

環境適応型機械学習および次元削減を適用した
KEK Linac 加速器制御の開発
(R & D of the KEK Linac accelerator tuning using
reinforcement learning and dimensionality reduction)

理学研究科

数物系専攻

久野彰浩

概要

本研究では、高エネルギー加速器研究機構（KEK）で稼働中の、電子・陽電子入射器 Linac における加速器運転調整手法の開発を行った。具体的には、機械学習を用いた次元削減による加速器の状態の可視化手法及び環境適応型機械学習を用いた加速器運転パラメータの最適化手法の開発を行った。

Linac 加速器は長さ 600m の電子・陽電子線形加速器で、Belle II 実験に用いられている SuperKEKB 加速器を含む複数の加速器への入射加速器として運用されている。Linac 加速器における入射効率の向上と安定は、高輝度化と大量データ収集を目標としている Belle II 実験において非常に重要な要素である。

Linac 加速器における運転調整には次の問題点が存在する。

1. 制御するパラメータ数が多く ($\sim O(1000)$)、複雑なシステムである
2. 運転環境変化 (加速器構成機器の温度変化や振動、潮汐力など) に適応するために、常時運転調整を行う必要がある

そこで、本研究では、機械学習を用いた加速器制御手法の開発を行った。多パラメータ間の相関関係の記述に強みを持つ機械学習は、上述の問題点を解決し、運転調整の精度向上や高速化を実現するために有効な手法だと考えられる。具体的には、1. を解決するためにディープニューラルネットワーク (DNN) を用いた次元削減（特徴抽出）による運転状況可視化手法の開発を行った。また、2. 解決するためには環境駆動型の機械学習が有効であると考え、その実装を行うための準備研究として、DNN を用いて加速器の運転状況と入射効率の相関関係を記述する手法や加速器の運転シミュレータの開発を行った。

これらの加速器制御手法の開発を、2018 年 11 月から 2021 年 6 月までの間に蓄積された実際の加速器運用データを用いて行った。開発手法、および性能評価について詳細を述べる。

目次

第 1 章	序論	1
第 2 章	加速器	3
2.1	加速器の構成要素	3
2.2	加速器の加速原理	4
2.3	加速器で使用する電磁石	6
2.4	Linac	7
第 3 章	機械学習	10
3.1	機械学習について	10
3.2	ニューラルネットワーク	11
3.3	変分オートエンコーダ (Variational AutoEncoder)	19
3.4	敵対的生成ネットワーク (Generative Adversarial Networks)	20
第 4 章	収集したデータについて	22
4.1	収集データのパラメータについて	22
4.2	加速器運転における入射効率の指標	22
4.3	収集データにおける入射効率の分布	23
第 5 章	加速器の状態の可視化手法の開発	25
5.1	機械学習を用いた加速器の状態の可視化の方針	25
5.2	実装したニューラルネットワーク	25
5.3	可視化手法の開発結果と考察	26
第 6 章	加速器調整におけるパラメータ群と入射効率の相関関係の記述手法の開発	29
6.1	機械学習を用いたパラメータ群と入射効率の相関関係記述の方針	29
6.2	実装したニューラルネットワーク	29
6.3	加速器のパラメータ群と入射効率の相関関係記述手法開発	30
第 7 章	加速器の運転シミュレート手法の開発	34
7.1	加速器の運転シミュレータの開発方針	34
7.2	実装したニューラルネットワーク	34
7.3	加速器運転シミュレータの開発結果と考察	35
第 8 章	まとめ・今後の方針	42
	参考文献	44

目次

2.1	静電場を使用した加速の例	5
2.2	1 つの静電場を使用し繰り返し加速する例	5
2.3	加速電波による電子の加速の例	6
2.4	KEK における加速器の配置図	8
2.5	Linac 加速器の配置図	9
3.1	ニューラルネットワークの例	11
3.2	ニューラルネットワークの計算の様子	12
3.3	ReLU 関数	13
3.4	シグモイド関数	13
3.5	ドロップアウトの概念図	17
3.6	VAE の模式図	19
3.7	GAN の学習の流れ	21
4.1	Linac 加速器 A Sector の配置図	23
4.2	Linac 加速器 5 Sector の配置図	23
4.3	2018/11 から 2021/06 までの期間での蓄積データにおける入射効率分布	24
5.1	VAE による加速器の状態の可視化の概略	25
5.2	実装したエンコーダーの構造	26
5.3	実装したデコーダーの構造	26
5.4	2018/11 から 2021/05 までの期間での蓄積データの可視化結果	27
5.5	学習に用いていない 2021/06 での蓄積データの可視化結果	28
6.1	加速器のパラメータ群と入射効率の相関関係記述の方針	29
6.2	実装した DNN の構造	30
6.3	2018 年 11 月から 2021 年 6 月の蓄積データによるデータセットでの学習結果のプロット	31
6.4	学習時期の異なるデータセットの概略	32
6.5	図 6.4 のデータセットでの学習結果のプロット	32
7.1	GAN を用いた加速器シミュレータ開発の方針	34
7.2	実装した Generator の構造	35
7.3	実装した Discriminator の構造	35
7.4	データセット 1-1 で学習した GAN の出力	36
7.5	データセット 1-2 で学習した GAN の出力	36
7.6	データセット 2 で学習した GAN の出力: 運転パラメータ	37

7.7	データセット 2 で学習した GAN の出力: 入射効率	37
7.8	実データ (左図) と模倣データ (右図) の二次元ヒストグラム	37
7.9	データ可視化を用いた 1233 パラメータの加速器データの比較の概略	38
7.10	データセット 3 による、実データ (左図) と模倣データ (右図) の潜在変数の二次元ヒストグラム	39
7.11	データセット 4-1 による、実データ (左図) と模倣データ (右図) の潜在変数の二次元ヒストグラム	40
7.12	データセット 4-2 による、実データ (左図) と模倣データ (右図) の潜在変数の二次元ヒストグラム	40

表目次

6.1	図 6.4 のデータセットでの学習結果における入射効率予測精度	33
-----	---	----

第 1 章

序論

本研究では、高エネルギー加速器研究機構（KEK）で稼働中の、電子・陽電子入射器 Linac における機械学習を用いた加速器運転調整手法の開発を行った。

ここで、Linac では大別して以下二つの目的に対して運転調整が行われている。

1. Linac 加速器ビーム源から下流までビームを通すための調整
2. ビームの入射効率を上げるための調整

1. の調整は Linac が長期シャットダウン（夏期、冬期の年二回）後に再立ち上げを行う際に行われる。調整を完了するのに要する時間は通常 1 日程度である。一方で、2. の調整は Linac の運用中にビームの入射効率を改善し、また高い入射効率でのビーム入射を維持するために行うものだが、Linac の運転状況は温度変化や振動等の外部環境の影響を受けて常時変化しているため、この調整は Linac の運用中に常時行われている。高度の調整が求められ、またその頻度も高いことから、本研究では 2. の調整についての開発を行った。

上述した 2. の目的の Linac の運転調整には以下の問題がある。

1. 調整に関わるマシンパラメータの数が多く ($O(1000)$)、パラメータ間の相関が複雑であるため、加速器の状態を把握することが困難となる。
2. 振動や温度変化等、加速器周囲の環境変化により、同じマシンパラメータ設定でも同じ入射性能が得られるとは限らない。高い入射効率を得るために、常に環境変化に適応した設定値最適化が必要となる。

そこで、本研究では、機械学習を用いた加速器制御手法の開発を行った。多パラメータ間の相関関係の記述に強みを持つ機械学習を導入することで、運転調整における精度の向上や高速化に伴う入射効率の向上が期待できると考えられる。

1. の問題については、複雑に相関する多次元のパラメータから特徴を抽出し、比較が容易な低次元のデータへと次元削減を行うことで、加速器運転状況の把握が容易になり、また、これまでの運転状況の変化も明らかにすることが出来ると考えられる。

2. の問題については、変化する環境に適応することのできる環境駆動型の機械学習（強化学習）の適用が有効だと考えられる。しかしながら、加速器運転調整に強化学習を導入するには、加速器の入射効率とパラメータの相関関係を記述する必要がある。また、強化学習の最適化のために実際の加速器を運転することには、想定されない動作を引き起こす恐れがあるため、実際の加速器の運転で学習を行う前に、事前学習を行う必要がある。そのためには、リアリスティックな加速器シミュレーターの開発が必

要である。

故に、本研究では以下の述べる方針で運転調整手法の開発を行った。

1. 加速器の状態の可視化手法の開発
2. 加速器調整におけるパラメータ群と入射効率の相関関係の記述手法の開発
3. 加速器の運転シミュレート手法の開発

1. の開発については、ニューラルネットワークによる教師無し学習を用いた加速器パラメータ群の次元削減による加速器の状態の可視化手法を、2. については、ニューラルネットワークによる教師あり学習を用いた、加速器のパラメータ群を入力としてそのパラメータ環境下における入射効率を予測する手法を、3. については、敵対的生成ネットワーク (GAN) を用いた手法の開発を行い、またそれらの性能評価を行った。

第 2 章

加速器

本章では、高エネルギー実験で用いられる加速器の基本原理解 [1],[2],[3] 及び本研究で注目した KEK Linac について述べる。

加速器とは荷電粒子を加速する装置の総称である。原子核や素粒子実験など物理学分野のみならず、薬学や化学、生物学など幅広い分野の研究で用いられており、実験の目的に応じて、加速する荷電粒子には電子・陽電子を始めとするレプトンや、陽子・反陽子などのハドロン、また重イオンが用いられることもある。

以降、2.1節で加速器の構成要素の概略について、2.2節で加速器における荷電粒子の加速原理について説明し、2.3節で加速器で使用されている電磁石について説明する。また2.4節で本研究において注目した KEK Linac について線形加速器としての概略や KEK における運用について述べる [4],[5],[6],[7]。

2.1 加速器の構成要素

加速器は以下の 5 つの基本的な要素から構成されている。

- 粒子源によるビーム生成

粒子源は加速器で加速する荷電粒子を生成する部分を指す。有効かつ安定的な加速器運用のために、粒子源には高強度かつ広がり少ないビームを生成する事が要求される。先述の通り、加速器で加速される荷電粒子は目的とする実験に応じて多岐にわたるが、一例として電子ビームの粒子源は電子銃、イオンビームの粒子源はイオン源と呼ばれる。以下で本研究で使用した電子ビームの粒子源として一般的な電子銃の手法について述べる。

電子銃による電子ビーム生成については、高温に加熱された固体 (熱陰極) の表面から電子が放出される現象である熱電子放出を利用する手法が一般的だ。また、この手法は、1 つの面に孔をあけた金属製の箱の中で行うことが多い。これは、箱の中に配置した金属に負電荷を帯びさせることで反発によって金属内部の電子の放出を促し、さらに孔をあけた面に電氣的に正のポテンシャルをかけて放出された熱電子がこの面の孔に向かって進みやすくすることで熱電子の収集効率を高めることが出来るためである。この孔を通り抜けてきた電子をビームに利用している。

なお、電子銃からビームを引き出す際には 50 ~100kV の高電圧をかけるため、電子銃から出た段階で 50 ~100keV のエネルギーを粒子は得ることになる。

- ビームの加速

粒子源から引き出したビームを実験目的に応じたエネルギーまで加速する必要がある。ここで、ビームの加速には電場を利用する。そのため、加速器で加速することの出来る粒子は荷電粒子に

限られる。加速の手法については2.2節で詳しく述べる事とする。

- ビームの収束

加速器は理想的な条件下において、設定した中心軌道上をビームが運動するように設計されている。しかしながら、実際のビームを想定すると、粒子源でビームが生成された時点においてもビーム中の粒子は一塊となっておらず、また粒子の放出される位置やその運動の方向もそろっていないため、ビームを進行方向に対して直角な面で切断した場合、断面はある程度の大きさを持つてしまう。また、その断面はビームが進むにつれて広がってしまうため、ビームに外部から何らかの収束力を与えなければ中心軌道から逸れてしまい、さらには発散してしまう。このことから、加速器においては電場や磁場を用いてビームに収束力をかける必要がある。なお、ビームの収束に使われる電磁石については2.3節において紹介する。

- ビーム経路の真空

加速器において、ビームの通る経路にガスが存在していると、ビームがこのガスと衝突することで軌道から外れてしまい、ビーム中の粒子が減少してしまう。故に、ビームの減衰を最小限に抑えるために、加速器ではビームの通る経路内を真空に保つことが必要とされる。また、このことからビームの通る容器は真空ダクトや真空チェンバーと呼ばれる。

加えて、加速器における真空システムには、加速器ならではの留意すべき問題が複数存在している。例えば、本研究で注目した電子加速器であれば、電子ビームが曲げる際には放射光が発生するため、放射光が真空ダクトにあたる際に光脱離による吸着ガスの発生や、放射光が真空ダクトに吸収されることによる発熱など、複数の問題が加速器環境下では発生しうる。そのため、加速器における真空システムは通常の真空システムと異なる幅広い問題に対応する事が求められる。

通常、ビーム中の粒子数がビーム生成時の半分となる時間をビームの寿命と呼ぶため、ビームの寿命はビーム経路内の残留ガス圧力に反比例する。故に、加速器運用中に起こる様々な現象を踏まえたうえでビーム経路における真空を維持する技術は加速器にとって非常に重要な要素となる。

- 加速器の動作制御

加速器は上述した様々な要素からなる複雑なシステムであり、また外部環境の影響等を受けながら変化する動作環境に対して適切な制御を行う必要がある。ただ、制御にはリアルタイム性を強く求められる制御と、そこまでの速さは求められない制御が存在する。これらに対して、前者の制御に関してはハードワイヤーを用いた制御が行われ、後者の制御に関してはネットワーク分散型制御システムを用いた制御が行われている。

2.2 加速器の加速原理

加速器の加速原理は目標とするエネルギーや実験目的に応じて様々である。本節では、加速器で使われている加速方法のうち、静電場を用いた加速と、現在幅広く利用され、本研究で注目した KEK Linac でも利用している高周波加速について述べる。

2.2.1 静電場による加速

加速器において荷電粒子を加速する上で最も単純な手法は静電場を用いた加速手法である。図2.1に静電場を利用した荷電粒子加速の例を示す。この加速手法では、帯電させた金属平面電極により生じたクーロン力を用いて荷電粒子を加速させている。

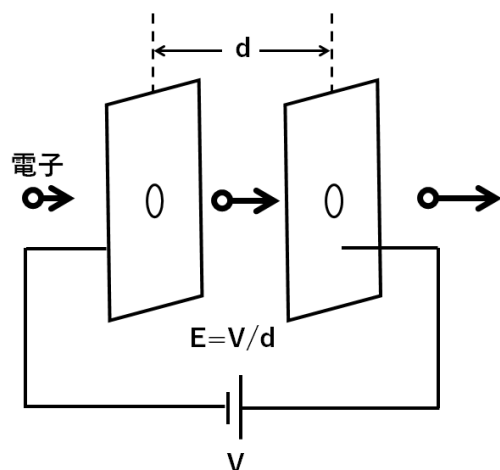


図2.1 静電場を使用した加速の例

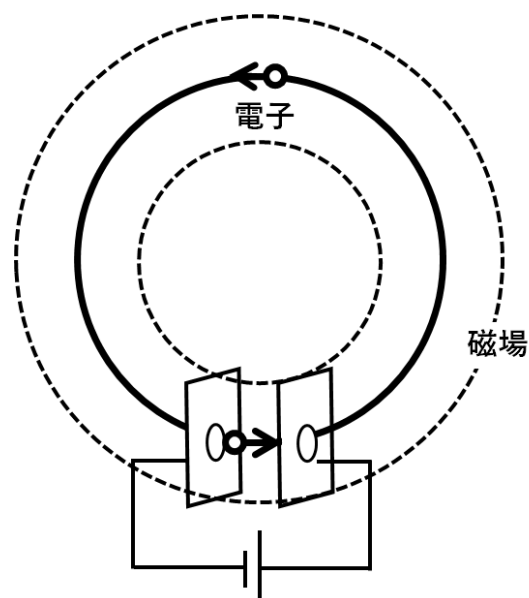


図2.2 1つの静電場を使用し繰り返し加速する例

しかしながら、図2.1に示した静電場による加速では、金属平面間の最大印加電圧に放電限界に起因する制約があるため、一つの金属平面間での加速には限界が生じてしまう。この限界金属を考慮して一つの金属平面間に荷電粒子を繰り返し入射させることで加速させる手法も考えられるが、それは原理的に不可能である。図2.2に一つの金属平面間に荷電粒子を繰り返し入射させて加速させる手法の例を示す。この場合だと、磁場を用いて荷電粒子を回転させる事で一つの金属平面間に荷電粒子を繰り返し入射させているが、この場合だと金属平面間で加速された荷電粒子は回転軌道を取って右側電極から左側電極へ移動する間に電極間電位差によって減速されてしまうために加速分が相殺されてしまう。そのため、このような手法を用いて一つの金属平面間の静電場で荷電粒子を繰り返し加速させる事を試みても、結果としては荷電粒子を加速させる事が出来ない。故に、単体の静電場で荷電粒子を加速させる手法にはエネルギーの限界が存在する。この問題を解決し、さらなる高エネルギーへの加速器を実現するべく、多重加速の手法が開発され、高周波加速の手法に至った。次項においてこの手法について説明を行う。

