University of Missouri

Neural Networks: Spring 2023

Project 1

Hi class,

In project 1, you will create a supervised learning-based neural net (NN). Your goal is to NOT simply obtain "96.456784 accuracy". Your goal is to code, vary, and explore!!! I care more about your journey and report than your results. Specifically, in project 1 you will:

- Implement and compare a (1) multi-layer perceptron (MLP), a (2) radial basis function (RBF) net, and a (3) convolutional NN (CNN).
- Investigate these different NNs relative to classification (two data sets outlined below).
- Do qualitative (your eyeballs, plots, etc.) & quantitative (confusion matrix, F1-score, etc.) analysis.

You can build on:

- https://vcs.missouri.edu/andersondt/mizzou_spring_2023_nn/-/blob/main/NN/LinearRegressionInPyTorch.ipynb
- https://vcs.missouri.edu/andersondt/mizzou_spring_2023_nn/-/blob/main/NN/MlpInPyTorch.ipynb
- https://vcs.missouri.edu/andersondt/mizzou_spring_2023_nn/-/blob/main/NN/SimpleCNN.ipynb
- https://vcs.missouri.edu/andersondt/mizzou_spring_2023_nn/-/blob/main/NN/RBF.ipynb

Classification experiments

- **"EASY": Data set 1: MNIST digits**
    - Dataset: http://yann.lecun.com/exdb/mnist/
    - Input size 28x28 (so 28*28 = 784 inputs)
    - (60,000 train, 10,000 test) samples for {0, 1, 2, …, 8, 9} digits
        - Depends on what type of fancy computer you have (or don't). I am OK if you end up running on something like the first X samples for each class (something that fits into your compute) and test on the remainder. Yes, this will bias your outcome, but if you don't have the computing resources, it is what it is …
        - If you down sample, pay attention and try to keep a balanced number of each digit. For example, don't have a training dataset of 5,000 "5"s and no other digits :)
    - Build a CNN and explore away, e.g.,
        - Can you learn a two class net that classifies "1"s vs "8"s?
        - Can you learn a two class net that classifies "3"s vs "8"s? Was it simpler than the "1" vs "8" experiment? The same? How do you determine that?
        - Can you learn a network that classifies "0" vs "1" vs … "8" vs "9" (all 10 digits)? Was that harder than the "1" vs "8" or "3" vs "8" experiment?

        - What happens if you change: nonlinearity, criteria/cost function, learning algorithm, learning parameters, weight initialization, add/remove layers, filter sizes, input scaling, etc.? Can you see an analytical and/or experimental winners?
        - Can you solve this using a simple MLP vs a CNN or an RBF? How can you compare one to the other, e.g., an "apples-2-apples" experiment?
        - Can you solve this using a single convolutional layer? What if you try 10 filters that are the size of your input image? What filters did you learn (what are they doing)? You can visualize them. What did the machine do…? Did you end up in a black box or can you explain what your network learned? What if you increase the number of filters? Decrease the number of filters?
        - With respect to an RBF, do you do random weight initialization, cluster and allow the radial units to update, use a single sigma or multiple, diagonal versus full covariance matrix, etc.?
    - Common gang, the world is your oyster, explore, have fun, dig into that NN!!! Again, your goal is to NOT simply obtain "96.456784 accuracy. Your goal is to understand how these things "think" and you should be able to justify your approach.
    - At the end of the day, you have control over
        - Data set and how you set your experiment up
        - Network architecture
        - Network hyperparameters
- **"LESS EASY": Data set 2 : Fashion MNIST (https://www.kaggle.com/zalando-research/fashionmnist) or CiFAR-10 or CiFAR-100 (https://www.cs.toronto.edu/~kriz/cifar.html)**
    - OK, digit MNIST is the "hello world" or the IRIS dataset of NNs … (aka its simple!)
    - Let's try a harder data set.
    - Do your ideas and implementation hold up …?

Alternative … Yeah, ok class, I know you are possibly chomping at the bit to use your own data set … I outlined two above for y'all. However, if you have another data set AND you already have the data and it's a supervised learning problem, you could try to convince me to let you use that data set instead … I like it when the problem is real to you. But, not saying I will approve it, you can sure ask though! Now, if you do use a custom data set, I expect to see the same 90%+ work and analysis on the NN side, not your data set or application. I don't want "here is a report that is my problem with a sprinkle of NNs". Be sure that what you will work on results in your building, applying, varying, and learning about the above.

Expectations:

I am happy if you complete the above. Really. Ecstatic. However, I will grade you based on your report. Your goal is not to implement and run the above and be done. *At the end of the day, I will be looking for how well you understand these theories, tools, and what you did!*

Code:

You must program this yourself. Here are DOs and DONTs:

NOT ALLOWED:

- Absolutely NO working with others. NO robots, aliens, internet, this is YOU and your brain! Y'all can talk about NNs, CNNs, MNIST, etc. General class stuff. But you CANNOT talk about this project, how you are solving it, you cannot share your code (any of it), etc. DO NOT cross that line. You will get a 0. Really. If you have any questions, ask me first!!!
- You can code this all up from scratch (what I use to make y'all do!) or build on the Python and PyTorch examples I gave you. No unapproved libraries. You can always ask me if you want to use something else. You also can NOT start from/with anyone else's code (just mine)!

ALLOWED:

- You CAN use existing libs to load data (e.g., load MNIST) and analyze results (e.g., plot)
- You CAN use "BASIC PYTORCH". For example, you can extend my MlpInPyTorch.ipynb and SimpleCNN.ipynb examples.

Why!?!? Well, I use to make you code everything 100%... It took forever (was a nice exercise) and y'all did not get very far as a result. Many of you also were unable to produce numerically stable codes (crashed all the time, e.g., Inf's or Nan's coming out of nonlinearities). So, I embraced our new NN programming era. But it's a tightrope. I want you to use as basic of functionality as possible and still program up a NN. I just want to provide you more stable codes and simplify some things (e.g., loading datasets). What is your best bet? Start from scratch or start from my examples and go from there. AND IF I did not cover a question that you have, ASK!!! I will not penalize for asking, but if you cross the line and ask for forgiveness, that is another thing… Overall, gang I am trying to work with you and help you.

Final Report:

Remember, this is what I grade!!!!!

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

1) Title Page (1 page)

- name, pawprint, date

2) Technical Description (5 pages minimum and 10 pages max)

- high-level details (e.g., text description and figures)
- low-level details (e.g., equations and algorithms)
- use text, equations, figures, etc.
- end of day, do I believe you understand this stuff?!?!

3) Experiments and Results (8 pages minimum and 15 pages max)

- what did you run?
- why did you run it?
- 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) Reflection (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