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

March 12, 2019

ABSTRACT

In order to entrust all sentient beings with the **AI-First** mindset and with the best AI tools to enhance their prosperity and well-being and to inspire those who, with AI, will shape the 21st Century, the **General Secretariat of Montréal.AI** introduces this ***VIP AI 101 CheatSheet*** for All.

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

***K*eywords** AI-First · Artificial Intelligence · Deep Learning · GANs · Intelligent Agent

1 AI-First

TODAY'S ARTIFICIAL INTELLIGENCE IS POWERFUL AND ACCESSIBLE TO ALL. AI opens up a world of new possibilities. To pioneer AI-First innovations advantages: start by exploring how to apply AI in ways never thought of.

2 Getting Started

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

2.1 In the Cloud

Colab [2]. Practice Immediately [3]. Labs[4]: Introduction to Deep Learning (MIT 6.S191)

- Free GPU compute via Colab https://colab.research.google.com/notebooks/welcome.ipynb.

2.2 On a Local Machine

JupyterLab is an interactive development environment for working with notebooks, code and data [5].

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

*Founding Chairman at MONTRÉAL.AI http://www.montreal.ai.
[2] https://medium.com/tensorflow/colab-an-easy-way-to-learn-and-use-tensorflow-d74d1686e309
[3] https://colab.research.google.com/github/GokuMohandas/practicalAI/
[4] https://colab.research.google.com/github/aamini/introtodeeplearning_labs
[5] https://blog.jupyter.org/jupyterlab-is-ready-for-users-5a6f039b8906

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 both[6]. See Figure 1 and Table 1. Introductory course `http://introtodeeplearning.com`.

"*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

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*

"*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

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

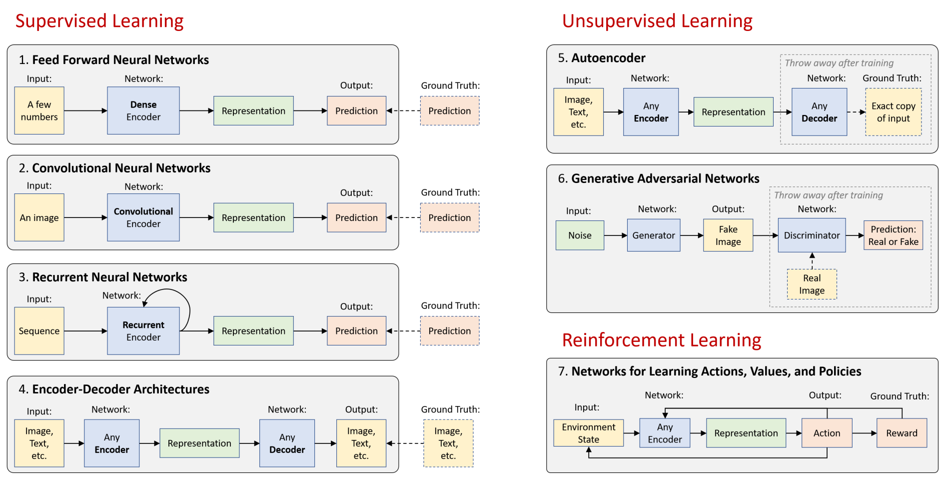


Figure 1: Deep learning can be used in supervised, unsupervised, or RL. Source: Fridman et al. | MIT Deep Learning

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

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 + GPU[7]:

- Infinitely flexible function: *Neural Network* (multiple hidden layers: Deep Learning)[8].
- All-purpose parameter fitting: *Gradient Descent*[9][10].

[6] `https://github.com/lexfridman/mit-deep-learning`
[7] `http://wiki.fast.ai/index.php/Lesson_1_Notes`
[8] `http://neuralnetworksanddeeplearning.com/chap4.html`
[9] `https://github.com/DebPanigrahi/Machine-Learning/blob/master/back_prop.ipynb`
[10] `https://www.jeremyjordan.me/neural-networks-training/`

- Fast and scalable: *GPU*.

