HackConf 2019

Playing ATARI Games with Deep Reinforcement Learning

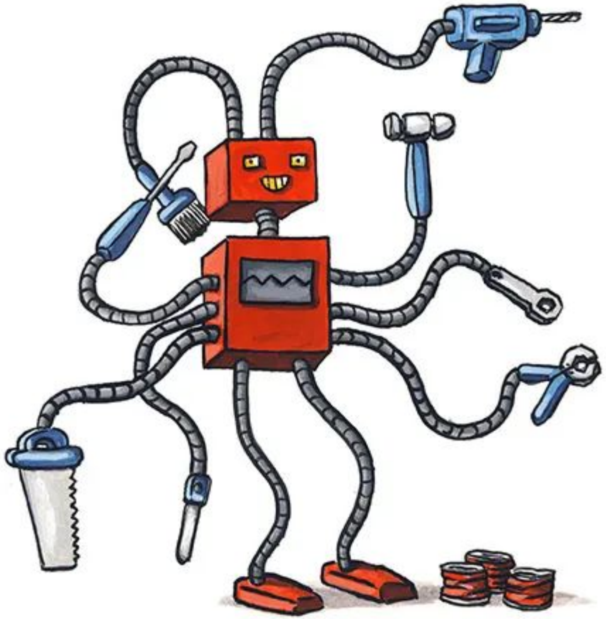
Dr. Svetlin Penkov

SCIRO RESEARCH

SCIRO RESEARCH

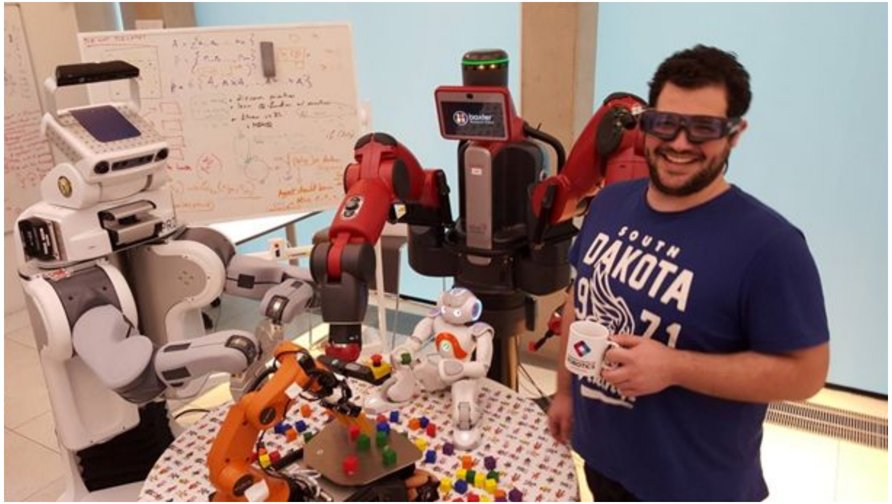
About me

Dreaming about building intelligent robots since the age of 6...



by [Becky Barnicoat]

About me



PhD in Robotics & AI

THE UNIVERSITY of EDINBURGH

FIVE AI

Research Scientist & Team Lead

About me

Robots should learn to program themselves…

- Team of world experts in AI and robotics
- Design, develop and deploy AI based solutions in **challenging domains**
- Research new **state-of-the-art AI methods**

Agenda

www.github.com/svepe/atari-dqn-workshop

10:30 - 11:20	Introduction to RL and OpenAI gym
Break	
11:30 - 12:20	Introduction to deep neural networks and Chainer
Break	
12:30 - 13:20	Implementing a Deep Q-learning Network (DQN) agent to play ATARI games

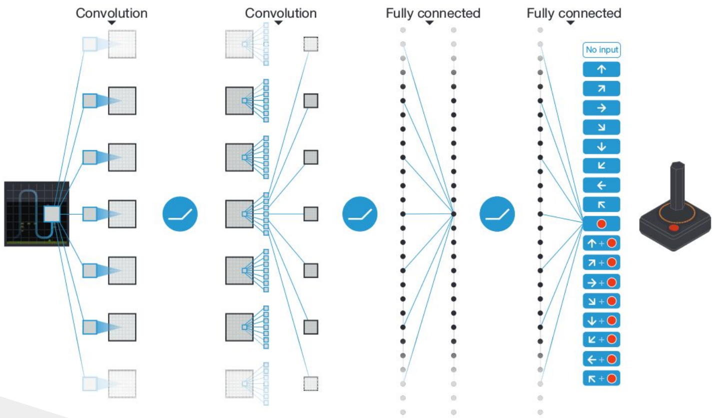
Goal

LETTER

doi:10.1038/nature14236

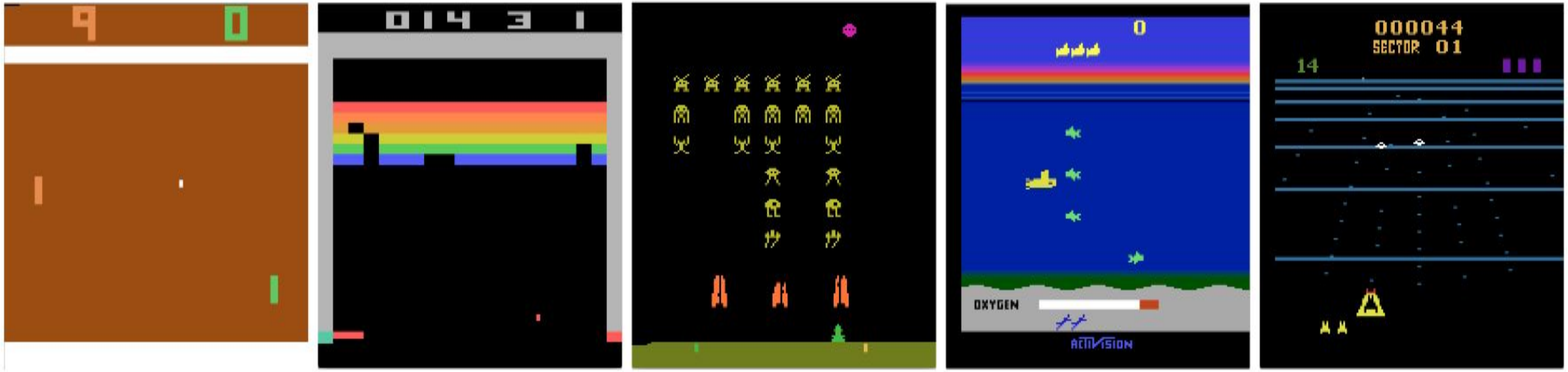
Human-level control through deep reinforcement learning

Volodymyr Mnih[1]*, Koray Kavukcuoglu[1]*, David Silver[1]*, Andrei A. Rusu[1], Joel Veness[1], Marc G. Bellemare[1], Alex Graves[1], Martin Riedmiller[1], Andreas K. Fidjeland[1], Georg Ostrovski[1], Stig Petersen[1], Charles Beattie[1], Amir Sadik[1], Ioannis Antonoglou[1], Helen King[1], Dharshan Kumaran[1], Daan Wierstra[1], Shane Legg[1] & Demis Hassabis[1]

