広島県立総合技術研究所東部工業技術センター研究報告　No. 35（2022）
技術ノート

強化学習における各種手法の比較（第 3 報）

廣川　勝久

Simple Comparison of Reinforcement Learnings (Ⅲ)

HIROKAWA Katsuhisa

従来の強化学習では，報酬の設定が課題とされている。本報告では，報酬の設定を必要としない逆強化学習に敵対的生成ネットワークを用いた模倣学習の評価について述べる。性能の異なるエキスパートに対する学習能力の比較から，エキスパートと同等な強化学習が可能であることを示す。

キーワード：模倣学習，逆強化学習，GAIL, GAN，敵対的生成ネットワーク，方策勾配法，PyTorch，深層学習

1. 緒　　言

強化学習は，学習前に教師データを用意する必要は無く，環境内を探索しながら，最も良い行動または最も報酬が得られる行動を見つけ出す。この環境内の探索能力により，人間を超えた最良の答えや行動を見出すことが可能となった。

しかし，テレビゲームの得点のように，予め報酬が決まっている環境内では，最良の行動を見出すことができるが，環境内に報酬の設定を必要とする場合では，報酬の設定が適切でなければ最良の結果が得られない。また，囲碁などでは，最後の一手まで打たなければ勝負が決まらず，報酬が得られないため，学習には膨大な時間を必要とするなどの課題がある。

一方で，すでに車の運転やドローンの操縦など匠と呼ばれるような高い能力や経験を持つ人も存在する。このような高い能力を持つ人達からその行動を測定し，最も良い行動を見出すことができれば，行動パターンを報酬に置き換えて強化学習することが可能と考えられる。

逆強化学習はすでに存在する最も良い行動を学習する手法である。逆強化学習と敵対的生成ネットワークを組み合わせたものが模倣学習と呼ばれる深層学習による強化学習であり，一部の企業ではすでに模倣学習を用いた産業応用が始まっている [1,2]。

我々は，これまでの報告により，様々な種類の強化学習について評価を行った [3,4]。本報告ではそれら強化学習により得られたエージェントの行動を模倣学習により方策勾配法のネットワークに学習させ，実験・評価を行ったので報告する。

2. 模倣学習の概要

2.1　逆強化学習

強化学習では，エージェントが実行環境内を探索しながら，環境から与えられる報酬により最も価値ある行動を決定した。強化学習は，報酬の設定が困難であるが，既知の価値ある行動が明らかな場合，その行動軌跡に近づくように，その行動を機械学習させるための手法である。逆強化学習では，良いとされる行動軌跡に対して，尤度が最も高くなるように学習を行う。

2.2　敵対的生成ネットワーク

敵対的生成ネットワーク（**G**enerative **A**dversarial **Net**works：**GAN**）[5]は２種類のニューラルネットワークで構成される（**図１**）。一つは，入力に対して，入力データが本物であるかどうか判定する識別器と，もう一方は乱数から本物そっくりのデータを作る生成器から構成される。識別器は生成器からのデータを偽物と判断出来るように学習を行い，生成器は識別器を騙せるようなデータを出力出来るように学習する。この敵対的生成ネットワークにおける識別器の価値関数は次式で与えられる。

$$V(D) = \log D(x) + \log\left(1 - D\left(G(z)\right)\right) \qquad (1)$$

今識別器が，本物のデータに対しては１に近い値を出力し，偽物のデータに対しては 0 に近い値を出力するとする。すると，識別器に本物のデータが入力された場合，前項が大きくなり，偽物が入力された場合は，第 2 項が大きくなることで，価値関数がどちらのデータに対して

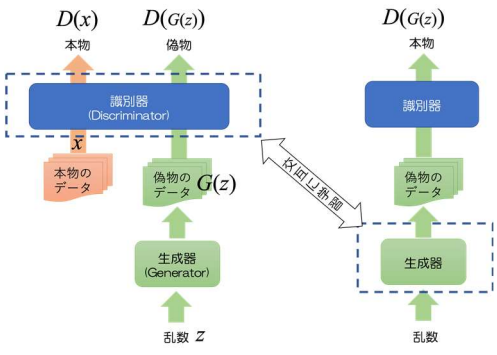

図１　敵対的生成ネットワークの学習

2022. 9. 7　加工技術研究部　受理

も評価可能となる。従って，識別器の学習には，この価値関数を最大化するよう学習を行えば良い。

一方，生成器の学習時には，生成器が作り出したデータに対して，識別器が本物と判断させる必要があるため識別器からの出力に対して

$$\mathrm{V}(G) = \log D\big(G(z)\big) \qquad (2)$$

を最大化するように，生成器を学習する。識別器の学習時は生成器には損失の逆伝搬を止め識別器のみの学習を行う。生成器の学習時は(2)式の損失を識別器から生成器へと逆伝搬させ，生成器の学習を行う。敵対的生成ネットワークでは，学習が進むつれ，生成器に乱数を入力すると本物と全く同じではないが，非常によく似たデータを出力するようになる。

