

150101002 - Berat Göktuğ ÖZDEMİR

150101045 - Uğur İPEKDÜZEN

# YAPAY SİNİR AĞLARI

## Giriş

Bu proje, Yapay Sinir Ağları ve Multilayer Perceptron fonksiyonu ile Mısır'daki Hepatit-C hastalarından toplanmış veriler göz önünde bulundurularak, veri seti içerisindeki hastaların Hepatit-C hastalığının durumuna göre F0-F4 aralığında sınıflandırmaktadır.

Bu aralıkta F0 sağlıklı bireyi temsil ederken, F4 siroz olmuş hastayı göstermektedir. Kullanılan veri setinde sağlıklı (F0) birey bulunmamaktadır.

Bu veriler baş ağrısı, mide bulantısı, kusma gibi fiziksel belirtilerin yanında kan tahlillerini de içermekte olup cinsiyet, yaş ve vücut kitle indeksi gibi sınıflandırılmış toplam 28 giriş içermektedir.

Python dilinde, bazı hazır kütüphaneler kullanılarak hazırlanmış bu proje de fazla giriş olması, yetersiz hasta verisi ve ilişkisi düşük girişler nedeniyle makine yeterli öğrenme düzeyine ulaşamadığı görülmüştür. Sonuç alanında bulunan Korelasyon Matrisi ile girişlerin birbirleri ile bağlantısının az olduğu gösterilmiştir.

Ortalama %25 oranında doğru tahminde bulunabilen bu proje, epoch değerine, gizli katman ile nöron sayısı değişikliğine ve aktivasyon fonksiyonlarına bağlı olarak %22-30 aralığında doğru tahminlerde bulunduğu görülmüştür.

## Tanım

Bu projenin amacı Mısır'daki Hepatit-C hastalarından toplanan veriler ile bilgileri girilen hastanın, hastalığının geldiği seviyeleri sınıflandırarak tahmin etmektir.

### \* Girişler:

- |   |   |
|---|---|
| 1. Age (Yaş)                              | 15. Platelet (Trombosit)                      |
| 2. Gender (Cinsiyet)                      | 16. AST 1 (Aspartat Aminotransferaz 1. Hafta) |
| 3. BMI (Vücut Kitle İndeksi)              | 17. ALT 1 (Alanin Aminotransferaz 1. Hafta)   |
| 4. Fever (Ateş)                           | 18. ALT 4 (Alanin Aminotransferaz 4. Hafta)   |
| 5. Nausea/Vomiting (Mide bulantısı/Kusma) | 19. ALT 12 (Alanin Aminotransferaz 12. Hafta) |
| 6. Headache (Baş ağrısı)                  | 20. ALT 24 (Alanin Aminotransferaz 24. Hafta) |
| 7. Diarrhea (İshal)                       | 21. ALT 36 (Alanin Aminotransferaz 36. Hafta) |
| 8. Fatigue (Yorgunluk)                    | 22. ALT 48 (Alanin Aminotransferaz 48. Hafta) |
| 9. Bone ache (Kemik ağrısı)               | 23. RNA Base                                  |
| 10. Jaundice (Sarılık)                    | 24. RNA 4 (RNA 4. Hafta)                      |
| 11. Epigastria pain (Karın ağrısı)        | 25. RNA 12 (RNA 12. Hafta)                    |
| 12. WBC (Beyaz Kan Hücresi)               | 26. RNA EOT (RNA Tedavi Sonu)                 |
| 13. RBC (Kırmızı Kan Hücresi)             | 27. RNA EF (RNA Uzama Faktörü)                |
| 14. HGB (Hemoglobin)                      | 28. BHG (Temel Histolojik Notlandırma)        |

### \* Çıkışlar:

29. Baseline Histological Staging (
- F0: Tümör yok
  - F1: Geçiş Fibröz
  - F2: Az Septum
  - F3: Çok Septum
  - F4: Siroz

Fibröz Doku: Bağ Doku

Karaciğer dokusunun (hücrelerinin) alkol ya da yağlanmasıyla ilgili olarak hasarı yüzünden fibrozis gelişebilir. Bu da ilerleyince siroz ve karaciğer yetmezliğine neden olur.

1385 örneğin kullanımına göre dağılımı şu şekildedir:

- ★%70 Training
- ★%15 Validation
- ★%15 Testing

Veri setinin alındığı adres: <https://archive.ics.uci.edu/ml/datasets/>

### \* Projenin Faydası:

Hastalık belirtileri ve kan sonuçları gibi bilgiler işlenerek ve makineye öğretilerek, hastanın Hepatit C'nin hangi evresinde olduğuna dair tahminler yapılarak tedavinin kolaylaştırılmasına yardımcı olmaktadır.