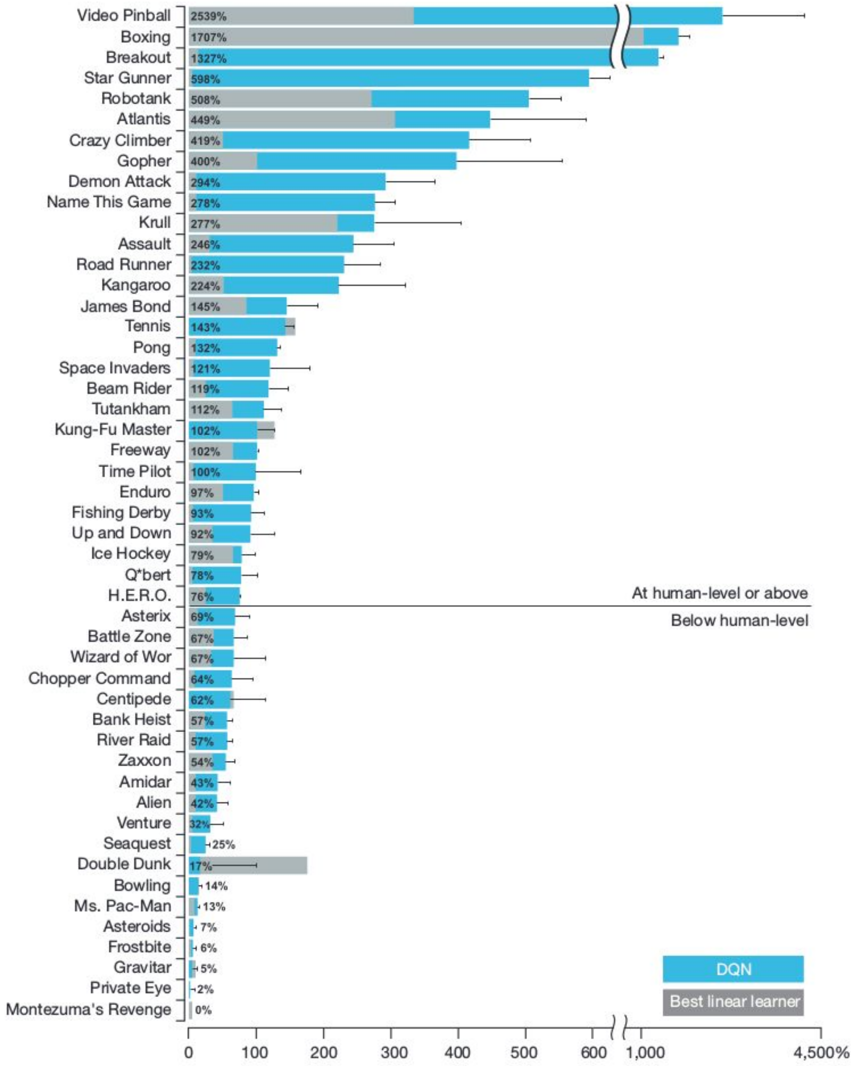


Mnih et al., *Nature*, 2015

Playing ATARI Games

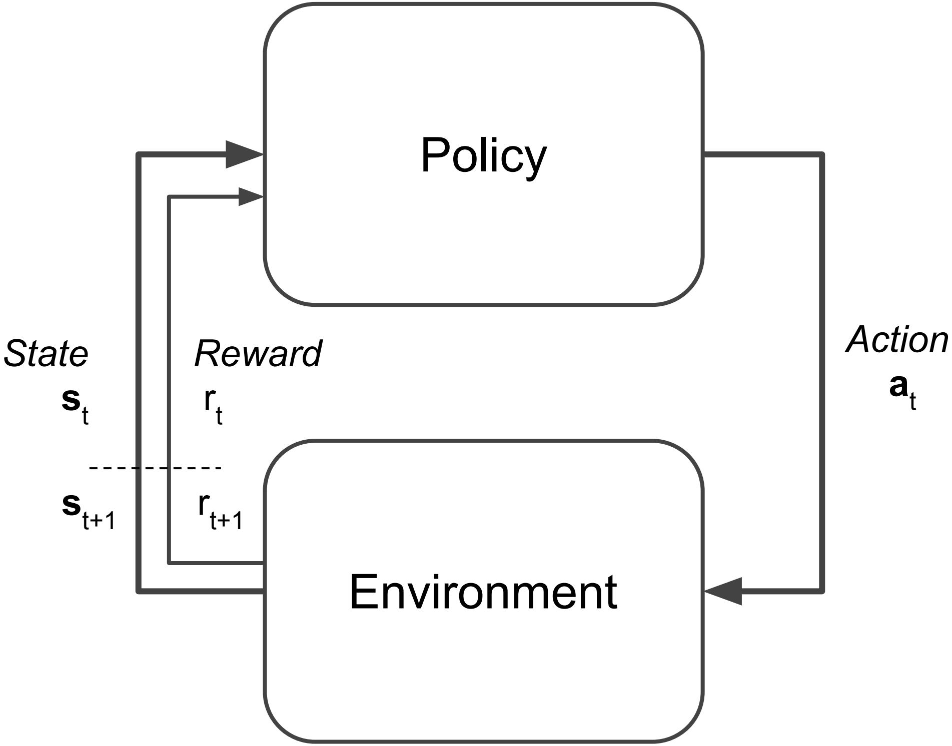


Mnih et al., *Nature*, 2015

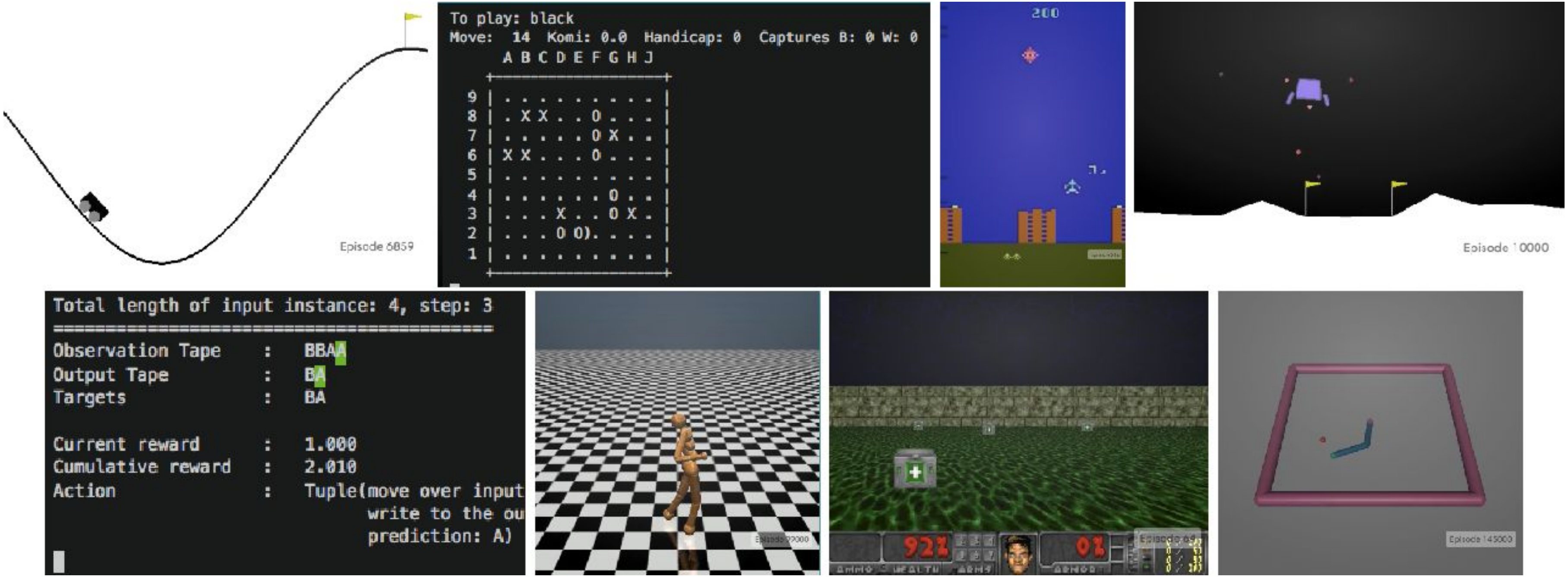


Mnih et al., *Nature*, 2015

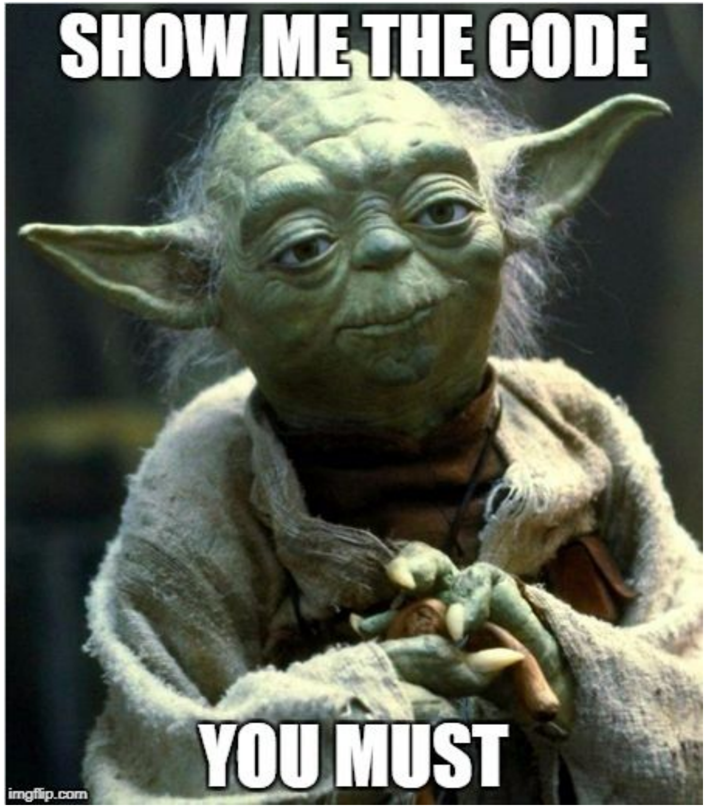
Reinforcement Learning 101



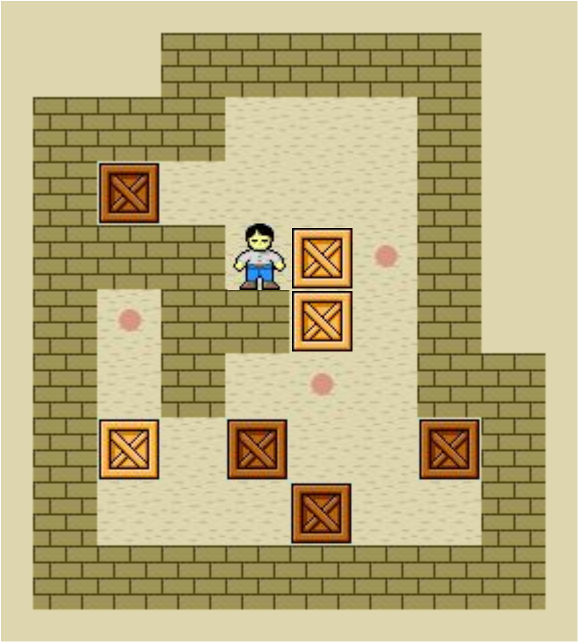
OpenAI Gym


Episode 6859
To play: black
Move: 14 Komi: 0.0 Handicap: 0 Captures B: 0 W: 0
A B C D E F G H J
200
Episode 10000
Total length of input instance: 4, step: 3
Observation Tape : BBAA
Output Tape : BA
Targets : BA
Current reward : 1.000
Cumulative reward : 2.010
Action : Tuple(move over input
write to the ou
prediction: A)

OpenAI Gym

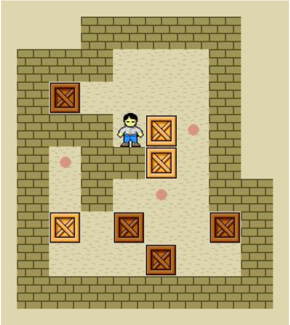


Sokoban Example



Episode 1

$\mathbf{s}_0 \in \mathbf{S}$

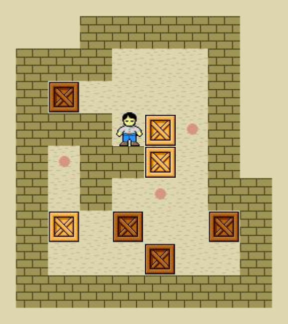


$\mathbf{a}_1 \in \{\mathrm{move}_1, \mathrm{move}_2, \mathrm{move}_3, \mathrm{move}_4\} = \mathbf{A}$

