まず，DPとDDPはどちらもGPUを複数利用して学習を行いたいときに用いる手法である．

DPの方が実装が簡単で用いられることも多いが，公式としてはDDPの方を推奨している．

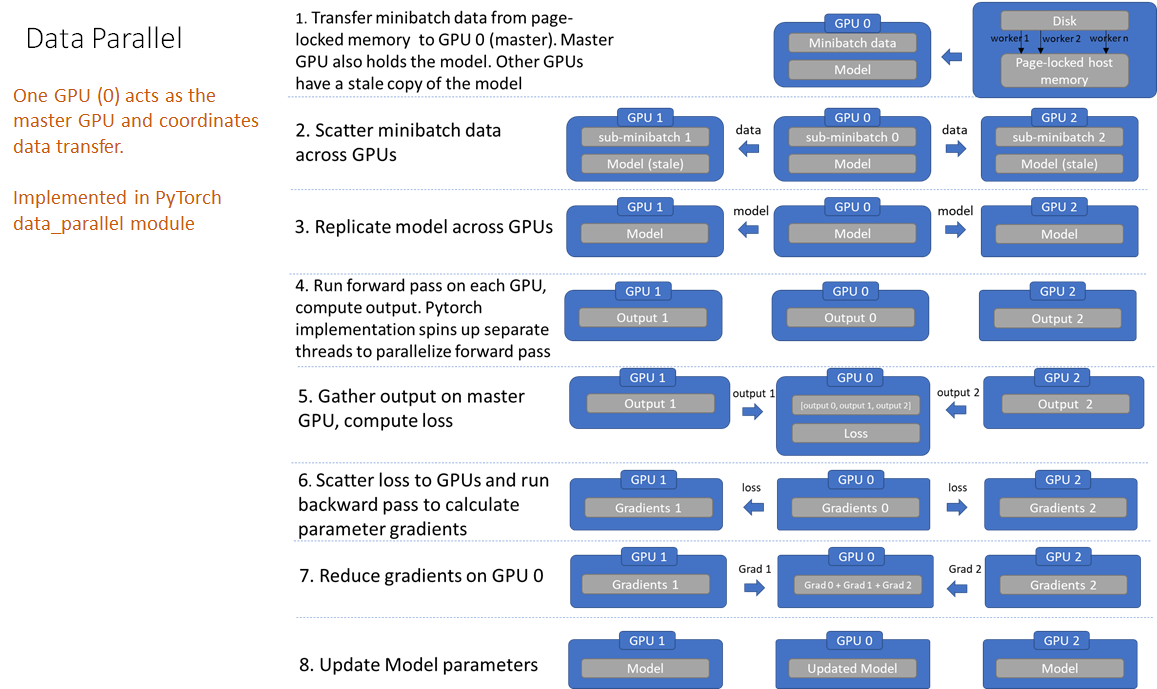
理由は，学習速度が速く，精度も高いため．当然DDPを使うべきである．

<https://qiita.com/fam_taro/items/df6061b589c3ccf86089>

上記の記事を参考にすると両者の違いがよくわかる

DPについて

DPは基本的に一つの親モデルがその他の子モデルを使って学習を進めている．



図で見ると，2,3,5,6,7で各GPU間とやりとりをしているためここで必ず通信ロスが生じる．また，3を見ればわかるように親モデルを子モデルに複製しているのでモデルのパラメタ数が多ければ多いほど負担が大きくなる．

実装方法

modelに対してラッピングをするだけで良い

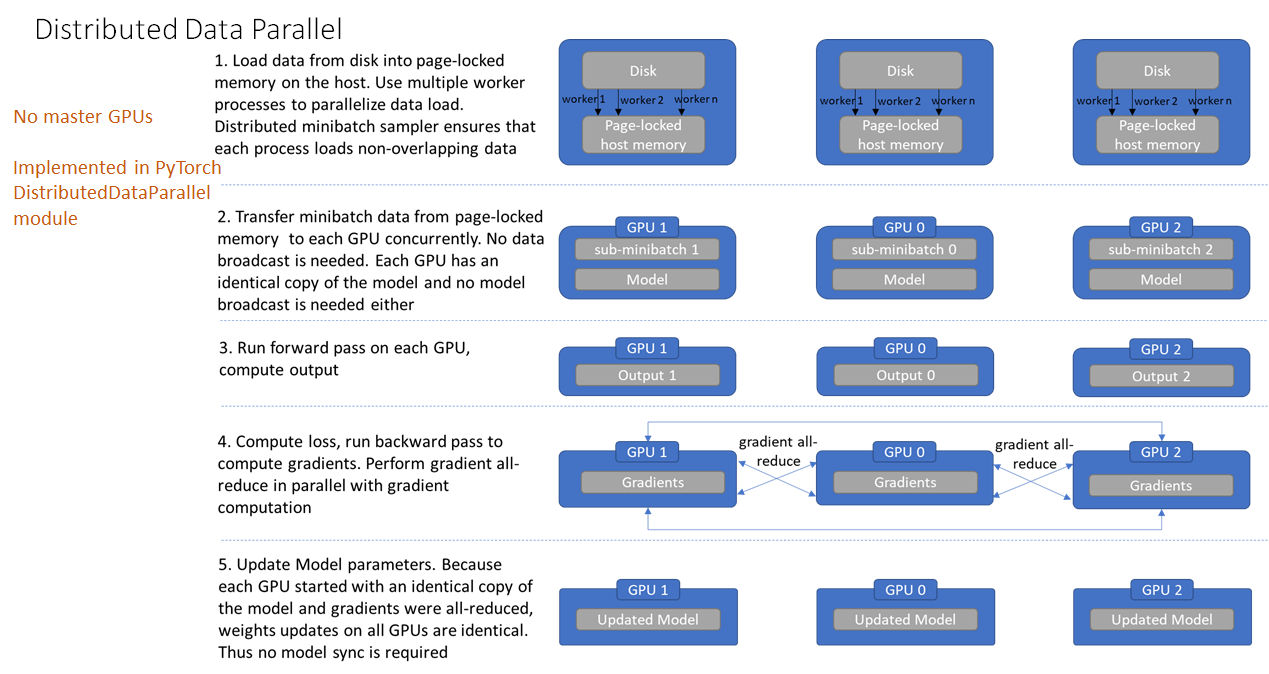
例

model = eval(args.model)(num\_class)

model = torch.nn.DataParallel(model)

model = model.cuda()

DDPについて



DDPでは各GPUが各々独立したデータセットとモデルを有していて勾配計算を行う際に各自の勾配情報のみをGPU間で通信している．また各モデルは一番初めの初期化以降，複製をすることはない．4で共有される勾配値が同じ値になるのでパラメタ更新後も同じパラメタを持ったモデルになるためである．

実装方法

[番外編\_DDP\_learning.ipynb](https://drive.google.com/open?id=1eahURenTmjnoZpIBMn3Vv3dtnfWY1jIx&usp=drive_copy)