公立はこだて未来大学 2023 年度 システム情報科学実習
グループ報告書

Future University Hakodate 2023 Systems Information Science Practice

Group Report

プロジェクト名

脳をつくるプロジェクト

Project Name

Make Brain Project

グループ名

音楽生成

Group Name

Music Generation

プロジェクト番号/Project No.

13-A

プロジェクトリーダー/Project Leader

太田怜志　Reiji Ota

グループリーダー/Group Leader

太田怜志　Reiji Ota

グループメンバー/Group Member

山内大翔	Daito Yamauchi	太田怜志	Reiji Ota
岩崎誠也	Seiya Iwasaki	工藤大	Hiro Kudo
小林未佳	Mika Kobayashi	田中柊真	Shuma Tanaka
中村允洸	Masahiro Nakamura	山谷璃輝	Riki Yamaya

指導教員

香取勇一　栗川知己　加藤譲　佐々木博昭　冨永敦子　ヴラジミールリアボフ　佐藤直行

Advisor

Yuichi Katori　Tomoki Kurikawa　Yuzuru Kato　Hiroaki Sasaki　Atsuko Tominaga
Volodymyr Riabov　Naoyuki Sato

提出日

2024 年 1 月 17 日

Date of Submission

January 17, 2024

概要

日々進歩を遂げる人工知能の研究は、我々の生活のあらゆる側面に影響を与えている。その研究成果は、我々が日常的に利用する多くの便利なツールとなり、生活の質を向上させている。その中でも近年、生成 AI が多くの分野で注目を集めている。しかし、生成 AI の課題の 1 つとして、学習コストが膨大であることが挙げられる。生成 AI は大量のデータを必要とし、そのデータを学習するために高性能なコンピュータを必要とし、学習の過程で大量の電力を消費する。しかし、脳では同様のタスクを低コストで行っていると考えられる。そこで、本グループでは、脳の仕組みを取り入れた音楽生成 AI を開発することで、学習コストの削減を試みた。

本グループでは、音楽生成 AI の開発を 2 つの異なるアプローチで試みた。1 つ目は、テキストから音楽を生成するアプローチ、2 つ目は、ある音楽を別のジャンルの音楽に変更するアプローチである。テキストから音楽を生成するアプローチでは、既存の Stable Diffusion[1] に対し、Fully Spiking Variational Autoencoder[2] を導入し、低コスト化を試みた。ある音楽を別のジャンルの音楽に変更するアプローチでは、CycleGAN[3] に対し、レザバー計算 [4] を導入し、低コスト化を試みた。理由は、リスクの分散と開発の効率化である。2 つのアプローチで並行して開発を進めることで、それぞれの収集したデータや得られた知見を共有することが可能となる。これにより、同じ失敗を繰り返すことを防ぎ、開発の効率化ができると考えた。

本グループでは、低コストな音楽生成 AI の実現に向けて活動した。その結果、生成した音楽にはノイズを含んでいるものの、音楽生成 AI の低コスト化を実現できた。また、今回用いた音楽生成 AI のモデルは、その他の生成 AI との共通部分も多い。そのため今回の成果は、低コストな音楽生成 AI 実用化だけでなく、その他の生成 AI の低コスト化に貢献するものと考えている。今後は、音楽生成 AI の性能をさらに向上させ、実世界における活用の可能性を考えている。

キーワード　CycleGAN, レザバー計算, 拡散モデル, Fully Spiking Variational Autoencoder

（※文責: 太田怜志）

Abstract

Research in artificial intelligence, which is advancing daily, is influencing in various aspects of daily life. It produces have become many useful tools that we use on a daily basis to improve our quality of our every life. Among them, generative AI has recently attracted much attention in many fields. However, one of the challenges with generative AI is the enormous learning cost. Generative AI requires a large amount of data, a high-performance computer to learn the data, and a large amount of power in the learning process. But the brain is thought to perform similar tasks at a lower cost. Therefore, our group has attempted to reduce these learning costs in a music generation AI that incorporating brain mechanisms.

Our group had two approaches to develop a music generation AI. The first approach is to generate music from text, and the second is to change one genre of music to another. In the approach to generate music from text, we attempted to reduce the cost by introducing a Fully Spiking Variational Autoencoder[2] to the existing Stable Diffusion[1]. For the approach to change one genre of music to another, we attempted to reduce the cost by introducing Reservoir Computing[4] to CycleGAN[3]. The reason is to diversify risk and increase development efficiency. By developing the two approaches in parallel, it is possible to share the data collected and knowledge gained from each. We believed that this would prevent repeating the same mistakes and improve development efficiency.

Our group worked to achieve low-cost music generation AI. As a result, although the generated music contained noise, the group was able to realize low-cost music generation AI. In addition, the model of music generation AI used in our group shares many common features with other generative AI. Therefore, we believe that our achievement will contribute not only to the practical application of low-cost music generative AI, but also to the cost reduction of other generative AI. In the future, we plan to further improve the performance of the music generation AI and consider the possibility of using it in the real world.

Keyword CycleGAN, Reservoir Computing, Diffusion Model, Fully Spiking Variational Autoencoder

（※文責: 太田怜志）

目次

第 1 章　はじめに

1.1　背景

音楽は、人間の感情や表現を伝える芸術文化の一つであり、その創造的側面をコンピュータによって再現することは困難な課題であると長年認識されていた。しかし、機械上で音楽プロセスを実現しようとする取り組みは、他の計算機科学の研究領域と比べて比較的歴史が長く、早期に立ち上がった領域とされている。楽譜や楽典の中で見られるように、音楽が他の芸術分野よりも計算機科学に馴染みやすかったことなどがその理由に挙げられる。近代科学における取り組みとしては、リズム認識モデルや旋律認識に関連した研究、音楽研究と心理学的研究の融合などの、音楽理論と音楽的洞察に基づいて音楽の分析を進める解析的な研究などがある。そしてそれらと並行して、機械学習を用いた生成的なアプローチを計算機上で再現する研究にも取り組まれてきた。近年、機械学習の技術応用の広がりによって新たな可能性が開かれている。機械学習とは、コンピューターシステムが経験から学習し、新しいデータに対して予測や意思決定を行う能力のことを指す。機械学習のアプリケーションは広範であり、画像認識、音声認識、自然言語処理、医療診断、金融予測、ゲームプレイなど多岐にわたり、データが豊富に利用可能である場合や複雑なパターンが存在する場合、機械学習は高い性能を発揮することがある。最も早期の取り組みとしては、1956 年に行われたコンピュータによる自動作曲が挙げられる。当時の音楽生成のアプローチは規則ベースの手法に基づいており、特定の音楽理論や規則をプログラムに組み込んで音楽を生成していた。そして現在までに機械学習や深層学習のさらなる発展により、既存の音楽の構成や特徴を捉えることで機械学習による新たな音楽の生成が可能となった。具体的にはリカレントニューラルネットワーク（RNN）や長短期記憶（LSTM）などのモデルが利用され、より複雑で表現力豊かな音楽生成が可能になった。

既存の生成モデルの例として「拡散モデル（Diffusion Model）」「敵対的生成ネットワーク（GAN）」の二つのモデルを挙げる。「拡散モデル（Diffusion Model）」は機械学習や統計学の分野で使用される確率モデルの一種であり、情報や影響が時間の経過とともに拡散するプロセスをモデル化したものである。代表的なモデルとしては、テキストから画像への変換を可能とするモデルである「Stable Diffusion」がある。「敵対的生成ネットワーク（GAN）」は 2 種類のニューラルネットワークで構成された生成モデルであり、これらを互いに競い合わせることで精度を高めていく。代表的なモデルとしては、GAN のアーキテクチャを応用した手法であり、画像のドメイン変換を実現する深層学習モデルの一種である「CycleGAN」が挙げられる。教師なし学習の一形態であり、ペアとなる画像を必要とせずにドメイン変換を行うことを可能としている。

既存の生成モデルにはいくつかの欠点が存在するが、その一つに、大量の学習コストがかかるという問題点がある。具体的には、高性能なハードウェアを必要とするため必要な電力消費が激しいという点、大規模なデータセットや学習データを必要とする点、学習に時間がかかってしまう点がある。大規模な生成モデルの訓練には、多くの計算資源が必要である。これらの訓練プロセスは、特に GPU や TPU などの高性能なハードウェアを使用することが一般的だが、これには相応の電力が必要である。訓練は数週間から数ヶ月かかる場合もあり、その間、大規模な計算資源が連続的

に使用されることになる。また、学習に必要な大規模なデータセットを収集するためには、時間と労力がかかり、時には高いコストがかかることがある。特にラベル付けが必要な場合、それが専門的な作業を必要とする場合もある。

（※文責: 工藤大）

1.2 目的

本グループの目的は、既存の生成モデルにおける学習コストの削減である。前述のように、既存の生成モデルには大量の学習コストがかかるという問題点がある。この点を改善するにあたって、我々は脳の仕組みを取り入れた改善案を提示する。具体的な検証方法としては、前節で挙げた二つのモデルに対しての、「FSVAE」、「レザバー計算」の導入である。まずはじめに、「FSVAE」とは生物の神経発火の仕組みを利用してデザインされた変分オートエンコーダー（VAE）の一種である。神経細胞が発火していなければ 0 を、発火したら 1 を出力させるため、扱う数値が極端に少ない。つまり、計算量の削減が期待できる [2]。次に、「レザバー計算」とは特に脳の神経回路の動作に類似したニューラルネットワークであり、時系列データの処理や予測に適したアーキテクチャである。非常に単純かつ線形な学習アルゴリズムを利用しており、従来の再帰型ニューラルネットワーク（RNN）よりも学習が容易である [4]。
上記二つはどちらの特性も脳の動作に触発されている。我々は「テキストから音楽を生成する」「音楽のジャンル変換」という二つのアプローチから、上記の手法で従来よりも学習コストが削減された独自の生成モデルを作成し、最終的に音楽の生成を行う。この過程の中で、電力消費量を減らす、学習の収束を少量のデータで行えるようにする、短時間で学習可能にするという目的を設定した。

（※文責: 工藤大）

1.3 先行研究

Kamata ら (2021) によって行われた研究では、SNN(Spiking Neural Network) を使用して画像生成を可能にするために、変分オートエンコーダ（VAE）を構築している [2]。SNN とは、生物の神経細胞の発火をモデル化したニューラルネットワークである。深層学習などで利用される従来のニューラルネットワークとは異なり、入力信号を時系列のバイナリ信号に変換して処理を行う。バイナリ信号を活用して演算を実施するため、SNN は加算演算のみで入出力関係を記述することができる。しかし、時系列のバイナリ信号を扱う必要があるため、従来の深層学習モデルのような多層化が困難であるという課題もまた存在する。Kamata らは、既存の AE に対して全てのモジュールが SNN で構築された「FSVAE(Fully Spiking Variational Autoencoder)」を構築し、その品質が既存の人工ニューラルネットワークと比較して同等、またはそれ以上であることを示した。また、SNN は加算演算のみで入出力関係を記述するため、FSVAE は速度の面で既存モデルを大幅に凌駕できる可能性も示唆している。

図 1.1 は VAE と既存の AE との計算量を比較した表である。AdamWoptimizer (LoshchilovandHutter2019) を使用し、学習率は 0.001、バッチサイズは 250、150 エポックのトレーニングを行った際の結果である。測定のためには 5,000 の画像をデータセットとして使用し、比較対象

Model	Computational complexity	
	Addition	Multiplication
ANN	$\mathbf{7.4 \times 10^9}$	7.4×10^9
FSVAE (Ours)	5.0×10^{10}	$\mathbf{5.6 \times 10^8}$

図 1.1　既存の AE と計算量を比較した表

として、同じアーキテクチャで構築された ANN を使用した通常の VAE を用意し、同じ設定でトレーニングを行っている。この結果から、全てのデータセットにおいて FSVAE が ANN を上回っていることが示されている。

レザバー計算について、Georg ら (2009) によって行われた、非線形の音声処理にレザバー計算を用いた先行研究がある [4]。「非線形音声予測」は、与えられた音声の過去のデータをもとに未来の音声サンプルを予測するタスクを指している。リザバー計算は、再帰型の結合を持つリカレントニューラルネットワークの一種であり、前述のように時系列情報の機械学習に優れている。Georg らは、レザバー計算は多くの非線形音声処理の問題に適していると考え、リザバー計算を使用した音声予測アプリケーションを用いて、その動作について、一般的に使用される代替モデルと比較検証を行った。本検証においては、リザバーコンピューティングの予測性能は、パターンマッチングアルゴリズム（PatMat）および線形自己回帰（AR）モデルと比較されている。

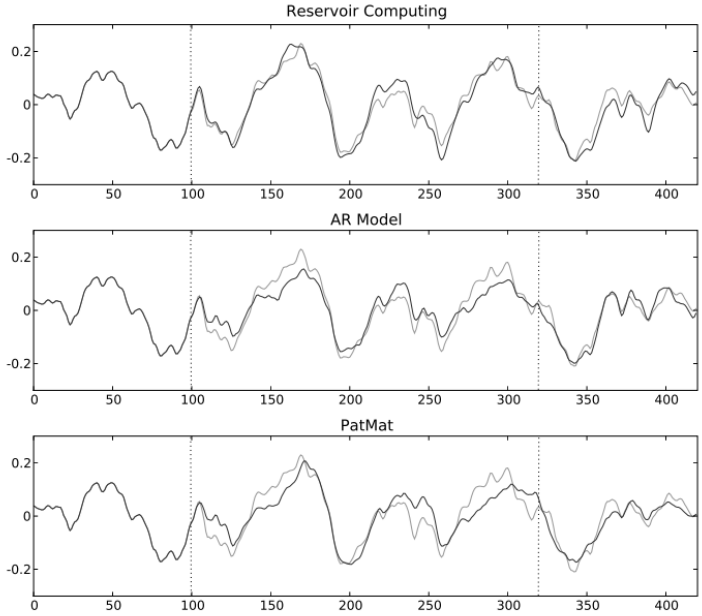

図 1.2　音声予測の性能比較

図 1.2 は、音声予測タスクの例として、オリジナルの信号（薄い灰色で表示）と予測された信号（濃い灰色で表示）のドロップアウトの様子を示している。ドロップアウトの開始と終了は点線で示されている。レザバー計算は、ドロップアウトの予測において良好な性能を発揮したといえる。

パターンマッチングアルゴリズムと線形自己回帰モデルも評価されたが、レザバー計算が総じて僅かに優れていることが示された。この結果から、音楽情報の分類タスクにレザバー計算を応用できる可能性があるといえる。

（※文責: 工藤大）

第 2 章　プロジェクト学習の概要

2.1　問題の設定

本グループでは、脳の仕組みを取り入れた機械学習モデルを用いて、1.2 節で述べた問題点の改善を目指す。そこで、スパイキングニューラルネットワークを実装した変分オートエンコーダー、および、レザバー計算を取り入れた敵対的生成ネットワークを用いて工学的応用性を検証する。また、生成タスクとしては、音楽生成を目標として設定した。

（※文責: 山内大翔）

2.2　課題の設定

本グループでは、2.1 節で述べた通り、脳の仕組みを取り入れた機械学習モデルを作成する。敵対的生成ネットワークでは、生成機のニューロンモデルにレザバー計算の一種であるエコーステートネットワークを取り入れた。また、変分オートエンコーダーでは、変分オートエンコーダーのニューラルモデルにスパイキングニューラルネットワークを取り入れた。また、敵対的生成ネットワークでは通常のニューラルネットワークを用いたモデルとの学習時間を比較し、変分オートエンコーダーではスペクトログラムを生成する際の学習の収束の速さを比較した。

（※文責: 山内大翔）

2.3　到達目標

2.2 節の課題を達成するために本グループでは、目標を 2 段階に分けて活動を行った。まず、1 つ目の目標は、音楽のジャンル変換である。具体的には敵対的生成ネットワークを 2 つ組み合わせる CycleGAN を利用して、片方では音楽生成を行い、もう片方では音楽のジャンル変換を行うことを目指した。2 つ目の目標は、文章からの音楽生成である。変分オートエンコーダーと拡散モデルを組み合わせることでの音楽生成を目指した。

（※文責: 山内大翔）

第 3 章　活動内容

本章では、音楽のジャンル変換をする班の活動内容を簡潔にまとめたものである。より詳細な活動内容に関しては、第 8 章「活動内容の詳細」で確認されたし。

3.1　text2music 班

3.1.1　夏季休暇までの活動

まず、一番初めのアイデア出しの時点でグループで取り組むテーマとして音楽と人工知能を合わせたテーマを取り上げることでまとまった。次に、学んでみたい技術としてレザバー計算や生成 AI、SNN などが挙げられた。また、人工知能を使用して音楽を生成する方法にはどのようなものがあるか、知識を共有した。例えば深層学習技術を使用して音楽を生成する方法としては、音楽をリズム、旋律などの要素に対する機構を用意し、それぞれ学習させる RNN(Recurrent Neural Network) が考えられた。最終的には音楽のジャンルを変換する CycleGAN と、近年生成 AI モデルとして盛り上がりを見せている拡散モデルをテーマとして取り上げることになった。

