Towards Principled Approaches for Empirical Problems

Yuandong Tian

Research Scientist and Manager

Facebook AI Research
Great Empirical Success of AI

ImageNet classification (Top-5 error)

- 2010: AlexNet, 0.28
- 2011: AlexNet, 0.26
- 2012: Clarifi, 0.16
- 2013: VGG, 0.12
- 2014: SENets, 0.07
- 2015: SENets, 0.036
- 2016: SENets, 0.03
- 2017: SENets, 0.023

https://www.kaggle.com/getting-started/149448
Great Empirical Success of AI

- AlphaGo (2016)
- Chess
- Shogi
- Dota 2
- StarCraft 2
Great Empirical Success of AI

The importance of being on twitter
by Jerome K. Jerome
London, Summer 1897

It is a curious fact that the last remaining form of social life in which the people of London are still interested is Twitter. I was struck with this curious fact when I went on one of my periodical holidays to the sea-side, and found the whole place twittering like a starling-cage. I called it an anomaly, and it is.

I spoke to the sexton, whose cottage, like all sexton’s cottages, is full of antiquities and interesting relics of former centuries. I said to him, "My dear sexton, what does all this twittering mean?" And he replied, "Why, sir, of course it means Twitter." "Ah!" I said, "I know about that. But what is Twitter?"
Will this trend continue?

Exponential Growth

~10^20 operations

https://openai.com/blog/ai-and-compute/
At first, I cannot do parameter sweeping
Then I cannot train the model
Then I cannot do fine-tuning
Then I cannot run one forward pass
Then I cannot even download the model

“..."

Will this trend continue?

At first, I cannot do parameter sweeping
Then I cannot train the model
Then I cannot do fine-tuning
Then I cannot run one forward pass
Then I cannot even download the model

...”

https://openai.com/blog/ai-and-compute/
At first, I cannot do parameter sweeping
Then I cannot train the model
Then I cannot do fine-tuning
Then I cannot run one forward pass
Then I cannot even download the model…

Will this trend continue?

What’s Next?

""

Exponential Growth

facebook Artificial Intelligence

https://openai.com/blog/ai-and-compute/
Is Black-box Model Enough?

Input: This is an apple
Output: “Some Nonlinear Transformation”
Using Black-box Model is tricky

Adversarial samples

Data Poisoning

Interpretability

Stop sign $\rightarrow$ a 45 mph sign

Let’s Check the History

Alchemy

Periodic Table of the Elements

Chemistry
The Black Powder

$$\text{2KNO}_3 + S + 3\text{C} \rightarrow \text{K}_2\text{S} + \text{N}_2 \uparrow + 3\text{CO}_2 \uparrow$$

Best mass ratio.

74.64% : 11.85% : 13.51%

(KNO₃) (S) (C)
# Black Powder Ratio in the History

<table>
<thead>
<tr>
<th></th>
<th>KNO$_3$</th>
<th>S</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Song Dynasty (1044 AD)</td>
<td>50%</td>
<td>25%</td>
<td>25%</td>
</tr>
<tr>
<td>Early Ming Dynasty (~1400 AD)</td>
<td>71.4%</td>
<td>14.3%</td>
<td>14.3%</td>
</tr>
<tr>
<td>Mid Ming Dynasty (~1550 AD)</td>
<td>75.8%</td>
<td>10.6%</td>
<td>13.6%</td>
</tr>
<tr>
<td>Qing Dynasty (1753 AD)</td>
<td>80%</td>
<td>10.51%</td>
<td>9.88%</td>
</tr>
<tr>
<td>Qing Dynasty (1818 AD)</td>
<td>77.8%</td>
<td>9.7%</td>
<td>12.5%</td>
</tr>
<tr>
<td>Qing Dynasty (1839 AD)</td>
<td>74%</td>
<td>11%</td>
<td>15%</td>
</tr>
<tr>
<td><strong>Current Standard</strong></td>
<td><strong>75%</strong></td>
<td><strong>10%</strong></td>
<td><strong>15%</strong></td>
</tr>
</tbody>
</table>

Human Parameter Tuning
Kepler’s laws of planetary motion

Tycho Brahe’s Mars Observations

How many curves can you fit with modern machine learning?
Tycho Brahe’s Mars Observations

The true curve computed from the modern methods

http://www.pafko.com/tycho/observe.html
Will History Repeat Itself?

Observe empirical results/success

Many crazy theories

Some good but imperfect theory

Established common sense

Where are we now?
Theory that matches with Practice

**Linear Regression**

**Convex Function**
Theory that **doesn’t match** with Practice

How can we move forward?
Theory and Practice

How to develop theory?

- New formulation
- New assumptions
- New technical finding
- Beautiful math
- New connections
- New problems
Theory and Practice

How to develop empirical work?

Practice

- Larger dataset
- Analysis of existing approach
- Improve algorithms
- Handle new cases
- SoTA performance
- New models
Theory and Practice

The best research work we could imagine:

Theory
- Demystify existing works
- Good performance
- Guaranteed Improvement
- Solid theoretical foundations
- Understandable trade-off
- Reproducibility

Practice

Super Hard ... But that’s the way to go!
Career Path

Theoretical Understanding of Models and Algorithms

Computer Vision

2006  2008  2013  2015  2020

PhD  Waymo  Facebook AI Research

Reinforcement Learning
The Charm of Games

Complicated long-term strategies.

