目次：

Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model

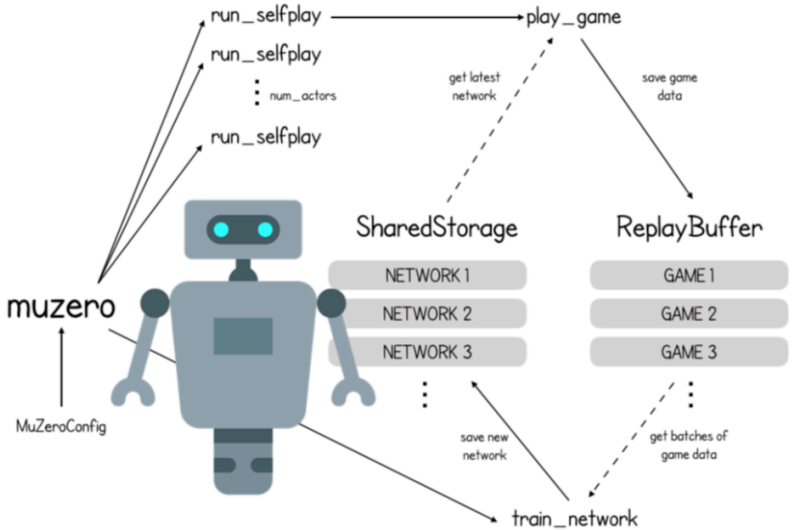
（２０１９）著者：Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, David Silver

概要

AIが囲碁・チェス・将棋などのボードゲームおよびゲームを学習、マスターし、ボードゲームにおいては世界チャンピオンに圧勝している。ゲームにおいても多くを人間以上のレベルに達しており、今後は実世界への応用が期待される。

名前はMuZero。

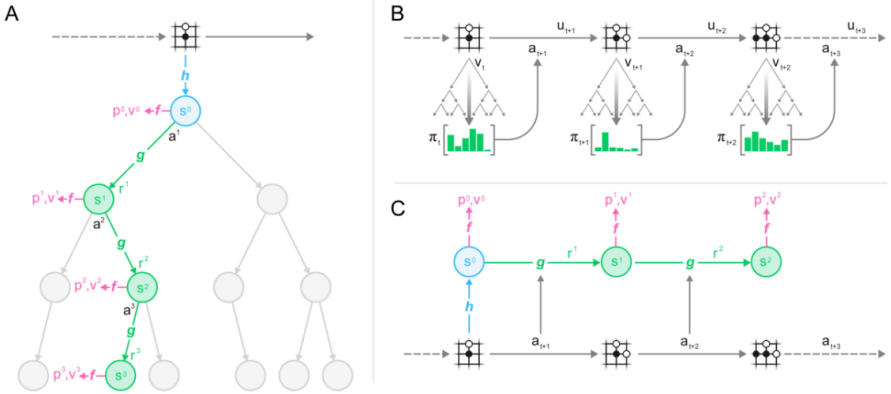
◇MuZeroの構造



先行研究：AlphaZeroからの進化

・表現関数(h)と遷移関数(g)をニューラルネットワーク(ResNet)で表現。
(全部で３つのニューラルネットを学習)

・ニューラルネットで表現したことでAIにルールを教えなくとも学習過程で自らルールを学習。
(ただし勝敗引き分けは教える必要あり)



次に読むべき文献

・Identity Mappings in Deep Residual Networks

・Single-Player Monte-Carlo Tree Search

・Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments

・Ray- A Distributed Framework for Emerging AI Applications

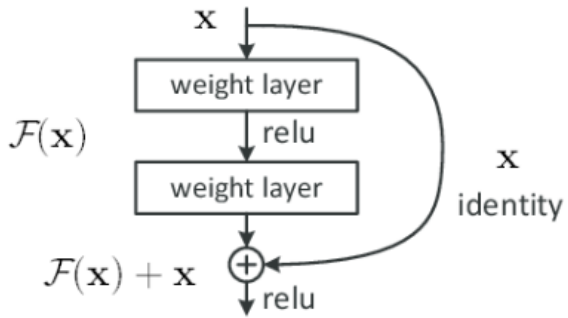
Deep Residual Learning for Image Recognition