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

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)

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 2. Demo `https://ml4a.github.io/demos/convolution/`.

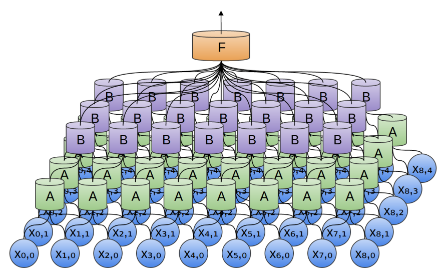


Figure 2: A 2D Convolutional Neural Network. Source: Colah et al., 2014

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[11]. Reading[12].

In images, local combinations of edges form motifs, motifs assemble into parts, and parts form objects[13][14].

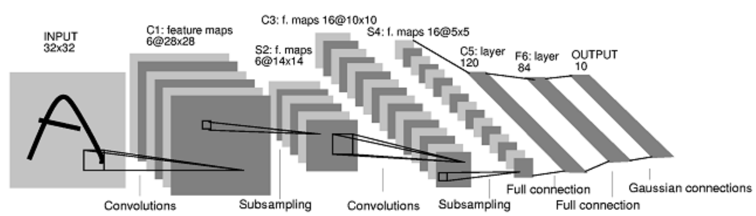


Figure 3: Architecture of LeNet-5, a Convolutional Neural Network. LeCun et al., 1998

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[15]. 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 4.

"*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

[11] `http://cs231n.github.io/convolutional-networks/`
[12] `https://ml4a.github.io/ml4a/convnets/`
[13] `http://yosinski.com/deepvis`
[14] `https://distill.pub/2017/feature-visualization/`
[15] `http://colah.github.io/posts/2015-08-Understanding-LSTMs/`

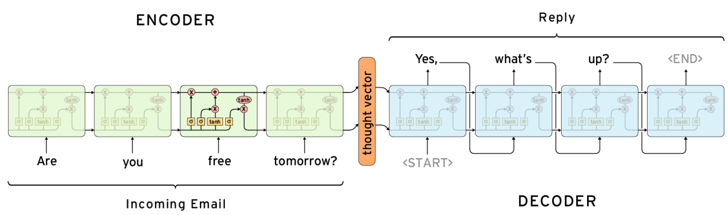


Figure 4: "Long Short-Term-Memory" Network (LSTM). Diagram by Chris Olah

3.4 Unsupervised Learning

Unsupervised learning 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][16].

An embedding is a mapping from discrete objects, such as words, to vectors of real numbers[17].

3.4.1 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].

"*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 5.

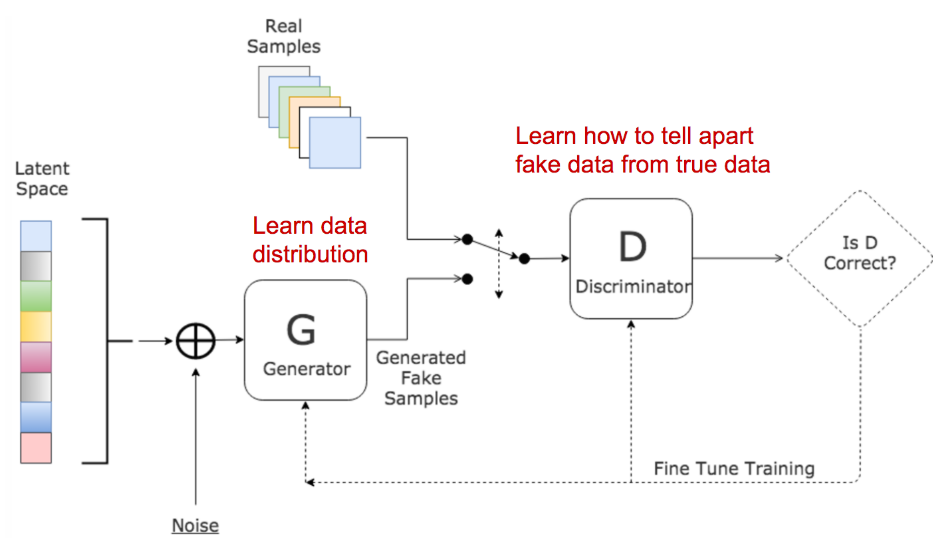


Figure 5: GAN: Neural Networks Architecture Pioneered by Ian Goodfellow at University of Montreal (2014).

