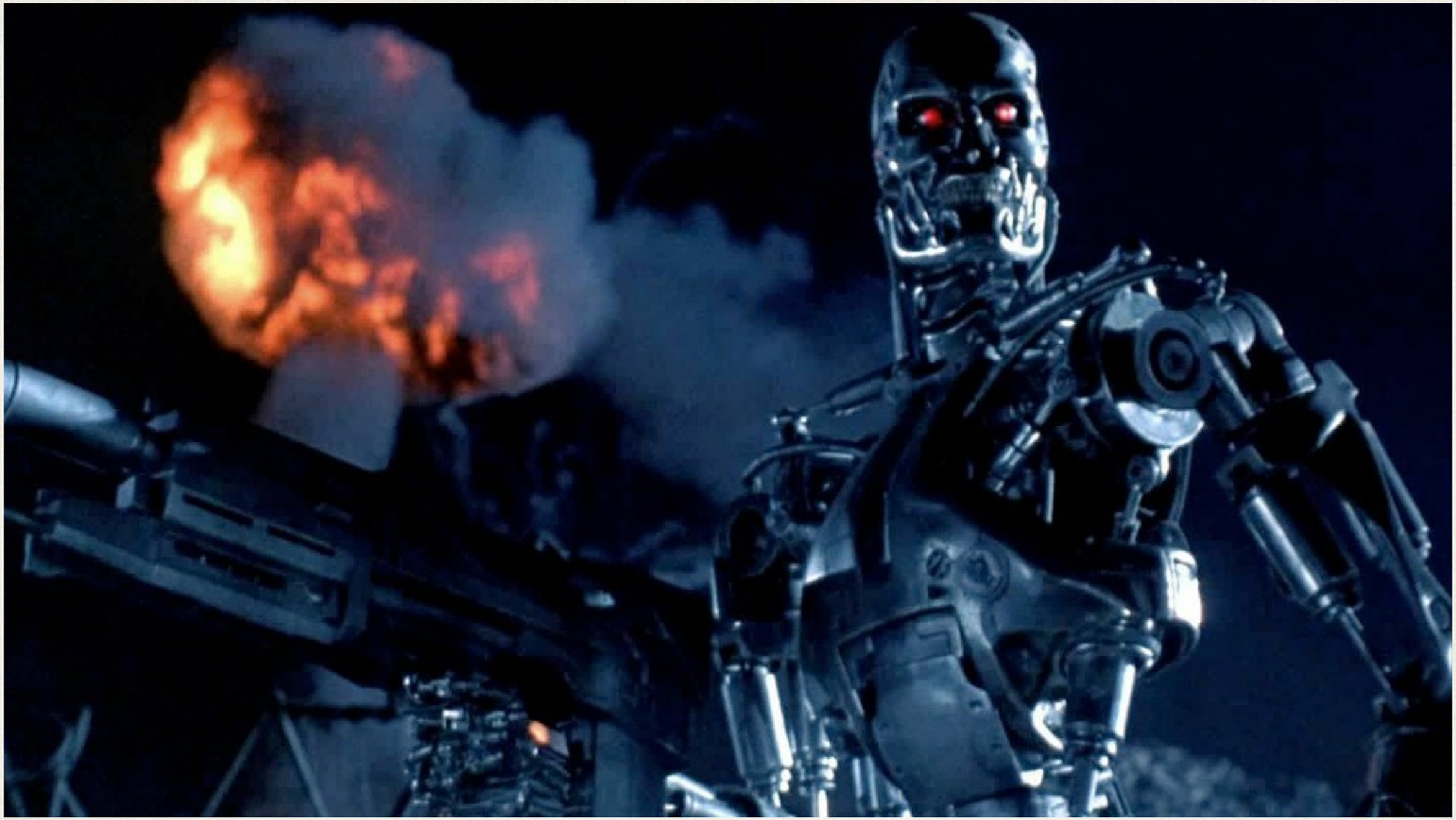


Reinforcement Learning ~ Space invaders

Daniel Parada

08/28/2018

Research Question

- What is reinforcement learning ?
  - Training an agent using rewards : positive and negative
- What are convolutional neural networks ?
  - Applying successive filters to an image or video
- Combining these two using OpenAI Gym
  - Platform for testing RL algorithms

Capstone project

- Can we train a neural network to “play” Space Invaders by maximising its reward and reducing its cost function ?

