Deep Learning for Artificial General Intelligence

Survey of Recent Developments

Cosmo Harrigan
cosmoh@uw.edu

University of Washington

July 17, 2016

The New School, New York City

The Ninth Conference on
Artificial General Intelligence
at The Joint Multi-Conference on
Human-Level Artificial Intelligence

Overview

Why is Deep Learning relevant for AGI?

Abstract

The relevance of deep learning to the field of Artificial General Intelligence research is described, in terms of the expanding scope of deep learning model designs and the increasing combination of deep learning with other methods to form hybrid architectures.

Deep learning is a rapidly expanding research area, and various groups have recently proposed novel extensions to earlier deep learning models, including: generative models; the ability to interface with external memory and other external resources; Neural Turing Machines which learn programs; deep reinforcement learning; neuroevolution; intrinsic motivation and unsupervised learning; and more complex network models.

These slides constitute a brief survey of selected work from recent papers in the field. Citations for the content are provided in the footnotes.

The presentation is organized with a view towards the integration of additional abilities into deep learning architectures, including: planning; reasoning and logic; data efficient learning and one-shot learning; program induction; additional learning algorithms other than backpropagation; more sophisticated techniques for unsupervised learning and reinforcement learning; and structured prediction.

We can view deep learning research as making significant contributions relevant to AGI, but also note that future progress in the field will likely depend on integrating threads of research from cognitive science, machine learning, universal artificial intelligence and symbolic artificial intelligence, resulting in systems that significantly extend the boundaries of what might be considered “deep learning” today.

Why is Deep Learning relevant for AGI?

Summary

Deep learning is relevant to AGI research for two reasons: because its methods are expanding in scope, and because it is effectively being combined with other methods to form hybrid architectures.

Expanding in Scope

It is expanding to encompass a wide range of methods, including: memory, unsupervised learning, learning to act, program learning and attention.

Hybrid Systems

It is increasingly being used in conjunction with other methods to form hybrid architectures.

Universal Intelligence

We can frame deep learning for AGI as a specific research direction with the goal of designing appropriate methods for **approximating universal intelligence**.

AIXI Approximation: Planning and Learning

There are two parts to AIXI. The first is the expectimax search into the future which we will call **planning**. The second is the use of a Bayesian mixture over Turing machines to predict future observations and rewards based on past experience; we will call that **learning**. **Both parts need to be approximated for computational tractability.**[a]

[a] Joel Veness et al. "A Monte-Carlo AIXI Approximation". In: *Journal of Artificial Intelligence Research* 40.1 (2011), pp. 95–142.

Defining Deep Neural Networks I

What is a useful definition for deep neural networks?

Schmidhuber[1] presents a non-traditional, useful definition of deep neural networks, in terms of the following concepts:

1. Definition/Program implemented by the network
2. Partially Causal Sequences
3. Topology
4. Weight Sharing
5. Credit Assignment Paths
6. Potential Causal Connections
7. Depth

Defining Deep Neural Networks II

Definition/Program

The NN's behavior or program is determined by a set of real-valued, possibly modifiable, parameters or weights w_i $(i = 1, \ldots, n)$.[a]

[a] Jürgen Schmidhuber. "Deep learning in neural networks: An overview". In: *Neural Networks* 61 (2015), pp. 85–117.

Partially Causal Sequences

During an episode, there is a *partially causal sequence* $x_t(t = 1, \ldots, T)$ of real values called events. Each x_t is either an input set by the environment, or the activation of a unit that may directly depend on other $x_k(k < t)$ through a current NN topology-dependent set in_t of indices k representing incoming causal connections or links.[a]

[a] Jürgen Schmidhuber. "Deep learning in neural networks: An overview". In: *Neural Networks* 61 (2015), pp. 85–117.

Defining Deep Neural Networks III

Topology

Let the function v encode topology information and map such event index pairs (k, t) to weight indices.

x_t may directly affect certain $x_k (k > t)$ through outgoing connections or links represented through a current set out_t of indices k with $t \in in_k$. Some of the non-input events are called *output events*.[a]

[a] Jürgen Schmidhuber. "Deep learning in neural networks: An overview". In: *Neural Networks* 61 (2015), pp. 85–117.

Defining Deep Neural Networks IV

Weight Sharing

Note that many of the x_t may refer to different, time-varying activations of the *same* unit in sequence-processing RNNs (*“unfolding in time”*).

During an episode, the same weight may get reused over and over again in topology-dependent ways, e.g., in RNNs, or in convolutional NNs.

This is called weight sharing *across space and/or time*. Weight sharing may greatly reduce the NN's descriptive complexity, which is the number of bits of information required to describe the NN.[a]

[a] Jürgen Schmidhuber. “Deep learning in neural networks: An overview”. In: *Neural Networks* 61 (2015), pp. 85–117.

Defining Deep Neural Networks V

Credit Assignment Paths

To measure whether credit assignment in a given NN application is of the *deep* or *shallow* type, we introduce the concept of *Credit Assignment Paths* or CAPs, which are chains of possibly causal links between the events; e.g., from input through hidden to output layers in FNNs, or through transformations over time in RNNs.[a]

[a] Jürgen Schmidhuber. "Deep learning in neural networks: An overview". In: *Neural Networks* 61 (2015), pp. 85–117.

Defining Deep Neural Networks VI

Potential Causal Connections

More general, possibly indirect, *Potential Causal Connections* (PCC) are expressed by the recursively defined Boolean predicate $pcc(p, q)$, which in the SL case is true only if $pdcc(p, q)$, or if $pcc(p, k)$ for some k and $pdcc(k, q)$.

In the latter case, appending q to any CAP from p to k yields a CAP from p to q (this is a recursive definition, too).

The set of such CAPs may be large but is finite. Note that the *same* weight may affect *many* different PDCCs between successive events listed by a given CAP, e.g., in the case of RNNs, or weight-sharing FNNs.[a]

[a] Jürgen Schmidhuber. "Deep learning in neural networks: An overview". In: *Neural Networks* 61 (2015), pp. 85–117.

