3-14-18-ReenforcementLearning-2019-5-4

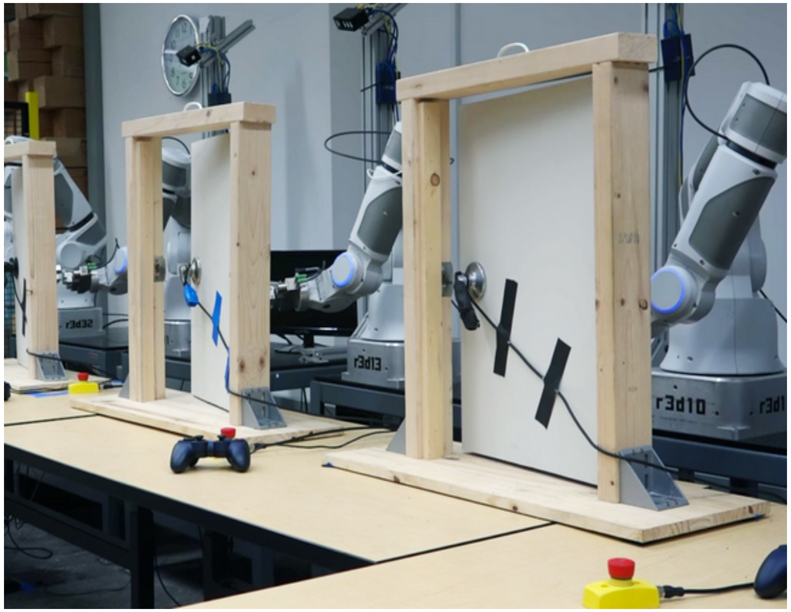
CTI One Corporation

Version: x0.4
Date: May 4, 2019
Project Lead: Harry Li, Ph.D.

Team members:

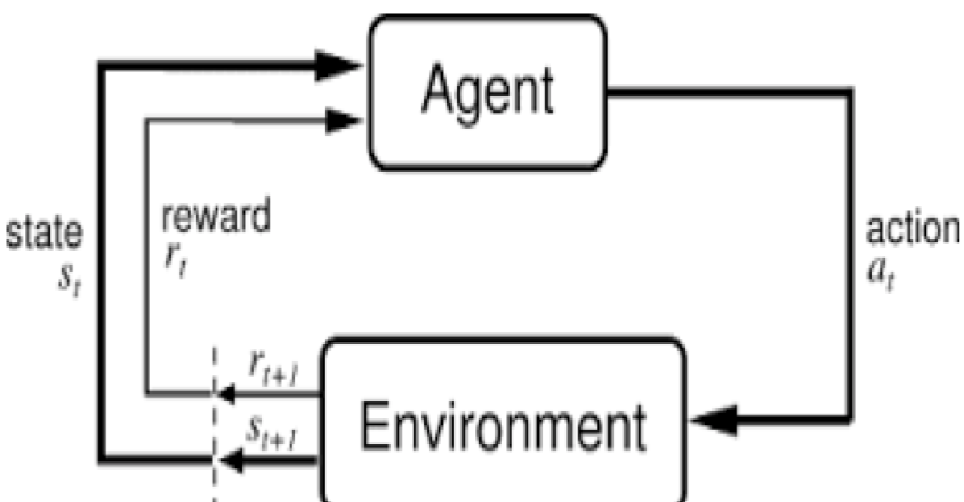
May-4-2019 Reenforcement Learning

https://blog.floydhub.com/robotic-arm-control-deep-reinforcement-learning/



Google's robot arms opening doors

How OpenAI 5 featurizes its space



Model free: the algorithm doesn’t need any internal details of how the robot works, no need to compute differential equations. All it needs is low level observations like the positions of joints

Deterministic: the algorithm will always run the same way on the same test examples which makes it easy to debug when things aren’t working well

Harry Li, Ph.D.

