UNIVERSITÉ MOHAMMED V

Faculté des Sciences Rabat

Université Mohammed V
Faculté des Sciences
Rabat

Analyse des sentiments des fichiers audios

Professeur : Mahmoudi

Réalisé par :

Wiam Zellou

Safae Zellou

Année universitaire 2020-2021

Sommaire

Introduction

Grâce à tous les sens disponibles, les humains peuvent ressentir l'état émotionnel de leur partenaire de communication. Cette détection émotionnelle est naturelle pour les humains, mais c'est une tâche très difficile pour les ordinateurs; bien qu'ils puissent facilement comprendre les informations basées sur le contenu, il est difficile d'accéder à la profondeur du contenu.

Détecter les sentiments est l'une des stratégies de marketing les plus importantes dans le monde d'aujourd'hui. nous pourrions personnaliser différentes choses pour un individu spécifiquement en fonction de ses intérêts. Obtenir le sentiment des clients améliorera le résultat des produits. Normalement, l'analyse des sentiments se fait à travers les données textuelles, mais nous avons beaucoup de données audio non utilisées,et c'est ce que la reconnaissance des émotions vocales (SER) se propose de faire. Il s'agit d'un système par lequel divers fichiers vocaux audio sont classés en différentes émotions telles que le bonheur, la tristesse, la colère et la neutralité par les ordinateurs. détecter les émotions d'une personne simplement par sa voix, ce qui nous permettra de gérer de nombreuses applications liées à l'IA. Certains exemples pourraient inclure des centres d'appels pour jouer de la musique lorsque l'on est en colère lors de l'appel. Un autre pourrait être une voiture intelligente qui ralentit lorsque l'on est en colère ou craintif. Un autre pourrait être la cybersécurité. En conséquence, ce type d'application a beaucoup de potentiel dans le monde

l'objectif de notre projet est de démontrer SER en utilisant le jeu de données audio RAVDESS fourni sur Kaggle .

Environnement de travail

1. **Jupyter Notebook**

Un notebook, en programmation, permet de combiner des sections en langage naturel et des sections en langage de programmation dans un même document. Jupyter est une application web permettant de créer des notebooks. Jupyter permet de programmer en direct en langage python, langage Julia et le langage R. Le code s'y exécute. Le langage de balisage Markdown permet de commenter ce code en langage naturel.

2. **Le Langage de Programmation Python**

Python est un langage de programmation puissant et facile à apprendre. Il dispose de structures de données de haut niveau et permet une approche simple mais efficace de la programmation orientée objet. Parce que sa syntaxe est élégante, que son typage est dynamique et qu'il est interprète, Python est un langage idéal pour l'écriture de scripts et le développement rapide d'applications dans de nombreux domaines et sur la plupart des plateformes.

3. **TensorFlow**

TensorFlow est une bibliothèque de Machine Learning, il s'agit d'une boîte à outils permettant de résoudre des problèmes mathématiques extrêmement complexes avec aisance. Elle permet aux chercheurs de développer des architectures d'apprentissage expérimentales et de les transformer en logiciels.

4. **Keras**

Keras est un API de réseaux de neurones de haut niveau, écrite en Python et ineffaçable avec TensorFlow, CNTK et Theano. Elle a été développée pour permettre des expérimentations rapides.
Les avantages de Keras :

- Permet le prototypage rapide et facile (de par sa convivialité, sa modularité et son extensibilité).
- Supporte à la fois les réseaux convolutifs et les réseaux récurrents ainsi que la combinaison des deux.
- Fonctionne de façon transparente sur CPU et GPU.

5. **Librosa**

Librosa est un package python pour l'analyse musicale et audio. Il fournit les blocs de construction nécessaires pour créer des systèmes de recherche d'informations musicales.

Dataset

Pour démarrer un système de reconnaissance des émotions, trois éléments principaux doivent être pris en considération : le choix d'un ensemble de données appropriées, la sélection des caractéristiques à partir des données audio et les classificateurs pour détecter les émotions.

