Diziden Diziye Modeli ve MIDI Müzik Veri Tabanı Kullanımıyla Gerçekçi Bir Davul Eşliği Üreteci

A Realistic Drum Accompaniment Generator Using Sequence-to-Sequence Model and MIDI Music Database

Yavuz Batuhan AKYÜZ ve Şevket GÜMÜŞTEKİN
Elektrik-Elektronik Mühendisliği
İzmir Yüksek Teknoloji Enstitüsü
35430 Urla/İzmir, Türkiye
batuhan.akyuz@gmail.com, sevketgumustekin@iyte.edu.tr

2022 30th Signal Processing and Communications Applications Conference (SIU) | 978-1-6654-5092-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/SIU55565.2022.9864843

***Özetçe*— Bu çalışmada, Müzik Enstrümanları Dijital Arabirimi (MIDI) formatında verilen müzik parçalarının davul kısımlarının yapay zeka ile yeniden yorumlanması ve/veya eklenmesi gerçekleştirilmektedir. Bunun için diziden diziye (Sequence-to-Sequence) öğrenme yöntemi ve Kodlayıcı-Kodçözücü (Encoder-Decoder) Uzun Kısa-Vadeli Bellek (LSTM) yapay sinir ağı modeli kullanılmıştır. Bu yapay sinir ağının öğrenmesini geliştirmek için öğretmen zorlama (Teacher Forcing) yöntemi kullanılmıştır. Yeni davul kısımların üretiminde ise, sıcaklık örneklenmesi kullanılarak örneklerin kalitesi ve orijinalliği geliştirilmiştir. Önerdiğimiz yöntem karmaşıklığı ayarlanabilen yüksek kalitede davul eşliği üretmektedir.**

Anahtar Kelimeler—MIDI, dizi-dizi, kodlayıcı ve kodçözücü, uzun-kısa vadeli bellek, öğretmen zorlama, sıcaklık örneklenmesi, otonom müzik eşliği

***Abstract*— In this work, artificial intelligence reinterpretation and/or addition of drum parts for musical pieces supplied in Musical Instruments Digital Interface (MIDI) format, have been carried out. To achieve this, Sequence-to-Sequence learning method and Encoder-Decoder Long Short-Term Memory (LSTM) artificial neural network model have been used. In order to improve training of this neural network, teacher forcing method was utilized. In the generation of new drum parts, the quality and the originality of the samples were improved by using temperature sampling. Our proposed method produces high quality drum accompaniments with adjustable complexity.**

Keywords—MIDI, sequence-to-sequence, encoder and decoder, long-short term memory, teacher forcing, temperature sampling, autonomous music accompany

I. GİRİŞ

Yapay zeka ile müzik üretmek için çeşitli yöntemler kullanılmıştır. Bu yöntemlerde, sadece yapay sinir ağı yapısı değil, aynı zamanda müziğin hangi formatta işlendiği, ön işleme tekniği, vb. gibi müziği temsil edebilecek her türlü bilgi ve farklı yapı önemli rol oynamaktadır [1].

İlk olarak gerçekçi bir şekilde müzik üretimi LSTM yapay sinir ağı kullanılarak başlamıştır. Dougles Eck ve Jürgen Schmidhuber RNN (Recurrent Neural Network) yapısından yola çıkmış fakat bu yapının birbirinden uzak olayların ilişkisini kurmakta başarısız olduğundan LSTM kullanmışlardır [2]. Daha sonra çift eksenli LSTM modeli, polifonik müzik üretmek için kullanılmıştır. Bu yaklaşım nicel ve nitel olarak başarılı sonuçlar vermiştir [3]. Günümüzde, başarılı bir şekilde kullanılan müzik yaratma algoritmaların başında MuseGAN gelmektedir. MuseGAN, çekişmeli üretici yapay sinir ağı (GAN) modeli hem sembolik müzik üretmek için hem de eşlik etmek için kullanılmışlardır [4]. Bu yapıda, müzik barlar halinde alınmış ve yapay sinir ağı bu şekilde eğitilmiştir. Bunun yanı sıra, aynı yapay sinir ağı modeli, polifonik müzik üretmek için kullanılmıştır [5]. MuseGAN'ın başarısına rağmen, bu çalışmamızda MuseGAN yerine, bir müzik parçası içerisindeki spesifik bir enstrümana uygun melodiler üretmek için dizi-dizi kodlayıcı ve kodçözücü LSTM yapay sinir ağı modelinin kullanımı tercih edilmiştir.

