Assignment 5: Neuroevolution: Neural Architecture Search (NAS)

Goal: Get familiar with neural architecture search by applying an evolutionary algorithm to select the best architecture of a neural network.

Submission: The assignment consists of two parts: implementation and analysis. You are supposed to present results for both parts in the following manner:

1. Upload your code.
2. Prepare a report with an analysis of results obtained at home.

The code and the report must be uploaded due to the deadline to Canvas.
UPLOAD A **SINGLE FILE** (a zip file) containing your code and the report. Name your file as follows: [vunetid]_[assignment number].

Introduction

In this assignment, we are going to use an evolutionary algorithm for selecting the best architecture of a neural network. As such, this assignment can be seen as a combination of assignment 3 and assignment 4. Similar to assignment 4, the neural network is trained on images.

In this task, **the code for a neural network must be implemented in PyTorch.**

Part 1: Implementation

Data Loaders

See assignment 4.

Neuroevolution for NAS

1. In this assignment, we are interested in implementing a convolutional neural network of the following form:
   Conv2d → f(.) → Pooling → Flatten → Linear 1 → f(.) → Linear 2 → Softmax
   However, we allow different choices in each building block:
   - Conv2d:
     - Number of filters: 8, 16, 32

- kernel=(3,3), stride=1, padding=1 OR kernel=(5,5), stride=1, padding=2
- f(.):
  - ReLU OR sigmoid OR tanh OR softplus OR ELU
- Pooling:
  - 2x2 OR Identity
  - Average OR Maximum
- Linear 1:
  - Number of neurons: 10, 20, 30, 40, 50, 60, 70, 80, 90, 100

Altogether, there are 4500 possible configurations.

2. Think of how to represent these options (e.g., binary variables, one-hot-representation, integers).
3. Adapt the implementation from assignment 4 accordingly so that a network could be easily created for given options.
4. Implement an evolutionary algorithm for NAS:
   a. Think of appropriate mutation, recombination, and selection for chosen representation.
   b. Modify the objective function (i.e., classification error, ClassError) by adding a penalty for the number of weights in the CNN:

      $$\text{Objective} = \text{ClassError} + \lambda \frac{N_p}{N_{max}}$$

      where N_p denotes the number of weights of a model and N_{max} is a number of weights of the largest possible neural network in the search space.
      Take $\lambda = 0.01$.
   c. Each candidate network is trained for a couple of epochs (e.g., up to 10) in order to avoid potential problems with learning times.
5. Evaluate the best-performing candidate network on the test set.

Please notice that we train a CNN using the cross-entropy, but in the evolutionary algorithm, we use the classification error on the validation set!

Part 2: Analysis

1. Compare the performance of the best performing network with FCN and CNN trained in assignment 4.
2. How can you explain the differences in performance of these three networks?

Grading the assignment: A Guide

The assignment is graded from 0 to 10 points and consists of two parts:

1. Evaluation of a code.
2. Evaluation of results and analysis.

The general structure of a report

A report must be structured as follows (**please follow the provided template**):

1. Introduction (up to 0.5 pages)
    a. Explain the problem in plain words.
    b. Explain the used methodology (i.e., an algorithm) in plain words.
2. Problem statement (up to 1 page)
    a. Formulate the problem, e.g., function minimization, neural network training (the cross-entropy minimization for given data).
    b. Specify what is the objective function.
    c. Comment on possible difficulties (e.g., no analytical form, optimization is time-consuming).
3. Methodology (up to 2 pages)
    a. Explain your approach.
    b. Define and outline all steps including mathematical formula if necessary.
    c. Provide a pseudo-code if this could be helpful.
4. Experiments (up to 1 page)
    a. Provide a detailed description of your experiment, e.g., what are the hyperparameters, what are their values, a neural network architecture, learning step value.
    b. Provide goals of the experiments, i.e., what you are going to verify.
5. Results and discussion (up to 2 pages, but it could be more)
    a. Provide all results (plots and tables).
    b. Discuss all results and check whether goals are achieved.
    c. Compare methods (if applicable).
    d. Comment on deficiencies and advantages of the proposed/used methods and think of possible extensions.
6. References (unlimited)
    a. Indicate papers/books you used for the assignment.

The report should be **up to 6 pages** long. Try to be concise and to the point. Make your statement clear and simple.

Follow the provided template! All deviations from the template could result in 0 points.

Code assessment (3 points)

You must upload the code to Canvas.

The code will be evaluated according to the following criteria:

1. Structure of the code (e.g., classes, methods). (0, 0.5 or 1 point)
2. Effectiveness of the code (e.g., exception handling, code repetition). (0, 0.5 or 1 point)
3. Cleanliness of the code (e.g., readability of the code, comments). (0, 0.5 or 1 point)

DO NOT COPY code from StackExchange or other websites. If you have any questions, please ask the TAs or the course coordinator!

Report assessment (7 points)

The report is evaluated only if the code is uploaded. **If there is no code, the report is NOT evaluated (i.e., it results in 0 points for the report).**

The report is evaluated according to the following criteria:

1. Problem statement (1 point)
    a. Is the problem properly formulated? (0 or 0.5 point)
    b. Is the problem clear enough? (0 or 0.5 point)

    Typical issues: Do not present the problem statement for a specific problem (i.e., a specific neural network) It should be general. For instance, a problem statement for a function minimization could look as follows:

    We aim at minimizing a differentiable function f(x) on a closed domain X

    min f(x)

    s.t. x in X

    Here, we consider a specific function f(x) = x^2.

    Obviously, it is a simplified version, but please do not say explicitly that you start with optimizing the quadratic function, be more general first, then you can specify your problem.
2. Methodology (3 points)
    a. Present your approach in a comprehensive manner. (0 or 0.5 point)

   b. Explain all steps of your approach. Use mathematics to avoid ambiguities. (0, 0.5, 1 or 1.5 points)
   c. Be imaginative! Try to think outside of the box. (0, 0.5 or 1 point)
3. Results and discussion (3 points)
   a. Present all results visually! Plots and tables are crucial to properly evaluate a method. (0, 0.5, 1, 1.5 or 2 points)
   b. Discuss the results, check whether the goals are achieved, and think of possible extensions of the proposed/used approach. (0, 0.5 or 1 point).