# MONTRÉAL.AI ACADEMY: ARTIFICIAL INTELLIGENCE 101 FIRST WORLD-CLASS OVERVIEW OF AI FOR ALL **VIP AI 101 CHEATSHEET**

A PREPRINT

**Vincent Boucher***
MONTRÉAL.AI
Montreal, Quebec, Canada
info@montreal.ai

September 25, 2020

## ABSTRACT

**For the purpose** of entrusting all sentient beings with powerful AI tools to learn, deploy and scale AI in order to enhance their prosperity, to settle planetary-scale problems and to inspire those who, with AI, will shape the 21st Century, **MONTRÉAL.AI** introduces this ***VIP AI 101 CheatSheet*** for All.

**MONTRÉAL.AI is preparing a global network of education centers.*
******ALL OF EDUCATION, FOR ALL**. MONTRÉAL.AI is developing a teacher (**Saraswati AI**) and an agent learning to orchestrate synergies amongst academic disciplines (**Polymatheia AI**).*

Curated Open-Source Codes and Science: `http://www.academy.montreal.ai/`.

***K*eywords** AI-First · Artificial Intelligence · Deep Learning · Reinforcement Learning · Symbolic AI

## 1 AI-First

TODAY'S ARTIFICIAL INTELLIGENCE IS POWERFUL AND ACCESSIBLE TO ALL. AI is capable of transforming industries and opens up a world of new possibilities. **What's important is what you do with AI and how you embrace it**. To pioneer AI-First innovations advantages: start by exploring how to apply AI in ways never thought of.

**The Emerging Rules of the AI-First Era**: Search and Learning.

> "***Search*** *and* ***learning*** *are general purpose methods that continue to scale with increased computation, even as the available computation becomes very great.*" — Richard Sutton in *The Bitter Lesson*

**The Best Way Forward For AI[2].**

> "*... so far as I'm concerned, system 1 certainly knows language, understands language... system 2... it does involve certain manipulation of symbols... Gary Marcus ... Gary proposes something that seems very natural...* ***a hybrid architecture****... I'm influenced by him... if you look introspectively at the way the mind works... you'd get to that distinction between implicit and explicit... explicit looks like symbols.*" — Nobel Laureate Danny Kahneman at *AAAI-20 Fireside Chat with Daniel Kahneman* `https://vimeo.com/390814190`

In *The Next Decade in AI*[3], Gary Marcus proposes a hybrid, knowledge-driven, reasoning-based approach, centered around cognitive models, that could provide the substrate for a richer, more robust AI than is currently possible.

---

*Founding Chairman at MONTRÉAL.AI `http://www.montreal.ai` and QUÉBEC.AI `http://www.quebec.ai`.
[2] `https://montrealartificialintelligence.com/aidebate/`
[3] `https://arxiv.org/abs/2002.06177v3`

## 2 Getting Started

Tinker with neural networks in the browser with *TensorFlow Playground* `http://playground.tensorflow.org/`.

- **CS231n Python Tutorial With Google Colab**[4].
- **Learn with Google AI** `https://ai.google/education/`.
- **Made With ML Topics** `https://madewithml.com/topics/`.
- **One Place for Everything AI** `https://aihub.cloud.google.com/`.
- **Deep Learning Drizzle** `https://deep-learning-drizzle.github.io`.
- **Google Dataset Search** (Blog[5]) `https://datasetsearch.research.google.com`.
- **AI Literacy for K-12 School Children** `https://aieducation.mit.edu/resources`.
- **Learning resources from DeepMind** `https://deepmind.com/learning-resources`.
- **Papers With Code** (*Learn Python 3 in Y minutes*[6]) `https://paperswithcode.com/state-of-the-art`.

"*Dataset Search has indexed almost 25 million of these datasets, giving you a single place to search for datasets and find links to where the data is.*" — Natasha Noy

**The Measure of Intelligence** (*Abstraction and Reasoning Corpus*[7]) `https://arxiv.org/abs/1911.01547`.

❖ Growing Neural Cellular Automata, Mordvintsev et al. `https://distill.pub/2020/growing-ca/`.

### 2.1 In the Cloud

Colab [8]. Practice Immediately [9]. Labs[10]: Introduction to Deep Learning (MIT 6.S191)

- Free GPU compute via Colab `https://colab.research.google.com/notebooks/welcome.ipynb`.
- Colab can open notebooks directly from GitHub by simply replacing "*http://github.com*" with "*http://colab.research.google.com/github/*" in the notebook URL.
- Colab Pro `https://colab.research.google.com/signup`.

### 2.2 On a Local Machine

JupyterLab is an interactive development environment for working with notebooks, code and data [11].

- Install Anaconda `https://www.anaconda.com/download/` and launch '*Anaconda Navigator*'
- Update Jupyterlab and launch the application. Under Notebook, click on '*Python 3*'

IDE: Visual Studio Code `https://code.visualstudio.com/`.

"*If we truly reach AI, it will let us know.*" — Garry Kasparov

## 3 Deep Learning

Learning according to Mitchell (1997):

"*A computer program is said to learn from experience* **E** *with respect to some class of tasks* **T** *and performance measure* **P**, *if its performance at tasks in* **T**, *as measured by* **P**, *improves with experience* **E**." — Tom Mitchell

[4] `https://colab.research.google.com/github/cs231n/cs231n.github.io/blob/master/python-colab.ipynb`
[5] `https://blog.google/products/search/discovering-millions-datasets-web/`
[6] `https://learnxinyminutes.com/docs/python3/`
[7] `https://github.com/fchollet/ARC`
[8] `https://medium.com/tensorflow/colab-an-easy-way-to-learn-and-use-tensorflow-d74d1686e309`
[9] `https://colab.research.google.com/github/madewithml/practicalAI/`
[10] `https://colab.research.google.com/github/aamini/introtodeeplearning_labs`
[11] `https://blog.jupyter.org/jupyterlab-is-ready-for-users-5a6f039b8906`

After the **Historical AI Debate**[12]: "*Yoshua Bengio and Gary Marcus on the Best Way Forward for AI*" `https://montrealartificialintelligence.com/aidebate/`, there have been clarifications on the term "***deep learning***"[13].

"*Deep learning is inspired by neural networks of the brain to build learning machines which discover rich and useful internal representations, computed as a composition of learned features and functions.*" — Yoshua Bengio

"*DL is constructing networks of parameterized functional modules and training them from examples using gradient-based optimization.*" — Yann LeCun

"*... replace symbols by vectors and logic by continuous (or differentiable) functions.*" — Yann LeCun

Deep learning allows computational models that are composed of multiple processing layers to learn REPRESENTATIONS of (raw) data with multiple levels of abstraction[2]. At a high-level, neural networks are either encoders, decoders, or a combination of both[14]. Introductory course `http://introtodeeplearning.com`. See also Table 1.

Table 1: Types of Learning, by Alex Graves at NeurIPS 2018

| Name | With Teacher | Without Teacher |
|---|---|---|
| Active | *Reinforcement Learning / Active Learning* | *Intrinsic Motivation / Exploration* |
| Passive | *Supervised Learning* | *Unsupervised Learning* |

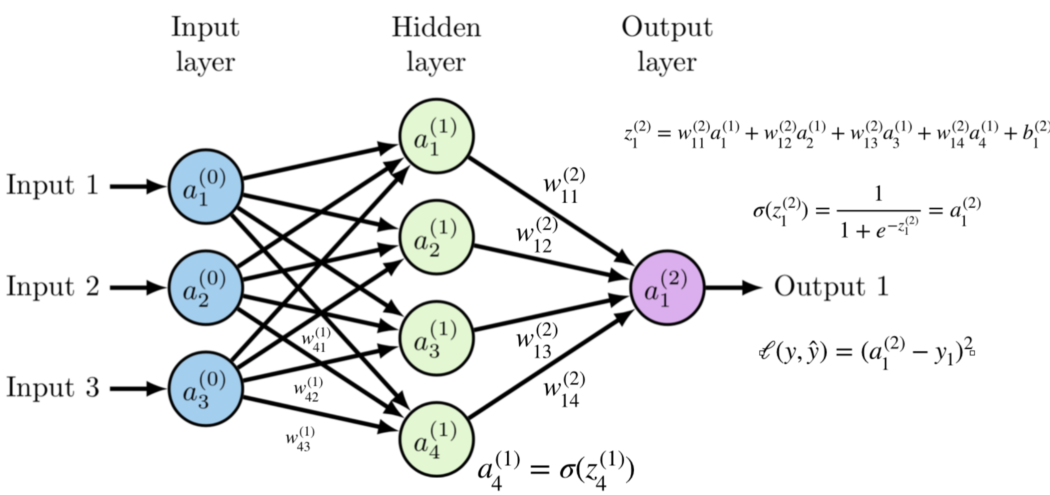



Figure 1: **Multilayer perceptron (MLP)**.

Deep learning assumes that the data was generated by the composition of factors potentially at multiple levels in a hierarchy[15]. Deep learning (*distributed representations + composition*) is a general-purpose learning procedure.

"*When you first study a field, it seems like you have to memorize a zillion things. You don't. What you need is to identify the 3-5 core principles that govern the field. The million things you thought you had to memorize are various combinations of the core principles.*" — J. Reed

"*1. Multiply things together*
*2. Add them up*
*3. Replaces negatives with zeros*
*4. Return to step 1, a hundred times.*"
— Jeremy Howard

[12] `https://www.zdnet.com/article/devils-in-the-details-in-bengio-marcus-ai-debate/`
[13] `https://www.zdnet.com/article/whats-in-a-name-the-deep-learning-debate/`
[14] `https://github.com/lexfridman/mit-deep-learning`
[15] `https://www.deeplearningbook.org`

❖ Linear Algebra. Prof. Gilbert Strang[16].
❖ Dive into Deep Learning http://d2l.ai.
❖ Minicourse in Deep Learning with PyTorch[17].
❖ How to do Research At the MIT AI Lab (1988)[18].
❖ Introduction to Artificial Intelligence, Gilles Louppe[19].
❖ Fast and Easy Infinitely Wide Networks with Neural Tangents[20].
❖ Deep Learning. The full deck of (600+) slides, Gilles Louppe[21].
❖ These Lyrics Do Not Exist https://theselyricsdonotexist.com.
❖ AI and Wargaming, Goodman et al. https://arxiv.org/abs/2009.08922.
❖ Backward Feature Correction: How Deep Learning Performs Deep Learning[22].
❖ A Selective Overview of Deep Learning https://arxiv.org/abs/1904.05526.
❖ The Missing Semester of Your CS Education https://missing.csail.mit.edu.
❖ fastai: A Layered API for Deep Learning https://arxiv.org/abs/2002.04688.
❖ Anatomy of Matplotlib https://github.com/matplotlib/AnatomyOfMatplotlib.
❖ Data project checklist https://www.fast.ai/2020/01/07/data-questionnaire/.
❖ Using Nucleus and TensorFlow for DNA Sequencing Error Correction, Colab Notebook[23].
❖ Machine Learning for Physicists https://machine-learning-for-physicists.org.
❖ Flow-edge Guided Video Completion, Gao et al. https://arxiv.org/abs/2009.01835.
❖ The world as a neural network, Vitaly Vanchurin https://arxiv.org/abs/2008.01540.
❖ Generalized Energy Based Models, Michael Arbel, Liang Zhou and Arthur Gretton, 2020[24].
❖ Representing Scenes as Neural Radiance Fields for View Synthesis. Mildenhall et al., 2020[25].
❖ PoseNet Sketchbook https://googlecreativelab.github.io/posenet-sketchbook/.
❖ The Neural Network, A Visual Introduction. Vivek Verma : https://youtu.be/UOvPeC8WOt8.
❖ Synthetic Data for Deep Learning, Sergey I. Nikolenko https://arxiv.org/abs/1909.11512.
❖ Removing people from complex backgrounds in real time using TensorFlow.js in the web browser[26].
❖ A Recipe for Training Neural Networks https://karpathy.github.io/2019/04/25/recipe/.
❖ TensorFlow Datasets: load a variety of public datasets into TensorFlow programs (Blog[27] | Colab[28]).
❖ Denoising Diffusion Probabilistic Models, Ho et al., 2020 https://arxiv.org/abs/2006.11239.
❖ The Markov-Chain Monte Carlo Interactive Gallery https://chi-feng.github.io/mcmc-demo/.
❖ NeurIPS 2019 Implementations https://paperswithcode.com/conference/neurips-2019-12.
❖ Involutive MCMC: a Unifying Framework, Neklyudov et al. https://arxiv.org/abs/2006.16653.
❖ Algebra, Topology, Differential Calculus, and Optimization Theory For Computer Science and Machine Learning[29].
❖ How to Choose Your First AI Project https://hbr.org/2019/02/how-to-choose-your-first-ai-project.
❖ Blog | MIT 6.S191 https://medium.com/tensorflow/mit-introduction-to-deep-learning-4a6f8dde1f0c.
❖ A Fortran-Keras Deep Learning Bridge for Scientific Computing, Ott et al. https://arxiv.org/abs/2004.10652. GitHub[30].
❖ A Wholistic View of Continual Learning with Deep Neural Networks: Forgotten Lessons and the Bridge to Active and Open World Learning. Mundt et al., 2020[31].

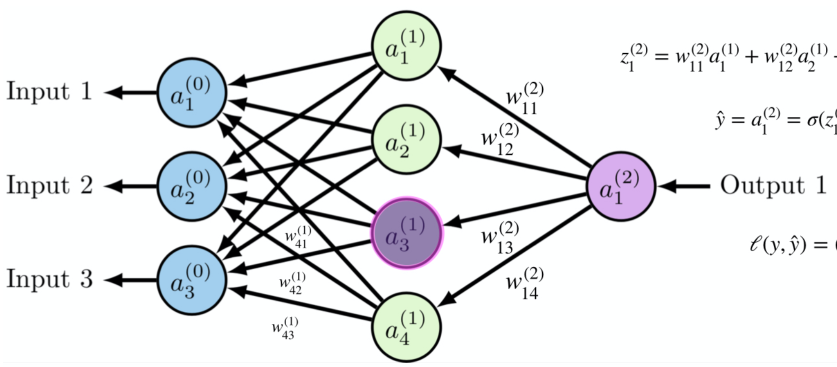
[16] https://ocw.mit.edu/courses/mathematics/18-06-linear-algebra-spring-2010/video-lectures/
[17] https://github.com/Atcold/pytorch-Deep-Learning-Minicourse
[18] http://dspace.mit.edu/bitstream/handle/1721.1/41487/AI_WP_316.pdf
[19] https://glouppe.github.io/info8006-introduction-to-ai/pdf/lec-all.pdf
[20] https://ai.googleblog.com/2020/03/fast-and-easy-infinitely-wide-networks.html
[21] https://github.com/glouppe/info8010-deep-learning/raw/v2-info8010-2019/pdf/lec-all.pdf
[22] https://arxiv.org/abs/2001.04413
[23] https://colab.research.google.com/github/google/nucleus/blob/master/nucleus/examples/dna_sequencing_error_correction.ipynb
[24] https://arxiv.org/abs/2003.05033
[25] http://www.matthewtancik.com/nerf
[26] https://github.com/jasonmayes/Real-Time-Person-Removal
[27] https://medium.com/tensorflow/introducing-tensorflow-datasets-c7f01f7e19f3
[28] https://colab.research.google.com/github/tensorflow/datasets/blob/master/docs/overview.ipynb
[29] https://drive.google.com/file/d/1sJvLQwxMyu89t2z4Zf9tD7O7efnbIUyB/view
[30] https://github.com/scientific-computing/FKB
[31] https://arxiv.org/abs/2009.01797

### 3.1 Universal Approximation Theorem

The universal approximation theorem states that a feed-forward network with a single hidden layer containing a finite number of neurons can solve any given problem to arbitrarily close accuracy as long as you add enough parameters.

