Apprentissage profond et génération de musique

Jean-Pierre Briot

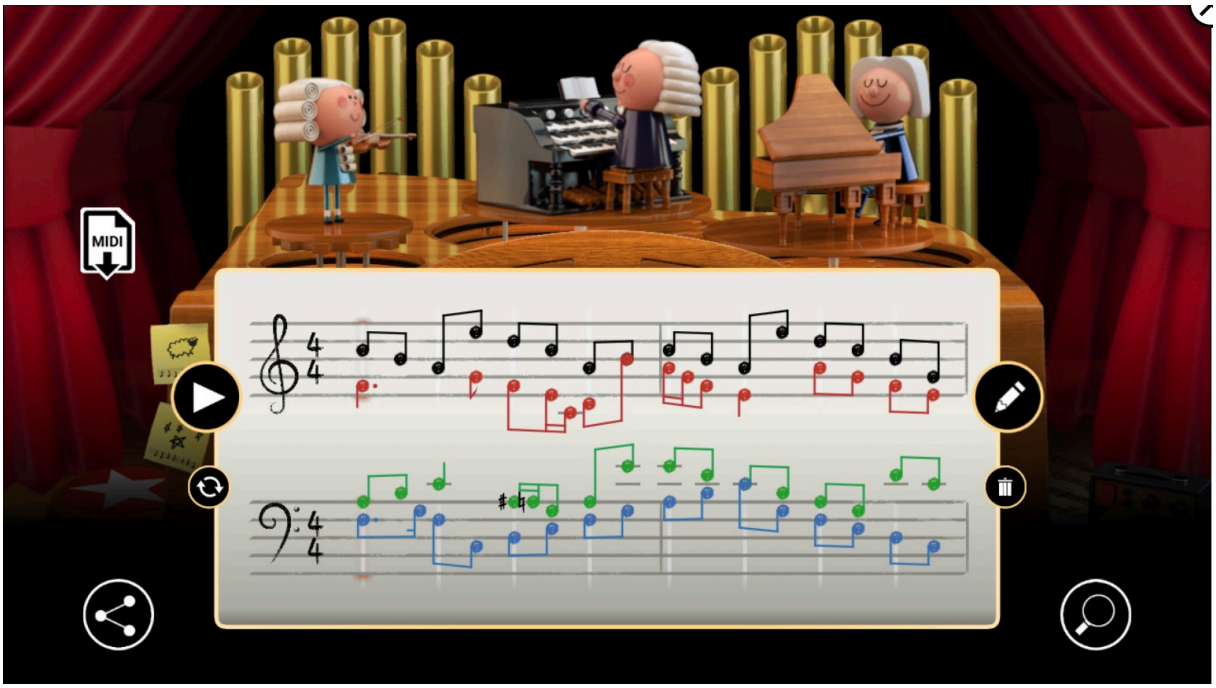
Apprentissage profond et génération de musique

Jean-Pierre Briot

Introduction

Le tsunami actuel de l'apprentissage profond (le retour hypervitaminé des réseaux de neurones artificiels) a récemment montré qu'il s'applique non plus seulement aux tâches et applications plus traditionnelles de l'apprentissage machine statistique, c'est-à-dire, la prédiction (par exemple, le prix d'appartements en fonction de différents critères) et la classification (de chiffres manuscrits, voir l'article « Apprentissage automatique et réseaux de neurones » du numéro hors série n° 68 « Intelligence artificielle » du magazine Tangente), mais qu'il a d'ores et déjà conquis d'autres domaines, tels que la traduction (par exemple, le service Google Translate), la reconnaissance (l'assistant Siri d'Apple) ou la synthèse de la voix (l'assistant Alexa d'Amazon).

Un nouveau domaine est la génération de musique, et de manière plus générale la génération de contenu créatif (texte, image, musique, son, vidéo). En mars 2019, à l'occasion de l'anniversaire de la naissance de Johann Sebastian Bach, Google a présenté un Doodle (une variation sur le logo de Google) créant un accompagnement par contrepoint associé à une mélodie définie interactivement par l'utilisateur, dans le style des chorals de Bach. L'architecture sous-jacente est un réseau de neurones profond, dont le principe de génération sera introduit plus loin. Par ailleurs, en octobre 2018, la célèbre société de vente aux enchères Christie's a adjugé pour plus de 400 000 dollars un tableau intitulé « Edmond De Belamy », créé par un autre type d'architecture de réseau profond (appelé réseau antagoniste génératif/créatif, GAN/CAN en anglais, voir la fin de l'article). Plusieurs jeunes entreprises, telles que AIVA (Artificial Intelligence Virtual Artist), Amper Music et JukeDeck, se partagent déjà le marché tout juste naissant de la génération de musique pour des documentaires ou des publicités, voire de la pop (la chanson Break Free de Taryn Southern en août 2017), et sont plusieurs à utiliser des architectures de réseaux profonds.