May-4-2019 Controlling 2 Link Robot Arm

https://github.com/MorvanZhou/train-robot-arm-from-scratch/tree/master/part1

MorvanZhou / **train-robot-arm-from-scratch**

Reference: google ai robotics team
https://ai.google/research/teams/brain/robotics/

Google principal scientist

1992~	出生	湖南人(Hunan, China)	
2004~2010	中学	雅礼中学, 长沙	Email:
2010~2012	本科, 土木工程	桂林理工	morvanzhou@hotmail.com
2013~2015	Bachelor, Civil Engineering	Griffith University, Australia	
2015~2018	PhD, Intelligent Transportation	Griffith University(Supervisor: Prof.	
2015~2016	Internship	National ICT Australia	
2016~2018	Visiting PhD	UTS, Australia(Supervisor: Prof. Xia	
2017~2018	Internship	Sydney Trains, Australia	
2018~	深度学习,推荐系统	腾讯, 深圳, 中国	

https://morvanzhou.github.io/about/

https://github.com/msaroufim

Founder at Yuri.ai | Formerly Applied Scientist, Dev and PM at Microsoft AI & Research

GitHub, Inc. (US)

Mark Saroufim

May-4-2019 Temporal Difference For Robot Control

https://www.youtube.com/watch?v=VO1mCjHvzlo

Temporal Difference Models (TDM)

The tabular TD(0) method

Estimates the state value function of a finite-state Markov decision process (MDP) under a policy π

$$V^\pi(s) = E_\pi\left\{\sum_{t=0}^{\infty} \gamma^t r_t \middle| s_0 = s\right\} \quad \ldots (1)$$

Where V is a state value function under policy pi, s is a state, r is reward, and discount rate gamma.

E_pi : expectation under policy pi for all reward r(t) for the time t=0 to infinity, with each instance of discount rate gamma, given initial state s(t)_{t=0}

Next state value function V at s_1:

$$V^\pi(s) = E_\pi\{r_0 + \gamma V^\pi(s_1) | s_0 = s\} \quad \ldots(2)$$

Temporal difference (TD) learning refers to a class of model-free reinforcement learning methods which learn by bootstrapping from the current estimate of the value function. These methods sample from the environment, like Monte Carlo methods, and perform updates based on current estimates, like dynamic programming methods.[1]

https://en.wikipedia.org/wiki/Temporal_difference_learning

Repeatedly evaluate state value function, with positive learning rate alfa

$$V(s) \leftarrow V(s) + \alpha(\overbrace{r + \gamma V(s')}^{\text{The TD target}} - V(s)) \quad \ldots(3)$$

Where $r + \gamma V(s')$ is known as TD target

TD to explain many aspects of behavioral research.[11] It has also been used to study conditions of the consequences of pharmacological manipulations of dopamine on learning.[12]

Harry Li, Ph.D.

May-4-2019 Hamilton–Jacobi–Bellman (HJB) equation

https://en.wikipedia.org/wiki/Hamilton%E2%80%93Jacobi%E2%80%93Bellman_equation

The Hamilton–Jacobi–Bellman (HJB) equation is a partial differential equation which is central to optimal control theory.[1] The solution of the HJB equation is the value function which gives the minimum cost for a given dynamical system with an associated cost function.

$$V(x(0),0) = \min_u \left\{ \int_0^T C[x(t),u(t)]\,dt + D[x(T)] \right\}$$

$$\dot{x}(t) = F[x(t),u(t)]$$

$$\dot{V}(x,t) + \min_u \left\{ \nabla V(x,t) \cdot F(x,u) + C(x,u) \right\} = 0$$

$$V(x,T) = D(x),$$

May-4-2019 HJB Equation Application to LQG Control

https://en.wikipedia.org/wiki/Hamilton%E2%80%93Jacobi%E2%80%93Bellman_equation

Application to LQG Control

Example, we can look at a system with linear stochastic dynamics and quadratic cost.
If the system dynamics is given by

$$dx_t = (ax_t + bu_t)dt + \sigma dw_t$$

The cost accumulate at the rate $C(x_t, u_t) = r(t)u_t^2/2 + q(t)x_t^2/2$

The HJB equation is given:

$$-\frac{\partial V(x,t)}{\partial t} = \frac{1}{2}q(t)x^2 + \frac{\partial V(x,t)}{\partial x}ax - \frac{b^2}{2r(t)}\left(\frac{\partial V(x,t)}{\partial x}\right)^2 + \frac{\sigma^2}{2}\frac{\partial^2 V(x,t)}{\partial x^2}$$

With optimal action given by

$$u_t = -\frac{b}{r(t)}\frac{\partial V(x,t)}{\partial x}$$

May-4-2019 Google Learning Hand-Eye Coordination

Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection

Sergey Levine SLEVINE@GOOGLE.COM
Peter Pastor PETERPASTOR@GOOGLE.COM
Alex Krizhevsky AKRIZHEVSKY@GOOGLE.COM
Deirdre Quillen DEQUILLEN@GOOGLE.COM
Google

