LSTM ile Müzik Oluşturma

Proje Raporu

Ders: Makine Öğrenmesine Giriş

Ad-Soyad: Aydın Bozkır

No: 2111502047

İçindekiler

Amaç: Bu projede Keras kütüphanesindeki LSTM(Long Short-Term Memory) ağını kullanarak bir müzik oluşturma ağı tasarlandı. Amaç, mevcut müzik verilerinden öğrenerek, benzer tarzda yeni müzik notaları üretmektir. Bu, müzik besteleme sürecini otomatikleştirmek için kullanılabilecek projedir.

Music21 Kütüphanesi: Music21, bilgisayar destekli müzikoloji için kullanılan Python araç takımıdır. Müzik örnekleri üretmemizi ve müzik çalışmamızı sağlar. Araç takımı, MIDI dosyalarının müzik notasyonunu edinmek için basit bir arayüz sağlar. Ayrıca kendi MIDI dosyalarımızı kolayca oluşturabilmemizi için Nota ve Akor nesneleri oluşturmamızı sağlar. Bu projede, veri setimizin içeriğini çıkarmak ve sinir ağının çıktısını alıp müzik notasyonuna çevirmek için Music21' i kullanacağız.

Özet: Projede MIDI formatındaki müzik dosyaları ve müziksel notalar bir veri kümesine dönüştürülüp 200 epochluk bir eğitim yapıldı. Loss fonksiyonumuz yaklaşık 5 değerlerinde başlayıp yaklaşık 0.2 değerlerine indi. Daha fazla eğitim ile daha detaylı ve farklı müzikler elde edilebilir.

