

KÜTAHYA SAĞLIK BİLİMLERİ ÜNİVERSİTESİ



Yapay Zeka Tabanlı Ses Üretimi

Beyzanur Dağdelen

Vize Raporu

Danışman

Dr.Öğr.Üyesi Emre Güngör

Yapay Zeka & Veri Odaklı Sistem Tasarımı & Latex İle Rapor Hazırlama

25.04.2024

İçindekiler

1	Giriş	2
1.1	Variational Autoencoder (VAE): Yaratıcı ve Esnek Veri Üretimi	3
1.1.1	VAE'nin İşlevi ve Autoencoder (AE) ile Farkı	4
1.1.2	VAE'nin Çalışma Prensipleri	4
1.1.3	GAN ile Bağlantısı	5
1.1.4	Kullanım Alanları	5
1.1.5	VAE & VQ-VAE & VAE-GAN	6
1.1.5.1	VAE	7
1.1.5.2	VQ-VAE	7
1.1.5.3	VAE-GAN	8
2	Entropi	9
3	Çapraz Düzensizlik (Cross-Entropy) ve KL Iraksaklığı (KL Divergence)	13
4	Kullback-Leibler Iraksaklığı (KL-Divergence)	14
5	Optimizer	15
6	Aktivasyon Fonksiyonu	17
7	Sound datasets	19
7.1	N-synth	19
7.2	MAESTRO	19
8	Ses Türleri	20

9 Ses Temsil Etme Yöntemi	20
10 Raw-audio	20
11 Spektrogram	21
12 Spektrum	21
13 Mel-Frekans Cepstral Katsayıları (MFCC-Mel-Frequency Cepstral Coefficients)	22
14 Üretimde Inputlar	22

1 Giriş

Bu proje için ilk hafta, yapay zeka ile ses üretimine odaklanırken unreal engine ortamındaki sanal gerçeklik teknolojisi ile aynı zamanda insanı huzurlu hissettiren ortamlarda bu deneyimi sağlaması amaçlanmıştır. İyileştirici etkiye sahip farklı frekanslardaki sesleri, yapay zeka ile üreterek insanlar üzerindeki olumsuz duygularını azaltarak, iyileştirmek ve motivasyon sağlayarak hayatlarındaki gelişen süreci desteklemektir. Bu hedeflere ulaşmak için, VQ-VAE (Vector Quantized-Variational Autoencoder) gibi ileri yapay zeka tekniklerinden faydalanılabilir. VQ-VAE, seslerin temsili için kullanılabilir ve farklı frekanslardaki sesleri analiz edip, daha sonra bu sesleri yeniden oluşturmak için kullanılabilir. Bu teknik, seslerin duygusal etkilerini inceleyerek, kullanıcıların duygusal durumlarını iyileştirmek için özelleştirilmiş sesler oluşturabilir. Ayrıca, bu seslerin Unreal Engine gibi sanal gerçeklik ortamlarında kullanılmasıyla, kullanıcılar huzur ve motivasyon sağlayan görsel ve işitsel deneyimler yaşayabilirler. Bu şekilde, VQ-VAE ve sanal gerçeklik bir araya getirilerek, insanların duygusal refahlarını artırmak için yenilikçi bir yaklaşım sunulabilir. OpenAI Jukebox, farklı müzik türlerinde ve tarzlarda insan yapımı şarkılara benzeyen yeni müzik parçaları üretmek için tasarlanmış bir yapay zeka (AI) projesidir. Jukebox, insanlar tarafından oluşturulan çeşitli müzik türlerini analiz ederek, benzer tarzda orijinal parçalar üretebilen bir yapay zeka sistemini içerir. Bu proje, derin öğrenme algoritmaları kullanarak geniş bir müzik veri kümesi üzerinde eğitilmiş bir model içerir. Model, farklı müzik enstrümanlarını ve vokal stillerini kapsayan geniş bir yelpazede müzikal varyasyonları üretebilir. Ayrıca, belirli müzik türleri veya

sanatçıların tarzlarını taklit etme yeteneğiyle dikkat çeker. OpenAI Jukebox, müziğin sınırlarını genişletirken aynı zamanda yaratıcılık ve yapay zeka alanında büyük bir ilerleme sunar [1] .

Jukebox'ın temel özellikleri şunlardır:

- Geniş bir müzik yelpazesini kapsar: Rock, pop, klasik, caz, elektronik ve daha fazlası dahil olmak üzere çok çeşitli müzik türlerinde şarkılar oluşturabilir.
- Şarkı sözleri de üretebilir: Sadece melodiler ve armoniler değil, aynı zamanda insan yapımı şarkı sözlerine benzeyen metinler de oluşturabilir.
- Kullanıcı tarafından özelleştirilebilir: Kullanıcılar, belirli bir sanatçı, tür veya ruh hali gibi kriterlere göre müzik oluşturmak için Jukebox'ı yönlendirebilirler.
- Etkileyici sonuçlar verir: Jukebox tarafından oluşturulan müzik parçaları, insan yapımı müziklerden ayırt edilemeyecek kadar yüksek kalitede olabilir.