（２０１５）著者：Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun

概要

より深層なニューラルネットワーク（最大１０００層以上）を構築することを可能にした論文。もともと層を深くすれば性能が悪化すると言われていたが、ResNetでは「ある層で求める最適な出力を学習するのではなく、層の入力を参照した残差関数を学習する」ことで最適化しやすくした。

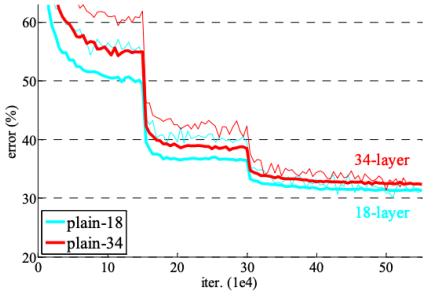
◇ResNetの構造

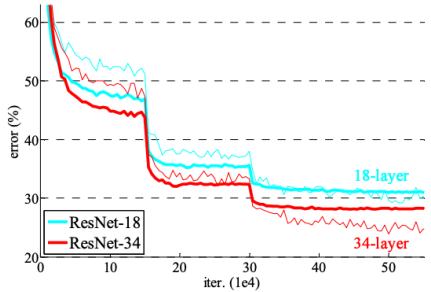


求める写像をH(x)、入力xとの残差をF(x)とするとF(x)：H(x) - xで表すことができ元の画像はF(x) + xで表すことができる。
ショートカットさせているだけなのでパラメータが増えず計算が効率的に行える。
x層を足すことで勾配消失が起こりづらい形式で学習ができる。
深いネットワークを構成する場合はブロックを３層にする。

実装による検証結果

ImageNetを使ったPlainとResNetでの比較。18層と34層で実験しており、より深い34層においてはResNetの性能が高いことがわかる。その他複数のデータで実験しているがいずれも高性能が出ている。