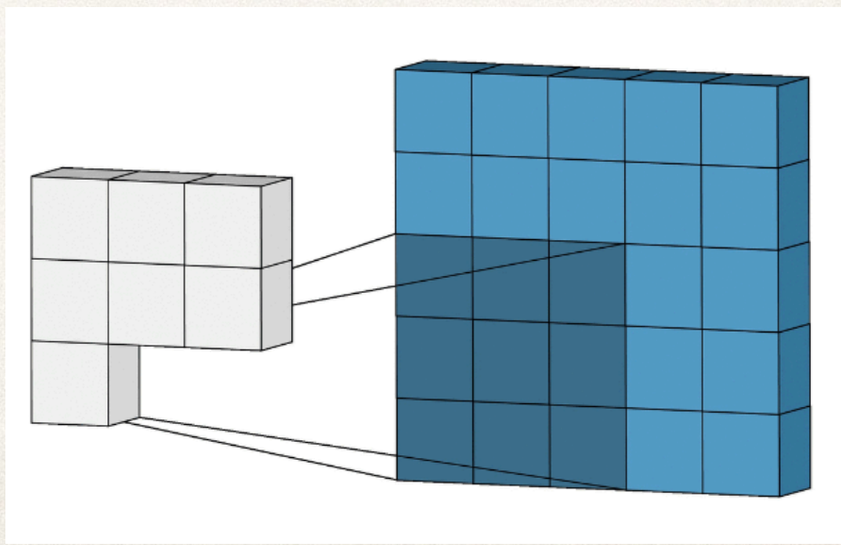
1. What type of data do we need ?

