Text Generation with Machine Learning

Decreux-Duchêne Théo
Hachemi Ilies

January 2021

1 Context

Natural language processing refers to the techniques used by machines to process human language. In this project, we will attempt to generate text using a recurrent neural network. We present the data first, then we will see the processing before we can give it as input for our neural network. We will quickly come back to the theory behind these neural networks to finally build a model, and train it in order to generate text. The goal of this project is the following :

Given a sequence of characters, we would like to be able to predict the next character in the sequence.

2 Data presentation

2.1 Pride and Prejudice, 1813

Pride and Prejudice is a novel written in 1813 by Jane Austen, an English author of the time. This book contains 723,133 characters including 87 unique characters. We will use this novel to feed our model.

2.2 Word processing - Tokenization

The first step is to vectorize the text. We will do a mapping of each of these 87 unique characters and associate them with a key. For example, we could have the following dictionary for the first 3 characters :
{'A': 0, 'B': 1, 'C': 2, 'D': 3 ... }

The interest being to be able to cut out the words and more generally the sequences of characters in vector form.
So the sentence "HELLO WORLD!" will correspond to the sequence: [37 34 41 41 44 2 52 44 47 41 33 3]. The computer understands us, using reverse mapping we understand the computer. We can be interested in the modeling

of the network. Since our data are presented in the form of a sequence: we are going to use recurrent neural networks.

3 The model

3.1 Recurrent neural network - Theoretical framework

Recurrent neural networks are a class of neural networks that allow past predictions to be used as inputs, through hidden states. We can represent it as follows:

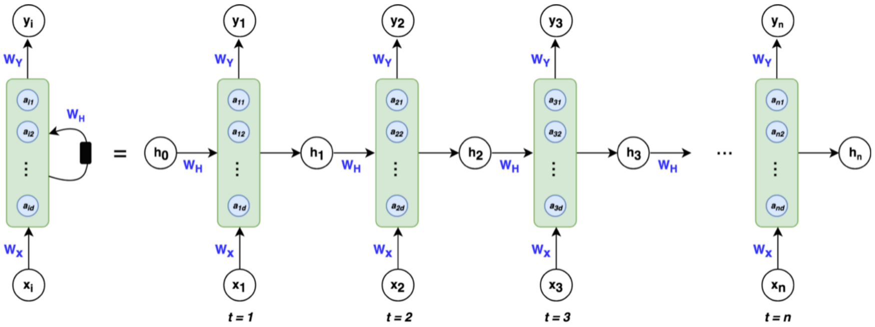


Figure 1: A RNN schema

So we have,

$a_t = W_H h_{t-1} + W_X X_t$

$h_t = g(a_t)$ where g() is the activation function

$y_t = softmax(W_Y h_t)$

Let's remind the cost function, denoted L

$L_t(y_t \hat{y}_t) = -y_t log(\hat{y}_t)$

The overall loss is the sum of $L_t(y_t \hat{y}_t)$, so we have :

$L_{total} = -\sum_{t=1}^{n} y_t log(\hat{y}_t)$

Now we have define the total loss, we would like to use it to compute the gradient of W_h, W_X and W_Y to be able to update the weights[1], considering and taking to account the loss at each step.

[1] with backpropagation

Considering the chain rule with respect to W_Y, we have

$$\frac{\partial L_{total}}{\partial W_Y} = \sum_{t=1}^{n} \frac{\partial L_t}{\partial W_Y}$$

Reasoning the same way for W_X,

$$\frac{\partial L_{total}}{\partial W_Y} = \sum_{t=1}^{n} \sum_{k=0}^{n} \frac{\partial L_t}{\partial \hat{y_t}} \frac{\partial \hat{y_t}}{\partial z_t} \frac{\partial z_t}{\partial \hat{h_t}} \frac{\partial h_t}{\partial \hat{h_k}} \frac{\partial h_k}{\partial \hat{W_X}}$$

The advantage of this type of structure lies in its ability to take into account the past, in the fact that the size of the model does not depend on the input vector and that the size of the input vector is unlimited. The major disadvantage is related to the computation time [2].

Vanishing/Exploding Gradient and activation function : Insofar as we want to keep temporal dependencies, RNNs involve multiplying several gradients between them, we immediately realize the problem that this can imply on the updating of coefficients of our neurons. Indeed, let us take the "vanishing" case, multiplying a lot of small gradient between them would make the result converge towards 0 and therefore limit the updating of the coefficients.

