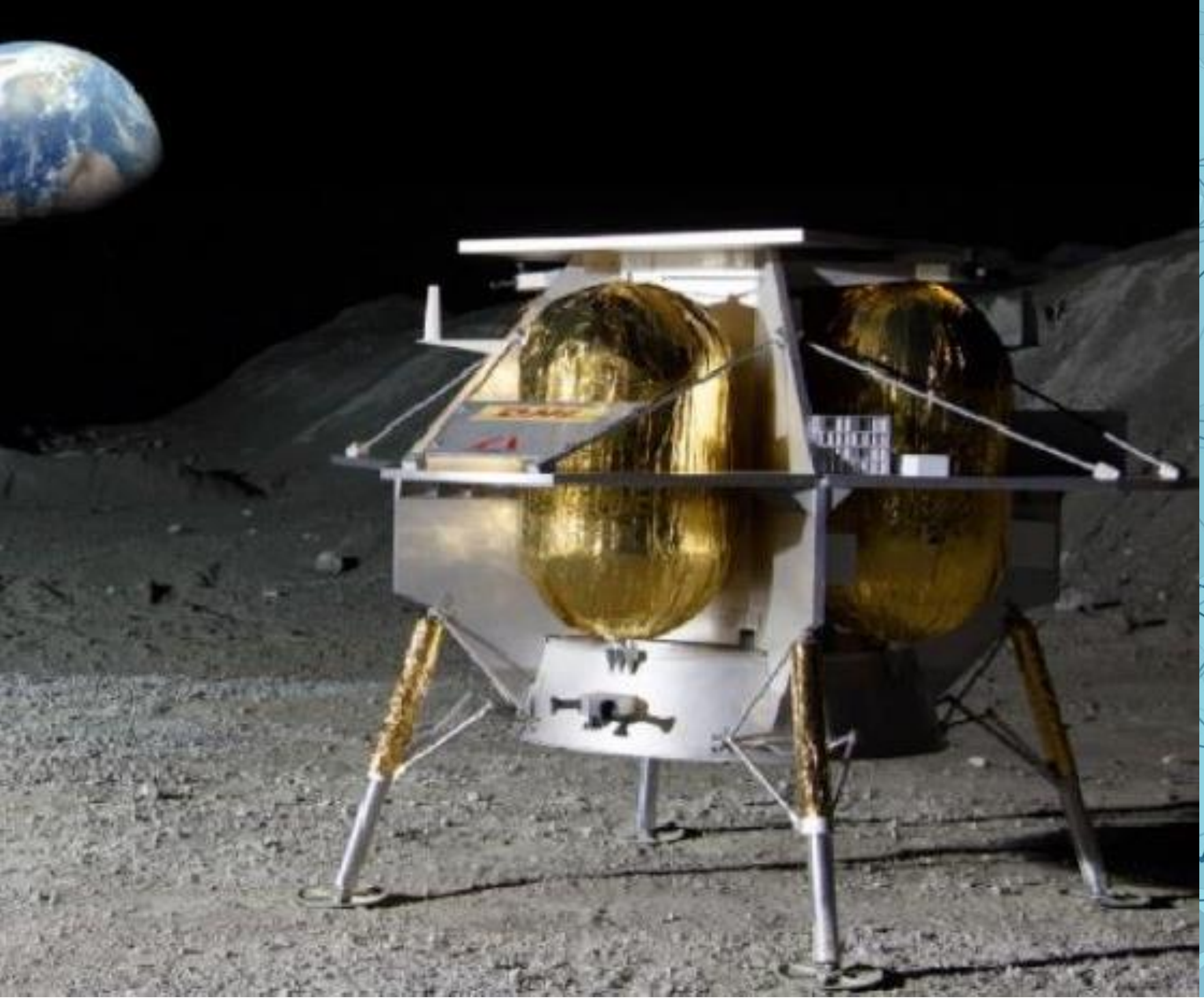
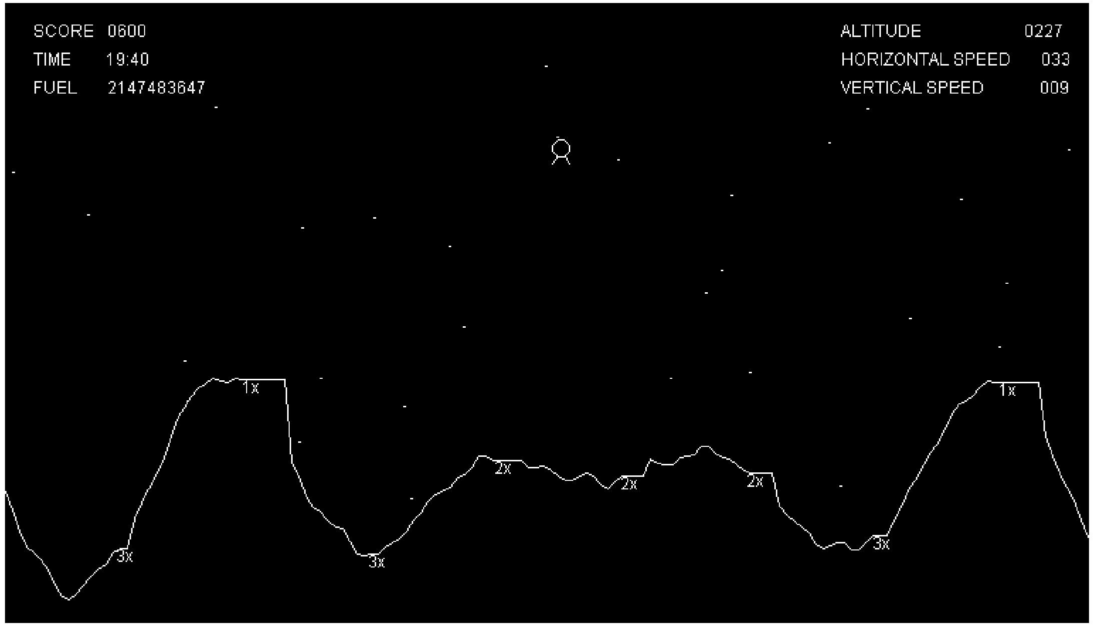