Müzik verisi yapı bakımdan bir zaman serisidir. Dizi-dizi LSTM yapay sinir ağı modeli zaman serileri için iyi performans veren bir modeldir. Bu model ile geliştirilmiş olan makine çevirileri ve sohbet robotları en çok bilinen uygulamalardandır. Dizi-dizi yapay sinir ağları, sabit boyutlu bir giriş dizisini alarak çok katmanlı LSTM'den geçirir. Daha sonra buradan çıkan vektörü kullanarak, hedef dizinin kodunun çözülmesi için başka bir çok katmanlı LSTM kullanılır [6]. Bu yapay sinir ağı modeli, makine tercüme alanında hem kelime kelime tercümanlıkta hem de harf harf tercümanlıkta oldukça yaygın bir şekilde kullanılmış ve olumlu sonuçlar vermiştir [7]. Bunun dışında aynı model, videoları metne çevirmek için de kullanılmıştır. Bunu yapmak için giriş olarak her videonun

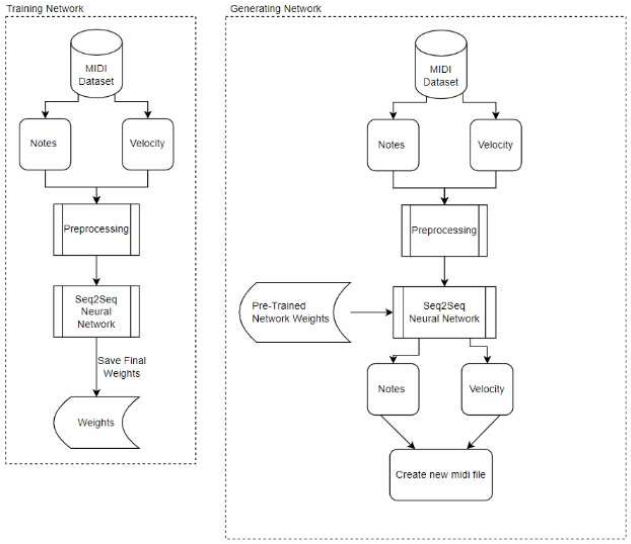
978-1-6654-5092-8/22/$31.00 ©2022 IEEE

kareleri alınmış ve çıkış olarak bunlara karşılık gelen metinler kullanılmıştır [8].

Bilgisayar ortamında otonom müzik üretmenin başlıca zorluğu, kuralları ve beklentileri karşılamanın yanında müziğin yaratıcı bir şekilde üretmek ve yapaylık hissettirmemektir. Dinleyicinin müziği doğal ve akıcı olarak hissetmesi gerekir. Bu sebepten dolayı, müziğin doğasını anlamak ve bunu doğru bir şekilde işlemek önemlidir. Bu hedefimizin ışığında, çalışmamızda, bir müziğe eşlik etmesi için dizi-dizi yapay sinir ağı modeli kullanılarak, davul enstrümanına ait nota ve nota vuruş hızları üretilmiştir. Yapay sinir ağımız, farklı müzik türleri ile eğitilmiş ve bunun sonucu olarak bu müzik türlerinin özelliklerini barındıran davul kısımları elde edilmiştir. Bunun yanı sıra, üretilen davul enstrümanlarının karmaşıklıkları sıcaklık örneklenmesi ile ayarlanabilmektedir. Bu sayede davul performansının kullanıcının tercih ettiği seviyede öne çıkması sağlanmıştır.

