Université de Montréal

Expressivité et contrôle de modèles d'apprentissage automatique dans un corpus d'installations audiovisuelles

Par

Gabriel Lavoie Viau

Faculté de musique

Mémoire présenté en vue de l'obtention du grade de maîtrise

en musique, option composition et création sonore

Décembre 2022

Université de Montréal

Unité académique : Faculté de musique

Ce mémoire intitulé

Expressivité et contrôle de modèles d'apprentissage automatique dans un corpus d'installations audiovisuelles

Présenté par

Gabriel Lavoie Viau

A été évalué par un jury composé des personnes suivantes

Nicolas Bernier
Président-rapporteur

Dominic Thibault
Directeur de recherche

Myriam Boucher
Membre du jury

Résumé

L'appropriation d'algorithmes existants, la création d'outils numériques et des recherches conceptuelles ont mené à la création de deux installations audiovisuelles interactives. La première, *Deep Duo*, met en scène des réseaux de neurones artificiels contrôlant des synthétiseurs modulaires. La deuxième, *Morphogenèse*, l'œuvre d'envergure de ce mémoire, met en relation le spectateur avec des modèles profonds génératifs et le place face à des représentations artificielles de sa voix et de son visage.

Les installations et leurs fonctionnements seront décrits et, à travers des exemples de stratégies créatives et des concepts théoriques en lien avec l'interactivité et l'esthétique des comportements, des pistes pour favoriser l'utilisation d'algorithmes d'apprentissage automatique à des fins créatives seront proposées.

Mots-clés : art interdisciplinaire, installation audiovisuelle, interactivité, apprentissage automatique, modèles génératifs, synthétiseur modulaire, synthèse faciale, synthèse vocale, génération mélodique.

Abstract

The appropriation of existing algorithms, the creation of digital tools and conceptual research have led to the creation of two interactive audiovisual installations. The first, *Deep Duo*, features artificial neural networks controlling modular synthesizers. The second, *Morphogenesis*, the major work of this dissertation, connects the viewer with generative deep models and places them in front of artificial representations of their voice and face.

We will describe these installations and their functioning and, through examples of creative strategies and theoretical concepts related to interactivity and the aesthetics of behaviour, we will propose ways to promote the use of machine learning algorithms for creative purposes.

Keywords : interdisciplinary art, audiovisual installation, interactivity, machine learning, generative models, modular synthesizer, facial synthesis, voice synthesis, melodic generation.

Table des matières

Liste des tableaux

Liste des figures

Liste des extraits disponibles en annexe

À Pulque, qui a passé une bonne partie de la rédaction de ce mémoire à ronfler paisiblement à mes côtés, comme pour me rappeler de ne pas en faire tout un plat.

Remerciements

Merci à Dominic Thibault pour son immense générosité, ses conseils judicieux, son soutien sans failles, son écoute, sa curiosité, son savoir, sa motivation, sa disponibilité et son ouverture. Merci à Catherine pour son soutien, autant vocal que silencieux, d'une qualité rare et précieuse n'ayant d'égale dans sa constance que son absence totale d'attentes et de pression. Merci à Dominique de m'avoir donné de son temps et prêté son œil de correctrice émérite pendant le sprint final. Mais surtout, merci pour les encouragements et pour le soutien indéfectible en toute circonstance, et cela, depuis toujours.

Finalement, merci aussi à Nicolas Bernier, Guillaume Boutard et, encore une fois, Dominic Thibault de m'avoir donné du boulot et permis de travailler sur vos projets pendant mes études. Ce fut très agréable et apprécié.

Introduction

Pour la plupart d'entre nous, les algorithmes d'apprentissage automatique sont drapés d'une aura de mystère et de complexité et cela, en dépit de leur omniprésence et de l'influence qu'ils ont sur les réalités numériques que nous habitons. Dans les dernières années, certains de ces algorithmes ont démontré une capacité surprenante à générer du contenu visuel, musical et textuel, tel que le démontrent les modèles issus des recherches de Ramesh *et al.* (2022), Huang *et al.* (2018) et Brown *et al.* (2020). Mais pour l'instant, ceux-ci restent principalement dans les mains d'une communauté d'initiés. La majorité d'entre eux, incluant ceux créés pour générer du contenu médiatique, n'ont pas été pensés pour répondre aux besoins des artistes : « [...] a major limitation that we observe in this area, is that the majority of the methods available offer very limited, if any, creative control to a human user » (Akten, 2021, p.4). Ainsi, des questions fondamentales ont orienté ma recherche-création tout au long de mon parcours. Par quels moyens est-il possible d'exploiter le potentiel créatif de ces algorithmes malgré ces limitations actuelles? Aussi, certains de ces algorithmes génératifs ont une signature esthétique très typée et reconnaissable. Y a-t-il des moyens qui permettent de se servir de ceux-ci afin qu'ils soient au service d'une vision artistique? Finalement, est-ce que mettre en contact le public avec ces algorithmes par le biais d'installations audiovisuelles peut participer à les démocratiser?

Ces interrogations se résument dans la question de recherche suivante : comment peut-on adapter et utiliser les algorithmes d'apprentissage automatique comme outils pour créer des œuvres artistiques qui participeront à révéler au public le fonctionnement de ces algorithmes?

La méthodologie de recherche-création s'est articulée dans des allers-retours entre trois types d'activités qui s'influencent et se nourrissent tout au long du parcours. Cette méthode a permis de mieux comprendre comment adapter et utiliser des algorithmes d'apprentissage automatique comme outils de création :

- Recherche et appropriation d'algorithmes existants.
- Création d'outils créatifs menant à des installations audiovisuelles.
- Revue de littérature

Ce document, organisé en trois chapitres, résume le processus de création, tant au niveau technique que conceptuel. Mais surtout, il vise à partager les concepts et les idées rencontrés en cours de route qui me sont apparus comme les plus importants pour un artiste intéressé à l'utilisation de ces outils à des fins créatives. Le premier chapitre décrira l'installation sonore *Deep Duo* qui met en scène des réseaux de neurones artificiels contrôlant des synthétiseurs modulaires. Créée lors de ma première année de maîtrise, la conception de cette installation m'a permis de me familiariser avec les réseaux de neurones artificiels. Ce chapitre sera donc aussi une occasion de partager certains des concepts à la base de ces réseaux et de s'assurer de mettre à niveau le lecteur pour les sections subséquentes. Le chapitre suivant expose le fonctionnement de l'œuvre d'envergure de ce mémoire, *Morphogenèse*, qui met en relation le spectateur avec des modèles profonds génératifs et le place face à des représentations artificielles de sa voix et de son visage. Cette installation audiovisuelle utilise trois modèles génératifs différents, chacun d'une plus grande complexité que celui à l'œuvre dans *Deep Duo*. Par conséquent, le système de cette installation est lui aussi d'une plus grande complexité. Ce chapitre en fera la description technique, ce qui permettra de comprendre le rôle des trois modèles au sein de l'installation et de détailler l'implémentation de différentes stratégies créatives mise en place pour les utiliser. Le troisième et dernier chapitre porte un regard plus conceptuel sur les processus créatifs que j'ai engagés avec les différents modèles d'apprentissage à l'œuvre dans les deux installations décrites jusque-là. Il met aussi en lumière un changement de paradigme dans notre interaction avec l'ordinateur lors de la création avec les algorithmes d'apprentissage automatique. Finalement, à travers des exemples pratiques et des concepts théoriques en lien avec l'interactivité et l'esthétique des comportements, il propose des pistes pour favoriser l'utilisation de ces algorithmes à des fins créatives.

Bien que ce document aborde et, dans certains cas, tente de vulgariser des concepts liés aux algorithmes d'apprentissage automatique, il n'a ni les intentions ni les prétentions d'un mémoire en informatique ou en recherche opérationnelle. L'objectif est plutôt de décrire, d'interroger et de partager les processus qui m'ont permis d'utiliser certains de ces algorithmes comme outils de création.

Chapitre 1 - *Deep Duo* : une introduction à l'apprentissage automatique

Deep Duo est une création qui prend la forme d'une installation sonore interactive. L'installation utilise les mécanismes de l'apprentissage automatique dans un contexte de contrôle de synthèse modulaire.

Ce premier chapitre permettra de décrire cette première installation, tant sur le plan expérientiel que technique, et de faire un survol des concepts d'apprentissage automatique nécessaire à la compréhension de son fonctionnement.

1.1. Scénario d'interactivité

Cette section fait la description de l'installation du point de vue d'un spectateur, qui sera appelé interacteur pour souligner le fait que celui-ci peut interagir avec l'œuvre. Cette description subjective, écrite à la première personne du singulier, aidera à comprendre la création d'un point de vue expérientiel. Le document suivant, disponible en annexe, appuie cette description :

Extrait 1 - *Deep Duo* : documentation vidéo.

J'arrive dans un espace où je vois, à chaque extrémité d'une table, un écran et un synthétiseur modulaire. Face à cette table, un iPad est déposé sur un socle. J'entends des sons aux timbres électroniques qui semblent provenir de l'endroit où se trouvent les synthétiseurs. C'est une musique sans tonalité qui donne l'impression d'une série d'impulsions variant légèrement en termes de durée et de spectre harmonique.

Instinctivement, je m'approche de l'iPad, me doutant qu'il doit s'agir d'une sorte d'interface de contrôle. Sur l'écran, je vois un diagramme qui représente plusieurs couches de points successives. Chaque point est lié par une ligne à tous les points de la couche suivante. Je remarque aussi une série de menus déroulants dont je ne comprends pas bien les titres. Ces titres sont composés de termes tels que neurones, couches cachées et taux d'apprentissage. Un peu au hasard, je sélectionne des valeurs dans les menus déroulants. J'appuie sur le gros bouton où est écrit « entraînement ». Soudainement, le nombre de couches et de points dans le diagramme change et le paysage sonore se transforme. Le

timbre, les hauteurs et la durée des impulsions sonores sont en transformation. Simultanément, une courbe apparaît à l'écran. Plus cette courbe s'approche de zéro, plus le paysage sonore se stabilise. Je remarque que l'interface graphique contient un deuxième onglet avec des contrôles similaires. Ces onglets sont intitulés « système 1 » et « système 2 ». Je comprends alors que chaque onglet permet de contrôler un des deux systèmes posés sur la table.

Pendant que j'expérimente avec différents paramètres des systèmes, je remarque que les sons que j'émets ont une incidence sur le rendu sonore de l'installation. Taper du pied ou des mains crée une explosion d'impulsions sonores. Cette réalisation me motive à expérimenter avec les sons de ma voix. Je remarque que l'amplitude, la hauteur et les formants de ma voix sont pris en compte par le système et que je peux me servir de ceux-ci pour jouer avec le paysage sonore de l'installation. Je passe alors un certain temps à changer les valeurs de paramètres des deux systèmes, à les entraîner et à expérimenter avec les différentes manières qu'ils ont de réagir aux sons que je produis. Bien que le sens exact des noms des paramètres affichés sur l'écran de l'iPad m'échappe toujours, mes interactions avec le système m'amènent peu à peu à développer une connaissance intuitive des effets qu'ils ont sur le comportement de l'installation.

1.2. Description physique et technique

L'installation requiert une surface d'environ quatre mètres par quatre mètres où se trouvent deux synthétiseurs modulaires et deux écrans posés sur une table, un iPad et un microphone sur pied, une paire de haut-parleurs amplifiée, un ordinateur et une carte de son.

Tel qu'exemplifié par la figure 1 (fig. 1), chaque synthétiseur est associé à un écran et est placé à une extrémité de la table pour souligner l'indépendance des deux systèmes. Le pied, sur lequel sont placés l'iPad et le microphone, se trouve face à la table à une distance d'environ deux mètres. Le microphone est de type condensateur pour être en mesure de capter les sons de l'interacteur même s'il se trouve à une certaine distance de sa bouche. Les haut-parleurs sont placés à chaque extrémité de la table et sont chacun responsable de projeter le signal sonore en provenance de l'un des deux systèmes. La carte de son et l'ordinateur sont cachés sous la table.

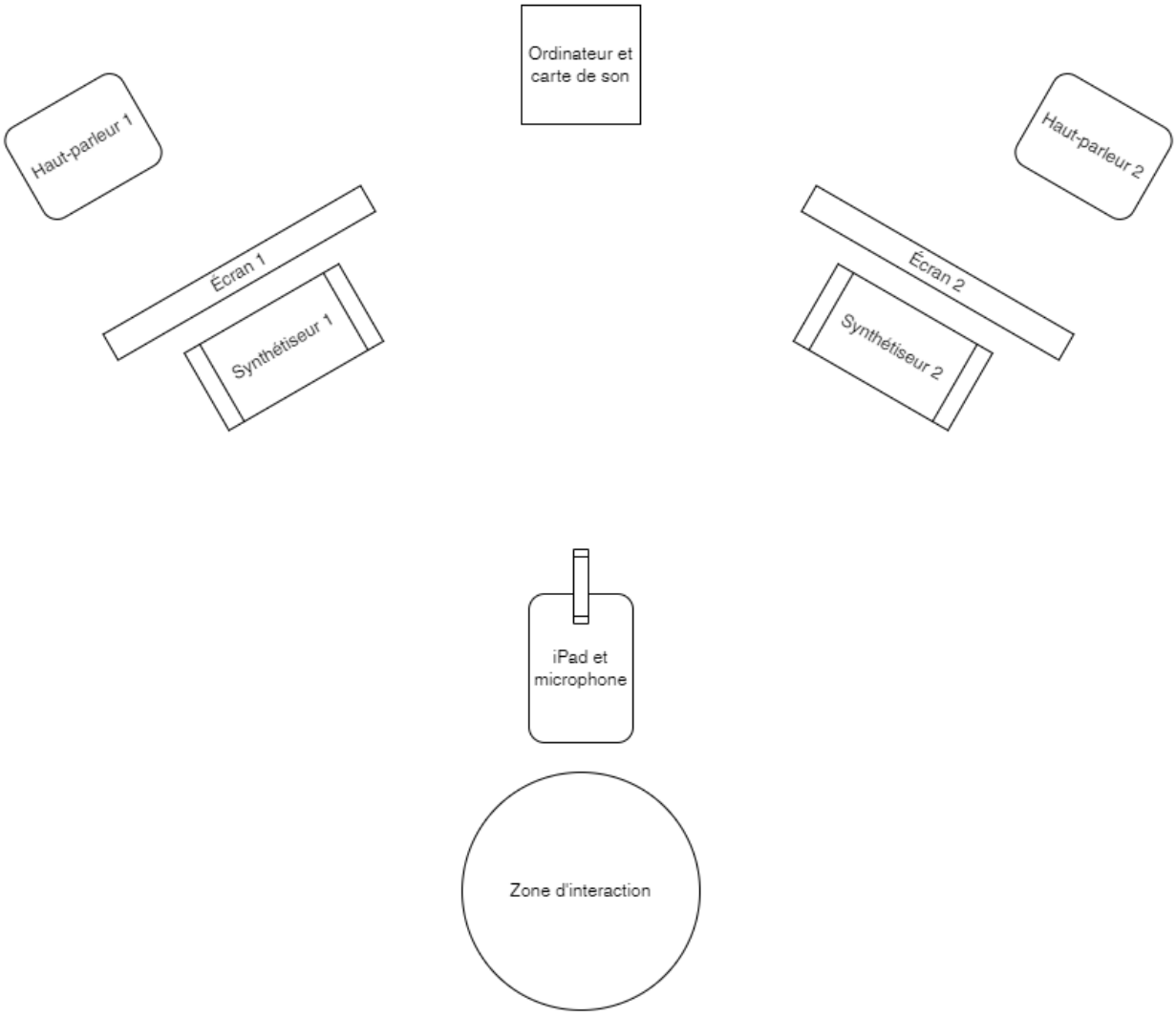

Figure 1 – *Deep Duo* : vue de dessus de la disposition des éléments physiques

Le système derrière l'installation consiste en un programme Max[1] s'occupant du volet sonore et un programme TouchDesigner[2] s'occupant du volet visuel. Le système créé avec Max contient deux instances d'un même sous-programme contrôlant chacune l'un des deux synthétiseurs[3]. Ces synthétiseurs sont câblés et calibrés pour être en mesure de faire de la synthèse par modulation de fréquence (synthèse FM). La première instance Max contrôle les paramètres de synthèse du premier synthétiseur en générant du voltage de contrôle. Le son produit par ce synthétiseur est envoyé en entrée de la deuxième instance

[1] Un langage de programmation visuel pour la musique et le multimédia.
[2] Un langage de programmation visuel pour le contenu multimédia interactif.
[3] Lien vers le code sur le site d'hébergement GitHub : https://github.com/gabriel-lavoie-viau/DeepDuo

Max qui utilise des descripteurs sonores pour analyser le signal sonore entrant et en extraire des données. Ces données sont formatées puis mappées vers les contrôles du deuxième synthétiseur. La sortie audio de ce deuxième synthétiseur est à son tour envoyé en entrée de la première instance et ainsi de suite (fig. 2). Les données extraites du signal sonore et les contrôles des synthétiseurs étant multiples, le mappage qui les met en relation sera appelé un mappage multiple (Marier, 2017, p.33).

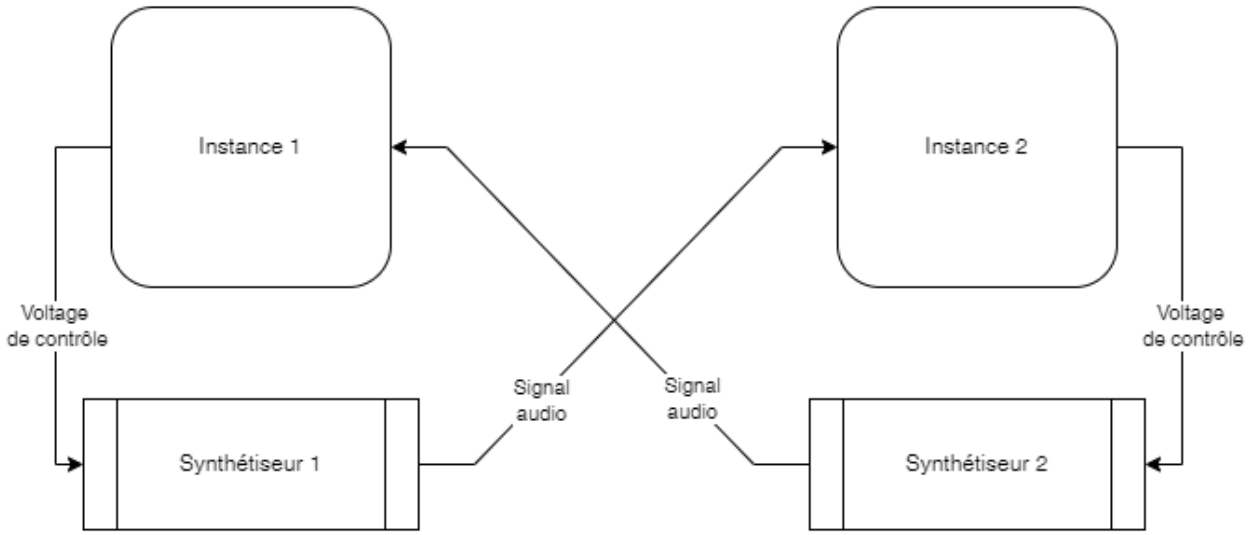

Figure 2 – *Deep Duo* : mise en boucle des deux instances

Cette mise en boucle des deux instances au sein du système crée un échange sous la forme d'un miroir musical où les deux sous-systèmes s'influencent et s'alimentent ad vitam aeternam. Le contenu de cette boucle peut être altéré par le signal en provenance du microphone, qui est envoyé en entrée des deux instances. Les données en sortie des descripteurs sonores générées par chacune des deux instances sont envoyées vers le programme TouchDesigner et servent à animer des formes géométriques affichées sur les écrans.

1.3. Concepts et outils d'apprentissage automatique en jeu

Deep Duo permet de contrôler un synthétiseur modulaire à partir d'un signal audio. Le contrôle des paramètres de ce synthétiseur se fait par le biais de voltage de contrôle. Étant donné qu'un signal audio ne fournit pas directement de valeurs pouvant être mises en

relation avec les différents paramètres de contrôle du synthétiseur, il semble raisonnable d'utiliser des analyses spectrales, timbrales et dynamiques pour extraire de ce signal des informations qui vivent dans le même paradigme que des paramètres de synthèse sonore, tels que des hauteurs de notes, des amplitudes ou des valeurs qui permettent de parler de timbre. Mettre en relation les valeurs produites par ces descripteurs et des valeurs de contrôle de voltage devient alors un problème de mappage. Étant donné la grande quantité de descripteurs pouvant être utilisés, un tel mappage est d'une grande complexité. L'utilisation d'un algorithme d'apprentissage machine permet de créer automatiquement et rapidement ce mappage multiple et donc d'expérimenter facilement avec différentes combinaisons de descripteurs.