Realistic Worlds
Game as a Vehicle of AI

- **Infinite supply of fully labeled data**
- **Controllable and replicable**
- **Low cost per sample**
- **Faster than real-time**
- **Less safety and ethical concerns**
- **Complicated dynamics with simple rules.**
How Game AI works

Even with a super-super computer, it is not possible to search the entire space.
How Game AI works

Even with a super-super computer, it is not possible to search the entire space.

Current game situation

Lufei Ruan vs. Yifan Hou (2010)
Alpha-beta Pruning

A good counter move eliminates other choices.

Move order is important!
Monte Carlo Tree Search

Aggregate win rates, and search towards the good nodes.

\[ a_t = \arg \max_a Q(s_t, a) + u(s_t, a) \]

\[ u(s, a) = c_{\text{uct}} P(s, a) \sqrt{\frac{\sum_b N(s, b)}{1 + N(s, a)}} \]
How to model Policy/Value function?

Non-smooth + high-dimensional
Sensitive to situations. One stone changes in Go leads to different game.

Traditional approach

• Many manual steps
• Conflicting parameters, not scalable.
• Need strong domain knowledge.

Deep Learning

• End-to-End training
  • Lots of data, less tuning.
• Minimal domain knowledge.
• Amazing performance
AlphaGo Series

AlphaGo Lee (Mar. 2016)

AlphaGo Master (May. 2017)

AlphaGo Zero (Oct. 2017)

Without Human Knowledge
The Mystery

• Mystery
  • Is the proposed algorithm really universal?
  • Is the bot almighty? Is there any weakness in the trained bot?

• Lack of Ablation Studies
  • What factor is critical for the performance?
  • Is the algorithm robust to random initialization and changes of hyper parameters?
  • Any adversarial samples?

Impressive Results, No code, No model
Demystify existing empirical results
Good performance
Reproducibility

OpenGo project

Yuandong Tian  Jerry Ma*  Qucheng Gong*  Shubho Sengupta*  Zhuoyuan Chen  James Pinkerton  Larry Zitnick

*Equal Contributions

[Y. Tian et al., ELF OpenGo: An Analysis and Open Reimplementation of AlphaZero, ICML 2019]
AlphaGoZero / AlphaZero

Generate Training data

Update Models

Self-Replays

Without human knowledge

[Silver et al, Mastering the game of Go without human knowledge, Nature 2017]
The idea of Self-Play

Stroke left and right (左右互搏)
Generate Self-play Games

Monte Carlo Tree Search with current model $\theta_i$
Update Models

Input features (19x19x17): \((X, Y, X_{-1}, Y_{-1}, \ldots, X_{-7}, Y_{-7}, C)\)

Objective:

\[
J(\theta) = (z - V_\theta)^2 - \pi^T \log \mathbf{p}_\theta + c \|\theta\|^2
\]
AlphaGo Zero Strength

- 3 days version
  - 4.9M Games, 1600 rollouts/move
  - 20 block ResNet
  - Defeat AlphaGo Lee.

- 40 days version
  - 29M Games, 1600 rollouts/move
  - 40 blocks ResNet.
  - Defeat AlphaGo Master by 89:11
ELF OpenGo

- System can be trained with 2000 GPUs in 2 weeks (20 block version)
- Superhuman performance against professional players and strong bots.
- Abundant ablation analysis
- Decoupled design, code reusable for other games.

We open source the code and the pre-trained model for the Go and ML community.
**ELF OpenGo Performance**

**Vs top professional players**

<table>
<thead>
<tr>
<th>Name (rank)</th>
<th>ELO (world rank)</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kim Ji-seok</td>
<td>3590 (#3)</td>
<td>5-0</td>
</tr>
<tr>
<td>Shin Jin-seo</td>
<td>3570 (#5)</td>
<td>5-0</td>
</tr>
<tr>
<td>Park Yeonghun</td>
<td>3481 (#23)</td>
<td>5-0</td>
</tr>
<tr>
<td>Choi Cheolhan</td>
<td>3466 (#30)</td>
<td>5-0</td>
</tr>
</tbody>
</table>

Single GPU, 80k rollouts, 50 seconds
Offer unlimited thinking time for the players

**Vs professional players**

Single GPU, 2k rollouts, 27-0 against Taiwanese pros.

**Vs strong bot (LeelaZero)**

[158603eb, 192x15, Apr. 25, 2018]: 980 wins, 18 losses (98.2%)
Distributed System

Evaluate/Selfplay

Client

Send request (game params)

Client

Receive experiences

Client

Client

Client

Client

Server

Client

Putting AlphaGoZero and AlphaZero into the same framework

AlphaGoZero (more synchronization)

AlphaZero (less synchronization)

Server controls synchronization

Server also does training.

Training Stage of Final Model

Prototype-\(\alpha\) = strong amateur level

Prototype-\(\beta\) = professional level

Prototype = superhuman level
(model against professional players)

A lot of zig-zag in the training process
Overfitting issues

Dip of the value function

- Overestimate white winrate
- Black resigns prematurely
- Black loses many games
- Imbalanced replay buffer

Large replay buffer is the key
Adaptive resign threshold has delays
Ladder Issues

Run a ladder and lost

Run shorter ladder and lost

Doesn’t run ladder

There is only one long path that is correct
Value propagation is really slow.
Did we solve ladder?