II. MIDI Davul Eşliği Yapay Sinir Ağı

Bu projede kullanılan yapay sinir ağı dizi-dizi yapar sinir ağı olarak ve veri formatı MIDI formatı olarak seçilmiştir. Yapay sinir ağına giriş dizisi olarak veri kümesi içerisindeki müziklerdeki enstrüman notaları verilmiş ve bunları takip eden davul notaları ise çıkış dizisi olarak verilmiştir. Bu sayede yapay sinir ağının bu enstrümanlar arasındaki etkileşimi öğrenmesi sağlanmıştır. Bununla kalmayıp, yeni şarkılar için üretilecek davul kısımlarının daha az robotik hale getirmek için, aynı işlem notların vuruş hızları için de kullanılmıştır. Takip eden figürde, yapılan projenin eğitim ve üretim akış şemaları gösterilmektedir.



Şekil. 1. Yapay Sinir Ağı Eğitim ve Üretim Akış Şemaları

A. MIDI Formatı

MIDI formatı, müziğin bilgisayar ortamında tanımlanması için kullanılan dijital bir formattır. Bu format müzik ile ilgili notaların başlangıç ve bitiş sürelerini, çalınan notanın hangi enstrümana ait olduğunu, ritmini, nota değerlerini, notaların vuruş hızlarını, vb. verileri kapsamaktadır. Bu format içerisinde, yapay sinir ağını eğitmek için notaların değerleri, vuruş hızları ve enstrüman bilgileri kullanılmıştır. Nota ve vuruş hız değerleri, 0 ile 127 arasındaki bütün tam sayılarla tanımlanmaktadır. Ayrıştırılmış bir midi verisindeki örnek girdiler tablo 1'de gösterilmiştir.

TABLO I. Ayrıştırılmış MIDI Verisi Örneği

Start	End	Pitch	ID-Instrument	Velocity
0.000	0.250	49	0-Drum	30
0.000	0.250	38	0-Drum	105
0.000	0.250	64	30-Distortion Guitar	109
0.000	2.000	66	33-Electric Bass Guitar	95
0.000	2.000	67	30-Distortion Guitar	95
0.000	2.000	72	30-Distortion Guitar	95
0.000	2.000	33	30-Distortion Guitar	95
0.250	0.375	58	30-Distortion Guitar	95
0.250	0.500	38	0-Drum	109
0.250	0.500	42	0-Drum	109
0.375	0.625	61	30-Distortion Guitar	95
0.500	0.625	38	0-Drum	109

B. Veri Ön İşlemesi

Önceden de bahsedildiği üzere projedeki ana amacımız; davulun verilen bir müziğe gerçeğe yakın bir şekilde eşlik etmesi olduğundan, MIDI verisinde bizim için önemli olan bilgiler enstrüman cinsi, nota değerleri ve nota vuruş hızlarıdır.

Enstrüman bilgisi kullanılarak, giriş (1) ve çıkış (2) matrisleri oluşturulmuştur. Yapay sinir ağına giriş olarak davul haricindeki bütün enstrümanlara ait olan nota değerleri ve nota vuruş hızları alınmış, davul enstrümanına ait olan nota değerleri ve nota vuruş hızları ise çıkış bilgisi olarak kullanılmıştır.