Le système derrière l'installation *Deep Duo* met donc en place un réseau de neurones artificiels ayant pour fonction de créer un mappage multiple. Ce procédé est responsable de transformer les données extraites du signal audio entrant en valeurs de contrôle du synthétiseur modulaire. Les données en entrées sont produites par des analyses faites par une série de descripteurs audio spectraux, harmoniques et perceptuels capables de fournir des informations quant à l'amplitude, la hauteur, le timbre et le contenu harmonique du signal tel que ceux décrits par Peeters (2004). Plusieurs dizaines de ces types d'analyses peuvent être activées et mappées vers les quatre paramètres d'un programme de synthèse FM du synthétiseur modulaire. Les paramètres de cette synthèse, contrôlés par voltage de contrôle, sont la fréquence et l'amplitude de l'onde porteuse et la fréquence et l'amplitude de l'onde modulante.

C'est donc à travers ce mappage multiple dynamique entre des descripteurs audio et du voltage de contrôle que *Deep Duo* explore le fonctionnement des réseaux de neurones artificiels. Afin d'accompagner une description plus détaillée du système de l'installation et partager certains des apprentissages faits au cours de cette première année d'étude, la suite de ce chapitre tentera d'offrir certains éléments de compréhension des algorithmes d'apprentissage automatique. Leurs histoires et leurs fonctionnements sont particulièrement bien décrits par la littérature des dernières années (Alpaydin, 2020; Alzubi, Nayyar, et Kumar, 2018; Audry, 2021a; Das et Behera, 2017; Priddy et Keller, 2005; Ruder, 2016; Sharma, Sharma, et Athaiya, 2017; Thibodeau-Laufer, 2014).

1.3.1. Pourquoi les machines apprennent-elles?

Les origines de l'apprentissage automatique remontent aux premiers jours de la cybernétique au cours des années 1940. Le terme *Machine Learning* a été utilisé pour la première fois en 1959 par Arthur L. Samuel qui le définit comme un champ d'études qui donne aux ordinateurs la possibilité d'apprendre sans être programmés explicitement. Il fut le premier à proposer un modèle d'apprentissage fonctionnel. C'est aussi pendant cette période que Frank Rosenblatt inventa le Perceptron, un algorithme d'apprentissage à la base des réseaux de neurones artificiels, que l'on connaît aujourd'hui comme une technique fondamentale de l'apprentissage automatique contemporain.

L'intérêt soulevé par ce type de recherches eut un deuxième souffle pendant les années 1980 grâce au connexionnisme, une approche des sciences cognitives et de l'intelligence artificielle qui utilise des modèles mathématiques basés sur des versions simplifiées des réseaux de neurones se trouvant dans le cerveau humain. Plus récemment, la réémergence de l'apprentissage automatique est contextuelle:

> Its emergence since the beginning of the millennium is inseparable from the increased access to raw computing power—in particular, due to the development of graphical processing units (GPUs) incidentally pushed by the game and cinema industries—and the exponential growth of data, thanks to the explosive expansion of the internet as a platform for mass social media. (Audry, 2021c, p.4)

L'apprentissage automatique permet d'accomplir des tâches allant au-delà de la capacité d'un humain, tel que l'analyse d'immenses quantités de données complexes. Elle permet aussi à l'ordinateur d'accomplir des tâches qui seraient difficiles, voire impossibles, de traduire en une série de règles logiques, telles que reconnaître des objets, se déplacer ou encore traduire un langage vers un autre. Effectivement, une partie du savoir que nous possédons est encodé implicitement par notre intelligence. Nous n'avons donc pas nécessairement accès explicitement aux concepts qui permettraient de traduire ce savoir en instructions lisibles par un ordinateur, tel que nous le faisons avec la programmation procédurale. L'apprentissage automatique propose de laisser l'ordinateur apprendre de manière incrémentale à partir de données, comme nous le faisons par l'expérience de la réalité, et d'y découvrir lui-même des structures sous-jacentes.

1.3.2. Comment les machines apprennent-elles?

Avant d'observer l'approche générale de l'apprentissage automatique, il peut être intéressant de faire la distinction entre ce dernier, l'intelligence artificielle et

l'apprentissage profond. L'intelligence artificielle peut être considérée comme un vaste champ de recherche qui inclut plusieurs approches différentes, dont l'apprentissage automatique. L'apprentissage profond, quant à lui, est une branche particulière de l'apprentissage automatique dont les systèmes sont basés sur des réseaux de neurones artificiels.

De manière générale, les algorithmes d'apprentissage servent à effectuer des prédictions. Pour y arriver, ces algorithmes sont mis en relation avec des ensembles de données à partir desquelles ils entraînent automatiquement des modèles. Une fois ces modèles entraînés, il est possible de leur fournir des données en entrée en fonction desquelles ils génèrent des données en sortie, appelées prédictions, qui ont une cohérence par rapport à l'organisation des données ayant servi à l'entraînement. Ces algorithmes doivent donc être mis en relation avec un ensemble de données sur la base desquelles se fera l'apprentissage.

Les algorithmes d'apprentissage automatique peuvent être classés en grandes familles en fonction de leurs styles d'apprentissages. Le nombre de ces familles varie selon les auteurs (Ray, 2019; Alzubi, Nayyar, et Kumar, 2018; Das et Behera, 2017; Nasteski, 2017). Par souci de simplicité, je parlerai ici de deux grandes familles : les algorithmes d'apprentissage supervisé et les algorithmes d'apprentissage non supervisé. Ces deux familles englobent les algorithmes utilisés par mes travaux de recherche-création[4].

Dans le cas de l'apprentissage supervisé, les données servant lors de la phase d'entraînement des modèles sont étiquetées. C'est-à-dire que, en amont de la phase d'apprentissage, les données qui seront envoyées en entrée du système sont associées à des valeurs en sortie. Les données en entrée sont les caractéristiques et les données en sortie les étiquettes. L'objectif d'un tel système est que le modèle résultant de l'entraînement soit en mesure de faire de bonnes prédictions. C'est-à-dire qu'il fournit les bonnes valeurs lorsqu'il est exposé à des données qui n'étaient pas présentes lors de l'entraînement.

Un algorithme d'apprentissage non supervisé, quant à lui, est entraîné avec des données qui ne sont pas étiquetées. Dans ce cas, l'entraînement produira un modèle qui aura découvert des structures ou des motifs présents dans les données d'entraînement. Comme

[4] D'autres familles d'apprentissage mentionnées dans la littérature sont, par exemple, l'apprentissage par renforcement, l'algorithmie évolutionniste et l'apprentissage semi-supervisé.

dans le cas des algorithmes d'apprentissage supervisé, différents types de tâches et de résultats peuvent être attendus de ces modèles. La nature des données, le type d'algorithme d'apprentissage choisi, la configuration et le paramétrage de cet algorithme feront varier les valeurs en sortie du modèle.

Les trois éléments principaux des systèmes d'apprentissage automatique sont les données, le processus d'entraînement et le modèle. Les données peuvent être vues comme le savoir auquel le modèle a accès. Ultimement, ce savoir sera contenu à l'intérieur du modèle qui, pour être efficace, doit être assez complexe pour représenter fidèlement les données d'entraînement sans être trop précis. Autrement, le modèle ne sera pas en mesure de fournir de bons résultats, lorsqu'exposé à de nouvelles données. Le processus d'entraînement intervient entre les données et le modèle. Ce processus utilise les données pour ajuster le modèle de manière itérative en se fiant sur une fonction d'évaluation qui mesure la performance du modèle.

Le modèle est en quelque sorte le produit du processus d'entraînement. Une fois ce processus accompli, le modèle est autonome. Il n'est plus nécessaire d'avoir en main les données puisque le modèle est en quelque sorte une version compacte de ces données capable de faire des prédictions.

1.4. Le système de l'installation *Deep Duo*

Le système de l'installation *Deep Duo* peut être décrit comme une structure facilitant l'utilisation d'un réseau de neurones dans le cadre d'une tâche spécifique. Effectivement, le système est un programme Max construit autour d'un script, codé précédemment pour les besoins de l'installation, qui permet l'utilisation de réseaux de neurones artificiels. L'ensemble du système sert à :

- Créer une base de données constituée de descripteurs audio.
- Sélectionner les éléments d'entraînement dans la base de données.
- Définir l'architecture et les paramètres d'entraînement du réseau de neurones.
- Entraîner le réseau de neurones à partir des éléments sélectionnés dans la base de données.
- Faire des demandes de prédictions au modèle.

Le programme Max est donc un algorithme faisant le pont entre le réseau de neurones et le synthétiseur modulaire. Il facilite les étapes d'utilisations du modèle dans le cadre de son utilisation pour les besoins spécifiques de *Deep Duo*.

Dans cette section, le fonctionnement général du système sera décrit. Cette description permettra de comprendre plus en détail son architecture ainsi que les différentes étapes nécessaires à l'entraînement et la demande de prédiction d'un réseau de neurones artificiels.

1.4.1. Création de la base de données

Comme c'est le cas pour tous les algorithmes d'apprentissage automatique, il est nécessaire d'avoir accès à des données à partir desquelles le modèle sera entraîné. *Deep Duo* utilise un algorithme d'apprentissage supervisé. Ces données doivent donc être étiquetées de telle sorte que les données en entrée soient associées à des données en sortie ou, autrement dit, aux résultats attendus. Dans le cas présent, l'objectif est de cartographier les possibilités sonores d'un programme de synthèse FM fait au synthétiseur modulaire. Pour ce faire, les valeurs de plusieurs descripteurs qui analysent la sortie audio du synthétiseur seront associées avec les valeurs des paramètres de la synthèse FM correspondantes. Ainsi, la base de données contiendra des listes de valeurs de descripteurs audio, telles que la quantité d'énergie ou de bruit contenu dans le signal, associées aux valeurs de contrôle qui permettent de générer le signal correspondant. L'objectif est d'entraîner un modèle capable de produire des valeurs de voltage de contrôle à partir de valeurs d'analyse de n'importe quel signal audio. Ainsi, n'importe quelle source sonore pourra devenir un moyen de contrôler le synthétiseur modulaire par le biais du modèle.

Une section du programme Max sert à générer les données d'entraînement. Pour ce faire, un processus algorithmique permet de parcourir de manière systématique l'ensemble des combinaisons possibles des quatre paramètres de la synthèse FM du synthétiseur. L'étendue du voltage de contrôle de chaque paramètre est divisée en 10 échelons, ce qui fait donc 10 000 combinaisons différentes (10^4). La sortie audio du synthétiseur est récupérée par le système et 45 descripteurs audio différents analysent le signal de chaque combinaison des quatre paramètres. L'ensemble de ces valeurs sont emmagasinées dans une base de données sous la forme de 10 000 exemples contenant chacun 45 données en

entrée (les descripteurs) associées à 4 données en sortie (les valeurs de voltage de contrôle). Théoriquement, ce processus pourrait servir à cartographier d'autres types de synthèse faite à partir d'un synthétiseur modulaire, tel que de la synthèse additive. La figure 3 (fig. 3) permet de visualiser un des exemples contenus dans la base de données ayant servi à l'entraînement d'une des deux instances de *Deep Duo*. Les descripteurs en entrée du réseau sont regroupés dans le groupe *features* et les valeurs de voltage de contrôle en sortie sous le groupe *labels*.


```
"step_7115" :   {
    "features" :        {
        "SignalZeroCrossingRate" : 1590.35089111328125,
        "TotalEnergy" : 0.0142563120462,
        "SpectralCentroid" : 5221.3408203125,
        "SpectralSpread" : 3113.8270263671875,
        "SpectralSkewness" : -0.042748931795359,
        "SpectralKurtosis" : 1.930769205093384,
        "SpectralRolloff" : 9808.3740234375,
        "SpectralVariation" : 0.110993832349777,
        "SpectralDecrease" : -0.016533663962036,
        "Loudness" : 4.745449542999268,
        "RelativeSpecificLoudness0" : 0.029040585272014,
        "RelativeSpecificLoudness1" : 0.027399607934058,
        "RelativeSpecificLoudness2" : 0.02920282073319,
        "RelativeSpecificLoudness3" : 0.031616461463273,
        "RelativeSpecificLoudness4" : 0.033156210556626,
        "RelativeSpecificLoudness5" : 0.035878591239452,
        "RelativeSpecificLoudness6" : 0.036939593032002,
        "RelativeSpecificLoudness7" : 0.037195812910795,
        "RelativeSpecificLoudness8" : 0.037762138992548,
        "RelativeSpecificLoudness9" : 0.03802452981472,
        "RelativeSpecificLoudness10" : 0.036496238783002,
        "RelativeSpecificLoudness11" : 0.032208457589149,
        "RelativeSpecificLoudness12" : 0.030935046263039,
        "RelativeSpecificLoudness13" : 0.033779941499233,
        "RelativeSpecificLoudness14" : 0.039209568873048,
        "RelativeSpecificLoudness15" : 0.043914860114455,
        "RelativeSpecificLoudness16" : 0.047205425798893,
        "RelativeSpecificLoudness17" : 0.050968099385,
        "RelativeSpecificLoudness18" : 0.052860958501697,
        "RelativeSpecificLoudness19" : 0.05308199301362,
        "RelativeSpecificLoudness20" : 0.05416720174253,
        "RelativeSpecificLoudness21" : 0.057464718818665,
        "RelativeSpecificLoudness22" : 0.062418146058917,
        "RelativeSpecificLoudness23" : 0.063275564461946,
        "PerceptualTristimulus0" : 0.018292037770152,
        "PerceptualTristimulus1" : 0.056872537359595,
        "PerceptualTristimulus2" : 0.924835413694382,
        "PerceptualOddToEvenRatio" : 0.941860467195511,
        "Sharpness" : 3.072569012641907,
        "Spread" : 0.87566351890564,
        "SpectralFlatness0" : 0.984097123146057,
        "SpectralFlatness1" : 0.99543759226799,
        "SpectralFlatness2" : 0.922639548778534,
        "SpectralFlatness3" : 0.948035329580307,
        "SpectralCrest0" : 1.369402527809143,
        "SpectralCrest1" : 1.175360798835754,
        "SpectralCrest2" : 1.409007549285889,
        "SpectralCrest3" : 1.494779706001282,
        "SpectralSlope" : -0.000000028152922,
        "Chroma0" : 0.480507299304008,
        "Chroma1" : 0.48307178914547,
        "Chroma2" : 0.459384799003601,
        "Chroma3" : 0.420514345169067,
        "Chroma4" : 0.518180072307587,
        "Chroma5" : 1.0,
        "Chroma6" : 0.460431784391403,
        "Chroma7" : 0.600452661514282,
        "Chroma8" : 0.327712625265121,
        "Chroma9" : 0.529097884893417,
        "Chroma10" : 0.48779085278511,
        "Chroma11" : 0.31853061914444,
        "MFCC0" : -2.118028104305267,
        "MFCC1" : 0.235790953040123,
        "MFCC2" : -0.62829378247261,
        "MFCC3" : -0.66468420624733,
        "MFCC4" : 0.103400330990553,
        "MFCC5" : 0.334931924939156,
        "MFCC6" : -0.208163686096668,
        "MFCC7" : -0.104319583624601,
        "MFCC8" : 0.139253258705139,
        "MFCC9" : 0.124770224094391,
        "MFCC10" : 0.018744978122413,
        "MFCC11" : -0.071155995130539,
        "MFCC12" : 0.075691137462854,
        "PerceptualSpectralDeviation" : 0.017586495727301,
        "PerceptualSpectralCentroid" : 4519.138427734375,
        "PerceptualSpectralSpread" : 2900.4873046875,
        "PerceptualSpectralSkewness" : 0.177679643034935,
        "PerceptualSpectralKurtosis" : 2.027081072330475,
        "PerceptualSpectralRolloff" : 9776.07421875,
        "PerceptualSpectralVariation" : 0.028052538633347,
        "PerceptualSpectralDecrease" : 0.045223413035274,
        "PerceptualSpectralSlope" : 0.000007319140195,
        "FundamentalFrequency" : 0.0,
        "Inharmonicity" : 0.0,
        "HarmonicEnergy" : 0.0,
        "NoiseEnergy" : 0.0142563120462,
        "Noisiness" : 1.0,
        "HarmonicTristimulus0" : 0.0,
        "HarmonicTristimulus1" : 0.0,
        "HarmonicTristimulus2" : 0.0,
        "HarmonicOddToEvenRatio" : 0.0,
        "HarmonicSpectralDeviation" : 0.0,
        "HarmonicSpectralCentroid" : 0.0,
        "HarmonicSpectralSpread" : 0.0,
        "HarmonicSpectralSkewness" : 0.0,
        "HarmonicSpectralKurtosis" : 0.0,
        "HarmonicSpectralVariation" : 0.0,
        "HarmonicSpectralDecrease" : 0.0,
        "HarmonicSpectralSlope" : 0.0,
        "HarmonicSpectralRolloff" : 0.0,
        "Frequency" : 2164.026611328125,
        "Energy" : 0.012273068074137,
        "Periodicity" : 0.07207390293479,
        "AC1" : 0.639056593179703
    }
,
    "labels" :      {
        "cv_out_4" : 0.555555555555556,
        "cv_out_3" : 0.111111111111111,
        "cv_out_2" : 0.111111111111111,
        "cv_out_1" : 0.777777777777778
    }
```


Figure 3 – Un exemple contenu dans une base de données de *Deep Duo*.

1.4.2. Sélection des données d'entraînement

Ce ne sont pas nécessairement toutes les valeurs contenues dans une base de données qui sont pertinentes pour l'entraînement d'un modèle d'apprentissage. En fonction des caractéristiques que l'on veut que le modèle prenne en compte, il est possible qu'il soit désirable de filtrer une partie de ces données. Le système permet de faire une sélection parmi les 45 descripteurs disponibles pour chaque exemple présent dans la base de données. La figure 4 (fig. 4) fait voir l'interface du système qui permet de faire cette sélection. Ces descripteurs sont divisés en plusieurs familles telles que des descripteurs spectraux, perceptuels et harmoniques. Les descripteurs qui seront sélectionnés pour l'entraînement influenceront la nature du modèle. Par exemple, si aucun descripteur lié à la quantité d'énergie présente dans le signal n'est sélectionné, le modèle, une fois entraîné, ne tiendra probablement pas compte des variations d'amplitude du signal en entrée. Il est donc possible à ce stade d'influencer la sensibilité acoustique qu'aura le modèle final.

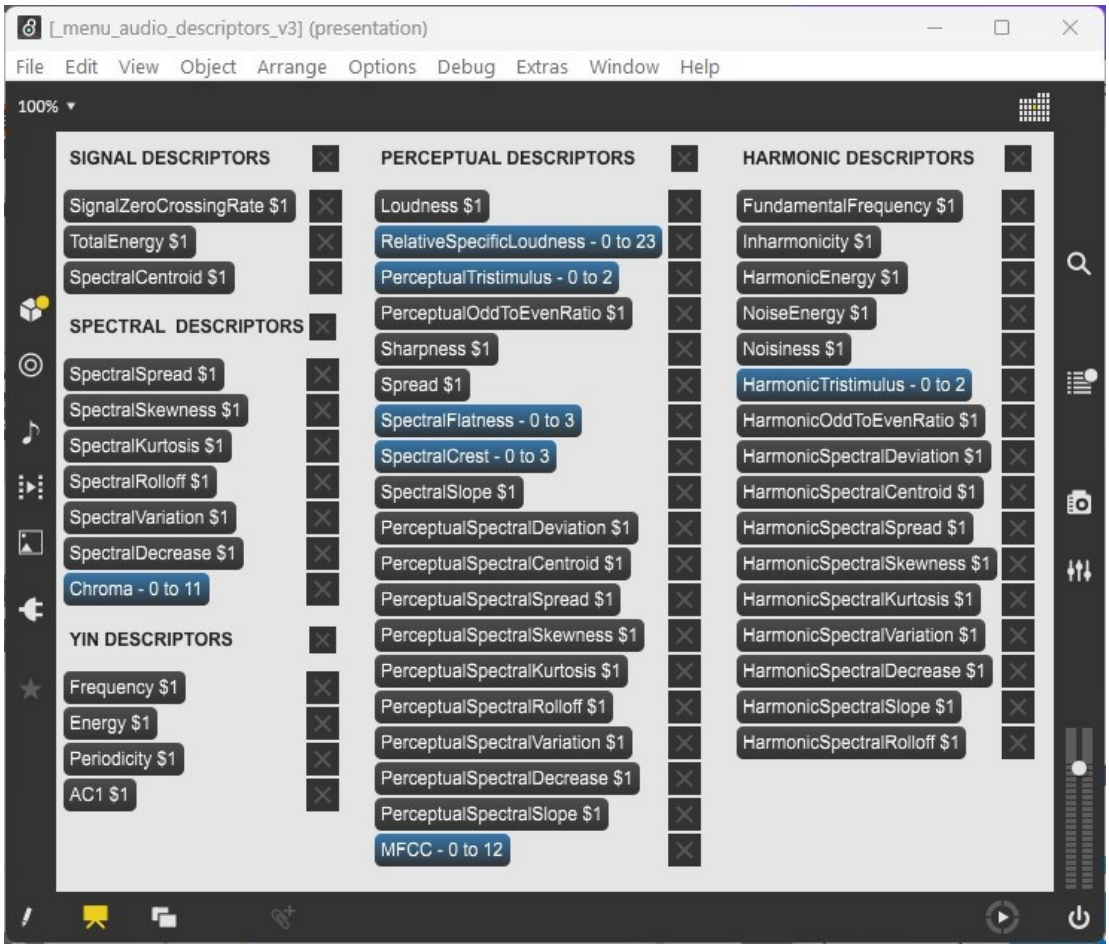

Figure 4 – *Deep Duo* : menu de sélection des descripteurs audio.