No

Why is the model still strong? → It plays alternative moves to avoid these situations.
Why MCTS is so important?

Look-ahead is how new knowledge is created.

On Final Model

White rollouts 2x \rightarrow ~85\% winrate

Black rollouts 2x \rightarrow ~65\% winrate

Training is almost always constrained by model capacity (why 40b > 20b)
Joint Policy Search and Contract Bridge Bidding

Principled Algorithm
Guaranteed Performance
Good Empirical Results

Yuandong Tian  Qucheng Gong  Tina Jiang

[Y. Tian et al., Joint Policy Search for Collaborative Multi-agent Imperfect Information Game, NeurIPS 2020]
When Self-Play Fails?

Training with self-play + A2C get stuck in local minima
An example

Switch to English??
No...she speaks French and might be unhappy...

C'est la vie...

A **unilateral** change of policy doesn’t improve co-operative communication
(many single-agent DRL approach improves by unilateral changes of agent policy)
Communication Game (Incomplete Information)

Player 2 makes the decision without knowing player 1’s action.

(French, French): local Nash Equilibrium +0.5

(English, English): global Nash Equilibrium +1.0

A joint optimization of policy $\sigma(I_1)$ and $\sigma(I_2)$ yields optimal solution.
Another Illustrative Example (Imperfect Information)

Private Card

<table>
<thead>
<tr>
<th>Private card</th>
<th>Alice’s Action</th>
<th>Bob’s Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>♥ A</td>
<td>1</td>
<td>Guess ♥ A</td>
</tr>
<tr>
<td>A</td>
<td>3</td>
<td>Guess A</td>
</tr>
<tr>
<td>--</td>
<td>2</td>
<td>--</td>
</tr>
</tbody>
</table>

What if Alice and Bob never use signal 2, but sending signal 2 has additional rewards?
Optimize Policies in Multiple Infosets

A sparse set of active infosets to be optimized

Policy: \( \sigma(I) \)

Perfect information \( \rightarrow \) A Subtree
Imperfect information \( \rightarrow \) A Graph
Lots of dependencies!
Dependency between policies

A change of $\sigma(I_1, a)$ affects all the reachability of down-stream states and/or infosets, no matter they are active or not.

A trajectory could re-enter into another active set and leave and re-enter again.

The value of an inactive infoset $I_3$ will change since the reachability to $I_3$ changes.

An infoset might contain both affected states and unaffected states.

Is there a good way to track value changes?
Optimize Policies in Multiple Infosets

\[ \sigma \rightarrow \sigma' \]
Current Policy \rightarrow New Policy

\[ \bar{\nu}^\sigma \rightarrow \bar{\nu}^{\sigma'} \]
Current Game Value \rightarrow New Game Value
Policy-change Density

Density \( \rho^{\sigma,\sigma'}(h) = \pi^{\sigma'}(h) \left[ \sum_{a \in A(I)} \sigma'(I, a)\psi^\sigma(ha) - \psi^\sigma(h) \right] \)

Two key properties:

(a) Its summation yields overall value changes

\[ \tilde{\psi}^{\sigma'} - \tilde{\psi}^\sigma = \sum_{h \notin Z} \rho^{\sigma,\sigma'}(h) \]

(b) For regions with the same policy, it vanishes even if the overall reachability changes.

\[ \rho^{\sigma,\sigma'}(h) = 0 \]
Value Changes w.r.t Localized Policy Change

Theorem

$$
\overline{\nu}^{\sigma'} - \overline{\nu}^{\sigma} = \sum_{I \in \mathcal{I}} \sum_{h \in I} \rho^{\sigma,\sigma'}(h)
$$

Overall value changes

All active Infosets where $\sigma' \neq \sigma$

All complete states, doesn’t matter whether their reachability is affected or not
JPS (Joint Policy Search)

1. Initial infosets $I_{cand} = \{I_1\}$
2. Pick $I \in I_{cand}$
3. Pick an action $a$
4. Set $\sigma'(I, b) = \delta(a = b)$
5. Compute $\rho^{\sigma,\sigma'}$
6. Set $I_{cand} = \text{Succ}(I, a)$

Repeat until maximal depth is reached.
Algorithm 1 Joint Policy Search (Tabular form)

1: function JSP-MAIN($\sigma$)
2:     for $i = 1 \ldots T$ do
3:         Compute reachability $\pi^\sigma$ and value $v^\sigma$ under $\sigma$. Pick initial infoсет $I_1$.
4:         $\sigma \leftarrow$ JPS($\sigma$, $\{I_1\}$, 1).
5:     end for
6: end function

7: function JPS($\sigma$, $\mathcal{I}_{\text{cand}}$, $d$) \quad \triangleright \mathcal{I}_{\text{cand}}$: candidate infoсетs
8:     if $d \geq D$ then
9:         return 0.
10:    end if
11:     for $I \in \mathcal{I}_{\text{cand}}$ and $h \in I$ do
12:         Compute $\pi^\sigma'(h)$ by back-tracing $h' \sqsubset h$ until $I(h')$ is active. Otherwise $\pi^\sigma'(h) = \pi^\sigma(h)$.
13:     end for
14:     Compute $J^{\sigma,\sigma'}(I) = \sum_{h \in I} \rho^{\sigma,\sigma'}(h)$ for each $I \in \mathcal{I}_{\text{cand}}$ using Eqn. 5.
15:     for $I \in \mathcal{I}_{\text{cand}}$ and $a \in A(I)$ do
16:         Set $I$ active. Set $\sigma'(I)$ and reachability accordingly Eqn. 6.
17:         Set $r(I, a) = \text{JPS}(\sigma, \text{succ}(I, a), d + 1) + J^{\sigma,\sigma'}(I)$
18:     end for
19:     return $\max(0, \max_{I, a} r(I, a))$ \quad \triangleright \text{Also consider if no infoсет in } \mathcal{I}_{\text{cand}} \text{ is active.}
20: end function
Results on Simple Games

