SCIENCES SORBONNE UNIVERSITÉ

Projet en traitement avancé du son - M2 ISI 2020/2021

Groupe 7 / Sujet 4.1 : Reconnaissance de genres musicaux

RACHEDI Nasr Eddine - TOUATI Tania Camelia - YAHIA Yacine

1 Présentation et objectif du projet

La musique est complètement partie prenante de notre vie quotidienne. Selon un rapport de la Fédération Internationale de l'Industrie Phonographique (IFPI) publié en 2019, nous consommons en moyenne 18 heures de musique par semaine. Face à cet engouement, de nouveaux services ont fait leur apparition, à l'instar des systèmes de recommandation de musique ou encore des générateurs de playlists. Ces services innovants se basent en partie sur la reconnaissance de sons et plus précisément, l'indexation de contenus audio.

C'est dans ce cadre là que s'inscrit notre projet. Il a pour objectif de classer des morceaux musicaux suivant leurs genres, à travers la mise en pratique des techniques d'apprentissage automatique profond visant une performance aussi bonne voire meilleure que celle de l'oreille humaine.

Il est à noter que plusieurs approches existent pour la classification de genres (machines à vecteurs de support multi classe, K-moyennes, K plus proches voisins) mais nous nous concentrerons dans ce projet uniquement sur les réseaux de neurones Feedforward et convolutifs.

Dans un premier temps, nous nous intéresserons au traitement des données qui serviront d'entrée au réseau de neurones (section 2) puis, nous détaillerons l'architecture des modèles employés pour la classification (section 3). Enfin, nous terminerons sur une discussion des résultats obtenus avant de conclure.

L'intégralité de notre projet peut être schématisé par la figure 1.

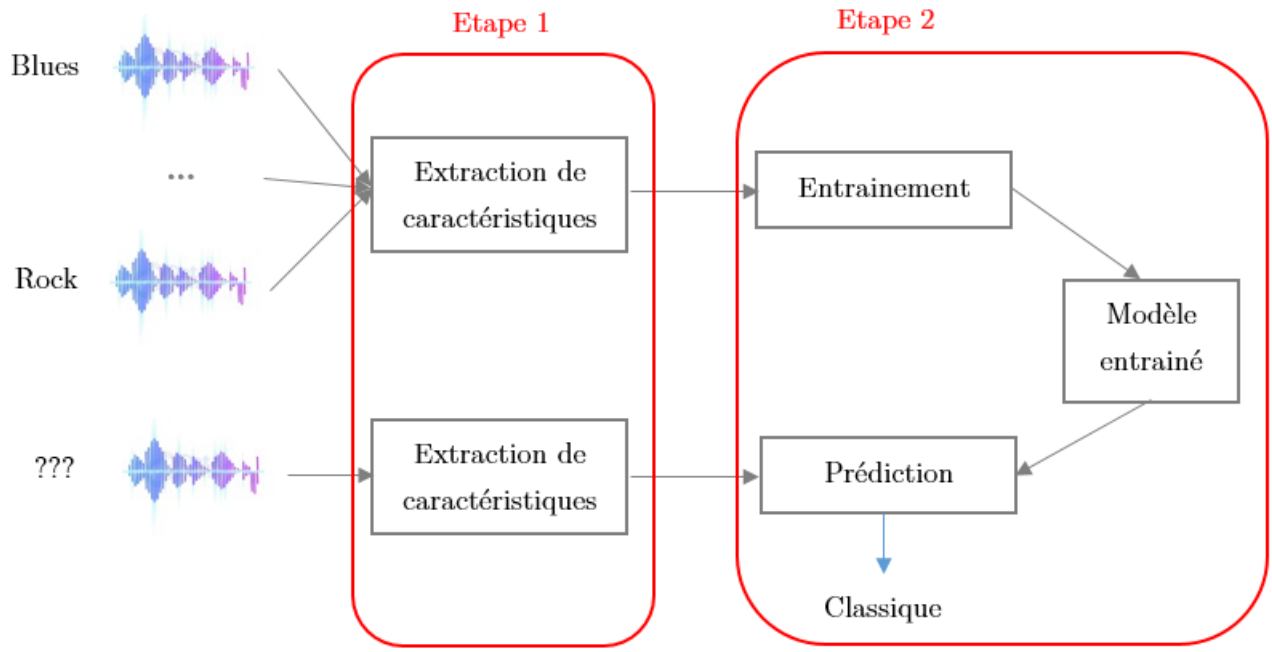


FIGURE 1: Schématisation des étapes du projet

2 Étape 1 : Traitement de la base de données

2.1 Description et objectifs

Base de données GTZAN
Le jeu de données dont nous disposons est le GTZAN. C'est une collection de genres qui a été utilisée dans un article populaire traitant de la classification de genres en 2002 [1].
GTZAN est constitué de 10 genres (blues, classique, country, disco, hip hop, jazz, metal, pop, reggae et rock), avec 100 extraits de 30 secondes par genre. Les extraits y sont organisés dans des dossiers, chacun portant le nom d'un genre, et enregistrés sous format .wav.

Problématique : Comment transformer des extraits audio en informations exploitables pour une tâche de classification ?
La première étape, et sans doute la plus cruciale, du projet de classification des genres musicaux consiste à préparer la base de données sur laquelle sera effectué l'apprentissage par la suite. Il s'agit donc d'extraire des caractéristiques de différenciation à partir de données audio qui pourraient être introduites dans un modèle.