ARE NEURAL NETWORKS NECESSARY TO PLAY LUNAR LANDER: LIMITATIONS OF FEATURE BASED Q-LEARNING

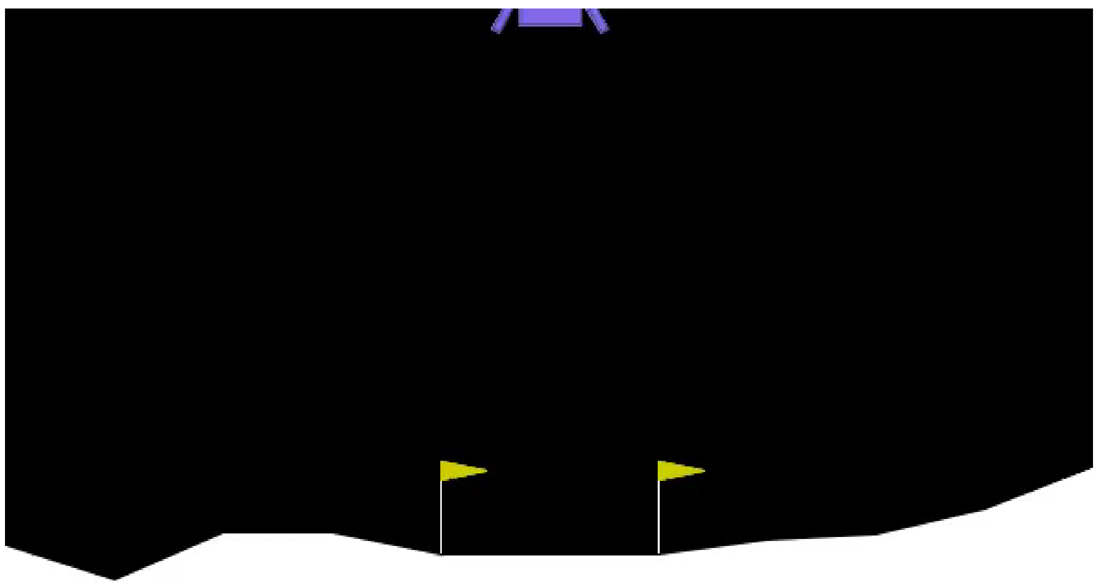
By Ian Nacke

LUNAR LANDER

RECREATION OF ORIGINAL ARCADE GAME



ENVIRONMENT PROVIDED BY OPENAI’S GYM LIBRARY



remove fixed framerate (faster learning), wait for AI move between each frame

MOTIVATION

- Reinforcement learning can model robots which perform specific tasks
    - High accuracy
    - Low time
- Digital clones useful for modelling real-world situations
- Simple is better than complex

IMPLEMENTATION WITH TABULAR Q-LEARNING

State Space

- (X, Y) coordinate
- (dx, dy) velocity vector
- Facing direction (0-360)
- ~~Fuel~~