1.1 Variational Autoencoder (VAE): Yaratıcı ve Esnek Veri Üretimi

Variational Autoencoder (VAE), veri temsili öğrenmek ve yaratıcı bir şekilde yeni veri noktaları üretmek için kullanılan önemli bir modeldir. VAE, derin öğrenme modelleri sınıfında yer alan bir tür yapay sinir ağıdır. Amacı, veri setindeki örüntüleri öğrenerek, veriyi daha az boyutta bir temsile sıkıştırmak ve bu temsili kullanarak yeni veri noktaları

oluşturmaktır. Genellikle denetimsiz öğrenme yöntemi olarak kullanılan VAE, veri setinin içsel yapısını anlama yeteneğine sahiptir [2].

1.1.1 VAE'nin İşlevi ve Autoencoder (AE) ile Farkı

VAE, geleneksel bir autoencoder (AE) gibi çalışır, ancak belirgin bir farklılık bulunur. AE, gizli bir temsil (latent representation) oluşturmak için veriyi sıkıştırır ve ardından bu temsil aracılığıyla veriyi yeniden oluşturur. VAE ise sadece veriyi sıkıştırmakla kalmaz, aynı zamanda verinin olası dağılımını modelleyerek, veri noktalarının olasılık dağılımını öğrenir. Bu da VAE'nin veri üretiminde daha esnek ve çeşitlilik yaratabilme yeteneğine sahip olmasını sağlar.

1.1.2 VAE'nin Çalışma Prensibi

VAE'nin temel prensibi, veriye gizli bir yapısal temsil oluşturmak ve bu temsil aracılığıyla yeni veri noktaları üretmektir. Bu süreç iki ana bileşen üzerine kuruludur: encoder ve decoder. Veriyi latent uzayın parametreleri

olan bir dağılımın parametreleri haline getiren encoder ağıdır. Bu dağılım genellikle Gaussian (normal) dağılımı olarak belirtilir. Encoder, veriyi sıkıştırarak, her bir veri noktasını latent uzayın içindeki bir noktaya eşler. Latent uzaydaki noktaları orijinal veriye çeviren ise decoder ağıdır. Encoder

tarafından elde edilen latent uzay temsili alr ve bu temsili orijinal veriye dönüştürür.

1.1.3 GAN ile Bağlantısı

GAN, gerçekçi veri üretmek için kullanılırken, VAE daha yapılandırılmış ve kontrollü bir şekilde veri üretir. Bazı çalışmalar, VAE'nin öğrendiği temsili GAN yapısında kullanarak daha kaliteli ve çeşitli veri üretimi sağladığını göstermiştir.

1.1.4 Kullanım Alanları

Kullanım alanları geniş olan VAE'ye şunlar örnek verilebilir:

- Görüntü ve Video İşleme: Görüntü tamamlama, yeniden oluşturma ve sentezleme işlemlerinde kullanılabilir.
- Doğal Dil İşleme: Dil modelleme, metin üretimi ve dilin öğrenilmesi gibi alanlarda kullanılabilir.
- Yaratıcı İçerik Üretimi: Sanat, müzik ve edebiyat gibi yaratıcı içeriklerin oluşturulmasında kullanılabilir.

Variational Autoencoder, derin öğrenme ve generatif modelleme alanındaki önemli bir gelişmedir. Verinin daha esnek bir temsiliyi öğrenerek hem veri üretiminde kullanılabilirliği hem de verinin içsel yapısını anlama yeteneği ile birçok alanda geniş bir kullanım potansiyeline sahiptir.

1.1.5 VAE & VQ-VAE & VAE-GAN

Yapay zeka tabanlı model türleri olan bu üç model, özellikle ses ve görüntü üretmede kullanılırlar [3].

