Northeastern University
CS7180 – Special Topics in Artificial Intelligence
Spring 2024 Semester, Jonathan Mwaura

Problem Set #3

This Problem Set #3 is due at **11.30PM on 19^{th},April 2024**, and will be submitted via Canvas.

This Problem Set will be marked out of 45. Question #1 is worth 20 points and Question #2 is worth 25 points

The problem set is to be completed individually.

Please type your solutions on standard 8.5×11 paper, with your name on top of the page. Ensure that you submit you solutions in one PDF file.

While a solution must be absolutely perfect to receive full marks, I will be generous in awarding partial marks for incomplete solutions that demonstrate progress.

In this class, **I encourage the exploration and use of generative AI tools** to help learn and understand the topics we are discussing. Note that learning is something the human (yourself or myself does); Its hard to measure someones learning and these assignments helps me to try to understand and see your progress in this class. Below is some guidelines on proper use of AI tools:

(a) Given a correct prompt, the AI tool can help you refine the areas of learning , "the need to knows". Think about the general topic area we are exploring and use AI tool to investigate those topics.

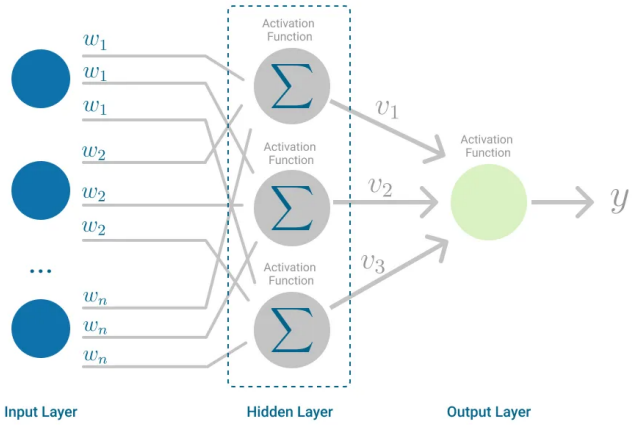
(b) You can use AI tools to measure the correctness of what you have written or to see any new suggestions you can add. If new suggestion arises try to dig deeper into the conversation with intention of generating new knowledge. Note that these tools can make mistakes, so be careful to also check references in scholarly articles.

(c) In submitting your final solutions, share (you can do this with a link) your entire chat with the AI generative tools. The instructor shall comment on your prompt engineering skills.

(d) The answers/solutions generated by AI tools are not your answers. Those are the AI tools answer. When you pass those answers to me as yourself, then this is not only an academic violation but also denies you the chance to really showcase your learning and your talent in the course.

Problem #1

In class we discussed approaches to solving supervised learning tasks using artificial neural network (ANN). One of the most common ANN approaches we saw was the multi-layered perceptron (MLP). Briefly, the MLP is an extension of the percetron by adding hidden layers (see the image below).



In this problem, we are going to use the ANN implementation (ANN.py) to analyse how the ANN works and to see how change of parameters affects the working of the algorithm. The attached code uses ANN to train a model that helps classify glass type. The glass data set: *https* : *//www.kaggle.com/datasets/uciml/glass* has been included to help you run the technique.

(a) Download the code and go through it trying to understand the implementation. Based on your analysis, describe the architecture of the ANN used. In your description , and:

   (a) State the number of input features
   (b) Number hidden layers (if any)
   (c) Number of nodes in hidden layers (if any)
   (d) Describe how the architecture selected relates to the Glass Dataset in terms of the number of input nodes and output nodes, why does this relationship matter?

   In addition to these, as seen in class some networks include a bias term, what does this bias term mean/represent? How a bias been used here?

(b) Training a ANN involves finding a set of weights that helps to transform the input to the output. One such a way to address the training is the use of back propagation (BP) algorithm. A requirement to use BP is that the function used must be differentiable. In this implementation, what activation function(s) have been used in the hidden layers (if any) and the output layer? Why is this activation function sufficient for the given problem? In addition, BP uses gradient descent to update the weights. Describe Gradient Descent and its use in BP.