$\mathbf{a}_1 = ? \mid \mathbf{s}_0$

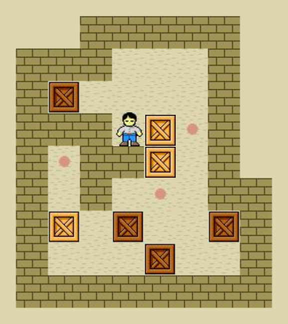
Episode 1

$s_0 \in S$

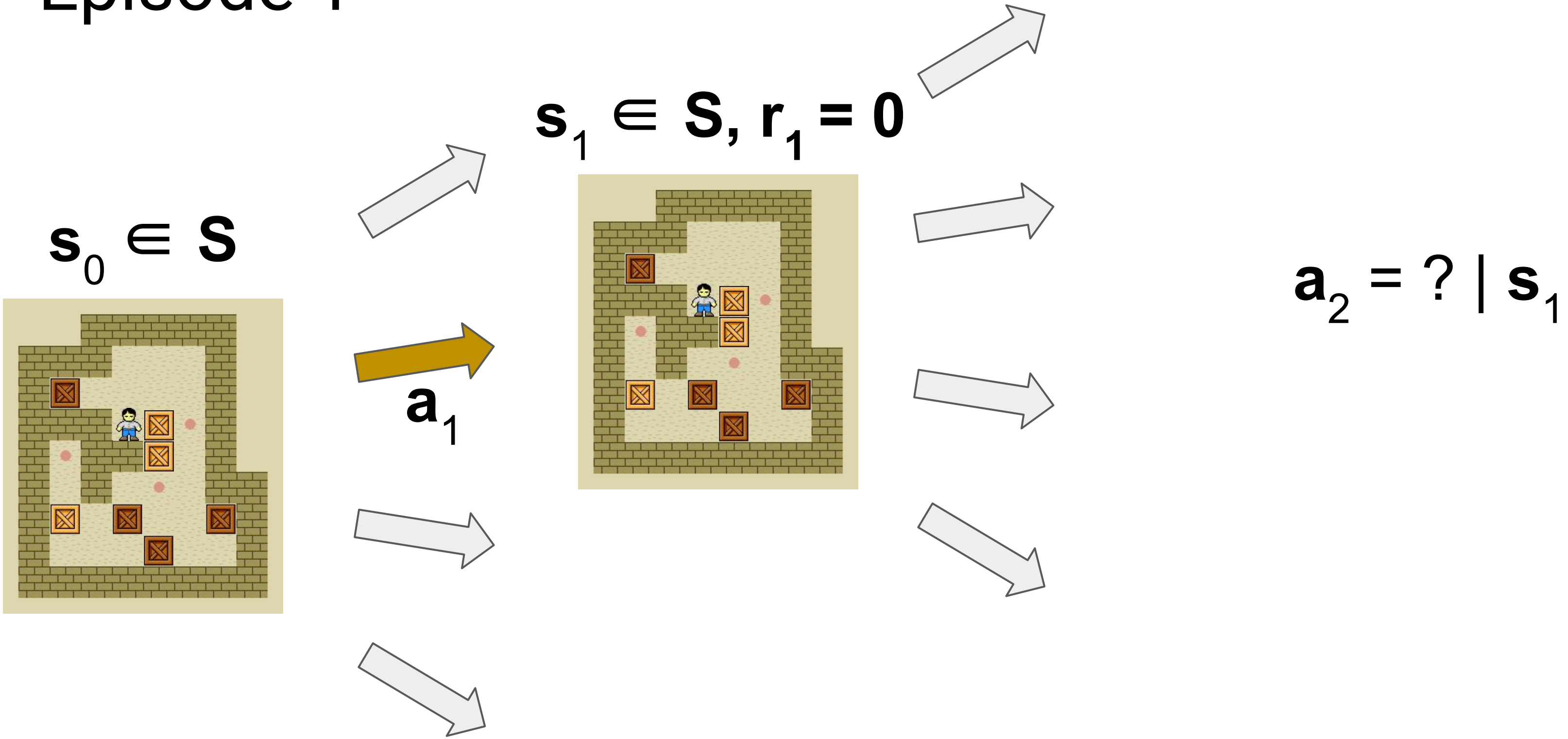


a_1

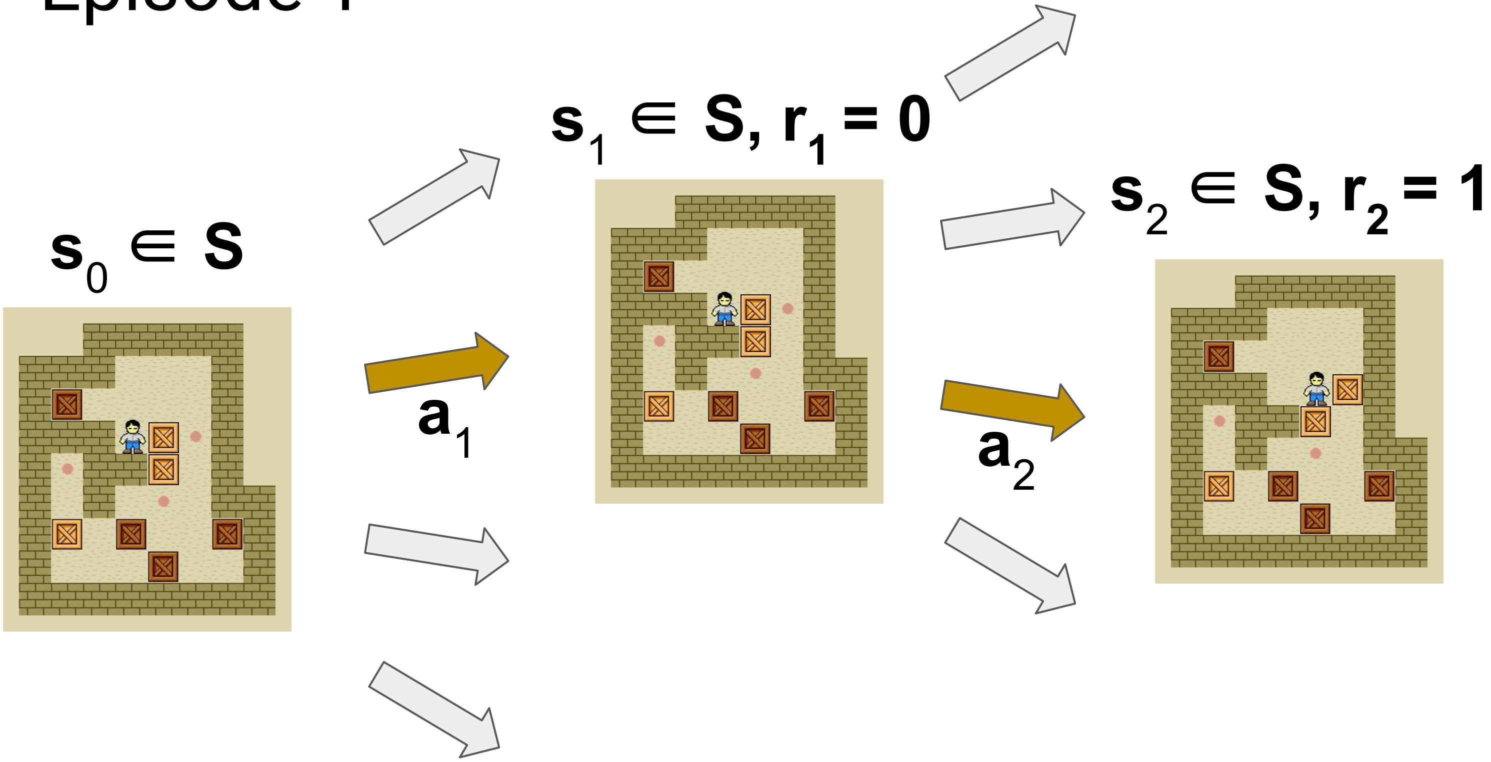
$s_1 \in S, r_1 = 0$



Episode 1



Episode 1



Episode Trace

$$
\begin{array}{ccccccccccccccccccccccc}
\mathbf{s}_0 & \rightarrow & \mathbf{a}_1 & \rightarrow & \mathbf{s}_1 & \rightarrow & \mathbf{a}_2 & \rightarrow & \mathbf{s}_2 & \rightarrow \cdots \rightarrow & \mathbf{a}_t & \rightarrow & \mathbf{s}_t & \rightarrow & \mathbf{a}_{t+1} & \rightarrow & \mathbf{s}_{t+1} & \rightarrow \cdots \rightarrow & \mathbf{a}_T & \rightarrow & \mathbf{s}_T \\
 & & & & r_1 & & & & r_2 & & & & r_t & & & & r_{t+1} & & & & r_T
\end{array}
$$

State Value

$$\mathbf{s}_0 \to \mathbf{a}_1 \to \mathbf{s}_1 \to \mathbf{a}_2 \to \mathbf{s}_2 \to \cdots \to \mathbf{a}_t \to \mathbf{s}_t \to \mathbf{a}_{t+1} \to \mathbf{s}_{t+1} \to \cdots \to \mathbf{a}_T \to \mathbf{s}_T$$

$$r_1 \qquad r_2 \qquad r_t \qquad r_{t+1} \qquad r_T$$

$$\mathbf{V}(\mathbf{s}_t) = r_t + r_{t+1} + r_{t+2} + \cdots + r_T$$

Discounted State Value

$$\mathbf{s}_0 \rightarrow \mathbf{a}_1 \rightarrow \mathbf{s}_1 \rightarrow \mathbf{a}_2 \rightarrow \mathbf{s}_2 \rightarrow \cdots \rightarrow \mathbf{a}_t \rightarrow \mathbf{s}_t \rightarrow \mathbf{a}_{t+1} \rightarrow \mathbf{s}_{t+1} \rightarrow \cdots \rightarrow \mathbf{a}_T \rightarrow \mathbf{s}_T$$

$$r_1 \qquad r_2 \qquad r_t \qquad r_{t+1} \qquad r_T$$

$$\mathbf{V}(\mathbf{s}_t) = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} \cdots + \gamma^{T-t} r_T$$

$\gamma \in [0, 1)$ - discount factor

RL Objective

$$\mathbf{s}_0 \rightarrow \mathbf{a}_1 \rightarrow \mathbf{s}_1 \rightarrow \mathbf{a}_2 \rightarrow \mathbf{s}_2 \rightarrow \cdots \rightarrow \mathbf{a}_t \rightarrow \mathbf{s}_t \rightarrow \mathbf{a}_{t+1} \rightarrow \mathbf{s}_{t+1} \rightarrow \cdots \rightarrow \mathbf{a}_T \rightarrow \mathbf{s}_T$$

$$r_1 \qquad r_2 \qquad r_t \qquad r_{t+1} \qquad r_T$$

Choose $\mathbf{a}_1, \mathbf{a}_2 \ldots, \mathbf{a}_T$ such that we maximise

$$\mathbf{V}(\mathbf{s}_1) = r_1 + \gamma r_2 + \gamma^2 r_3 \cdots + \gamma^T r_T$$

SCIRO RESEARCH

Temporal Difference (TD-learning)

$$\mathbf{s}_0 \rightarrow \mathbf{a}_1 \rightarrow \mathbf{s}_1 \rightarrow \mathbf{a}_2 \rightarrow \mathbf{s}_2 \rightarrow \cdots \rightarrow \mathbf{a}_t \rightarrow \mathbf{s}_t \rightarrow \mathbf{a}_{t+1} \rightarrow \mathbf{s}_{t+1} \rightarrow \cdots \rightarrow \mathbf{a}_T \rightarrow \mathbf{s}_T$$

$$r_1 \quad r_2 \quad r_t \quad r_{t+1} \quad r_T$$

$$\mathbf{V}(\mathbf{s}_1) = r_1 + \gamma r_2 + \gamma^2 r_3 \cdots + \gamma^T r_T$$

$$= r_1 + \gamma (r_2 + \gamma^1 r_3 \cdots + \gamma^{T-1} r_T)$$

$$= r_1 + \gamma \mathbf{V}(\mathbf{s}_2)$$

Temporal Difference (TD-learning)

$$\mathbf{s}_0 \rightarrow \mathbf{a}_1 \rightarrow \mathbf{s}_1 \rightarrow \mathbf{a}_2 \rightarrow \mathbf{s}_2 \rightarrow \cdots \rightarrow \mathbf{a}_t \rightarrow \mathbf{s}_t \rightarrow \mathbf{a}_{t+1} \rightarrow \mathbf{s}_{t+1} \rightarrow \cdots \rightarrow \mathbf{a}_T \rightarrow \mathbf{s}_T$$

$$r_1 \qquad r_2 \qquad r_t \qquad r_{t+1} \qquad r_T$$

$$\mathbf{V}(\mathbf{s}_1) = r_1 + \gamma \mathbf{V}(\mathbf{s}_2)$$

$$\Downarrow$$

$$r_1 + \gamma \mathbf{V}(\mathbf{s}_2) - \mathbf{V}(\mathbf{s}_1) = 0$$

Temporal Difference (TD-learning)

$$\mathbf{s}_0 \rightarrow \mathbf{a}_1 \rightarrow \mathbf{s}_1 \rightarrow \mathbf{a}_2 \rightarrow \mathbf{s}_2 \rightarrow \cdots \rightarrow \mathbf{a}_t \rightarrow \mathbf{s}_t \rightarrow \mathbf{a}_{t+1} \rightarrow \mathbf{s}_{t+1} \rightarrow \cdots \rightarrow \mathbf{a}_T \rightarrow \mathbf{s}_T$$

$$r_1 \quad r_2 \quad r_t \quad r_{t+1} \quad r_T$$

$$\mathbf{V}(\mathbf{s}_t) = r_t + \gamma \mathbf{V}(\mathbf{s}_{t+1})$$

$$\Downarrow$$

$$r_t + \gamma \mathbf{V}(\mathbf{s}_{t+1}) - \mathbf{V}(\mathbf{s}_t) = 0$$

Temporal Difference (TD-learning)

$$\mathbf{s}_0 \rightarrow \mathbf{a}_1 \rightarrow \mathbf{s}_1 \rightarrow \mathbf{a}_2 \rightarrow \mathbf{s}_2 \rightarrow \cdots \rightarrow \mathbf{a}_t \rightarrow \mathbf{s}_t \rightarrow \mathbf{a}_{t+1} \rightarrow \mathbf{s}_{t+1} \rightarrow \cdots \rightarrow \mathbf{a}_T \rightarrow \mathbf{s}_T$$

$$r_1 \quad r_2 \quad r_t \quad r_{t+1} \quad r_T$$

$$r_t + \gamma \mathbf{V}(\mathbf{s}_{t+1}) - \mathbf{V}(\mathbf{s}_t) = 0 \quad \text{for any} \quad (\mathbf{s}_t, r_t, \mathbf{s}_{t+1})$$

Choose $\mathbf{a}_1$, $\mathbf{a}_2$..., $\mathbf{a}_T$ such that… but how?

Action Value Function (Q-value)

$$\mathbf{s}_0 \rightarrow \mathbf{a}_1 \rightarrow \mathbf{s}_1 \rightarrow \mathbf{a}_2 \rightarrow \mathbf{s}_2 \rightarrow \cdots \rightarrow \mathbf{a}_t \rightarrow \mathbf{s}_t \rightarrow \mathbf{a}_{t+1} \rightarrow \mathbf{s}_{t+1} \rightarrow \cdots \rightarrow \mathbf{a}_T \rightarrow \mathbf{s}_T$$

$$r_1 \quad r_2 \quad r_t \quad r_{t+1} \quad r_T$$

Q(**s**, **a**) - the value of choosing **a** in state **s**

a* = argmax$_{\mathbf{a}}$ Q(**s**, **a**) - the best action in state **s**

Action Value Function (Q-value)

$$\mathbf{s}_0 \rightarrow \mathbf{a}_1 \rightarrow \mathbf{s}_1 \rightarrow \mathbf{a}_2 \rightarrow \mathbf{s}_2 \rightarrow \cdots \rightarrow \mathbf{a}_t \rightarrow \mathbf{s}_t \rightarrow \mathbf{a}_{t+1} \rightarrow \mathbf{s}_{t+1} \rightarrow \cdots \rightarrow \mathbf{a}_T \rightarrow \mathbf{s}_T$$

$$r_1 \quad r_2 \quad r_t \quad r_{t+1} \quad r_T$$

$$\mathbf{V}(\mathbf{s}) = \max_{\mathbf{a}} \mathbf{Q}(\mathbf{s}, \mathbf{a})$$

$$r_t + \gamma \mathbf{V}(\mathbf{s}_{t+1}) - \mathbf{V}(\mathbf{s}_t) = 0 \quad \text{for all} \quad (\mathbf{s}_t, r_t, \mathbf{s}_{t+1})$$

Action Value Function (Q-value)

$$\mathbf{s}_0 \rightarrow \mathbf{a}_1 \rightarrow \mathbf{s}_1 \rightarrow \mathbf{a}_2 \rightarrow \mathbf{s}_2 \rightarrow \cdots \rightarrow \mathbf{a}_t \rightarrow \mathbf{s}_t \rightarrow \mathbf{a}_{t+1} \rightarrow \mathbf{s}_{t+1} \rightarrow \cdots \rightarrow \mathbf{a}_T \rightarrow \mathbf{s}_T$$

$$r_1 \quad r_2 \quad r_t \quad r_{t+1} \quad r_T$$

$$\mathbf{V}(\mathbf{s}) = \max_{\mathbf{a}} \mathbf{Q}(\mathbf{s}, \mathbf{a})$$

$$r_t + \gamma \max_{\mathbf{a}} \mathbf{Q}(\mathbf{s}_{t+1}, \mathbf{a}) - \max_{\mathbf{a}} \mathbf{Q}(\mathbf{s}_t, \mathbf{a}) = 0 \quad \text{for all} \quad (\mathbf{s}_t, r_t, \mathbf{s}_{t+1})$$