2. How do we collect it ?

Tools

- Docker

- OpenAI Gym
- Convolutional Neural Networks

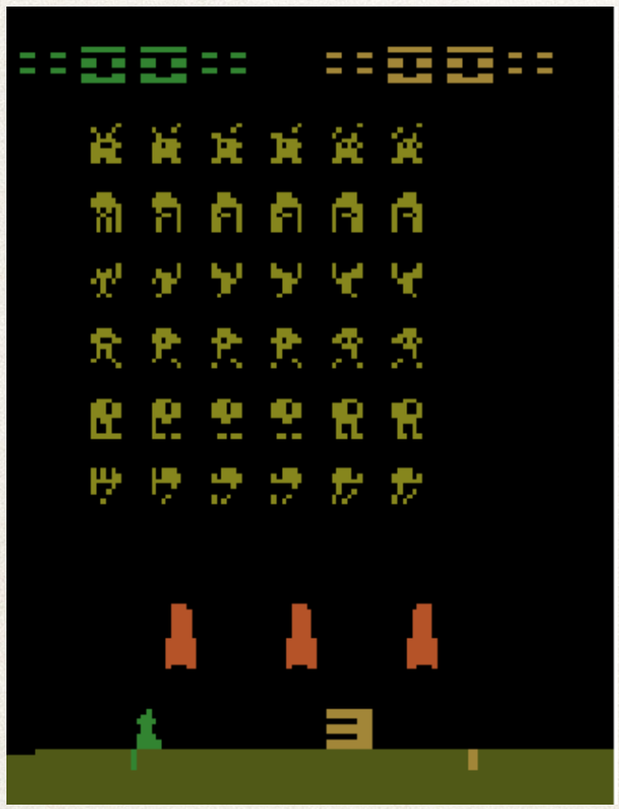


- Reinforcement Learning

Data wrangling

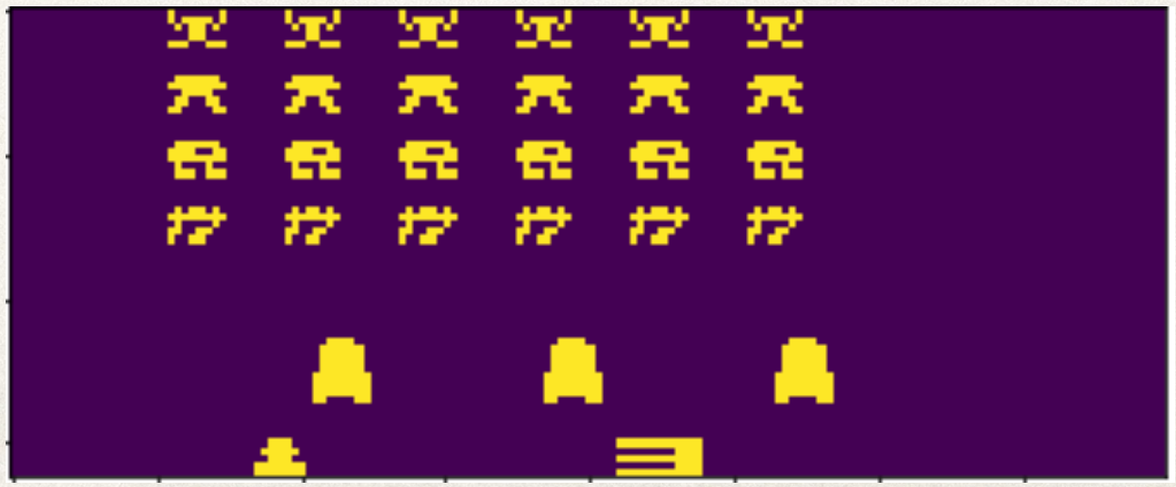
- Raw input images

(210 x 160 x 3)



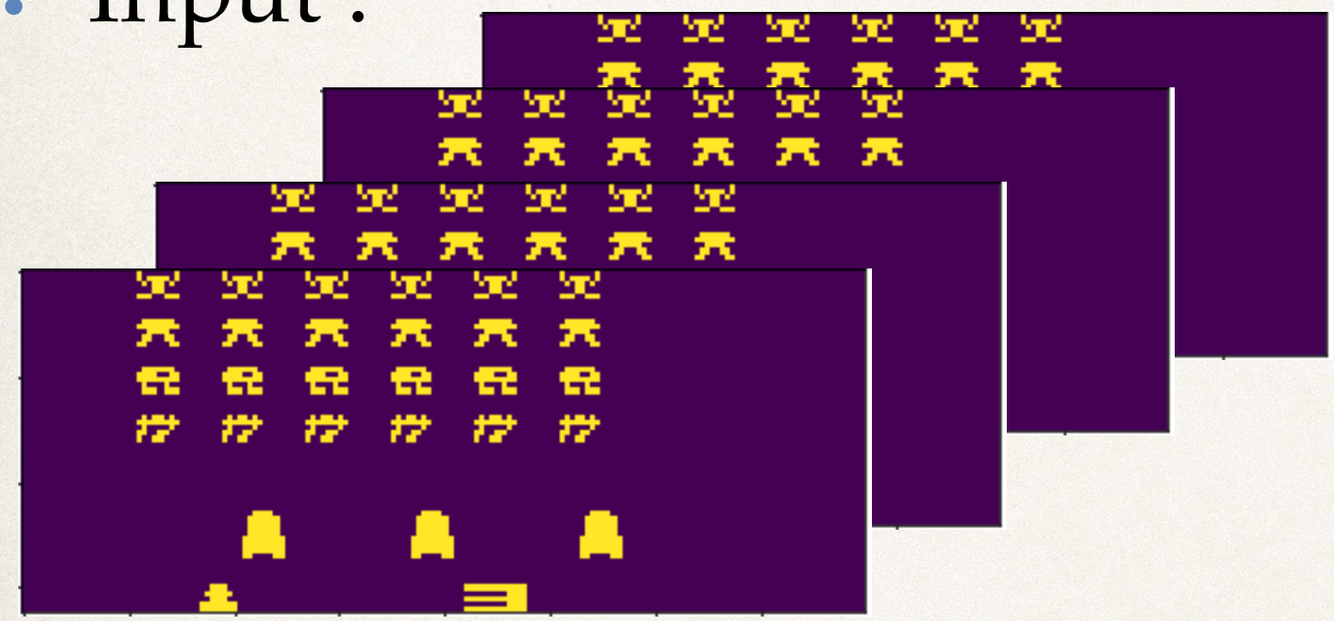
- Reformatting images to single channel and resizing