## Metot

Python ile PyCharm üzerinde Keras kullanarak projeyi geliřtirdik. Bařta veriler iin Numpy kütüphanesini kullanırken, daha sonra Feature Extruction(Özellik ıkarma) iřleminde Pandas kütüphanesi kolaylık sağladığı iin Numpy yerine Pandas'ı tercih ettik.

### ● Kullanılan Kütüphaneler:

```
from keras.models import Sequential
from keras.layers import Dense
from keras.optimizers import *
from keras.metrics import *
from keras.utils.np_utils import to_categorical
from sklearn import model_selection
from sklearn.preprocessing import LabelEncoder
from sklearn.multiclass import OneVsRestClassifier
from sklearn.svm import LinearSVC
from sklearn.utils import shuffle
from sklearn.metrics import confusion_matrix, auc, roc_curve
import numpy as np
from pandas import DataFrame
import matplotlib.pyplot as plt
import seaborn as sns
import itertools
from termcolor import cprint
```

### ● Kullanılan Algoritma:

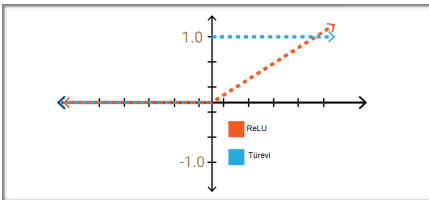
Keras, varsayılan olarak Backpropagation algoritmasını kullanır.

*Backpropagation, Çok Katmanlı Perceptron (Yapay Sinir Ağları) eğitimi iin denetimli bir öğrenme algoritmasıdır.*

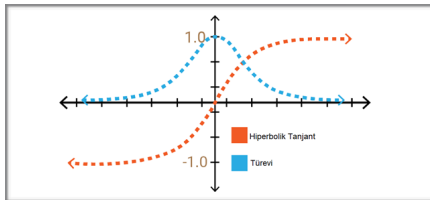
*Backpropagation algoritması, delta kuralı veya gradyan iniř olarak adlandırılan bir teknik kullanarak ağırlık alanındaki hata fonksiyonunun minimum deęerini arar. Hata fonksiyonunu en aza indiren ağırlıkların öğrenme problemine bir çözüm olduęu düşünölmektedir.*

