YMH451 Yapay Zeka ve Uzman Sistemler Dersi

Hafta4 Uygulama Sorularının Cevapları

Soru1

# -\*- coding: utf-8 -\*-

"""

Created on Fri Oct 25 15:17:48 2019

@author: Emrullah

"""

**from** keras**.**datasets **import** fashion\_mnist

**((**trainX**,** trainY**),** **(**testX**,** testY**))** **=** fashion\_mnist**.**load\_data**()**

**import** numpy **as** np

**def** veriVektorleme **(**diziler**,**boyut**=**10000**):**

sonuclar **=** np**.**zeros**((**len**(**diziler**),**boyut**))**

**for** i**,** dizi **in** enumerate**(**diziler**):**

sonuclar**[**i**,**dizi**]** **=** 1.

**return** sonuclar

x\_egitim **=** veriVektorleme**(**trainX**)**

x\_test **=** veriVektorleme**(**testX**)**

y\_eğitim **=** np**.**asarray**(**trainY**).**astype**(**"float32"**)**

y\_test **=** np**.**asarray**(**testY**).**astype**(**"float32"**)**

**from** keras **import** layers

**from** keras **import** models

model **=** models**.**Sequential**()**

model**.**add**(**layers**.**Dense**(**16**,**activation**=**"relu"**,** input\_shape**=(**10000**,)))**

model**.**add**(**layers**.**Dense**(**16**,** activation**=**"relu"**))**

model**.**add**(**layers**.**Dense**(**1**,** activation**=**"sigmoid"**))**

model**.**compile**(**optimizer**=**"rmsprop"**,** loss**=**"binary\_crossentropy"**,** metrics**=[**"accuracy"**])**

ysa **=** model**.**fit**(**x\_egitim**,**y\_eğitim**,** epochs**=**10**,** batch\_size**=**512**,** validation\_data**=(**x\_egitim**,**y\_eğitim**))**

**import** matplotlib**.**pyplot **as** plt

kayıp **=** ysa**.**history**[**"loss"**]**

doğrulamaKayıp **=** ysa**.**history**[**"val\_loss"**]**

epok **=** range**(**1**,** len**(**kayıp**)** **+** 1**)**

plt**.**plot**(**epok**,** kayıp**,**"b"**,** label**=**"Eğitimn kaybı"**)**

plt**.**plot**(**epok**,**doğrulamaKayıp**,** "r"**,** label **=** "Doğrulama kaybı"**)**

plt**.**title**(**"Eğitim ve doğrulama kaybı"**)**

plt**.**xlabel**(**"Epoklar"**)**

plt**.**ylabel**(**"Kayıp"**)**

plt**.**legend**()**

plt**.**show**()**

**import** matplotlib**.**pyplot **as** plt

başarım **=** ysa**.**history**[**"acc"**]**

doğrulamaBaşarım **=** ysa**.**history**[**"val\_acc"**]**

epok **=** range**(**1**,** len**(**kayıp**)** **+** 1**)**

plt**.**plot**(**epok**,** başarım**,**"b"**,** label**=**"Eğitimn kaybı"**)**

plt**.**plot**(**epok**,**doğrulamaBaşarım**,** "r"**,** label **=** "Doğrulama kaybı"**)**

plt**.**title**(**"Eğitim ve doğrulama kaybı"**)**

plt**.**xlabel**(**"Epoklar"**)**

plt**.**ylabel**(**"Kayıp"**)**

plt**.**legend**()**

plt**.**show**()**

ortalamBaşarım **=** np**.**mean**(**doğrulamaBaşarım**)**

**print(**"Ortlama başarım. "**,** ortalamBaşarım**)**

**Soru2**

**import** pandas **as** pd

**from** google**.**colab **import** files

file **=** files**.**upload**()**

data **=** pd**.**read\_csv**(**"breastCancer.csv"**,** header **=** **None)**

data**.**info**()**

data**.**head**(**15**)**

df **=** pd**.**DataFrame**(**data**)**

df**[**6**]** **=** df**[**6**].**replace**(**'?'**,**0**)**

df**[**6**]** **=** df**[**6**].**astype**(**int**)**

mean **=** int**(**df**[**6**].**mean**())**

df**[**6**]** **=** df**[**6**].**replace**(**0**,**mean**)**

df**.**head**()**

df **=** pd**.**DataFrame**(**data**)**

df**[**6**].**value\_counts**()**

df**[**10**]** **=** df**[**10**].**replace**(**2**,**0**).**replace**(**4**,**1**)**

df**.**head**(**10**)**

**from** sklearn**.**model\_selection **import** train\_test\_split

X **=** df**.**iloc**[:,** 1**:**10**]**

Y **=** df**.**iloc**[:,** 10**]**

X\_train**,** X\_test**,** Y\_train**,** Y\_test **=** train\_test\_split**(**X**,** Y**,** test\_size **=** 0.2**,** random\_state **=** 0**)**

**from** sklearn**.**preprocessing **import** StandardScaler

scaler **=** StandardScaler**()**

X\_train **=** scaler**.**fit\_transform**(**X\_train**)**

X\_test **=** scaler**.**transform**(**X\_test**)**

**from** keras**.**models **import** Sequential

**from** keras**.**layers **import** Dense

classifier **=** Sequential**()**

classifier**.**add**(**Dense**(**units **=** 5**,** activation **=** 'relu'**,** input\_dim**=**9**))**

classifier**.**add**(**Dense**(**units **=** 3**,** activation **=** 'relu'**))**

classifier**.**add**(**Dense**(**units **=** 1**,** activation **=** 'sigmoid'**))**

classifier**.**compile**(**optimizer**=**'adam'**,** loss**=**'mean\_squared\_error'**,** metrics**=[**'accuracy'**])**

classifier**.**fit**(**X\_train**,** Y\_train**,** batch\_size **=** 10**,** epochs **=** 100**)**

Y\_pred **=** classifier**.**predict**(**X\_test**)**

Y\_pred **=** **[** 1 **if** y**>=**0.5 **else** 0 **for** y **in** Y\_pred**]**

**print(**Y\_pred**)**

**from** sklearn**.**metrics **import** confusion\_matrix

cm **=** confusion\_matrix**(**Y\_test**,** Y\_pred**)**

**print(**cm**)**