Yöntem ve Teknikler

Veri Seti: Veri seti Final Fantasy film müziklerinden oluşmuş bir veri setidir. Tek bir enstrümandan oluşmuş olması yeterli olacaktır. Tek enstrümandan oluşan herhangi bir veri seti de kullanılabilir.

Bir MIDI dosyası Music21 kütüphanesi ile aşağıdaki gibi okunuyor;

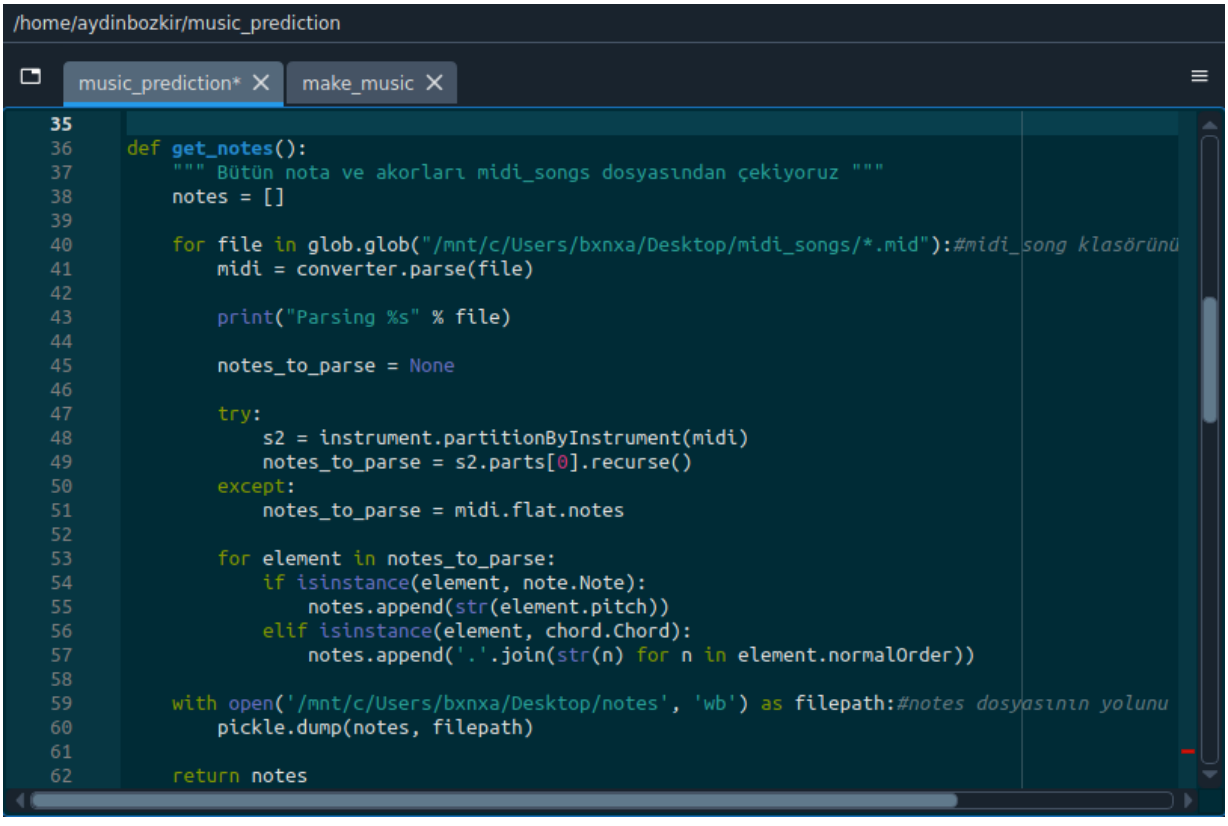
```
...
<music21.note.Note                          F>
<music21.chord.Chord                A2      E3>
<music21.chord.Chord                A2      E3>
<music21.note.Note                          E>
<music21.chord.Chord                B-2     F3>
<music21.note.Note                          F>
<music21.note.Note                          G>
<music21.note.Note                          D>
<music21.chord.Chord                B-2     F3>
<music21.note.Note                          F>
<music21.chord.Chord                B-2     F3>
<music21.note.Note                          E>
<music21.chord.Chord                B-2     F3>
...
```