Bu dizilerin uzunluğu belli bir periyot ile alınmıştır. Deneme yanılma yöntemi ile bu periyot 20 örnek olarak belirlenmiştir. Eğer, giriş verimizde periyottan daha az örnek varsa, bu dizinin sonuna sıfır dolgulaması yapılarak örnek sayısı 20'ye tamamlanmıştır. Bu matrislerdeki sütun sayımız periyodu; satır sayımız ise veri kümemizden her birinin uzunluğu 20 olan toplam dizi sayımızı temsil etmektedir.

$$notes_{in} = \begin{bmatrix} 64 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 42 & \cdots & 40 \end{bmatrix}_{MxN}, vel_{in} = \begin{bmatrix} 109 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 105 & \cdots & 0 \end{bmatrix}_{MxN} \quad (1)$$

$$notes_{out} = \begin{bmatrix} 49 & \cdots & 47 \\ \vdots & \ddots & \vdots \\ 56 & \cdots & 0 \end{bmatrix}_{MxN}, vel_{out} = \begin{bmatrix} 30 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 90 & \cdots & 0 \end{bmatrix}_{MxN} \quad (2)$$

Matrislerindeki (*i, j*) konumunda,

i: O andaki, nota veya vuruş hızı değerini,
j: Örneğin periyot içindeki konumunu,

ifade etmektedir.

Verilerimizi, yapay sinir ağında kategorik bir şekilde eğitmek amacıyla bir değişken kodlaması yapılmıştır. II. A'da belirtildiği gibi nota değerleri ve nota vuruş hızları 0 – 127

arasında değer almaktadır. Bu yüzden toplam kategori sayımız 128 adettir.

C. Yapay Sinir Ağı ile Eğitim

Bu çalışmada, yapay sinir ağı modeli olarak, tercüme alanında da çokça kullanılan dizi-dizi kodlayıcı – kodçözücü yapay sinir ağı modeli kullanılmıştır. Bu modeli seçmemizdeki başlıca neden, buradaki problemin, yöntemin başarılı olduğu tercüme problemine olan benzerliğidir.

Modelimizde ayrıca öğretmen zorlama yöntemi kullanılmıştır. Bu yöntemde, eğitim aşamasında kodçözücünün ürettiği çıkışı kullanması yerine, önceki aşamada beklenen/istenilen gerçek çıkışı kullanması sağlanır.

Eğitim sırasında, eniyileme işlevi olarak "Adam" kullanılmış ve öğrenme oranımız 0.001'den başlatılmıştır. Algoritmanın öğrenme hızını artırmak, ağın öğrenme sürecini kolaylaştırmak ve minimum yitime yakınsamasını sağlamak için öğrenme hızı [9]'da olduğu gibi kademeli olarak azaltılmıştır.

Kullandığımız yapay sinir ağının çalışma adımları aşağıdaki gibi sıralanabilir:

Adım 1. Kodlayıcıya giriş vektörü (x_t) verilir.

$$x_t = [64\ \ 66\ \ 67\ \ 72\ \ 33\ \ \dots\ \ 0\ \ 0\ \ 0] \quad (3)$$

Adım 2. Kodlayıcı bu vektörü (3) kullanarak, kendisine ait bağlam vektörleri olan h ve c'yi (4) oluşturur.

$$h, c = LSTM(x_t) \quad (4)$$

Adım 3. Kodçözücüye istenilen/beklenilen gerçek çıkış bir zaman adımı kaydırılarak ve başına özel "BAŞLA" komutu anlamına gelen işaret konarak verilir. Bizim örneğimizde (şekil 2) ve denklem 5'te, bu "BAŞLA" komutu sıfır rakamı ile temsil edilmiştir.

Not: Dizi başındaki sıfır rakamı "BAŞLA" komutuna denk gelirken, dizi sonundaki sıfır rakamları dolgulama için kullanılmışlardır.

