Deep Learning – HW3

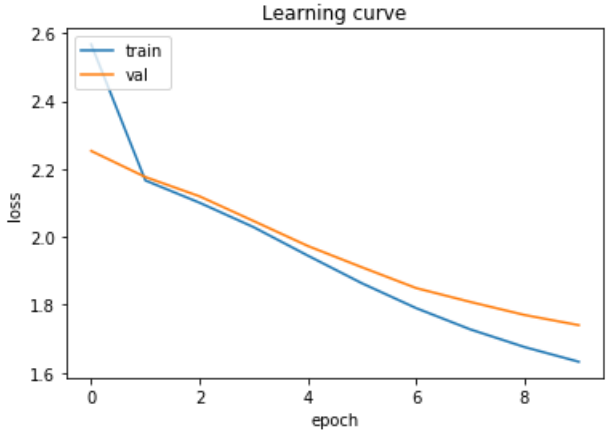
I. Standard RNN

Architecture :

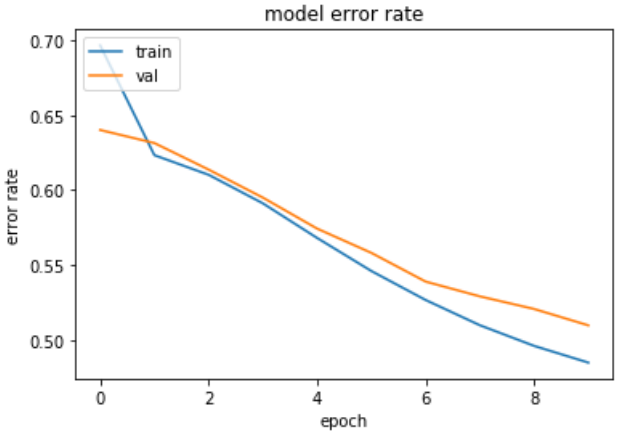
```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
simple_rnn_1 (SimpleRNN)     (None, 1024)              1118208
_________________________________________________________________
dense_1 (Dense)              (None, 67)                68675
=================================================================
Total params: 1,186,883
Trainable params: 1,186,883
Non-trainable params: 0
```

I have one simpleRNN layer with 1024 units, for a total of over 1 million parameter. I used it on google colab in order to make the computation possible. My computer doesn't have any graphic card, so it can't compute that large networks.

Learning curve :



Training and validation error :



Breaking Points :

INPUT : « No more talking »

2 epochs

```
No more talking.
Death tom thit whic  hae the wardaghay goof Kingt.-
Forbees to.

ARENIO:
Eull we the mars cop woxyich it of his my fithem.
Dersweea halomat no gome than?
Why sn it bue
thingthes: 't hy sfee sins beo
```

4 epochs

```
No more talking it thing hor sheve be nge the prouthous dyourthir for t willly: 'lond, tive me tound;
You fremint ofe gaver:
Fhim dof mas ard tous thoury gand, tolden! I an tchice you Pougr
By sald not wringtt notre
```

6 epochs

```
No more talking of
Nay not browhen: I lodreand
It high it allntw, I' wordny, thit wsulf
I feat an gobst a meane tremont charnews:
with coman muser Sheps thameemas for an for had st oppeotchis With.
Gid Goves ser,
Se
```

8 epochs

```
No more talking, of drrungs and oly,
My highterssboripl ap rysold:
Eadle cluit commentes ous fall'd for hemees on my good my poor: Scerian simjed with wime thy llanfe
Am you, good covillt of good fime
The orbout for
```