Description des données

Les datasets (ou jeux de données) sont couramment utilisés en machine learning. Ils regroupent un ensemble de données cohérents qui peuvent se présenter sous différents formats (textes, chiffres, images, vidéos etc…).

Pour ce travail, nous choisissons de travailler avec la base de données audiovisuelle Ryerson de la parole et des chansons émotionnelles RAVDESS (Ryerson Audio Visual Database of Emotional Speech and Song).
Il contient 7356 fichiers (taille totale : 24,8 Go) et 24 acteurs professionnels (12 femmes, 12 hommes), vocalisant deux déclarations lexicalement appariées dans un accent nord-américain neutre.
Chaque fichier audio dure 3 secondes et contient de la parole classée comme une émotion spécifique. Le format des fichiers audio est au format WAVE 16 bits, 48 kHz (.wav). Le discours comprend des expressions : calme, heureux, triste, en colère, craintives, surpris et de dégoût, et la chanson contient des émotions calmes, heureuses, tristes, en colère et craintives. Chaque expression est produite à deux niveaux d'intensité émotionnelle (normal, fort), avec une expression neutre supplémentaire.
Toutes les conditions sont disponibles dans trois formats de modalité : audio uniquement (16 bits, 48 kHz .wav), audio-vidéo (720p H.264, AAC 48 kHz, .mp4) et vidéo uniquement (pas de son). Remarque, il n'y a aucun fichier de morceau pour Actor 18 [48]. Chacun des fichiers 7356 RAVDESS a un nom de fichier unique. Le nom de fichier se compose d'un identifiant numérique en 7 parties (**par exemple, 02-01-06-01-02-01-12.mp4**).Ces identifiants définissent les caractéristiques du stimulus :

Modalité	01 = full-AV, 02 = vid´eo uniquement, 03 = audio uniquement
Canal vocal	01 = discours, 02 = chanson
Emotion	01 = neutre, 02 = calme, 03 = heureux, 04 = triste, 05 = en colère, 06 = peureux, 07 = d'égoût, 08 = surpris
Intensité émotionnelle	01 = normal, 02 = fort. Remarque : Il n'y a pas d'intensité forte pour l'émotion «neutre».
Déclaration	01 = “Kids are talking by the door”, 02 = “Dogs are sitting by the door”

Répétition	01 = 1ère répétition, 02 = 2ème répétition
Acteur	De 01 à 24. Les acteurs impairs sont des hommes, et les pairs sont des femmes

Identificateurs de nom de fichier

Exemple de nom de fichier **: 02-01-06-01-02-01-12.mp4**
Vidéo uniquement (02).
Discours (01).
Peur (06).
Intensité normale (01).
Déclaration «chiens» (02).
1ère répétition (01).
12ème acteur (12).
Femme, car le numéro d'identification de l'acteur est pair.

Implémentation et mise en oeuvre

1. Import du dataset

Après avoir téléchargé la base de données "Ravdess" du site,et la dézipper, l'importer dans le drive, notre base de données est prête à être utilisée.

2. Extraction de caractéristiques (Feature Extraction)

Le défi avec les réseaux de neurones est de savoir comment gérer la taille des donn´ees de l'entr´ee d'origine, qui est souvent tr`es importante en termes de m´emoire. Par exemple, les fichiers image et audio ont souvent la taille de plusieurs Mo. Cela rend le processus de formation très coûteux, en termes d'allocation de mémoire et de nombre d'opérations de calcul nécessaires. Avant de prendre les données brutes telles qu'elles, nous extrayons des caractéristiques spécifiques afin qu'elles soient des entrées pour le réseau neuronal, de sorte que la taille sera réduite et donc le nombre d'opérations nécessaires pour entraîner le réseau au fur et à mesure pour augmenter les performances du modèle.
Dans ce travail, nous utilisons la méthode d'extraction de caractéristiques **MFCC (The Mel-Frequency Cepstral Coefficients)**, qui est une approche de pointe pour l'extraction de caractéristiques vocales.