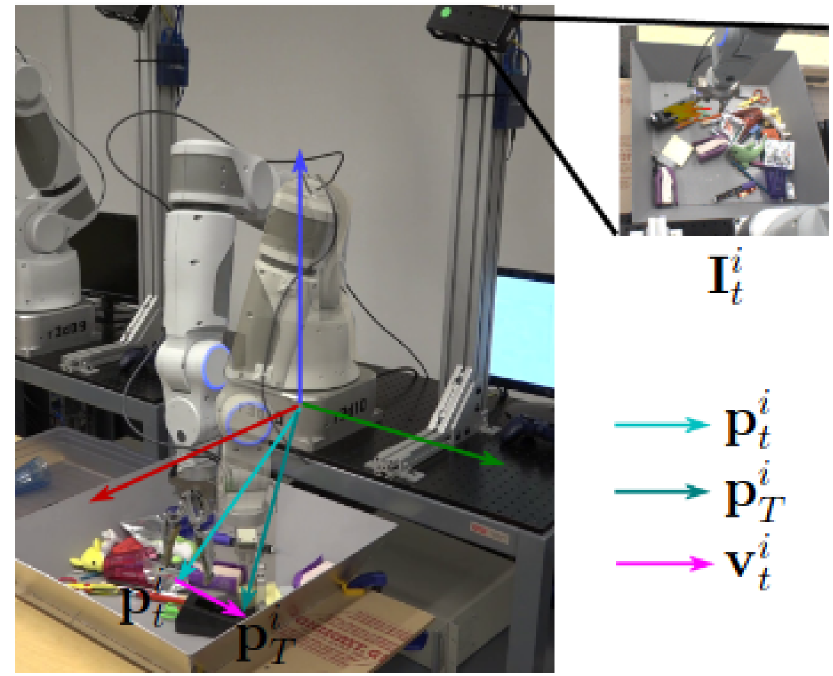
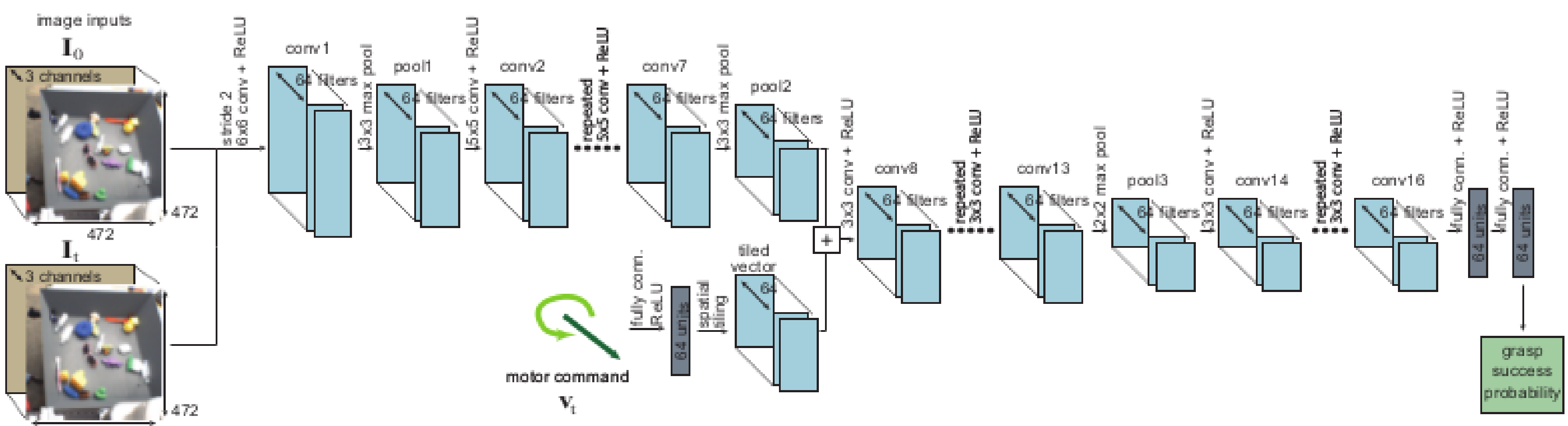


