
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

November 25, 2019

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 to pioneer an impactful understanding of AI and to foster a vector for safe humanitarian artificial general intelligence (AGI).

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

Keywords AI-First · Artificial Intelligence · Deep Learning · Reinforcement Learning · Unsupervised Learning

A task of historic proportions — MONTRÉAL.AI is looking for associates and partners to join us in empowering Humanity on an unprecedented scale : *Captains of Industries, Iconic Tech Entrepreneurs, Philanthropists, Scholars and Luminaries*. We are tackling the most ambitious scientific quest in human history. "**GET ON A ROCKET SHIP!!**"

1 AI-First | Pre-AGI Technologies

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

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

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*

2 Getting Started

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

*Founding Chairman at MONTRÉAL.AI and QUÉBEC.AI <http://www.montreal.ai>.

Deep Learning Drizzle <https://deep-learning-drizzle.github.io>.

Papers With Code (*Learn Python 3 in Y minutes*²) <https://paperswithcode.com/state-of-the-art>.

The Measure of Intelligence (*Abstraction and Reasoning Corpus*³) <https://arxiv.org/abs/1911.01547>.

2.1 In the Cloud

Colab⁴. Practice Immediately⁵. Labs⁶: 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.

2.2 On a Local Machine

JupyterLab is an interactive development environment for working with notebooks, code and data⁷.

- Install Anaconda <https://www.anaconda.com/download/> and launch ‘Anaconda Navigator’
- Update Jupyterlab and launch the application. Under Notebook, click on ‘Python 3’

3 Deep Learning

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 both⁸. Introductory course <http://introtodeeplearning.com>. See also Table 1.

“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

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

Name	With Teacher	Without Teacher
Active	<i>Reinforcement Learning / Active Learning</i>	<i>Intrinsic Motivation / Exploration</i>
Passive	<i>Supervised Learning</i>	<i>Unsupervised Learning</i>

“If you have a large big dataset and you train a very big neural network, then success is guaranteed!” — Ilya Sutskever

“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

²<https://learnxinyminutes.com/docs/python3/>

³<https://github.com/fchollet/ARC>

⁴<https://medium.com/tensorflow/colab-an-easy-way-to-learn-and-use-tensorflow-d74d1686e309>

⁵<https://colab.research.google.com/github/GokuMohandas/practicalAI/>

⁶https://colab.research.google.com/github/aamini/introtodeeplearning_labs

⁷<https://blog.jupyter.org/jupyterlab-is-ready-for-users-5a6f039b8906>

⁸<https://github.com/lexfridman/mit-deep-learning>

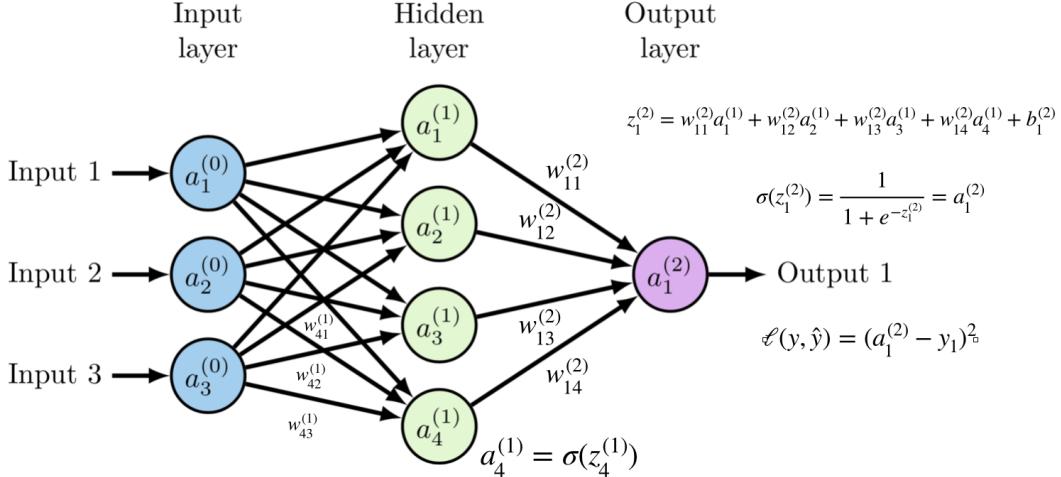


Figure 1: Multilayer perceptron (MLP).

Deep learning (*distributed representations + composition*) is a general-purpose learning procedure.

- ❖ Linear Algebra. Prof. Gilbert Strang⁹.
- ❖ Dive into Deep Learning <http://d2l.ai>.
- ❖ Minicourse in Deep Learning with PyTorch¹⁰.
- ❖ Deep Learning. The full deck of (600+) slides, Gilles Louppe¹¹.
- ❖ A Selective Overview of Deep Learning <https://arxiv.org/abs/1904.05526>.
- ❖ Anatomy of Matplotlib <https://github.com/matplotlib/AnatomyOfMatplotlib>.
- ❖ PoseNet Sketchbook <https://googlecreativelab.github.io/posenet-sketchbook/>.
- ❖ A Recipe for Training Neural Networks <https://karpathy.github.io/2019/04/25/recipe/>.
- ❖ Algebra, Topology, Differential Calculus, and Optimization Theory For Computer Science and Machine Learning¹².
- ❖ 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>.

3.1 Universal Approximation Theorem

Neural Networks + Gradient Descent + GPU¹³:

- Infinitely flexible function: *Neural Network* (multiple hidden layers: Deep Learning)¹⁴.
- All-purpose parameter fitting: *Backpropagation*¹⁵¹⁶.
- 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

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

⁹<https://ocw.mit.edu/courses/mathematics/18-06-linear-algebra-spring-2010/video-lectures/>

¹⁰<https://github.com/Atcold/pytorch-Deep-Learning-Minicourse>

¹¹<https://glouppe.github.io/info8010-deep-learning/pdf/lec-all.pdf>

¹²<https://drive.google.com/file/d/1sJvLQwxMyu89t2z4Zf9tD707efnbIUyB/view>

¹³http://wiki.fast.ai/index.php/Lesson_1_Notes

¹⁴<http://neuralnetworksanddeeplearning.com/chap4.html>

¹⁵https://github.com/DebPanigrahi/Machine-Learning/blob/master/back_prop.ipynb

¹⁶<https://www.jeremyjordan.me/neural-networks-training/>

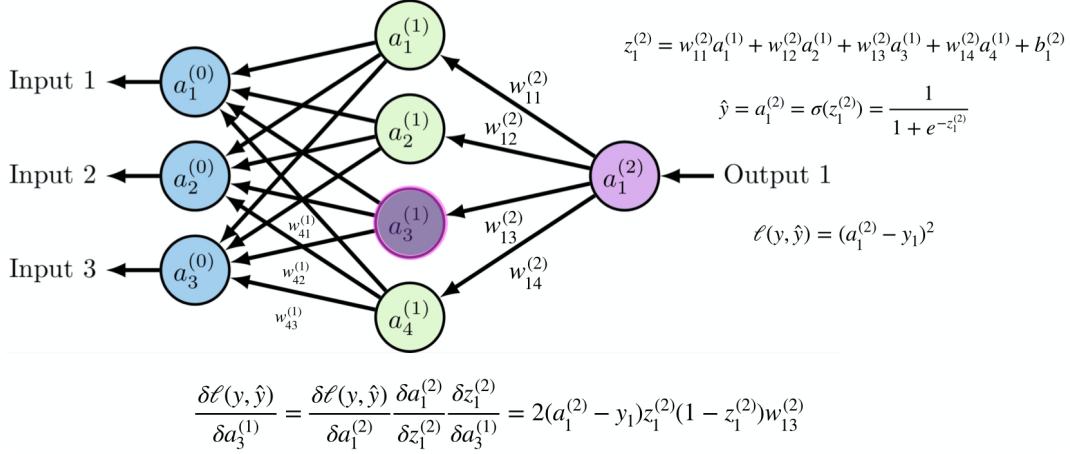


Figure 2: All-purpose parameter fitting: Backpropagation.