### ● Kullanılan Aktivasyon Fonksiyonu:

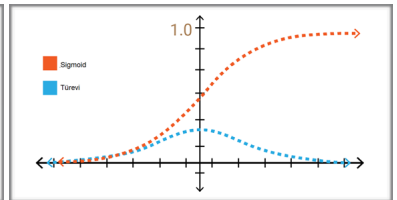
ReLu



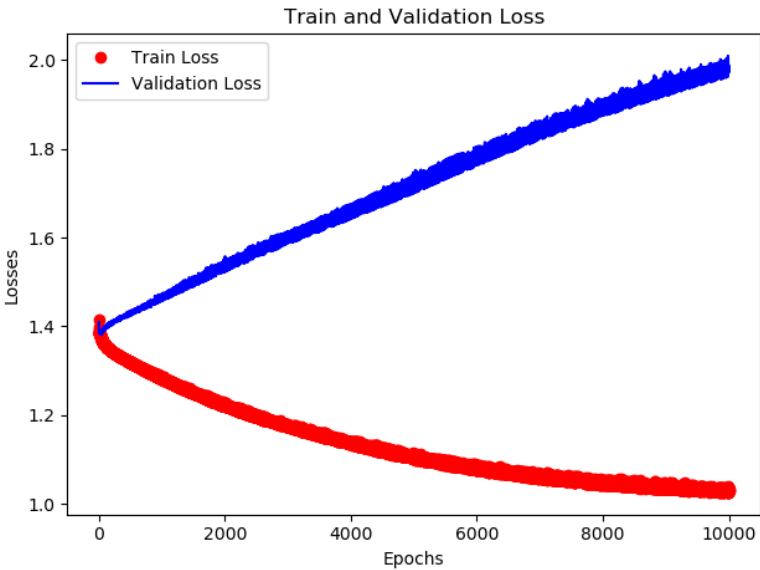
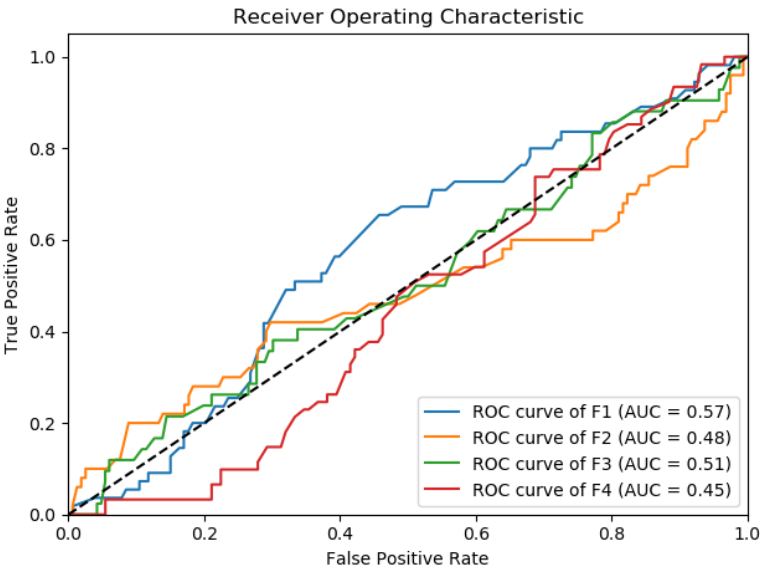
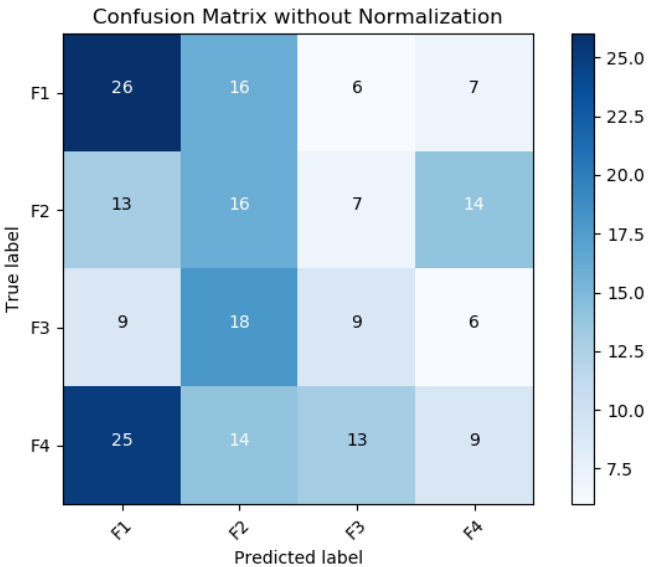
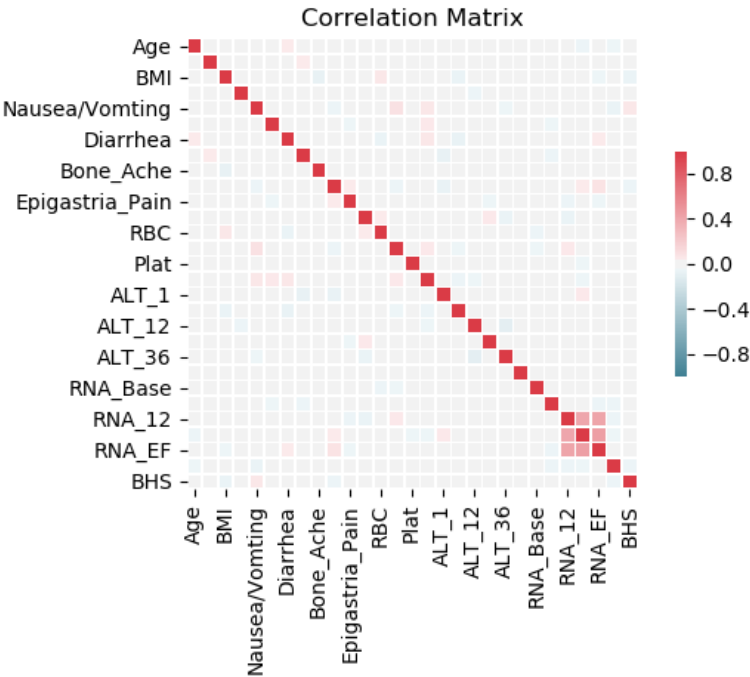
Hyperbolic Tangent



Sigmoid



Sonuç



### ✓ Gizli Katmandaki Değişiklikler ve Sonuçları

Gizli katmanların sayısını azalttıkça, accuracy'nin azalması ve loss değerinin artmasının yanında programın tahmin yeteneği de zayıfladı. Genel olarak hep aynı tahminlerde bulundu. Fakat gizli katman miktarını arttırdıkça program farklı tahminlerde de bulunmaya başladı.