Action Value Function (Q-value)

$$\mathbf{s}_0 \rightarrow \mathbf{a}_1 \rightarrow \mathbf{s}_1 \rightarrow \mathbf{a}_2 \rightarrow \mathbf{s}_2 \rightarrow \cdots \rightarrow \mathbf{a}_t \rightarrow \mathbf{s}_t \rightarrow \mathbf{a}_{t+1} \rightarrow \mathbf{s}_{t+1} \rightarrow \cdots \rightarrow \mathbf{a}_T \rightarrow \mathbf{s}_T$$

$$r_1 \quad r_2 \quad r_t \quad r_{t+1} \quad r_T$$

$$r_t + \gamma \max_{\mathbf{a}} \mathbf{Q}(\mathbf{s}_{t+1}, \mathbf{a}) - \boxed{\max_{\mathbf{a}} \mathbf{Q}(\mathbf{s}_t, \mathbf{a})} = 0 \quad \text{for all} \quad (\mathbf{s}_t, r_t, \mathbf{s}_{t+1})$$

Find the best action $\mathbf{a}_t$ for state $\mathbf{s}_t$

Q-learning

$$r_t + \gamma \max_{\mathbf{a}} \mathbf{Q}(\mathbf{s}_{t+1}, \mathbf{a}) - \mathbf{Q}(\mathbf{s}_t, \mathbf{a}_t) = 0 \quad \text{for all} \quad (\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})$$

Q-learning (annoying details)

$$r_t + \boxed{\gamma \max_{\mathbf{a}} \mathbf{Q}(\mathbf{s}_{t+1}, \mathbf{a})} - \mathbf{Q}(\mathbf{s}_t, \mathbf{a}_t) = 0 \quad \text{for all} \quad (\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})$$

If $\mathbf{s}_t$ is terminal there is no $\mathbf{s}_{t+1}$

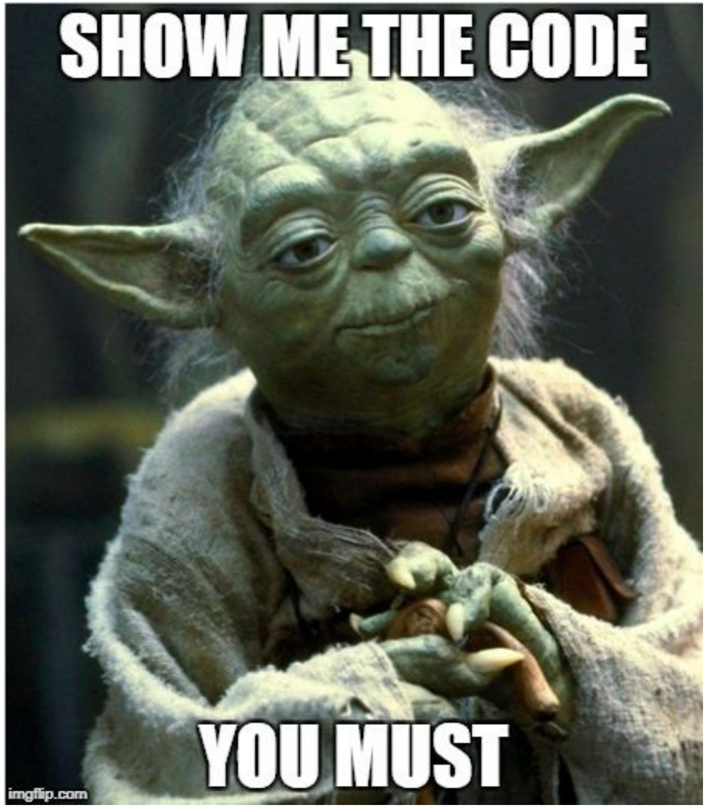
Q-learning

$$y_t - \mathbf{Q}(\mathbf{s}_t, \mathbf{a}_t) = 0 \quad \text{for all} \quad (\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})$$

$$y_t = \begin{cases} r_t + \gamma \max_{\mathbf{a}} \mathbf{Q}(\mathbf{s}_{t+1}, \mathbf{a}) & \text{if } \mathbf{s}_t \text{ not terminal} \\ r_t & \text{otherwise} \end{cases}$$

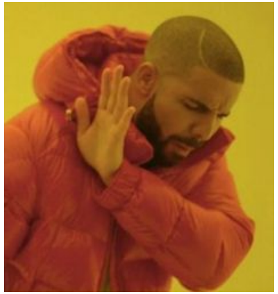
SCIRO RESEARCH

Gathering Experience



What is the Q-value function?

Q: $\boldsymbol{S} \times \boldsymbol{A} \rightarrow \mathbb{R}$




```
def Q(s: State, a: Action) -> float:
```

Tabular Q-learning

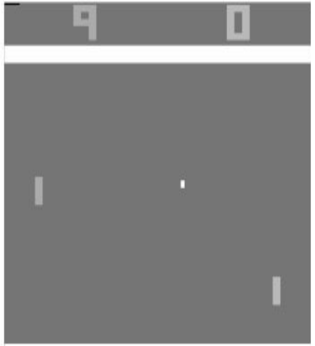
```
def Q(s: State, a: Action) -> float:
    return q[s, a]
```

.... for all $(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})$

Learning from Images

```
def Q(s: int, a: int) -> float:
    return q[s, a]
```

210



160

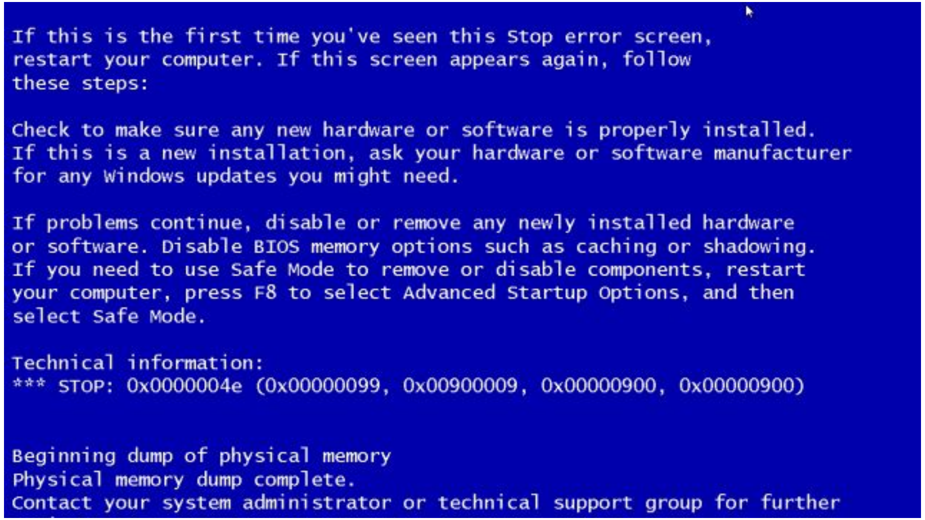
$255^{210*160}$ states

left, right, up, down, fire, NOOP

6 actions

Learning from Images

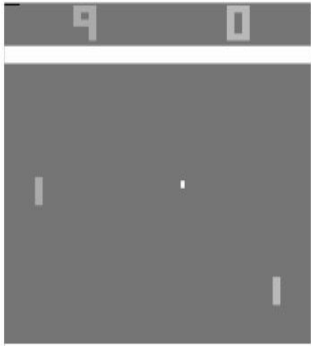
```
q = np.zeros((255**33600, 6))
```



Learning from Images

```
def Q(s: int, a: int) -> float:
    ???
```

210



160

$255^{210*160}$ states

left, right, up, down, fire, NOOP

6 actions

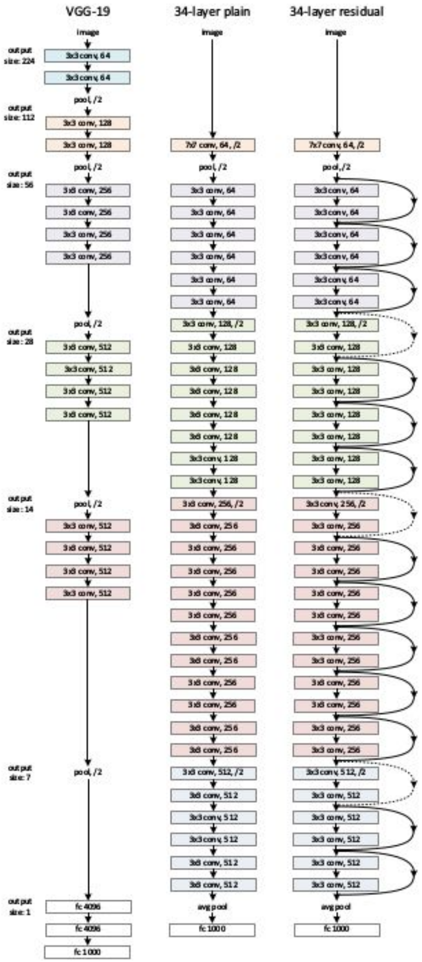
Deep Neural Networks

- Universal function approximators
- Lots of linear algebra run on GPU
- Typically gigabytes of parameters

architecture

$$y = f_{\theta}(x)$$

output parameters input



Neural Networks as Computational Graph

$$f(x, y)_\theta = ax^2 + bxy + cy^2$$

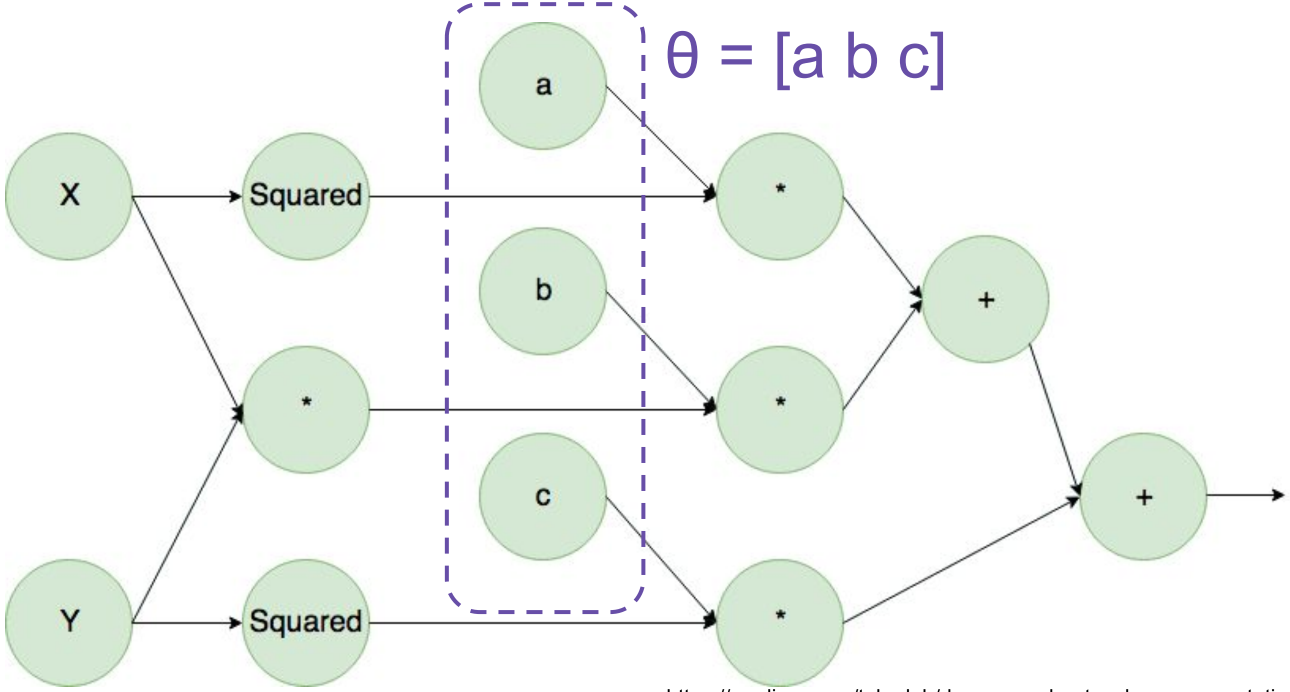
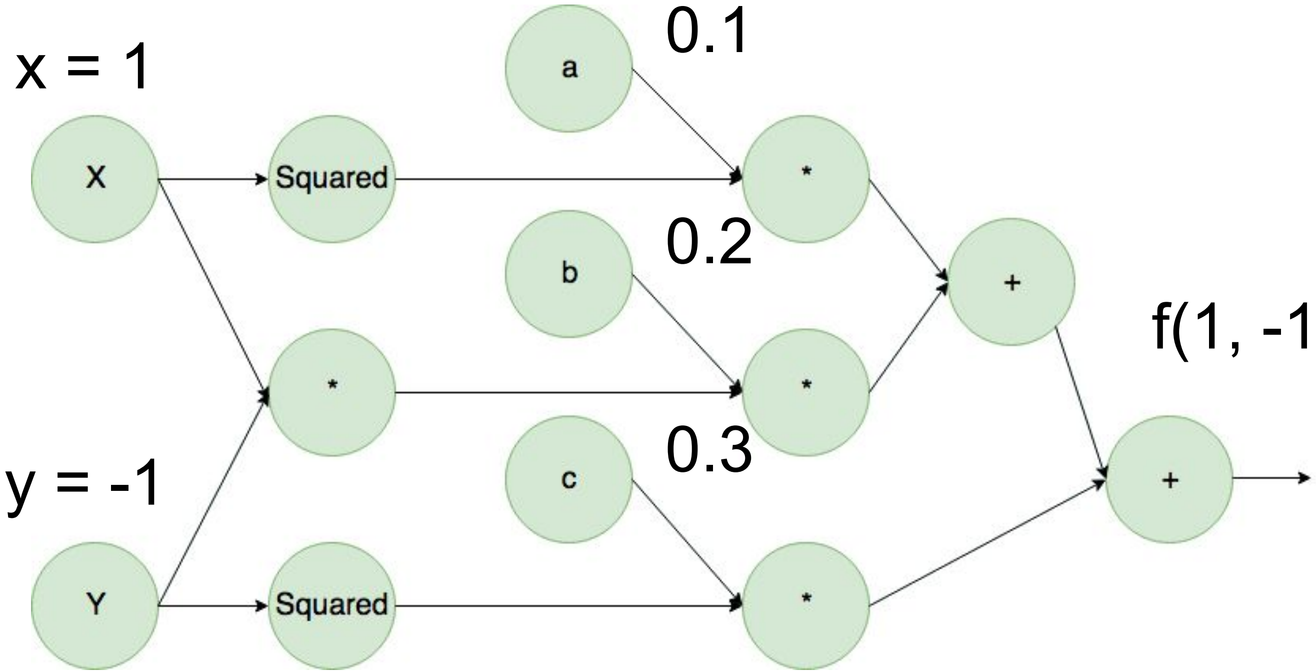


