Mémoire Master 1 MIAGE Apprentissage

Université Paris 1 Panthéon-Sorbonne

La génération de musique par deep learning

Pouvons-nous reproduire la créativité des artistes grâce à l'Intelligence artificielle ?

Nelly UNG

Encadrée par : Mme. Irina Rychkova

Maître d'apprentissage : M. Guillaume OSTER

Septembre 2021

Remerciement

Je voudrais dans un premier temps remercier ma tutrice Madame RYCHKOVA Irina pour m’avoir guidée et conseillée tout au long de l’élaboration de ce mémoire. Elle m’a fait confiance et a tout fait pour que cette expérience soit instructive, en partageant son expertise avec moi. Elle a été très disponible et pédagogue, et je tiens à l'en remercier.

Je remercie également toute l’équipe pédagogique de l’université Paris 1 Panthéon Sorbonne et les intervenants professionnels pour m’avoir apporté toutes les connaissances nécessaires à la réalisation de ce mémoire.

Je voudrais enfin remercier l’équipe MOA de BNP Paribas GSS au sein de laquelle j’effectue mon alternance et plus particulièrement mon tuteur Monsieur OSTER Guillaume pour m’avoir soutenu.

Résumé

Les modèles de Deep Learning ont montré des résultats très prometteurs dans la composition automatique de morceaux de musique polyphonique. Cependant, il est très difficile de contrôler de tels modèles pour guider les compositions vers un objectif souhaité.

Cet état de l'art vise à explorer l'utilisation de réseaux de neurones profonds pour la génération de musiques. Avec l'arrivée récente des approches par réseaux de neurones profonds et leur capacité à fournir des représentations de haut niveau, plusieurs travaux ont été menés pour tenter de les utiliser.

Mots-clés : Apprentissage automatique, Génération de musique, Deep learning, LSTM, RNN

Introduction

“There are not more than five musical notes, yet the combinations of these five give rise to more melodies than can ever be heard.”

Sun Tzu

La musique fait partie de notre quotidien et est utilisée depuis des siècles partout dans le monde. Il existe de nombreux types de musiques (folklorique, blues, soul, rap, rock etc …) qui n’ont cessé d’évoluer et d’être une source d’inspiration pour l’Homme.

L’évolution des technologies informatiques et d’internet a fait émerger de nouvelles façons de faire de la musique au point de créer des œuvres douées d’une véritable valeur esthétique. En effet, un ordinateur peut maintenant composer une musique de manière totalement autonome. De plus en plus d’artistes explorent ces nouvelles technologies afin de créer de nouvelles musiques.

De ce fait, le deep learning (apprentissage profond) et l'intelligence artificielle révolutionnent lentement de nombreux domaines d'application, ils ont le potentiel de remplacer les humains dans une variété de tâches et d'emplois. Leur montée en puissance annonce une émergence d’un nouveau type de musique dans le monde et à mesure que des ensembles de données musicales à plus grande échelle sont mis à disposition, le système de génération basé sur l'apprentissage automatique pourra automatiquement apprendre un style musical à partir d'échantillons et en générer de nouveaux. Il est donc intéressant de s’intéresser à la question suivante :

Pouvons-nous reproduire la créativité des artistes grâce à l'intelligence artificielle ?

Cet état de l’art se concentre sur les progrès récents de l'apprentissage profond appliqué à la génération de contenus musicaux. Si ces modèles sont capables de produire des résultats qui pourraient être considérés comme de la musique, le rôle du musicien humain reste toujours prépondérant dans la production d'une pièce musicale.

Après avoir présenté ce que le Deep learning peut apporter dans l'univers musical, nous étudierons comment il peut aider à la mise en place d'une nouvelle forme de musique. Pour cela, nous verrons plus en détail quelques-unes des techniques utilisées et nous porterons un regard critique sur l'utilisation de de l'intelligence artificielle dans le domaine musical. Enfin, nous conclurons sur un bilan et les perspectives d'évolution.

I. Impact du Deep Learning dans le secteur musical

A. Définition

L'apprentissage profond (« deep learning ») est un domaine récent en pleine expansion et est utilisé dans la plupart des cas pour des prédictions ou encore de la classification. C'est le cas de la reconnaissance d'image ou vocale.

C'est en 2012 [8] que le deep learning connaît une forte croissance, notamment grâce à plusieurs facteurs : la disponibilité de plus en plus de données "big data", d'ordinateurs efficaces et puissants, et enfin des avancées technologiques importantes. L'apprentissage profond est basé sur des « réseaux de neurones artificiels », composés de milliers d'unités (les « neurones ») qui effectuent chacune de petites opérations simples [6]. Les résultats d'une première couche de neurones » servent d'entrée aux calculs d'une deuxième couche et ainsi de suite.

Si le domaine est récent, il n'en demeure pas moins qu'il a déjà produit un ensemble de théories, de techniques, de mécanismes et de systèmes qui ont fait leurs preuves. C'est une discipline incontournable qui s'utilise dans des domaines de recherche de plus en plus variés comme le traitement du signal audio.

De plus, les techniques récentes de réseaux de neurones profonds [1] ont relancé la tendance avec une avancée significative : la capacité à apprendre automatiquement des concepts de haut niveau, proches du langage naturel.

Dans le domaine de la musique, le processus de composition musicale est divisé en plusieurs tâches telles que la mélodie, l'accompagnement ou encore le rythme. La figure ci-dessous regroupe les algorithmes d'intelligence artificielle les plus populaires (à gauche) utilisés pour automatiser les tâches de composition musicale (à droite). Nous nous intéresserons plus particulièrement aux techniques de “deep neural network”.

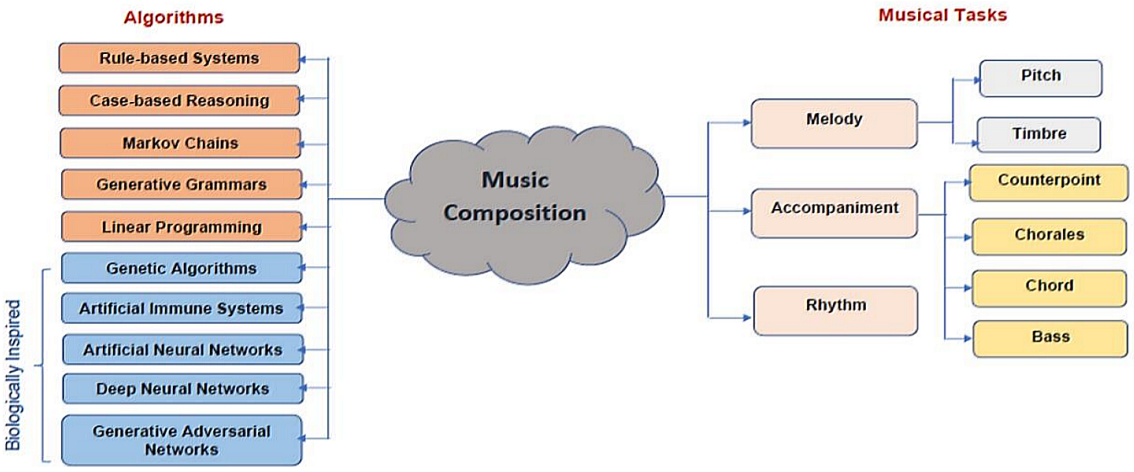


Figure. 1: Music Composition Tasks and Algorithms [3]

B. Motivation

À l’heure du big data et de l’avènement des services de streaming, la génération de musiques est devenue une nécessité. Le contenu musical entre en concurrence avec les autres acteurs du secteur. De ce fait, la génération de musiques est très prisée. En effet, comme nous l’avons vu précédemment, elle est devenue un axe de recherche important, avec plus d’une centaine de publications depuis les années 2000.

Le champ de recherche a émergé de la nécessité de gérer une quantité de données musicales digitales toujours plus grande et ces recherches ont permis d’effectuer des travaux d'analyse musicale. Cependant, ils doivent faire face à un défi majeur : l'accès à des données musicales en libre-service.

Malgré les plus de 150 millions de morceaux disponibles [19] sur les plateformes de streaming, seuls quelques milliers de morceaux sont utilisés pour la recherche scientifique. La mise en place d'une base de données musicale académique reste donc difficile.