3.2 Convolution Neural Networks (Useful for Images | Space)

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/>.

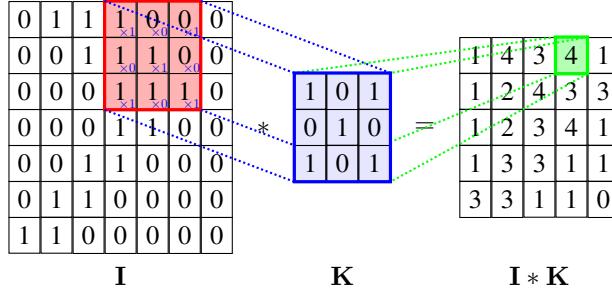


Figure 3: 2D Convolution. Source: Cambridge Coding Academy

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 parameters¹⁷. Reading¹⁸.

In images, local combinations of edges form motifs, motifs assemble into parts, and parts form objects¹⁹²⁰.

- ❖ CS231N : Convolutional Neural Networks for Visual Recognition²¹.
- ❖ 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 persist²². 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.

¹⁷<http://cs231n.github.io/convolutional-networks/>

¹⁸<https://ml4a.github.io/ml4a/convnets/>

¹⁹<http://yosinski.com/deepvis>

²⁰<https://distill.pub/2017/feature-visualization/>

²¹https://www.youtube.com/playlist?list=PLzUTmXVwsnXod6WNdg57Yc3zFx_f-RYsq

²²<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

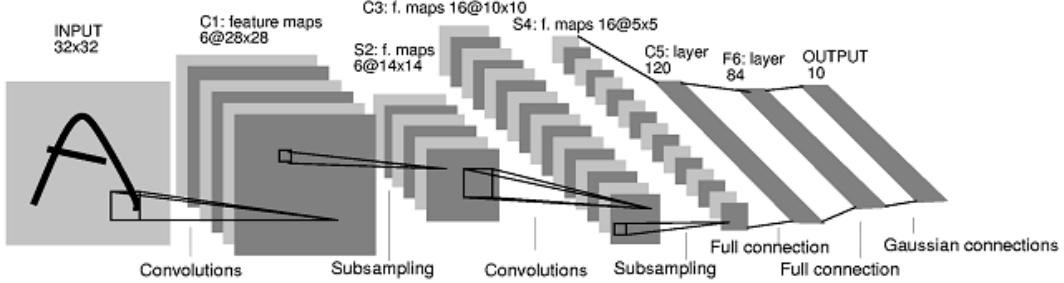


Figure 4: Architecture of LeNet-5, a Convolutional Neural Network. LeCun et al., 1998

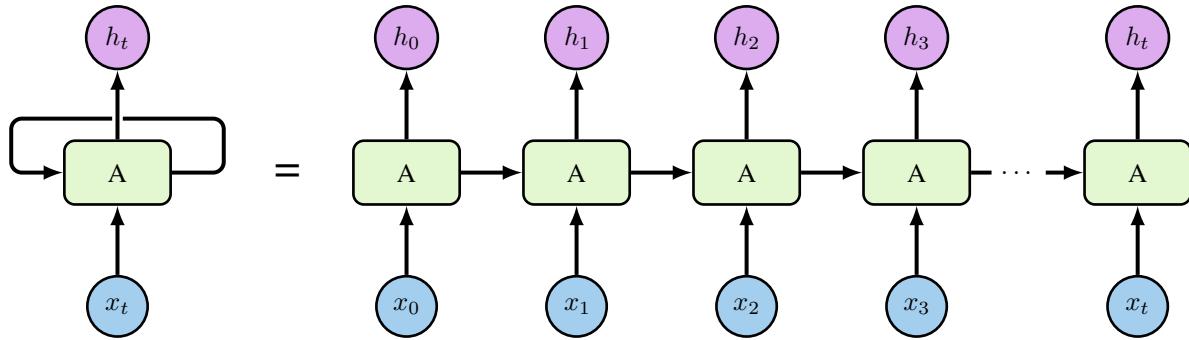


Figure 5: RNN Layers Reuse Weights for Multiple Timesteps.

"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

- ❖ Long Short-Term-Memory (LSTM), Sepp Hochreiter and Jürgen Schmidhuber²³.
- ❖ CS224N : Natural Language Processing with Deep Learning²⁴.
- ❖ Can Neural Networks Remember? Slides by Vishal Gupta: http://vishalgupta.me/deck/char_lstms/.
- ❖ Understanding LSTM Networks <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>.
- ❖ The Unreasonable Effectiveness of Recurrent Neural Networks, blog (2015) by Andrej Karpathy²⁵.
- ❖ Attention and Augmented Recurrent Neural Networks <https://distill.pub/2016/augmented-rnns/>.
- ❖ Attention Is All You Need, Vaswani et al. <https://arxiv.org/abs/1706.03762>.
- ❖ Transformer model for language understanding. Tutorial showing how to write Transformer in TensorFlow 2.0²⁶.

3.4 Transformers

Transformers are generic, simple 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)²⁷.

- ❖ The Illustrated Transformer <http://jalammar.github.io/illustrated-transformer/>.
- ❖ The annotated transformer (code) <http://nlp.seas.harvard.edu/2018/04/03/attention.html>.
- ❖ Transformer in TensorFlow 2.0 (code) <https://www.tensorflow.org/beta/tutorials/text/transformer>.
- ❖ Making Transformer networks simpler and more efficient²⁸.