2.2.2 高周波加速

高周波加速 (RF) 加速についての説明を行う前に、その源流となった多重加速と呼ばれる加速手法の説明を行う。多重加速とは、先述した静電場による加速で用いた金属平面電極を直線上に複数個配置してそれぞれの電極間で繰り返し加速を行うことで個々の電極における加速のエネルギー的制約を上回る高エネルギーまで加速を行う手法である。また、この手法においては個々の電極に過度の高電圧を印加する必要がなくなるため、電極が放電を起こしてしまう問題も解決する事が出来る。この手法を発展させた加速手法が高周波加速 (RF) 加速である。

高周波加速は、多重加速において各電極に印加していた電圧を加速器中の荷電粒子の運動に対して周期的に変化するように設定した手法である。本研究で注目した KEK Linac もこの高周波加速を電子ビームの加速を用いている。この手法を用いることで通常の多重加速よりも高いエネルギーを得ることが出来る。

高周波加速における電子の加速の原理について述べる。高周波加速による電子の加速では、加速器内部の電子の進行方向に電場の成分を持つ進行波の位相速度を、電子が常に加速位相に乗って加速電場の影響を受けることが出来るように調整している。図2.3に高周波加速による電子の加速の例を示す。図2.3では、左から右へ位相速度 v_p で進む電波を示している。ここで、図中の A-B 間は電子が加速される位相、B-C 間は電子が減速される位相である。この場合、位相速度 v_p を電子の速度 v_e と同期させることで、電子は常に加速位相の影響を受けて加速される。

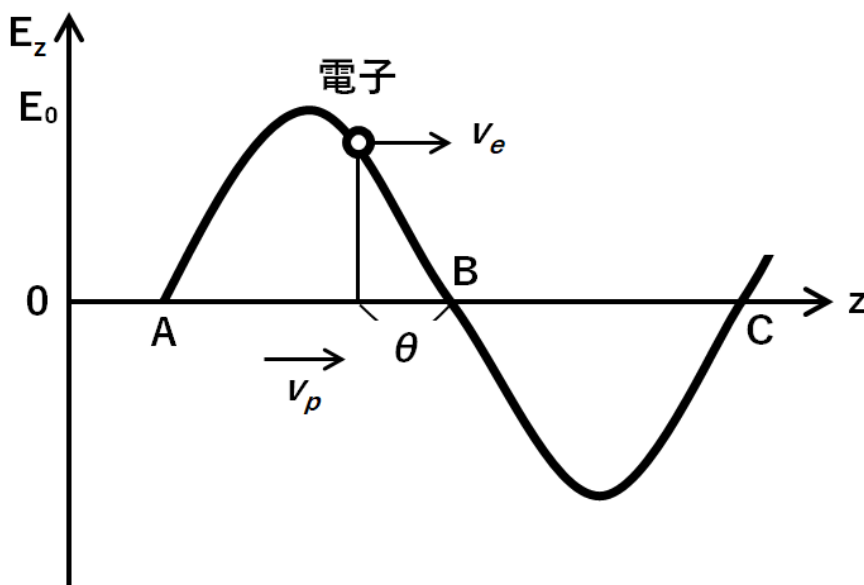


図2.3 加速電波による電子の加速の例

2.3 加速器で使用される電磁石

本節では、加速器で使用される種々の電磁石について、その特徴と使用目的を説明する。

- 偏向電磁石

偏向電磁石はビームの方向を変える電磁石である。荷電粒子は磁力線の中を通過するとフレミングの左手の法則による力を受けその進行方向が曲げられる。偏向電磁石では上下に磁場をかけることで荷電粒子の進行方向を左右に曲げることができる。また偏向電磁石を 90 度傾ければ、上下方向へも向きを曲げられる。線形加速器などの直線加速器ではあまり使用されないが、シンクロトロンなどの円形加速器ではこの偏向電磁石を適切に配置することで、ビームを曲げビームが円形に近い軌道を保つようにしている。円形加速器では一見、なめらかな円形に見えるが、実際は偏向電磁石でビームを曲げている部分以外は直線となっている。

- 4 極電磁石

4 極電磁石は加速器で粒子ビームを細く絞るために使用される電磁石である。N 極の電磁石と S

極の電磁石それぞれ2つずつ、同じ極の電磁石が対面になるように配置する。ビームは同じ電荷を持つ荷電粒子が集まったバンチとなっているため、加速器を通っているうちに、それぞれの粒子が電氣的に反発し合って発散してしまう。またビームは加速中あるいは輸送中にも発散する傾向にある。そのため4極電磁石はビームの発散を防ぎ、ビームを中心軌道周辺の狭い空間に保持するため、ビームを中心に集める働きをする。4極電磁石は中心から遠ざかるに従って磁場が強くなるように製作されているため、発散して中心から外れた粒子をよりビーム軌道中心方向へ曲げて集めることができ、ビームを細く絞ることができる。光学系の凸レンズ、凹レンズと同じ役割を果たすが、四極電磁石1台では水平、垂直方向それぞれに逆のレンズとして働くため、2台以上を組み合わせ使用して使用する。

- 6 極電磁石

6 極電磁石はビームの運動量差に伴う閉軌道のズレや焦点位置のズレの補正に使用する電磁石である。ビームを加速している間の粒子の進行速度は完全に揃っているのではなくばらついており、速度分布をもっている。これより粒子の速さによって4極電磁石から受ける力が異なり、速度の速い粒子ほど受ける力が弱くなる。これによってビームが発散する方向に向かってしまうので、これを防ぎビームを収束させるために6極電磁石を使用する。またこの焦点位置のズレは光学系で色収差とよばれるものと同じである。

- 補正電磁石 (ステアリング電磁石)

補正電磁石は偏向電磁石や四極電磁石でのビーム軌道のズレの補正に使用する電磁石である。偏向電磁石や四極電磁石の電源は強磁場を作り出すために大電流を出せるが、そのため細かな磁場の微調整が苦手である。これらの電磁石が苦手としている調整を行い、また磁石の設置位置が理想的な軌道からどうしてもずれてしまっている誤差の吸収など行う。

2.4 Linac

2.4.1 線形加速器

線形加速器とは高周波加速を用いて加速を行う直線型の加速器である。高周波加速については2.2節で説明した。荷電粒子の走行時間に合わせて電極を並べて加速を行う発想は1920年代からあったが、当時は電波技術が未発達であったことと、イオン収束の研究が進んでいなかったために実現されなかった。第2次世界大戦後に電波技術が急速に発展したことで、荷電粒子を加速することに成功した。

線形加速器では加速する荷電粒子により構造が異なる。ここでは本研究で使用した線形電子加速器について簡単に説明する。電子を線形加速器で加速する場合、80keVのエネルギーですでに $\beta = \frac{v}{c} = 0.5$ に達するので、電子銃から電子が射出された段階で電子は高速となっている。その後の高周波を用いた加速では、電子の進行方向に電場をもち、位相速度を電子の速さに合致させた導波管を用いる。円形導波管で、円形の「ひだ」を適当に作って、位相速度を小さくし、しかも電子の加速につれて位相速度が変化するようにする。高周波としては1,000~10,000MHzのマイクロ波を用い、電源はクライストロンを使用する。

線形電子加速器は信頼度を高く、非常に多くの用途で使用されている。加速粒子のエネルギーが1MeV前後のものは医療、材料検査、放射線化学のために使用されることが多く、10MeV以上のものは原子核物理学実験のために使用されることが多い、また数GeV以上のものが素粒子実験に使用される

ことが多い。

また線形加速器は直接加速器実験に使用されるのではなく、しばしば入射器として使用される。入射器の簡単な説明は次項で行う。

2.4.2 入射器

粒子のエネルギーが低いと、加速器内の残留気体による散乱、残留磁気による磁場の乱れの影響を受けやすく、また速度変化の幅が大きいため加速周波数の変調の幅が広がる。これらの理由からシンクロトロンでは、加速器に入射する前に荷電粒子のエネルギーを予めある程度の高さまで上げておく必要がある。このような目的で使われる前段の加速器を入射器と呼ぶ。一般的に線形加速器が用いられる。

2.4.3 Linac 加速器

Linac 加速器は KEK にある全長 600m の線形電子・陽電子加速器である [4],[5][6],[7]。入射器として使用され、電子を Photon Factory (PF)[8]、PF Advanced Ring (PF-AR)[9]、SuperKEKB 加速器電子リング (HER)[10] へ入射し、陽電子を SuperKEKB 加速器陽電子リング (LER)[10] へ入射させている。入射先の加速器の要請により、異なるエネルギーの粒子を異なる間隔で振り分け入射することが可能となっている。また運転性能としては最大 50Hz、2 バンチ（96ns 間隔）運転が可能である。図2.4に Linac 加速器及び、Linac 加速器から入射される種々の加速器の配置図を示す。

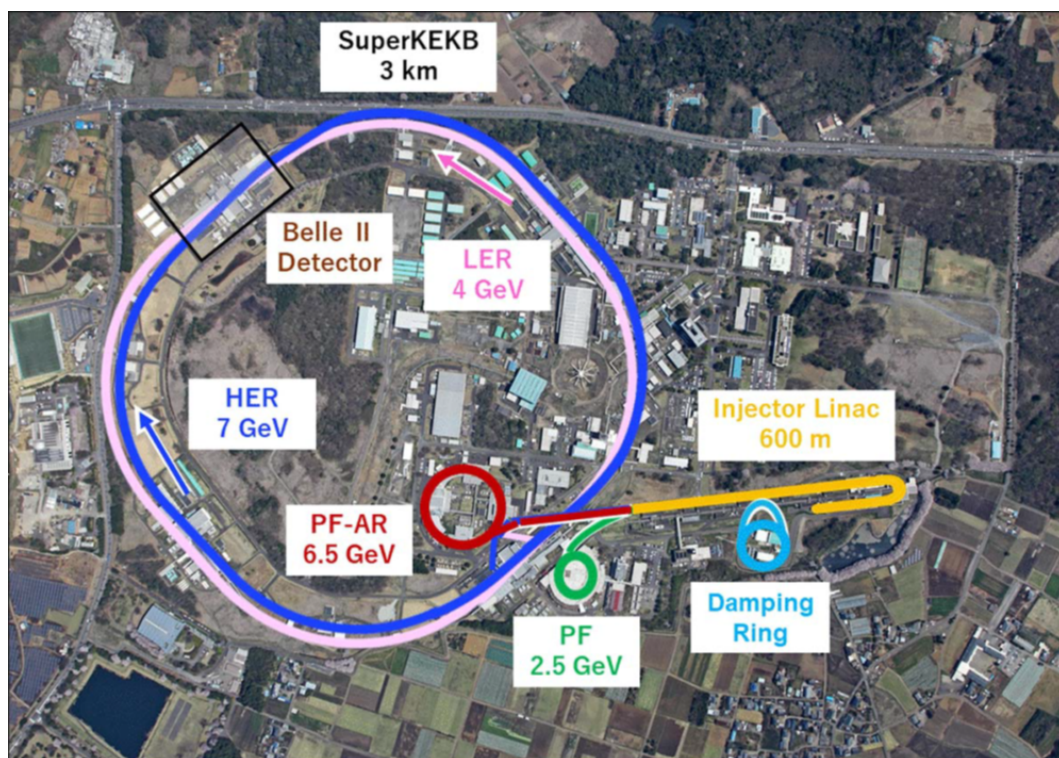


図2.4 KEK における加速器の配置図

また図2.5に Linac 加速器の全体配置図を示す。加速器の上流が A Sector となっており、一番下流が 5 Sector となっている。また Linac 加速器は線形加速器であるが、図のように厳密な直線ではなく J のような形をしている。建設当初の Linac 加速器では 1 Sector から 5 Sector までの直線状となっており、エネルギーも 2.5GeV であった。そこから super KEKB 加速器の前身となる KEKB 加速器の入

射器としてエネルギーを 8GeV に upgrade した際に A Sector から C Sector が増設され、形状も J のような形になった。その歴史背景から厳密に直線でなくなった今でも線形加速器とされている。また super KEKB 加速器の入射器として upgrade した際に、陽電子のために 2 Sector と 3 Sector の間の D Sector に陽電子ビーム用のダンピングリングを増設した。

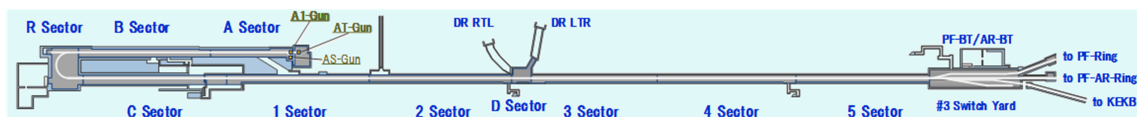


図2.5 Linac 加速器の配置図

第 3 章

機械学習

本章では、3.1節で機械学習について簡単に紹介し、3.2節で今回使用したディープニューラルネットワークと前提知識となるニューラルネットワークの説明を行う [11]。また、3.3節,3.4節でそれぞれ、本研究で用いた機械学習の手法である変分オートエンコーダ (VAE)[12] と敵対的生成ネットワーク (GAN)[13] についての説明を行う。

3.1 機械学習について

機械学習とはその字が表す通り「機械（コンピュータ）が学習する」ことで、説明変数（入力データ）をもとに学習を行い、そのデータの持つパターンを抽出することで目的変数（正当値、出力データ）を出力するモデルを構築する手法である。また機械学習の種類は主に「教師あり学習」「教師なし学習」「強化学習」の3つに分類される。

- 教師あり学習

教師あり学習は入力データと正答値（ラベルと呼ばれることが多い）が既知のデータセットを用い学習を行い、入力データと正答値の関係からモデルを構築する手法である。あらかじめ正答値が付与されていることから「教師「あり」学習」と呼ばれる。一度モデルを構築すれば、未知の入力データに適用し、出力値として正答値の予測を行うことが可能である。またモデルが出力する値の種類によって「分類学習」と「回帰学習」がある。分類学習は学習の結果として離散値を出力する学習である。例えば MNIST と呼ばれる手書き数字の画像認識が有名である。手書き数字の画像データを入力データとし、その画像が 0 から 9 までのどの数字を表すかを正答値として、画像データからそこに書かれている数字の予測を行うことで画像分類を行う。一方、回帰学習は学習の結果として連続値を出力する問題である。例えばある土地の気温と湿度のデータがある場合、そのデータから学習を行ったモデルを使用し、入力データとして気温（あるいは湿度）を入力することで、湿度（あるいは気温）を連続値として出力し予測することが可能である。本研究では、この教師あり学習の中で回帰学習の手法を加速器データから入射効率を予測する手法として利用している。

- 教師なし学習

教師なし学習は入力データからそのデータの構造や特性などを学習し、特徴を見つけ出す手法である。教師あり学習と異なり、あらかじめ正答値が付与されていないことから「教師「なし」学習」と呼ばれる。また教師あり学習と同様に、一度モデルを構築すれば未知の入力データに適用し、以前に学習させたデータと同じ特徴を見つけ出すことが可能である。次元削減やクラスタリ

ングなどを目的に用いられることも多い。例えば、ある店舗での購入者の購入情報をもとに、購入者のグルーピングを行う際に使用される。またデータに正答値が含まれていないため、モデルの正当性を判断する際は、抽出したパラメータや相関から正当性の判断を行うことが可能である。本研究で利用している変分オートエンコーダ (VAE: Variational AutoEncoder) や敵対的生成ネットワーク (GAN: Generative Adversarial Networks) も教師無し学習に分類されている。