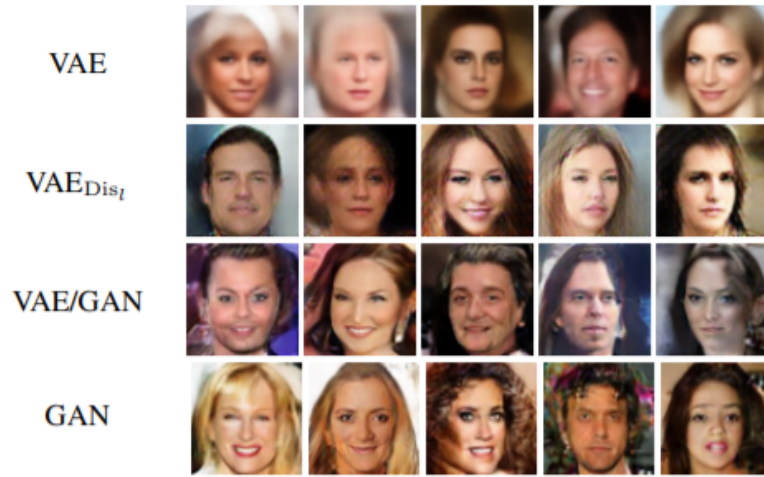


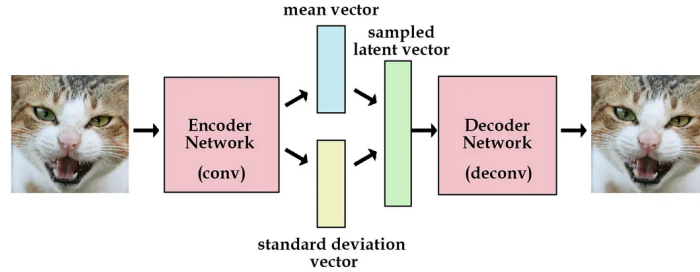
Figure 3. Samples from different generative models.



Figure 4. Reconstructions from different autoencoders.

Şekil 1: query&zoutput
[3]

1.1.5.1 VAE Vanilya VAE en iyi bilinen üretken modellerden biridir (GAN dışında). Bir aşağı örnek, genellikle z olarak adlandırılan bir gizli vektör ve bir yukarı örnekten oluşan üç bölümü vardır. Aşağı örnek genellikle konvolüsyonlardan oluşur ve yukarı örnek ya transpoze edilmiş bir konvolüsyondur ya da normal bir yukarı örnek ve yukarı örnekten tutulan uzamsal boyutlarla konvolüsyon içeren bir bloktur. Bununla birlikte, konvolüsyonlu yukarı örnekleme genellikle normal transpoze konvolüsyonlardan daha düzgün sonuçlar üretir.

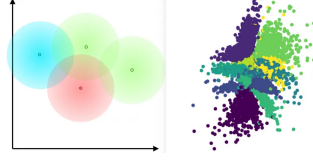


Şekil 2: VAE

[3]

1.1.5.2 VQ-VAE VQ-VAE, vanilya VAE'nin ayırt edici özelliği olan gizli uzayın yeniden yaratılmasıdır. Bu, gizli uzayı ortadan kaldırmak anlamına gelmemektedir, sadece sürekli yerine ayırık olmasına izin verir, dolayısıyla adım Vektör Niceleme VAE'dir. Gizli uzaydaki noktalar VQ-VAE'de ayırıktır ancak VAE'de sürekli [4].

En yakın embeddingi ya da gizli uzayda örnekleme yapılacak en yakın noktayı hesaplayan bir fonksiyon ekleyerek bulanık çıktılar sorununu çözülmesi hedeflenmektedir. Çünkü VQ-VAE eğitildiği bir noktadan örnekleme alınırsa daha keskin olacaktır. VQ-VAE 2 görüntü kalitesi ve çoklu nesil çeşitliliği ölçütlerinde BIG-GAN'ı geride bırakmıştır. Bununla



Şekil 3: VQ-VAE

[4]

birlikte, aynı sonucu elde etmenin bir başka yolu da GAN'ın discriminatorunu kullanmaktır.

1.1.5.3 VAE-GAN GAN'lar bazen insanları bile kandırabilecek hiper-gerçekçi görüntüler üretebilmektedir fakat amaçlanan özelliklere sahip görüntüler üretmek için VAE'nin gizli uzay gibi bazı kısımlarına hala ihtiyaç duyulabilmektedir. Bunu yapmak için, GAN'ın ayırt edicisini VAE'nin dönüştürülmüş konvolüsyonlarından sonra eklemelidir. Görüntülerin daha keskin ve daha gerçekçi olmasına yardımcı olmak için VAE'nin loss fonksiyonuna bir ayırıcı kayıp bileşeni eklenmelidir [5].

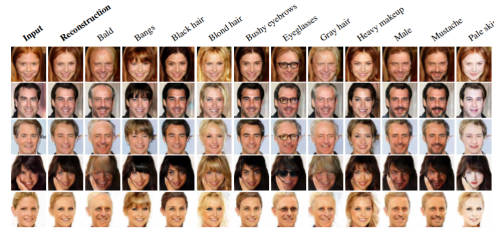


Figure 5. Using the VAE/GAN model to reconstruct dataset samples with visual attribute vectors added to their latent representations.

Şekil 4: VAE-GAN

[3]

2 Entropi

Entropi(düzensizlik), bir olayın veya iletinin içerdği bilgi miktarıdır [11].

Bilgiyi ise bilgisayarlarda bitler halinde depolanır. Bilginin depolama işlemi gerçekleştirilirken minimum bit sayısı olacak şekilde saklanması amaçlanır.

Olayların gerçekleşme olasılığı ile içerdği bilgi miktarı karşılaştırılırsa;

- Düşük olasılıklı gerçekleşen olaylar büyük miktarda bilgi içermektedir.
- Yüksek olasılıkla gerçekleşen olaylar az sayıda bilgi içermektedir.

