

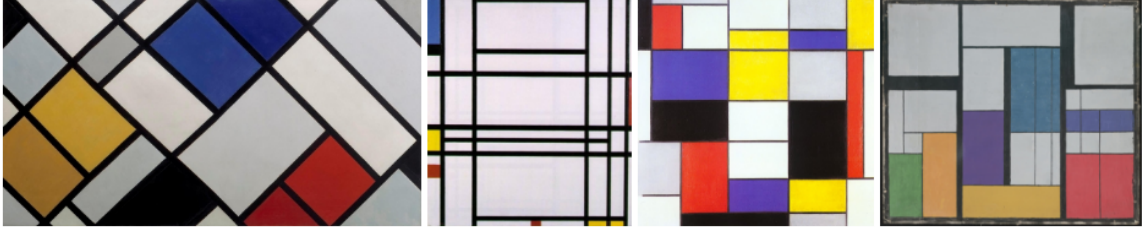
1.1 Amaç-Hedef

Bu ödevde; 4 farklı sanat türünde resmedilmiş eserleri tanıyan geri-beslemeli sinir ağı kodlanacaktır. Ödevdeki problemin; geri-beslemeli ağı, aynı ağı **keras** kullanılarak kodlanması ve (opsiyonel olarak) konvolüsyonel sinir ağı kullanılarak çözülmesi beklenmektedir.

1.2 Veriler

Kullanılacak veriler ödev ile birlikte verilmektedir. Eserler, şu 4 farklı sanat türünde resmedilmiştir:

- Modern Sanat



Şekil 1: Modern sanat resim örnekleri

- Peyzaj (Manzara) Sanatı



Şekil 2: Manzara resim örnekleri

- Dışavurum Sanatı



Şekil 3: Dışavurum sanatı resim örnekleri

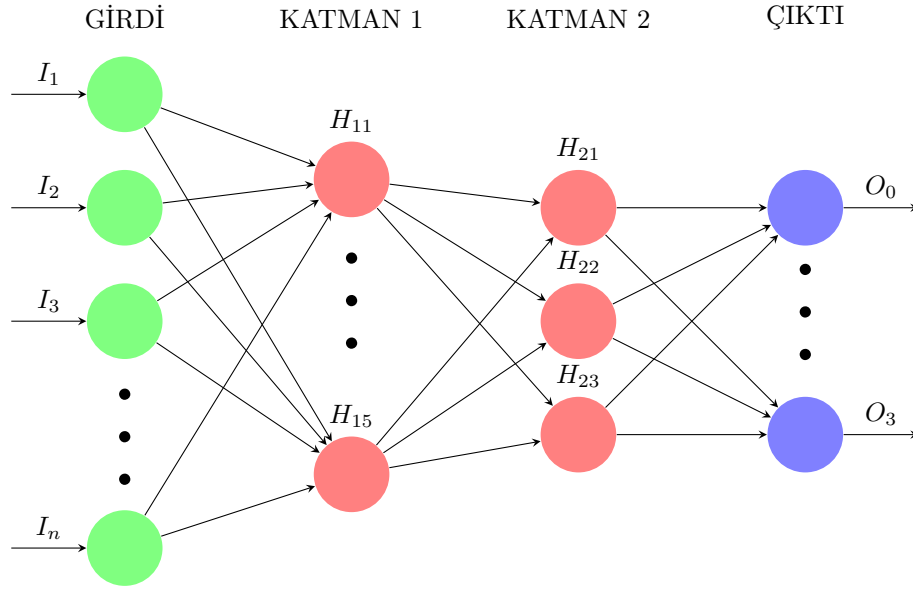
- Renkli Alan Boyamaları



Şekil 4: Renkli alan boyama örnekleri

Tüm resimler bir dosya yapısı içerisinde "eğitim" ve "test" verileri olarak ikiye ayrılmıştır. Resimlerin boyutları sabit değildir. Bu resimlerden, çeşitli resim işleme yöntemleri (veya farklı teknikler) kullanarak *en az 3* özellik elde etmelisiniz. Özellikleri elde ederken hazır fonksiyonlar kullanabilirsiniz; ancak hazır fonksiyonlar kullanacağınız tekniğin parçası olmalıdır, tamamı olmamalıdır.