Solution : Plusieurs solutions s'offrent à nous. En effet, il est possible d'extraire des caractéristiques numériques propres à chaque extrait et de les stocker dans un vecteur qu'on donnera en entrée au modèle ou encore d'exploiter des représentations visuelles, et donc plus intuitives à notre sens, telles que les spectrogrammes. C'est sur le spectrogramme Mel et les MFCC (Mel-Frequency Cepstral Coefficients) que notre choix s'est porté pour ce projet. Ces deux représentations peuvent être générées en quelques lignes de code à l'aide de la bibliothèque Librosa [2] pour le traitement audio et musical sous Python.

Le spectrogramme Mel
Le spectrogramme Mel est une représentation visuelle d'un signal audio qui illustre la variation du spectre fréquentiel dans le temps. Le spectrogramme est obtenu par transformation de Fourier sur des fenêtres du signal qui se chevauchent puis, les fréquences sont mises à l'échelle Mel.

Les MFCC (Mel-Frequency Cepstral Coefficients)
Les MFCC sont des coefficients cepstraux calculés par une transformée en cosinus discrète appliquée au spectre de puissance d'un signal. Les bandes de fréquence de ce spectre sont espacées logarithmiquement selon l'échelle de Mel.

Les différences idiosyncratiques entre les genres sont bien capturées dans les deux représentations mentionnées pus haut. Ainsi, dans les deux cas, il est tout à fait envisageable de transformer le problème en une tâche de classification d'images.

2.2 Configuration expérimentale

- On commence par récupérer les chemins des fichiers .wav contenus dans la base de données GTZAN. Pour cela, on a écrit une fonction **[fichier] = listdirectory(path)** qui reçoit en entrée le chemin d'un dossier donné (blues, par exemple) et renvoie en sortie une liste des chemins de tous les fichiers contenus dans ce dossier (les 100 extraits de blues).

- On stocke ensuite les chemins récupérés dans un dataframe de taille 10×100 où les lignes correspondent aux 10 genres musicaux et les colonnes aux 100 extraits.

On écrit une fonction **[Mel, labels] = segmentation_mel(chemin,label)** (**[mfcc, labels] = segmentation_mfcc(chemin,label)**, respectivement) qui reçoit en entrée les chemins des extraits audio ainsi que leurs labels et retourne en sortie les signaux exploitables donc les spectrogrammes Mel (les MFCC, respectivement) ainsi que leurs labels associés.

Dans cette fonction, nous avons utilisé la méthode **librosa.load** qui retourne un vecteur d'échantillons audio ainsi que le taux d'échantillonnage à partir du chemin d'un fichier au format .wav. Nous avons également utilisé **librosa.feature.mfcc** et **librosa.feature.melspectrogram** pour générer les MFCC et les spectrogrammes Mel, respectivement. Ces deux dernières méthodes reçoivent en entrée un segment (cela sera expliqué par la suite dans la section discussion) du vecteur des échantillons audio et un certain nombre de paramètres que nous avons choisi comme suit : $n\_mfcc = 13$, $n\_fft = 2048$, $hop\_length = 512$ en nous basant sur ce qui est le plus employé dans la littérature.

- On applique la fonction précédente sur notre Dataframe.

- On partitionne ensuite notre nouveau jeu de données à l'aide de la fonction **train_test_split** en consacrant 70% pour l'entraînement et 30% pour le test.

2.3 Résultats obtenus

A l'issue du traitement de la base de données GTZAN, nous avons pu obtenir deux nouvelles bases de données labelisées et totalement exploitables par un réseau de neurones.
La première, correspondant aux coefficients MFCC, constituée de 9996 exemples de taille 130×13 (à noter que 4 extraits dont la taille était différente de 130×13 ont été supprimés).
La seconde, correspondant aux spectrogrammes Mel, composée également de 9996 exemples de taille 130×13.
La visualisation de ces deux bases est donnée par les figures 2 et 2.