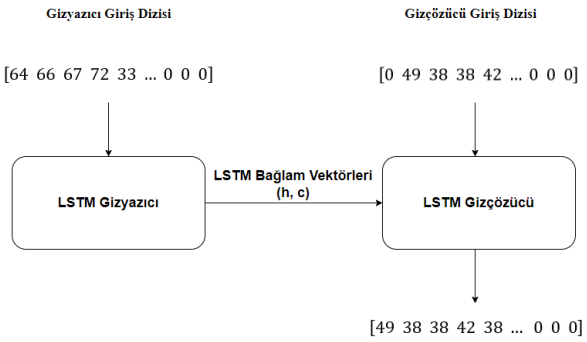
$$y_{real_t} = [0\ \ 49\ \ 38\ \ 38\ \ 42\ \dots\ 0\ \ 0\ \ 0] \quad (5)$$

Adım 4. Kodçözücü, kodyazıcıdan çıkan bağlam vektörlerini kullanarak (4) ve beklenilen girişteki dizi elemanlarını (5) sırasıyla, bir sonraki dizi elemanı çıkışı olarak verir. Bunu yaparken aktivasyon fonksiyonu olarak "düzgeli üstel fonksiyon (softmax)" fonksiyonunu kullanır.

$$y_{pred_t} = softmax\left(LSTM(y_{real_t}, h, c)\right) \quad (6)$$

Adım 5. Bu işlem, çıkış dizimiz tam bir periyot kadar uzunlukta oluncaya kadar veya sıfır rakamını buluncaya kadar devam eder.

Model bu veri kümesindeki bütün dizileri tek tek giriş olarak, ağırlık matrislerini her epokta günceller. Ağırlık matrislerini güncellemek için yitim fonksiyonu olarak kategorik çapraz entropi kullanılmaktadır.



Şekil. 2. Yapay Sinir Ağı Kodyazıcı –Kodçözücü Eğitim Modeli

D. Yapay Sinir Ağı ile Üretim

Yeni parçaları üretmek için, daha önceden eğittiğimiz modelin ağırlık matrisleri kullanılır. Davul eklenmesi istenilen şarkının, nota ve nota vuruşları giriş olarak kullanılır. Yapay sinir ağı, yeni bir çıkış üretmek için her kategorinin olasılığının bulunduğu bir vektör $\vec{z} = (z_1, z_2, \dots, z_n)$ oluşturur. Denklem 7 kullanılarak, bu vektörün her bir elemanı, içerisindeki diğer elemanlarla karşılaştırılarak ve eşiksiz en büyük işlevinden geçirilerek, yeni bir olasılık vektörü olan $\vec{q} = (q_1, q_2, \dots, q_n)$ vektörünü oluşturur.

$$q_i = \frac{e^{z_i}}{\sum_j e^{z_j}} \quad (7)$$

Bu yöntem sorunsuz bir şekilde çalışmakla birlikte çıkış olasılık değerlerimiz çok büyük olduğunda, yapay sinir ağı kendine çok güvenir ve hep doğruyu bulduğuna inanır. Bunun sonucu olarak hep kendini tekrarlayan, gerçeğe uzak olan ve ilginç olmayan diziler elde edilir.

Bu durumun üstesinden gelmek için sıcaklık örneklenmesi kullanılmıştır. Standart bir teknik olan sıcaklık örneklenmesinin asıl amacı üretilen örneklerin kalitesini arttırmaktır. Bunu gerçekleştirmek için, çıkışın olasılık dağılımlarını değiştirmeye yarayan sıcaklık parametresi "T" kullanılır. Bu sayede kendini tekrarlamayan, benzersiz ve ilginç sonuçlar elde edilir. Sıcaklık örneklenmesi için [10]'da önerildiği gibi, (7)'ye "T" parametresi eklenmiş ve (8) içerisinde kullanılmıştır.

$$q_i = \frac{e^{\left(\frac{z_i}{T}\right)}}{\sum_j e^{\left(\frac{z_j}{T}\right)}} \quad (8)$$

Sıcaklık parametresi, projemizde karmaşıklık değeri olarak kullanılmıştır. Üretilen parçaların karmaşıklık değeri, sıcaklık parametresi T=1 seçilirse; daha karmaşıklığı az, kendini tekrar eden davul dizileri üretirken, bu değer 0'a yaklaştıkça daha benzersiz ve özgün davul dizileri oluşturmaktadır.