Image adapted from:
https://medium.com/tebs-lab/deep-neural-networks-as-computational-graphs-867fcaa56ac9

Example

$$f(x, y)_\theta = 0.1x^2 + 0.2xy + 0.3y^2$$

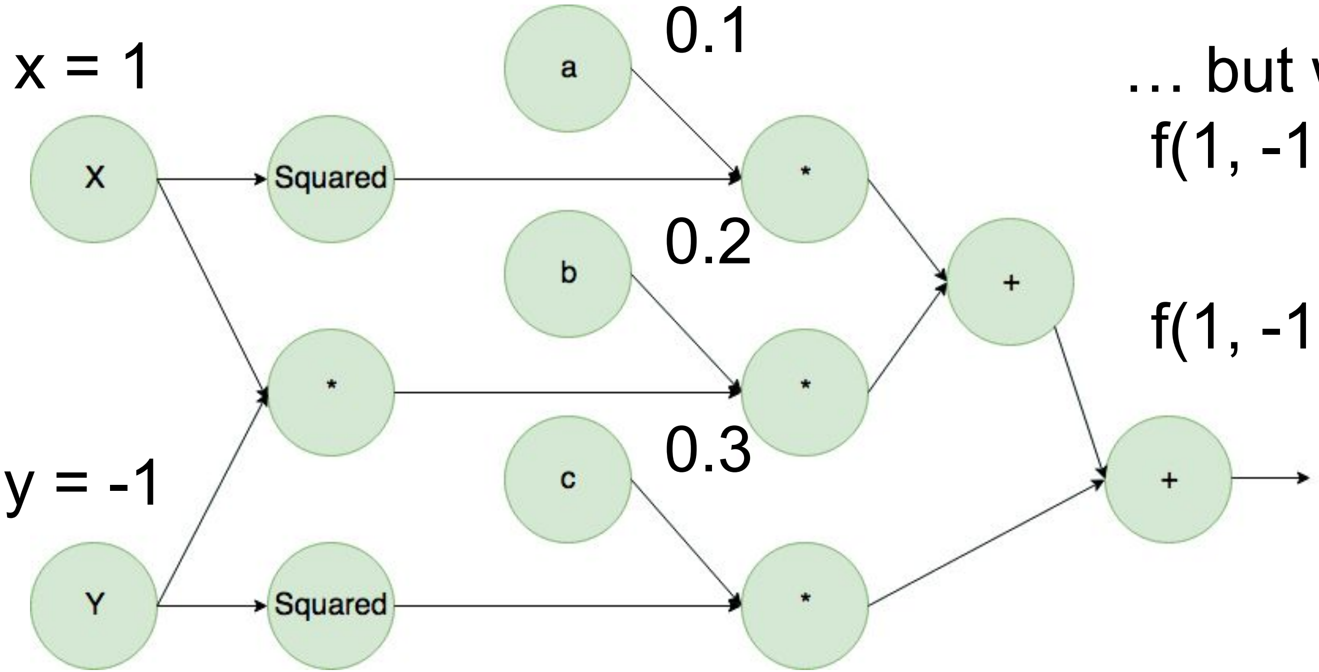


$$f(1, -1)_\theta = 0.2$$

SCIRO RESEARCH

Example

$$f(x, y)_\theta = 0.1x^2 + 0.2xy + 0.3y^2$$

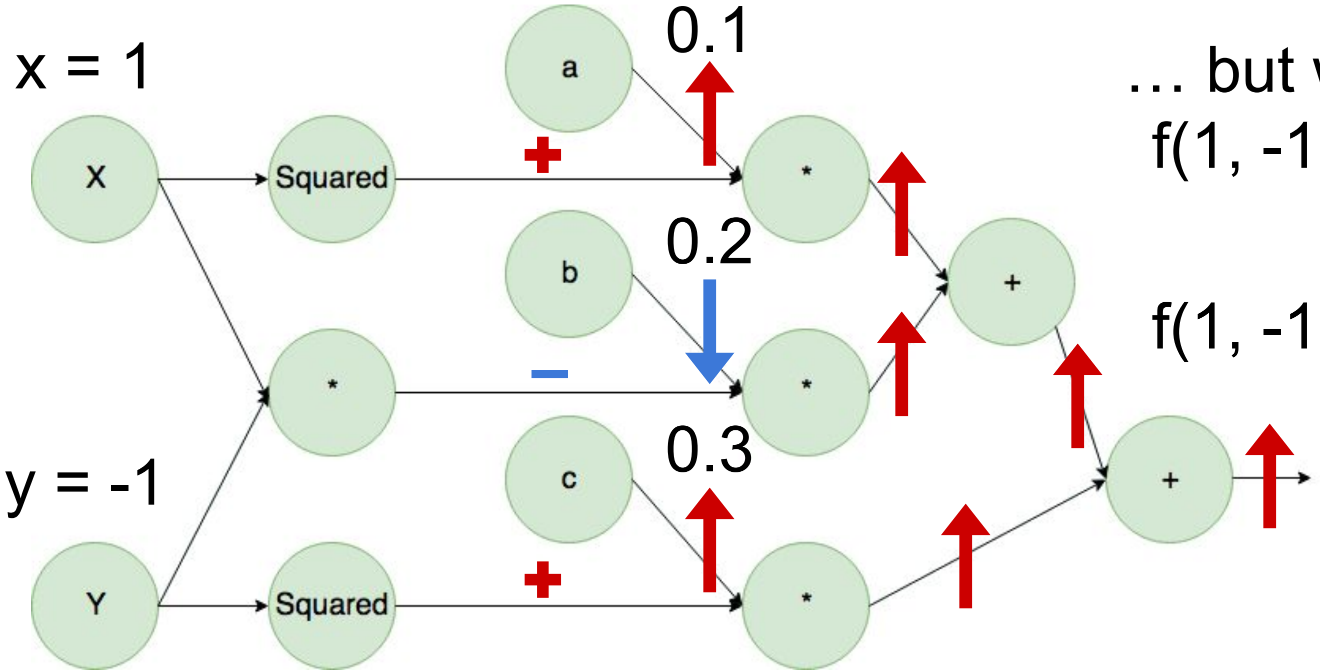


… but we want $f(1, -1)_\theta = 0.5$

$f(1, -1)_\theta = 0.2$

Automatic Differentiation

$$f(x, y)_\theta = 0.1x^2 + 0.2xy + 0.3y^2$$



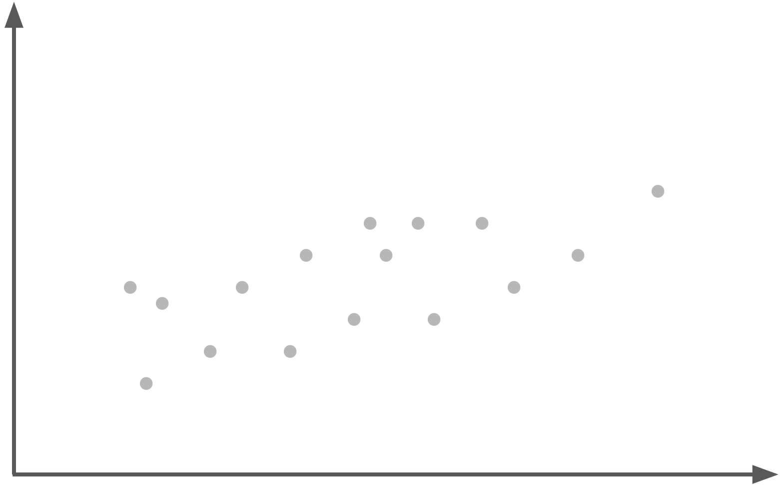
… but we want $f(1, -1)_\theta = 0.5$

$f(1, -1)_\theta = 0.2$

Automatic Differentiation

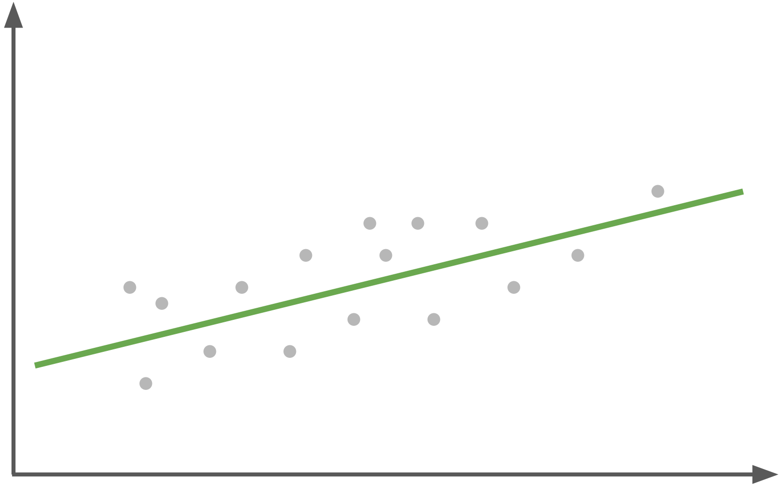
- Automatic differentiation packages calculate gradients automatically
- Deep learning packages are essentially fancy automatic differentiation libraries

