Генерисање музике применом неуронских мрежа

Петар Никић, Вукан Антић

15. јануар 2022.

Сажетак

У овом раду демонстрирамо како се неуронске мреже могу користити за генерисање музике. Фокус рада је на генерисању мелодија за клавир. Приказане су две архитектуре: једна заснована на рекурентним неуронским мрежама и друга заснована на конволутивним неуронским мрежама. Крајњи модели су у стању да са делимичним успехом генеришу интересантне мелодије.

1 Увод

Прављење музике је задатак који је пре пар деценија био незамислив за рачунаре, и резервисан за човека због креативности коју човек поседује. Међутим, у скорије време, показано је да и машине могу да буду добри композитори ([3]).

Напредак на овом пољу омогућила је појава неуронских мрежа, и то посебних неуронских мрежа које раде са секвенцама података: рекурентним и конволутивним неуронским мрежама.

2 Проблем

У овом раду бавимо се генерисањем музике из *Lo-fi* жанра. Lo-fi (low fidelity) жанр одабрали смо због своје једноставности. Музика овог жанра се састоји из следећих елемената:

- Музичког узорка (sample)
- Бубњева
- Позадинских звукова

Музички узорак представља (обично кратки) низ тонова и акорда који се понављају током песме.

Узорак песми даје мелодију, бубњеви ритам, а позадински звукови се додају као стилски елемент жанра.

Фокус нашег рада је био генерисање управо музичког узорка употребом неуронских мрежа.

2.1 Формулација проблема

Када размишљамо о музици, једна од карактеристика коју лако можемо уочити је да постоји одређена временска структура музике. Тонови нису независни, већ се заједно комбинују у дуже структуре.

Проблем постављамо на начин који је карактеристичан за проблеме који се баве секвенцама података ([2], [3]). Модел учи функцију f која представља условну расподелу вероватноће за наредни тон ако су познати претходни тонови у секвенци.

Прецизније, нека је T дужина секвенце коју посматрамо, x_t представља улаз модела у тренутку t, и $x = (x_1, x_2, \ldots, x_T)$ представља секвенцу улазних вектора. Модел учи условну расподелу

$$P\{x_{T+1} \mid x_1, x_2, \ldots, x_T\}$$

3 Имплементација

Имплементација модела је рађена у програмском језику *Python*, у *Jupyter* свескама, на сервису *Google Colaboratory*. Од библиотека смо користили *Music21* за обраду музичких података и *Keras* за дефинисање модела и рад са моделом.

Коришћен скуп података за учење је Lo-Fi Hip Hop MIDIs, који је саставио Zachary Katsnelson.

3.1 Обрада улазних података

Подаци су организовани као колекција песама и узорака у MIDI формату.

`music21` библиотеку смо користили да из узорака издвојимо низове тонова, акорда и пауза из изворних MIDI фајлова.

Након тога, музичке елементе из свих узорака смо спојили у један низ, и пресликали (бијективно) у целе бројеве.

Из великог низа музичких елемената свих узорака смо издвојили секвенце дужине која одговара хиперпараметру `sequence_length`, заједно са наредним елементом који се користи за проверу тачности ноте која је предвиђена на крају. Овде треба напоменути да ће неке секвенце садржати елементе из различитих песама, али смо проценили да је прихватљиво допустити ове случајеве, јер се ради о једном жанру музике, и претпоставка је да су узорци довољно слични.

У случају рекурентне архитектуре, цели бројеви улазних секвенци се кодирају у асиметричне бинарне атрибуте (*one-hot* кодирање), док у конволутивној архитектури цели бројеви првобитно пролазе кроз `Embedding` слој, који сваку вредност пресликава у вектор, који се прослеђује даљим слојевима.

Наредни елемент се у оба случаја кодира one-hot кодирањем.

3.2 Модел

Истражили смо две архитектуре модела, једну рекурентну и једну конволутивну. Рекурентна архитектура је заснована на архитектури описаној у раду [2], док је конволутивна инспирисана *WaveNet* архитектуром, коју је представио *DeepMind* у [3].