Gizli katmanlardaki nöron sayısında yapılan değişikliklerde ise nöron sayısı azaldıkça accuracy oranı azaldı. Loss oranı aşırı etkilenmemekle beraber nöron sayısı azaldıkça az miktarda arttı.

Bunların dışında nöron ve gizli katman sayısı azaldıkça programın çalışma hızı artmaktadır.

Gizli katmanda yapılan bütün değişiklikleri ve sonuçlarını raporun sonunda görebilirsiniz.

### ✓ Öğrenme Oranındaki Değişiklikler ve Sonuçları

Denenen öğrenme oranları: 0.01, 0.005, 0.001, 0.0001

En İyi Sonuç: 0.01 - Sonuçlar her seferinde farklı olsa da en yüksek başarı yüzdesi ve accuracy bu öğrenme oranında görülmüştür. 0.6074 accuracy, %28 başarı yüzdesi.

En Stabil Sonuç: 0.0001 - Yapılan denemelerde loss, accuracy ve başarı yüzdesi hemen hemen sabit kaldı.

### ✓ Momentum Katsayısındaki Değişiklikler ve Sonuçları

SGD Optimizer'ı genel olarak kötü sonuçlar verdiği için kendi içinde bir kıyaslama yapacak olursak; Denenen momentum katsayıları: 0.90, 0.70, 0.50, 0.15, 0.05

En İyi Sonuç: 0.50 momentum katsayısı diğerlerine göre nispeten yüksek başarı yüzdesi vermesinin yanında diğer SGD sonuçlarına oranla stabil sonuçlar da verdi.

En Stabil Sonuç: 0.05 momentum katsayısı ile sağlandı. loss, accuracy ve başarı yüzdesi büyük oranda değişmedi.

*Öğrenme hızı ve momentum ile ilgili bazı özellikler aşağıdaki gibidir;*

- Öğrenme hızı yüksek tutuyor olmam veriden çok etkileniyorum demektir.
- Öğrenme hızı yüksek olması salınım neden olacaktır. Buna karşı küçük olması da küçük adımlarla ilerleyeceğinden öğrenimin çok uzun sürmesine neden olacaktır.
- Öğrenme hızı için en uygun çözüm başlangıçta öğrenme hızı yüksek tutmak, gittikçe azaltmaktır.
- Öğrenme hızı başlangıçta çok küçük olması, lokal optimum değere takılarak, global optimum değere hiç ulaşamamasına neden olabilir.
- Öğrenme hızı değeri genelde varsayılan değer olarak 0.01 kullanılmakta belli bir epoch'dan sonra 0.001'e düşürülmektedir.
- Stochastic gradient descent(SGD) momentum ile kullanıldığında hızı artırmakta, salınımları azaltmaktadır.
- Genelde kullanılan momentum beta katsayısı 0.9'dur. Uygun parametre aralığı ise 0.8-0.99'dur.

# RHO

28

100 - ReLu

100 - Tanh

100 - Tanh

100 - Tanh

4 - Sigmoid

**Learning Rate:** 0.01 | **Rho:** 0.90 | **Batch Size:** 1024 | **Epoch:** 10000

**Optimizer:** RMSprop | **Loss:** Mean Square Error

**Loss:** 0.1562 | **Accuracy:** 0.4660 | **Başarı Yüzdesi:** 25%

	F1	F2	F3	F4
F1	33	13	20	7
F2	35	9	23	12
F3	39	14	22	9
F4	35	6	24	19

**Loss:** 0.1180 | **Accuracy:** 0.6074 | **Başarı Yüzdesi:** 28%

	F1	F2	F3	F4
F1	37	10	13	17
F2	34	8	14	13
F3	40	10	22	18
F4	33	9	19	23

28

100 - ReLu

100 - Tanh

100 - Tanh

100 - Tanh

4 - Sigmoid

**Learning Rate:** 0.01 | **Rho:** 0.75 | **Batch Size:** 1024 | **Epoch:** 10000**Optimizer:** RMSprop | **Loss:** Mean Square Error**Loss:** 0.1581 | **Accuracy:** 0.4707 | **Başarı Yüzdesi:** 23%