(65 x 160 x 1)

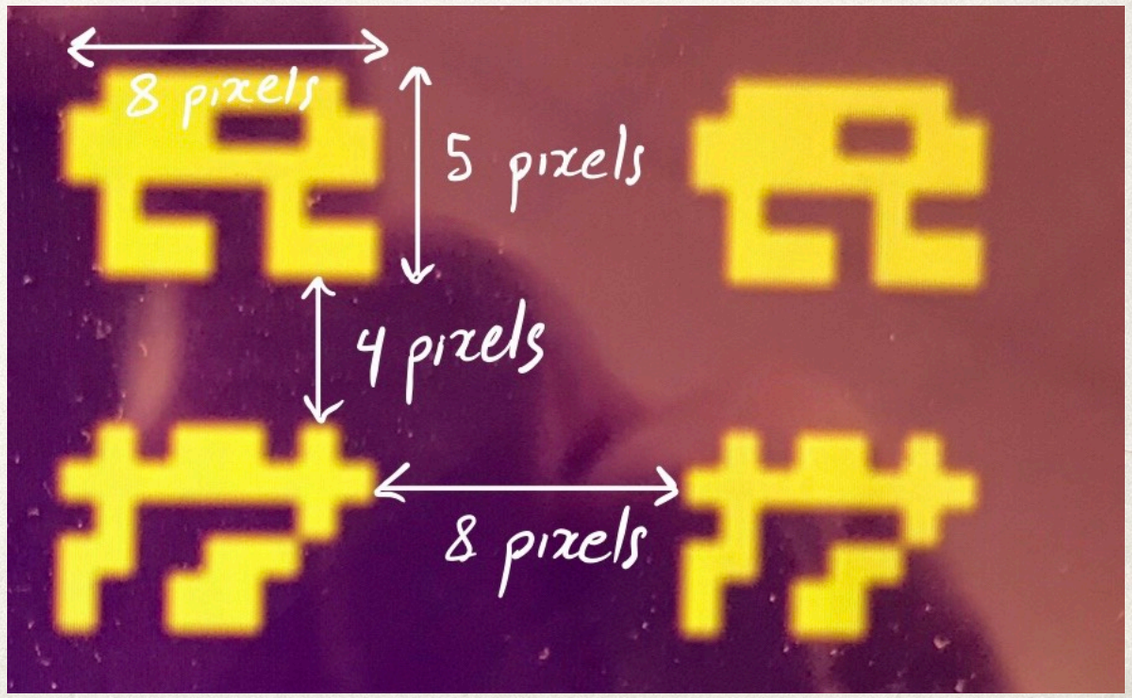


Visualising our data

- Input :



- Filter size and stride :



Visualising our Convolutional Neural Network

- CNN :

