

Student details: Apurv Mishra (1006695211)

Assignment IV: Function Approximation and SARSA control

Environment used: **CART-POLE V1**

Algorithm being used is from the section 10.1 from the text, i.e. **Episodic Semi-gradient SARSA for estimating q^*** .

Parameters, hyperparameters and NN model:

- alpha or learning rate = 0.001
- gamma or discount factor = 0.9
- epsilon for policy = 0.1
- layers in the neural network = 5
- hidden terms per hidden layer = 100
- input terms = 4, for each state
- output terms = 2, for each action
- optimizer used: SGD
- error used: Mean Squared Error

```
In [6]: '''code is in the solver.py file'''
```

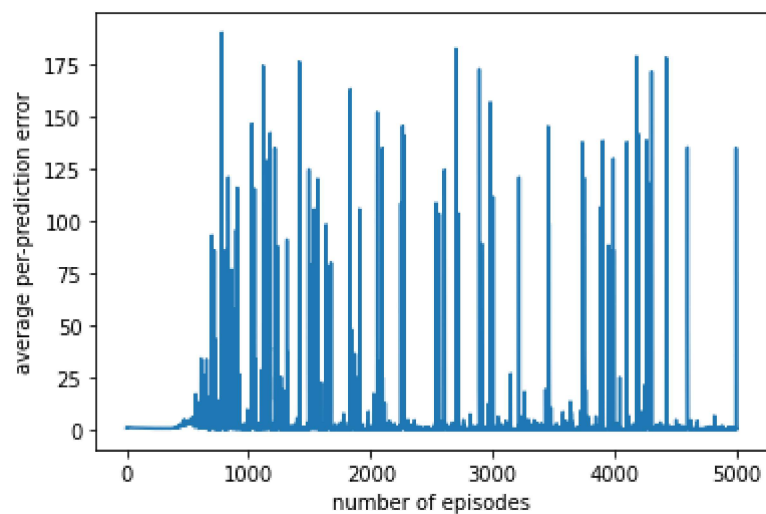
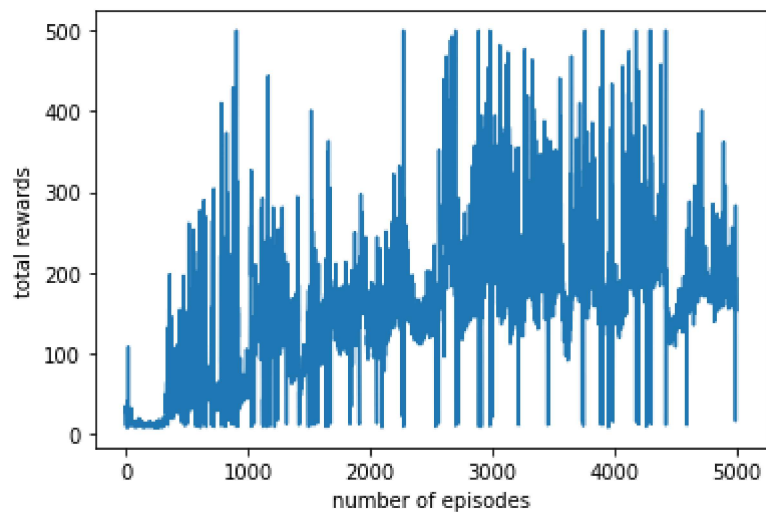
Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 100)	500
dense_1 (Dense)	(None, 100)	10100
dense_2 (Dense)	(None, 100)	10100
dense_3 (Dense)	(None, 2)	202

Total params: 20,902
Trainable params: 20,902
Non-trainable params: 0

Function approximation and control policy using semi-gradient SARSA

```
In [9]: '''code is in the solver.py file'''
```



For semi-gradient SARSA:
Total episodes = 5001 with average steps = 162

The above result is for 5000 episodes.