Veriler iki nesne türüne ayrılır: Notalar ve Akorlar. Nota nesneleri notanın perdesi, oktavı ve ofseti hakkında bilgi içerir. Perde, sesin

frekansını veya yüksekliğini belirtir. Oktav piyanoda hangi perde setini kullandığınızı belirtir. Ofset notanın parçada nerde bulunduğunu belirtir. Ve akor nesneleri, esasen aynı anda çalışan bir dizi nota yığınıdır.

Eğitim: Müziği doğru bir şekilde üretmek için sinir ağımızın hangi notanın veya akorun bir sonraki olacağını tahmin edebilmesi gerekir. Bu, tahmin dizimizin eğitim setimizde karşılaştığımız her notayı ve akor nesnesini içermesi gerektiği anlamına gelir. Eğitim setinde toplam farklı nota ve akor sayısı 352 idi. Bu, ağın işleyebileceği çok sayıda olası çıktı tahmini gibi görünüyor, bu yüzden de LSTM ağını kullandık.

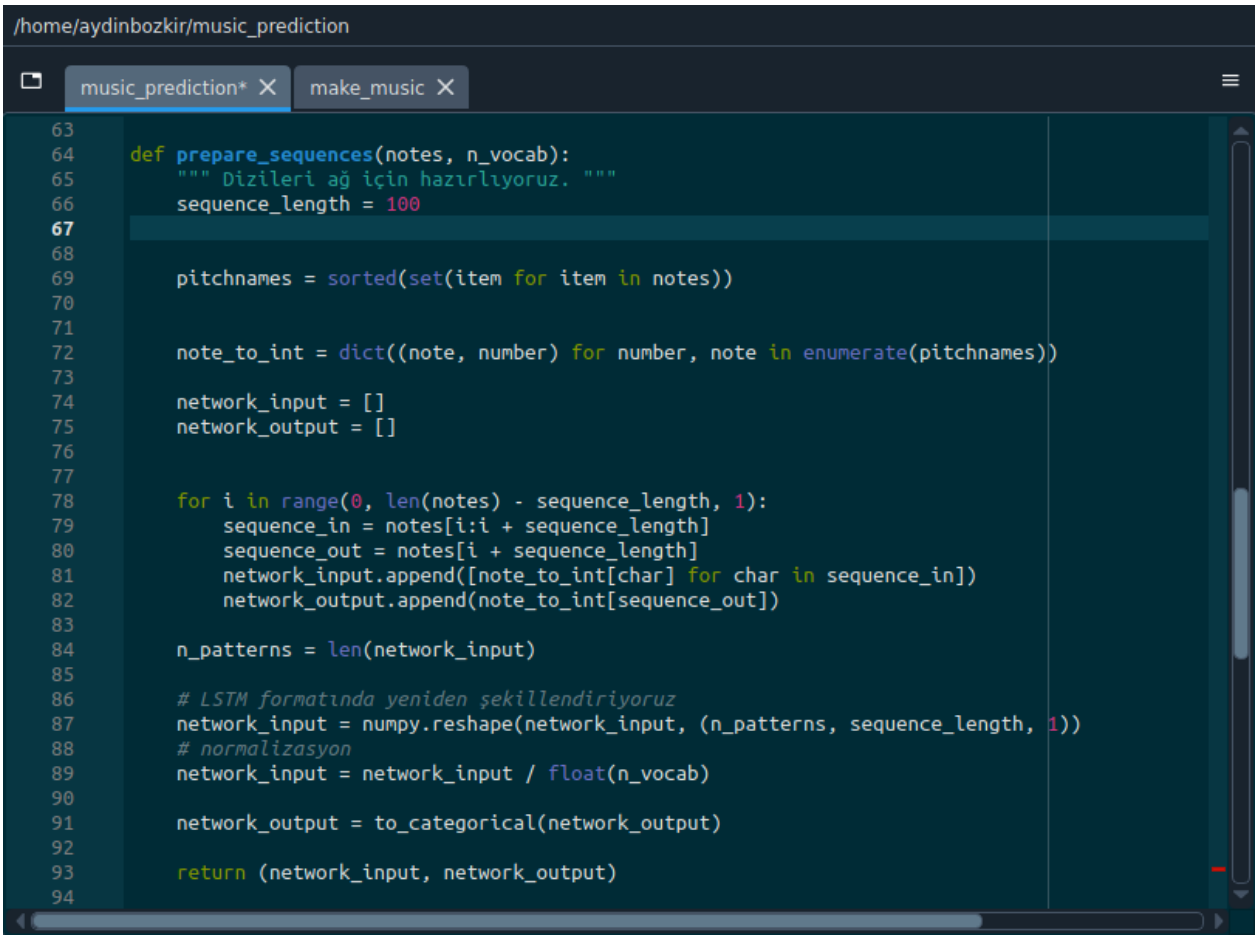
Veri Setini Hazırlama: Verilerimizin nota ve akorlar olacağına karar verdikten sonra artık bir dizi haline dönüştürebiliriz.



Her dosyayı *converter.parse(file)* fonksiyonunu kullanarak bir Music21 akış nesnesine yükleyerek başlıyoruz. Bu akış nesnesini kullanarak dosyadaki tüm notaların ve akorların bir listesini elde ediyoruz. Notanın en önemli kısımları perdenin dize notasyonunu kullanarak yeniden oluşturabildiğinden, her nota nesnesinin perdesini dize notasyonunu kullanarak ekliyoruz.

