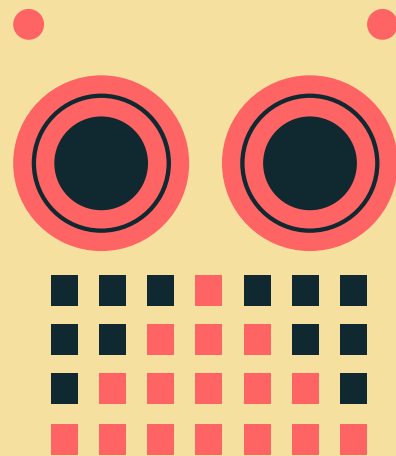
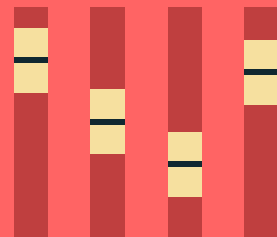
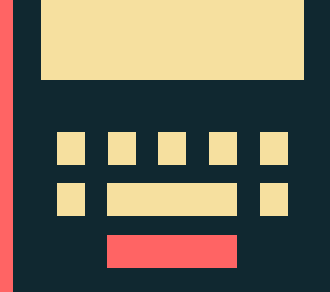
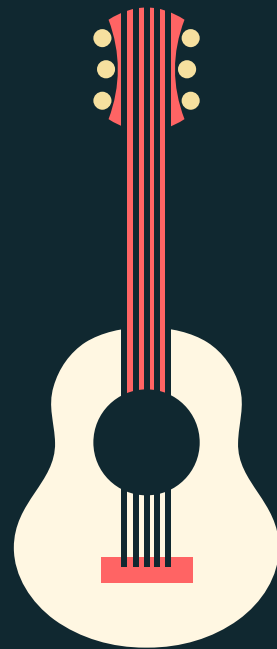




# Génération de mélodies

8INF887 – Apprentissage profond

Amandine Lapique-Favre et Hélène Barbillon



# ★ PROBLÉMATIQUE

→ Génération de mélodies par apprentissage profond

- Exploration de méthodes
- Manipulation d'un nouveau type de données (musique)





# METHODOLOGIE ★



## DONNÉES

*EsAC*  
*Essen Associative  
Code and  
Folksong Database*

- Mélodies simples
- Dynastie Han  
+1600 morceaux
- Comptines allemandes  
+1700 morceaux



## OUTILS

music21



musescore



PyTorch



TensorFlow



## MODELES

- Génération aléatoire
- RNN-LSTM
- VAE

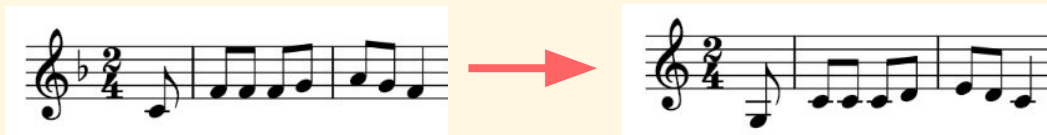
# ★ RNN-LSTM

## Pré-traitement

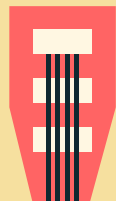
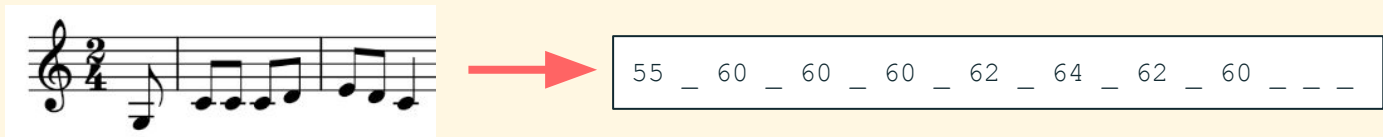
Enlever durées de notes non acceptables