議論

・構造は現状が最良であるか。
・最良な最適化手法は。
・学習時間がかなりかかる。

次に読むべき文献

・Identity Mappings in Deep Residual Networks

Identity Mappings in Deep Residual Networks

（２０１６）著者：Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun

概要

前述のDeep Residual Learning for Image Recognition の派生研究。

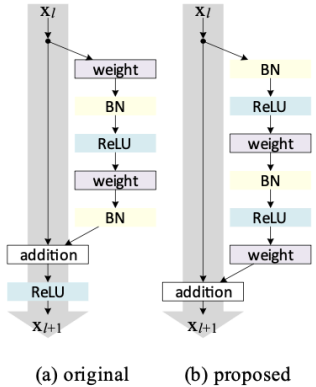
ResNetにおけるモデルの構造をどうすれば最善になるかの議論。

以下４つの考察を記載

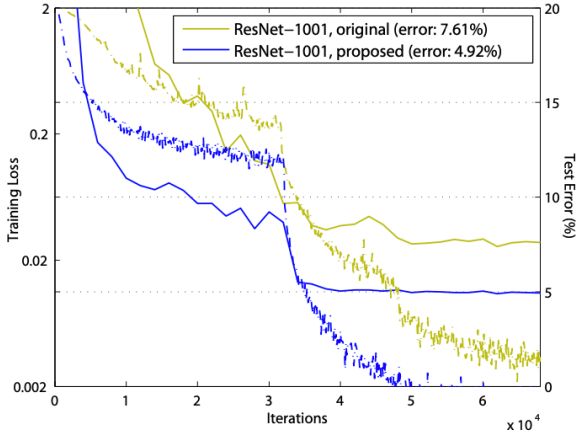
考察①

ResNetブロック内の順番変更にによるlossの減少

(a)のoriginalはlossが7.61%に対し、(b)proposedは4.92%と大幅に改善。



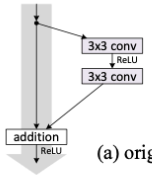
(a) original (b) proposed



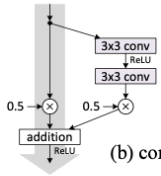
考察②

ResNetブロック内に様々なショートカットコネクトを適用。

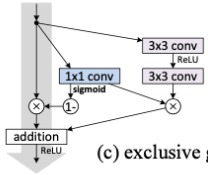
ここでの結果は、ショートカットの乗算操作（スケーリング、ゲーティング、1×1畳み込み、ドロップアウト）は情報の伝播を妨げ、最適化の問題になる可能性があることが示されている。



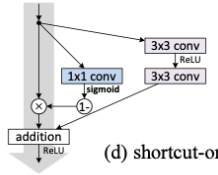
(a) original



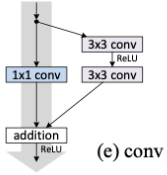
(b) constant scaling



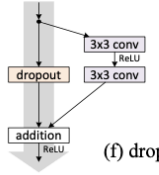
(c) exclusive gating



(d) shortcut-only gating



(e) conv shortcut



(f) dropout shortcut

Table 1. Classification error on the CIFAR-10 test set using ResNet-110 [1], with different types of shortcut connections applied to all Residual Units. We report "fail" when the test error is higher than 20%.

case	Fig.	on shortcut	on $\mathcal{F}$	error (%)	remark
original [1]	Fig. 2(a)	1	1	**6.61**	
constant scaling	Fig. 2(b)	0	1	fail	This is a plain net
		0.5	1	fail	
		0.5	0.5	12.35	frozen gating
exclusive gating	Fig. 2(c)	$1-g(\mathbf{x})$	$g(\mathbf{x})$	fail	init b_g=0 to −5
		$1-g(\mathbf{x})$	$g(\mathbf{x})$	8.70	init b_g=-6
		$1-g(\mathbf{x})$	$g(\mathbf{x})$	9.81	init b_g=-7
shortcut-only gating	Fig. 2(d)	$1-g(\mathbf{x})$	1	12.86	init b_g=0
		$1-g(\mathbf{x})$	1	6.91	init b_g=-6
1×1 conv shortcut	Fig. 2(e)	1×1 conv	1	12.22	
dropout shortcut	Fig. 2(f)	dropout 0.5	1	fail	

考察③
いろいろな順番を試行
結果：BN→Relu→weight→BN→Relu→weight→ additionの順番が最善。

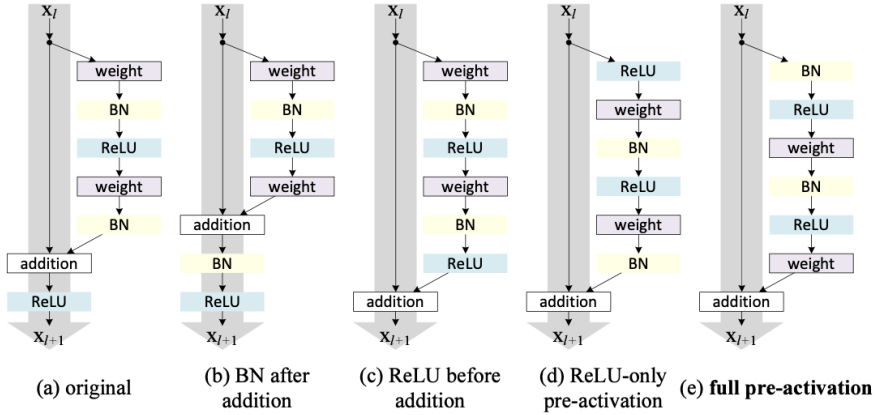


Table 2. Classification error (%) on the CIFAR-10 test set using different activation functions.