1.3 Sinir Ağı'nın Ayarlanması



Şekil 5: Örnek Sinir Ağı Yapısı

Sinir ağında sizin belirleyeceğiniz tekniğe göre en az 3 giriş ve bir adet bias olmalıdır. Sizin için tanımlanan sinir ağı'nın gizli katmanlarında kaç adet nöron olduğu bilgisi size Tablo-1'de verilmiştir. Örneğin 5, 3 yazılıysa ayarlamanız gereken sinir ağı'nın yapısı Şekil-5'deki gibi olacaktır (Not: Rahat anlaşılması, kodlamayı rahat planlamak için ilk katman dışında ara katmanlar için bias kullanmayabilirsiniz).

- Şekil-5'deki ağ yapısı üzerinden düşünülürse, katman 1'e gelen ağırlıklar $n \times 5$ 'lik bir matris ile, katman 2'ye gelen ağırlıklar ise 5×3 'lük bir matrisle, çıkışa gelen ağırlıklar ise 3×4 'lük bir matrisle ifade edilebilir.
- Ağdaki tüm ağırlıklar rastgele olarak ve $[0 - 1]$ aralığında atanmalıdır. Bunun için *numpy* kütüphanesindeki *random* fonksiyonlarını kullanabilirsiniz.
- Aktivasyon fonksiyonu olarak *sigmoid* fonksiyonunu kullanmalısınız.

Çıktı katmanında 4 adet çıkış bulunmasını şu şekilde yorumlamalısınız. Hangi çıkıştaki değer en büyükse sınıflandırmak istediğiniz veri, o sınıfa aittir. Modern sanat 0, manzara sanatı 1, dışavurum sanatı 2, renkli alan boyamaları 3 olarak numaralandırıldığı varsayılırsa; çıktıdaki 4 sayı içerisinde en büyük değer 2 numaralı çıkışta ise girdideki resim dışavurum sanatı ile resmedilmiş demektir. Bu durumda eğitim işlemi yaparken **target** olarak belirteceğimiz sayı aslında vektör olmalıdır. Bunun için aşağıdaki kodu kullanabilirsiniz:

```
1 from sklearn.preprocessing import LabelBinarizer
2 label_binarizer = LabelBinarizer()
3 label_binarizer.fit(target)
4 target_output = label_binarizer.transform(target)
```

Bu kod **target_output** matrisi içerisine her bir veri için çıkış bilgisini oluşturur. Örneğin matrisin herhangi bir satırında $[0, 0, 1, 0]$ yazılıysa bu satırda temsil edilen verinin dışavurum sanatı ile resmedilmiş bir eser olduğu anlaşılmaktadır.

Test resimleri ile işlem yaptıktan sonra çıktıda hangi çıkıştan en büyük sayı geliyorsa cevabın o olduğundan bahsedilmiştir. Bu işlemi aşağıdaki kod ile hızlı bir şekilde yapabilirsiniz. **my_result** yazılı olan yere verdiğiniz matrisin her bir satırındaki en büyük sayısının indisini seçip size cevap vektörü döndürecektir.

```
1 result = label_binarizer.inverse_transform(my_result)
```

1.4 Ödevde istenilenler

1.4.1 Ödevin 1. Bölümü

- Resimlerden *en az 3* adet özellik çıkarmalısınız.

- Tüm resimler için elde edilen özellikleri kullanarak eğitim ve test veri matrislerini oluşturmalsınız.
- Size verilen ağı yapısına uygun bir şekilde ağırlıklara rasgele değerler atamalsınız.
- Çıktıda 4 adet çıkış bulunduğundan, **target** vektörünü yukarıda verilen kod ile matrise çevirmelisiniz (**target_output**).
- Ağıınızı geri-besleme yöntemi (backpropagation) ile eğitiniz.
- Epoch sayısını siz belirleyebilirsiniz.
- Eğittiğiniz ağırlıkları kullanarak, test verileri için sonuçları alınız. Sonuçlar 4 çıkışlı olduğundan hangi sınıfa karşılık geldiğini anlamak için yine yukarıda verilen tek satırlık kod parçasını kullanabilirsiniz.
- Elde ettiğiniz sonuçlar ile gerçek sonuçları kullanarak **sklearn** kütüphanesinde bulunan **accuracy_score** fonksiyonunu kullanarak test başarısını ekrana yazdırmalsınız.