Neural Networks + Gradient Descent + GPU[32]:

- Infinitely flexible function: *Neural Network* (multiple hidden layers: Deep Learning)[33].
- All-purpose parameter fitting: *Backpropagation*[34][35]. Backpropagation is the key algorithm that makes training deep models computationally tractable and highly efficient[36]. The backpropagation procedure is nothing more than a practical application of the chain rule for derivatives.




$$z_1^{(2)} = w_{11}^{(2)}a_1^{(1)} + w_{12}^{(2)}a_2^{(1)} + w_{13}^{(2)}a_3^{(1)} + w_{14}^{(2)}a_4^{(1)} + b_1^{(2)}$$

$$\hat{y} = a_1^{(2)} = \sigma(z_1^{(2)}) = \frac{1}{1+e^{-z_1^{(2)}}}$$

$$\ell(y,\hat{y}) = (a_1^{(2)} - y_1)^2$$

$$\frac{\delta\ell(y,\hat{y})}{\delta a_3^{(1)}} = \frac{\delta\ell(y,\hat{y})}{\delta a_1^{(2)}}\frac{\delta a_1^{(2)}}{\delta z_1^{(2)}}\frac{\delta z_1^{(2)}}{\delta a_3^{(1)}} = 2(a_1^{(2)} - y_1)z_1^{(2)}(1 - z_1^{(2)})w_{13}^{(2)}$$

Figure 2: **All-purpose parameter fitting: Backpropagation**.

- Fast and scalable: *GPU*.

"*You have relatively simple processing elements that are very loosely models of neurons. They have connections coming in, each connection has a weight on it, and that weight can be changed through learning.*" — Geoffrey Hinton

Deep learning : connect a dataset, a model, a cost function and an optimization procedure.

"*Deep learning has fully solved the curse of dimensionality. It vanished like an RNN gradient!*" — Ilya Sutskever

When a choice must be made, just feed the (raw) data to a deep neural network (Universal function approximators).

### 3.2 Convolution Neural Networks (Useful for Images | Space)

**Richer innate priors : *innateness that enables learning.***

A significant percentage of Deep Learning breakthroughs comes from reusable constructs and parameters sharing. The deep convolutional network is a construct that reuses weights in multiple locations (parameters sharing in space)[37].

"*Virtually all modern observers would concede that genes and experience work together; it is "nature and nurture", not "nature versus nurture". No nativist, for instance, would doubt that we are also born with specific biological machinery that allows us to learn. Chomsky's Language Acquisition Device should be viewed precisely as an innate learning mechanism, and nativists such as Pinker, Peter Marler (Marler, 2004) and myself (Marcus, 2004) have frequently argued for a view in which a significant part of a creature's innate armamentarium consists not of specific*

[32] `http://wiki.fast.ai/index.php/Lesson_1_Notes`
[33] `http://neuralnetworksanddeeplearning.com/chap4.html`
[34] `https://github.com/DebPanigrahi/Machine-Learning/blob/master/back_prop.ipynb`
[35] `https://www.jeremyjordan.me/neural-networks-training/`
[36] `https://colah.github.io/posts/2015-08-Backprop/`
[37] `https://twitter.com/iamtrask/status/949439556499230720`

*knowledge but of learning mechanisms, a form of* ***innateness that enables learning.***" — Gary Marcus, Innateness, AlphaZero, and Artificial Intelligence[38]

The deep convolutional network, inspired by Hubel and Wiesel's seminal work on early visual cortex, uses hierarchical layers of tiled convolutional filters to mimic the effects of receptive fields, thereby exploiting the local spatial correlations present in images[1]. See Figure 4. Demo `https://ml4a.github.io/demos/convolution/`.

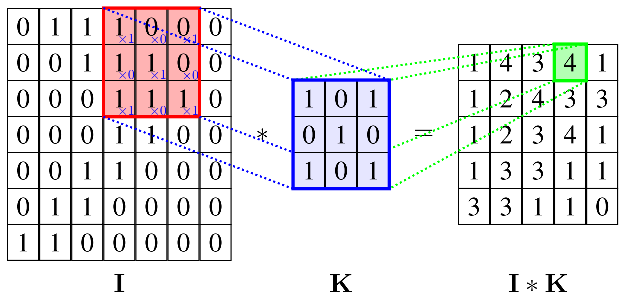



Figure 3: **2D Convolution**. Source: Cambridge Coding Academy

A ConvNet is made up of Layers. Every Layer has a simple API: It transforms an input 3D volume to an output 3D volume with some differentiable function that may or may not have parameters[39]. Reading[40].

In images, local combinations of edges form motifs, motifs assemble into parts, and parts form objects[41][42].

**Representation learning :** ***the language of neural networks.*** The visual vocabulary of a convolutional neural network seems to emerge from low level features such as edges and orientations, and builds up textures, patterns and composites, . . . and builds up even further into complete objects. This relates to Wittgenstein's "*language-game*" in ***Philosophical Investigations***[43], where a functional language emerge from simple tasks before defining a vocabulary[44].

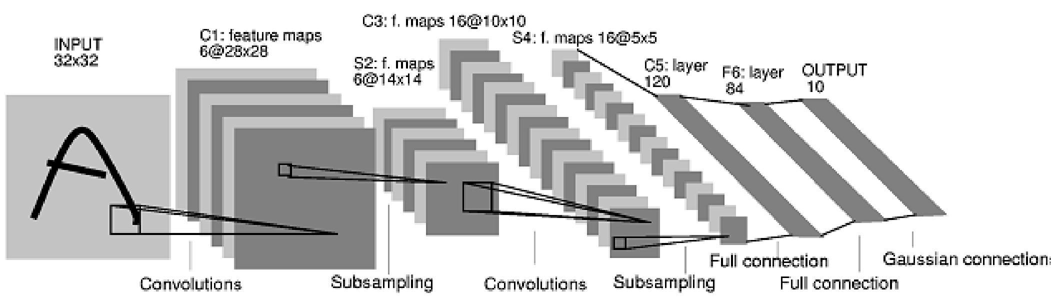



Figure 4: Architecture of LeNet-5, a Convolutional Neural Network. LeCun et al., 1998

"*DL is essentially a new style of programming – "differentiable programming" – and the field is trying to work out the reusable constructs in this style. We have some: convolution, pooling, LSTM, GAN, VAE, memory units, routing units, etc.*" — Thomas G. Dietterich

❖ Image Classification from Scratch[45].
❖ CS231N : Convolutional Neural Networks for Visual Recognition[46].

[38] `https://arxiv.org/abs/1801.05667`
[39] `http://cs231n.github.io/convolutional-networks/`
[40] `https://ml4a.github.io/ml4a/convnets/`
[41] `http://yosinski.com/deepvis`
[42] `https://distill.pub/2017/feature-visualization/`
[43] `https://en.wikipedia.org/wiki/Philosophical_Investigations`
[44] `https://media.neurips.cc/Conferences/NIPS2018/Slides/Deep_Unsupervised_Learning.pdf`
[45] `https://colab.research.google.com/drive/1umJnCp8tZ7UDTYSQsuWdKRhqbHts38AC`
[46] `https://www.youtube.com/playlist?list=PLzUTmXVwsnXod6WNdg57Yc3zFx_f-RYsq`

❖ Introduction to Graph Convolutional Network (GCN). Alfredo Canziani[47].
❖ Deep Plastic Surgery: Robust and Controllable Image Editing with Human-Drawn Sketches. Yang et al.[48].
❖ CNN Explainer: Learning Convolutional Neural Networks with Interactive Visualization. Wang et al. [49] [50].
❖ An Overview of Early Vision in InceptionV1 `https://distill.pub/2020/circuits/early-vision/`.
❖ Neural Voice Puppetry: Audio-driven Facial Reenactment. Thies et al. `https://arxiv.org/abs/1912.05566`.
❖ TensorSpace (`https://tensorspace.org`) offers interactive 3D visualizations of *LeNet*, *AlexNet* and *Inceptionv3*.

### 3.3 Recurrent Neural Networks (Useful for Sequences | Time)

Recurrent neural networks are networks with loops in them, allowing information to persist[51]. RNNs process an input sequence one element at a time, maintaining in their hidden units a ‘state vector’ that implicitly contains information about the history of all the past elements of the sequence[2]. For sequential inputs. See Figure 5.

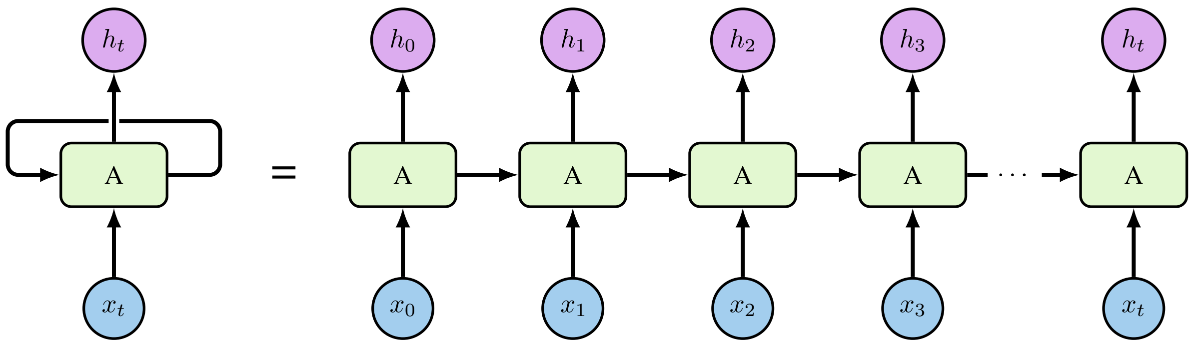



Figure 5: **RNN Layers Reuse Weights for Multiple Timesteps**.

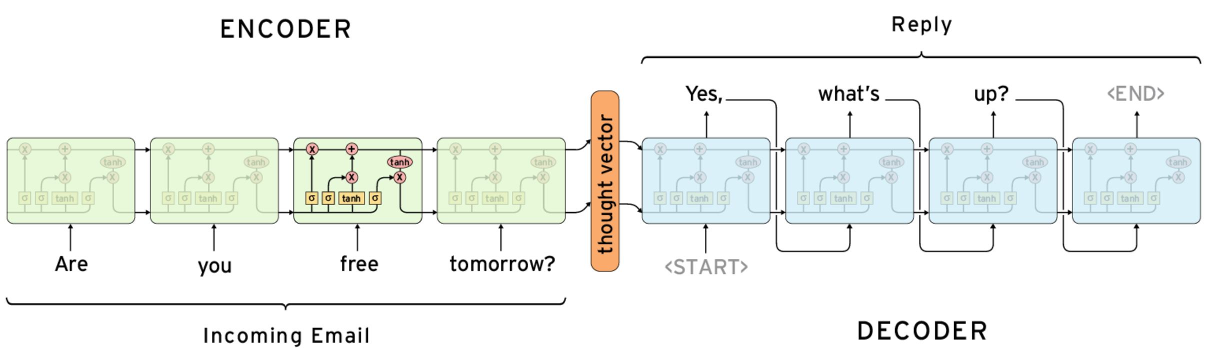



Figure 6: **Google *Smart Reply System* is built on a pair of recurrent neural networks**. Diagram by Chris Olah

"*I feel like a significant percentage of Deep Learning breakthroughs ask the question “how can I reuse weights in multiple places?” – Recurrent (LSTM) layers reuse for multiple timesteps – Convolutional layers reuse in multiple locations. – Capsules reuse across orientation.*" — Andrew Trask

❖ CS224N : Natural Language Processing with Deep Learning[52].
❖ Long Short-Term-Memory (LSTM), Sepp Hochreiter and Jürgen Schmidhuber[53].
❖ The Unreasonable Effectiveness of Recurrent Neural Networks, blog (2015) by Andrej Karpathy[54].

[47] `https://atcold.github.io/pytorch-Deep-Learning/en/week13/13-3/`
[48] `https://arxiv.org/abs/2001.02890`
[49] `https://arxiv.org/abs/2004.15004`
[50] `http://poloclub.github.io/cnn-explainer/`
[51] `http://colah.github.io/posts/2015-08-Understanding-LSTMs/`
[52] `https://www.youtube.com/playlist?list=PLU40WL8Ol94IJzQtileLTqGZuXtGlLMP_`
[53] `https://www.bioinf.jku.at/publications/older/2604.pdf`
[54] `http://karpathy.github.io/2015/05/21/rnn-effectiveness/`

❖ Understanding LSTM Networks `http://colah.github.io/posts/2015-08-Understanding-LSTMs/`.
❖ Can Neural Networks Remember? Slides by Vishal Gupta: `http://vishalgupta.me/deck/char_lstms/`.

### 3.4 Transformers

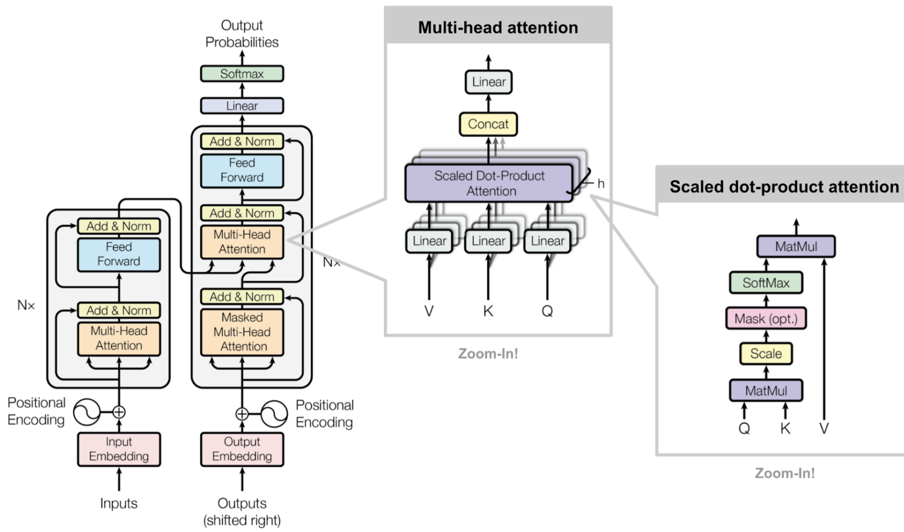



Figure 7: Attention Is All You Need. *Vaswani et al., 2017* : `https://arxiv.org/abs/1706.03762`.

Transformers are generic, simples and exciting machine learning architectures designed to process a connected set of units (tokens in a sequence, pixels in an image, etc.) where the only interaction between units is through self-attention. Transformers' performance limit seems purely in the hardware (how big a model can be fitted in GPU memory)[55].