Bach Doodle – Exemple de génération contrapuntique de choral (mélodie soprano originelle en noir et celles générées en contrepoint en couleur). ©2019 Google LLC, utilisé avec permission.

Historique

Pour mieux comprendre, il est utile de revenir momentanément au début de l'informatique musicale à la fin des années 50, avec la création en 1957 par Lejaren Hiller et Leonard Isaacson de la composition intitulée « ILLIAC Suite » sur l'ordinateur ILLIAC I à l'Université de l'Illinois à Urbana-Champaign (UIUC) aux États-Unis. Les deux compositeurs, ou plutôt « méta-compositeurs », sont à la fois scientifiques et musiciens. Il s'agit d'un des tous premiers exemples de composition algorithmique à l'aide d'un ordinateur. L'approche consiste en la combinaison de modèles de Markov (modèles stochastiques de transitions) pour la partie génération et de règles (contraintes) pour la sélection. Cependant, la composition algorithmique peut être remontée dès la fin du 18ème siècle et attribuée (peut-être à tort) en 1787 à Mozart, avec son jeu de dés musical (« Muzikalisches Wurfelspiel ») dans lequel, en lançant à plusieurs reprises deux dés, on choisit chaque élément successif parmi un ensemble de segments mélodiques prédéfinis (et fixes).

Apprentissage de style musical

D'autres modèles peuvent être utilisés, telles des grammaires génératives, à l'image des grammaires pour une langue. Cependant, la définition de tels modèles par l'expert est difficile. L'idée d'apprendre automatiquement un tel modèle à partir de nombreux exemples est à la base du changement de paradigme de l'approche actuelle. En plus d'une moindre difficulté pour définir un modèle (qui n'exempt néanmoins pas des ajustements en partie empiriques des caractéristiques de l'architecture et de l'apprentissage, appelés hyperparamètres), le processus est automatique et générique, puisque le style musical dépendra du corpus d'exemples musicaux choisis.

Un des pionniers en matière d'utilisation de réseaux de neurones artificiels pour générer de la musique est Peter Todd en 1989. Il sera suivi par d'autres, puis, du fait de progrès notables, entre autres en matière de réseaux récurrents (voir l'encadré), puis de l'avènement des réseaux profonds (voir l'article « Apprentissage automatique et réseaux de neurones » du numéro hors série n° 68 « Intelligence artificielle » du magazine Tangente), le mouvement s'est à la fois accéléré et diversifié, comme nous allons le voir.

Principes

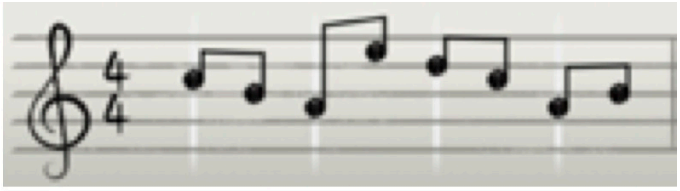
Pour commencer, considérons l'objectif suivant : générer un accompagnement d'une mélodie. L'accompagnement peut, par exemple, se traduire sous la forme d'une séquence harmonique formée de suites d'accords, ou bien sous la forme de plusieurs mélodies formant un contrepoint, par exemple, une polyphonie vocale. Considérons ce deuxième cas, qui est exactement celui du Bach Doodle (mentionné au début de l'article). Nous choisissons comme corpus (et donc comme style) les chorals de Johann Sebastian Bach, un (sinon « le ») maître en la matière de contrepoint. Dans le modèle des chorals, trois voix (alto, ténor et basse) sont créées et associées à la voix (soprano) originelle. Le problème d'apprentissage consiste donc à apprendre en fonction d'une entrée (la mélodie soprano), la sortie (les trois mélodies associées).

