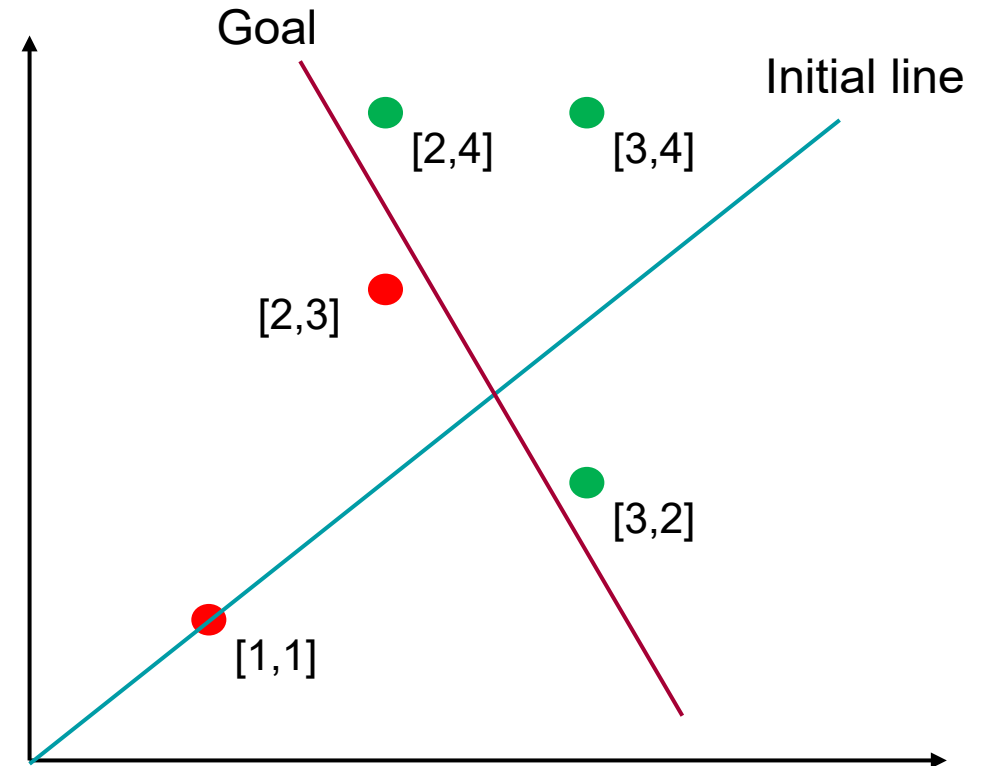


# Project 4

Implement a perceptron learning algorithm for data classification:

1. Use a constant learning rate of 1 and run for 100 iterations.
2. Use the `accuracy_score` from `sklearn` to evaluate after each iteration
3. Plot the `accuracy_score` and each of the three weights over the 100 iterations
4. The implementation above will not converge. Implement a second version of perceptron learning with a learning rate that decays over time in the form of  $c/(t+c)$ . Choose an appropriate  $c$  that converges within 100 iterations.
5. Repeat #2 and #3 for the improved version



# Submission

Use the provided Python notebook and add your implementation code to the specified areas. Submit one notebook file containing your implementation code, output, and documentation. Provide ample comments to explain your implementation.

## Output Requirements:

1. **The plots as specified**
2. **A brief discussion on how you arrived at the appropriate learning rate constant and what you learned in the process. Reverse engineering from the target line is not an acceptable approach.**

Each student should complete the project individually.

You can use utility packages, but you must implement the core algorithm of perceptron learning by yourself.

# Grading:

- 50% for successful execution and correct output for the initial implementation
- 30% for the improved implementation. (Must converge)
- 20% for the discussion
- Extra credit: up to 20% if you animate how the line evolves over the learning process for both implementations