Bir olayın içerdği bilgi miktarını formülize edildiğinde aşağıdaki(??) formül elde edilir.

$$Bilgi(x) = -\log_2(p(x)) [11] \quad (1)$$

Hilesiz bir madeni para için para yazı-tura oyunu için havaya atıldığında aşağıdaki tablo ortaya çıkmaktadır(1).

Table 1: Hilesiz Para Örneği [11]

Çıktılar	Yazı	Tura
Olasılık	0.5	0.5
Bit Gösterimi	0	1
Bit Sayısı	1	
Bilgi: $H(p) = -\log_2(p(x))$	1	1

Bu olayların gerçekleşme olasılığı eşit olduğundan olaya dahi taşıdıkları bilgi miktarları birbirine eşit ve 2'dir.

Madeni para hileli olduğunda yazı ve tura gelme olasılıkları eşit değildir(2).

Table 2: Hileli Para Örneği [11]

Çıktılar	Yazı	Tura
Olasılık	0.25	0.75
Bit Gösterimi	0	1
Bit Sayısı	1	
Bilgi: $H(p) = -\log_2(p(x))$	2	0.415

Hilesiz 2 adet bağımsız madeni paranın değerleri aşağıdaki gibidir(3);

Table 3: Hileli Para Örneği [11]

Çıktılar	Yazı-Yazı	Yazı-Tura	Tura-Yazı	Tura-Tura
Olasılık	0.25	0.25	0.25	0.25
Bit Gösterimi	00	01	10	11
Bit Sayısı	2			
Bilgi: $H(p) = -\log_2(p(x))$	2	2	2	2

Olasılıkları farklı olan 4 olayı incelendiğinde(4);

Table 4: Hileli Para Örneği [11]

Çıktılar	X_1	X_2	X_3	X_4
Olasılık	0.5	0.25	0.125	0.125
Bit Gösterimi	1	01	000	001
Bit Sayısı	1	2	3	3
Bilgi: $H(p) = -\log_2(p(x))$	1	2	3	3

Burada bit gösterimlerinin ve sayılarının farklı olduğu gözükmemektedir.

Burada olaylar 3'e ayrılmış gibi görülebilmektedir. Yani p: olayın gerçekleşme olasılığı gösterilirse(3);

$$[!hb]P(X_1) = P(x_2) + P(x_3) + P(x_4) \quad (2)$$

$$[!ht]P(x_2) = P(x_3) + P(x_4) \quad (3)$$

şeklindedir.

İçerdikleri bilgi miktarlarına bakıldığında gerçekleşme olasılığı düşük olan olayların daha yüksek bilgi içerdikleri gözükmemektedir.

İçerdiği bilgi hesaplamasını rastgele bir X değişkeni için hesaplırsak tüm gerçekleşen olaylar üzerinden beklenen bilgiye(expected information) bakılmaktadır. Entropi formüllerinden birisi olan Shannon Entropi aşağıdaki şekildedir(??).

$$H(X) = - \sum_{i=1}^n p_i \log_2 p_i \quad (4)$$

Sırayla bir adil ve bir adil olmayan madeni paraların yazı tura olaylarının entropileri ise(??);

Olaydaki belirsizlik, eşit olasılıklı yazı turaya göre daha düşüktür. Yani

Table 5: Bir Adil Madeni Para
Örneği [11]

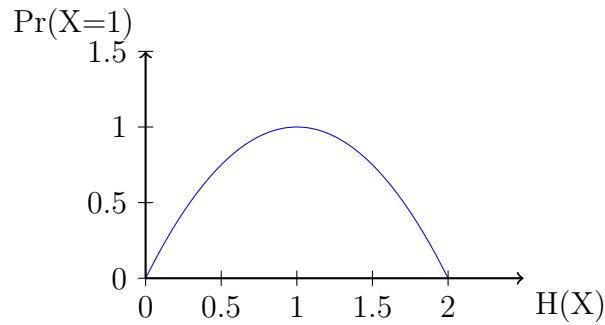
Çıktılar	Yazı	Tura
Olasılık	0.5	0.5
Bit Gösterimi	0	1
Bit Sayısı	1	
Bilgi: $H(p) = -\log_2(p(x))$	1	1
Entropi: $: p * -\log_2(p(x))$	0.5	0.5
Beklenen Entropi	1	

Table 6: Bir Adil Olmayan
Para Örneği [11]

Çıktılar	Yazı	Tura
Olasılık	0.25	0.75
Bit Gösterimi	0	1
Bit Sayısı	1	
Bilgi: $H(p) = -\log_2(p(x))$	2	0.415
Entropi: $: p * -\log_2(p(x))$	0.5	0.307
Beklenen Entropi	0.807	

entropisi daha düşüktür.

$P(X)$ dağılımı Bernoulli dağılımı olarak 2 ayrı x ve y olaylarının gerçekleştiği tanımlanırsa Bernoulli dağılımı altında $p(x) = 1|p(y)$ denilebilir. x olayının gerçekleşme olasılığı $p(x)$ e göre entropi fonksiyonunun grafiği ise aşağıdaki şekildedir(2);



İki olayın gerçekleşme olasılığı eşit olduğunda (yani $p(x)=0=1-p(y)$) dolayısıyla $p(y)=0.5$) entropinin maksimum değerini aldığı gözükmemektedir. İki olayın gerçekleşme olasılıkları birbirine eşit olduğundan hangi sonucun geleceği diğer durumlara göre daha belirsizdir ve bu yüzden entropi maksimum değeri alır.