Représentation des données

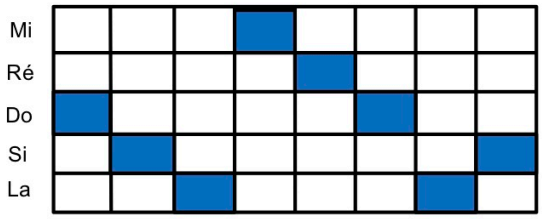
Il nous faut trouver un mode de représentation des notes d'une mélodie en vue de les transformer en variables d'entrée du réseau de neurones (et de manière duale pour les variables de sortie). L'approche la plus courante est le format de papier à musique (« piano roll »), inventé pour les pianos mécaniques et les orgues de barbarie, et dont nous considérons ici la version numérique. Le remplissage du rectangle correspondant à une note donnée et à un temps donné indique que la note correspondante sera jouée à ce moment. L'encodage naturel du piano roll est de faire correspondre à chaque segment temporel successif (correspondant à la durée minimale des notes, pour la mélodie soprano, une croche) un vecteur

ayant comme taille l'intervalle entre la note la plus basse et la note la plus haute (la tessiture, le nombre de lignes du piano roll) 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, inventée au départ pour l'électronique, se nomme ainsi « one hot »).

Partition



Piano roll



One hot

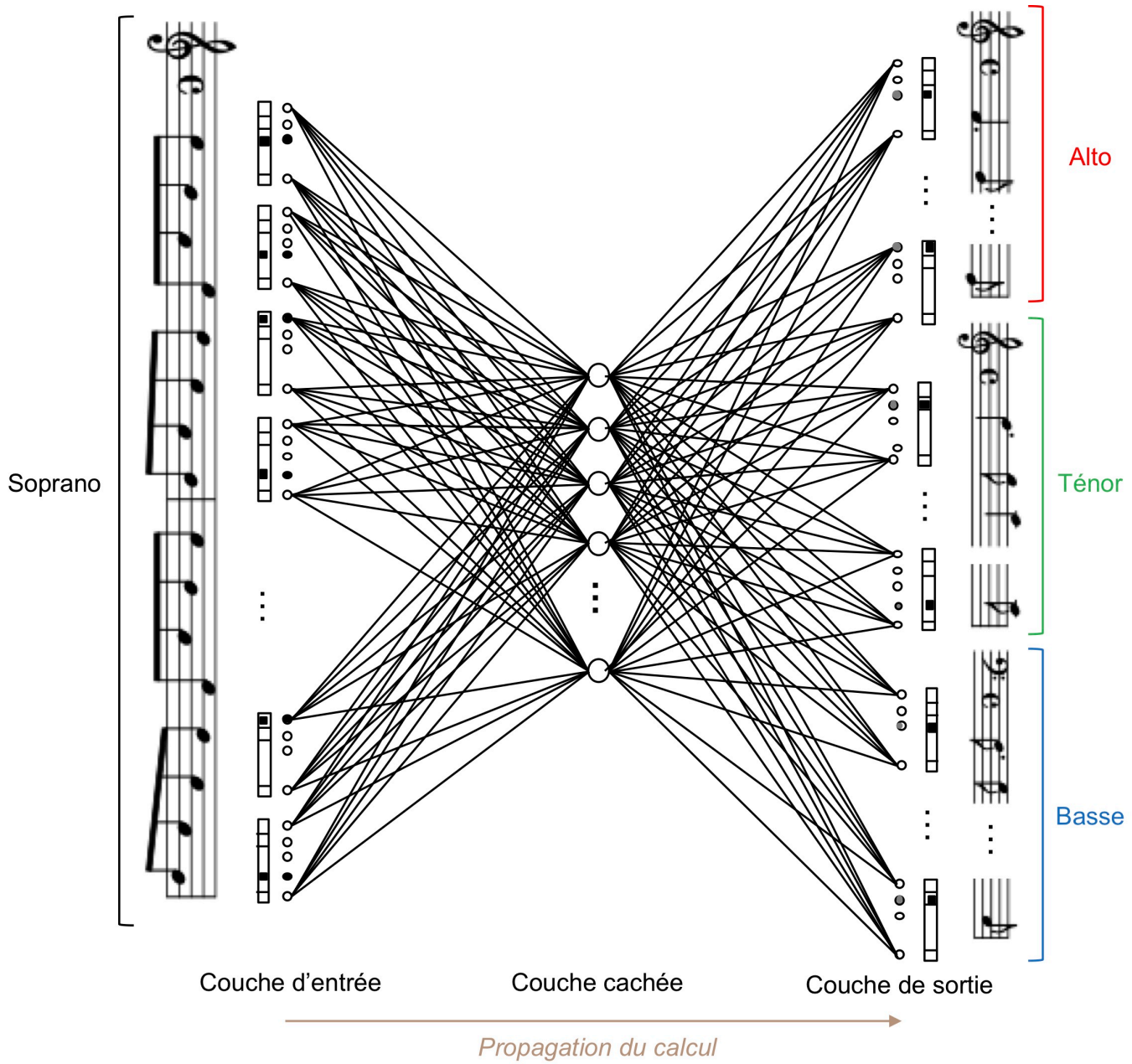
0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1
0	0	1	0	0	0	1	0