GANSynth: Generate high-fidelity audio with GANs! Colab `http://goo.gl/magenta/gansynth-demo`.
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`

[16] https://media.neurips.cc/Conferences/NIPS2018/Slides/Deep_Unsupervised_Learning.pdf
[17] http://projector.tensorflow.org

3.4.2 Variational AutoEncoder

Variational Auto-Encoders (VAEs) are powerful models for learning low-dimensional representations See Figure 5. 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).

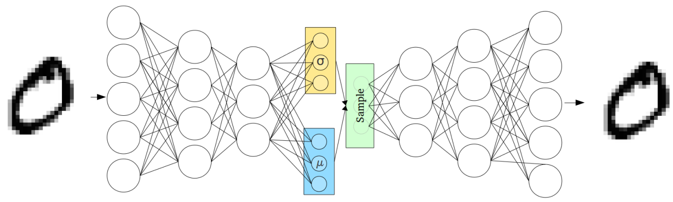


Figure 6: Variational Autoencoders (VAEs): Powerful Generative Models.

Colab[18]: "*Debiasing Facial Detection Systems.*" ***AIEthics***
MusicVAE: Learning latent spaces for musical scores `https://magenta.tensorflow.org/music-vae`.
Slides: A Few Unusual Autoencoders, by Colin Raffel `https://colinraffel.com/talks/vector2018few.pdf`.
Reading: Disentangled VAE's (DeepMind 2016) `https://arxiv.org/abs/1606.05579`.

3.4.3 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)[19]. Blog[20].

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 `https://blog.openai.com/better-language-models/`.

"*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

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`.

4 Autonomous Agents

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.

4.1 Deep Reinforcement Learning

Reinforcement learning (RL) studies how an agent can learn how to achieve goals in a complex, uncertain environment (Figure 7) [5]. Recent superhuman results in many difficult environments combine deep learning with RL (*Deep Reinforcement Learning*). See Figure 8 for a taxonomy of RL algorithms.

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:

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

[18] `https://colab.research.google.com/github/aamini/introtodeeplearning_labs/blob/master/lab2/Part2_debiasing_solution.ipynb`
[19] `https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html`
[20] `https://jalammar.github.io/illustrated-bert/`

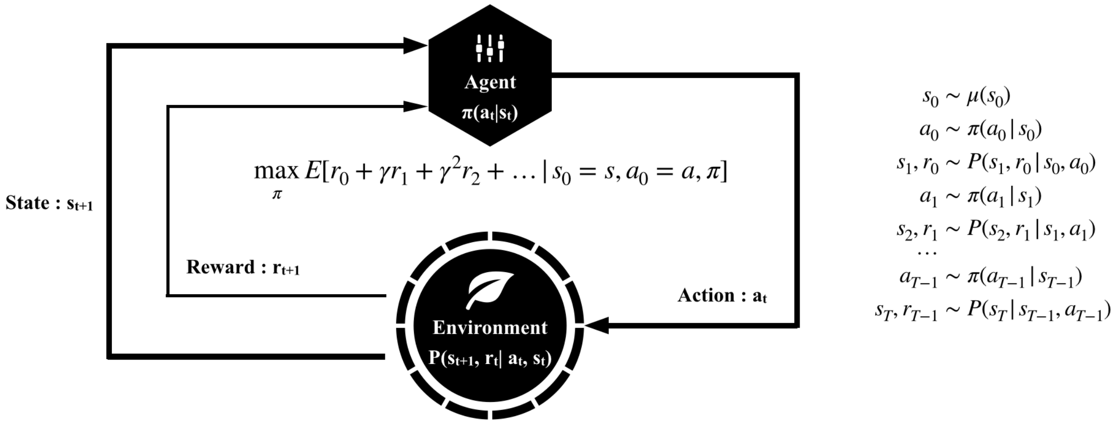


Figure 7: An Agent Interacts with an Environment.