2.3 模倣学習

模倣学習（**G**enerative **A**dversarial **I**mitation **L**earning: **GAIL**）は，逆強化学習の考え方を敵対的生成ネットワークに実装したものである。模倣学習の構成を**図２**に示す。模倣学習では敵対的生成ネットワークの本物の入力を，エキスパートの行動に置き換え，生成器は学習器として良いエキスパートの行動を学習させる。また，識別器には，環境からの状態sと状態sにおけるエキスパートの行動aが入力され学習される。学習器からの出力についてもエキスパートの状態sと同じ状態を入力し，行動a'を偽物として入力し，学習する。

方策勾配法を使った強化学習では，損失関数に

$$\nabla J(\theta_p) = \log\Big(\pi\big(a_p|s_t,\theta_p\big)\Big)\,Q(s_t) \qquad (3)$$

を用いた。方策勾配法の模倣学習では，行動に伴う報酬から得られる$Q(s_t)$値を識別器からの出力値に置き換え，学習器の学習を行う。

$$Q(s,a') = \log D(s,a') \qquad (4)$$

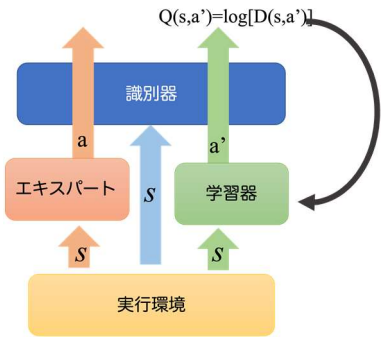

図２　模倣学習の学習

識別器からの出力値$Q(s,a')$が大きければ，学習器の行動が，エキスパートの行動に近いと識別器が判断したと考えられる。即ち，エキスパートの行動に対して学習器は尤もらしい行動を選択したことを意味する。

3. 敵対的生成ネットワークから模倣学習へ

3.1 敵対的生成ネットワークによる画像生成

模倣学習を実装する事前準備として敵対的生成ネットワークを実装し，その動作や学習について基本的な確認を行った。実装に当たり，PyTorch の DCGAN Tutorial[6]を参考にニューラルネットワークの重み部分をコンボリューション型から全結合型に修正した[7]。学習には MNIST: Modified National Institute of Standards and Technology[8]のデータベースによる手書き数字を用いた。実装した識別器のニューラルネットワーク構造を次に示す。

```
def forward(self, x):
    x = self.fc1(x)
    x = self.relu(x)

    x = self.fc2(x)
    x = self.bn2(x)
    x = self.relu(x)

    x = self.fc3(x)
    x = self.relu(x)

    x = torch.sigmoid(self.fc4(x))
    return x
```

識別器，生成器ともに入力層，中間層 2 層，出力層の 4 層からなるネットワーク構造とした。但し，識別器の 2 層目通過後に，BatchNorm1d()を追加し，重みの正規化を行っている。これは，敵対的生成ネットワークの学習速度が低速であり，加えて生成器の出力画像の明るさに大きな差が生じたためである。また，sigmoid 関数により識別器は，0～1 の値を出力する仕様とした。

図３に生成器からの画像出力を示す。これらの画像は，６万種類の手書き画像を 300 回学習した後に，生成器に 200 桁の乱数を入力し，得られた画像である。一部の画像は数字として読み取れないものもあるが，手書き数字が生成されつつあるのが見て取れる。敵対的生成ネットワ