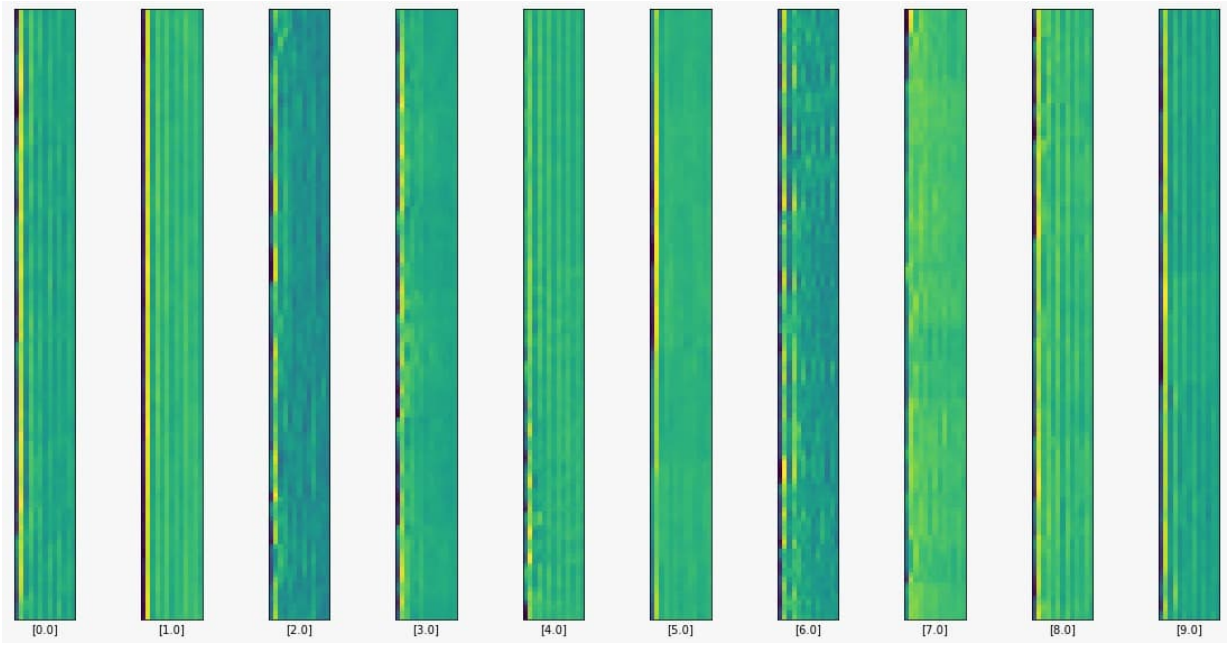


FIGURE 2: MFCC du premier extrait de chaque genre musical

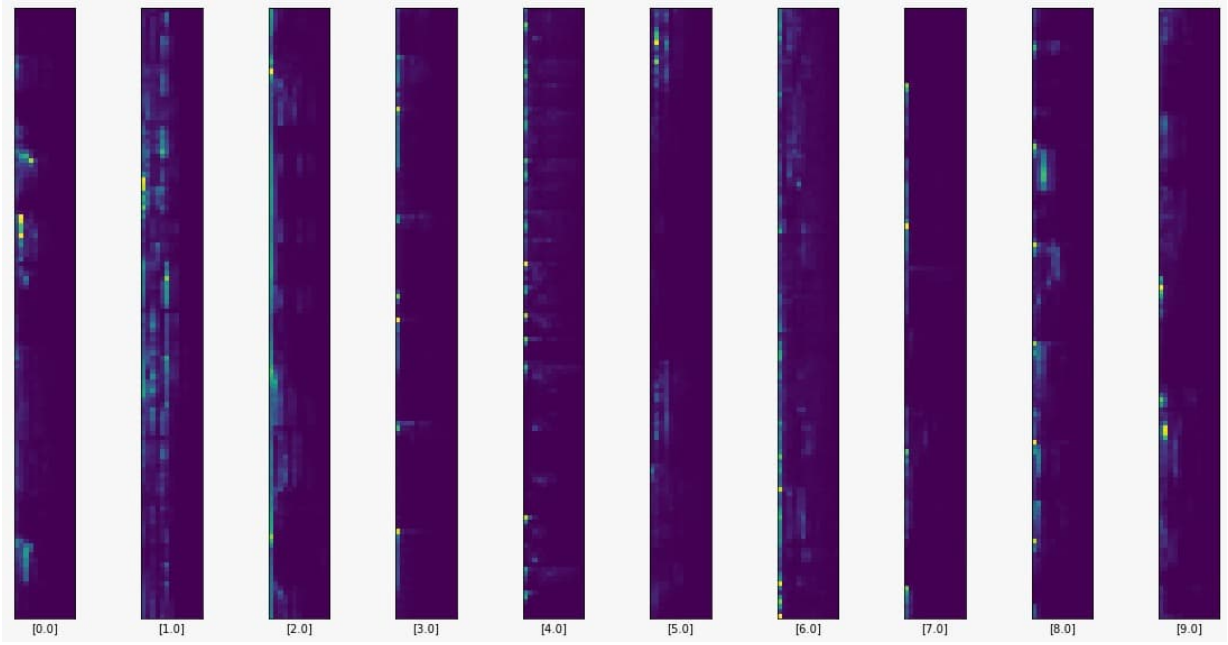


FIGURE 3: Mel spectrogrammes du premier extrait de chaque genre musical

2.4 Discussion

Afin de nous assurer de la cohérence de nos bases de données, nous avons conduit des essais préliminaires sur celles-ci en tentant de premiers tests de classification. Les résultats étant très mauvais, nous avons d'abord commencé par changer l'architecture de notre modèle mais en vain. En effet, malgré ses caractéristiques attrayantes, GTZAN présente un certain nombre d'inconvénients, notamment le fait qu'elle contienne seulement 100 éléments par classe, ce qui est relativement petit dans le contexte de l'apprentissage automatique.
C'est ainsi que nous avons décidé d'explorer une autre piste et de ré-exploiter nos données.

En effet, la base de données initiale composées de 1000 exemples est trop petite pour pouvoir

donner des résultats pertinents lors de la classification. C'est pour cette raison que nous avons décidé d'augmenter la taille de la base en segmentant les extraits. Ainsi, chaque extrait de 30 secondes a été subdivisé en 10 de 3 secondes, multipliant de ce fait le nombre d'exemples par 10. Ce qui explique comment nous sommes passés d'une base de 1000 à 10000 (9996) exemples.