Defining Deep Neural Networks VII

Depth

Suppose a CAP has the form $(\ldots, k, t, \ldots, q)$, where k and t (possibly $t = q$) are the first successive elements with *modifiable* $w_{v(k,t)}$. Then the length of the suffix list $(t, \ldots, q)$ is called the CAP's *depth*.

Thus, we arrive at the concept of *Deep Learning*.

[1] Jürgen Schmidhuber. "Deep learning in neural networks: An overview". In: *Neural Networks* 61 (2015), pp. 85–117.

Intrinsic Motivation

Definition

- Intrinsically motivated agents explore new behaviors simply to satisfy an internal drive for discovery, defined in one of multiple possible ways, rather than to directly solve problems
- Intrinsic behaviors could eventually help the agent to solve tasks presented by the environment
- Useful in settings with sparse, delayed rewards
- Examples: hunger, boredom, curiosity

Hierarchical DQN (h-DQN) is a framework to integrate hierarchical value functions, operating at different temporal scales, with intrinsically motivated deep reinforcement learning.

[2] Tejas D. Kulkarni et al. "Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation". In: (Apr. 2016), p. 13. arXiv: 1604.06057. URL: http://arxiv.org/abs/1604.06057%7B%5C#%7D.

Intrinsic Motivation

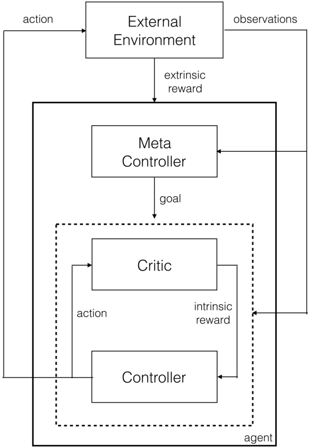
Hierarchical DQN[3]

- Scheme for temporal abstraction that involves simultaneously learning options (intrinsic goals) and a control policy to compose options in a deep reinforcement learning setting
- Allows for flexible goal specifications, such as functions over entities and relations
- Provides an efficient space for exploration in complicated environments

[3] Tejas D. Kulkarni et al. "Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation". In: (Apr. 2016), p. 13. arXiv: 1604.06057. URL: http://arxiv.org/abs/1604.06057%7B%5C#%7D.

Intrinsic Motivation

Hierarchical DQN[4]



[4] Tejas D. Kulkarni et al. "Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation". In: (Apr. 2016), p. 13. arXiv: 1604.06057. URL: http://arxiv.org/abs/1604.06057%7B%5C#%7D.

Intrinsic Motivation

Hierarchical DQN[5]

Algorithm 1 Learning algorithm for h-DQN

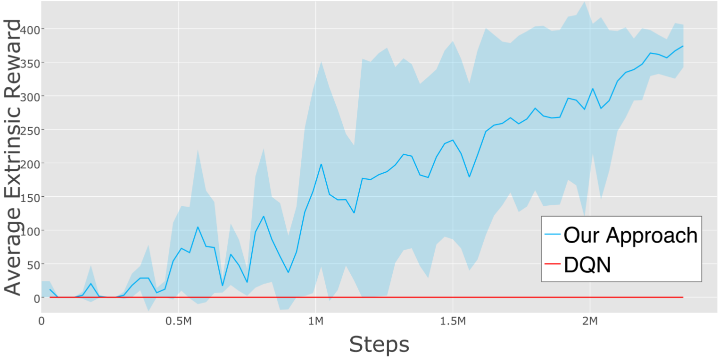
Initialize experience replay memories $\{\mathcal{D}_1, \mathcal{D}_2\}$ and parameters $\{\theta_1, \theta_2\}$ for the controller and meta-controller respectively.
Initialize exploration probability $\epsilon_{1,g} = 1$ for the controller for all goals g and $\epsilon_2 = 1$ for the meta-controller.
for $i = 1, num\_episodes$ **do**
    Initialize game and get start state description s
    $g \leftarrow \text{EPSGREEDY}(s, \mathcal{G}, \epsilon_2, Q_2)$
    while s is **not** terminal **do**
        $F \leftarrow 0$
        $s_0 \leftarrow s$
        while not (s is terminal **or** goal g reached) **do**
            $a \leftarrow \text{EPSGREEDY}(\{s, g\}, \mathcal{A}, \epsilon_{1,g}, Q_1)$
            Execute a and obtain next state s' and extrinsic reward f from environment
            Obtain intrinsic reward $r(s, a, s')$ from internal critic
            Store transition $(\{s, g\}, a, r, \{s', g\})$ in $\mathcal{D}_1$
            $\text{UPDATEPARAMS}(\mathcal{L}_1(\theta_{1,i}), \mathcal{D}_1)$
            $\text{UPDATEPARAMS}(\mathcal{L}_2(\theta_{2,i}), \mathcal{D}_2)$
            $F \leftarrow F + f$
            $s \leftarrow s'$
        end while
        Store transition (s_0, g, F, s') in $\mathcal{D}_2$
        if s is **not** terminal **then**
            $g \leftarrow \text{EPSGREEDY}(s, \mathcal{G}, \epsilon_2, Q_2)$
        end if
    end while
    Anneal ϵ_2 and adaptively anneal $\epsilon_{1,g}$ using average success rate of reaching goal g.
end for

[5] Tejas D. Kulkarni et al. "Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation". In: (Apr. 2016), p. 13. arXiv: 1604.06057. URL: http://arxiv.org/abs/1604.06057%7B%5C#%7D.

Intrinsic Motivation

Hierarchical DQN[6]

Applied to Montezuma's Revenge (delayed reward setting). Comparison of h-DQN with DQN demonstrates increased effectiveness for delayed rewards:



[6] Tejas D. Kulkarni et al. "Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation". In: (Apr. 2016), p. 13. arXiv: 1604.06057. URL: http://arxiv.org/abs/1604.06057%7B%5C#%7D.