case	Fig.	ResNet-110	ResNet-164
original Residual Unit [1]	Fig. 4(a)	6.61	5.93
BN after addition	Fig. 4(b)	8.17	6.50
ReLU before addition	Fig. 4(c)	7.84	6.14
ReLU-only pre-activation	Fig. 4(d)	6.71	5.91
full pre-activation	Fig. 4(e)	**6.37**	**5.46**

考察④
Residual Unitsの先頭にactivationを置くのが効率が良い

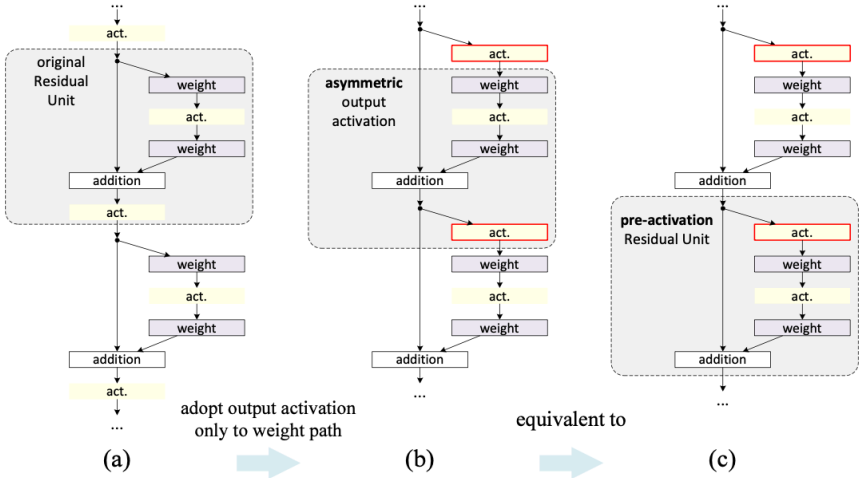


Table 3. Classification error (%) on the CIFAR-10/100 test set using the original Residual Units and our pre-activation Residual Units.

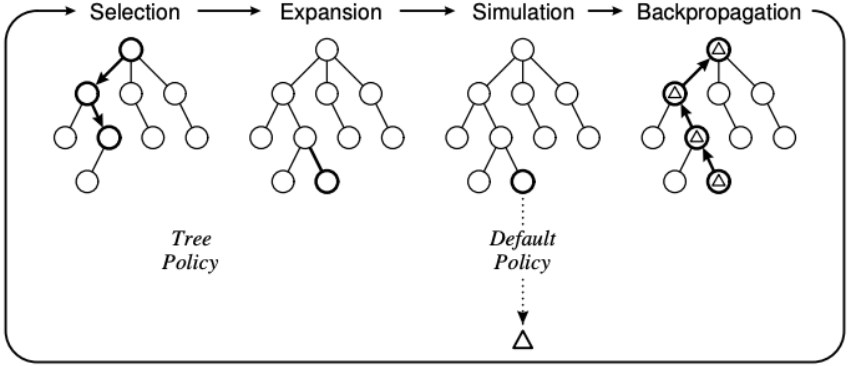
dataset	network	baseline unit	pre-activation unit
CIFAR-10	ResNet-110 (1layer skip)	9.90	8.91
	ResNet-110	6.61	6.37
	ResNet-164	5.93	5.46
	ResNet-1001	7.61	4.92
CIFAR-100	ResNet-164	25.16	24.33
	ResNet-1001	27.82	22.71

A Survey of Monte Carlo Tree Search Methods

（２０１２）著者：Cameron Browne, Member, IEEE, Edward Powley, Member, IEEE, Daniel Whitehouse, Member, IEEE, Simon Lucas, Senior Member, IEEE, Peter I. Cowling, Member, IEEE, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis and Simon Colton