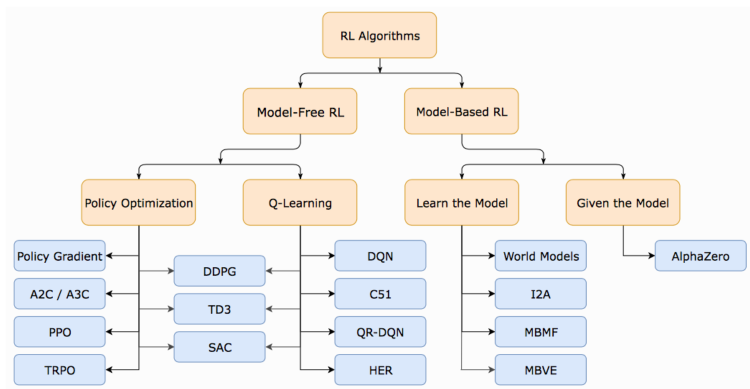


Figure 8: A Taxonomy of RL Algorithms. Source: Spinning Up in Deep RL by Achiam et al. | OpenAI

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

$$Q(s, a) = E[R_t] \tag{2}$$

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

$$\pi^*(s) = argmax_a Q(s, a) \tag{3}$$

- **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`.

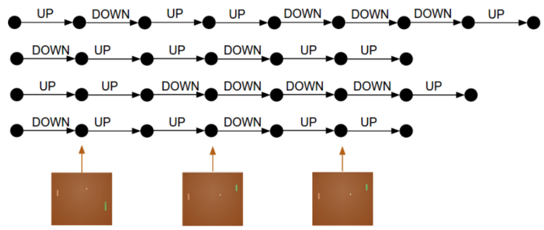
4.1.2 Model-Free RL | Policy-Based

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

$$\tau = (s_0, a_0, r_0, s_1, a_1, r_1, \ldots, s_{T-1}, a_{T-1}, r_{T-1}, s_T) \tag{4}$$

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

$$\nabla_\theta E_\tau[R(\tau)] = E_\tau \left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi(a_t | s_t, \theta) R(\tau) \right] \tag{5}$$



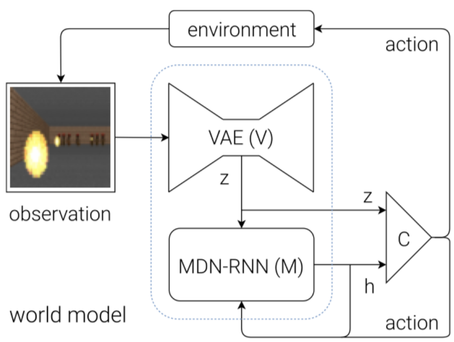

```
function REINFORCE
  Initialize θ
  for episode ~ π_θ
   {s_i, a_i, r_i}_{i=1}^{T-1} ← episode
   for t = 1 to T-1
     ∇ ← ∇_θ log π_θ(a_t|s_t) R_t
     θ ← θ + α∇
  return θ
```

Figure 9: Policy Gradient Directly Optimizes the Policy.

- **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.




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

- **Learn the Model**: *Recurrent World Models Facilitate Policy Evolution* (World Models[21]). 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 9. 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[22].

4.1.4 OpenAI Baselines

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

Colab `https://colab.research.google.com/drive/1KKq9A3dRTq1q6bJmPyFOgg917gQyTjJI`.

4.1.5 Google Dopamine and A Zoo of Agents

Dopamine is a research framework for fast prototyping of reinforcement learning algorithms.[23].

A Zoo of Atari-Playing Agents: Code[24], Blog[25] and Colaboratory notebook `https://colab.research.google.com/github/uber-research/atari-model-zoo/blob/master/colab/AtariZooColabDemo.ipynb`.