	F1	F2	F3	F4
F1	35	11	27	8
F2	38	8	20	3
F3	33	14	22	7
F4	36	19	28	11

**Loss:** 0.1698 | **Accuracy:** 0.4014 | **Başarı Yüzdesi:** 24%

	F1	F2	F3	F4
F1	13	20	39	12
F2	12	15	41	10
F3	15	19	40	4
F4	15	13	42	10

28

100 - ReLu

100 - Tanh

100 - Tanh

100 - Tanh

4 - Sigmoid

**Learning Rate:** 0.01 | **Rho:** 0.50 | **Batch Size:** 1024 | **Epoch:** 10000**Optimizer:** RMSprop | **Loss:** Mean Square Error**Loss:** 0.1556 | **Accuracy:** 0.4561 | **Başarı Yüzdesi:** 29%

	F1	F2	F3	F4
F1	9	14	12	35
F2	5	25	21	32
F3	8	27	22	30
F4	7	14	22	37

**Loss:** 0.1651 | **Accuracy:** 0.4131 | **Başarı Yüzdesi:** 27%

	F1	F2	F3	F4
F1	2	11	17	42
F2	12	15	14	41
F3	8	14	16	38
F4	8	10	17	55



28

100 - ReLu

100 - Tanh

100 - Tanh

100 - Tanh

4 - Sigmoid

**Learning Rate:** 0.01 | **Rho:** 0.30 | **Batch Size:** 1024 | **Epoch:** 10000**Optimizer:** RMSprop | **Loss:** Mean Square Error**Loss:** 0.1444 | **Accuracy:** 0.5059 | **Başarı Yüzdesi:** 25%

	F1	F2	F3	F4
F1	14	30	18	22
F2	15	27	15	20
F3	17	23	17	20
F4	14	30	16	22

**Loss:** 0.1052 | **Accuracy:** 0.6309 | **Başarı Yüzdesi:** 29%

	F1	F2	F3	F4
F1	23	23	15	18
F2	17	29	12	16
F3	26	17	17	14
F4	20	28	21	24

28

100 - ReLu

100 - Tanh

100 - Tanh

100 - Tanh

4 - Sigmoid

**Learning Rate:** 0.01 | **Rho:** 0.10 | **Batch Size:** 1024 | **Epoch:** 10000**Optimizer:** RMSprop | **Loss:** Mean Square Error**Loss:** 0.1735 | **Accuracy:** 0.3857 | **Başarı Yüzdesi:** 22%

	F1	F2	F3	F4
F1	20	14	45	7
F2	21	15	41	5
F3	20	20	29	6
F4	18	18	32	9

**Loss:** 0.0910 | **Accuracy:** 0.6787 | **Başarı Yüzdesi:** 25%

	F1	F2	F3	F4
F1	17	18	27	17
F2	19	24	25	16
F3	32	16	17	17
F4	17	20	15	23

# LEARNING RATE

28

100 - ReLu

100 - Tanh

100 - Tanh

100 - Tanh

4 - Sigmoid

**Learning Rate:** 0.01 | **Rho:** 0.90 | **Batch Size:** 1024 | **Epoch:** 10000

**Optimizer:** RMSprop | **Loss:** Mean Square Error

**Loss:** 0.1562 | **Accuracy:** 0.4660 | **Başarı Yüzdesi:** 25%

	F1	F2	F3	F4
F1	33	13	20	7
F2	35	9	23	12
F3	39	14	22	9
F4	35	6	24	19

**Loss:** 0.1180 | **Accuracy:** 0.6074 | **Başarı Yüzdesi:** 28%

	F1	F2	F3	F4
F1	37	10	13	17
F2	34	8	14	13
F3	40	10	22	18
F4	33	9	19	23

28

100 - ReLu

100 - Tanh

100 - Tanh

100 - Tanh

4 - Sigmoid

**Learning Rate:** 0.005 | **Rho:** 0.90 | **Batch Size:** 1024 | **Epoch:** 10000**Optimizer:** RMSprop | **Loss:** Mean Square Error**Loss:** 0.1792 | **Accuracy:** 0.3359 | **Başarı Yüzdesi:** 23%

	F1	F2	F3	F4
F1	2	15	56	3
F2	2	26	53	2
F3	2	21	45	2
F4	1	27	60	2

**Loss:** 0.1680 | **Accuracy:** 0.4023 | **Başarı Yüzdesi:** 25%

	F1	F2	F3	F4
F1	14	5	28	29
F2	11	1	39	22
F3	16	6	41	13
F4	19	4	47	25

28

100 - ReLu

100 - Tanh

100 - Tanh

100 - Tanh

4 - Sigmoid

**Learning Rate:** 0.001 | **Rho:** 0.90 | **Batch Size:** 1024 | **Epoch:** 10000**Optimizer:** RMSprop | **Loss:** Mean Square Error**Loss:** 0.1155 | **Accuracy:** 0.6016 | **Başarı Yüzdesi:** 28%