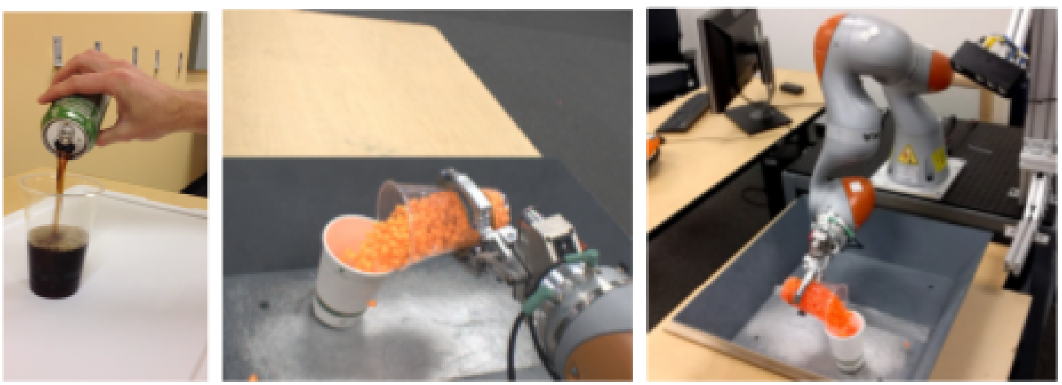
Figure 4. The architecture of the CNN grasp predictor.

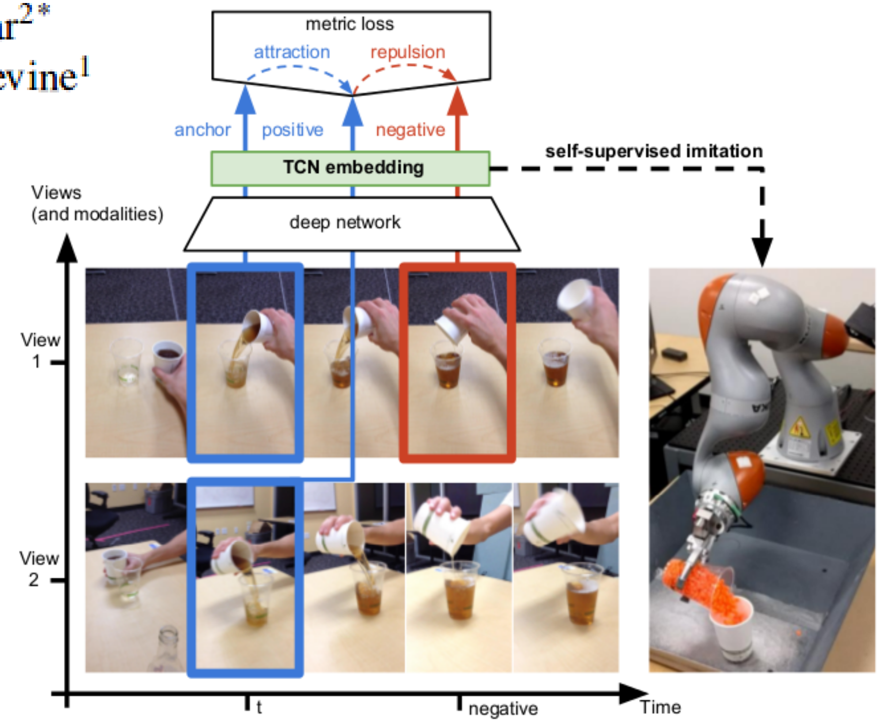


May-4-2019 Google AI Team's Publication

Time-Contrastive Networks: Self-Supervised Learning from Video

Pierre Sermanet[1*@] Corey Lynch[1R*] Yevgen Chebotar[2*]
Jasmine Hsu[1] Eric Jang[1] Stefan Schaal[2] Sergey Levine[1]
[1]Google Brain [2]University of Southern California

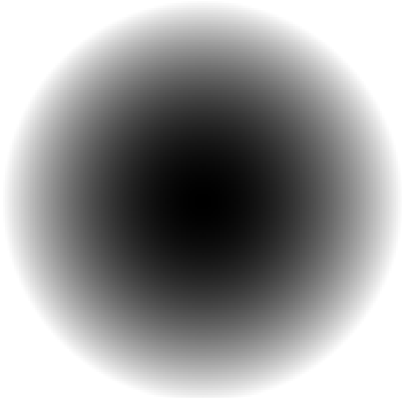


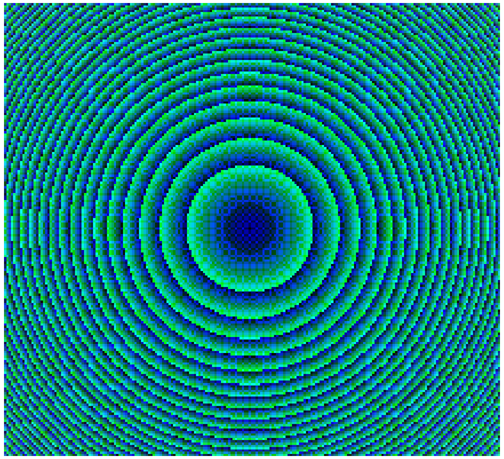


Harry Li, Ph.D.