ここで、GroupA は 8 人と少し人数が多かったので、手の空いたメンバーが時間を持て余すことを防ぐために、1 つのグループの中でテーマを分けて取り組むことになった。1 つ目のタスクは拡散モデルを使用した音楽生成タスクに取り組む text2music 班である。具体的には、脳の仕組みを取り入れることによって工夫したモデルを作成したいという思いから、Huggingface が公開している OSS（オープンソースソフトウェア）で、画像生成 AI として有名な Stable Diffusion のヘッドにあたる変分オートエンコーダ部分を、SNN を利用した FSVAE(Fully Spiking Neural Network) に置き換えることで学習コストの低減を図るというテーマを取り上げた。この時点でレザバー計算、拡散モデル、SNN など各自学んでみたい技術がばらばらだったため、夏季休暇中はそれぞれ学びたい技術を学ぶ時間に充てることになり、具体的な開発は夏休み以降に行うことになった。

（※文責: 小林未佳）

3.1.2　夏季休暇以降の活動

データセットについて

夏季休暇が明けてから本格的に開発を行った。まず、音楽生成 AI の班では Stable Diffusion の学習に使用するデータセットを作る必要があった。そこで、wav ファイル（音楽）とその音楽について説明したテキストデータを Google の MusicCaps データセットから拝借した。MusicCaps では、youtube の url についた id を利用して、5521 個のデータを入手することができる。一部データの内容のバランスをとったサブセットデータが 2600 個ほど用意されているものの、データの数が足りないと判断した。しかし全て学習に使用すると音楽の種類やジャンルが偏るので、音楽の種類を分類して classic と判定されたデータを使用した。データセット内で指定された 10 秒間について、動画の内容について専門家が説明したテキストが用意されている。これらの Youtube 動画の

音声についを 22050Hz でサンプリングし、長さ 20 秒程度の wav ファイルとして保存した後、これを長さ 4 秒程度に区切ってグレースケールのスペクトログラムを生成し、画像データとして学習に使用する。MusicCaps データセットはライセンス cc-by-sa 4.0 に従っており、商業目的を含めコピーや再配布、データに対する加工を加えることができる。ただし、作成したデータセットは元のデータセットである MusicCaps と同じライセンスに従う。

Huggingface Hub

またこの際、Huggingface Hub のデータセットリポジトリを利用することにした。Huggingface Hub を使用するメリットはいくつかある。

一つ目は、データセットとモデルの共有とアクセスが容易になることだ。Huggingface Hub は、モデルやデータセットを共有できるプラットフォームであり、organization として登録された他の開発者とデータセットを共有し、いつでも load_dataset メゾットを利用してデータセットをダウンロードすることが可能になる。

二つ目はバージョン管理と追跡が容易になることだ。Huggingface Hub のデータセットリポジトリにデータをアップロードする際は Git Large File Storage を利用する。このため普段ファイル管理に使用している git と同じようにデータを扱うことができるのでローカル PC でもリモートでも、データ操作の間違いが減る。またアップロードすると、データセットに対する変更履歴や、修正したバージョンが追跡される。これにより、データセットの変更や改善が透明かつ追跡可能になり、プロジェクトの進捗を管理しやすくなる。

3 つ目は、Huggingface Hub を使用することで、データセットへの操作やアクセスが簡単になることだ。例えば、API を介してアクセスするとデータセットのメタ情報（データセット名、訓練データの数、テストデータの数、データセットの内容、カラムの構成など）取得や更新がスムーズに行える。また新しくデータセットを用意する際に pandas や polar で作成された Datasets オブジェクトからデータセットを簡単にアップロードすることもできる。

またこのほかには、データセットを PostgreSQL 等大規模データに適する DBMS を用いて管理する方法も考えたが、そこまでの技量と習得に要する時間的余裕はその時点ではないと判断した事情もある。

Huggingface Hub の公式ドキュメントを読み、Huggingface Hub 上でデータセットを作成する方法を 3 つ見つけた。まず最初の方法は、画像などのデータを zip ファイルなどで加工し、それに加えて jsonl や csv ファイルから成るメタデータファイルを組み合わせて直接アップロードする方法である。次に、Python で書かれたローディングスクリプトを用いてデータの解析をしながら、取得したデータをデータセットとして加工させる方法がある。そして、最後の方法は、事前にノートブック上で実行済みの ‘datasets‘モジュールの ‘Datadict‘型から ‘Dataset‘ オブジェクトを生成し、Huggingface Hub の API を用いて直接データセットリポジトリにプッシュする方法だ。これらの方法の中で、最も手軽で効果的だったのは、最後の方法であった。

開発環境構築とモデル開発

FSVAE を使用する際には、論文とともに公開されている Github のスクリプトを参照した。VAE や FSVAE の構造については次項を読んでほしい。

開発環境構築　PC の OS として Ubuntu をインストールした。また、GPU を使った学習を行うため Nvidia のドライバのインストールを行った。Rye と呼ばれる環境管理ツールの導入を行い、Windows 上での開発よりもより素早く開発を行えるように環境を整えた。

工夫点　まずモデル開発で工夫した点の一つ目は、学習コストの評価のため wandb を使用したコードを追加したり、学習を進めるための最適化手法を変更したことだった。最初の試みでは、SNN を使用した AE(オートエンコーダ) ではうまくスペクトログラムを再構成することができたものの、VAE を使うとうまく学習が進まないことが分かった。そこでデータセットのスペクトログラムを見直して正規化した画像データを用意した。正規化の方法には 3 種類ほどあり、一つ目はヒスログラム平滑化、二つ目は適応的ヒストグラム平坦化、三つめは輝度値の平均と標準偏差を指定して正規化する方法が見つかった。これらすべてを試した結果、画素値がより明確に 2 値化する 1 番目の方法を使用してデータを作成した。またこの作業は、それぞれの変換方法を適用した画像から元の音楽が再生できることを確認しながら行った。また、そもそもデータセットに歌声や雑音などとても音楽とは呼べないデータが含まれていたことから、音楽分類モデルを利用して、このモデルに classic と判断された音楽のみを抽出したデータセットを用意して試すなどした。最終的には音楽が上手く生成されていないのは、リズムなどの時系列情報が上手く学習されていないためであると判断し、music2music 班で開発していた waveGAN を参考に、アプローチの方法を切り替えて waveGAN（音素の時系列を入力データとする方法）などを試した。また、拡散モデルのコードについては Huggingface が github 上で、Huggingface Diffusrs リポジトリで公開しているを参照した。このコードを動かすにあたっては、Lora という技術を使用した。クロスアテンション部分に低次元の層を層を追加することで、U-NET 全体のパラメータを学習せずモデルをファインチューニングできる。これを使うことで、半精度技術を使っても最低 24GB は学習に必要なメモリ量が 12GB 程度になる。しかし最終的には Huggingface のリポジトリで公開されているコードを工夫して音楽を生成するには至らなかった。

その他の作業

データセットづくりやモデル開発以外の作業としては、スペクトログラムから音楽を再生する関数を定義するなどの作業があった。また、これらの関数についてはグループで共有しやすいように python モジュール化し PyPI（オープンソースソフトウェアを公開するリポジトリ）上で公開した。

（※文責: 小林未佳）

3.1.3　活用したニューラルネットワーク

AutoEncoder

オートエンコーダ（Autoencoder, AE）は、ニューラルネットワークの一種でエンコーダとデコーダで構成される。エンコーダは入力データを低次元の表現に変換し、データの重要な特徴や構造を抽出する。デコーダは低次元の埋め込み表現を元の画像と同じ次元まで戻し、データを再構成する。したがって、訓練データとして与えられた入力データと再構築されたデータとの誤差を最小化するように学習を行う。この誤差を再構成誤差と呼ぶ。オートエンコーダの応用としては、次元

削減による特徴抽出などが挙げられる。

Variational AutoEncoder

変分オートエンコーダ（Variational Autoencoder, VAE）では、通常のオートエンコーダのアイデアをベースに、潜在空間内の点がデータの潜在分布からサンプリングされる過程を考慮する。通常のオートエンコーダでは、潜在空間の点は単なる埋め込み表現に過ぎない。これに対して、VAE ではデータの潜在分布がある確率に従うことを仮定する。データの潜在的な分布としてガウス分布（正規分布）を仮定し、エンコーダから出力された平均と分散を用いてサンプリングを行い（再パラメータ化、Reparametrization Trick)、このサンプルをデコーダへの入力とする。VAE の損失関数では、通常の再構成誤差項と、潜在空間の分布が正規分布に近づくような正則化項が加えられた、KL ダイバージェンス項からなる損失関数を使う。変分オートエンコーダは、特にデータ生成や変分推論のタスクで優れた性能を発揮する。

FSVAE

Fully Spiking Variational Autoencoder (FSVAE) は、スパイクニューラルネットワーク (SNN) を活用した変分オートエンコーダ (VAE) の一種である。SNN で扱われるデータはバイナリ値であり、超高速かつ超低消費電力のニューロモーフィックデバイスで実行することを想定している。FSVAE の公式実装は GitHub 上で公開されている。この研究では、考察として MNIST や CelebA などのデータセットを用いて通常のニューラルネットワークを用いたモデルとの計算量比較を行い、計算量が通常の VAE よりも少なくなると主張している。

アーキテクチャ　FSVAE モデルは全ての層を SNN で構築している。具体的には、torch.nn モジュールの nn.Conv3d 関数とバッチ正規化、LIF モデルを参考に作成された活性化関数からなるモジュールを何層か組み合わせたエンコーダから、潜在的な低次元の表現を得る。デコーダもエンコーダと同様に上記のモジュールを何層か組み合わせた構造を持つ。この際、エンコーダから出力されるデータはバイナリの時系列データであるため、通常の VAE のように、平均と分散を出力して標準正規分布に従うよう学習する再パラメータ化を用いることはできない。したがって FSVAE の潜在空間は VRNN(Variational Recurrent Neural Network) の仕組みを利用する自己回帰 SNN モデルで構成されており、ベルヌーイ過程に従う時系列を出力するよう学習する。

損失関数　FSVAE を学習する際には、損失関数として元画像と復元画像との差を計算する再構成誤差と、事前分布と事後分布の差を計算する MMD(Maximum Mean Discrepancy) を用いる。

一つ目の再構成誤差には、MSE(Mean Squared Error, 平均二乗誤差) 関数を用いる。元の画像データは画像の高さ $H\times$ 幅 $W\times$ チャネル数 C で構成されているが、デコーダの最終層から出力されるのはこれに時間の次元 T を追加した時系列データであることから、時系列情報から膜電位を計算して元の画像データと次元を合わせたうえで、再構成誤差を計算する。

二つ目の MMD 関数について、通常の VAE は正規分布を事前分布として用い、学習には KL ダイバージェンスを用いて確率分布同士の差を、再構成誤差と合わせて損失関数として用いる。しかし KL ダイバージェンスは発散しやすく、多峰性のシナプス膜電位の変化を記述するスパイク列のモデル化には適していない。したがって FSVAE の元の論文では、自己回帰 ANN を用いてスパイク列がポアソン過程に従うモデル化した研究を参考に、損失関数として MMD 関数を導入している。

拡散モデル

潜在拡散モデルは機械学習分野における潜在変数モデルであり、拡散モデルとも呼ばれる。このモデルは、まず VAE モデルのエンコーダでデータの潜在表現ベクトルを抽出する。次にデータの各点が潜在空間上でノイズによって拡散され（拡散過程)、これを逆にたどる逆拡散過程で、ニューラルネットワークを用いた U-NET と、U-NET の各層に対して LLM からの入力を組み込むクロスアテンションを用いてデータの潜在的な構造を学習する。Lora を用いて学習する際には、クロスアテンション層に低次元の層を追加し、ファインチューニングを行う。Stable Diffusion は英国企業の stability.ai が開発した画像生成 AI であり、潜在拡散モデル（Latent Diffusion Models）を利用している。stability.at 社はこのプロジェクトのテーマである音楽生成を行うモデルも開発しており、web サイト上で無料で利用できる。また、他にプロンプトから画像を生成する技術として有名なモデルとして Google 社が開発した DALLE-2 などが有名である。

（※文責: 小林未佳）

3.2 music2music 班

3.2.1 夏休みまでの活動

モデルを作成するためのニューラルネットワークの決定

我々の班では、音楽のジャンル変換をすることを目標に定めた。そして次に、どうやって音楽のジャンル変換を実現するかについて決めることにした。それらを決定するために、私たちはいくつかのニューラルネットワークを候補に挙げた。その候補は、DNN、CNN、GAN、レザバー計算だ。どのニューラルネットワークを使うか検討するため、私たち 4 人で 1 つずつこれらのニューラルネットワークを調査することにした。調査した内容を精査し、スライドにまとめるなどして班のメンバーに共有した。また、GAN を調べていく中で、CycleGAN というニューラルネットワークの存在を発見した。CycleGAN とは、GAN を 2 つ組み合わせた構造を持っているニューラルネットワークである。この CycleGAN についても、音楽ジャンル変換を実現するためのニューラルネットワークの候補に加えた。議論の結果、CycleGAN にレザバー計算を導入して、音楽のジャンル変換を実現することとなった。

この決定に至った主な理由は 2 つある。1 つ目は、CycleGAN で音楽のジャンル変換をした先行研究がある点だ。先行研究があると、その研究を参考にしてある程度の所までモデルを構築でき、円滑にプロジェクトを進めることができる。2 つ目は、新規性がある点だ。音楽のジャンル変換を実現する方法を決定した段階では、まだ GAN にレザバー計算を導入した例はなく、我々の班が先陣を切って取り組むことができるという事実がとても魅力的なものに映った。3 つ目は、具体的な目標が設定しやすい点だ。レザバー計算の利点として挙げられるのは、消費する電力や学習に掛かる時間を抑えながら、ある程度の計算性能を発揮できるということである。消費した電力や学習に掛かる時間というものは、測定が容易であるので、既存のモデルと比較する評価指標として適している。よって、レザバー計算を導入したことにより、消費した電力や学習に掛かる時間といった評価指標を減少させるという具体的な目標を設定できるようになった。

（※文責: 中村允洸）

中間発表スライドの作成

本グループの方針としては、発表スライドの制作に一か月ほど時間をかけ、中間発表で求められる以上のクォリティーのものを作っておくということになった。その理由は、中間発表の時点で完成度の高いスライドを作っておくことで、追い込まれることが予想される最終発表間近の時、スライド作成に時間を割く必要がないようにするためだ。まず私たちは、スライドのデザインを決めることにした。デザインの候補は、メンバーが自身で作成したデザインや、あるいはインターネット上でダウンロードできる既存のデザインで気に入ったものだ。スライドのデザインを決めた後、私たちはスライドを共同で編集できるサービスを探した。その結果、canva というサービスがいいのではないかという結論に至った。その理由は、canva が無料で使用可能であり、グループメンバー全員でスライドを作成することが可能だからだ。また、canva は elements という他のユーザーが作成したデザインを使用することが出来る。そのデザインを使用することで、私たちがデザインを作成する時間を削減し、より優れたスライドを作成することが期待できる。

スライドのデザイン、作成するサービスが決まったので、私たちは協力してスライド作成に取り組んだ。その活動の中で役に立ったのが、前述した GAN, レザバー計算について班のメンバーがまとめたスライドである。中間発表では、使用することになるニューラルネットワークについても当然紹介しなければならない。そのため、あらかじめ班のメンバーが作成したスライドを流用することが出来た。その結果、作業時間を大幅に短縮することが出来た。私たちは、他のユーザーが作成したデザインを使いながら、スライドを完成させた。しかし、完成したスライドで発表練習をした際、「スライドに余計なデザインが多く見にくい」という指摘を担当教員の方にいただいてしまった。その指摘を受け、スライドからデザインを減らし、シンプルで見やすいものになるように修正を行った。その結果、中間発表では、スライドの見やすさという点で多くの方々から高い評価をいただけた。

（※文責: 中村允洸）

3.2.2 夏休み後の方針

夏休みの方針

私たちは前期の期間中、中間発表のスライド作成にかなり時間をかけた。ゆえに、CycleGAN にレザバー計算を導入したニューラルネットワークを用いて、モデルを開発する作業までは進まなかった。また、夏休み期間中はインターンなど学生にとって重要な期間であるため、本格的なモデル開発は後期から始めることに決めた。ただ、夏休み期間に何も行わないとなると、成果発表会までに成果物の作成が間に合わない。よって、夏休み期間中は、github を使いこなせるようにするという方針を立てた。github は、複数人でモデル開発をする際に、ソースコードを管理するサービスとしてなくてはならないものだ。また、最終報告書も github で管理して作成をすることに決めた。これを受けて、私たちは各自で web サイトや書籍を調査し、github について学んだ。また、自分が読んだ web サイトで勉強になったものを共有することで、習得がより円滑になることを図った。