[21] `https://worldmodels.github.io`
[22] `https://applied-data.science/static/main/res/alpha_go_zero_cheat_sheet.png`
[23] `https://github.com/google/dopamine`
[24] `https://github.com/uber-research/atari-model-zoo`
[25] `https://eng.uber.com/atari-zoo-deep-reinforcement-learning/`

4.2 Evolution Strategies (ES)

Evolution and neural networks proved a potent combination in nature. 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.[21]

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

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)[20].

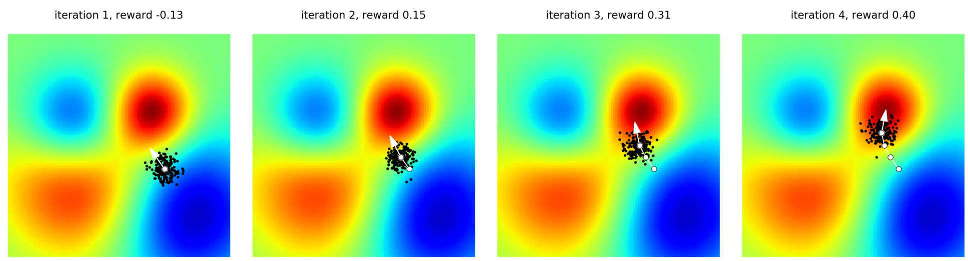


Figure 11: `https://colab.research.google.com/github/karpathy/randomfun/blob/master/es.ipynb`.

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.

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

Demos: ES on CartPole-v1[26] and ES on LunarLanderContinuous-v2[27].
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 world[28].

> "*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 Demo[29] | Code[30]
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`.

[26] `https://colab.research.google.com/drive/1bMZWHdhm-mT9NJENWoVewUks7cGV10go`
[27] `https://colab.research.google.com/drive/1lvyKjFtc_C_8njCKD-MnXEW8LPS2RPr6`
[28] `http://metalearning-symposium.ml`
[29] `https://frpays.github.io/lc0-js/engine.html`
[30] `https://github.com/frpays/lc0-js/`