State Size: WH*2*V*360=36000000000

*Fuel excluded to reduce state space size

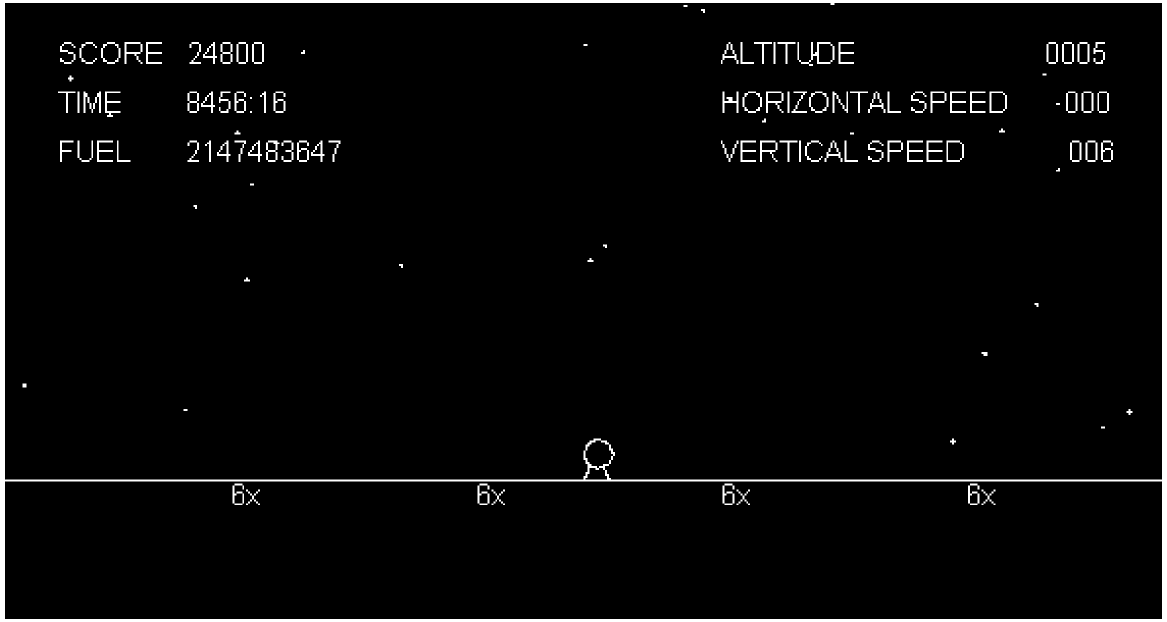
Actions

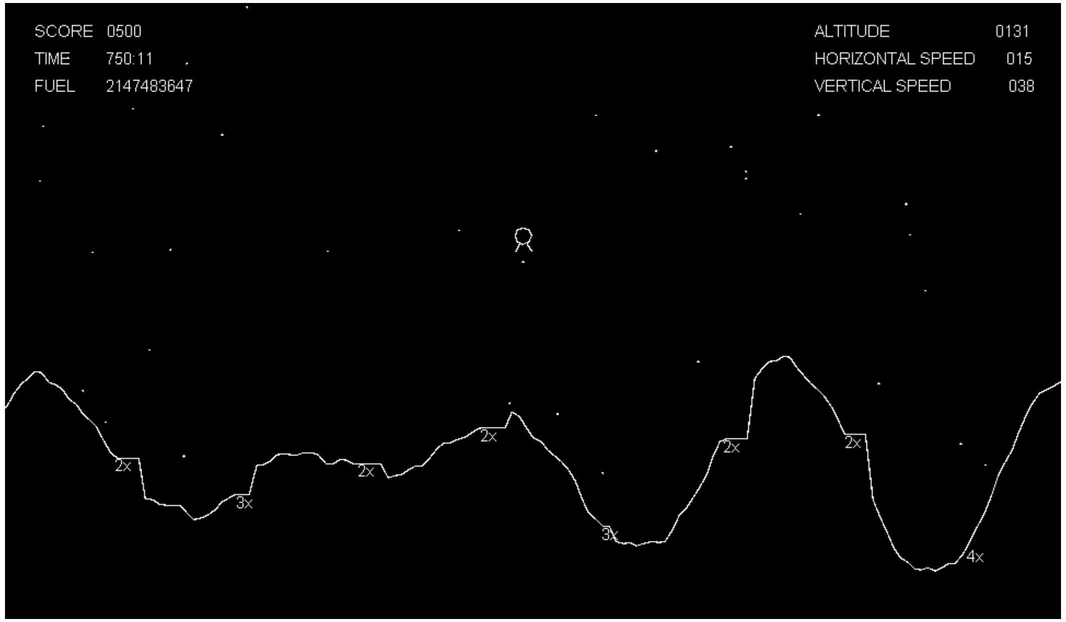
- Turn left
- Turn right
- Accelerate
- Turn left and accelerate
- Turn right and accelerate

Reward Function

- +50 for regular landing
- +100 for landing on score multiplier
- -100 for crashing

PROBLEM: CANNOT GENERALIZE





SOLUTION: FEATURE BASED

Bonus: needs less training time

Questions:

- Translate state space into qualitative values or use raw data?