10 epochs

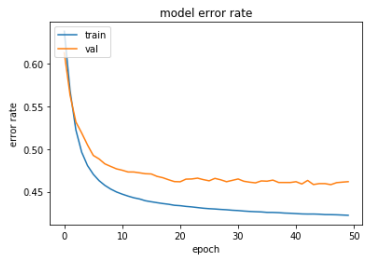
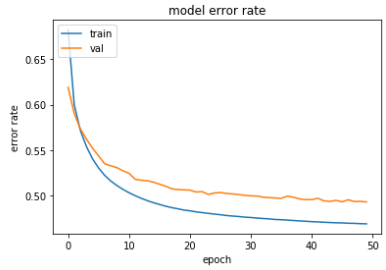
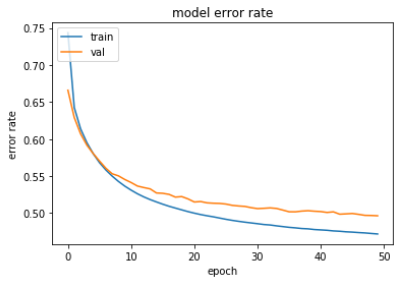
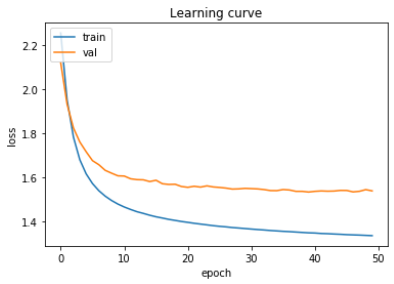
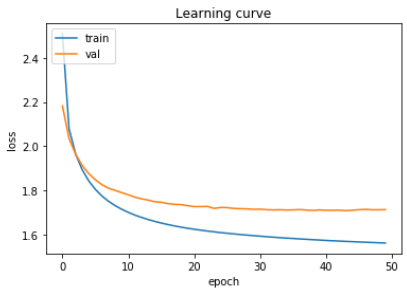
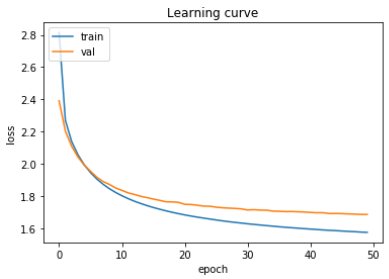
```
No more talking of, cown.
The evenes with some to dight roth resead?

DUCHETS REGBORDECBESHY:
Yes, lady: why, you shall mada i' the soult.

MONBACHO
TERY A:
Thark'st t of anare.

SHALLOWO:
I carrunt their offer such
```

The computation were too long for me to plot a training loss with different sequence length and hidden layer sizes. So I tried several architectures and compare the results. See table below :

Sequence length : 5 Hidden layer : 1024 units epochs : 50	Sequence length : 5 Hidden layer : 128 units epochs : 50	Sequence length : 15 Hidden layer : 128 units epochs : 50
		
		

We observe that by lowering the number of units in the hidden layer, the results are less good, but the computations are a bit faster. We have an error rate increased by around 0.05, and the loss function increased by 0,2. I felt like the predictions were less understandable too by trying on a few examples.
When increasing the sequence length, still with 128 units to compute faster, we observe that the learning curve and the error rate are quite the same. But the predictions shall consider more the context of the sentence. But watching some examples of predictions, I felt like it was less easy to understand the meaning.

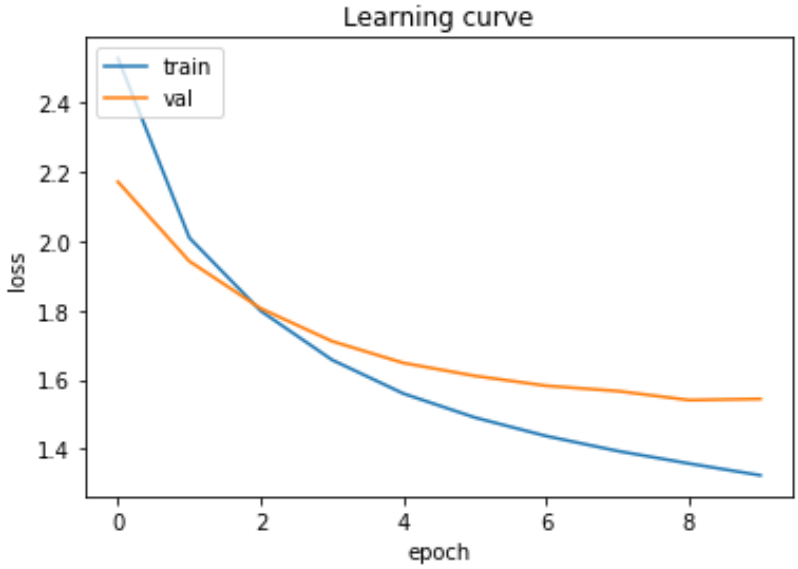
II. **LSTM**

Network architecture :