Some possible reward functions for intrinsic motivation:[7]

1. Missing information or Bayesian surprise, measuring the change in an agents internal belief after the observation of new data
2. Measures based on prediction errors of future states
3. Salient event prediction
4. Measures based on information-theoretic quantities
5. Empowerment

[7] Shakir Mohamed and Danilo Jimenez Rezende. *Variational Information Maximisation for Intrinsically Motivated Reinforcement Learning*. 2015.

Intrinsic Motivation

Empowerment[8]

- Many ways in which to formally define internal drives
- What all such definitions have in common is that they, in some unsupervised fashion, allow an agent to reason about the value of information in the action-observation sequences it experiences
- The mutual information allows for exactly this type of reasoning and forms the basis of one popular intrinsic reward measure, known as empowerment

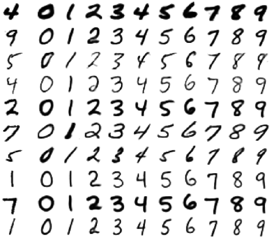
[8] Shakir Mohamed and Danilo Jimenez Rezende. *Variational Information Maximisation for Intrinsically Motivated Reinforcement Learning*. 2015.

- Variational Autoencoders (latent-variable probabilistic models) are used for unsupervised learning of abstract features
- Employing rich parametric density estimators formed by the fusion of probabilistic modeling and deep neural networks[9]

[9] Diederik P Kingma and Max Welling. "Auto-Encoding Variational Bayes". In: (Dec. 2013). arXiv: 1312.6114. URL: http://arxiv.org/abs/1312.6114.

Generative Models

Analogical Reasoning[10]



- Leftmost columns: images from the test set.
- Subsequent columns: analogical fantasies by the generative model, where the latent variable of each row is set to the value inferred from the test-set image on the left by the inference network.
- Each column corresponds to a class label.

[10] Diederik P Kingma et al. "Semi-Supervised Learning with Deep Generative Models". In: *arXiv.org* cs.LG (June 2014), pp. 1–9. arXiv: arXiv:1406.5298v1. URL: http://arxiv.org/abs/1406.5298v2.

Generative Models

Generative Adversarial Networks[11]

- Simultaneously train two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G
- Training procedure for G is to maximize the probability of D making a mistake
- Corresponds to a minimax two-player game

[11] Ian Goodfellow et al. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems*. 2014, pp. 2672–2680. URL: http://papers.nips.cc/paper/5423-generative-adversarial-nets.

Generative Models

Example: Varying Features in a Learned Model of Chairs[12]

- Train a neural network to generate accurate images of chairs from a high-level description: class, orientation with respect to the camera, and additional parameters such as color, brightness, etc.
- Interpolation between examples
- Generating new examples by varying specific features

[12] Alexey Dosovitskiy, Jost Tobias Springenberg, and Thomas Brox. *Learning To Generate Chairs With Convolutional Neural Networks*. 2015. arXiv: 1411.5928.

Generative Models

Activating Various Transformations[13]

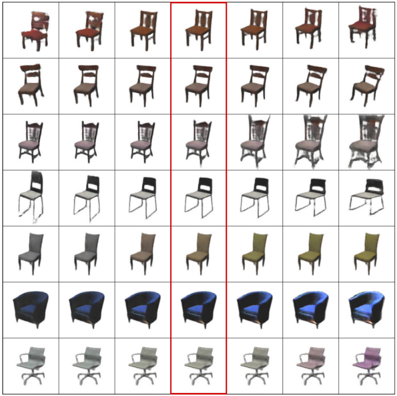
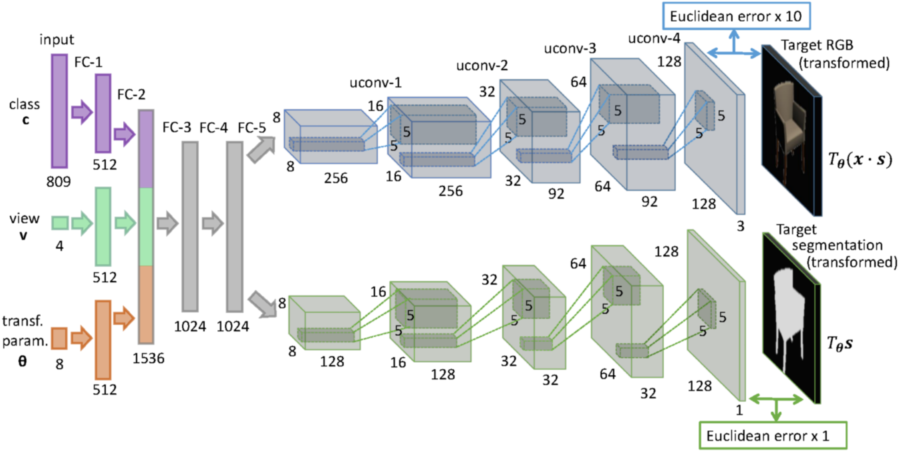


Figure 5. Generation of chair images while activating various transformations. Each row shows one transformation: translation, rotation, zoom, stretch, saturation, brightness, color. The middle column shows the reconstruction without any transformation.

[13] Alexey Dosovitskiy, Jost Tobias Springenberg, and Thomas Brox. *Learning To Generate Chairs With Convolutional Neural Networks*. 2015. arXiv: 1411.5928.

Architecture[14]

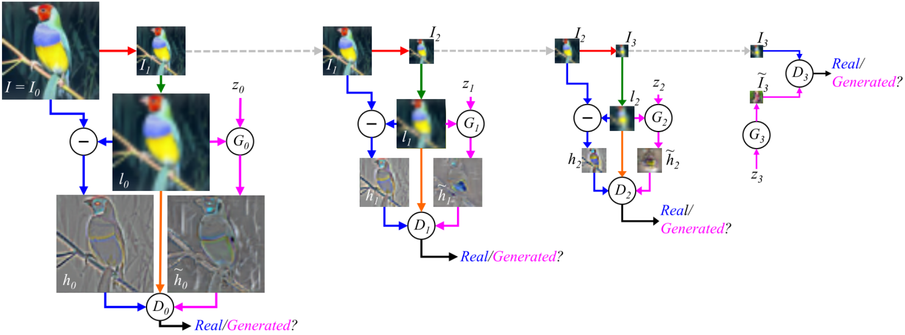


[14] Alexey Dosovitskiy, Jost Tobias Springenberg, and Thomas Brox. *Learning To Generate Chairs With Convolutional Neural Networks*. 2015. arXiv: 1411.5928.