²³<https://www.bioinf.jku.at/publications/older/2604.pdf>

²⁴https://www.youtube.com/playlist?list=PLU40WL80194IJzQtileLTqGZuXtG1LMP_

²⁵<http://karpathy.github.io/2015/05/21/rnn-effectiveness/>

²⁶<https://www.tensorflow.org/alpha/tutorials/sequences/transformer>

²⁷<http://www.peterbloem.nl/blog/transformers>

²⁸<https://ai.facebook.com/blog/making-transformer-networks-simpler-and-more-efficient/>

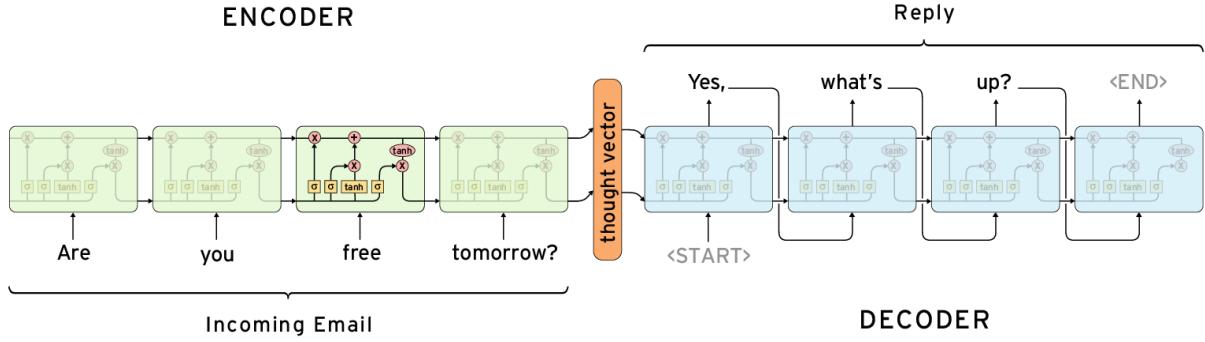


Figure 6: **Google Smart Reply System** is built on a pair of recurrent neural networks. Diagram by Chris Olah

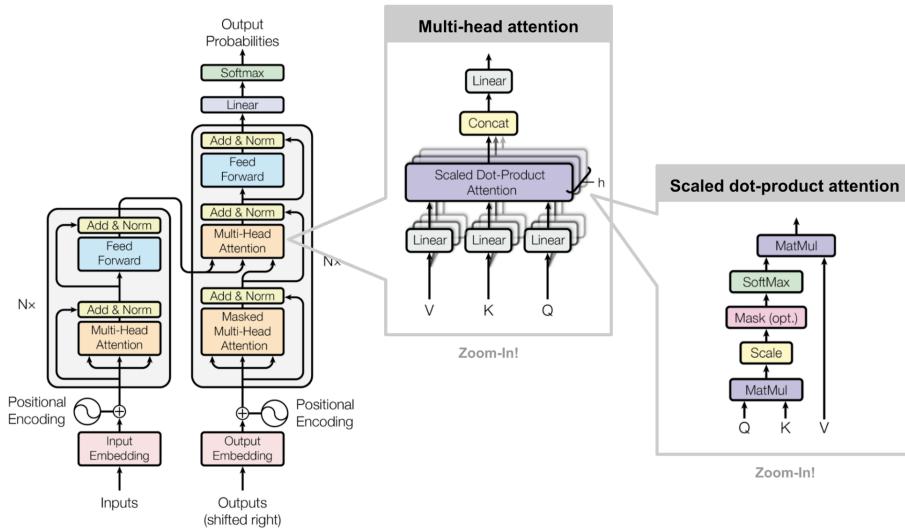


Figure 7: Attention Is All You Need. Vaswani et al., 2017 : <https://arxiv.org/abs/1706.03762>.

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)²⁹.

- 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 Implications³⁰.

"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

- ❖ How to Build OpenAI's GPT-2: "The AI That's Too Dangerous to Release"³¹.
- ❖ 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.

²⁹<https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html>

³⁰<https://blog.openai.com/better-language-models/>

³¹<https://blog.floydhub.com/gpt2/>



Write With Transformer

Get a modern neural network to
auto-complete your thoughts.

This web app, built by the Hugging Face team, is the official demo of the `pytorch-transformers` repository's text generation capabilities.

Figure 8: **Write With Transformer** allows text generation on general-purpose Transformer models (BERT, GPT-2, XLNet ...). By *Hugging Face* : <https://transformer.huggingface.co>.

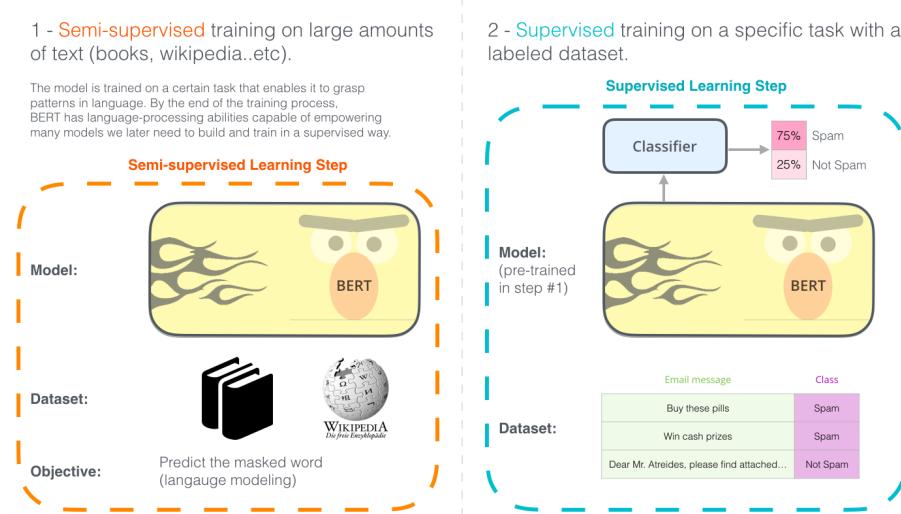


Figure 9: The two steps of how BERT is developed. Source <https://jalammar.github.io/illustrated-bert/>.

3.5 Unsupervised Learning

True intelligence will require independent learning strategies.

Unsupervised learning is a paradigm for creating AI that learns without a particular task in mind: learning for the sake of learning³². 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]³³.

³²<https://deepmind.com/blog/unsupervised-learning/>

³³https://media.neurips.cc/Conferences/NIPS2018/Slides/Deep_Unsupervised_Learning.pdf

"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

Self-supervised learning is derived from unsupervised learning where the data provides the supervision. E.g. Word2vec³⁴, 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 numbers³⁵.

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].

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

"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 indistinguishable from the genuine articles. See Figure 10.

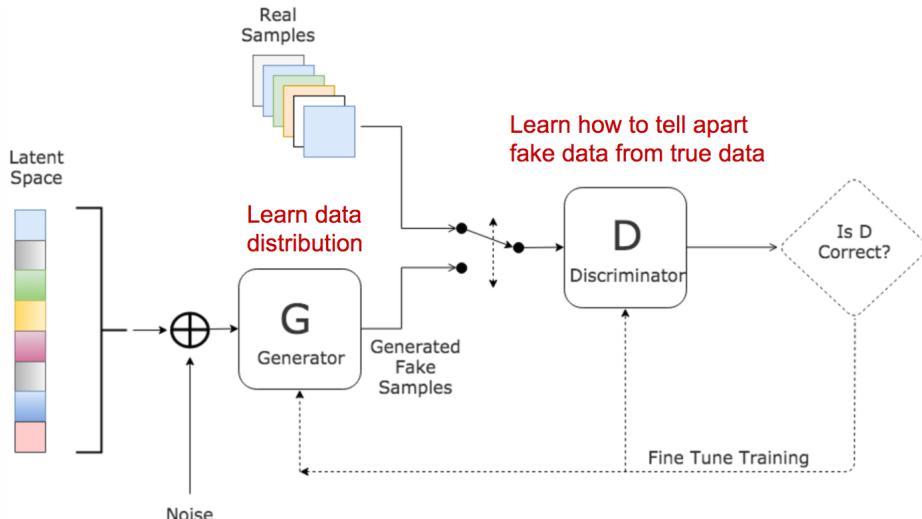


Figure 10: GAN: Neural Networks Architecture Pioneered by Ian Goodfellow at University of Montreal (2014).