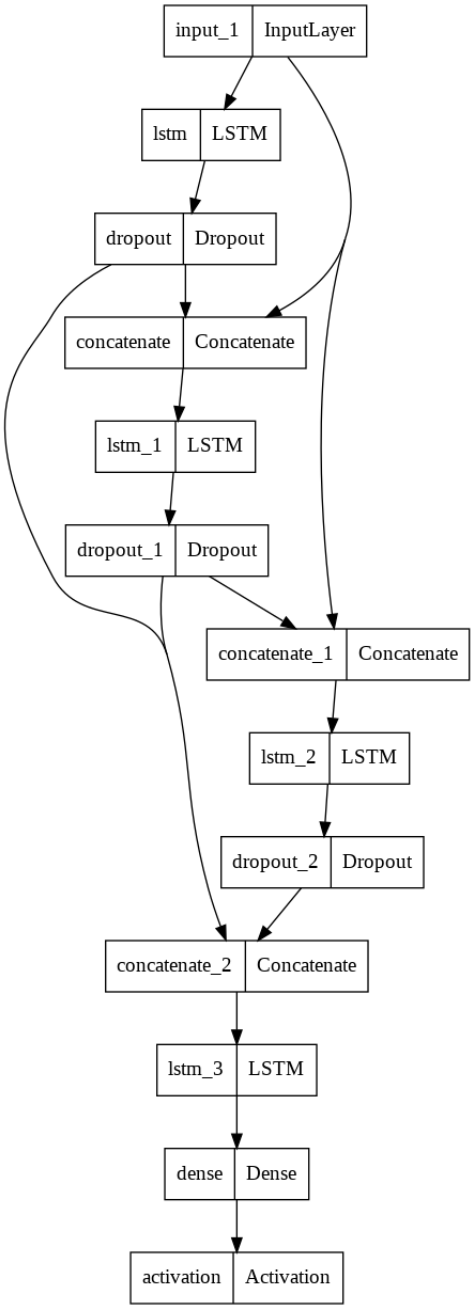
Модели су дефинисани функционалним апликативним програмским интерфејсом који нуди библиотека *Keras*.

3.2.1 Рекурентна архитектура

Рекурентна аритектура је организована у 4 скривена, рекурентна слоја. За рекурентне елементе се могу користити прости рекурентни елемент (RNN), Long Short-Term Memory (LSTM) или Gated Recurrent Unit (GRU).

У мрежу су додати скокови: улаз се прослеђује свим сем последњег скривеног слоја, и сви скривени слојеви прослеђују улаз последњем. На овај начин се смањује број корака од почетка до краја мреже, чиме се умањује проблем *нестајања градијента*, и обезбеђује боље учење дубоких мрежа.

На крају се налази обичан, густи слој са *softmax* активационом функцијом, који описује расподелу вероватноћа за наредни елемент секвенце (музички елемент).



Слика 1: Изглед рекурентне архитектуре

3.2.2 Конволутивна архитектура

Конволутивну архитектуру карактерише наизменични низ једнодимензионих конволуција са *ReLU* активационом функцијом и једнодимензионих max агрегација.

Конволуције у моделу су узрочне конволуције са дилатацијом (causal dilated convolutions).

Узрочне конволуције резултат формирају искључиво на основу претходних вредности секвенце, што одговара полазном проблему у којем су зависности од прошлих ка будућим вредностима.

Дилатација омогућава да се на дубљим слојевима повећа ширина поља вредности на основу којих се врши предикција, односно број улазних вредности које учествују у предикцији. Шире поље нам је битно за учење дугорочне структуре података, које је природно присутно у рекурентним мрежама, али није у конволутивним. Други начини на који се може обезбедити шире поље су коришћење више слојева или увећавање филтера. Оба наведена случаја значајно повећавају количину потребног рачуна.