Yapay sinir ağında üretilen notaların uygunluğu, kontrol edilmesi için uygunluk testinden geçmektedir. Uygunluk testi, bizim oluşturmuş olduğumuz ve davulun robotik hissettirmemesi için yapılan bir testtir. Bir insan, davul

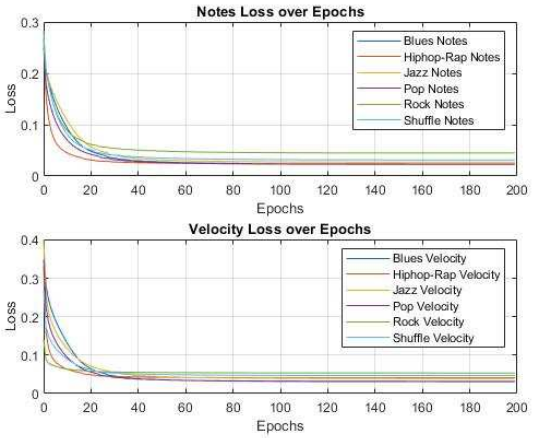
çalarken iki kolu ve iki ayağı sayesinde maksimum dört uzvunu aynı anda kullanabilir. Bu sebepten ötürü eğer üretilen dizinin uzunluğu dörtten fazlaysa, bu koşula uyan ilk dört nota alınır. Bir dizide, eğer aynı zaman çerçevesinde iki adet aynı nota veya ikiden fazla aynı uzvun kullanılması gereken notalar varsa, bu notalar göz ardı edilir.

III. SONUÇLAR

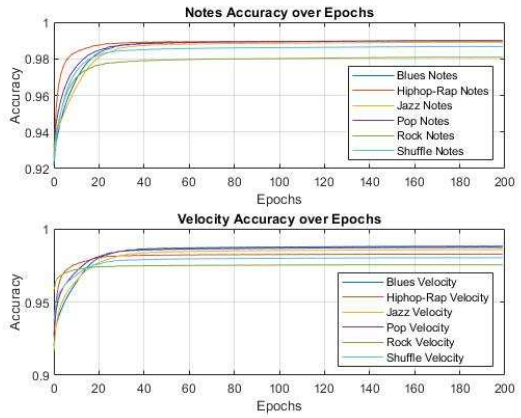
Bu çalışma kapsamında; yapay sinir ağı kullanarak blues, caz, hiphop-rap, pop, rock ve bu türlerin hepsinin bir arada bulunduğu shuffle veri kümelerinin kullanımıyla davul sesi üretilmiştir. Bunun sonucu olarak, kullanıcı davul eklemek istediği parça için, istediği müzik türünü seçebilir ve bu müzik türünün etkilerini görebilir. Bu sayede, kullanıcıya farklı deneyimler ve seçenekler sunulması hedeflenmiştir.

Yapay sinir ağımızın, 200 epok ve 64 parti boyutu ile eğitimi sağlanmıştır. Her bir veri kümesi ve üç farklı karmaşıklık değeri kullanılarak, eğitim veri seti içerisinde bulunmayan bir MIDI dosyası üzerindeki sonuçlarına, kaynaklar kısmında belirtilen soundcloud bağlantısı [11] aracılığıyla ulaşılabilmektedir.