**Definition 1** (Simple Communication Game of length $L$). Consider a game where $s_1 \in \{0, \ldots, 2^L - 1\}$, $a_1 \in A_1 = \{0, 1\}$, $a_2 \in A_2 \in \{0, \ldots, 2^L - 1\}$. $P1$ sends one binary public signal for $L$ times, then $P2$ guess $P1$'s private $s_1$. The reward $r = 1[s_1 = a_2]$ (i.e. 1 if guess right).
Results on Simple Games

**Definition 2** (Simple Bidding Game of size $N$). $P1$ and $P2$ each dealt a private number $s_1, s_2 \sim \text{Uniform}[0, \ldots, N - 1]$. $A = \{\text{Pass}, 2^0, \ldots, 2^k\}$ is an ordered set. The game alternates between $P1$ and $P2$, and $P1$ bids first. The bidding sequence is strictly increasing. The game ends if either player passes, and $r = 2^k$ if $s_1 + s_2 \geq 2^k$ where $k$ is the latest bid. Otherwise the contract fails and $r = 0$. 

(Private = 0)

Alice

Bid = 1

Bid = P

Bob

(Private = 4)
### Performance

<table>
<thead>
<tr>
<th></th>
<th>Comm (Def. 1)</th>
<th>Mini-Hanabi [15]</th>
<th>Simple Bidding (Def. 2)</th>
<th>2SuitBridge (Def. 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$L = 3$</td>
<td>$L = 5$</td>
<td>$L = 6$</td>
<td>$L = 7$</td>
</tr>
<tr>
<td>CFR1k [43]</td>
<td>0.89*</td>
<td>0.85</td>
<td>0.85</td>
<td>0.85</td>
</tr>
<tr>
<td>CFR1k+JPS</td>
<td>1.00*</td>
<td>1.00*</td>
<td>1.00*</td>
<td>1.00*</td>
</tr>
<tr>
<td>A2C [26]</td>
<td>0.60*</td>
<td>0.57</td>
<td>0.51</td>
<td>0.02</td>
</tr>
<tr>
<td>BAD [15]</td>
<td>1.00*</td>
<td>0.88</td>
<td>0.50</td>
<td>0.29</td>
</tr>
<tr>
<td><strong>Best Known</strong></td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>#States</td>
<td>633</td>
<td>34785</td>
<td>270273</td>
<td>2129793</td>
</tr>
<tr>
<td>#Infosets</td>
<td>129</td>
<td>2049</td>
<td>8193</td>
<td>32769</td>
</tr>
</tbody>
</table>
Contract Bridge

- 25 million US players
- 100 years of history
- Incomplete Information
- Collaborative + Competitive
- Large State Space ($5.4 \times 10^{28}$)
Bridge Bidding

Player only knows the private cards

Sequences of non-decreasing bids

The last bid is the contract

Fundamental Trade-off: bid high via efficient communication, but not too much!
Evaluation against SoTA software (1000 games)

<table>
<thead>
<tr>
<th>Methods</th>
<th>Vs. WBridge5 (IMPs/board)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Previous SoTA (Rong et al, 2019)</td>
<td>+ 0.25 (on 64 games)</td>
</tr>
<tr>
<td>Our A2C baseline</td>
<td>+ 0.29 ± 0.22</td>
</tr>
<tr>
<td>1% JPS (2 days)</td>
<td>+ 0.44 ± 0.20</td>
</tr>
<tr>
<td>5% JPS (2 days)</td>
<td>+ 0.37 ± 0.19</td>
</tr>
<tr>
<td>1% JPS (14 days)</td>
<td>+ 0.63 ± 0.20</td>
</tr>
</tbody>
</table>

## Bidding Visualization

<table>
<thead>
<tr>
<th>Opening bids</th>
<th>Ours</th>
<th>SAYC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1♣</td>
<td>10+ HCP</td>
<td>12+ HCP, 3+♣</td>
</tr>
<tr>
<td>1♦</td>
<td>8-18 HCP, &lt;4 ♥, &lt;4 ♠</td>
<td>12+ HCP, 3+♦</td>
</tr>
<tr>
<td>1♥</td>
<td>4-16 HCP, 4-6♥</td>
<td>12+ HCP, 5+♥</td>
</tr>
<tr>
<td>1♠</td>
<td>4-16 HCP, 4-6♠</td>
<td>12+ HCP, 5+♠</td>
</tr>
<tr>
<td>1NT</td>
<td>12-17 HCP, bal</td>
<td>15-17 HCP, bal</td>
</tr>
<tr>
<td>2♣</td>
<td>6-13 HCP, 5+♣</td>
<td>22+ HCP</td>
</tr>
<tr>
<td>2♦</td>
<td>6-13 HCP, 5+♦</td>
<td>5-11 HCP, 6+♦</td>
</tr>
<tr>
<td>2♥</td>
<td>8-15 HCP, 5+♥</td>
<td>5-11 HCP, 6+♥</td>
</tr>
<tr>
<td>2♠</td>
<td>8-15 HCP, 5+♠</td>
<td>5-11 HCP, 6+♠</td>
</tr>
</tbody>
</table>
Learning Action Space in Monte Carlo Tree Search

