DATA.ML.200 Pattern Recognition and Machine Learning

Exercise week 8: Do-It-Yourself RNN (advanced)

Be prepared for the exercise sessions (watch the demo lecture). You may ask TAs to help if you cannot make your program to work, but don't expect them to show you how to start from the scratch.

1. `python` *Letter prediction MLP v3 (20 pts)*

   In the previous RNN exercise we found that the model with immediate input only,

   $$\mathbf{x}_{t+1} = MLP(\mathbf{x}_t) \ ,$$

   works for the text file containing sequence 'abcde abcde', but does not work for the file containing sequences 'abcde edcba'.

   We further found that the model that uses the two last inputs,

   $$\mathbf{x}_{t+1} = MLP(\mathbf{x}_t, \mathbf{x}_{t-1}) \ ,$$

   works well with the both files.

   This results gives an idea that a model with internal state,

   $$\mathbf{x}_{t+1}, \mathbf{h}_{t+1} = MLP(\mathbf{x}_t, \mathbf{h}_t) \ ,$$

   is able to solve the both problems if the model learns to pass the previous input via its state ($\mathbf{h}_{t+1} \sim \mathbf{x}_t$.

   Implement a neural network state model that uses the current input and previous state to learn a new state

   $$\mathbf{h}_{t+1} = f_1(\mathbf{x}_t, \mathbf{h}_t) \ ,$$

   and uses the new state to estimate the next output

   $$\mathbf{x}_{t+1} = f_2(\mathbf{h}_{t+1}) \ ,$$

   Implement a training procedure for the network and make sure it works well with the both data files.

   For evaluation, test the network with a fixed sequence, and test the network also for pure text generation. Note that for generation you may need to convert network output vectors to one-hot encoded vectors to robustify generation process.

   Return the following items:

   - Python code: char_rnn_v3.py that does all above.
   - PNG image: your full desktop screenshot that includes a terminal where the python file is executed and printing the requested results: char_rnn_v3_screenshot.png