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>.
- 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 McDonald³⁶.

³⁴<https://jalammar.github.io/illustrated-word2vec/>

³⁵<http://projector.tensorflow.org>

³⁶<https://medium.com/@kcimc/how-to-recognize-fake-ai-generated-images-4d1f6f9a2842>

- ❖ Introducing **TF-GAN**: A lightweight GAN library for TensorFlow 2.0³⁷.
- ❖ Generative Adversarial Networks (GANs) in 50 lines of code (PyTorch)³⁸.
- ❖ 100,000 Faces Imagined by a GAN <https://generated.photos>.
- ❖ Infinite-resolution (CPPNs, GANs and TensorFlow.js) <https://thispicturedoesnotexist.com>.
- ❖ Few-Shot Adversarial Learning of Realistic Neural Talking Head Models³⁹.
- ❖ Wasserstein GAN <http://www.depthfirstlearning.com/2019/WassersteinGAN>.
- ❖ 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>.
- ❖ CariGANs: Unpaired Photo-to-Caricature Translation. Cao et al.: <https://cari-gan.github.io>.
- ❖ GANpaint Paint with GAN units <http://gandissect.res.ibm.com/ganpaint.html>.
- ❖ PyTorch pretrained BigGAN <https://github.com/huggingface/pytorch-pretrained-BigGAN>.
- ❖ 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-Encoders⁴⁰ (VAEs) are powerful models for learning low-dimensional representations See Figure 11. 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)).

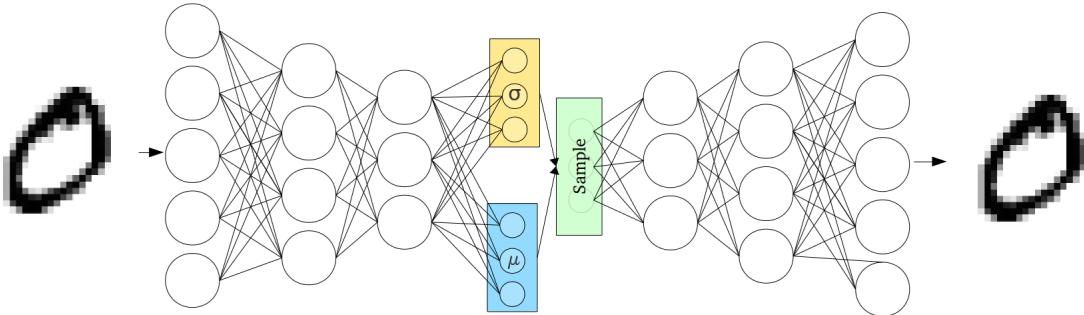


Figure 11: Variational Autoencoders (VAEs): Powerful Generative Models.

- ❖ Colab⁴¹: "Debiasing Facial Detection Systems." *AIEthics*
- ❖ SpaceSheet: Interactive Latent Space Exploration with a Spreadsheet <https://vusd.github.io/spacesheet/>.
- ❖ MusicVAE: Learning latent spaces for musical scores <https://magenta.tensorflow.org/music-vae>.
- ❖ Slides: A Few Unusual Autoencoders <https://colinraffel.com/talks/vector2018few.pdf>.
- ❖ Generative models in **Tensorflow 2** <https://github.com/timsainb/tensorflow2-generative-models/>.
- ❖ Reading: Disentangled VAE's (DeepMind 2016) <https://arxiv.org/abs/1606.05579>.

³⁷<https://medium.com/tensorflow/introducing-tf-gan-a-lightweight-gan-library-for-tensorflow-2-0-36d767e1abae>

³⁸<https://medium.com/@devnag/generative-adversarial-networks-gans-in-50-lines-of-code-pytorch-e81b79659e3f>

³⁹<https://arxiv.org/abs/1905.08233>

⁴⁰<https://arxiv.org/abs/1906.02691v2>

⁴¹https://colab.research.google.com/github/aamini/introtodeeplearning_labs/blob/master/lab2/Part2_debiasing_solution.ipynb

4 Autonomous Agents

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 definition⁴² of the *universal intelligence* of agent π ⁴³:

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

"Intelligence measures an agent's ability to achieve goals in a wide range of environments." — Shane Legg

4.1 Deep Reinforcement Learning

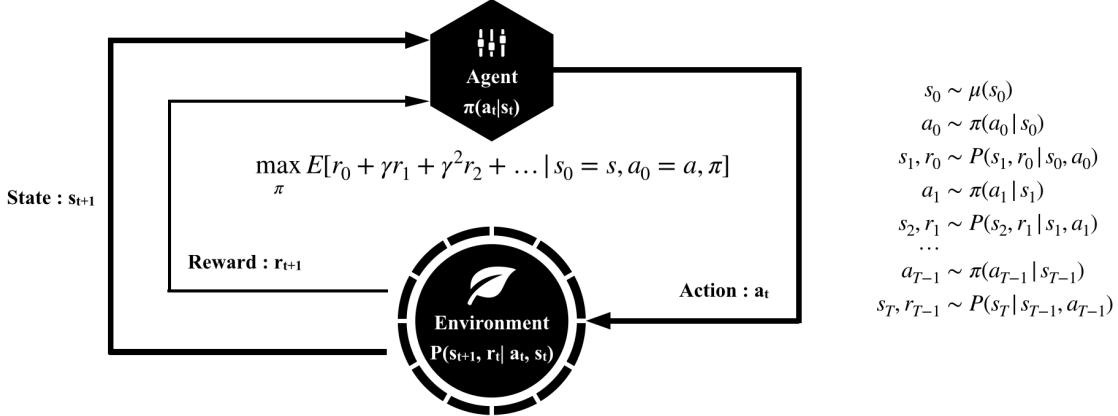


Figure 12: An Agent Interacts with an Environment.

Reinforcement learning (RL) studies how an agent can learn how to achieve goals in a complex, uncertain environment (Figure 13) [5]. Recent superhuman results in many difficult environments combine deep learning with RL (*Deep Reinforcement Learning*). See Figure 13 for a taxonomy of RL algorithms.

- ❖ CS 188 : Introduction to Artificial Intelligence⁴⁴.
- ❖ Introduction to Reinforcement Learning by DeepMind⁴⁵.

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 R_t , called the return:

$$R_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots \quad (3)$$

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

$$Q(s, a) = E[R_t] \quad (4)$$

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

$$\pi^*(s) = \text{argmax}_a Q(s, a) \quad (5)$$

⁴²<https://arxiv.org/abs/0712.3329>

⁴³Where μ 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_μ^π is the agent's "ability to achieve".