(c) The performance of ANN can be affected by the parameters used in the algorithm. By considering the given implementation, what parameters can we vary/tune to ensure that we develop a model that produces higher performance (note these performance also include any parameters used in BP). Choose any two parameters and generate 10 different values for each of those parameters. Experiment the effect the parameters have on the ANN performance by running the code 10 times with each of your parameter values and record the average performance (in terms of accuracy, precision and recall) for each parameter value. You will output a table and a graph and a written analysis on effect of these? (experiment design involves varying one parameter at a time while keeping others constant).

Problem #2

Ant Colony Optimization (ACO) is a probabilistic technique, inspired by behaviour of real ants, for solving computational problems.

In this problem you are given an implementation of ACO and two TSP maps (Burma with 14 cities and Brazil with 58 cities). The code implements the standard ACO and ACO with Elitism.

Your goal is to explore how different parameters and operators affect the ACO's performance on the Traveling Salesman Problem. The experiment design involves varying one parameter at atime while keeping others constant. Your ouPotential parameters/operators to explore include:

- ACO approach variations (e.g., Max–Min Ant System, ACO with Elitism etc -)..
- Evaporation rate variations (e.g.0.1,0.2,0.9 etc).
- Colony size variations (e.g., 10, 50, 100).
- Local heuristic function variations (e.g., 1/d , Q/d etc)
- Local search methods (e.g. Hill Climbing, Simulated Annealing etc)

Each experiment involves running the ACO algorithm with a specific parameter or operator variation and observing its impact on the algorithm's performance in finding optimal solutions for the Traveling Salesman Problem in the given city datasets.

Using the following questions as a guide, describe your observations of the results and explanations or conclusions you can draw from them. In addition, you should describe any further experiments you conducted to better answer the questions below and to demonstrate your understanding of the parameters used in the ACO.

(a) Which combination of parameters produces the best results?

(b) What do you think is the reason for your findings in Question 1?

(c) How does each of the parameter settings influence the performance of the algorithm?.

(d) Can you think if any variation for this algorithm to improve your results? Explain your answer.

(e) Do you think of any any other nature inspired algorithms that might have provided better results? Explain your answer.

Note: You should think carefully about how best to present your results to show the behaviours of the algorithm during your experiments.

For extra guidance, please see next page (Appendix A for guidance on ACO. Also see ACO materials in Canvas.

Appendix A - The Ant Colony Optimisation Algorithm

From a modelling perspective, ACO uses a construction graph to hold the pheromone values for each decision made by the ants. This graph reflects the paths and associated pheromone levels for every possible city-to-city movement. The ACO algorithm can then be structured as follows

- Random Pheromone Initialization: Initialize the construction graph by randomly assigning small amounts of pheromone (value between 0 and 1) on the edges representing the connections between cities.
- Ant Path Generation: Generate a set of:
  - m ant paths from the starting node.
  - Each ant will traverse the construction graph, making decisions at each city based on the amount of pheromone on potential paths and a heuristic function.
  - The selection will be biased by the amount of pheromone on the paths ahead and the heuristic information.
  - Ants move to the next city until all cities are visited.
- Pheromone Update:
  - After an ant completes its path, update the pheromone in the construction graph based on its fitness.
  - Update the pheromone according to the formula $\frac{Q}{fitness}$ to reward paths that decrease the fitness function. Here, Q is a parameter.
- Pheromone Evaporation: Evaporate the pheromone for all links in the graph by multiplying all pheromone values by the evaporation rate e.
- Termination Criterion: Terminate the algorithm when a maximum number of fitness evaluations (e.g. $10,000$) is reached. The result will be the best fitness found so far by the ants. Here the "max number of evaluations" is a parameter.

Acknowledgement: Question 2 originally formulated by a former college mate, Prof David Walker over at University of Exeter, UK. The code is written by his former student,