Deep Structured Learning (IST, Fall 2018)

Homework 4

Instructor: André Martins
TAs: Vlad Niculae and Erick Fonseca

Deadline: Wednesday, December 12, 2018.

Please turn in the answers to the questions below together with the code you implemented to solve them (when applicable). Please email your solutions in **electronic format** (a single zip file) with the subject "Homework 4" to:

`deep-structured-learning-instructors@googlegroups.com`

Hard copies will not be accepted.

Question 1

Transliteration. Transliteration is the problem of converting text (usually entity names) from one script to another. For example, `васильевич` in Russian (Cyrillic script) is transliterated as `Vassiljevitch` in English (Latin script). We can regard this is as a sequence-to-sequence problem, where the sizes of the two sequences do not necessarily match.

In this exercise, we will use the Arabic-English transliteration data released by Google (`https://github.com/googlei18n/transliteration`).

Run the following commands to download the train, validation, and test partitions (resp. 12877, 1431, and 1590 word pairs):

```
wget https://raw.githubusercontent.com/googlei18n/transliteration/master/ar2en-train.txt
wget https://raw.githubusercontent.com/googlei18n/transliteration/master/ar2en-eval.txt
wget https://raw.githubusercontent.com/googlei18n/transliteration/master/ar2en-test.txt
```

1. You are going to implement a sequence-to-sequence model for this task. The input and output should respectively be an Arabic and a English word, represented left-to-right as a sequence of characters. The evaluation metric is Word Accuracy (which counts the fraction of words that were fully transliterated correctly).
   (a) (10 points) Start by determining the source and target vocabularies (don't forget to include special symbols, such as START, STOP, UNK, and PAD (if you pad). What are the vocabulary sizes?
   (b) (30 points) Implement a vanilla sequence-to-sequence model using an encoder-decoder architecture with two unidirectional LSTMs (one encoder LSTM and one decoder LSTM). Report the validation accuracy as a function of the epoch number and the final test accuracy. Hint: if you're using Pytorch, use the function `nn.LSTM` for this exercise.
   (c) (10 points) Repeat the previous exercise reverting the source string.

(d) (30 points) Turn the encoder into a bidirectional LSTM and add an attention mechanism to the decoder. Report the validation accuracy as a function of the epoch number and the final test accuracy.

Question 2

Baselines in reinforcement learning. The REINFORCE algorithm commonly uses a "baseline" to reduce the variance of its parameter updates. In this exercise, you will prove the underlying result that makes this strategy possible.

1. (10 points) Let $x \in \mathcal{X}$ be a random variable distributed according to $p_{\boldsymbol{\theta}}(x)$, where $\boldsymbol{\theta} \in \mathbb{R}^d$ is a parameter, and let $f : \mathcal{X} \to \mathbb{R}$ be a function independent of $\boldsymbol{\theta}$. Show that $\nabla_{\boldsymbol{\theta}} \mathbb{E}[f(x)] = \mathbb{E}[f(x) \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(x)]$, where the expectation is with respect to $p_{\boldsymbol{\theta}}$.

2. (10 points) Use the above fact to show that $\mathbb{E}[\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(x)] = 0$ and that $\mathbb{E}[f(x) \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(x)] = \mathbb{E}[(f(x) - b) \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(x)]$ for any constant $b \in \mathbb{R}$. Comment how b can be used to reduce the variance of a Monte Carlo gradient estimator of $\nabla_{\boldsymbol{\theta}} \mathbb{E}[f(x)] \approx \frac{1}{k} \sum_{i=1}^{k} f(x_i) \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(x_i)$.