⁴⁴<https://inst.eecs.berkeley.edu/~cs188/fa18/>

⁴⁵<https://www.youtube.com/watch?v=2pWv7G0vuf0&list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzF0bQ>

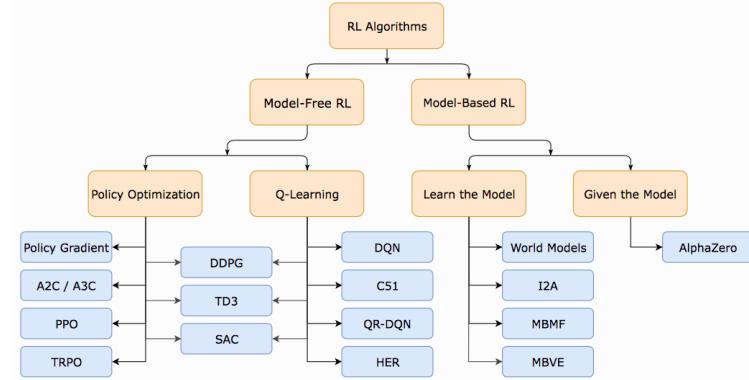


Figure 13: A Taxonomy of RL Algorithms. Source: Spinning Up in Deep RL by Achiam et al. | OpenAI

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

TF-Agents (DQN Tutorial) | Colab <https://colab.research.google.com/github/tensorflow/agents>.

4.1.2 Model-Free RL | Policy-Based

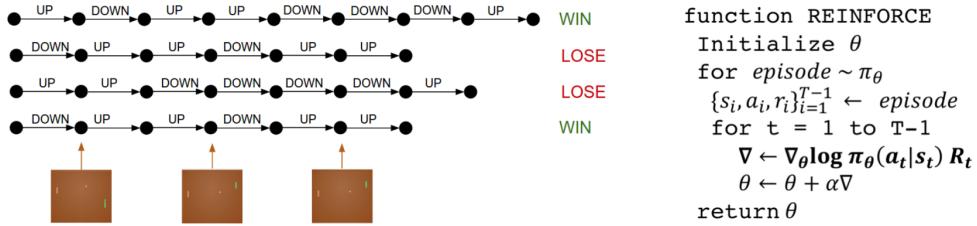


Figure 14: Policy Gradient Directly Optimizes the Policy.

Run a policy for a while (code: <https://gist.github.com/karpathy/a4166c7fe253700972fcbe77e4ea32c5>):

$$\tau = (s_0, a_0, r_0, s_1, a_1, r_1, \dots, s_{T-1}, a_{T-1}, r_{T-1}, s_T) \quad (6)$$

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

$$\nabla_{\theta} E_{\tau}[R(\tau)] = E_{\tau} \left[\sum_{t=0}^{T-1} \nabla_{\theta} \log \pi(a_t | s_t, \theta) R(\tau) \right] \quad (7)$$

- **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].

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 Models⁴⁶). 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 17. Ha et al, 2018[11].
 - **Learn the Model:** *Learning Latent Dynamics for Planning from Pixels* <https://planetrl.github.io/>.

⁴⁶<https://worldmodels.github.io>

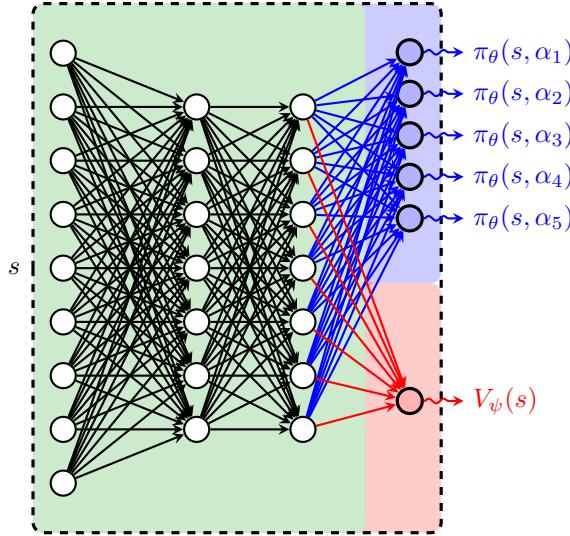


Figure 15: Asynchronous Advantage Actor-Critic (A3C). Source: Petar Velickovic

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA
1	Project	Maintainer	Framework	Execution	Algorithms (discrete & continuous)												Additional features										
2				Parallel	Distributed	DQN	Rainbow	REINFORCE	A2C	PPO	DDPG	SAC	TD3	REINFORCE	A2C	PPO	n-step return	prioritized experience replay	distributional value function approximation	hyperbolic discounting	dict observations support	?					
3	OpenAI baselines	OpenAI	Tensorflow	✓	✗	✓	✗	✗	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗	last commit 3 June	8.1k	commit activity 1/month	code size 1.22 MB	
4	stable_baselines	Antonin Raffin, Ashish Kapoor	Tensorflow	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	last commit 1 July	1.1k	commit activity 6/month	code size 893 kB	
5	CatalystRL	Sergey Kolesnikov	PyTorch	✗?	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	last commit 10 June	710	commit activity 19/month	code size 4.92 kB	
6	Ray.rllib	Ray Team	Tensorflow	✗?	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	last commit 10 July	8k	commit activity 100/month	code size 1.22 kB	
7	TF.agents	Google	Tensorflow	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	last commit 10 June	749	commit activity 34/month	code size 1.28 kB	
8	Horizon	Facebook	PyTorch	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	last commit 10 July	2k	commit activity 28/month	code size 1.04 kB	
9	Coach	Intel	Tensorflow	✗?	✗?	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	last commit 10 June	1.4k	commit activity 7/month	code size 1.99 kB	
10	Garage	community	Tensorflow	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	last commit 10 July	404	commit activity 27/month	code size 1.54 kB	
11	SLM-Lab	Wah Loon Kong, Laura Graesser	PyTorch	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	last commit 10 July	540	commit activity 147/month	code size 315 kB	
12	Dopamine	Google	Tensorflow	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	last commit 10 July	6.1k	commit activity 3/month	code size 1.13 kB	
13	OpenAI_zoo	OpenAI	Tensorflow	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	last commit 10 July	3.2k	commit activity 100/month	code size 403 kB	
14	tf1	DeepMind	Tensorflow	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	last commit 10 July	2.7k	commit activity 100/month	code size 122 kB	
15	scalable_agent	DeepMind	Tensorflow	?	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	last commit 10 July	654	commit activity 0/month	code size 564 kB	
16	ELF	Facebook	PyTorch	?	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	last commit 10 July	1.3k	commit activity 0/month	code size 364 kB	
17	keras-d	Matthias Piappe	Tensorflow	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	last commit 10 July	4k	commit activity 0/month	code size 191 kB	
18	icositkov	Ilya Kostrikov	PyTorch	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	last commit 10 July	1.4k	commit activity 0/month	code size 95.9 kB	
19	Rainbow	Kai Arulkumar	PyTorch	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	last commit 10 July	777	commit activity 2/month	code size 30.4 kB	
20	Vel	Jerry (?)	PyTorch	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	last commit 10 July	23k	commit activity 0/month	code size 668 kB	
21	tensorforce		Tensorflow	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	last commit 10 July	2.4k	commit activity 0/month	code size 870 kB	
22	RL_Adventure		PyTorch	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	last commit 10 July	1.6k	commit activity 0/month	code size 1.07 kB	
23	DeepRL_Tutorials		PyTorch	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	last commit 10 July	410	commit activity 0/month	code size 4.15 kB	
24	surreal		TensorX	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	last commit 10 July	342	commit activity 0/month	code size 40.9 kB	
25	lagent		PyTorch	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	last commit 10 July	11k	commit activity 0/month	code size 2.24 kB	
26	demisrlb		Tensorflow	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	last commit 10 July	71	commit activity 0/month	code size 1.39 kB	
27	scaler		Tensorflow	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	last commit 10 July	251	commit activity 2/month	code size 73.9 kB	
28	symal	WhiRL	PyTorch	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	last commit 10 July	1.1k	commit activity 0/month	code size 1.11 kB	

Figure 16: Open-Source RL Algorithms <https://docs.google.com/spreadsheets/d/1EeFPd-XIQ3mq-9snT1AZSsFY7Hbnmd7P5bbT8LPuMn0/>

- 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 Diagram⁴⁷.