	F1	F2	F3	F4
F1	24	16	16	18
F2	18	19	19	24
F3	19	17	19	26
F4	13	17	26	29

**Loss:** 0.1532 | **Accuracy:** 0.4414 | **Başarı Yüzdesi:** 21%

	F1	F2	F3	F4
F1	26	26	17	10
F2	29	11	18	10
F3	32	22	24	10
F4	33	20	23	9

28

100 - ReLu

100 - Tanh

100 - Tanh

100 - Tanh

4 - Sigmoid

**Learning Rate:** 0.0001 | **Rho:** 0.90 | **Batch Size:** 1024 | **Epoch:** 10000**Optimizer:** RMSprop | **Loss:** Mean Square Error**Loss:** 0.1598 | **Accuracy:** 0.4756 | **Başarı Yüzdesi:** 25%

	F1	F2	F3	F4
F1	16	14	27	21
F2	10	20	22	21
F3	14	16	28	30
F4	12	21	30	18

**Loss:** 0.1618 | **Accuracy:** 0.4736 | **Başarı Yüzdesi:** 25%

	F1	F2	F3	F4
F1	22	25	19	16
F2	16	18	16	24
F3	27	12	21	25
F4	19	25	13	22

# HIDDEN LAYERS

28

100 - ReLu100 - Tanh100 - Tanh4 - Sigmoid**Learning Rate:** 0.001 | **Rho:** 0.90 | **Batch Size:** 1024 | **Epoch:** 10000**Optimizer:** RMSprop | **Loss:** Mean Square Error**Loss:** 0.1507 | **Accuracy:** 0.4732 | **Başarı Yüzdesi:** 17%

	F1	F2	F3	F4
F1	20	1	42	21
F2	22	1	39	25
F3	26	2	25	24
F4	27	3	31	11

**Loss:** 0.1660 | **Accuracy:** 0.4209 | **Başarı Yüzdesi:** 19%

	F1	F2	F3	F4
F1	26	13	18	20
F2	38	6	11	21
F3	47	12	16	18
F4	43	7	9	15

28

100 - ReLu100 - Tanh4 - Sigmoid**Learning Rate:** 0.001 | **Rho:** 0.90 | **Batch Size:** 1024 | **Epoch:** 10000**Optimizer:** RMSprop | **Loss:** Mean Square Error**Loss:** 0.1818 | **Accuracy:** 0.3320 | **Başarı Yüzdesi:** 27%

	F1	F2	F3	F4
F1	9	32	13	21
F2	5	37	6	26
F3	8	51	8	24
F4	6	37	2	35

**Loss:** 0.1814 | **Accuracy:** 0.3340 | **Başarı Yüzdesi:** 22%

	F1	F2	F3	F4
F1	6	15	7	44
F2	7	19	2	59
F3	8	15	5	56
F4	9	19	6	43



28

100 - ReLu4 - Sigmoid**Learning Rate:** 0.001 | **Rho:** 0.90 | **Batch Size:** 1024 | **Epoch:** 10000**Optimizer:** RMSprop | **Loss:** Mean Square Error**Loss:** 0.3745 | **Accuracy:** 0.2520 | **Başarı Yüzdesi:** 26%

	F1	F2	F3	F4
<b>F1</b>	0	0	79	0
<b>F2</b>	0	0	69	0
<b>F3</b>	0	0	85	0
<b>F4</b>	0	0	87	0

**Loss:** 0.2500 | **Accuracy:** 0.2393 | **Başarı Yüzdesi:** 26%

	F1	F2	F3	F4
<b>F1</b>	85	0	0	0
<b>F2</b>	70	0	0	0
<b>F3</b>	67	0	0	0
<b>F4</b>	98	0	0	0

28

16 - ReLu4 - Sigmoid**Learning Rate:** 0.001 | **Rho:** 0.90 | **Batch Size:** 1024 | **Epoch:** 10000**Optimizer:** RMSprop | **Loss:** Mean Square Error**Loss:** 0.6230 | **Accuracy:** 0.2383 | **Başarı Yüzdesi:** 24%

