Assignment 5 - Keyboard Optimization

Sreeram R EE23B075

October 13, 2024

1 Introduction

The goal of this assignment is to create a Python program that generates an optimal keyboard layout for a given input string using optimization algorithms like simulated Annealing. The program generates a heat map for the input string on the optimized layout and also generates an animations showing the distance cost as a function of number of iterations.

2 Code and Logic Explanation

The QWERTY layout used is a modified version of the layout (coordinates have been changed) given in the problem statement which is attached in the code itself in the form of dictionaries `keys` and `characters`. The problem statement can be split into 4 parts :

- Distance calculation
- Optimizing the layout using simulated annealing
- Heat map generation for the input text on the optimized layout
- Creating an animation showing distance cost as a function of number of iterations

,

2.1 Distance Calculation

The function `calc_dist` uses the Euclidean distance formula for calculating the distance between the home row key and the key typed. The modified layout contains two dictionaries `keys` and `characters`. The dictionary `keys` contain various characters, their position, and the home row keys from which the distance travelled to type the particular character has to be calculated. The `characters` dictionary stores the characters and the keys that need to be pressed to type that character.

Now before calculating the euclidean distance, in order to access the coordinates easily and also to make it easy to swap the keys later on while implementing simulated annealing, the `start` key of the dictionary `keys` which initially stored the home row key , is now modified so that it stores the coordinates of the home row key. The advantages are as mentioned, the ease of accessing the coordinates and more importantly when we implement simulated annealing which is explained in the optimizing layout section.

2.2 Optimizing the layout

Simulated annealing is a probabilistic optimization algorithm that tries out various possible layouts by making random changes to the keyboard layout (swapping two keys). It attempts to minimize the total distance the fingers travel to type the input text. The function `get_new_layout` selects two keys at random, swaps them and generates a new layout. It is made sure that the non special keys are not included because, their key sizes are not the same and rearranging them might mess up the heatmap.

The `simulated_annealing` function takes the layout, calls the `get_new_layout` function in order to get a new layout and then calculates the distance travelled before swapping and after swapping and store them in arrays, later used for plotting. It iteratively does this a large number of times till you get the optimized layout. Hyperparameter tuning resulted in the following: `initial_temp` = 1000, `cooling_rate` = 0.98, `num_iterations`= 1000.

2.3 Heatmap Generation

The `draw_key` function draws the keyboard layout using the Matplotlib library based on the coordinate system. The `normalize` function transforms the frequency values to a range of [0, 1], linking each key's color to its relative frequency, ensuring that higher frequencies correspond to more intense colors. The `counter` function returns the frequency of a character in the input text. The `create_heatmap` function creates a color mapping system using the normalize function to scale key frequencies between 0 and 1. It then iterates through the main keys, plotting each with a color that corresponds to its frequency using the `draw_key` function.

2.4 Creating Animation

To visualize the optimized layout, an animated plot is generated. The `update` function is called repeatedly by the animation to update the plot for each frame. It modifies two plot lines: one showing the best distance found so far and another showing the current distance for the layout being evaluated at each iteration. As the animation progresses, it updates these lines with the data up to the current iteration. The `main` function sets up the animations, calls the `update` function iteratively, to update the data.

3 Results

Sample Text 1:

Artificial Intelligence (AI) and Machine Learning (ML) are transformative technologies reshaping various industries. AI encompasses systems that simulate human intelligence, enabling machines to perform tasks such as reasoning, learning, and problem-solving. Within AI, ML focuses on algorithms that allow computers to learn from data and improve their performance over time without explicit programming. This capability drives innovations in fields like healthcare, finance, and autonomous vehicles, where predictive analytics and pattern recognition enhance decision-making and efficiency. As AI and ML continue to evolve, they promise to unlock new opportunities and challenges, influencing the future of technology and society.

Distance traveled in the QWERTY layout with modified coordinates = 356.52449 whereas before optimization it was 767.97589.

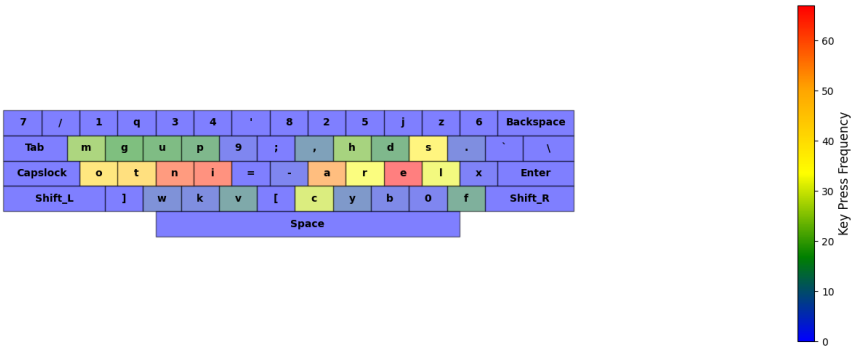

Figure 1: Heatmap of keyboard usage for Sample Text 1.

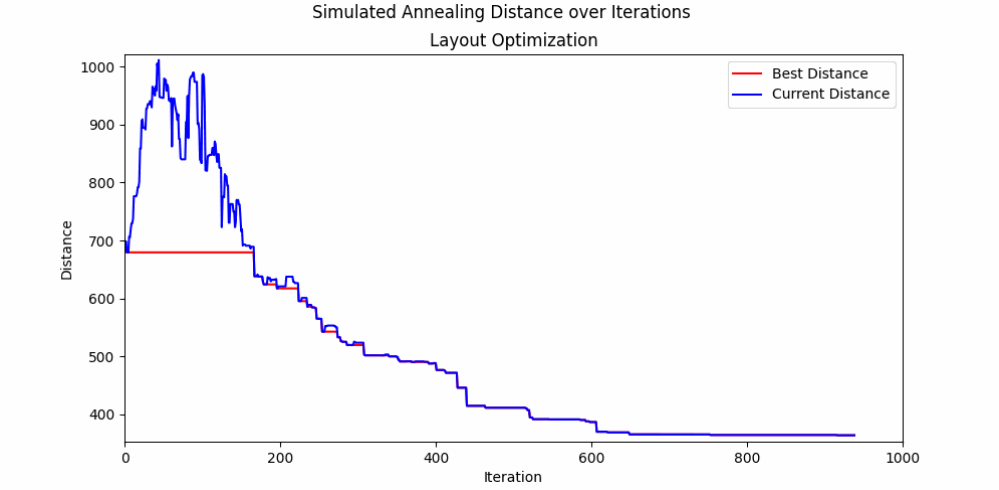

Figure 2: Animation plot of the given text for the optimized layout

4 Observation

- On applying simulated annealing optimization algorithm, the results are really impressive,as in this case for the above sample text the distance travelled on the optimized layout is half that of the distance traveeled in the modified QWERTY layout.
- From the animation it can clearly be observed that on large number of iterations, the distance cost function decreases and eventually saturates.

5 Note :

The code takes a string as input and gives the calculated distance and the heatmap (saved as keyboard_heatmap.png) and the animation plot as animation.gif for the QWERTY layout with modified coordinates as the output.