Transposer en do majeur/la mineur



Encoder les musiques en texte



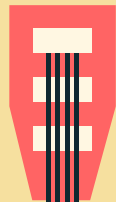
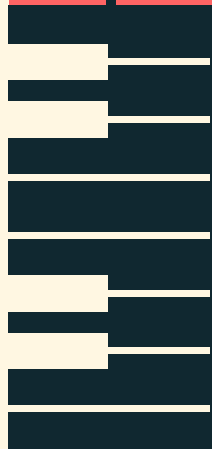
# ★ RNN-LSTM

## Entrainement



input output

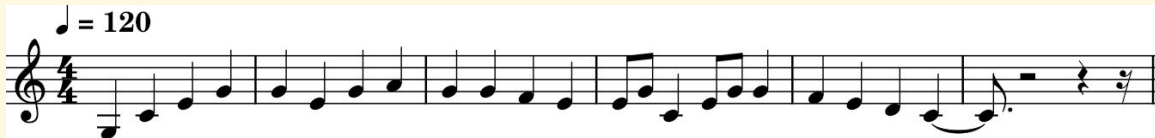
Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, None, 38)	0
lstm (LSTM)	(None, 256)	302,080
dropout (Dropout)	(None, 256)	0
dense (Dense)	(None, 38)	9,766



# ★ RNN-LSTM

## Méodies générées

Modèle deutsch

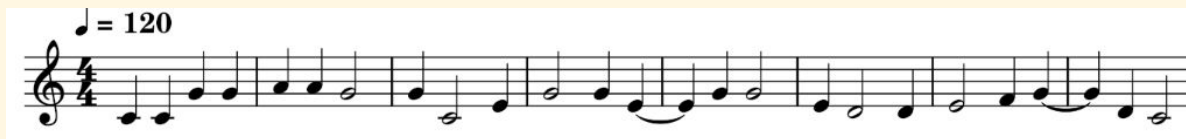


Modèle han



A partir de "ah vous dirais-je maman" :

Modèle deutsch



Modèle han



# VARIATIONAL AUTOENCODER ★



## PRÉ-TRAITEMENT



## NOTES



## RYTHME

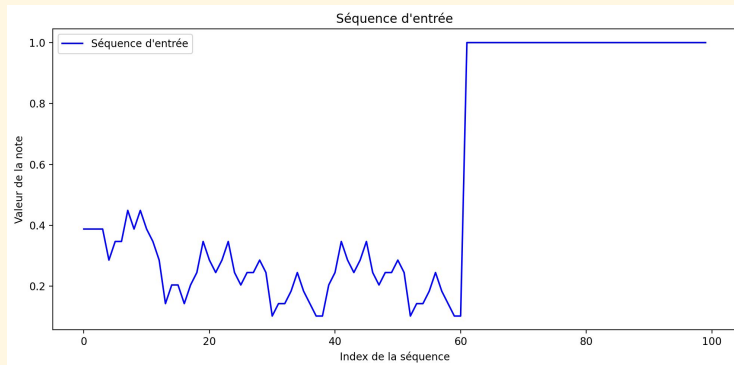


# VARIATIONAL AUTOENCODER



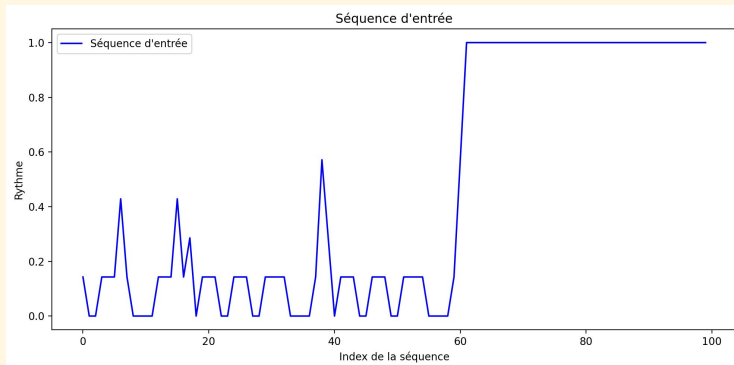
## NOTES

- ★ Transposition
- ★ Encodage
  - Hauteur  $\in [50;97]$
  - Silence = 99
- ★ Padding (ajout de silences)
- ★ Normalisation