Good Empirical Performance
No theory yet

[L. Wang et al, Sample-Efficient Neural Architecture Search by Learning Action Space, arXiv]
[L. Wang et al, Learning Search Space Partition for Black-box Optimization using Monte Carlo Tree Search, NeurIPS 2020]

Linnan Wang¹  Saining Xie²  Teng Li²  Rodrigo Fonseca¹  Yuandong Tian²

¹Brown University, ²Facebook AI Research
What else can Monte Carlo Tree Search (MCTS) be used for?

(Non-Convex) Optimization
Motivating Examples in Architecture Search

Depth = \{1, 2, 3, 4, 5\}
Channels = \{32, 64\}
KernelSize = \{3x3, 5x5\}

1364 networks.

Action space

Sequential = \{ add a layer, set K, set C \}
Global = \{ Set depth, set all K, set all C \}

Global is better!
Empty network

Action: Add one conv layer

Action: More filters

Conv

Conv
ReLU

Conv
ReLU

Conv
ReLU
Conv
ReLU

Full network

Evaluation
Empty network

Action: Add one conv layer

Action: More filters

Full network

Current approaches

Ensemble network design space

Action = Learnable Constraint

Entire network design space

Good models

Bad models

Subset of design space (lots of models)

Value function

Our approaches

Conv

Conv

ReLU

Conv

ReLU

Conv

ReLU

Evaluation

Good models

Bad models

Current approaches

Our approaches

70
Learn action space

Current node whose action space is learned

Action 1="right"

Action 1="left"
Approach

Monte Carlo Tree Search (MCTS)

Value sampled from the current subset of networks (E.g., from truth table)

(a) Search using current action space until a fixed #rollouts are used.

(b) Train the action space.

<table>
<thead>
<tr>
<th>Network Hyperparameters</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>(filter=2, depth=5)</td>
<td>85%</td>
</tr>
<tr>
<td>(filter=3, depth=7)</td>
<td>92%</td>
</tr>
<tr>
<td>(filter=3, depth=2)</td>
<td>30%</td>
</tr>
</tbody>
</table>
Performance

NASBench-101 (CIFAR-10, 420k models, NASNet Search Space)

Each curve is repeated 100 times. We randomly pick 2k models to initialize.
Performance

Customized dataset: ConvNet-60K (CIFAR-10, VGG style models)
Performance

Customized dataset: LSTM-10K (PTB)
### Open Domain

**CIFAR-10**

(NASNet style architecture)

<table>
<thead>
<tr>
<th>Model</th>
<th>Using ImageNet</th>
<th>Params</th>
<th>Top1 err</th>
<th>M</th>
<th>GPU days</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>search based methods</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NASNet-A+c/o [22]</td>
<td>X</td>
<td>3.3 M</td>
<td>2.65</td>
<td>20000</td>
<td>2000</td>
</tr>
<tr>
<td>AmoebaNet-B+c/o [10]</td>
<td>X</td>
<td>2.8 M</td>
<td>2.55±0.05</td>
<td>27000</td>
<td>3150</td>
</tr>
<tr>
<td>PNASNet-5 [29]</td>
<td>X</td>
<td>3.2 M</td>
<td>3.41±0.09</td>
<td>1160</td>
<td>225</td>
</tr>
<tr>
<td>NAO+c/o [30]</td>
<td>X</td>
<td>128.0 M</td>
<td>2.11</td>
<td>1000</td>
<td>200</td>
</tr>
<tr>
<td>AmoebaNet-B+c/o</td>
<td>X</td>
<td>34.9 M</td>
<td>2.13±0.04</td>
<td>27000</td>
<td>3150</td>
</tr>
<tr>
<td>EfficientNet-B7</td>
<td>√</td>
<td>64M</td>
<td>1.01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BiT-M</td>
<td>√</td>
<td>60M</td>
<td>1.09</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>LaNet+c/o</strong></td>
<td>X</td>
<td>3.2 M</td>
<td><strong>1.63±0.05</strong></td>
<td>800</td>
<td>150</td>
</tr>
<tr>
<td><strong>LaNet+c/o</strong></td>
<td>X</td>
<td>44.1 M</td>
<td><strong>0.99±0.02</strong></td>
<td>800</td>
<td>150</td>
</tr>
<tr>
<td><strong>one-shot NAS based methods</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ENAS+c/o [18]</td>
<td>X</td>
<td>4.6 M</td>
<td>2.89</td>
<td>-</td>
<td>0.45</td>
</tr>
<tr>
<td>DARTS+c/o [20]</td>
<td>X</td>
<td>3.3 M</td>
<td>2.76±0.09</td>
<td>-</td>
<td>1.5</td>
</tr>
<tr>
<td>BayesNAS+c/o [31]</td>
<td>X</td>
<td>3.4 M</td>
<td>2.81±0.04</td>
<td>-</td>
<td>0.2</td>
</tr>
<tr>
<td>ASNG-NAS+c/o [32]</td>
<td>X</td>
<td>3.9 M</td>
<td>2.83±0.14</td>
<td>-</td>
<td>0.11</td>
</tr>
<tr>
<td>XNAS+c/o [33]</td>
<td>X</td>
<td>3.7 M</td>
<td>1.81</td>
<td>-</td>
<td>0.3</td>
</tr>
<tr>
<td>oneshot-LaNet+c/o</td>
<td>X</td>
<td>3.6 M</td>
<td><strong>1.68±0.06</strong></td>
<td>-</td>
<td>3</td>
</tr>
<tr>
<td>oneshot-LaNet+c/o</td>
<td>X</td>
<td>45.3 M</td>
<td><strong>1.2±0.03</strong></td>
<td>-</td>
<td>3</td>
</tr>
</tbody>
</table>