The fundamental operation of transformers is **self-attention** (a sequence-to-sequence operation, Figure 8): *an attention mechanism relating different positions of a single sequence in order to compute a representation of the same sequence*[56].

Let's call the input vectors (of dimension $k$) :

$$\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_t \tag{1}$$

Let's call the corresponding output vectors (of dimension $k$) :

$$\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_t \tag{2}$$

The **self attention** operation takes a weighted average over all the input vectors :

$$\mathbf{y}_i = \sum_j w_{ij} \mathbf{x}_j \tag{3}$$

The weight $w_{\mathrm{ij}}$ is derived from a function over $\mathbf{x}_{\mathrm{i}}$ and $\mathbf{x}_{\mathrm{j}}$. The simplest option is the dot product (with softmax) :

$$w_{ij} = \frac{e^{\mathbf{x}_i^T \mathbf{x}_j}}{\sum_j e^{\mathbf{x}_i^T \mathbf{x}_j}} \tag{4}$$

Transformers are Graph Neural Networks[57].

❖ The Transformer Family. By *Lilian Weng*[58].
❖ Transformers Notebooks. By *Hugging Face*[59].

[55] `http://www.peterbloem.nl/blog/transformers`
[56] `https://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html`
[57] `https://graphdeeplearning.github.io/post/transformers-are-gnns/`
[58] `https://lilianweng.github.io/lil-log/2020/04/07/the-transformer-family.html`
[59] `https://github.com/huggingface/transformers/tree/master/notebooks`

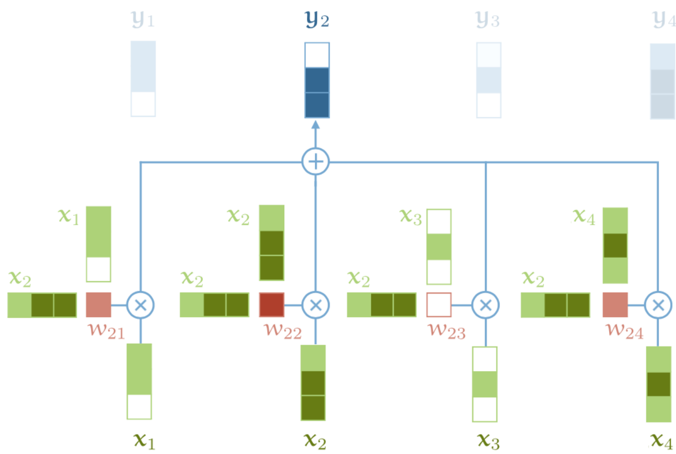



Figure 8: Self-attention. By *Peter Bloem* : `http://www.peterbloem.nl/blog/transformers`.

❖ Text classification with Transformer. Colab[60].
❖ Making Transformer networks simpler and more efficient[61].
❖ AttentioNN: All about attention in neural networks described as colab notebooks[62].
❖ Attention Is All You Need, Vaswani et al. `https://arxiv.org/abs/1706.03762`.
❖ Efficient Transformers: A Survey, Tay et al. `https://arxiv.org/abs/2009.06732`.
❖ How to train a new language model from scratch using Transformers and Tokenizers[63].
❖ Write With Transformer. By *Hugging Face*: `https://transformer.huggingface.co`.
❖ The Illustrated Transformer `http://jalammar.github.io/illustrated-transformer/`.
❖ How to generate text: using different decoding methods for language generation with Transformers[64].
❖ The annotated transformer (code) `http://nlp.seas.harvard.edu/2018/04/03/attention.html`.
❖ Attention and Augmented Recurrent Neural Networks `https://distill.pub/2016/augmented-rnns/`.
❖ Transformer model for language understanding. Tutorial showing how to write Transformer in TensorFlow 2.0[65].
❖ End-to-End Object Detection with Transformers, Carion et al. `https://arxiv.org/abs/2005.12872`. Colab[66].
❖ Transformer in TensorFlow 2.0 (code) `https://www.tensorflow.org/beta/tutorials/text/transformer`.

### 3.4.1 Natural Language Processing (NLP) | BERT: A New Era in NLP

BERT (Bidirectional Encoder Representations from Transformers)[6] is a *deeply bidirectional, unsupervised language representation*, pre-trained using only a plain text corpus (in this case, Wikipedia)[67].

- Reading: Unsupervised pre-training of an LSTM followed by supervised fine-tuning[7].
- TensorFlow code and pre-trained models for BERT `https://github.com/google-research/bert`.
- Better Language Models and Their Implications[68].

"*I think transfer learning is the key to general intelligence. And I think the key to doing transfer learning will be the acquisition of conceptual knowledge that is abstracted away from perceptual details of where you learned it from.*" — Demis Hassabis

[60] `https://colab.research.google.com/github/keras-team/keras-io/blob/master/examples/nlp/ipynb/text_classification_with_transformer.ipynb`
[61] `https://ai.facebook.com/blog/making-transformer-networks-simpler-and-more-efficient/`
[62] `https://github.com/zaidalyafeai/AttentioNN`
[63] `https://huggingface.co/blog/how-to-train`
[64] `https://huggingface.co/blog/how-to-generate`
[65] `https://www.tensorflow.org/tutorials/text/transformer`
[66] `https://colab.research.google.com/drive/1rPm0-UrWHpJJRX9PsNb5SpzZiUlMh7wm`
[67] `https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html`
[68] `https://blog.openai.com/better-language-models/`

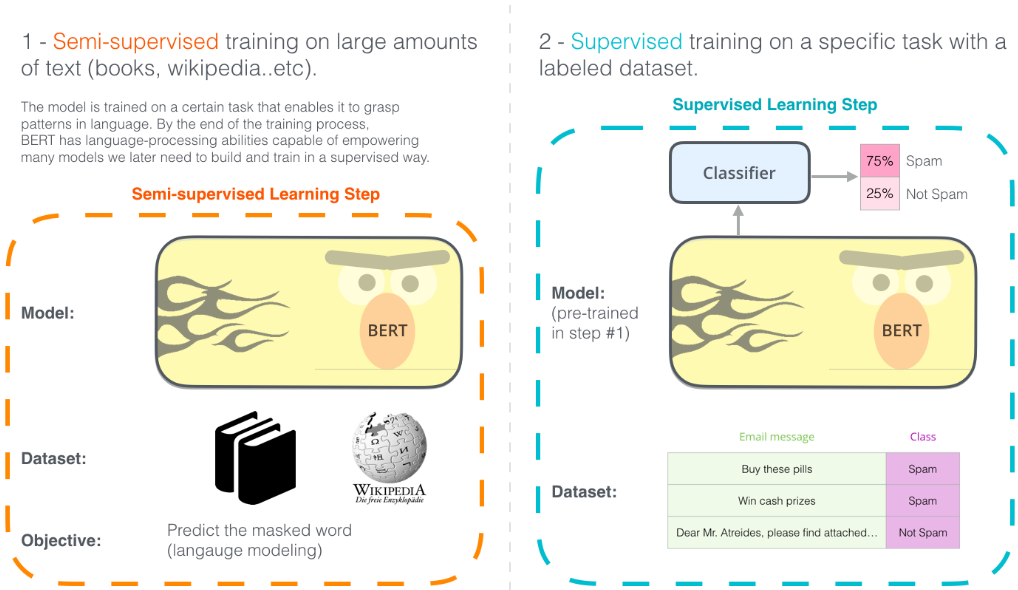



Figure 9: The two steps of how BERT is developed. Source https://jalammar.github.io/illustrated-bert/.

❖ Towards a Conversational Agent that Can Chat About. . . Anything[69].
❖ How to Build OpenAI's GPT-2: "*The AI That's Too Dangerous to Release*"[70].
❖ A Primer in BERTology: What we know about how BERT works, Rogers et al., 2020[71].
❖ Play with BERT with your own data using TensorFlow Hub https://colab.research.google.com/github/google-research/bert/blob/master/predicting_movie_reviews_with_bert_on_tf_hub.ipynb.

### 3.5 Unsupervised Learning

True intelligence will require independent learning strategies.

"*Give a robot a label and you feed it for a second; teach a robot to label and you feed it for a lifetime.*" — Pierre Sermanet

Unsupervised learning is a paradigm for creating AI that learns without a particular task in mind: learning for the sake of learning[72]. It captures some characteristics of the joint distribution of the observed random variables (learn the underlying structure). The variety of tasks include density estimation, dimensionality reduction, and clustering.[4][73].

"*The unsupervised revolution is taking off!*" — Alfredo Canziani

**Self-supervised learning** is derived form unsupervised learning where the data provides the supervision. E.g. Word2vec[74], a technique for learning vector representations of words, or word **embeddings**. An embedding is a mapping from discrete objects, such as words, to vectors of real numbers[75].

"*The next revolution of AI won't be supervised.*" — Yann LeCun

"*Self-supervised learning is a method for attacking unsupervised learning problems by using the mechanisms of supervised learning.*" — Thomas G. Dietterich

[69] https://ai.googleblog.com/2020/01/towards-conversational-agent-that-can.html
[70] https://blog.floydhub.com/gpt2/
[71] https://arxiv.org/abs/2002.12327
[72] https://deepmind.com/blog/unsupervised-learning/
[73] https://media.neurips.cc/Conferences/NIPS2018/Slides/Deep_Unsupervised_Learning.pdf
[74] https://jalammar.github.io/illustrated-word2vec/
[75] http://projector.tensorflow.org

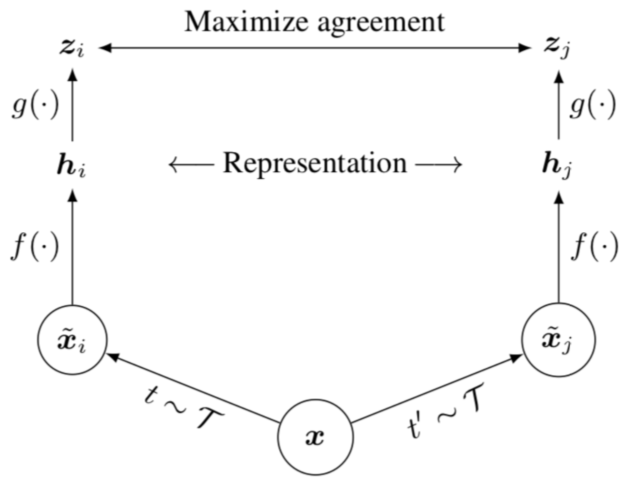



Figure 10: A Simple Framework for Contrastive Learning of Visual Representations, Chen et al., 2020

❖ Self-Supervised Image Classification, Papers With Code[76].
❖ Self-supervised learning and computer vision, Jeremy Howard[77].
❖ Momentum Contrast for Unsupervised Visual Representation Learning, He et al.[78]
❖ Data-Efficient Image Recognition with Contrastive Predictive Coding, Hénaff et al.[79]
❖ A Simple Framework for Contrastive Learning of Visual Representations, Chen et al.[80]
❖ FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence, Sohn et al.[81]
❖ Self-classifying MNIST Digits, Randazzo et al.: `https://distill.pub/2020/selforg/mnist/`.
❖ Self-Supervised Learning of Pretext-Invariant Representations, Ishan Misra, Laurens van der Maaten[82].

### 3.5.1 Generative Adversarial Networks

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. The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game[3].

$$\min_{\theta_g} \max_{\theta_d} [\mathbb{E}_{\mathrm{x} \sim \mathrm{p_{data}(x)}} [log D_{\theta_d}(x)] + \mathbb{E}_{\mathrm{z} \sim \mathrm{p_z(z)}} [log(1 - D_{\theta_d}(G_{\theta_g}(z)))]] \tag{5}$$

"*What I cannot create, I do not understand.*" — Richard Feynman

Goodfellow et al. used an interesting analogy where the generative model can be thought of as analogous to a team of counterfeiters, trying to produce fake currency and use it without detection, while the discriminative model is analogous to the police, trying to detect the counterfeit currency. Competition in this game drives both teams to improve their methods until the counterfeits are indistiguishable from the genuine articles. See Figure 11.

StyleGAN: A Style-Based Generator Architecture for Generative Adversarial Networks

- Paper `http://stylegan.xyz/paper` | Code `https://github.com/NVlabs/stylegan`.
- **StyleGAN for art.** Colab `https://colab.research.google.com/github/ak9250/stylegan-art`.
- This Person Does Not Exist `https://thispersondoesnotexist.com`.
- Which Person Is Real? `http://www.whichfaceisreal.com`.

---

[76] `https://paperswithcode.com/task/self-supervised-image-classification`
[77] `https://www.fast.ai/2020/01/13/self_supervised/`
[78] `https://arxiv.org/abs/1911.05722`
[79] `https://arxiv.org/abs/1905.09272`
[80] `https://arxiv.org/abs/2002.05709`
[81] `https://arxiv.org/abs/2001.07685`
[82] `https://arxiv.org/abs/1912.01991`

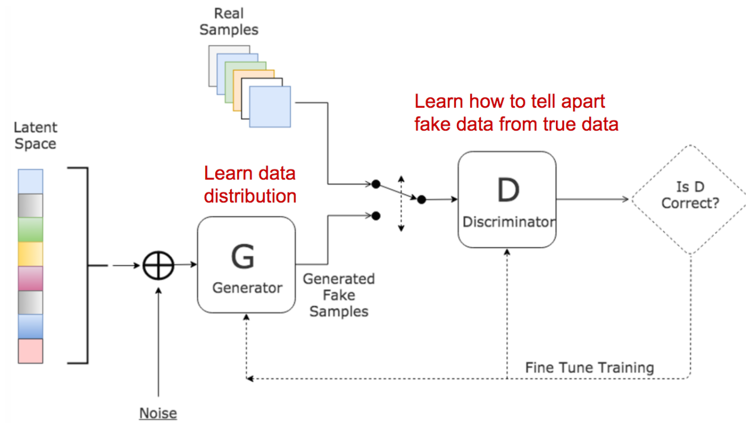



Figure 11: GAN: Neural Networks Architecture Pioneered by Ian Goodfellow at University of Montreal (2014).

- This Resume Does Not Exist https://thisresumedoesnotexist.com.
- This Waifu Does Not Exist https://www.thiswaifudoesnotexist.net.
- Encoder for Official TensorFlow Implementation https://github.com/Puzer/stylegan-encoder.
- How to recognize fake AI-generated images. By Kyle McDonald[83].