Generative Models

Combination of GANs + Laplacian Pyramids[15]

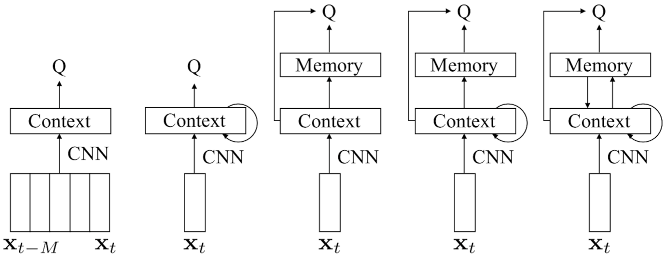
Laplacian Pyramids combined with Generative Adversarial Networks for generating images of a class.



[15] Emily Denton et al. "Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks". In: *Advances in Neural Information Processing Systems* (2015), pp. 1486–1494.

Feedback Recurrent Memory Q-Network (FRMQN)[16]

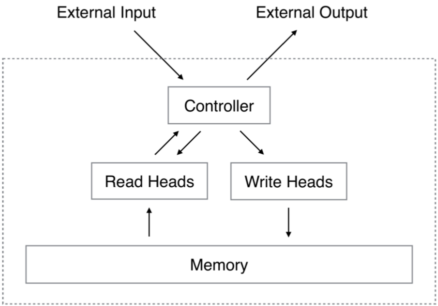
Context-dependent memory retrieval for deep reinforcement learning. Utilizes attention to decide which memories to focus on for computing the value function.



(a) DQN (b) DRQN (c) MQN (d) RMQN (e) FRMQN

[16] Junhyuk Oh et al. "Control of Memory, Active Perception, and Action in Minecraft". In: (May 2016). arXiv: 1605.09128. URL: http://arxiv.org/abs/1605.09128.

Neural Turing Machines extend the capabilities of neural networks by coupling them to external memory resources, which they can interact with by attentional processes. The combined system is analogous to a Turing Machine or Von Neumann architecture but is differentiable end-to-end, allowing it to be efficiently trained with gradient descent.[17]



[17] Alex Graves, Greg Wayne, and Ivo Danihelka. "Neural Turing Machines". In: *arXiv preprint arXiv:1410.5401* (2014).

Program Learning

Neural Turing Machines[18]

- NTMs infer simple algorithms such as copying, sorting, and associative recall from input and output examples
- Enrich the capabilities of standard recurrent networks to simplify the solution of algorithmic tasks by adding a large, addressable memory
- Capacity for short-term storage of information and its rule-based manipulation
- Rules are simple programs, and the stored information constitutes the arguments of these programs
- An NTM resembles a working memory system, as it is designed to solve tasks that require the application of approximate rules to "rapidly-created variables."

[18] Alex Graves, Greg Wayne, and Ivo Danihelka. "Neural Turing Machines". In: *arXiv preprint arXiv:1410.5401* (2014).

- Unlike the NTM, the Neural GPU is highly parallel which makes it easier to train and efficient to run
- An essential property of algorithms is their ability to handle inputs of arbitrary size
- The Neural GPU can be trained on short instances of an algorithmic task and successfully generalize to long instances.
- Verified on a number of tasks including long addition and long multiplication of numbers represented in binary

[19] Łukasz Kaiser and Ilya Sutskever. "Neural GPUs Learn Algorithms". In: (Nov. 2015). arXiv: 1511.08228. URL: http://arxiv.org/abs/1511.08228.

Neural Programmer-Interpreter (NPI): a recurrent and compositional neural network that learns to represent and execute programs. NPI has three learnable components:

- Task-agnostic recurrent core
- Persistent key-value program memory
- Domain-specific encoders that enable a single NPI to operate in multiple perceptually diverse environments with distinct affordances.

By learning to compose lower-level programs to express higher-level programs, NPI reduces sample complexity and increases generalization ability compared to sequence-to-sequence LSTMs.
Program memory allows efficient learning of additional tasks by building on existing programs.

[20] Scott Reed and Nando de Freitas. "Neural Programmer-Interpreters". In: (Nov. 2015). arXiv: 1511.06279. URL: http://arxiv.org/abs/1511.06279.

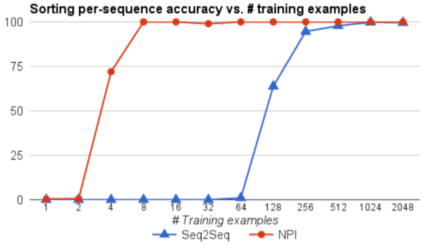


Figure 5: **Sample complexity**. Test accuracy of sequence-to-sequence LSTM versus NPI on length-20 arrays of single-digit numbers. Note that NPI is able to mine and train on subprogram traces from each bubblesort example.

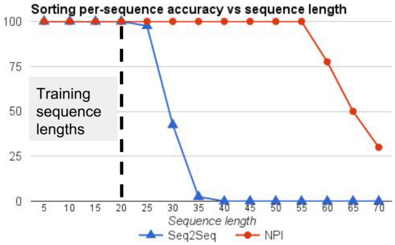
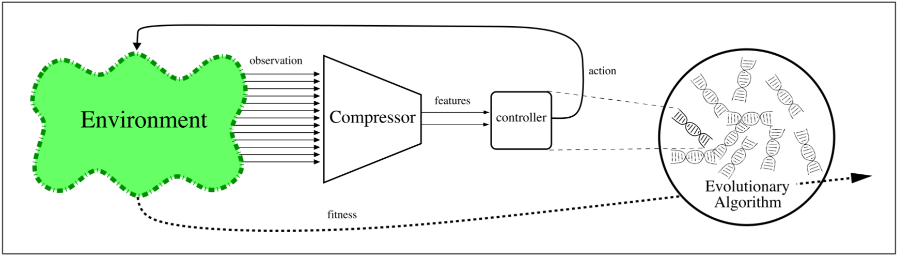


Figure 6: **Strong vs. weak generalization**. Test accuracy of sequence-to-sequence LSTM versus NPI on varying-length arrays of single-digit numbers. Both models were trained on arrays of single-digit numbers up to length 20.