Linear Regression Example



Data: $\{(x_i, y_i)\}^N_{i=1}$

Linear Regression Example



<u>Data:</u> $\{(x_i, y_i)\}^N_{i=1}$

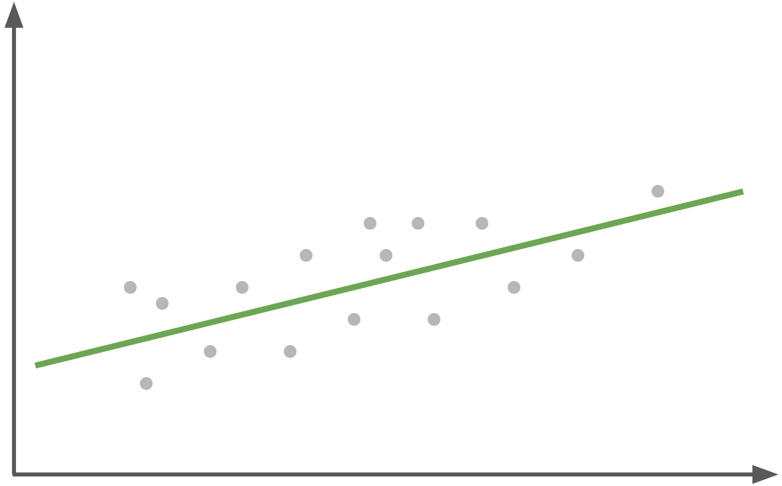
<u>2D line:</u>

$y = w_2x + w_1$

<u>Linear algebra notation:</u>

$y = \mathbf{w}^T\mathbf{x}$

Linear Regression Example



__Data:__ $\{(x_i, y_i)\}^N_{i=1}$

__2D line:__

$y = w_2 x + w_1$

__Linear algebra notation:__

$y = \mathbf{w}^T\mathbf{x}$

__Loss:__

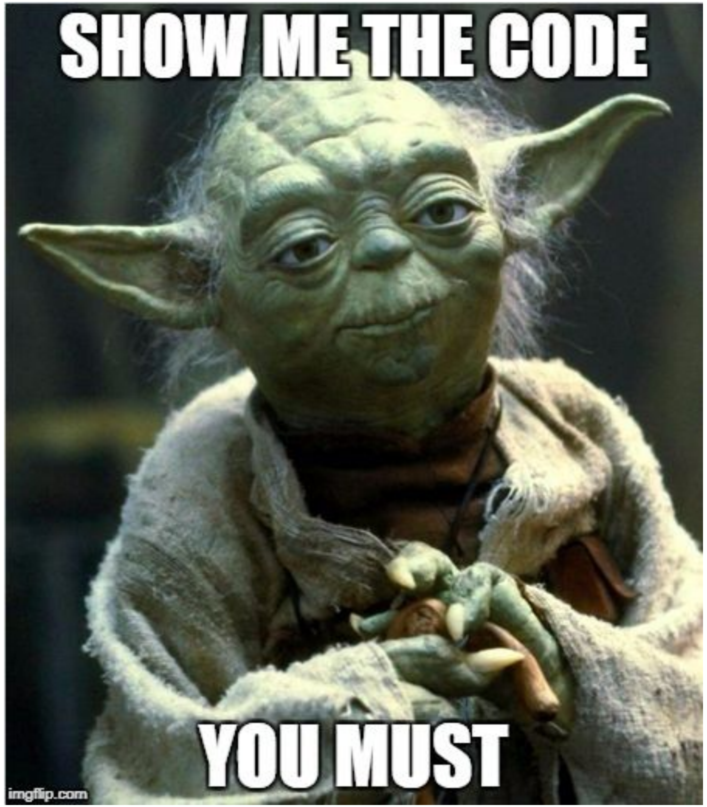
$L_{\mathbf{w}}(D) = (1 / N)\Sigma^N_{i=1}(y_i - \mathbf{w}^T\mathbf{x}_i)^2$

Chainer & CuPy

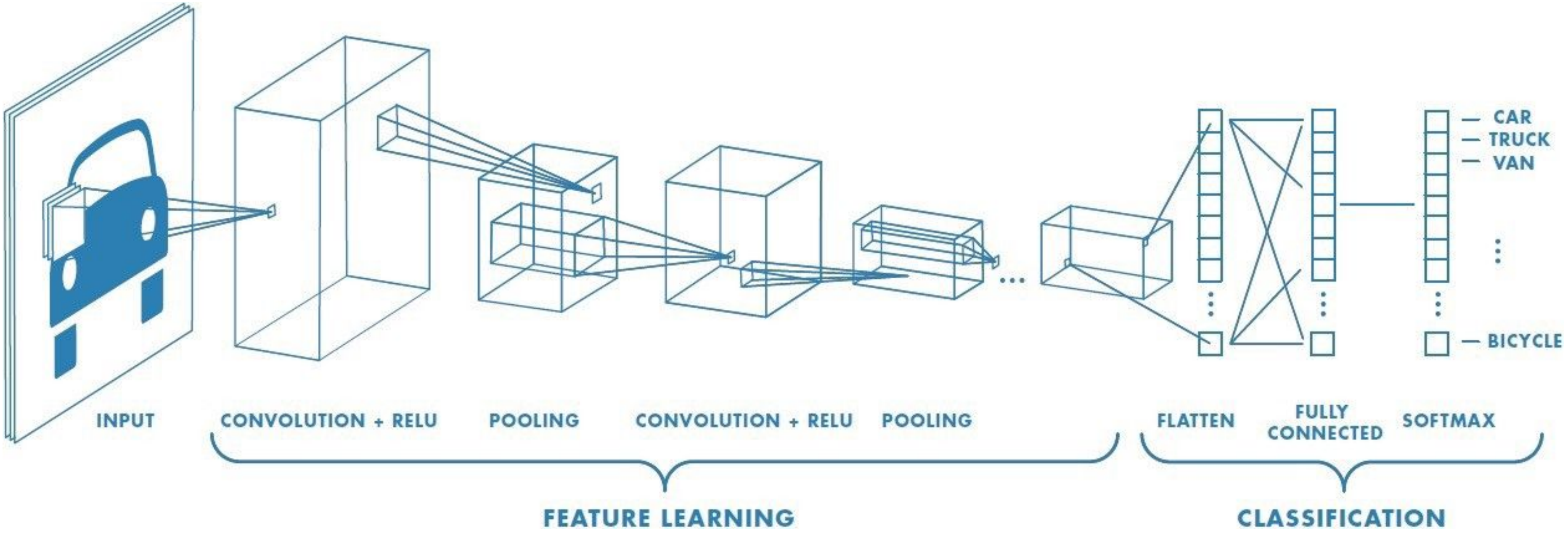
Chainer

CuPy

Chainer & CuPy

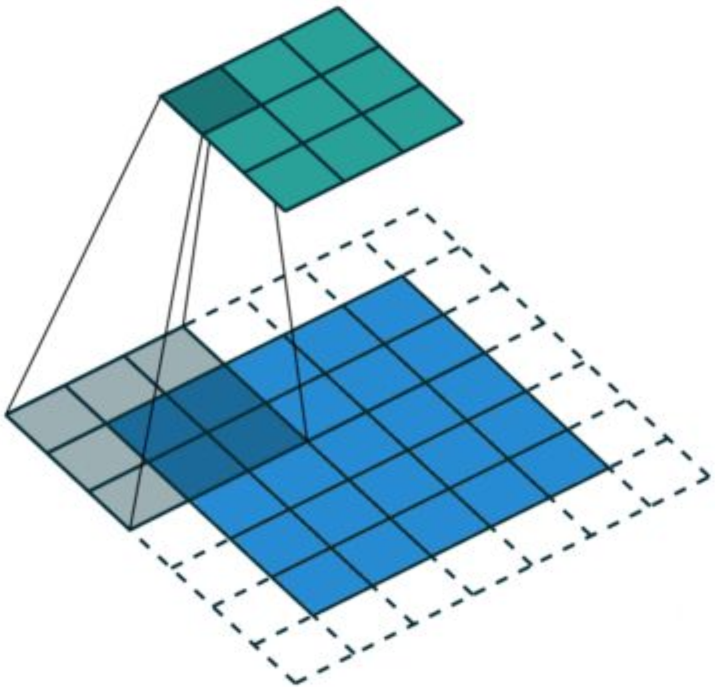


Convolutional Neural Networks (CNNs)



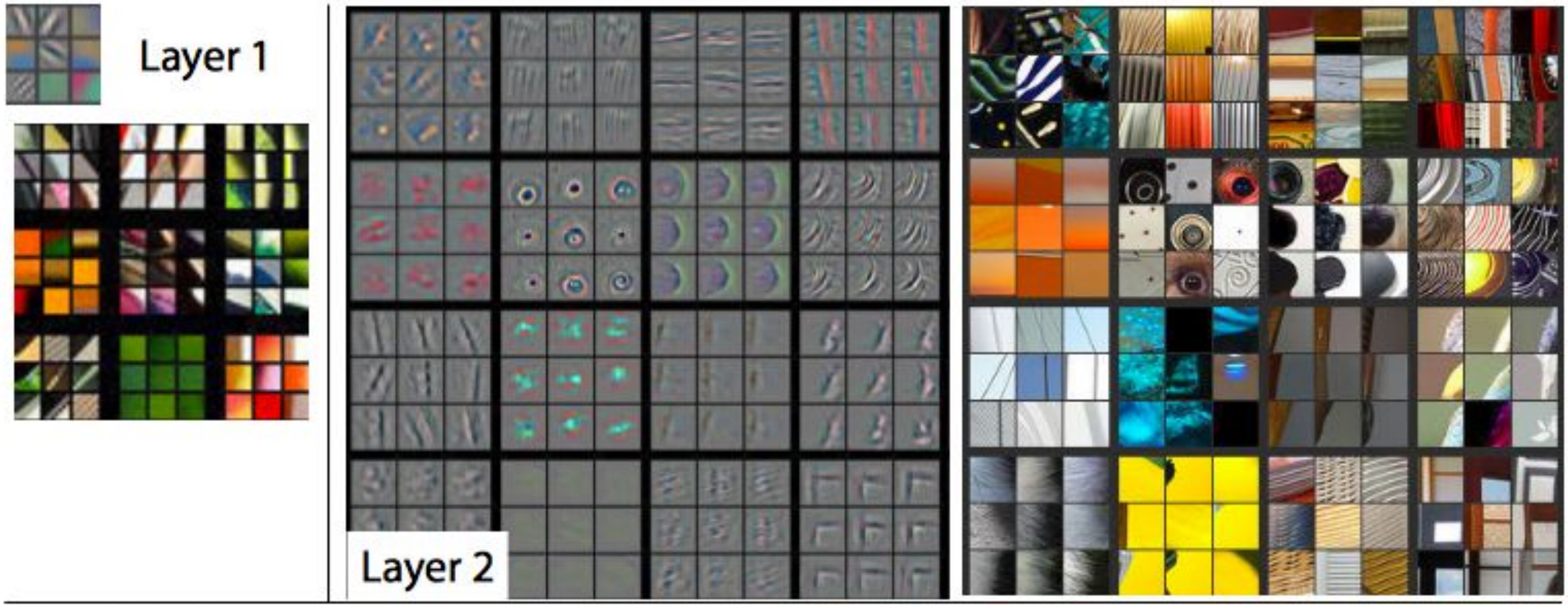
https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Image Convolution



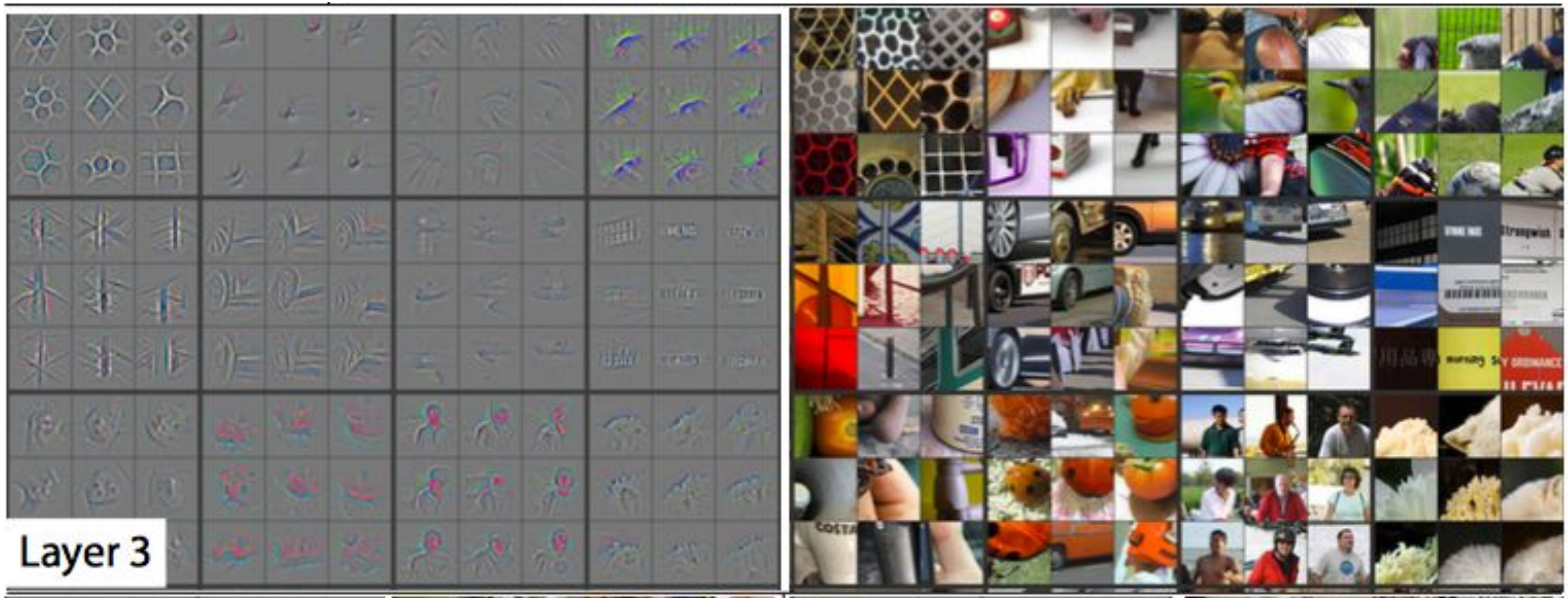
https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Why CNNs?