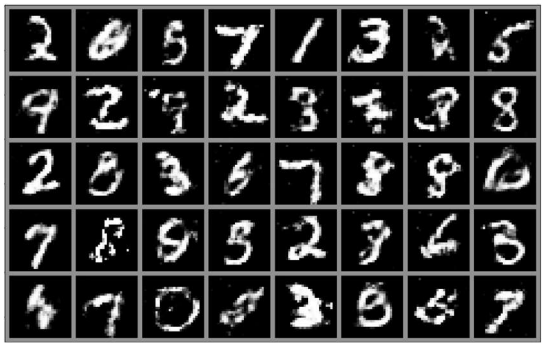

図３　生成器が出力した手書き文字

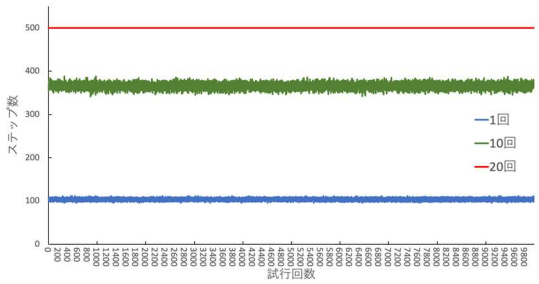

図４　エキスパートの性能

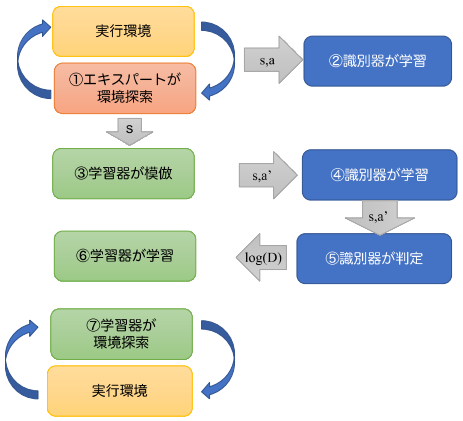

図５　模倣学習の学習フロー

ークでは，識別器と生成器が敵対するように学習が行われるため，両方のニューラルネットの学習速度を同程度にする必要がある。例えば，識別器の学習速度が速い場合，生成器はいくら学習を繰り返しても損失を修正することができず，画像は生成されない状態に陥る。

3.2 模倣学習による方策勾配法の学習

模倣学習の実験では，エキスパートに DQN(Deep Q Network)を用いて [3]，その行動を方策勾配法のニューラルネットワークに模倣させることを試みた。本実験の実施環境にもこれまでの報告と同様に倒立振子の環境，OpenAI Gym の’CartPole-v1'を用いた。エキスパートには，行動価値に基づく DQN に実行環境の倒立振子を立てるための行動を予め学習させた。用いたエキスパートには，能力の異なる 3 種類の DQN を用意した（学習時に 1 度だけ 500 回倒立振子を立てることができたもの，10 回連続で立て続けることができたもの，さらに 20 回連続でできたもの）。学習後，保存した重みを読み込み，各 DQN を CartPole-v1 の環境内で動作させた時の能力評価を**図４**に示す。評価環境では能力に応じて，100 回，380 回，500 回と一定程度安定して倒立振子を立てることができた。

図５に模倣学習のフローを示す。まず，エキスパートは倒立振子が倒れるまで環境内で動作する。そのデータ（環境変数と行動変数）は識別器に送られ，本物と認識できるように学習ネットワークの更新値が計算される。次に，エキスパートが動作した一連の環境変数が学習器に入力され環境変数に応じた出力である行動変数が識別器に送られる。識別器では，学習器からの行動変数が偽物と見分けられるように更新値が計算される。最後に識別器はエキスパートと学習器の両方の損失から，重みの更新による学習を行う。以下に識別器の学習時のソースコードを示す。netD, netG はそれぞれ，識別器，学習器のニューラルネットワークである．

```
netD.zero_grad()
output = netD(actions, states)
errD_real = criterion(output.view(-1), one_labels)
errD_real.backward()

probs = netG(states)
m = Categorical(probs)
actions = m.sample()
log_probs = m.log_prob(actions)

actions = actions.reshape([-1, 1]).to(torch.float32)
output = netD(actions.detach(), states)
errD_fake = criterion(output.view(-1), zero_labels)
errD_fake.backward()

optimizerD.step()
```

ソースコード中の actions.detach()は，学習器への損失の逆伝搬を防いでいる。また，損失関数は(1)式の独自関数ではなく，DCGAN の Tutorial と同じバイナリー交差エントロピー（BCELoss()）関数を用いた。学習器には方策勾配法の入力層，中間層 1 層（ノード数 128 個）出力層，3 層から構成されるニューラルネットワークを用いた [4]。識別器は敵対的生成ネットワークと同じ 4 層構造としたが，識別器の学習速度が速いため BatchNorm1d()は実装していない。

識別器の学習後，再度識別器に学習器の行動変数と環境変数を入力し，以下のように識別器からの出力を求め，学習を行う。

```
netG.zero_grad()
output = netD(actions, states).view(-1)
base = rewind(output)
errG = criterionG(log_probs, output.detach())
errG.backward()
optimizerG.step()
```

学習に際し，学習器は(3)，(4) 式に従って，識別器からの出力値を対数値に変換し報酬とする。

```
for l in reversed(range(steps)):
    g_reward = torch.log(output[l]) + gamma * g_reward
    output[l] = g_reward
    tmp.append(g_reward.item())
```

学習後，学習器を環境内で動作させ学習の進捗状況を確認する。

図６にエキスパートの能力に対する学習器の学習速度を示す。用意したエキスパートは上述のとおり a)学習時に 1 度だけ 500 回倒立振子を立てることができたもの，b)10 回連続で立て続けたもの，c)20 回連続のものをそれぞれ環境内で動作させ，その動作を模倣させた。学習が進むにつれ，エキスパートの能力が高いほど学習器の能力も向上した。但し，1 度だけ 500 回倒立振子を立てることができたエキスパートを用いた場合，エキスパート