Pour l'exemple, imaginons que les coefficients cepstraux de fréquence Mel (*Mel-Frequency Cepstral Coefficients* ou MFCC) sont sélectionnés, un descripteur audio qui, avec peu de valeurs, arrivent à représenter le profil spectral d'un signal sonore (Peeters, 2004, p.16). Dans le cas présent, le descripteur MFCC génère 13 coefficients. Pour chacun des 10 000 exemples de la base de données, 13 données en entrées seront donc conservées (les MFCC) associées à quatre données en sortie (les voltages de contrôle). Ce nouvel ensemble de valeurs sera appelé les données d'entraînement. Éventuellement, lorsque le modèle sera entraîné, des valeurs de MFCC en provenance de n'importe quelle source sonore pourront être envoyées en entrée du modèle pour qu'il prédise les voltages de contrôle qui reproduiront le plus fidèlement possible le spectre entrant.

1.4.3. Définition de l'architecture et des paramètres du réseau de neurones

L'architecture du réseau neuronal et ses paramètres d'entraînements doivent aussi être précisés. Ceux-ci auront un impact significatif sur la manière dont le modèle se comportera. Mais tout d'abord, qu'est-ce qu'un réseau de neurones artificiels? Initialement, les systèmes de ce genre s'inspiraient du fonctionnement des neurones biologiques. Mais dans leurs formes actuelles, ils s'apparentent plus à des méthodes statistiques. Un réseau de neurones est organisé en couches successives qui contiennent toutes un nombre variable de neurones. Minimalement, le réseau contient une couche d'entrée et une couche de sortie. Entre ces deux couches, il peut y avoir un nombre variable de couches cachées, c'est-à-dire des couches qui ne connectent pas avec l'extérieur du réseau, contenant chacune un nombre variable de neurones. L'ajout de couches cachées permet au réseau d'avoir une réponse non linéaire aux données en entrées contrairement à un réseau qui contiendrait uniquement une couche d'entrée et une couche de sortie. Chaque neurone présente dans le réseau connecte avec toute les neurones de la couche suivante et est associé à un poids et un seuil qui influencent les données qu'elle produira en sortie. Le processus d'entraînement consiste à ajuster le poids et le seuil des neurones du réseau. C'est ainsi que le comportement de ce dernier peut s'adapter à différentes bases de données et accomplir différentes fonctions.

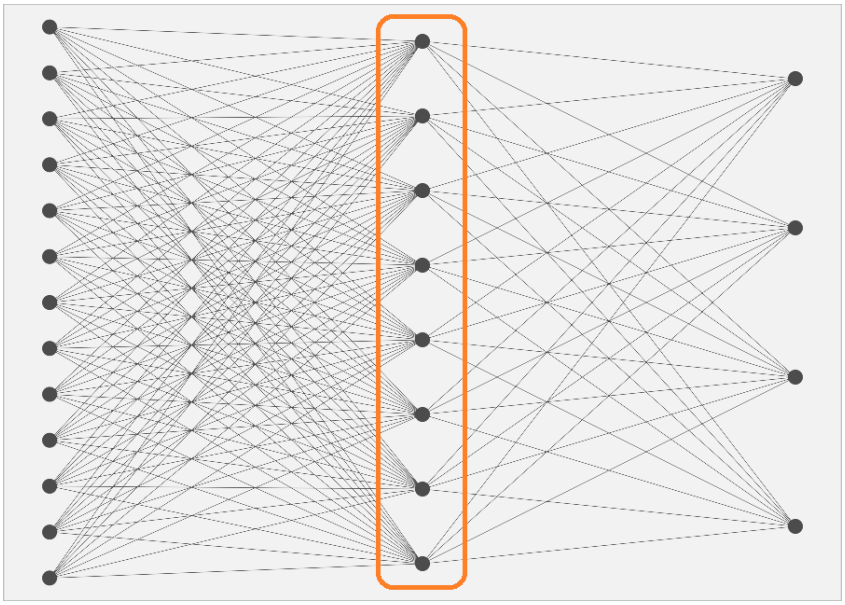

Figure 5 – Diagramme d'un réseau de neurones : la couche cachée est encadrée en orange.

Pour satisfaire les besoins de l'exemple, 13 neurones pour la couche en entrée, correspondant aux 13 coefficients du descripteur MFCC, et 4 neurones en sortie, correspondant au voltage de contrôle des quatre paramètres de la synthèse FM, seront nécessaires. Entre les deux sera placée une couche cachée contenant huit neurones. La figure 5 (fig. 5) permet de voir un diagramme de cette configuration. Il n'y a pas de mode d'emploi qui permet de déterminer le nombre optimal de couches cachées et leurs nombres de neurones respectifs. Une des approches envisageables est de procéder de manière empirique en entraînant plusieurs réseaux de configuration différente puis en comparant leurs degrés de précision.

À travers une interface affichée sur l'iPad (fig. 6), le système permet aussi de modifier les paramètres d'entraînement. Ces paramètres auront une incidence sur la durée que prendra l'entraînement ainsi que sur le degré de précision et la réponse du modèle résultant. Voici une courte définition de certains des paramètres importants :

- Taille de lot : le nombre d'exemples utilisé pour chaque itération de l'ajustement du modèle. Effectivement, l'ensemble des données d'entraînement peut être divisé en plusieurs lots de taille égale qui seront utilisés successivement lors de l'entraînement. Ce procédé permet au processus d'entraînement d'être plus stable et efficace.

- Nombre d'époques : le nombre de fois que l'entièreté des données est présentée à l'algorithme d'apprentissage. Une époque se termine à chaque fois que l'ensemble des lots a été présenté à l'algorithme.
- Taux d'apprentissage : un facteur utilisé pour ajuster la vitesse à laquelle le poids et le seuil des neurones seront ajustés pendant l'entraînement. Ce facteur influence directement le temps que prendra une époque avant d'être complétée.
- Fonction de perte : le choix de la fonction mathématique qui sera responsable de calculer la différence entre le résultat espéré et le résultat fourni par le modèle. La valeur résultante de cette fonction permet de savoir à quel point le modèle apprend quelque chose de l'entraînement.

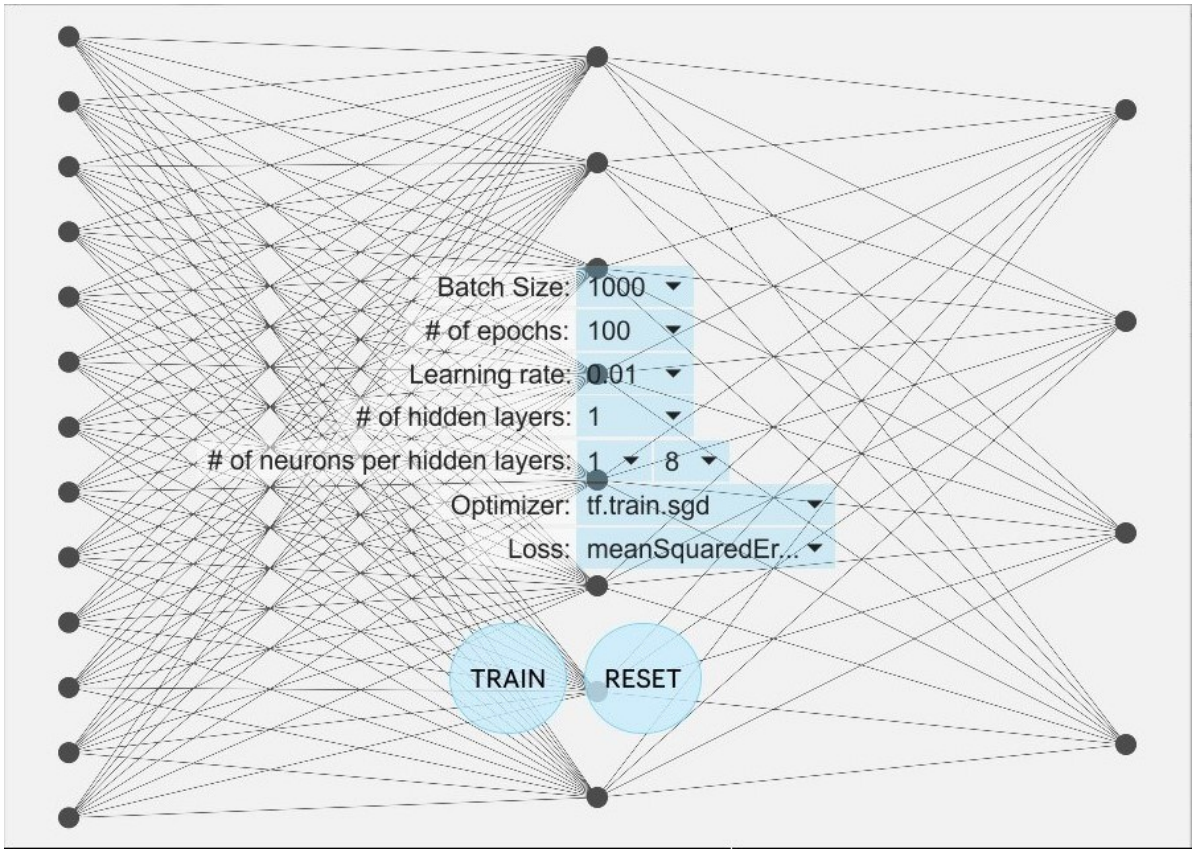

Figure 6 – *Deep Duo* : image de l'interface de l'iPad permettant de modifier les paramètres d'entraînements.

Dans un contexte d'utilisation typique d'un réseau neuronal, la sélection des données d'entraînement, le choix de l'architecture du réseau ainsi que l'ajustement des paramètres d'entraînement ont comme objectif d'optimiser la précision du modèle. Dans le cadre d'un

projet artistique comme *Deep Duo*, les critères d'évaluation du modèle sont plutôt subjectifs, voire esthétiques. Ainsi, la manipulation de ces différents paramètres devient un moyen de modifier la nature et le comportement sonore du système et d'en explorer les possibilités expressives. Si l'on considère le système de *Deep Duo* comme un instrument musical avec lequel on joue par le biais des sons que l'on émet, l'entraînement devient alors le processus par lequel cet instrument est créé. En conséquence, ces paramètres sont les outils qui permettent de déterminer le comportement de l'instrument.

1.4.4. Entraînement du réseau de neurones

À ce stade, l'algorithme d'apprentissage a en main tous les éléments dont il a besoin pour que l'entraînement du modèle débute.

Pendant l'entraînement, à la fin de chaque époque, le système donne accès à des informations qui font état du processus. Ces valeurs, notamment le numéro de l'époque et la valeur résultante de la fonction de perte, permettent à l'utilisateur d'avoir une vue d'ensemble sur le processus d'entraînement en cours et elles sont rendues accessibles aux interacteurs via un graphique se trouvant sur l'iPad. Le système continue d'être fonctionnel pendant l'entraînement, ce qui permet d'entendre en temps réel l'évolution du comportement sonore du modèle alors qu'il passe par ses différentes phases d'apprentissage.

1.4.5. Demande de prédictions

Les demandes de prédiction, comme mentionnée dans la section précédente, peuvent être faites en cours d'entraînement alors que le modèle est en train de prendre forme. Une fois l'entraînement terminé, la forme du modèle est cristallisée et son comportement restera le même. C'est-à-dire que pour une même liste de valeurs données en entrée, il produira toujours les mêmes valeurs en sortie. Le modèle s'attend à recevoir une liste de valeurs de la même taille que le nombre de données en entrée contenu dans chaque exemple des données d'entraînement. La longueur de cette liste correspond aussi au nombre de neurones de la couche d'entrée du modèle. De la même façon, le modèle produira toujours quatre valeurs en sortie.

Le modèle, une fois créé, peut être sauvegardé sur le disque sous la forme d'un fichier. Effectivement, une fois le modèle entraîné, il n'est plus nécessaire d'avoir les données ni

les paramètres de l'entraînement en main. Pour faire des prédictions, il suffit de charger un modèle et de lui envoyer des valeurs en entrée.

1.5. Conclusion : vers une mise en corps de l'interaction avec un réseau neuronal

L'installation *Deep Duo* met en scène le fonctionnement interne du système et permet à l'interacteur d'intervenir dans les différentes étapes de l'entraînement du réseau neuronal. Elle tente de révéler au public la nature d'un algorithme d'apprentissage automatique non pas par la voie de la raison mais plutôt par celle de l'intuition. Effectivement, l'intention derrière cette installation n'est pas de vulgariser les concepts de l'apprentissage automatique, mais, par le biais de la mise en corps d'une interaction avec un modèle d'apprentissage, de permettre à l'interacteur de développer une intuition de l'influence de ces différents concepts sur le comportement que le modèle peut exhiber. Son interaction avec le modèle passe par une mise en corps puisqu'il sent les variations de comportements du modèle par le biais d'un échange sonore qui impliquent la voix et l'écoute. L'œuvre, exposée au chapitre suivant, continue d'explorer cette mise en corps, mais cette fois en proposant des interactions avec des modèles d'une plus grande complexité. Les sonorités abstraites de *Deep Duo* seront abandonnées au profit de timbres vocaux et d'images de visages afin d'établir un lien plus intime entre l'œuvre et l'interacteur.

Chapitre 2 - *Morphogenèse* : installation audiovisuelle interactive

Morphogenèse, l'œuvre d'envergure de ce mémoire, est une installation interactive et audiovisuelle qui met en relation un interacteur avec différents modèles d'apprentissage automatique. Au fil de l'expérience, l'interacteur se trouve peu à peu confronté à l'image que la machine se fait de lui. Par un effet à la fois de familiarité et d'étrangeté, l'installation tente de rendre sensible le clivage qui existe entre l'image que l'on a de soi et celle renvoyée par les modèles d'apprentissage qui façonnent les différentes réalités numériques.

Le titre de l'installation est emprunté au travail de l'artiste et chercheur Sofian Audry. Ce dernier propose un cadre esthétique qui permet de qualifier les comportements et la morphologie d'agents adaptatifs présents au sein de certaines œuvres numériques (Audry, 2018). La *morphogenesis* y est définie comme le processus par lequel des comportements émergent de manière continue. L'aspect visuel de l'œuvre est une narration autour de cette catégorie de comportements.

Ce chapitre présentera d'abord l'installation et ses modalités d'interactions du point de vue d'un interacteur. Ensuite, une description physique et technique de l'installation sera faite. Puis, les parcours du signal sonore et visuel seront suivis à partir de leurs captations par des senseurs jusqu'à leurs sorties dans les haut-parleurs et sur l'écran, en passant par tous les processus actifs au sein du système de *Morphogenèse*. Finalement les manières et les mécanismes par lesquels l'expérience de l'interacteur évolue au fil de ses interactions avec l'installation seront exposés.

Ce chapitre est en grande partie descriptif et technique afin de mettre la table pour le chapitre trois où seront davantage exposées les stratégies créatives mises en place avec les différents algorithmes d'apprentissage automatique utilisé dans *Morphogenèse* ainsi que dans *Deep Duo*.

2.1. Scénario d'interactivité

Le document suivant, disponible en annexe, appuie ce scénario d'interactivité :

Extrait 2 - *Morphogenèse* : documentation vidéo.

J’entre dans une salle dénudée et sombre. Mon attention se porte sur un grand panneau accroché au plafond, sur lequel un immense visage est projeté. En face du panneau, un dispositif fixé sur un pied émet une faible lumière oscillante. Une voix légèrement synthétique crée des rythmes à travers des consonnes bruiteuses. Le visage, aux yeux fermés, a une apparence protéiforme où s’écoule lentement une multiplicité de traits.

Une lumière en douche souligne la présence d’un dispositif sur pied. La lumière oscillante qu’il émet invite à l’interaction. Lorsque je me place devant celui-ci, la lumière cesse d’osciller et s’éteint. Le visage projeté tombe en silence et ses yeux s’ouvrent. Lorsque j’émets un son, la lumière du dispositif allume, activée par l’amplitude de la voix, et le visage se met à imiter la mélodie et les articulations que je produis. Au fil des interactions, des mélodies émergent, portées par une succession de visages et de timbres vocaux, à travers un jeu sonore fait de répétitions, de transpositions et de déformations.

Peu à peu, je me rends compte que les traits projetés me renvoient des versions déformées de mon propre visage. J’ai l’impression d’être face à une version parallèle de moi-même dont l’âge et le genre sont en constantes mutations.

Lorsque je quitte la zone d’interaction, les notes formant les mélodies se dispersent. Pendant que le visage ferme les yeux, mes traits se fondent tranquillement dans le flot d’autres traits où surgissent les visages des visiteurs passés. L’installation retourne à sa phase initiale : des sons rythmiques sans tonalité et un visage en constante métamorphose.

2.2. Description physique et technique

Dans cette section se trouve un survol des principaux éléments physiques et numériques ainsi qu’une explication de la manière dont ces éléments sont organisés entre eux.

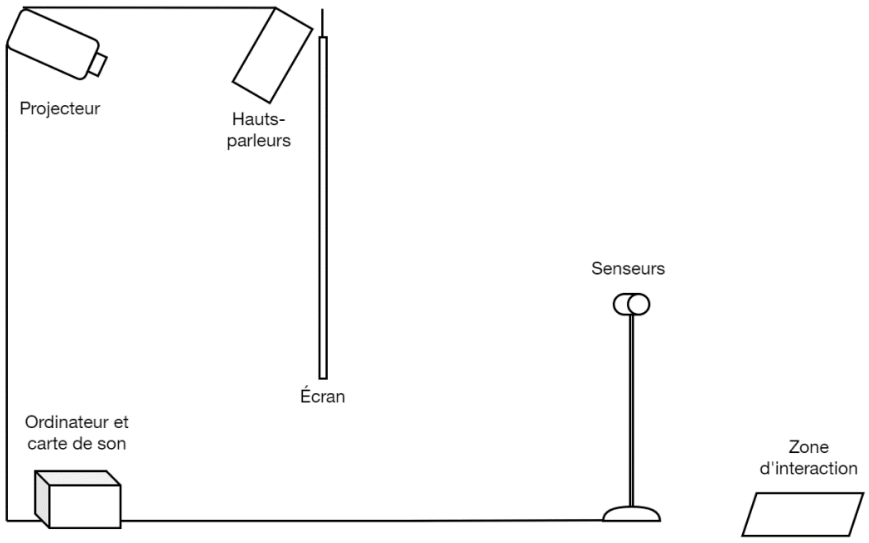

Figure 7 – *Morphogenèse* : plan de coupe de l'installation.

L'installation requiert une surface d'environ six mètres sur trois mètres où se trouvent un espace d'interaction, des senseurs sur pied, une surface de projection de forme carrée, un projecteur à focale courte, une paire de haut-parleurs préamplifiées ou d'écouteurs, un ordinateur et une interface audio.

Les senseurs consistent en un microphone condensateur unidirectionnel et une caméra. Ils sont placés sur un pied à environ 50 centimètres de la zone d'interaction de manière à faire face au visage de l'interacteur. Les données générées par ces senseurs sont envoyées vers l'ordinateur. La figure 7 (fig. 7) permet de voir l'ensemble de ces éléments disposés dans un plan de coupe de l'installation.

Le système numérique qui fait fonctionner l'installation peut être divisé en cinq sections principales[5] (fig. 8) :

- Un système de génération mélodique.
- Un système de génération sonore.
- Un système de génération visuelle.
- Un système de gestion du signal audio.
- Un système de contrôle central qui reçoit et envoie des informations aux quatre autres systèmes et, ainsi, orchestre le déroulement de l'expérience.