❖ GAN in Keras. Colab[84].
❖ 100,000 Faces Imagined by a GAN https://generated.photos.
❖ Introducing **TF-GAN**: A lightweight GAN library for TensorFlow 2.0[85].
❖ Generative Adversarial Networks (GANs) in 50 lines of code (PyTorch)[86].
❖ Few-Shot Adversarial Learning of Realistic Neural Talking Head Models[87].
❖ Wasserstein GAN http://www.depthfirstlearning.com/2019/WassersteinGAN.
❖ GANpaint Paint with GAN units http://gandissect.res.ibm.com/ganpaint.html.
❖ StyleGAN2 Distillation for Feed-forward Image Manipulation. Viazovetskyi et al.[88] Code[89].
❖ A Review on Generative Adversarial Networks: Algorithms, Theory, and Applications. Gui et al.[90].
❖ CariGANs: Unpaired Photo-to-Caricature Translation. Cao et al.: https://cari-gan.github.io.
❖ Infinite-resolution (CPPNs, GANs and TensorFlow.js) https://thispicturedoesnotexist.com.
❖ PyTorch pretrained BigGAN https://github.com/huggingface/pytorch-pretrained-BigGAN.
❖ GANSynth: Generate high-fidelity audio with GANs! Colab http://goo.gl/magenta/gansynth-demo.
❖ SC-FEGAN: Face Editing Generative Adversarial Network https://github.com/JoYoungjoo/SC-FEGAN.
❖ Demo of BigGAN in an official Colaboratory notebook (backed by a GPU) https://colab.research.google.com/github/tensorflow/hub/blob/master/examples/colab/biggan_generation_with_tf_hub.ipynb

### 3.5.2 Variational AutoEncoder

Variational Auto-Encoders[91] (VAEs) are powerful models for learning low-dimensional representations See Figure 12. Disentangled representations are defined as ones where a change in a single unit of the representation corresponds to a change in single factor of variation of the data while being invariant to others (Bengio et al. (2013).

[83] https://medium.com/@kcimc/how-to-recognize-fake-ai-generated-images-4d1f6f9a2842
[84] https://colab.research.google.com/drive/1CQ2XTMoUB7b9i9USUh4kp8BoCag1z-en
[85] https://medium.com/tensorflow/introducing-tf-gan-a-lightweight-gan-library-for-tensorflow-2-0-36d767e1abae
[86] https://medium.com/@devnag/generative-adversarial-networks-gans-in-50-lines-of-code-pytorch-e81b79659e3f
[87] https://arxiv.org/abs/1905.08233
[88] https://arxiv.org/abs/2003.03581
[89] https://github.com/EvgenyKashin/stylegan2-distillation
[90] https://arxiv.org/abs/2001.06937
[91] https://arxiv.org/abs/1906.02691v2

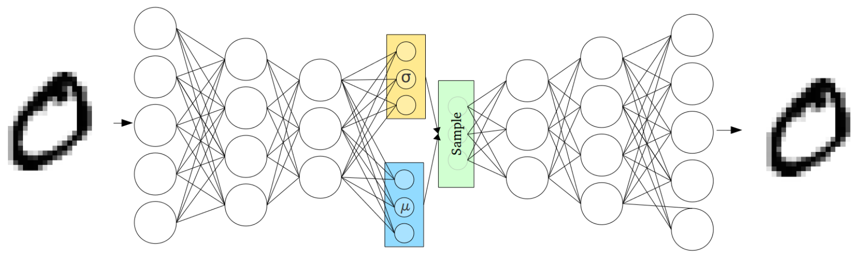



Figure 12: Variational Autoencoders (VAEs): Powerful Generative Models.

❖ Colab[92]: "*Debiasing Facial Detection Systems.*" ***AIEthics***
❖ NVAE: A Deep Hierarchical Variational Autoencoder, Arash Vahdat and Jan Kautz [93].
❖ Reading: Disentangled VAE's (DeepMind 2016) `https://arxiv.org/abs/1606.05579`.
❖ Slides: A Few Unusual Autoencoders `https://colinraffel.com/talks/vector2018few.pdf`.
❖ **MusicVAE**: Learning latent spaces for musical scores `https://magenta.tensorflow.org/music-vae`.
❖ Generative models in **Tensorflow 2** `https://github.com/timsainb/tensorflow2-generative-models/`.
❖ **SpaceSheet**: Interactive Latent Space Exploration with a Spreadsheet `https://vusd.github.io/spacesheet/`.

### 3.5.3 Capsule

**Stacked Capsule Autoencoders.** The inductive biases in this unsupervised version of capsule networks give rise to object-centric latent representations, which are learned in a self-supervised way—simply by reconstructing input images. Clustering learned representations is enough to achieve unsupervised state-of-the-art classification performance on MNIST (98.5%). Reference: blog by Adam Kosiorek.[94] Code[95].

Capsules learn *equivariant object representations* (applying any transformation to the input of the function has the same effect as applying that transformation to the output of the function).

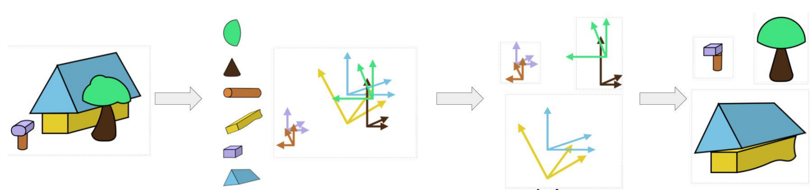



Figure 13: Stacked Capsule Autoencoders. Image source: Blog by Adam Kosiorek.

## 4 Autonomous Agents

**We are on the dawn** of *The Age of Artificial Intelligence.*

"*In a moment of technological disruption, leadership matters.*" — Andrew Ng

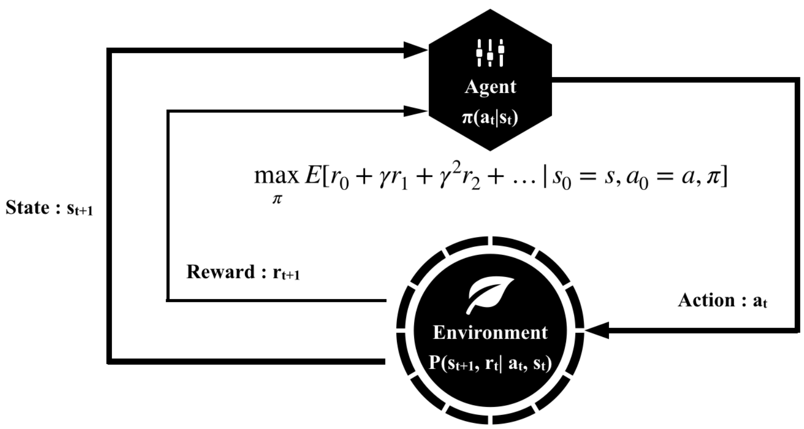
[92] `https://colab.research.google.com/github/aamini/introtodeeplearning_labs/blob/master/lab2/Part2_debiasing_solution.ipynb`
[93] `https://arxiv.org/abs/2007.03898`
[94] `http://akosiorek.github.io/ml/2019/06/23/stacked_capsule_autoencoders.html`
[95] `https://github.com/google-research/google-research/tree/master/stacked_capsule_autoencoders`

An **autonomous agent** is any device that perceives its environment and takes actions that maximize its chance of success at some goal. At the bleeding edge of AI, autonomous agents can learn from experience, simulate worlds and orchestrate meta-solutions. Here's an informal definition[96] of the *universal intelligence* of agent $\pi$[97]:

$$\Upsilon(\pi) := \sum_{\mu \in E} 2^{-K(\mu)} V_{\mu}^{\pi} \tag{6}$$

"*Intelligence measures an agent's ability to achieve goals in a wide range of environments.*" — Legg and Hutter, 2007

### 4.1 Deep Reinforcement Learning




$$\begin{aligned} s_0 &\sim \mu(s_0) \\ a_0 &\sim \pi(a_0 \mid s_0) \\ s_1, r_0 &\sim P(s_1, r_0 \mid s_0, a_0) \\ a_1 &\sim \pi(a_1 \mid s_1) \\ s_2, r_1 &\sim P(s_2, r_1 \mid s_1, a_1) \\ &\dots \\ a_{T-1} &\sim \pi(a_{T-1} \mid s_{T-1}) \\ s_T, r_{T-1} &\sim P(s_T \mid s_{T-1}, a_{T-1}) \end{aligned}$$

Figure 14: An Agent Interacts with an Environment.

Reinforcement learning (RL) studies how an agent can learn how to achieve goals in a complex, uncertain environment (Figure 14) [5]. Recent superhuman results in many difficult environments combine deep learning with RL (*Deep Reinforcement Learning*). See Figure 15 for a taxonomy of RL algorithms.

➵ Spinning Up in Deep RL - Proximal Policy Optimization (PPO), Colab Notebook[98].

❖ RL Tutorial, Behbahani et al.[99].
❖ An Opinionated Guide to ML Research[100].
❖ CS 188 : Introduction to Artificial Intelligence[101].
❖ Introduction to Reinforcement Learning by DeepMind[102].
❖ Discovering Reinforcement Learning Algorithms, Oh et al.[103].
❖ "My Top 10 Deep RL Papers of 2019" by Robert Tjarko Lange[104].
❖ Deep tic-tac-toe `https://zackakil.github.io/deep-tic-tac-toe/`.
❖ A Framework for Reinforcement Learning and Planning, Moerland et al. [105].
❖ Automatic Curriculum Learning For Deep RL: A Short Survey, Portelas et al.[106].
❖ ALLSTEPS: Curriculum-driven Learning of Stepping Stone Skills, Xie et al. [107].

[96] `https://arxiv.org/abs/0712.3329`
[97] Where $\mu$ is an environment, $K$ is the Kolmogorov complexity function, $E$ is the space of all computable reward summable environmental measures with respect to the reference machine $U$ and the value function $V_{\mu}^{\pi}$ is the agent's "ability to achieve".
[98] `https://colab.research.google.com/drive/1piaU0x7nawRpSLKOTaCEdUG0KAR2OXku`
[99] `https://github.com/eemlcommunity/PracticalSessions2020/blob/master/rl/EEML2020_RL_Tutorial.ipynb`
[100] `http://joschu.net/blog/opinionated-guide-ml-research.html`
[101] `https://inst.eecs.berkeley.edu/~cs188/fa18/`
[102] `https://www.youtube.com/watch?v=2pWv7GOvuf0&list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ`
[103] `https://arxiv.org/abs/2007.08794`
[104] `https://roberttlange.github.io/posts/2019/12/blog-post-9/`
[105] `https://arxiv.org/abs/2006.15009`
[106] `https://arxiv.org/abs/2003.04664`
[107] `https://www.cs.ubc.ca/~van/papers/2020-allsteps/`

❖ Decoupling Representation Learning from Reinforcement Learning, Stooke et al.[108] Code[109]
❖ Chip Placement with Deep Reinforcement Learning https://arxiv.org/abs/2004.10746.
❖ RL Unplugged: Benchmarks for Offline Reinforcement Learning, Gulcehre et al.[110] GitHub[111].
❖ CS 287: Advanced Robotics[112]. https://people.eecs.berkeley.edu/~pabbeel/cs287-fa19/.
❖ Combining Deep Reinforcement Learning and Search for Imperfect-Information Games, Brown et al.[113].
❖ MDP Homomorphic Networks: Group Symmetries in Reinforcement Learning, Elise van der Pol et al.[114].
❖ One Policy to Control Them All: Shared Modular Policies for Agent-Agnostic Control, Huang et al.[115]. Code[116].
❖ Decentralized Reinforcement Learning: Global Decision-Making via Local Economic Transactions, Chang et al. [117].

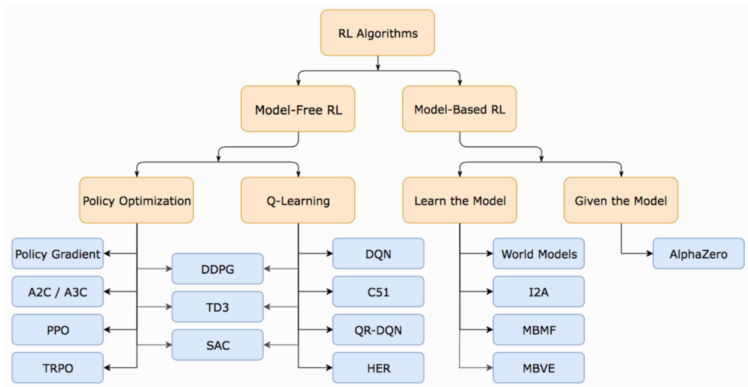



Figure 15: A Taxonomy of RL Algorithms. Source: Spinning Up in Deep RL by Achiam et al. | OpenAI

| | | | Execution | | Algorithms (discrete & continuous) | | | | | | | | | | | Additional features | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Project | Maintainer | Framework | Parallel | Distributed | DQN | Rainbow | REINFORCE | A2C | PPO | DDPG | SAC | TD3 | REINFORCE | A2C | PPO | n-step return | prioritized experience replay | distributional value funciton approximation | hyperbolic discounting | dict observations support | ? | | | | |
| OpenAI baselines | OpenAI | Tensorflow | ✓ | ✗ | ✓ | ✗ | ✗ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | | last commit june | Stars 8.1k | commit activity 1/month | code size 1.22 MB |
| stable baselines | Antonin Raffin, Ashley Hill | Tensorflow | ✓ | ✗ | ✓ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | | last commit last monday | Stars 1.1k | commit activity 6/month | code size 893 kB |
| Catalyst.RL | Sergey Kolesnikov | PyTorch | ✓? | ✓ | ✓ | ✗? | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ (any critic model) | ✗ | ✓ (any critic model) | ✓ (any critic model) | ✓ | | last commit today | Stars 718 | commit activity 19/month | code size 611 kB |
| Ray.rllib | Ray Team | Tensorflow | ✓? | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ? (DQN, DDPG only) | ✓ | ? (DQN only) | ✗ | ? | | last commit today | Stars 8k | commit activity 100/month | code size 4.92 MB |
| TF agents | Google | Tensorflow | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | | last commit yesterday | Stars 749 | commit activity 34/month | code size 2.18 MB |
| Horizon | Facebook | PyTorch | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | | last commit last monday | Stars 2k | commit activity 28/month | code size 1.04 MB |
| Coach | Intel | Tensorflow | ✓? | ✓? | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ? (DQN only) | ✓ | ? (DQN only) | ✗ | ✗ | | last commit today | Stars 1.4k | commit activity 7/month | code size 1.99 MB |
| Garage | community | Tensorflow | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | | last commit yesterday | Stars 404 | commit activity 27/month | code size 1.54 MB |
| SLM-Lab | Wah Loon Keng, Laura Graesser | PyTorch | ✗ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | | last commit last monday | Stars 548 | commit activity 147/month | code size 315 kB |
| Dopamine | Google | Tensorflow | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✗ | ✗ | | last commit july | Stars 8.1k | commit activity 3/month | code size 2.34 MB |
| OpenAi spinningup | OpenAI | Tensorflow | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | | last commit july | Stars 3.2k | commit activity 2/month | code size 218 kB |
| trfl | DeepMind | Tensorflow | ✗ | ✗ | ? | ? | | | ? | ? | ? | | | | ? | | | | | | | last commit april | Stars 2.7k | commit activity 0/month | code size 403 kB |
| scalable_agent | DeepMind | Tensorflow | ? | ✓ | ? | ? | | | ? | ? | ? | | | | ? | | | | | | | last commit march | Stars 654 | commit activity 0/month | code size 122 kB |
| ELF | Facebook | PyTorch | ? | ? | ? | ? | | | ? | ? | ? | | | | ? | | | | | | | last commit march | Stars 1.9k | commit activity 0/month | code size 964 kB |
| keras-rl | Matthias Plappert | Tensorflow | ✗ | ✗ | ✓ | ✗ | | | ✗ | ✓ | ✗ | ✗ | | | ✗ | | | | | | | last commit july | Stars 4k | commit activity 0/month | code size 191 kB |
| ikostrikov | Ilya Kostrikov | PyTorch | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | | last commit may | Stars 1.4k | commit activity 0/month | code size 95.9 kB |
| Rainbow | Kai Arulkumaran | PyTorch | ✗ | ✗ | ✓ | ✓ | | | ✗ | ✗ | ✗ | | | | ✗ | | | | | | | last commit last saturday | Stars 777 | commit activity 2/month | code size 30.4 kB |
| Vel | Jerry (?) | PyTorch | ✓ | ✗ | ✓ | ✗ | | | ✓ | ✓ | ✗ | | | | ✗ | | | | | | | last commit june | Stars 238 | commit activity 0/month | code size 668 kB |
| tensorforce | | Tensorflow | ✓ | ✓ | ✓ | ✗ | | | ✓ | ✓ | ✗ | | | | ✓? | | | | | | | last commit march | Stars 2.4k | commit activity 0/month | code size 870 kB |
| RL-Adventure | | PyTorch | ✗ | ✗ | ✓ | ✓ | | | ✗ | ✓ | ✓ | | | | ✗ | | | | | | | last commit april 2018 | Stars 1.6k | commit activity 0/month | code size 1.07 MB |
| DeepRL-Tutorials | | PyTorch | ✗ | ✗ | ✓ | ✓ | | | ✗ | ✗ | ✗ | | | | ✓ | | | | | | | last commit march | Stars 416 | commit activity 0/month | code size 4.15 MB |
| surreal | | TorchX | ✓ | ✓ | ✗ | ✗ | | | ✗ | ✓ | ✗ | | | | ✓ | | | | | | | last commit april | Stars 336 | commit activity 0/month | code size 467 kB |
| lagom | | PyTorch | ✓ | ✗ | ✗ | ✗ | | | ✗ | ✗ | ✗ | | | | ✗ | | | | | | | last commit july | Stars 342 | commit activity 12/month | code size 2.24 MB |
| dennybritz | | Tensorflow | ? | ? | ? | ? | | | ? | ? | ? | | | | ? | | | | | | | last commit june | Stars 11k | commit activity 0/month | code size 2.28 MB |
| scitator | | Tensorflow | ? | ? | ✓ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | | last commit june 2017 | Stars 71 | commit activity 0/month | code size 1.39 MB |
| pymarl | WhiRL | PyTorch | | | MARL | | | | | | | | | | | | | | | | | last commit july | Stars 251 | commit activity 2/month | code size 73.9 kB |