- 強化学習

強化学習とは環境駆動型の学習である。教師あり学習や教師なし学習のような固定的で明確なデータを基にした学習ではなく、プログラム自体が与えられた環境を観測し、各試行での行動の評価を行うことでモデルの更新を行い、連続した一連の試行の結果、「報酬が最大となる行動」を学習する手法である。例えば、倒立振り子（例：箒や定規を手の上で直立、維持させる遊び）について、倒立を維持し続けるモデルがある。これは振り子の傾きや加速度から支点の位置を調整し、倒立を維持できた場合に正の報酬を与え、倒立が崩れた場合に負の報酬を与えることで、振り子を維持し続けるモデルを構築することが出来る。社会での使用例としては、自動車の自動運転技術やゲームの自動操作がある。

3.2 ニューラルネットワーク

3.2.1 ニューラルネットワークの構造

ニューラルネットワークとは、脳神経細胞ニューロンを模した基本単位と、そのつながりを数式的なモデルで表現したものである。ニューラルネットワークを図で表すと、図3.1のようになる。

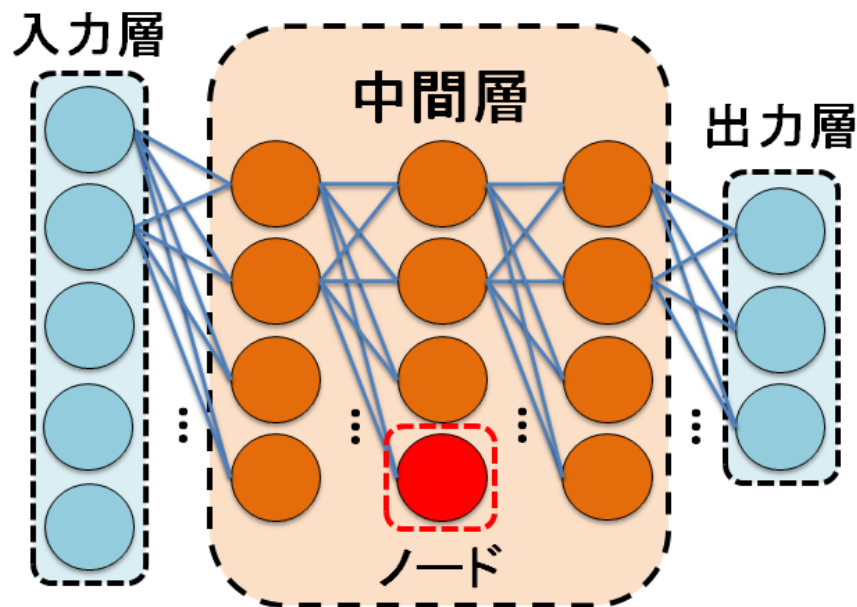


図3.1 ニューラルネットワークの例

図3.1の模式図において、左端の列を入力層、右端の列を出力層と呼び、中間部分の列を中間層と呼ぶ。また、図3.1を構成する丸はノードと呼ぶ。ノードは入力信号を受け取った際にそれぞれに固有の重

みが計算された値を出力する。ノード単位での計算の詳細は3.2.2項で述べる。また、ニューラルネットワーク全体では各ノードからの出力が次のノードへと伝播することで計算が進められる。加えて、ニューラルネットワークの層数や各層でのノード数などはハイパーパラメータと呼ばれる。ハイパーパラメータの詳細については3.2.6項に譲ることとする。

なお、本研究で用いたニューラルネットワークについては、オープンソースライブラリであるKeras[15]やTensorFlow[14]を用いて実装した。

3.2.2 ニューラルネットワークの計算

ニューラルネットワークでどのように計算が行われるかを詳しく記載する。図3.2はニューラルネットワークの中の1つのノードで行われる計算を図示したものである。

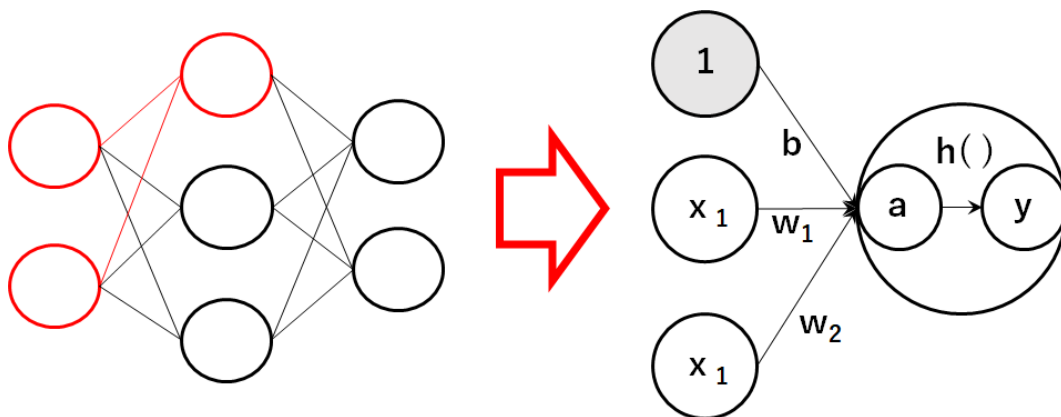


図3.2 ニューラルネットワークの計算の様子

左図はニューラルネットワークの全体図を、右図は左図の赤色の部分の拡大図を示している。ここで x_1 と x_2 と 1 の 3 つの信号がノードへの入力となっている。また b はバイアス、 w_1 と w_2 は各信号の重みを表している。ノードの中に書かれている a は重み付きの入力信号とバイアスの総和を計算したものであり、 $h()$ は活性化関数と呼ばれるものである。ここで x_1, x_2 の入力データが w_1, w_2 のフィルターを通して、次のノードへ伝わっていることがわかる。またバイアスとはノードの発火しやすさを制御するパラメータとなっている。これらが足し合わされ次のノードへの入力となっている。受け取った入力を数式で表すと式3.1となり、この入力を活性化関数を通して出力することで、次のノードへの入力となる。 y を出力とすると式3.2となる。

$$a = b + w_1 x_1 + w_2 x_2 \quad (3.1)$$

$$y = h(a) \quad (3.2)$$

活性化関数 $h(a)$ は様々なものが考案されているが、本研究で使った関数を含めて代表的な関数を次項で紹介する。

3.2.3 活性化関数

活性化関数として本研究で使った ReLU 関数と同じく代表的なシグモイド関数について紹介する。

- ReLU 関数

ReLU (Rectified Linear Unit) 関数は入力が 0 を超えれば、その入力をそのまま出力し、0 以下

ならば 0 を出力する関数である。入力データとして余分な情報（ノイズ）が多いデータについて、負の値の出力を切り捨てることで、より特徴をつかみやすい効果を期待できる。

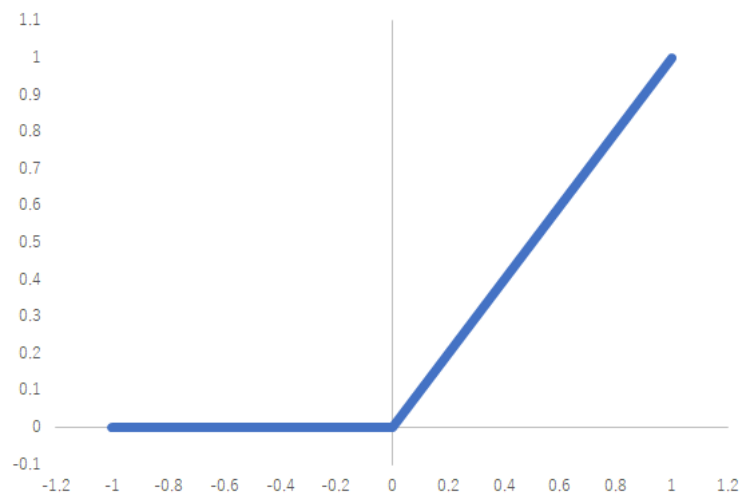


図3.3 ReLU 関数

ReLU 関数を数式で表すと、以下のようになる。

$$h(x) = \begin{cases} x & (x > 0) \\ 0 & (x \leq 0) \end{cases} \quad (3.3)$$

- シグモイド関数

シグモイド関数は入力した値が大きければ大きいほど 1 に近づき、小さければ小さいほど 0 に近づく関数である。特徴として ReLU 関数と異なり滑らかな曲線となっていることである。これにより入力の値を全て使用することが可能となる。

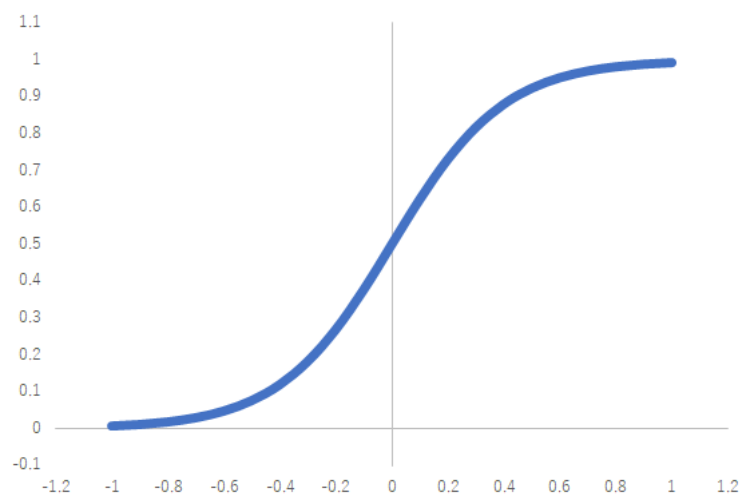


図3.4 シグモイド関数

シグモイド関数を数式で表すと、以下のようになる。

$$h(x) = \frac{1}{1 + \exp(-x)} \quad (3.4)$$

3.2.4 ニューラルネットワークの学習

最終的に出力されたデータは目的変数と比較される。そこで目的変数に対し誤差が大きい場合は、中間層での重み w が適切ではないということなので重み w の更新を行う。この重み w の調整を繰り返し、出力結果が正しい値となるようにする作業がニューラルネットワークの学習である。本項では、ニューラルネットワークの学習に重要な損失関数や勾配、学習上の問題とされることの多い過学習と呼ばれる状態とその対策手法について説明する。

損失関数

損失関数 (loss function) とはニューラルネットワークの学習で用いられる指標であり、このニューラルネットワークからの出力と目的変数との誤差を用いて計算を行う。ニューラルネットワークの学習では損失関数から得られる値が小さくなるように重み w を更新する。一般的に回帰問題については 2 乗和誤差 (mean squared error) を、分類問題については交差エントロピー誤差 (cross entropy error) を用いられる。ここでは 2 乗誤差と交差エントロピー誤差について、正答値からの誤差の評価方法について簡単に説明する。

- 2 乗和誤差

2 乗和誤差は数式で次のように表される。

$$E = \frac{1}{2} \sum_k (y_k - t_k)^2 \quad (3.5)$$

ここで y_k はニューラルネットワークの出力、 t_k は目的変数、 k はデータの次元数を表す。

- 交差エントロピー誤差

交差エントロピー誤差は数式で次のように表される。

$$E = - \sum_k t_k \log y_k \quad (3.6)$$

ここで $\log$ は底が e の自然対数を表し、 y_k はニューラルネットワークの出力、 t_k は目的変数、 k はデータの次元数を表す。また目的変数は正答値のインデックスのみが 1 でその他が 0 である、one-hot 表現を用いて表される。例えば、1,2,3 という目的変数があるとすると、one-hot 表現では $[1,0,0], [0,1,0], [0,0,1]$ となる。そのため式 3.6 は実質的に正解の目的変数が 1 に対応するインデックスの値のみ計算することになる。式 3.6 からわかるように y_k が 0 に近づくにつれ (正答値から離れるにつれ) E の値は大きくなり、1 に近づくにつれ (正答値に近づくにつれ) E の値が小さくなるのがわかる。

ミニバッチ学習

ニューラルネットワークが学習を行う際に、損失関数は全ての訓練データを対象として求める必要がある。そのため訓練データの数だけ損失関数が存在し、その和が学習の指標となる。学習に使用される際は、損失関数の和をその数で割った平均の損失関数を使用される。しかし一般的にニューラルネット

ワークに使用されるデータ数は膨大であり、その和を求めるには少々時間がかかってしまう。そこで訓練データの中から一部を選び出し、それを全体の近似として利用する。ここで選び出された塊をミニバッチと呼び、このような学習をミニバッチ学習と呼ぶ。例えば、訓練データが10000個あった場合、ミニバッチのサイズを50とすると200個のデータとなるので、学習時間の高速化が望める。またミニバッチごとに少ないデータでまとめて学習を行いパラメータの更新を行うので、局所解にはまりにくく学習停滞が起きにくい、同時に訓練データにおいて外れ値がある場合に影響を受けやすいという特徴がある。

勾配法

勾配法とはニューラルネットワークにおいて重み w を更新する手法である。ニューラルネットワークにおいて最適な重み w は、損失関数の最小化に必要不可欠である。しかし損失関数は複雑であり、重み w のパラメータ空間も広大であるため、最小値を見つけ出すことは困難である。そこで勾配を利用し、関数の最小値を探し出す手法が勾配法である。各地点において関数の値を最も減らす方向を示すのが勾配とし、現在の位置から勾配方向に一定距離だけパラメータを更新する。この作業を繰り返すことで、関数の値を徐々に減らすとができる。

勾配法を数式で表すと以下になる。

$$w = w - \eta \frac{\partial f}{\partial w} \quad (3.7)$$

ここで η は更新の量、 f は最小にしたい関数を表す。ニューラルネットワークの学習においては学習率 (Learning rate) と呼ばれる。式3.7は1回の更新式を示しており、このステップを繰り返し行うことで目的の関数の値を徐々に減らしていくことができる。

また勾配法を行う最適化アルゴリズム (オプティマイザー) は多数存在する。以下に例としてニューラルネットワークでよく使用される SGD、RMSprop、本研究で使用した Adam について説明する。

• SGD

SGD とは stochastic gradient descent(確率的勾配降下法) の略である。数式で表すと式3.8となる。

$$w^{t+1} \leftarrow w^t - \eta \frac{\partial f(w^t)}{\partial w^t} \quad (3.8)$$

ここで更新する重みパラメータを w 、 w に関する損失関数の勾配を $\frac{\partial f}{\partial w}$ である。また η は学習係数を表している。また式中の $\leftarrow$ は右辺の値で左辺の値を更新するということを意味している。SGD は勾配方向へのある一定距離だけ進むという単純な方法である。また w の更新ごとにランダムにサンプルを選び直すことで局所的な最適解への収束を防いでいる。

• RMSprop

RMSprop とは Adagrad という学習率の変化を可能にしたオプティマイザーの改善策である。数式で表すと式3.9となる。

$$\begin{aligned} h_t &= \alpha h_{t-1} + (1 - \alpha) \nabla f(w^t)^2 & (h_0 = 0) \\ \eta_t &= \frac{\eta_0}{\sqrt{h_t} + \epsilon} \\ w^{t+1} &\leftarrow w^t - \eta_t \nabla f(w^t) \end{aligned} \quad (3.9)$$

ここで更新する重みパラメータを w 、 η は学習係数を示し、式中の $\leftarrow$ は右辺の値で左辺の値を更新するということを意味している。また α は過去の勾配の影響を抑えるパラメータ、 ϵ は正の微

小量となっている。過去の勾配を記憶し、過去の勾配から学習係数の自動調整を行う。また過去の勾配の記憶は徐々に失われ、新しい勾配の情報が大きく反映されるようになっている。これにより学習が進むにつれ学習係数を次第に小さくできるので、学習の収束を早める効果が得られる。

- Adam

Adam とは Adaptive moment estimation の略である。数式で表すと式3.10となる。

$$\begin{aligned} m_{t+1} &= \beta_1 m_t + (1 - \beta_1) \nabla f(w^t) \\ v_{t+1} &= \beta_2 v_t + (1 - \beta_2) \nabla f(w^t)^2 \\ \hat{m} &= \frac{m_{t+1}}{1 - \beta_1^t} \\ \hat{v} &= \frac{v_{t+1}}{1 - \beta_2^t} \\ w^{t+1} &\leftarrow w^t - \alpha \frac{\hat{m}}{\sqrt{\hat{v}} + \epsilon} \end{aligned} \quad (3.10)$$

ここで更新する重みパラメータを w 、 α は学習係数を示し、式中の $\leftarrow$ は右辺の値で左辺の値を更新するということを意味している。また β は過去の勾配を抑えるパラメータ、 ϵ は正の微小量となっている。RMSprop と同様に過去の勾配を記憶し、過去の勾配から学習係数の自動調整を行う。RMSprop と異なり過去の勾配を指数関数的に減衰させることによって、学習の収束をさらに早めることができる。また既存のオプティマイザーの利点を集めているので、現在最も評価されているオプティマイザーとなっている。