[21] Scott Reed and Nando de Freitas. "Neural Programmer-Interpreters". In: (Nov. 2015). arXiv: 1511.06279. URL: http://arxiv.org/abs/1511.06279.

Neuroevolution

In addition to being used for feature learning, neural networks can also be applied to reinforcement learning as a policy search method, by representing the controller as a neural network and optimizing the parameters of the controller using a genetic algorithm.

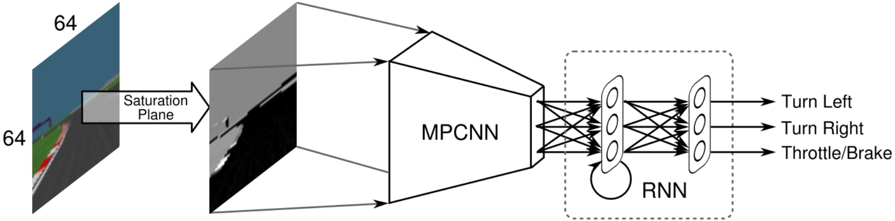


[22]

[22] Jan Koutník, Jürgen Schmidhuber, and Faustino Gomez. "Online evolution of deep convolutional network for vision-based reinforcement learning". In: *Lecture Notes in Computer Science* 8575 LNAI (2014), pp. 260–269.

Deep Reinforcement Learning

Learning to Act from Pixels: TORCS[23][24][25]



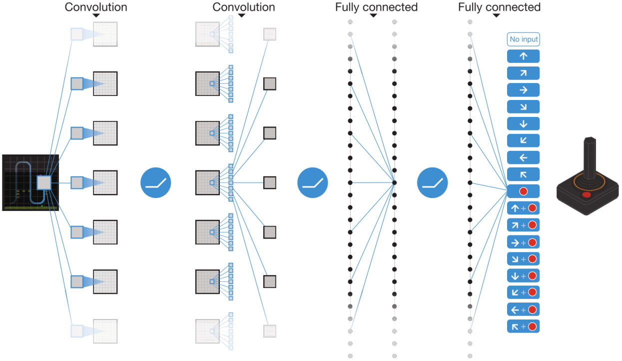
[23] Jan Koutník et al. "Evolving large-scale neural networks for vision-based reinforcement learning". In: *Proceeding of the 2013 Conference on Genetic and Evolutionary Computation*. New York, New York, USA: ACM Press, July 2013, p. 1061.

[24] Jan Koutník, Jürgen Schmidhuber, and Faustino Gomez. "Evolving deep unsupervised convolutional networks for vision-based reinforcement learning". In: *Proceedings of the 2014 Conference on Genetic and Evolutionary Computation*. ACM. 2014, pp. 541–548.

[25] Jan Koutník, Jürgen Schmidhuber, and Faustino Gomez. "Online evolution of deep convolutional network for vision-based reinforcement learning". In: *Lecture Notes in Computer Science* 8575 LNAI (2014), pp. 260–269.

Deep Reinforcement Learning

Learning to Act from Pixels: Atari[26][27]



[26] Volodymyr Mnih et al. "Playing Atari with Deep Reinforcement Learning". In: (Dec. 2013). arXiv: 1312.5602. URL: http://arxiv.org/abs/1312.5602.

[27] Volodymyr Mnih et al. "Human-level control through deep reinforcement learning". In: *Nature* 518.7540 (2015), pp. 529–533. ISSN: 0028-0836. DOI: 10.1038/nature14236. arXiv: 1312.5602. URL: http://dx.doi.org/10.1038/nature14236.

One-Shot Learning[28]

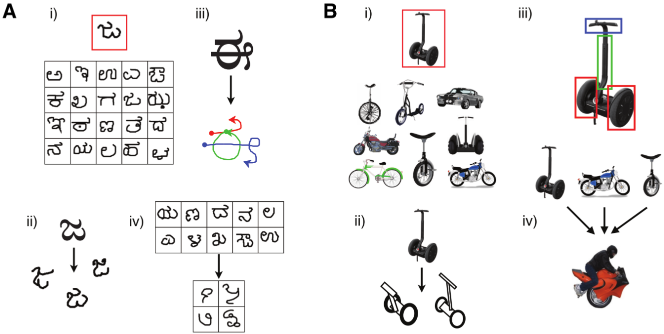


Figure 1: The characters challenge: human-level learning of a novel handwritten characters (A), with the same abilities also illustrated for a novel two-wheeled vehicle (B). A single example of a new visual concept (red box) can be enough information to support the (i) classification of new examples, (ii) generation of new examples, (iii) parsing an object into parts and relations, and (iv) generation of new concepts from related concepts. Figure reprinted from Lake, Salakhutdinov, and Tenenbaum (2015).

[28] Brenden M. Lake et al. "Building Machines That Learn and Think Like People". In: (Apr. 2016). arXiv: 1604.00289. URL: http://arxiv.org/abs/1604.00289.

Deep reinforcement learning is very data inefficient.

How to discover of relevant aspects of the environment efficiently?

This episodic control method presents a hybrid system, combining deep learning (variational autoencoders) for feature selection with an episodic memory.[29]

- Episodic control[30] is an approach that can rapidly re-enact observed, successful policies
- Episodic control records highly rewarding experiences and follows a policy that replays sequences of actions that previously yielded high returns
- Tackles a critical deficiency in current reinforcement learning systems: their inability to learn in a one-shot fashion.
- A fast-learning system based on non-parametric memorization of experience

[29] Charles Blundell et al. "Model-Free Episodic Control". In: (June 2016). arXiv: 1606.04460. URL: http://arxiv.org/abs/1606.04460.

