



## **TP1 : Connexionnisme et deep learning**

TP : Réalisation d'un projet basé sur l'intelligence artificielle à l'aide de  
Tensorflow

02/12/2018


**Attributs du document**

<b>Objet du document</b>	<b>Rapport TP1 M2 INFO en Connexionnisme et deep learning</b>
<b>Référence du document</b>	<b>Réf :</b>
<b>Nom du client</b>	<b>Public</b>

**Auteur**

<b>Nom</b>	<b>Fonction</b>	<b>Contact</b>
<b>JEAN-CHARLES Loïc</b>	<b>Etudiant en M2 Informatique</b>	<b>0690 158 668</b>

**Destinataires principaux**

<b>Nom</b>	<b>Fonction</b>	<b>Société</b>	
<b>Mr PAGÉ</b>	<b>Professeur</b>	<b>Université des Antilles</b>	

**Critères de diffusion**

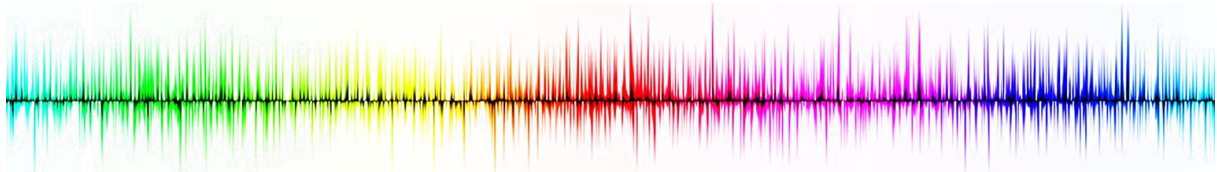
<b>Confidentiel UA</b>	<b>Confidentiel Client</b>	<b>Public</b>
<b>[ ]</b>	<b>[ ]</b>	<b>[X]</b>

# TP : Réalisation d'un projet basé sur l'intelligence artificielle à l'aide de Tensorflow

---

**Sujet :** Générer de la musique avec Tensorflow.

**Contexte :** En utilisant, un RNN (réseau de neurones récurrents), je vais générer de la musique du style du groupe 'The Chainsmokers' c'est-à-dire de la musique POP.



## Mes recherches et modifications ?

Pour ce projet, j'ai utilisé les tutoriaux suivants :

- Générer de la musique avec un RNN :  
<http://web.mit.edu/jisoomin/www/6S198/ps/ps6/notebook.html>
- J'ai modifié le modèle de réseau de neurones en adaptant avec un plus à jour. J'ai utilisé la méthode `tf.nn.rnn_cell.LSTMCell` au lieu de `rnn.BasicLSTMCell`.
- Tutoriel sur l'implémentation basic d'un rnn-rbm :  
<http://deeplearning.net/tutorial/rnnrbm.html>
- Tutoriel vidéo YouTube par Siraj RAVAL (excellents tutoriaux sur tous ce qui est en rapport avec l'intelligence artificielle et la science des données) :  
<https://www.youtube.com/watch?v=ZE7qWXX05T0&list=PL2-dafEMk2A7EEME489Dsl468AB0wQsMV&index=3>
- J'ai ajouté le logiciel « Ffmpeg.exe » qui fonctionne en ligne de commande afin de transformer les fichiers .midi en .mp3.
- J'ai modifié les paramètres du réseau de neurones, afin d'obtenir la mélodie la plus potable que possible. Les tests et leurs résultats se trouvent dans le fichier `parameters_results.txt`. La meilleure mélodie que j'ai obtenue est pour le test 2. Pour écouter, un fichier midi il faut utiliser le lecteur Windows Media Player (lecteur par défaut) ou utiliser la commande suivante pour transformer le fichier en .mp3 :

```
..\Projet\ffmpeg -i "E:\WPy3670\Projet\generated\gen_song_0.mid" -vn  
-ar 44100 -ac 2 -ab 192k  
-f mp3 "E:\WPy3670\Projet\generated\output.mp3"
```

La page suivante indique le fonctionnement interne de ce projet.

## Comment ça fonctionne ?

### Dépendances et jeu de données

Tout d'abord, il faut télécharger le paquet MIDI qui autorise la modification des fichiers .midi dans Python avec la commande suivante :

```
pip install git + https://github.com/vishnubob/python-midi@feature/python3
```

J'ai utilisé une version de python 2.7 portable (version 3.6) sur Windows avec Tensorflow sur ma clé USB pour les tests. Donc, pour faire fonctionner le projet, il faut modifier les chemins d'accès aux fichiers :

- À la ligne 112 du fichier main.py :  
`noteStateMatrixToMidi (gen_song, name = "E:\WPy3670\Projet\generated\gen_song_0")`
- À la ligne 6 du fichier create\_dataset.py :  
`songs = glob.glob (r'E:\WPy3670\Projet\util\data\*.mid *')`

### La base de données

Cette dernière est générée grâce aux musiques du dossier data. L'ensemble des musiques sont des fichiers mp3 du groupe « The Chainsmokers » en version piano (ce sont des reprises trouvées sur YouTube libre de droits) que j'ai converti en fichier midi.

L'ensemble de données est une liste de *np. arrays*, un pour chaque morceau. Chaque chanson doit avoir la forme (*longueur\_de\_la\_chanson, num\_possible\_notes*), où *longueur\_de\_la\_chanson* > = *min\_long\_length*. Les vecteurs de caractéristiques individuels des notes de la chanson sont traités en un codage *one\_hot*, ce qui signifie qu'il s'agit de vecteurs binaires où une et une seule entrée est 1.

### Le modèle du réseau neuronal récurrent (RNN)

On forme un modèle RNN sur notre base de données musicale, puis on utilise ce modèle formé pour générer une nouvelle chanson. On forme le RNN en utilisant des lots d'extraits de chansons de notre base de données grâce à Tensorflow. Puis, on définit un optimiseur pour l'apprentissage du réseau: AdamOptimizer, la fonction de perte (la perte moyenne d'entropie croisée) et la fonction de précision.

Ce modèle est basé sur une seule cellule LSTM (Réseaux de mémoire à long terme et à court terme), avec un vecteur d'état utilisé pour maintenir les dépendances temporelles entre les notes de musique consécutives. À chaque pas de temps, on alimente une séquence de notes précédentes. La sortie finale du LSTM (c.-à-d. de la dernière unité) est introduite dans une seule couche entièrement connectée pour produire une distribution de probabilité sur la note suivante. De cette façon, on modélise la distribution de probabilité :  $P(x_t | x_{t-L}, \dots, x_{t-1})$  où  $x_t$  est un encodage one-hot de la note jouée au temps t et L est la longueur d'un extrait de chanson.

### Générer la musique

Après avoir entraîné le modèle, les différentes notes que ce dernier sortira seront retranscrites dans un fichier .midi. Le fichier généré se trouve dans le dossier generated. On peut améliorer le modèle en modifiant les paramètres suivants *hidden\_size* = 256, *learning\_rate* = 0.001, *training\_steps* = 2000, *batch\_size* = 128.