[5] Lien vers le code sur le site d'hébergement GitHub: https://github.com/gabriel-lavoie-viau/Morphogenese

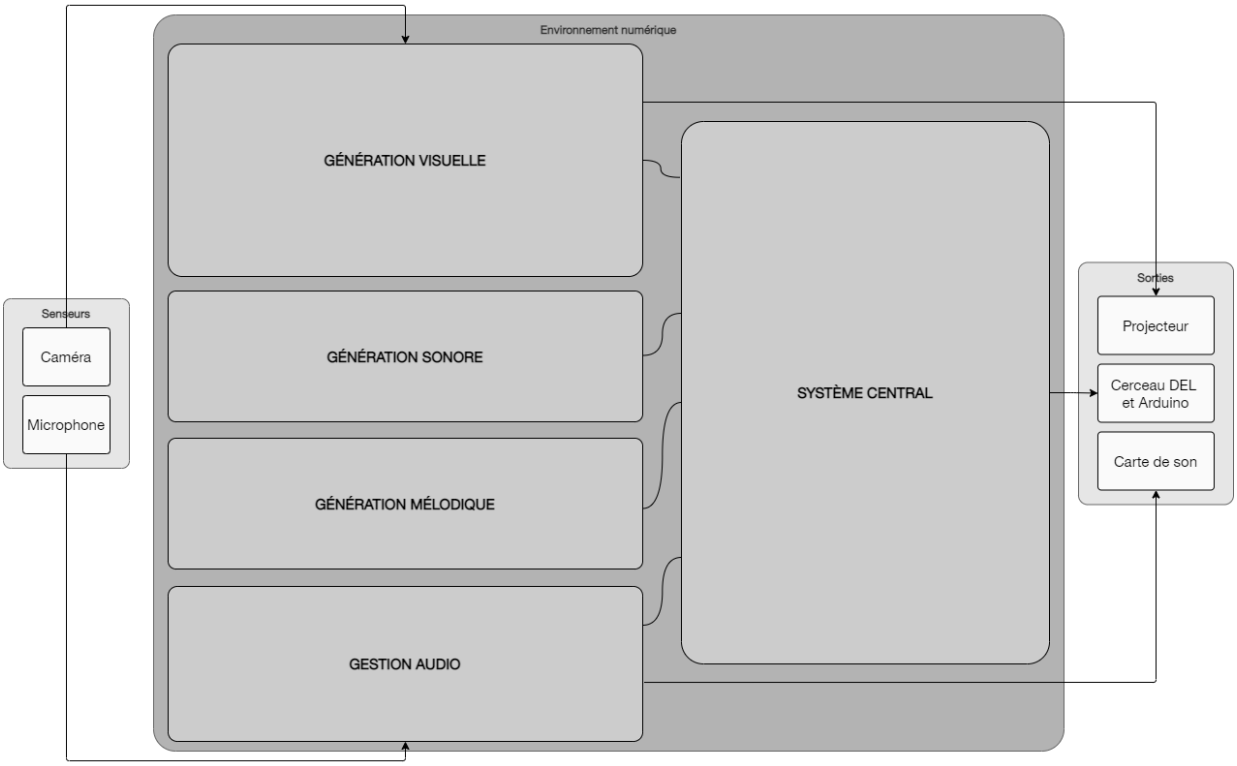

Figure 8 – *Morphogenèse* : vue d'ensemble du système.

L'image en provenance de la caméra est d'abord envoyée à un algorithme de détection faciale qui recadre l'image autour du visage et l'enregistre sur le disque. L'algorithme fournit aussi des informations sur le nombre de visages détecté et le nombre de secondes écoulé depuis le début de la détection en cours. Ces informations sont envoyées vers le système de contrôle central.

À chaque intervention sonore de l'interacteur, le système de génération faciale produit deux images : un visage généré aléatoirement et un visage généré à partir de celui de l'interacteur. Le visage projeté à l'écran est le résultat d'une interpolation entre ces deux visages. Au temps zéro de l'interaction, l'interpolation penche complètement du côté du visage aléatoire. Plus l'interacteur fait d'interventions sonores, plus l'interpolation penche du côté du visage encodé. Éventuellement, l'interpolation penche complètement du côté de l'image encodée du visage de l'interacteur. Une fois arrivés à cette étape, des procédés algorithmiques semi-aléatoires font transiter les traits du visage encodé sur des axes qui font varier l'âge et le genre du visage.

Le son en provenance du microphone est d'abord envoyé vers un système d'analyse sonore. Si le système considère qu'une interaction est en cours et que le son reçut dépasse

un seuil d'amplitude prédéterminé, ce dernier est enregistré et un message est envoyé vers le système central pour indiquer qu'un nouvel enregistrement a été effectué.

Lorsqu'un enregistrement est en cours, la valeur d'analyse d'amplitude sert aussi à faire varier l'intensité lumineuse d'un cerceau de lumière placé sur l'avant de la caméra. Cet effet sert de rétroaction à l'interacteur et a pour objectif de lui indiquer que les sons qu'il produit sont perçus et pris en compte par le système.

Chaque nouvel enregistrement sonore passe par un système d'analyse en sortie duquel sont obtenues plusieurs listes de valeurs. Une de ces listes, contenant des hauteurs de notes, est envoyée vers le générateur de mélodie et lui sert de point de départ pour la génération d'une nouvelle mélodie. Cette mélodie, générée artificiellement par un modèle d'apprentissage, est retournée vers le système central, responsable de gérer les données envoyées au générateur vocal. Le générateur utilise ces données pour produire des fichiers sonores de voix chantée. Ces fichiers sont enregistrés sur le disque et un message est envoyé au système central qui déclenche alors leurs lectures. Un suivi d'enveloppe sur le son en sortie du système permet de faire bouger la bouche du visage à l'écran.

2.3. Parcours du signal sonore

Cette section permettra de détailler davantage le fonctionnement de l'aspect sonore de l'installation. Le parcours du signal sonore sera suivi à partir de sa captation par le microphone, en passant par la génération mélodique et sonore, jusqu'à sa sortie par les haut-parleurs (fig. 9). Les contrôles exposés qui ont permis de créer l'atmosphère et la musique de *Morphogenèse* seront détaillés.

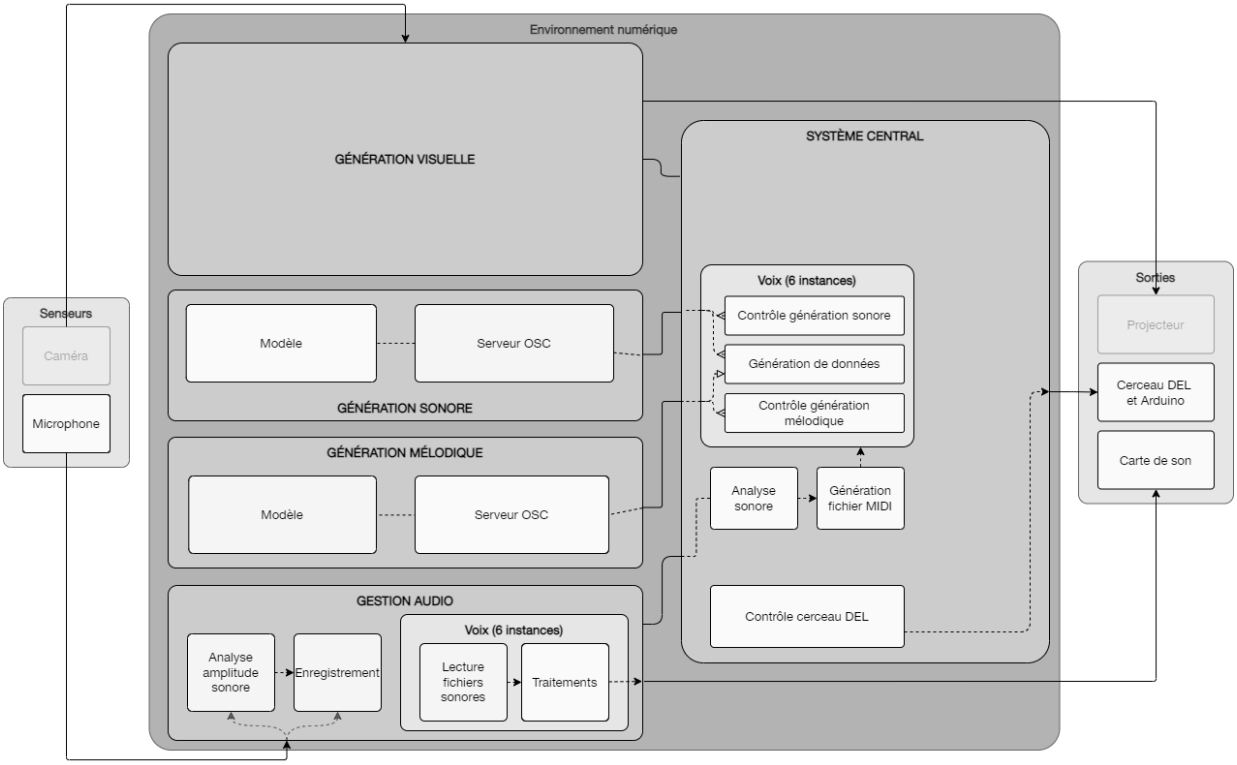

Figure 9 – *Morphogenèse* : parcours du signal sonore.

2.3.1. Signal en entrée

Le microphone utilisé est un condensateur de type hypercardioïde qui permet de capter les sons, même ténus, des interacteurs, mais qui permet d'éviter de capter les sons générés par l'installation. Le signal du microphone est récupéré par un programme Pure Data[6]. Ce programme s'occupe de gérer l'ensemble du signal audio entrant et sortant du système et est en communication avec le système central. Une porte de bruit permet de déterminer si un son est produit dans la zone d'interaction. Si tel est le cas, le programme enregistre le son et avise le système central. En cours d'enregistrement, une analyse d'amplitude est faite sur le signal sonore. Les valeurs de cette analyse servent à faire varier l'intensité lumineuse du cerceau de lumière.

2.3.2. Génération de l'amorce mélodique

Au cœur du système de génération mélodique de l'installation se trouve un algorithme d'apprentissage automatique appelé MelodyRNN créé par le projet de recherche open source Magenta. Cet algorithme applique la modélisation de langage à la génération de

[6] Un langage de programmation visuelle pour la musique et le multimédia.

mélodie avec un réseau de neurones de type *long short-term memory* (LSTM). Le modèle utilisé par le système est un des modèles préentraînés disponibles à partir de la plateforme GitHub[7] du projet. Tel que décrit par Waite (2016), ce modèle, appelé AttentionRNN, est conçu de manière à identifier et à s'inspirer de structures mélodiques relativement longues. Le modèle prend en entrée une amorce, sous la forme d'une suite de notes contenue dans un fichier au format *Musical Instrument Digital Interface* (MIDI), de laquelle il s'inspire pour générer des mélodies de longueurs variables.

Dans le cadre de l'installation, ce sont les interventions des interacteurs qui servent à créer les amorces mélodiques. Lorsque le système central est avisé qu'un nouvel enregistrement sonore a été effectué, il l'analyse pour en extraire des hauteurs de notes qui servent ensuite à générer un fichier midi. C'est ce fichier qui deviendra l'amorce du modèle de génération mélodique.

2.3.3. Contrôle de la génération mélodique

Le système central contient six instances d'un sous-programme servant aux contrôles des six voix qui habitent le paysage sonore de l'installation. Ces sous-programmes se nommeront les modules de voix. Ils permettent de contrôler les modèles de génération mélodique et sonore. Deux voix produisent des sons qui n'impliquent pas les cordes vocales, telles que des consonnes fricatives et des chuchotements, une des voix prend le rôle d'un drone dans un registre grave, une voix est la voix principale et deux autres voix harmonisent cette dernière (tab. 1).

[7] Service d'hébergement et de gestion de développement de logiciel en ligne.

	Mélodie	Synthèse	Effets	Rythme
Voix 1	Pas de contenu mélodique.	Timbre non consonnant, respiration et bruit de bouche.	Légère distortion. Légère réverbération. Spatialisé à gauche.	Séquences de quelques secondes, entrecoupées de courts silences.
Voix 2	Pas de contenu mélodique.	Timbre non consonnant, respiration et bruit de bouche.	Légère distortion. Légère réverbération. Spatialisé à droite.	Séquences de quelques secondes, entrecoupées de courts silences.
Voix 3	Drone dans le registre grave.	Timbre consonnant. Généré avec synthèse concaténative.	Filtre passe-bas. Légère réverbération	En continu.
Voix 4	Séquence générée à partir de la mélodie analysée.	Timbre consonnant. Imitation du timbre de l'interacteur.	Légère réverbération. Dédoublement grâce à un très court délai. Spatialisation à gauche et à droite.	Généré immédiatement après chaque intervention sonore.
Voix 5	Harmonisation de la voix 4 dans les aigus.	Timbre consonnant. À mi-chemin entre le timbre du participant et la synthèse concaténative.	Effet de chorus. Légère réverbération. Spatialisé à gauche.	Lorsqu'actif, généré immédiatement après chaque intervention sonore.
Voix 6	Harmonisation de la voix 4 dans les graves.	Timbre consonnant. À mi-chemin entre le timbre du participant et la synthèse concaténative.	Effet de chorus. Légère réverbération. Spatialisé à droite.	Lorsqu'actif, généré immédiatement après chaque intervention sonore.

Tableau 1 – *Morphogenèse* : caractéristiques des voix.

Les six instances du module de voix contiennent des paramètres qui permettent de faire varier le comportement et le timbre des mélodies et des sonorités générées par les modèles d'apprentissage. Lorsque le modèle de génération mélodique créer un fichier MIDI, celui-ci contient l'amorce mélodique ainsi que la mélodie générée à partir de celle-ci. Les paramètres de ces modules permettent de choisir quelle proportion de l'amorce ainsi que de la mélodie inventée sera préservée. Ils permettent aussi de choisir la température du modèle. Ce paramètre fera varier ce qu'on pourrait appeler l'audace du générateur ou, autrement dit, fera en sorte que le modèle génère des mélodies simples et monotones ou, inversement, denses et s'éloignant harmoniquement de l'amorce. Finalement, ils permettent d'utiliser la fin de la dernière mélodie générée comme amorce plutôt que le fichier MIDI afin de générer des séquences mélodiques ad vitam aeternam même lorsqu'aucune intervention sonore n'a lieu.

2.3.4. Génération de mélodies

Les scripts du générateur de mélodie, écrits avec le langage de programmation Python, sont des bons exemples de l'approche adoptée pour l'organisation de l'ensemble des scripts de l'installation. Ce code est organisé en deux sections contenues à l'intérieur d'un script principal. Une de ces sections, le cœur du générateur mélodique, s'occupe de définir les paramètres de la génération, charger le modèle, préparer les données pour la prédiction, effectuer la prédiction, ajuster le fichier MIDI résultant, le sauvegarder sur le

disque et retourner le chemin d'accès vers ce fichier. L'autre section est un serveur de messages *Open Sound Control* (OSC). Ce script s'occupe de relayer les différents messages envoyés entre le système central et la section responsable de la génération mélodique.

2.3.5. Contrôle de la génération sonore

Le modèle utilisé pour la génération sonore est une adaptation de la librairie DDSP[8], nommé DDSP_gm2[9]. Elle permet de créer des modèles qui contrôlent les paramètres d'éléments de traitements sonores numériques, tels que des filtres et des oscillateurs, afin d'imiter les sonorités du corpus sonore sur lequel ils ont été entraînés. Dans sa version originale, le système prend en entrée un fichier sonore et génère un nouveau fichier sonore à partir de celui-ci. On pourrait, par exemple, envoyer un extrait sonore de violon au modèle pour qu'il renvoie un extrait sonore similaire, mais interprété à la trompette. À condition, bien sûr, que le modèle ait préalablement été entraîné sur un corpus d'extraits sonores de trompettes.

DDSP_gm2, l'adaptation de la librairie DDSP, a été créée expressément pour la voix chantée. Lors de l'entraînement sur un corpus d'extraits sonores de voix, les fichiers sont analysés pour en extraire les hauteurs de notes, les intensités et les MFCC. L'utilisation des MFCC pour l'entraînement n'est pas présente dans la version originale de DDSP. L'ajout de ce descripteur permet la création d'un modèle qui tient compte des formants et du timbre de la voix. La librairie DDSP_gm2 est construite de telle sorte que, autant lors de l'entraînement du modèle que lors des demandes de prédictions, l'algorithme s'attend à recevoir directement des fichiers sonores à partir desquelles il extrait des hauteurs de notes, des intensités sonores et des valeurs de MFCC. J'ai modifié le code de la librairie pour que ces données d'analyse puissent être envoyées directement à l'algorithme. C'est donc le programme Pure Data principal qui procède à ces analyses ce qui permet, comme ce sera bientôt vu, d'altérer de différente façon les valeurs avant de les envoyer vers le modèle.

Aucun modèle préentraîné n'étant disponible, j'ai entraîné un modèle expressément pour l'installation à partir de fichiers sélectionnés dans une base de données regroupant une multitude d'extraits de voix chantée de 20 chanteurs (Wilkins *et al.*, 2018).

[8] Une librairie Python créée par l'équipe de recherche Magenta.
[9] Une librairie Python créée par Gian-Marco Hutter dans le cadre d'un projet de maîtrise à l'École polytechnique fédérale de Zurich.

L'entraînement a été fait de telle sorte que le modèle soit en mesure de reproduire différents timbres de voix humaine, de reproduire différents phonèmes et de générer des sons de voix sans tonalité. Plus de détails sur le processus d'entraînement seront donnés dans le chapitre trois.

Voici comment l'algorithme de l'installation procède pour contrôler le modèle de génération sonore. Lorsque le système central est averti qu'une nouvelle mélodie a été générée, il relaie l'information à la voix concernée. Pour générer un fichier audio, le modèle a besoin de listes de hauteurs de notes en fréquence associées à des intensités sonores et à des valeurs de MFCC. Le fichier MIDI est donc converti en une liste de fréquence. Alors que la liste de fréquence déterminera la mélodie, des listes de valeurs d'analyses d'intensité sonore et de MFCC détermineront les variations d'intensité, le timbre et les formants du fichier sonore final. Le système de *Morphogenèse* permet de générer des fichiers sonores qui peuvent ressembler ou différer des sons faits par l'interacteur. Pour y arriver, le système génère des listes de MFCC à partir de la voix de l'interacteur et à partir de la banque de sons ayant servi à l'entraînement du modèle. Ceci permet de choisir si le timbre et les formants seront ceux de l'interacteur, de la banque de sons, ou encore d'une interpolation entre les deux.

Afin de pouvoir modifier le timbre et le comportement des différentes voix, divers paramètres ont été rendus accessibles :

- Un multiplicateur de hauteur de note.
- Un multiplicateur d'intensité sonore.
- Une valeur d'interpolation d'intensité sonore entre l'extrait sonore en provenance de l'interacteur et celui généré à partir de la banque de son.
- Une valeur d'interpolation similaire, mais pour les valeurs MFCC.
- Deux valeurs permettant de choisir le caractère sonore des fichiers en provenance de la banque de son.

2.3.6. Génération sonore

Les scripts python du générateur sonore sont organisés de la même façon que ceux du générateur mélodique. Un script principal est divisé en deux sections : le modèle de génération sonore et un serveur de messages OSC. Le serveur OSC permet de relayer les informations entre le système central et le modèle de génération sonore. Pour chacune des

voix, toutes les données nécessaires sont envoyées au modèle lorsque le système central fait une demande de prédiction. En retour, le modèle crée un fichier sonore et retourne un message contenant le chemin d'accès vers le fichier ainsi que le numéro de voix pour laquelle il a été généré. Finalement, le serveur OSC relaie ces deux informations au système central.

2.3.7. Signal en sortie

Lorsque le système central reçoit l'indication qu'un nouveau fichier sonore a été généré pour une voix donnée, il relaie l'information au module s'occupant de gérer l'audio entrant et sortant.

À l'intérieur de ce programme se trouvent six instances d'un même sous-programme, une instance pour chacune des voix. Ces instances gèrent la lecture de fichiers audio ainsi que l'ajout d'effets sonores avant que le signal soit envoyé vers l'interface audio. La nature de ces effets varie en fonction des différentes voix. Ils consistent, entre autres, en de légères réverbérations qui aident à placer les voix dans l'image stéréo ou encore en des effets de distorsions subtiles qui enrichissent le spectre aigu des extraits sonores. Ces extraits ont peu de définitions dans cette zone spectrale étant donné que le modèle de génération sonore produit des fichiers dont le taux d'échantillonnage n'est que de 16 kHz.

Lorsque la lecture d'un fichier est terminée, un message est envoyé au système central et relayé vers la voix correspondante. Selon le paramétrage de cette voix, ce message peut potentiellement provoquer une nouvelle demande de prédiction au système de génération mélodique et, ainsi, réenclencher toute la série d'opérations qui mène à la lecture d'un fichier sonore.

Les huit extraits sonores suivants, disponibles en annexe, donnent à entendre le comportement de chacune des voix en fonction d'un fichier sonore en entrée du système (le son original) :

Extrait 3 à extrait 10 – *Morphogenèse* : extraits sonores

2.4. Parcours du signal visuel

Cette section permettra de détailler davantage le fonctionnement de l'aspect visuel de l'installation. Le parcours du signal sera suivi à partir de sa captation par la caméra, en

passant par la détection faciale et la génération de visage, jusqu'à la création de l'image finale (fig. 10). Les contrôles ayant été exposés qui ont permis de créer les effets et l'atmosphère visuelle de *Morphogenèse* seront détaillés.

L'organisation générale du code servant à la détection faciale et à la génération de visage ressemble à celle du générateur de mélodie et du générateur sonore. Cependant, ce code contient davantage de sections qui consistent en un serveur de message OSC, de la détection faciale, un modèle de génération faciale et le contrôle de cette génération faciale. Le script principal met en relation ces différentes sections et contient aussi une boucle, exécutée 20 fois par seconde, à l'intérieur de laquelle sont orchestrés la prise de photo, la détection faciale, la génération faciale, le traitement de l'image et l'envoi de l'image finale vers le projecteur.