3. Classification

a. Decision tree classifier

Les arbres de décision (AD) sont une catégorie d'arbres utilisée dans l'exploration de données et en informatique décisionnelle. Ils emploient une représentation hiérarchique de la structure des données sous forme des séquences de décisions (tests) en vue de la prédiction d'un résultat ou d'une classe. Chaque individu (ou observation), qui doit être attribué(e) à une classe, est décrit(e) par un ensemble de variables qui sont testées dans les nœuds de l'arbre. Les tests s'effectuent dans les nœuds internes et les décisions sont prise dans les nœuds feuille.

b. Random forest

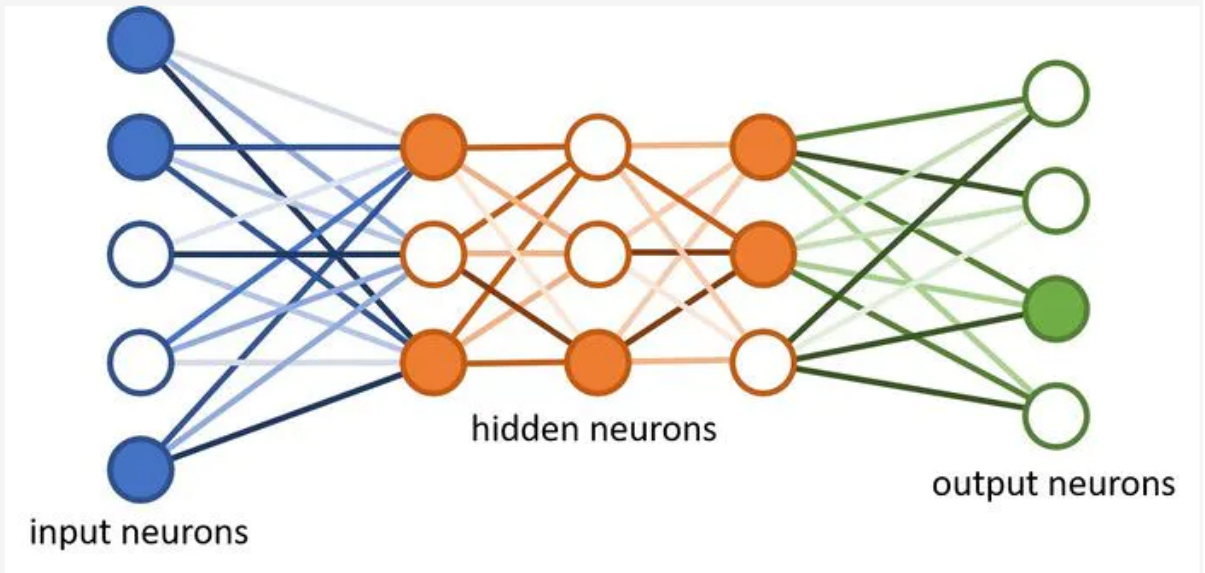
Random forest est une technique d'apprentissage automatique utilisée pour résoudre des problèmes de régression et de classification. Il utilise l'apprentissage d'ensemble, qui est une technique qui combine de nombreux classificateurs pour fournir des solutions à des problèmes complexes.
Un algorithme de forêt aléatoire (Random forest) se compose de plusieurs arbres de décision.
L'algorithme (Random forest) établit le résultat sur la base des prédictions des arbres de décision. Il prédit en prenant la moyenne ou la moyenne de la sortie de divers arbres.
L'augmentation du nombre d'arbres augmente la précision du résultat.

4. Construction du réseau de neurone

a. Qu'est ce qu'un réseau de neurones?