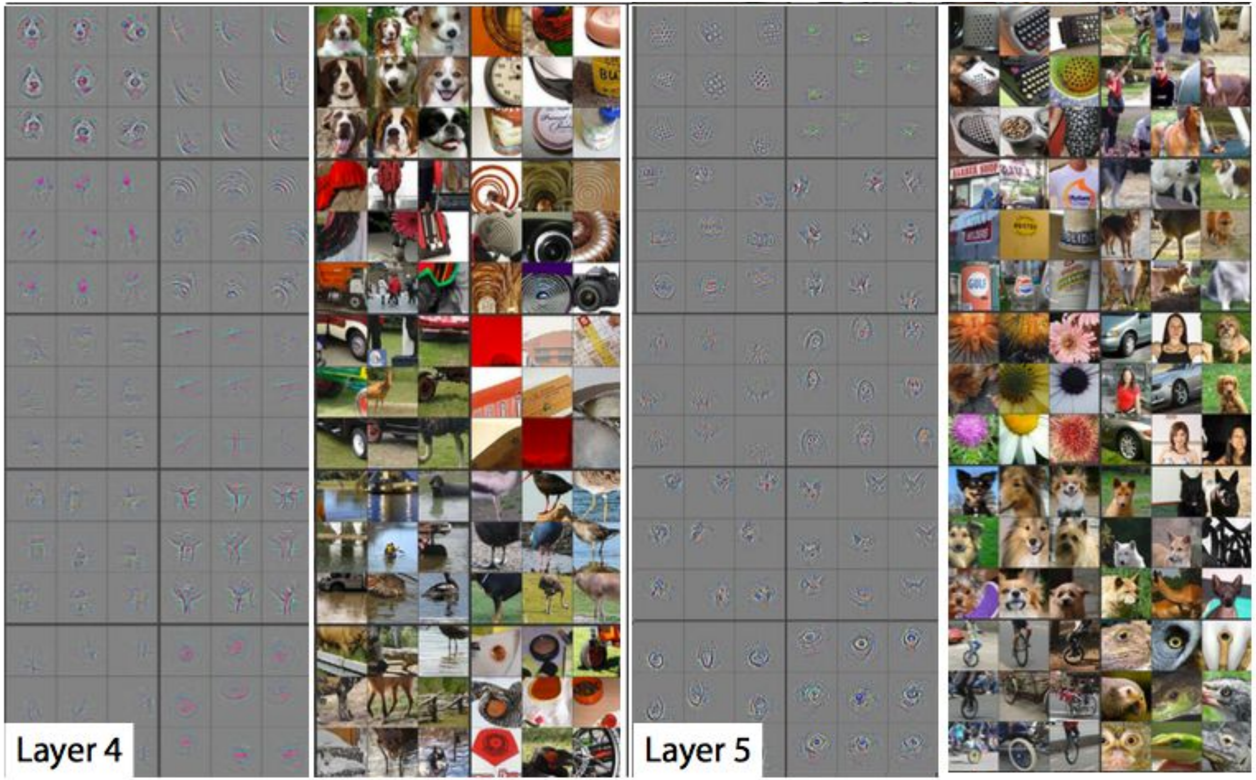
Zeiler, M. D., and R. Fergus. "Visualizing and understanding convolutional networks.", 2013

Why CNNs?



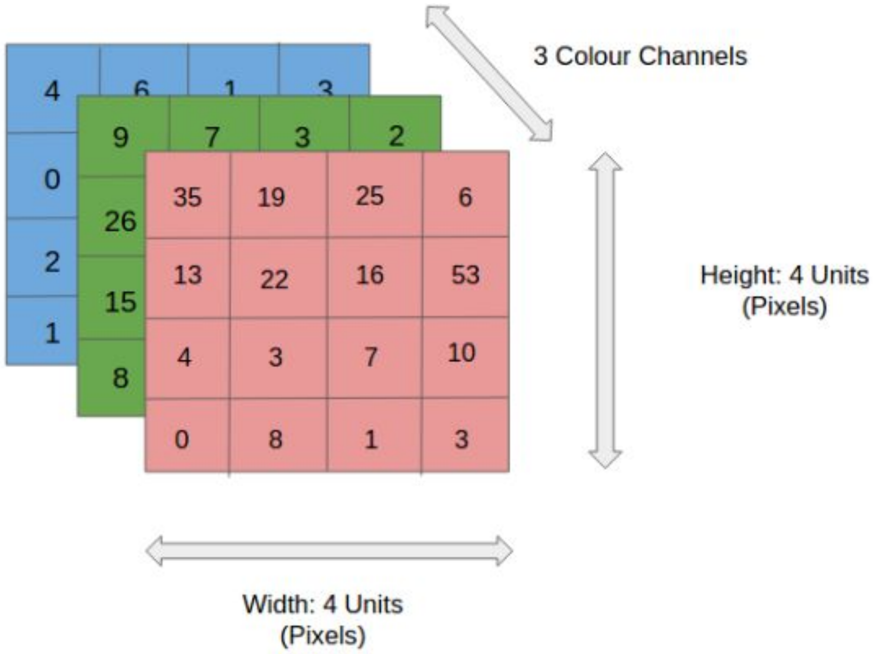
Zeiler, M. D., and R. Fergus. "Visualizing and understanding convolutional networks.", 2013

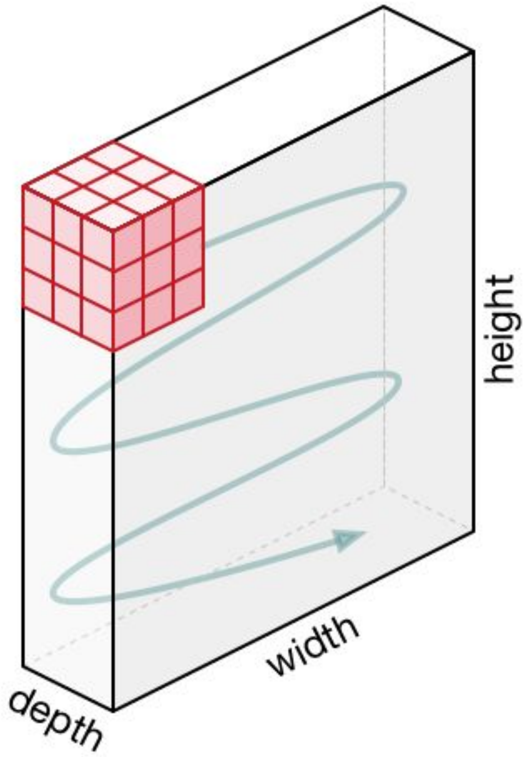
Why CNNs?



Zeiler, M. D., and R. Fergus. "Visualizing and understanding convolutional networks.", 2013

Image Convolution Kernels



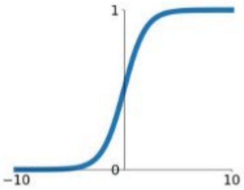


https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Activation Functions

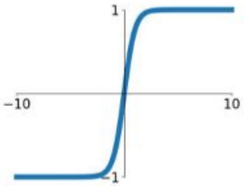
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



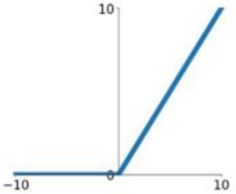
tanh

$\tanh(x)$



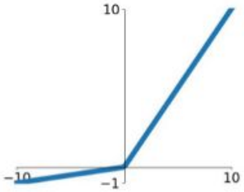
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$

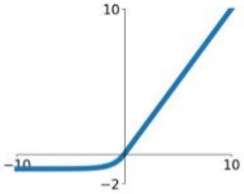


Maxout

$\max(w_1^T x + b_1, w_2^T x + b_2)$

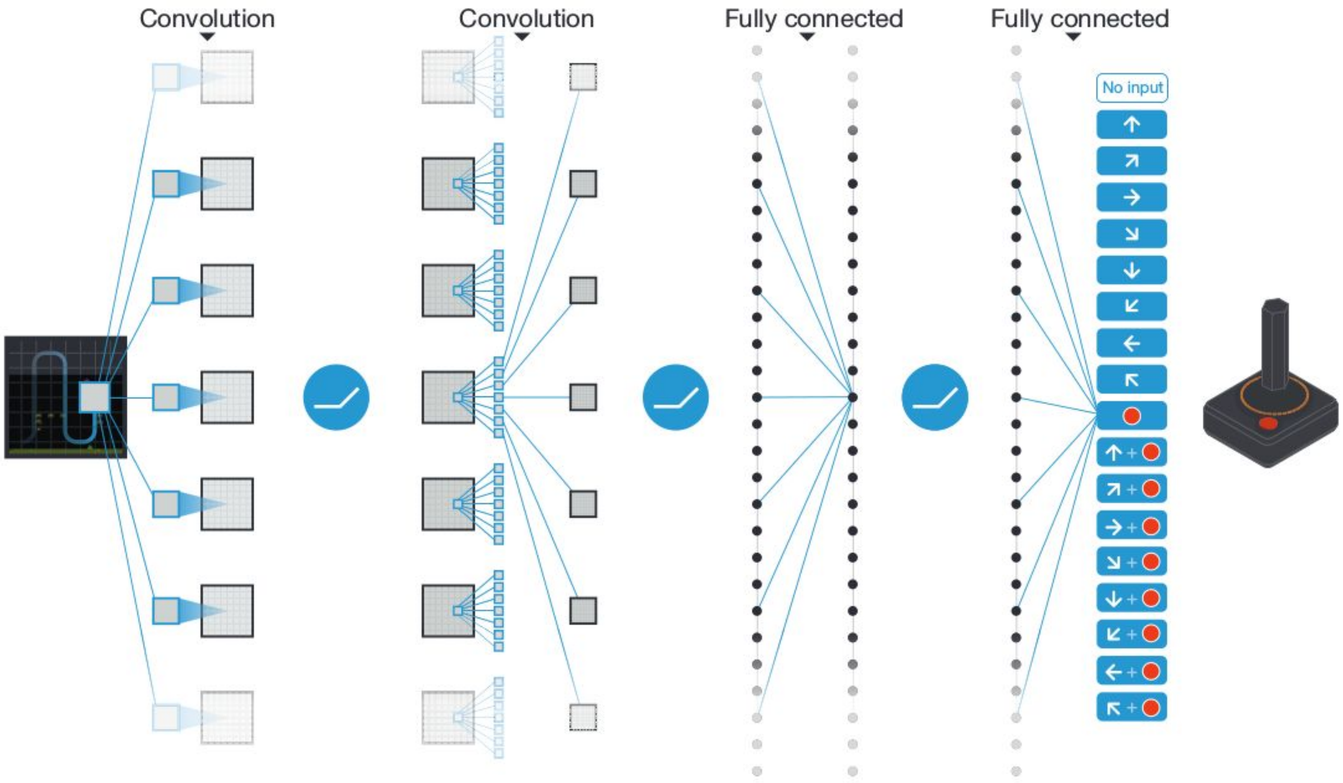
ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

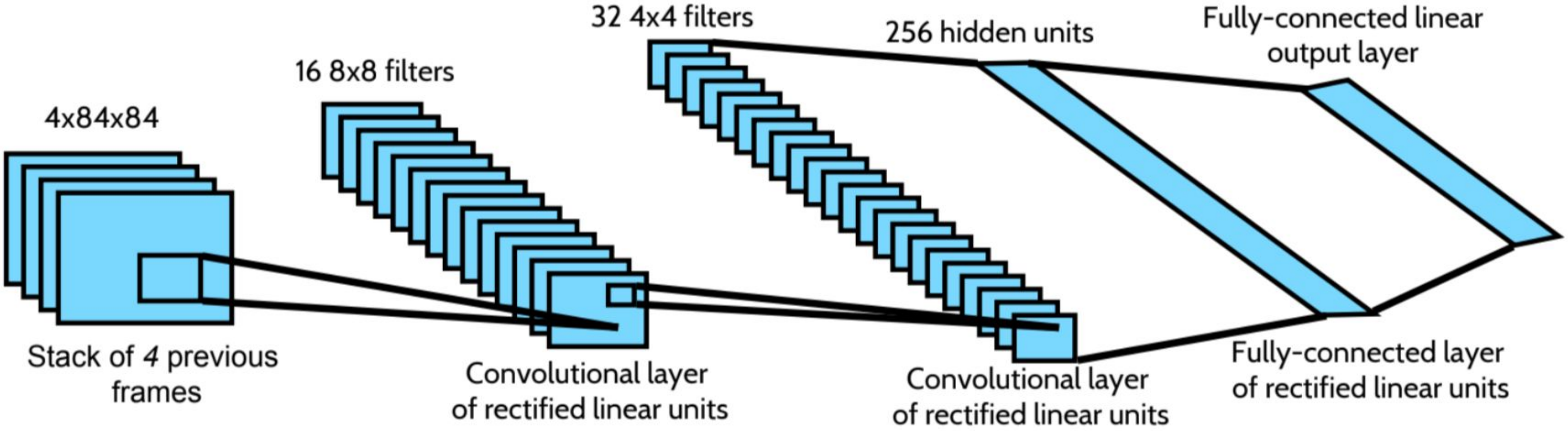


https://towardsdatascience.com/complete-guide-of-activation-functions-34076e95d044

