Deep Learning Carom Billiards

Mrinal Sourav, Northeastern University - Seattle

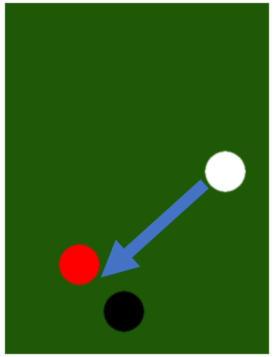
1. Objective:

- Train a Neural Network to play Carom Billiards.
- Observe the limits of the model's generalization.

2. Video Demo:

[YouTube search with title]

3. Simulation for Q-Learning:



State: X and Y coordinates of each ball. Future states are not dependent on the current state

Action: Angle and Speed for the q-ball

Q(Quality)-Value: Reward earned from applying the action on the state

- 0 for no collision
- 10 for collision with one other ball
- 100 for collision with both the other balls

Game round starts when all balls are frozen. Game round ends when all balls are frozen.

4. Key Challenges:

- Capturing a balanced dataset from random states and actions.
- Action space of 3600 : 360 angles and 10 speeds.
- Non-Linear separation.

5. Traditional approach to Q-Learning:

q_ball.x
q_ball.y
red_ball.x
red_ball.y
black_ball.x
black_ball.y

Neural Net

3600 Possible Actions:

Angle1, Speed1 + q_value
Angle1, Speed2 + q_value
.
.
.
Angle360, Speed10 + q_value

6. Approach Used:

q_ball.x
q_ball.y
red_ball.x
red_ball.y
black_ball.x
black_ball.y
+
q_ball.angle
q_ball.speed

Neural Net

q_value | action

7. During Test:

q_ball.x
q_ball.y
red_ball.x
red_ball.y
black_ball.x
black_ball.y
+
3600 Possible Actions:

Angle1, Speed1
Angle1, Speed2
.
Angle360, Speed10

Neural Net

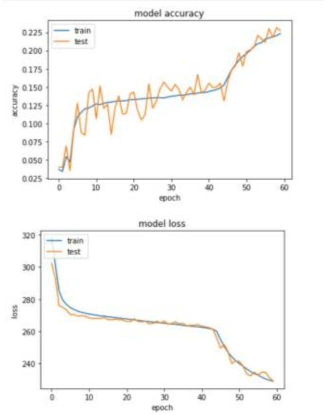
q_value1
q_value2
q_value3
.
.
.
.
q_value3600

Concatenate and sort to select the action with the highest q_value

8. Model & Training:

```
model = Sequential()
model.add(Dense(8, input_dim=8, activation='relu'))
model.add(Dense(10, activation='relu'))
model.add(Dense(100, activation='relu'))
model.add(Dense(300, activation='relu'))
model.add(Dense(200, activation='relu'))
model.add(Dense(50, activation='relu'))
model.add(Dense(20, activation='relu'))
model.add(Dense(5, activation='relu'))
model.add(Dense(1, activation='relu'))

model.compile(loss='mean_squared_error', optimizer='nadam',
```



9. Key take-aways:

- It is easy to land up with a non linear system.
- The real "black-box" here is the non-linear, real-world data; not the model.