M: number of samples selected.
## Open Domain

**ImageNet (mobile setting, Flop < 600M)**

<table>
<thead>
<tr>
<th>Model</th>
<th>FLOPs</th>
<th>Params</th>
<th>top1 / top5 err</th>
</tr>
</thead>
<tbody>
<tr>
<td>NASNet-A (Zoph et al. (2018))</td>
<td>564M</td>
<td>5.3 M</td>
<td>26.0 / 8.4</td>
</tr>
<tr>
<td>NASNet-B (Zoph et al. (2018))</td>
<td>488M</td>
<td>5.3 M</td>
<td>27.2 / 8.7</td>
</tr>
<tr>
<td>NASNet-C (Zoph et al. (2018))</td>
<td>558M</td>
<td>4.9 M</td>
<td>27.5 / 9.0</td>
</tr>
<tr>
<td>AmoebaNet-A (Real et al. (2018))</td>
<td>555M</td>
<td>5.1 M</td>
<td>25.5 / 8.0</td>
</tr>
<tr>
<td>AmoebaNet-B (Real et al. (2018))</td>
<td>555M</td>
<td>5.3 M</td>
<td>26.0 / 8.5</td>
</tr>
<tr>
<td>AmoebaNet-C (Real et al. (2018))</td>
<td>570M</td>
<td>6.4 M</td>
<td><strong>24.3 / 7.6</strong></td>
</tr>
<tr>
<td>PNASNet-5 (Liu et al. (2018a))</td>
<td>588M</td>
<td>5.1 M</td>
<td>25.8 / 8.1</td>
</tr>
<tr>
<td>DARTS (Liu et al. (2018b))</td>
<td>574M</td>
<td>4.7 M</td>
<td>26.7 / 8.7</td>
</tr>
<tr>
<td>FBNet-C (Wu et al. (2018))</td>
<td>375M</td>
<td>5.5 M</td>
<td>25.1 / -</td>
</tr>
<tr>
<td>RandWire-WS (Xie et al. (2019))</td>
<td>583M</td>
<td>5.6 M</td>
<td>25.3 / 7.8</td>
</tr>
<tr>
<td>BayesNAS (Zhou et al. (2019))</td>
<td>-</td>
<td>3.9 M</td>
<td>26.5 / 8.9</td>
</tr>
<tr>
<td>LaNet</td>
<td>570M</td>
<td>5.1 M</td>
<td><strong>25.0 / 7.7</strong></td>
</tr>
</tbody>
</table>
Black-Box Optimization (LaMCTS)

Nonlinear boundary learnt by SVM.

Build local models.

La-MCTS as a meta method

- Ackley-20d
  - Solver: using LAMCTS
  - TuRBO: yes
  - BO: no

- Ackley-100d
  - Solver: using LAMCTS
  - TuRBO: yes
  - BO: no

- Rosenbrock-20d
  - Solver: using LAMCTS
  - TuRBO: yes
  - BO: no

- Rosenbrock-100d
  - Solver: using LAMCTS
  - TuRBO: yes
  - BO: no
Optimizing linear policy for Mujoco tasks

(a) Swimmer, #params = 16
(b) Hopper, #params = 33
(c) Walker-2d, #params = 102
(d) Half-Cheetah, #params = 102
(e) Ant, #params = 888
(f) Humanoid, #params = 6392
TODO: A theory is needed ...
Principled framework
Demystify existing work

A theoretical framework that explains
1. Why self-supervised learning with deep ReLU models works
2. Why a good representation is learned without supervision
3. Why BYOL doesn’t need negative samples

Understand Deep ReLU Models

[Y. Tian., Student Specialization in Deep ReLU Networks With Finite Width and Input Dimension, ICML 2020]
Self-supervised Learning

\[ \mathbf{x} \sim p(\cdot) \]

\[ \mathbf{x}_1, \mathbf{x}_2 \sim p_{\text{aug}}(\cdot | \mathbf{x}) \]

Data Augmentation

Online \( \mathcal{W}_1 \)

Predictor

Target \( \mathcal{W}_2 \)

Loss
Similarity with Teacher Student Setting

\[ x \sim p(\cdot) \]

Student \( \mathcal{W}_1 \)

Teacher \( \mathcal{W}_2 = \mathcal{W}^* \)

\[ r := \| f_{1,L} - f_{2,L} \|^2 \]

The mathematical framework is similar!