Deep Q-value Network (DQN)

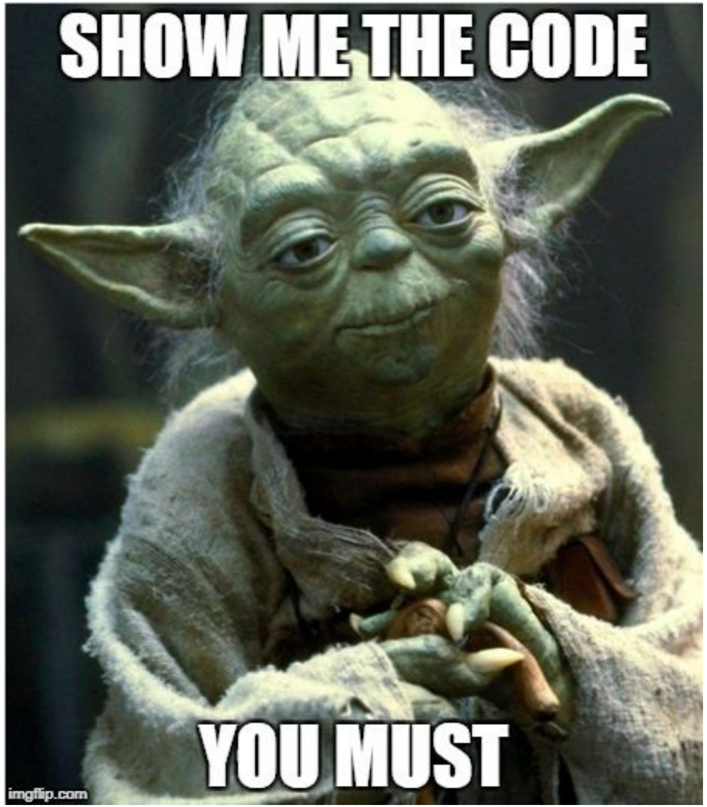


Deep Q-value Network (DQN)



https://medium.com/@jonathan_hui/rl-dqn-deep-q-network-e207751f7ae4

Implementing DQN



Putting Everything Together

Q-learning

$$y_t - \mathbf{Q}(\mathbf{s}_t, \mathbf{a}_t) = 0 \quad \text{for all} \quad (\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})$$

$$y_t = \begin{cases} r_t + \gamma \max_{\mathbf{a}} \mathbf{Q}(\mathbf{s}_{t+1}, \mathbf{a}) & \text{if } \mathbf{s}_t \text{ not terminal} \\ r_t & \text{otherwise} \end{cases}$$

Q-learning

$$r_t + \gamma \max_{\mathbf{a}} \mathbf{Q}(\mathbf{s}_{t+1}, \mathbf{a}) - \mathbf{Q}(\mathbf{s}_t, \mathbf{a}_t) = 0 \quad \text{for all} \quad (\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})$$

SCIRO RESEARCH

Q-learning (annoying details cont'd)

$$r_t + \gamma \boxed{\max_{\mathbf{a}} \mathbf{Q}(\mathbf{s}_{t+1}, \mathbf{a}) - \mathbf{Q}(\mathbf{s}_t, \mathbf{a}_t)} = 0 \quad \text{for all} \quad (\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})$$

If we update **Q** every timestep there is too much noise

Q-learning (annoying details cont'd)

$$r_t + \gamma \max_{\mathbf{a}} \mathbf{Q}_{\theta-}(\mathbf{s}_{t+1}, \mathbf{a}) - \mathbf{Q}_{\theta}(\mathbf{s}_t, \mathbf{a}_t) = 0 \quad \text{for all} \quad (\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})$$

- Update $\mathbf{Q}_{\theta}$ every timestep
- Periodically Set $\mathbf{Q}_{\theta-} = \mathbf{Q}_{\theta}$

Q-learning

$$y_t - \mathbf{Q}_{\boldsymbol{\theta}\text{-}}(\mathbf{s}_t, \mathbf{a}_t) = 0 \quad \text{for all} \quad (\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})$$

$$y_t = \begin{cases} r_t + \gamma \max_{\mathbf{a}} \mathbf{Q}_{\boldsymbol{\theta}}(\mathbf{s}_{t+1}, \mathbf{a}) & \text{if } \mathbf{s}_t \text{ not terminal} \\ r_t & \text{otherwise} \end{cases}$$

SCIRO RESEARCH

Exploration vs. Exploitation

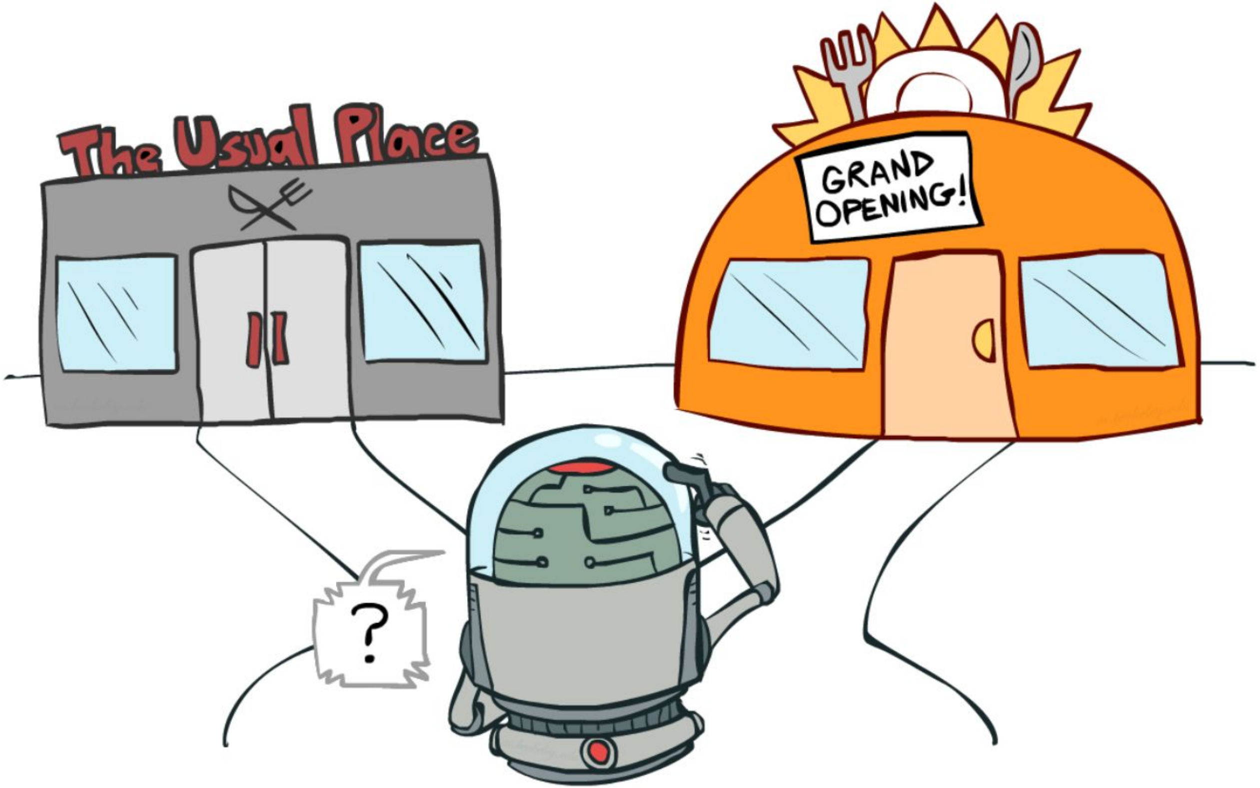


Image source: UC Berkeley AI course

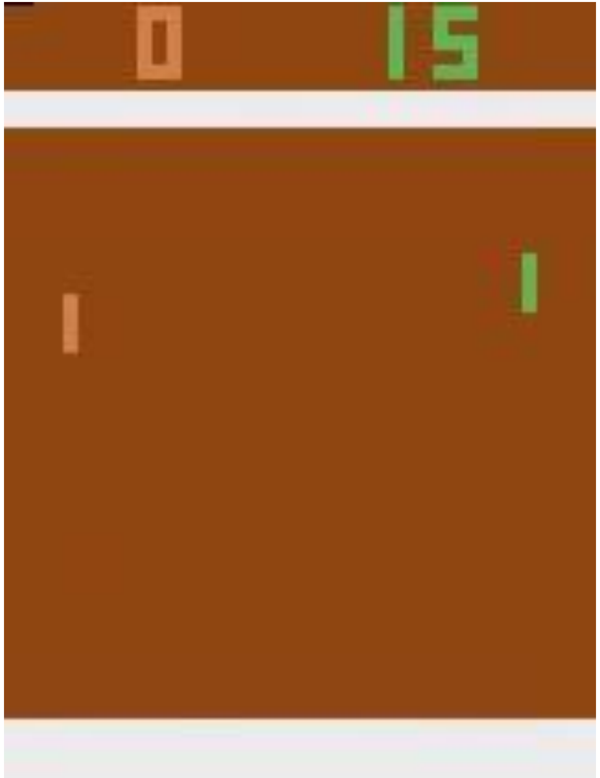
ε-greedy Policy

- Choose random action with probability ε
- Full exploration ε = 1.0
- Full exploitation ε= 0.0

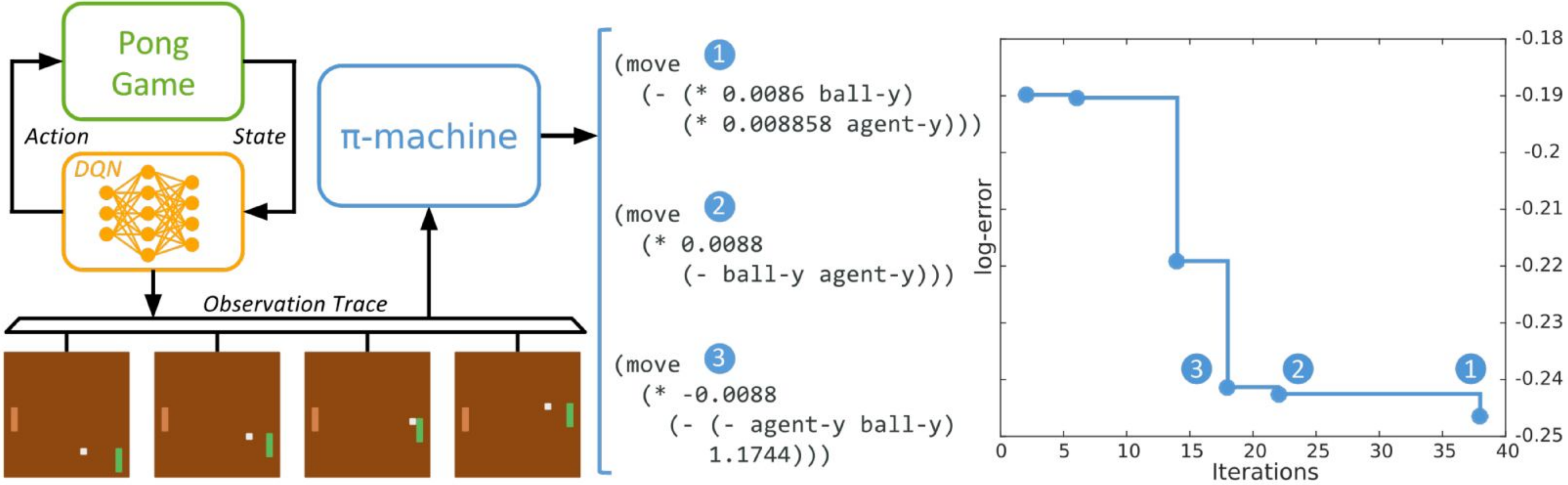
Training

- ~24 hours on i7 laptop with GTX1050
- Can be taken down to ~1 hour with a beefy machine and a few optimisations
- Deep RL is notorious for irreproducability
- Sometimes multiple runs are needed to get a working agent

Playing Better Than Human



Deep Q-learning Network



Human: 9.3 Program agent: 11.1 DQN: 18.9

Penkov & Ramamoorthy, ICML-WHI, 2017

SCIRO RESEARCH

Starting soon...

AI MASTERCLASS by SCIRO

Be the first one to get an invite to apply!

sciro.ai