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 1024)              4472832
_________________________________________________________________
dense_1 (Dense)              (None, 67)                68675
=================================================================
Total params: 4,541,507
Trainable params: 4,541,507
Non-trainable params: 0
```

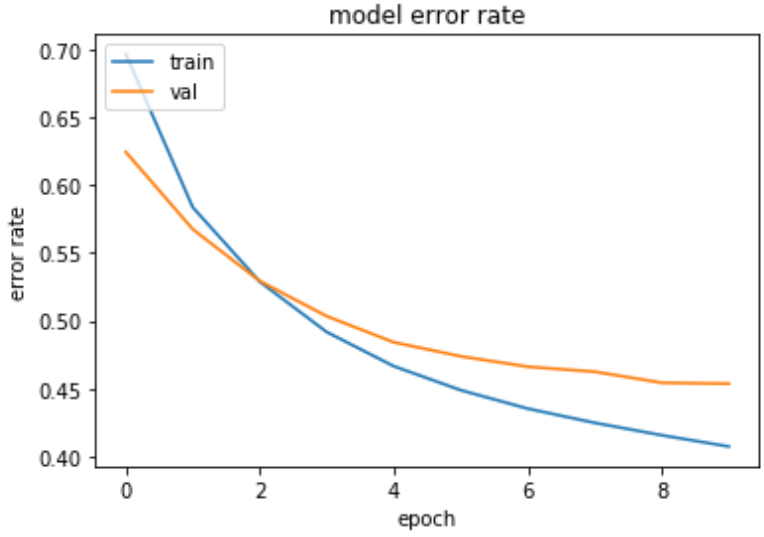
One layer type LSTM with 1024 units, when I had a lower number of units, my NN was predicting the same sentence again and again, which was frustrating for me. I tried to increase the number of units and had much better results.
Then one dense layer which is our output layer, to predict the character. I didn't look for something more complicated, because this network has already 4,542,507 parameters, and took more than 30 minutes to compute.

Learning Curve :



We see that the model starts overfitting the trainig datas, so increasing the number of epochs wouldn't make the results any better without the add of a regularization term.

Training and validation error rate :



Breaking Points :

INPUT : « No more talking »

2 epochs

```
No more talking
ar ustinn's, he cup, noveMa they shall hat streforings dearst Bustow!

BYOWMTFUMY US:
To shill torests of Fllane; a wist
REnol-Ro andy cing if make I rooke ceme to the Forst, and
nith attis make hom.
```

4 epochs

```
No more talking, them to exprict,
I lyevny it nower take one the
mentown is by the rakes in hosour,
and whow I hear bedief dayd
And wost my love, foot
This brec onimans is sonsoures not inconden you!

CARINSA:
How h
```

6 epochs

```
No more talking o'er-thing this ring words
Mance't in he it.
Scyipus, I go about there is his all,
That so geadly re
uppgin a chuel-mangered partly's chain.
Come, come,
Now I shall ever me,
ABret my burn: them, call
```

8 epochs

```
No more talking father's.

SHALLOW:
Set at the head his gaw-s
hear death's father behold
Euts hold with his
mouthed and injuries shumenor's cown;
Thriw sit uson your heart.
And make it.