Figure 16: **Open-Source RL Algorithms** https://docs.google.com/spreadsheets/d/1EeFPd-XIQ3mq_9snTlAZSsFY7Hbnmd7P5bbT8LPuMn0/

[108] https://arxiv.org/abs/2009.08319
[109] https://github.com/astooke/rlpyt/tree/master/rlpyt/ul
[110] https://arxiv.org/abs/2006.13888
[111] https://github.com/deepmind/deepmind-research/tree/master/rl_unplugged
[112] https://people.eecs.berkeley.edu/~pabbeel/cs287-fa19/exam/cs287-fa19-exam-study-handout.pdf
[113] https://arxiv.org/abs/2007.13544
[114] https://arxiv.org/abs/2006.16908
[115] https://arxiv.org/abs/2007.04976
[116] https://huangwl18.github.io/modular-rl
[117] https://arxiv.org/abs/2007.02382

#### 4.1.1 Model-Free RL | Value-Based

The goal in RL is to train the agent to maximize the discounted sum of all future rewards $\boldsymbol{R_t}$, called the return:

$$R_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots \tag{7}$$

The Q-function captures the expected total future reward an agent in state $\boldsymbol{s}$ can receive by executing a certain action $\boldsymbol{a}$:

$$Q(s, a) = E[R_t] \tag{8}$$

The optimal policy should choose the action $\boldsymbol{a}$ that maximizes $\boldsymbol{Q(s,a)}$:

$$\pi^*(s) = argmax_a Q(s, a) \tag{9}$$

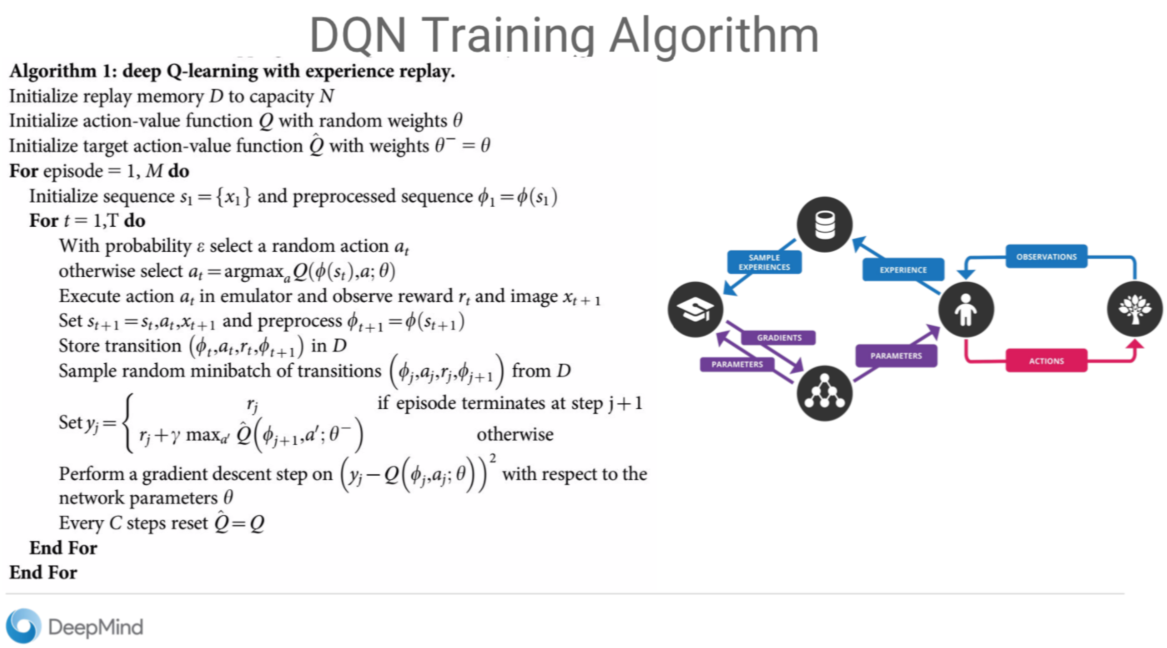



Figure 17: **DQN Training Algorithm.** Volodymyr Mnih, Deep RL Bootcamp

- **Q-Learning**: *Playing Atari with Deep Reinforcement Learning* (DQN). Mnih et al, 2013[10]. See Figure 17.

"*There's no limit to intelligence.*" — David Silver

❖ Q-Learning in enormous action spaces via amortized approximate maximization, de Wiele et al.[118].
❖ TF-Agents (DQN Tutorial) | Colab `https://colab.research.google.com/github/tensorflow/agents`.

#### 4.1.2 Model-Free RL | Policy-Based

An RL agent learns the stochastic policy function that maps state to action and act by sampling policy.

Run a policy for a while (code: `https://gist.github.com/karpathy/a4166c7fe253700972fcbc77e4ea32c5`):

$$\tau = (s_0, a_0, r_0, s_1, a_1, r_1, \dots, s_{T-1}, a_{T-1}, r_{T-1}, s_T) \tag{10}$$

[118] `https://arxiv.org/abs/2001.08116`

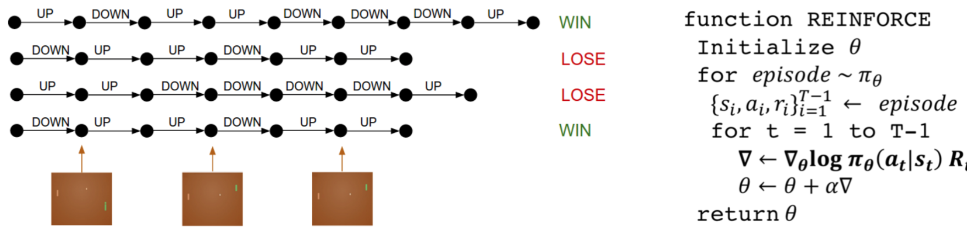



Figure 18: Policy Gradient Directly Optimizes the Policy.

Increase probability of actions that lead to high rewards and decrease probability of actions that lead to low rewards:

$$\nabla_\theta E_\tau[R(\tau)] = E_\tau \left[ \sum_{t=0}^{T-1} \nabla_\theta \log \pi(a_t|s_t, \theta) R(\tau) \right] \tag{11}$$

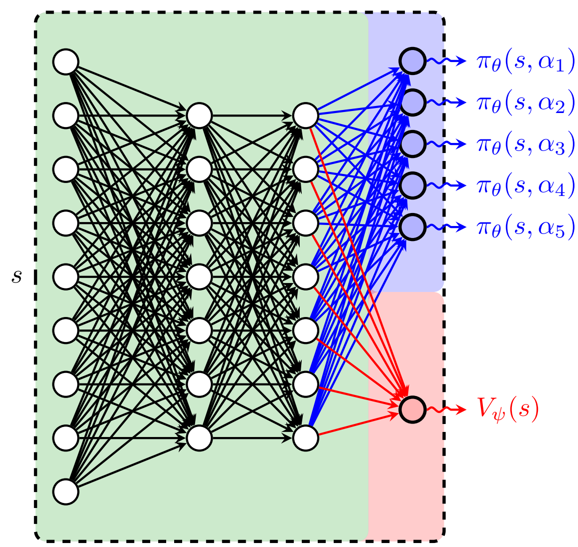



Figure 19: **Asynchronous Advantage Actor-Critic (A3C)**. Source: Petar Velickovic

- **Policy Optimization**: *Asynchronous Methods for Deep Reinforcement Learning* (A3C). Mnih et al, 2016[8].
- **Policy Optimization**: *Proximal Policy Optimization Algorithms* (PPO). Schulman et al, 2017[9].

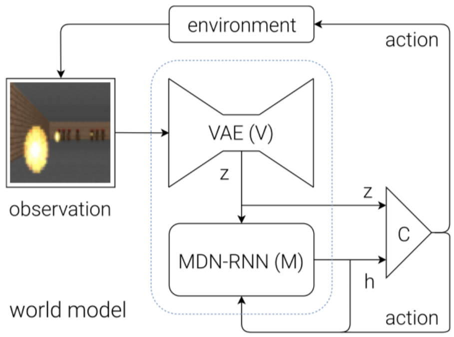
❖ Phasic Policy Gradient. Cobbe et al.[119] Code[120].
❖ Deep Reinforcement Learning for Playing 2.5D Fighting Games. Li et al.[121].
❖ rlpyt: A Research Code Base for Deep Reinforcement Learning in PyTorch. Adam Stooke, Pieter Abbee[122].

### 4.1.3 Model-Based RL

In Model-Based RL, the agent generates predictions about the next state and reward before choosing each action.

- **Learn the Model**: *Recurrent World Models Facilitate Policy Evolution* (World Models[123]). The world model agent can be trained in an unsupervised manner to learn a compressed spatial and temporal representation of the environment. Then, a compact policy can be trained. See Figure 20. Ha et al, 2018[11].

[119] https://arxiv.org/abs/2009.04416
[120] https://github.com/openai/phasic-policy-gradient
[121] https://arxiv.org/abs/1805.02070
[122] https://arxiv.org/abs/1909.01500
[123] https://worldmodels.github.io




```
def rollout(controller):
  ''' env, rnn, vae are '''
  ''' global variables  '''
  obs = env.reset()
  h = rnn.initial_state()
  done = False
  cumulative_reward = 0
  while not done:
    z = vae.encode(obs)
    a = controller.action([z, h])
    obs, reward, done = env.step(a)
    cumulative_reward += reward
    h = rnn.forward([a, z, h])
  return cumulative_reward
```

Figure 20: World Model's Agent consists of: Vision (V), Memory (M), and Controller (C). | Ha et al, 2018[11]

- **Learn the Model**: *Learning Latent Dynamics for Planning from Pixels* `https://planetrl.github.io/`.
- **Given the Model**: *Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm* (AlphaZero). Silver et al, 2017[14]. AlphaGo Zero Explained In One Diagram[124].

❖ Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model. Schrittwieser et al.[125]. Pseudocode[126].

#### 4.1.4 Toward a General AI-Agent Architecture: SuperDyna (*General Dyna-style RL Agent*)

"*Intelligence is the computational part of the ability to predict and control a stream of experience.*" — Rich Sutton

**SuperDyna.**[127] The ambition: a general AI agent for Artificial Biological Reinforcement Learning.

1. Interact with the world: sense, update state and take an action
2. Learn from what just happened: see what happened and learn from it
3. Plan: (while there is time remaining in this time step) imagine hypothetical states and actions you might take
4. Discover : curate options and features and measure how well they're doing

The first complete and scalable general AI-agent architecture that has all the most important capabilities and desiderata:

- Acting, learning, planning, model-learning, subproblems, and options.
- Function approximation, partial observability, non-stationarity and stochasticity.
- Discovery of state features, and thereby of subproblems, options and models.
- All feeding back to motivate new, more-abstract features in a virtuous cycle of discovery.

Presentation by Richard Sutton (starts at 15 min.)[128].

"*In practice, I work primarily in reinforcement learning as an approach to artificial intelligence. I am exploring ways to represent a broad range of human knowledge in an empirical form–that is, in a form directly in terms of experience–and in ways of reducing the dependence on manual encoding of world state and knowledge.*" — Richard S. Sutton

#### 4.1.5 Improving Agent Design

Via Reinforcement Learning: Blog[129]. arXiv[130]. ASTool `https://github.com/hardmaru/astool/`.
Via Evolution: Video[131]. Evolved Creatures `http://www.karlsims.com/evolved-virtual-creatures.html`.

[124] `https://applied-data.science/static/main/res/alpha_go_zero_cheat_sheet.png`
[125] `https://arxiv.org/abs/1911.08265`
[126] `https://arxiv.org/src/1911.08265v2/anc/pseudocode.py`
[127] `https://insidehpc.com/2020/02/video-toward-a-general-ai-agent-architecture/`
[128] `https://slideslive.com/38921889/biological-and-artificial-reinforcement-learning-4`
[129] `https://designrl.github.io`
[130] `https://arxiv.org/abs/1810.03779`
[131] `https://youtu.be/JBgG_VSP7f8`

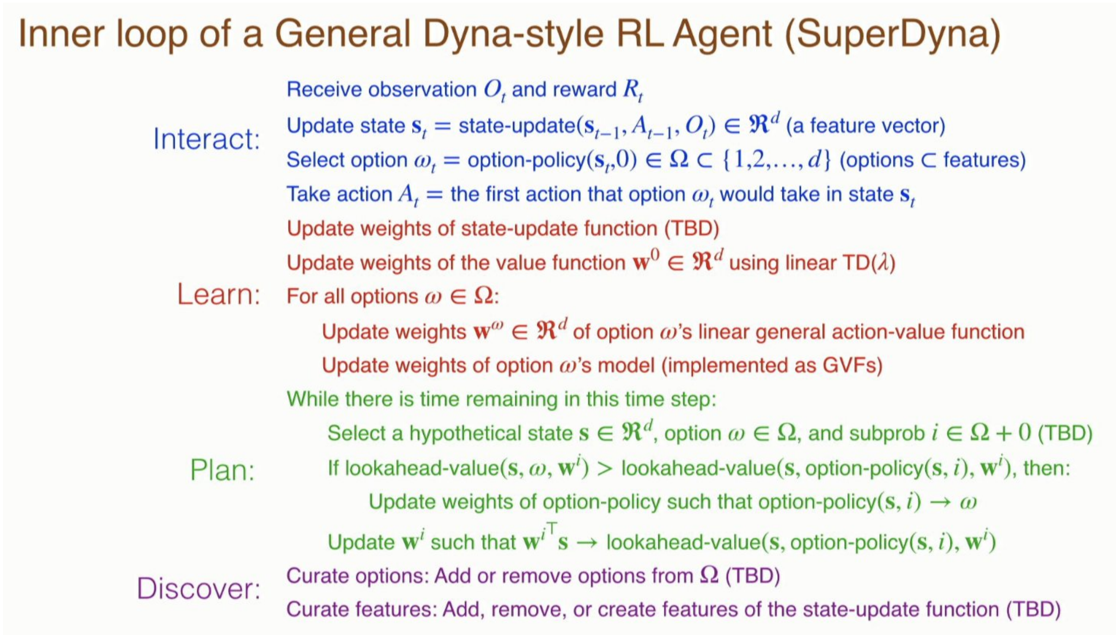



Figure 21: Inner Loop of a General Dyna-Style RL Agent (**SuperDyna**).