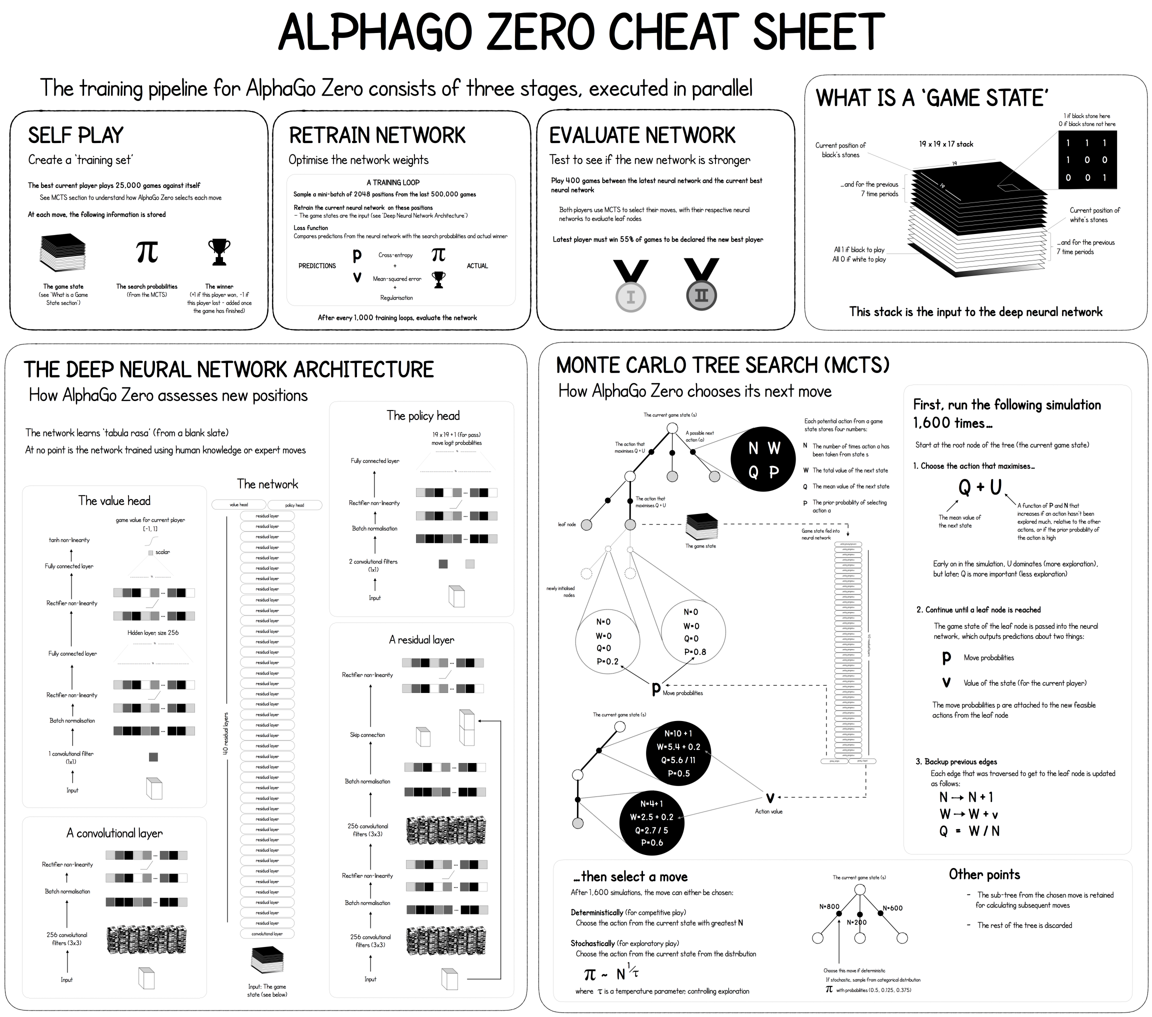


Figure 12: Ref.: `https://applied-data.science/static/main/res/alpha_go_zero_cheat_sheet.png`.

4.4 Deep Meta-Learning

Learning to Learn[16]. 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].

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

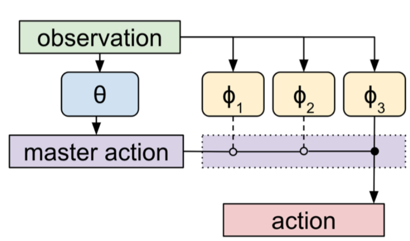


Figure 13: Meta Learning Shared Hierarchies[18] (The Lead Author is in High School!)

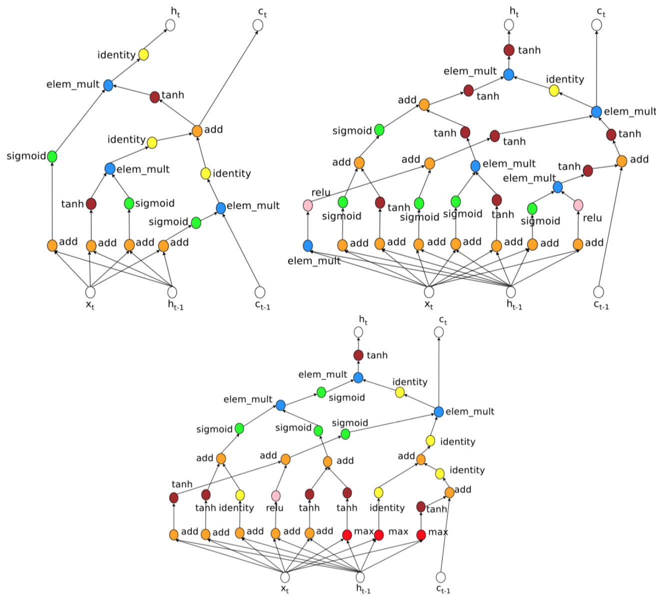


Figure 14: A comparison of the original LSTM cell vs. two new good generated. Top left: LSTM cell. [19]

"*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

"*Causal Reasoning from Meta-reinforcement Learning,*" Dasgupta et al.: `https://arxiv.org/abs/1901.08162`

4.5 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.[24]

Cooperative Agents. Learning to Model Other Minds, by OpenAI[23], is an algorithm which accounts for the fact that other agents are learning too, and discovers self-interested yet collaborative strategies. Also: OpenAI Five[31].