[Y. Tian, Student Specialization in Deep ReLU Networks With Finite Width and Input Dimension, ICML 2020]
## Compare with Teacher-Student Setting

<table>
<thead>
<tr>
<th>Teacher-Student Setting</th>
<th>SimCLR Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Training Setup</strong></td>
<td>Teacher is fixed and assumed to be optimal $\mathcal{W}^*$.</td>
</tr>
<tr>
<td></td>
<td>Teacher and student are both under training.</td>
</tr>
<tr>
<td><strong>Loss function</strong></td>
<td>L2 loss</td>
</tr>
<tr>
<td></td>
<td>Contrastive Loss</td>
</tr>
<tr>
<td><strong>Data Augmentation</strong></td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Yes (and critical)</td>
</tr>
<tr>
<td><strong>Architectures</strong></td>
<td>Same architecture for Teacher and Student</td>
</tr>
<tr>
<td></td>
<td>Same architecture for the two networks</td>
</tr>
</tbody>
</table>
SimCLR Setting

\[ \mathbf{x} \sim p(\cdot) \]

\[ \mathbf{x}_1, \mathbf{x}_2 \sim p_{aug}(\cdot | \mathbf{x}) \]

Data Augmentation

Multi-layer ReLU network \( \mathcal{W} \)

Contrastive Loss

If \(|u| = |v| = 1\), then the formulation is the same as SimCLR’s formulation since 
\[ -r = -|u - v|^2 = 2\text{sim}(u, v) - 2 \]
The Covariance Operator

Connection

\[ K_l(x) := f_{l-1}(x) \otimes J_l^T(x) \]
\( \otimes: \) Kronecker Product

Augment-Average Connection

\[ \tilde{K}_l(x) := \mathbb{E}_{x' \sim p_{aug}(\cdot|x)}[K_l(x')] \]

Weight Update for SimCLR at layer \( l \):

\[
W_l(t + 1) = W_l(t) + \alpha \Delta W_l(t)
\]

\[
\text{vec}(\Delta W_l(t)) = \beta \nabla_x [\tilde{K}_l(x)] \text{vec}(W_l(t))
\]

Learning rate

Positive number related to Contrastive loss
What does it mean?

The Covariance Operator \( \nabla_x [\bar{K}_l(x; \mathcal{W}(t))] \)

- Always PSD at any stage of training
- Weight at each layer undergoes a PSD transformation
- Strong eigen mode leads strong weight growth along that direction

What are the strong eigen models in the covariance operator?

To understand that, we need a generative model of the data.
Using Generative Models to understand Covariance Operator

\[ z_0: \] Class (sample) label
\[ z': \] Nuisance Transformations given by Data Augmentation
One-layer one-neuron example

Two objects \textbf{11} and \textbf{101} translating in 1D space

Nuisance $z'$

$z_0 = 1$

$z_0 = 2$

$x(z_0, z')$

$\nabla_{z_0} \left[ \overline{K}(z_0) \right] = \frac{1}{4d^2} uu^\top$

$u := x_{11} + x_{00} - x_{01} - x_{10}$

Linear neuron: Nothing is learned.

ReLU neuron: Enforce what is initialized!

Feature to represent pattern 10
A two-layer example

Augment-Average Connection for both layers:

\[ \tilde{K}_1(z) = [w_{2,1}u_1, \ldots, w_{2,n_1}u_{n_1}] \quad \tilde{K}_2(z) = [w_{1,1}^Tu_1, \ldots, w_{1,n_1}^Tu_{n_1}] \]

where \( u_j(z) := \mathbb{E}_{z' \mid z}[x(z, z') \mathbb{I}(w_{1,j}^Tx(z, z') \geq 0)] \)

**Theorem 4.** If \( \text{Cov}_z[u_j, u_k] = 0 \) for \( j \neq k \), then the time derivative of \( w_{2,j} \) and \( w_{1,j} \) satisfies:

\[ \dot{w}_{2,j} = w_{2,j}w_{1,j}^T A_j w_{1,j}, \quad \dot{w}_{1,j} = w_{2,j}^2 A_j w_{1,j}, \]

where \( A_j := \nabla_z[u_j(z)] \).

Weights of two layer are ***enforcing*** each other
Hierarchical Latent Tree Models (HLTM)

Nuisance latent $z'$

$$\mathbb{P}(z_\nu|z_\mu)$$

Hierarchical Latent Tree Model (HLTM)

$$x(z_0, z')$$

Deep ReLU networks
BYOL Setting

\[ \mathbf{x} \sim p(\cdot) \]

\[ \mathbf{x}_1, \mathbf{x}_2 \sim p_{\text{aug}}(\cdot|\mathbf{x}) \]

Data Augmentation

Multi-layer ReLU network \( \mathbf{w} \)

predictor

\[ (\text{normalized}) \]

L2 Loss

No Negative Pairs!!!