4.1.4 Improving Agent Design

Via Reinforcement Learning: Blog⁴⁸. arXiv⁴⁹. ASTool <https://github.com/hardmaru/astool/>.

Via Evolution: Video⁵⁰. Evolved Creatures <http://www.karlsims.com/evolved-virtual-creatures.html>.

"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 environments⁵¹.

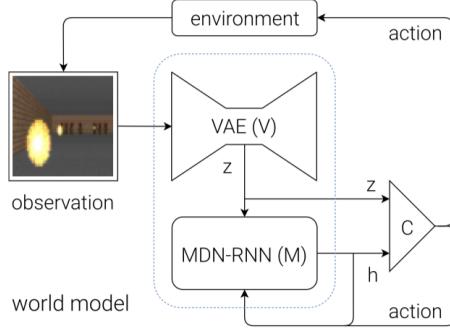
⁴⁷https://applied-data.science/static/main/res/alpha_go_zero_cheat_sheet.png

⁴⁸<https://designrl.github.io>

⁴⁹<https://arxiv.org/abs/1810.03779>

⁵⁰https://youtu.be/JBgG_VSP7f8

⁵¹<https://arxiv.org/abs/1910.07224>



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

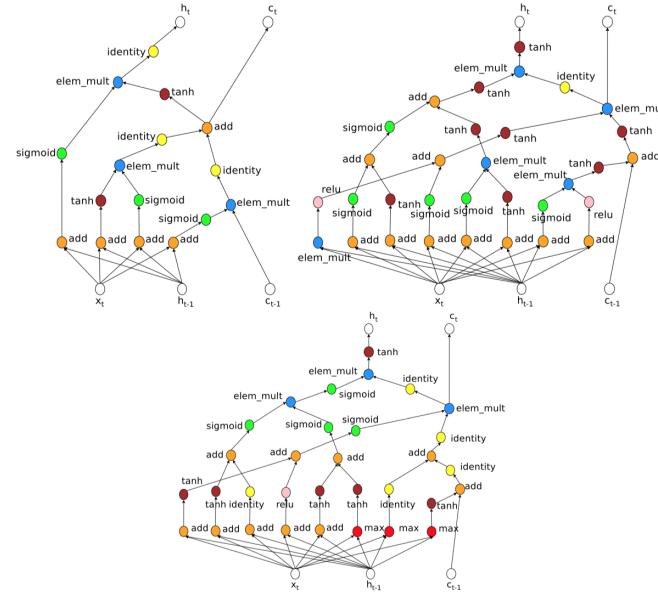


Figure 18: A comparison of the original LSTM cell vs. two new good generated. Top left: LSTM cell. [19]

4.1.5 OpenAI Baselines

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

Colab <https://colab.research.google.com/drive/1KKq9A3dRTq1q6bJmPyF0gg917gQyTjJI>.

4.1.6 Google Dopamine and A Zoo of Agents

Dopamine is a research framework for fast prototyping of reinforcement learning algorithms.⁵²

A Zoo of Atari-Playing Agents: Code⁵³, Blog⁵⁴ and Colaboratory notebook <https://colab.research.google.com/github/uber-research/atari-model-zoo/blob/master/colab/AtariZooColabDemo.ipynb>.

4.1.7 TRFL : TensorFlow Reinforcement Learning

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

⁵²<https://github.com/google/dopamine>

⁵³<https://github.com/uber-research/atari-model-zoo>

⁵⁴<https://eng.uber.com/atari-zoo-deep-reinforcement-learning/>

4.1.8 *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)^{†⁵⁵}, 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

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].

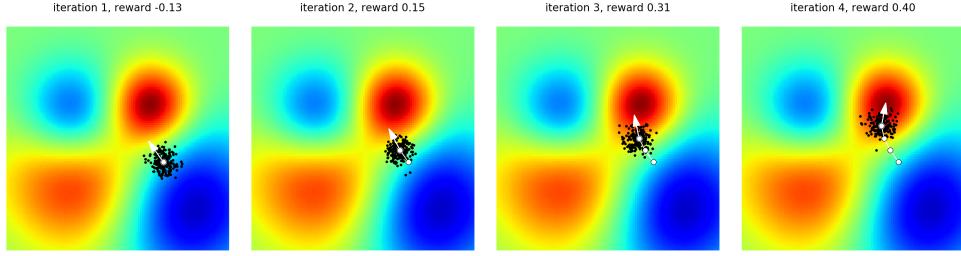


Figure 19: <https://colab.research.google.com/github/karpathy/randomfun/blob/master/es.ipynb>.

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 Strategies⁵⁶.
- ❖ VAE+CPPN+GAN⁵⁷.

⁵⁵<https://onlinelibrary.wiley.com/doi/epdf/10.1002/anie.201907729>

⁵⁶<https://lilianweng.github.io/lil-log/2019/09/05/evolution-strategies.html>

⁵⁷https://colab.research.google.com/drive/1_OoZ3z_C5J15gnxD0E9VEMCTs-F18pvM

- ❖ Demos: ES on CartPole-v1⁵⁸ and ES on LunarLanderContinuous-v2⁵⁹.
- ❖ Spiders Can Fly Hundreds of Miles Riding the Earth's Magnetic Fields⁶⁰.
- ❖ A Visual Guide to ES <http://blog.otoro.net/2017/10/29/visual-evolution-strategies/>.

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 world⁶¹.

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

TensorFlow.js Implementation of DeepMind's AlphaZero Algorithm for Chess. Live Demo⁶² | Code⁶³

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 Five⁶⁴.

