Reinforcement Learning

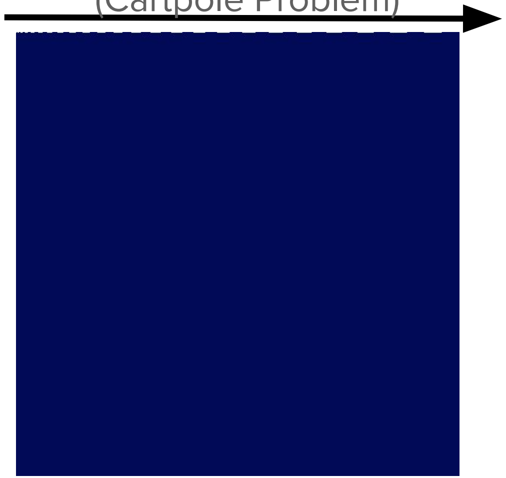
Alexis W. Mills, Daniel Chai, Weishi Yan

Reinforcement Learning Overview

The system is made up of 5 key components:

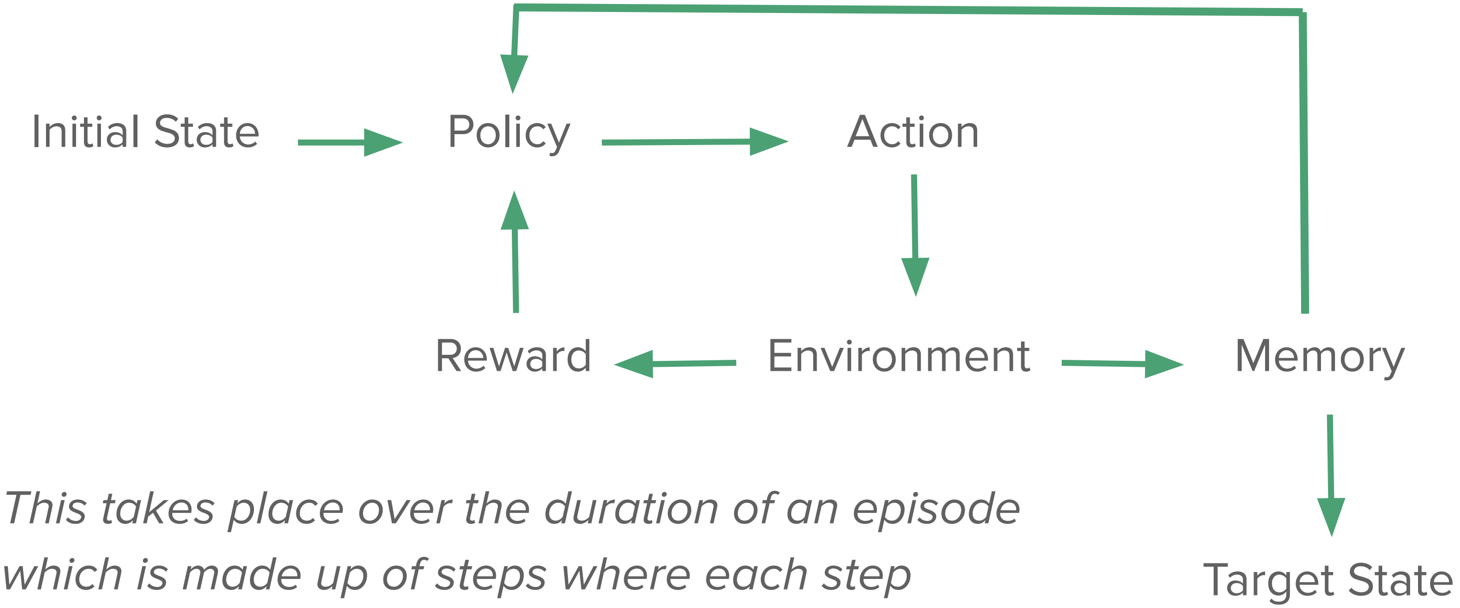
1. Agent
2. Environment
3. Action
4. Reward
5. State

(Cartpole Problem)