[30] Máté Lengyel and Peter Dayan. "Hippocampal Contributions to Control: The Third Way". In: *Advances in Neural Information Processing Systems 2007.* (2007), pp. 889–896.

Data Efficiency

Episodic Control[31]

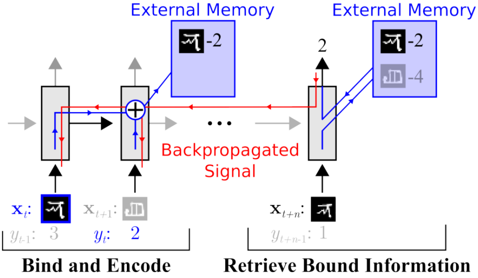
- Humans and animals utilize multiple learning, memory, and decision systems each best suited to different settings
- When an accurate model of the environment is available, and there are sufficient time and working memory resources, the best strategy is model-based planning associated with prefrontal cortex
- When there is no time or no resources available for planning, the less compute-intensive immediate decision systems must be employed
- Quick-to-learn instance-based control policies serve as a rough approximation while a slower more generalizable decision system is trained up
- Deep learning is used for embedding observations in state space

[31] Charles Blundell et al. "Model-Free Episodic Control". In: (June 2016). arXiv: 1606.04460. URL: http://arxiv.org/abs/1606.04460.

Data Efficiency

Memory Augmented Neural Networks[32]

- Rapidly assimilate new data, leverage this data to make accurate predictions after only a few samples
- Method for accessing an external memory that focuses on memory content, unlike previous methods that use memory location-based focusing mechanisms



[32] Adam Santoro et al. *One-shot Learning with Memory-Augmented Neural Networks*. 2016. arXiv: 1605.06065.

Data Efficiency

Prioritized Experience Replay[33]

- Traditionally, experience transitions are uniformly sampled from a replay memory
- Simply replays transitions at the same frequency that they were originally experienced, regardless of their significance
- New framework for prioritizing experience, so as to replay important transitions more frequently, and therefore learn more efficiently
- More frequently replay transitions with high expected learning progress, as measured by the magnitude of their temporal-difference (TD) error

[33] Tom Schaul et al. "Prioritized Experience Replay". In: (Nov. 2015). arXiv: 1511.05952. URL: http://arxiv.org/abs/1511.05952.

Transfer Learning

Actor-Mimic Multitask Learning[34]

- Ability to act in multiple environments and transfer previous knowledge to new situations a critical aspect of any intelligent agent
- Novel method of multitask and transfer learning that enables an autonomous agent to learn how to behave in multiple tasks simultaneously, and then generalize its knowledge to new domains
- Train a single policy network that learns how to act in a set of distinct tasks by using the guidance of several expert teachers.
- Representations learnt by the deep policy network are capable of generalizing to new tasks with no prior expert guidance

[34] Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov. "Actor-Mimic: Deep Multitask and Transfer Reinforcement Learning". In: (Nov. 2015). arXiv: 1511.06342. URL: http://arxiv.org/abs/1511.06342.

Transfer Learning

Deep Skill Networks[35]

- Hierarchical Deep RL Network (H-DRLN) extends DQN to facilitate skill reuse in lifelong learning
- The H-DRLN learns a policy that determines when to execute primitive actions and when to reuse pre-learned skills

[35] Chen Tessler et al. "A Deep Hierarchical Approach to Lifelong Learning in Minecraft". In: (2016). arXiv: 1604.07255.

The pre-learned skills are represented with deep networks and are referred to as Deep Skill Networks (DSNs).

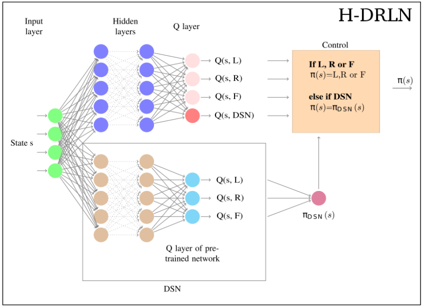
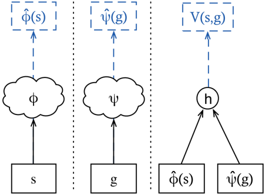


Figure 2: The H-DRLN architecture: Primitive actions include Left (L), Right(R) and Forward (F).

[36] Chen Tessler et al. "A Deep Hierarchical Approach to Lifelong Learning in Minecraft". In: (2016). arXiv: 1604.07255.

Universal Value Function Approximators[37]

- Extends the idea of value function approximation to both states s and goals g
- Uses a universal value function approximator $V(s, g, \theta)$
- A sufficiently expressive function approximator can in principle identify and exploit structure across both s and g
- By universal, we mean that the value function can generalize to any goal g in a set G of possible goals.



[37] Tom Schaul et al. "Universal Value Function Approximators". In: *Proceedings of the 32nd International Conference on Machine Learning* (2015).

Other New Architectures

Deep Recurrent Q-Networks[38]

- Adds recurrency to the Deep Q-network (DQN) by replacing the first post-convolutional fully-connected layer with a recurrent LSTM
- Intended to address partial observability and noisy state information

[38] Matthew Hausknecht and Peter Stone. "Deep Recurrent Q-Learning for Partially Observable MDPs". In: (July 2015). arXiv: 1507.06527. URL: http://arxiv.org/abs/1507.06527.

Other New Architectures

Highway Networks[39]

- Allow unimpeded, direct information flow across many layers
- Gating units learn to regulate the flow of information through a network

[39] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. "Training Very Deep Networks". In: (July 2015), p. 11. arXiv: 1507.06228. URL: http://arxiv.org/abs/1507.06228.

Other New Architectures

Ladder Networks[40]

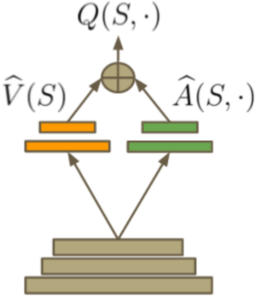
- For deep unsupervised learning
- Autoencoder with lateral shortcut connections from the encoder to decoder at each level of the hierarchy
- Lateral shortcut connections allow the higher levels of the hierarchy to focus on abstract invariant features
- While standard autoencoders are analogous to latent variable models with a single layer of stochastic variables, the proposed network is analogous to hierarchical latent variables models.

