［ディープラーニング活用事例と使いこなしの勘所］ 最適化・推論分野

7 ディープラーニングを使った将棋 AI の学習

山岡忠夫

コンピュータ囲碁においてディープラーニングを使用した手法が成功している．私は，将棋においてもディープラーニングが有効かに興味があり，ディープラーニングを使った将棋 AI「dlshogi」の開発を行っている．本稿では，ディープラーニングを将棋 AI に応用した際の入力特徴と出力の設計，学習に使用したデータセットと学習方法，対局プログラムの実装について紹介したい．

ニューラルネットワークの構成

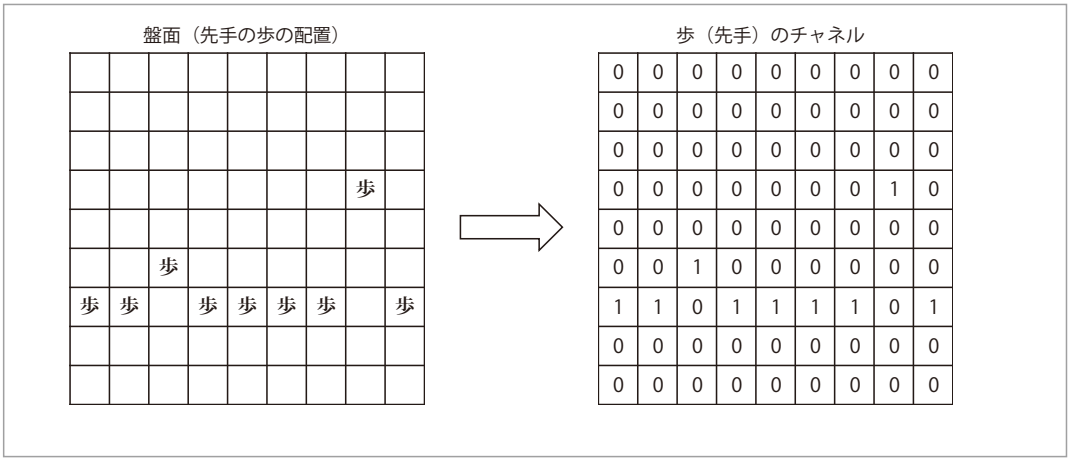
ある局面における指し手を予測する「方策ネットワーク」と，局面の勝率を予測する「価値ネットワーク」の 2 つのニューラルネットワークを学習する．ニューラルネットワークの構成には，画像認識の分野で高い精度をあげている Residual Network (ResNet) を使用し，10 ブロックの ResNet ブロックで構成した．畳み込み層のフィルタは，サイズ 3 × 3 で 192 枚とした．プーリング層は使用していない．

入力特徴の設計

将棋の盤面を複数チャネルの 9 × 9 の画像として入力する．盤上の駒は，駒の種類ごとにチャネルを分けて，駒が存在する座標の画素を 1 とし，ほかは 0 とする（図 -1 参照）．持ち駒は駒の種類ごとに，持ち駒の枚数分のチャネルを割り当て，持ち駒がある場合は画素をすべて 1，持ち駒がない場合は画素をすべて 0 とする（図 -2 参照）．持ち駒の歩の枚数は 8 枚を上限とした．それ以外に，駒の利き情報と王手の情報も加えている．王手の情報を除いて，これらのチャネルは先手と後手で分けている．チャネル数は合計で，119 になる．

出力の設計

方策ネットワークの出力は，指し手を駒の移動先の座標と移動元の方向の組合せで表す．将棋の駒の移動は，方向が分かれば移動元の駒を一意に特定で



■図 -1
歩（先手）のチャネルの例

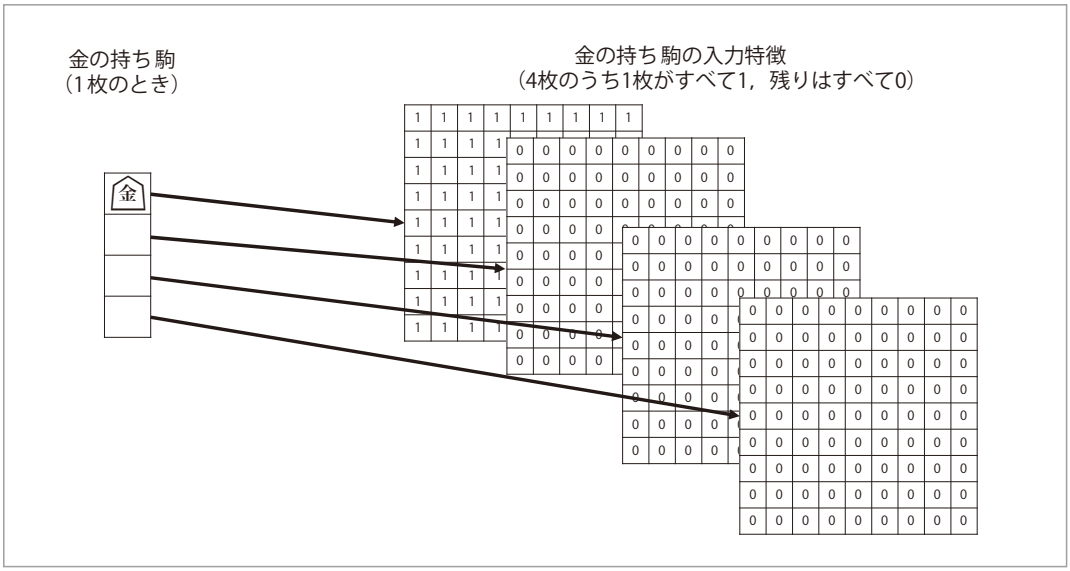
きる．移動元を座標ではなく方向で表すことで分類クラス数を削減している．クラス数を削減することで予測精度の向上を期待できる．方向は，8 方向に加えて桂馬の動きを 2 方向として表す．持ち駒を打つ場合は，駒の種類を 1 つの方向として表す．クラス数は合計で，2,187 になる．価値ネットワークの出力は，勝ちを 1，負けを 0 として，勝率を 0 から 1 の実数で表す．