3 Çapraz Düzensizlik (Cross-Entropy) ve KL İraksaklığı (KL Divergence)

Cross-Entropy, gerçek olasılık dağılımı P iken bulunan Q olasılık dağılımı için beklenen entropi değeridir. Cross-Entropy formülü (??);

$$H(p, q) = - \sum_x p(x) \log_q(x) \quad (5)$$

Cross-Entropy'yi Kullback-Leibler(KL) ıraksaklığını kullanarak da aşağıdaki şekilde gösterilebilmektedir;

$$H(p, q) = H(p) + D_w(p||q) \quad (6)$$

$H(p)$, doğru olasılık dağılımı P'nin entropisidir. P doğru olasılık dağılımını bulmak için bir model kurulduğunda P'ye yaklaşık olan Q olasılık dağılımını vermektedir. Bu 2 olasılık dağılımı arasındaki farklılığa bakmak için tanımlanan ölçülerden biri Cross-Entropy fonksiyonunda gördüğünüz $D_{kl}(p||q)$ ölçüsü Kullback-Leibler(KL) ıraksaklığıdır.

4 Kullback-Leibler Iraksaklığı (KL-Divergence)

Makine öğrenmesi alanında P olasılık dağılımı yerine Q olasılık dağılımı kullanılırken elde edilen bilgi kazanım anlamınındayken, Bayes çıkarımında ise önceki dağılım Q'dan sonraki dağılım P'ye geçerken ki elde edilen bilgi kazanımıdır. Yani P olasılık dağılımını tahmin ederken Q olasılık dağılımı kullanıldığında kaybedilen bilgi miktarıdır. Formülünün olabilirlik oranından elde edilmesi ise adım adım aşağıdaki şekildedir(??).

$$LR = \frac{p(x)}{q(x)} \quad (7)$$

x değeri bilinmeyen bir olasılık dağılımından ise yukarıda görülen (??) olabilirlik oranı x örnekleme için P dağılımından gelmenin Q dağılımından gelmeye nazaran ne kadar olası olduğunu göstermektedir. Birden fazla bağımsız örneklem için olabilirlik fonksiyonu hesaplanırken her örneklem için bulunan olabilirlik oranının çarpılması gerekmektedir.

$$LR = \prod_{i=0}^n \frac{p(x_i)}{q(x_i)} \quad (8)$$

$$LR = \sum_{i=0}^n \log\left(\frac{p(x_i)}{q(x_i)}\right) \quad (9)$$

Örnekleme Q dağılımı üzerinde P'ye dair ortalama ne kadar bilgi verdiğini hesaplamak için olabilirlik oranının beklenen değeri??;

$$D_k l(P||Q) = \sum_{i=0}^n p(x_i) \log\left(\frac{p(x_i)}{q(x_i)}\right) \quad (10)$$

KL ıraksaklığı formülünü yukarıdaki şekliyle elde etmiş olunmaktadır. KL ıraksaklığının 0 olması P ve Q dağılımlarının aynı olduğunu ifade etmektedir. Yukarıdaki formül P ve Q dağılımları ayırık olasılık dağılımları için kullanılabilir (Örneğin Bernoulli, Poisson, Binom dağılımları).

$$D_k l(P||Q) = \int_{-\infty}^{\infty} p(x_i) \log\left(\frac{p(x_i)}{q(x_i)}\right) dx \quad (11)$$

Olasılık dağılımları sürekliyse formül yukarıdaki gibi integral şeklinde tanımlanır.

5 Optimizer

Optimizasyon algoritmaları, modelin kayıp fonksiyonunu minimize ederek modelin daha iyi performans göstermesini sağlar. Bu algoritmalar, ağırlıkları güncellemek için gradyan iniş (gradient descent) ve türev tabanlı teknikler kullanır. Optimizatörler, sınır ağı eğitiminde önemli bir rol oynar. Ağırlık ve önyargılar gibi parametreleri ayarlayarak ağı bir fonksiyonu en iyi şekilde temsil etmesini sağlarlar [13].

1. Gradyan İniş (Gradient Descent): Gradyan İniş (Gradient Descent), belki de en temel optimizasyon algoritmasıdır ve derin öğrenme modellerinin eğitiminde sıkça kullanılır. Amacı, kayıp fonksiyonunun ağırlıklar üzerinden türevini alarak, negatif gradyana doğru ağırlıkları güncellemektir [14].