Représentations successives de la 1ère moitié de la mélodie soprano : partition, piano roll, vecteurs one-hot.

Notez que le cas du Bach Doodle est assez simplifié : pour la mélodie soprano il n'y a pas d'altération (par exemple, pas de note La# entre La et Si), une tessiture très réduite et une durée unique d'une croche. Dans le cas général, il faut pouvoir représenter une note tenue, qui doit être distinguée d'une note répétée. Une solution simple est de représenter la note tenue comme un élément additionnel au vecteur de notes. Le silence peut être implicite et correspondre à une absence de note et en conséquence un vecteur « zero one hot », sans aucun 1.

Architecture

La dernière étape est enfin de juxtaposer bout à bout les vecteurs de notes successifs correspondant aux segments temporels élémentaires successifs et les faire correspondre aux différentes variables d'entrée (également appelés nœuds de la couche d'entrée) du réseau. La couche de sortie représente quant à elle la concaténation des trois mélodies d'accompagnement. L'architecture présente un certain nombre de couches cachées, selon la profondeur du réseau (une seule dans le cas simplifié de la figure). Le décodage des valeur produites procède à l'inverse de l'encodage de l'entrée. Pour chacune des trois voix, chaque vecteur successif représente la note produite. En pratique, dans l'interprétation déterministe la plus directe, il suffit de choisir la note dont la valeur de l'élément correspondant est la plus grande (et ainsi la plus probable).



Architecture d'un réseau de neurones feedforward à une seule couche cachée générant un contrepoint.

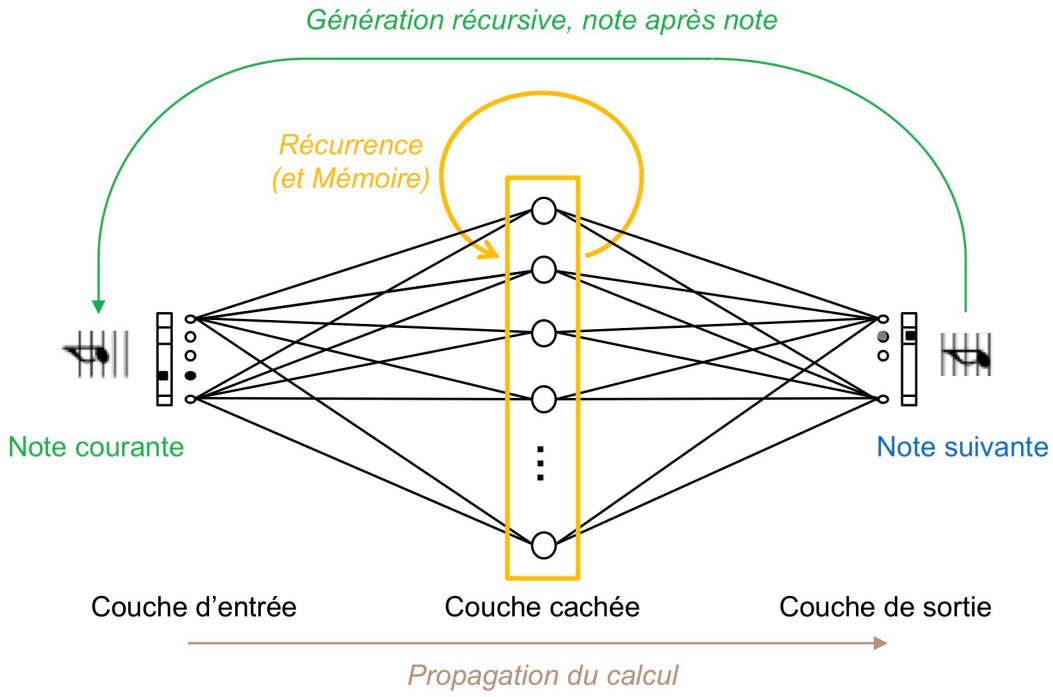
Lors de la phase d'apprentissage, le réseau est entraîné avec un grand nombre d'exemples (plus de 350 pièces du répertoire de chorals de Bach), en présentant pour chaque exemple la mélodie soprano en entrée et les trois mélodies d'accompagnement en référence pour la sortie. L'algorithme d'apprentissage, à base de descente de gradient (ou plus sophistiqué) et de rétropropagation, conduit à ajuster de manière incrémentale les poids des connexions entre les neurones, de manière à réduire l'erreur de classification (prédiction de chaque note pour chaque segment temporel et pour chaque voix).