3 Étape 2 : Classification des données

3.1 Description et objectifs

Pour réaliser la tâche demandée qui consiste à classifier des extraits musicaux suivant leurs genres, on construit dans cette partie des réseaux de neurones avec deux architectures distinctes : un réseau de neurones Feedforward et un réseau de neurones convolutif. La seconde architecture est ensuite testée sur les deux bases de données introduites à l'étape précédente.

La classification des genres musicaux a pour objectif d'identifier chaque morceau musical passé en entrée du réseau et de l'affecter en sortie à la classe qui lui correspond. Dans la base de données GTZAN, on dispose de 10 genres musicaux, ce qui signifie que les architectures doivent prendre en entrée des morceaux de même taille et en sortie une couche de 10 neurones.

3.2 Configuration expérimentale

3.2.1 Réseau de neurones Feedforward

On commence par instancier un réseau de neurones Feedforward pour avoir, dans un premier temps, un aperçu de la qualité des résultats. L'architecture proposée est détaillée ci-dessous et illustrée par la figure 4.

- Une première couche de 512 neurones avec une fonction d'activation 'relu' ;
- Une deuxième couche de 256 neurones avec une fonction d'activation 'relu' ;
- Une troisième couche de 64 neurones avec une fonction d'activation 'relu' ;
- Une couche de sortie de 10 neurones avec une fonction d'activation 'softmax'.

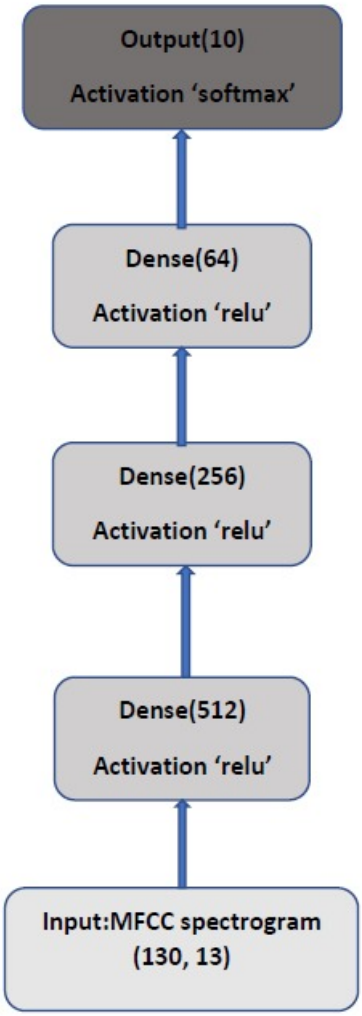


FIGURE 4: Architecture du réseau de neurones Feedforward

3.2.2 Réseau de neurones convolutif

Les réseaux de neurones convolutifs désignent une sous-catégorie de réseaux de neurones qui interviennent dans le cas où les entrées du réseau sont des images. Le principe de cette méthode est basé sur l'extraction et le zoom automatique des caractéristiques. De ce fait, ils sont adaptés à la tâche que nous souhaitons accomplir puisque nos entrées sont des spectrogrammes. L'architecture proposée est détaillée ci-dessous et illustrée par la figure 5.

- 256 filtres de taille 3×3 appliqués à l'entrée dans la première couche convolutive avec une fonction d'activation 'relu' suivis d'un max pooling de taille 3×3, d'un stride de 2×2 et d'une normalisation par batch ;
- 256 filtres de taille 3×3 avec une fonction d'activation 'relu' suivis d'un max pooling de taille 3×3, d'un stride de 2×2 et d'une normalisation par batch ;
- 128 filtres de taille 2×2 avec une fonction d'activation 'relu' suivis d'un max pooling de taille 2×2, d'un stride de 2×2 et d'une normalisation par batch ;
- Une couche dense de 256 neurones avec une fonction d'activation 'relu' ;
- Une couche dense de 128 neurones avec une fonction d'activation 'relu' ;
- Une couche dense de 64 neurones avec une fonction d'activation 'tanh' ;

- Une couche dense de 32 neurones avec une fonction d'activation 'tanh' et un dropout de 0.5 ;
- Une couche de sortie de 10 neurones avec une fonction d'activation 'softmax'.