過学習

ニューラルネットワークの学習においての問題の1つに過学習がある。過学習とはニューラルネットワークが学習を進める際に、訓練データだけに適応しすぎてしまい、訓練データに含まれない他のデータにはうまく対応できない状態を言い、汎用性の高いモデルを目指す上で問題となる。また過学習が起きる原因としては主に以下が挙げられる。

- パラメータを大量に持ち、表現力の高いモデルであること
- 訓練データが少ないこと

またここで過学習を抑制する手法として一般的な手法である荷重減衰と、本研究で使ったドロップアウトを以下で説明する。

- 荷重減衰 (Weight decay)

荷重減衰 (Weight decay) とは過学習抑制において古くから使用される手法である。過学習は重み W が大きな値をとることで発生することが多いことから、学習の過程において大きな重み W を持つことに対してペナルティを課すことで過学習を抑制する。ニューラルネットワークの学習において、損失関数に重み W の 2 乗ノルム (L2 ノルム) である $\frac{1}{2} \lambda W^2$ を加算する。ここで λ は正則化の強さを表す。 λ を大きく設定すればするほど、大きな重み W に対し強いペナルティを科すことができる。

- ドロップアウト (Dropout)

ドロップアウト (Dropout) とは中間層のノードをランダムに消去しながら学習する手法である。ニューラルネットワークの学習において、訓練時に中間層のノードをランダムに選出し、その消

去を行う。図3.5左図のニューラルネットワークから、右図のようにノードは消去される。消去されたノードは図3.5右図のように信号伝達が行われなくなる。テスト時には全てのノードに信号を伝達する。ノードをランダムに消去することから過学習を抑制することができる。

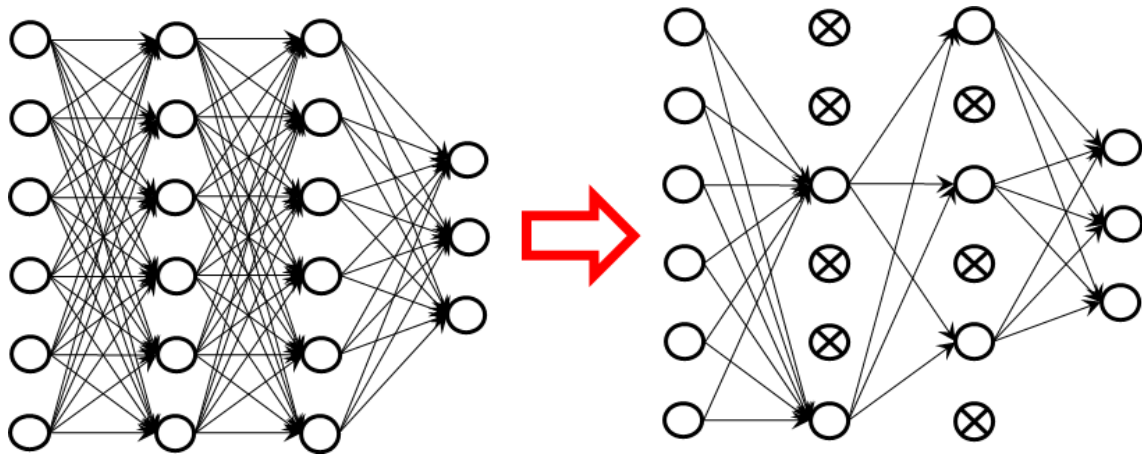


図3.5 ドロップアウトの概念図

3.2.5 ディープニューラルネットワーク

ここまでニューラルネットワークの計算方法や学習方法について述べてきた。ここで、本研究で利用したディープニューラルネットワークとは文字通り、ニューラルネットワークをベースに中間層を増やした、つまり層を深くしたネットワークである。中間層を増やすことで、ニューラルネットワークと比較し以下のような利点がある。

- ネットワークのパラメータ数を少なくすることができる。
中間層を増やすことは、ネットワークの中で中間層が1層のニューラルネットワークを繰り返すことと同じである。これにより1つのニューラルネットワークのパラメータ数を抑えることができる。またパラメータ数の差は中間層を深くするにつれて大きくなる。
- 学習の効率を上げることができる。
中間層が浅いネットワークと比較し、中間層を深いネットワークでは、学習データを少なくすることができ、高速で学習が行うことができる。例えば、「犬」を認識する問題について考えると、中間層が浅いネットワークでは、一度に「犬」の特徴を理解する必要がある。「犬」には様々な犬種があり、また撮影環境に見え方は依存します。そのため「犬」の特徴を理解するためには、多くのバリエーションに富んだ学習データが必要となるので、学習に時間がかかる。中間層が深いネットワークでは学習すべき問題について階層的に分解することが可能なので、それぞれの層が学習すべき問題は単純な問題に落とし込むことができ、学習の高速化が望める。
- 高度なパターン認識を行うことができる。
それぞれの中間層で階層的に情報を渡しているので、中間層から次の中間層への学習において、抽出した情報を効率的に使用することができる。それによってネットワークで学習したモデルが、より高度なパターンを表現することができる。

図3.1に示したニューラルネットワークの模式図は、中間層が複数層になっているため、分類上はディープニューラルネットワークに該当する。

3.2.6 ハイパーパラメータ

ハイパーパラメータとは機械学習アルゴリズムが力を発揮するために必要不可欠なものとなっている。しかし重み W などの学習アルゴリズムが自動で獲得できるパラメータと異なり、人の手で設定する必要があるパラメータのことである。ハイパーパラメータの値を適切な値に設定しなければ、性能の悪いモデルになり正しく学習を行うことができない。以下でハイパーパラメータの詳細について説明する。

- エポック数

エポック (epoch) 数とは一度の学習において訓練データを使用した回数である。エポックとは単位を表しており、学習において訓練データ全てを1度使い切った時が1エポックとなる。エポック数が多いほど訓練データを繰り返し使用するので、学習に時間がかかる。またエポック数が少なすぎると学習を終えることができず、逆に多すぎると過学習を起こす。

- バッチサイズ

バッチサイズとはミニバッチ学習の際のミニバッチの大きさである。バッチサイズが大きいほど学習を行う回数が減るので時間を短縮できるが、学習が局所解にはまりやすくなる。逆にバッチサイズが小さいほど学習を行う回数が増えるので時間がかかるが、学習が局所解にはまりにくくなる。またバッチサイズが大きいほど一度の学習に使用するデータサイズが増える、パソコンのメモリに負荷がかかる、そのためバッチサイズの最大値はパソコンに依存する。

- ドロップアウト率

ドロップアウト率とは、過学習を学習する手法の一種であるドロップアウトを行う比率である。ドロップアウト率が高いほど消去するノードが増えるので過学習を抑制することができる。しかしノードを消去してしまうので、ドロップアウト率が高いほど学習が進みにくくなる。

- オプティマイザーと学習率

オプティマイザーとは勾配法の最適化アルゴリズムであり、学習率とは一度の学習でどれだけの学習を行うか、どれだけのパラメータの更新を行うかを決定する割合である。オプティマイザーにより学習の収束の速さが異なる。また学習率は式3.7において η で表され、学習率が大きいほど学習が早くなり、学習率が小さいほど学習が遅くなる。また学習の結果としてはエポック数と同じ様な効果が得られる。

- 中間層ノード数

中間層ノード数とは各中間層にあるノードの個数である。ノード数が大きいほど様々な特徴を抽出することができるが、余計な特徴まで拾い、またそれぞれノードごとの相互作用が小さくなり学習がうまく進まない場合もある。またノード数が小さいほど入力データの圧縮ができるが、本来必要な情報まで捨ててしまう可能性もある。

- 中間層数

中間層数とは中間層の層数である。中間層の層数が多いほど、より複雑なモデルを学習できる。

しかしニューラルネットワークの学習は誤差の伝播により行われているので、層数が増えるほど正しく誤差を伝播していくことが困難になる。そのため過度に層数を増やすと、局所的な解にばかりやすくなり、過学習を起こす可能性もある。

3.3 変分オートエンコーダ (Variational AutoEncoder)

変分オートエンコーダ (VAE: Variational AutoEncoder) とはデータの特徴を獲得するためのニューラルネットワークである。機械学習の中でも教師無し学習に該当する。また、学習の過程において、獲得した特徴を示す変数を元に入力されたデータの復元を行うことから、生成モデルにも分類される。本節では、3.3.1項において VAE の構造を述べ、3.3.2項において VAE の特性を述べる。

3.3.1 VAE の構造

VAE はエンコーダーとデコーダーと呼ばれる二つのニューラルネットワークから構成されている。エンコーダーは入力である画像などの高次元データの特徴を抽出し、それを表現する任意次元の変数を出力するネットワークである。なお、この特徴を表現する任意次元の変数を潜在変数と呼ぶ。一方で、デコーダーはエンコーダーが出力した潜在変数を入力として、元データを復元するニューラルネットワークだ。図3.6に VAE の例を示す。図3.6において、上段のニューラルネットワークがエンコーダー、下段のニューラルネットワークがデコーダーを示している。

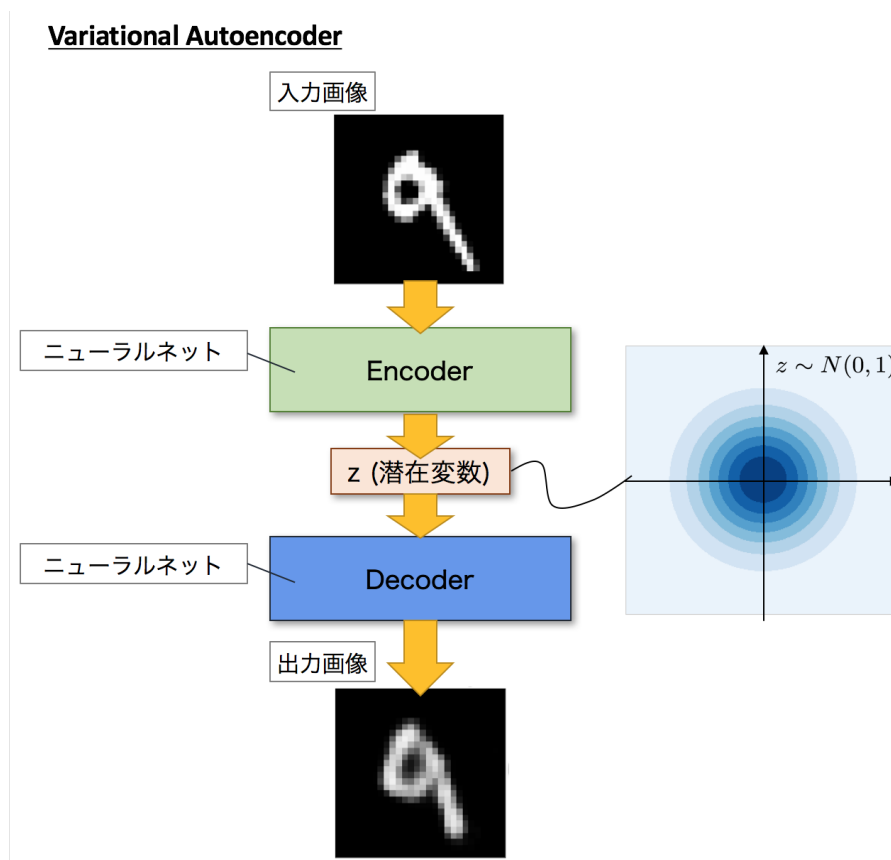


図3.6 VAE の模式図

3.3.2 VAE の特性

VAE では、エンコーダーの出力である潜在変数 z に確率分布を仮定している。通常、 $z \sim N(0, 1)$ と仮定される。このことによって、類似する傾向や特徴を持つ入力データから得られた潜在変数は、パラメータ空間上において近接して分布する特性を持つ。この特性によって、高次元データの類似度を潜在変数のパラメータ空間上での分布によって類推する事が可能となる。本研究では、加速器データを VAE によって 2 次元の潜在変数へと変換し、そのマッピングを行うことで加速器データの可視化を行っている。

3.4 敵対的生成ネットワーク (Generative Adversarial Networks)

敵対的生成ネットワーク (GAN: Generative Adversarial Networks) は学習時に利用したデータを元として、同じ構造の模倣データの生成を行うニューラルネットワークである。教師無し学習に分類され、また目的から生成モデルとも分類される。GAN は競合する二つのネットワークである、生成ネットワーク (Generator) と識別ネットワーク (Discriminator) から構成されている。本節では 3.4.1 項で生成ネットワークと識別ネットワークの概略について述べる。また、3.4.2 項で GAN の学習について述べる。

3.4.1 GAN の構造

- 生成ネットワーク (Generator)

生成ネットワークはノイズデータ (一様乱数) を入力とし、生成した模倣データを出力するニューラルネットワークである。敵対的生成ネットワークにおけるデータ生成はこの生成ネットワークによるものである。そのため、学習後の敵対的生成ネットワークでデータ生成を行う場合、生成ネットワーク部分のみを抜き出し、ノイズデータを入力してデータ生成を行う。本研究でも、加速器データの模倣データ生成を行う際、学習を行った GAN の生成ネットワーク部分のみを抜きだし、データの生成器として利用している。

- 識別ネットワーク (Discriminator)

識別ネットワークは学習用の実データか生成ネットワークの出力である模倣データを入力とし、それらの真偽判定を出力とするネットワークである。GAN の学習中に利用され、学習終了後は通常利用されない。

3.4.2 GAN の学習

GAN の学習の流れ

図 3.7 に GAN の学習の流れの概略を示す。先述の通り、生成ネットワーク (Generator) はノイズデータの入力から模倣データを生成し、識別ネットワークは実データもしくは模倣データの入力に対して真偽判定を行う。ここで、学習を行う際に、生成ネットワークは識別ネットワークを騙す (実データと誤認させる) ことのできる模倣データを生成できるネットワーク構造の獲得を目的とし、識別ネットワークは実データと模倣データを識別することのできるネットワーク構造の獲得を目的としている。つまり、生成ネットワークと識別ネットワークは互いに競合する関係にある。これが「敵対的」と呼ばれる所以である。GAN の学習において、生成ネットワークの学習と識別ネットワークの学習は通常交互に行われる。これにより、両ネットワークの学習は競合しながら進展し、結果として生成ネットワークは

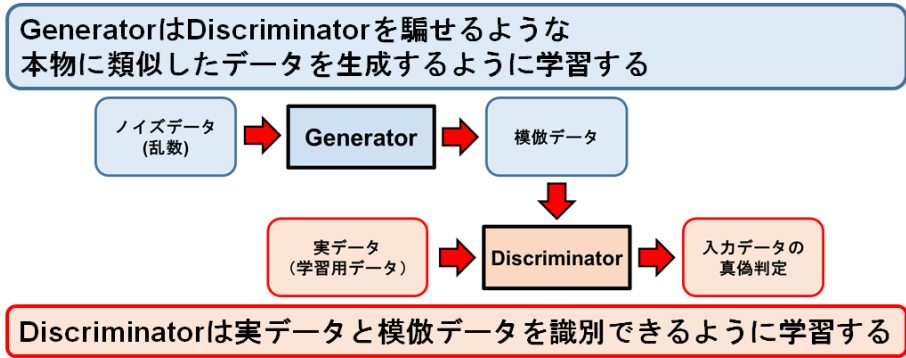


図3.7 GAN の学習の流れ

優れた模倣データ生成能力を獲得することが出来る。

モード崩壊

GAN の学習を行う中で陥りがちな問題として、モード崩壊と呼ばれる現象がある。これは、GAN が学習に用いた実データの一部分のみを模倣してしまう現象だ。先目で GAN の学習の流れを述べたが、GAN の生成ネットワークは識別ネットワークを騙すことが出来るように学習を進めるため、実データの一部分のみを模倣している状態でも、識別ネットワークを騙すことさえ出来ればその方向に学習が収束してしまう。その結果として、モード崩壊と呼ばれる現象に陥ってしまう。例を挙げると、手書き文字データ (MNIST データ) の模倣を GAN で行うために学習を行ったが、学習済みの GAN は一部の文字の画像のみを出力した、といった状態がモード崩壊に該当する。

第 4 章

収集したデータについて

本研究では、加速器運転調整の入力向上に資する加速器運転調整手法の開発を行うために、Linac 加速器の実際の運転データを収集した。具体的には、2018 年 11 月から 2021 年 6 月までの期間に蓄積された Linac 加速器における SuperKEKB 加速器入射用電子ビームデータを利用した。4.1 節で収集したデータの詳細を述べる。また、4.2 節で本研究で定義した入射効率の指標について述べる。