概要

モンテカルロ木探索(以後MCTS)のまとめ論文。MCTSが考えられてから当時までの５年間の技術を記載している。
MCTSはツリー検索の精度とランダムサンプリングの一般性を組み合わせた検索方法で、AlphaGo～MuZeroでも採用されている。



$$UCT = \overline{X}_j + 2C_p\sqrt{\frac{2\ln n}{n_j}}$$

モンテカルロ木探索の手順説明

①選択：ルートノードから開始し、UCB1に従ったポリシーが再帰的に適用されリーフに達するまでツリーを降下。
②展開：利用可能なアクションに従い、ツリーを展開するために一つ（もしくは複数）の子ノードが追加される。
③シミュレーション：結果を生成するためにデフォルトポリシーに従い新しいノードからシミュレーションが実行される。
④バックプロパゲーション：シミュレーション結果は、選択したノードを介してバックプロパゲーションされて、統計量が更新される。

Xj → 平均報酬，n = これまでのプレイ総数，nj = jが選択された回数

MuZeroのアクション選択

$$a^k = \arg\max_a \left[Q(s,a) + P(s,a) \cdot \frac{\sqrt{\sum_b N(s,b)}}{1+N(s,a)} \left(c_1 + \log\left(\frac{\sum_b N(s,b) + c_2 + 1}{c_2} \right) \right) \right]$$

モンテカルロ木探索により、過去は教師付きモデルを踏襲し過去の対戦データを学習したのちにプログラムが試行を開始、アクションを指定していたのに対し、人の関与を必要としない、かつ人間の対戦に夜教師データをも使用しないself_playを確立した手法。

Maximum_Entropy_Deep_Inverce_Reinforcement_Learning

（２０１６）著者：Mobile Robotics Group, Department of Engineering Science, University of Oxford

概要

マルコフ決定過程における逆強化学習。

強化学習は報酬を決め、それに伴った生涯報酬を最大にする方策を選択するが、逆強化学習は目的までの方策（エキスパート）を示すことで最適な報酬関数を導き出すもの。

人間や動物の行動を調べる際には、経験に基づいて報酬関数を考慮すべきとの考え。

→エキスパートの報酬関数を模倣する。

$$P(\varsigma|r) \propto \exp\{\sum_{s,a\in\varsigma} r_{s,a}\}.$$

$$\mathcal{L}(\theta) = \log P(\mathcal{D},\theta|r) = \underbrace{\log P(\mathcal{D}|r)}_{\mathcal{L}_\mathcal{D}} + \underbrace{\log P(\theta)}_{\mathcal{L}_\theta}.$$

$$\frac{\partial\mathcal{L}}{\partial\theta} = \frac{\partial\mathcal{L}_\mathcal{D}}{\partial\theta} + \frac{\partial\mathcal{L}_\theta}{\partial\theta}.$$

$$\begin{aligned}\frac{\partial\mathcal{L}_\mathcal{D}}{\partial\theta} &= \frac{\partial\mathcal{L}_\mathcal{D}}{\partial r}\cdot\frac{\partial r}{\partial\theta} \\ &= (\mu_\mathcal{D} - \mathbb{E}[\mu])\cdot\frac{\partial}{\partial\theta}g(f,\theta),\end{aligned}$$

・ニューラルネットワークを使った複雑な非線形報酬関数の近似

＜アルゴリズム＞

①初期化したNNにより報酬を推定→②経路分布を計算→③熟練者と②の誤差を計算→④誤差逆伝搬法によってパラメーターを更新

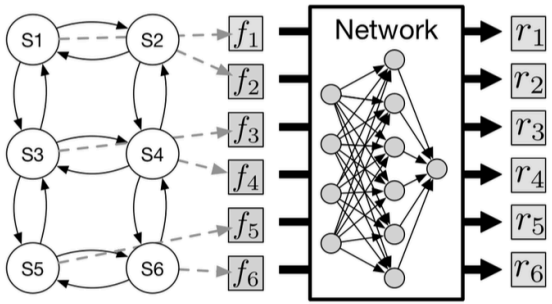


Figure 2: Schema for Neural Network based reward function approximation based on the feature representation of MDP states

・報酬関数は人が設定した特徴量の線形結合に限られていた。

→複雑な非線形な報酬関数でも推定可。

→高速化