- Binary or continuous?
- How to normalize data?

Altitude

The lander's distance from the ground directly below.

$$altitude = \frac{Y(x) - y}{W}$$

Nearest_barrier

Min of distance from the lander to the nearest barrier (left, up, right).

Above_landing_zone

Binary variable representing whether lander is directly above a landing zone.

Delta Y, Delta X

Normalized with $dy = \frac{dy}{Y}, dx = \frac{|dx|}{X}$.

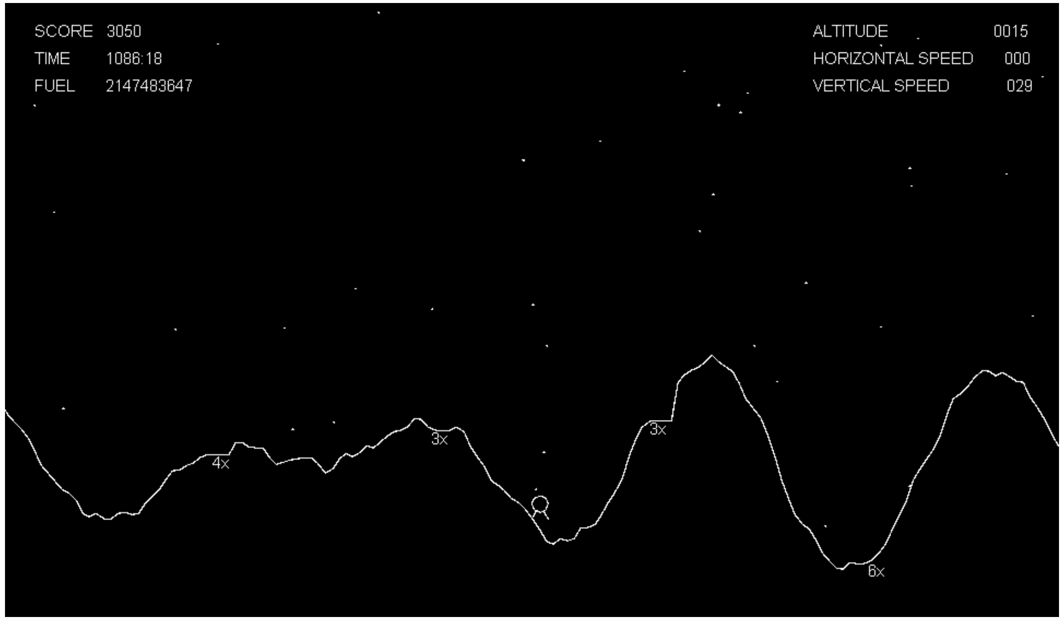
Distance_to_landing_rotation?