2. Stochastic Gradient Descent: Stokastik gradyan azalma, gradyan azalma gibidir ancak gradyan azalmada bütün veri seti kullanılırken, stokastik gradyan azalmada tek bir örneklem kullanılır. Stokastik gradyan azalma yönteminde tek seçilen veri ile işlem yapıldığı için global minimum noktasına kararlı bir şekilde ilerleme gözlenmez. Diğer taraftan, rastgele parametre katsayıları ile bir örnek üzerinde işlem yaptığından local minimaya düşme ihtimali daha düşüktür [14]. Stokastik gradyan azalma, gradyan azalmaya nazaran daha gürültülü bir grafik çizer. Minima noktasına ulaşması için daha fazla iterasyon sayısına ihtiyaç duymaktadır ancak tek bir örnek üzerinde çalıştığı için, hesaplama maliyeti gradyan azalmadan daha verimlidir.
3. Uyarlanabilir Gradyan Algoritması (Adagrad): Adagrad, parametre güncellemelerini her bir parametre için ayrı ayrı ölçeklendirerek gradyan iniş optimizatörünün bir modifikasyonudur . Minibatch Gradyan azalmada başarıyı etkileyen faktörlerden biride kullanılan adımın boyutuydu. Bu adım boyutunu doğru belirleyememek, bizi minimuma yakınsamama problemi ile karşı karşıya bırakabilirdi. AdaGrad burada devreye girmekte ve adım boyutunu otomatik olarak belirlemektedir [14]. Uyarlanabilir Gradyan Algoritması, gradyan tabanlı optimizasyon için bir algoritmadır. Öğrenme oranı, geçmiş gözlemleri hesaba dahil edilerek parametrelere bileşen olarak uyarlanır. Her parametrenin performansı artıran kendi öğrenme oranı vardır. NLP ve Görüntü tanıma konularında kullanımı uygundur.
4. Momentum: Momentum, Stokastik gradyan azalma ile kullanılan popüler bir tekniktir. Sadece mevcut durumda hesaplanan gradyan

değerini kullanmak yerine eski gradyanların değerlerini de kullanarak gideceği yönü belirlemeye çalışır. Salınım yaparak minimuma ulaşmaya çalışmaktadır. Çünkü bir batch için hatayı azaltacak olan parametre güncellemeleri verinin tamamında aynı etkiyi oluşturmayabilir [15].

5. RMSprop Optimizasyon (Root Mean Square Propagation): RMSprop optimizasyon, momentum optimizasyon ile benzerdir. RMSprop optimizasyon dikey olarak meydana gelen salınımı minimize eder. Dolayısıyla öğrenme oranını artırarak, yatay boyutta daha hızlı hareket edip minimuma daha hızlı ulaşma imkanı elde ederiz. RMSprop optimizasyon ve gradyan azalma arasındaki fark, matematiksel formülden kaynaklanmaktadır[15].
6. Adam Optimizasyonu (Adam Optimization): Adam, Adagrad ve RMSProp optimizatörlerinin bir kombinasyonudur[13]. Parametre güncellemelerini her bir parametre için ayrı ayrı ölçeklendirerek ve geçmiş gradyan bilgilerini kullanarak gradyan iniş optimizatörünün bir modifikasyonudur. Adam Optimizasyonu, gradyan iniş algoritmasının bir türevidir ve daha hızlı ve verimli bir şekilde çalışır. Bu algoritma, adaptif momentum ve adaptif öğrenme hızı kullanarak ağırlıkları günceller.

6 Aktivasyon Fonksiyonu

Yapay sinir ağlarına kompleks verileri öğretebilmemiz için aktivasyon fonksiyonları gereklidir. Aktivasyon fonksiyonlarının amacı weight(ağırlık) ve bias değerlerini ayarlamaktır.

”Weight” (ağırlık), yapay sinir ağlarında girdi verileriyle çarpılarak nöronlara iletilen değerlerdir. Bu ağırlıklar, ağırlık öğrenme sürecinde

güncellenir ve çeşitli veri desenleri arasında ilişkileri öğrenmesine yardımcı olur. Ağırlıklar, ağıın çıktılarını belirlemek için kullanılır ve doğru şekilde ayarlandığında, ağıın istenen sonuçları üretmesine katkıda bulunurlar.

”Bias” (yatay kayma), yapay sinir ağlarında her bir nöronun çıktısına eklenen sabit bir değerdir. Bu değer, nöronun aktivasyon fonksiyonunu değiştirerek ağıın esnekliğini artırır. Bias, ağıın veri setindeki karmaşıklıkları öğrenmesine ve girdi verileriyle daha iyi uyum sağlamasına yardımcı olur. Ayrıca, bias terimi, nöronların doğrusal olmayan ilişkileri öğrenmesine olanak tanır ve ağıın genelleme yeteneğini artırabilir.