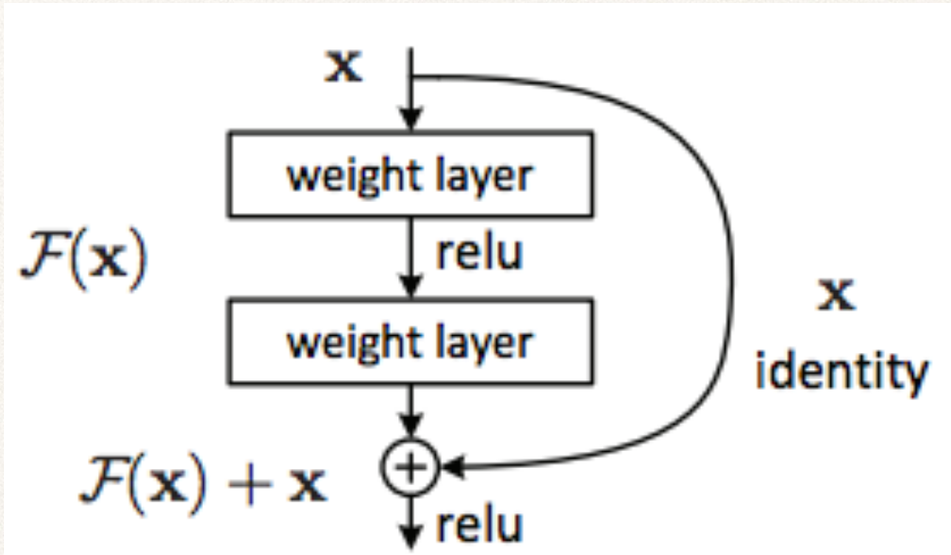
1. “Hidden” layer 1
2. Max pooling
3. “Hidden” layer 2
4. “Hidden” layer 3
5. Vectorisation (flatten)
6. Fully connected layer 1
7. Fully connected layer 2
8. Fully connected layer 3

Layer (type)	Output Shape	Param #
h_conv1 (Conv2D)	(None, 65, 160, 16)	2320
max_pooling2d_1 (MaxPooling2	(None, 32, 80, 16)	0
h_conv2 (Conv2D)	(None, 29, 77, 32)	8224
h_conv3 (Conv2D)	(None, 28, 76, 32)	4128
flatten_1 (Flatten)	(None, 68096)	0
h_fc1 (Dense)	(None, 176)	11985072
h_fc2 (Dense)	(None, 44)	7788
h_fc3 (Dense)	(None, 6)	270

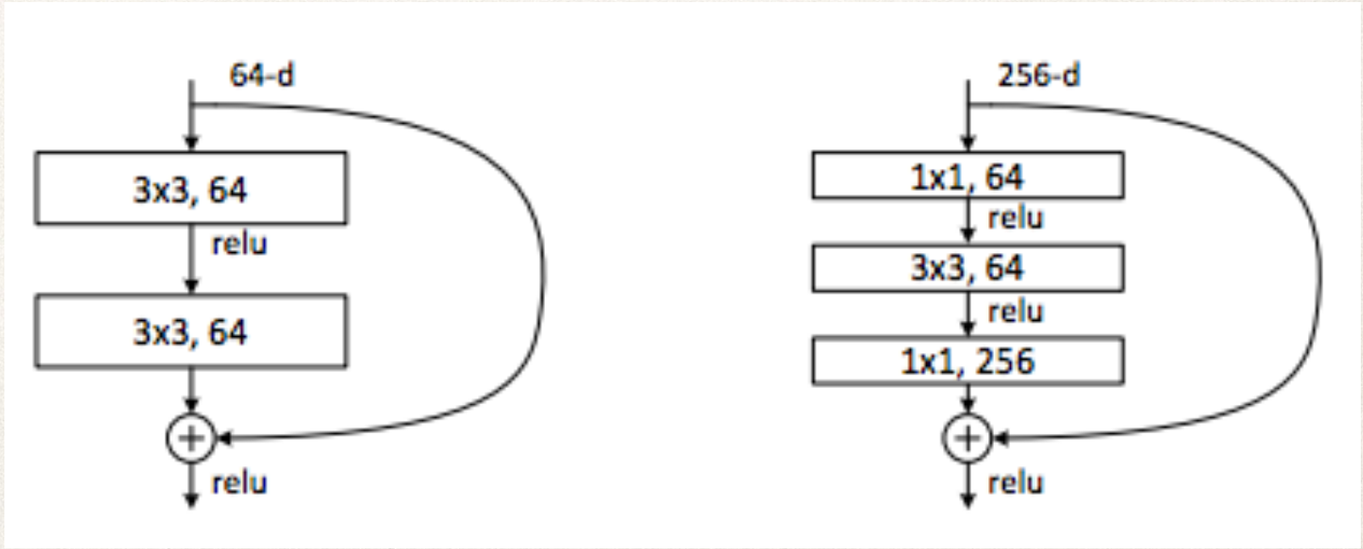
Total params: 12,007,802
Trainable params: 12,007,802
Non-trainable params: 0