Une fois le réseau entraîné, on peut alors débuter la phase de production (génération) : présenter une (ou plusieurs) mélodie arbitraire, et générer le contrepoint associé, selon le principe d'une telle architecture, de type « feedforward » (à propagation avant). C'est exactement sur ce principe que fonctionne le Doodle Bach de Google et également des systèmes plus sophistiqués, qui produisent des chorals dans le style de Bach difficiles à distinguer d'un original.

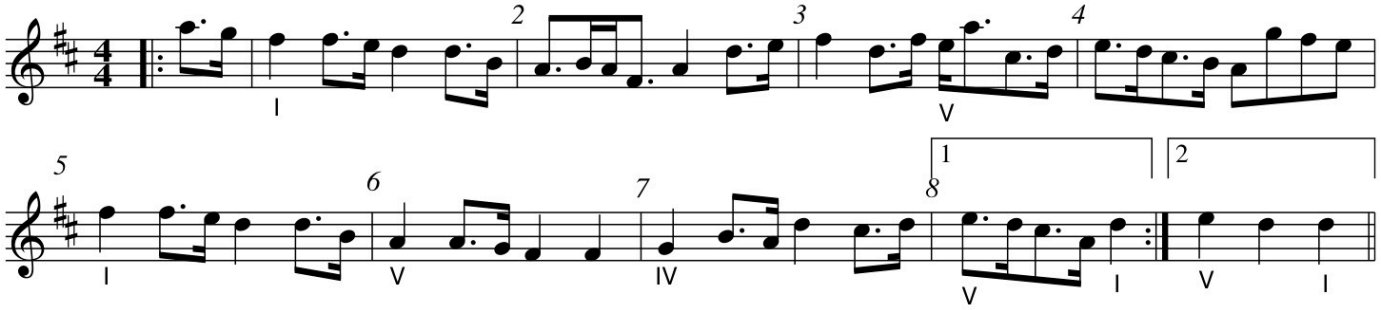
Réseaux récurrents (RNN)

Une limitation importante de l'architecture actuelle est que la taille des mélodies générées est fixée par l'architecture. De manière à générer une mélodie de taille arbitraire, on peut utiliser une autre stratégie, en entraînant un réseau de neurones plus simplifié sur l'apprentissage de la correspondance entre une note et la suivante. On entraine alors le réseau sur un ensemble d'exemples comportant en entrée une note et en sortie la note suivante, selon une combinatoire de segments extraits d'un grand nombre de partitions, par exemple, de mélodies utilisées par Bach ou bien d'un tout autre style, telles des mélodies de type celtique (Breton ou Irlandais). La génération procède cette fois de manière itérative, et plus précisément récursive, en présentant une note de départ, en générant la note suivante, qui servira de nouvelle entrée au réseau, et ainsi de suite pour générer une suite de notes (mélodie), correspondant au style appris. Cela fonctionne avec une architecture à propagation avant. Intuitivement, la transitivité va faire que les corrélations entre l'ensemble des notes successives découlent des corrélations deux à deux successives. En pratique, et même si le résultat est convenable, on s'aperçoit vite que le réseau n'a pas la capacité de capturer les relations à long terme et, par voie de conséquence, la mélodie générée manque de cohérence. L'utilisation d'une architecture récurrente (voir l'encadré) offre cette capacité de capture de relations à plus long terme.