"*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 [32].

"*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. Code[33] and 3D Client[34].

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)

[31] `https://blog.openai.com/openai-five/`
[32] `https://github.com/eugenevinitsky/sequential_social_dilemma_games`
[33] `https://github.com/openai/neural-mmo`
[34] `https://github.com/jsuarez5341/neural-mmo-client`

5.1 OpenAI Gym

The OpenAI Gym `https://gym.openai.com/` (Blog[35] | GitHub[36]) is a toolkit for developing and comparing reinforcement learning algorithms. What makes the gym so great is a common API around environments.

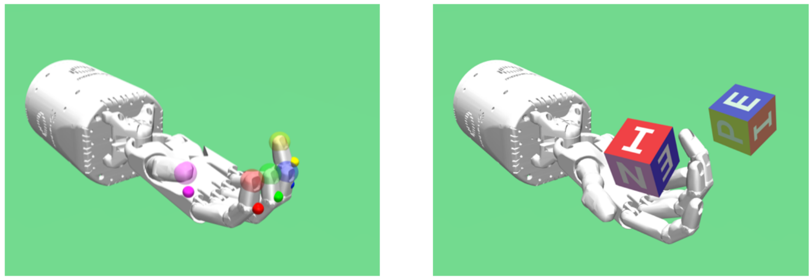


Figure 15: Robotics Environments `https://blog.openai.com/ingredients-for-robotics-research/`

Here's how to create an environment `https://github.com/openai/gym/tree/master/gym/envs`.

Examples: OpenAI Gym Environment for Trading[37].

5.2 DeepMind Lab

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

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`.

5.4 POET: Paired Open-Ended Trailblazer

Diversity is the premier product of evolution. Endlessly generate increasingly complex and diverse learning environments[38]. Open-endedness could generate learning algorithms reaching human-level intelligence[22].

6 Datasets

Google Dataset Search Beta (Blog[39]) `https://toolbox.google.com/datasetsearch`.
TensorFlow Datasets: load a variety of public datasets into TensorFlow programs (Blog[40] | Colab[41]).

7 Hardware

A Full Hardware Guide to Deep Learning, by Tim Dettmers[42]

8 TensorFlow

- TensorFlow 2.0: basic ops, gradients, data preprocessing and augmentation, training and saving. Colab[43].

[35] `https://blog.openai.com/openai-gym-beta/`
[36] `https://github.com/openai/gym`
[37] `https://github.com/hackthemarket/gym-trading`
[38] `https://eng.uber.com/poet-open-ended-deep-learning/`
[39] `https://www.blog.google/products/search/making-it-easier-discover-datasets/`
[40] `https://medium.com/tensorflow/introducing-tensorflow-datasets-c7f01f7e19f3`
[41] `https://colab.research.google.com/github/tensorflow/datasets/blob/master/docs/overview.ipynb`
[42] `http://timdettmers.com/2018/12/16/deep-learning-hardware-guide/`
[43] `https://colab.research.google.com/github/zaidalyafeai/Notebooks/blob/master/TF_2_0.ipynb`

Figure 16: Edge TPU - Dev Board https://coral.withgoogle.com/products/dev-board/

- TensorBoard in Jupyter Notebooks. Colab[44].
- TensorFlow Lite for Microcontrollers[45].

9 AI Macrostrategy: Aligning AGI with Human Interests

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

"*I believe that the answer here is to figure out how to create superintelligent A.I. such that even if – when – it escapes, it is still safe because it is fundamentally on our side because it shares our values. I see no way around this difficult problem.*" — Nick Bostrom

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

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

[45] https://petewarden.com/2019/03/07/launching-tensorflow-lite-for-microcontrollers/

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

[21] 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

[22] 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

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

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