PROBLEM: LINEAR EQUATIONS ARE LIMITED

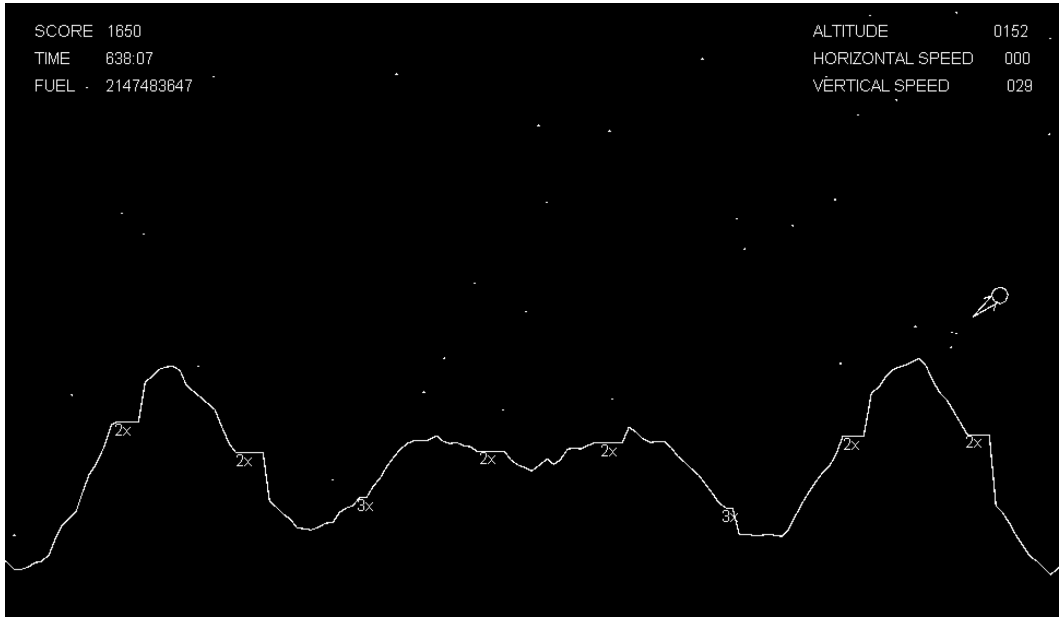
$$Q(s,a) = \sum_i w_i f_i(s')$$

Linear approach defines features as always good/always bad.