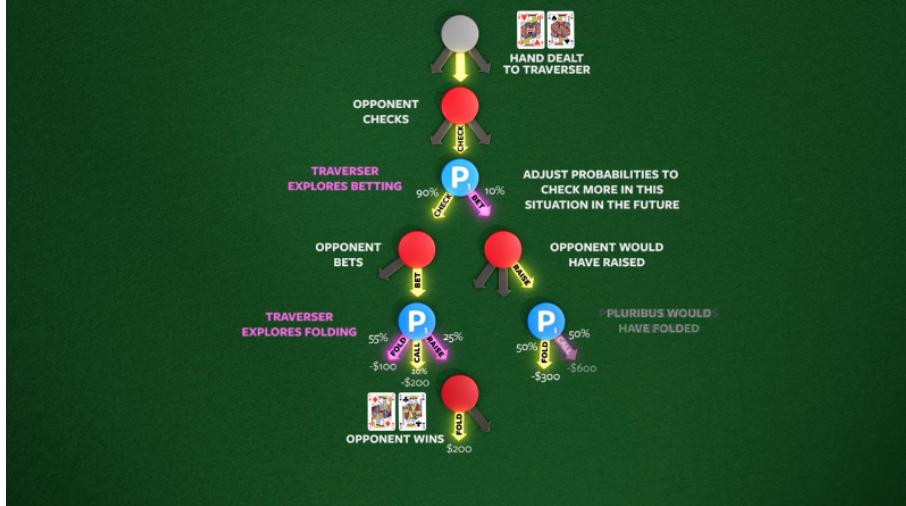


Figure 20: Facebook, Carnegie Mellon build first AI that beats pros in 6-player poker <https://ai.facebook.com/blog/pluribus-first-ai-to-beat-pros-in-6-player-poker>

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

⁵⁸<https://colab.research.google.com/drive/1bMZWHdhm-mT9NjENWoVewUks7cGV10go>

⁵⁹https://colab.research.google.com/drive/1lvyKjFtc_C_8njCKD-MnXEW8LPS2RPr6

⁶⁰[https://www.cell.com/current-biology/fulltext/S0960-9822\(18\)30693-6](https://www.cell.com/current-biology/fulltext/S0960-9822(18)30693-6)

⁶¹<http://metalearning-symposium.ml>

⁶²<https://frpays.github.io/lc0-js/engine.html>

⁶³<https://github.com/frpays/lc0-js/>

⁶⁴<https://blog.openai.com/openai-five/>

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 implementation⁶⁵.

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

Neural MMO: a massively multiagent env. for simulations with many long-lived agents. Code⁶⁶ and 3D Client⁶⁷.

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

- ❖ Meta Learning Shared Hierarchies[18] (*The Lead Author is in High School!*)
- ❖ Causal Reasoning from Meta-reinforcement Learning <https://arxiv.org/abs/1901.08162>

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].

$$\theta \leftarrow \theta - \beta \nabla_{\theta} \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i}) \quad (8)$$

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].

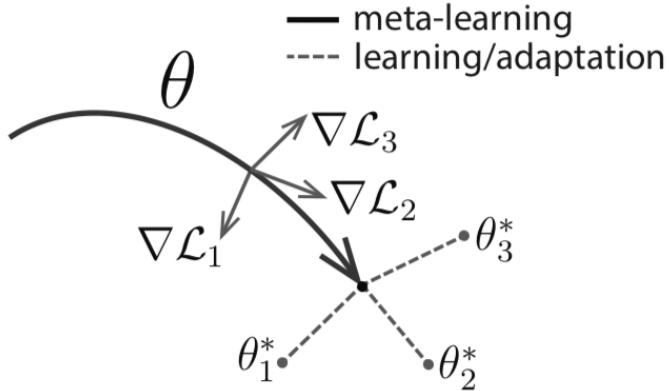


Figure 21: Diagram of Model-Agnostic Meta-Learning (MAML)

- ❖ Meta-Learning with Implicit Gradients <https://arxiv.org/abs/1909.04630>
- ❖ How to Train MAML (Model-Agnostic Meta-Learning)⁶⁸
- ❖ Colaboratory reimplementation of MAML (Model-Agnostic Meta-Learning) in TF 2.0⁶⁹.

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*⁷⁰, 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.

⁶⁵https://github.com/eugenevinitsky/sequential_social_dilemma_games

⁶⁶<https://github.com/openai/neural-mmo>

⁶⁷<https://github.com/jsuarez5341/neural-mmo-client>

⁶⁸<https://medium.com/towards-artificial-intelligence/how-to-train-maml-model-agnostic-meta-learning-90aa093f8e46>

⁶⁹<https://colab.research.google.com/github/mari-linhares/tensorflow-maml/blob/master/maml.ipynb>

⁷⁰<https://arxiv.org/abs/1905.10985>

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 transformers, 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.

5 Environments

Platforms for training autonomous agents.

"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)

5.1 OpenAI Gym

The OpenAI Gym <https://gym.openai.com/> (Blog⁷¹ | GitHub⁷²) is a toolkit for developing and comparing reinforcement learning algorithms. What makes the gym so great is a common API around environments.

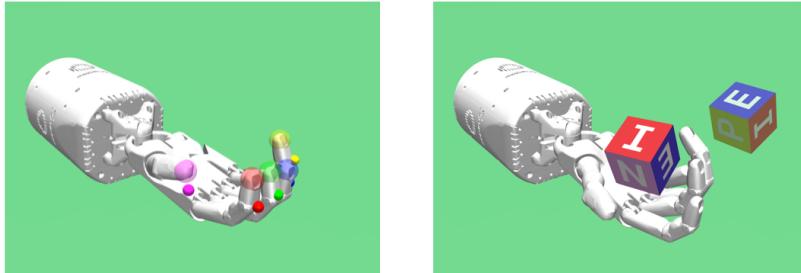


Figure 22: Robotics Environments <https://blog.openai.com/ingredients-for-robotics-research/>

How to create new environments for Gym⁷³. **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`

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

⁷¹<https://blog.openai.com/openai-gym-beta/>

⁷²<https://github.com/openai/gym>

⁷³<https://github.com/openai/gym/blob/master/docs/creating-environments.md>

1. Download *gym-foo-v3*⁷⁴
2. cd gym-foo-v3
3. pip install -e .
4. python ES-foo-v3.py

❖ OpenAI Gym Environment for Trading⁷⁵.

5.2 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/zuxingdong/dm2gym>.

5.3 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>.

- Getting Started with Marathon Environments for Unity ML-Agents⁷⁶.
- Arena: A General Evaluation Platform and Building Toolkit for Multi-Agent Intelligence⁷⁷.

5.4 POET: Paired Open-Ended Trailblazer

Diversity is the premier product of evolution. Endlessly generate increasingly complex and diverse learning environments⁷⁸. Open-endedness could generate learning algorithms reaching human-level intelligence[23].

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

6 Datasets

Google Dataset Search Beta (Blog⁷⁹) <https://toolbox.google.com/datasetsearch>.

TensorFlow Datasets: load a variety of public datasets into TensorFlow programs (Blog⁸⁰ | Colab⁸¹).

7 Deep-Learning Hardware

- ❖ A Full Hardware Guide to Deep Learning, by Tim Dettmers⁸².
- ❖ Which GPU(s) to Get for Deep Learning, by Tim Dettmers⁸³.
- ❖ Build AI that works offline with Coral Dev Board, Edge TPU, and TensorFlow Lite, by Daniel Situnayake⁸⁴.
- ❖ Jetson Nano. A small but mighty AI computer to create intelligent systems⁸⁵.