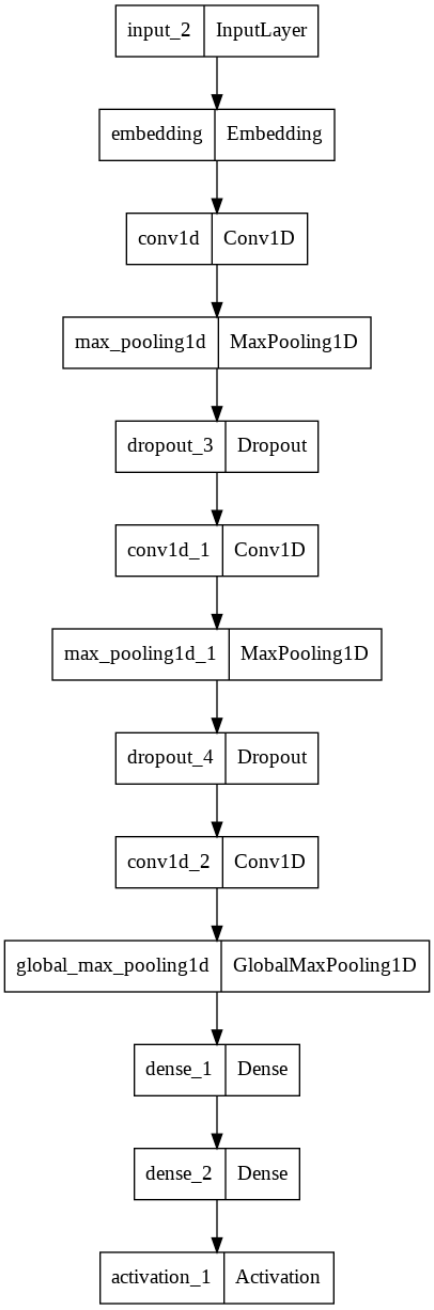
3.3 Генерисање музике

Музика се генерише давањем иницијалног узорка који представља замишљене ноте пре почетка генерисања.

На основу узорка се генерише следећа нота, узорковањем из расподеле коју дефинише излаз мреже. Узорак се помера ка прошлости, прва нота се избацује, а на место последње ноте се смешта генерисана нота, након чега се нови узорак може користи за даље генерисање наредних нота.

4 Резултати

У току развоја цео скуп је коришћен за тренирање. Скупови за валидацију и тестирање нису коришћени. Први разлог за доношење такве одлуке је мањак података. Додатном поделом скупа података би се додатно смањио доступан број података за тренирање. Други разлог је због природе проблема који решавамо. Музика је субјективна ствар, и сматрали смо да постоји шанса да модел који нема велику тачност прави бољу мелодију од модела који има велику тачност. Начин обраде података није потпуно природан, и доста поједностављује изворне податке, што даје додатне назнаке за претходно наведену тврдњу.



Слика 2: Изглед конволутивне архитектуре

Прихватајући субјективне карактеристике музике, процену смо вршили генерисањем и слушањем узорака. Неки од резултата које смо приметили следе.

- Прости RNN елементи нису могли да адекватно моделују структуру музике, и генерисани узорци су често били једна или две ноте које се врте у петљи.
- LSTM и GRU су показали сличне резултате, што је било и очекивано ([1]). Субјективно, више су нам се допали резултати генерисани GRU моделом.
- Конволутивни модел боље моделује локалну структуру музике, због чега обично звучи природније од рекурентних модела.
- LSTM, GRU и CNN модели генеришу различите резултате за различите улазе. То је добар индикатор да могу да науче одређени стил, али да се не фиксирају за једну песму.

Међутим, главни утисак је да генерисана музика није на нивоу способности човека. Неопходна је обрада генерисаних резултата, попут одабира интересантних делова и исецање тих делова од остатка песме. Алтернативна употреба модела може да буде као инспирација за професионалне уметнике, који могу да очувају идеју иза генерисане музике, уз преправке грешака које начини модел.

Литература

[1] Junyoung Chung и др. "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling". У: *arXiv:1412.3555 [cs]* (11. дец. 2014.). arXiv: 1412.3555. URL: http://arxiv.org/abs/1412.3555.

[2] Alex Graves. "Generating Sequences With Recurrent Neural Networks". У: (4. авг. 2013.). URL: https://arxiv.org/abs/1308.0850v5.

[3] Aaron van der Oord и др. "WaveNet: A Generative Model for Raw Audio". У: *arXiv:1609.03499 [cs]* (19. септ. 2016.). arXiv: 1609.03499. URL: http://arxiv.org/abs/1609.03499.