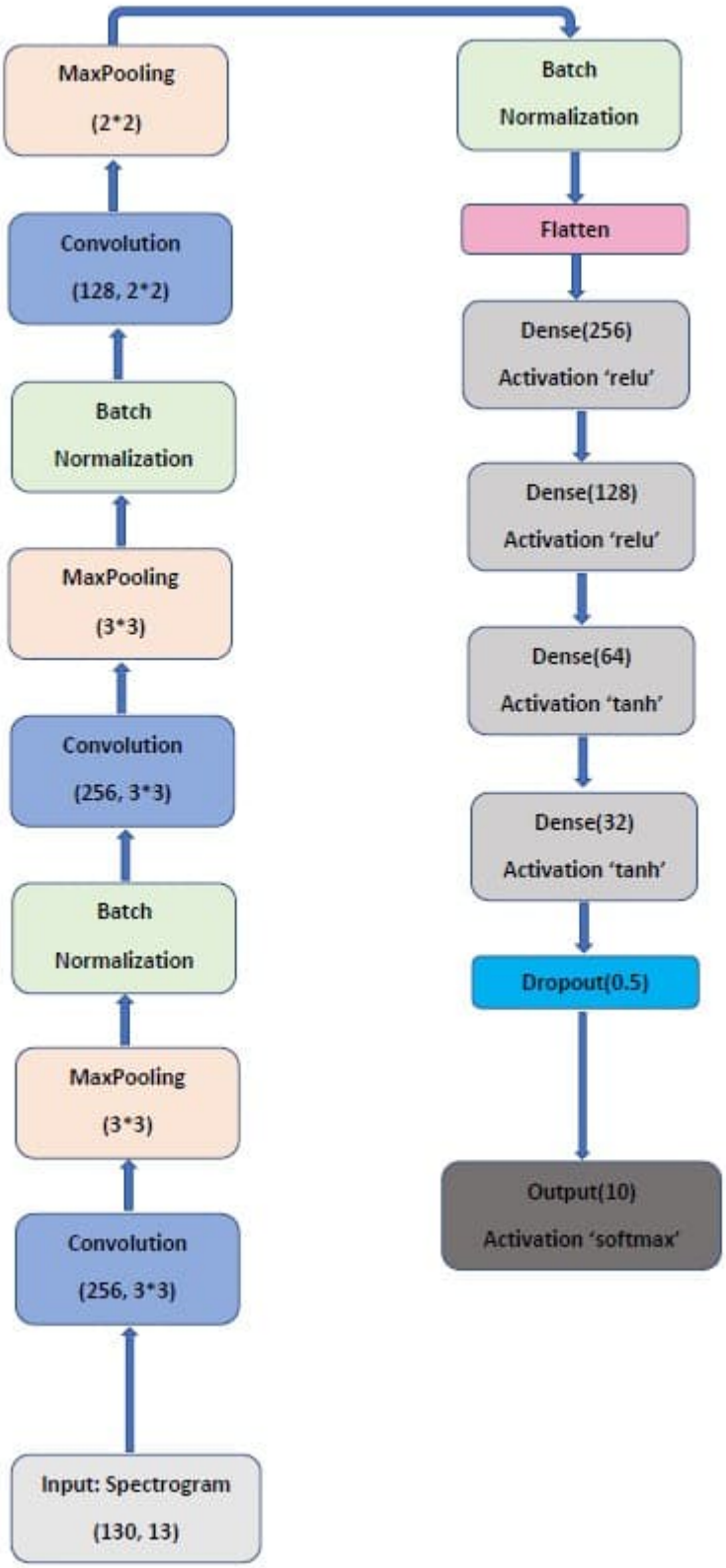


FIGURE 5: Architecture du réseau de neurones convolutif

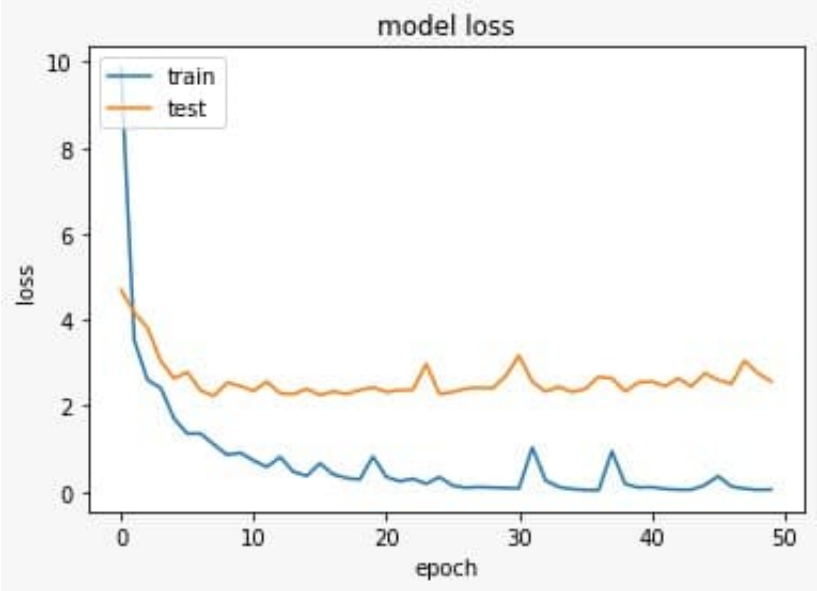
3.2.3 Apprentissage

L'apprentissage nécessite la définition de certains paramètres, en l'occurrence une fonction coût à optimiser, un algorithme d'optimisation et une précision. A cet effet, on opte pour la fonction 'sparse categorical crossentropy', l'algorithme 'Adam' avec un pas d'apprentissage de 0.0001 et une précision de type 'accuracy'.
On utilise la méthode 'fit' pour l'apprentissage avec un batch_size de taille 32, 50 epochs et un validation split de 0.3, qui permet d'allouer 30% des données d'entraînement pour la validation du modèle lors de l'apprentissage.

3.3 Résultats obtenus

3.3.1 Sur la base des MFCC

- Avec la première architecture (**réseau de neurones Feedforward**) appliquée sur la base des MFCC, on obtient un taux d'apprentissage élevé de 98%. On constate cependant sur la base de validation une précision de 58% après 50 epochs. Sur la base de test, la précision obtenue est de 57.58%. A partir des ces résultats, il est clair qu'il y a un surapprentissage qui s'est produit, probablement en raison de l'architecture choisie, ce qui nous a donc conduit à faire d'autres essais avec des CNN.