1. Sigmoid: Sigmoid, S şeklinde bir eğriye sahip bir aktivasyon fonksiyonudur. Giriş değerlerini 0 ile 1 arasında bir aralığa dönüştürür [16].
2. ReLU: ReLU (Rectified Linear Unit), derin öğrenme modellerinde kullanılan doğrusal olmayan bir aktivasyon fonksiyonudur. Girdi değeri pozitif ise, giriş değerini olduğu gibi çıktı olarak verir, aksi takdirde 0 değerini çıktı olarak verir. ReLU, vanishing gradient problemini çözmek ve modelin eğitimini hızlandırmak için yaygın olarak kullanılır [17].
3. Tanh: Hiperbolik tanjant fonksiyonudur. Sigmoid’e benzer, ancak -1 ve 1 arasında değerler üretir.
4. Softmax: Çoklu sınıflandırma problemleri için kullanılır.

7 Sound datasets

- N-synth
- MAESTRO

7.1 N-synth

Benzersiz perde ve tınıya sahip 305.979 müzik notası içeren, tek seferlik enstrümantal notalardan oluşan bir veri kümesidir. Sesler, ticari örnek kütüphanelerindeki 1006 enstrümandan toplanmıştır ve kaynaklarına (akustik, elektronik veya sentetik), enstrüman ailesine ve sonik niteliklerine göre açıklanmıştır. Açıklamada kullanılan enstrüman aileleri bas, piring, flüt, gitar, klavye, tokmak, org, kamış, yaylı, synth lead ve vokaldir. Enstrümanlar için dört saniyelik monofonik 16kHz ses parçacıkları (notalar) oluşturulmuştur [6].

7.2 MAESTRO

MAESTRO (MIDI and Audio Edited for Synchronous Tracks and Organization), nota etiketleri ve ses dalga formları arasında ince bir hizalama (3 ms) ile yakalanan 200 saatin üzerinde virtüöz piyano performansından oluşan bir veri kümesidir [7]. MIDI, dijital enstrümanların birbirleriyle 'mesajlar' aracılığıyla iletişim kurmasını sağlayan bir protokoldür. Bu mesajlar, çalma için kullanılacak enstrümanın türü, çalınacak notalar, bir notanın ne zaman başlayacağı, bir notanın ne zaman biteceği vb. hakkında bilgi depolar.

8 Ses Türleri

- Konuşma (Metinden Sese)
- Müzik
- Müzik notaları (örnekler)
- Ses tasarımı

9 Ses Temsil Etme Yöntemi

- Raw-audio
- Spectrogramlar

10 Raw-audio

Bu ayar, standart .mp4 ses parçasına ek olarak videonuz için ayrı bir .wav dosyası oluşturur. Bu ayar, ayrı bir .wav dosyasının gelişmiş bir düzenleme programında paylaşmasını veya kullanmasını istiyorsanız kullanışlıdır.

RAW ses parçasına uygulanacak işlem seviyesini seçebilirsiniz: Düşük:

Minimum işleme uygular; post prodüksiyonda ses işleme uygularsanız

idealdir. Orta: Manuel Ses Kontrolü ayarınıza (rüzgar ve / veya stereo)

göre işleme uygular. Manuel Ses Kontrolünüz kapalıysa, kamera otomatik

olarak rüzgar filtreleme ve stereo ses arasında geçiş yapar. Yüksek: Tam ses

işleme (otomatik kazanç ve AAC kodlaması) uygular.

11 Spektrogram

Spektrogram, belirli bir dalga formunda bulunan çeşitli frekanslarda zaman içinde bir sinyalin sinyal gücünü veya "ses yüksekliğini" temsil etmenin görsel bir yoludur. Örneğin 2 Hz ile 10 Hz arasında daha fazla veya daha az enerji olup olmadığı görülebildiği gibi, enerji seviyelerinin zaman içinde nasıl değiştiği de görülebilir. Diğer bilim dallarında spektrogramlar genellikle insanlar, makineler, hayvanlar, balinalar, jetler vb. tarafından üretilen ve mikrofonlar tarafından kaydedilen ses dalgalarının frekanslarını görüntülemek için kullanılır. Sismik dünyada, spektrogramlar, farklı deprem türlerini veya yeryüzündeki diğer titreşimleri ayırt etmeye ve karakterize etmeye yardımcı olmak için bireysel veya sismometre grupları tarafından kaydedilen sürekli sinyallerin frekans içeriğine bakmak için giderek daha fazla kullanılmaktadır. [8]

12 Spektrum

Spektrum, elektromanyetik radyasyon dalgalarının sürekli bir aralığıdır. En uzun radyo dalgalarından en kısa X-ışınlarına ve gama ışınlarına kadar uzanır. Radyofrekans spektrumu, spektrumun alt kısmında yer alır [9].

13 Mel-Frekans Cepstral Katsayıları (MFCC-Mel-Frequency Cepstral Coefficients)

MFCC, ses sinyallerinin öznitelik çıkarma sürecinde kullanılan bir yöntemdir. Özellikle konuşma tanıma ve ses işleme alanlarında sıkça kullanılır.

MFCC'nin kullanılmasının nedenleri arasında insan işitme sisteminin özelliklerini modellemeye uygunluğu, ses sinyallerinin temsilinin kompakt olması, ve gürültüye karşı dirençli olması gibi özellikler bulunur. Bu özellikler sayesinde, MFCC ses sinyallerinin analizi ve tanımlanması için etkili bir şekilde kullanılabilir. Konuşma tanıma bir denetimli öğrenme görevidir, belirli konuşma örnekleriyle eğitilerek konuşma seslerini tanıyabilen sistemler geliştirilir [10].

