

Project 3

Growing Neural Gas or RNN/LSTM

In project 3, investigate ONE of the following topics

- Recurrent neural network (or long short term memory)
- Growing neural gas

If you select RNN/LSTM, then either

- Find two *simple* datasets to explore
 - e.g., <https://machinelearningmastery.com/time-series-datasets-for-machine-learning/>
- Generate your own synthetic dataset
 - I like this route because you know the answer and can control things
 - e.g., sequence length, noise, number of samples, etc.

If you select GNG

- Create your own datasets
 - My recommendation, load "paint", make {black, white} pixels, convert to data set
 - load image, make each pixel that is black a data point, i.e., (x,y) location
 - This will allow you to create a variety of different and wild patterns
 - Easy then to create (aka draw!) ways to break the algorithm

You can use Python/NumPy/PyTorch

- GNG
 - I want you to code it up yourself
 - Play around with the algorithm: i.e., proximities, topologies, update rules, etc.
 - Search the literature for an extension or two?
- RNN/LSTM
 - Option 1, from scratch
 - If you want to code up an RNN or LSTM and code up BTT or TBTT, wonderful!
 - That is a level of depth that I am OK with
 - Option 2, build on tools
 - If you go with say built in PyTorch support (existing class/implementation), then I want you to experiment with 3 different types (e.g., classic, peepholes, GRUs, etc. (<https://dzone.com/articles/5-types-of-lstm-recurrent-neural-networks-and-what>))

NOTE: the above is what is my minimum expectation. If you want to deviate from the above and you have something compelling and you can convince me that you will learn the same or more and stay connected to the underlying NN jazz, shoot me an email! ;-)

Project Details

*This is an **individual assignment**. All work should be the author's and in accordance with the university's academic integrity policies.*

More specifically:

- You can **NOT** work with anyone else (person, internet, robot, etc.).
 - This assignment is for you and you only!!!
- If you turn in late work, the discount rate is
 - 1-3 days late, one letter grade (i.e., max of 100% goes to 90%)
 - 3-5 days late, two letter grades (best you can get is an 80%)
 - each day after 5 is one additional letter grade

NOTE: start (really) EARLY gang (like today)!!!

Submission Procedure

Submit via Canvas (< 10MB file PDF).

Code

Your code needs to be documented and you need to make a README.txt. The README.txt should tell ME how to install your code/software. Your README.txt should also include step-by-step instructions on how to run your program.

Submit via Canvas (< 10MBs).

YOU CAN USE PyTorch (see details above).

Report

Turn in a 20-30 page report (double spaced, one column format, Times New Roman 12 point font) in the following format:

1) Title Page (1 page)

- name, pawprint, date

2) Technical Description (3-10 pages)

- story form
- in your OWN WORDS
- show me you know the high-to-low level details
- combination of pictures, equations, description, etc.
- **have you convinced me that you know the in's and out's?**

3) Experiments and Results (10-20 pages)

- discussion, tables, plots, etc.
- co-"show/plot" as much as possible (comparion)
- do NOT DUMP data/results, find the STORY and report those parts
- **think then write!**

4) Lessons Learned (2-4 pages)

- quick summary paragraph
- did everything turn out like you thought it would? surprises? etc.
- discuss shortcomings
- discuss what you would do in the future if you had more time

Brownie points if you develop it in LaTeX, perhaps consider using OverLeaf? ;-)

NOTE: I think its grand if you coded up this project. BUT, I grade based ON YOUR REPORT.