L'objectif principal est de créer des œuvres imitant le plus possible le compositeur humain.

C. Emergence de deep learning pour la génération de musique

La construction de méthodes de calcul pour l'imitation de style musical a été beaucoup plus difficile qu'on ne l'imaginait initialement, les modèles utilisés se sont avérés incapables de générer des mélodies même simples et bien formées.

Un premier exemple populaire est le jeu de dés musical, où de petits fragments de musique sont réorganisés au hasard en lançant un dé pour créer une pièce musicale [11]. Depuis, les recherches sur la composition algorithmique se sont multipliées.

La première musique générée par ordinateur est apparue en 1957. Il s'agissait d'une mélodie de 17 secondes nommée «The Silver Scale» par son auteur Newman Guttman et générée par un logiciel de synthèse sonore nommé Music I, développé par Mathews aux Bell Laboratories [8].

Par la suite, de nombreuses avancées ont fait leurs preuves comme, «The Illiac Suite», première partition composée par ordinateur, ou encore « Ural-1 », le premier article sur la composition algorithmique de la musique à l'aide de l'ordinateur [10].
« The Illiac suite » utilise des modèles stochastiques dépendant des concepts statistiques (chaînes de Markov) [12] pour la génération ainsi que des règles de filtrages générées selon les propriétés souhaitées.

Le projet Magenta de Google[1] et le projet Flow Machine de Sony CSL[2] sont deux projets de recherche axés sur la génération musicale. Ils ont développé plusieurs outils pour aider les musiciens à être plus créatifs, par exemple. Melody Mixer, Beat

[1] https://magenta.tensorflow.org/
[2] http://www.flow-machines.com/

Blender, Flow Composer, etc. De plus, au cours des dernières années, plusieurs startups ont également utilisé l'IA pour créer de la musique.

Aujourd'hui on peut voir apparaître de plus en plus d'outils et logiciels dans le monde permettant de créer de la musique. Ces programmes ne remplacent pas les compositeurs, mais les aides de manière intelligente en fournissant des idées musicales.

II. Génération de musique par deep learning

A. Contenu musical et destination

Nous verrons dans cette partie les différents types de contenus musicaux et leurs traitements. Tout d'abord, il existe différents types de contenus musicaux [6] :

- La Mélodie: C'est une séquence de notes pour un seul instrument ou voix, avec au plus une note à la fois. Par exemple, la flûte.

- La polyphonie : C'est une séquence de notes pour un seul instrument, où plus d'une note peut être jouée en même temps. Par exemple une musique produite par un instrument polyphonique tel qu'un piano ou une guitare.

- Multipiste: Il s'agit d'un ensemble de voix / pistes multiples. Les exemples sont: une chorale avec des voix de soprano, alto, ténor et basse ou un trio de jazz avec piano, basse et batterie.

- Accompagnement : cela peut être plusieurs mélodies (voix); ou bien une progression d'accords, qui fournit une certaine harmonie.

Ces contenus musicaux peuvent avoir différentes finalités :

- Un système audio qui joue le contenu généré, comme dans le cas de la génération d'un fichier audio.
- Un logiciel séquenceur qui traitera le contenu généré, comme dans le cas de la génération d'un fichier MIDI.
- Ou encore une action humaine qui exécutera et interprétera le contenu généré, comme dans le cas de la génération d'une partition musicale.

Générer un contenu musical peut être fait de manière autonome et automatisé sans aucune intervention humaine; ou de manière interactive avec une interface de contrôle permettant aux utilisateurs humains d'avoir un contrôle interactif sur le processus de génération.

Concernant le style musical du contenu à générer, il sera régi par le choix de l'ensemble de données d'exemples musicaux qui seront utilisés. Le choix d'un jeu de données et des propriétés est donc fondamental pour une bonne génération de musiques.

B. Représentations des données musicales

Le choix de la représentation des données musicales peut être important pour la précision de l'apprentissage et pour la qualité du produit généré. En effet, il est nécessaire d'avoir une représentation des notes de musique pour ensuite les transformer en variables d'entrée d'un réseau de neurones.

Par exemple, dans le cas d'une représentation audio, il est possible d'utiliser une représentation spectrale au lieu d'une représentation de forme d'onde brute.

Avant d'entrer dans les choix de représentation des différentes données à traiter par une architecture de deep learning, il est important d'identifier les deux phases principales liées à l'activité d'une architecture de deep learning [5] :

La phase d'apprentissage implique la formation des données, elle est constituée d'un ensemble d'exemples pour entraîner le système de deep learning et de données de validation appelées données de test pour tester le système.

La phase de génération implique

- les données utilisées en entrée pour la génération. Par exemple, la mélodie, pour laquelle le système va générer un accompagnement, ou bien une note qui sera la première note générée de la mélodie.
- Les données générées (en sortie) correspondent aux données produites par la génération.

La représentation des données musicales peut prendre plusieurs formes :

Waveform (forme d'onde)

La première est le signal audio, que ce soit dans sa forme brute appelée waveform ou bien transformée. L'axe des abscisses x représente le temps et celui des ordonnées y représente l'amplitude du signal [24].

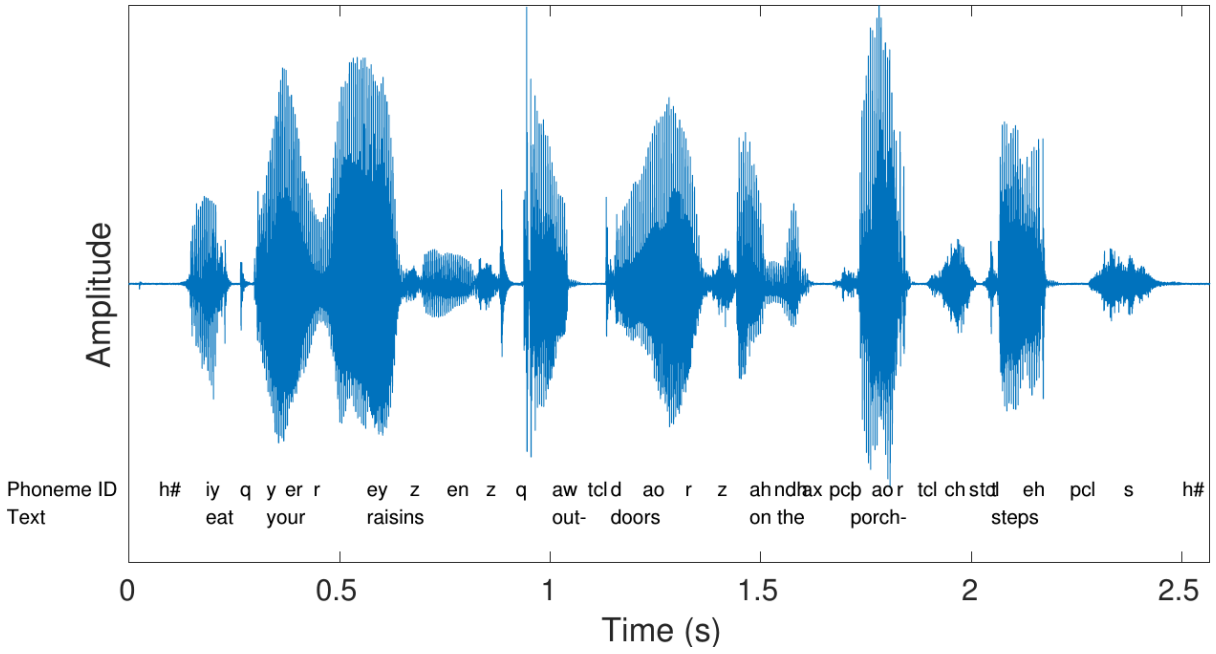


Figure 2 : Waveform[3]

L'avantage d'utiliser une forme d'onde (waveform) est de considérer la matière première non transformée, avec sa résolution initiale complète. L'inconvénient réside dans la charge de calcul, en effet le signal brut de bas niveau est exigeant en matière de mémoire et de traitement [6].