1.4.2 Ödevin 2. Bölümü

- Size tanımlanan ağı, **keras** (arka planda **tensorflow** olmalıdır) kullanarak kodlamalsınız.
- Kodladığınız modeli; 1. bölümde bahsedilen kurallara dikkat ederek, elde ettiğiniz özellikleri girdi olarak verip eğitim ve test işlemlerini yapmalısınız.
- Epoch sayısını 5'den 50'ye kadar beşer beşer arttırarak eğittiğiniz modelin test verisi üzerindeki başarı grafiğini ekrana çizdiriniz. (**batch_size**, 1 olmalıdır)
- En yüksek başarıyı veren epoch sayısı T olsun. T epoch kullanarak **batch_size** parametresini 1'den 4'e kadar birer birer arttırıp, modelin test verisi üzerindeki başarı grafiğini ekrana çizdiriniz. (Test verileri için **batch_size**, her zaman 1 olmalıdır).

1.4.3 Ödevin 3. Bölümü (Bonus)

- Bu bölüm ödevin ek puan alabileceğiniz kısmıdır. Bu kısım tam çalıştığı durumda puan alabilirsiniz. Eksik veya kısmi doğru olmasında puan verilmeyecektir.
- En fazla iki adet **Conv2D** fonksiyonu içeren, arka planda **tensorflow** kullanan CNN modelini **keras** kullanarak tasarlamalsınız.
- CNN modelinde özellik çıkarma işlemini modelin kendisinin yapacağını unutmayınız. Dolayısıyla önceki kısımlarda çıkardığınız özellikler burada geçerli olmayacaktır.
- Modelin doğru çalışması için tüm resimleri sabit bir boyut getirmelisiniz.
- Epoch sayısını en fazla 10 yapınız. Kernel boyutlarını sizin belirlemeniz gerekmektedir.
- Belirtilen kısıtlar dahilinde elde edilen test başarının yüksek olması bu bölümden alacağınız puanı arttıracaktır. Kısıtların dışına çıkılmaması gerekmektedir. Başarınızı arttırmak için **dropout**, **batch normalization** kullanabilir, parametrelerle oynayabilirsiniz.
- Oluşturduğunuz modelde tüm aktivasyon fonksiyonları **sigmoid** olmalıdır. Buna göre modeli eğitip, eğitim süresini ekrana yazdırmalsınız. Aynı modeli tüm aktivasyon fonksiyonları **relu**, **tanh**, **prelu** olacak şekilde değiştirerek eğitim sürelerini ekrana yazdırmalsınız.

1.4.4 Genel Açıklamalar

- Tüm işlemler (bonus kısım da dahil) tek bir kod içerisinde olmalıdır. Diğer bir deyişle kod çalıştırıldığında yukarıda istenilen herşey çıktıda yer almalıdır.
- Resimlerin olduğu dosya yapısını değiştirmeyiniz. Yazacağınız kod size verilen klasörle aynı dizinde bulunduğunda, resimlere erişebilmelidir. Kodunuzu "artpictures" klasörünün içine koymayınız.
- Kodunuzun içerisinde açıklamalar yapmayı unutmayınız. Yorumlarınızı ve açıklamalarınızı kodun içerisinde ikna edici seviyede ifade ediniz.
- **numpy**, **sklearn**, **skimage**, **os**, **glob**, **matplotlib**, **keras**, **tensorflow**, **cv2** kütüphaneleri dışında farklı kütüphaneler kullanmayınız.

1.5 Ödev Teslimi

Ödevin son teslim tarihi **4 Mayıs Cuma 2018 23:59**'dur. Ödevi **ce** uzantılı e-mail adresine gönderiniz. E-mailin konusuna **YSA 2. Ödev** yazınız. Sadece **.py** uzantılı Python kodunuzu gönderiniz. Zip halinde *göndermeyiniz*. Kodunuza isim ve soyisminizi yazınız. Örnek: *aliveli.py*

Not: Öncelikli olarak kodun doğru çalışmasına ve bulduğunuz karakteristik özelliklere not verilecektir. Doğruluk oranının yüksek veya az olması önemli değildir; ancak doğruluk oranı yüksek olan ve ödevinde bütünlük olanlara fazladan puan verilebilir. (Bu kurallar bonus kısmı için geçerli değildir)

