fbnet 使い方

コードは 2 種類のネットワークから成る

①supernet_main_file

②architecture_main_file

全体の流れとしては、①supernet でパラメーターθ(ブロック選択用のパラメータ)と w(重み)を交互に学習して、θ で決定されたネットワークを用いて②architecture_main_file の方で w(重み)だけを学習して最終的な accuracy を出す。

supernet_main_file の大まかな参照関係としては

supernet_main_file --- model_supernet.py　　nn.Module を継承してモデルを定義

--- lookup_table_builder.py　候補ブロックや、選択する層の数を定義

└fbnet_builder.py　候補ブロックの定義を行う

--- config_for_supernet　学習率や保存場所などの様々な設定

--- dataloaders.py

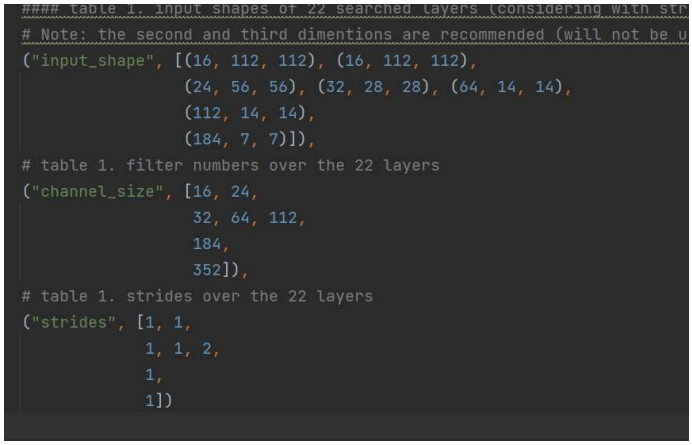
supernet_main_file の学習の方法

1. dataloders.py の get_loaders と get_test_loader を学習したいデータセット用に変更する
2. lookup_table_builder.py の CANDIDATE_BLOCKS に選択肢としたいブロックを書き込む。
   選択肢として書き込めるブロックは、fbnet_builder.py の PRIMITIVES にあるブロック
3. lookup_table_builder.py の input_shape,channel_size,strides を変更する。この時この 3 つのリストの要素の数は探索するブロック数なので一致させる。

input_shape は、タプルの数で探索するブロック数を表している。デフォルトではタプル内にチャネルサイズ、画像サイズが入っているが、処理では使われていないため適当でもいい

channel_size これによって探索するブロックのチャネル方向の出力サイズを決定している。

strides これによって探索するブロックのストライドを決定している。



4. 2 で CANDIDATE_BLOCKS を変更した場合 config_for_supernet の lookup_table-create_from_scratch を True にする。（変更した場合レイテンシーを再計算するため）

5. model_supernet.py の FBNet_Stochastic_SuperNet クラスで探索するブロックの前後の層を変更する

6. 以下のコードで実行する

```
python supernet_main_file.py --train_or_sample train
```

・勾配爆発したら config_for_supernet の w の学習率を下げる

・GPU メモリが溢れた場合 config_for_supernet の batch_size を下げる

その他 Loss のレイテンシー項の調節やエポック数などを config_for_supernet で変更する

学習が終わったら以下のコードで実行する

```
python supernet_main_file.py --train_or_sample sample --architecture_name my_unique_name --hardsampling_bool_value True
```

(--my_unique_name はモデルにつけたい名前を入れる)

このコードを実行することで、発見されたモデルが fbnet_modeldef.py に書き込まれる。

lookup_table_builder からブロック数などを読み取っているため、input_shape のリストの改行なども反映されるようにした。(この部分 utils.py は塚本が作成したのでエラーがあれば教えてください)

あと modeldef.py に first[16,2] ←[channel, stride]と必ず書き込むようになっています

が、model.supernet の self.first の output とストライドを変更した場合は、これを変えてください。

architecture_main_file の実行方法

1. supernet の学習で用いたデータセットのチャネルサイズを fbnet_builder.py の一番下の get_model 関数の model = FBNet(dim_in=チャネルサイズ)としてください
2. その他 supernet の学習段階で model_supernet の FBNet_Stochastic_SuperNet の self.first を変えた場合は fbnet_builder.py の add_first 関数
   self.last を変えた場合は add_last_states を合わせる。
3. 以下のコードを実行する

```
python architecture_main_file.py --architecture_name my_unique_name
```

(--my_unique_name はモデルにつけた名前を入れる)

LOG に最終的な accuracy が書き込まれる。

注意

・PRIMITIVES に need expanssion とコメントしたものは、supernet 学習時に lookup_table_builder.py の_generate_layers_parameters 関数の layers_parameters の 3 つ目の要素がデフォルトでは-999 とある部分を 1 などにすれば全ての need expanssion とコメントしたものに対してその expansion が使用される。-999 のままでは使用することはできない。

・PRIMITIVES に新しい候補を追加する時は“basic_block”の CascadeConv3×3 のようにチャネルサイズの in と out とストライドを引数にとり、それらに合わせて出力、self.output_depth を用意し、出力チャネルサイズ(引数の C_out)を代入しておけば追加することはできそう

→候補ブロックの出力のサイズはすべて同じに調節しなければならない

・model_supernet の構造を変更した場合、architecture_main_file の方には反映されていないため、fbnet_builder の add_first,add_last_states を合わせて変更しなければならない