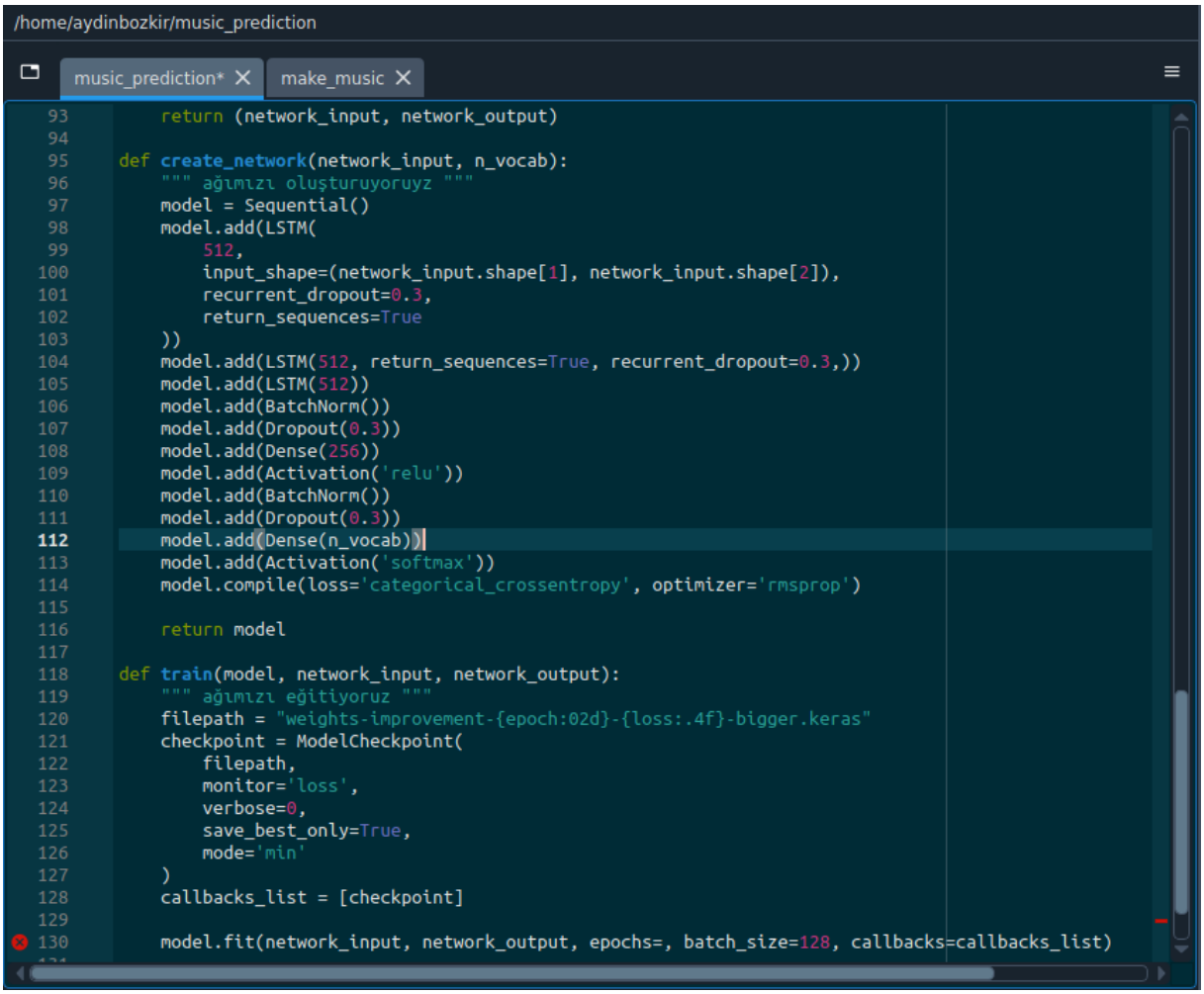
Ve her akoru, akordaki her notanın kimliğini tek bir dizeye kodlayarak ekliyoruz, her nota bir nokta ile ayrılıyor. Bu kodlamalar, ağ tarafından üretilen çıktıyı doğru notalara ve akorlara kolayca çözmemizi sağlar.

Dize tabanlı kategorik verilerden tam sayı tabanlı verilere eşleme yapmak için Mapping fonksiyonu kullanacağız. Bu, sinir ağının dize tabanlı kategorik verilerden çok daya iyi tam sayı tabanlı sayısal verilerle performans göstermesi nedeniyle yapılır. Daha sonra, ağ giriş dizileri ve bunlarla ilgili çıktılar oluşturmamız gerekir. Her giriş dizisinin çıktısı, nota listemizdeki giriş dizisindeki nota dizisinden sonra gelen ilk nota veya akor olacaktır.