SAY SANG:
Marry, at somaties
```

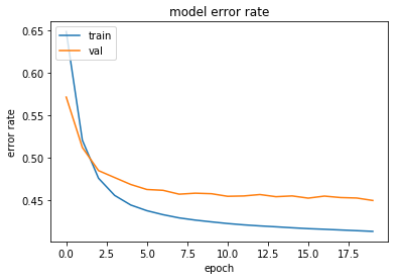
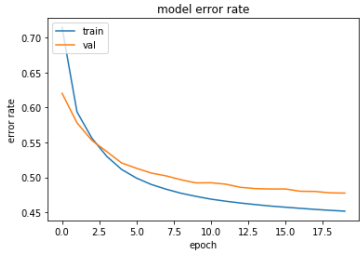
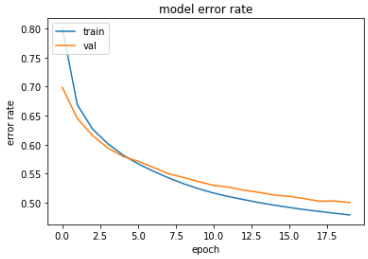
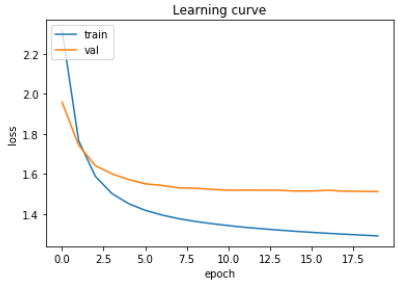
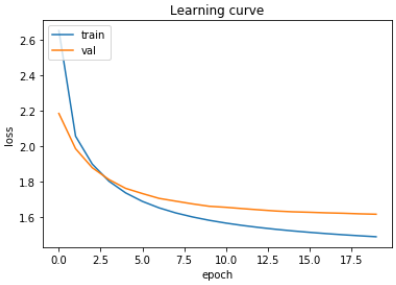
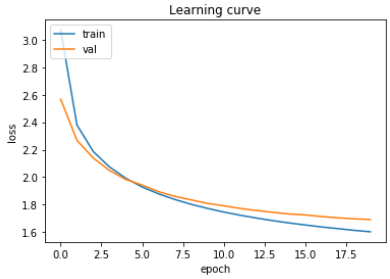
10 epochs

```
No more talking the gods. My ground and her Hest.

PRINCE HENRY:
But what's my mistress short
The vew some flowly fortune father, hulldy, fathermag from them.

KING HENRY VI:
O, wasing yet the grace to separt poor m
```

Different parameters for LSTM :

Sequence length : 5 Hidden layer : 1024 units epochs : 20	Sequence length : 5 Hidden layer : 128 units epochs : 20	Sequence length : 15 Hidden layer : 128 units epochs : 20
		
		

We see that lowering the number of units in the hidden layer didn't decrease the accuracy so neither the error rate. The learning curve also shows that that loss isn't as low as with a high number of units in the hidden layer. When increasing the sequence length to 15, the error rate is getting higher, without changing the loss by much. The results overall are still pretty similar. I think we should have more significant change to observe bigger difference between the models.

Discuss the difference between standard RNN and LSTM

It is difficult to train standard RNNs to learn long-term temporal dependencies, because the gradient of the loss function decays exponentially with time (cf vanishing gradient problem).
LSTM uses special units, to include a 'memory cell' that can maintain information in memory for long periods of time. It can controle entering informations in the memory, as well as forgetting other informations. This architecture allows us to learn longer-term dependencies.

Generate words with « JULIET » as context using LSTM :

```
JULIET:
I say a mine own
dogs fath,
Embateay came,
The heart of suched budy
Up: at no fite parter--
As food, sport;
Say in the sugar, if not vain;
And he is him
Ede a lambirious wit,
Yes of glory of wize say, Anros!

Mistress Awonding!

THOISS:
Who' pray's
Yoster drash maids and in solems to Lisuo,
Let not the revenged, and the devil be
present them to get thee this cost, and nature as I call my colsents it stine is dead, sir, it wish
As I am comparion.
ALONCO:
What! ha!' it old yes: be agid to thee.
```

I have the following results, which is really cool. We have close from full sentences, which shows how powerfull are the LSTM for langage prediction. If we can manage to have such results by ourselves, we can bearly imagine what Google or Facebook can do using their optimized tool !