Let’s Design Neural Networks for Playing Pong!

Exercise 1

Experiment with this neural network architecture and document your experiments:
Note: Always train for around 1000 epochs (top right of the page).

Input Features	Network Structure	Training Loss	Test Loss
x_1, x_2	[2, 2, 2]		

Exercise 2

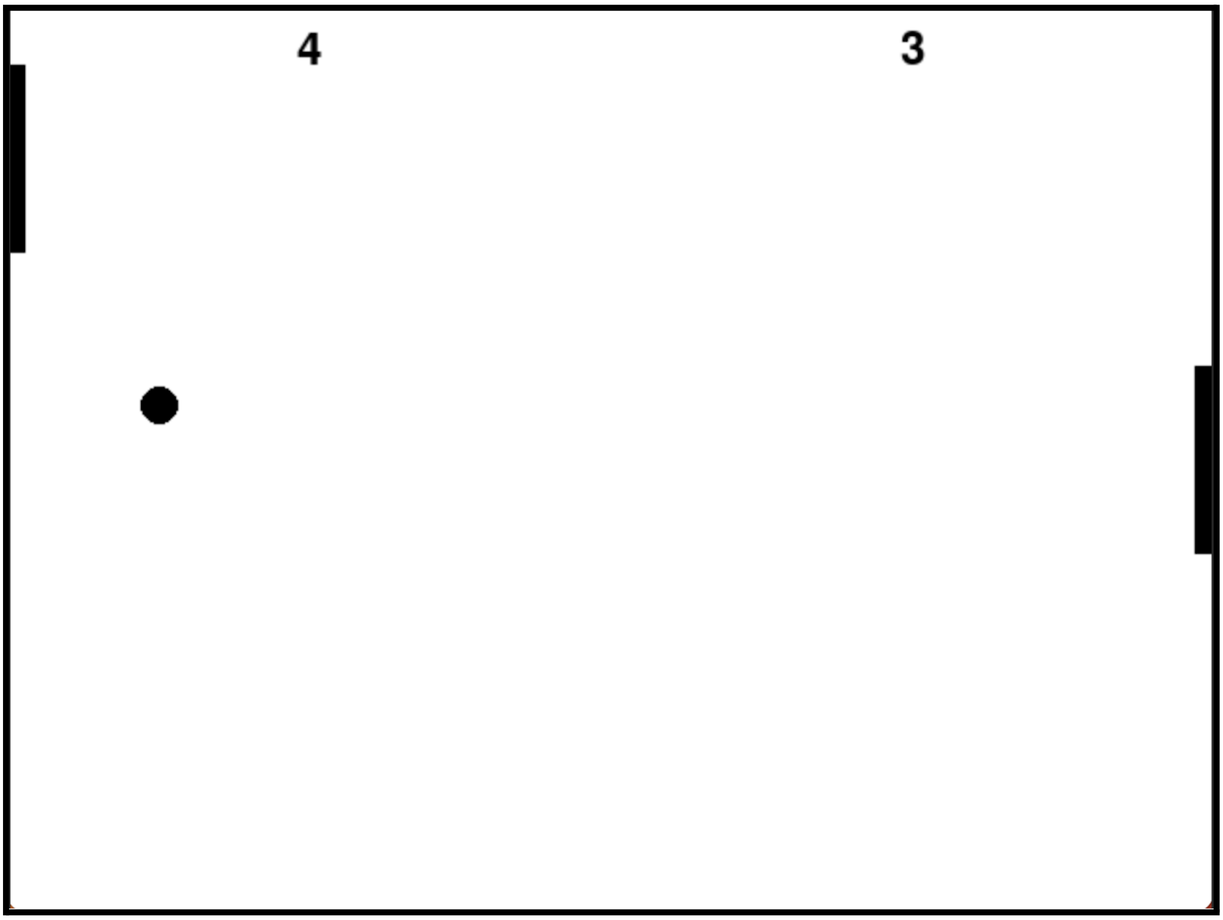
Write down three hypotheses on how the network structure influences the abilities of the neural network. For each, do you think these hold in general or just for this example?

1.

2.

3.

Exercise 3 (before “Example: Implementing a strategy”)
Take a look at the Pong class. Mark all attributes you can find on the drawing below.
Hint: The top left has coordinates (0, 0) and the bottom right the coordinates (width, height).



Exercise 4 (after “reward strategy”)
Could your reward function be problematic if one of the following changes is applied? Shortly state why you think it would (not) be problematic! If possible propose a fix!