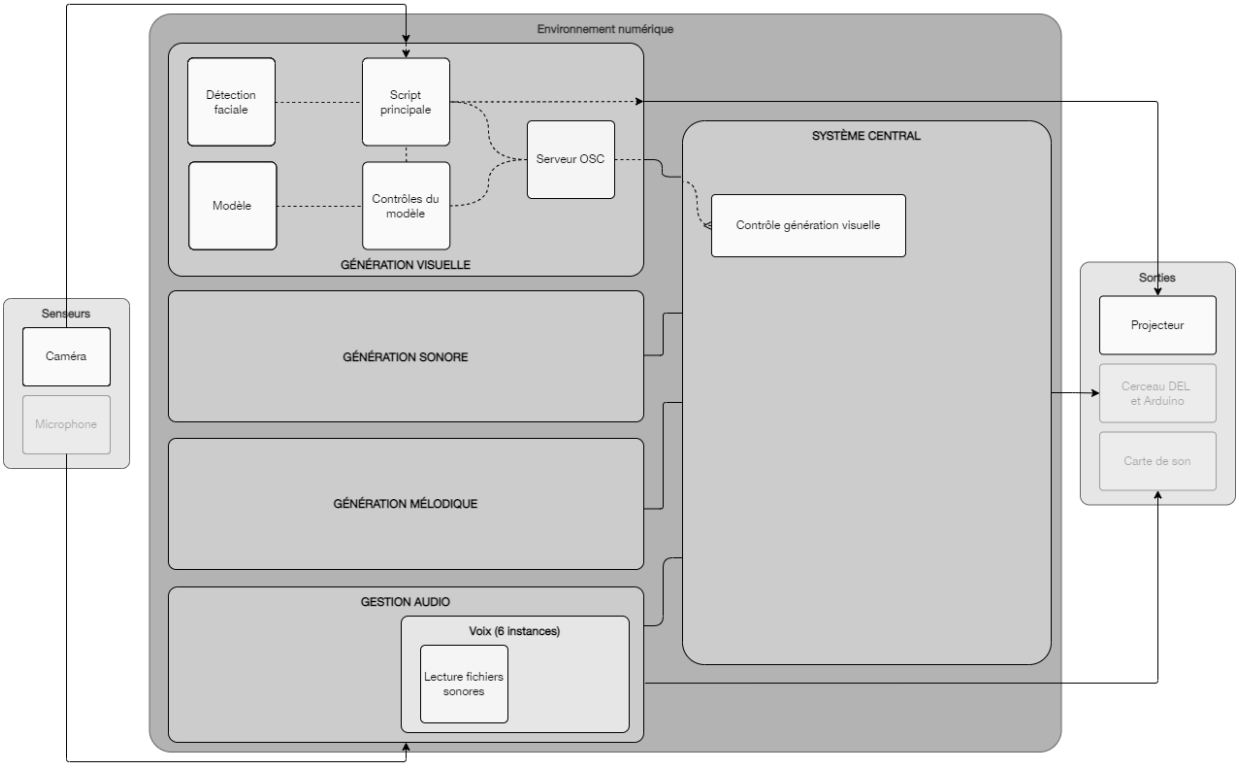

Figure 10 – *Morphogenèse* : parcours du signal visuel.

2.4.1. Signal en entrée

La caméra qui filme les interacteurs est une webcaméra haute définition. La webcaméra a été choisie pour sa lentille grand-angle et pour sa forme cylindrique qui permet d'apposer le cerceau de lumière autour de la lentille.

2.4.2. Détection faciale

La section s'occupant de la détection faciale prend en entrée une image et retourne, pour chaque objet détecté, une valeur correspondant à la confiance qu'a l'algorithme que l'élément détecté est bien un visage ainsi que les coordonnés d'un rectangle à l'intérieur duquel se trouve l'objet en question. Ces valeurs permettent donc de connaître la taille et la position des visages dans l'image. À partir de ces valeurs initiales, des conditions sont appliquées qui permettent de filtrer les visages détectés. Si le visage est trop petit ou trop en périphérie de l'image, le visage n'est pas retenu. Ces conditions ont pour objectif de ne retenir que les visages se trouvant dans la zone d'interaction. Une fois la détection accomplie, le temps d'interaction est mis à jour et, pour chaque visage détecté, une nouvelle image est créée, cadrée autour de la détection.

La détection faciale retourne le temps écoulé depuis qu'un visage a été détecté, le nombre de visages détecté, une liste de photo des visages recadrés et l'image originale. Le script principal, via le serveur OSC, envoie alors le nombre de visages détectés et le temps d'interaction vers le système central.

2.4.3. Génération de visages

Au cœur de la génération de visage se trouve l'algorithme d'apprentissage profond StyleGAN créée par Karras, Laine, et Aila (2019). L'installation utilise un modèle préexistant, entraîné à partir de la base de données Flickr-Faces-HQ (Karras, Laine, et Aila, 2018), qui contient 70 000 images en haute définition de visages de personnes d'âge et d'ethnicité variés. Un autre modèle, nommé encoder4editing et créé par Tov *et al.* (2021), s'occupe d'encoder les visages des interacteurs sous forme de coordonnées. Ces coordonnées peuvent être envoyées en entrée du modèle qui retournera l'image d'un visage similaire généré artificiellement. L'encodage d'un visage sous forme de coordonnées permet différentes manipulations, tel que l'interpolation graduelle vers d'autres visages artificiels et la modification des traits dudit visage. Techniquement, ces manipulations consistent à appliquer des translations à un point se trouvant dans l'espace latent créé par le modèle.

Il est aussi possible d'identifier des axes précis sur lesquels faire des translations dans l'espace latent qui ont comme effet de modifier des aspects spécifiques du visage. Se déplacer sur l'un de ces axes identifiables équivaut à dire qu'une direction est appliquée

au visage encodé. Ainsi, il m'a été possible de choisir une série d'axes qui m'apparaissaient esthétiquement pertinents et qui me permettent de faire ouvrir et fermer la bouche et les yeux, faire varier l'âge, le genre et la direction de la lumière sur le visage.

Dans le chapitre trois se trouve une explication plus détaillée du concept d'espace latent et du potentiel créatif lié à sa navigation.

2.4.4. Contrôle de la génération de visage

Cette section du code a comme rôle d'appliquer des transformations et d'interpoler entre des points dans l'espace latent. Une série d'opérations qui permettent de générer une animation d'un visage en constante métamorphose ont lieu plusieurs fois par seconde. D'abord, deux visages sont générés en parallèle, un visage inspiré des traits de l'interacteur et un visage généré aléatoirement. Selon un poids qui varie au fil de l'expérience, un visage qui mélange les traits de ces deux visages est créé. Au début de l'expérience, ce poids penche du côté du visage aléatoire. Plus l'expérience évolue, plus le poids bascule vers le visage de l'interacteur, qui se reconnaît peu à peu dans l'image projetée. Sur ce visage est appliqué un effet de fluctuation paramétrable (communément appelé *jitter*). Ainsi, les traits du visage varient avec plus ou moins d'amplitude autour de son point de départ. La dernière opération consiste à appliquer les directions, des effets qui permettent d'animer de manière ciblée les traits du visage. Comme détaillé au chapitre trois, ces effets permettent par exemple de modifier l'ouverture des yeux ou de la bouche.

Cette section du code est donc constamment en train de procéder à des calculs qui aboutissent à l'image d'un visage final. En plus de générer de nouveau point dans l'espace latent, elle permet de choisir et d'ajuster la vitesse et le poids des interpolations, l'amplitude et la vitesse de l'effet de fluctuation et l'intensité des directions à appliquer. C'est donc elle qui permet de faire évoluer l'animation et l'identité du visage au fil de l'expérience.

2.4.5. Génération de l'image finale

Comme mentionné précédemment, le script principal contient une boucle qui est exécutée vingt fois par seconde. À chaque exécution, une requête est faite à la section s'occupant de la navigation de l'espace latent pour obtenir l'image finale générée par le modèle. Ensuite, un léger effet de flou est appliqué à cette image pour lui donner un fini plus uniforme. Puis, en fonction des valeurs envoyées par le système central, un masque flouté et un

masque noir entourant le visage sont appliqués avec plus ou moins d'intensité. Ces effets ont pour objectif d'adoucir les contours de l'image et de mettre davantage l'accent sur les traits du visage.

Selon le stade de l'évolution de l'expérience, des conditions sont appliquées qui peuvent provoquer l'ajout de texte sur l'image. Si aucune interaction n'a lieu, la phrase « placez-vous sur la marque au sol » est ajoutée. L'intensité lumineuse du texte oscille inversement à la luminosité du cerceau de lumière. Si un interacteur est détecté, mais qu'il n'a pas encore fait d'intervention sonore, la phrase précédente disparaît et la phrase « faites des sons » est ajoutée. Finalement, l'image finale est envoyée vers le projecteur.

Ce script, en plus d'organiser la boucle d'animation, permet donc de fournir des indications aux interacteurs et de faire évoluer l'intensité des effets appliqués sur l'image.

2.5. Évolution de l'expérience

Le système central est l'outil qui permet d'ajuster et d'organiser l'ensemble de l'expérience en rendant accessibles les différents paramètres et en envoyant diverses requêtes à la génération mélodique, sonore et visuelle. Il reçoit des données en provenance de la détection faciale et de la gestion de l'audio en entrée qui lui fournit des informations sur le monde extérieur (fig. 11). Ces informations permettent d'organiser le déroulement de l'expérience. Elles incluent le nombre d'interacteurs présents dans la zone d'interaction, la quantité de temps depuis le début de l'interaction et la présence d'interventions sonores faites par un interacteur.

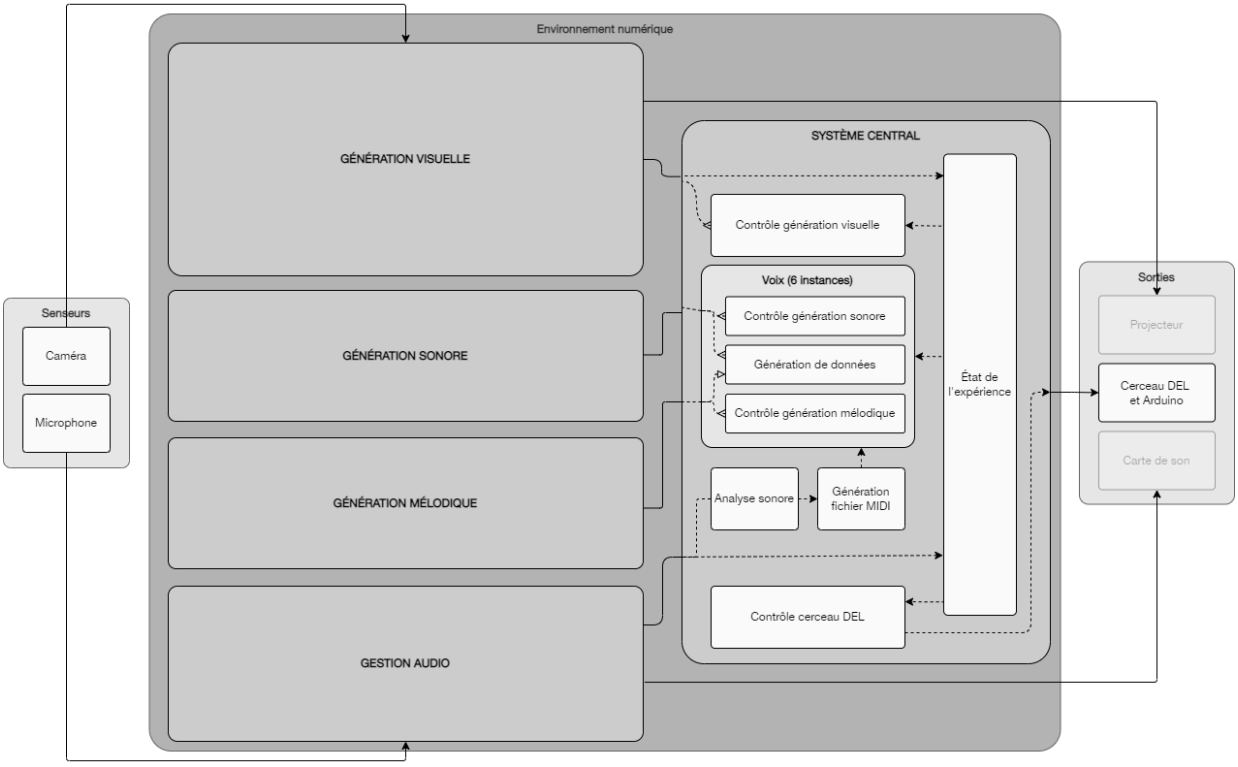

Figure 11 – *Morphogenèse* : système central.

Avec ces données, le système procède à des calculs desquels résulte un chiffre qui représente l'état de l'évolution de l'expérience. L'état zéro correspond à l'absence d'interacteur alors que l'état un correspond à la présence d'interacteur. L'état deux survient lorsqu'un interacteur est présent et qu'il a fait au moins une intervention sonore. Les états trois, quatre, cinq et six surviennent après deux, quatre, six et dix interventions sonores. Les six états de l'expérience servent à faire varier les différents paramètres qui influencent le comportement de l'installation. La valeur de ces paramètres passe de l'ancienne à la nouvelle de manière instantanée ou en fonction d'une rampe dont la vitesse est ajustable.

Lorsque l'état zéro est en cours, donc lorsqu'aucune interaction n'a lieu, la luminosité du cerceau de lumière oscille lentement. Lorsque l'installation passe de l'état zéro à l'état un, donc lorsqu'un interacteur est détecté, le cerceau fait un flash lumineux puis sa luminosité descend au plus bas. À partir de ce moment, tant que l'interaction est en cours, la luminosité du cerceau est proportionnelle à l'intensité des interventions sonores de l'interacteur.

Du côté sonore, lorsque l'état zéro est en cours, ce sont les deux voix sans tonalité qui se font entendre. Lorsque l'installation passe vers l'état numéro un, les voix sans tonalité disparaissent pour laisser place à la voix de drone. Par la suite, plus l'interacteur fait d'interventions sonores, donc plus les états s'enchaînent, plus les réponses mélodiques de la voix principale sont longues. À partir de l'état numéro six, les deux voix d'harmonie s'ajoutent à la voix principale et à la voix de drone.

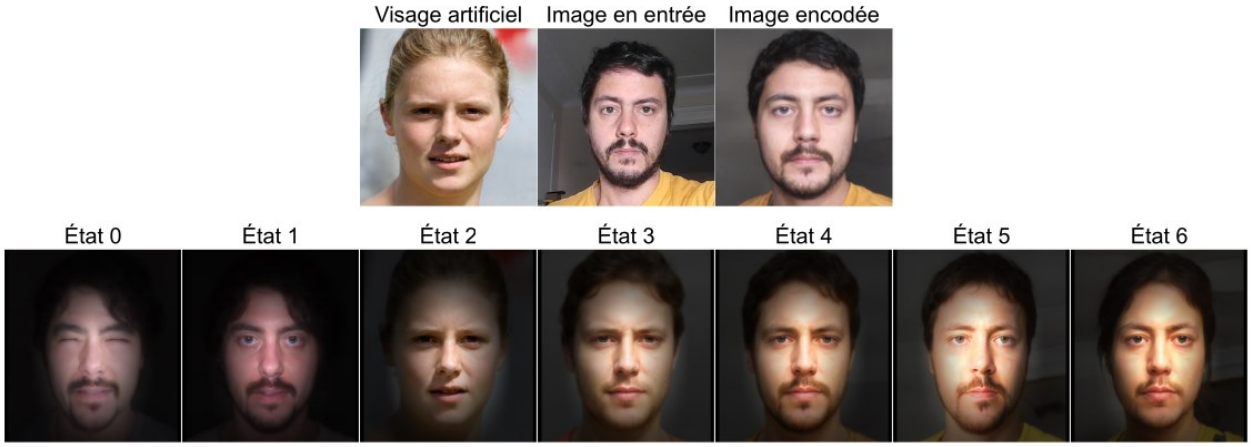

Figure 12 – *Morphogenèse* : rendu visuel lors des différents états.

Les différents états exercent aussi une influence sur les paramètres de la génération d'images (fig. 12). À l'état zéro, lorsqu'il y a absence d'interaction, le système central envoie un message au générateur visuel qui a pour effet de fermer les yeux du visage généré et les traits du visage affiché sont pigés dans une liste de coordonnées contenant tous les visages encodés des interacteurs passés. Pour tous les autres états, donc quand un interacteur est présent, les yeux sont ouverts et le passage des différents états amène le visage artificiel à arborer des traits de plus semblables à ceux de l'interacteur. L'intensité de la variation de l'angle de la lumière, de l'âge et du genre du visage augmente au fil de l'expérience alors que la quantité de flou et le masque noir entourant le visage diminuent peu à peu.

2.6. Conclusion : apprentissage automatique et stratégies créatives

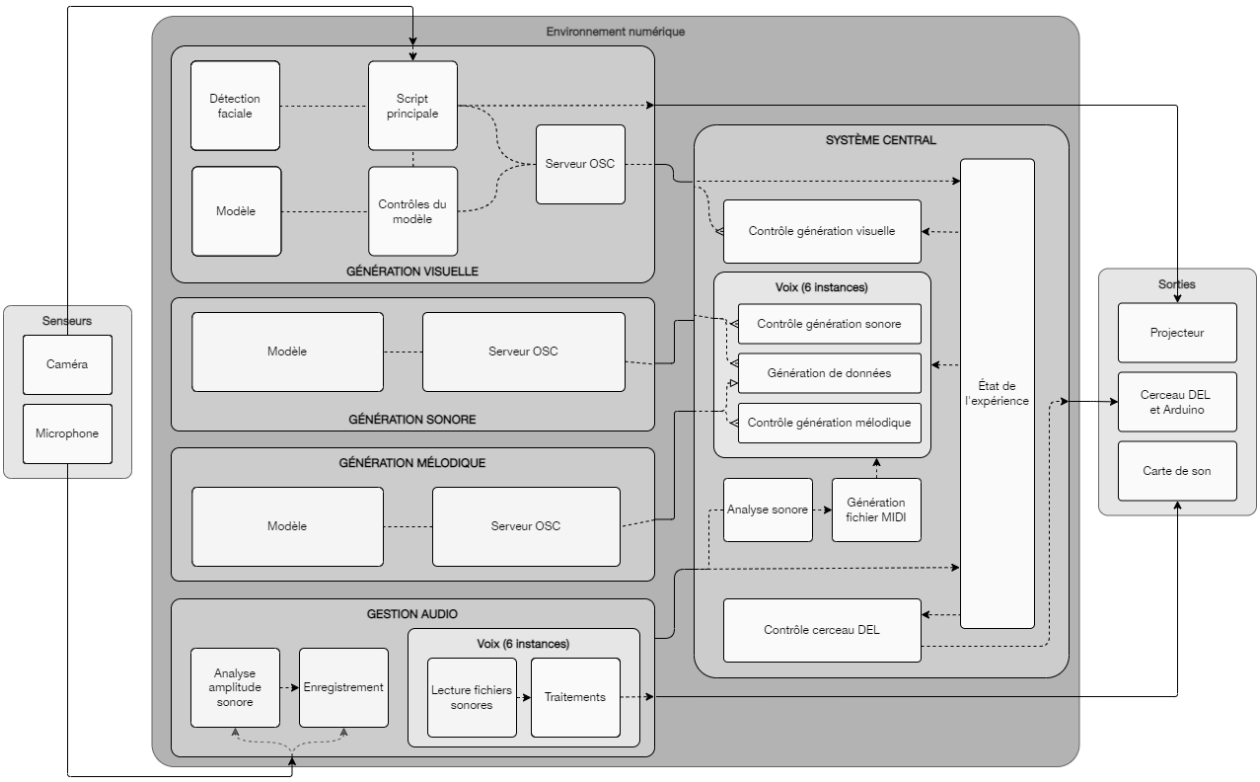

Figure 13 – *Morphogenèse* : détails du système.

La création de *Morphogenèse* fut une occasion de travailler avec des algorithmes d'apprentissage automatique d'une plus grande complexité qu'avec *Deep Duo*, capables de générer des mélodies, des fichiers sonores et des images et, aussi, de trouver des manières de mettre en commun ces algorithmes. À cet effet, le schéma ci-dessus (fig. 13) illustre la complexité du système mis en place. En revanche, ces algorithmes en provenance de sources externes viennent avec une architecture et des contrôles prédéfinis. Il m'a donc fallu trouver des stratégies qui allaient me permettre de satisfaire mes objectifs créatifs. Ces stratégies incluent la manipulation des données en entrée, le paramètre de température, la modification d'un algorithme préexistant et la navigation dans l'espace latent d'un modèle génératif. Alors que ce chapitre s'est concentré à faire la description de l'implémentation technique de ces différentes stratégies, le chapitre qui suit permettra de comprendre davantage leurs natures et leurs potentiels créatifs.