Le réseau de neurones est un concept. Ce n'est pas physique. Le concept de réseaux de neurones artificiels (Artificial Neural Networks ANN) a été inspiré par les neurones biologiques. Dans un réseau de neurones biologiques, plusieurs neurones travaillent ensemble, reçoivent des signaux d'entrée, traitent des informations et déclenchent un signal de sortie.Les réseaux de neurones tirent profit des expériences passées.

b. Réseau de neurones d'Intelligence artificielle (IA)



Bien que le concept sous-jacent soit le même que celui des réseaux biologiques, le réseau de neurones de l'IA est un groupe d'algorithmes mathématiques produisant une donnée de sortie (output) à partir des données d'entrée (input). Ces algorithmes peuvent être groupés pour produire les résultats souhaités. Les réseaux de neurones sont entraînés avec une multitude de données d'entrées couplées à leurs données de sortie respectives. Ils calculent ensuite la donnée de sortie, ils la comparent à la donnée de sortie réelle connue et se mettent à jour en permanence pour améliorer les résultats (si nécessaire).

c. Architecture des réseaux de neurones

Il existe un grand nombre d'architectures profondes. Nous allons détailler les réseaux de neurones convolutifs (CNNs),

Les réseaux de neurones convolutifs (CNNs) sont un type de réseau de neurones spécialisés pour le traitement de données ayant une topologie semblable à une grille leurs principe c'est le filtrage. Les exemples comprennent des données de type série temporelle sont considérées comme une grille 1D en prenant des échantillons à des intervalles de temps réguliers, les données de type image se représentent en 2D ou en 3D, il y a 2 dimensions qui correspondent à la largeur et à la hauteur de l'image et une troisième dimension qui correspond à la composante couleur.

Le nom « réseau de neurones convolutif » indique que le réseau emploie une opération mathématique appelée convolution à la place de la multiplication matricielle dans au moins une de leurs couches. La convolution est une opération linéaire spéciale.

Le rôle du ConvNet est de réduire les images sous une forme plus facile à traiter, sans perdre les fonctionnalités qui sont essentielles pour obtenir une bonne prédiction.

Il existe différentes architectures de CNN disponibles qui ont été essentielles dans la construction d'algorithmes qui alimentent et alimenteront l'IA dans son ensemble dans un avenir prévisible. Certains d'entre eux ont été énumérés ci-dessous:

LeNet, AlexNet, VGGNet, GoogLeNet, ResNet, ZFNet.

- **Couche de convolution (CONV) :**

C’est la composante clé des réseaux de neurones convolutifs, et constitue toujours au moins leur première couche. C’est la couche qui effectue le plus de calculs lourds.

3 paramètres pour dimensionner cette couche (la profondeur, le pas, la marge) :

- La profondeur(le nombre de neurones associés à un même champ de récepteur).
- Le pas(il contrôle le chevauchement des champs récepteurs).
- La marge(permet de contrôler la dimension spatiale du volume de la sortie).

- **Couche de pooling (POOL) :**

Sa fonction est de réduire progressivement la taille spatiale de la représentation pour réduire le nombre de paramètres et de calcul dans le réseau, elle contrôle également le sur-apprentissage. Entre chaque deux couches de CONV il est préférable de mettre une couche pooling. Il y a plusieurs types de couches pooling

- Max pooling (donne la valeur maximale des entrées)
- Average pooling (donne la moyenne des entrées)
- L2-norm pooling

- **Couche de correction (ReLU) :**

Pour améliorer l'efficacité du traitement on intercale entre les couches de traitement une couche qui opère une fonction mathématique sur les signaux de sortie.