[J. Grill et al, Bootstrap your own latent: A new approach to self-supervised Learning, arXiv]
## BYOL Setting

<table>
<thead>
<tr>
<th>SimCLR Setting</th>
<th>BYOL Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Loss function</strong></td>
<td>Contrastive Loss</td>
</tr>
<tr>
<td><strong>Architectures</strong></td>
<td>Symmetric $\mathcal{W}_1 = \mathcal{W}_2 = \mathcal{W}$</td>
</tr>
<tr>
<td><strong>BatchNorm in predictor/projector</strong></td>
<td>Optional</td>
</tr>
</tbody>
</table>

**Why BYOL doesn’t need contrastive loss?**

**Why BYOL needs an extra predictor?**

**Why BYOL needs to have BN in predictor/projector to work?**
BYOL Setting (Top-1 Performance in STL-10)

Using Predictor is critical

<table>
<thead>
<tr>
<th></th>
<th>EMA</th>
<th>BN</th>
<th>EMA, BN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>38.7 ± 0.6</td>
<td>39.3 ± 0.9</td>
<td>33.0 ± 0.3</td>
</tr>
</tbody>
</table>

BN is critical

<table>
<thead>
<tr>
<th></th>
<th>P</th>
<th>P, EMA</th>
<th>P, BN</th>
<th>P, EMA, BN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>39.5 ± 3.1</td>
<td>44.4 ± 3.2</td>
<td>63.6 ± 1.06</td>
<td><strong>78.1 ± 0.3</strong></td>
</tr>
</tbody>
</table>
How to analyze BatchNorm?

Zero-mean property.
After BN, Backpropagated Gradient is zero-mean in each minibatch:

\[
\tilde{g}_l^i := g_l^i - \frac{1}{|B|} \sum_{i \in B} g_l^i = g_l^i - \bar{g}_l
\]
Zero-mean Gradient matters.

Ablation Study of Batch components

<table>
<thead>
<tr>
<th></th>
<th>-</th>
<th>$\mu$</th>
<th>$\sigma$</th>
<th>$\mu, \sigma$</th>
<th>$\mu^\dagger$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>43.9 ± 4.2</td>
<td>64.8 ± 0.6</td>
<td>72.2 ± 0.9</td>
<td>78.1 ± 0.3</td>
<td>44.2 ± 7.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>$\sigma^\dagger$</th>
<th>$\mu^\dagger, \sigma$</th>
<th>$\mu, \sigma^\dagger$</th>
<th>$\mu^\dagger, \sigma^\dagger$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>54.2 ± 0.6</td>
<td>48.3 ± 2.7</td>
<td>76.3 ± 0.4</td>
<td>47.0 ± 8.1</td>
</tr>
</tbody>
</table>

\[ x = x - x.\text{mean}(0) \quad \text{or} \quad x = x - x.\text{mean}(0).\text{detach()} \]
\[ x = x / x.\text{std}(0) \quad \text{or} \quad x = x / x.\text{std}(0).\text{detach()} \]
Explanation with the Framework

\[
\text{vec}(\Delta W_l) = \text{vec}(\Delta W_l)_{\text{sym}} - \mathbb{E}_x \left\{ \bar{K}_l(x) \left[ \bar{K}_l^T(x) \text{vec}(W_l) - \bar{K}_l^T(x; \mathcal{W}') \text{vec}(W_l') \right] \right\}
\]

Without BN or \( \mathcal{W} = \mathcal{W}' \)

\[
\text{vec}(\tilde{\Delta W}_l) = \text{vec}(\Delta W_l)_{\text{sym}} - \nabla_x \left[ \bar{K}_l(x) \right] \text{vec}(W_l) + \text{Cov}_x \left[ \bar{K}_l(x), \bar{K}_l(x; \mathcal{W}') \right] \text{vec}(W_l')
\]

With BN and \( \mathcal{W} \neq \mathcal{W}' \)

*Some assumption is need to get to here, see paper for the details.

\[
\text{vec}(\Delta W_l)_{\text{sym}} = -\mathbb{E}_x \left\{ \nabla_x' [K_l(x') \right] \text{vec}(W_l)
\]
Why BatchNorm and Predictor matters

\[-\nabla_{\mathbf{x}} \left[ \tilde{K}_l(\mathbf{x}; \mathcal{W}) \right] \text{vec}(W_l) + \text{Cov}_{\mathbf{x}} \left[ \tilde{K}_l(\mathbf{x}; \mathcal{W}), \tilde{K}_l(\mathbf{x}; \mathcal{W}') \right] \text{vec}(W'_l)\]

**Negated** covariance operator

**Approximate** covariance operator

Small when there is a predictor in $\mathcal{W}$ with small Jacobian
Reinitializing Predictors Works

Table 5: Top-1 performance of BYOL using reinitialization of the predictor every $T$ epochs.

<table>
<thead>
<tr>
<th></th>
<th>Original BYOL</th>
<th>ReInit $T = 5$</th>
<th>ReInit $T = 10$</th>
<th>ReInit $T = 20$</th>
</tr>
</thead>
<tbody>
<tr>
<td>STL-10 (100 epochs)</td>
<td>78.1</td>
<td>78.6</td>
<td><strong>79.1</strong></td>
<td>79.0</td>
</tr>
<tr>
<td>ImageNet (60 epochs)</td>
<td>60.9</td>
<td>61.9</td>
<td><strong>62.4</strong></td>
<td><strong>62.4</strong></td>
</tr>
</tbody>
</table>

The predictor is not necessarily “optimal” as suggested in the original BYOL paper.
Homework

• What’s the best mass ratio in Black powder?

• Is that possible to enumerate all possible states in a game like Go?

• How does AlphaZero work? Does AlphaZero use human knowledge?

• Explain how Monte Carlo Tree Search works?

• Explain how Alpha Beta Pruning works?

• Why do we want to open the black-box for deep models?
Thanks!