	F1	F2	F3	F4
F1	0	73	0	0
F2	0	77	0	0
F3	0	91	0	0
F4	0	79	0	0

**Loss:** 0.3784 | **Accuracy:** 0.2432 | **Başarı Yüzdesi:** 24%

	F1	F2	F3	F4
F1	79	0	0	0
F2	66	0	0	0
F3	83	0	0	0
F4	92	0	0	0

28

16 - ReLu16 - Tanh4 - Sigmoid**Learning Rate:** 0.001 | **Rho:** 0.90 | **Batch Size:** 1024 | **Epoch:** 10000**Optimizer:** RMSprop | **Loss:** Mean Square Error**Loss:** 0.1872 | **Accuracy:** 0.2676 | **Başarı Yüzdesi:** 27%

	F1	F2	F3	F4
F1	1	0	43	35
F2	0	0	42	24
F3	0	0	47	34
F4	1	0	54	39

**Loss:** 0.1862 | **Accuracy:** 0.2861 | **Başarı Yüzdesi:** 22%

	F1	F2	F3	F4
F1	0	0	76	6
F2	0	0	65	8
F3	0	0	63	14
F4	0	0	75	13

28

16 - ReLu16 - Tanh16 - Tanh4 - Sigmoid**Learning Rate:** 0.001 | **Rho:** 0.90 | **Batch Size:** 1024 | **Epoch:** 10000**Optimizer:** RMSprop | **Loss:** Mean Square Error**Loss:** 0.1862 | **Accuracy:** 0.2930 | **Başarı Yüzdesi:** 23%

	F1	F2	F3	F4
F1	31	20	6	19
F2	39	15	3	15
F3	43	13	6	19
F4	38	23	6	24

**Loss:** 0.1851 | **Accuracy:** 0.2939 | **Başarı Yüzdesi:** 22%

	F1	F2	F3	F4
F1	0	15	53	23
F2	0	7	45	20
F3	0	16	46	19
F4	0	5	53	18

28

16 - ReLu16 - Tanh16 - Tanh16 - Tanh4 - Sigmoid**Learning Rate:** 0.001 | **Rho:** 0.90 | **Batch Size:** 1024 | **Epoch:** 10000**Optimizer:** RMSprop | **Loss:** Mean Square Error**Loss:** 0.1873 | **Accuracy:** 0.2695 | **Başarı Yüzdesi:** 25%

	F1	F2	F3	F4
F1	5	0	17	51
F2	6	0	22	54
F3	14	0	15	42
F4	10	0	21	63

**Loss:** 0.1858 | **Accuracy:** 0.2871 | **Başarı Yüzdesi:** 24%

	F1	F2	F3	F4
F1	5	30	0	47
F2	0	27	0	35
F3	0	34	0	54
F4	1	41	1	45

# OPT. & MOMENTUM

28

100 - ReLu

100 - Tanh

100 - Tanh

100 - Tanh

4 - Sigmoid

**Learning Rate:** 0.001 | **Momentum:** 0.90 | **Batch Size:** 1024 | **Epoch:** 10000

**Optimizer:** SGD | **Loss:** Mean Square Error

**Loss:** 0.1870 | **Accuracy:** 0.2656 | **Başarı Yüzdesi:** 24%

	F1	F2	F3	F4
<b>F1</b>	0	2	14	63
<b>F2</b>	0	0	15	57
<b>F3</b>	0	0	9	76
<b>F4</b>	0	0	15	69

**Loss:** 0.1872 | **Accuracy:** 0.2754 | **Başarı Yüzdesi:** 23%

	F1	F2	F3	F4
<b>F1</b>	0	0	0	83
<b>F2</b>	1	0	0	70
<b>F3</b>	0	0	0	90
<b>F4</b>	0	0	0	76

28

100 - ReLu

100 - Tanh

100 - Tanh

100 - Tanh

4 - Sigmoid

**Learning Rate:** 0.001 | **Momentum:** 0.70 | **Batch Size:** 1024 | **Epoch:** 10000**Optimizer:** SGD | **Loss:** Mean Square Error**Loss:** 0.1858 | **Accuracy:** 0.2969 | **Başarı Yüzdesi:** 25%

	F1	F2	F3	F4
F1	11	24	15	21
F2	11	19	22	28
F3	6	30	22	25
F4	14	21	28	31

**Loss:** 0.1873 | **Accuracy:** 0.2559 | **Başarı Yüzdesi:** 26%

	F1	F2	F3	F4
F1	0	0	1	73
F2	0	0	2	67
F3	0	0	0	89
F4	0	1	2	85

28

100 - ReLu

100 - Tanh

100 - Tanh

100 - Tanh

4 - Sigmoid

**Learning Rate:** 0.001 | **Momentum:** 0.50 | **Batch Size:** 1024 | **Epoch:** 10000**Optimizer:** SGD | **Loss:** Mean Square Error**Loss:** 0.1866 | **Accuracy:** 0.2852 | **Başarı Yüzdesi:** 28%