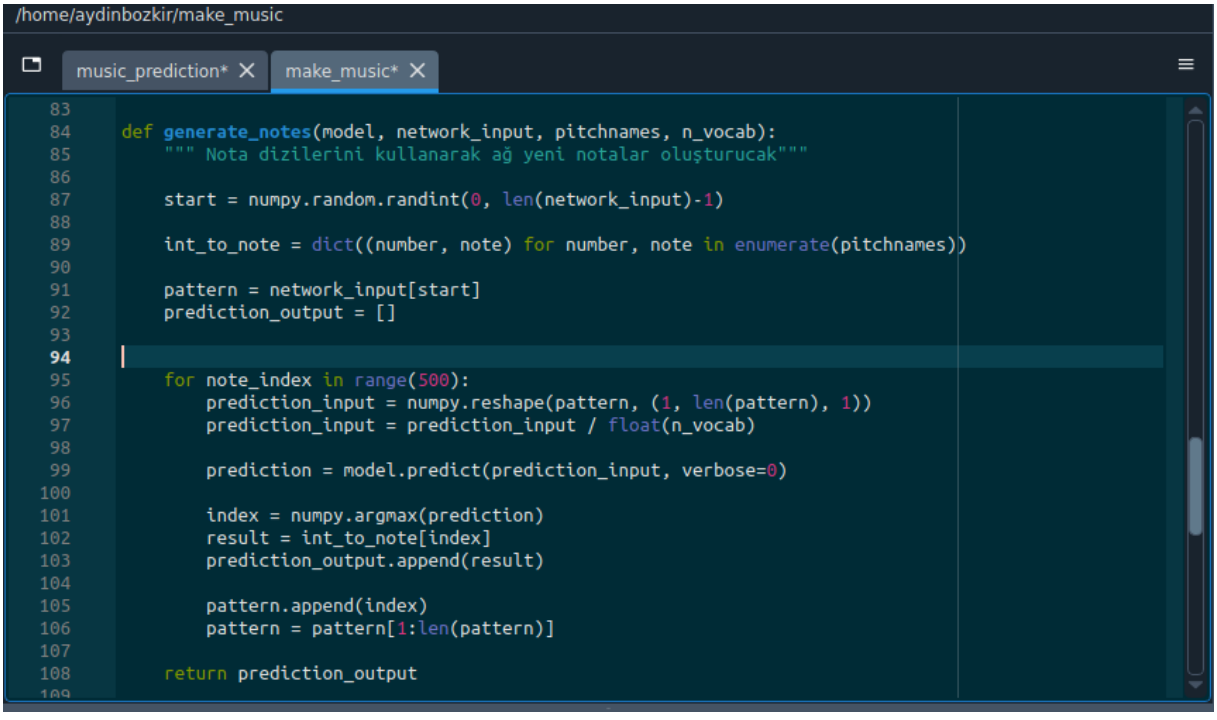
Kod örneğimizde, her dizinin uzunluğunu 100 nota/akor olarak belirledik. Bu, dizideki bir sonraki notayı tahmin etmek için ağın tahminde bulunmasına yardımcı olmak üzere önceki 100 notaya sahip olduğu anlamına gelir. Son olarak girişi normalizasyon yaptık ve çıktıyı one-hot-encoding kodladık.

Model: Ağımızı LSTM tekrarlayan bir ağ olarak kurduk. Bu eğitim için 3 LSTM katmanı, 3 Dropout katmanı ve 2 Dense katmanı kullandık. İlk katmanda input_shape parametresini verdik. Son katman her zaman sistemimizin sahip olduğu farklı çıktı sayısıyla aynı sayıda düğüm içermelidir. Bu, ağın çıktısının doğrudan sınıflarımıza eşleneceğini garanti eder. Eğitimin her yinelemesi için kaybı hesaplamak için categorical_crossentropy kullandık çünkü çıktılarımızın her biri yalnızca tek bir sınıfa ait ve çalışmak için ikiden fazla sınıfımız var. Ve ağımızı optimize etmek için genellikle yinelemeli sinir ağları için çok iyi bir seçim olduğu için bir RMSprop optimizasyon kullandık.