Chapitre 3 - Perspectives sur la création avec l'intelligence artificielle

La création avec des algorithmes d'apprentissage automatique a été le fil conducteur de mon parcours à la maîtrise. Partant de connaissances en programmation et de plusieurs années d'expérience à travailler avec des installations interactives, je me suis lancé le défi de créer des œuvres audiovisuelles utilisant l'intelligence artificielle. J'ai donc exploré cet univers en travaillant avec des modèles de plus en plus complexes et en lisant les textes d'artistes et de chercheurs s'intéressant à ce type d'outil. Ces années d'études m'ont permis de me rendre compte de quelque chose qui m'apparaît aujourd'hui fondamental : bien que l'on considère l'apprentissage automatique comme un outil informatique, son utilisation implique des processus fort différents de la programmation procédurale ou de l'algorithmique au sens traditionnel du terme. Cela apparaît peut-être comme une évidence, mais étant donné que travailler avec ces algorithmes passe par les mêmes outils qu'avec des systèmes classiques, intégrer réellement cette information peut prendre du temps. Leurs présences dans notre ordinateur peuvent donner l'impression que l'esprit d'un inconnu s'est glissé dans le corps d'un être familier. Intégrer réellement cette information amène à s'engager dans de nouveau processus, à avoir de nouvelles attentes et à développer de nouvelles façons d'interagir avec la machine. Un des apprentissages importants de cette maîtrise a été de comprendre et m'ouvrir à ce changement de paradigme. J'utiliserai donc ce dernier chapitre pour tenter de rendre tangible cette différence puisque c'est à mon sens le point de départ pour aborder et réfléchir la création avec l'intelligence artificielle.

Dans ce chapitre, je soutiendrai que pour créer une œuvre aboutie, il faut que le créateur trouve des moyens engageants de travailler avec les algorithmes d'apprentissage et qu'il soit en mesure de porter un jugement esthétique sur les processus en cours. Le travail de création avec l'intelligence artificielle n'est pas unidirectionnel. Il ne s'agit pas de mettre en place un ensemble de règles qui régissent le comportement de la machine. Il s'agit plutôt d'un aller-retour entre le créateur et l'ordinateur et, en ce sens, d'un processus interactif entre un agent humain et un agent machine. Pour cette raison, un intérêt sera porté à la tradition de l'art interactif qui tente par différents moyens d'établir un échange entre l'homme et la machine. Des parallèles seront établis entre cette tradition et le

processus de création avec l'intelligence artificielle. Ce n'est pas uniquement le processus de création qui est différent. Pour celui qui observe les comportements de ces modèles, c'est aussi la nature des éléments générés. Pour tenter de construire un regard critique face à ces éléments pour lesquels il y a encore peu de références, des théories relativement récentes qui jettent les bases d'un cadre esthétique pour des œuvres créées avec l'apprentissage automatique seront examinées.

Ensuite, différents processus qui permettent l'utilisation d'algorithmes d'apprentissage automatique en tant qu'outils de création seront examinés. Ces processus peuvent prendre place en amont de l'entraînement du modèle ou au moment de lui faire faire des prédictions. Le créateur peut intervenir sur la base de données, l'architecture du modèle, ses paramètres d'entraînement et, par différentes techniques, sur les données envoyées en entrée d'un modèle entraîné.

Dans chacune des sections de ce dernier chapitre, je tenterai d'établir des liens entre ces différents exemples pratiques et les concepts évoqués plus tôt dans ce chapitre.

3.1. Changement de paradigme

Un créateur travaillant avec des outils numériques s'est habitué à un certain type de processus avec l'ordinateur. Qu'il travaille avec des logiciels, un langage de programmation visuel ou un langage de programmation textuel, il s'est habitué à ce que l'ordinateur exécute les instructions qu'il lui fournit. Autrement dit, son travail consiste à expliciter les règles qui régiront le comportement de la machine. Avec l'apprentissage automatique, le processus est différent puisque c'est l'algorithme lui-même qui, en entraînant un modèle, définit ces règles. D'ailleurs, plus les modèles sont complexes, plus il est difficile, voire impossible, de comprendre la logique derrière les règles créées par l'algorithme. Le processus n'est donc plus le même. Il ne s'agit plus de définir un comportement, mais plutôt de rassembler les éléments qui permettront à la machine de générer des comportements qui conviennent à nos objectifs. Généralement, ce processus passe par une série d'essais erreurs où le comportement du modèle est testé, puis les éléments que l'on fournit à l'algorithme sont modifiés pour que celui-ci produise un modèle qui semble plus approprié. Il s'agit donc réellement d'un échange entre le créateur et l'ordinateur qui passe par un processus itératif et interactif où les deux parties sont engagées à faire des ajustements jusqu'à ce que le résultat soit satisfaisant. Une part de ce

processus est d'ailleurs mis en scène avec *Deep Duo*, qui permet de réentraîner les modèles en fonction de différents paramètres.

Pour illustrer la différence entre la programmation procédurale et l'utilisation d'algorithmes d'apprentissage, je me permets d'utiliser une analogie tirée du travail de Lucy Suchman (1987a) à propos de différents processus de navigation, au sens maritime du terme. Elle place en opposition la navigation européenne et la navigation des Chuukois des États fédérés de Micronésie. La navigation européenne, comme la programmation procédurale, adopte une approche cartésienne. Elle utilise un plan qui s'appuie sur des principes de navigation et qui n'est pas dépendant d'événements locaux pouvant survenir pendant le voyage. Le navigateur s'efforce constamment de rester sur la trajectoire prévue. Si, en cours de route, un événement survient qui l'oblige à bifurquer de la trajectoire prévue, il doit d'abord mettre à jour son plan pour déterminer la suite de ses actions. L'approche du navigateur chuukois est différente et plus proche de celle à adopter lorsque l'on travaille avec des algorithmes d'apprentissage. Celui-ci ne fait pas de plans avant d'entamer son périple. Dès le début du voyage, son objectif est clair, mais la trajectoire qu'il empruntera pour y arriver n'est pas définie puisque cette trajectoire dépend de circonstances qu'il n'est pas en mesure de prévoir. Ainsi, Suchman dira que ses actions sont situées (*situated actions*), en opposition à planifiées, puisqu'elles répondent aux événements au fur et à mesure qu'ils se présentent.

L'apprentissage automatique oblige un certain lâcher-prise et suggère un rapport différent avec la machine, plus proche de l'interaction que de la simple utilisation. Ce changement de paradigme, puisqu'il s'opère à l'intérieur d'un objet familier qui a habitué l'utilisateur à des comportements bien précis, m'aura pris un certain temps à assimiler. Au fil du temps, je me suis rendu compte qu'une bonne part du travail avec les algorithmes consiste à passer du temps à interagir avec les modèles entraînés afin de développer une connaissance intuitive qui permet d'avoir une meilleure idée de la manière d'ajuster leurs comportements. Je me suis aussi rendu compte que, contrairement à ce que m'avait habitué la programmation procédurale, je n'avais pas complètement le contrôle sur le fonctionnement du système et qu'il fallait que j'accepte de céder une part de ce contrôle à la machine. S'en rendre compte et accepter ce changement de paradigme permet d'avoir un rapport plus fécond avec ces algorithmes.

3.2. Interactivité

Ce paradigme invite donc à revisiter les processus par lesquels s'opère la création avec la machine. Pour arriver à des résultats satisfaisants, le créateur doit trouver des moyens engageants d'interagir avec ces algorithmes et ces moyens passent plus par une interaction avec les modèles que part une approche procédurale. L'histoire de la création avec l'intelligence artificielle est encore jeune et il me semble pertinent d'aller puiser du côté de traditions artistiques plus établies. Je propose donc ici d'examiner des réflexions à propos de l'art interactif pour trouver des pistes qui faciliteront cette recherche. En effet, depuis plus d'un demi-siècle, l'art interactif étudie et explore nos modes de communication avec l'ordinateur et propose des œuvres dont le sens émerge d'un dialogue entre l'homme et la machine.

Selon l'archéologue des médias et historien Erkki Huhtamo, l'art interactif émerge de courants artistiques qui, pendant les années 1960, ont participé à changer la relation entre l'œuvre d'art et son spectateur :

> The roots of interactive media art are found in the 1960s [...]. The expansion of the traditional field of art, the dream about "Total Art," the annihilation of the barrier between life and art, the "dematerialization" of the art object" (Lucy Lippard), process art, participation art, concept art, Fluxus, the Happening-movement and Situationism, "Art and Technology," kinetic art, cybernetic art (Jack Burnham), closed circuit video installations—these phenomena may be heterogenous, [sic] but they are part of one and the same process which had a profound effect on the relationship between art and its audience. (Huhtamo cité dans Salter, 2010, p.305)

Tel que décrit dans la littérature (Cangiano, Fornari, et Seratoni, 2022; Garwood, 2007; Pask, 1971), on sent chez plusieurs artistes de cette époque, tels que Gordon Pask avec son système *Musicolor*, le collectif Gruppo T et leur exposition *Miriorama 1* ou encore les organisateurs de l'événement new-yorkais de 1966 *9 Evenings : Theater and Engineering*, un désir de créer des espaces et des œuvres qui permettront au spectateur de se libérer d'une certaine forme de passivité et de prendre part activement à de nouvelles formes d'expérience artistiques.

L'artiste et chercheur Chris Salter met en opposition deux types d'œuvres interactives. L'une mettant l'accent sur la réactivité de l'ordinateur par rapport à son environnement et l'autre sur l'engagement corporel du spectateur. Le premier type d'œuvre constitue, selon lui, la majorité des œuvres interactives : « the inherent performative constitution of

interactive media [...] took a secondary role in comparison to the emphasis on computer responsiveness » (Salter, 2010, p.321).

Le deuxième type d'œuvre tente d'élargir le discours de l'art interactif vers des questions qui abordent la mise en corps, l'intimité, le sens du touché et le jeu social par le biais, entre autres, d'espaces augmentés et d'interfaces disposées sur les corps des spectateurs (Salter, 2010, p.338). Dans ce type d'œuvre, le centre d'attention est déplacé vers « [...] the articulation of a space of meeting between the artwork and the viewer [...] » (Shaw, 1996, p.171) à travers un engagement sensori-moteur des spectateurs. Ainsi, l'interaction n'est plus uniquement un moyen d'activer ou de générer le contenu de l'œuvre, mais devient une part de l'œuvre telle que ressentie à la lisière entre la machine et le corps du spectateur : « [the] artwork is more and more embodied in the interface [...] where the artwork as an artifact seems to disappear altogether » (Shaw, 1996, p.171). Ainsi, puisque c'est l'interaction elle-même qui est au cœur de la création, pour faire l'expérience de l'œuvre, il faut s'engager physiquement avec celle-ci. L'auteur Simon Penny abonde dans ce sens lorsqu'il affirme :

> [...] a substantial part of the meaning of the [interactive] work resides in or emerges from the sensorimotor experience of doing it. Like riding a bicycle, we cannot "understand" the work without bodily engagement—not simply immersion, which implies the possibility of inaction, but fully dynamic proprioceptive/kinesthetic engagement. (Penny, 2017b, p.393)

Si l'art interactif peut effectivement libérer le spectateur de son rôle passif, c'est probablement avec ce deuxième type d'œuvres qui passent par une mise en corps où l'engagement actif et sensoriel du spectateur avec la machine, l'espace et les autres spectateurs sont constituants de l'œuvre elle-même. Il est possible de dire que, à la manière du navigateur Chuukois, les actions du spectateur sont situées, puisqu'elles répondent aux événements au fur et à mesure qu'ils se présentent. En ce sens, et en partant de la prémisse que « [...] our common sense of the social world is not the precondition for our interaction, but its product » (Suchman, 1987b, p.40), c'est donc par l'interaction que l'intelligibilité et l'objectivité de l'œuvre est produite.

Ainsi, selon un certain pan de l'art interactif, la mise en corps de l'œuvre permet à l'interaction entre l'homme et la machine d'être générateur de sens. Cette mise en corps implique un engagement proprioceptif à partir duquel « [it] is impossible to adopt an

objective view [...] » (Penny, 2017a, p.360). Le spectateur est donc obligé de poser des actions de types situés à travers lesquelles se développe un sens commun de l'œuvre.

Avec *Morphogenèse*, l'engagement physique du spectateur se concentre sur sa voix et sur les expressions de son visage. Le reflet imparfait que lui renvoie l'installation sollicite son corps et l'oblige à prendre conscience de sa position dans le système. C'est cet engagement qui lui permet de sentir les subtilités des échanges sonores avec la machine et l'effet d'étrangeté qui émerge lorsque l'on voit les traits de son propre visage apparaître à l'écran. Cet engagement physique a été d'une tout aussi grande importance au moment de la création avec les différents modèles à l'œuvre dans *Morphogenèse* puisque c'est grâce à celui-ci que j'ai pu réellement sentir les détails de leurs comportements. Cette connaissance physique, presque intuitive, m'a permis de mieux comprendre comment ajuster leurs comportements en fonction de ma vision créative.

3.3. L'esthétique des comportements

Dans le but d'affiner le regard que le créateur pose sur les éléments générés par ces algorithmes lorsqu'ils sont utilisés dans un cadre créatif, je propose de s'attarder aux théories de deux artistes et chercheurs. D'abord, Simon Penny, qui propose d'observer les œuvres interactives sous l'angle d'une esthétique des comportements, puis Sofian Audry qui propose une esthétique des comportements adaptatifs pour parler d'œuvres interactives créées avec des algorithmes d'apprentissage automatique.

Penny part de la prémisse que les machines, ou ordinateurs, ont un comportement, dans le sens que ce qu'elles produisent en sortie varie selon un « raisonnement » basé sur une certaine « conscience » de changements dans leurs environnements (Penny, 2017c, p.18). Il s'intéresse à un type d'art créé à l'aide de ces machines qu'il nomme « behaving art » (Penny, 2017a, p.355), c'est-à-dire des créations qui, par le biais de senseurs, répondent aux variations de leurs environnements. Il affirme que, culturellement, ces créations représentent quelque chose de nouveau qu'il nomme des artefacts culturels comportementaux. Ces « behaving cultural artifact » (Penny, 2017a, p.357) amène le spectateur à avoir des interactions culturelles et ceci, dit-il, est étranger à l'univers des beaux-arts. Il faut donc jeter les bases d'un cadre esthétique nouveau qu'il appelle « an aesthetics of behavior » (Penny, 2017a, p.356).

Cette approche met l'accent sur les sensations et les actions d'un corps structurellement couplé avec la machine. Elle examine le phénomène de l'interaction culturel en termes de boucles sensori-motrices, intégrant le ressenti et l'action de soi et de son environnement à travers un couplage dynamique, performatif et structurel (Penny, 2017a, p.357). L'interaction, dit Penny, est performative et implique une mise en corps. Le spectateur d'une œuvre interactive n'assiste donc pas seulement aux événements découlant de ses actions, mais aussi à ses propres sensations kinesthésiques et proprioceptives. L'effet esthétique d'un tel système sur le spectateur est donc « a combination of perceptions of the behavior of the system and awareness of the spectator's own activity and agency in the system » (Penny, 2017a, p.369). Finalement, Penny affirme qu'un système, pour être considéré comme interactif, doit nécessairement contenir des éléments capables de comportements et que le design de ces comportements doit donc précéder le design de l'interaction. En ce sens, il m'a inévitablement fallu mettre en place un réseau de neurones artificiels capable de réagir aux sons environnants pour faire de *Deep Duo* une œuvre interactive. Même si j'avais une idée préalable de la nature des interactions que je désirais voir exister, ce n'est qu'une fois que cet élément a été mis en place que j'ai pu réellement commencer à expérimenter et à travailler sur le design de l'interaction elle-même. Ce design a consisté à ajuster de différentes façons le système et l'étendue des comportements du réseau de neurones pour que les interactions aient l'impact esthétique souhaité.

L'esthétique des comportements n'est pas un cadre esthétique formel, mais plutôt une manière d'aborder et de réfléchir sur le design d'œuvres interactives. Elle suggère que les éléments en cause au sein de telles œuvres, humains et artificiels, peuvent être vus comme des agents en situation d'interaction. Créer une œuvre interactive implique donc avant tout le design des comportements d'agents artificiels structurellement couplé et interagissant avec leurs environnements.

Pour sa part, Audry s'intéresse à une facette bien précise de ce type d'œuvres d'art : des installations artistiques numériques adaptatives basées sur des agents artificiels qui utilisent des algorithmes d'apprentissage automatique. La nature des algorithmes d'apprentissage automatique permet de créer des œuvres dont le comportement peut changer dans le temps. En effet, lorsqu'on réentraîne un modèle sur la base de nouvelles données ou en changeant ses paramètres d'entraînement, il réagira différemment aux valeurs qu'il reçoit en entrée. Son comportement évoluera en cours d'entraînement, puis

se stabilisera de nouveau lorsque l'entraînement sera terminé. C'est cette caractéristique particulière qui amène le chercheur à proposer une « […] aesthetics of adaptive agents rooted in the distinctive way their behavior evolves and stabilizes as they couple with their environment » (Audry, 2018, p.3).

Ce cadre esthétique examine donc l'évolution des comportements dans le temps. Un comportement, rappelle Audry, bien qu'il implique une séquence d'événements toujours changeants, a une forme générale qui reste constante au fil du temps. Des exemples de cela existent avec l'art visuel génératif, tel que les animations de formes géométriques utilisées dans *Deep Duo* qui servent à illustrer les variations de valeurs des descripteurs audio analysant la voix. Les carrées et les cercles qui forment ces animations varient constamment de tailles et de positions à l'écran. La séquence de ces variations est toujours changeante et ne se répète pour ainsi dire jamais exactement de la même façon. Pourtant, assez rapidement, nous savons à quoi nous attendre de ces animations. Les variations ne nous surprennent plus, car nous avons perçu les limites et le contour des variations à l'écran. Ainsi, nous sentons que la forme de leurs comportements restera la même au fil du temps. Pour que la forme de leurs comportements change, il faudrait, par exemple, que l'étendue des variations de tailles et de positions change, ou encore que les formes géométriques elles-mêmes se transforment. Pour un moment, nous ne saurions plus à quoi nous attendre et il nous faudrait observer un moment ce nouveau paradigme avant de discerner la nouvelle forme de ces comportements.

Audry utilise différentes catégories qui permettent de classer des systèmes en fonctions des types de comportements dont ils font preuve. Ces catégories sont les comportements d'ordre zéro, d'ordre premier et d'ordre second. L'ordre zéro correspond à des systèmes dénués de comportements. Ces systèmes peuvent être comparés à des fonctions mathématiques. Ils n'ont pas de mémoire du passé et sont donc incapables d'accumuler de l'expérience : « Devoid of any kind of autonomy and agency […] their conduct relying almost entirely upon the data that is fed into them » (Audry, 2018, p.13). L'ordre premier est associé à des systèmes ayant une certaine forme de mémoire pouvant être modifiée par leurs interactions avec leur environnement. Leurs expériences passées laissent une trace qui influence leur comportement futur. Les visuels génératifs de *Morphogenèse* évoqués plus tôt sont un exemple de système d'ordre premier. Un tel système est capable de comportements, mais ces comportements ne changeront pas avec le temps : « Given

enough time, it will, inexorably, come to repeat similar patterns. » (Audry, 2018, p.14). Un système d'ordre second se distingue par sa capacité à changer son propre comportement. C'est exactement ce qui se passe lorsqu'un spectateur réentraîne le modèle à l'œuvre dans *Morphogenèse* avec de nouveaux paramètres. La forme des comportements du système, s'exprimant à travers les sons produits par les deux synthétiseurs, se métamorphose pour prendre une forme nouvelle, créée par le système lui-même. Cette différence fondamentale affecte l'effet esthétique que ces systèmes peuvent avoir et c'est dans cette catégorie que se trouvent les agents adaptatifs.

Le créateur qui travaille avec des systèmes d'ordre second, c'est-à-dire des systèmes utilisant des algorithmes d'apprentissage automatique, doit interagir avec ceux-ci pour mieux comprendre et canaliser les différents comportements qu'ils déploient. Ces comportements s'exprimeront par le biais des interactions qu'aura le modèle avec son environnement et, ultimement, auront un impact esthétique sur le spectateur. Pour mieux comprendre et utiliser ces systèmes dans un contexte créatif, Audry propose trois concepts qui caractérisent l'émergence et l'existence de leurs comportements. « […] the process whereby a behavior hovers around a stable state of being » (Audry, 2018, p.15) réfère au concept de *morphostasis*. Les motifs créés par ce type de processus changent avec le temps, mais éventuellement ces motifs se répéteront. La forme du comportement ne change donc pas. La forme du comportement ne change donc pas. Le processus de *morphogenesis* correspond au moment ou un comportement est en phase de changement. Tant que ce processus est actif, il sera impossible d'associer une forme définie aux comportements. La *metamorphosis* correspond au parcours de la forme d'un comportement qui, ayant été stable, donc en morphostasis, passe par un processus de morphogenesis puis se stabilise de nouveau. Il existe deux dimensions à cette metamorphosis : l'ampleur de la transformation encourue, qu'Audry nomme *metaboly*, et la vitesse avec laquelle le comportement est passé d'une forme à une autre.