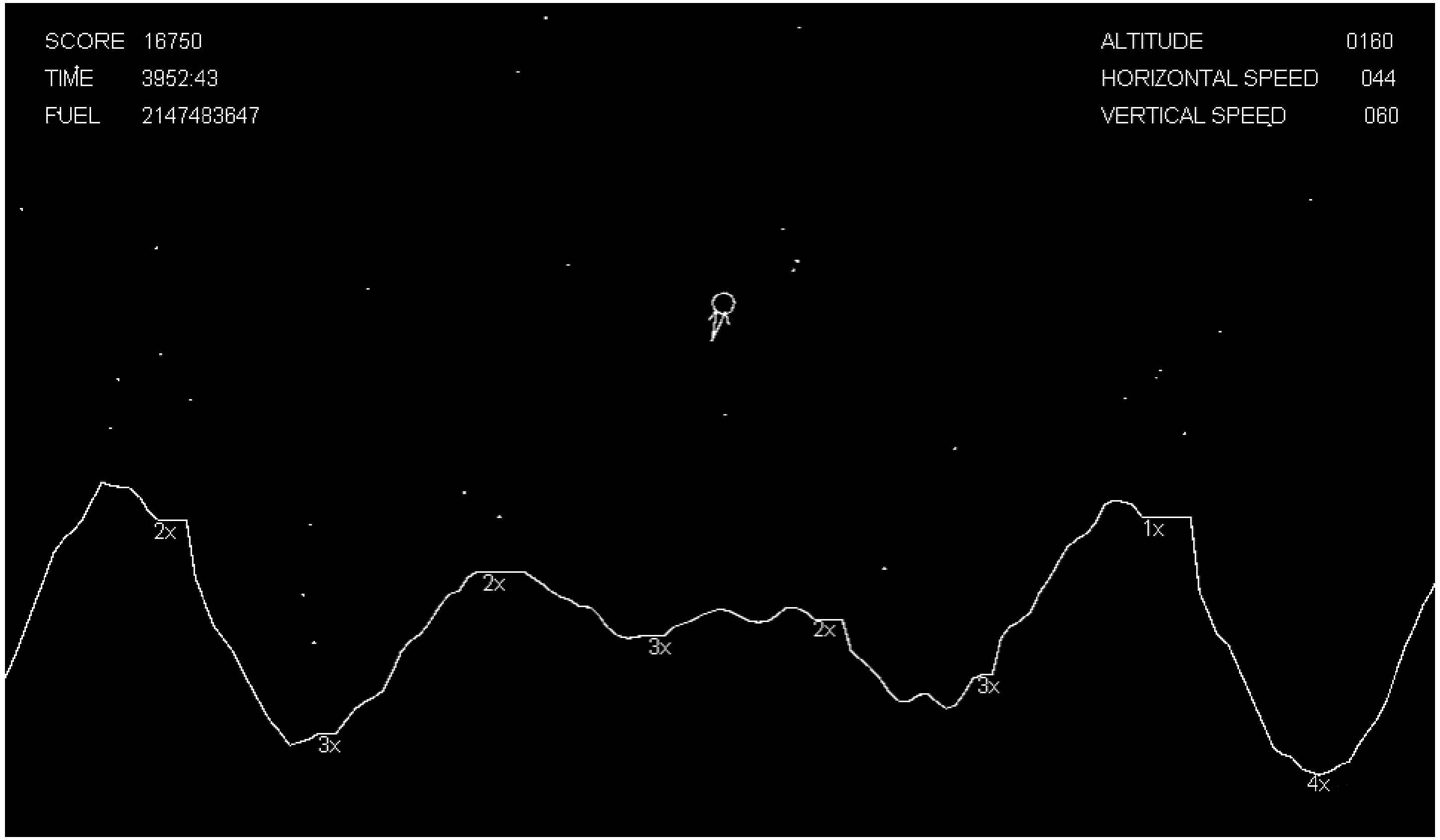
Can we use backpropagation?



With distance_to_landing_rotation



Without distance_to_landing_rotation



RESULTS

Weights after 700 iterations:

{altitude: -12.559258,
landing_speed: -3.998225,
above_zone: 1.5887228,
nearest_zone: -4.376875,
nearest_barrier: 2.7242012,
delta_y: 1.9004742,
delta_x: -3.6711361, }

BENEFITS OF DEEP Q-LEARNING

- Non-Linear
  - Can adjust importance of certain features based on situation
- Model free
  - Does not require information about system i.e. physics
- No human-defined features
- Does not suffer from curse of dimensionality

CONCLUSION

- Neural networks are partially necessary for Lunar Lander
  - Can model complex contextual situations

- Features of neural networks can be applied to feature based Q-learning
  - Backpropagation

Github: https://github.com/ian41284128/RLLunarLander