



به نام خدا



دانشگاه تهران

دانشکده برق و کامپیوتر

شبکه‌های عصبی مصنوعی و یادگیری عمیق

مینی پروژه دوم

دی ماه 1400

## سوال اول

یکی از کاربردهای جالب شبکه‌های عصبی بازگشتی، استفاده از آنها برای تولید موزیک است. با توجه به اینکه در موسیقی هر نوت به نوت قبلی خود وابسته است، میتوان از دیتای موجود در شبکه‌های عصبی بازگشتی برای تولید موزیک استفاده کرد. در این پروژه به پیاده سازی این تسک یعنی music generation با استفاده از شبکه های عصبی بازگشتی می پردازیم.

### قسمت اول: load کردن دیتا

برای این سوال ما از دیتای Classical Music MIDI استفاده خواهیم کرد. فایل دیتایی که در اختیار شما قرار داده می شود دارای آهنگهای آهنگسازان مختلف می باشد. ما فقط از آهنگ های آهنگساز معروف لهستانی [Frédéric Chopin](#) استفاده خواهیم کرد.

برای load کردن دیتا آهنگ های این هنرمند را از پوشه هم نام خودش با استفاده از کتابخانه music21 بخوانید. سپس باید `cord` ها و `note` های موجود در هر آهنگ را به دست بیاوریم. برای اینکار لازم است که ابتدا یک تعریفی از `note` و `cord` در موسیقی به دست بیاورید. در نهایت همه `note` ها موجود در هر آهنگ را در یک لیست ذخیره کنید. برای `extract` کردن `note` ها میتوانید از تابع `extract_notes` داده شده استفاده کنید.

### قسمت دوم: پیش پردازش دیتا

در این مرحله پس از اینکه لیست `corpus` را به دست آوردید، تعداد تکرار هر `note` را شمرده و `note` های کم تکرار یا به اصطلاح `rare` را از `corpus` حذف کنید. لازم به ذکر است که تعداد حداقل تکرار برای حذف کردن یک `note` اختیاری است و میتوانید این پارامتر را خودتان تعیین کنید. جهت درست کردن دیتای آموزش و تست شبکه میتوانید از `corpus` حاصل از مرحله قبل `note` ها را به صورت `sequence` های 40 تایی از `corpus` خوانده و `note` 41 امی بعد از `sequence` را به عنوان لیبل و دیتای تست در نظر بگیرید. برای مثال ایندکس 0 تا 39 به عنوان یک `sequence` و ایندکس 40 به عنوان لیبل این `sequence` در نظر گرفته می شود. ایندکس 1 تا 40 به عنوان `sequence` بعدی و ایندکس 41 به عنوان لیبل این `sequence` در نظر گرفته می شود و الی آخر ...

پس از ایجاد `sequence` ها با استفاده از یک دیکشنری `sequence` ها و لیبل های مورد نظر را `encode` کنید و در این مرحله دیتای آموزشی و لیبل های شما در دسترس است. توجه کنید که برای آموزش شبکه لیبل ها را باید به صورت `one hot encoding` تبدیل کنید.

پس از این کار دیتای `train` و `test` را جدا کرده و در مرحله بعد شبکه بازگشتی خود را پیاده سازی میکنیم.

### قسمت سوم: پیاده سازی شبکه عصبی بازگشتی

میتوانید از مدل زیر یا هر مدل دلخواه خود استفاده کنید. از تابع `categorical_crossentropy` برای محاسبه `loss` استفاده کنید. هم چنین تابع فعال سازی لایه `fully connected` را `softmax` در نظر بگیرید. ساختار انتخابی خود را در گزارش شرح دهید.

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 40, 512)	1052672
lstm_1 (LSTM)	(None, 256)	787456
dense (Dense)	(None, 256)	65792
dense_1 (Dense)	(None, 266)	68362
Total params: 1,974,282		
Trainable params: 1,974,282		
Non-trainable params: 0		

## قسمت چهارم: ارزیابی شبکه

مقدار loss را برای فرایند آموزش چاپ کنید.

یک آهنگ با استفاده از تابع داده شده تولید کرده و آن را به همراه فایل گزارش خود آپلود کنید. آهنگ تولید شده با فرمت midi را میتوانید با استفاده از ابزارهای آنلاین موجود به فرمت wav تبدیل کنید.

همین مراحل را برای موسیقی‌هایی که آهنگ سازی آن [Mozart](#) است انجام دهید.

تاثیر dropout بر روی عملکرد شبکه را آزمایش کنید. پس از هر لایه lstm و لایه‌های fully connected یک لایه dropout اضافه کنید.