（※文責: 中村允洸）

後期の方針

後期は、本格的にモデル開発をすることとなった。それに伴って、やらなくてはならない課題が出来た。

1 つ目は、データセットの作成である。CycleGAN を用いてニューラルネットワークの学習をするためには、データセットが必要不可欠である。私たちは音楽のジャンル変換をするので、音楽データをインターネット上で収集した。その際、著作権等の問題に配慮し、学習に利用してもよい音楽データのみを収集することに努めた。集めた音楽のジャンルは、ロックとジャズだ。その理由は、ロックとジャズという音楽ジャンルは誰もが聞いたことがあるので分かりやすいからだ。また、ロックとジャズの音楽ジャンルはかなり異なっているため、ジャンル変換したときの差異が判別しやすい、というのも理由の一つだ。

2 つ目は、スペクトログラムと音楽を相互に変換することだ。スペクトログラムとは、時間と周波数の二次元表示である音声や音楽などの信号の周波数成分を可視化するためのグラフである。CycleGAN の学習には、このスペクトログラムの画像を用いることにした。よって、音楽データから得られた音の波形を、スペクトログラムに変換する必要がある。また、CycleGAN が生成したスペクトログラムを音の波形に変換する必要がある。

モデル開発と同時に、成果発表に向けた準備も進めなくてはならない。成果発表会は、私たちが作成した成果物を、様々な方々に評価していただくための重大なイベントである。その中で、質問対応というのは特に重要な役割だ。なぜなら質問というものは、ただ単に準備された発表を聞くという受動的なものではなく、知りたいことを能動的に尋ねるものだからだ。もし、質問者の知りたいことに過不足なく適切な応答を返すことが出来れば、高い評価を得ることが出来る。ゆえに、成果発表会において、最低一人は専門性の高い知識に詳しい人材を配置することにした。その役割を任された人間は、関連する書籍や論文を調査し、それを notion にまとめて、知識の定着を図った。

（※文責: 中村允洸）

3.2.3　モデル開発

モデル開発は、いきなり CycleGAN を用いて音楽ジャンル変換を試みるようなことはせず、段階的に行うことにした。まず、GAN を構築してスペクトログラムの画像生成を試みた。しかし、GAN が生成するスペクトログラムの画像には、市松模様のようなものが周期的に表れてしまった。これでは、スペクトログラムを正しく学習できず、モデルの精度に悪影響が出てしまう。私たちはこの問題を解決するため、インターネット上で論文を調査した。調査の結果、Donahue らの論文で紹介されていた、WaveGAN というものを使えば、この問題を解決できると考えた。その論文では、WaveGAN は、Discriminator にフェーズシャッフルという手法を導入することで、チェッカーボードアーティファクト（市松模様）の発生を防ぐことができると示唆されていた。また、WaveGAN にはもう一つ利点がある。それは、WaveGAN は、音の波形のデータをそのまま学習に使えるということだ。この利点により、音の波形とスペクトログラムを相互に変換する必要がなくなり、作業量を減らすことが出来た。以上の利点から、GAN ではなく、WaveGAN で音楽生成を試みることにした。試行錯誤の結果、音楽生成におおむね成功した。生成した音楽のジャンルは、ハウスミュージックである。その理由は 2 つある。1 つ目は、4 つうちのドラムなので、リズ

ムを学習したかどうかわかりやすいからだ。2 つ目は、もともと電子音なので、コンピュータに取り込む際、失われる情報がなく、音楽生成と相性がいいからだ。WaveGAN を用いての音楽生成に成功したので、次の段階として、Generator にレザバー計算を導入することにした。その結果、1 エポック当たりの学習時間が約 57 秒から約 37 秒に短縮され、「学習コストの削減」という大目標を達成することが出来た。しかし、活動時間が足りず、CycleGAN を用いて音楽のジャンル変換を行う段階までは到達することができなかった。

（※文責: 中村允洸）

3.2.4　活用したニューラルネットワークの解説

GAN

GAN（Generative Adversarial Network）は、2014 年に Ian J. Goodfellow らによって提案された教師なし学習の生成モデルの一種であり、敵対的生成ニューラルネットワークとも呼ばれる [5]。Ganerator（生成器）と Discriminator（識別器）の 2 つのニューラルネットワークから構成され、この 2 つが互いに競い合うことで学習する。Generator はランダムなノイズを入力として受け取り、データの生成を行う。Discriminator は「Generator が生成した偽物のデータ」と「訓練データとして用意された本物のデータ」を入力として受け取り、データの真偽を判別する。Generator は生成したデータを Discriminator に本物として判別されるように学習される。Discriminator は Generator が生成したデータを偽物であると判別できるように学習する。この学習を交互に繰り替えることにより、互いに精度を高め合い、Generator が本物により近い偽物を生成できるようにする。

通常の GAN は、Generator と Discriminator に CNN を導入し、学習を行う。これを DCGAN（Deep Convolutional GAN）と呼ぶ。DCGAN は、主に画像生成を行うために用いられる。音楽生成をする際には、音楽をスペクトログラムの画像に変換してから学習させる必要がある。

（※文責: 中村允洸）

CycleGAN

CycleGAN は、2017 年に Jun-Yan Zhu らによって提案された GAN の一種である [3]。GAN のニューラルネットワークを 2 つ組み合わせたような構造を持っている。この CycleGAN には、GAN にはない利点がある。通常の GAN の場合、Image-to-Image 変換では、対となる画像ペアのトレーニングセットが必要だ。しかし、CycleGAN の場合、異なるドメイン間の画像変換を行うことができる。2 つのドメインをドメイン X、ドメイン Y としたときドメイン X をドメイン Y に変換する Generator（以後、Generator（X → Y）とする。）に X を入力し、Y’ を生成する。この Y’ を、Y を訓練データとして学習した Discriminator（以後、Discriminator Y とする。）に入力し真偽を判別させる。次に Generator（Y → X）に Y’ を入力し、X” を生成する。このときに X を訓練データとして学習した Discriminator X に X” を入力し、X と一致しているかを確認しながら学習が行われる。これらの学習が逆方向からも行われることで、循環による一貫性を持たせることができ、2 つのドメイン間の相互変換を可能にする。

（※文責: 中村允洸）

レザバー計算

レザバー計算は、リカレントニューラルネットワークの一種で、主に時系列データの機械学習に利用される。典型的なレザバー計算モデルは入力層、レザバー層、出力層から構成される。レザバー層と出力層の間の結合重みだけを更新し、入力層とレザバー層の間の結合重みとレザバー層内の重みは固定するという特徴がある。これによって、すべての重みを更新する他のニューラルネットワークと比較して、高速な学習が可能となる。また、レザバー計算には音声予測に用いられているという先行研究がある。そこでは、音楽情報の分類タスクに, レザバー計算を応用できる可能性があるということが示唆されている [4]。

（※文責: 中村允洸）

WaveGAN

WaveGAN は、GAN の一種であり、教師無し学習により音の波形を合成することが出来るニューラルネットワークである。また、利点として、DCGAN より効率的に学習を進めることが出来るというものがある [6]。DCGAN では、画像生成をした際、画像内に特徴的な「チェッカーボード」アーティファクトを生成してしまう [6]。画像では周期的なパターンはあまり一般的ではないため、Discriminator はそれらを含む画像を拒否することを学習できる [6]。音の波形の場合、類似のアーティファクトは実際のデータで一般的な周波数と重なる可能性のあるピッチ付きノイズとして認識され、Discriminator の学習がより困難になる [6]。ただし、アーティファクトの周波数は常に特定の位相で発生するため、Discriminator は生成された偽物のデータを拒否する簡単なポリシーを学習できる [6]。これにより、他の判別要素があまり考慮されず、全体的な最適化の問題が阻害される可能性がある [6]。

Discriminator がそのようなポリシーを学習するのを防ぐために、フェーズシャッフルという操作を行う [6]。フェーズシャッフルは、各レイヤーのアクティベーションのフェーズをランダムにシャッフルする [6]。これを Discriminator に導入することで、Discriminator に偽物のデータを拒否する簡単なポリシーを学習させないようにすることができる [6]。

（※文責: 中村允洸）

第 4 章　成果とその評価

4.1　text2music 班

text2music 班では、脳の仕組みを取り入れることで、低コストで音楽生成を行うモデルの作成を目指し、活動を進めてきた。ここで、text2music 班による成果物を作成するうえで用いた手法を説明する。

（※文責: 田中柊真）

4.1.1　Stable Diffusion

前節で述べたように、text2music 班では、テキストを打ち込み、音楽を生成するタスクを行わせる。text2music 班では、既存の Stable Diffusion というモデルを改善する。Stable Diffusion とは、Diffusion model を利用した、テキストから画像を生成するモデルであり、オープンソースで公開されている [1]。Stable Diffusion は、Autoencoder、U-NET と Text Encoder によって構成されている [7]。まず、画像を Autoencoder に入力させ、拡散プロセスでノイズを連続的に付与させる。続いて、プロンプトにテキストを入力し、入力されたテキストを Text Encoder に入力させ、ノイズが付与された画像と一緒に U-NET を介し、逆拡散プロセスでノイズを除去させる。最後に除去された画像を Autoencoder に入力させることで、画像を生成する。

text2music 班では、既存の Stable Diffusion に対して、Autoencoder に Fully Spiking Variational Autoencoder を導入する。さらに、U-NET に Token-shift を組み込む。

（※文責: 田中柊真）

4.1.2　Autoencoder

Autoencoder とは、Encoder と Decoder からなる [8]。Encoder では、画像から特徴量を抽出させることで、画像の特徴量の次元圧縮をさせる。次元圧縮により、画像のより基本的な意味を捕らえることが可能となる [7]。Decoder では、Encoder から出力された潜在変数に対して、次元をもとに戻し、画像に戻すものである。

Stable Diffusion においては、画像を Encoder に入力させることで、画像のより基本的な意味を捕らえられることが期待できる。

text2music 班では、さらなる計算コストの削減のため、Autoencoder を Fully Spiking Variational Autoencoder に置き換える。

（※文責: 田中柊真）

4.1.3 Fully Spiking Variational Autoencoder

Fully Spiking Variational Autoencoder とは、Autoencoder にスパイキングニューラルネットワークを適用したものである [2]。なお、以下では、Fully Spiking Variational Autoencoder を FSVAE と記述する。

スパイキングニューラルネットワークとは、生物の神経細胞の発火をモデル化したニューラルネットワークである。神経細胞が発火していなければ 0 を、発火したら 1 を出力させる。これにより、扱う数値は 0 または 1 の 2 値となる。

FSVAE は、既存の人工ニューラルネットワークのモデルと比較して、より高い精度で画像の認識を行えたことが明らかになった [2]。さらに、FSVAE はスパイキングニューラルネットワークを適用していることから、扱う数値は 0 または 1 のどちらかとなる。そのため、計算量の削減が期待できる。

text2music 班では、計算量の削減のため、Autoencoder の Encoder および Decoder を、それぞれ FSVAE を用いた Encoder および Decoder に置き換える。

（※文責: 田中柊真）

4.1.4 拡散プロセス

拡散プロセスでは、入力された画像に対して、ガウス分布に従うノイズを連続的に付与していく。拡散プロセスにおいては、Stable Diffusion における構成と同様に構成する。

（※文責: 田中柊真）

4.1.5 U-NET

U-NET とは、画像を縮小、拡大するような畳み込み層が連結されている、畳み込みニューラルネットワークである [9]。U-NET では、まず、入力された画像を何度も畳み込み、その画像の特徴を抽出する。続いて、畳み込みの逆処理を行い、抽出された特徴から、画像に付与されたノイズを学習させる。

text2music 班では、U-NET における自己注意機構を Token-shift に置き換えたうえで、組み込む。

（※文責: 田中柊真）

4.1.6 自己注意機構

自己注意機構では、U-NET において畳み込まれた特徴量を、各層へ反映する。

text2music 班では、計算量を削減させるために、自己注意機構に Token-shift を組み込む。

（※文責: 田中柊真）

4.1.7 Token-shift

Token-shift は、Token と呼ばれる、テキストを意味単位ごとに分割したものを、わずかに移動させる機構である [10]。この機構を利用することで、言語処理において、計算量の削減に成功している [10]。また、Token-shift は、入力された特徴量のベクトルを 1 つ分移動させることができることから、ベクトルを 1 つ分移動させた特徴量に対し、重みをかけて混ぜ合わせることで、ネットワーク不要な自己注意機構の代替となると期待している。

text2music 班では、以上のことを利用して、計算量を削減させるために、U-NET 内に Token-shift を組み込む。

（※文責: 田中柊真）

4.1.8 逆拡散プロセス

逆拡散プロセスは、拡散プロセスの逆のことを行う。つまり、ノイズが付与された画像に対して、Token-shift を組み込んだ U-NET で学習したノイズを引いていくことで、画像を生成させる。

（※文責: 田中柊真）

4.1.9 提案手法

まず、音楽をスペクトログラムへ変換させる。スペクトログラムとは、音声に対して窓付きフーリエ変換を用いることで、横軸を時間、縦軸を周波数として分解し、音声の強弱を色の濃淡で表すようにプロットしたものである。これにより、音声を画像として扱うことが可能となる。変換されたスペクトログラムをコピーして重ね、FSVAE による Encoder に入力させ、特徴量を抽出させる。抽出させた特徴量について、拡散プロセスにて、ガウス分布に従うノイズを連続的に付与させる。続いて、プロンプトにテキストを入力し、入力されたテキストを Text Encoder に入力させ、ノイズが付与されたスペクトログラムの特徴量と一緒に、Token-shift を組み込んだ U-NET による逆拡散プロセスにて、ノイズが付与されたスペクトログラムの特徴量におけるノイズを連続的に除去させる。続いてノイズが除去された特徴量を、FSVAE による Decoder に入力させることで、連なったスペクトログラムを生成させる。最後に、連なったスペクトログラムを足し合わせ、スペクトログラムを逆フーリエ変換をさせることで、生成されたスペクトログラムを音楽に変換させる。

（※文責: 田中柊真）

4.1.10 成果と評価

以上の手法を用いて、text2music 班では、入力テキストに即した音楽生成を果たすことはできなかったが、FSVAE による学習によって、スペクトログラムを生成する機構を作り上げた。また、生成されたスペクトログラムを逆フーリエ変換することで、音楽を得ることができた。

ここで、訓練段階にて入力として与えたスペクトログラムを図 4.1 に、訓練によって得られたスペクトログラムを図 4.2 に示す。

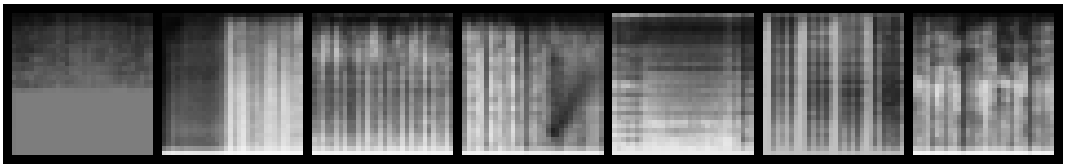

図 4.1　訓練段階における入力スペクトログラム

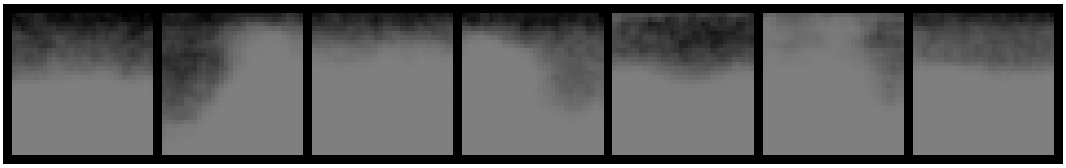

図 4.2　訓練によって得られたスペクトログラム

図 4.2 より、生成されたスペクトログラムは、入力として与えたスペクトログラムである図 4.1 と比較しても明らかなように、目視でも全体的にぼやけているように思えた。そのため、変換した音楽もノイズが目立つ結果となった。