## RYTHMES

- ★ Encodage de la durée
  - $[0.25, 0.5, 0.75, 1, 1.5, 2, 3, 4]$
  - $\rightarrow [0, 1, 2, 3, 4, 5, 6, 7]$
- ★ Padding (ajout de rondes)
- ★ Normalisation





# VARIATIONAL AUTOENCODER



## MODÈLE - VAE

### Encodeur

- 1e couche linéaire de `input_dim` → `hidden_dim`
- `LeakyReLU(0.2)`
- 2e couche linéaire de `hidden_dim` → `hidden_dim`
- `LeakyReLU(0.2)`
- Deux sorties parallèles :
  - `FC_mean` : moyenne de la distribution latente
  - `FC_var` : log-variance de la distribution latente

### Reparamétrisation

$\epsilon \sim N(0, I)$  : bruit gaussien  
 $z = \text{mean} + \text{logvar} * \epsilon$  : vecteur latent échantillonné

### Décodeur

- 1e couche linéaire de `latent_dim` → `hidden_dim`
- `LeakyReLU(0.2)`
- 2e couche linéaire de `hidden_dim` → `hidden_dim`
- `LeakyReLU(0.2)`
- `FC_output` : couche linéaire de `hidden_dim` → `output_dim`
- `Sigmoid` : activation finale (entre 0 et 1)



## NOTES

```
hidden_dim = 500
latent_dim = 250
epochs = 200
batch_size = 32
```

### Optimiseur Adam :

- Taux d'apprentissage : `lr = 1e-5`
- Régularisation L2 : `weight_decay = 1e-6`



## RYTHMES

```
hidden_dim = 500
latent_dim = 300
epochs = 8
batch_size = 64
```

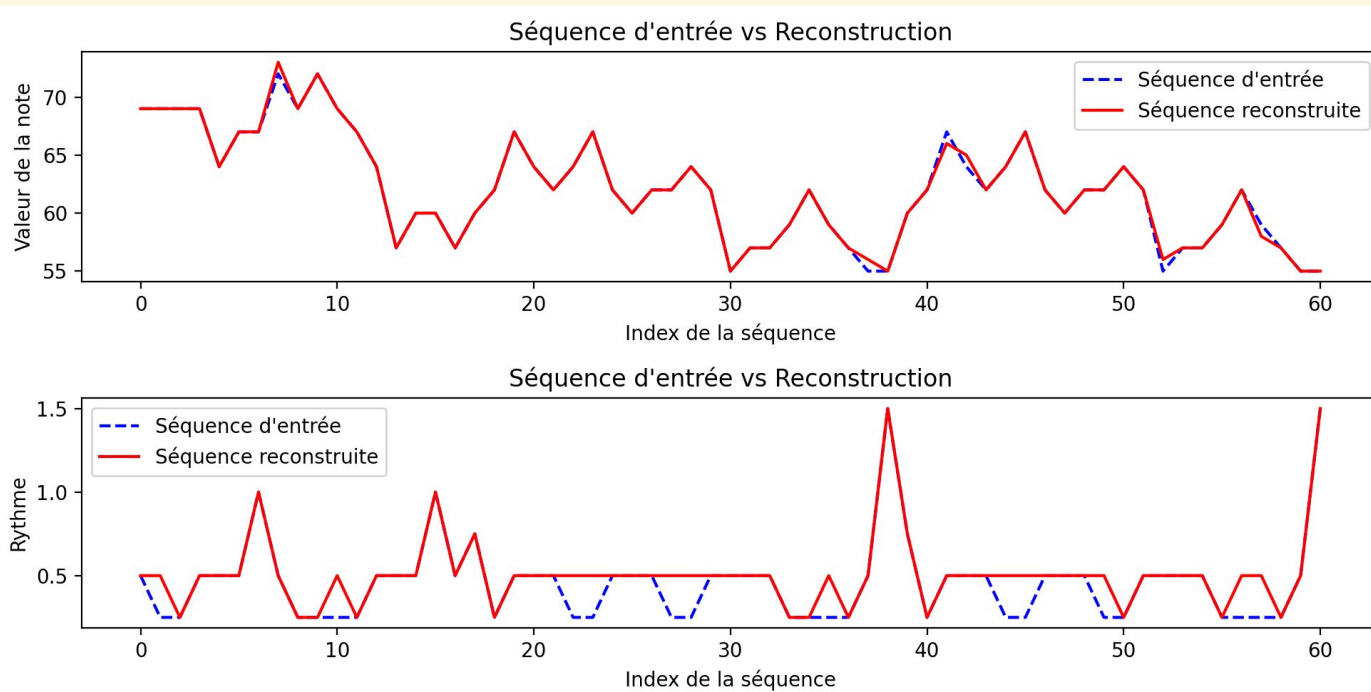
### Optimiseur Adam :