[3] wikipedia.org (https://wiki.aalto.fi/display/ITSP/Waveform)

Piano roll

L'approche la plus courante et utilisée est le format « piano roll ».[6]

La représentation au piano roll d'une mélodie (monophonique ou polyphonique) est inspirée des pianos automatisés. Il s'agissait d'un rouleau de papier continu avec des perforations (trous). Chaque perforation représente un élément d'information de contrôle de note, pour déclencher une note donnée. La longueur de la perforation correspond à la durée d'une note [22].

L'encodage du piano roll vise à faire correspondre à chaque segment de temps un vecteur ayant comme taille l'intervalle entre la note la plus basse et la note la plus haute avec une valeur égale à 1 pour l'élément correspondant à la note actuelle et une valeur nulle (0) pour tous les autres . (cette forme d'encodage a été inventée au départ pour l'électronique et se nomme « one hot ») [10].

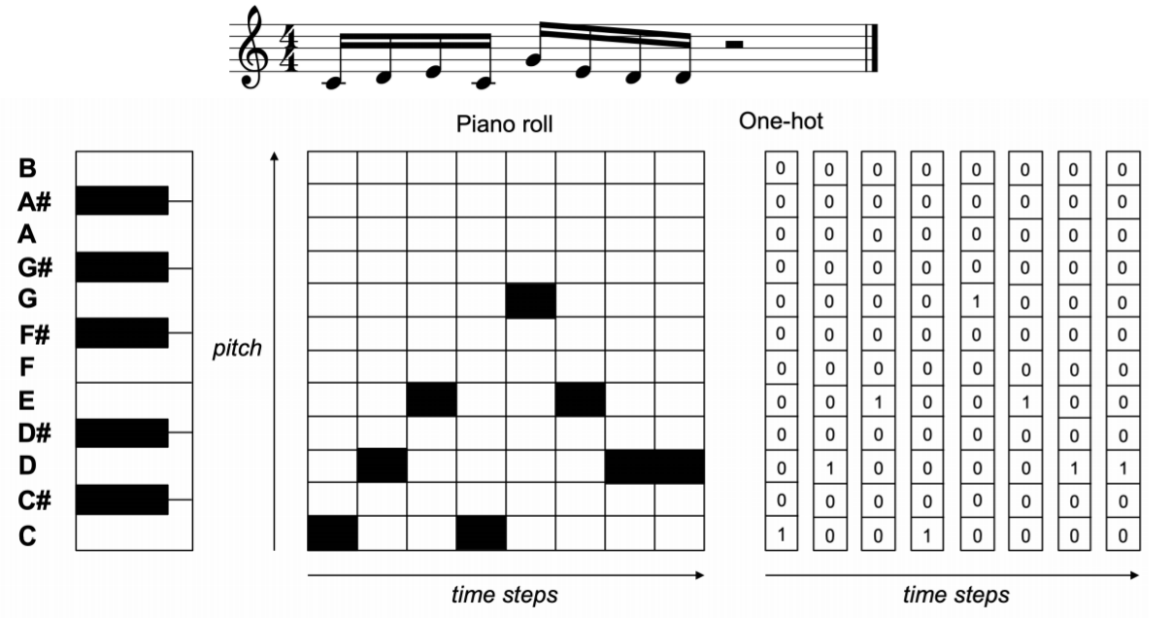


Figure 3 : Exemple de piano roll et sa correspondance après l'encodage one hot encoding [10]

Fichiers MIDI (Musical Instrument Digital Interface)

Musical Instrument Digital Interface (MIDI) est une norme technique qui décrit un protocole, une interface numérique et des connecteurs pour l'interopérabilité entre divers instruments de musiques électroniques, logiciels et dispositifs [23].

Un fichier MIDI transporte les données de performance et les données de contrôle de notes spécifiques en temps réel. Les deux informations d'événement les plus cruciales sont Note-on et Note off. [5]

• Note on pour indiquer qu'une note est jouée. Il contient

– un numéro de canal, qui indique l'instrument ou la piste

– un numéro de note MIDI, qui indique la hauteur de note

– une vélocité, qui indique à quel niveau la note est jouée

Un exemple est « Note on, 0, 60, 50 » qui signifie « Sur le canal 1, commencez à jouer un do moyen avec une vélocité de 50 » ;

• Note off pour indiquer qu'une note se termine.

Plusieurs modèles utilisent le format MIDI au lieu de Waveform car la taille du fichier MIDI est comparativement plus petite [1], ce qui se traduit par un apprentissage plus rapide.

Ce format est couramment utilisé, car un nombre considérable de données sont disponibles.

```
2,   96, Note_on,   0, 60, 90
2,  192, Note_off,  0, 60,  0
2,  192, Note_on,   0, 62, 90
2,  288, Note_off,  0, 62,  0
2,  288, Note_on,   0, 64, 90
2,  384, Note_off,  0, 64,  0
```

Figure 4 : Échantillons d'un fichier MIDI [8]

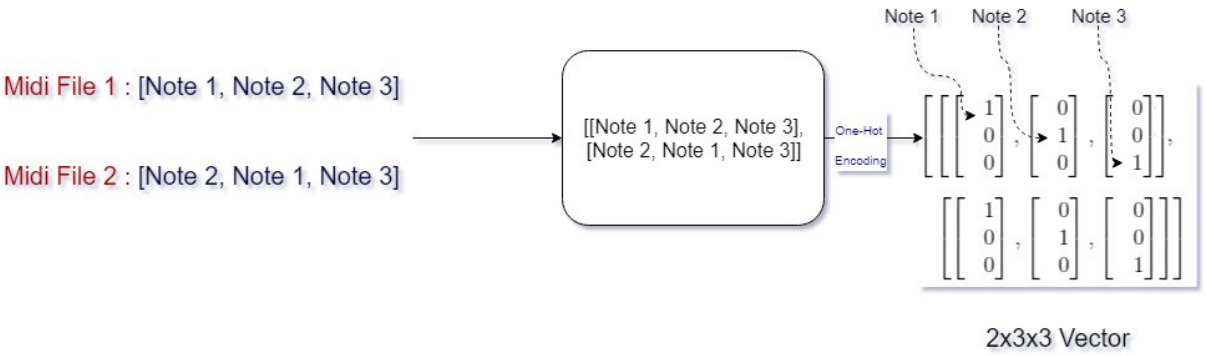


Figure 5 : architecture du fichier MIDI et sa conversion en vecteur à droite[4]

D'autres techniques [1] transforment les notes et les accords en une forme numérique qui peut être utilisée dans le réseau. C'est le cas de la technique "dictionary coding" utilisée par Music21, librairie basée sur Python pour la musicologie assistée par ordinateur. Elle est capable de lire des fichiers MIDI et d'obtenir des informations sur les notes et les accords. Après avoir traité une chanson, le fichier peut être présenté comme une liste de caractères ['G5', '10.3', 'B-4', 'G5', 'G5', 'B-4', 'F5',… , 'E-5', 'F4', 'D4', '5.10']. Les informations représentent l'ensemble de ses notes et informations sur les accords.

Notation ABC

La notation ABC se compose de deux parties :

[4] https://laptrinhx.com/generating-music-with-seq2seq-models-362271969/

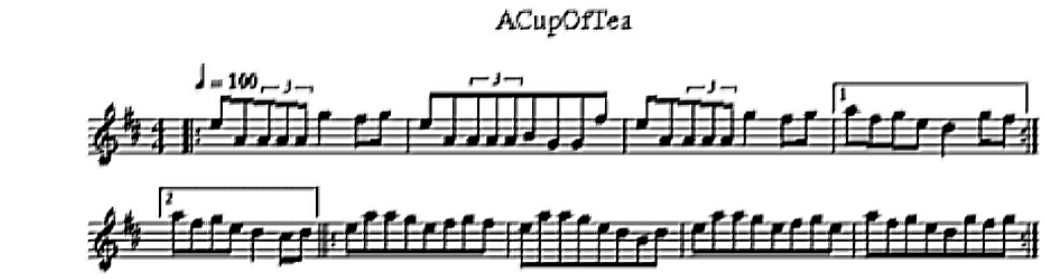
L'en-tête qui fournit les métadonnées de la musique telle que la signature, le titre, et le corps qui détaille les notes de la chanson [21]. La figure ci-dessous montre un exemple de chanson au format ABC. Les balises d'en-tête, telles que T pour le titre et X pour le numéro de la chanson, n'affectent pas la synthèse musicale. Les espaces et les nouvelles lignes dans le corps sont également des caractères étrangers et ne sont pas nécessaires pour la génération musicale. Par conséquent, les chansons sont ensuite traitées pour ne contenir que 5 informations d'en-tête :

type de morceau (R), signature rythmique(M), longueur de note unitaire (L), nombre de bémols [5]et mode de morceau (K) [27].