	F1	F2	F3	F4
F1	8	19	16	27
F2	6	26	21	23
F3	15	25	22	25
F4	6	31	15	35

**Loss:** 0.1866 | **Accuracy:** 0.2998 | **Başarı Yüzdesi:** 24%

	F1	F2	F3	F4
F1	3	14	33	30
F2	5	13	27	36
F3	6	13	23	31
F4	4	15	28	39



28

100 - ReLu

100 - Tanh

100 - Tanh

100 - Tanh

4 - Sigmoid

**Learning Rate:** 0.001 | **Momentum:** 0.15 | **Batch Size:** 1024 | **Epoch:** 10000**Optimizer:** SGD | **Loss:** Mean Square Error**Loss:** 0.1866 | **Accuracy:** 0.2793 | **Başarı Yüzdesi:** 28%

	F1	F2	F3	F4
F1	14	17	28	8
F2	14	31	21	8
F3	10	28	29	14
F4	16	25	41	16

**Loss:** 0.1863 | **Accuracy:** 0.3105 | **Başarı Yüzdesi:** 19%

	F1	F2	F3	F4
F1	6	16	6	42
F2	23	14	14	28
F3	19	14	12	43
F4	17	16	19	31

28

100 - ReLu

100 - Tanh

100 - Tanh

100 - Tanh

4 - Sigmoid

**Learning Rate:** 0.001 | **Momentum:** 0.05 | **Batch Size:** 1024 | **Epoch:** 10000**Optimizer:** SGD | **Loss:** Mean Square Error**Loss:** 0.1858 | **Accuracy:** 0.3047 | **Başarı Yüzdesi:** 26%

	F1	F2	F3	F4
F1	13	12	21	34
F2	12	10	19	36
F3	7	11	21	50
F4	10	3	19	42

**Loss:** 0.1863 | **Accuracy:** 0.3105 | **Başarı Yüzdesi:** 26%

	F1	F2	F3	F4
F1	14	21	33	10
F2	16	13	35	10
F3	17	16	35	11
F4	9	15	43	22

28

100 - ReLu

100 - Tanh

100 - Tanh

100 - Tanh

4 - Sigmoid

**Learning Rate:** 0.001 | — | **Batch Size:** 1024 | **Epoch:** 10000**Optimizer:** RMSprop | **Loss:** Mean Square Error**Loss:** 0.1128 | **Accuracy:** 0.6133 | **Başarı Yüzdesi:** 27%

	F1	F2	F3	F4
F1	15	12	10	31
F2	17	14	19	35
F3	17	9	20	31
F4	16	16	18	40

**Loss:** 0.1397 | **Accuracy:** 0.5176 | **Başarı Yüzdesi:** 22%

	F1	F2	F3	F4
F1	14	15	26	17
F2	22	12	27	25
F3	18	19	20	26
F4	14	17	23	25

## Kaynakça

- ★ Python ile Derin Öğrenme, François Chollet, Buzdağı Yayınları, Eylül 2019
- ★ <https://archive.ics.uci.edu/ml/datasets/Hepatitis+C+Virus+%28HCV%29+for+Egyptian+patients>
- ★ <https://devhunteryz.wordpress.com/2018/06/20/geri-yayilimbackpropagation/>
- ★ <https://medium.com/deep-learning-turkiye/derin-ogrenme-uygulamalarinda-en-sik-kullanilan-hiper-parametreler-ece8e9125c4>
- ★ <https://datascience.stackexchange.com/questions/40067/confusion-matrix-three-classes-python>
- ★ [https://scikit-learn.org/stable/auto\\_examples/model\\_selection/plot\\_roc.html](https://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html)
- ★ <https://stackoverflow.com/questions/33547965/computing-auc-and-roc-curve-from-multi-class-data-in-scikit-learn-sklearn>
- ★ <https://mc.ai/keras-ile-derin-ogrenme-modeli-olusturma/>
- ★ <https://www.turkcerrahi.com/tip-sozlugu/fibrozis/>
- ★ <https://jovianlin.io/keras-models-sequential-vs-functional/>
- ★ <https://keras.io/models/sequential/>
- ★ <https://keras.io/optimizers/>
- ★ <https://keras.io/metrics/>
- ★ <https://keras.io/losses/>
- ★ <https://mlfromscratch.com/optimizers-explained/>
- ★ <https://towardsdatascience.com/learning-rate-schedules-and-adaptive-learning-rate-methods-for-deep-learning-2c8f433990d1>