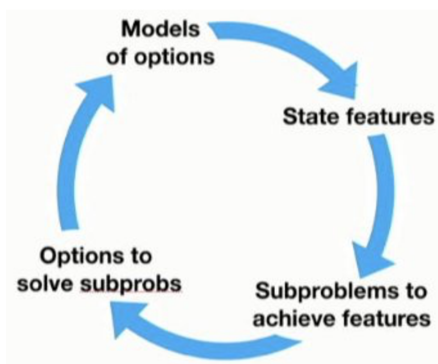



Figure 22: **SuperDyna:** Virtuous cycle of discovery.

"*The future of high-level APIs for AI is... a problem-specification API. Currently we only search over network weights, thus "problem specification" involves specifying a model architecture. In the future, it will just be: "tell me what data you have and what you are optimizing"*." — François Chollet

❖ Teacher algorithms for curriculum learning of Deep RL in continuously parameterized environments[132].

### 4.1.6 OpenAI Baselines

High-quality implementations of reinforcement learning algorithms `https://github.com/openai/baselines`.

➺ Colab Notebook `https://colab.research.google.com/drive/1amdIQaHWyc8Av_DoM5yFYHyYvyqD5BZX`.

### 4.1.7 Google Dopamine and A Zoo of Agents

Dopamine is a research framework for fast prototyping of reinforcement learning algorithms.[133].

[132] `https://arxiv.org/abs/1910.07224`
[133] `https://github.com/google/dopamine`

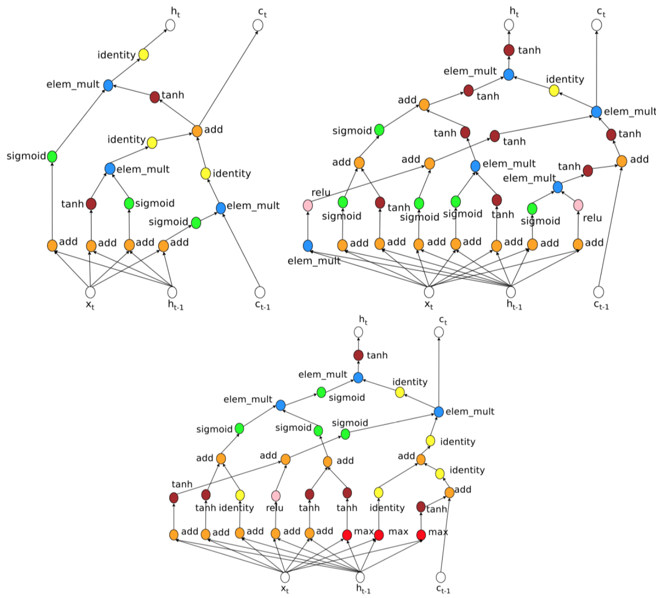



Figure 23: A comparison of the original LSTM cell vs. two new good generated. Top left: LSTM cell. [19]

A Zoo of Atari-Playing Agents: Code[134], Blog[135] and Colaboratory notebook `https://colab.research.google.com/github/uber-research/atari-model-zoo/blob/master/colab/AtariZooColabDemo.ipynb`.

#### 4.1.8 TRFL : TensorFlow Reinforcement Learning

TRFL ("truffle"): a library of reinforcement learning building blocks `https://github.com/deepmind/trfl`.

#### 4.1.9 *bsuite* : Behaviour Suite for Reinforcement Learning

A collection of experiments that investigate core capabilities of RL agents `http://github.com/deepmind/bsuite`.

### 4.2 Evolution Strategies (ES)

In her Nobel Prize in Chemistry 2018 Lecture "*Innovation by Evolution: Bringing New Chemistry to Life*" (Nobel Lecture)†[136], Prof. Frances H. Arnold said :

"*Nature ... invented life that has flourished for billions of years. (...) Equally awe-inspiring is the process by which Nature created these enzyme catalysts and in fact everything else in the biological world. The process is evolution, the grand diversity-generating machine that created all life on earth, starting more than three billion years ago. (...)* ***evolution executes a simple algorithm of diversification and natural selection, an algorithm that works at all levels of complexity from single protein molecules to whole ecosystems.***" — Prof. Frances H. Arnold

**⇝ Demo: ES on LunarLanderContinuous-v2. Colab Notebook[137]. Python Code[138]**

Evolution and neural networks proved a potent combination in nature.

"*Evolution is a slow learning algorithm that with the sufficient amount of compute produces a human brain.*" — Wojciech Zaremba

Natural evolutionary strategy **directly evolves the weights of a DNN** and performs competitively with the best deep reinforcement learning algorithms, including deep Q-networks (DQN) and policy gradient methods (A3C)[21].

[134] `https://github.com/uber-research/atari-model-zoo`
[135] `https://eng.uber.com/atari-zoo-deep-reinforcement-learning/`
[136] `https://onlinelibrary.wiley.com/doi/epdf/10.1002/anie.201907729`
[137] `https://colab.research.google.com/drive/1PpYYaihoJWiszZh1vhKXvmN2X9KnLA7i`
[138] `https://drive.google.com/file/d/1YlKNorK21GMffz-29omEB7q_iLYRlmXD/view?usp=sharing`

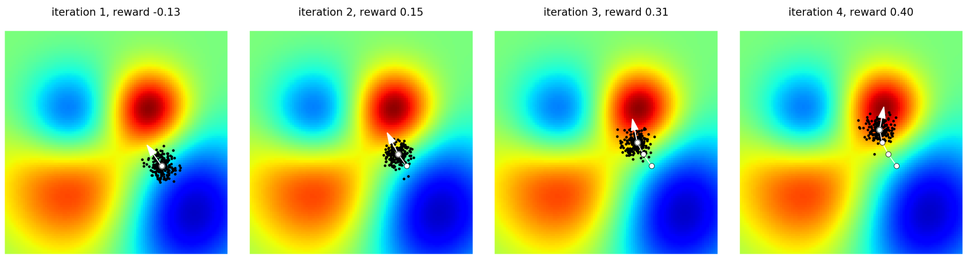



Figure 24: https://colab.research.google.com/github/karpathy/randomfun/blob/master/es.ipynb.

Neuroevolution, which harnesses evolutionary algorithms to optimize neural networks, enables capabilities that are typically unavailable to gradient-based approaches, including learning neural network building blocks, architectures and even the algorithms for learning[12].

> "*. . . evolution — whether biological or computational — is inherently creative, and should routinely be expected to surprise, delight, and even outwit us.*" — The Surprising Creativity of Digital Evolution, Lehman et al.[22]

The ES algorithm is a "guess and check" process, where we start with some random parameters and then repeatedly:

1. Tweak the guess a bit randomly, and
2. Move our guess slightly towards whatever tweaks worked better.

Neural architecture search has advanced to the point where it can outperform human-designed models[13].

"*Caterpillar brains LIQUIFY during metamorphosis, but the butterfly retains the caterpillar's memories!*" — M. Levin

"Open-ended" algorithms are algorithms that endlessly create. Brains and bodies evolve together in nature.

> *"We're machines,"* says Hinton. "*"We're just produced biologically (...)"* — Katrina Onstad, Toronto Life

❖ Evolution Strategies[139].
❖ VAE+CPPN+GAN[140].
❖ Demo: ES on CartPole-v1[141].
❖ AutoML-Zero: Evolving Machine Learning Algorithms From Scratch, Real et al.[142] Code[143].
❖ Spiders Can Fly Hundreds of Miles Riding the Earth's Magnetic Fields[144].
❖ A Visual Guide to ES http://blog.otoro.net/2017/10/29/visual-evolution-strategies/.
❖ **Xenobots** A scalable pipeline for designing reconfigurable organisms, Kriegman et al.[145]. Learn[146]. Evolve[147].

### 4.3 Self Play

Silver et al.[15] introduced an algorithm based solely on reinforcement learning, without human data, guidance or domain knowledge. Starting tabula rasa (and being its own teacher!), AlphaGo Zero achieved superhuman performance. AlphaGo Zero showed that algorithms matter much more than big data and massive amounts of computation.

"*Self-Play is Automated Knowledge Creation.*" — Carlos E. Perez

Self-play mirrors similar insights from coevolution. Transfer learning is the key to go from self-play to the real world[148].

[139] https://lilianweng.github.io/lil-log/2019/09/05/evolution-strategies.html
[140] https://colab.research.google.com/drive/1_OoZ3z_C5Jl5gnxDOE9VEMCTs-Fl8pvM
[141] https://colab.research.google.com/drive/1bMZWHdhm-mT9NJENWoVewUks7cGV10go
[142] https://arxiv.org/abs/2003.03384
[143] https://github.com/google-research/google-research/tree/master/automl_zero
[144] https://www.cell.com/current-biology/fulltext/S0960-9822(18)30693-6
[145] https://www.pnas.org/content/early/2020/01/07/1910837117
[146] https://cdorgs.github.io
[147] https://github.com/skriegman/reconfigurable_organisms
[148] http://metalearning-symposium.ml

"*Open-ended self play produces: Theory of mind, negotiation, social skills, empathy, real language understanding.*" — Ilya Sutskever, Meta Learning and Self Play

❖ How To Build Your Own MuZero AI Using Python[149].
❖ AlphaGo - The Movie | Full Documentary `https://youtu.be/WXuK6gekU1Y`.
❖ TensorFlow.js Implementation of DeepMind's AlphaZero Algorithm for Chess. Live Demo[150]. | Code[151].
❖ An open-source implementation of the AlphaGoZero algorithm `https://github.com/tensorflow/minigo`.
❖ ELF OpenGo: An Open Reimplementation of AlphaZero, Tian et al.: `https://arxiv.org/abs/1902.04522`.

### 4.4 Multi-Agent Populations

"*We design a Theory of Mind neural network – a ToMnet – which uses meta-learning to build models of the agents it encounters, from observations of their behaviour alone.*" — Machine Theory of Mind, Rabinowitz et al.[25]

**Cooperative Agents.** Learning to Model Other Minds, by OpenAI[24], is an algorithm which accounts for the fact that other agents are learning too, and discovers self-interested yet collaborative strategies. Also: OpenAI Five[152].

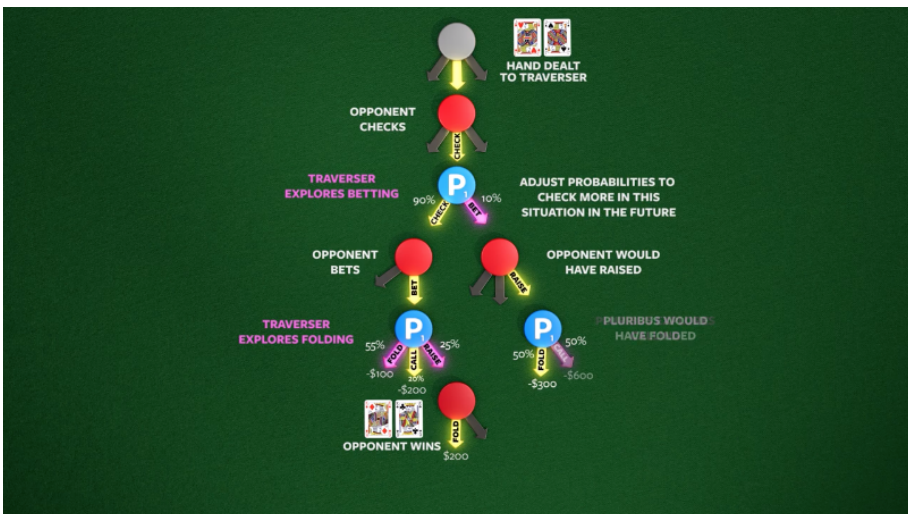



Figure 25: **Facebook, Carnegie Mellon build first AI that beats pros in 6-player poker** `https://ai.facebook.com/blog/pluribus-first-ai-to-beat-pros-in-6-player-poker`

"*Artificial Intelligence is about recognising patterns, Artificial Life is about creating patterns.*" — Mizuki Oka et al.

**Active Learning Without Teacher.** In *Intrinsic Social Motivation via Causal Influence in Multi-Agent RL*, Jaques et al. (2018) `https://arxiv.org/abs/1810.08647` propose an intrinsic reward function designed for multi-agent RL (MARL), which awards agents for having a causal influence on other agents' actions. Open-source implementation [153].

"*Open-ended Learning in Symmetric Zero-sum Games,*" Balduzzi et al.: `https://arxiv.org/abs/1901.08106`

❖ Lenia and Expanded Universe, Bert Wang-Chak Chan `https://arxiv.org/abs/2005.03742`.
❖ Neural MMO v1.3: A Massively Multiagent Game Environment for Training and Evaluating Neural Networks,

[149] `https://medium.com/applied-data-science/how-to-build-your-own-muzero-in-python-f77d5718061a`
[150] `https://frpays.github.io/lc0-js/engine.html`
[151] `https://github.com/frpays/lc0-js/`
[152] `https://blog.openai.com/openai-five/`
[153] `https://github.com/eugenevinitsky/sequential_social_dilemma_games`

Suarezet al.[154] Project Page https://jsuarez5341.github.io, Video[155] and Slides[156].
❖ Neural MMO: A massively multiagent env. for simulations with many long-lived agents. Code[157] and 3D Client[158].

### 4.5 Deep Meta-Learning

Learning to Learn[16].

> "*The notion of a neural "architecture" is going to disappear thanks to meta learning.*" — Andrew Trask

❖ Stanford CS330: Multi-Task and Meta-Learning, Finn et al., 2019[159].
❖ Meta Learning Shared Hierarchies[18] (*The Lead Author is in High School!*).
❖ Causal Reasoning from Meta-reinforcement Learning https://arxiv.org/abs/1901.08162.
❖ Meta-Learning through Hebbian Plasticity in Random Networks, Elias Najarro and Sebastian Risi, 2020[160].
❖ Meta-Learning Symmetries by Reparameterization, Zhou et al., 2020 https://arxiv.org/abs/2007.02933.