Visualising the Resnet Neural Network

- Principle



- Main motivation



Output while training

- TimeStep : The training step of the simulation

```
('TIMESTEP', 60000, '/ STATE', 'explore', '/ EPSILON', 0.9910000000002963)
('step:', 62000, ', reward:', 1.0, ', terminal:', False)
('step:', 64000, ', reward:', 1.0, ', terminal:', False)
('step:', 66000, ', reward:', 1.0, ', terminal:', False)
('step:', 68000, ', reward:', 1.0, ', terminal:', False)
('step:', 70000, ', reward:', 1.0, ', terminal:', False)
('TIMESTEP', 70000, '/ STATE', 'explore', '/ EPSILON', 0.9820000000005926)
('step:', 72000, ', reward:', 1.0, ', terminal:', False)
('step:', 74000, ', reward:', 1.0, ', terminal:', False)
('step:', 76000, ', reward:', 1.0, ', terminal:', False)
('step:', 78000, ', reward:', 1.0, ', terminal:', False)
('step:', 80000, ', reward:', 1.0, ', terminal:', False)
('TIMESTEP', 80000, '/ STATE', 'explore', '/ EPSILON', 0.9730000000008889)
```

- State :

1. Observe (TimeStep < 50,000)
2. Explore (TimeStep >= 50,000)

- Epsilon : $\epsilon[0,1]$

1. Probability of random action performed by agent

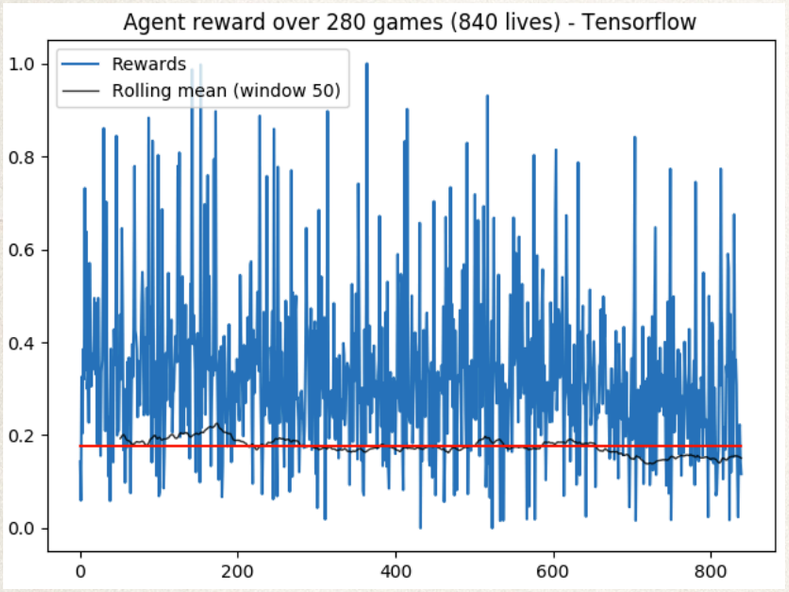
Training the Agent

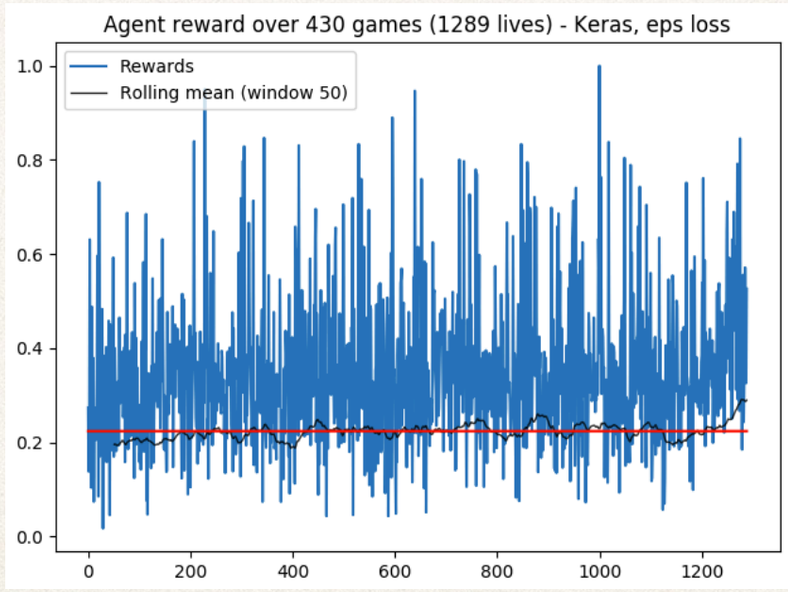
- Strategy :
  - resize output to capture lasers from space invaders
  - increase the number of fully connected layers, i.e. deeper networks outperform shallow ones
  - modify the loss function, use an epsilon “tube” approach

