**NN Description**

Mj2 (No backward hu check), ai not chow ing or pong ing

Mj3 (backward hu plus rewards for draw in the game), same issue as above, realised that when chow/pong happens, current player don’t change, thus resulting in the wrong agent responding to the call for chow/pong, thus leading to the AI pressing no to chow/pong most of the time.

Mj4 (increased wining reward to 2 and reduced chow pong reward)

Mj5 (separated player exposedtiles from ComputeOthers)

Mj6 (Insert game phase information into observation buffer)

**Plan for AI Training**

1. Do up trainer file for AI training
2. Plan for training-reward function
3. Conversion of input data types to facilitate AI training
4. Implement reward function into the game

**Trainer File**

Hyperparameters:

batch\_size: 256

buffer\_size: 4096

learning\_rate: 3e-4

epsilon: 0.15

lambd: 0.95

num\_epoch: 6

learning\_rate\_schedule: constant

Network Settings:

normalize: false

hidden\_units: 256

num\_layers: 2

vis\_encode\_type: simple

reward\_signals:

extrinsic:

gamma: 0.99

strength: 1.0

keep\_checkpoints: 5

max\_steps: 10000000

time\_horizon: 32

summary\_freq: 1000

threaded: true

self\_play:

save\_steps: 20000

team\_change: 100000

swap\_steps: 10000

window: 20

play\_against\_latest\_model\_ratio: 0.5

initial\_elo: 1000.0

**Reward function plan (Start off training)**

Win/Lose (Require further discussion): +1f/-0.5f

Pong: +0.05f

Kong: +0.15f

Chow: +0.05f

Draw: +0.01f

How much should chow pong kong be worth?.

Penalties for keeping/waiting dead tiles (Require further discussion):

mlagents-learn trainingConfig\mj.yaml --env=env\MJ.exe --num-envs=9 --run-id=mj3

**ML Agents Data Types**

<https://github.com/Unity-Technologies/ml-agents/blob/master/docs/Learning-Environment-Design-Agents.md#agentcollectobservations>

To check whether the logic set for the AI can be used well, use the Heuristics() method to test it out (eg. 0 to chow set 1, 1 to chow set 2 etc).

If Mahjong is considered an asymmetric game, more trainer files are required due to adversarial conditions.

CollectObservations():

The agent should be able to see their own hidden tiles and everyone’s exposed tiles. Observations should be normalised either in the range of [-1, +1] or [0, 1] to enable faster training time (Think this is not possible since the tiles themselves have unique IDs). Stacking the observations is most likely required since playerHidden and playerExposed are part of a set of a larger observation that the AI should be aware of.

All observations can be pushed from an array since they are int variables.

CollectDiscreteActionMasks():

Don’t think this is required because there are no actions that the AI cannot take since everything is handled by the game logic.

OnActionRecieved():

How to let the AI know the difference between chow, pong and kong?

Actions can be discrete but the AI must be able to tell the difference between chow pong and kong.

Output wil be in form of int, in order to “select” the choice.