May-4-2019 Google Depth Image Calibration

https://sites.google.com/site/brainrobotdata/home/depth-image-encoding

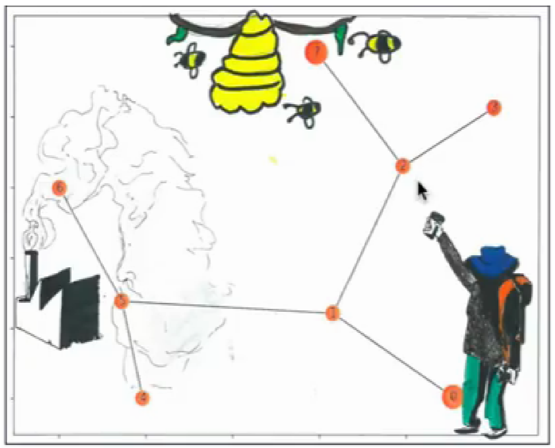


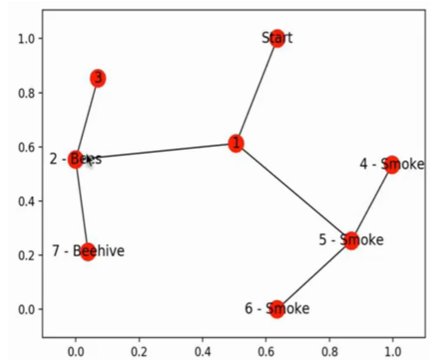


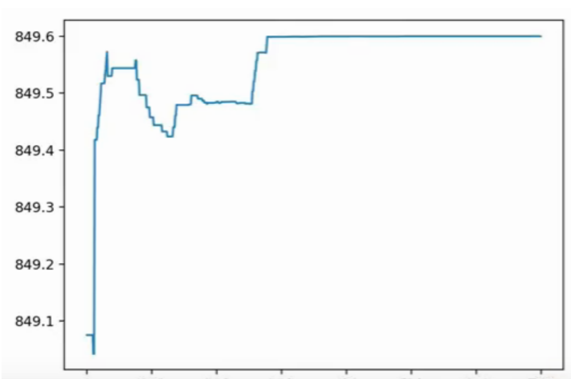
May-5-2019 Reenforcement Learning Tutorial Find the “Honey”

Reinforcement Learning - A Step Closer to AI with Assisted Q-Learning

https://amunategui.github.io/reinforcement-learning/index.html





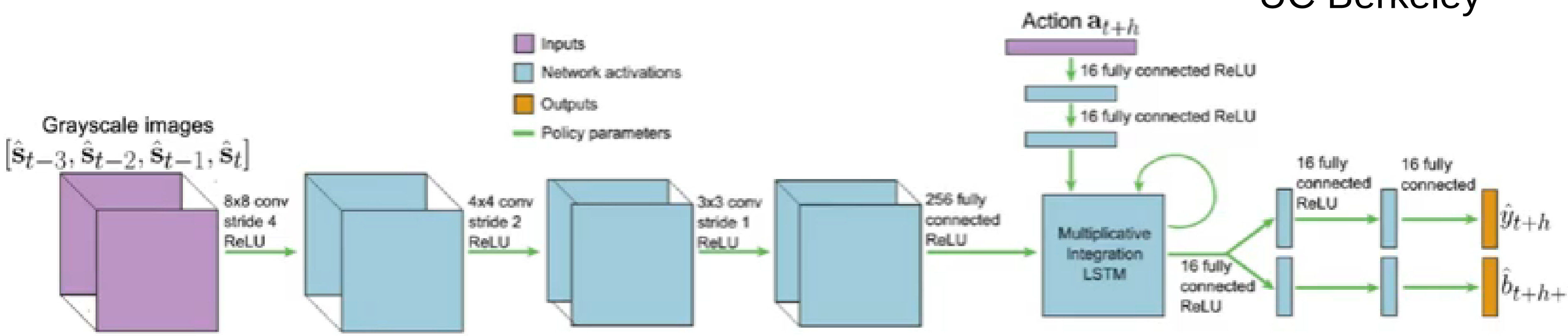


May-4-2019 Reenforcement Learning for Navigation

YouTube https://www.youtube.com/watch?v=vgiW0HlQWVE

Self-supervised Deep Reinforcement Learning with Generalized Computation Graphs for Robot Navigation

UC Berkeley



Use past 4 gray scale images