HACETTEPE
UNIVERSITY
COMPUTER
VISION LAB

CMP722: Deep Learning for Computer Vision
Spring 2017

PRACTICAL 3

Due on April 22, 2017 (23:59:59)

Instructions. In Practical 3, you will work on extending Andrej Karpathy's min-char-rnn.py, a character-level vanilla recurrent neural network for language modelling. You will experiment with the Shakespeare dataset[1].

```
PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.
```

Part 1

Write a function to sample text from the model using different temperatures ($1/\alpha$), in which you will choose word $w^{(j)}$ according to:

$$\frac{\exp\left(\alpha\widehat{P}(w_1 w_2, \dots w_{k-1} w^{(j)})\right)}{\sum_j \exp\left(\alpha\widehat{P}(w_1 w_2 \dots w_{k-1} w^{(j)})\right)} \tag{1}$$

where $\widehat{P}$ is the input to the softmax that produces the probability in the RNN. Try different temperatures, and, in your report, include examples of texts generated using different temperatures. Briefly discuss what difference the temperature makes. Include the listing (i.e., source code) of the function you wrote/modified to accomplish the task in the report. You should use the weights from Part 3.

Part 2

Write a function that uses an RNN to *complete* a string. That is, the RNN should generate text that is a plausible continuation of a given starter string. In order to do that, you will need to compute the hidden activity h at the end of the starter string, and then to start generating new text.

Include **at least five interesting examples** of outputs that your network generated using a starter string. (This part need not be easily reproducible). Again, you should use the weights from Part 3.

Bonus (Extra 40 pts): Train the model on a corpus of your choice (other than Shakespeare) and generate samples using your trained model.

Part 3

[1] Adapted from a homework developed for Toronto CSC321 Introduction to Machine Learning and Neural Networks course, offered by Michael Guerzhoy.

The weights for a trained RNN are available here. Some sample from the RNN (at temperature=1) are here, and code to read in the weights is here (if this doesn't work, try the pickle file, and get a using import cPickle as pickle; a = pickle.load(open("char-rnn-snapshot.pkl")).

In the samples that the RNN generated, it seems that a newline or a space usually follow the colon (i.e., ":") character. In the weight data provided, identify the specific weights that are responsible for this behaviour by the RNN. In your report, specify the coordinates and values of the weights you identified, and explain how those weights make the RNN generate newlines and spaces after colons.

Part 4

Identify another interesting behaviour of the RNN, identify the weights that are responsible for it. Specify the coordinates and the values of the weights, and explain how those weights lead to the behaviour that you identified. To obtain more than 2/10 for the bonus, the behaviour has to be more interesting than the behaviour in Part 3 (i.e., character A following character B).

What to turn in

The practical should be implemented using Python 2. You will be turning in a written report of your solutions in a single PDF file. **These reports should be prepared using LaTeX using NIPS style** and will be submitted via e-mail. A zip file containing the corresponding template and style files are provided at the course website. **The write-up must be no more than 6 pages long.** You should present your results in a clear way using well-designed tables, plots and figures. Do not include your codes.

Grading

The practical will be graded out of 100 points: 0 (no submission), 20 (an attempt at a solution), 40 (a partially correct solution), 60 (a mostly correct solution), 80 (a correct solution), 100 (a particularly creative or insightful solution). **Note that the grading depends on both the content and clarity of your report.**

Late Policy

You may use up to five *slip days* (in total) over the course of the semester for the three practicals you will take. Any additional unapproved late submission will be weighted by 0.5 and no submission after five days will be accepted.

Academic Integrity

All work on assignments must be done individually unless stated otherwise. You are encouraged to discuss with your classmates about the given practical, but these discussions should be carried out in an abstract way. That is, discussions related to a particular solution to a specific problem (either in actual code or in the pseudocode) will not be tolerated. In short, turning in someone else's work, in whole or in part, as your own will be considered as a violation of academic integrity. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else.