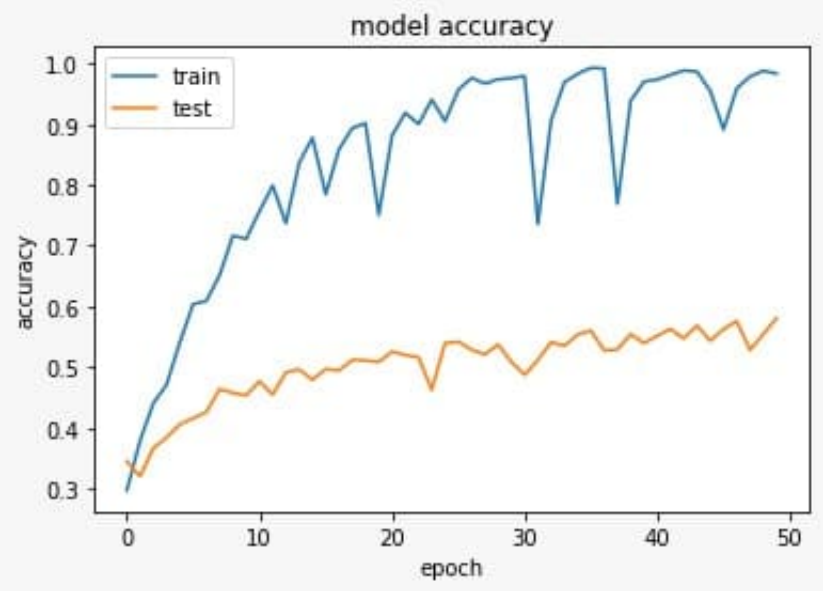
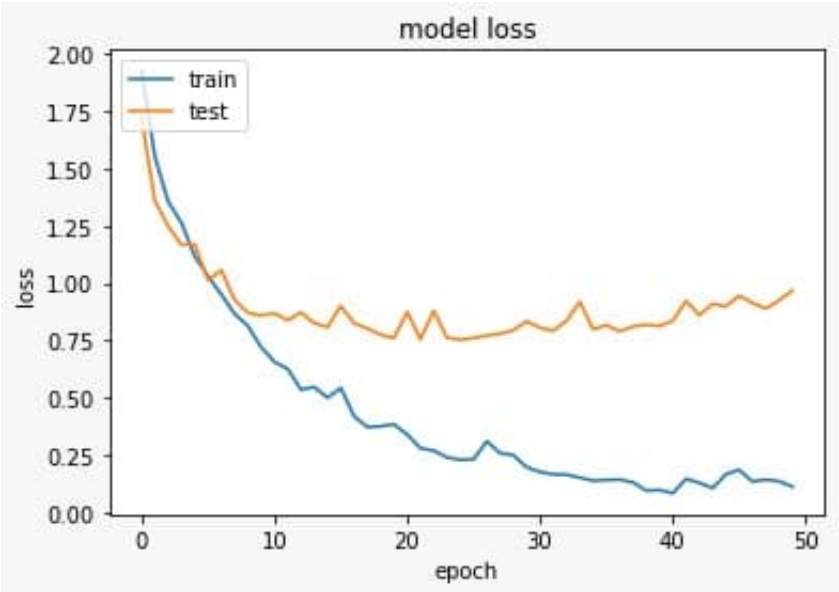


FIGURE 6: Courbes de la fonction coût et de la précision en fonction des epochs avec l'architecture Feedforward sur la base des MFCC

Sur la figure 6, on observe une différence significative entre la précision de la base d'apprentissage et celle de la base de test. De même, sur la courbe de l'erreur, on constate que l'erreur d'apprentissage diminue contrairement à celle du test qui croit au fil des epochs, ce qui confirme la présence du phénomène de surapprentissage.

- Avec la seconde architecture (**réseau de neurones convolutif**) appliquée sur la même base des MFCC, on obtient un taux d'apprentissage de 97.75% mais cette fois-ci, avec une précision de 70% après 50 epochs sur la base de validation. Pour ce qui est de la base de test, nous avons obtenu une précision de 75.69%. Cela signifie que le modèle arrive à généraliser et parvient à reconnaître plus de 75% des genres musicaux sur des extraits qu'il n'a jamais vu.



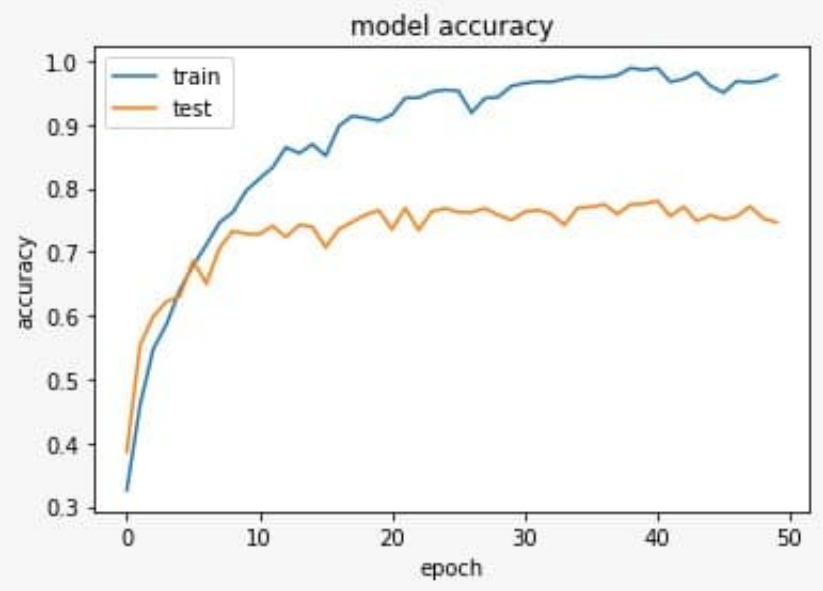


FIGURE 7: Courbes de la fonction coût et de la précision en fonction des epochs avec l'architecture convolutive sur la base des MFCC

A partir de la figure 7, on constate que les précisions sont bien meilleures et les erreurs moindres qu'avec l'architecture précédente car les valeurs de l'apprentissage et du test sont quasiment égales après 50 epochs, et le taux de reconnaissance est de 75%.

