

## **Unit-1**

### **Lab experiments**

**A financial institution wants to optimize its investment strategy.**

#### **Aim:**

A financial institution wants to optimize its investment strategy. Use a basic policy gradient method to simulate and optimize the investment policy for maximum returns. Implement this in Python.

#### **Algorithm:**

1. Initialize policy parameters  $\theta$  randomly
2. For each episode:
  - o Observe current state  $s$  o Select action  $a$  using softmax policy  $\pi_\theta(a|s)$
  - o Receive reward  $r$
3. Update policy parameters using:
$$\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi_\theta(a | s) \cdot r$$
4. Repeat until policy converges
5. Output the learned policy probabilities

#### **Code Github Link:**

<https://github.com/syekumar/MLA0316-Reinforcement-learning->

## Output:

```
Optimized Investment Policy:
```

```
State 0 (Market): [0.93 0.07]
```

```
State 1 (Market): [0.18 0.82]
```

## Result:

The policy gradient algorithm successfully learns:

- **Risk-averse behavior in bad markets**
- **Risk-seeking behavior in good markets**

This demonstrates how policy gradients directly optimize decision-making policies for maximum returns.