> *Un réseau de neurones récurrent (en anglais RNN) ajoute une mémoire au niveau de chaque neurone d'une couche cachée ainsi que des connexions récurrentes (réentrantes, également pondérées et ainsi ajustées comme toutes les connexions lors de la phase d'apprentissage). Cela permet de tenir compte du calcul des éléments précédents. La version moderne, appelée LSTM (pour Long short-term memory), apprend à réguler l'accès à la mémoire et ainsi à minimiser les risques d'erreurs accumulées lors des calculs numériques des gradients par rétroprogagation pendant la phase d'apprentissage (le problème initial des « vanishing or exploding gradients » est ainsi résolu). Les réseaux récurrents sont utilisés de manière routinière pour des prédictions de séries (temporelles ou non), par exemple, en prédiction du climat, en traduction (voir l'article « La traduction automatique, un enjeu pour l'avenir » du numéro hors série n° 6 « Intelligence artificielle » du magazine Tangente), et donc ainsi en musique, où l'on peut considérer une mélodie comme une série temporelle de notes.*



Architecture de réseau récurrent et génération récursive note par note.



« The Mal's Copporim », mélodie celtique générée automatiquement par l'architecture de réseau récurrent folk-rnn de Bob Sturm et al., 2016. L'harmonie (degrés I, IV et V) est annotée manuellement par les auteurs. Reproduction avec la permission des auteurs.

Analyse et Défis

Les musiques générées par ces deux types d'architectures et approches, quand le corpus (ensemble d'exemples d'entraînement) est bien choisi (suffisamment important et cohérent) et que l'architecture est bien configurée (hyperparamètres bien choisis : bon nombre de couches, d'itérations, etc.), donne de bons résultats, voire de très bons résultats. Les musiques produites correspondent étonnamment bien au style appris et peuvent être confondues avec des originaux par la plupart des auditeurs. Cependant, au delà de l'aspect test de Turing musical réussi (voir l'article « Les pionniers » du numéro hors série n° 68 « Intelligence artificielle » du magazine Tangente), l'intérêt artistique reste limité. À quoi bon recréer des musiques dans un style déjà connu ? Un certain nombre de questions restent ainsi ouvertes :

- Structure – La musique générée, surtout si elle est longue, manque en général de direction et de structure.
- Contrôle – Un musicien va en général vouloir spécifier certaines contraintes, par exemple, sur la durée des notes, sur l'harmonie à suivre, etc.
- Originalité – Le musicien souhaite un équilibre entre une conformité à un style mais également que le contenu généré soit capable de le surprendre (et de lui plaire).
- Interactivité – La génération reste autonome et mécanique, avec pas ou peu d'interactivité avec le musicien, qui en général souhaite pouvoir retravailler certaines portions sans avoir à chaque fois à regénérer tout l'ensemble.

Ces défis sont des questions de recherche en cours et diverses stratégies et architectures plus complexes sont proposées et évaluées par les chercheurs du domaine, comme nous allons le voir.