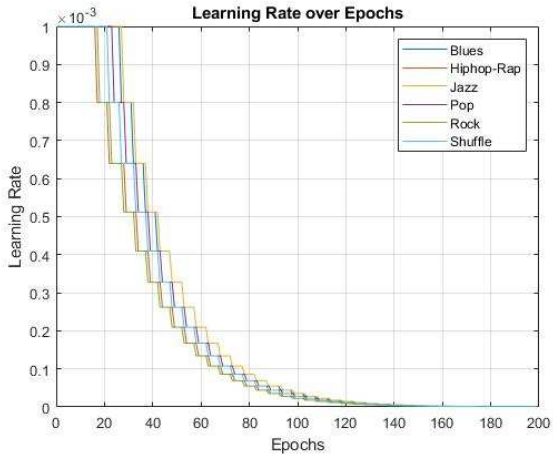
Takip eden grafiklerde, yapay sinir ağımızın yinelemeye karşı yitiminin, doğruluğunun ve eniyileme değerinin nasıl değiştiğini gösterilmektedir.



Şekil. 3. Nota ve Nota Vuruş Hızı Yitiminin Yinelemeye Karşı Grafikleri



Şekil. 4. Nota ve Nota Vuruş Hızı Doğruluğunun Yinelemeye Karşı Grafikleri



Şekil. 5. Eniyileme Değerinin Yinelemeye Karşı Grafiği

Çalışma sonucunda, hedeflenen gerçekçi sonuçlara ulaşılmış olsa da, kullanılmakta olan yapay sinir ağı daha da geliştirilebilir. Bunun için öncellikli olarak, veri kümesinin periyot sayısı ve veri kümesinin boyutu ve periyot sayısı arttırılarak yitim daha da düşürülebilir. Bunun yanı sıra, üretilen müziği daha da gerçekçi yapmak amacıyla, notaların zamanlamaları giriş olarak veya yardımcı giriş olarak kullanılabilir.

KAYNAKLAR

[1] J. Briot, G. Hadjeres and F. Pachet, "Deep Learning Techniques for Music Generation -- A Survey", arXiv.org, 2019. [Online]. Available: https://arxiv.org/abs/1709.01620

[2] D. Eck and J. Schmidhuber, "Finding temporal structure in music: blues improvisation with LSTM recurrent networks," Proceedings of the 12th IEEE Workshop on Neural Networks for Signal Processing, 2002, pp. 747-756, doi: 10.1109/NNSP.2002.1030094.

[3] N. Kotecha and P. Young, "Generating Music using an LSTM Network", arXiv.org, 2018. [Online]. Available: https://arxiv.org/abs/1804.07300

[4] H. Dong, W. Hsiao, L. Yang and Y. Yang, "MuseGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment", arXiv.org, 2017. [Online]. Available: https://arxiv.org/abs/1709.06298

[5] H. Dong and Y. Yang, "Convolutional Generative Adversarial Networks with Binary Neurons for Polyphonic Music Generation", arXiv.org, 2018. [Online]. Available: https://arxiv.org/abs/1804.09399

[6] I. Sutskever, O. Vinyals and Q. Le, "Sequence to Sequence Learning with Neural Networks", arXiv.org, 2014. [Online]. Available: https://arxiv.org/abs/1409.3215v3

[7] G. Neubig, "Neural Machine Translation and Sequence-to-sequence Models: A Tutorial", arXiv.org, 2017. [Online]. Available: https://arxiv.org/abs/1703.01619v1

[8] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell and K. Saenko, "Sequence to Sequence -- Video to Text", arXiv.org, 2015. [Online]. Available: https://arxiv.org/abs/1505.00487v3

[9] Z. Li and S. Arora, "An Exponential Learning Rate Schedule for Deep Learning", arXiv.org, 2019. [Online]. Available: https://arxiv.org/abs/1910.07454

[10] G. Hinton, O. Vinyals and J. Dean, "Distilling the Knowledge in a Neural Network", arXiv.org, 2015. [Online]. Available: https://arxiv.org/abs/1503.02531

[11] "MIDI Drum Accompaniment Network", SoundCloud, 2022. [Online]. Available: https://soundcloud.com/yba_ee/sets/samples/s-xIWP39WCj9A?utm_source=clipboard&utm_medium=text&utm_campaign=social_sharing (short URL: https://tinyurl.com/2n5atj9s)