

Rapport (sprint 1) du projet de 8INF892

Wenhao LUO
LUOW09129805

March 24, 2023

1 Introduction

Dans le cadre du cours 8INF892, j'ai besoin de réaliser un projet en utilisant les outils de l'apprentissage profond. Je choisis de créer un réseau neurone qui permet de produire de la musique en imitant le style de Bach. Le projet sera réalisé en utilisant `Python`, avec l'aide de la bibliothèque `TensorFlow`.

Dans le premier sprint, je commence par chercher l'algorithme que j'ai besoin d'utiliser ainsi que les outils dont j'ai besoin. Dans les deux sections suivantes, je vais présenter les recherches que j'ai réalisées.

Le dépôt du projet est disponible sur Github : <https://github.com/LuoQuestionmark/projet-apprentissage-profond>.

2 Algorithme recherché

L'inspiration de ce projet est un "Google Doodle" il y a quelques années¹. Il y a un article qui explique en détails de ce projet[2]. Le réseau neurone est entraîné pour "compléter" la partition musicale : on cache certaines informations de la partition, et le réseau doit retrouver ces informations ; c'est donc un apprentissage supervisé.

Une fois ce réseau est réalisé, un morceau de la musique peut être créé en utilisant le réseau plusieurs fois. Chaque fois des nouvelles notes seront ajoutées.

3 Dataset

On peut trouver un dataset sur l'internet[3]. Le format `midi` est un format connu. Il y a donc une bibliothèque disponible avec le langage `Python`[1]. Un exemple de la lecture d'un fichier `midi` avec ce module est montré dans le listing 1.

Le plan du sprint 2 est analyser la sortie de ce module et encoder cette sortie sous le format compréhensible par le réseau neurone.

4 Conclusion

Dans le premier sprint, j'ai exploré l'algorithme à utiliser et les bibliothèques utiles. J'ai fini donc la préparation pour la création du réseau neurone de ce projet.

¹<https://www.google.com/doodles/celebrating-johann-sebastian-bach>

```

1 for i, track in enumerate(mid.tracks):
2     print('Track {}: {}'.format(i, track.name))
3     for msg in track:
4         print(msg)
5
6 # output:
7 # note_on channel=1 note=60 velocity=64 time=0
8 # note_off channel=1 note=60 velocity=51 time=60
9 # note_on channel=1 note=62 velocity=64 time=0
10 # note_off channel=1 note=62 velocity=49 time=60
11 # note_on channel=1 note=60 velocity=64 time=0
12 # note_off channel=1 note=60 velocity=55 time=60
13 # note_on channel=1 note=59 velocity=64 time=0
14 # note_off channel=1 note=59 velocity=59 time=60
15 # note_on channel=1 note=57 velocity=64 time=0

```

Listing 1: Un exemple du programme qui permet de lire le fichier midi.

References

- [1] Mido - midi objects for python. <https://mido.readthedocs.io/en/latest/index.html>.
- [2] Cheng-Zhi Anna Huang, Tim Cooijmans, Adam Roberts, Aaron Courville, and Douglas Eck. Counterpoint by convolution. In *International Society for Music Information Retrieval (ISMIR)*, 2017.
- [3] Wolfgang Schmieder. A john sebastian bach midi page. <https://www.bachcentral.com/midiindexcomplete.html>. Accessed: 2023-03-24.