4.1 収集データのパラメータについて

収集データには以下のパラメータデータが含まれている。

- ビーム位置モニター (BPM) 情報 (ビームの位置や電荷量) : 2 個
- 電磁石情報 (電磁石励起電流値読み返し値) : 500 個
- 環境情報 (環境温度や冷却水温など) : 732 個

収集データにおける BPM 情報 2 つは加速器運転における入射効率の指標を計算するために収集した。入射効率指標の定義については 4.2 節で詳細を述べる。

それぞれのパラメータの収集方法については、KEK 側のアーカイブデータを利用させていただいた。しかしながら、電磁石情報と BPM 情報については各 shot 毎に同期したデータを取得することが出来たが、環境情報については 10 秒から 1 分に一度程度の間引かれたデータしか取得することが出来なかった。ただ、環境情報に関しては急激な変化が秒単位で起きているとは考えにくいため、同期している電磁石情報と BPM 情報のタイムスタンプをベースとして、環境情報の値は時系列に沿って補間する事で shot 毎のイベントデータを再構成した。

本研究では、収集データにおいて制御可能なパラメータである電磁石情報のパラメータ群を運転パラメータ、制御できない環境情報のパラメータ群を環境パラメータと呼称している。

4.2 加速器運転における入射効率の指標

本研究の最終的な加速器運転調整の目標は入射効率向上である。入射効率とは前段の加速器から後段の加速器にビームを入射する際、前段の加速器でのビームのうち、後段の加速器で入射されるビームの割合である。そのため入射効率は、ビームの位置やタイミング、粒子の大きさなど様々なパラメータに依存し、また 1shot ごとで決まる量でもない。このように入射効率は様々な要因を含んだ値である。そこで本研究では加速器運転調整における入射効率の指標として、Linac 加速器の上流と下流のビーム位置モニター (BPM) における、電荷量の比 Q_{ratio} を用いた。以降、本研究において特に断りなく入射

効率について言及した場合、 Q_{ratio} を指すものとする。

$$Q_{ratio} = \frac{\text{Linac 加速器下流の BPM(SP_58_0) での電荷量}}{\text{Linac 加速器上流の BPM(SP_A1_M) での電荷量}} \quad (4.1)$$

Q_{ratio} はビームが生成されたところでの電荷量から見た、Linac 加速器から出るところでの電荷量となっている。これより Q_{ratio} が高くなるように調整を行うことで、Linac 加速器の上流から下流までのビーム損失が少なくすることができる。

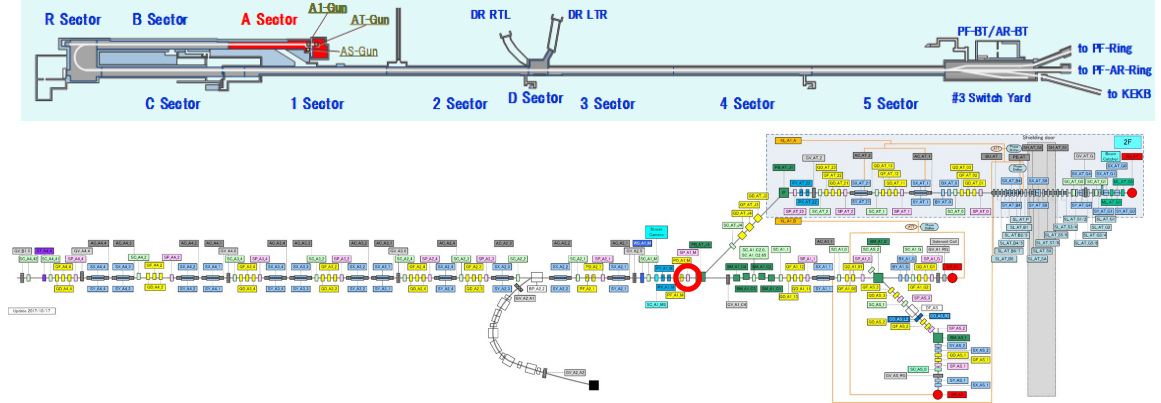


図4.1 Linac 加速器 A Sector の配置図

上図の赤色の部分が A Sector の位置であり、下図が A Sector の拡大図となっている。また下図の右方が Linac 加速器の上流側、左方が下流側である。

下図の赤い円で囲まれた BPM が加速器上流として使用した BPM(SP_A1_M) である。

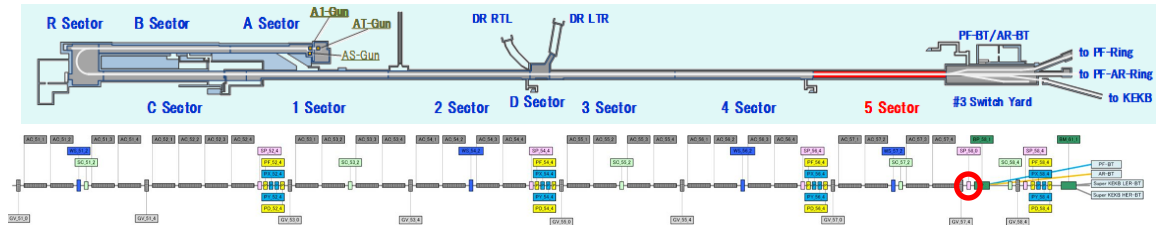


図4.2 Linac 加速器 5 Sector の配置図

上図の赤色の部分が 5 Sector の位置であり、下図が 5 Sector の拡大図となっている。また下図の左方が Linac 加速器の上流側、右方が下流側である。

下図の赤い円で囲まれた BPM が加速器下流として使用した BPM(SP_58_0) である。

Linac 加速器の最下流には Switch Yard が存在するが、経路がわかれており複雑であるため、Linac 加速器下流の BPM としては全てのビームが通る 5 Sector 最後の BPM(SP_58_0) を使用した。

4.3 収集データにおける入射効率の分布

加速器は高い入射効率を維持できるように調整されているため、本研究で収集した 2018 年 11 月から 2021 年 6 月までの期間での蓄積データにおける入射効率は高効率側に偏った分布をしている。図4.3に蓄積データにおける入射効率の分布を示す。

図4.3を見ると、入射効率の平均値は 0.9 程度であることが分かる。つまり、入射効率 0.85 前後の状態から 0.90 以上へと向上させる事が出来るような運転調整が重要だと考えられる。よって、本研究では入射効率が 0.85 以上の中～高効率側に注目した。

また、学習データの均一化を目的として、各開発でデータセットを作成した際には特に言及がない場合、入射効率の分布が均一になるよう抽出を行っている。具体的には、以下3つの区間でデータ数が均一になるようデータの抽出を行っている。

- ・ 入射効率 0.85 以上 0.90 未満
- ・ 入射効率 0.90 以上 0.95 未満
- ・ 入射効率 0.95 以上

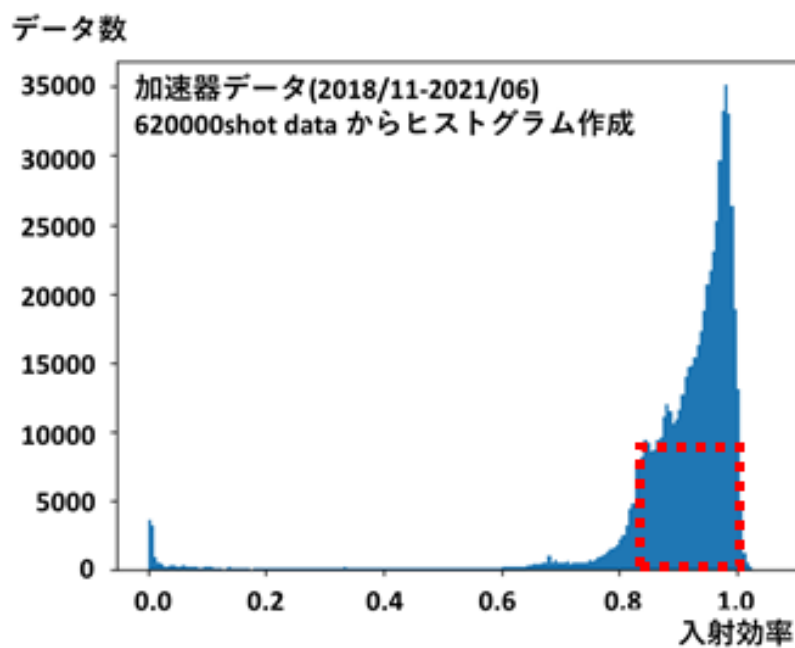


図4.3 2018/11 から 2021/06 までの期間での蓄積データにおける入射効率分布

第 5 章

加速器の状態の可視化手法の開発

本章では、本研究で行った加速器の状態の可視化手法の開発について、その手法と結果を示す。5.1節で可視化手法の方針について述べ、5.2節で加速器の状態の可視化を行うために実装したニューラルネットワークについて述べる。また、5.3節で開発結果の詳細を述べる。

5.1 機械学習を用いた加速器の状態の可視化の方針

本研究では、高次元の加速器データからなる加速器の状態を可視化する手法として、3.3節で紹介した変分オートエンコーダ (VAE) を用いた次元削減 (特徴抽出) を用いた。比較が困難な高次元データを VAE を利用することで特徴を抽出した低次元の潜在変数へと次元削減する。

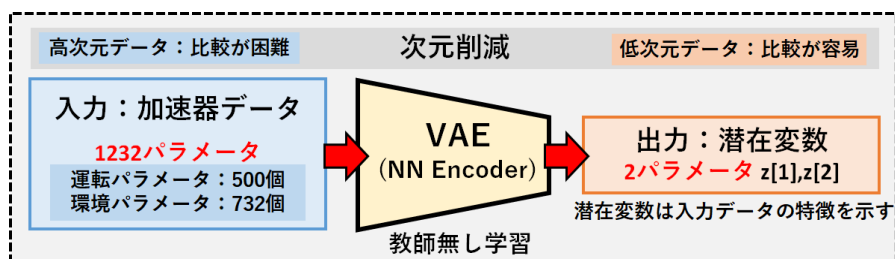


図5.1 VAEによる加速器の状態の可視化の概略

図5.1に加速器の状態の可視化手法の概略を示す。本研究では VAE の出力である潜在変数を 2 次元になるように設定している。そのため、加速器データの入力に対して VAE が出力した 2 次元 (2 パラメータ) の潜在変数 $z[1], z[2]$ をパラメータ平面上でマッピングすることで、加速器の状態の可視化を行う。

5.2 実装したニューラルネットワーク

本節では、本研究で実装した VAE のネットワークについて説明する。3.3節で述べたように、VAE はエンコーダーとデコーダーの二つのネットワークから構成される。実装したエンコーダーとデコーダーの構造を、図5.2と図5.3に示す。

学習時のオプティマイザーは Adam、活性化関数は Relu を用いている。また、学習は 10 エポックで行った。

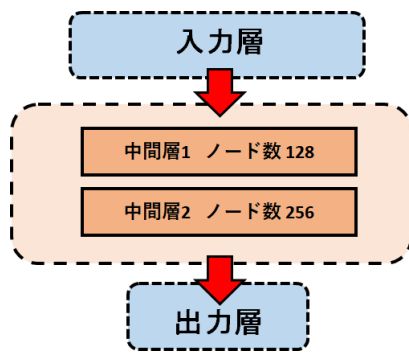


図5.2 実装したエンコーダーの構造

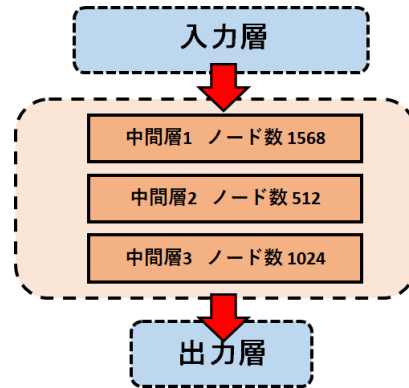


図5.3 実装したデコーダーの構造

5.3 可視化手法の開発結果と考察

5.3.1 VAE を用いた次元削減による加速器の状態の可視化手法の検証

VAE を用いた次元削減による加速器の状態の可視化手法を開発するにあたり、まず初めに学習データと同時期同構造のデータを可視化できるか検証を行った。

使用したデータセット

本研究では、VAE の学習と性能検証用のデータセットとして、2018 年 11 月から 2021 年 5 月に蓄積されたデータから、運転パラメータ 500 個と環境パラメータ 732 個の合計 1232 パラメータのデータを 150,000 event 分抽出し、9 割を学習用データ、1 割を検証用データとした。

- 1233 パラメータ/150,000 event
 入射効率：1 パラメータ (0.85 以上を均一抽出)
 運転パラメータ：500 パラメータ
 環境パラメータ：732 パラメータ

学習データの可視化結果

作成したデータセットの学習用データで VAE の学習を行った。また、学習後の VAE を用いてデータセットの検証用データの可視化を行った。可視化結果を図5.4に示す。

図5.4の横軸、縦軸はそれぞれ、VAE の出力である 2 パラメータの潜在変数 $z[1]$, $z[2]$ を示している。また、プロットの着色と図右のカラーバーは、次元削減前の加速器データが持つ入射効率の値に対応している。

結果として、加速器の状態を示す 1232 パラメータの加速器データを 2 パラメータの潜在変数へと次元削減し、そのマッピングを行うことによって加速器の状態を可視化することが出来た。

可視化結果である図5.4における潜在変数の分布に注目すると、月ごとに密集した分布となっている。ここで、VAE への入力には時期を明確に示すタイムスタンプの様なパラメータは含まれていない。つまり、この結果は、VAE が学習によって加速器の状態の月ごとの違いを学び取り、それが 2 次元の潜在変数へと反映した結果と考えられる。また、潜在変数が月ごとに密集した分布になったことは、短期的 (1 カ月程度) には加速器の状態が類似あるいは連続的に変化している事を示している。加えて、異なる

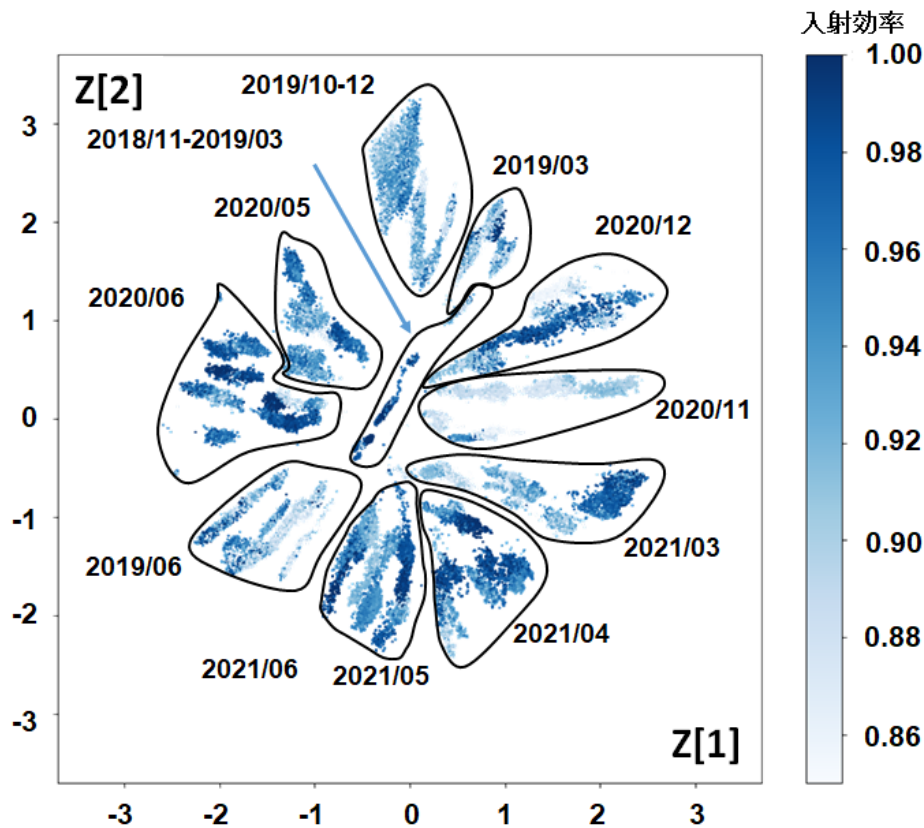


図5.4 2018/11 から 2021/05 までの期間での蓄積データの可視化結果

期間の潜在変数が離れて分布している事からは、長期的 (3 カ月以上、あるいはシャットダウン前後) には加速器の状態に不連続かつ大きな変化が生じている事を読み取ることが出来る。

