

TP : Réseaux récurrents pour génération musicale

Objectifs

- Prétraiter des partitions musicales pour les rendre compatibles avec un modèle RNN.
- Implémenter un modèle LSTM capable de prédire la prochaine note.
- Entraîner le modèle avec gestion des batches, padding et logging.
- Générer de nouvelles séquences musicales à partir d'une séquence initiale.
- (Bonus) Appliquer une augmentation de données par transposition des partitions.

1. Chargement et exploration des données

- Les datasets **train.json** et **validation.json** ont été chargés via datasets.
- **Nombre de chansons** : 214 122 (train), 2 162 (validation).
- Chaque partition est au format ABC, avec des en-têtes (X, L, M, K) et le corps de la mélodie.

2. Prétraitement des données

- Extraction des **caractères uniques** : 95.
- Création de mappings **caractère : index** (char2idx et idx2char).
- **Vectorisation** et **padding** des séquences pour uniformiser leur longueur maximale (2968 caractères).
- Les séquences sont préparées pour l'entraînement :

x = séquence d'entrée, y = séquence cible décalée d'un caractère.

3. Création du dataset PyTorch

- Classe MusicDataset implémentée avec gestion du padding et renvoi (x, y).
- DataLoaders créés pour l'entraînement et la validation (batch_size = 8).
- Vérification : x.shape = [8, 299], y.shape = [8, 299].

4. Architecture du modèle

- **Embedding** : projection des indices en vecteurs continus.
- **LSTM** : capture des dépendances temporelles.
- **Couche dense** : prédiction du caractère suivant.
- Hyperparamètres pour test :

`embedding_dim = 256, hidden_size = 512, learning_rate = 5e-3.`

5. Entraînement

- Fonction `train_one_epoch` implémentée avec **CrossEntropyLoss** et **Adam optimizer**.
- Logging des pertes sur TensorBoard et sauvegarde du modèle.
- Résultat après 1 epoch :
 - Train Loss = 1.2787
 - Val Loss = 1.2816

6. Génération de musique

- Fonction **generate_music** utilise le modèle pour générer une séquence de 200 caractères à partir d'une séquence initiale.
- Sauvegarde du résultat en fichier .abc et conversion en .mid pour lecture via music21.

7. Bonus : augmentation des données

- Transposition des notes (+1, -1) pour augmenter la diversité.
- Nombre de partitions après augmentation : 642 366 (vs 214 122 original).

Conclusion

Le TP a permis de mettre en œuvre un pipeline complet pour la génération musicale : prétraitement, vectorisation, dataset PyTorch, entraînement d'un LSTM, génération et

augmentation de données. Le modèle est opérationnel et peut générer des mélodies cohérentes à partir d'une séquence initiale.