Practical work 13 – 12/12/2019
Recurrent Neural Networks with Keras - part 2

Objectives

The objective of this PW is to practice with some applications of Recurrent Neural Networks (RNN) for equence generation.

Submission

— **Deadline** : Thursday 19 December, 12 :00 (noon)

— **Format** : Zip with report and iPython notebook.

Exercise 1 Sequence generation - startup names

The objective of this exercise is to build a generator of startup names using a RNN. We will use to train the system a corpus of existing startup names composed of 172K entries. Figure 1 illustrates the 20 first entries of the corpus.

Many-to-one approach

a) Read again the examples on the Shakespeare text generation in the section *Generative RNN* from the class support. Make sure you understand the different steps for the data preparation. You can also start from the ipython notebooks provided on Moodle for the Shakespeare text genration.

b) Get from Moodle the file `companies.csv` that will be used to train the system.

c) Read the data from the file. Treat the whole set of names as a unique string of text, like in the Shakespeare example.

d) Follow similar steps for the data preparation as in the Shakespeare example. First extract the set of chars and define the encoding and decoding dictionaries. Then chop the data into X and y, you may define here a sequence length of 10 and a skip of 3. Finally transform your data and labels into one-hot vectors.

```
Hashplay Inc.
New Incentives
GrabJobs
MediBookr
MelissaWithLove.co
Starting 11
The CarShare Guy
Allahabad Bank
Anlaiye
Any Time Loan
Asieris Pharmaceuticals
Birner Dental Management Services (PERFECT TEETH)
Blockchain Foundry
Breethe
Buffalo Wild Wings
Canadian Solar
Convert Group
DeepCam
Doctaly
Gaze Coin
```

FIGURE 1 – 20 first entries of the training set of startup names

e) Define your model using a SimpleRNN (64 units) and a Dense with softmax activation layers.

f) Train your model.

g) Generate startup names using different values of the hyperparameters : number of epochs, number of cells in the RNN, etc.

Report about your findings in the pdf report. Select 5-10 generated startup names that you find interesting.

Many-to-many approach

Redo the previous exercise using this time a many-to-many approach. You will need to modify slightly the way the target tensors are prepared in order to do so. Pay also attention to the parameters of the layer definition that are a bit different.

Using LSTM and GRU

Do some experiments replacing the simple RNN with LSTM or GRU layers and comment on your observations.

Exercise 2 Optional : Simple Seq2Seq in Keras

Read and try to re-implement the basic Seq2Seq model using Keras functional API from https://blog.keras.io/a-ten-minute-introduction-to-sequence-to-sequence-learning-in-keras.html.

The model is applied to translating short English sentences into short French sentences, ?character-by-character. Note that it is fairly unusual to do character-level machine translation, as ?word-level models are more common in this domain.

Report on your experiments.

Exercise 3 Optional : Review Questions

a) How can you define a generative system ? Describe the two approaches seen in the class to build generative systems with RNNs.

b) Draw the architecture of a Seq2Seq model and explain the encoder/decoder concept.

c) What is the main problem of a basic Seq2Seq models used for machine translation and the solution that can be used to overcome this problem ?