Using Simulated Annealing With an OR Gate

Grant Hughes

February 20, 2025

1 Introduction

This project aims to optimize the weights of an artificial neuron that represents an OR gate using the Simulated Annealing (SA) algorithm. This is accomplished by tracing an algorithm and implementing it using the python coding language Python. The goal is to find the optimal weight values that minimize the error in predicting the output of the OR gate.

2 Methodology

2.1 OR Gate Representation

The OR gate is a basic Boolean function with the following truth table:

Input X	Input Y	Output(X OR Y)
0	0	0 0
1	1 1	0
1 1	1	1 height

The neuron has three weights: (for input X), (for input Y), and (bias). The output is calculated using the weighted sum and an activation function.

2.2 Simulated Annealing Algorithm

We implemented the SA algorithm as described in the textbook (Figure 4.5 in the book). The main steps include the following.

1. Randomly initialize the weights.
2. Compute the error function.
3. Set an initial temperature and define a cooling schedule.
4. Generate new weights by applying small disruption.
5. Calculate the error change.
6. Accept or reject the new weights on the basis of the acceptance probability.
7. Repeat until the stopping condition is met.

2.3 Google Sheets Tracing

The SA process was first traced in Google Sheets, logging iterations, temperature, weights, error, and acceptance probability to help visualize optimization behavior. This is to help us understand how the algorithm works.

3 Implementation

The algorithm was implemented in Python, strictly following the textbook steps. Key components include:

- Random weight initialization.
- Temperature schedule.
- Objective function evaluation.
- Iterative optimization with SA rules.

4 Results and Discussion

The optimization results were analyzed by iteration (Figure 1):

- Temperature vs. Iterations
- Delta E vs. Iterations
- Objective Function vs. Iterations

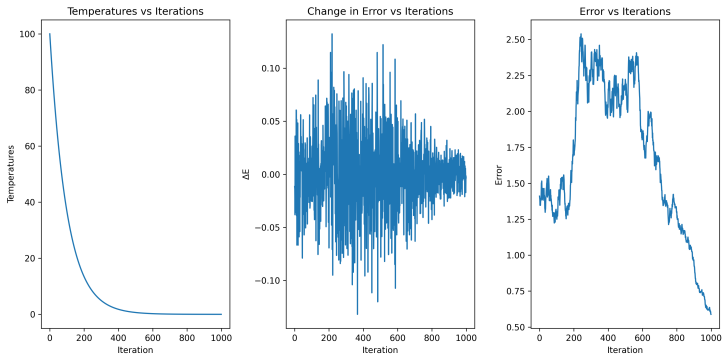

Figure 1: Temperature, Delta E, and Objective Function vs Iterations.

Some challenges encountered while understanding the tracing of SA were: 1) I found it extremely confusing to understand the values and how they relate to one another. 2) Changing the AND data to OR data was not easy. It took time understanding how to calculate the weights, values, error, and so on. 3) Getting the shape() correct during code implementation created confusion.

5 Conclusion

Simulated Annealing effectively optimized the OR gate by refining weight values over iterations. The choice of temperature schedule significantly influenced convergence speed and accuracy.