Müzik Oluşturma: Sinir ağını müzik üretmek için kullanabilmek için onu daha öncekiyle aynı duruma getirmemiz gerekecek. Basitleştirmek için, verileri hazırlamak ve ağ modelini daha öncekiyle aynı şekilde kurmak için eğitim bölümündeki kodu yeniden kullanacağız. Ancak ağı eğitmek yerine eğitim bölümünde kaydettiğimiz ağırlıkları modele yükledik.

Elimizde tam bir nota dizi listesi olduğundan, başlangıç noktası olarak listedeki rastgele bir dizi seçeceğiz ve bu da hiçbir şeyi değiştirmeden üretim kodunu yeniden çalıştırmamızı ve her seferinde farklı sonuçlar elde etmemizi sağlayacak.



Ağ kullanarak 500 nota üretmeyi seçtik çünkü bu yaklaşık 2 dakikalık bir müzik oluşturur. Üretmek istediğimiz her nota için ağa bir dizi göndermemiz gerekiyor.

Daha sonra ağdan gelen tüm çıktıları tek bir dizide toplarız. Artık notaların ve akorların tüm kodlanmış gösterimlerine bir dizide sahip olduğumuza göre, bunlar kod çözmeye ve Nota, Akor nesnelerinden oluşan bir dizi oluşturmaya başlayabiliriz.

Öncelikle kod çözdüğümüz çıktının Nota mı yoksa Akor mu olduğunu belirlemeliyiz.

Desen bir Akor ise, dizeyi bir nota dizisine bölmemiz gerekir. Daha sonra her notanın dize gösteriminde döngüye gireriz ve her biri için bir Nota nesnesi oluştururuz. Daha sonra bu notaların her birini içeren bir Akor nesnesi oluşturabiliriz. Desen bir Not ise, desende bulunan perdenin dize gösterimini kullanarak bir Nota nesnesi oluştururuz.

Her yinelemenin sonunda ofseti 0,5 artırırız ve oluşturulan Nota/Akor nesnesini bir listeye ekleriz.

Son olarak ağ tarafından üretilen Notalar ve Akorlar listemiz olduğuna göre, listeyi parametre olarak kullanarak bir Music21 Stream nesnesi oluşturabiliriz. Son olarak ağ tarafından üretilen müziği içeren bir MIDI dosyası oluşturmak için akışı bir dosyaya yazmak üzere Music21 kütüphanesindeki write fonksiyonunu kullanırız.

/home/aydinbozkir/make_music

music_prediction* × make_music* ×

```
109
110 def create_midi(prediction_output):
111     """ tahminlerimizi midi dosyasına çeviriyoruz. """
112     offset = 0
113     output_notes = []
114
115
116     for pattern in prediction_output:
117
118         if ('.' in pattern) or pattern.isdigit():
119             notes_in_chord = pattern.split('.')
120             notes = []
121             for current_note in notes_in_chord:
122                 new_note = note.Note(int(current_note))
123                 new_note.storedInstrument = instrument.Piano()
124                 notes.append(new_note)
125             new_chord = chord.Chord(notes)
126             new_chord.offset = offset
127             output_notes.append(new_chord)
128
129         else:
130             new_note = note.Note(pattern)
131             new_note.offset = offset
132             new_note.storedInstrument = instrument.Piano()
133             output_notes.append(new_note)
134
135
136         offset += 0.5
137
138     midi_stream = stream.Stream(output_notes)
139
140     midi_stream.write('midi', fp='test_output.mid')
141
142 if __name__ == '__main__':
143     generate()
```

Sonuç: Son olarak ağımızı kullanarak yaklaşık 2 dakikalık anlamlı melodiler elde etmeyi başardık. Yanlış notalar tabi ki olacaktır. Daha iyi sonuçlar elde edebilmek için daha büyük bir ağa ihtiyacımız olacaktır. Üretilen müzikler ek dosyasına eklenecektir.