5.3.2 未知期間のデータに対する可視化結果

前項の結果から、VAE を用いて加速器の状態を可視化できる事が示された。ただ、この結果は学習に用いた期間と同じ期間のデータを可視化したものだ。ここで、VAE による加速器の状態の可視化手法が、学習に用いていない未知の期間にも有効かどうか検証を行う必要がある。

そこで、前項で VAE の学習を行った際のデータセットに含まれていない 2021 年 6 月に蓄積されたデータから 5,000 event 分を検証用データとして抽出し、前項で学習した VAE を用いて加速器の状態の可視化を行った。結果を図5.5に示す。

図5.5は図5.4の結果に検証用データ (2021/06) の可視化結果を重ねてプロットしたものだ。結果を見ると、VAE は検証用データについても同様に潜在変数へと変換して可視化する事が出来ている。よって、学習に用いていない期間のデータであっても、VAE を用いた次元削減によって加速器の状態を可視化できると示された。

また、検証用データの可視化結果に注目すると、その潜在変数は、図5.4における 2021 年 5 月のデータの潜在変数に近接して分布していることが分かる。これは、加速器の状態が短期的 (1 カ月程度) では

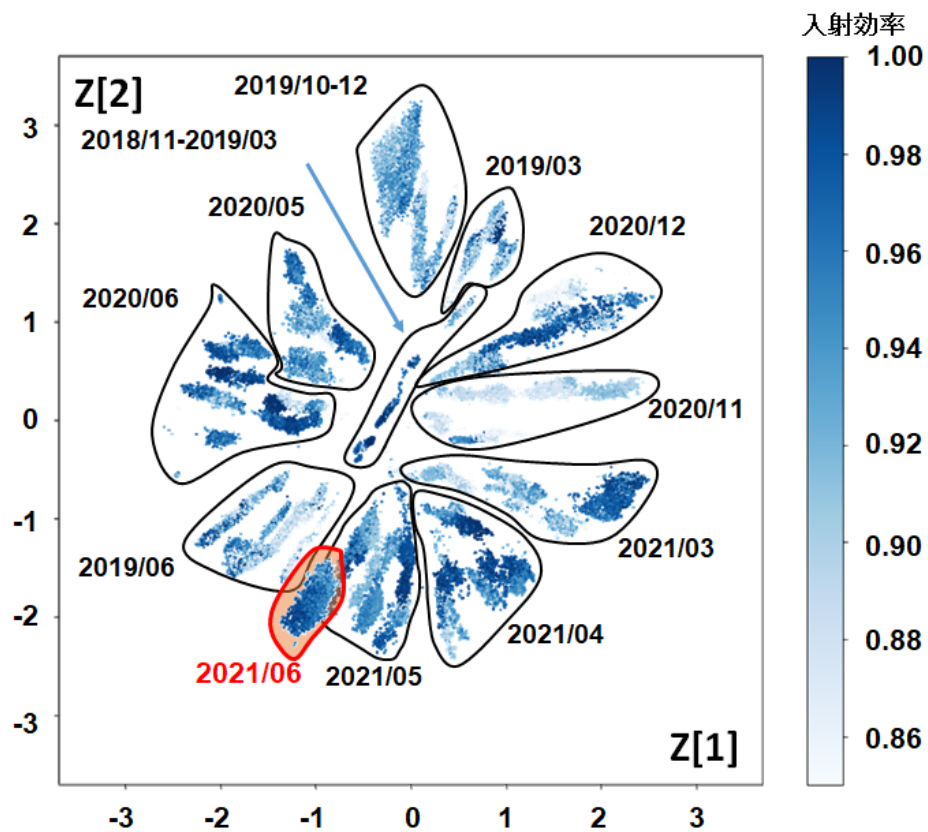


図5.5 学習に用いていない 2021/06 での蓄積データの可視化結果

類似している、という結果を肯定するものである。

第 6 章

加速器調整におけるパラメータ群と入射効率の相関関係の記述手法の開発

本章では、本研究で行った加速器調整におけるパラメータ群と入射効率の相関関係の記述手法の開発について、その手法と結果を示す。6.1節で可視化手法の方針について述べ、6.2節で加速器の状態の可視化を行うために実装したニューラルネットワークについて述べる。また、6.3節で開発結果の詳細を述べる。

6.1 機械学習を用いたパラメータ群と入射効率の相関関係記述の方針

本研究では、加速器のパラメータ群と入射効率の相関関係を記述する手法として、教師あり学習に分類される回帰型ニューラルネットワークを利用した。そこで、加速器データを入力として入射効率を出力する DNN をデザインした。



図6.1 加速器のパラメータ群と入射効率の相関関係記述の方針

図6.1に研究における加速器のパラメータ群と入射効率の相関関係を記述する手法の概略を示す。教師あり学習は入力データと出力データの間を学習に利用したデータから学び取り、学習モデルを構築する手法である。そのため、加速器のパラメータ群からなるデータを入力として入射効率を出力とする DNN は、学習によって加速器のパラメータ群と入射効率の相関関係を学び取り、学習モデルという形で相関関係を記述することが出来ると考えられる。また、この手法での相関関係の記述は、入射効率の予測精度という形で検証が可能である。

6.2 実装したニューラルネットワーク

本節では、本研究で実装した入射効率の値を回帰する DNN について説明する。実装した DNN の構造を図6.2に示す。

学習時のオプティマイザーは Adam、活性化関数は Relu を用いている。また、学習は 60 エポックで

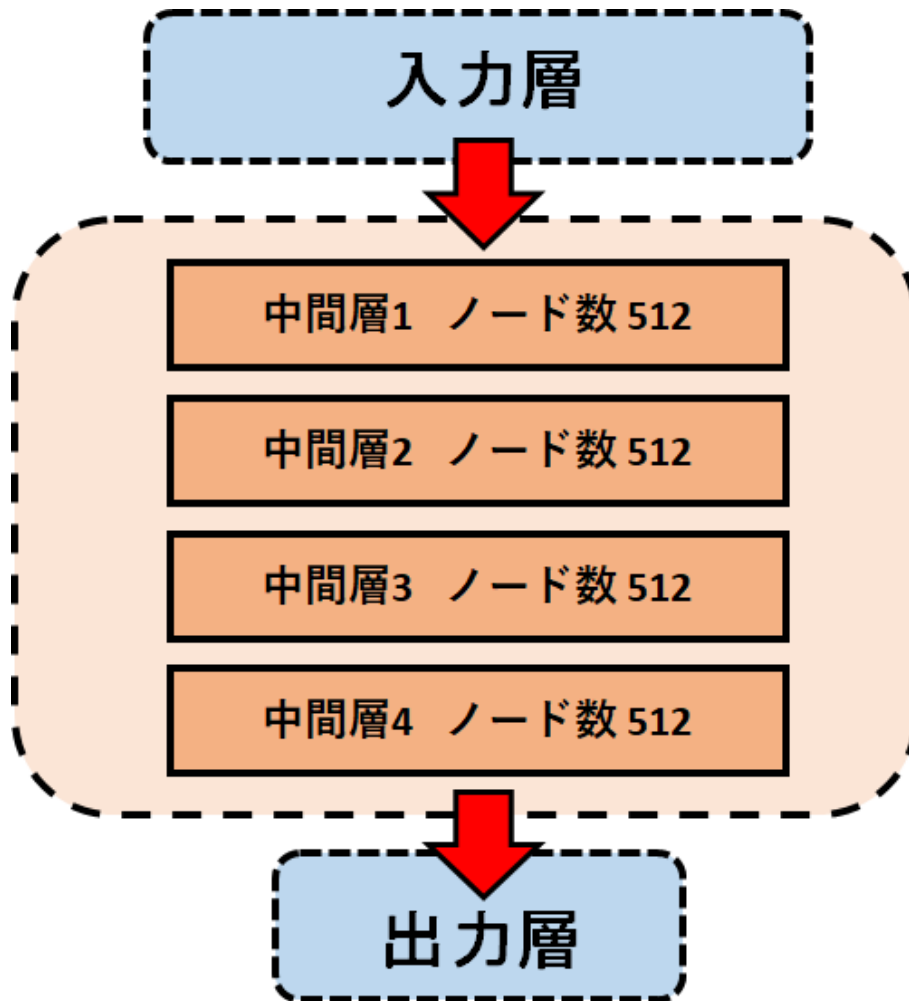


図6.2 実装した DNN の構造

行った。

6.3 加速器のパラメータ群と入射効率の相関関係記述手法開発

6.3.1 DNN を用いた加速器のパラメータ群と入射効率の相関関係記述の検証

DNN を用いた加速器のパラメータ群と入射効率の相関関係記述手法を開発するにあたり、まず初めに DNN が加速器のパラメータ群と入射効率の相関関係を記述できるか検証を行った。

使用したデータセット

DNN を用いた加速器のパラメータ群と入射効率の相関関係記述手法の検証のために、2018 年 11 月から 2021 年 6 月に蓄積されたデータから、運転パラメータ 500 個と環境パラメータ 732 個の合計 1232 パラメータのデータを 150,000 event 分抽出し、9 割を学習用データ、1 割を検証用データとした。データセットの概略を以下に示す。

- 1233 パラメータ/150,000 event
入射効率：1 パラメータ (0.85 以上を均一抽出)

運転パラメータ：500 パラメータ

環境パラメータ：732 パラメータ

データセットによる学習結果

実装した DNN について、前目で詳細を述べたデータセットで学習と性能検証を行った。結果を図6.3に示す。

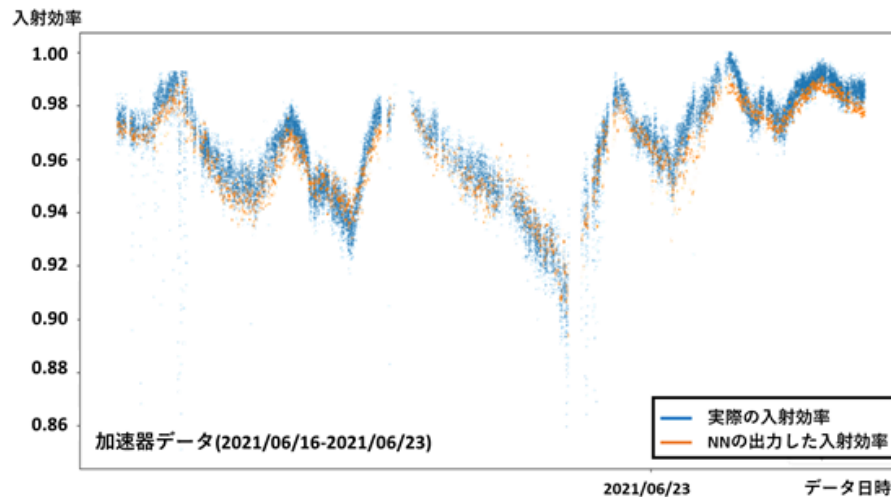


図6.3 2018 年 11 月から 2021 年 6 月の蓄積データによるデータセットでの学習結果のプロット

図6.3は横軸がデータの日時、縦軸が入射効率を示している。また、青色のプロットが検証用データにおける実際の入射効率の値を示し、橙色のプロットが検証用データの入力に対して DNN が予測した入射効率の値を示している。また、この結果は 2018 年 11 月から 2021 年 6 月までのデータを含む検証用データから、2021 年 6 月 16 日から 2021 年 6 月 23 日までの一週間分を抜き出したものだ。

結果より、学習した DNN は入力である加速器データから入射効率の値を正確に予測することが出来ている。これは、DNN を用いて加速器パラメータ群と入射効率の相関関係を記述出来た事を示している。

6.3.2 未知の期間に対しても有効な相関関係記述手法の探索

前節の結果から、DNN を用いることで加速器のパラメータ群と入射効率の相関関係を記述することが出来ると示された、しかしながら、この結果は、学習用データと検証用データが同時期の場合のものだ。実際の加速器運転調整に DNN を用いるためには、DNN にとって未知の期間のデータに対しても同様にパラメータ群と入射効率の相関関係を記述出来る必要がある。そこで、学習用データと検証用データの抽出時期を分け、DNN が未知の期間のデータに対しても有効なパラメータ群と入射効率の相関関係を記述できるかどうか、検証を行った。

使用したデータセット

前章の図5.5の結果より、2021 年 6 月の加速器データは 2021 年 5 月の加速器データと類似した関係にあると類推する事が出来る。本節における検証ではこの結果に注目し、2021 年 6 月のデータを 20,000 event 分抽出して検証用データとした。これは、2021 年 6 月の加速器データと 2021 年 5 月の加

速器データの類似性が学習に与える影響を明らかにするためである。よって、本研究では、2021 年 6 月に蓄積されたデータを検証用データとする一方で、抽出時期の異なる 3 つの学習用データを作成した。図6.3.2に作成した 3 つの学習用データの概略を示す。

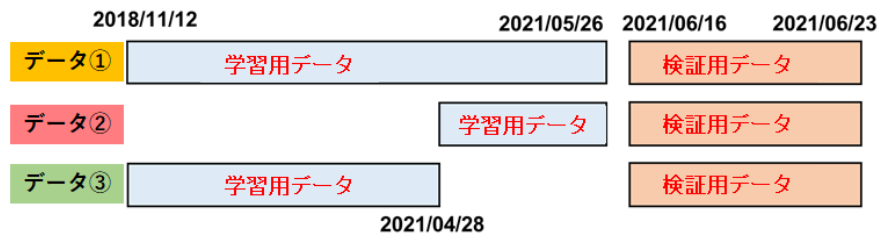


図6.4 学習時期の異なるデータセットの概略

学習用データとして、1. 検証用データの直前 (2021 年 5 月末) までの全期間を含むデータセット、2. 検証用データの直前一カ月分 (2021 年 5 月分) のみを含むデータセット、3. 検証用データの 1 カ月前 (2021 年 4 月末) までのデータのみを含むデータセットを作成した。なお、各学習用データのパラメータ構成は前節で用いたデータセットと同構造なため、詳細については割愛する。

6.3.3 データセットによる学習結果

実装した DNN について、図6.4に示した 3 つのデータセットでそれぞれ学習と性能検証を行った。結果を図6.5に示す。

また、各データセットでの学習モデルについて、検証用データに対する入射効率の予測精度の定量的な指標として、検証用データにおける実際の入射効率の値と検証用データの入力に対して DNN が予測した入射効率の値との間で平均二乗誤差 (MSE) を計算した。結果を表6.1に示す。

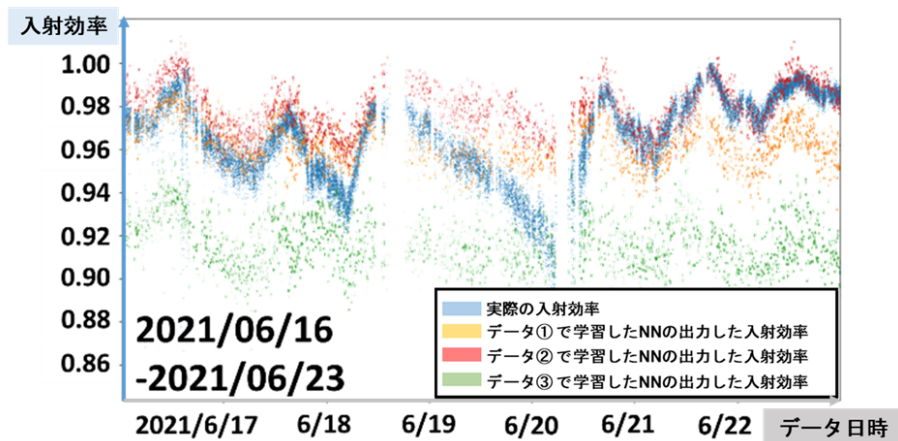


図6.5 図6.4のデータセットでの学習結果のプロット

図6.5は横軸がデータの日時、縦軸が入射効率を示している。また、青色のプロットが検証用データにおける実際の入射効率の値を示し、黄色、赤色。緑色のプロットがそれぞれ図6.4におけるデータ 1, データ 2, データ 3 で学習した DNN が検証用データの入力に対して予測した入射効率の値を示している。

結果より、検証用データの直前までの全期間を含むデータ 1 や、検証用データの直前一カ月分のみを含むデータ 2 で学習した DNN は、学習時には用いていない検証用データにおける 2021 年 6 月のデータに対しても有効なパラメータと入射効率の相関関係を記述し、入射効率を正確に予測することが出来