La matrice de confusion résultante est donnée sur la figure 8.

Genres	Blues	Classique	Country	Disco	Hiphop	Jazz	Metal	Pop	Reggae	Rock
Blues	84	1	2	3	1	4	6	0	1	3
Classique	0	92	2	0	0	4	0	0	0	0
Country	7	1	76	5	1	7	2	5	3	20
Disco	0	0	3	70	5	1	2	5	6	8
Hiphop	3	1	1	5	73	0	5	6	6	1
Jazz	1	5	1	0	1	78	0	1	1	1
Metal	0	0	0	0	1	0	77	0	1	6
Pop	0	0	4	2	6	1	0	75	3	2
Reggae	0	0	2	7	10	1	1	3	74	2
Rock	3	0	10	8	3	4	8	5	4	57

FIGURE 8: Matrice de confusion du réseau de convolution avec la base des MFCC

3.3.2 Sur la base des Mel spectrogrammes

Étant donnés les résultats encourageants obtenus avec l'architecture convolutive, nous avons voulu la tester sur notre seconde base de données contenant les spectrogrammes Mel.

A l'issue de l'apprentissage, la précision obtenue a convergé vers 76.22% sur la base d'entraînement, 59.29% sur la base de validation et 75.69% sur la base de test.

Le coût ainsi que la précision de l'architecture convolutive sur la base des spectrogrammes Mel sont donnés par la figure 9.

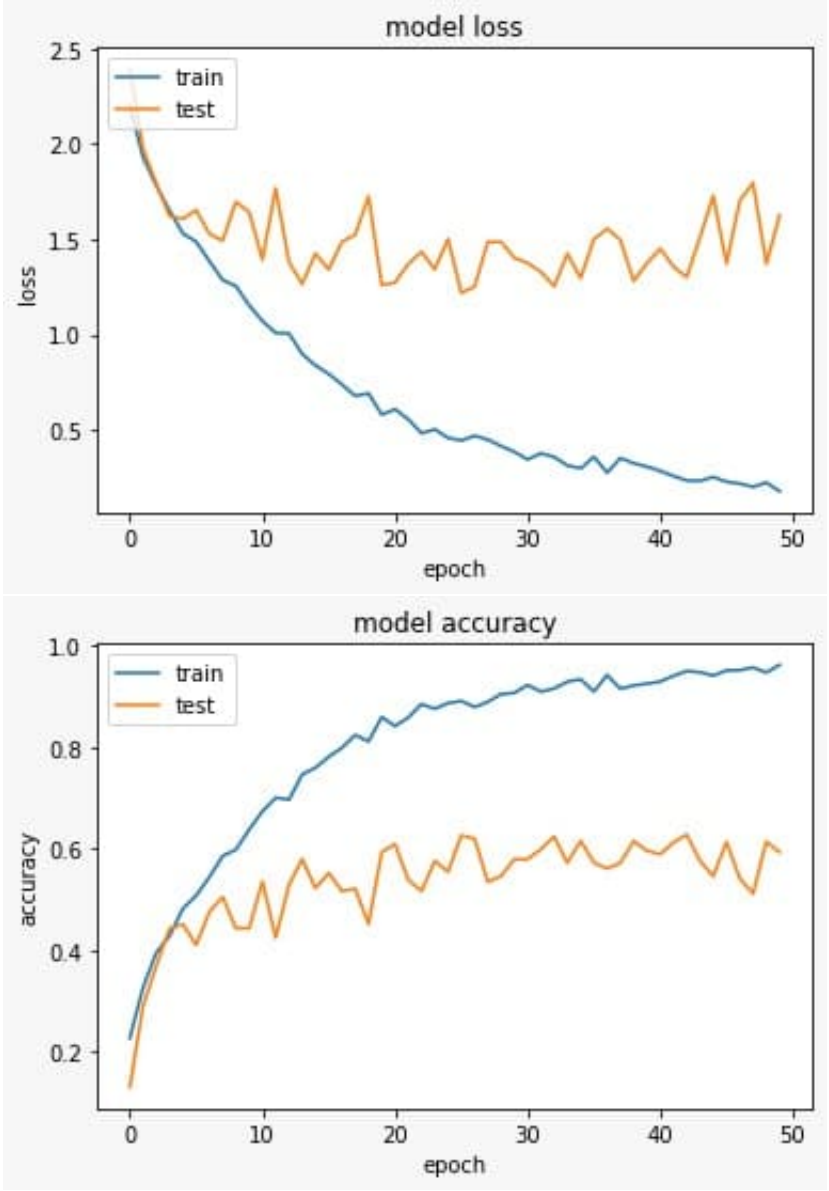


FIGURE 9: Courbes de la fonction coût et de la précision en fonction des epochs avec l'architecture convolutive sur la base des spectrogrammes Mel

La matrice de confusion obtenue est illustrée dans la figure 10.