Ces concepts permettent de réfléchir au comportement des agents adaptatifs actifs dans une œuvre d'un point de vue esthétique. Ainsi, lorsqu'un spectateur démarre un nouvel entraînement avec *Deep Duo*, celui-ci peut percevoir que le système entre dans un processus de metamorphosis qui, pendant un moment, modifiera de manière continue ses comportements et ébranlera le paradigme d'interaction qui s'était établi entre celui-ci et le spectateur. En fonction de l'ampleur et de la vitesse de cette transformation, les effets

de surprises et d'étonnements sur le spectateur seront variables. Lorsque le système se stabilise de nouveau, les premières interactions du spectateur lui permettront d'établir les bases d'un nouveau paradigme d'interaction, qui restera valide pendant toute la durée de cette phase de morphostasis.

L'esthétique des comportements adaptatifs propose donc une taxonomie qui permet d'identifier et de parler d'une catégorie spécifique de comportements, ceux rendus possibles par les algorithmes d'apprentissage automatique. Avec cette taxonomie, Sofian Audry joint sa voix à celle de Simon Penny pour soutenir qu'une esthétique de l'interaction entre l'ordinateur et l'humain doit passer par une esthétique des comportements. Il souligne aussi l'importance d'une mise en corps de cette interaction : « [...] in order to experience [...] behaviors in all their richness, one needs to "get to know them" phenomenologically, through her own sensorimotor body » (Audry, 2018, p.22).

3.4. Algorithmes d'apprentissage automatique comme outils de création

Dans cette section, des exemples pratiques d'utilisation d'algorithmes d'apprentissage automatique comme outils de création seront examinés. Tous ces exemples, à l'exception du premier, proviennent des créations faites lors de cette maîtrise. Ils seront une occasion de mettre en pratique l'approche de la mise en corps et l'esthétique des comportements tels que vus dans les propositions de Simon Penny et Sofian Audry. Le premier exemple est tiré du travail de la chercheuse Rebecca Fiebrink. Il justifie sa présence par le fait qu'il permet particulièrement bien de mettre en valeur les concepts évoqués dans ce chapitre jusqu'à maintenant. Cet exemple mènera naturellement à parler des différentes approches utilisées pour la création dynamique de mappages avec l'installation *Deep Duo*. Il sera ensuite le moment de voir comment la manipulation des données d'entraînements, l'ajustement du paramètre de température et la navigation de l'espace latent m'ont permis d'utiliser des modèles génératifs pour créer les effets esthétiques de l'installation *Morphogenèse*.

La création avec les algorithmes d'apprentissage impliquant une forme de collaboration entre l'artiste et la machine, il importe de savoir de quel type de créativité la machine est capable. Margaret Boden, chercheuse en sciences cognitives, écrit qu'il existe trois formes

de créativité : combinatoire, exploratoire et transformationnelle (2009, p.23). Selon elle, la créativité combinatoire, qui consiste à créer des associations inusitées à partir d'idées convenues, est celle que l'intelligence artificielle a le plus de mal à reproduire. La créativité exploratoire, qui se base sur l'exploration d'un espace conceptuel défini par un ensemble de règles souvent implicites, résonne bien avec le concept d'espace latent, concept que nous détaillerons plus loin de cette section. La créativité transformationnelle, quant à elle, peut être mise en relation avec le processus de metamorphosis vu précédemment, puisqu'elle consiste en une transformation de l'espace conceptuel lui-même et fait donc émerger des idées radicalement nouvelles.

3.4.1. Processus d'entraînement

Dans cette sous-section, différentes approches créatives avec les algorithmes d'apprentissage qui interviennent en amont du processus d'entraînement d'un modèle. Comme nous le verrons, ces approches consistent à modifier le contenu de la base de données, filtrer certains des éléments de la base de données, modifier l'architecture du modèle et ajuster les paramètres d'entraînement.

3.4.1.1. Mappage

La mise en relation des gestes du musicien avec les paramètres sonores d'un instrument, ou mappage, est un des aspects fondamentaux du design d'un instrument numérique : « [It] defines the ways a performer may move or act, the dimensions of musical engagement that are possible, the means for an audience to perceive the relationship between a performer's intention and the music, and so on » (Fiebrink, 2017, p.2). Les mappages multiples, comme le suggèrent les recherches de Hunt et Wanderley (2002), se prêtent mieux à des applications musicales que les mappages un à un, c'est-à-dire des « […] mappage[s] où chaque capteur n'affecte qu'un seul paramètre » (Marier, 2017, p.32). Mais réussir à créer un mappage multiple engageant et cohérent n'est pas une tâche simple. Avec son logiciel Wekinator, Rebecca Fiebrink propose de laisser des algorithmes d'apprentissage automatique effectuer ce travail à notre place. Le rôle du créateur consiste alors à fournir à l'algorithme une série d'exemples où des gestes musicaux, captés par les senseurs, sont mis en relation avec les valeurs de paramètres sonores de l'instrument. Sur la base de ces données, l'algorithme crée un modèle avec lequel le créateur peut expérimenter en manipulant directement l'instrument et en écoutant les sons produits. Il

pourra ensuite décider de modifier le comportement de l'instrument, en fournissant des exemples supplémentaires ou en retirant des exemples indésirables de la base de données, pour modifier, renforcer ou diminuer l'impact de certains gestes ou encore pour ajouter de nouvelles zones de jeux. Les algorithmes d'apprentissage, dit Fiebrink, « can facilitate new types of design outcomes [...]. But they are also valuable in facilitating new types of design processes allowing the instrument creation process to become a more exploratory, playful, embodied, expressive partnership between human and machine » (Fiebrink, 2017, p.1). Avec le processus mis de l'avant par Wekinator, l'utilisateur ne met pas en place des règles logiques, il utilise plutôt les gestes pour sentir comment le mappage se comporte et pour l'orienter dans la direction souhaitée. Cet exemple démontre bien comment la création avec un algorithme d'apprentissage peut être un processus exploratoire et itératif passant par un échange avec la machine. Cet échange, ou cette interaction, passe par cette mise en corps évoqué plus tôt qui permet de sentir le comportement du modèle à la lisière entre le geste et l'interface. Cette interaction est réellement significative puisque, contrairement à un clic de souris, c'est directement par elle que l'information transige et, ultimement, par elle qu'émerge un sens.

Avec *Deep Duo*, il n'est pas possible de modifier le contenu de la base de données, mais il est tout de même possible d'intervenir sur le comportement des modèles actifs dans le système. En permettant cela, mon objectif est de mettre en scène ces comportements et de rendre sensibles leurs variations. Ainsi, ceux-ci deviennent des éléments à part entière de l'œuvre à travers l'impact esthétique qu'ils ont sur l'interacteur.

3.4.1.2. Filtrage des caractéristiques

Le système derrière l'installation *Deep Duo* est en partie inspiré du travail de Fiebrink et de son Wekinator. Un réseau de neurones artificiels y est utilisé pour créer un mappage qui permet d'utiliser une source audio (la voix ou autre chose) pour générer des valeurs de voltages contrôlant un synthétiseur modulaire. Différemment d'avec Wekinator, les données d'entraînement ne sont pas générées en mettant en relation la source de contrôle et les paramètres de synthèses. Ces données mettent en relation des descripteurs audio qui analysent le signal sonore du synthétiseur et les valeurs de voltage de contrôle correspondant. L'idée derrière cette approche est de créer automatiquement une base de données qui prend en compte l'ensemble des possibilités sonore de l'instrument et d'essayer de faire en sorte que différentes sources sonores puissent contrôler le système,

plutôt qu'une source en particulier. Pour générer une base de données représentative de la relation entre descripteurs audio entrants et voltages de contrôle sortants, un algorithme parcourt l'étendue et les combinaisons possibles des paramètres du synthétiseur. Théoriquement, ce processus permet donc de créer automatiquement des bases de données pour différents types de systèmes de synthèses sonores. Le système permet à l'utilisateur de réentraîner le modèle sur de nouvelles bases. L'accent est donc mis sur la possibilité de changer le comportement du système en temps réel et d'expérimenter avec ces différents comportements. L'installation permet d'explorer ce que Sofian Audry appelle la metamorphosis puisqu'elle met en scène différents comportements ainsi que la transition entre ceux-ci. Comme avec Wekinator, cette expérience passe par une mise en corps puisque c'est par la voix que le spectateur interagit avec le modèle et sent les différents comportements qu'il exhibe.

Une des approches que propose *Deep Duo* pour changer le comportement du modèle est de sélectionner les descripteurs audio qui feront partie de la base de données lors de l'entraînement. Ainsi, la quantité totale d'exemples présents dans la base de données ne change pas, mais les descripteurs (*features*) présents dans chacun de ces exemples peuvent être filtrés. La base de données initiale contient, pour chaque exemple, une grande variété de descripteurs qui permettent d'analyser différentes caractéristiques d'un signal sonore. Entraîner le modèle avec différentes combinaisons de ces descripteurs influence donc la sensibilité qu'aura le modèle à différentes caractéristiques du son.

3.4.1.3. Architecture du modèle

Deep Duo permet aussi à l'interacteur de modifier l'architecture du modèle avant de le réentraîner. L'algorithme d'apprentissage derrière cette installation consiste en un réseau de neurones dont le nombre de couches cachées et le nombre de neurones contenu dans ces différentes couches peuvent être modifiés par le biais d'une interface graphique. Comme mentionné dans le premier chapitre, il n'y a pas de règles qui permettent de déterminer l'architecture idéale d'un réseau de neurones pour un problème donné. Différentes architectures aboutiront à des modèles aux comportements différents. L'interacteur peut donc expérimenter avec différentes configurations et interagir avec les modèles résultants.

3.4.1.4. Manipulation des paramètres d'entraînement

Plusieurs paramètres régissent le processus d'entraînement et influencent la forme du comportement d'un modèle. Comme expliqué au premier chapitre, l'interface de *Deep Duo* permet de choisir la taille du lot, le nombre d'époques, le taux d'apprentissage, la fonction d'activation, la fonction de perte et, encore une fois, d'expérimenter avec les modèles résultants. Par exemple, selon la fonction d'activation choisie, le modèle sera en mesure ou non d'avoir une réponse non linéaire aux données en entrée. Ou encore, augmenter le taux d'apprentissage risque d'accélérer le processus d'entraînement et donc la vitesse à laquelle le comportement du modèle changera pendant l'entraînement. Autrement dit, cela influencera la vitesse du processus de metamorphosis et, ultimement, aura différents effets sur l'expérience de l'interacteur.

3.4.1.5. Manipulation de la base de données

Le modèle responsable de la génération sonore dans l'installation *Morphogenèse* peut lui aussi être contrôlé à partir d'une source audio. Son fonctionnement consiste aussi à recevoir des données d'analyse sonore faite sur le signal en entrée et, en fonction de ceux-ci, générer des valeurs de contrôle d'un synthétiseur, qui dans ce cas est numérique. Ce dernier combine de la synthèse additive avec du bruit filtré et réussit ainsi, avec une quantité finie de paramètres, à imiter un large éventail de timbres. Les données en entrée proviennent d'une analyse de hauteur de note, d'amplitude et de MFCC. L'entraînement de ce modèle se fait à partir de fichiers sonores. Une quinzaine de minutes d'audio peuvent suffire pour que le modèle soit en mesure de reproduire le timbre d'un instrument. Pour *Morphogenèse,* je souhaitais que l'installation puisse :

- Générer différents timbres de voix humaine.
- Reproduire des phonèmes.
- Générer des sons de voix sans tonalité, tels que des chuchotements et des sons de respirations.

Pour que l'algorithme génère un modèle ayant ces capacités, il me fallait lui fournir des fichiers sonores contenant les différentes caractéristiques que je recherchais. Il me fallait aussi ajuster la proportion de ces différents types de timbres vocaux présents dans ces fichiers. En effet, si cet ensemble de fichiers ne contenait pas suffisamment de sons non consonants, le modèle peinait à imiter des sons de chuchotements. À l'opposé, une trop

grande quantité de sons non consonants aboutissait à une présence trop grande de ce type de sons. Ainsi, créer un modèle qui correspondait à ma vision nécessitait d'entraîner un modèle, d'expérimenter avec celui-ci, d'ajuster les fichiers sonores présents dans la base de données, d'entraîner un nouveau modèle et ainsi de suite jusqu'à arriver à un résultat convaincant. Pendant cette période de création, mon rôle n'était plus de coder, mais plutôt de développer une intuition du fonctionnement de cet algorithme qui me permettrait de l'amener à avoir le comportement que je désirais. Cette manipulation subjective de la base de données est caractéristique d'une utilisation créative d'un algorithme d'apprentissage automatique. Comme le dit Rebecca Fiebrink (2017, p.6), « [in] more conventional machine learning applications […] changing the training data is not a reasonable action to take to improve a model, because the training dataset is assumed to be a (more or less) accurate representation of some phenomenon in the world ». L'objectif d'un tel procédé est de créer un biais dans le modèle en fonction d'une vision créative. Ce qui est possible puisque les « [m]achine learning systems are subjective representational devices », « [they] are inherently biased toward the data they are fed » (Audry, 2021d, p.145).

3.4.2. Prédictions

Cette sous-section abordera d'autres types de manipulation créative d'un algorithme d'apprentissage automatique. Ceux-ci interviennent une fois que le modèle est entraîné, au moment de lui faire des demandes de prédiction. La manipulation des données envoyées en entrée du modèle, le paramètre de température et la navigation de l'espace latent seront les types de manipulations discutés.

3.4.2.1. Manipulation des données en entrée

Comme mentionné précédemment, le modèle de synthèse vocale utilisé dans l'installation *Morphogenèse* s'attend à recevoir en entrée différentes listes de données : des hauteurs de notes, des amplitudes sonores et des résultats d'analyse MFCC. Quand la voix d'un interacteur est détectée et enregistrée, le système derrière *Morphogenèse*, grâce à des outils d'analyse, génère ces listes d'information permettant la prédiction du modèle. En parallèle, il fait de la synthèse concaténative à partir des fichiers sonores ayant servi à l'entraînement du modèle. Cette synthèse permet de générer un fichier sonore fait d'un amalgame de segments de voix dont les timbres sont similaires. Ce fichier sonore est analysé de la même façon que celui de la voix de l'interacteur. Ainsi, deux listes de

hauteurs de notes, d'amplitudes sonores et d'analyses MFCC sont produites. Il devient alors possible de mélanger ces listes. Par exemple, les hauteurs de notes et les amplitudes en provenance de l'interacteur peuvent être mélangées avec l'analyse MFCC en provenance de la synthèse concaténative. J'ai mis ce système en place pour être en mesure de transfigurer la voix de l'interacteur vers différents timbres et articulations vocales. Le système permet aussi de transposer et de changer l'étendue des valeurs contenues dans ces listes avant qu'elles soient envoyées au modèle pour qu'il resynthétise la voix. Le processus permet de transposer la hauteur de la voix générée ou encore de modifier la force avec laquelle la voix chante. Encore une fois, ce genre de processus serait à proscrire dans le cadre d'une utilisation plus classique où l'on voudrait vérifier que le modèle est en mesure de faire des prédictions justes. Mais dans le cadre d'une utilisation créative, cela m'apparaît comme des moyens relativement simples et efficaces d'influencer le comportement du modèle et de l'orienter vers les zones désirées.

3.4.2.2. Température

L'algorithme de génération de mélodie utilisé dans l'installation *Morphogenèse* offre la possibilité d'ajuster un paramètre appelé température. Ce paramètre est disponible dans certaines configurations de réseaux de neurones artificiels. La valeur de ce paramètre permet d'ajuster le caractère aléatoire des prédictions. Une valeur d'un n'affecte pas le comportement du modèle. Augmenter cette valeur rendra l'algorithme plus téméraire alors que le diminuer le rendra plus conservateur. Dans le cas d'un modèle de génération de mélodie, cela se traduit par des mélodies harmoniquement et rythmiquement plus expérimentales et audacieuses ou, inversement, plus simples et répétitives. Dans le cadre de *Morphogenèse*, où plusieurs voix jouent simultanément pour créer des harmonies, ce paramètre permet d'ajuster le comportement de chacune de ces voix. Pour la voix grave, par exemple, la valeur de ce paramètre est plutôt basse, ce qui fait en sorte que la mélodie générée contient peu de notes et que ces notes sont majoritairement des toniques et des quintes. Les voix harmonisant la voix principale, quant à elles, ont une température plus élevée, de manière à créer des harmonies riches et des mélodies plus audacieuses que celle de la voix principale. Le paramètre de température me semble aussi être un moyen simple et efficace d'ajuster le comportement d'un modèle et, lorsque plusieurs instances d'un même modèle sont simultanément actives, d'élargir l'éventail des comportements que celles-ci peuvent exhiber.

3.4.2.3. Navigation de l'espace latent

Un espace latent est un « mathematical space that constitutes a distributed representation of data learned by a deep neural network » (Audry, 2021b, p.104). On peut l'imaginer comme un espace multidimensionnel dans lequel se trouve une multitude de points. Chacun de ces points correspond à un élément que le modèle est capable de générer. Une des choses qui rendent les espaces latents intéressants est la manière dont ces points sont organisés les uns par rapport aux autres. Plus la distance entre deux points est petite, plus les éléments auxquels ils correspondent sont similaires. Par exemple, dans le cas d'un modèle capable de générer des visages, la distance entre deux points de l'espace latent est proportionnelle à la similarité des traits, de la couleur des cheveux, de l'âge, etc. Aussi, la relation géométrique des points dans l'espace révèle une relation logique établie par le modèle. Avec le modèle qu'utilise *Morphogenèse* pour la génération de visage, cette caractéristique de l'espace latent fait en sorte que la distance et la direction séparant le visage d'un homme à l'expression neutre et le visage d'un homme souriant risque d'être approximativement les mêmes que celles séparant le visage d'un bébé à l'expression neutre et le visage d'un bébé souriant. Pour reprendre l'explication que Sofian Audry (2021b, p.109) fait de ce phénomène, cette logique, qui s'exprime par la phrase « un homme souriant est à un homme neutre ce qu'un bébé souriant est à un bébé neutre », s'exprime dans l'espace latent par l'opération : homme souriant – homme neutre + bébé neutre = bébé souriant. À l'aide d'opérations arithmétiques simples, il devient donc possible de naviguer l'espace latent et, par le fait même, le potentiel génératif d'un modèle.

Des techniques plus complexes ont aussi été mises sur pied pour faciliter la navigation dans un espace latent. Une de ces techniques s'appelle l'encodage. Elle consiste à trouver un point dans l'espace latent qui est similaire à un élément donné. Par exemple, l'algorithme utilisé pour la génération de visage dans *Morphogenèse*, encoder4editing, permet d'encoder la photo d'un visage dans l'espace latent. C'est-à-dire que l'algorithme trouve le point dans l'espace qui partage le plus de similarité avec l'image fournie. Cela permet de générer des visages ayant de grandes similarités avec celui de l'interacteur.

Une autre de ces techniques est l'analyse en composantes principales, une technique de réduction de dimensions. Elle permet d'interpréter plus facilement des ensembles de données tout en préservant un maximum d'informations. Elle fait ceci en identifiant des directions autour desquelles se trouvent de grandes variations de données. Autrement dit,

elle regroupe les points de l'espace latent autour de certaines tendances présentes dans l'organisation de ces points. Utiliser une de ces directions pour se promener dans l'espace latent équivaut à passer par une série de points qui se distinguent les uns des autres par une caractéristique principale. Ces directions étant calculées automatiquement, il faut tester chacune de celles-ci et tenter d'identifier la caractéristique qu'elles font varier. Ces directions donnent parfois des résultats inattendus. Par exemple, une des directions principales identifiées par une analyse en composante principale de l'espace latent du modèle de génération de visage de *Morphogenèse* fait varier la couleur de l'arrière-plan des visages, alors que les autres caractéristiques restent relativement stables. Effectivement, un des éléments qui varie le plus dans un portrait est l'arrière-plan. Mais cela ne correspond probablement pas à ce qui serait venu naturellement à l'esprit d'un humain si on lui demandait d'énumérer les caractéristiques qui varient le plus dans un ensemble de photos de visages. Or, les résultats d'une analyse en composantes principales sont disponibles pour l'espace latent du modèle de génération de visages utilisé dans *Morphogenèse*[10]. Les directions résultantes ayant été préalablement testées et identifiées par l'auteur, il m'a été possible de choisir et d'utiliser celles qui me semblaient pertinentes dans le cadre de ma création. Celles-ci agissent sur l'ouverture des yeux, l'ouverture de la bouche, l'âge, le genre et la direction horizontale et verticale de la source lumineuse (fig. 14).