#### 4.5.1 *MAML*: Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks

The goal of *model-agnostic meta-learning for fast adaptation of deep networks* is to train a model on a variety of learning tasks, such that it can solve new learning tasks using only a small number of training samples[20].

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}\left(f_{\theta_i'}\right) \tag{12}$$

A meta-learning algorithm takes in a distribution of tasks, where each task is a learning problem, and it produces a quick learner — a learner that can generalize from a small number of examples[17].

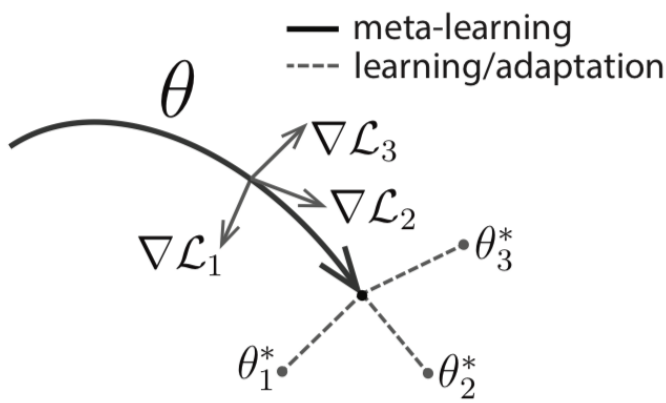



Figure 26: Diagram of Model-Agnostic Meta-Learning (MAML)

❖ How to Train MAML (Model-Agnostic Meta-Learning)[161].
❖ Meta-Learning with Implicit Gradients https://arxiv.org/abs/1909.04630.
❖ Colaboratory reimplementation of MAML (Model-Agnostic Meta-Learning) in TF 2.0[162].
❖ Torchmeta: A Meta-Learning library for PyTorch[163] https://github.com/tristandeleu/pytorch-meta.

[154] https://arxiv.org/abs/2001.12004
[155] https://youtube.com/watch?v=DkHopV1RSxw
[156] https://docs.google.com/presentation/d/1tqm_Do9ph-duqqAlx3r9lI5Nbfb9yUfNEtXk1Qo4zSw/edit?usp=sharing
[157] https://github.com/openai/neural-mmo
[158] https://github.com/jsuarez5341/neural-mmo-client
[159] http://youtube.com/playlist?list=PLoROMvodv4rMC6zfYmnD7UG3LVvwaITY5
[160] https://arxiv.org/abs/2007.02686
[161] https://medium.com/towards-artificial-intelligence/how-to-train-maml-model-agnostic-meta-learning-90aa093f8e46
[162] https://colab.research.google.com/github/mari-linhares/tensorflow-maml/blob/master/maml.ipynb
[163] https://medium.com/pytorch/torchmeta-a-meta-learning-library-for-pytorch-f76c2b07ca6d

#### 4.5.2 The Grand Challenge for AI Research | *AI-GAs*: AI-Generating Algorithms, an Alternate Paradigm for Producing General Artificial Intelligence

In *AI-GAs: AI-generating algorithms, an alternate paradigm for producing general artificial intelligence*[164], Jeff Clune describes an exciting path that ultimately may be successful at producing general AI. The idea is to create an AI-generating algorithm (AI-GA), which automatically learns how to produce general AI.

Three Pillars are essential for the approach: (1) **Meta-learning architectures**, (2) **Meta-learning the learning algorithms themselves**, and (3) **Generating effective learning environments**.

- **The First Pillar**, meta-learning architectures, could potentially discover the building blocks : *convolution, recurrent layers, gradient-friendly architectures, spatial tranformers, etc*.
- **The Second Pillar**, meta-learning learning algorithms, could potentially learn the building blocks : *intelligent exploration, auxiliary tasks, efficient continual learning, causal reasoning, active learning, etc*.
- **The Third Pillar**, generating effective and fully expressive learning environments, could learn things like : *co-evolution / self-play, curriculum learning, communication / language, multi-agent interaction, etc*.

On Earth,

> "*( . . . ) a remarkably simple algorithm (Darwinian evolution) began producing solutions to relatively simple environments. The 'solutions' to those environments were organisms that could survive in them. Those organism often created new niches (i.e. environments, or opportunities) that could be exploited. Ultimately, that process produced all of the engineering marvels on the planet, such as jaguars, hawks, and the human mind.*" — Jeff Clune

**Turing Complete** (universal computer) : an encoding that enables the creation any possible learning algorithm.
**Darwin Complete** : an environmental encoding that enables the creation of any possible learning environment.

❖ **Learning to Continually Learn**. Beaulieu et al. `https://arxiv.org/abs/2002.09571`. Code[165].
❖ **Fully Differentiable Procedural Content Generation through Generative Playing Networks**. Bontrageret et al.[166]

# 5 Symbolic AI

❖ **Neural Module Networks for Reasoning over Text**. Gupta et al.[167] Code.[168]
❖ **Neural-Symbolic Learning and Reasoning: A Survey and Interpretation**. Besold et al.[169]
❖ On neural-symbolic computing: suggested readings on foundations of the field. Luis Lamb[170].
❖ **Neuro-symbolic A.I. is the future of artificial intelligence. Here's how it works**. Luke Dormehl[171].
❖ **DDSP: Differentiable Digital Signal Processing**. Engel et al. Blog[172], Colab[173], Paper[174] and Code[175].
❖ **The compositionality of neural networks: integrating symbolism and connectionism**. Hupkes et al.[176]
❖ **Graph Neural Networks Meet Neural-Symbolic Computing: A Survey and Perspective**. Lamb et al.[177]
❖ Discovering Symbolic Models from Deep Learning with Inductive Biases, Cranmer et al.[178]. Blog and code[179].
❖ **Differentiable Reasoning on Large Knowledge Bases and Natural Language**. Minervini et al.[180] Open-source

[164] `https://arxiv.org/abs/1905.10985`
[165] `https://github.com/uvm-neurobotics-lab/ANML`
[166] `https://arxiv.org/abs/2002.05259`
[167] `https://arxiv.org/abs/1912.04971`
[168] `https://nitishgupta.github.io/nmn-drop`
[169] `https://arxiv.org/abs/1711.03902`
[170] `https://twitter.com/luislamb/status/1218575842340634626`
[171] `https://www.digitaltrends.com/cool-tech/neuro-symbolic-ai-the-future/`
[172] `http://magenta.tensorflow.org/ddsp`
[173] `http://g.co/magenta/ddsp-demo`
[174] `http://g.co/magenta/ddsp-paper`
[175] `http://github.com/magenta/ddsp`
[176] `https://arxiv.org/abs/1908.08351`
[177] `https://arxiv.org/abs/2003.00330`
[178] `https://arxiv.org/abs/2006.11287`
[179] `https://astroautomata.com/paper/symbolic-neural-nets/`
[180] `https://arxiv.org/abs/1912.10824`

neuro-symbolic reasoning framework, in TensorFlow https://github.com/uclnlp/gntp.

# 6 Environments

Platforms for training autonomous agents.

"*Run a physics sim long enough and you'll get intelligence.*" — Elon Musk

## 6.1 OpenAI Gym

"*Situation awareness is the perception of the elements in the environment within a volume of time and space, and the comprehension of their meaning, and the projection of their status in the near future.*" — Endsley (1987)

The OpenAI Gym https://gym.openai.com/ (Blog[181] | GitHub[182]) is a toolkit for developing and comparing reinforcement learning algorithms. What makes the gym so great is a common API around environments.

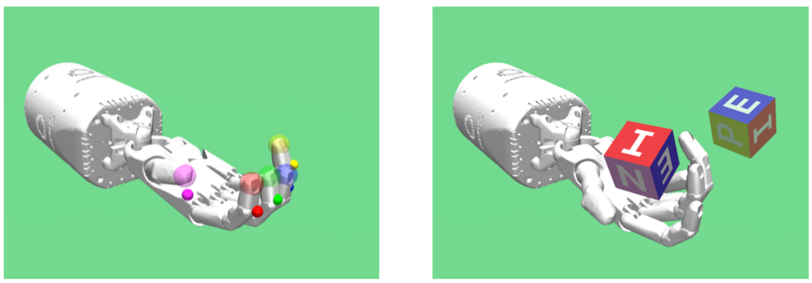


Figure 27: Robotics Environments https://blog.openai.com/ingredients-for-robotics-research/

"*By framing the approach within the popular OpenAI Gym framework, design firms can create more realistic environments – for instance, incorporate strength of materials, safety factors, malfunctioning of components under stressed conditions, and plug existing algorithms into this framework to optimize also for design aspects such as energy usage, easy-of-manufacturing, or durability.*" — David Ha[183]

**⇝ Getting Started with the OpenAI Gym, Colab Notebook**[184]

How to create new environments for Gym[185]. **Minimal example with code and agent** *(evolution strategies on foo-v0):*

1. Download *gym-foo* https://drive.google.com/file/d/1r2A8J9CJjIQNwss246gATeDOLLMzpUT-/view?usp=sharing
2. cd gym-foo
3. pip install -e .
4. python ES-foo.py

He're another more difficult (*for the agent!*) new environment for Gym *(evolution strategies on foo-v3):*

1. Download *gym-foo-v3*[186]
2. cd gym-foo-v3
3. pip install -e .
4. python ES-foo-v3.py

[181] https://blog.openai.com/openai-gym-beta/
[182] https://github.com/openai/gym
[183] https://designrl.github.io
[184] https://colab.research.google.com/drive/1fBDH7xfpwH9SKj5J9TAH9XOTGJF61vJZ
[185] https://github.com/openai/gym/blob/master/docs/creating-environments.md
[186] https://drive.google.com/file/d/1cGncsXJ56UUKCO9MaRWJVTnxiQEnLuxS/view?usp=sharing

➾ **Create a New Environment (foo) from Scratch, Colab Notebook**[187]

❖ OpenAI Gym Environment for Trading[188].
❖ Fantasy Football AI Environment `https://github.com/njustesen/ffai`.
❖ Create custom gym environments from scratch — A stock market example[189].
❖ Spot Mini Mini OpenAI Gym Environment. Maurice Rahme, blog[190] et code[191].
❖ IKEA Furniture Assembly Environment `https://clvrai.github.io/furniture/`.
❖ Minimalistic Gridworld Environment `https://github.com/maximecb/gym-minigrid`.
❖ **DoorGym**: A Scalable Door Opening Environment and Baseline Agent, Urakami et al., 2019[192].
❖ **gym-gazebo2**, a toolkit for reinforcement learning using ROS 2 and Gazebo, Lopez et al., 2019[193].
❖ OFFWORLD GYM Open-access physical robotics environment for real-world reinforcement learning[194].
❖ Safety Gym: environments to evaluate agents with safety constraints `https://github.com/openai/safety-gym`.
❖ **TensorTrade:** An open source reinforcement learning framework for training, evaluating, and deploying robust trading agents `https://github.com/tensortrade-org/tensortrade`.

### 6.2 DeepMind Lab

**DeepMind Lab:** A customisable 3D platform for agent-based AI research `https://github.com/deepmind/lab`.

- DeepMind Control Suite `https://github.com/deepmind/dm_control`.
- Convert DeepMind Control Suite to OpenAI Gym Envs `https://github.com/zuoxingdong/dm2gym`.

### 6.3 Unity ML-Agents

**Unity ML Agents** allows to create environments where intelligent agents (*Single Agent*, *Cooperative and Competitive Multi-Agent* and *Ecosystem*) can be trained using RL, neuroevolution, or other ML methods `https://unity3d.ai`.

- Announcing ML-Agents Unity Package v1.0! Mattar et al.[195].
- Getting Started with Marathon Environments for Unity ML-Agents[196] `https://github.com/Unity-Technologies/marathon-envs`.
- Arena: A General Evaluation Platform and Building Toolkit for Multi-Agent Intelligence[197].

### 6.4 AI Habitat

**AI Habitat** enables training of embodied AI agents (virtual robots) in a highly photorealistic and efficient 3D simulator, before transferring the learned skills to reality. By Facebook AI Research `https://aihabitat.org/`.

Why the name Habitat? Because that's where AI agents live!

### 6.5 POET: Paired Open-Ended Trailblazer

Diversity is the premier product of evolution. Endlessly generate increasingly complex and diverse learning environments[198]. Open-endedness could generate learning algorithms reaching human-level intelligence[23].

- Implementation of the POET algorithm `https://github.com/uber-research/poet`.

[187] `https://colab.research.google.com/drive/1hXW5hQn1MO4kjgc2W2wjyTwDcId5QGCD`
[188] `https://github.com/hackthemarket/gym-trading`
[189] `https://towardsdatascience.com/creating-a-custom-openai-gym-environment-for-stock-trading-be532be3910e`
[190] `https://moribots.github.io/project/spot-mini-mini`
[191] `https://github.com/moribots/spot_mini_mini`
[192] `https://arxiv.org/abs/1908.01887`
[193] `https://arxiv.org/abs/1903.06278`
[194] `https://gym.offworld.ai`
[195] `https://blogs.unity3d.com/2020/05/12/announcing-ml-agents-unity-package-v1-0/`
[196] `https://towardsdatascience.com/gettingstartedwithmarathonenvs-v0-5-0a-c1054a0b540c`
[197] `https://arxiv.org/abs/1905.08085`
[198] `https://eng.uber.com/poet-open-ended-deep-learning/`

- Enhanced POET: Open-Ended Reinforcement Learning through Unbounded Invention of Learning Challenges and their Solutions. Wang et al., 2020 https://arxiv.org/abs/2003.08536. Code[199].

# 7 Deep-Learning Hardware

Figure 28: Edge TPU - Dev Board https://coral.ai/products/dev-board/

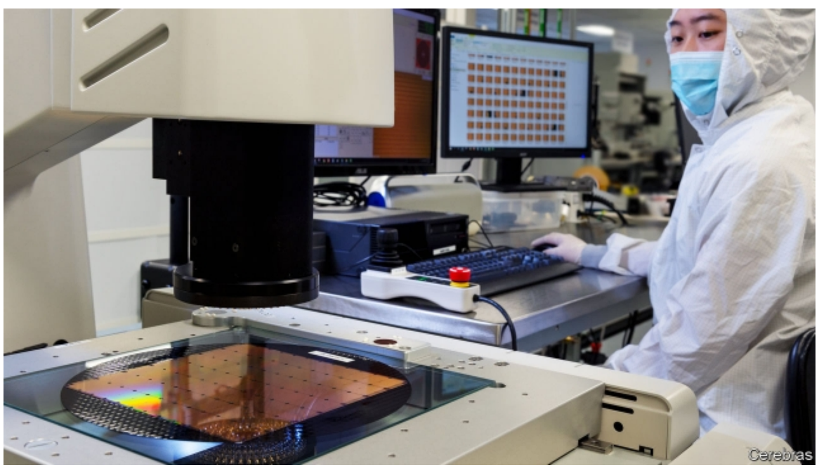



Figure 29: The world's largest chip : Cerebras Wafer Scale Engine https://www.cerebras.net

❖ Which GPU(s) to Get for Deep Learning, by Tim Dettmers[200].
❖ A Full Hardware Guide to Deep Learning, by Tim Dettmers[201].
❖ Jetson Nano. A small but mighty AI computer to create intelligent systems[202].
❖ Build AI that works offline with Coral Dev Board, Edge TPU, and TensorFlow Lite, by Daniel Situnayake[203].