- Taux d'apprentissage : `lr = 1e-3`

# VARIATIONAL AUTOENCODER



## RÉSULTATS



# VARIATIONAL AUTOENCODER



## RÉSULTATS

ORIGINALE



RECONSTRUCTION





# BILAN



## DIFFICULTÉS

- Données de bonnes qualités
- Pré-traitement
- Données inhabituelles

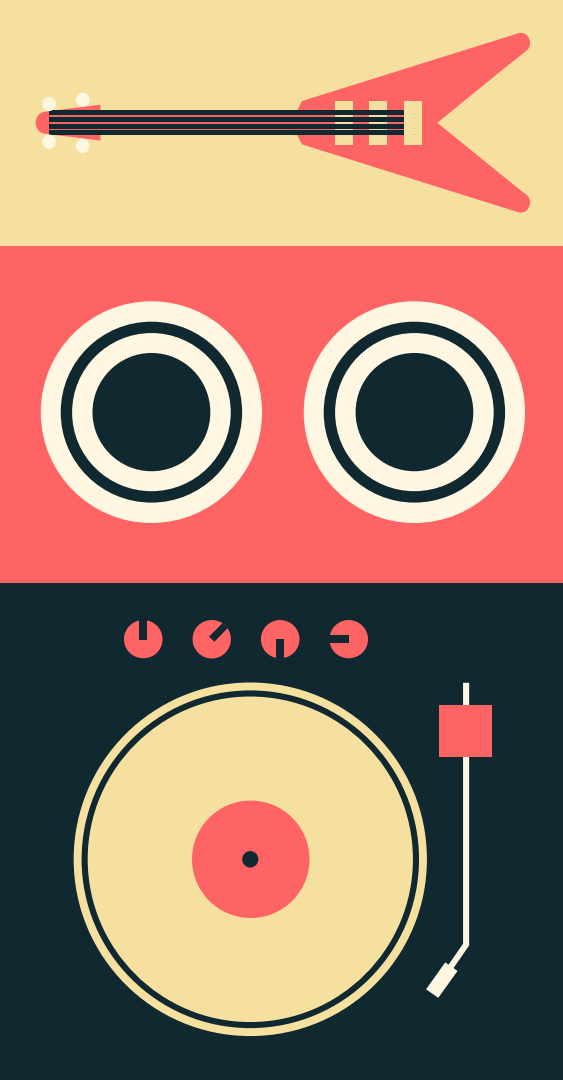


## PERSPECTIVES

- GAN
- Transformeurs



## PERSONNEL



# Merci

**CREDITS:** This presentation template was created by [Slidesgo](#), and includes icons by [Flaticon](#), and infographics & images by [Freepik](#)