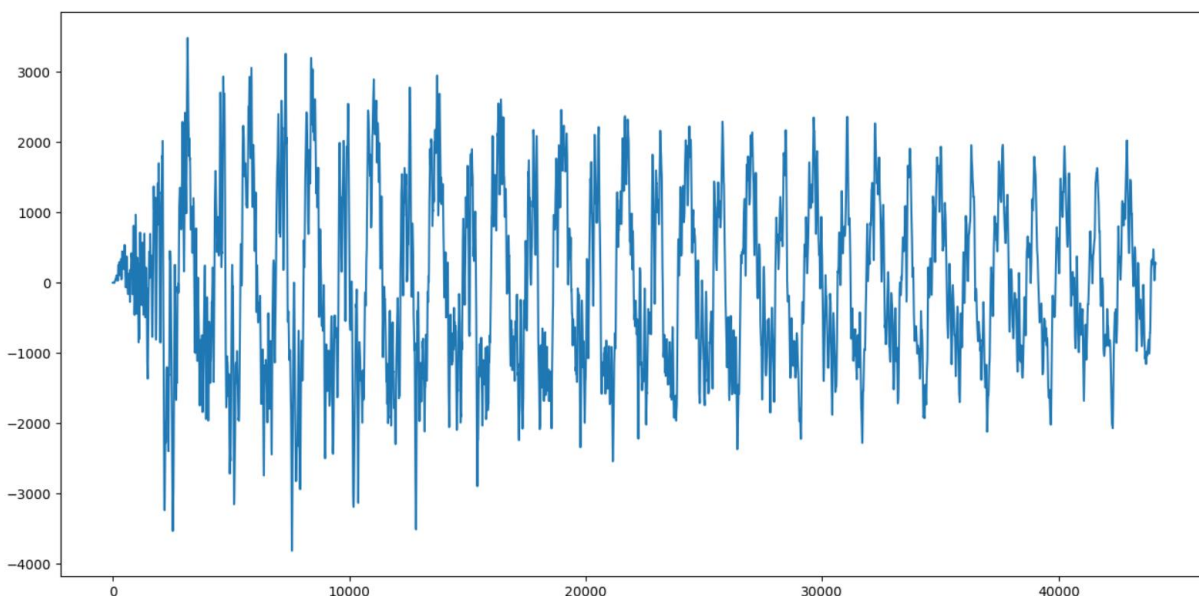
## سوال دوم - تشخیص نت موسیقی

یکی از کاربردهای شبکه‌های عصبی بازگشتی در پردازش صوت است. در این پروژه قصد داریم شبکه‌ای عصبی طراحی کنیم تا با شنیدن یک سمپل صوتی کوتاه، نت موسیقی بکار رفته در آن سمپل را شناسایی کند. هر نت موسیقی فرکانس مشخصی دارد که در سازهای مختلف موسیقی یکسان بوده و تغییر نمی‌کند. در [این لینک](#) می‌توانید فرکانس ۱۰۸ نت مختلف را مشاهده نمایید.

مجموعه داده‌ای که در پوشه‌ی notes/midi قرار گرفته شامل فایل‌های صوتی‌ای به طول ۱ ثانیه است. نام هر فایل به فرمت "instrument\_note.mid" بوده که instrument شماره ساز موسیقی و note شماره نت موسیقی از آن ساز است. در این مجموعه داده ۱۰۸ نت از ۱۲۸ ساز مختلف قرار دارد. این نمونه‌ها توسط فایل generate\_midi.py تولید شده‌اند.

### قسمت اول: مجموعه داده

فایل‌های فرمت mid را به فرمت wav تبدیل کنید تا با استفاده از کتابخانه‌ی `scipy.io.wavfile` (یا سایر کتابخانه‌های صوتی) قابل پردازش باشد. سپس سیگنال موجود در فایل‌های `1_0.wav` و `1_12.wav` را پلات کرده و شباهت و تفاوت آن‌ها را بیان کنید.



تصویر ۱: سیگنال موجود در فایل `0_0.wav`

محور افقی و عمودی در تصویر بالا بیانگر چیست؟

### قسمت دوم: تولید مجموعه‌های `test` و `validation`, `training`

مجموعه داده را به ۳ قسمت تقسیم کنید. نحوه‌ی تقسیم‌بندی (نمونه‌هایی که در هر دسته قرار می‌گیرد، اندازه دسته، سازهای داخل هر دسته و ...) و علت این نوع تقسیم‌بندی را توضیح دهید.

### قسمت سوم: پیاده‌سازی شبکه عصبی

در این قسمت باید شبکه‌ای طراحی کنید که یک قطعه صوتی به طول ۲۵۰ میلی ثانیه را به عنوان ورودی گرفته و در خروجی نت موسیقی مربوط به آن قطعه را از میان ۱۰۸ نت موجود مشخص کند.

الف) شبکه را با سلول‌های `RNN`، `LSTM` و `GRU` طراحی کرده و نتیجه را بر اساس سرعت و دقت تحلیل کنید.

ب) تاثیر dropout را بر روی هر سه شبکه در قسمت قبل بررسی کنید.

ج) بهترین شبکه‌ای که در قسمت قبل بدست آمد را با فریم‌های زمانی ۳۵، ۷۰، ۱۵۰ و ۳۰۰ آزمایش کرده و نتیجه را مقایسه و تحلیل کنید.