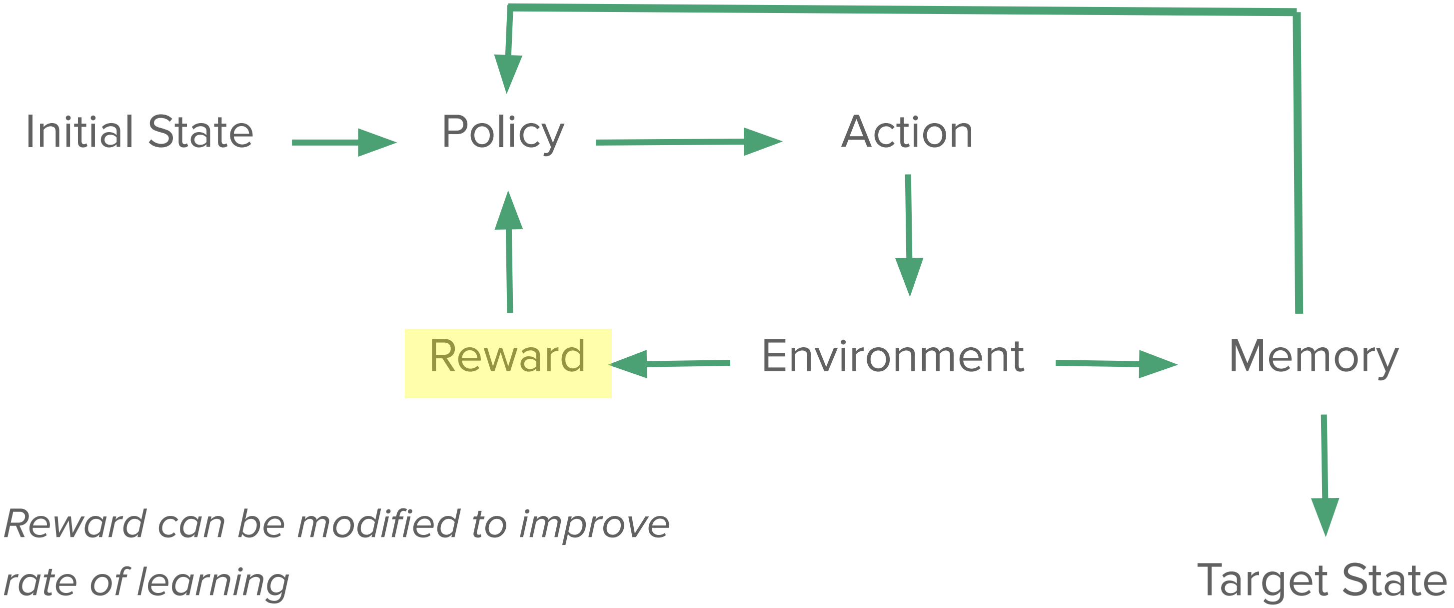
1. Controls the cart (black square)
2. Angle of pole and position of cart
3. Move in positive or negative direction along 1D surface
4. Reward=1 if agent has not failed (moved cart beyond threshold or allow pole angle to fall beyond threshold)
5. Consists of cart position, cart velocity, pole angle, and pole velocity at tip

Learning Algorithm



This takes place over the duration of an episode which is made up of steps where each step allows the agent to take an action

Learning Algorithm

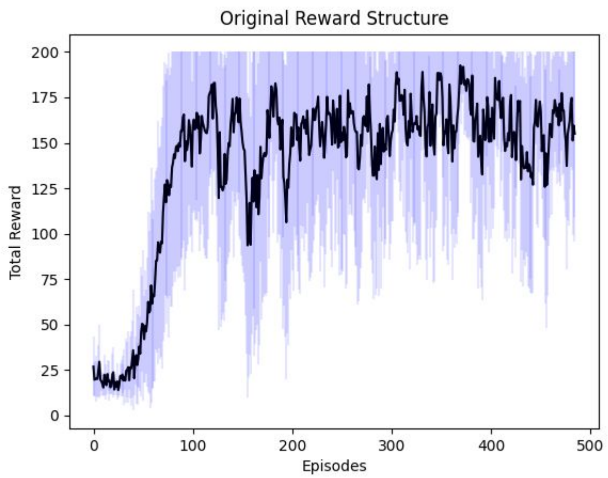


Reward can be modified to improve rate of learning

Reward Structure

Original

A reward of value 1.0 is granted so long as the pole angle is -41.8° < θ < 41.8° and the cart position is -2.4 < x < 2.4