14 Üretimde Inputlar

- Conditioning, belirli koşullara bağlı olarak bir sistem veya modelin davranışının değiştirilmesidir.
- Autonomous, bir şeyin dış yardım veya kontrol olmaksızın kendi kendine hareket etme yeteneği veya özelliğidir.
- Continuation, sürecin veya durumun devam etmesini ifade eder.

Kaynakça

- [1] Author/OpenAI, “Müzik kutusu - jukebox (openai).”
<https://openai.com/research/jukebox/>, Year Published/ 02.2024.
Accessed: 15.03.2024.
- [2] A. Gezer, “Vae.” <https://www.yapayzekatr.com/2023/11/14/variational-autoencoders-vaes-nedir/>, Year Published/
14.11.2023. Accessed: 15.03.2024.
- [3] A. Kumar, “A crash course on vaes, vq-vaes, and vae-gans.”
<https://ammesatyajit.medium.com/a-crash-course-on-vaes-vq-vaes-and-vae-gans-3fdcc40b059e>,
Year Published/ 29.01.2020. Accessed: 17.03.2024.
- [4] A. Yadav, “Understanding vector quantized variational autoencoders (vq-vae).” <https://shashank7-iitd.medium.com/understanding-vector-quantized-variational-autoencoders-vq-vae-323d710a888a>,
Year Published/ 01.09.2019. Accessed: 17.03.2024.
- [5] A. Mishra, “An introduction to vae-gans.”
<https://wandb.ai/shambhavicodes/vae-gan/reports/An-Introduction-to-VAE-GANs--VmlldzoxMTcxMjM5>, Year Published/
08.12.2021. Accessed: 17.03.2024.
- [6] Engel, “Nsynth.” <https://paperswithcode.com/dataset/nsynth/>,
01.04.2017. 25.03.2024.
- [7] J. Vial, “The maestro dataset v2.” <https://www.kaggle.com/datasets/jackvial/themaestrodatasetv2/>,
01.01.2018. 25.03.2024.

- [8] PNSN, “What is a spectrogram?.”
<https://pnsn.org/spectrograms/what-is-a-spectrogram/>,
01.01.2020. 28.03.2024.
- [9] ACMA, “What is spectrum?.”
<https://www.acma.gov.au/what-spectrum#:~:text=The%20spectrum%20is%20a%20continuous,Frequency%20bands/>,
17.10.2022. 28.03.2024.
- [10] U. Kiran, “Mfcc technique for speech recognition.”
https://community.gopro.com/s/article/What-is-RAW-Format?language=en_US/, 14.08.2023. 28.03.2024.
- [11] Author/Emre Yüksel, “Entropi&KL Divergence.”
<https://medium.com/kaveai/d%C3%BCzensizlik-entropy-%C3%A7apraz-d%C3%BCzensizlik-cross-entropy-ve-kl-iraksakl%C4%B1%C4%9F%C4%B1-kl-divergence-89d26735789f/>, Year Published/
02.2024. Accessed: 15.03.2024.
- [12] Author/Unreal Engine, “UE BP.”
https://dev.epicgames.com/documentation/en-us/unreal-engine/making-macros-in-unreal-engine?application_version=5.0/,
Year Published/ 02.2024. Accessed: 02.03.2024.
- [13] A. O. YILMAZ, “Derin Öğrenmede kayıp fonksiyonları ve optimizasyon algoritmaları.” <https://aoyilmaz.medium.com/derin-%C3%B6%C4%9Frenmede-kay%C4%B1p-fonksiyonlar%C4%B1-ve-optimizasyon-algoritmalar%C4%B1-8fc754e7a639/>, Year Published/ 27.07.2023. Accessed: 25.03.2024.

- [14] Author/wikipedia, “Stochastic gradient descent.” https://en.wikipedia.org/wiki/Stochastic_gradient_descent#AdaGrad/. Accessed: 25.03.2024.
- [15] A. ATCILI, “Yapay sinir aglarinda kullanilan optimizasyon algoritmaları.” <https://medium.com/machine-learning-t%C3%BCrkiye/yapay-sinir-a%C4%9Flar%C4%B1nda-kullan%C4%B1lan-optimizasyon-algoritmalar%C4%B1-3e87cd738cb5/>, Year Published/ 06.11.2020. Accessed: 25.03.2024.
- [16] Author/wikipedia, “Sigmoid function.” https://en.wikipedia.org/wiki/Sigmoid_function/. Accessed: 25.03.2024.
- [17] Author/wikipedia, “Rectifier (neural networks).” [https://en.wikipedia.org/wiki/Rectifier_\(neural_networks\)/](https://en.wikipedia.org/wiki/Rectifier_(neural_networks)). Accessed: 25.03.2024.