

ponkotsu アピール文書

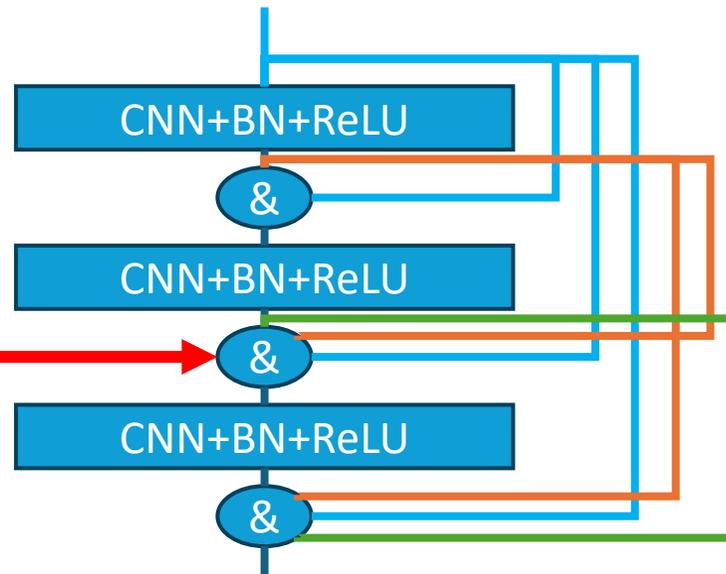
WCSC33からの改良内容

- ネットワーク構造の改良
- 学習率スケジューラの変更
- 最適化関数の変更
- 学習データのアップデート

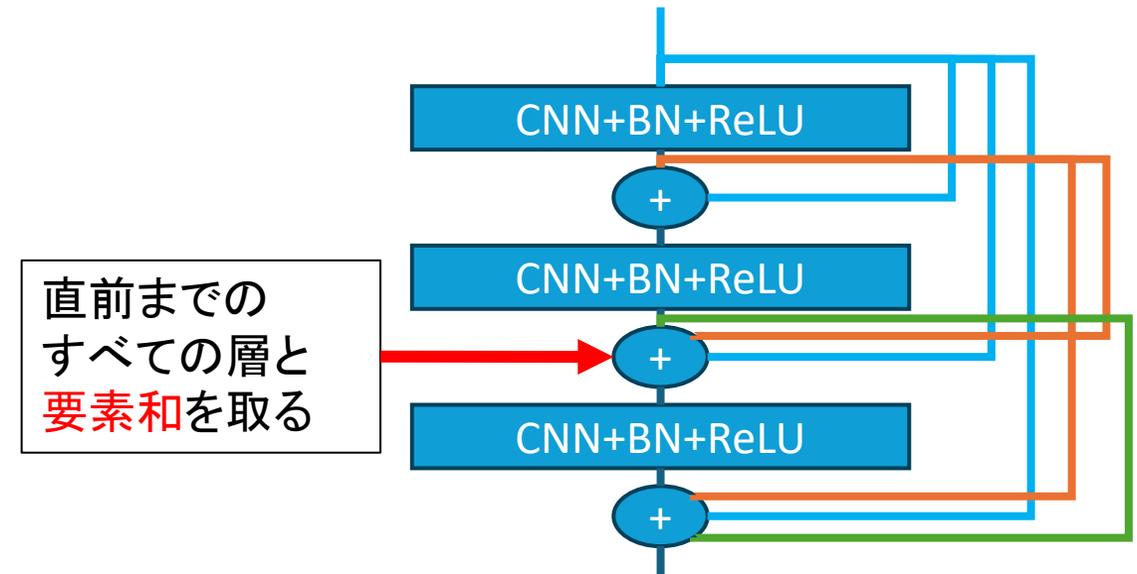
ネットワーク構造の改良

- 従来版で問題となっていた層が深くなるにつれチャンネル数が増大する問題の対策として、スキップ接続をチャンネル方向に結合するのではなく要素和を取るようになった

従来版(DenseNet)



改良版



※オーバーフロー対策として要素和を取った後平均を取っている

ネットワーク構造の改良(識別精度の比較)

- 学習データ: DL将棋本付属訓練データ
- epoch数: 22 epoch
- テストデータ: DL将棋本付属テストデータ
- Policy・Valueともに改良後のほうが高精度

Model	Policy Accuracy(%)	Value Accuracy(%)
DenseNet10	43.4	72.2
改良版DenseNet10	<u>44.3</u>	<u>72.9</u>

学習率スケジューラの変更

- 学習率スケジューラをReduceLROnPlateauに変更

ReduceLROnPlateauでは

loss_{best}: 最も良かったloss

threshold: 閾値

patiance: 改善が見られなかったepoch数

としたときlossがpatience epoch連続で

$$\text{loss}_{\text{best}} * (1 - \text{threshold})$$

以上だった場合学習率を下げる

最適化関数の変更

- 最適化関数をSAM(Sharpness-Aware Minimization)に変更

以下の手順でパラメータを更新する

1. パラメータ w の周辺で損失が最大となる $w + \hat{\epsilon}(w)$ を算出
2. $w + \hat{\epsilon}(w)$ における損失・勾配を算出
3. 2.で算出した勾配で w を更新

最適化関数の変更(識別精度の比較)

- 学習データ: DL将棋本付属訓練データ
- epoch数: SGD 22 epoch、SAM 11epoch
- テストデータ: DL将棋本付属テストデータ
- Policy・ValueともにSAMを使用したほうが高精度

Model	Policy Accuracy(%)	Value Accuracy(%)
DenseNet10+SGD 22 epoch	43.4	72.2
DenseNet10+SAM 11 epoch	<u>44.0</u>	<u>72.7</u>

学習データのアップデート

- DL将棋本の付属データのうちfloodgateの棋譜と水匠の棋譜を以下のようにアップデート

種類	従来の内容	アップデート後の内容
floodgateの棋譜	floodgateの2019年～2021年5月のレーティング3500以上のプレイヤーの棋譜	floodgateの2019年～ 2024年3月 のレーティング 3800以上 のプレイヤーの棋譜
水匠の棋譜	水匠3改を使用して、1手120万ノードで自己対局した棋譜	たややんさん作成の 最新の水匠 を使用して 1手100万ノード で自己対局した棋譜(リンク)

その他

- 以上の改良に加え、ネットワークのブロック数を10→15に増やす予定
- 後日昨年のバージョンとレーティングを比較する予定

参考文献

- [ReduceLROnPlateau — PyTorch 2.2 documentation](#)
(2024年3月27日閲覧)
- ["Sharpness-Aware Minimization for Efficiently Improving Generalization", Foret, P., Kleiner, A., Mobahi, H., Neyshabur, B. \(2020\)](#)
- [SoTAを総なめ！衝撃のオプティマイザー「SAM」爆誕&解説！ #機械学習 – Qiita](#) (2024年3月27日閲覧)