[40] Harri Valpola. "From neural PCA to deep unsupervised learning". In: (Nov. 2014). arXiv: 1411.7783. URL: http://arxiv.org/abs/1411.7783.

Other New Architectures

Deuling Network with Advantage Learning[41]

- Dueling network architecture consists of two streams
- Sharing a common convolutional feature learning module
- Representing the value and advantage functions



[41] Ziyu Wang, Nando de Freitas, and Marc Lanctot. "Dueling Network Architectures for Deep Reinforcement Learning". In: (Nov. 2015), p. 14. arXiv: 1511.06581. URL: http://arxiv.org/abs/1511.06581.

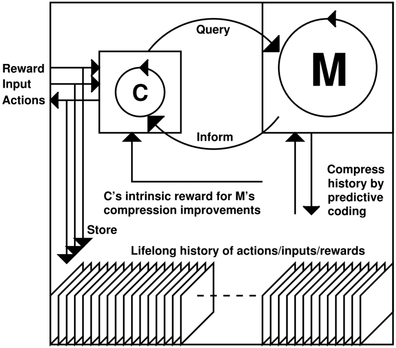
Other New Architectures

Premise Selection for Automated Theorem Proving[42]

- Given a large set of premises P, an ATP system A with given resource limits, and a new conjecture C, predict those premises from P that will most likely lead to an automatically constructed proof of C by A.
- Strong premise selection requires models capable of reasoning over mathematical statements, here encoded as variable-length strings of first-order logic
- Mimics some higher-level reasoning on simple algorithmic tasks
- Extract learned representations of mathematical statements to assist in premise selection and proof

[42] Alex A. Alemi et al. *DeepMath - Deep Sequence Models for Premise Selection*. 2016. arXiv: 1606.04442.

Proposes an architecture with a predictive RNN world-model M along with an RNN controller C which learns to exploit M.



[43] Juergen Schmidhuber. "On Learning to Think: Algorithmic Information Theory for Novel Combinations of Reinforcement Learning Controllers and Recurrent Neural World Models". In: (Nov. 2015), p. 36. arXiv: 1511.09249. URL: http://arxiv.org/abs/1511.09249.

Recent developments, as described in *Towards an integration of deep learning and neuroscience*[44]:

- Structured architectures are used, including dedicated systems for attention, recursion and various forms of short- and long-term memory storage
- Heterogeneously optimized systems, enabled by a series of interacting cost functions, serve to make learning data-efficient and precisely targeted to the needs of the organism

[44] Adam H Marblestone, Greg Wayne, and Konrad P Kording. "Towards an integration of deep learning and neuroscience". In: (2016). DOI: 10.1101/058545. arXiv: 1606.03813.

Hybrid Models

Deep Learning with Monte-Carlo Tree Search: UCT to CNN[45]

Use slow planning-based agents to provide training data for a deep-learning architecture capable of real-time play.

Methods for combining UCT-based RL with DL:

- UCT to CNN via Regression
- UCT to CNN via Classification
- UCT to CNN via Classification-Interleaved

Focus planning on that part of the state space experienced by the (partially trained) CNN player. Continue alternating between training the CNN and UCT planning rollouts.

[45] Xiaoxiao Guo et al. "Deep Learning for Real-Time Atari Game Play Using Offline Monte-Carlo Tree Search Planning". In: *Advances in Neural Information Processing Systems*. 2014, pp. 3338–3346.

AlphaGo: MCTS + Deep Learning

New search algorithm that combines Monte Carlo simulation with value networks (to evaluate board positions) and policy networks (to select moves).

[46] David Silver et al. "Mastering the game of Go with deep neural networks and tree search". In: *Nature* 529.7585 (2016), pp. 484–489. ISSN: 0028-0836. DOI: 10.1038/nature16961. URL: http://dx.doi.org/10.1038/nature16961.

Future

“Future generations of neural networks will look very different from the current state-of-the-art.

They may be endowed with intuitive physics, theory of mind, and causal reasoning.

More structure and inductive biases could be built into the networks or learned from previous experience with related tasks, leading to more human-like patterns of learning and development.”[47]

Additional possibilities:

- Planning
- Reasoning
- Cognitive Architectures

[47] Brenden M. Lake et al. “Building Machines That Learn and Think Like People”. In: (Apr. 2016). arXiv: 1604.00289. URL: http://arxiv.org/abs/1604.00289.

Conclusion

Deep learning methods are relevant to the field of Artificial General Intelligence research, since they are expanding in scope to encompass many types of functionality, and are effectively being combined with other methods to form hybrid architectures.

Expanding in Scope

Expanding to encompass a wide range of methods, including: memory, unsupervised learning, learning to act, program learning and attention.

Hybrid Systems

Increasingly being used in conjunction with other methods to form hybrid architectures.

Contact:

`cosmoh@uw.edu`

`@cosmosquared`

References I

Alex A. Alemi et al. *DeepMath - Deep Sequence Models for Premise Selection*. 2016. arXiv: 1606.04442.

Charles Blundell et al. "Model-Free Episodic Control". In: (June 2016). arXiv: 1606.04460. URL: http://arxiv.org/abs/1606.04460.

Emily Denton et al. "Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks". In: *Advances in Neural Information Processing Systems* (2015), pp. 1486–1494.

Alexey Dosovitskiy, Jost Tobias Springenberg, and Thomas Brox. *Learning To Generate Chairs With Convolutional Neural Networks*. 2015. arXiv: 1411.5928.