学習データセット

コンピュータ将棋は従来の $\alpha\beta$ 法を使用したトップレベルのソフトがオープンソースで公開されている．それらのソフトを使って自己対局を行うことで，学習局面を大量に生成することができる．そこで，2017 年の世界コンピュータ将棋選手権で優勝した elmo を使用して学習局面を生成する．10 コアの CPU を搭載した PC を使用して約 2 週間で 35 億局面を生成した．1 つの学習局面は局面（盤面と持ち駒），指し手，勝敗から構成される．ディスクサイズを節約するため，局面はハフマン符号で圧縮し 256 bit で表している．学習局面はファイルとして保存する．35 億局面のファイルサイズは合計で，123 GB になった．

学習方法

前処理として，学習局面から局面と指し手が一致するデータの重複排除と，シャッフルを行い，学習の中断や再開が行いやすいように，学習局面を 1,000 万局面単位に分割を行う．学習の処理は，ディープラーニングフレームワークに Chainer を使用し，Python で実装した．前処理を行った学習局面をミニバッチの単位で読み込み入力特徴に変換しながら学習を行う．その際，上で述べたハフマン符号で圧縮した局面をデコードする処理が Python では遅いため，ミニバッチ単位で局面から入力特徴に変換する処理を C＋＋ で実装した．学習の最適化手法には Momentum SGD を使用した．学習率は 0.01 から開始し，訓練損失が横ばいになったところで 10 分の 1 にした．学習局面とは別に用意したテスト局面に対して，方策ネットワークでは指し手の正解率，価値ネットワークでは勝敗の正解率を測定し，それぞれの正解率が下降しはじめたところで，学習を停止した．学習局面は 35 億局面用意したが，11 億局面学習したところで，収束した．学習にかかった時間は，GPU に NVIDIA 社の GeForce GTX 1080 を使用して約 12 日であった．



■図 -2　金の持ち駒のチャネルの例

対局プログラム

将棋の対局を行うプログラムは，AlphaGo Zero と同様のアルゴリズムで，モンテカルロ木探索を使用して指し手を決定する．探索するノードの選択に方策ネットワークを利用し，葉ノードの評価に価値ネットワークを使用する．対局プログラムは探索速度を重視して C++ で実装した．方策ネットワークと価値ネットワークの推論部分は，Python モジュールを呼び出し，Chainer で推論を行う．複数の探索スレッドからの推論の要求をキューにためて，推論は 1 スレッドでキューをミニバッチで処理する．Python モジュールの呼び出しにはオーバヘッドがあるが，ミニバッチで処理することでオーバヘッドの影響を少なくしている．GPU に GeForce GTX 1080 を用いた場合，1 秒間におよそ 7,000 回シミュレーションを行える．

学習結果の評価

学習したモデルのテスト局面に対する正解率は，方策ネットワークが 46.1%，価値ネットワークが 78.1% となった．対局プログラムの強さは，インターネットのコンピュータ将棋対局サイトの floodgate でレーティング 3,000 程度となっており，人間のプロに近いレベルに達している．しかし，従来手法である $\alpha\beta$ 法と 3 駒関係の評価関数を使用したソフトのレーティングは 4,000 以上であり，従来手法を上回る強さには到っていない．レーティング 1,000 の差は勝率では 99.68% である．モデルの精度，探索手法にまだ課題があると考えている．

今後の計画

AlphaGo Zero のアルゴリズムを将棋に応用した AlphaZero では，強化学習を行うことで，elmo を上回る強さになっている．私の開発している将棋 AI においても，教師ありではモデルの精度が頭打ちになったため，強化学習の手法を試みている．ディープラーニングを使った手法でトップレベルを目指したい．

ディープラーニングの勉強法

私は趣味で将棋 AI を開発しており，ディープラーニングの勉強は独学で行った．参考書としては『深層学習（機械学習プロフェッショナルシリーズ）』（岡谷貴之，講談社），『ゼロから作る Deep Learning —Python で学ぶディープラーニングの理論と実装』（斎藤康毅，オライリー・ジャパン）が理解の役に立った．ディープラーニングフレームワークを使ってサンプルコードを実装して，一部を変更して実行することで，参考書で理解した内容の確認を行った．また，AlphaGo の論文を読んで実装してみたことで理解が深まったと考えている．

（2018 年 7 月 29 日受付）

■山岡忠夫　tadaoyamaoka@gmail.com

東京工業大学工学部電子物理工学科卒業．システムエンジニア．ディープラーニングを使用した将棋 AI「dlshogi」開発者．