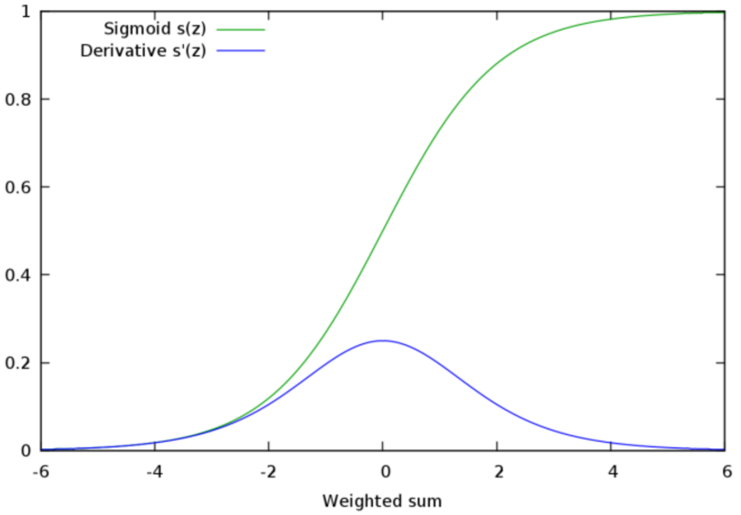


Figure 2: The example of the sigmoid function

[2] By the way, it was very problematic given our personal configurations. Launching a slightly evolved model often led to unrealistic computational times for our machines (several hours or even days), in that sense we probably weren't able to experience the model as much as we would have liked.

GRU / LSTM : Gated Recurrent Units (GRU) and Long Short-Term Memory units (LSTM) limit the problem of the disappearing gradient encountered by traditional RNNs. We will use GRUs in our modeling, they have performance comparable to LSTMs but require fewer parameters to be estimated.[3]

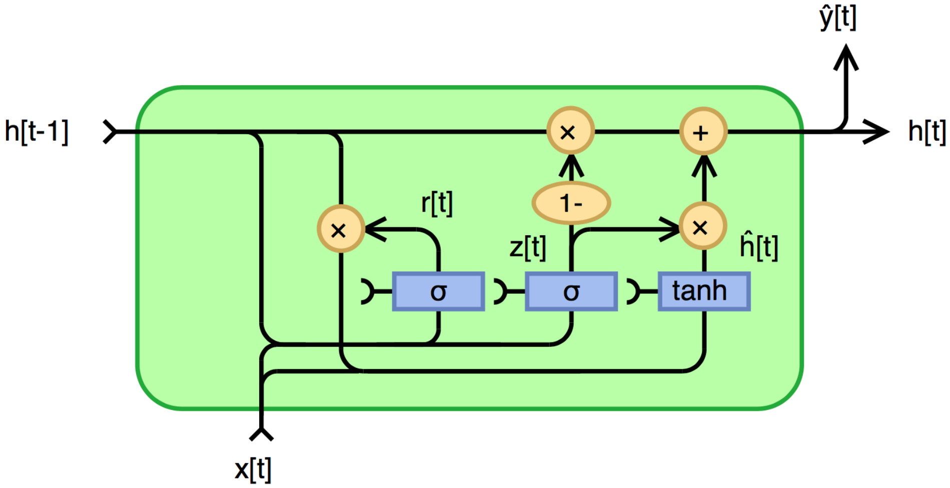


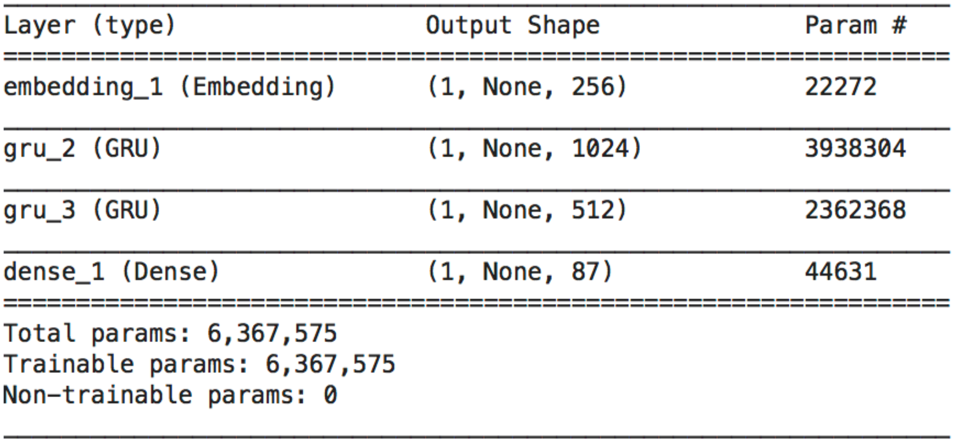
Figure 3: GRU : Gates principle

The GRU consists of two doors and an exit state. The reset gate controls how much past information should be retained, while the update gate controls what should be retained and forgotten.