⁷⁴<https://drive.google.com/file/d/1cGncsXJ56UUKC09MaRWJVTnxiQEnLuxS/view?usp=sharing>

⁷⁵<https://github.com/hackthemarket/gym-trading>

⁷⁶<https://towardsdatascience.com/gettingstartedwithmarathonenvs-v0-5-0a-c1054a0b540c>

⁷⁷<https://arxiv.org/abs/1905.08085>

⁷⁸<https://eng.uber.com/poet-open-ended-deep-learning/>

⁷⁹<https://www.blog.google/products/search/making-it-easier-discover-datasets/>

⁸⁰<https://medium.com/tensorflow/introducing-tensorflow-datasets-c7f01f7e19f3>

⁸¹<https://colab.research.google.com/github/tensorflow/datasets/blob/master/docs/overview.ipynb>

⁸²<http://timdettmers.com/2018/12/16/deep-learning-hardware-guide/>

⁸³<http://timdettmers.com/2019/04/03/which-gpu-for-deep-learning/>

⁸⁴<https://medium.com/tensorflow/build-ai-that-works-offline-with-coral-dev-board-edge-tpu-and-tensorflow-lite-70>

⁸⁵<https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-nano/>



Figure 23: Edge TPU - Dev Board <https://coral.withgoogle.com/products/dev-board/>

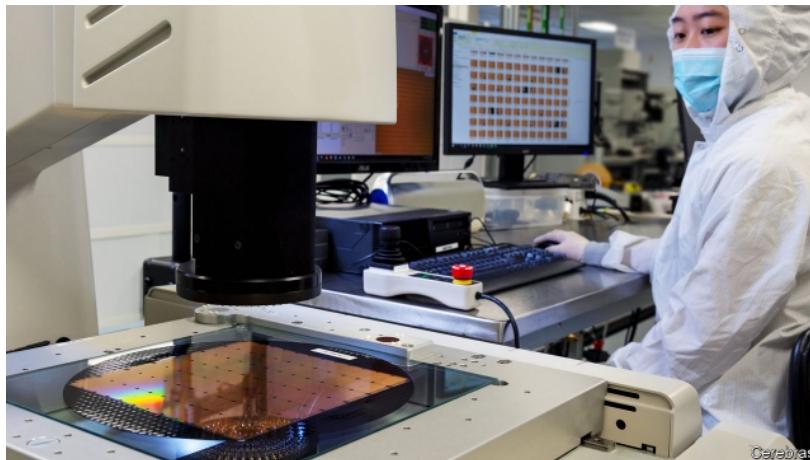


Figure 24: The world's largest chip : Cerebras Wafer Scale Engine <https://www.cerebras.net>

8 Deep-Learning Software

TensorFlow

- TensorFlow 2.0 + Keras Crash Course. Colab⁸⁶.
- tf.keras (TensorFlow 2.0) for Researchers: Crash Course. Colab⁸⁷.
- TensorFlow 2.0: basic ops, gradients, data preprocessing and augmentation, training and saving. Colab⁸⁸.
- TensorBoard in Jupyter Notebooks. Colab⁸⁹.
- TensorFlow Lite for Microcontrollers⁹⁰.

PyTorch

- PyTorch primer. Colab⁹¹.

⁸⁶<https://colab.research.google.com/drive/1UCJt8EYjlzCs1H1d1X0iDGYJsHKwu-N0>

⁸⁷<https://colab.research.google.com/drive/14CvUNTaX10FHDfaKaaZzrBsvMfhCOHIR>

⁸⁸https://colab.research.google.com/github/zaidalyafeai/Notebooks/blob/master/TF_2_0.ipynb

⁸⁹https://colab.research.google.com/github/tensorflow/tensorboard/blob/master/docs/r2/get_started.ipynb

⁹⁰<https://petewarden.com/2019/03/07/launching-tensorflow-lite-for-microcontrollers/>

⁹¹<https://colab.research.google.com/drive/1DgkVm6GksW0ByhYVQpyUB4Rk3PUq0Cp>

- PyTorch internals <http://blog.ezyang.com/2019/05/pytorch-internals/>

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



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

The code (*art-DCGAN*) for the first artificial intelligence artwork ever sold at Christie's auction house (Figure 25) is a modified implementation of DCGAN focused on generative art: <https://github.com/robbiebarrat/art-dcgan>.

- **TensorFlow Magenta.** An open source research project exploring the role of ML in the creative process.⁹²
- **Magenta Studio.** A suite of free music-making tools using machine learning models!⁹³
- **Style Transfer Tutorial** https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/r2/tutorials/generative/style_transfer.ipynb
- **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

MuseNet. Generate Music Using Many Different Instruments and Styles!⁹⁴

Tuning Recurrent Neural Networks with Reinforcement Learning⁹⁵.

Discovering Visual Patterns in Art Collections with Spatially-consistent Feature Learning. Shen et al.⁹⁶.

Deep Multispectral Painting Reproduction via Multi-Layer, Custom-Ink Printing. Shi et al.⁹⁷.

10 AI Macrostrategy: Aligning AGI with Human Interests

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

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

- ❖ **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>.
- ❖ **Ethically Aligned Design, First Edition**⁹⁸. From Principles to Practice <https://ethicsinaction.ieee.org>.

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

⁹²<https://magenta.tensorflow.org>

⁹³<https://magenta.tensorflow.org/studio>

⁹⁴<https://openai.com/blog/musenet/>

⁹⁵<https://magenta.tensorflow.org/2016/11/09/tuning-recurrent-networks-with-reinforcement-learning>

⁹⁶<https://arxiv.org/pdf/1903.02678.pdf>

⁹⁷<http://people.csail.mit.edu/liangs/papers/ToG18.pdf>

⁹⁸<https://standards.ieee.org/content/dam/ieee-standards/standards/web/documents/other/ead1e.pdf>

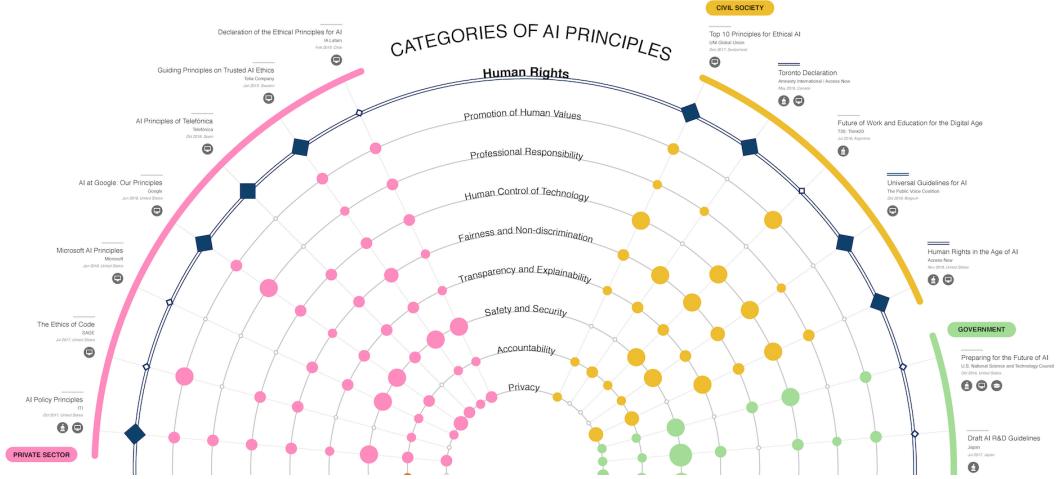


Figure 26: A Map of Ethical and Right-Based Approaches <https://ai-hr.cyber.harvard.edu/primp-viz.html>

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>
- [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>