References II

- Ian Goodfellow et al. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems*. 2014, pp. 2672–2680. URL: http://papers.nips.cc/paper/5423-generative-adversarial-nets.
- Alex Graves, Greg Wayne, and Ivo Danihelka. "Neural Turing Machines". In: *arXiv preprint arXiv:1410.5401* (2014).
- Xiaoxiao Guo et al. "Deep Learning for Real-Time Atari Game Play Using Offline Monte-Carlo Tree Search Planning". In: *Advances in Neural Information Processing Systems*. 2014, pp. 3338–3346.
- Matthew Hausknecht and Peter Stone. "Deep Recurrent Q-Learning for Partially Observable MDPs". In: (July 2015). arXiv: 1507.06527. URL: http://arxiv.org/abs/1507.06527.

References III

Łukasz Kaiser and Ilya Sutskever. "Neural GPUs Learn Algorithms". In: (Nov. 2015). arXiv: 1511.08228. URL: http://arxiv.org/abs/1511.08228.

Diederik P Kingma and Max Welling. "Auto-Encoding Variational Bayes". In: (Dec. 2013). arXiv: 1312.6114. URL: http://arxiv.org/abs/1312.6114.

Diederik P Kingma et al. "Semi-Supervised Learning with Deep Generative Models". In: *arXiv.org* cs.LG (June 2014), pp. 1–9. arXiv: arXiv:1406.5298v1. URL: http://arxiv.org/abs/1406.5298v2.

Jan Koutník, Jürgen Schmidhuber, and Faustino Gomez. "Evolving deep unsupervised convolutional networks for vision-based reinforcement learning". In: *Proceedings of the 2014 Conference on Genetic and Evolutionary Computation*. ACM. 2014, pp. 541–548.

References IV

Jan Koutník, Jürgen Schmidhuber, and Faustino Gomez. "Online evolution of deep convolutional network for vision-based reinforcement learning". In: *Lecture Notes in Computer Science* 8575 LNAI (2014), pp. 260–269.

Jan Koutník et al. "Evolving large-scale neural networks for vision-based reinforcement learning". In: *Proceeding of the 2013 Conference on Genetic and Evolutionary Computation*. New York, New York, USA: ACM Press, July 2013, p. 1061.

Tejas D. Kulkarni et al. "Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation". In: (Apr. 2016), p. 13. arXiv: 1604.06057. URL: http://arxiv.org/abs/1604.06057%7B%5C#%7D.

Brenden M. Lake et al. "Building Machines That Learn and Think Like People". In: (Apr. 2016). arXiv: 1604.00289. URL: http://arxiv.org/abs/1604.00289.

References V

Máté Lengyel and Peter Dayan. "Hippocampal Contributions to Control: The Third Way". In: *Advances in Neural Information Processing Systems 2007.* (2007), pp. 889–896.

Adam H Marblestone, Greg Wayne, and Konrad P Kording. "Towards an integration of deep learning and neuroscience". In: (2016). DOI: 10.1101/058545. arXiv: 1606.03813.

Volodymyr Mnih et al. "Human-level control through deep reinforcement learning". In: *Nature* 518.7540 (2015), pp. 529–533. ISSN: 0028-0836. DOI: 10.1038/nature14236. arXiv: 1312.5602. URL: http://dx.doi.org/10.1038/nature14236.

Volodymyr Mnih et al. "Playing Atari with Deep Reinforcement Learning". In: (Dec. 2013). arXiv: 1312.5602. URL: http://arxiv.org/abs/1312.5602.

References VI

- Shakir Mohamed and Danilo Jimenez Rezende. *Variational Information Maximisation for Intrinsically Motivated Reinforcement Learning*. 2015.
- Junhyuk Oh et al. "Control of Memory, Active Perception, and Action in Minecraft". In: (May 2016). arXiv: 1605.09128. URL: http://arxiv.org/abs/1605.09128.
- Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov. "Actor-Mimic: Deep Multitask and Transfer Reinforcement Learning". In: (Nov. 2015). arXiv: 1511.06342. URL: http://arxiv.org/abs/1511.06342.
- Scott Reed and Nando de Freitas. "Neural Programmer-Interpreters". In: (Nov. 2015). arXiv: 1511.06279. URL: http://arxiv.org/abs/1511.06279.

References VII

- Adam Santoro et al. *One-shot Learning with Memory-Augmented Neural Networks*. 2016. arXiv: 1605.06065.
- Tom Schaul et al. "Prioritized Experience Replay". In: (Nov. 2015). arXiv: 1511.05952. URL: http://arxiv.org/abs/1511.05952.
- Tom Schaul et al. "Universal Value Function Approximators". In: *Proceedings of the 32nd International Conference on Machine Learning* (2015).
- Juergen Schmidhuber. "On Learning to Think: Algorithmic Information Theory for Novel Combinations of Reinforcement Learning Controllers and Recurrent Neural World Models". In: (Nov. 2015), p. 36. arXiv: 1511.09249. URL: http://arxiv.org/abs/1511.09249.
- Jürgen Schmidhuber. "Deep learning in neural networks: An overview". In: *Neural Networks* 61 (2015), pp. 85–117.

References VIII

David Silver et al. "Mastering the game of Go with deep neural networks and tree search". In: *Nature* 529.7585 (2016), pp. 484–489. ISSN: 0028-0836. DOI: 10.1038/nature16961. URL: http://dx.doi.org/10.1038/nature16961.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. "Training Very Deep Networks". In: (July 2015), p. 11. arXiv: 1507.06228. URL: http://arxiv.org/abs/1507.06228.

Chen Tessler et al. "A Deep Hierarchical Approach to Lifelong Learning in Minecraft". In: (2016). arXiv: 1604.07255.

Harri Valpola. "From neural PCA to deep unsupervised learning". In: (Nov. 2014). arXiv: 1411.7783. URL: http://arxiv.org/abs/1411.7783.

Joel Veness et al. "A Monte-Carlo AIXI Approximation". In: *Journal of Artificial Intelligence Research* 40.1 (2011), pp. 95–142.

References IX

Ziyu Wang, Nando de Freitas, and Marc Lanctot. "Dueling Network Architectures for Deep Reinforcement Learning". In: (Nov. 2015), p. 14. arXiv: 1511.06581. URL: http://arxiv.org/abs/1511.06581.