さらに、テスト段階でも、以下の結果が得られた。

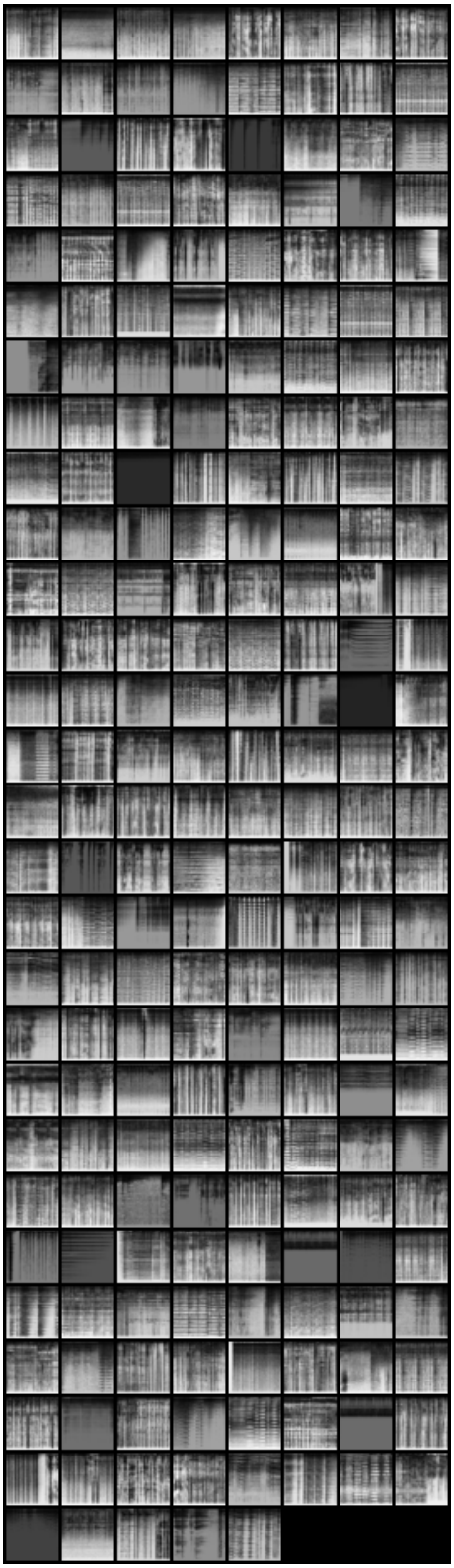

図 4.3　テスト段階における入力スペクトログラム

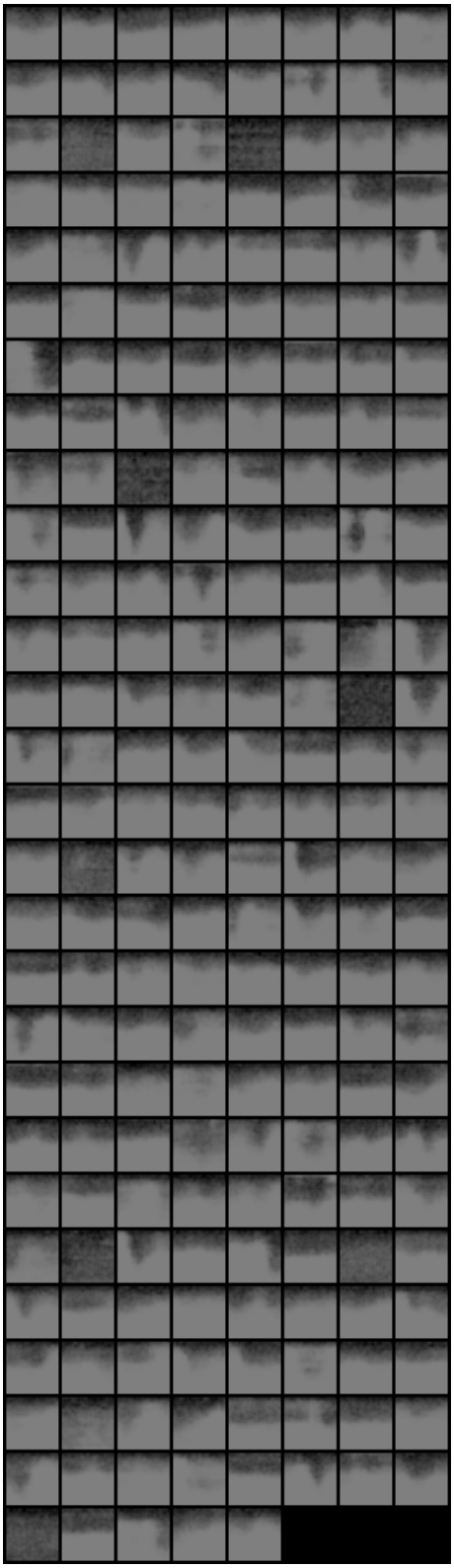

図 4.4　テスト段階によって得られたスペクトログラム

しかし、学習段階での損失関数の値、loss 値の推移は以下のようになった。

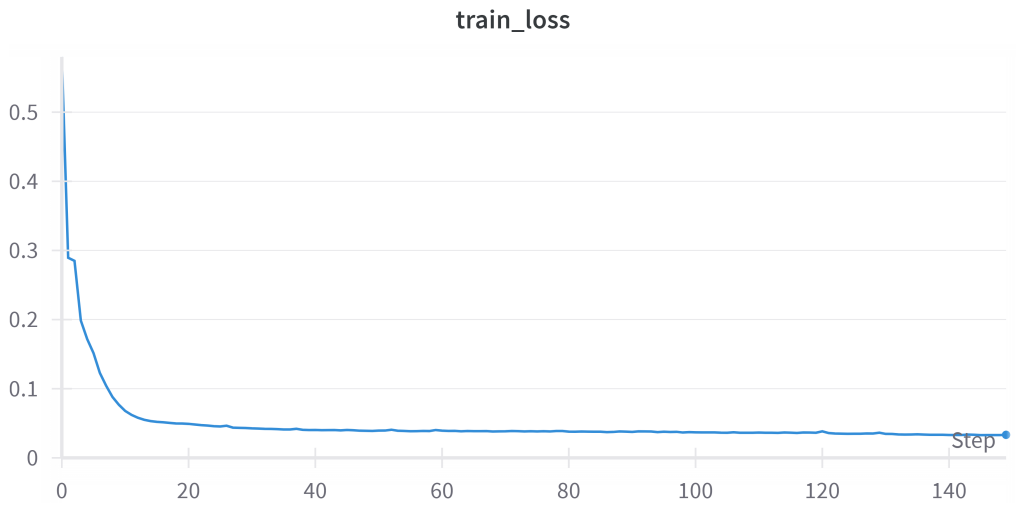

図 4.5　訓練段階における loss 値の推移

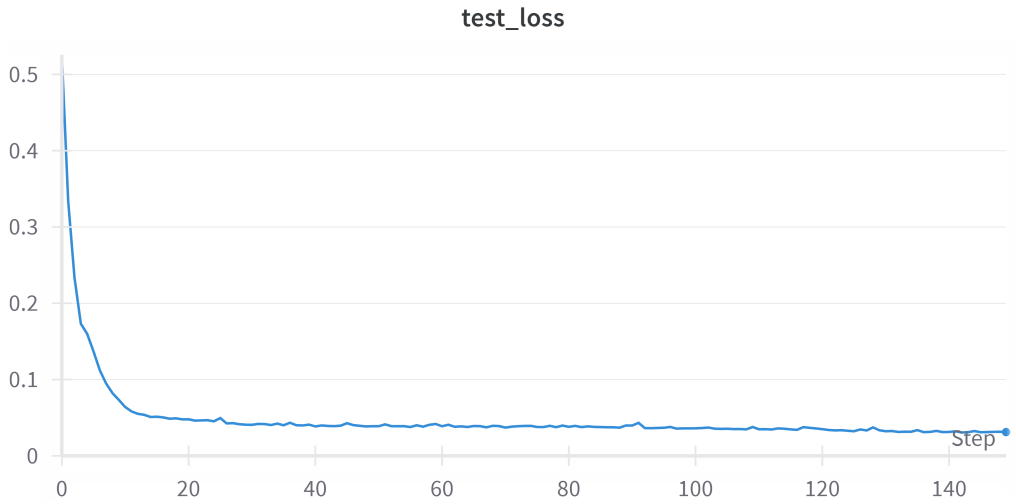

図 4.6　テスト段階における loss 値の推移

図 4.5、4.6 では、横軸をエポック数、縦軸を loss 値としている。しかし、学習段階において、訓練データおよびテストデータとして、学習データを用いて生成したスペクトログラムでは、loss 値はともに 0.1 程度まで抑えて収束させることができていた。つまり、ぼやけを排除したスペクトログラムを生成するには、さらに小さい loss 値で収束することが必要であることが示された。

ここで、第**??**部の 2.2 節で挙げた、スペクトログラムを生成する際の学習の収束の速さの比較をしたところ、以下のような結果が得られた。

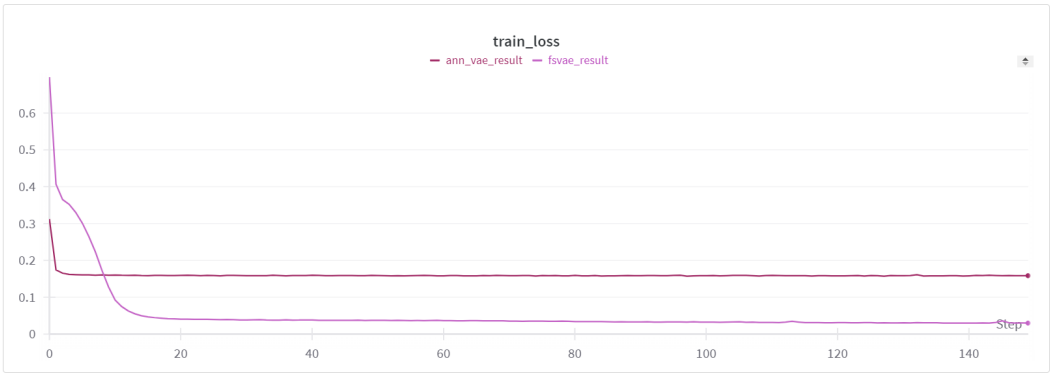

図 4.7　訓練段階における ANN と SNN の学習過程の比較

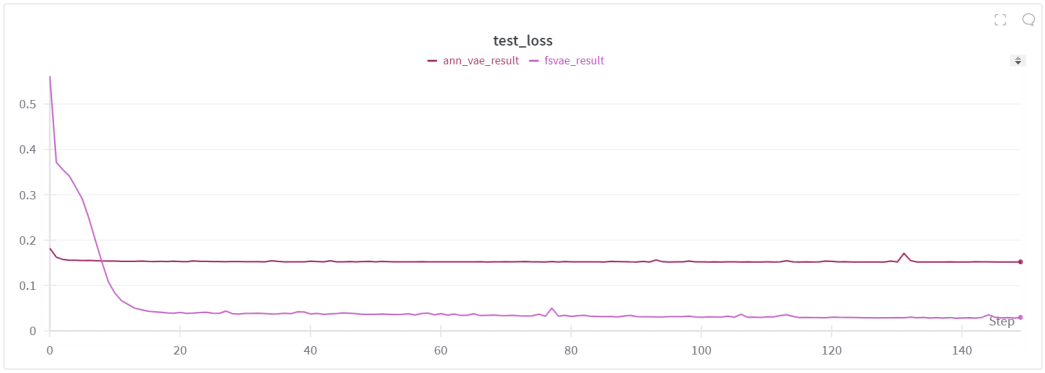

図 4.8　テスト段階における ANN と SNN の学習過程の比較

図 4.7 と 4.8 では、赤色のグラフが従来のニューラルネットワークである ANN、紫色のグラフが、スパイキングニューラルネットワークである SNN を表し、訓練段階において、横軸をエポック数、縦軸を loss 値としている。以上の図より、ANN と比較して、SNN の方が学習においてより小さい loss 値で収束していることがわかる。したがって、スペクトログラムの学習では、SNN の方が、性能が良いことが示された。

なお、学習段階で FSVAE を用いることで、216×1025 次元ものスペクトログラムの情報を保持するために、大量のメモリを消費してしまうことが明らかになった。これは、「学習コストの削減」の目的に反している。

（※文責: 田中柊真）

4.1.11　改善後の FSVAE による手法

以上のように、学習データとして音楽をスペクトログラムとして扱う手法ではなく、音楽を学習データとして学習させることにした。これにより、音楽をいくつもコピーして重ね、FSVAE の Encoder に入力させる。そして、FSVAE の Decoder から重ねられて出力された音楽を、足し合わせることで、音楽を生成させる。

さらに、FSVAE がメモリ消費量が大きくなることに対して、勾配累積と呼ばれる、勾配を計算するときに、バッチのサイズを小さくし、複数回累積した勾配の平均を取ることで、小さくなる前のバッチサイズのときよりも、メモリの消費が抑えられる手法を用いる。

加えて、学習率スケジューリングと呼ばれる、学習率を訓練段階で変化させる手法も組み込む。これにより、学習率を徐々に下げ、学習の収束を促す。

（※文責: 田中柊真）

4.1.12　改善後の成果と評価

訓練段階で得られた音楽は、ノイズが聞こえるものの、音楽として聴くことは可能であるほどであった。しかし、テスト段階で得られた音楽は、全体的にノイズがかかっており、音楽として聴くことは難しかった。

ここで、学習段階での損失関数の値、loss 値の推移は以下のようになった。

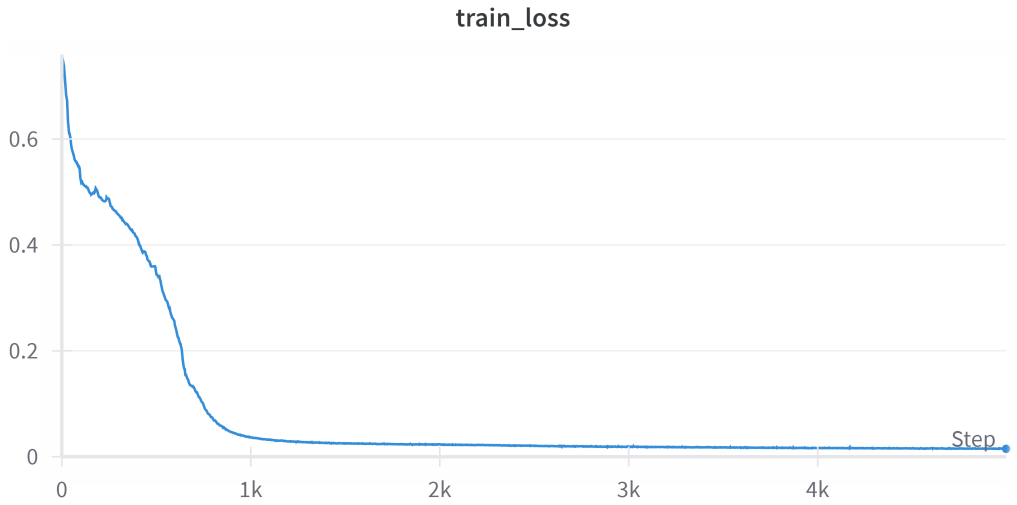

図 4.9　訓練段階における loss 値の推移

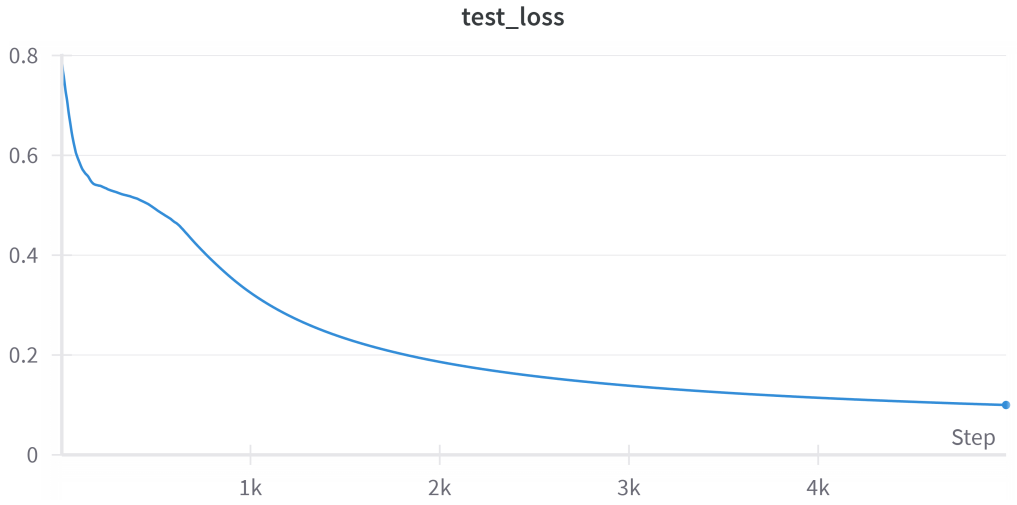

図 4.10　テスト段階における loss 値の推移

図 4.9、4.10 では、横軸をエポック数、縦軸を loss 値としている。改善後では、訓練段階におけ

る loss 値は、0.015 程度と、スペクトログラムによる学習と比較しても、小さな値で収束した。しかし、テスト段階の loss 値に関しては、改善前と変わらず、0.1 程度で収束した。音楽として聴くことができるようにするには、テスト段階の loss 値を訓練段階の値程度まで抑える必要があるという結論が得られた。

（※文責: 田中柊真）

4.2 music2music 班

music2music 班では、脳の仕組みを取り入れることで、低コストで学習を行える音楽ジャンル変換モデルを作成することを目的としていた。しかし、時間的な都合により、指定したジャンルの音楽を生成するモデルの作成までを行った。実際に、成果物の作成にあたって用いた手法を以下で説明する。

（※文責: 岩崎誠也）

4.2.1 提案手法

学習モデルの作成には、Python のオープンソースの機械学習ライブラリである「PyTorch」を用いた。このモデルは通常の GAN（Generative Adversarial Network）の Discriminator（識別器）に、Phase Shuffle という操作を行う層を追加した WaveGAN を基本とした。この WaveGAN の Generator（生成器）部分にレザバー計算を導入することで、高速な学習を行うことを可能にした。

（※文責: 岩崎誠也）

4.2.2 成果と評価

以上の手法を用いて、music2music 班では、特定のジャンルの音楽生成に成功したが、最終的に生成した音楽には全体的にノイズが現れてしまった。これは、入力として与えた音楽の高周波数成分を適切に学習できず、ノイズとして現れてしまったと思われる。しかし、音楽におけるリズムの学習と、その学習に必要とする時間的コストの削減を実現でき、1 エポックあたりにかかる学習時間を、レザバー計算を導入していないモデルと比較して、約 57 秒から約 37 秒へ短縮することに成功した。

