

2022年引き継ぎ資料

津村周作

この引き継ぎ資料について

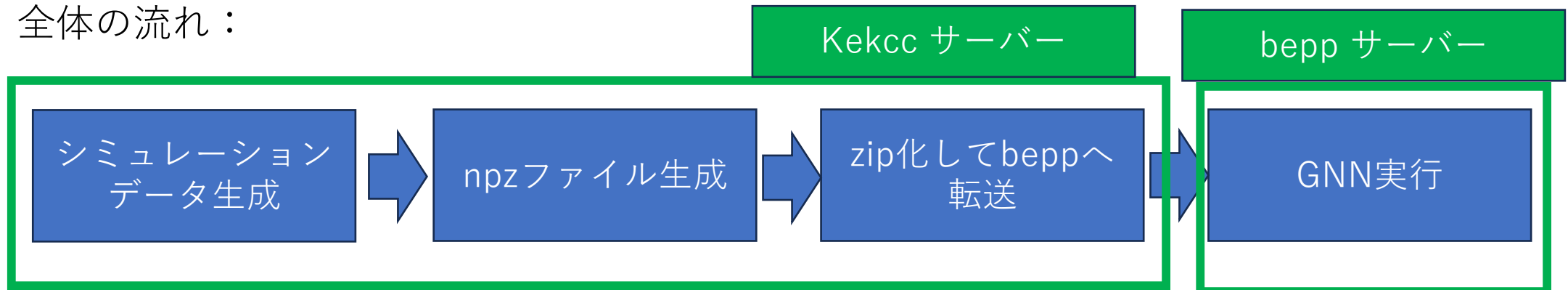
- 研究の全体的な流れは以下の修論にあります。
 - https://github.com/Shusaku-T/MasterThesis_2022.git
- また、機械学習・深層学習については後藤さんの資料も参考になります。
 - https://github.com/Goto-K/Master_Thesis
- 主な違いとしては、
 - ライブラリとしてTensorflowかPyTorchか
 - NetworkとしてDense・RNNかグラフか
 - TaskとしてVertexFinderかParticleFlowかなどが挙げられます
- この資料では技術的な部分についていくつか説明します。

Contents

- GravNetを用いたPFAの実装
 - ILD検出器シミュレーション
 - LCIOファイル→NPZファイルへの変換
 - Kekcc→beppへの転送
 - GravNetによる深層学習
 - Network の評価
- EBES実験のための検出器性能評価
 - エネルギー較正
 - エネルギー分解能評価
 - HV scan

GravNetを用いたPFAの実装

- ILD検出器シミュレーションをkekccで回して、深層学習をbepp gpuで行なっています
 - iLCSoftをbepp gpuに入れて二つを同じ環境で回してもいいと思います。ただし、CPUやメモリの消費には気をつけてください。
 - beppでの環境構築については後藤さんの資料にあります。
- kek ccの私のディレクトリpathは
/home/ilc/stsumura
です。(tsumuraではないので注意)
- 論文の二光子事象のシミュレーションはdoublePG/exampleで実行し、できたデータは
/group/ilc/users/stsumuraに置いてます
- 全体の流れ：



ILC検出器シミュレーション

- 実行手順
- まず、データを置くディレクトリを作成します
 - ./prepareSim.sh
- 100000ファイル作成する工程を並列処理するためにbsubシステムを使っています。
 - ./bsub_sim.sh
 - ./bsub_convert.sh
 - ./bsub_sim.shが全て終わった後で./bsub_convert.shを実行してください
 - 終わったかどうかはbjobsで確認できます。
 - 中身はexe_LCIO_sim.shおよびexe_LCIO_convert.shにあります。
 - Kekccサーバーでの一連の流れ：lcio_particle_gun.py →ddsim→Marlin
→LCIO2npz_dPG.py→zip

prepare.sh

- シミュレーションデータを置くディレクトリを作成しています
- 新しく作成されるディレクトリの名前は固定していて、もし古いディレクトリが作成されるとその古いディレクトリの名前が現在時刻で上書きされます。
- このコードを実行していないと、データを作成するときにエラーが出ます。

LCIOファイル→NPZファイルへの変換

- bsubを行うと自動的に実行されます。
 - 実行ファイルはLCIO2npz_dPG.py
- LCIOからHitの位置・正解ラベル・PDGなどを引っ張ってきます。
 - MC粒子の参照元が間違っている可能性あり
- 生成されたファイルは圧縮されてzipファイルとしても保存されています
 - beppへ送るのはこちらのファイルです

KEKcc→Beppへの転送

- scp -rでeppを経由してbeppサーバーまで転送します
 - 場所は/gluster/maxi/ilcです
- 持ってきたファイルはunzipで解凍した上でそれぞれのイベントごとにファイルを分割します。
 - 生成した段階で分割すると時間がかかるのでここで分割しています
 - 分割するための実行コードはnpz_concate.pyです。引数に注意してください

深層学習の実行

- Zipファイルをunzipで解凍します。
- ここで作成したnpzファイルは転送時間を減らすために1000イベントが一つのファイルに入っているので、npz_concat.pyでそれぞれのイベントごとにファイルを作成します。

深層学習の実行

- Zipで解凍したディレクトリを深層学習のinputファイルとして設定します。
- Settingにあるtxtファイルをこのinputのディレクトリに設定してください。
- train_ILC.pyのReadFileメソッドの引数に設定されているのが上記のtxtファイルであることを確認して、train_ILC.pyを実行してください

ネットワークの評価

- train_ILC.pyを実行すると、ディレクトリcheckpointの中に各エポックごとのGravNetモデルが出力されています。
- Settingのtxtファイルのinput modelにそれぞれのモデルの名前を入力することで評価対象のモデルを指定できます
- plot_statistics_ILC.pyを実行してネットワークの定量的評価を行います。