- **Couche entièrement connectée(Fully Connected FC) :**

Les neurones de cette couche ont des connexions complètes à toutes les activations de la couche précédentes

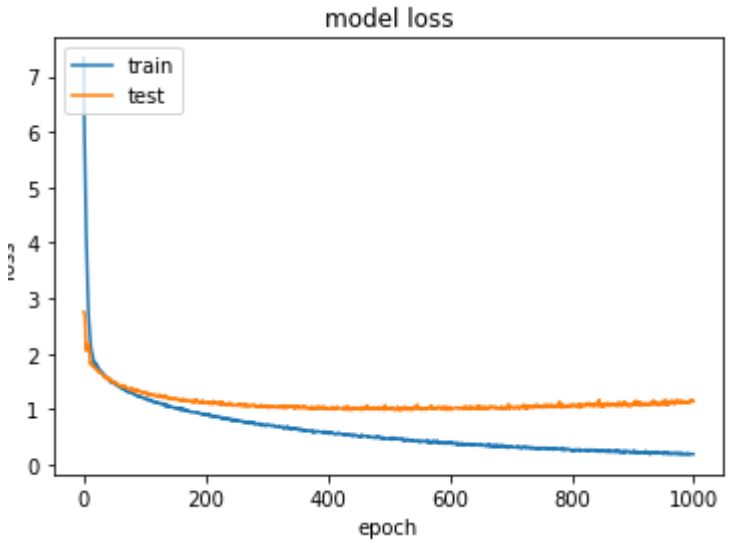
- **Couche de perte (LOSS) :**

Elle est normalement la dernière couche du réseau. Elle spécifie comment l'entraînement du réseau pénalise l'écart entre le signal prévu et réel. Les fonctions les plus utilisées softmax, la perte par entropie, la perte euclidienne

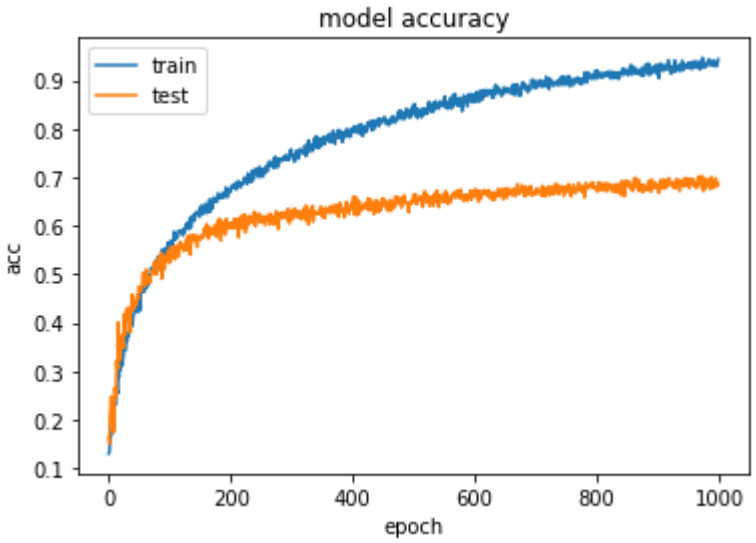
Visualisation des données pour le modèle d'apprentissage à l'aide de la bibliothèque Matplotlib :

La visualisation des performances de n'importe quel modèle d'apprentissage automatique est un moyen facile de donner un sens aux données sortant du modèle et de prendre une décision éclairée sur les modifications qui doivent être apportées aux paramètres ou hyperparamètres qui affectent le modèle d'apprentissage automatique.

- Le graphique ci-dessous représente la perte d'entraînement par rapport à la perte de validation sur le nombre d'époques.



- Le graphique ci-dessous représente la précision de l'entraînement par rapport à la précision de la validation sur le nombre d'époques.



La visualisation des données est l'un des meilleurs moyens d'humaniser les données pour les rendre faciles à comprendre et à en tirer les tendances pertinentes.