（※文責: 岩崎誠也）

4.3 中間発表会

4.3.1 発表形式

中間発表会は、2023 年 7 月 7 日の 4・5 限時に開催され、本プロジェクトはプレゼンテーションベイ B で発表を行った。発表・質疑応答の合計 15 分を前半と後半でそれぞれ 3 回ずつ行い、発表後、聴講者に本プロジェクトの評価を行ってもらった。評価フォームには Google フォームを利用

し、プロジェクトポスターにアクセス用の QR コードを掲載した。評価項目は発表技術・プロジェクト内容の 2 つの項目について 10 段階での評価と、その評価理由と改善のためのコメントなどである。発表技術の評価については、プロジェクトの目的・成果を伝えるために効果的な発表が行われているか、プロジェクト内容の評価については、プロジェクトの目的と成果は優れているかという観点で行われた。

（※文責: 岩崎誠也）

4.3.2 発表スライド・ポスター

発表スライドについては、制作に無料のデザインツールである「Canva」を利用した。「Canva」を選択した理由は、無料で利用できる点と、オンラインで複数人が同時に編集を行えるためである。スライドの構成としては、最初に本グループ全体の背景・目的を述べた後、text2music 班、music2music 班の順番でそれぞれの概要・詳細を述べた。発表ポスターについては、本グループ単体では制作せず、プロジェクト全体のポスターを制作し、その中に背景・目的・活動計画を記載した。このプロジェクトポスターの制作にも、発表スライドと同様に「Canva」を利用した。

（※文責: 岩崎誠也）

4.3.3 発表練習

各グループで制作したスライド、またはポスターを利用して発表練習を行った。他グループと相互に発表の評価をして、意見を交換した。その後、指導教員も交えて発表練習を行い、課題や改善点を指摘していただいた。また、普段の活動場所だけではなく、より本番に近い状況を想定するため、実際に発表するプレゼンテーションベイ B でも発表練習を実施した。

（※文責: 岩崎誠也）

4.3.4 評価の集計

評価件数は 39 件で、発表技術についての平均点は 8.05 点、プロジェクト内容についての平均点は 8.41 点であった。そのうち、評価者の種別は学生が 89.7%、教員が 10.3% であった。ただし、グループ A 単体への評価ではなくプロジェクト全体での集計である。それぞれの分布を、に示す。また、評価理由と改善のためのコメントを一部抜粋し記載する。まず、高評価であったものを以下にまとめる。

- 「図を多く用いて、視覚的にわかりやすいスライドになっている。」
- 「やりたいことが分かりやすくまとめられていました。図や矢印があって見やすかったです」
- 「グループごとの発表の中で、課題や手法などの説明がされていてよかった。」
- 「具体的な技術の説明があって良かった」
- 「目標や計画が具体的で良いと思います」

次に、低評価であったものを以下にまとめる。

- 「スライドの文章が少し難しくわかりにくいなと思ったと感じた。知識があまりない人にも分かるような説明を増やせると良いと思った。」
- 「グラフの説明が不十分だったかも？」
- 「全てのグループの説明が聞けると良かった。」
- 「専門用語についての説明がもう少し欲しい。」
- 「どの程度の性能のものが出来上がるのか予想がつかないので、現時点では評価がしにくい。」

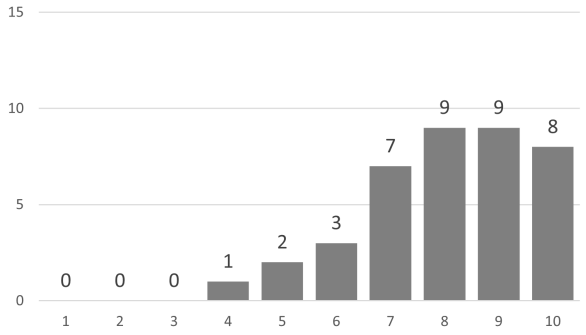

図 4.11　発表技術の評価

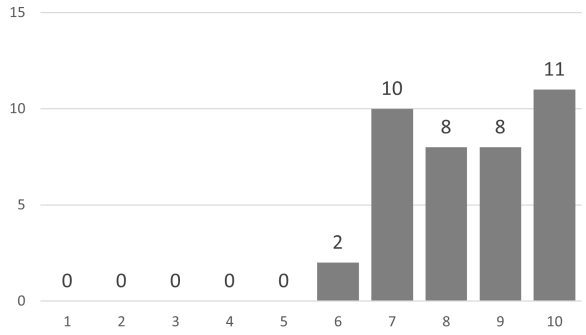

図 4.12　プロジェクト内容の評価

（※文責: 岩崎誠也）

4.3.5　総評

発表技術については、図を多用することで内容の理解がしやすくなったが、文章が簡素になり情報が読み取りにくくなってしまった。また、発表形式について 3 つのグループに分かれて発表したため、すべてのグループの説明を聴講できないという意見が見られた。プロジェクト内容については、中間発表会の時点で目的・手法について述べることができたが、生成した音楽などの制作物は提示できなかったため、最終的な着地点が伝わりにくかったと考える。

（※文責: 岩崎誠也）

4.4 成果発表会

4.4.1 発表形式

成果発表会は、2023 年 12 月 8 日の 4・5 限時に開催され、本プロジェクトはプレゼンテーションベイ B の半分のエリアで発表を行った。発表・質疑応答の合計 15 分を前半と後半でそれぞれ 3 回ずつ行い、発表後、聴講者に本プロジェクトの評価を行ってもらった。評価フォームには Google フォームを利用し、プロジェクトポスターと発表スライドの最終ページにアクセス用の QR コードを掲載した。評価項目は発表技術・プロジェクト内容の 2 つの項目について 10 段階での評価と、その評価理由と改善のためのコメントなどである。発表技術の評価については、プロジェクトの目的・成果を伝えるために効果的な発表が行われているか、プロジェクト内容の評価については、プロジェクトの目的と成果は優れているかという観点で行われた。

（※文責: 岩崎誠也）

4.4.2 発表スライド・ポスター

発表スライドについては、中間発表会と同様、制作に無料のデザインツールである「Canva」を利用した。中間発表会で使用した発表スライドに対して、実際の活動内容や結果などを加筆修正した。発表ポスターについては、中間発表会と同様、本グループ単体では制作せず、プロジェクト全体のポスターに背景・目的・活動記録を記載した。このプロジェクトポスターの制作も、発表スライドと同様に「Canva」を利用し、中間発表会で使用したプロジェクトポスターのデザインを変更した。

（※文責: 岩崎誠也）

4.4.3 発表練習

各グループで制作したスライドを利用して発表練習を行った。また、中間発表会と同様、普段の活動場所だけではなく、より本番に近い状況を想定するため、実際に発表するプレゼンテーションベイ B でも発表練習を実施した。

（※文責: 岩崎誠也）

4.4.4 評価の集計

評価件数は 48 件で、発表技術についての平均点は 8.02 点、プロジェクト内容についての平均点は 8.06 点であった。そのうち、評価者の種別は学生が 85.4%、教員が 6.3%、一般が 8.3% であった。ただし、グループ A 単体への評価ではなくプロジェクト全体での集計である。それぞれの評価分布を、に示す。また、評価理由と改善のためのコメントを一部抜粋し記載する。まず、高評価であったものを以下にまとめる。

- 「身振り手振りがあってスライドのどの部分を見れば良いのかが伝わりやすかった。少し聞きづらい部分があったように感じた。」
- 「難しい内容でしたが、画像や図を適宜用いていて、わかりやすかったです」
- 「実際に生成した音声を流してくれるなどして研究の成果が明確に表されていた。」
- 「レザバーコンピューティングについての説明がわかりやすかった」
- 「高速化までできており、成果物（音）を流すことで非常にわかりやすかった」

次に、低評価であったものを以下にまとめる。

- 「途中から成果発表というより、用語解説の印象を受けました」
- 「専門用語が多くて何を伝えたいのかがよく分からなかった。（レザバー計算、モジュール、GAN など）」
- 「スライド枚数が多い。グラフや図の説明が不足気味でした。」
- 「声量がもっとあるといいかも」
- 「本来の目的のジャンル変換できると良かった。」

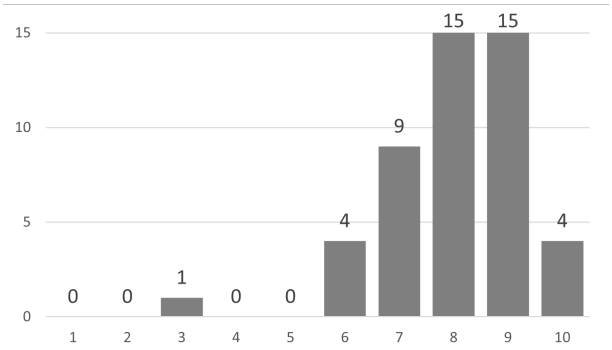

図 4.13　発表技術の評価

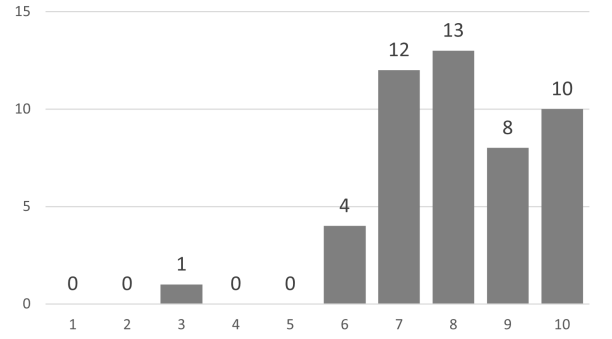

図 4.14　プロジェクト内容の評価

（※文責: 岩崎誠也）

4.4.5　総評

発表技術については、身振り手振りや生成した音楽を聴講者に聞かせることなど、プロジェクトの目的・成果を効果的に伝えるための工夫があった。しかし、作成したモデルを説明する上で、専門用語の使用回数が増えてしまい、用語自体の説明に費やす時間が多くなってしまった。また、それに伴い、グラフや図などの説明が不足したことと、発表者の声量が小さいという意見も見られた。プロジェクト内容については、音楽生成のコスト削減についての肯定的な評価や、当初の目的であった音楽ジャンル変換についての言及も見られた。

（※文責: 岩崎誠也）

第 5 章　まとめ

5.1　前期

前期の活動は、各々のやりたいテーマを考えて、そこからテーマを決定した。その後、使用するモデルについての勉強会や開発に必要な Git などの基礎的な知識を習得するための勉強会を行った。その後は、香取教授から借用した PC の開発環境整備を行ない、中間発表に向けてのスライド作成や練習、中間報告書の作成を行なった。スライド作成では、プロジェクトロゴの作成から、スライドのデザインまでを行い、図などを用いて分かりやすく発表できるようにした。また、開発環境の整備として、PC には昨年と同様に、Ubuntu をインストールした。Nvidia のドライバのインストールや、Rye と呼ばれる環境管理ツールの導入を行い、Windows 上での開発よりもより素早く開発を行える用、環境を整えた。

中間報告書では、共同で作業を進めるにあたり、Docker を用いた Tex 環境の構築や、GitHub を用いた共同作業の方法を行なった。結果として、後期での活動に向けて、効率的なコードの共有や、共同作業ができる方法を学ぶことができた。

良かったことは、早々にグループの課題設定を明確にできたことだ。これにより、今後の活動が明確になり、必要な知識を共有するために勉強会を即座に行うことができた。その他にも Slack や Notion などで意思疎通や議事録の管理を行う環境を活動初期に整備できたことで、中間報告書の作成や、知識の共有などを円滑に行うことができた。これは、前もって先輩などからプロジェクト学習の活動についてのアドバイスなどを受け取っていたことも大きいと感じた。中間発表では何度も先生方に発表のレビューをいただき、グループ内でも何度もレビューを行ったことで、発表のクオリティを高めることができた。結果として発表評価アンケートでも高い評価をいただくことができた。

反省は、知識の収集や課題の解決方法を決める時間が長くなってしまい、後期に行う実装作業が多くなってしまったことだ。作業工数がどの程度かかるのかを明確に把握していなかったため、このようなことが起こった。また、活動が本格化してからは、先生方や先輩方への相談する回数が少なく、自分たちで文献の調査などを全て行ってしまったため、課題と関係のない知識の習得にも時間を使ってしまった。これは、先輩などに相談をすることで本筋とは関係のない作業を減らすことができると感じた。全体を通して、効率的な作業を行うことをグループ全体で意識していたが、この反省は後期まで続いてしまった。効率的な開発を行うためにも他の人に相談することの大切さを感じた。

（※文責: 山内大翔）

5.2　夏季休暇

夏季休暇の活動は、メンバーが帰省やインターンシップなどにより、全員が集まっての活動ができなかった。しかし、夏季休暇前に必要な作業を明確にしていたため、時間のあるメンバーがそれ

ぞれの作業を行うことができた。これは、Notion や Slack などのツールを用いて、情報や方針の共有を常に行っていたことが大きかった。

（※文責: 山内大翔）

5.3 後期

前期で実装の出だしが遅れた反省を生かし、後期では実装を中心に活動を行った。また、前期で行っていた解決策では工数的に厳しいと判断し、変更を行った。具体的には、音楽ジャンルの変換や文章からの音楽生成ではなく、音楽の生成に注力し、CycleGAN や拡散モデルの構築ではなく、変分オートエンコーダーと敵対的生成ネットワークの構築に注力した。作業が個人に集中してしまうことを防ぐために、Slack などでプロジェクト時間外でも連絡を取り合い、作業の進捗や、実装中に発生した問題の共有を行った。特に、敵対的生成ネットワークの班では、毎週のプロジェクトの時間にて、各々の進捗を報告し、生成された音楽の評価を行った。開発では 2 つの班で共通して使えるコードなどは積極的に共有し、開発の効率化を図った。また、開発を進めていく上で、敵対的生成ネットワークを用いた音楽生成モデルでは、スペクトログラムを利用した学習を行うと、市松模様のようなパターンが含まれた画像が生成されてしまうことがわかった。そのため、市松模様のパターンを持った生成データに対する対処法が示されていた WaveGAN の論文を参考に、音声の波形をランダムにシャッフルし、識別する識別器を用いて、音声の波形を生成するようにした。こちらも他の文献調査と同様に班内で手法についての共有を行い、成果発表に向けたメンバー同士の理解度の向上を図った。

また、変分オートエンコーダーの班では、毎週のプロジェクトの時間にて、各々の進捗を報告し、実装にあたって遭遇した問題点などを共有した。実装で困った部分があれば対面で集まる水曜日、金曜日にコードを共有し、問題点の解決を行った。しかし、実際には、週 2 日では解決しきれない問題もあり、より多くの日程を設けるべきだったと感じた。また、FSVAE を使ったスペクトログラムの生成においては、変分オートエンコーダー特有の生成画像のぼやけが発生してしまった。これは DNN を用いた変分オートエンコーダーでも同様の現象が発生した。また、レザバー計算と違い、モデル全体で逆誤差伝播法を用いた学習を行っているため、自動微分の結果の保存のため学習に必要なメモリが大きくなってしまった。結果として GPU のメモリ不足により、十分な解像度を持ったスペクトログラムの学習ができないという問題が発生した。こちらは、学習データをスペクトログラムから音声波形に変更することで、解決したが、それまでに多くの時間を費やしてしまった。また、メモリ不足に対応するために 1epoch あたりのバッチサイズを小さくしたため、学習に時間がかかってしまった。これは、学習率スケジューラの利用や最適化関数の変更などを行い、解決した。

良かった点は、前期での反省を生かし、実装に取り掛かることで活動時間を有効に使えたことだ。また、前期での経験を生かし、プロジェクト時間外でも連絡を取り合い、作業の進捗や、実装中に発生した問題の共有を行った。特に、できるだけ毎週の授業時間に作業を終了させることを意識し、進捗を随時共有しながら作業を行うことができた。その結果、過度な追加の作業を行うことなく、前年度と違い負傷者が出ることなく活動を終えることができた。反省点は、成果物の作成において明確な期限を設定していなかったため、2 つの班での作業の進捗が異なってしまったことだ。また、できるだけ毎週の授業時間に作業を終了させることを意識したが、それでも終わらない

作業はその担当者の負担が大きくなってしまった。より柔軟な作業量の調整や、作業進捗を Slack などを用いて手動で管理するのではなく、GitHub issue などを用いてより効率的に管理するべきだったと感じた。また、変分オートエンコーダーの班では、音声の生成が遅れてしまい、最終発表のスライドの作成に割く時間が極端少なくなってしまった。

（※文責: 山内大翔）

5.4 成果について

本グループの目的は脳構造から工学的な応用というアプローチのもと、脳の仕組みを取り入れたレザバー計算やスパイキングニューラルネットワークを生成ネットワークにおける実用性を検証することであった。具体的には、レザバー計算を生成器に取り入れた敵対的生成ネットワークを用いた、音楽生成や、スパイキングニューラルネットワークを取り入れた変分オートエンコーダーを用いての音楽生成を行なった。1 つ目のレザバー計算を生成器に取り入れた敵対的生成ネットワークを用いた音楽生成では、WaveGAN の論文で用いられている Phase Shuffle を用いた識別器を用いて、音声の波形を生成するようにした。この際、生成器の最終層は一層の全結合ネットワークを用いて、勾配降下法によって学習された。識別器、生成器ともに最適化関数は Adam を使用している。結果として、DNN を利用した音楽生成モデルよりも、1 エポック当たりの学習時間が 57 秒から 37 秒に短縮された。また、ジャンルを絞った音楽の生成に成功した。ノイズが生成された音声に乗ってしまうなどの問題もあったが、ノイズ除去を行うなどの改善を行うことで、より人間が聞きやすい音声の波形を生成することが確認できた。特にハウスやテクノなどのジャンルにおいては、リズムが強く現れた音声の波形を生成できることが確認できた。

次に 2 つ目の目標であるスパイキングニューラルネットワークを取り入れた変分オートエンコーダーを用いての音楽生成では、スペクトログラムの生成では、変分オートエンコーダー特有のぼやけが発生してしまい、音声の波形を利用した音楽生成では、音声の波形の生成に成功し、訓練データ、テストデータ共に Loss は減少を確認できた。DNN よりもスペクトログラムの生成において、精度の向上が確認できた。しかし、Loss は減少したものの人間が音楽として聴けるような音声の波形のサンプリングを潜在空間から行うことができなかった。以上のことから、レザバー計算を取り入れた敵対的生成ネットワークを用いた音楽生成は、既存の音楽生成手法と比較して、学習時間が短縮され、ジャンルを絞った音楽の生成が可能なことが確認された。また、スパイキングニューラルネットワークを取り入れた変分オートエンコーダーを用いた音楽生成では、訓練データに対しては人間が音楽として聴けるような音声の波形のサンプリングを行うことができたが、テストデータやランダムに生成した潜在空間からのサンプリングでは、人間が音楽として聴けるような音声の波形のサンプリングを行うことができなかった。

したがって、レザバー計算を取り入れた敵対的生成ネットワークを用いた音楽生成は、工学的応用性があると言えるが、スパイキングニューラルネットワークを取り入れた変分オートエンコーダーを用いた音楽生成は、現状の音声波形を利用したモデルでは不完全であり、MusicVAE[11] のように midi データを学習に利用するなどの使用するメモリやタスクの複雑性に関する改善が必要であると言える。

（※文責: 山内大翔）

第 6 章　今後の課題

text2music 班については、FSVAE を導入した StableDiffusion の開発という最終的な到達目標までは行き着いていないため、今後の課題としては、StableDiffusion まで組み合わせることが挙げられる。また、同じデータをコピーする段階で同一の内容が出てしまう問題が解決できていないので、ベクトル列にずらすことで、時系列に対して学べるように改善することも挙げられる。

music2music 班については、今回生成した音楽に全体的に乗ってしまったノイズを軽減することが今後の課題の一つである。また、本来の目的である CycleGAN を用いた音楽のジャンル変換まで最終的には到達できておらず、指定したジャンルの音楽の生成で留まってしまっているため、本来の目標であるジャンル変換を行うことも挙げられる。

レザバー計算に関連した手法として、物理系の動的な挙動を利用して情報を処理する物理レザバー（Physical Reservoir）が存在する。音楽生成においても、物理レザバーは一部の研究や実験で利用されており、物理系のダイナミクスを音楽の生成に組み込むことで新しい音楽のパターンや表現を生み出すことが可能とされている。そのため、上記の今後の課題に加えて、物理レザバーを用いた音楽の生成も検討したいと考えている。

（※文責: 工藤大）

第 7 章　個人の取り組み

7.1　山内大翔

プロジェクト学習では、スライドの作成から前半部分の発表、文献の調査や、実装などを行った。中間発表におけるスライドでは、文献調査を担当した箇所のスライドを作成した。成果発表会のスライドの作成では、自分が実装に関わった変分オートエンコーダーの学習についてのスライドや学習結果のスライドを作成した。どちらのスライドでも図やグラフなどを用いて、直感的に理解できるよう、スライドを工夫した。特に図では話したい部分に注目が集まるように、図を一部だけ表示したり、拡大し注釈をつけたりすることで、聴衆が理解しやすいようなデザインを心がけた。

成果物作成のために必要な文献の調査では、前期は課題に対する理解を深めるために、音楽生成に関する文献の調査や、スパイキングニューラルネットワークを利用した学習に関する文献の調査を行った。最初期の段階では使用するモデルの候補すら決まっていなかったため、LSTM を利用した音楽生成に関する文献の調査や、画像生成を応用した音楽生成に関する文献の調査など、様々な文献の調査を行いグループメンバーに共有することを心がけた。また、後期においては、実装も進んできたため、メモリ不足に関する問題を解決するための勾配累積に関する文献の調査や、学習データの前処理に関する文献の調査など、実装に関する文献の調査を行った。特に、調査した内容をグループメンバーに共有し、実装までを円滑に進めるため、実装されたコードと調査した文献を見比べながら説明するなど、実装をグループ内で効率的に進めるために、わかりやすい調査内容の共有を徹底した。

成果物の作成では、主に学習用のコードの実装や、調べた文献を元にしたコードの改良などを行った。スペクトログラムの学習においてメモリ不足が発生した際は音声波形ベースの変分オートエンコーダーの実装などを進めた。また、モデルの構造を変更したことにより、学習速度が低下した際は、学習率スケジューリングや、最適化関数の変更などを行い、学習速度を 2 倍にすることに成功した。その他にも、メモリ不足問題においてデータの次元数の大きさも原因の一つであったため、サンプリングレートの調整や、音声の長さの調整に必要なメソッドの作成を行なった。また作成したメソッドは、容易にグループ間で共有できるように python パッケージ化を行い Pypl 上に公開した。実験結果をグラフ化し管理できるように、グラフを保存するためのコードの実装や、生成した音声やスペクトログラムを保存するためのコードの実装を行った。実装したコードで作成されたグラフは、今回のレポートや成果発表のスライドにも使用した。VAE ベースのモデルにてぼやけたスペクトログラムが生成される問題に対して、解決策を模索するために、スパイキングニューラルネットワークを利用した AE の実装を行い、VAE で作成されたスペクトログラムと比較を行った。また、プロジェクトを円滑に進めるために、調査書の作成に必要な LaTeX 環境をまとめた Docker ファイルの作成や、プロジェクト学習で利用する PC を大学の外から接続するための環境構築など、環境の整備を主に担当し、効率化を目指した。

（※文責: 山内大翔）

7.2 田中柊真

text2music 班に所属し、既存モデルの改善を目的とすると決定してからは、計算コストの削減を実現するための手法を模索した。既存モデルとして、拡散モデルを選択し、拡散モデルについての知識を、Stable Diffusion の論文や、書籍を用いて収集した。さらに、計算コストの削減を期待できるニューラルネットワークの枠組みであるレザバー計算に着目した。既存モデルに対して、レザバー計算を用いることで、計算コストを削減できると考えた。そこで、レザバー計算の基礎事項や活用例などの文献調査を進めた。文献調査を進めていくうちに、班の方針として、FSVAE をはじめとする手法を中心に進めていくことになったため、レザバー計算の知識は成果には現れていない。中間発表までに、text2music 班において、Stable Diffusion にレザバー計算を組み込む見通しは立っていないものの、主に Stable Diffusion とレザバー計算の調査を行った。

中間発表するにあたって、既存モデルの概要および背景を理解できるようなスライド作成を行った。スライド作成においては、簡潔な文章かつ、事前知識がない聴衆が見ても理解できるように工夫をした。また、発表においては、既存の Stable Diffusion と CycleGAN における問題点を、事前知識を持っていない聴衆が理解できるように、具体例を交えて発表するように、構成を工夫した。

後期開始後は、モデルの開発を進めた。開発を進めていくなかで、FSVAE の Decoder によって出力されるスペクトログラムが、全体的にぼやけている問題が起きた。原因として、学習データのスペクトログラムにあると考え、学習データのスペクトログラムの輝度を調整し、より濃淡を強調させるデータセットを作成した。また、もともとの FSVAE のコードについて、勾配累積の機能を追加し、メモリの消費量を軽くすることを目指した。さらに、前期にレザバー計算の文献調査をしていたことを活かし、再度、レザバー計算を重点的に調査を行った。レザバー計算のなかでも、「FORCE 学習」と呼ばれる枠組みについて、学習を進めた。レザバー計算の学習を進めることになった理由として、開発したモデルに、FORCE 学習を加えることで、ぼやけたスペクトログラムに対し、FORCE 学習で学習したリズムを合わせることで、リズムを持った、ぼやけが解消されたスペクトログラムを生成できるという仮定をしたからである。なお、FORCE 学習の知識収集の着手を始めたタイミングが 10 月と遅れたため、成果物の機構に含めることはできなかった。しかし、text2music 班の成果物に加えれば、学習の精度が向上し、誤差が小さくなることで、より高度な成果物を目指すことができると考えている。さらに、レザバー計算を用いることで、FSVAE によってメモリの消費量が大きくなったことによる計算のコストを、軽量化できると考えており、実現すると、text2music 班の目的を達成させることができるようになると考える。

成果発表会においても、このグループがこのプロジェクトを取り組むことになった背景や、目的を、プロジェクトの内容を知らない聴衆にわかるような発表構成を考えた。また、このプロジェクト全体で 1 年を通して行ってきたことを、簡潔に冒頭で発表をした。また、発表会に参加した高校生に対して、このプロジェクト全体で行ったことや、脳や人工知能に関する基礎的な内容の紹介をした。

（※文責: 田中柊真）

7.3 山谷璃輝

前期は、勉強を中心に行った。始めは、生物の脳を模倣することに取り組みたいと考えていたが、アイデアを出していくうちに音楽生成に興味を持ち、文章から音楽生成に取り組むことにした。その後、学習コストの削減という目標を立て、目標を達成するうえで役立ちそうなものや音楽生成をする AI について勉強した。具体的には、DiffusionModel について解説している本や動画、Stable Diffusion や FSVAE についての論文を読んで勉強をした。また、他のグループメンバーと比べて、AI に関する基礎的な知識が不足していると感じたため、その勉強も行った。プロジェクト学習の時間には、読んできた論文についての情報をみんなに共有した。

中間発表では、AI に関する知識が不足している点を活かして、分かりやすいスライドや発表にするよう心掛けた。発表練習を通して、先生方からいただいたフィードバックを踏まえ、見やすいスライドを作成することができた。しかし、2 つの班に分かれて活動を行ったため、他のグループより発表時間が少ないということもあり、専門知識に関する説明が不足している点が少し目立ったように感じた。

後期は、前期に引き続き、勉強をしつつ、実装に取り掛かっていった。基本的に、毎週やるべきことをメンバーで話し合って、役割を分担して、作業に取り掛かっていった。初めは、diffuser という既存の StableDiffusion を動かすことを中心に行っていった。わからないことが多かったため、すぐに調べたり、他のグループメンバーに聞いたりすることを徹底した。まずは、diffuser の使用例で紹介されていたデータセットを用いて学習させた後、他のグループメンバーが作成したデータセットを用いて学習させた。その後、FSVAE を通したスペクトログラムによる音楽生成ができないという問題が発生し、グループで話し合ったところ、画像の濃淡をはっきりさせることでノイズではなく音楽が生成できるのではないかと考えた。そこで、スペクトログラムに exp をかけてスペクトログラムの値を大きくし、画像の白黒をはっきりさせた。しかし、音楽生成はできず、FSVAE とスペクトログラムの相性があまりよくないという考えになり、扱うデータをスペクトログラムから音声波形に変更することになった。そこで、音声波形のデータセットを他のグループメンバーと協力して作成した。

最終発表では、中間発表の反省と先生方からいただいたフィードバックを踏まえ、専門的な説明をできるだけ省き、成果物を示すことや活動内容について詳しく説明することに時間をかけることで、短い時間内でも分かりやすい発表ができたのではないかと考えている。しかし、発表に対する質問の受け答えに関しては私自身が対応できることがなかった。それは、私自身が実装した部分以外の知識が不足していたことによるものだと考えており、発表前のグループ内での情報共有や勉強不足があったと反省している。

活動の全体を通して, 想定外の問題が起きて、どう対処するかを話し合う場面が多かったが、対処法を提案することがあまりできなかった。大きな原因として考えられるのは、講義以外でのグループでの開発を今までしてこなかったことと AI に関する知識の不足である。また, 分からないことがあった時に、一人で調べることに時間をかけすぎてしまい、解決できなかった時に、他の人に聞くまでの時間がもったいなかったように感じた。そこで、グループでの作業は頻繁に状況を共有しておくことが大切だと感じた。反省点としては、勉強不足、班のメンバーとのコミュニケーション不足であると感じている。プロジェクト学習の活動以前と比べると、格段に AI に関する知識は身に着けることができたが、それでもこの活動の目標に対して、知識の量が足りていないと感

じた。また、開発経験もほとんどなかったため、開発の流れをつかむまでに、多くの時間を要してしまった。コミュニケーション面では、班のメンバーの作業の進捗状態があまり把握できていなかったことや、不明な部分を班のメンバーに確認することを迅速に行えなかったことが問題だったと考えている。共有する必要がないかもしれないことでも班のメンバーと共有するような習慣を身に着けるといったことを行っていれば、このような問題は防げていたと考えている。これらのような、プロジェクト学習を通じて得たグループ開発の経験や知識は、今後のグループ活動に大いに役立っていくだろうと考えている。

（※文責: 山谷璃輝）

7.4 岩崎誠也

前期の活動では、最初にプロジェクトとして取り組むテーマの決定を行った。テーマの決定に際して、各自が興味のあるテーマを提示していくブレインストーミングを行った。まずアイデアの質より量を優先することで、新しい発想を得やすくなったり、より良いアイデアに昇華させたりできる効果的な方法であると再認識できた。これを元に、脳の仕組みを取り入れた機械学習に取り組むグループ A と、前年度の活動を引き継ぎ、AI カーをテーマとするグループ B に分かれた。その後、グループ A で取り組む具体的なテーマを話し合った結果、音楽生成をテーマに設定することとなった。また、音楽生成というテーマを軸にしたうえで、学習コストの削減をグループ A 全体の目的として据えた。さらにグループ A 内で、文章から音楽を生成する「text2music 班」と、音楽のジャンルを変換する「music2music 班」に分かれた。音楽ジャンル変換を実現するために使用するモデルの決定と、機械学習についての基礎的な知識の情報収集を分担して行い、グループ内で相互に共有した。これに並行して中間発表会に向けての計画や、発表スライドのデザイン考案、グループロゴの作成、購入物品の決定などを行った。発表スライドのデザインについては、各メンバーが 1 つ以上のサンプルを考案し、文字の配置や色彩の明瞭さなどを検討して多数決にて決定した。次に、完成した発表スライドを用いて、プロジェクトメンバー内で発表練習を行った。実際の発表を想定して練習しながら、プレゼンテーションの見直しやスライドの修正を行うことで、発表としての完成度を高めていくことができた。中間発表会を終えてから発表技術・内容の反省や、中間提出物を作成した。

後期の活動では、前期での計画を実現するために音楽データの収集やモデルの構築をグループ内で分担して行った。当初「music2music 班」では、音楽ジャンル変換を目的としていたが、時間的な都合で指定したジャンルの音楽生成を目的とした。私は、主に発表スライドとプロジェクトポスターの作成を担当した。成果発表会で用いる発表スライドは、中間発表会で用いた発表スライドをベースとして加筆修正を行った。成果発表会での使用を想定して、文章量を可能な限り減らし、図を増やすことで視覚的に内容が伝わるように心がけた。また、成果発表会では、実際に生成した音楽を聴講者に聞かせることで発表に説得力を持たせた。自分としては、中間発表会の時の反省を活かして発表技術を向上させることができたと考える。

今回実際に環境構築や、機械学習のモデル構築は知識不足から他のメンバーに頼りきってしまったことや、発表技術の未熟さが反省点であった。しかし、プロジェクト学習全体を通して、グループで活動する上での情報共有の方法や課題の設定、活動の役割分担、スライド作成や発表練習などの準備、実際の発表など、卒業研究や社会に出たときに有用な経験を得ることができたと考える。

（※文責: 岩崎誠也）

7.5　小林未佳

生成 AI を使ったり、興味はあったものの、あまり詳しくその仕組みを調べたことはなかった。このプロジェクト学習が実施されたのは 2023 年だが、その前年に当たる 2022 年は「生成 AI 元年」と呼ばれ、この技術が急速に進化し、未開の領域を切り開いていた。人工知能が自律的に新しいデータやコンテンツを生成し、これまでにない創造性や革新性をもたらした。最近では、OpenAI の GPT シリーズや Stable Diffusion など有名な生成 AI モデルが多く開発され、絵画、音楽、文章生成などで驚くべき成果を上げている。同様に、音楽生成 AI は、その可能性が広がる中で、未知の美と創造性を生み出す新たな局面を切り開く手段として注目されることになるだろう。また世界各国でもデータやモデルの取り扱いに関する法整備が進んでいる。一方、90 年代のバブル崩壊後御不況から立ち直ることができず、2000 年代に急速に進んだ世界的な情報産業化の流れに後れを取ったと言われている日本は AI 研究・産業活性化のため、比較的データに対する規制が緩いと言われている。したがって、これは日本人にとってはビジネスチャンスであるとも言えるだろう。個人的には生成 AI について詳しく調べたことで、視野がとても広がったと考えている。

具体的な活動としては、5 月は様々な音楽生成のアプローチを検討した。6 月には生成 AI に関する論文を調査し、習得するべき知識の確認を行った。8 月には、拡散モデル、レザバー計算、SNN に焦点を当てた本や論文、ネット記事の情報を通じて、各技術の利点と課題を明確にした。これにより、異なる技術の統合に対する理解を深めた。9 月からは、プロジェクトにおいて Huggingface の導入を検討し、API やライブラリを利用して Google が提供する「MusicCaps」で音楽のスペクトログラムとその説明テキストを含むデータセットを作成した。この際、Huggingface Hub の公式ドキュメントを読んで、Google Colaboratory 上へデータセットをダウンロードする仕組みに関する知見を得た。

今回のプロジェクトで特に苦労した点は、Huggingface のドキュメントを読むことだった。結果的にはデータセットの作成方法を 3 つも試すこととなり、私はこの仕事に 1 ヶ月もの時間を要してしまった。データセットづくりに関する方法を調べていた段階では、データセット作成を無料枠の Google Colaboratory と自分のパソコンで行っており、メモリ容量が少なかったことも原因だった。分からなかいことが多かったので、データセットをロードする関数を実行した際にデータをロードする時間がかかりすぎるのはデータセットリポジトリの構成が悪いせいだと考えて 1 から作り直したり、Huggingface Hub のリポジトリにある質問コーナーで Huggingface 社の人に連絡を取ってエラーの内容を教えてもらうこともあった。またいざ Huggingface Hub の公式の train_text_to_image.py ファイルを実行しようとして愕然としたのは、その python スクリプトを動かすには半精度などの技術を使っても 24GB 以上の GPU が必要だとちらっと書いてあったことだった。しかし、この問題は山内君が LORA という技術を使うことで 12GB ほどに減らすことができるという情報を得ていたため、何とかコードを動かすことができた。

また、細かいところでは最終的にモデルから生成されたスペクトログラムから wav ファイルを再生する関数とその要件を定義したり、逆にデータセットを使用する時点での wav ファイルからスペクトログラムを生成する際の周波数の調整をする作業があった。これらの作業を事前に定義する全体の工程に関する詳しい話し合いはあまりしていなかったと反省した。これはプロジェクトが

始まる前に考えておくべきことだったが、全ての工程に対して可能不可能の判別を事前に付けるのは難しい。実際に就職した先の企業でプロジェクトに取り組む機会があれば、事前の要件定義や工程管理、プロジェクトメンバーとのコミュニケーションなど、コーディング作業以外の気を付けること、学ぶべき知識を身に着けておきたいと痛切に感じた。しかし個人的に最も大変だったのは Huggingface Hub の公式ドキュメントの翻訳ではなく、FSVAE のコードを自分の目で理解することだった。とても長いコードだったうえ、一応夏休みに一通り SNN については学んだものの、そこから pytorch の tensor 操作やモデルのアルゴリズムを組み込んだ実装の複雑さにかなり苛まれた。しかし石の上にも三年、分からないプログラムの上には三か月ということで何とか最終的には構造を理解できたのでとても良い経験になったと思っている。

（※文責: 小林未佳）

7.6　工藤大

本研究に利用する学習手法の決定に当たって、まずはじめに基となる研究や生成 AI の基礎知識に関する論文を中心に読んで先行研究の調査を行い、既存の生成モデルに関する知識を深めた。主に敵対的生成ネットワーク、レザバーコンピューティングに関連する文献調査を重点的に行い、従来までの生成 AI に挙げられる問題点と、改善案の検討等について、図にまとめてグループ内で共有するという形で知識の共有を行った。この段階や後の報告書作成の際に、多くの論文に目を通すという機会が多々あったため、論文をなるべく早急に読み、要約をするという経験をすることができた。今までこのような経験がなかったわけではないが、短期間の間に多くの論文を読むという経験は初めてだったので、今後において非常に有益な経験ができたと感じている。

活動にあたってプロジェクト内で「AI カー」「動画要約」「音楽生成」の 3 つのグループに分かれ、私は「音楽生成」グループに所属した。さらに本グループ内で「music2music」「text2music」の 2 つに班分けを行い、私は music2music 班に所属し、成果物開発に向けて活動した。後期以降の主な開発にて敵対的生成ネットワークを用いて指定ジャンルの音楽の自動生成をするにあたり、私は指定のジャンルの音楽データを集める作業を担当した。膨大な数のデータを必要とするため、ジャンルを絞り、データの長さ等に注意しながら開発に必要なデータの収集を行った。当初は「ロック」「ジャズ」等、比較的知名度が高く聴衆が認識しやすいジャンルを指定しデータ収集を行っていたが、生成した音楽にノイズが目立つといった問題が発生することが発覚したため、生成する音楽のジャンルから見直し、低音が強くリズムがはっきりとしている「ハウスミュージック」から再度データ収集を行った。それ以降は成果発表会で使用するスライドの作成に加わった。

中間及び成果発表について、該当分野の知識を持ち合わせない聴講者が理解できる発表をするにはどうすればいいのかを考え、発表内容を練った。また使用する手法やモデルの背景や現状の問題点、その改善点を具体的且簡潔に示せるように工夫した。目標に対してどれほど達成できたか聴衆が視覚的に理解できるように図を作った。実際の発表では、台本は基本的に用意せずに発表を行い、できるだけ聴衆の表情を確認しながら発表を行うように意識をしたが、課題点は多々見つかったため、今後の卒業研究に生かしていきたいと考えている。

本プロジェクト学習によって、グループワークをする上での情報共有や役割分担の重要性を強く実感することができた。また、成果を発表する上での反省点も多く見つけることができたので、今後の他のグループ活動においても役に立つと考えている。

（※文責: 工藤大）

7.7　太田怜志

私はプロジェクト学習を行うにあたり、先輩方からお話を伺った。そこでは、プロジェクト学習の大変さ、特にリーダーはやるべきではないという意見を聞いた。しかし、様々な理由から、プロジェクトリーダーとグループリーダーを兼任することを決めた。私は、プロジェクトリーダーの業務は全体の方針の決定、プロジェクトで行う業務の責任を負うことであると考える。また、グループリーダーとしての業務は、グループの方針の決定、グループで行う業務の責任を負うことであると考える。これらのプロジェクトリーダー、グループリーダーのどちらの活動も求心力を必要とする。しかし、私には求心力がなく、私の希望方針をメンバーに反対されることも多かった。人は論理ではなく、感情で動くということを痛感した。

私はグループ A に所属し、AI を用いて、音楽のジャンル変換の実現を試みた。私はモデルの作成、改善を担当した。モデルの作成にあたり、私は膨大な量の論文を読んだ。ここでは、日々の研究活動で培ったスキルが役に立った。また、中間発表会や最終発表会での発表のため、作成したモデルについて、メンバーに説明した。開発したモデルは、様々な論文の内容を組み合わせており、メンバーには大変負担がかかったと反省している。プロジェクトの過程で、メンバーの同意や理解を得る過程を可能な限り減らしたが、この作業は避けられないものだった。メンバーの努力もあり、中間発表会や最終発表会では質疑応答の対応については、高い評価を得ることができた。学習データ数や時間などが足りない中でもなんとか最終発表会に間に合わせることができた。開発したモデルを用いて生成した音楽は、ノイズを含んでいたものの、リズムを刻むことができていた。私はプロジェクトを開始した段階ではもっと高いクオリティのモデルを開発できると考えていた。しかし、モデルの開発以外の雑務が多く、成果に納得できずに 1 年間のプロジェクト学習が終了した。プロジェクト活動全体としては、プロジェクト運営や管理、成果物の制作も何一つとしてうまく行くことはなかった。

プロジェクト学習を通じて私はリーダーに向いていないことを痛感した。プロジェクト学習では、プロジェクトリーダー、グループリーダーの裁量が小さく、解決できない問題が多い。このことは、プロジェクト学習を始める前からわかっていたことだが、想像を絶するほど苦痛であった。また、私がプロジェクトを通じて得たスキルは何もない。私にとってプロジェクト学習とは、学習ではなく、今まで培った技術を浪費する仕事だった。今後は、可能な限りプロジェクトを伴う活動を避けて生きていきたい。

密かに、私はメンバーの成長を個人的な目的としていた。本プロジェクトのメンバーが活動を通じて少しでも成長できていれば、幸いである。

（※文責: 太田怜志）

7.8　中村允洸

まず、私が前期の期間で主に取り組んだことを振り返える。

1 つ目は、どういったアプローチで音楽を生成するかを決めることである。また、それを決めるにあたって、それぞれニューラルネットワークについて調べ、それを班のメンバーに共有するとい

うことを行った。私は DNN について調査することにした。その理由は、ニューラルネットワーク構造の中でもっともシンプルで、他のニューラルネットワークを学習する際に身につけた知識が生きそうだからだ。私は書籍や web サイトを調査し、その内容を簡潔にまとめ、班のメンバーに共有した。その際の班のメンバーの反応はおおむね好意的なものだったので、いい発表が出来たと考える。班のメンバーと話し合った結果、CycleGAN にレザバー計算を導入して音楽のジャンル変換を目指すということになった。私が調査・共有した DNN は音楽生成に採用されなかったが、ニューラルネットワークに関する知識を着けたことは決して無駄ではないと考える。

2 つ目は、中間発表のために発表スライドを作成することである。本グループの方針としては、発表スライドの制作に一か月くらい時間をかけ、しっかりとしたものを作っておくということになった。その理由は、中間発表の時点で完成度の高いスライドを作っておくことで、忙しいことが予想される最終発表の時、スライド作成に時間を割かれないようにするためだ。このスライド制作で、私が DNN を調査・共有した経験が生きた。発表スライドとは、出来るだけ多くの内容を聴衆に分かりやすく伝えるために、情報を過不足なく与えてやらなければならない。その情報を取捨選択する過程は、私が DNN について共有するために情報を吟味したことに酷似していた。したがって、スライド作成の作業は円滑に進めることが出来た。また、グループ内で様々な意見交換をしながら作成したことも、グループに有益な貢献をすることが出来たことの要因の一つである。

3 つ目は、夏休み期間中の活動についてである。夏休み期間中の課題として、全員が github をある程度使えるようにしようという意見が上がった。私は Sourcetree というアプリを用いてのローカルリポジトリ操作には慣れていた。しかし、チーム開発の経験は無く、github のようなリモートリポジトリの基本的な機能についての知識すら無かった。よって、夏休み期間中に github を使いこなすため、相応の時間を割くことにした。その際、関連する web サイトに書いてある内容を hands on 形式で学ぶことで、学習の効率化を図った。結果的に、チーム開発に必要な github の機能に関しては問題なく扱えるようになった。

次に、私が後期の期間で主に取り組んだことを振り返る。

まず最初は、私が任された役割と、役割を果たすために取り組んだことだ。私は、GAN とレザバー計算の知識をしっかり身につけて、成果発表会の質問対応をこなすという役割に志願した。その理由としては、GAN やレザバー計算を少し調べる中で、この 2 つに対して興味を持ち、詳しく調べてみたいと考えたからだ。また、興味を持ったことに対しては、高いモチベーションをもって知識を身につける作業に取り組むことが出来るので、班に貢献しやすいと考えたのも理由の一つだ。私は私の所属する班で使う技術である GAN,WaveGAN, レザバー計算に関する論文や書籍を読み、得た知識を要約したものを Notion にまとめてして、効率的な知識の習得に努めた。その結果、私の所属する班が作成したモデルに関する知識をほぼ全て習得出来た。

２つ目は、成果発表である。成果発表では、プレゼンテーションに加えて質疑応答を行う役割というのはとても重要だ。私は任された役割を全うするため、入念な準備に取り組んだ。１つ目は、発表練習である。グループ内で発表練習を何度もして、メンバーにフィードバックをもらった。そのフィードバックをもとにプレゼンテーションを改善していき、本番で最高の発表が出来るように努めた。２つ目は、質疑応答の対策である。成果発表で投げかけられる質問を想定しておき、それに対する回答をあらかじめ用意する。そうすることにより、成果発表で落ち着いた質問対応を行えるようにした。これらの準備の結果、成果発表では満足のいくプレゼンテーションを行うことが出来た。また、質疑応答でも、過不足なく分かりやすい説明をすることが出来たと考える。

（※文責: 中村允洸）

第 8 章　活動内容の詳細

8.1　text2music 班の初期活動

文章から音楽を生成するグループでは、勉強会を中心に行った。グループのメンバーで使いたいモデルを話し合ったところ、スパイキングニューラルネットワーク、レザバー計算が候補として挙がった。そこで、Stable Diffusion にスパイキングニューラルネットワーク、レザバー計算を取り入れることで学習コストの削減を目指すという方針にし、それぞれの分野の担当を決めて、各々で本や論文を読み、グループ間で知識を共有した。これらのことを行った理由は、すべてのメンバーが同じものを学ぶより、一人が一つのことを調べ、理解したものをかみ砕いて他のメンバーに説明したほうが効率的だと考えたからである。これにより勉強時間を削減し、多くの知識を得ることができた。勉強会をしているうちに、FSVAE と Tokenshift という技術を取り入れることで、学習コストの削減が図れるだろうという考えになり導入する方針になった。その後、後期の活動の大まかな計画を立てた。計画としては、

1．スペクトログラム用に FSVAE を改変
2．StableDiffusion のソースコードを解読し、FSVAE を組み込む
3．U-NET 部分のソースコードを解読し、Tokenshift を組み込む

といったものだった。レザバー計算は具体的な導入方法が決まっていなかったため、時間に余裕ができ次第、手を付けることにした。

8.2　中間発表会

中間発表会の準備はスライド作成から行った。メンバー各自でスライドのテンプレを作成し、持ち寄って良いものを採用した。スライド作成後、発表練習を始めた。始めは、グループ内で発表練習を行い、改善点や想定される質問を話し合った。その後、プロジェクト全体で発表練習、質問に答える練習を行った。さらに、プロジェクトの先生方に発表練習を見てもらい、改善すべき点をお教えいただいた。お教えいただいた改善点としては、「スライドの前後関係がない」、「これからやることを分かりやすく示す」、「専門用語が多すぎる」といったものだった。発表練習で挙げられた改善点を修正し、中間発表会に臨んだ。中間発表会の評価アンケートにて、「専門用語についての説明がもっと欲しい」という声が多かったため、成果発表会に向けた改善点とした。

8.3　text2music 班の夏休み以降の活動

8.3.1　データセットの作成

著作権がない音楽とテキストがセットになっているデータセットを探し,MusicCaps を用いることにした。MusicCaps は、約 5000 のデータからなるものであり, 音楽の YouTube の動画の情報

と、その YouTube の動画の音声を文章で説明したものがセットになっているものである。その後、データセットの作成を行った。MusicCaps の Youtube の情報をもとに 20 秒の音声を抽出し、その音声をスペクトログラムに変換した。そして、スペクトログラムとその音声についての説明がセットになっているデータセットを作成し、HuggingFace Datasets で管理した。データセット作成後、4 つの改変を行った。

1 つ目は、スペクトログラムの画像をカラー画像からグレースケールに変更して作り直したことである。これにより、データサイズを小さくすることができ、学習コストの削減につながると考えた。

2 つ目は、データの量を増やしたことである。既存の生成モデルは学習に大量のデータを要するという課題があるため、データの量が足りない可能性があるという考えになり、20 秒の音声データを 4 秒の音声データで 5 つに切り取り、データの量を増やした。

3 つ目は、輝度の正規化だ。スペクトログラムの輝度のばらつきがあり、白い部分が多いものと黒い部分が多いものが極端に分かれていた。そこで、スペクトログラムの白と黒の割合を同程度に調整すればノイズが生成されなくなるのではないかと考え、画像のヒストグラム (輝度分布) を平坦化することで、輝度を正規化した。

4 つ目は、音楽のジャンルを統一したことだ。音楽のジャンルを統一する必要があると感じた理由は、MusicCaps には少し特殊な音声が含まれていたからである。例を挙げると、指パッチンしながら歌を歌うという音声が含まれていた。このような音声のデータが、音楽の生成の妨げになっている可能性があると考えた。そのため、distilhubert-finetuned-gtzan という音声の分類ができるモデルを使い、クラシック音楽のみのデータセットを作成した。

しかし、この 4 点を改変しても音楽生成はできなかった。そこで、班のメンバーと話し合った結果、入力データを画像から音声波形に変えるというアプローチと、FORCE 学習という技術を取り入れるというアプローチの 2 つにわかれて作業を行った。そのため、データセットをスペクトログラムから音声波形に変える作業を行った。扱うデータの種類が音声動画、音声波形、スペクトログラムと複数あったため、異なる種類のデータに変換する作業をスムーズに行えるように、PyPi を用いてパッケージ化した。パッケージ化したことにより、音声波形のデータセットはスムーズに作成できた。音声波形のデータセットでの学習によって、訓練データからの音楽生成は成功した。

8.3.2 FSVAE の開発

データセット作成後、データセットの読み込みメソッドを作成し、FSVAE で学習させてみたところ、Loss の値は訓練データを用いた場合もテストデータを用いた場合も下げることに成功した。しかし、FSVAE を通したスペクトログラムを音声に変換すると音楽ではなくノイズが生成された。スペクトログラムのデータセットに改変を加えた後に、代理勾配の改善を行い、再び FSVAE で学習させたが、結果は同じくノイズが生成された。そこで, 問題点が 2 つ挙げられた。

1 つ目の問題は、FSVAE を通すとスペクトログラムがぼやけるということだ。FSVAE を通したスペクトログラムは、元画像と比べて、時間軸の境目があいまいになっており、リズムの学習が上手くできていないのではないかと考えた。

2 つ目の問題は、スペクトログラムの画像を FSVAE に通すと、FSVAE は同じデータを複製するという工程があるため、スペクトログラムの画像に適用させると、メモリの消費量が大きくなってしまうということだ。

そこで入力データをスペクトログラムから音声波形に変更した。それに伴い、音声波形に対応した FSVAE（以下 WaveFSVAE)、音声波形のデータセット読み込みメソッド、WaveFSVAE のトレーニングスクリプトの作成を行った。入力データを音声波形に変更したことで、メモリの消費を抑えて学習させることに成功した。また、勾配累積や学習率スケジューラを用いることで、さらに目メモリの消費量を削減することができた。勾配累積というのは、小さいバッチで計算した重みを保存しておき、複数回分ためてから平均を取り、それを用いてモデルのパラメータを更新するというものである。勾配累積を利用することで、小さなバッチサイズでも大きなバッチサイズと同様の安定性能を出すことができる。学習率スケジューラというのは、学習率を学習の途中で変化させることができるものである。これにより、Loss の極小値を効率良く探すことができる。訓練データを用いたときは、Loss の値を非常に小さくすることができた。テストデータを用いたときも,Loss の値は小さくできたが訓練データほど小さくはできなかった。Loss の値のログは wandb を用いて確認した。結果としては、訓練データから音楽を生成することには成功した。しかし、テストデータやランダムサンプリングによる音声生成では、音楽を生成することはできず、ノイズが生成されてしまった。

8.3.3 結果

当初の予定とは異なったが、音声波形を FSVAE に通すことで、学習コストの削減をすることができた。時間の関係で FSVAE を StableDiffusion に組み込むこと、Tokenshift を取り入れることはできなかった。また、レザバー計算の枠組みである FORCE 学習で、ぼやけたスペクトログラムに対してリズムを学習することで、ぼやけたスペクトログラムからでも音楽を生成できると考え、FORCE 学習に関する勉強も並行して行っていた。しかし、こちらは勉強に多くの時間を要してしまい、実装まで進めることができなかった。

8.4 成果発表会

成果発表会では、スライドの改変から行った。基本的には、中間発表会で使用したスライドを改変して成果発表会でも使用することにした。スライド作成後、中間発表会の時と同様、発表練習に移った。グループ内で発表練習を行った後、先生方に発表練習を見ていただき、改善点をお教えいただいた。お教えいただいた改善点としては、「専門用語の説明に時間を使いすぎているため、成果物を示すことに時間を使った方が良い」、「スライドの情報量とスライドを見せる時間が釣り合っていない」といったものだった。これらの点を改善し、成果発表会に臨んだ。

成果発表会の評価アンケートでは、「スライドが見やすい」、「成果物が音声のため分かりやすい」という声もあった。中間発表会の時に引き続き、「専門用語のが多く、説明が不十分」という声もあった。しかし、１つのグループで２つの班に分かれているため、発表時間が他のグループの半分しかないということと、専門用語の解説に時間を使いすぎると成果発表がメインでなくなってしまうためこの点の改善は難しかった。

8.4.1 利用したサービス・モデルの詳細

GitHub

今回の開発では GitHub でソースコードの共有を行った。また、GitHub 上にアップされている既存の StableDiffusion のソースコードを参考にして開発を行った。GitHub は、ユーザのみなさんからヒントを得て作成された開発プラットフォームである。オープンソースプロジェクトやビジネスユースまで、GitHub 上にソースコードをホスティングすることで、他の開発者と一緒にコードのレビューを行ったり、プロジェクトの管理をしながら、ソフトウェアの開発を行うことができる。

HuggingFace

Hugging Face は、人工知能（AI）のモデルやデータを共有し、利用するためのオープンソースプラットフォームである。今回は、HuggingFace の Datasets というライブラリを利用した。Datasets というライブラリは、大規模なデータセットの処理と操作を効率的に行うためのツールである。データセットの読み込み、変換、フィルタリングなど、一般的な前処理タスクを簡単に行えるように設計されている。また、Datasets ライブラリは自分で作成したカスタムデータセットの読み込みと使用もサポートしている。

distilhubert-finetuned-gtzan

distilhubert-finetuned-gtzan は DistilHuBERT を微調整したものである。DistilHuBERT は、音声表現を学習し、多数の音声処理タスクを行うことができるものである。事前訓練のためのラベルなし音声データを利用して、多数の音声処理タスクのための良い表現を提供する Hidden-unit BERT(HuBERT) は、大きな記憶と高い予訓練コストを必要とした。DistilHuBERT は、HuBERT モデルから隠れ表現を蒸留する新しいマルチタスク学習フレームワークであり、訓練時間とデータがほとんど不要となった。

PyPi（Python Package Index）

PyPi は、Python を利用する開発者向けに、ソフトウェアの検索・インストール、コミュニティを提供している。ソフトウェアの開発に必要なパッケージ（ソースコードファイル群）をダウンロードし、活用することができる。今回は、PyPi 上に wav ファイルとスペクトログラムの相互変換、wav ファイルと音声波形の相互変換、スペクトログラムから音声波形への変換をパッケージ化した。

wandb（Weights Biases）

wandb は、機械学習のプラットフォームである。wandb では、モデルの学習記録、モデル・データセットのバージョン管理、モデルの性能評価、可視化ができる。今回は、スペクトログラム用の FSVAE、WaveFSVAE それぞれにおいて利用し、Loss の値の変化をグラフとして表示し、確認した。また、wandb 上にスペクトログラムの画像やそのスペクトログラムによって出力された音声を出力するようにした。それにより、Loss の値と出力結果を wandb のみで確認できるようになり、作業が効率化した。

（※文責: 山谷璃輝）

8.5　music2music 班の初期活動

music2music 班はまず初めに、方針を決めるため、勉強会を中心に行った。メンバーそれぞれが音楽のジャンル変換の実現に必要だと考えるモデルを調査し、共有した。その後、班のメンバー同士で議論を行い、CycleGAN を用いることを決定した。決定理由は、CycleGAN は先行研究が豊富かつ、学習コストの削減の可能性が見込まれたからである。

次に、我々はそれぞれの役割を分担した。班の 4 名をそれぞれデータ収集、データ加工、モデル開発、発表会担当者に分けた。以下の手順で音楽のジャンル変換の実現を試みた。初めに、データ収集担当者が収集したデータをデータ加工担当者が加工する。加工されたデータを用いて、モデル開発者がモデルの開発を行う。しかし、モデル開発者が 1 人だけでは、中間や最終発表会の質疑応答に対応できない可能祭がある。そこで、発表会担当者は、モデル開発の担当者からモデルの概要を学び、中間や最終発表会の質疑応答に備えた。

そして、夏休み中に行う活動についても話し合いを行った。夏休み中に行う活動として、GitHub の取り扱いを学ぶことが決まった。理由は、夏休み中は班のメンバーのほとんどが、インターンに参加する必要があり、並行してできる活動が絞られたからである。GitHub の取り扱いを夏休み中に行うことで、夏休み以降のモデルの開発、情報共有の効率化を図った。

（※文責: 太田怜志）

8.6　中間発表会

中間発表会の準備はスライド作成から行った。メンバー各自でスライドのテンプレートを作成し、持ち寄った。その後、使用するテンプレートを決定した。また、スライドの作成、共有には Canva というサービスを利用した。スライド作成後、発表練習を始めた。始めは、グループ内で発表練習を行い、想定問答集を作成した。次に、プロジェクト全体で発表練習し、スライドの改善を行った。最後に、プロジェクトの指導教員に発表練習を見てもらい、最終調整を行った。

（※文責: 太田怜志）

8.7　music2music 班の夏休み以降の活動

後期は本格的にモデルの開発に取り組むため、以下の方針を決定した。初めに、GAN の開発を行う。その後、GAN による音楽の生成が成功したら、レザバー計算を導入した GAN を開発する。レザバー計算を導入した GAN による音楽の生成が成功したら、レザバー計算を導入した CycleGAN を開発する。また、モデルの開発には、OS として Ubuntu 22.04.1、CPU として Intel Core i9-9900K、GPU として GeForce RTX 2080 SUPER を用いた。機械学習のフレームワークとして、PyTorch を用いた。PyTorch は、画像認識や自然言語処理などの多くの分野で活用されており、参照できる情報が多い。music2music 班では音楽のジャンル変換 AI の開発を行うため、少しでも参照できる情報は多いほうが良いため採用した。

（※文責: 太田怜志）

8.7.1 GAN の開発

はじめに、GAN を用いた音楽生成に取り組んだ。GAN は Generator と Discriminator の二つのネットワークから構成されるディープニューラルネットワークの一種である。Generator はデータを生成し、Discriminator はそのデータが本物か偽物かを判別する。Generator は Discriminator を欺こうと学習し、Discriminator はより正確に識別しようと学習する。これにより、GAN は本物に近いデータを生成する能力を向上させることが期待される。しかし、モデルが生成した音楽は高音が鳴るだけで、リズムやメロディーが生成できていなかった。原因を特定するため、生成した音楽のスペクトログラムを見てみることにした。スペクトログラムを見てみると、周期的なパターンが出現していることがわかった。調査を進めると Discriminator が意図しない学習を行っていることがわかった。

Generator と Discriminator には畳み込みニューラルネットワークというディープラーニングのアルゴリズムが使用されている。畳み込みニューラルネットワークは、入力層、畳み込み層、プーリング層からなる。畳み込み層の役割は局所的な特徴量の抽出で、プーリング層の役割は移動不変性の付与である。畳み込み層では、カーネルというフィルタのような重み行列を用いて、特徴量を抽出する。音声において、特定の位相で畳み込み層のカーネルが複合的に作用してしまい、Discriminator が意図しない学習をしてしまう。

（※文責: 太田怜志）

8.7.2 WaveGAN の開発

次に、Discriminator が意図しない学習を行う問題を解決するために、WaveGAN を導入した。WaveGAN は GAN の Discriminator に Phase Shuffle という操作を行う層を導入したものである。この層は Discriminator の、各畳み込み層の直後にそれぞれ配置される。Phase Shuffle を行う層では、入力された特徴量に対し、要素をずらす操作を実行する。Phase Shuffle ををを行うことで、音声のリズムやジャンルなどの特徴は変わらないが、畳み込み層のカーネルの複合的な作用を避けることができる。Phase Shuffle を行うことで、Discriminator の意図しない学習を防ぐことができる。生成した音楽に音の乱れはあったが、リズムやメロディーを生成することができた。

（※文責: 太田怜志）

8.7.3 レザバー計算を導入した WaveGAN の開発

WaveGAN の学習コスト削減を実現するため、Generator にレザバー計算を導入する。レザバー計算は時系列の学習が可能なリカレントニューラルネットワークの枠組みである。レザバー計算は、入力層、レザバー層、出力層から構成される。レザバー計算におけるネットワークの学習は、レザバー層から出力層への結合に限定しており、学習コストが少ない。つまり、レザバー計算は、ネットワークを最適化するのではなく、ネットワークの非線形なダイナミクスを利用している。今回は、レザバー計算の代表的なモデルであるエコーステートネットワーク（以下「ESN」とよぶ）

を用いた。生成した音楽はメロディーは不十分だったが、リズムは生成できていた。メロディーが不十分だった理由として、十分にモデルのパラメータを探索することができなかったことが挙げられる。しかし、1 エポックあたりの学習時間を約 20 秒短くすることができ、学習コストの削減に成功した。

（※文責: 太田怜志）

8.7.4 CycleGAN を用いた音楽のジャンル変換

最後に、CycleGAN を用いた音楽のジャンル変換の実現を試みた。CycleGAN とは 2 つの GAN から構成され、ソースドメインからターゲットドメインへの変換を行う手法である。例えば、CycleGAN を用いると、馬の写真をシマウマの写真にしたり、シマウマの画像を馬の画像に変換することができる。CycleGAN の学習では、データセットを 2 つ用意する必要がある。これらのデータセットをそれぞれ、データセット X, Y とする。データセット X を Y に変換するように学習を行った GAN とデータセット Y を X に変換するように学習を行った GAN を組み合わせる。この学習を行うことで、画像のドメインを相互変換する事が可能となる。music2music 班では、レザバー計算を導入した WaveGAN を 2 つ用いて CyclGAN の開発を試みた。しかし、レザバー計算を導入した WaveGAN を開発した時点で、成果発表会まで残り 2 週間を切っており、CycleGAN は学習途中で終わってしまった。前述したとおり、CycleGAN では 2 つのデータセットが必要である。レザバー計算を導入した WaveGAN の学習時間は短いとはいえ、CycleGAN の学習時間は少なくない。結果としては、CyclaGAN の開発には至らなかった。

（※文責: 太田怜志）

8.8 成果発表会

成果発表会のスライドは、中間発表会で使用したスライドに、後期の活動内容を加えて作成した。スライド作成後、中間発表会の時と同様の手順で発表練習を行った。グループ内での発表練習では中間発表会のときと同様に、想定問答集の作成をした。プロジェクト内での発表練習ではすらいどの改善の他に、発表時間についての話し合いを行った。本グループでは、どちらの班も音楽を流す必要があったため、他のグループの発表の邪魔をしないよう、話し合いをした。プロジェクトの指導教員に発表練習を見てもらったところ、成果物をもっと強調して発表するべきとの助言を頂いた。それを受けて、成果発表会に使う音楽を 1 曲から 3 曲に増やした。成果発表会後のアンケートでは、発表の仕方に関する指摘が多く見られた。

（※文責: 太田怜志）

8.9 結果

当初の目的である、CycleGAN の学習コスト削減を実現することはできなかった。しかし、その過程で音楽生成の学習コスト削減には成功できた。中間発表会や成果発表会では好意的な意見も多く、十分な成果を得ることができた。また、メンバー全員が様々な技術や知識を習得し、GitHub

Make Brain Project

の取り扱いを学び、効率よくプロジェクト管理を行うことができた。

（※文責: 太田怜志）

参考文献

[1] *stablediffusion*. https://github.com/Stability-AI/stablediffusion. (Accessed on 17/01/2024). Nov. 2022.

[2] Tatsuya Harada Hiromichi Kamata Yusuke Mukuta. *Fully Spiking Variational Autoencoder*. https://arxiv.org/pdf/2110.00375.pdf. (Accessed on 17/01/2024). Dec. 2021.

[3] Jun-Yan Zhu et al. *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*. https://arxiv.org/pdf/1703.10593.pdf. (Accessed on 17/01/2024). Mar. 2017.

[4] Georg Holzmann. *RESERVOIR COMPUTING: A POWERFUL BLACK-BOX FRAMEWORK FOR NONLINEAR AUDIO PROCESSING*. Proc. of the 12th Int. Conference on Digital Audio Effects (DAFx-09). (Accessed on 17/01/2024). Sept. 2009.

[5] Ian J. Goodfellow et al. *Generative Adversarial Networks*. https://arxiv.org/pdf/1406.2661.pdf. (Accessed on 17/01/2024). June 2014.

[6] Chris Donahue, Julian McAuley, and Miller Puckette. *Adversarial Audio Synthesis*. https://api.semanticscholar.org/CorpusID:52890982. (Accessed on 01/10/2024). 2018.

[7] Dominik Lorenz Patrick Esser Bjorn Ommer Robin Rombach Andreas Blattmann. *High-Resolution Image Synthesis with Latent Diffusion Models*. https://arxiv.org/pdf/2112.10752.pdf. (Accessed on 17/01/2024). Apr. 2022.

[8] Umberto Michelucci. *An Introduction to Autoencoders*. https://arxiv.org/pdf/2201.03898.pdf. (Accessed on 17/01/2024). Jan. 2022.

[9] Thomas Brox Olaf Ronneberger Philipp Fischer. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. https://arxiv.org/pdf/1505.04597.pdf. (Accessed on 17/01/2024). May 2015.

[10] Chong-Wah Ngo Hao Zhang Yanbin Hao. *Token Shift Transformer for Video Classification*. https://arxiv.org/pdf/2108.02432.pdf. (Accessed on 17/01/2024). Aug. 2021.

[11] Adam Roberts et al. "A hierarchical latent vector model for learning long-term structure in music". In: *International conference on machine learning*. PMLR. 2018, pp. 4364–4373.