[3] We would have liked to try the LSTMs, but our machines did not allow us to do so when we made the model a little more complex.

3.2 The model

We have trained the model for 30 epochs with batches of size 64.



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (1, None, 256)            22272
_________________________________________________________________
gru_2 (GRU)                  (1, None, 1024)           3938304
_________________________________________________________________
gru_3 (GRU)                  (1, None, 512)            2362368
_________________________________________________________________
dense_1 (Dense)              (1, None, 87)             44631
=================================================================
Total params: 6,367,575
Trainable params: 6,367,575
Non-trainable params: 0
_________________________________________________________________
```

Figure 4: GRU : Gates principle

3.3 Results

Text 1 :

Therefore, she found the whole of influence of their behaviour to the gentleman was the only one of the past was the only creature who could suppose there might be any extension. A great deal of concern, he thought it a most unforgiving speech," should her ladyship would think her like her not to herself, but to the very relations whom he had seen at Rosings, which is the last time I shall speak it. I am sure I shall be extremely well pleased. I do not believe that he will say that.

Text 2 :

She was to encounter Charlotte's observations on the subject of a pause, as well as her own way. –It was the only creature who could suppose there might be as little as possible, when the melancholy event takes place, as to any thing they had expected. "He is perfectly well behaved, polite, and who was equally poor?" "I believe it at this time have been in the habit of resentment, or any congratulations to Elizabeth was forced to put it out."
Perhaps he must have her own way.

Text 3 :

Arcy was not so palatable to your mother and sisters, though she foresaw little pleasure in the country. It was spoken of again while they were at the door of Lydia's prospects of Mr. Wickham, when last at Netherfield for the truth, I was too like him, I should think it occasional glance at home. Her heart will have to conceal it, from that very wish which Elizabeth had received at the same time haughters have been always at Longbourn.

A few comments:

- Although the sentences retain little meaning, we can rejoice in the relevance of the syntax. The sentences are grammatically correct, the tense concordance is often well respected.
- Punctuation is not always correct, but almost. It often is correct with regard to "." and the ",", the beginnings of sentences do indeed start with a capital letter. However, it is more difficult for the program to use quotes correctly.
- In green, we notice that although the sentences do not always make sense, the program sometimes succeeds in using words from the same lexical field, which shows a beginning of understanding of the vocabulary on its part.
- We also emphasize, and this is important, that the program does not invent words. Depending on our tests, some of our attempts have led us to abating models, this one has the merit of not inventing a new word at first glance.[4]
- We have noticed that certain expressions come up often, for example in orange "The only creature", knowing that this expression appears plenty of times in the book of Jane Austen, could underline that our input text is maybe not so various ? Other expressions, words come up often (for example "might", "could" ...) but the book being a British novel of the 1800s, like a book by Shakespeare it is normal that the level of strong English may seem redundant today.
- Finally the structure of the text is well respected, the program provides us paragraphs, quotes with dialogues between characters ...

[4] Maybe this can happen, it would be interesting to check it by generating a few thousand words and comparing them to an existing word list, in the dictionary for example.

3.4 How to improve this model

The first thing we can do to improve the model is to increase the epoch count to allow the model to train for longer. We could add more layer, and memory units to this layers. We only tried 64 batch-size, maybe we could do better at this step as well. We would have liked to try an LTSM with a drop-out parameter and several layers but our machines did not allow it. That being said, this slightly more modest model is giving encouraging results.

4 Bibliography and Sources

- Sherstinsky, Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network, 2020, Physica D: Nonlinear Phenomena, vol 404.

- https://towardsdatascience.com/how-our-device-thinks-e1f5ab15071e

- https://stanford.edu/ shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks

- https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

- https://www.tensorflow.org/tutorials/text/text_generation