D'autres architectures

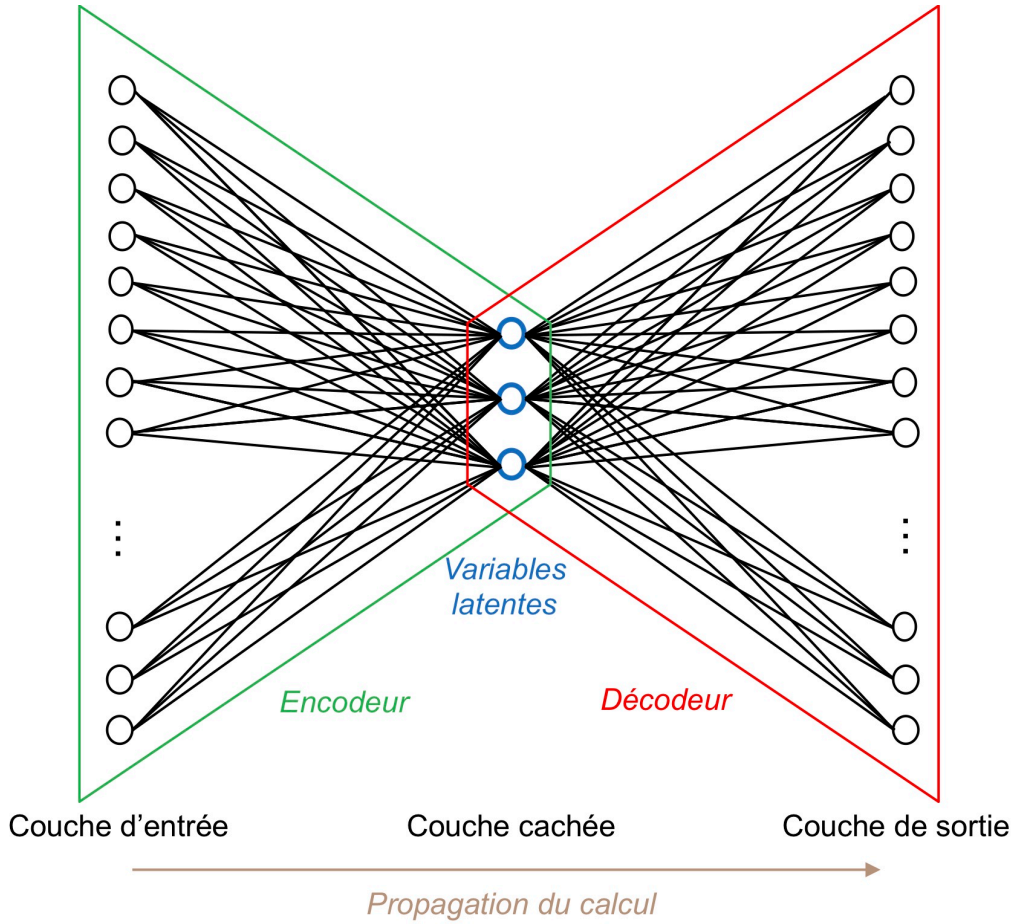
Notons tout d'abord que les deux exemples d'architectures présentées jusqu'à maintenant portent sur des représentations symboliques de la musique (portée, notes, accords…). Il existe cependant une tendance croissante à appliquer également de telles architectures à des représentations audio (spectre harmonique ou même signal brut de type forme d'onde). Les systèmes de reconnaissance et de génération de voix des assistants de Google ou d'Amazon sont des exemples de telles applications. On peut remarquer qu'au bout du compte, quelque soit la représentation initiale (notes, signal, pixel, lettres…), tout sera au final encodé sous la forme d'immenses vecteurs numériques pour alimenter les architectures de réseaux de neurones, ce qui assure cette généricité des architectures, indépendamment des représentations et des corpus. Néanmoins, les choix fins de représentation, par exemple, des notes tenues ou des silences, peuvent influer grandement sur la qualité des résultats.

Présentons maintenant brièvement quelques exemples d'architectures et de stratégies plus sophistiquées, en vue de répondre à certains des défis mentionnés ci-dessus. Les architectures d'autoencodeurs variationnels permettent d'apprendre les caractéristiques des variations entre les différents exemples d'un corpus musical. La couche cachée peut être réduite à deux neurones. Il est ensuite possible d'explorer l'espace latent à deux dimensions correspondant (les deux dimensions de variabilité peuvent, par exemple, correspondre à la tessiture et à la variance de la durée des notes, et vont dépendre du corpus d'exemples) et de générer différentes instances du style appris. Il suffit de choisir des valeurs correspondant aux variables latentes (par interpolation, extrapolation, combinaison…), et de les propager en avant dans la partie décodeur de l'autoencodeur pour générer une mélodie correspondant au style appris, tout en contrôlant les caractéristiques de variabilité. Il est possible d'emboîter un réseau récurrent dans la composante encodeur ainsi que dans la composante décodeur de l'autoencodeur. Cela permet ainsi de combiner les avantages de l'autoencodeur (exploration des caractéristiques discriminantes) et du réseau récurrent (taille variable). Ce type d'architecture, appelée RNN Encoder-Decoder, est d'ailleurs la base des systèmes de traduction actuels.