De plus, deux autres en-têtes de métadonnées sont générés : nombre de mesures et complexité du morceau (Q), qui est le nombre moyen de notes jouées chaque battement.

Il y a deux approches pour insérer les inputs dans le réseau de neurones. La première option consiste à alimenter le fichier ABC dans le modèle, caractère par caractère. La deuxième option est de tokeniser le fichier puis d'utiliser les tokens comme une séquence [27].

[5] Le bémol (symbole ♭) est un signe d'altération, destiné à indiquer sur une partition de musique un abaissement d'un demi-ton chromatique de la hauteur naturelle des notes associées.(wikipedia)




```
X: 1
T: A Cup Of Tea
Z: dafydd
S: https://thesession.org/tunes/3038#setting3038
R: reel
M: 4/4
L: 1/8
K: Amix
|:eA (3AAA g2 fg|eA (3AAA BGGf|
eA (3AAA g2 fg|1afge d2 gf:|2afge d2 cd||
|:eaag efgf|eaag edBd|eaag efge|afge dgfg:|
```

```
X:1
T:ACupOfTea
R:reel
M:4/4
L:1/8
K:Amix
Q:1/4=100
|:eA(3AAAg2fg|eA(3AAABG
Gf|eA(3AAAg2fg|1afged2gf:
|2afged2cd|||:eaagefgf|eaa
gedBd|eaagefge|afgedgfg:|
```

Figure 6 : En haut: la partition, à gauche : le fichier ABC original, à droite: fichier ABC formaté [27]

C. Techniques et exemples

Les réseaux de neurones sont une technique d'apprentissage automatique dans laquelle des couches contenant des nœuds sont empilées. Les données d'entrée entrent dans les nœuds des couches d'entrée et les informations sont combinées de manière pondérée et transmises à la couche suivante et ainsi de suite, jusqu'à ce qu'elles sortent de la couche de sortie.

FeedForward (Réseau de neurones à propagation avant)

Gaëtan Hadjeres et Jean-Pierre Briot proposent de générer une musique d'accompagnement d'une mélodie donnée [8], ils appellent ce système MicroBach.

La technique consiste à juxtaposer bout à bout les vecteurs de notes successifs et les faire correspondre aux différentes variables d'entrée du réseau. En sortie, on retrouve la concaténation des trois mélodies d'accompagnement [8].

Pour chacune des trois voix (alto, ténor et basse), chaque vecteur successif représente la note produite. En pratique, il suffit de choisir la note dont la valeur de l'élément correspondant est la plus grande (et ainsi la plus probable) [6].

On remarque que l'architecture est un assemblage de couches successives :

• La première couche, composée de nœuds d'entrée, est appelée couche d'entrée.

• La dernière couche, composée de nœuds de sortie, est appelée couche de sortie.

• Et toute couche entre la couche d'entrée et la couche de sortie est appelée couche cachée.

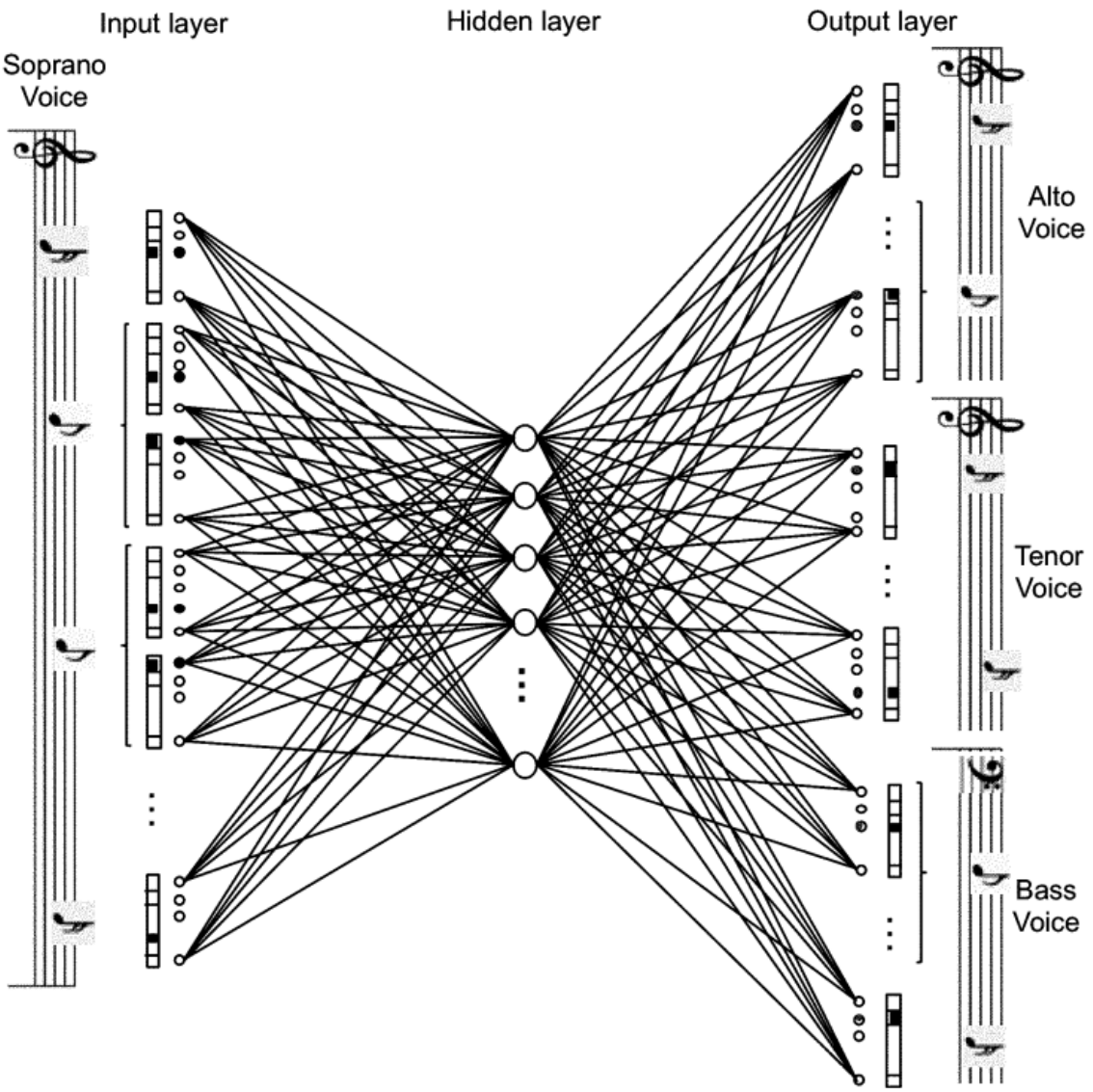


Figure 7 : FeedForward - [6]

Réseaux récurrents (RNN)

Le désavantage de l'architecture précédente est que la taille des mélodies générées est fixe. De ce fait, pour générer une mélodie de taille aléatoire, on peut utiliser la technique de RNN.

Le modèle RNN correspond à la première architecture de réseau neuronal pour la génération musicale. Dès 1989, Todd [13] utilisait RNN pour générer une mélodie monophonique pour la première fois. Cependant, il est difficile pour les RNN de stocker de longues informations historiques sur les séquences. Pour résoudre ce problème, Hochreiter [14] a conçu une architecture RNN spéciale : LSTM pour aider le réseau à mémoriser et à récupérer des informations dans la séquence. (cf partie suivante).

En 2016, l'équipe Magenta de Google Brain a proposé le modèle Melody RNN [20], améliorant encore la capacité de RNN à apprendre des structures à long terme. Cette technique vise à entraîner le réseau sur un ensemble d'exemples qui comporte en entrée une note et en sortie la note suivante [1]. C'est-à-dire que le réseau apprend à prédire les notes à l'instant t + 1 en utilisant les notes à l'instant t comme entrées.

Grâce à des échantillons extraits d'un grand nombre de partitions. La génération s'effectue de manière récursive, en présentant une note de début, puis en générant la note suivante, qui servira de nouvelle entrée au réseau, et ainsi de suite pour générer une suite de notes (mélodie). Cependant, même si le résultat est correct, on remarque rapidement que le réseau ne peut pas capturer les relations à long terme et, de ce fait, la mélodie générée manque de cohérence et de structure.

Un réseau feed-forward n'aurait aucune chance de composer de la musique de cette manière. Il lui manquerait la capacité de stocker des informations sur le passé, et de garder une trace d'où il se trouve dans une chanson.

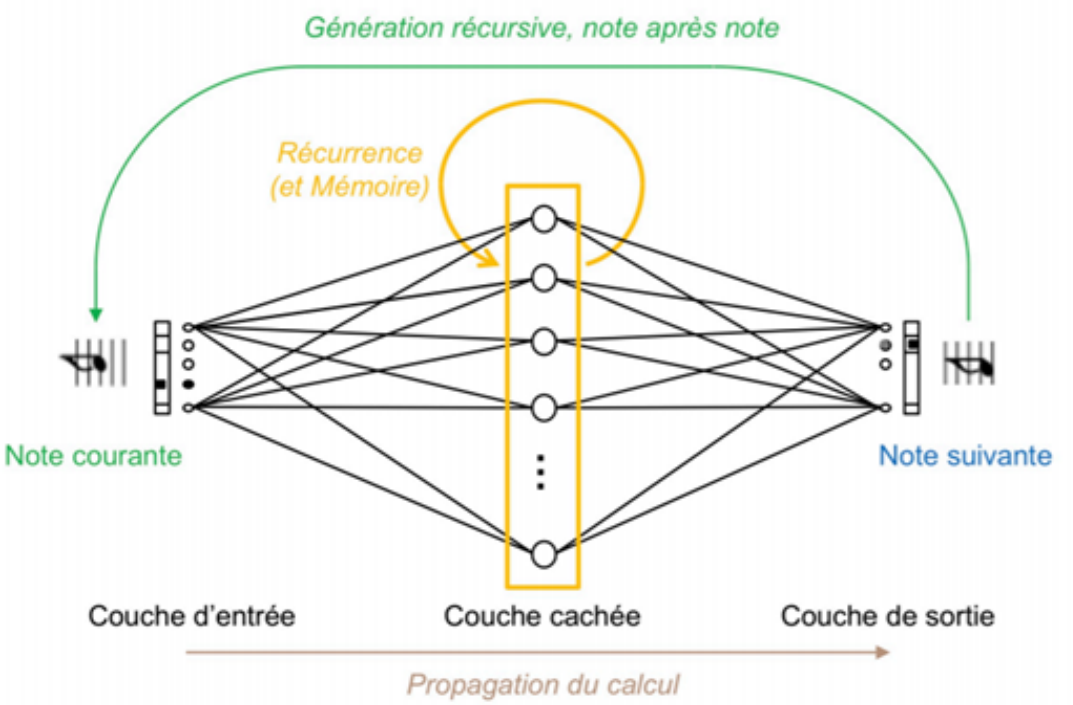


Figure 8 : Architecture d'un réseau récurrent et génération récursive note par note [15]

Long short term memory (LTSM)

En 2002, Eck et son équipe. [18] ont utilisé pour la première fois le LSTM dans la création musicale, improvisant du blues avec bon rythme et structure raisonnable basée sur un enregistrement court.

Les réseaux de mémoire à long court terme sont un type spécial de RNN, capables de se souvenir des informations pendant de longues périodes de temps.

Ils peuvent être utilisés pour générer des musiques et mélodies sans aucune intervention humaine. L'objectif principal est de développer un modèle qui peut apprendre à partir d'un ensemble de notes de musique, les analyser puis générer un nouvel ensemble de notes de musique. Le modèle doit être capable de se rappeler des détails passés sur la note pour une utilisation future dans la séquence d'apprentissage.

L’article [25] propose un algorithme qui peut être utilisé pour générer de la musique avec le système LSTM.

Plusieurs étapes sont nécessaires pour construire le réseau de neurones. Le modèle est appliqué sur des notes musicales polyphoniques. Le réseau LSTM est formé pour acquérir les connaissances et les probabilités d'occurrence d'une note de musique courante. La couche LSTM dépend d'une entrée sélectionnée. Toutes les notes ne subissent pas le processus d’apprentissage. Seules certaines sont sélectionnées et les notes spécifiées sont utilisées pour former ce modèle LSTM. Avec ces entrées, la couche LSTM apprend la cartographie et les corrélations entre les notes et leur projection.

- Input :

données d'entrée sous format de fichier MIDI. Les fichiers MIDI jouent un rôle important dans l'extraction d’informations sur la séquence de notes, la vélocité des notes et le rythme.

- Étape 1 : embedding word

L'embedding est une technique qui convertit les mots du langage naturel en formes vectorielles ou matricielles que les ordinateurs peuvent comprendre. [5]

- Étape 2 : LSTM

La couche prend une séquence en entrée et peut renvoyer des séquences ou une matrice.

Pour chaque note à générer, il est nécessaire de soumettre une séquence au réseau. La première séquence est la séquence de notes à l'index de départ. Pour chaque séquence suivante en entrée, la première note de la séquence est supprimée et la sortie de l'itération précédente est insérée à la fin de la séquence [26], comme le montre la figure ci-dessous.

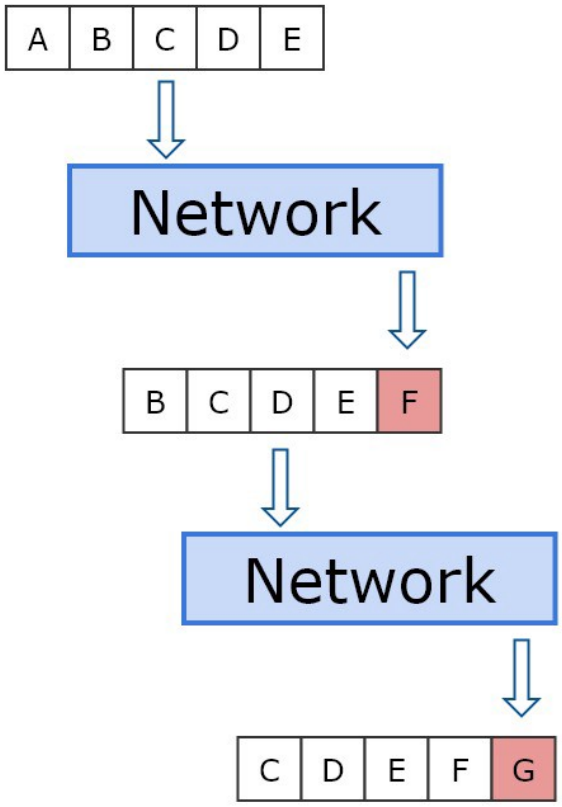


Figure 9 : Processus avec la séquence d'entrée ABCDE[6]

Pour déterminer la prédiction la plus probable à partir de la sortie du réseau, l'indice de la valeur la plus élevée est extrait. La valeur à l'index X dans le tableau de sortie correspond à la probabilité que X soit la note suivante. La figure suivante aide à expliquer cela.

[6] https://towardsdatascience.com/

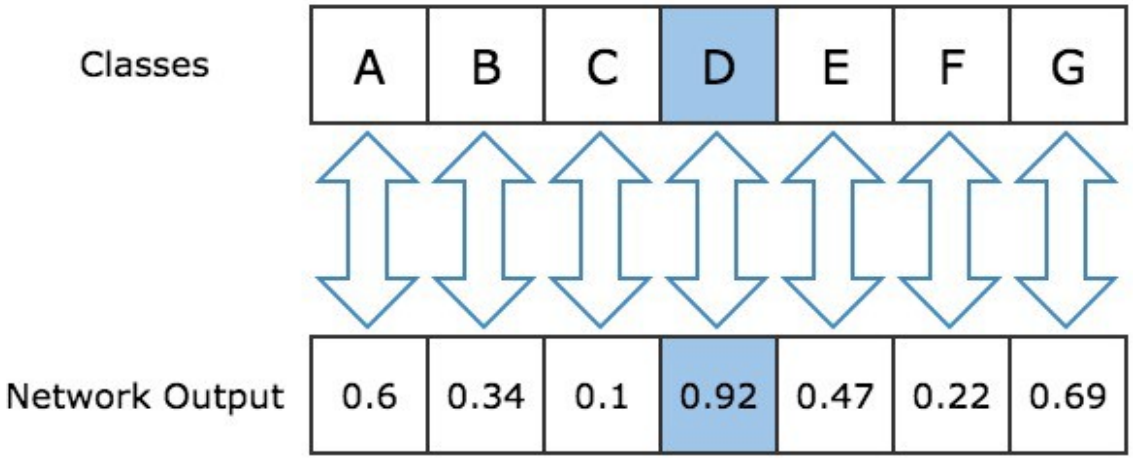


Figure 10 : Correspondance entre la prédiction de sortie du réseau et les classes.[7]

La probabilité la plus élevée est que la valeur suivante soit D, D est donc choisie comme classe la plus probable. Ensuite, les sorties du réseau sont regroupées dans un seul tableau.

- Étape 3 : Dropout

Dans le modèle de machine learning, s'il y a trop de paramètres de modèle et pas assez échantillons, le modèle d'apprentissage peut conduire à un phénomène appelé surapprentissage [21]. La technique de dropout supprime de manière aléatoire un nombre fixe d'unités dans une couche du réseau et généralise la portion d'apprentissage effectuée par la couche LSTM [1].

[7] https://towardsdatascience.com/

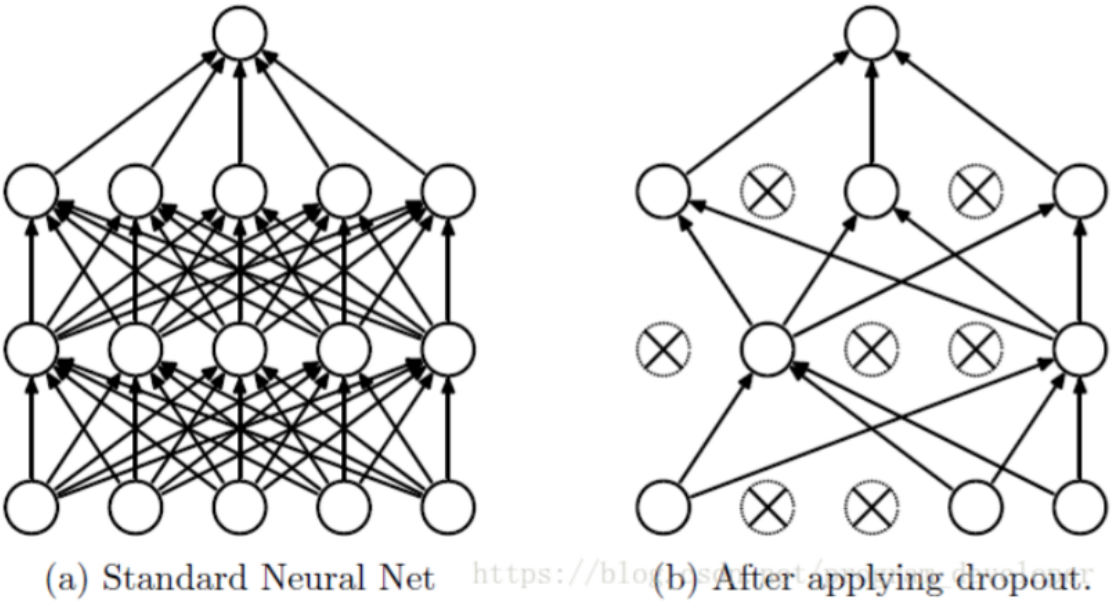


Figure 11 : Stratégie de Dropout [1]

- Dense :

Une fois que le modèle a appris la distribution de probabilité des notes et des séquences, il faut combiner toutes les cellules LSTM entre elles. Cet écart est subordonné à l'aide de la couche dense qui garantit que le modèle est entièrement connecté et produit la sortie requise par l'utilisateur [25].

- Activation

À la fin, une couche d'activation est ajoutée au modèle, ce qui aide à décider quels neurones (cellules LSTM) doivent être activés et si les informations obtenues par le neurone sont pertinentes, rendant la fonction d'activation très importante dans un neurone profond réseau [25].

En entrant un jeu de données original et en générant les notes ou les accords suivants de manière récurrente, le modèle est capable de générer de la nouvelle musique

Variational autoencoder (VAE)

Vineet Tiwari et Rushikesh Pokharkhar de l'université de Mumbai ont utilisé un autre système : le variational auto-encoder VAE [2]

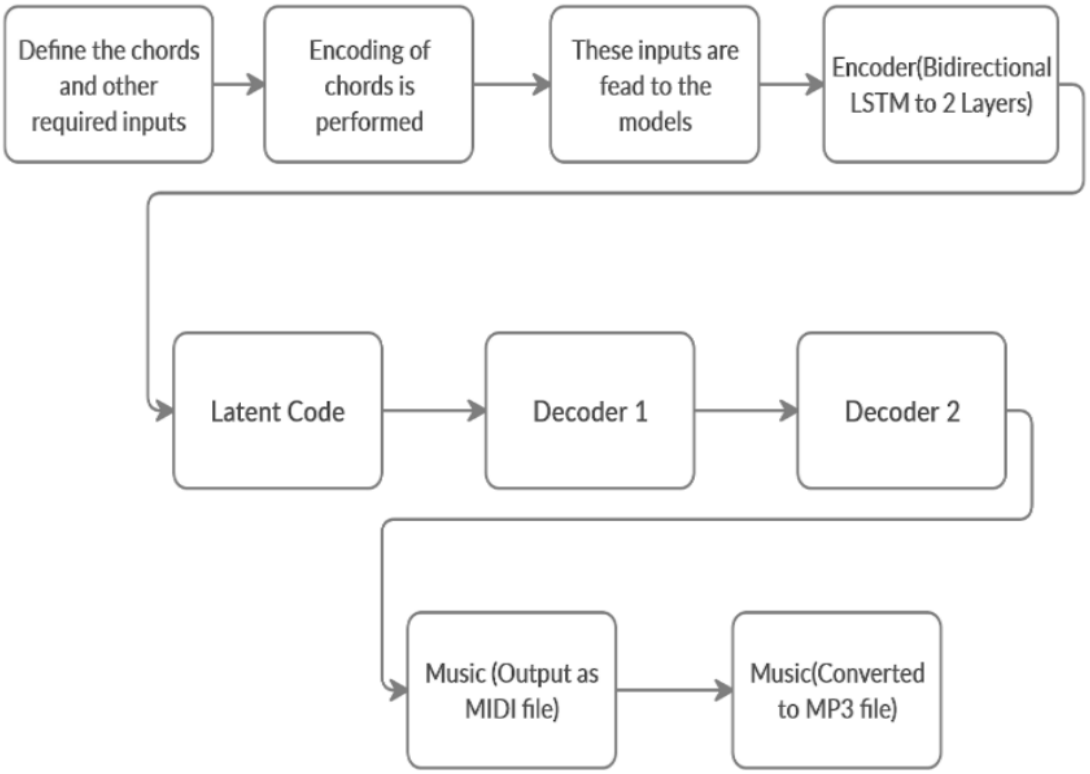


Figure 12 : Architecture d'une génération de musique polyphonique [2]

Ce processus est détaillé en neuf étapes, chacune étant décrite par un composant [2].

• Définir les accords et autres entrées requises: Quatre accords différents avec l'intensité de la musique, le nombre de mesures sera donné en entrée et en fonction de cela, le modèle générera de la musique contenant les accords.

• L'encodage des accords : une fois que l'utilisateur entre les accords requis, ils sont ensuite encodés en utilisant One hot encoding (cf partie piano-roll) pour que le modèle comprenne quels accords les chansons doivent inclure.

• Encodeur: Il consiste en un LTSM bidirectionnel à deux couches réseaux. Les entrées sont traitées à cette étape, puis alimentées à une couche entièrement connectée. La couche LSTM bidirectionnel est utilisée pour que le modèle puisse comprendre le long contexte du terme de la séquence [4].

• Décodeur: le décodeur de ce modèle est un RNN. Il utilise le code généré par l'encodeur et commence à générer la séquence de sortie de manière régressive. La raison d'utiliser un RNN hiérarchique est que le les RNN simples ont un échantillonnage et reconstruction des séquences médiocre dues au problème de à la disparition gradient [4].

• Sortie musicale: la séquence de sortie est ensuite convertie à un fichier de musique MIDI afin qu'il puisse être enregistré dans le système. Ce format MIDI n'est généralement pas utilisé comme une musique normale et nécessite un fichier de police sonore spécifique pour la lecture, donc nous convertissons ce fichier au format MP3 en utilisant FluidSynth et les bibliothèques LAME.

GAN (generative adversarial *network*)

Le réseau antagoniste génératif met en compétition deux réseaux au sein du système : le ” générateur ” et le ” discriminateur “ [27]. Le générateur a pour objectif de créer de nouvelles instances d’un objet (output) et l’envoyer au discriminateur. Celui-ci détermine alors l’authenticité de l’objet ou s’il fait ou non partie d’un ensemble de données.
Si le discriminateur n'est pas satisfait de la sortie du générateur, il la rétro propage au générateur pour une nouvelle génération de sorties [9].

Ainsi, le générateur produit des outputs de meilleure qualité tandis que le discriminateur détecte de mieux en mieux les faux. Ainsi, plus le temps avance, plus le processus devient meilleur. C’est ce que l’on appelle la rétropropagation.

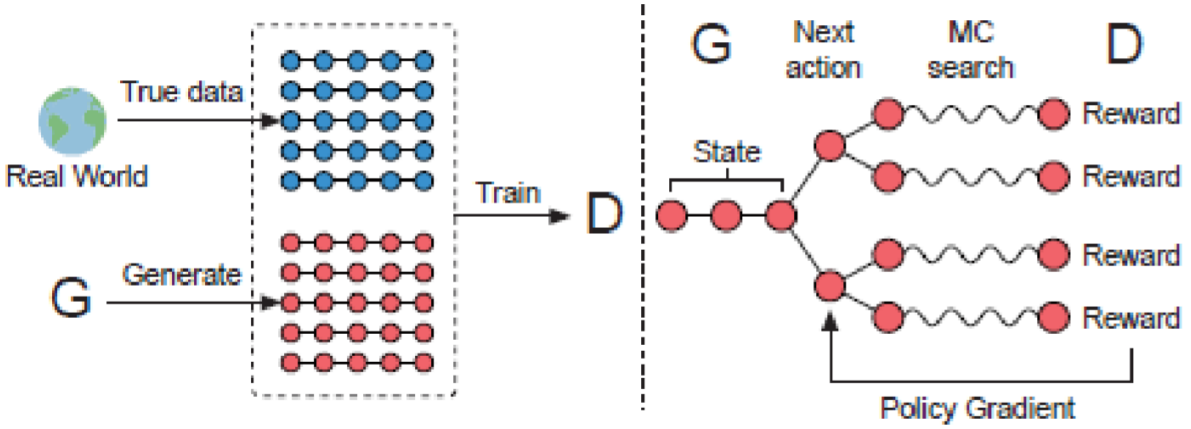


Figure 13 : Exemple de GAN avec G le générateur et D le discriminateur [27]

L'article [27] propose un modèle de RNN et de réseau antagoniste génératif (GAN). Il s'agit de la première équipe à combiner ces deux modèles pour construire un générateur de musique.

On remarque globalement que très peu d'articles existent concernant la génération de musiques à l'aide du modèle GAN.

D. Analyse des algorithmes présentés

Algorithmes proposé	Description	Avantages	Inconvénients
Feedforward	Dans ce réseau, les informations se déplacent vers l'avant, des nœuds d'entrée et nœuds cachés aux nœuds de sortie. Il n'y a pas de boucles dans le réseau.	Peut apprendre une fonction plus complexe.	est sujet au surapprentissage (overfitting). en raison du grand nombre de paramètres. Il faudra plus de temps et une grande taille de modèle pour générer de la musique.
RNN	Les réseaux de neurones récurrents se basent sur les prédictions précédentes pour être utilisées comme entrées,	- Capable de prendre des entrées de n'importe quelle taille - La taille du modèle n'augmente pas avec la taille de l'entrée - dépendant des informations antérieures	Temps de calcul élevé - l'accès à des informations lointaines est très difficile. - non prise en compte d'informations futures

LSTM	LSTM signifie mémoire à court long terme. Il s'agit d'un type particulier de RNN. Ils ont été introduits pour pallier le problème de la mémoire courte, ils ont 4 fois plus de mémoire que les	Capacité à modéliser les dépendances de séquence à long terme	Ils augmentent la complexité de calcul par rapport au RNN avec l'introduction de paramètres supplémentaires à apprendre. - La mémoire requise est supérieure à celle des RNN en raison de la présence de plusieurs cellules mémoire
VAE	Composé d'une paire de réseaux : L'encodeur fonctionne comme un compresseur de données, qui « résume » les données dans un espace de dimension plus petit. Le décodeur consiste à ramener les données à la distribution de probabilité d'origine. C'est-à-dire produire une musique comme celles que nous avons dans notre ensemble de	Nombre plus important de paramètres à régler ce qui permet un meilleur contrôle sur la modélisation.	Difficulté à mettre à l'échelle pour de grands ensembles de données et à lier à d'autres modèles.

	données.		
GAN	Le réseau antagoniste génératif est une technique qui repose sur la mise en compétition de deux réseaux au sein du modèle et permet de créer des imitations parfaites de données.	Le modèle utilise uniquement la rétropropagation. Il peut générer des échantillons plus rapidement car ils ne nécessitent pas de génération de données différentes.	Si le discriminateur est trop performant, la formation du générateur peut échouer en raison de la disparition des gradients. Un discriminateur optimal ne fournit pas suffisamment d'informations pour que le générateur progresse [27].

E. Exemple d'application de génération musicale

Plusieurs applications et librairies permettent de générer de la musique, elles utilisent ainsi les algorithmes vus précédemment :

Magenta :

Magenta est un projet de musique deep learning open source de Google. Le projet est devenu open source en juin 2016 et implémente actuellement un RNN régulier et deux LSTM [16]. Le site fournit une bonne documentation donc c'est relativement facile à mettre en place. L'équipe améliore activement les modèles et ajoute des fonctionnalités [16]. Il existe des modèles prédéfinis avec des milliers de fichiers midi. Il est donc possible de générer de nouveaux fichiers midi en utilisant ces modèles pré-entraînés

Bachbot :

Un projet de recherche de Feynman Liang à l'Université de Cambridge, utilise

également un LSTM. Cette fois, il sert à s'entraîner sur des chorals de Bach. Son objectif est de générer et d'harmoniser des chorals indiscernables de l'œuvre de Bach. Le site Web propose un test où l'on peut écouter deux flux et deviner lequel est une composition réelle de Bach. La recherche a montré que les gens ont du mal à distinguer le Bach généré du vrai. En outre, c'est l'un des meilleurs efforts pour gérer la musique polyphonique car l'algorithme peut gérer jusqu'à quatre voix

WaveNet :

Des chercheurs de DeepMind de Google ont créé Wavenet qui est basé sur les réseaux de neurones convolutifs. Leur objectif le plus prometteur est d'améliorer les applications de synthèse vocale en générant un flux plus naturel dans le son vocal [24]. Leur méthode peut également être appliquée à la musique car l'entrée et la sortie sont constituées d'audio brut.

Il utilise l'audio brut comme entrée. Par conséquent, il peut générer n'importe quel type d'instrument, et même n'importe quel type de son.

III. Discussion

A. Limites de l’utilisation du deep learning pour la génération de musique

La plupart de ces expérimentations montrent un intérêt technologique indéniable, mais l’intérêt artistique reste marginalisé. Les expériences décrites précédemment utilisent le plus souvent une approche similaire: un réseau de neurones artificiel entraîné sur un très grand nombre de partitions d’un ou plusieurs compositeurs. Si on lui présente quelques notes au départ, le réseau va tenter de prédire une suite en se basant sur ce qu’il a appris. Cela donne des résultats bluffants pendant une courte période, mais sur le long terme, cela devient rapidement ennuyeux.

B. Perspectives et challenges futur

L'enjeu serait donc de mettre dans le processus, des émotions, par exemple via la réaction de musiciens ou à terme du public et ainsi d'avoir un «feedback émotionnel» [17].
Le plus important est de remettre l'humain au centre. car sa place est souvent oubliée dans les projets d'intelligence artificielle. Ainsi, une intelligence artificielle pour la musique doit augmenter les capacités créatrices d'un artiste et non chercher implicitement à le remplacer.

Pour cela, il serait intéressant de mettre en place davantage d'intelligence artificielle interactive pour qu'elles puissent s'insérer dans le processus créatif du compositeur. Ce type d'interaction ouvre des perspectives nouvelles.

On peut également imaginer des travaux sur un ensemble de données beaucoup plus vaste, en utilisant plus d'unités LSTM et en essayant différentes combinaisons de paramètres pour voir les performances du modèle.

Le développement de la génération musicale utilisant les techniques numériques n'est pas seulement significatif pour la musicologie et l'informatique, mais peut également être appliqué à de nombreux domaines différents tels que la psychologie et la neuroscience en générant une musique apaisante pour le cerveau humain. Les ordinateurs peuvent générer automatiquement de la musique pour la thérapie en sélectionnant simplement plusieurs échantillons.

Avec l'aide de la génération musicale, de nombreux problèmes actuels peuvent être résolus et des idées créatives peuvent devenir réalité. Il est important d'accélérer le processus de formation et de minimiser la taille de l'ensemble de données pour générer de la musique plus efficacement à l'avenir.

Conclusion

Cet état de l'art a montré différents modèles qui peuvent être utilisés pour générer automatiquement de la musique et des mélodies sans aucune intervention humaine.

Bien que les résultats ne soient peut-être pas parfaits, ils sont néanmoins assez impressionnants et nous montrent que les réseaux de neurones peuvent créer de la musique et pourraient potentiellement être utilisés pour aider à créer des pièces musicales plus complexes.

Plus généralement, mon travail a permis de dégager la voie pour un futur travail de mémoire où la génération musicale par intelligence artificielle serait traitée plus en profondeur, notamment au travers des données utilisées et de mise en pratique.

Aujourd'hui, le nombre d'applications et de plugins basés sur de l'intelligence artificielle commercialisés n'est pas très important dans le secteur musical. L'état de l'art et les résultats m'amènent à penser que le domaine audio et musical à l'aide de réseaux de neurones n'en est qu'à ses débuts.

Table des illustrations

Figure 13 : Exemple de GAN avec G le générateur et D le discriminateur [27]
27

Annexes

Les musiques ci-dessous sont présentées afin d'illustrer les exemples d'algorithmes utilisés dans cet état de l'art.

Algorithme RNN : https://www.youtube.com/watch?v=A2gyidoFsoI

Algorithme LSTM : https://www.youtube.com/watch?v=wcmmDJeDRJ4

Algorithme VAE : https://www.youtube.com/watch?v=G5JT16flZwM

Références

[1] XIE, Jiatong. A Novel Method of Music Generation Based on Three Different Recurrent Neural Networks. In : Journal of Physics: Conference Series. IOP Publishing, 2020. p. 042034.

[2] TIWARI, Vineet, SHIVAPRASAD, Pratheesh, et RUSHIKESH, Rushikesh. Polyphonic Music Generation. Available at SSRN 3558389, 2020.

[3] SIPHOCLY, Nermin Naguib, SALEM, Abdel-Badeeh M., et EL-HORABTY, El-Sayed M. Applications of Computational Intelligence in Computer Music Composition. International Journal of Intelligent Computing and Information Sciences, 2021, vol. 21, no 1, p. 59-67

[4] Yamshchikov, Ivan P., and Alexey Tikhonov. "Music generation with variational recurrent autoencoder supported by history." SN Applied Sciences 2, no. 12 (2020): 1-7.

[5] Allen Huang, Yoshua Bengio et Raymond Wu. Deep Learning for Music. 2016

[6] Jean-Pierre Briot, Gaetan Hadjeres et François-David Pachet. Deep Learning Techniques for Music Generation. 2019

[7] SHUVAEV, Sergey, GIAFFAR, Hamza, et KOULAKOV, Alexei A. Representations of sound in deep learning of audio features from music. arXiv preprint arXiv:1712.02898, 2017.

[8] BRIOT, Jean-Pierre et PACHET, François. Music generation by deep learning-challenges and directions. arXiv preprint arXiv:1712.04371, 2017.

[9] Li, Shuyu, and Yunsick Sung. "INCO-GAN: Variable-Length Music Generation Method Based on Inception Model-Based Conditional GAN." Mathematics 9, no. 4 (2021): 387.

[10] Briot, Jean-Pierre. "From artificial neural networks to deep learning for music generation: history, concepts and trends." Neural Computing and Applications 33, no. 1 (2021): 39-65.

[11] Dorien Herremans, Ching-Hua Chuan, and Elaine Chew. A functional taxonomy of music generation systems. ACM Computing Surveys (CSUR), 50(5):1–30, 2017.

[12] Lejaren Arthur Hiller and Leonard M Isaacson. Experimental Music: Composition with an electronic computer. Greenwood Publishing Group Inc., 1979.

[13]] Peter M Todd. A connectionist approach to algorithmic composition. Computer Music Journal, 13(4):27–43, 1989.

[14] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997

[15] BRIOT, Jean-Pierre. Apprentissage profond et génération de musique. 2019.

[16] CASELLA, Pietro et PAIVA, Ana. Magenta: An architecture for real time automatic composition of background music. In : International Workshop on Intelligent Virtual Agents. Springer, Berlin, Heidelberg, 2001. p. 224-232.

[17] Ferreira, Lucas N., and Jim Whitehead. "Learning to generate music with sentiment." arXiv preprint arXiv:2103.06125 (2021).

[18] Douglas Eck and Juergen Schmidhuber. Finding temporal structure in music: Blues improvisation with lstm recurrent, 2002

[19] BAYLE, Yann. Apprentissage automatique de caractéristiques audio: application à la génération de listes de lecture thématiques. 2018. Thèse de doctorat. Université de Bordeaux.

[20] Elliot Waite et al. Generating long-term structure in songs and stories. Magenta Bolg: https: //magenta.tensorflow.org/blog/2016/ 07/15/ lookback-rnn-attention-rnn/, 2016.

[21] INGALE, Vaishali, MOHAN, Anush, ADLAKHA, Divit, et al. Music Generation using Deep Learning. arXiv preprint arXiv:2105.09046, 2021.

[22] MODRZEJEWSKI, Mateusz, DOROBEK, Mateusz, et ROKITA, Przemysław. Application of deep neural networks to music composition based on midi datasets and graphical representation. In : International Conference on Artificial Intelligence and Soft Computing. Springer, Cham, 2019. p. 143-152.

[23] MIDI Manufacturers Association (MMA). MIDI Specifications, Accessed on 14/04/2017. https://www.midi.org/specifications.

[24] OORD, Aaron van den, DIELEMAN, Sander, ZEN, Heiga, et al. Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499, 2016.

[25] Mangal, Sanidhya, Rahul Modak, and Poorva Joshi. "Lstm based music generation system." arXiv preprint arXiv:1908.01080 (2019).

[26]Xu, Xin. "LSTM Networks for Music Generation." arXiv preprint arXiv:2006.09838 (2020).

[27] WANG, Shao-Fan, CHEN, Tzu-Ping, HSU, Yuan-Lin, et al. Music Composition with Deep Learning. 2018.