Modified

The reward value is varied depending on how close the agent comes to failing during the step

Calculating the Reward:

$\boxed{\theta\_r = (\theta\_t - \theta\_a)/\theta\_t}$

ϴ_r: ϴ reward

ϴ_t: 41.8°

ϴ_a: Absolute value of ϴ angle (from current state)

$\boxed{x\_r = (x\_t - x\_a)/x\_t}$

x_r: x reward

x_t: 2.4

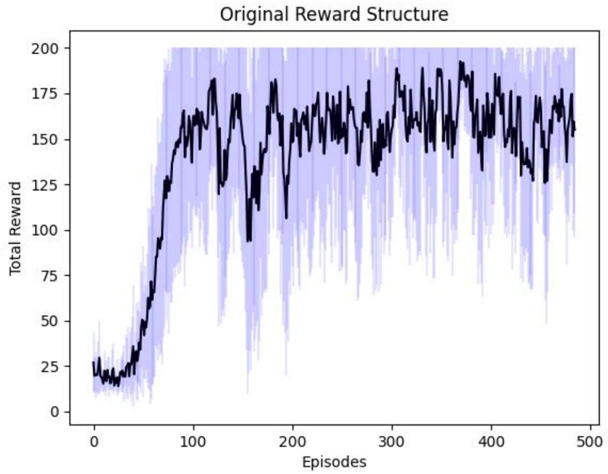
x_a: Absolute value of x position (from current state)

$\boxed{\textit{Modified reward} = (\theta\_r + x\_r)/2}$

Reward Structure

Original

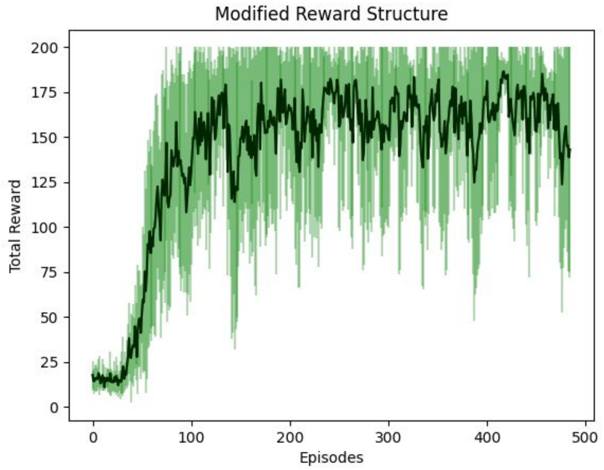
A reward of value 1.0 is granted so long as the pole angle is -41.8° < θ < 41.8° and the cart position is -2.4 < x < 2.4



Modified

The reward value is varied depending on how close the agent comes to failing during the step

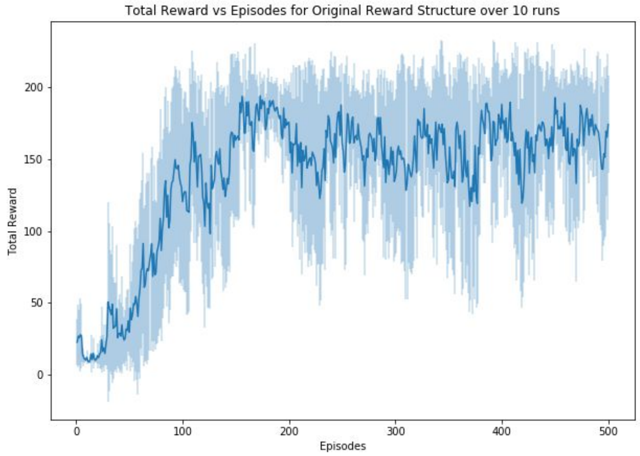
Modified reward = (Θ_r + x_r)/2



Reward Structure

Original

A reward of value 1.0 is granted so long as the pole angle is -41.8° < θ < 41.8° and the cart position is -2.4 < x < 2.4



Modified_version2

Since the goal of the problem is to have the pole stay upright as long as possible, physics equation was used to calculate the time the pole would stay up.