1. The ball would move way faster than it is now.

2. You could influence the reflection angle by hitting the ball with different parts of your paddle.

Exercise 5 (after “state strategy”)

Could your get_state function be problematic if one of the following changes is applied? Shortly state why you think it would (not) be problematic! If possible propose a fix!

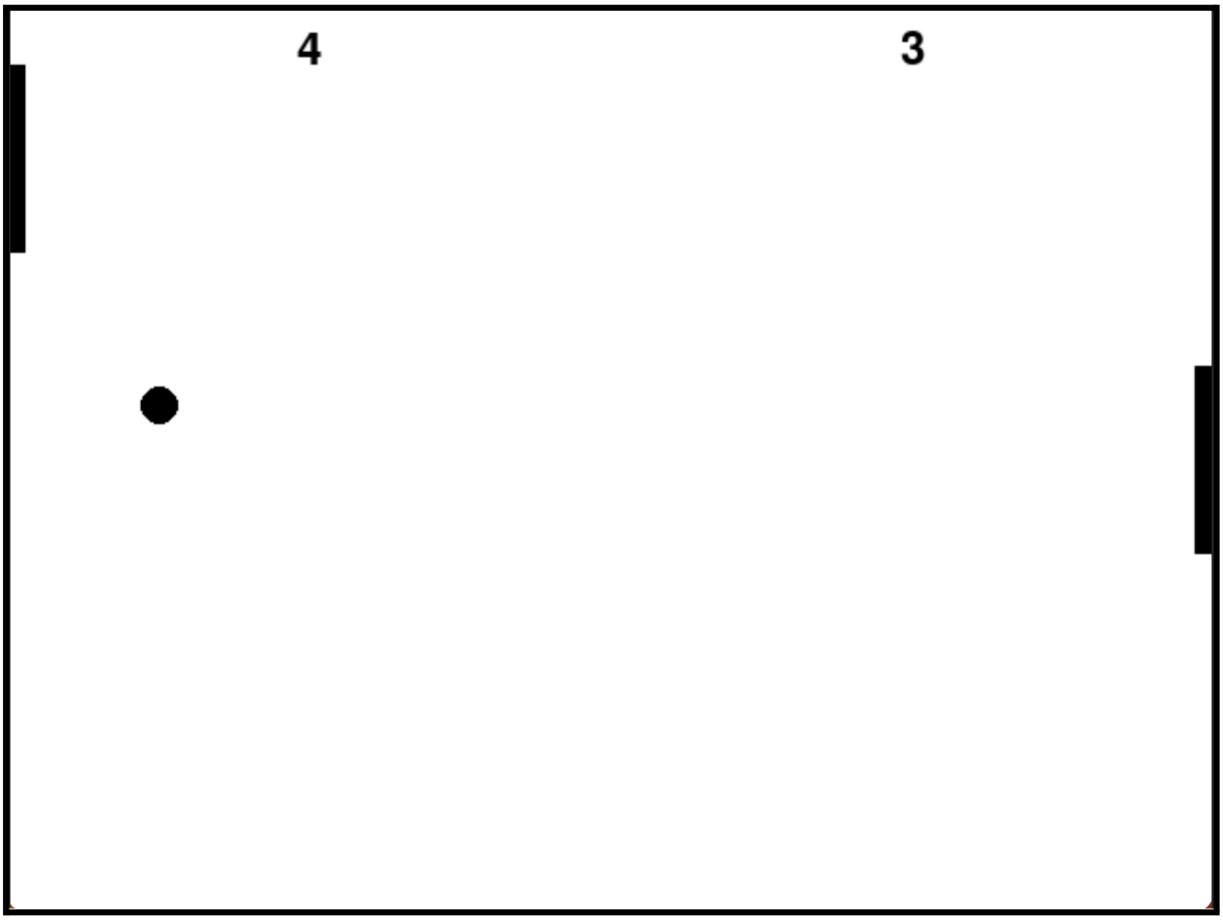
1. You could influence the reflection angle by hitting the ball with different parts of your paddle.

2. The ball's physics / flight behaviour changes over time.

Exercise 6 (after “network structure strategy”)

In the beginning we briefly mention overfitting (“the model remembering individual things by heart, instead of learning patterns”). Is reinforcement learning prone to overfitting, too? If no, shortly argue why. If yes, give a short example.

Exercise 7 (before doing the “transforming observations” implementation)
Perform the transformation by hand. i.e. transform the board state such that each player “thinks” they are on the other side.



Now choose an object. Explain how to calculate its new coordinates after the swap: