Deep Reinforcement Learning

Shivaram Kalyanakrishnan
shivaram@cse.iitb.ac.in

Department of Computer Science and Engineering
Indian Institute of Technology Bombay

February 2017

RoboCup Soccer

Objective of the RoboCup Federation:

"By the middle of the 21st century, a team of fully autonomous humanoid robot soccer players shall win a soccer game, complying with the official rules of FIFA, against the winner of the most recent World Cup."

RoboCup Soccer

Objective of the RoboCup Federation:

"By the middle of the 21st century, a team of fully autonomous humanoid robot soccer players shall win a soccer game, complying with the official rules of FIFA, against the winner of the most recent World Cup."

[RoboCup Nao video[1]]

1. `https://www.youtube.com/watch?v=b6Zu5fLUa3c`

Half Field Offense (KLS2007)

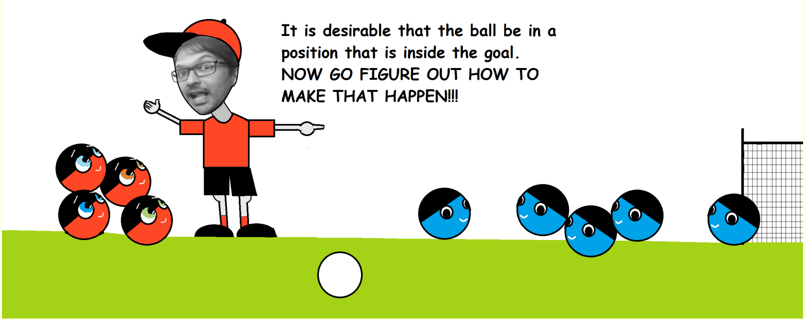
[Video of task[1]]

1. http://www.cs.utexas.edu/~AustinVilla/sim/halffieldoffense/

Half Field Offense (KLS2007)

[Video of task[1]]

Training

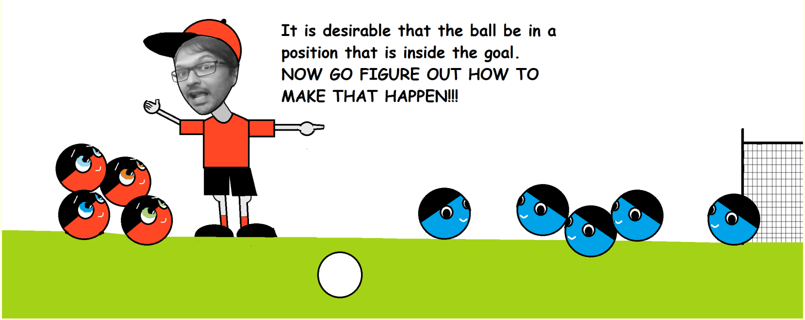


1. `http://www.cs.utexas.edu/~AustinVilla/sim/halffieldoffense/`

Half Field Offense (KLS2007)

[Video of task[1]]

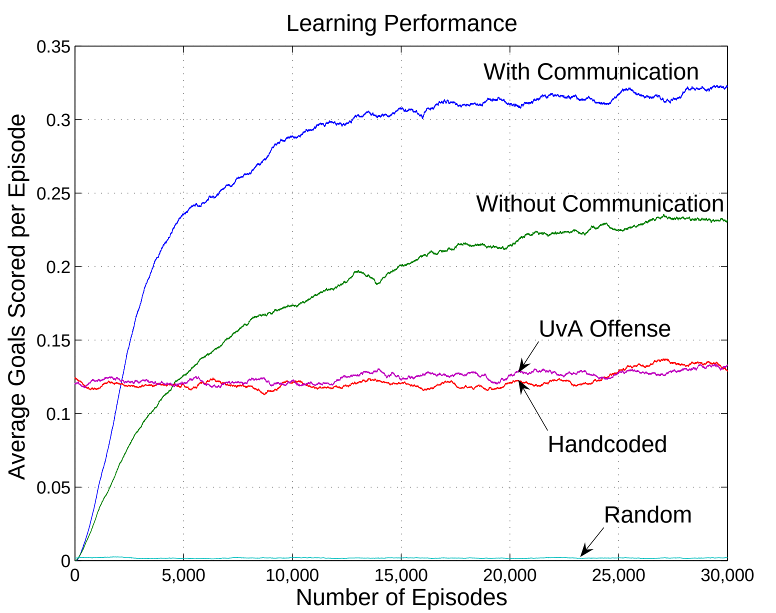
Training



[Video of task after training[2]]

1. http://www.cs.utexas.edu/~AustinVilla/sim/halffieldoffense/

Half Field Offense (KLS2007)



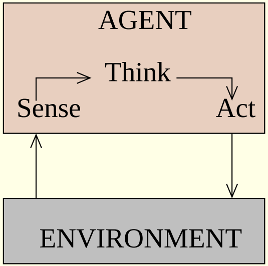
Overview

1. Reinforcement Learning
2. Neural Networks and Deep Learning
3. Deep Reinforcement Learning

Learning to Act Purposefully

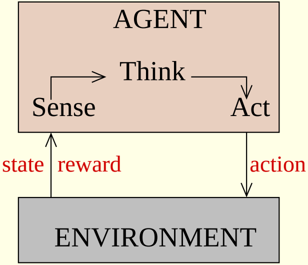
Answer: Reinforcement Learning (RL).

Learning to Act Purposefully



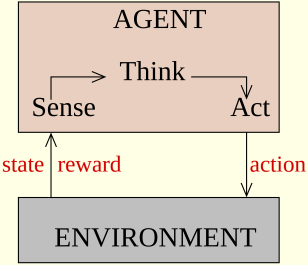
Answer: Reinforcement Learning (RL).

Learning to Act Purposefully



Answer: Reinforcement Learning (RL).

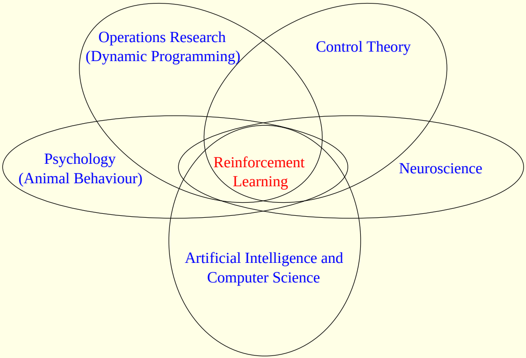
Learning to Act Purposefully



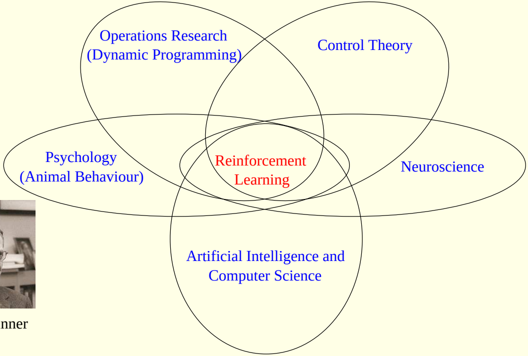
Question: How must an agent in an *unknown* environment act so as to maximise its long-term reward?

Answer: Reinforcement Learning (RL).

Reinforcement Learning: Historical Foundations



Reinforcement Learning: Historical Foundations

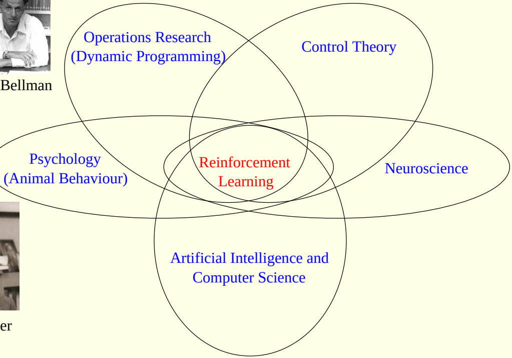


B. F. Skinner

Reinforcement Learning: Historical Foundations

R. E. Bellman

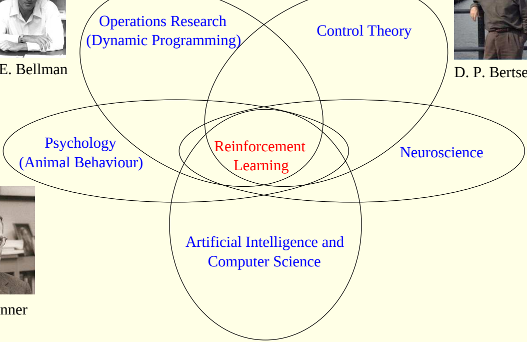
B. F. Skinner



Reinforcement Learning: Historical Foundations

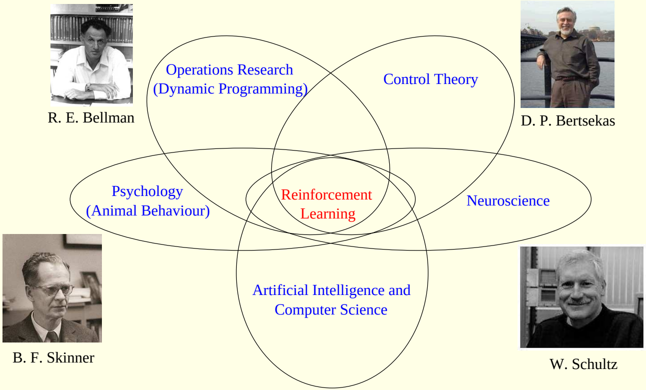
R. E. Bellman

D. P. Bertsekas

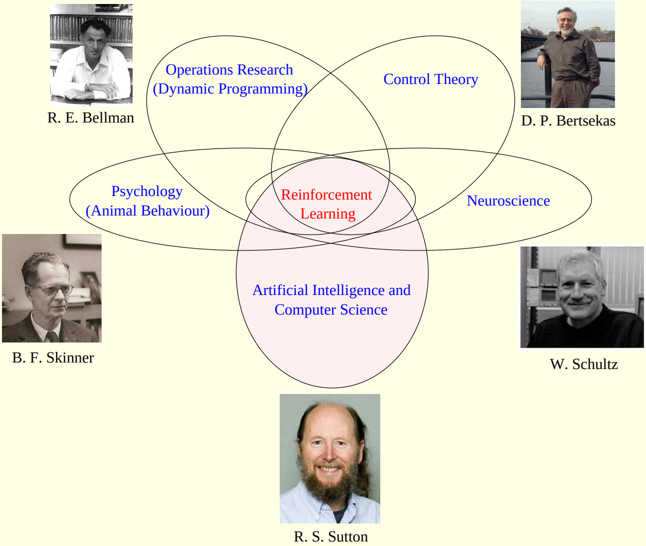


B. F. Skinner

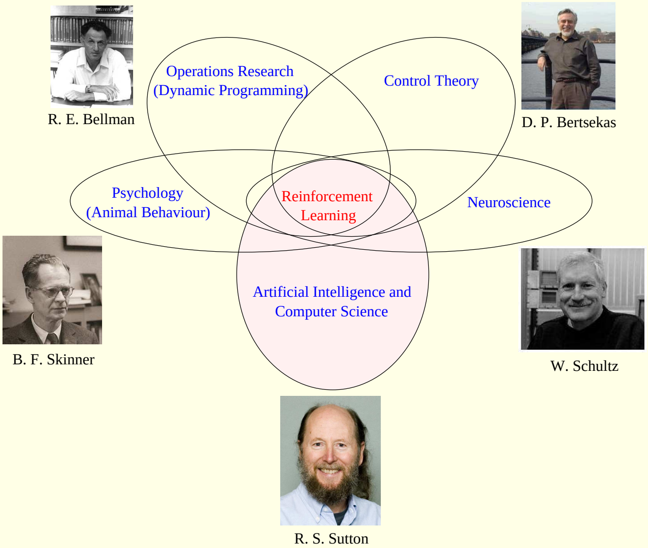
Reinforcement Learning: Historical Foundations



Reinforcement Learning: Historical Foundations

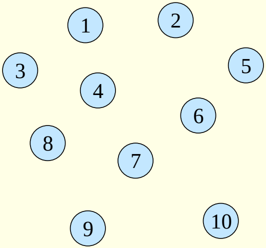


Reinforcement Learning: Historical Foundations

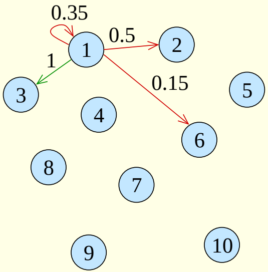


References: KLM1996, SB1998.

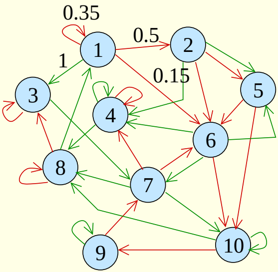
Markov Decision Problem (MDP) (P1994)



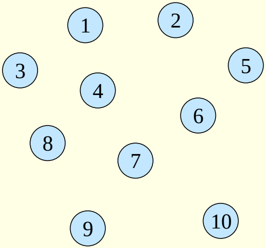
Markov Decision Problem (MDP) (P1994)



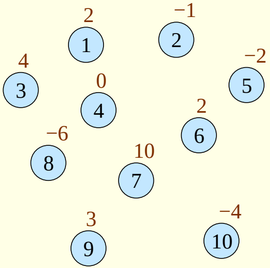
Markov Decision Problem (MDP) (P1994)



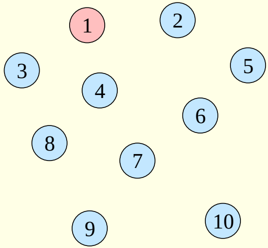
Markov Decision Problem (MDP) (P1994)



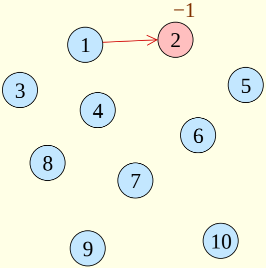
Markov Decision Problem (MDP) (P1994)



Markov Decision Problem (MDP) (P1994)

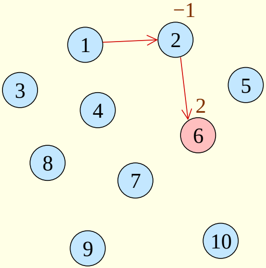


Markov Decision Problem (MDP) (P1994)



$r_1 = -1.$

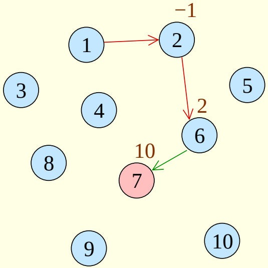
Markov Decision Problem (MDP) (P1994)



$r_1 = -1.$

$r_2 = 2.$

Markov Decision Problem (MDP) (P1994)

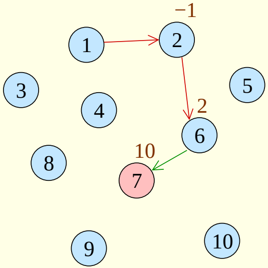


$r_1 = -1.$

$r_2 = 2.$

$r_3 = 10.$

Markov Decision Problem (MDP) (P1994)



$r_1 = -1.$

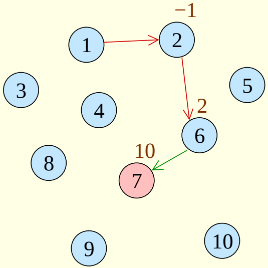
$r_2 = 2.$

$r_3 = 10.$

How to take actions so as to maximise expected long-term reward

$$\mathbb{E}[r_1 + r_2 + r_3 + \ldots]?$$

Markov Decision Problem (MDP) (P1994)



$r_1 = -1.$

$r_2 = 2.$

$r_3 = 10.$

How to take actions so as to maximise expected long-term reward

$$\mathbb{E}[r_1 + r_2 + r_3 + \ldots]?$$

[Note that there exists an (unknown) *optimal policy*.]

Q-Learning

- Keep a running estimate of the expected long-term reward obtained by taking each action from each state s, and acting *optimally* thereafter.

Q	red	green
1	-0.2	10
2	4.5	13
3	6	-8
4	0	0.2
5	-4.2	-4.2
6	1.2	1.6
7	10	6
8	4.8	9.9
9	5.0	-3.4
10	-1.9	2.3

Q-Learning

- Keep a running estimate of the expected long-term reward obtained by taking each action from each state s, and acting *optimally* thereafter.

Q	red	green
1	-0.2	10
2	4.5	13
3	6	-8
4	0	0.2
5	-4.2	-4.2
6	1.2	1.6
7	10	6
8	4.8	9.9
9	5.0	-3.4
10	-1.9	2.3

- Update these estimates based on experience (s_t, a_t, r_t, s_{t+1}):

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_t\{r_t + \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\}.$$

Q-Learning

- Keep a running estimate of the expected long-term reward obtained by taking each action from each state s, and acting *optimally* thereafter.

Q	red	green
1	-0.2	10
2	4.5	13
3	6	-8
4	0	0.2
5	-4.2	-4.2
6	1.2	1.6
7	10	6
8	4.8	9.9
9	5.0	-3.4
10	-1.9	2.3

- Update these estimates based on experience (s_t, a_t, r_t, s_{t+1}):

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_t \{r_t + \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\}.$$

Q-Learning

- Keep a running estimate of the expected long-term reward obtained by taking each action from each state s, and acting *optimally* thereafter.

Q	red	green
1	-0.2	10
2	4.5	13
3	6	-8
4	0	0.2
5	-4.2	-4.2
6	1.2	1.6
7	10	6
8	4.8	9.9
9	5.0	-3.4
10	-1.9	2.3

- Update these estimates based on experience (s_t, a_t, r_t, s_{t+1}):

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_t\{r_t + \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\}.$$

Q-Learning

- Keep a running estimate of the expected long-term reward obtained by taking each action from each state s, and acting *optimally* thereafter.

Q	red	green
1	-0.2	10
2	4.5	13
3	6	-8
4	0	0.2
5	-4.2	-4.2
6	1.2	1.6
7	10	6
8	4.8	9.9
9	5.0	-3.4
10	-1.9	2.3

- Update these estimates based on experience (s_t, a_t, r_t, s_{t+1}):

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_t\{r_t + \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\}.$$

- Act greedily based on the estimates (exploit) most of the time, but still
- Make sure to explore each action enough times.

Q-Learning

- Keep a running estimate of the expected long-term reward obtained by taking each action from each state s, and acting *optimally* thereafter.

Q	red	green
1	-0.2	10
2	4.5	13
3	6	-8
4	0	0.2
5	-4.2	-4.2
6	1.2	1.6
7	10	6
8	4.8	9.9
9	5.0	-3.4
10	-1.9	2.3

- Update these estimates based on experience (s_t, a_t, r_t, s_{t+1}):

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_t\{r_t + \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\}.$$

- Act greedily based on the estimates (exploit) most of the time, but still
- Make sure to explore each action enough times.

Q-learning will converge and induce an optimal policy (WD1992)!

Practice In Spite of the Theory!

Task	State Aliasing	State Space	Policy Representation (Number of features)
Backgammon (T1992)	Absent	Discrete	Neural network (198)
Job-shop scheduling (ZD1995)	Absent	Discrete	Neural network (20)
Tetris (BT1906)	Absent	Discrete	Linear (22)
Elevator dispatching (CB1996)	Present	Continuous	Neural network (46)
Acrobot control (S1996)	Absent	Continuous	Tile coding (4)
Dynamic channel allocation (SB1997)	Absent	Discrete	Linear (100's)
Active guidance of finless rocket (GM2003)	Present	Continuous	Neural network (14)
Fast quadrupedal locomotion (KS2004)	Present	Continuous	Parameterized policy (12)
Robot sensing strategy (KF2004)	Present	Continuous	Linear (36)
Helicopter control (NKJS2004)	Present	Continuous	Neural network (10)
Dynamic bipedal locomotion (TZS2004)	Present	Continuous	Feedback control policy (2)
Adaptive job routing/scheduling (WS2004)	Present	Discrete	Tabular (4)
Robot soccer keepaway (SSK2005)	Present	Continuous	Tile coding (13)
Robot obstacle negotiation (LSYSN2006)	Present	Continuous	Linear (10)
Optimized trade execution (NFK2007)	Present	Discrete	Tabular (2-5)
Blimp control (RPHB2007)	Present	Continuous	Gaussian Process (2)
9 $\times$ 9 Go (SSM2007)	Absent	Discrete	Linear ($\approx$1.5 million)
Ms. Pac-Man (SL2007)	Absent	Discrete	Rule list (10)
Autonomic resource allocation (TJDB2007)	Present	Continuous	Neural network (2)
General game playing (FB2008)	Absent	Discrete	Tabular (part of state space)
Soccer opponent "hassling" (GRT2009)	Present	Continuous	Neural network (9)
Adaptive epilepsy treatment (GVAP2008)	Present	Continuous	Extremely rand. trees (114)
Computer memory scheduling (IMMC2008)	Absent	Discrete	Tile coding (6)
Motor skills (PS2008)	Present	Continuous	Motor primitive coeff. (100's)
Combustion Control (HNGK2009)	Present	Continuous	Parameterized policy (2-3)

Practice In Spite of the Theory!

Task	State Aliasing	State Space	Policy Representation (Number of features)
Backgammon (T1992)	Absent	Discrete	Neural network (198)
Job-shop scheduling (ZD1995)	Absent	Discrete	Neural network (20)
Tetris (BT1906)	Absent	Discrete	Linear (22)
Elevator dispatching (CB1996)	Present	Continuous	Neural network (46)
Acrobot control (S1996)	Absent	Continuous	Tile coding (4)
Dynamic channel allocation (SB1997)	Absent	Discrete	Linear (100's)
Active guidance of finless rocket (GM2003)	Present	Continuous	Neural network (14)
Fast quadrupedal locomotion (KS2004)	Present	Continuous	Parameterized policy (12)
Robot sensing strategy (KF2004)	Present	Continuous	Linear (36)
Helicopter control (NKJS2004)	Present	Continuous	Neural network (10)
Dynamic bipedal locomotion (TZS2004)	Present	Continuous	Feedback control policy (2)
Adaptive job routing/scheduling (WS2004)	Present	Discrete	Tabular (4)
Robot soccer keepaway (SSK2005)	Present	Continuous	Tile coding (13)
Robot obstacle negotiation (LSYSN2006)	Present	Continuous	Linear (10)
Optimized trade execution (NFK2007)	Present	Discrete	Tabular (2-5)
Blimp control (RPHB2007)	Present	Continuous	Gaussian Process (2)
9 $\times$ 9 Go (SSM2007)	Absent	Discrete	Linear ($\approx$1.5 million)
Ms. Pac-Man (SL2007)	Absent	Discrete	Rule list (10)
Autonomic resource allocation (TJDB2007)	Present	Continuous	Neural network (2)
General game playing (FB2008)	Absent	Discrete	Tabular (part of state space)
Soccer opponent "hassling" (GRT2009)	Present	Continuous	Neural network (9)
Adaptive epilepsy treatment (GVAP2008)	Present	Continuous	Extremely rand. trees (114)
Computer memory scheduling (IMMC2008)	Absent	Discrete	Tile coding (6)
Motor skills (PS2008)	Present	Continuous	Motor primitive coeff. (100's)
Combustion Control (HNGK2009)	Present	Continuous	Parameterized policy (2-3)

Practice In Spite of the Theory!

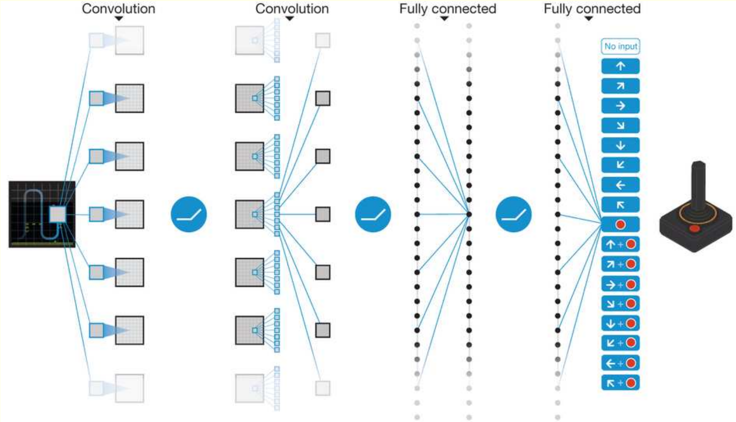
Task	State Aliasing	State Space	Policy Representation (Number of features)
Backgammon (T1992)	Absent	Discrete	Neural network (198)
Job-shop scheduling (ZD1995)	Absent	Discrete	Neural network (20)
Tetris (BT1906)	Absent	Discrete	Linear (22)
Elevator dispatching (CB1996)	Present	Continuous	Neural network (46)
Acrobot control (S1996)	Absent	Continuous	Tile coding (4)
Dynamic channel allocation (SB1997)	Absent	Discrete	Linear (100's)
Active guidance of finless rocket (GM2003)	Present	Continuous	Neural network (14)
Fast quadrupedal locomotion (KS2004)	Present	Continuous	Parameterized policy (12)
Robot sensing strategy (KF2004)	Present	Continuous	Linear (36)
Helicopter control (NKJS2004)	Present	Continuous	Neural network (10)
Dynamic bipedal locomotion (TZS2004)	Present	Continuous	Feedback control policy (2)
Adaptive job routing/scheduling (WS2004)	Present	Discrete	Tabular (4)
Robot soccer keepaway (SSK2005)	Present	Continuous	Tile coding (13)
Robot obstacle negotiation (LSYSN2006)	Present	Continuous	Linear (10)
Optimized trade execution (NFK2007)	Present	Discrete	Tabular (2-5)
Blimp control (RPHB2007)	Present	Continuous	Gaussian Process (2)
9 $\times$ 9 Go (SSM2007)	Absent	Discrete	Linear ($\approx$1.5 million)
Ms. Pac-Man (SL2007)	Absent	Discrete	Rule list (10)
Autonomic resource allocation (TJDB2007)	Present	Continuous	Neural network (2)
General game playing (FB2008)	Absent	Discrete	Tabular (part of state space)
Soccer opponent "hassling" (GRT2009)	Present	Continuous	Neural network (9)
Adaptive epilepsy treatment (GVAP2008)	Present	Continuous	Extremely rand. trees (114)
Computer memory scheduling (IMMC2008)	Absent	Discrete	Tile coding (6)
Motor skills (PS2008)	Present	Continuous	Motor primitive coeff. (100's)
Combustion Control (HNGK2009)	Present	Continuous	Parameterized policy (2-3)

Practice In Spite of the Theory!

Task	State Aliasing	State Space	Policy Representation (Number of features)
Backgammon (T1992)	Absent	Discrete	Neural network (198)
Job-shop scheduling (ZD1995)	Absent	Discrete	Neural network (20)
Tetris (BT1906)	Absent	Discrete	Linear (22)
Elevator dispatching (CB1996)	Present	Continuous	Neural network (46)
Acrobot control (S1996)	Absent	Continuous	Tile coding (4)
Dynamic channel allocation (SB1997)	Absent	Discrete	Linear (100's)
Active guidance of finless rocket (GM2003)	Present	Continuous	Neural network (14)
Fast quadrupedal locomotion (KS2004)	Present	Continuous	Parameterized policy (12)
Robot sensing strategy (KF2004)	Present	Continuous	Linear (36)
Helicopter control (NKJS2004)	Present	Continuous	Neural network (10)
Dynamic bipedal locomotion (TZS2004)	Present	Continuous	Feedback control policy (2)
Adaptive job routing/scheduling (WS2004)	Present	Discrete	Tabular (4)
Robot soccer keepaway (SSK2005)	Present	Continuous	Tile coding (13)
Robot obstacle negotiation (LSYSN2006)	Present	Continuous	Linear (10)
Optimized trade execution (NFK2007)	Present	Discrete	Tabular (2-5)
Blimp control (RPHB2007)	Present	Continuous	Gaussian Process (2)
9 $\times$ 9 Go (SSM2007)	Absent	Discrete	Linear ($\approx$1.5 million)
Ms. Pac-Man (SL2007)	Absent	Discrete	Rule list (10)
Autonomic resource allocation (TJDB2007)	Present	Continuous	Neural network (2)
General game playing (FB2008)	Absent	Discrete	Tabular (part of state space)
Soccer opponent "hassling" (GRT2009)	Present	Continuous	Neural network (9)
Adaptive epilepsy treatment (GVAP2008)	Present	Continuous	Extremely rand. trees (114)
Computer memory scheduling (IMMC2008)	Absent	Discrete	Tile coding (6)
Motor skills (PS2008)	Present	Continuous	Motor primitive coeff. (100's)
Combustion Control (HNGK2009)	Present	Continuous	Parameterized policy (2-3)

Perfect representations (fully observable, enumerable states) are impractical (K2011).

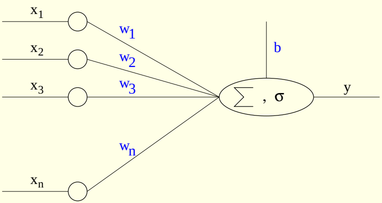
Typical Neural Network-based Representation of Q



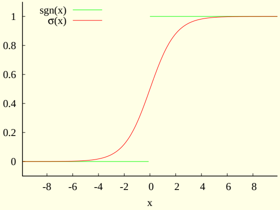
1. http://www.nature.com/nature/journal/v518/n7540/carousel/nature14236-f1.jpg

Overview

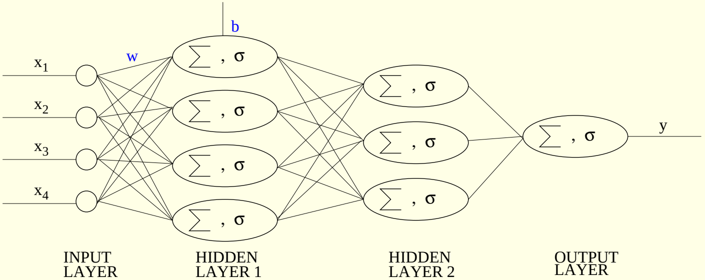
Artificial Neuron



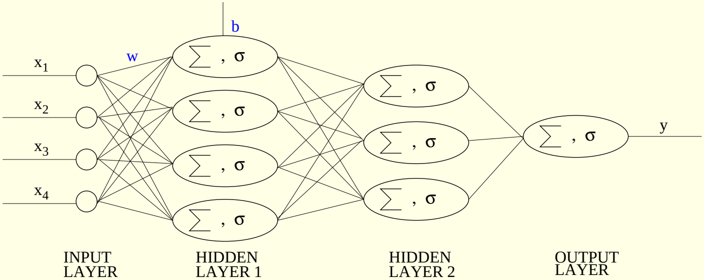
$$\sigma(x) \stackrel{\text{def}}{=} \frac{1}{1+\exp(-x)}.$$



Artificial Neural Networks



Artificial Neural Networks



Artificial neural networks are Universal Approximators.

For any function on a finite data set, there exists a single-hidden-layer neural network that fits it to an arbitrary degree of accuracy (HSW1989).

Backpropagation Algorithm (RHW1986)

We are given a training data set $\{(x^1, y^1), (x^2, y^2), \ldots, (x^D, y^D)\}$.

Let us start with some initial weights $\mathbf{w}$.

Backpropagation Algorithm (RHW1986)

We are given a training data set $\{(x^1, y^1), (x^2, y^2), \ldots, (x^D, y^D)\}$.

Let us start with some initial weights $\mathbf{w}$.

For each data point j,

the true label is y^j,

the prediction is $p^j(\mathbf{w})$; and

thus, the error is $(y^j - p^j(\mathbf{w}))^2$.

The aggregate error over the training data is $E(\mathbf{w}) = \sum_{j=1}^{D}(y^j - p^j(\mathbf{w}))^2$.

We move a step in the direction minimising error:

$\mathbf{w} \leftarrow \mathbf{w} - \alpha \nabla_{\mathbf{w}} E(\mathbf{w})$,

and iterate until convergence.

Backpropagation Algorithm (RHW1986)

We are given a training data set $\{(x^1, y^1), (x^2, y^2), \ldots, (x^D, y^D)\}$.
Let us start with some initial weights $\mathbf{w}$.

For each data point j,

the true label is y^j,

the prediction is $p^j(\mathbf{w})$; and

thus, the error is $(y^j - p^j(\mathbf{w}))^2$.

The aggregate error over the training data is $E(\mathbf{w}) = \sum_{j=1}^{D}(y^j - p^j(\mathbf{w}))^2$.

We move a step in the direction minimising error:

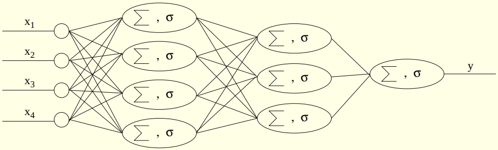
$\mathbf{w} \leftarrow \mathbf{w} - \alpha \nabla_{\mathbf{w}} E(\mathbf{w})$,

and iterate until convergence.

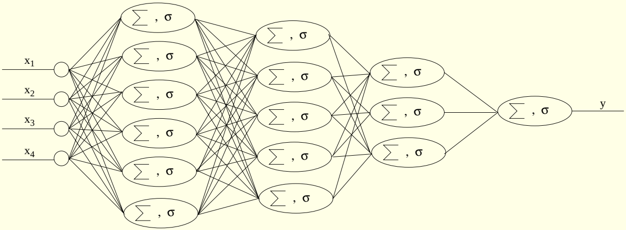
For a given neural network, $\nabla_{\mathbf{w}} E(\mathbf{w})$ can be easily computed.
Backpropagation will converge to a local minimum.

Expressiveness and Learnability

Small network



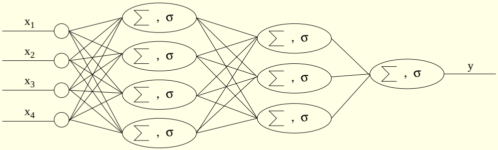
Large network



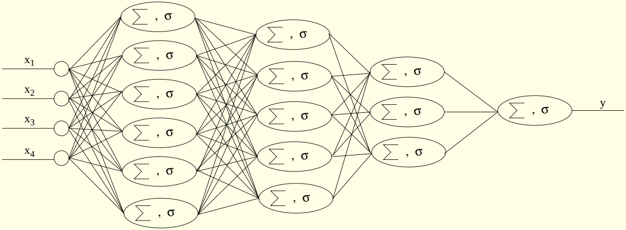
Is a larger network always better?

Expressiveness and Learnability

Small network



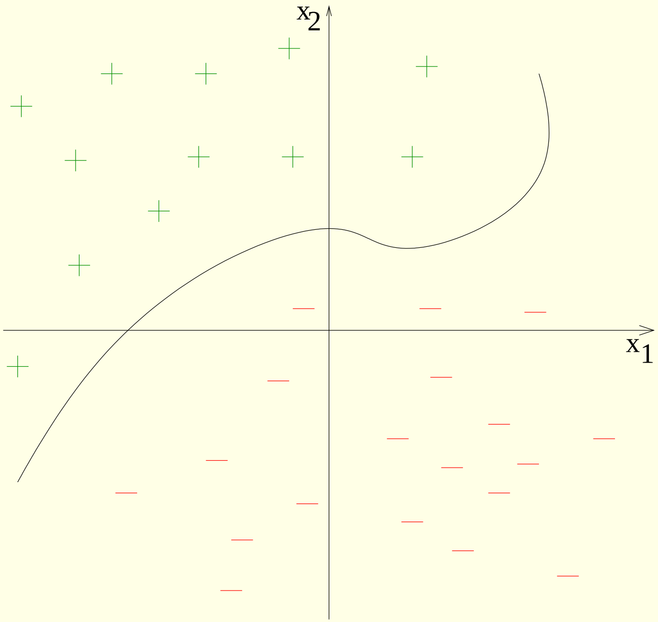
Large network



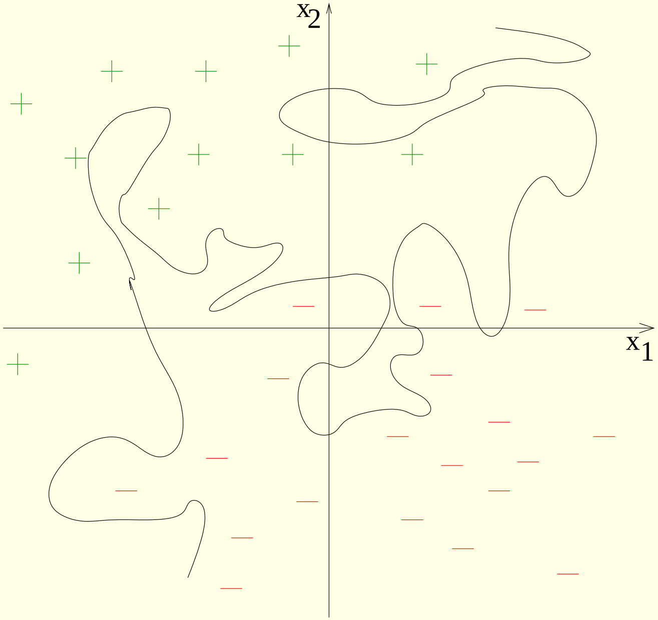
Is a larger network always better?

No!

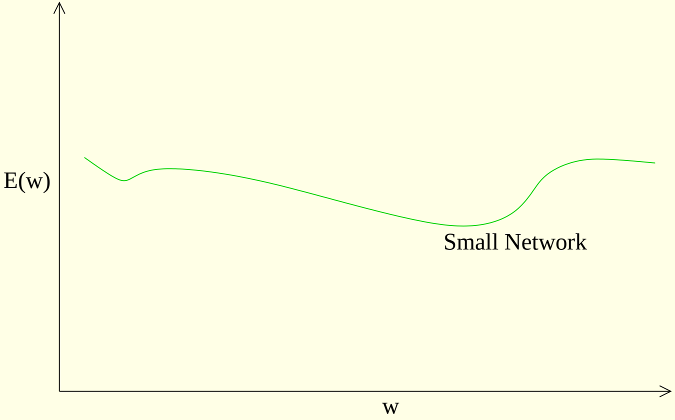
Expressiveness and Learnability: Overfitting



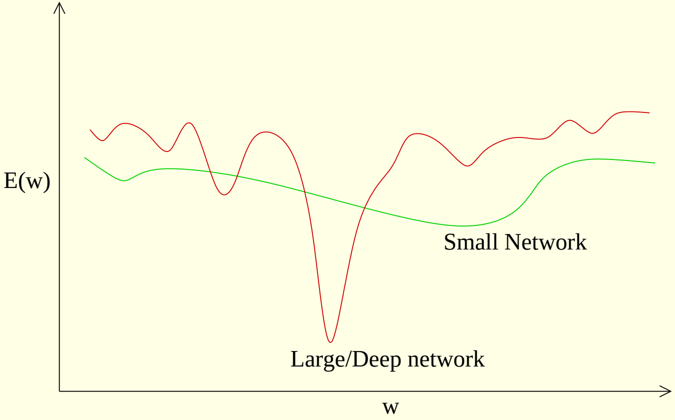
Expressiveness and Learnability: Overfitting



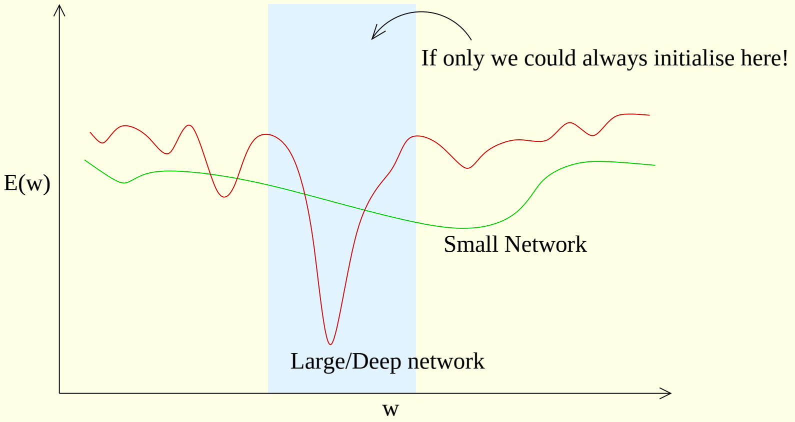
Expressiveness and Learnability: Initialisation



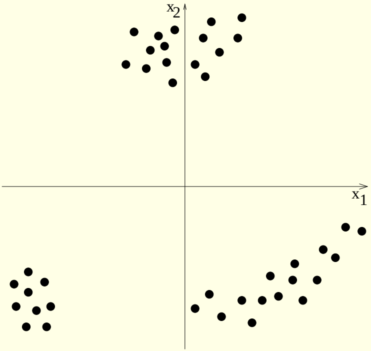
Expressiveness and Learnability: Initialisation



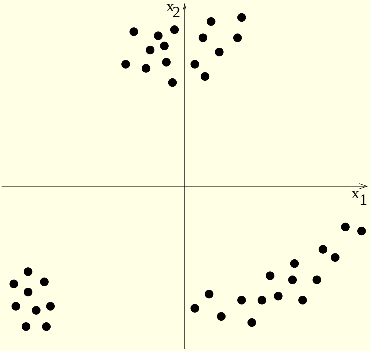
Expressiveness and Learnability: Initialisation



Deep Learning



Deep Learning

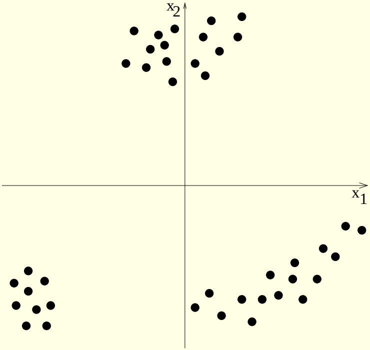


Preprocessing: Learn the distribution of the data (without seeing labels).

Deep Belief Networks: learned greedily, one layer at a time (HOT2006).

Subsequently apply backpropagation.

Deep Learning



Preprocessing: Learn the distribution of the data (without seeing labels).

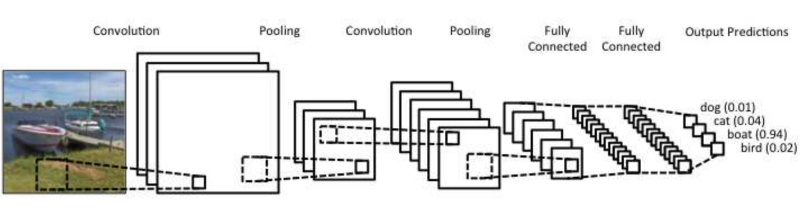
Deep Belief Networks: learned greedily, one layer at a time (HOT2006).

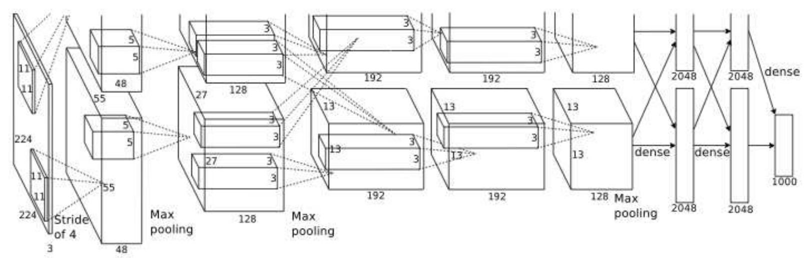
Subsequently apply backpropagation.

Aided by astronomical growth in data sizes and computational power.

Specifically successful in domains such as vision and speech.

Convolutional Neural Networks





253,440–186,624–64,896–64,896–43,264 –4096–4096–1000 network (KSH 2012).

1. https://www.clarifai.com/static/img_ours/cnn.png
2. http://mappingignorance.org/fx/media/2013/04/Deep-learning-4.png

Overview

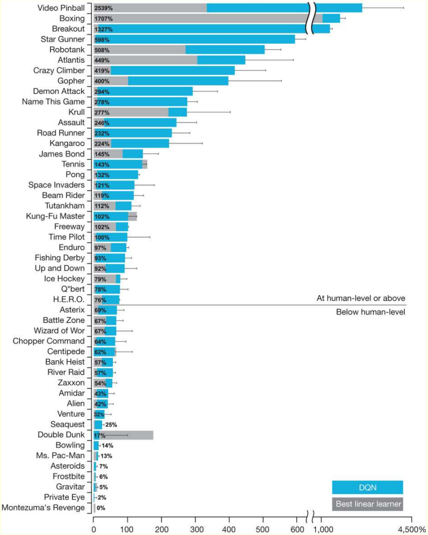
ATARI 2600 Games (MKSRVBGRFOPBSAKKWLH2015)

[Breakout video[1]]

1. http://www.nature.com/nature/journal/v518/n7540/extref/nature14236-sv2.mov

ATARI 2600 Games (MKSRVBGRFOPBSAKKWLH2015)

[Breakout video[1]]



1. `http://www.nature.com/nature/journal/v518/n7540/extref/nature14236-sv2.mov`

AlphaGo (SHMGSDSAPLDGNKSLLKGH2016)

March 2016: DeepMind's program beats Go champion Lee Sedol 4-1.

1. http://www.kurzweilai.net/images/AlphaGo-vs.-Sedol.jpg

AlphaGo (SHMGSDSAPLDGNKSLLKGH2016)

1. `http://static1.uk.businessinsider.com/image/56e0373052bcd05b008b5217-810-602/screen%20shot%202016-03-09%20at%2014.png`

Learning Algorithm

1. Represent action value function Q as a neural network.

2. Gather data (on the simulator) by taking ϵ-greedy actions w.r.t. Q:
   $(s_1, a_1, r_1, s_2, a_2, r_2, s_3, a_3, r_3, \ldots s_D, a_D, r_D, s_{D+1})$.

3. Train the network such that $Q(s_t, a_t) \approx r_t + \max_a Q(s_{t+1}, a)$.
   Go to 2.

Learning Algorithm

1. Represent action value function Q as a neural network.
   AlphaGo: Use both a policy network and an action value network.

2. Gather data (on the simulator) by taking ϵ-greedy actions w.r.t. Q:
   $(s_1, a_1, r_1, s_2, a_2, r_2, s_3, a_3, r_3, \dots s_D, a_D, r_D, s_{D+1})$.
   AlphaGo: Use Monte Carlo Tree Search for action selection

3. Train the network such that $Q(s_t, a_t) \approx r_t + \max_a Q(s_{t+1}, a)$.
   Go to 2.

AlphaGo: Trained using self-play.

References

(For references on slide 9, see Kalyanakrishnan's thesis (K2011).)

[R1957] Frank Rosenblatt, 1957. The Perceptron–a perceiving and recognizing automaton. *Report 85-460-1, Cornell Aeronautical Laboratory*, 1957.

[MP1972] Marvin Minsky and Seymour Papert, 1972. Perceptrons: An Introduction to Computational Geometry. *2nd edition with corrections, first edition 1969*, MIT Press, 1972.

[RHW1986] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams, 1986. Learning representations by back-propagating errors. *Nature*, 323(6088): 533–536, 1986.

[HSW1989] Kurt Hornik, Maxwell Stinchcombe, and Halber White, 1989. Multilayer Feedforward Networks are Universal Approximators. *Neural Networks*, 2: 359–366, 1989.

[WD1992] Christopher J. C. H. Watkins and Peter Dayan, 1992. Q-Learning. *Machine Learning*, 8(3–4):279–292, 1992.

[P1994] Martin L. Puterman. Markov Decision Processes. Wiley, 1994.

[KLM1996] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore, 1996. Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

[SB1998] Richard S. Sutton and Andrew G. Barto, 1998. Reinforcement Learning: An Introduction. MIT Press, 1998.

[HOT2006] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh, 2006. A Fast Learning Algorithm for Deep Belief Nets, *Neural Computation*, 18:1527–1554, 2006.

References

[KLS2007] Shivaram Kalyanakrishnan, Yaxin Liu, and Peter Stone. Half Field Offense in RoboCup Soccer: A Multiagent Reinforcement Learning Case Study. In *RoboCup 2006: Robot Soccer World Cup X*, pp. 72–85, Springer, 2007.

[K2011] Shivaram Kalyanakrishnan. Learning Methods for Sequential Decision Making with Imperfect Representations. *Ph.D. Thesis, Department of Computer Science, The University of Texas at Austin*, 2011.

[KSH2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*, pp. 1106–1114, NIPS, 2012.

[MKSRVBGRFOPBSAKKWLH2015] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518: 529–533, 2015.

[SHMGSDSAPLDGNKSLLKGH2016] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis, 2016. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529: 484–489, 2016.

Summary and Conclusion

Reinforcement Learning

Do not program behaviour! Rather, specify goals.

Rich history, at confluence of several fields of study, firm foundation.

Limited in practice by quality of the representation used.

Summary and Conclusion

Reinforcement Learning

Do not program behaviour! Rather, specify goals.

Rich history, at confluence of several fields of study, firm foundation.

Limited in practice by quality of the representation used.

Deep Learning

Ability to learn complex non-linear relationships in data.

Demonstrated successes in vision and natural language processing.

Summary and Conclusion

Reinforcement Learning

Do not program behaviour! Rather, specify goals.

Rich history, at confluence of several fields of study, firm foundation.

Limited in practice by quality of the representation used.

Deep Learning

Ability to learn complex non-linear relationships in data.

Demonstrated successes in vision and natural language processing.

Deep Reinforcement Learning

Proof of concept established.

Very promising technology that is changing the face of AI.

Summary and Conclusion

Reinforcement Learning

Do not program behaviour! Rather, specify goals.

Rich history, at confluence of several fields of study, firm foundation.

Limited in practice by quality of the representation used.

Deep Learning

Ability to learn complex non-linear relationships in data.

Demonstrated successes in vision and natural language processing.

Deep Reinforcement Learning

Proof of concept established.

Very promising technology that is changing the face of AI.

Thank you!