Genres	Blues	Classique	Country	Disco	Hiphop	Jazz	Metal	Pop	Reggae	Rock
Blues	53	3	12	4	2	6	3	3	5	6
Classique	1	47	1	0	0	2	0	0	0	2
Country	9	6	58	3	1	10	2	5	5	12
Disco	6	3	2	56	5	1	2	10	3	7
Hiphop	2	1	0	5	61	0	4	9	5	3
Jazz	4	22	4	0	0	67	0	0	1	0
Metal	2	2	2	2	5	3	82	1	0	19
Pop	6	2	6	10	8	2	0	64	5	7
Reggae	7	4	4	10	13	1	0	5	71	4
Rock	11	9	11	10	4	6	7	4	5	41

FIGURE 10: Matrice de confusion du réseau de convolution avec la base des spectrogrammes Mel

3.4 Discussion

Sur la base des MFCC, les deux architectures fournissent presque la même précision sur la base d'entraînement, dépassant les 95%. Cependant, à la différence du réseau convolutif, le Feedforward présente un surapprentissage. Cela nous permet de juger que le réseau convolutif est plus performant pour la tâche de classification, probablement car il tient compte de l'organisation spatiale des caractéristiques des signaux.

La même architecture convolutive déployée sur la base des spectrogrammes Mel donne des résultats beaucoup moins bons que ceux obtenus sur la base des MFCC.

Chacune des colonnes dans la matrice de confusion représente les classes prédites sur la base de test des genres musicaux tandis que les lignes représentent les classes réelles. Les pourcentages au niveau de la diagonale indiquent la proportion des classes correctement prédites par le réseau. Ainsi, nous remarquons que ces pourcentages sont élevés avec les MFCC et bas avec les spectrogrammes Mel, ce qui confirme notre analyse.
Le réseau entraîné sur la base des MFCC parvient à classifier tous les genres avec plus ou moins d'exactitude, à l'exception du rock qui présente le pourcentage le plus bas (57% seulement). De plus, il y a une confusion significative entre certains genres musicaux tel que la country et le rock (20%).
Le réseau entraîné sur la base des spectrogrammes Mel a du mal à classifier les genres musicaux de manière correcte d'une part, et présente une confusion plus importante sur presque toutes les classes, en particulier le jazz, le classique, le metal et le rock.

4 Résultats et commentaires

L'oreille humaine est capable de distinguer les genres musicaux avec une précision de l'ordre de 70% [3]. Nous jugerons donc qu'un modèle est performant s'il arrive à atteindre une précision similaire ou supérieure à la performance humaine.

A la lumière de cet élément, il est clair que l'architecture convolutive surpasse la performance humaine en terme de reconnaissance de genres musicaux tandis que l'architecture Feedforward est beaucoup moins efficace. Notre étude révèle également que le réseau de neurones convolutif entraîné sur la base des MFCC donne de meilleurs résultats que ceux obtenus par le même réseau entraîné sur la base des spectrogrammes Mel. Nous pouvons donc conclure que les caractéristiques propres aux extraits musicaux sont mieux conservées dans les MFCC que dans les spectrogrammes Mel.

Enfin, pour évaluer la pertinence de nos résultats, nous avons voulu les mesurer à ce qui est disponible dans la littérature. Un article publié en 2019 [4] présente une architecture basée sur des couches convolutives dupliquées dont les sorties sont connectées à différentes couches de pooling et prétend aboutir à une précision de 90.7% sur la base de données GTZAN. Nous avons repris cette même architecture mais ne sommes pas parvenus aux mêmes résultats. De plus, les scores de prédiction obtenus sont nettement inférieures à ceux issus de notre réseau convolutif entraîné sur les MFCC.

5 Conclusion

Au cours de ce projet, nous avons étudié deux architectures pour la classification des genres musicaux, et exploité deux bases de données pour l'entraînement que nous avons générées à partir de GTZAN. Nos résultats expérimentaux montrent que l'architecture convolutive fournit de meilleurs résultats que l'architecture Feedforward. En termes de données, il s'avère que les MFCC sont plus expressifs en termes d'informations intrinsèques aux signaux audio que les spectrogrammes Mel, et donc plus appropriés à la tâche de classification.

Annexes

Code : Le code développé au cours de ce projet est disponible sur la plateforme Github à l'adresse :
https ://github.com/orgs/Intelligent-Systems-MSc/teams/audio-grb4-1-team7.

Références

[1] Tzanetakis, George Cook, Perry. (2002). Musical Genre Classification of Audio Signals. IEEE Transactions on Speech and Audio Processing. 10. 293 - 302. 10.1109/TSA.2002.800560.

[2] McFee, Brian, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. "librosa : Audio and music signal analysis in python." In Proceedings of the 14th python in science conference, pp. 18-25. 2015.

[3] Dong, Mingwen. (2018). Convolutional Neural Network Achieves Human-level Accuracy in Music Genre Classification.

[4] Yang, Hansi Zhang, Wei-Qiang. (2019). Music Genre Classification Using Duplicated Convolutional Layers in Neural Networks. 3382-3386. 10.21437/Interspeech.2019-1298.