Tablo 1: Öğrenci Bazlı Girdi-Çıktı Katmanı Hariç Oluşturulacak Her Katmandaki Nöron Sayısı

#	Adı	Soyadı	Ağın Yapısı
1	İLYAS	ORTABAKAN	4, 5, 6
2	İLHAN	SUNYAR	4, 4, 6
3	CİHAN	SUVARİ	6, 4, 5
4	ÖMER	DEMİR	7, 4, 6
5	TUNAY	KOÇ	5, 5, 6
6	YUNUS	YILDIRIM	7, 5, 3
7	MURAT	ER	4, 6, 3
8	DOĞAÇ CEM	İŞOĞLU	6, 4, 3
9	MERVE NUR	USLU	5, 7, 5
10	MURAD	KIZILAY	4, 5, 5
11	ONUR	KILIÇ	6, 6, 5
12	MELİSA	SÖKMEN	6, 7, 5
13	ENES	BAŞAR	5, 4, 6
14	ERHAN	AYAKKABICIOĞLU	7, 5, 3
15	MÜCAHİT	KARAKAYA	7, 7, 5
16	ŞEYMA NUR	KOLAK	4, 7, 5
17	KASIM	SAĞIR	6, 4, 6
18	ENES	BİLGİN	4, 6, 6
19	FATİH	SARHAN	5, 7, 6
20	ÇAĞRI	GİDER	7, 4, 6
21	EGEHAN	GÜNDÜZ	7, 6, 5
22	UMUT	GÜNERİ	6, 6, 3
23	EMRE	KUL	7, 6, 6
24	ŞERİFE BENGİSU	KARAKILINÇ	4, 4, 3
25	BERKAY	ÖZTÜRK	5, 7, 5
26	BURAK	KARAKAYA	7, 6, 4
27	ÖZGÜR	KURU	5, 6, 4
28	ELGUN	GULİYEV	7, 4, 3
29	HASAN	SANLI	6, 7, 5
30	METİN MERT	AKÇAY	7, 4, 5
31	CEREN	YAŞAR	6, 4, 4
32	AKİF BERKAY	GÜRCAN	6, 6, 5
33	BURAK	ÖZTÜRK	7, 4, 5
34	MURATCAN	KILIÇ	6, 5, 5
35	SUNA	AKMELEZ	7, 5, 6
36	MUSTAFA	YOĞURTÇU	5, 5, 3
37	HİLMİ BİLAL	ÇAM	6, 7, 3
38	İZZETTİN	ERDEM	7, 6, 5
39	ELİF CANSU	YILDIZ	5, 4, 6
40	SONER	PAYCI	4, 5, 4
41	MUHAMMED MELİH	DURMAZ	5, 6, 7
42	OZAN	TEPE	4, 5, 7
43	EZGİ EKİN	ERGÜN	7, 7, 6
44	FURKAN	ÇELİK	7, 4, 6
45	JASSER	DHAOUADI	7, 5, 6
46	ÇETİN	KILIÇ	5, 4, 6
47	SARDOR	HAZRATOV	7, 6, 5
48	İLHAN	ÇELİK	5, 3, 6
49	CANER	EMEÇ	6, 3, 5
50	HALİL İBRAHİM	BESTİL	7, 3, 4
51	SÜLEYMAN EMRE	DEMİR	3, 5, 6
52	GİZEM	ÇAVUŞ	3, 6, 5
53	MÜRÜVVET	HASANBAŞOĞLU	3, 7, 5
54	OZAN TÜRKER	DEMİR	3, 4, 5
55	OĞUZHAN	ÖZBEK	4, 5, 7
56	MUHAMMED YASİN	SAĞLAM	5, 5, 7
57	HÜSEYİN MURAT	GEZER	6, 6, 7
58	MİRAY ASENA	AYGÜZEL	7, 4, 6
59	CANAN	SAYAR	5, 7, 7

60	EMRE	EĞRİBOZ	7, 4, 6
61	ERAY	TOPCU	5, 6, 4
62	EIMANTAS	MOCKEVICIUS	4, 3, 5