Enregistrement du modèle:

Après avoir construit le modèle vient l'étape de l'enregistrement du modèle sous le fichier " " `Ravdess_model/Emotion_Voice_Detection_Model.h5` " .

```
[ ] model_name = 'Emotion_Voice_Detection_Model.h5'
    save_dir = '/content/drive/My Drive/Ravdess_model'
    # Save model and weights
    if not os.path.isdir(save_dir):
        os.makedirs(save_dir)
    model_path = os.path.join(save_dir, model_name)
    model.save(model_path)
    print('Saved trained model at %s ' % model_path)
```

```
Saved trained model at /content/drive/My Drive/Ravdess_model/Emotion_Voice_Detection_Model.h5
```

```
loaded_model = keras.models.load_model('/content/drive/My Drive/Ravdess_model/Emotion_Voice_Detection_Model.h5')
loaded_model.summary()
```

Exploitation du modèle

```
class LivePredictions:
    """
    Main class of the application.
    """

    def __init__(self, file):
        """
        Init method is used to initialize the main parameters.
        """
        self.file = file
        self.path = MODEL_DIR_PATH + 'Emotion_Voice_Detection_Model2.h5'
        self.loaded_model = keras.models.load_model(self.path)

    def make_predictions(self):
        """
        Method to process the files and create your features.
        """
        data, sampling_rate = librosa.load(self.file)
        mfccs = np.mean(librosa.feature.mfcc(y=data, sr=sampling_rate, n_mfcc=40).T, axis=0)
        x = mfccs[np.newaxis, ..., np.newaxis]
        """x = np.expand_dims(mfccs, axis=1)
        x = np.expand_dims(x, axis=0)"""
        predictions = self.loaded_model.predict(x)
        """predicted = np.argmax(predictions)"""
        print( "Prediction is", " ", self.convert_class_to_emotion(predictions))
```

```python
    @staticmethod
    def convert_class_to_emotion(pred):
        """
        Method to convert the predictions (int) into human readable strings.
        """

        label_conversion = {'0': 'neutral',
                            '1': 'calm',
                            '2': 'happy',
                            '3': 'sad',
                            '4': 'angry',
                            '5': 'fearful',
                            '6': 'disgust',
                            '7': 'surprised'}

        for key, value in label_conversion.items():
            if int(key) == pred.any():
                label = value
        return label


if __name__ == '__main__':
    live_prediction = LivePredictions(file= EXAMPLES_PATH + '03-02-02-01-02-01-04.wav')
    live_prediction.loaded_model.summary()
    live_prediction.make_predictions()
    live_prediction = LivePredictions(file= EXAMPLES_PATH + '03-02-02-02-01-01-01.wav')
    live_prediction.loaded_model.summary()
    live_prediction.make_predictions()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d (Conv1D)              (None, 40, 128)           768
_________________________________________________________________
activation (Activation)      (None, 40, 128)           0
_________________________________________________________________
dropout (Dropout)            (None, 40, 128)           0
_________________________________________________________________
max_pooling1d (MaxPooling1D) (None, 5, 128)            0
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 5, 128)            82048
_________________________________________________________________
activation_1 (Activation)    (None, 5, 128)            0
_________________________________________________________________
dropout_1 (Dropout)          (None, 5, 128)            0
_________________________________________________________________
flatten (Flatten)            (None, 640)               0
_________________________________________________________________
dense (Dense)                (None, 8)                 5128
_________________________________________________________________
activation_2 (Activation)    (None, 8)                 0
=================================================================
Total params: 87,944
Trainable params: 87,944
Non-trainable params: 0
_________________________________________________________________
Prediction is   calm
```

Conclusion

Dans une interaction homme-homme la détection d'émotion est facile, elle est détectée soit par le visage et les expressions faciales, par les gestes du corps ou par la parole mais le défi c'est quand il s'agit d'une interaction homme-machine et que la machine soit capable de détecter l'émotion humaine. Afin d'améliorer cette interaction le terme SER apparaît qui a comme objectif la reconnaissance d'émotion en utilisant uniquement l'intonation vocale.
La reconnaissance des émotions vocales (SER) est une tâche difficile dans le domaine de l'analyse des signaux vocaux, c'est un problème de recherche qui tente de déduire l`émotion des signaux vocaux. Ces dernières années, les chercheurs ont investi intensivement dans le domaine de l'IA et de la robotique afin que ces robots puissent communiquer le plus normalement possible avec les humains.