example are longer.

Also in this case sample search exclude the possibility that the network overfit. Some sequence of word are copied from the text but no entirely phrases.