Using this physics equation:

x = x_initial + v * t

Transformed into:

Θ_time = (Θ_threshold - abs(Θ)) / Θ_dot

Reward Calculation - hyperbolic tangent (tanh):

Θ_reward = abs(tanh(Θ_time))

Same thing for x and calculate total reward :

total_reward = 0.3*x_reward + 0.7*Θ_reward

Reward Structure

Correlation Matrix

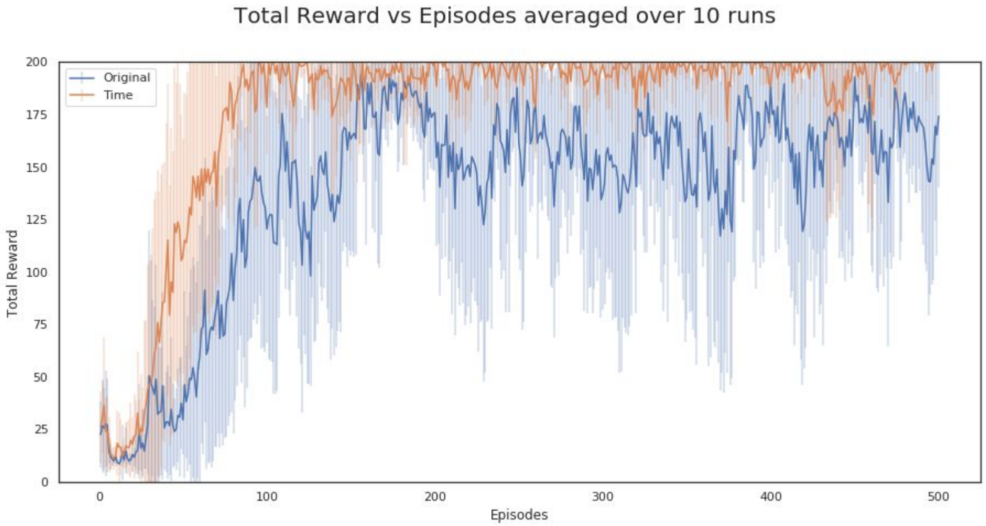
The values of 0.3 and 0.7 were retrieved from correlation matrix calculated from the original reward structure.
Absolute values of x-related terms and theta-related terms were combined to find the percentage of the total.
Example:

x_coefficient = (0.56 + 0.05) / 2.06 ≈ 0.3

	x	x_dot	theta	theta_dot	Total Reward
x	1.000000	0.685613	-0.209230	-0.524482	0.564305
x_dot	0.685613	1.000000	0.333420	0.004916	0.051151
theta	-0.209230	0.333420	1.000000	0.477523	-0.760483
theta_dot	-0.524482	0.004916	0.477523	1.000000	-0.682016
Total Reward	0.564305	0.051151	-0.760483	-0.682016	1.000000

Modified_version2

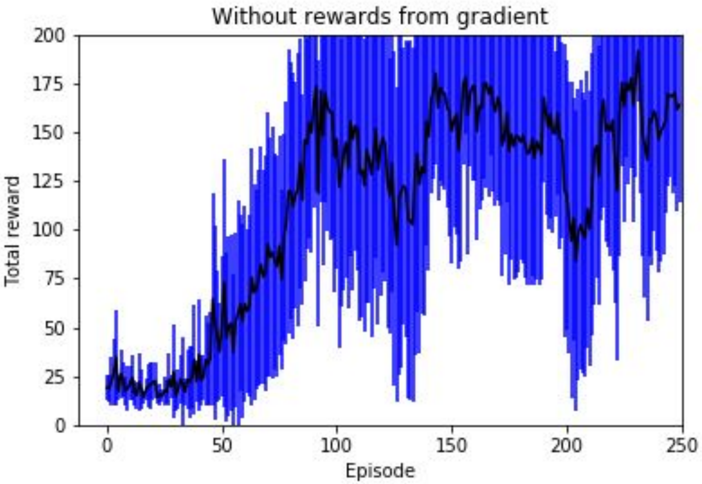
Modified reward = 0.3*x_r + 0.7*Θ_r



Reward Structure

Original

A reward of value 1.0 is granted so long as the pole angle is -41.8° < θ < 41.8° and the cart position is -2.4 < x < 2.4

