

進捗報告

1 今週やったこと

- SSL+GL の実験を再現してみました。
- 関連の論文と資料を読みました。
- 麻雀について何をやるのかかんがえました (?)

2 SSL+GL の実験

最初はなかなかできないし、説明も分かりませんでした。「順で実行する」の意味分かりませんでした。

現時点での使い方

・ encoder生成

import section

CL内

DataUtils, OtherUtils, Network, Pretrain

の順で実行することで作成できます

\

・ GA使用法

encoderがすでに生成されていれば

Pretrain以外順に実行すればできます

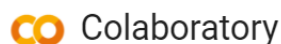
そのため現在Pre trainの実行部はコメントアウトしており

import section, CL, GA part で実行できます

気になった部分出ました。

```
#drive にデータ保存用
from google.colab import drive
drive.mount('/content/drive')
```

最初にこれはただ googledrive 使ってるだけだと思ってました。調べたら



順調になりました。

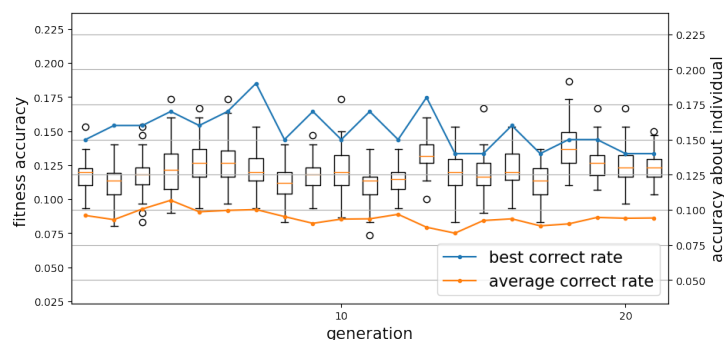
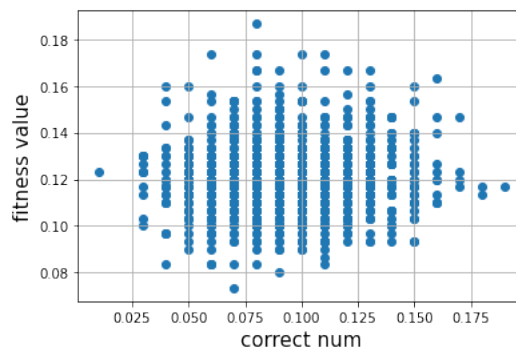
- 結果
まずは事前学習の結果時間がないので 100epoch (論文には 500) くらいやりましたが結果はほとんど変わっていません理由わかりません。

5	5.972951103	0.3212	0.8222
6	5.971372196	0.3251	0.824
7	5.97514284	0.3209	0.8225
8	5.974097824	0.3213	0.8232
9	5.973317704	0.3196	0.8221
10	5.97376959	0.322	0.8225
11	5.973565356	0.3219	0.8237
12	5.975173519	0.3231	0.8238
13	5.97223645	0.3237	0.8256

114	5.973672318	0.3206	0.8226
115	5.975586536	0.3235	0.8244
116	5.977025895	0.3197	0.822
117	5.972010427	0.3206	0.8221
118	5.976404865	0.3281	0.8239
119	5.975774625	0.3239	0.8205
120	5.972893263	0.3189	0.823
121	5.975280264	0.3256	0.8219

そして GA の結果も同じです。

20 代くらいやりましたがほとんど変わってない、しかも精度低い。



何かを見逃したかもしれない。

- 疑問
論文には事前学習はやるって書いてましたがコードの説明にいらないって言いました。もしか

表 5.7: 実験 2 : SimCLR の設定

model	Encoder	ResNet18
	Projection head	2 層 MLP(shape:2048to512)
	classifier	MLP(shape:2048to10)
事前学習		
train data	unlabeled	50000($D_1 + D_{ul} + D_s$)
batch size		1024
epochs		500
optimizer		RAdam($lr=1.0 * 10^{-3}$)

```
"""
encoderの学習クラス
net:ネットワーク
feature_dim:出力次元
temperature:Contrastive Learning(: CL)の温度係数
batch_size,epochs:バッチサイズ,エポック数
save_dir:保存フォルダ

k,cについて
test dataに対してtrain dataのc枚におけるfeatureが似ているものをk枚取り出して
それらのラベルの割合からtest dataのラベルを推定してtestの識別率の数値を出している

ラベル数が限られている実験なので不必要
一応残しているが消したほうがいいのかも
"""
```

したら結果が悪いのもこれと関係ありますか？

3 関連の論文と資料

気分転換として資料を読もうと思いました。

- RAdam という最適化アルゴリズム
Rectified Adam Warmup 付きの Adam
Adam:SGD に「モーメンタム」と「適応学習率」
を組み合わせたもの
もしかして warmup 使ってるから最初の結果が悪いの？
振り返ってみたら warmup の設定あったけど、全く使ってないです。笑

```
def get_cosine_schedule_with_warmup(optimizer,
                                     num_training_steps,
                                     num_cycles=.75,
                                     num_warmup_steps=0,
                                     last_epoch=-1):
    ...
    Get cosine scheduler (LambdaLR).
    if warmup is needed, set num_warmup_steps (int) > 0.
    ...
```

- Suphx: Mastering Mahjong with Deep Reinforcement Learning
本まで出しています



今でも更新つづいています。

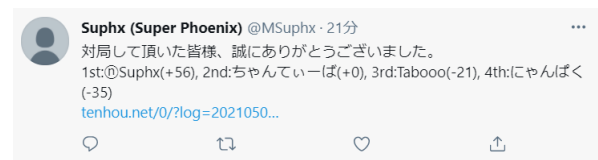
Suphx (Super Phoenix)
@MSuphx

An AI that's focused on the interesting and challenging problem: Mahjong. Please contact us via suphox@outlook.com or Twitter.

[自己紹介を翻訳](#)

📅 2019年3月からTwitterを利用しています

毎日対戦やっています。



- 爆打 (AI: 東京大学/HEROZ)
まさかの水上さん！

4 麻雀について何をやるのかかんがえました(?)

suphx のアルゴリズムのなかに、戦略を選ぶとき (攻撃的なのか、保守的なのか)、まず手牌を固定して、対戦相手の手牌と牌山をランダムに生成して、訓練されたモデル使ってゲームを進める。これを繰り返して十万回やるんですよ。今の状況により戦略を更新する。一種類の手牌に対して十万回をシミュレーションするのはちょっとやりすぎじゃない？必要ある？