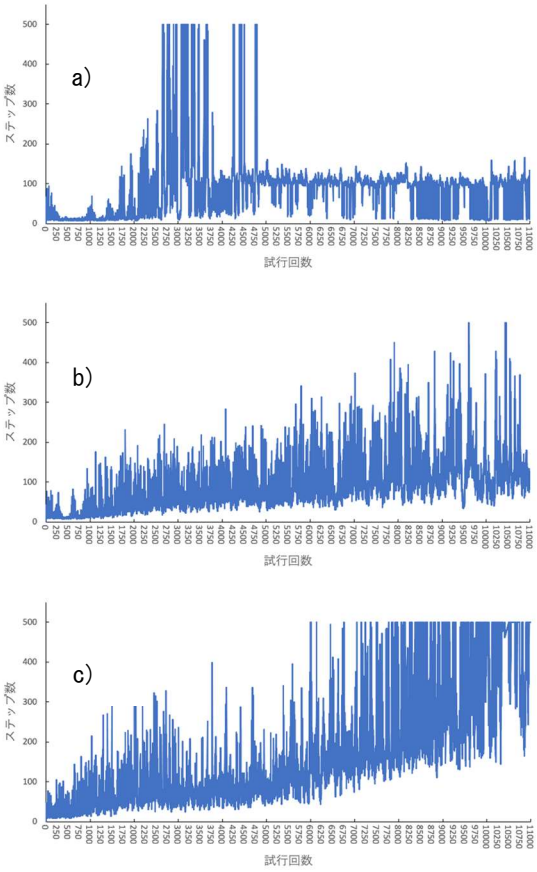

図６　エキスパートの能力に
対する学習器の学習速度
a)1 回，ｂ）10 回，c）20 回

は 100 回程度しか環境内では倒立振子を立てることが出来ないにも関わらず，学習器は早期には 500 回倒立振子を立てることができた。これは，何度学習を行っても同じ状態となった。当初，学習器はエキスパートの能力を上回ることは無いと考えられたが，予想外の結果となった。

学習器の報酬は(4)式で与えられるが，識別器からの出力値を対数値化せず，そのまま報酬として与えた場合の学習速度を**図 7** に示す。エキスパートには 20 回のものを用いた。**図 6** と比較して，対数値を用いない場合，学習は進まないことが分かる。

学習時，識別器から出力される損失の値は学習が進むにつれ，エキスパートと学習器からの入力に対して同程度になる方が良く，識別器と学習器の最適化関数の学習パラメータを調整し，識別器と学習器のどちらかが過学習に陥らないようにする必要がある。

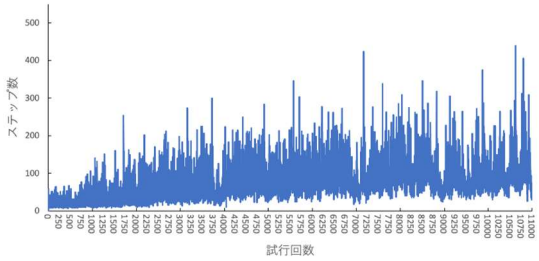

図７　識別器からの報酬を学習

4. 結　　言

強化学習の一手法として，逆強化学習を敵対的生成ネットワークに用いた模倣学習について，その能力等について検討した。実験の結果，エキスパートの能力に非常に近い学習が可能であることが示された。これにより，報酬の設定が困難であり，卓越した行動パターンが予め分かっている場合などには，模倣学習は強化学習の強力な手法と成り得ると考えられる。

文　　献

1) 松永沙織，森本卓爾，三塚由浩，佐藤 匠，毬山利貞：人と協調する AI，三菱電機技報，Vol.95，5，(2021)．
2) 中川陽介，小野仁意，筈井祐介，荒井幸代：熟練者の操作習得とノウハウ可視化に寄与する逆強化学習技術，三菱重工技報，Vol.59，2，(2022)．
3) 廣川 勝久：広島県立総合技術研究所東部工業技術センター研究報告，強化学習における各種手法の比較（第 1 報）**35**（2022）．
4) 廣川 勝久：広島県立総合技術研究所東部工業技術センター研究報告，強化学習における各種手法の比較（第 2 報）**35**（2022）．
5) Ho，J.，et al.：Generative adversarial imitation learning，In Advances in Neural Information Processing Systems，(2016)．
6) DCGAN Tutrial:https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html.
7) 大関真之：Python で機械学習入門 深層学習から敵対的生成ネットワークまで Python で機械学習入門，オーム社（2019）．
8) THE MNIST DATABASE of handwritten digits：http://yann.lecun.com/exdb/mnist/