	MSE
データ 1	0.00038
データ 2	0.00038
データ 3	0.00300

表6.1 図6.4のデータセットでの学習結果における入射効率予測精度

た。一方で、検証用データの直前一月分を含まないデータ 3 で学習した DNN は検証用データに対して入射効率を予測することが出来なかった。これは、学習データから構築したパラメータ群と入射効率の相関関係が、検証データに対しては通用しなかったことを示している。つまり、今回検証データとした 2021 年 6 月のデータに対して有効な加速器パラメータ群と入射効率の相関関係を記述するには、その直前である 2021 年 5 月のデータによる学習が必要なのだと考えられる。ここで、図5.5の結果より、2021 年 6 月と 2021 年 5 月の加速器の状態には類似性が推測されている。このことから、DNN を用いて学習時に含まれていない未知期間に対しても有効な加速器パラメータ群と入射効率の相関関係を記述するには、記述したい時期のデータと類似する傾向を持ったデータを学習用データに含める必要があると思われる。言い換えれば、類似する傾向のデータが学習用データに含まれていれば、未知の期間のデータに対しても DNN を用いて有効な加速器パラメータと入射効率の相関関係を記述することが出来ると考えられる。

第 7 章

加速器の運転シミュレート手法の開発

本章では、本研究で行った加速器の運転シミュレート手法の開発について、その手法と結果を示す。7.1節で開発の方針について述べ、7.2節で加速器の運転シミュレータ開発のために実装したニューラルネットワークについて述べる。また、7.3節で開発結果の詳細を述べる。

7.1 加速器の運転シミュレータの開発方針

本研究では、加速器運転調整手法に強化学習を適用するために必要となる事前学習用の加速器シミュレータとして敵対的生成ネットワーク (GAN) による加速器データ生成を利用した。本研究における GAN の運用についての概略を図7.1に示す。

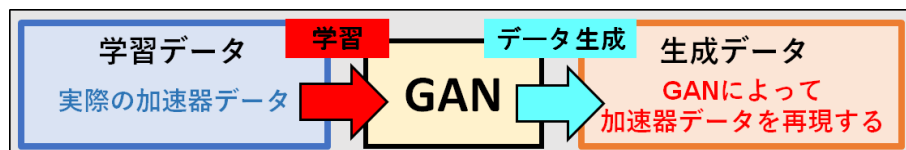


図7.1 GAN を用いた加速器シミュレータ開発の方針

運用としては、まず初めに、加速器の実際の運転データを用いて GAN の学習を行い、その後、学習済みの GAN の生成ネットワーク部分 (Generator) を抜き出してデータ生成器 (加速器シミュレータ) として利用した。

7.2 実装したニューラルネットワーク

本節では、本研究で実装した GAN について説明する。3.4節で述べたように、GAN は生成ネットワーク (Generator) と識別ネットワーク (Discriminator) の二つのネットワークから構成される。実装した Generator と Discriminator の構造を、図7.2と図7.3に示す。

学習時のオプティマイザーは Adam、活性化関数は LeakyRelu を用いている。また、学習は 300 エポックで行った。

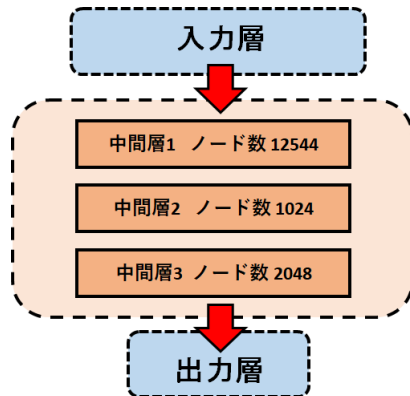


図7.2 実装した Generator の構造

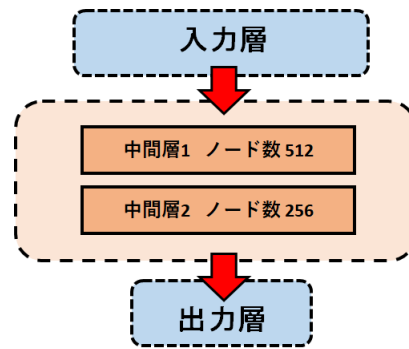


図7.3 実装した Discriminator の構造

7.3 加速器運転シミュレータの開発結果と考察

7.3.1 GAN によるデータ生成の検証

GAN を用いた加速器シミュレータを開発するにあたって、まず初めに GAN が加速器パラメータの傾向を再現した模倣データを生成できるか検証を行った。そのために、1 パラメータのみを含むデータセットによる学習とデータ生成の検証を行った。

利用したデータセット

GAN によるデータ生成の評価を行うために、2018 年 11 月から 2021 年 6 月に蓄積された加速器データから、入射効率のみをパラメータとして含む 600,000 event のデータセットを作成し、その 9 割を学習用データ、1 割を検証用データとした。データセットの概略を以下に示す。

- データセット 1-1 : 1 パラメータ/600,000 event
入射効率のみ (0.85 以上を均一抽出)
- データセット 1-2 : 1 パラメータ/600,000 event
入射効率のみ (蓄積データでの分布を維持)

ここで、入射効率については4.3節で 0.85 以上の範囲において均等な抽出を行ったと述べたが、本項で述べるデータセットに限り、均等な抽出を行ったものと、均等抽出を行わず蓄積データでの分布をそのまま用いたものを併用している。

データ生成の結果

作成したデータセットを用いて、GAN の学習を行った。学習後、GAN の Generator 部分を用いて模倣データの生成を行った。結果を図7.4と図7.5に示す。

図7.4と図7.5は実データ（検証用データ）と GAN による模倣データのそれぞれの分布をヒストグラムにしたものだ。縦軸がデータ数、横軸が入射効率の値を示している。また、図中の青色が実データの分布、橙色が GAN による模倣データの分布に対応している。

結果から、データセット 1-1 で学習した GAN に関してはデータ生成を行う際に入力するノイズデータ (一様乱数) をそのまま出力しているのではないかと考えられるが、データセット 1-2 で学習し

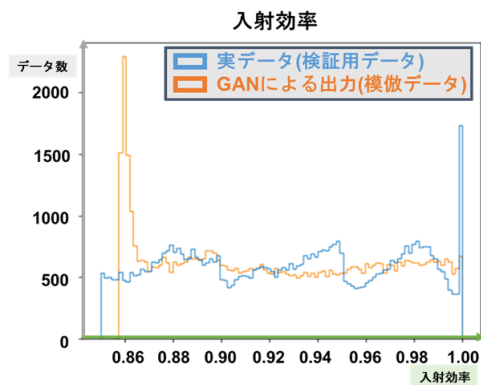


図7.4 データセット 1-1 で学習した GAN の出力

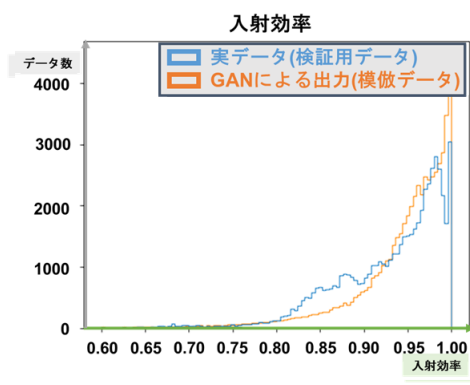


図7.5 データセット 1-2 で学習した GAN の出力

た GAN については実データの分布傾向を学習し、模倣データへ反映する事が出来たと言える。つまり、GAN を用いる事で、特定のパラメータの分布傾向を再現した模倣データを生成することが可能だと示された。

7.3.2 GAN によるパラメータ間の相関関係の再現

7.3.1項に示した結果より、GAN を用いてパラメータの分布傾向を再現した模倣データを生成する事が可能だと分かった。一方、実際の加速器シミュレータとして運用する場合は、個々のパラメータの分布傾向のみならず、複数パラメータ間での相関関係も再現する必要がある。よって、先の検証に続いて、GAN が複数パラメータ間の相関関係を模倣できるか検証を行うために、2 パラメータを含むデータセットによる学習とデータ生成の検証を行った。

利用したデータセット

GAN による模倣データ生成の評価を行うために、2018 年 11 月から 2021 年 6 月に蓄積された加速器データから、入射効率と特定の運転パラメータのみをパラメータとして含む 600,000 event のデータセットを作成し、その 9 割を学習用データ、1 割を検証用データとした。データセットの概略を以下に示す。

- データセット 2 : 2 パラメータ/600,000 event
 入射効率 : 1 パラメータ (0.85 以上を均一抽出)
 運転パラメータ : 1 パラメータ (第 3 セクター 2 番目のステアリング電磁石の励起電流読み値)

抽出する運転パラメータとしては、第 3 セクター 2 番目のステアリング電磁石の励起電流読み値を利用した。

データ生成の結果

作成したデータセットを用いて、GAN の学習を行った。学習後、GAN の Generator 部分を用いて模倣データの生成を行った。結果を図7.6と図7.7に示す

図7.6と図7.7は実データ (検証用データ) と GAN による模倣データの分布をヒストグラムにしたものだ。縦軸がデータ数、横軸がパラメータの値を示している。また、図中の青色が実データの分布、橙色が GAN による模倣データの分布に対応している。

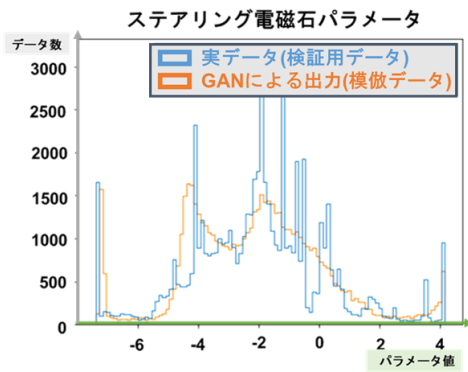


図7.6 データセット 2 で学習した GAN の出力:
運転パラメータ

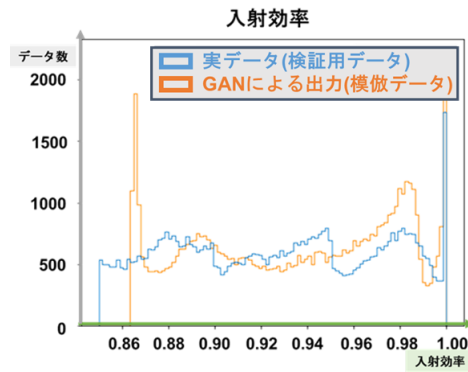


図7.7 データセット 2 で学習した GAN の出力:
入射効率

結果から、7.3.1項で示された結果と同様に、入射効率やステアリング電磁石パラメータの双方について、GAN は個々のパラメータの分布傾向を再現した模倣データを生成することができている。続いて、パラメータ間の相関関係を再現できたか検証を行うために、実データと GAN による模倣データのそれぞれについて、2 パラメータでの二次元ヒストグラムを作成した。結果を図7.8に示す。

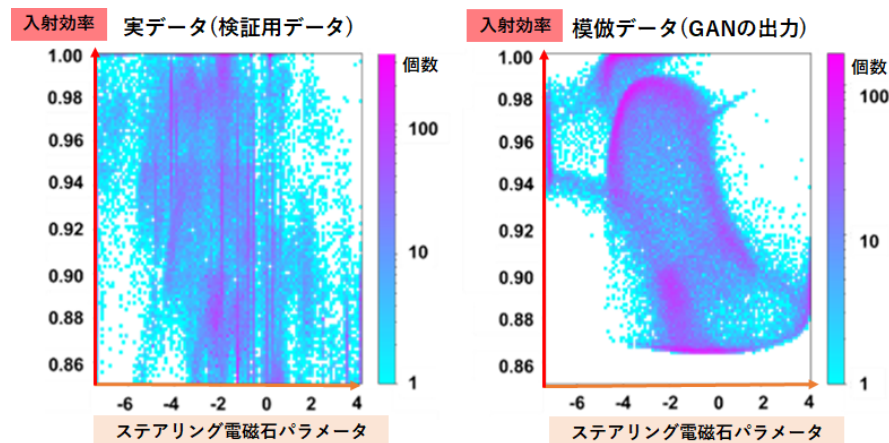


図7.8 実データ (左図) と模倣データ (右図) の二次元ヒストグラム

図7.8は縦軸が入射効率、横軸が運転パラメータ (第 3 セクター二番目のステアリング電磁石パラメータ) の値に対応している。また、着色はデータの分布密度を示し、グラフ右側のカラーバーと対応している。

結果から、GAN による模倣データにおける 2 パラメータ (入射効率とステアリング電磁石パラメータ) の分布は実データでの分布に重複している事が分かる。これは、GAN が実データにおける 2 パラメータの相関関係を学習し、模倣データに反映することが出来た事を示している。よって、GAN は 2 パラメータの分布や相関関係を大まかな傾向としては再現することが出来たと言える。ただ、GAN による模倣データには実データと一致しないピークや分布の偏りが存在する事も見受けられる。また、二次元ヒストグラムにおける分布領域に注目すると、模倣データの分布領域は実データの全範囲を網羅する事が出来ていない事が分かる。これは、GAN が模倣データの生成を行う際に、実データの一部分に偏ったデータを生成してしまった事を示している。

7.3.3 GAN による加速器データ全体の模倣データ生成

前項及び前々項の結果より、GAN は個々のパラメータの傾向を再現しつつ、他のパラメータとの相関関係も再現したデータの生成が可能だと分かった。続いて、加速器データ全体の再現が可能かどうか検証を行った。

利用したデータセット

GAN による加速器データ全体を再現した模倣データ生成の評価を行うために、2018 年 11 月から 2021 年 6 月に蓄積された加速器データから、入射効率と運転パラメータ及び環境パラメータ全てをパラメータとして含む 600,000 event のデータセットを作成し、その 9 割を学習用データ、1 割を検証用データとした。データセットの概略を以下に示す。

- データセット 3 : 1233 パラメータ / 600,000 event
 - 入射効率 : 1 パラメータ (0.85 以上を均一抽出)
 - 運転パラメータ : 500 パラメータ
 - 環境パラメータ : 732 パラメータ

データ生成の結果

作成したデータセットを用いて、GAN の学習を行った。学習後、GAN の Generator 部分を用いて模倣データの生成を行った。ここで、実データ (検証用データ) や模倣データは 1233 パラメータからなる高次元データであり、その分布関係をそのまま比較検討することは困難である。そこで、5.1 節で述べた、VAE による次元削減を利用したデータ可視化手法を利用した。実データと模倣データの可視化プロセスについて図 7.9 に概略を示す。

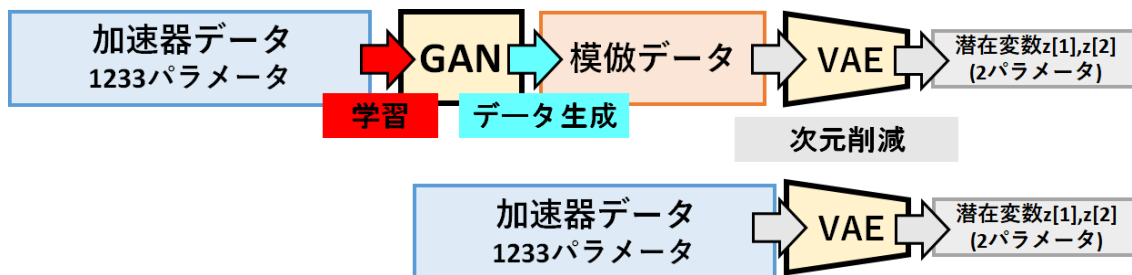


図 7.9 データ可視化を用いた 1233 パラメータの加速器データの比較の概略

1233 パラメータからなる実データ (検証用データ) と模倣データを、VAE による次元削減で、特徴が抽出された 2 次元の潜在変数 $z[1], z[2]$ へと変換する。この潜在変数について二次元ヒストグラムを作成することで、検証用データと模倣データの分布関係の比較を行った。結果を図 7.10 に示す。なお、この際に用いた VAE は、GAN の学習に用いたデータセットと同じもので学習を行っている。

図 7.10 は横軸と縦軸がそれぞれ潜在変数 $z[1], z[2]$ に対応している。また、着色はデータの分布密度を示し、グラフ右側のカラーバーに対応している。

結果から、GAN による模倣データの潜在変数は、実データの潜在変数の分布領域の一部分にしか分布していない事が見て取れる。これは、GAN が実データの一部のみを模倣してデータを生成してしまった事を示している。つまり、GAN の学習がモード崩壊に陥ってしまったと考えられる。この状態で