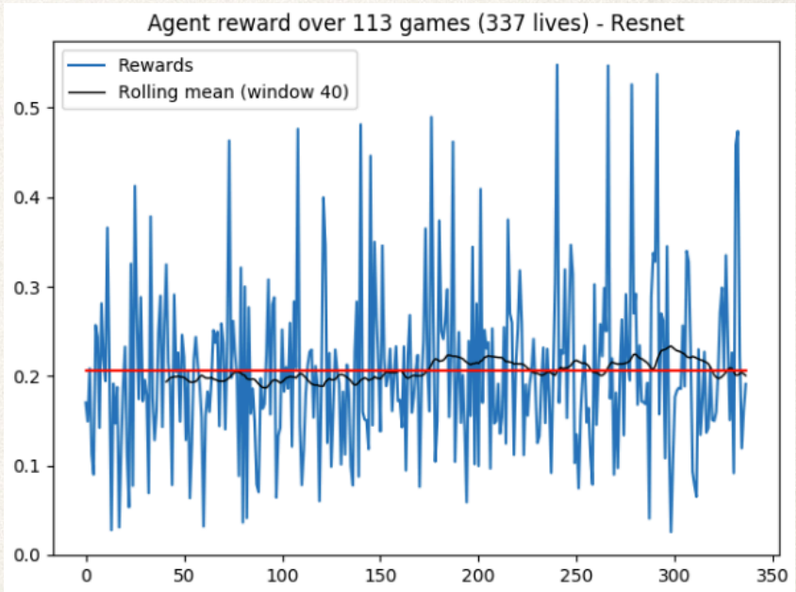
Model evaluation

- Multiple ways to analyse the performance of the algorithm

Cost/Loss Reward Length of game played







- Tensorflow/Keras same architecture but different loss function
- Resnet architecture : same loss function as Tensorflow

Model selection and parameters

- Previous results do not make it clear which approach is best, since all models could not be trained the same amount due to lack of computing resources.

- As seen earlier, there are over 12 million parameters to tune, which correspond to the various weights and biases of our network. There are hyper parameters such as the value of epsilon from the loss function that can be modified, the size of the filters of our CNN along with the stride and the number of fully connected layers

Conclusion

- Whilst there is no clear winner among the models used, the epsilon loss function seems to have the highest rolling average. However, the resnet architecture seems to have the most steady climb in reward despite the relatively short training it got.

Resources :

- https://nerdist.com/james-cameron-and-deadpools-tim-miller-look-to-make-a-new-terminator-movie/
- https://towardsdatascience.com/intuitively-understanding-convolutions-for-deep-learning-1f6f42faee1
- https://positively.com/dog-training/
- https://arxiv.org/pdf/1512.03385.pdf
- https://www.theverge.com/circuitbreaker/2018/5/25/17386716/docker-kubernetes-containers-explained
-