E70 meeting

2020/04/07

大橋　翼

運動量再構成をMLで代替

- 較正前の運動量はMLで置換できるか？2つの運動量について予測精度の差はでるか？
- 前回(1/28)まで…
  - E05 beam throughデータ
  - SDC(SKStrack(XY))→scatP(SKS再構成運動量、補正前)
  - MLPで$\sigma(p)$~3×10^{-3} GeV
  - Beam line側の位置座標データがなかったため、beamP(beam line spectrometer再構成運動量、補正前)についての評価ができない
- 3/24-
  - E05BTデータ
  - SDC (LocalTrackHit(X))→scatP
  - BDC, BFT(1次元)→beamP
  - ２つの予測精度の差を見てみる

補正前運動量再現

- データ概要（金築さんより提供）
  - E05 1.8GeV/c BeamThrough 実験データ
  - 105,090 events（うちscatPが算出できているのは104,820 events）
  - 変数の対応関係

	目的変数(output)	説明変数(input)	特徴量次元数
K1.8	beamP	BDC3-4pos (BC3=6面、BC4=6面)、BFTpos	13
SKS	scatP	SDC1-4pos (SDC1=6面, 2=4, 3=6, 4=6)	22

  - Train : test = 7 : 3で評価
  - 評価指標はRMSE[GeV](P_dataは既存運動量)

$$RMSE = \sigma = \sqrt{\frac{\sum_{i=1}^{N}(P_{ML} - P_{data})^2}{N}}$$

単純なDNN結果

- beamP: RMSE~4×10^{-3}

num_units	num_layers	batch_size	optimizer	training time[s]	train_rmse	test_rmse
64	3	32	adam	1716.6308901309967	0.002922591	0.0036575352
64	3	128	adam	719.6605031490326	0.0023825679	0.0052704653
64	5	32	adam	1744.6375629901886	0.0036413136	0.0069058817
64	5	128	adam	555.4256553649902	0.002797457	0.0073670107
128	3	32	adam	1341.3330075740814	0.002735309	0.005564817
128	3	128	adam	634.1119363307953	0.0023214903	0.006622402
128	5	32	adam	1179.1915137767792	0.0037957006	0.0067157126
128	5	128	adam	529.6820487976074	0.0023634795	0.004257694

- scatP: RMSE~3×10^{-2}

num_units	num_layers	batch_size	optimizer	training time[s]	train_rmse	test_rmse
64	3	32	adam	1759.5379824638367	0.012820818	0.039255224
64	3	128	adam	315.1544692516327	0.013001348	0.031689487
64	5	32	adam	1471.287977695465	0.022913342	0.026049722
64	5	128	adam	295.6483814716339	0.019892868	0.025008818
128	3	32	adam	1175.9535782337189	0.014532808	0.030645715
128	3	128	adam	209.67463064193726	0.01180116	0.032884184
128	5	32	adam	1167.2529435157776	0.013021205	0.03457619
128	5	128	adam	321.42857933044434	0.0124177	0.036827806

単純なDNN結果

- beamP: RMSE~4×10^{-3}
- scatP: RMSE~3×10^{-2}

- SKSTrackXYを用いた際はscatP RMSE~3×10^{-3}
  - 前回meeting「表現力の差？」
  - →複雑なモデルも含めて検証

- 残差分布をみる
  - インプットの分解能の伝搬
  - 分布によって残差の差が出るか

HP探索(HyperBand)

- Hyper ParameterのTuning手法
- 探索空間を与えるとその空間内で良いHPを探す
- （論文はまだよめていない）

- より複雑なモデルを取りえる空間で探索する
- 表現力が上がれば予測精度は上がるのか？（RMSEが下がるのか）

Hyperband結果

```
### log
```bash
Search space summary
|-Default search space size: 4
layers (Int)
|-default: 3
|-max_value: 30
|-min_value: 3
|-sampling: None
|-step: 1
units (Int)
|-default: None
|-max_value: 512
|-min_value: 32
|-sampling: None
|-step: 32
dropout (Float)
|-default: 0
|-max_value: 0.5
|-min_value: 0.0
|-sampling: None
|-step: 0.1
learning_rate (Choice)
|-default: 0.01
|-ordered: True
|-values: [0.01, 0.001, 0.0001]
```

```
[Results summary]
 |-Results in ../tune_log/scatP
 |-Showing 10 best trials
 |-Objective(name='val_loss', direction='min')
[Trial summary]
 |-Trial ID: 5c8e2cb434fa759fcd1f29106921b78a
 |-Score: 0.0034435830713746 53
 |-Best step: 0
 > Hyperparameters:
 |-dropout: 0.0
 |-layers: 28
 |-learning_rate: 0.0001
 |-tuner/bracket: 7
 |-tuner/epochs: 2
 |-tuner/initial_epoch: 0
 |-tuner/round: 0
 |-units: 64
```

- scatP RMSE~0.06
- よくはならない
```

考慮すべき問題

- HitがないSDC plane pos = -9999としてそのまま処理している
- 本来適切に処理しなくてはならない

- 欠損値を適切に処理するとどこまで良くなる？
- -9999 -> NaNと処理して以下進める

Light gbm

- Gradient boost decision tree アルゴリズムの１つ
- 決定木ベースなので欠損値を含んだまま扱える
- HPデフォルト、early stopping 10 step

	Test RMSE
K1.8	8.6×10^{-4}
SKS	7.9×10^{-3}

- DNNはK1.8 : 4×10^{-3}, SKS : 2.5×10^{-2}
- 3-4倍RMSEが改善
- 残差分布をみてガウシアンでフィットなどすると
  - スケールを合わせる

Light gbm

- 改善理由
  - 欠損値として扱ったこと
    - 欠損値を適切に処理してDNNすれば改善？

- SKSTrackXYを用いた際のscatP RMSE~3×10^{-3}に比べて悪い
  - 欠損値処理の余地がある？
  - そもそもトラック引いた後の位置なので勝てない？

データの理解

- E05Analyzerのコードを読む
  - 金築さんから説明

- Tdc -> pposition -> track(momentum)

前処理

- 欠損値処理をきちんとやる
- SDCごとにtrackfitしてhitがないplainをその結果で補間

- 実験中...
  - そこまで改善していなさそう...

展望

- MLで出したデルタを見る
  - 既存より細くいっていれば
  - 補正後との比較
- 関数系を仮定してやる
- 分布を取り出していろいろやってみる
  - 決まり切っているところはもういいのでは

- Inputの座標を振ってみてそれぞれのMLの精度の悪化を見ると振る舞いの違いを見れる