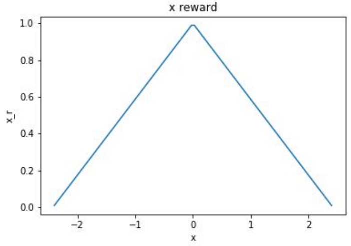


Modified

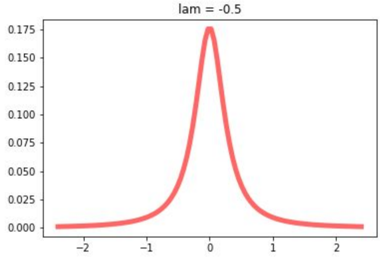
The reward value is varied depending on how close the agent comes to failing during the step

Calculating the Reward:

Θ_r and x_r:
Triangular Function



Θ_dot_r and x_dot_r:
Based on lam=-0.5 tukey lambda distribution

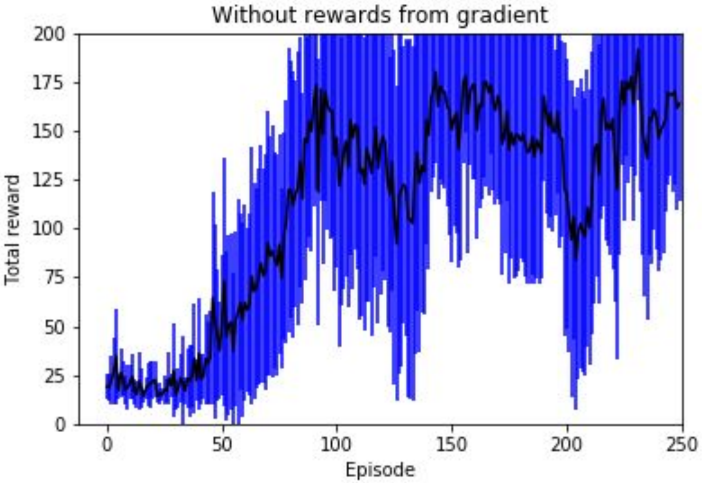


Modified reward = *x * Θ + Θ_r * x_r*

Reward Structure

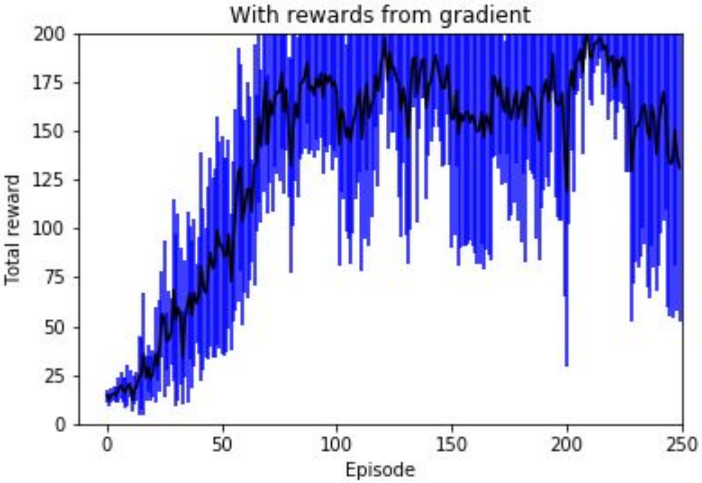
Original

A reward of value 1.0 is granted so long as the pole angle is -41.8° < θ < 41.8° and the cart position is -2.4 < x < 2.4



Modified

The reward value is varied depending on how close the agent comes to failing during the step



*Modified reward = x * Θ * Θ_r * x_r*

Conclusion

- Convergence and stability of the model improved with modified reward structure
- Incorporation of physics into the neural network may improve the results significantly