[199] http://github.com/uber-research/poet
[200] http://timdettmers.com/2019/04/03/which-gpu-for-deep-learning/
[201] http://timdettmers.com/2018/12/16/deep-learning-hardware-guide/
[202] https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-nano/
[203] https://medium.com/tensorflow/build-ai-that-works-offline-with-coral-dev-board-edge-tpu-and-tensorflow-lite-70

# 8 Deep-Learning Software

## 8.1 TensorFlow

**TensorFlow Hub** is a library for reusable ML modules https://www.tensorflow.org/hub. Tutorials[204].
**TensorFlow.js** allows machine learning to happen within the web browser https://www.tensorflow.org/js/.

- TF-Coder https://goo.gle/3gwTbB6.
- TensorFlow Lite for Microcontrollers[205].
- Intro to Keras for Researchers. Colab[206].
- Introduction to Keras for Engineers. Colab[207].
- TensorBoard in Jupyter Notebooks[208]. Colab[209].
- TensorFlow 2.0 + Keras Crash Course. Colab[210].
- tf.keras (TensorFlow 2.0) for Researchers: Crash Course. Colab[211].
- TensorFlow Tutorials https://www.tensorflow.org/tutorials.
- Exploring helpful uses for BERT in your browser with TensorFlow.js[212].
- TensorFlow 2.0: basic ops, gradients, data preprocessing and augmentation, training and saving. Colab[213].

## 8.2 PyTorch

- PyTorch primer. Colab[214].
- Get started with PyTorch, Cloud TPUs, and Colab[215].
- Effective PyTorch https://github.com/vahidk/EffectivePyTorch
- PyTorch internals http://blog.ezyang.com/2019/05/pytorch-internals/
- PyTorch Lightning Bolts https://github.com/PyTorchLightning/pytorch-lightning-bolts

# 9 AI Art | A New Day Has Come in Art Industry

The code (*art-DCGAN*) for the first artificial intelligence artwork ever sold at Christie's auction house (Figure 23) is a modified implementation of DCGAN focused on generative art: https://github.com/robbiebarrat/art-dcgan.

- **The Creative AI Lab** https://creative-ai.org.
- **TensorFlow Magenta**. An open source research project exploring the role of ML in the creative process.[216].
- **Magenta Studio**. A suite of free music-making tools using machine learning models![217].
- **Style Transfer Tutorial** https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/r2/tutorials/generative/style_transfer.ipynb

[204] https://www.tensorflow.org/hub/tutorials
[205] https://www.tensorflow.org/lite/microcontrollers
[206] https://colab.research.google.com/drive/1qKPITTI879YHTxbTgYW_MAWMHFkbOBIk
[207] https://colab.research.google.com/drive/1lWUGZarlbORaHYUZlF9muCgpPl8pEvve
[208] https://www.tensorflow.org/tensorboard/tensorboard_in_notebooks
[209] https://colab.research.google.com/github/tensorflow/tensorboard/blob/master/docs/tensorboard_in_notebooks.ipynb
[210] https://colab.research.google.com/drive/1UCJt8EYjlzCs1H1d1X0iDGYJsHKwu-NO
[211] https://colab.research.google.com/drive/14CvUNTaX1OFHDfaKaaZzrBsvMfhCOHIR
[212] https://blog.tensorflow.org/2020/03/exploring-helpful-uses-for-bert-in-your-browser-tensorflow-js.html
[213] https://colab.research.google.com/github/zaidalyafeai/Notebooks/blob/master/TF_2_0.ipynb
[214] https://colab.research.google.com/drive/1DgkVmi6GksWOByhYVQpyUB4Rk3PUq0Cp
[215] https://medium.com/pytorch/get-started-with-pytorch-cloud-tpus-and-colab-a24757b8f7fc
[216] https://magenta.tensorflow.org
[217] https://magenta.tensorflow.org/studio

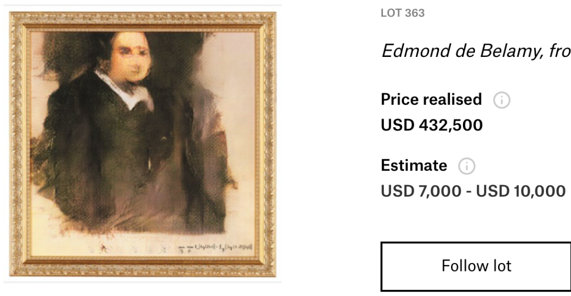



Figure 30: On October 25, 2018, the first AI artwork ever sold at Christie's auction house fetched USD 432,500.

- **AI x AR Paper Cubes** `https://experiments.withgoogle.com/paper-cubes`.
- **Photo Wake-Up** `https://grail.cs.washington.edu/projects/wakeup/`.
- **COLLECTION**. AI Experiments `https://experiments.withgoogle.com/ai`.

"*The Artists Creating with AI Won't Follow Trends; THEY WILL SET THEM.*" — The House of Montréal.AI Fine Arts

❖ Tuning Recurrent Neural Networks with Reinforcement Learning[218].
❖ **MuseNet**. Generate Music Using Many Different Instruments and Styles![219].
❖ Infinite stream of machine generated art. Valentin Vieriu `https://art42.net`.
❖ Deep Multispectral Painting Reproduction via Multi-Layer, Custom-Ink Printing. Shi et al.[220].
❖ Discovering Visual Patterns in Art Collections with Spatially-consistent Feature Learning. Shen et al.[221].
❖ Synthesizing Programs for Images using Reinforced Adversarial Learning, Ganin et al., 2018[222]. Agents[223].

## 10 AI Macrostrategy: Aligning AGI with Human Interests

**Montréal.AI Governance**: Policies at the intersection of AI, Ethics and Governance.

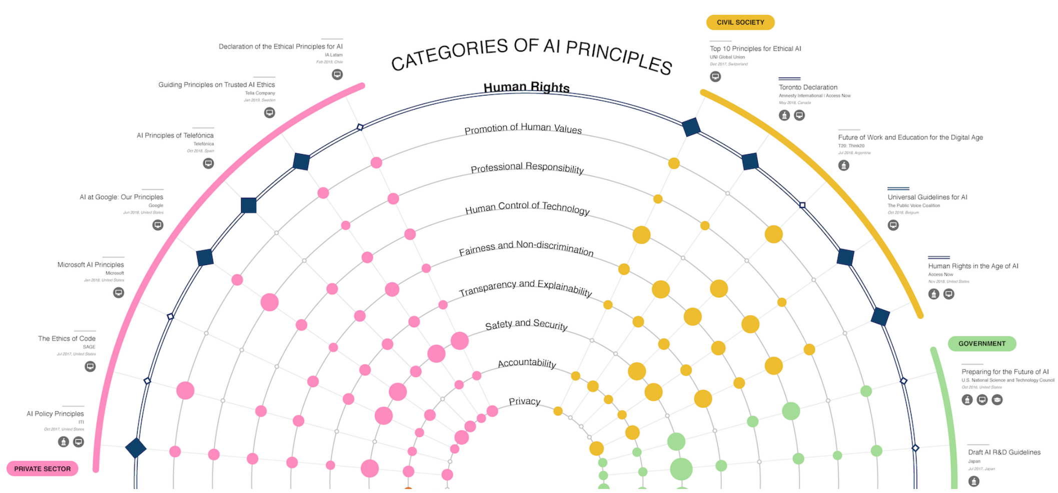



Figure 31: A Map of Ethical and Right-Based Approaches `https://ai-hr.cyber.harvard.edu/primp-viz.html`

[218] `https://magenta.tensorflow.org/2016/11/09/tuning-recurrent-networks-with-reinforcement-learning`
[219] `https://openai.com/blog/musenet/`
[220] `http://people.csail.mit.edu/liangs/papers/ToG18.pdf`
[221] `https://arxiv.org/pdf/1903.02678.pdf`
[222] `http://proceedings.mlr.press/v80/ganin18a.html`
[223] `https://github.com/deepmind/spiral`

"*(AI) will rank among our greatest technological achievements, and everyone deserves to play a role in shaping it.*" — Fei-Fei Li

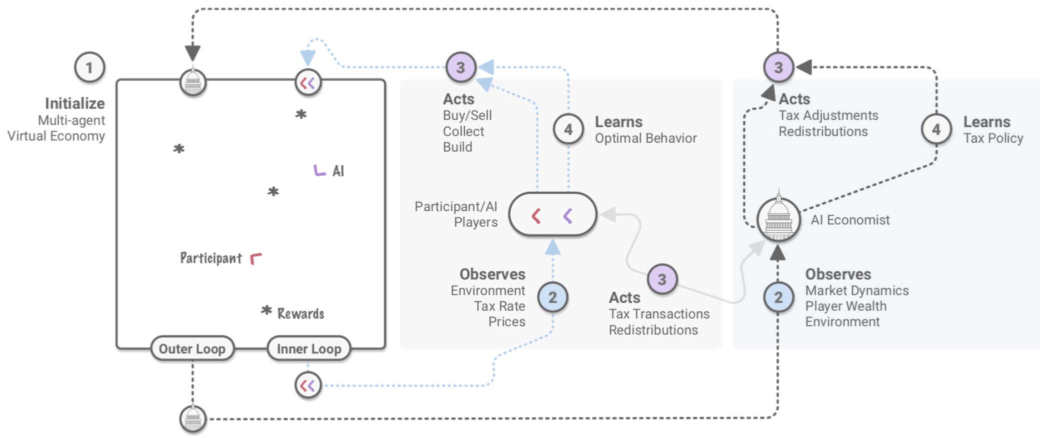



Figure 32: The AI Economist: Improving Equality and Productivity with AI-Driven Tax Policies. Zheng et al. `https://arxiv.org/abs/2004.13332`

❖ **AI Index**. `http://aiindex.org`.
❖ **Malicious AI Report**. `https://arxiv.org/pdf/1802.07228.pdf`.
❖ **Artificial Intelligence and Human Rights**. `https://ai-hr.cyber.harvard.edu`.
❖ **The AI Economist: Improving Equality and Productivity with AI-Driven Tax Policies**, Zheng et al.[224]. Blog[225].
❖ **Ethically Aligned Design, First Edition**[226]. From Principles to Practice `https://ethicsinaction.ieee.org`.
❖ **ADDRESS PREPARED BY POPE FRANCIS FOR THE PLENARY ASSEMBLY OF THE PONTIFICAL ACADEMY FOR LIFE**[227].

"*It's springtime for AI, and we're anticipating a long summer.*" — Bill Braun

## References

[1] Mnih et al. Human-Level Control Through Deep Reinforcement Learning. In *Nature* 518, pages 529–533. 26 February 2015. `https://storage.googleapis.com/deepmind-media/dqn/DQNNaturePaper.pdf`

[2] Yann LeCun, Yoshua Bengio and Geoffrey Hinton. Deep Learning. In *Nature* 521, pages 436–444. 28 May 2015. `https://www.cs.toronto.edu/~hinton/absps/NatureDeepReview.pdf`

[3] Goodfellow et al. Generative Adversarial Networks. *arXiv preprint arXiv:1406.2661*, 2014. `https://arxiv.org/abs/1406.2661`

[4] Yoshua Bengio, Andrea Lodi, Antoine Prouvost. Machine Learning for Combinatorial Optimization: a Methodological Tour d'Horizon. *arXiv preprint arXiv:1811.06128*, 2018. `https://arxiv.org/abs/1811.06128`

[5] Brockman et al. OpenAI Gym. 2016. `https://gym.openai.com`

[6] Devlin et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*, 2018. `https://arxiv.org/abs/1810.04805`

[7] Dai et al. Semi-supervised Sequence Learning. *arXiv preprint arXiv:1511.01432*, 2015. `https://arxiv.org/abs/1511.01432`

[8] Mnih et al. Asynchronous Methods for Deep Reinforcement Learning. *arXiv preprint arXiv:1602.01783*, 2016. `https://arxiv.org/abs/1602.01783`

[224] `https://arxiv.org/abs/2004.13332`
[225] `https://blog.einstein.ai/the-ai-economist/`
[226] `https://standards.ieee.org/content/dam/ieee-standards/standards/web/documents/other/ead1e.pdf`
[227] `http://w2.vatican.va/content/francesco/en/speeches/2020/february/documents/papa-francesco_20200228_accademia-perlavita.html`

[9] Schulman et al. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*, 2017. https://arxiv.org/abs/1707.06347

[10] Mnih et al. Playing Atari with Deep Reinforcement Learning. *DeepMind Technologies*, 2013. https://www.cs.toronto.edu/~vmnih/docs/dqn.pdf

[11] Ha et al. Recurrent World Models Facilitate Policy Evolution. *arXiv preprint arXiv:1809.01999*, 2018. https://arxiv.org/abs/1809.01999

[12] Kenneth et al. Designing neural networks through neuroevolution. In *Nature Machine Intelligence* VOL 1, pages 24–35. January 2019. https://www.nature.com/articles/s42256-018-0006-z.pdf

[13] So et al. The Evolved Transformer. *arXiv preprint arXiv:1901.11117*, 2019. https://arxiv.org/abs/1901.11117

[14] Silver et al. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. *arXiv preprint arXiv:1712.01815*, 2017. https://arxiv.org/abs/1712.01815

[15] Silver et al. AlphaGo Zero: Learning from scratch. In *DeepMind's Blog*, 2017. https://deepmind.com/blog/alphago-zero-learning-scratch/

[16] Andrychowicz et al. Learning to learn by gradient descent by gradient descent. *arXiv preprint arXiv:1606.04474*, 2016. https://arxiv.org/abs/1606.04474

[17] Nichol et al. Reptile: A Scalable Meta-Learning Algorithm. 2018. https://blog.openai.com/reptile/

[18] Frans et al. Meta Learning Shared Hierarchies. *arXiv preprint arXiv:1710.09767*, 2017. https://arxiv.org/abs/1710.09767

[19] Zoph and Le, 2017 Neural Architecture Search with Reinforcement Learning. *arXiv preprint arXiv:1611.01578*, 2017. https://arxiv.org/abs/1611.01578

[20] Finn et al., 2017 Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. *arXiv preprint arXiv:1703.03400*, 2017. https://arxiv.org/abs/1703.03400

[21] Salimans et al. Evolution Strategies as a Scalable Alternative to Reinforcement Learning. 2017. https://blog.openai.com/evolution-strategies/

[22] Lehman et al. The Surprising Creativity of Digital Evolution: A Collection of Anecdotes from the Evolutionary Computation and Artificial Life Research Communities. *arXiv preprint arXiv:1803.03453*, 2018. https://arxiv.org/abs/1803.03453

[23] Wang et al. Paired Open-Ended Trailblazer (POET): Endlessly Generating Increasingly Complex and Diverse Learning Environments and Their Solutions. *arXiv preprint arXiv:1901.01753*, 2019. https://arxiv.org/abs/1901.01753

[24] Foerster et al. Learning to Model Other Minds. 2018. https://blog.openai.com/learning-to-model-other-minds/

[25] Rabinowitz et al. Machine Theory of Mind. *arXiv preprint arXiv:1802.07740*, 2018. https://arxiv.org/abs/1802.07740