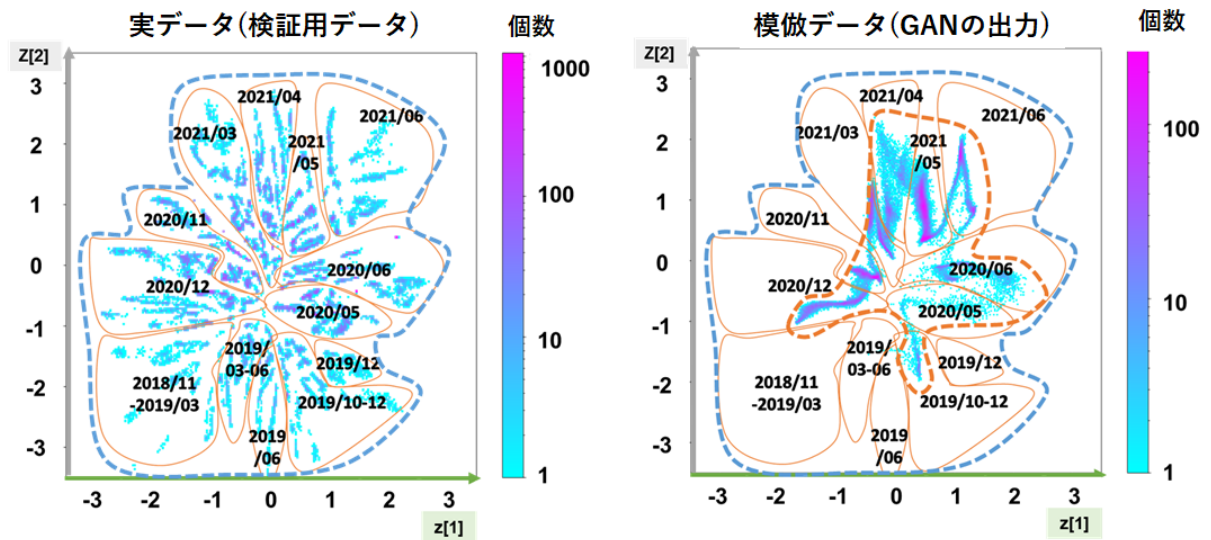


図7.10 データセット3による、実データ(左図)と模倣データ(右図)の潜在変数の二次元ヒストグラム

は、一部に偏ったデータのみを生成してしまうため、加速器シミュレータとして運用する事は出来ない。そこで、モード崩壊を回避する手法についての検討を行った。詳細は次項で述べる。

7.3.4 モード崩壊の回避手法の検討

前項の結果より、1233 パラメータからなる加速器データで学習した GAN は、実データの一部分のみを模倣したデータのみを生成してしまった。これは、GAN の学習がモード崩壊と呼ばれる状況に陥ってしまった事を示している。ここで、学習に利用したデータのパラメータ領域の範囲が、学習がモード崩壊に陥ってしまった原因ではないかと仮説を立てた。前項では 2018 年 11 月から 2021 年 6 月までの約 2 年半の期間からデータの抽出を行っていた。そこで、この期間を短縮し、データのパラメータ領域を削減することで、モード崩壊を回避出来るか検証を行った。

利用したデータセット

データの抽出期間を変更したデータセットを二つ作成し、それぞれの 9 割を学習用データ、1 割を検証用データとした。なお、パラメータ構成については 7.3.3 項で作成したデータセット 3 と同様なので割愛する。データセットの概略を以下に示す。

- データセット 4-1 : 1233 パラメータ/150,000 event
3 カ月分 (2021 年 4 月から 6 月) の蓄積データから抽出
- データセット 4-2 : 1233 パラメータ/120,000 event
1 カ月分 (2021 年 5 月) の蓄積データから抽出

データ生成の結果

作成したデータセットを用いて、GAN の学習を行った。学習後、GAN の Generator 部分を用いて模倣データの生成を行った。ここで、実データ (検証用データ) と模倣データの分布関係の比較検討につ

いては、7.3.3項と同様の手法で行った。また、利用した VAE も 7.3.3 項と同じものを用いた。まず、データセット 4-1 で学習した GAN について、模倣データ生成の結果を図 7.11 に示す。

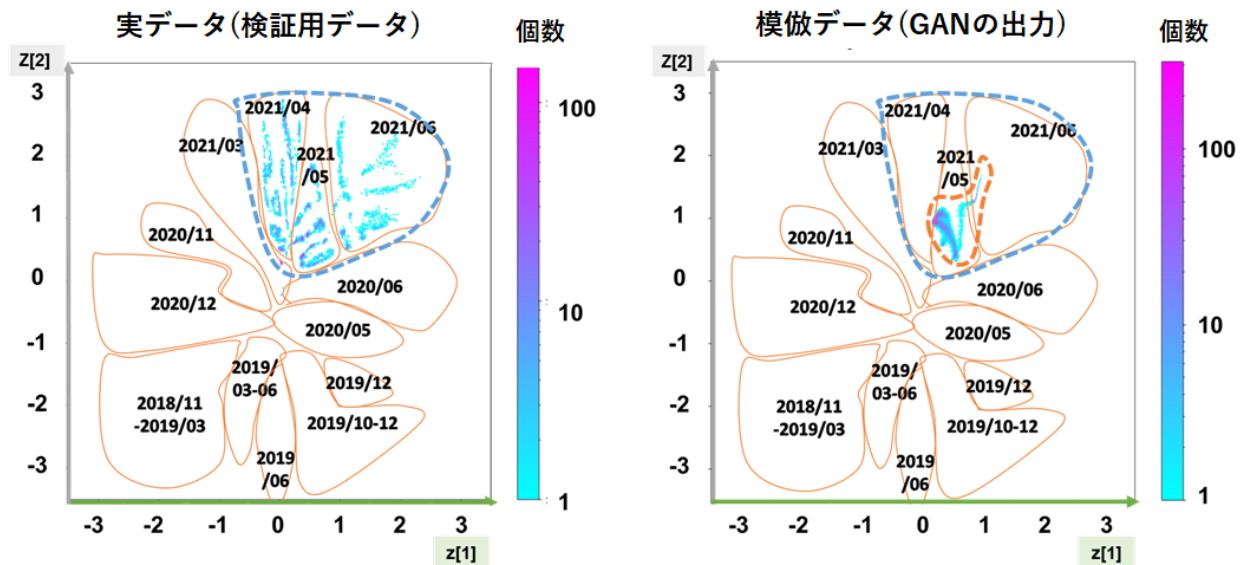


図 7.11 データセット 4-1 による、実データ (左図) と模倣データ (右図) の潜在変数の二次元ヒストグラム

図 7.11 は横軸と縦軸がそれぞれ潜在変数 $z[1]$, $z[2]$ に対応している。また、着色はデータの分布密度を示し、グラフ右側のカラーバーと対応している。

結果より、データセットに利用したデータの時期を 2021 年 4 月から 6 月の 3 カ月に短縮したことにより、図 7.11 での実データの分布領域は図 7.10 に比べて縮小したが、模倣データの分布領域は実データの全体を網羅できず、一部分のみを再現するにとどまっていることを読み取ることが出来る。つまり、GAN の学習に利用したデータのパラメータ領域を 3 カ月分に狭めたが、モード崩壊の回避には至らなかった。

続けて、データセット 4-2 で学習した GAN について、模倣データ生成の結果を図 7.12 に示す。

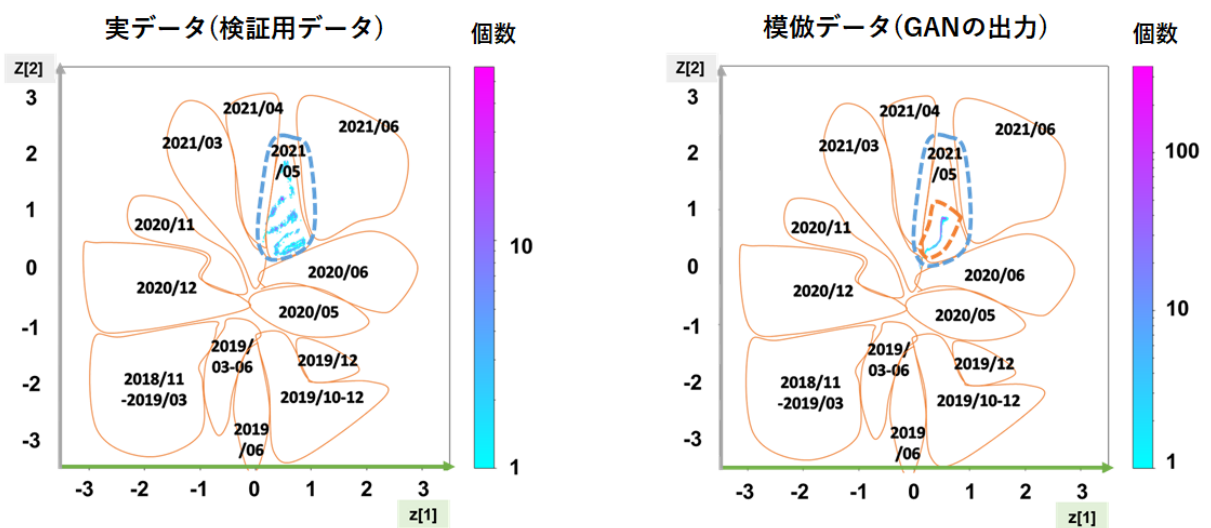


図 7.12 データセット 4-2 による、実データ (左図) と模倣データ (右図) の潜在変数の二次元ヒストグラム

図7.12は横軸と縦軸がそれぞれ潜在変数 $z[1]$, $z[2]$ に対応している。また、着色はデータの分布密度を示し、グラフ右側のカラーバーに対応している。

結果より、データセットに利用したデータの時期を 2021 年 5 月のみの 1 カ月に短縮し、図7.12での実データの分布領域はさらに縮小したが、模倣データの分布領域はなおも実データの全体を網羅できず、一部分のみを再現するにとどまっていることを読み取ることが出来る。つまり、GAN の学習に利用したデータのパラメータ領域を 1 カ月分としても、モード崩壊を回避することはできなかった。

以上の結果より、学習に利用するデータのパラメータ領域を狭めても、モード崩壊を回避することはできなかった。この結果より、モード崩壊を回避するために学習用データのパラメータ領域を狭める事は有効ではないと明らかになった。モード崩壊を回避する方策としては、GAN のネットワーク構成の調整や、学習手法の変更が有効かと考えられるが、本研究ではいずれの手法も実証に至らなかった。

第 8 章

まとめ・今後の方針

本研究では、高エネルギー加速器研究機構 (KEK) で稼働中の、電子・陽電子入射器 Linac における機械学習を用いた加速器運転調整手法の開発を行った。

2018 年 11 月から 2021 年 6 月の期間に蓄積されたデータを用いて、以下の方針で運転調整手法の開発を行った。

1. 加速器の状態の可視化手法の開発
2. 加速器調整におけるパラメータ群と入射効率の相関関係の記述手法の開発
3. 加速器の運転シミュレート手法の開発

1. の開発については変分オートエンコーダ (VAE) を用いた次元削減による加速器の状態の可視化手法を開発した。その結果として、加速器の状態を示す 1232 パラメータからなる加速器データを、その特徴を示す 2 パラメータの潜在変数へと次元削減し、潜在変数のマッピングによって加速器の状態を可視化することが出来た。また、この可視化手法は、学習に用いていない時期のデータに対しても有効であり、加速器の現状が今までのどの時期に類似しているか可視化結果から類推する事が出来た。加えて、可視化結果から、加速器の状態は短期間 (1 カ月) では類似、あるいは連続的な変化をしているが、長期間 (3 カ月以上) では不連続かつ大きな変化が生じている事を示した。

2. の開発については、加速器のパラメータ群を入力として入射効率を出力とする回帰型ディープニューラルネットワーク (DNN) を用いて相関関係を記述する手法を開発した。その結果として、DNN は学習に用いた期間のデータについて加速器の 1232 個のパラメータと入射効率の相関関係を記述し、入射効率を正確に予測することが出来た。また、学習に用いていない時期のデータに対しても有効なパラメータ群と入射効率の相関関係を記述するには、学習用データに予測したい時期と傾向が類似しているデータを含んでいることが重要だと明らかにすることが出来た。

3. の開発については、敵対的生成ネットワーク (GAN) を用いた加速器シミュレータの開発を行った。その結果として、GAN は個々のパラメータの分布傾向や、他のパラメータとの相関関係を模倣することはできたが、学習に用いた実データの一部分しか模倣することが出来なかった。この状態は GAN の学習がモード崩壊に陥ってしまった事を示している。

本研究ではモード崩壊に陥ってしまった原因として、学習に用いたデータのパラメータ範囲に問題があると推測した。そこで、データを抽出する期間を短縮することで、パラメータ領域を狭めたデータセットを 2 パターン作成し、GAN の学習と模倣データ生成の検証を行った。しかし、2 パターンのデータセットでの学習結果は共に、モード崩壊の回避には至らなかった。この結果より、モード崩壊を回避

するために学習用データのパラメータ領域を狭める事は有効ではないと明らかになった。

今後の方針としては、GAN の性能向上について、モード崩壊の回避を主眼においた開発を行う。また、加速器シミュレータの開発に合わせて、加速器実機を用いたテストランを行い、そのデータを元とした加速器運転調整手法の開発も検討している。

参考文献

- [1] 亀井亨・木原元央,「パリティ物理学コース 加速器科学」 丸善株式会社 (1993)
- [2] 熊谷寛夫,「実験物理学講座 28 加速器」, 共立出版株式会社 (1975)
- [3] 神谷幸秀,「加速器の原理 -シンクロトロン及びストレージング-」,OHO '84 高エネルギー加速器セミナー (1984) II-1
- [4] 小川雄二郎,「電子線形加速器の概要」,OHO '02 高エネルギー加速器セミナー (2002) 1-1
- [5] 杉村高志,「KEKB 入射器入門」,OHO '04 高エネルギー加速器セミナー (2004) 5-1
- [6] R. Zhang et al. 「KEK 電子陽電子入射器の現状」 in Proceedings of the 16th Annual Meeting of Particle Accelerator Society of Japan, Kyoto, Japan, Jul.
- [7] M. Sato, “Injector beam operation” ,KEKB Accelerator Review Committee (2019)
- [8] I. Sato, “Accelerator structure and beam transport system for the KEK photon factory injector” Nucl. Inst. Meth 177, 1 (1980), 91
- [9] T. Ieiri et. Al., ”Bunch lengthening recently observed at PF-AR”, Proceedings of the 2003 Particle Accelerator Conference, (2003), 3077
- [10] Z. Dolezal and S. Uno ”Belle II Technical Design Report” High Energy Accelerator Research Organization (2010)
- [11] 斎藤康毅,「ゼロから作る Deep Learning -Python で学ぶディープラーニングの理論と実装」, 株式会社オライリー・ジャパン (2016)
- [12] D. P. Kingma, and M. Welling ”Auto-Encoding Variational Bayes” arXiv:1312.6114 (2014)
- [13] I. J. Goodfellow et al. ”Generative Adversarial Networks” arXiv:1406.2661 (2014)
- [14] TensorFlow <https://www.tensorflow.org/>
- [15] Keras <https://keras.io/>

謝辞

本研究を進めるにあたり、学外・学内を問わず多くの方にご指導ご鞭撻を賜りました。この紙面を借りて御礼申し上げます。

始めに、研究活動や本論文作成について直接ご指導を頂いた宇宙・素粒子実験物理学研究室の岩崎昌子先生に心より感謝いたします。ご多忙の中、様々な知見よりご指導いただき、また資料作成や学会発表の打ち合わせのために貴重なお時間を頂き、大変お世話になりました。また、同研究室の中野英一先生には、ゼミを通して高エネルギー物理学の基礎から丁寧にご指導を頂きました。加えて、研究活動についても多くのご助言を賜りました。本当にありがとうございます。

機械学習の手法については、大阪大学データビリティフロンティア機構（IDS）の長原一先生、中島悠太先生、武村紀子先生、大阪大学核物理研究センター（RCNP）の中野貴志先生に大変お世話になりました。研究を始めた当初はこの分野について素人であった自分に、ミーティングを介して多くご指導を頂きました。また、阪大 RCNP 研究プロジェクト、阪大 IDS 学際共同プロジェクトの皆様にも心から感謝いたします。

加速器データの収集については高エネルギー加速器研究機構 (KEK) の佐藤政則様、佐武いつか様に大変お世話になりました。ご多忙のところ、様々な面で助けていただいたことに感謝いたします。

宇宙・素粒子実験物理学研究室の皆様には、研究を行う上で多くの協力を頂きました。コロナ禍という特異な状況下での研究生活となりましたが、同期、後輩に恵まれ、充実した研究生活を過ごすことが出来ました。本当にありがとうございました。

最後に、学部・修士を合わせ物理学の面白さを感じ取れる学生生活・研究生活を送ることが出来たのは、支えてくださった皆様のおかげです。改めて、私をこれまで支えてくださったすべての方に感謝いたします。