> *Un autoencodeur est un réseau de neurones à propagation avant, pour lequel la couche de sortie est identique à la couche d'entrée. Lors de la phase d'apprentissage, on présente pour chaque exemple la même donnée en entrée et en sortie. L'objectif est d'entraîner l'autoencodeur successivement à : encoder l'information dans la couche cachée, puis la décoder en vue de reconstruire au plus près l'information initiale. La couche cachée présente en général un nombre de neurones très inférieur au nombre de neurones de la couche d'entrée (et de sortie) et peut intégrer des contraintes additionnelles : de parcimonie (sparsity, c'est-à-dire de minimisation du nombre de neurones actifs en même temps, ce qui va entraîner leur spécialisation en fonction des caractéristiques discriminantes entre les exemples), et/ou bien de suivre une distribution de loi normale – les autoencodeurs variationnels). L'intérêt d'un autoencodeur n'est pas dans la génération de la donnée de sortie, mais dans l'effet induit d'apprendre une représentation condensée et représentative d'un ensemble de données (exemples) d'un espace à un grand nombre de dimensions (appelé une variété, manifold, en mathématiques), vers un espace à peu de dimensions (appelé espace des variables latentes). L'autoencodeur variationnel a l'avantage supplémentaire d'assurer une bonne correspondance entre une petite variation au niveau de l'espace latent et une petite variation au niveau de l'espace des données. Une analogie est l'approximation de surfaces terrestres à trois dimensions dans des cartes à deux dimensions.*

Les architectures de réseaux antagonistes génératif (GAN en anglais) sont un autre exemple d'avancée très intéressante. Un GAN (generative adversarial networks) consiste en deux réseaux de neurones, le générateur – chargé de générer des éléments les plus proches possibles d'une base d'éléments (par exemple, des photos, des tableaux, des musiques) de référence –, et le discriminateur – chargé de déterminer si l'élément qu'on lui présente en entrée est un élément de la base ou bien un élément synthétique produit par le générateur. Les deux réseaux sont entraînés simultanément, jusqu'à ce que le générateur arrive à tromper suffisamment le discriminateur (lui-même devenu de plus en plus performant). Le tableau « Edmond De Belamy », introduit au début de cet article, a été généré ainsi. Il existe également comme alternatives récentes aux architectures récurrentes des architectures de réseaux convolutifs (voir l'article sur le sujet dans ce même numéro), ainsi que des architectures basées sur un mécanisme d'attention multiple (portant sur différents éléments d'une séquence d'entrée, et contrôlé par l'apprentissage). Par ailleurs, des architectures de transfert de style (voir l'encadré « Les réseaux de neurones artistes » dans l'article « Apprentissage automatique et réseaux de neurones » du numéro hors série n° 68 « Intelligence artificielle » du magazine Tangente) peuvent être utilisées comme point de départ pour imposer des caractéristiques (structure, tonalité, rythme…) à des musiques existantes ou, encore

mieux, en cours de génération. Enfin, il existe des reformulations et des combinaisons avec des modèles d'apprentissage par renforcement, qui permettent d'inclure des objectifs et le retour (feedback) de l'utilisateur.



Architecture d'un autoencodeur.

Conclusion

Comme on l'a vu, le domaine de la création de contenu artistique selon un style appris est un domaine en plein développement, aux niveaux scientifique, technique et économique. Les contenus peuvent être musicaux, mais également textuels (par exemple, poésie) ou visuels (images, tableaux, vidéos). Les réussites techniques sont d'ores et déjà là, tout en laissant encore nombre de questions ouvertes. Nous considérons qu'au delà des enjeux plus techniques, se posent les questions des usages et de la collaboration entre humain (musicien) et machine, pour aller non pas vers des générateurs mécaniques de musique, mais vers des assistants pro-actifs permettant aux musiciens d'explorer et de produire dans une collaboration homme-machine des œuvres qu'il leur serait difficile d'imaginer sinon.

Références

Deep Learning Techniques for Music Generation, Jean-Pierre Briot, Gaëtan Hadjeres et François Pachet, Computational Synthesis and Creative Systems Series, Springer Verlag, 2019.

Apprentissage artificiel – Deep learning, concepts et algorithmes, Antoine Cornuéjols, Laurent Miclet et Vincent Barra, Eyrolles, 3ème édition, 2018.

Algorithmic Composition: Paradigms of Automated Music Generation, Gerhard Nierhaus, Springer Verlag, 2009.

Deep Learning, Ian Goodfellow, Yoshua Bengio et Aaron Courville, Adaptive Computation and Machine Learning Series, MIT Press, 2016.

Jean-Pierre Briot est Directeur de recherche CNRS au LIP6, laboratoire de recherche en informatique commun à Sorbonne Université (Paris) et au CNRS.