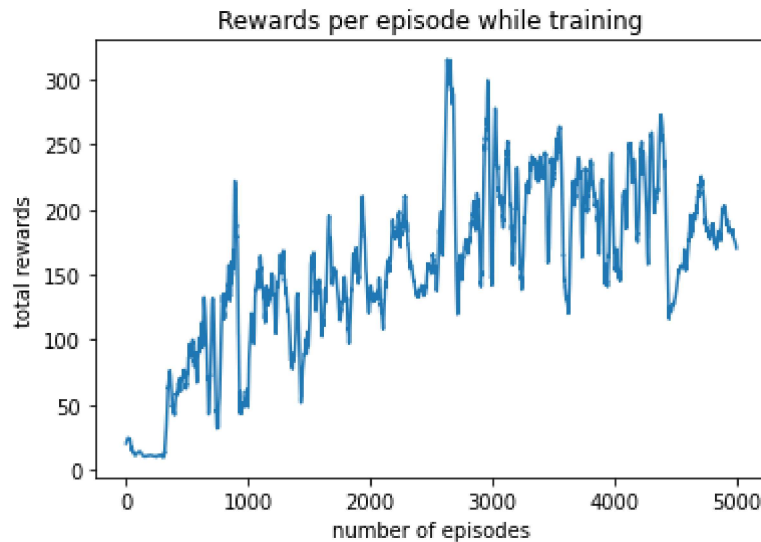
- first plot is for rewards per episode
- second plot is for mean per-prediction error per episode

Smoothened rewards plot against episodes:

```
In [23]: from scipy.signal import savgol_filter

smoothened_rewards = savgol_filter(reward_list, 55, 3)
plt.figure()
plt.plot(smoothened_rewards)
plt.xlabel("number of episodes")
plt.ylabel("total rewards")
plt.title("Rewards per episode while training")
```

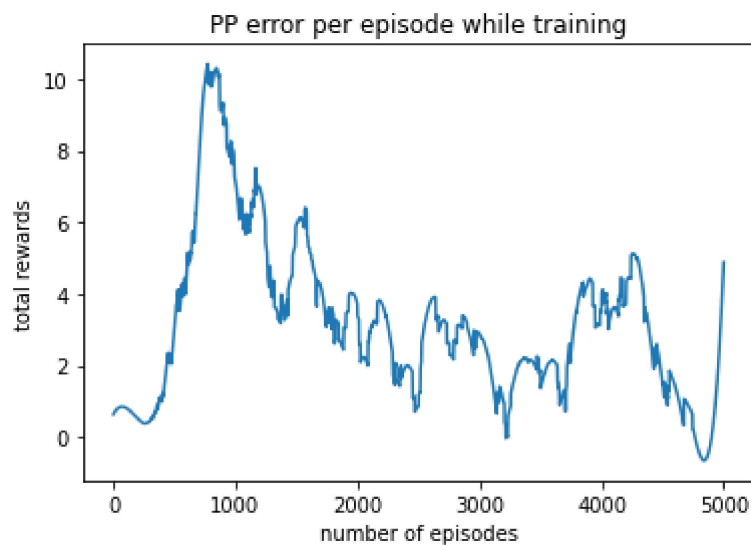
Out[23]: Text(0.5, 1.0, 'Rewards per episode while training')



Smoothened per-prediction error plot against episodes:


```
In [30]: smoothened_rewards = savgol_filter(pp_err_list, 501, 3)
plt.figure()
plt.plot(smoothened_rewards)
plt.xlabel("number of episodes")
plt.ylabel("total rewards")
plt.title("PP error per episode while training")
```

Out[30]: Text(0.5, 1.0, 'PP error per episode while training')



So, it can be observed that rewards are increasing and the per-prediction error is reducing with the progression in number episodes.

If more episodes were run, we would have gotten better performance.

```
In [33]:  jupyter nbconvert --to html /content/drive/MyDrive/Colab_Notebooks/APS1080/results.ipynb
```

```
[NbConvertApp] Converting notebook /content/drive/MyDrive/Colab_Notebooks/APS1080/results.ipynb to html
```

```
[NbConvertApp] Writing 387191 bytes to /content/drive/MyDrive/Colab_Notebooks/APS1080/results.html
```