10 https://github.com/harskish/ganspace/blob/65b0c4c7a4bbdcb5fedebb7c033dab59e27d61c0/notebooks/figure_teaser.ipynb

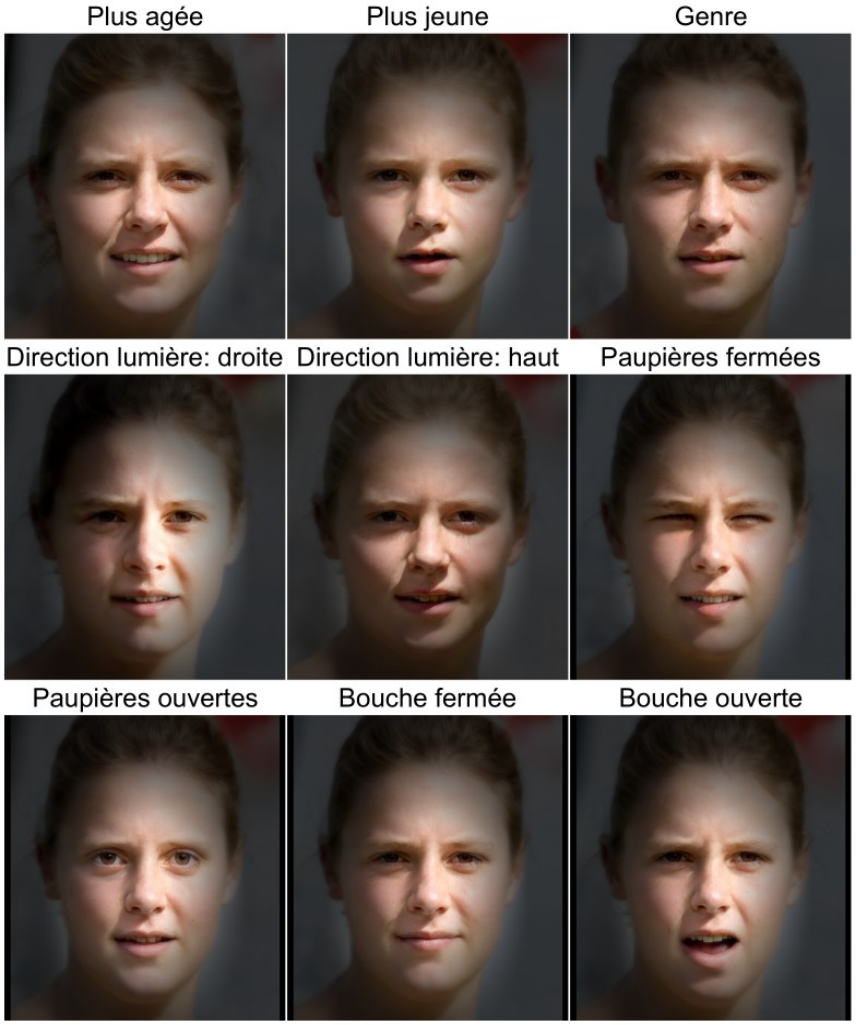

Figure 14 – Navigation de l'espace latent : exemples d'utilisation de directions.

L'ensemble de ces techniques sont utilisées pour créer les effets visuels de *Morphogenèse*. Des coordonnées de points dans l'espace latents sont choisies au hasard pour générer des visages artificiels. La technique de l'encodage permet de trouver d'autres points associés à des visages similaires à celui de l'interacteur. Une arithmétique simple permet de fluctuer légèrement autour de ces points pour donner vie aux visages. Une interpolation est appliquée à ces différents points qui font transiter les traits du visage fictif vers celui du visage ressemblant à l'interacteur. Le point résultant de cette interpolation transite sur des directions qui font varier l'âge, le genre, l'orientation et l'ouverture des yeux et de la bouche du visage généré. Chacune de ces opérations est appliquée algorithmiquement avec des intensités spécifiques et à des moments choisis pour créer une certaine narration

qui évolue en fonction des interventions et du temps que l'interacteur passe avec l'installation.

3.4.3. Algorithmie de modèles d'apprentissage

Plusieurs modèles d'apprentissage automatique sont actifs au sein de l'installation *Morphogenèse*. Ces modèles sont mis en commun et orchestrés de manière à générer le contenu sonore et visuel de l'installation en fonction des comportements de l'interacteur. La présence de l'interacteur, les sons qu'il émet et le temps qu'il passe devant la caméra deviennent des variables à la base d'un processus algorithmique qui module les comportements des modèles de génération mélodique, de synthèse vocale et de synthèse de visage.

Une des premières étapes pour me rendre à cette mise en commun fut d'apprendre à travailler avec chacun des modèles séparément. Ceci implique d'expérimenter avec eux, apprendre à connaître leurs potentiels génératifs, exposer certains de leurs paramètres et développer des processus qui permettent de les orienter vers des zones esthétiques spécifiques. Une fois tous ces éléments en place, il est devenu possible de les intégrer au sein d'une structure plus large qui allait permettre de contrôler algorithmiquement l'univers de *Morphogenèse*.

On peut dire d'un algorithme qu'il est « […] a finite sequence or structure of instructions […] » (McLean et Dean, 2018, p.2). Bien que les algorithmes d'apprentissage automatique marquent un changement de paradigme par rapport à une structure de programmation plus traditionnelle, une fois qu'ils sont entraînés et qu'une structure facilite la communication avec eux, il devient possible de les intégrer au sein d'une structure plus traditionnelle. Ainsi, dans cette séquence algorithmique, que l'on créera par le biais d'une série d'instructions, peuvent s'imbriquer des éléments créés automatiquement dont la structure et les comportements échappent à une compréhension procédurale.

Dans le cadre de *Morphogenèse*, cette algorithmie des modèles d'apprentissage permet d'obtenir des séquences musicales de voix chantée, créée grâce à des éléments générés par deux modèles distincts. Elle donne aussi à observer un visage, animé et généré par un modèle, mais donc les mouvements de la bouche dépendent des sons générés par un autre. Dans le cas de l'installation *Deep Duo*, la communication entre les modèles est bidirectionnelle et, en ce sens, il est possible de dire qu'il existe entre ceux-ci une forme

d'interaction. Effectivement, les sons produits par l'un influencent directement les sons produits par l'autre et vice versa. Bien sûr, les différents types de modèles pouvant être algorithmiquement mis en commun et les différentes manières de les placer en situation de communication ou d'interactions sont quasi infinis. Et c'est en ce sens que la mise en commun d'algorithmes d'apprentissage devient un outil puissant pour l'artiste. On peut imaginer que des procédés de ce genre pourraient survenir autant lors des étapes de préentraînement, pour qu'un modèle participe à l'entraînement d'un autre modèle, que lors des demandes de prédictions, pour que le comportement d'un modèle influence le comportement d'un autre. Dans tous les cas, il me semble que cette approche a un grand potentiel d'exploration et que, même si elle n'intervient pas directement sur les modèles eux-mêmes, elle a sa place dans une section sur les algorithmes d'apprentissage comme outils de création.

3.5. Conclusion du chapitre

La création de *Deep Duo* et de *Morphogenèse* m'a amené à explorer une pléiade d'approches pour utiliser les algorithmes d'apprentissage automatique en tant qu'outils de création. Ces approches interviennent tant en amont de l'entraînement des modèles qu'au moment de leur faire faire des prédictions. Elles interviennent sur la base de données, sur l'architecture du modèle, sur ses paramètres d'entraînement, sur les données qui lui sont envoyées en entrée, sur le paramètre de température, en naviguant l'espace latent des modèles génératifs et, finalement, sur une mise en commun de différents modèles par le biais d'un système algorithmique. Ultimement, toutes ces approches servent à influencer les valeurs en sortie des modèles. Les modèles d'apprentissage sont généralement trop complexes pour qu'on tente de les comprendre avec une pensée logique. Il semble donc cohérent et utile de penser ces valeurs en sortie des modèles en termes de comportements. La nature de ces algorithmes fait en sorte que ces comportements sont aussi capables de s'adapter et de changer. Il convient donc de prendre conscience des possibilités et de l'impact esthétique que peut avoir cette capacité de métamorphose au sein d'une œuvre d'art. Travailler avec ces modèles consiste donc à développer une connaissance instinctive de ces comportements et utiliser différentes approches pour les influencer. Si acquérir un sens commun de ces modèles passe moins par la logique que par une connaissance instinctive, ou holistique, il faut donc passer du

temps avec ceux-ci. Il faut interagir avec eux et, idéalement, s'engager corporellement dans cette interaction. Cet engagement corporel, qu'il passe par des gestes ou par la voix, permet d'engager les sens et, par le biais d'actions situées, de développer une connaissance intime de ces algorithmes qui permet de mieux travailler avec eux.

Conclusion

D'abord, j'espère que certains des éléments que ce document contient pourront contribuer à l'utilisation d'algorithmes d'apprentissage automatique comme outils de création.

Par quels moyens est-il possible d'exploiter le potentiel créatif de ces algorithmes? Y a-t-il des moyens qui permettent de se servir de ceux-ci afin qu'ils soient au service d'une vision artistique? Ces algorithmes sont nombreux et leurs degrés de complexités variés. Ainsi, avec certaines connaissances en programmation, il est possible de tripatouiller dans un réseau de neurones artificiels. Mais s'amuser à réimaginer un réseau antagoniste génératif demande considérablement plus de connaissances et n'est pas à la portée de tous. Certains de ces algorithmes, à cause de leurs complexités et des connaissances préalables qu'ils exigent, résistent donc plus fortement au processus d'appropriation. Par exemple, avec le réseau de neurones artificiels utilisé dans *Deep Duo*, j'ai été en mesure de mettre en place les blocs de base de l'algorithme et ainsi accéder et exposer un grand nombre de contrôles et de paramètres. Je n'ai pas modifié les équations mathématiques à la base de son fonctionnement, mais j'ai été en mesure d'avoir un contrôle au neurone près et par conséquent, à créer une œuvre qui permet d'expérimenter avec la structure même de l'algorithme. Je n'ai pu faire de même avec les modèles à l'œuvre dans *Morphogenèse*, ceux-ci étant d'une trop grande complexité pour mes connaissances. Il m'a donc fallu développer des stratégies différentes, dont l'influence sur la structure des algorithmes était plus limitée. Grâce à ces diverses stratégies, détaillées au chapitre trois, et à l'existence de modèles proche de mes besoins, j'ai pu malgré tout créer une œuvre reflétant assez bien la vision créative que j'avais. Mais je crois que ces limitations dans mes connaissances techniques des algorithmes en jeu ont participé à faire de cette œuvre une œuvre moins personnelle que *Deep Duo*, empreinte d'un esthétisme et ayant des comportements sur lesquels je n'avais pas le plein contrôle.

Est-ce que le contact d'une technologie à travers l'expérience que l'on fait d'une œuvre peut participer à la démocratisation de cette technologie? De manière générale, je crois que lorsqu'un artefact technologique se libère de l'emprise de ses concepteurs, puis de sa vocation primaire (généralement commerciale) et qu'elle aboutit dans les mains d'artistes, de bricoleurs ou autres curieux de tout acabit, il est destiné à perdre une part de son mystère. À force d'être utilisé en contexte créatif, il s'expose de plus en plus au savoir

collectif. Mais je crois aussi que le potentiel de démocratisation d'une œuvre est proportionnel à la capacité du créateur à s'approprier les technologies en jeu. Ainsi, une installation artistique comme *Deep Duo* permet de mettre en relation nos sens avec les comportements d'un modèle dont la structure est complètement à notre merci. Cette situation ne permet pas nécessairement de comprendre le fonctionnement technique de l'algorithme, mais elle permet de sentir, jusque dans ses extrêmes limites, les différents comportements que les modèles qu'il entraîne peuvent exhiber. Le système derrière *Morphogenèse* intervient plus en périphérie des modèles et cela se reflète sur la nature des interactions que l'on peut avoir avec eux. L'impact des interventions de l'interacteur est plus limité puisqu'il ne peut pas réellement influencer la structure des modèles. Ainsi, protégés par le design même du système, les algorithmes arrivent à préserver une plus grande part de mystère. Malgré ces limitations, je crois qu'à force d'interaction, l'installation permet tout de même de sentir peu à peu le contour de ces algorithmes. Je dirais donc qu'à divers degrés et en fonction des capacités et des intentions des créateurs, les œuvres d'art peuvent effectivement participer à la démocratisation de nouvelles technologies.

Cela mène finalement à la question de recherche posée par ce mémoire : comment peut-on adapter et utiliser les algorithmes d'apprentissage automatique comme outils pour créer des œuvres artistiques personnelles qui participeront à révéler au public la nature de ces algorithmes? Comme illustré au chapitre trois, je crois qu'une prise de conscience est nécessaire : la nature de ces algorithmes oblige à des processus différents de ceux auxquels nous sommes habitués. Ces processus consistent en un échange où le système ajuste ses comportements en fonction des informations qui lui sont fournies et où l'observation de ces comportements permet de mieux comprendre quelles informations doivent lui être fournir. Les règles logiques ne suffisent plus, il faut développer une connaissance plus holistique de l'algorithme. Cette connaissance, plus sensorielle qu'intellectuelle, se construit au fil de nos interactions avec le modèle. Plus ces interactions seront riches et engageront nos sens, plus intime sera la connaissance de l'algorithme. Le travail de création de *Deep Duo* et *Morphogenèse* a impliqué de longs moments d'interaction passés à solliciter, écouter et observer les comportements des différents modèles qui m'ont permis de mieux comprendre les manières de les influencer afin qu'ils portent ma vision créative. En fonction de la nature et de la complexité des modèles, j'ai mis en place différentes stratégies pour arriver à les influencer et les orienter.

Ces stratégies, expliquées en détail au chapitre trois, démontrent qu'il est possible d'utiliser les algorithmes d'apprentissage automatique comme outils de création. Cela étant dit, plus le créateur sera en maîtrise de ces algorithmes, plus il pourra s'en servir pour exprimer une vision personnelle.

L'utilisation de ces algorithmes à des fins créatives est encore très récente. Les recherches actuelles sont généralement préoccupées à optimiser la performance d'algorithmes existants ou à en créer de nouveaux. Mais peu de ces projets s'efforcent de réfléchir à des moyens qui permettraient d'avoir sur ces algorithmes des contrôles créatifs plus élaborés et subtils. De leur côté, les créateurs qui utilisent actuellement ces modèles se trouvent face à des outils ayant un potentiel énorme, mais qui offrent souvent un contrôle limité et qui sont difficiles à modifier étant donné leur grande complexité. Les créateurs de ces algorithmes pourraient sûrement s'inspirer de la pratique de certains artistes pour développer des contrôles réellement utiles à la création. De leur côté, les artistes, pour être en mesure de s'approprier davantage ces outils, gagneraient sûrement à travailler conjointement avec des spécialistes de ces technologies. Il me semble donc que, autant du point de vue du développement que du point de vue artistique, l'état actuel des choses appelle aux collaborations entre artistes, ingénieurs et scientifiques.

Références bibliographiques

Akten, Memo. 2021. « Deep Visual Instruments: Realtime Continuous, Meaningful Human Control over Deep Neural Networks for Creative Expression. » Thèse de doctorat, Computing, Goldsmiths, University of London.

Alpaydin, Ethem. 2020. *Introduction to machine learning*. MIT press.

Alzubi, Jafar*, et al.* 2018. « Machine learning from theory to algorithms: an overview. » Journal of physics: conference series.

Audry, Sofian. 2018. « Aesthetics of Adaptive Behaviors in Agent-based Art. » Proceedings of A Body of Knowledge - Embodied Cognition and the Arts Conference, Irvine, USA.

---. 2021a. *Art in the Age of Machine Learning*. MIT Press.

---. 2021b. « Deep Learning. » Dans *Art in the Age of Machine Learning*, 97-114. : MIT Press.

---. 2021c. « Introduction. » Dans *Art in the Age of Machine Learning*, 1-17. : MIT Press.

---. 2021d. « Watching and Dreaming. » Dans *Art in the Age of Machine Learning*, 143-52. : MIT Press.

Boden, Margaret A. 2009. « Computer models of creativity. » *AI Magazine* 30 (3): 23-23.

Brown, Tom*, et al.* 2020. « Language models are few-shot learners. » *Advances in neural information processing systems* 33: 1877-901.

Cangiano, Serena*, et al.* 2022. « Re-search, Re-enactment, Re-design, Re-programmed Art. » Dans *Over and Over and Over Again: Reenactment Strategies in Contemporary Arts and Theory*, 141-50. : ICI Berlin Press.

Das, Kajaree, et Rabi Narayan Behera. 2017. « A survey on machine learning: concept, algorithms and applications. » *International Journal of Innovative Research in Computer* 5 (2): 1301-09.

Fiebrink, Rebecca. 2017. « Machine learning as meta-instrument: Human-machine partnerships shaping expressive instrumental creation. » Dans *Musical instruments in the 21st century*, 137-51. : Springer.

Garwood, Deborah. 2007. « The Future of an Idea: 9 evenings—Forty Years Later. » *PAJ: A Journal of Performance and Art* 29 (1): 36-48.

Huang, Cheng-Zhi Anna*, et al.* 2018. « Music transformer. » *arXiv preprint arXiv:.04281*.

Hunt, Andy, et Marcelo M. Wanderley. 2002. « Mapping performer parameters to synthesis engines. » *Organised sound* 7 (2): 97-108.

Karras, Tero*, et al.* 2018. Flickr Faces HQ (FFHQ) 70K from StyleGAN.

Karras, Tero*, et al.* 2019. « A style-based generator architecture for generative adversarial networks. » Proceedings of the IEEE conference on computer vision and pattern recognition.

Marier, Martin. 2017. « Musiques pour éponge : la composition pour un nouvel instrument de musique numérique. » D. Mus., Université de Montréal.

McLean, Alex, et Roger T. Dean. 2018. « Musical Algorithms as Tools, Languages, and Partners: A Perspective. » Dans *The Oxford handbook of algorithmic music*, sous la direction de Alex McLean et Roger T. Dean. : Oxford University Press.

Nasteski, Vladimir. 2017. « An overview of the supervised machine learning methods. » *Horizons. b* 4: 51-62.

Pask, Gordon. 1971. « A comment, a case history and a plan. » Dans *Cybernetics, art and ideas*, sous la direction de Jasia Reichardt, 76-99. : Studio Vista London.

Peeters, Geoffroy. 2004. « A large set of audio features for sound description (similarity and classification) in the CUIDADO project. » *CUIDADO Ist Project Report* 54 (0): 1-25.

Penny, Simon. 2017a. « A Critical Aesthetics of Performative Technologies. » Dans *Making sense: Cognition, computing, art, and embodiment*, 355-71. : MIT Press.

---. 2017b. « Embodiment and Interaction. » Dans *Making sense: Cognition, computing, art, and embodiment*, 393-400. : MIT Press.

---. 2017c. « Introduction. » Dans *Making sense: Cognition, computing, art, and embodiment*. : MIT Press.

Priddy, Kevin L., et Paul E. Keller. 2005. *Artificial neural networks: an introduction*. Vol. 68. SPIE press.

Ramesh, Aditya, *et al.* 2022. « Hierarchical text-conditional image generation with clip latents. » *arXiv preprint arXiv:.06125*.

Ray, Susmita. 2019. « A quick review of machine learning algorithms. » 2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon).

Ruder, Sebastian. 2016. « An overview of gradient descent optimization algorithms. » *arXiv preprint arXiv:1609.04747*.

Salter, Chris. 2010. « Interaction. » Dans *Entangled: technology and the transformation of performance*, 302-48. : MIT Press.

Sharma, Sagar, *et al.* 2017. « Activation functions in neural networks. » *towards data science* 6 (12): 310-16.

Shaw, Jeffrey. 1996. « The dis-embodied re-embodied body. » *Kunstforum. Die Zukunft des Körpers I* 132: 168-71.

Suchman, Lucy A. 1987a. « Interactive Artifacts. » Dans *Plans and situated actions: The problem of human-machine communication*, 7-20. : Cambridge university press.

---. 1987b. « Situated Actions. » Dans *Plans and situated actions: The problem of human-machine communication*, 35-46. : Cambridge university press.

Thibodeau-Laufer, Éric. 2014. « Algorithmes d'apprentissage profonds supervisés et non-supervisés: applications et résultats théoriques. » Mémoire de maîtrise, Informatique, Université de Montréal.

Tov, Omer, *et al.* 2021. « Designing an encoder for stylegan image manipulation. » *ACM Transactions on Graphics* 40 (4): 1-14.

Waite, Elliot. 2016. « Generating Long-Term Structure in Songs and Stories. ». Consulté le 12 novembre 2022. https://magenta.tensorflow.org/2016/07/15/lookback-rnn-attention-rnn/.

Wilkins, Julia, *et al.* 2018. VocalSet: A Singing Voice Dataset. : Zenodo.

Annexes

Tous les extraits vidéos et sonores sont disponibles sous forme de fichiers complémentaires. Ces fichiers portent le même nom que les titres donnés à chacun des extraits dans le texte.

Extrait du chapitre 1

Extrait 1 – *Deep Duo* : documentation vidéo. [vidéo]

Extraits du chapitre 2

Extrait 2 - *Morphogenèse* : documentation vidéo. [vidéo]

Extrait 3 - *Morphogenèse* : son original. [audio]

Extrait 4 - *Morphogenèse* : son encodé. [audio]

Extrait 5 - *Morphogenèse* : voix 1. [audio]

Extrait 6 - *Morphogenèse* : voix 2. [audio]

Extrait 7 - *Morphogenèse* : voix 3. [audio]

Extrait 8 - *Morphogenèse* : voix 4. [audio]

Extrait 9 - *Morphogenèse* : voix 5. [audio]

Extrait 10 - *Morphogenèse* : voix 6. [audio]