د) چند مورد از الگوریتم‌های استخراج ویژگی (مثل تبدیل فوریه و تبدیل wavelet) که برای حل این مسئله مناسب هستند را انتخاب کرده و توضیح دهید. سپس از آن‌ها برای افزایش دقت بهترین شبکه‌ی قسمت قبل استفاده کنید.

ه) امتیازی: چند مورد از data augmentation‌هایی که برای حل این مسئله مناسب است را توضیح دهید. سپس سعی کنید دقت بهترین شبکه‌ی قسمت قبل را با اضافه کردن حداقل یکی از این چند مورد افزایش دهید.

و) امتیازی: با استفاده از ابزار spectrogram فریم صوتی را به تصویر تبدیل کرده و با ترکیب CNN و RNN پیش‌بینی را انجام دهید و نتایج را با بهترین شبکه‌ی بدست آمده در قسمت‌های قبل مقایسه کنید.

## سوال سوم

از دیگر کاربرد های شبکه های عصبی تولید متن در کاربرد های جذاب در این زمینه تقلید از اشعار شاعران و تولید شعر جدید است. در این سوال مجموعه داده های (LYRICS\_DATASET) اشعار تعدادی از موسیقی های برتر دنیا در اختیار شما قرار گرفته است.

**الف)** مدلی طراحی کنید که بتواند یک بیت شعر تولید کند و نمودار خطا و دقت آن را رسم کنید همچنین ذکر کنید به چه دلیل پیش پردازش انتخاب شده میتواند باعث بهبود مدل شود؟

**ب)** با استفاده از ۳ تابع زیان مختلف و ۳ معیار متفاوت عملکرد مدل را مقایسه کنید.

**پ)** بیت تولید شده توسط مدل خود را مشاهده کنید و توضیح دهید چه تفاوتی در اولین کلمات تولید شده با کلمات آخر بیت شما وجود دارد و دلیل آن را توضیح دهید.

**ت)** چه روش هایی برای کاهش هزینه محاسباتی مدل پیشنهاد میکنید.

**ث)** توضیح دهید چگونه حافظه سلول های عصبی استفاده شده در مدل شما در عملکرد مدل موثر است.

## نکات:

- مهلت تحویل این تمرین/ مینی پروژه 19 دی است .
  - گزارش را در قالب تهیه شده که روی صفحه درس در Elearn بارگذاری شده، بنویسید.
  - گزارش شما در فرآیند تصحیح از اهمیت ویژهای برخوردار است. لطفا تمامی نکات و فرضیهایی که برای پیاده‌سازیها و محاسبات خود در نظر میگیرید را در گزارش ذکر کنید.
  - در گزارش خود برای تصاویر زیرنویس و برای جداول هم بالانویس اضافه کنید .
  - الزامی به ارائه توضیح جزئیات کد در گزارش نیست. اما باید نتایج بدست آمده را گزارش و تحلیل کنید.
  - دستیاران آموزشی ملزم به اجرا کردن کدهای شما نیستند. بنابراین هرگونه نتیجه و یا تحلیلی که در شرح سوال از شما خواسته شده است را به طور واضح و کامل در گزارش بیاورید. در صورت عدم رعایت این مورد، بدیهی است که از نمره تمرین کسر میشود.
  - در صورت مشاهده تقلب امتیاز تمامی افراد شرکت کننده در آن، 100 - لحاظ میشود.
  - برای انجام تمرین ها و مینی پروژه ها، تنها زبان برنامه نویسی مجاز Python است .
  - استفاده از کدهای آماده برای تمرینها به هیچ وجه مجاز نیست. اما برای مینی پروژه‌ها فقط برای قسمتهایی از کد و به عنوان راهنمایی برای پیاده‌سازی، میتوانید از کدهای آماده استفاده کنید.
  - نحوه محاسبه تاخیر به این شکل است: مهلت ارسال بدون جریمه تا تاریخ اعلام شده است؛ و پس از آن به مدت هفت روز تا \_ بارگذاری با کسر جریمه تاخیر ممکن است و در نهایت، پس از بازه تاخیر نمره تکلیف صفر خواهد شد .
  - لطفا گزارش، فایل کدها و سایر ضمایم مورد نیاز را با فرمت زیر در سامانه مدیریت دروس بارگذاری نمایید.
- PROJECT2\_[Lastname]\_[StudentNumber].zip
- در صورت وجود هرگونه ابهام یا مشکل میتوانید از طریق رایانامه‌های زیر با دستیاران آموزشی مربوطه آقایان امیر حسین مصباح (سوال اول) ، محمد امین حق پناه (سوال دوم) و سینا سبزواری (سوال سوم) در تماس باشید:

[mesbahamirhossein@gmail.com](mailto:mesbahamirhossein@gmail.com)

[mdan.hagh@gmail.com](mailto:mdan.hagh@gmail.com)

[sinasabzevar@ut.ac.ir](mailto:sinasabzevar@ut.ac.ir)