

HW5 Rubrics

Total: 6 points (6 + 1) capped at 6

If not submitted as a .zip, -1

Late penalty: 0.5/day after 12/5 11:59PM

Q1. (2 points)

+1: weight.h5

+1: screenshot showing values for the second dense layer matches weight.h5

HDFView 2.14

File Window Tools Help

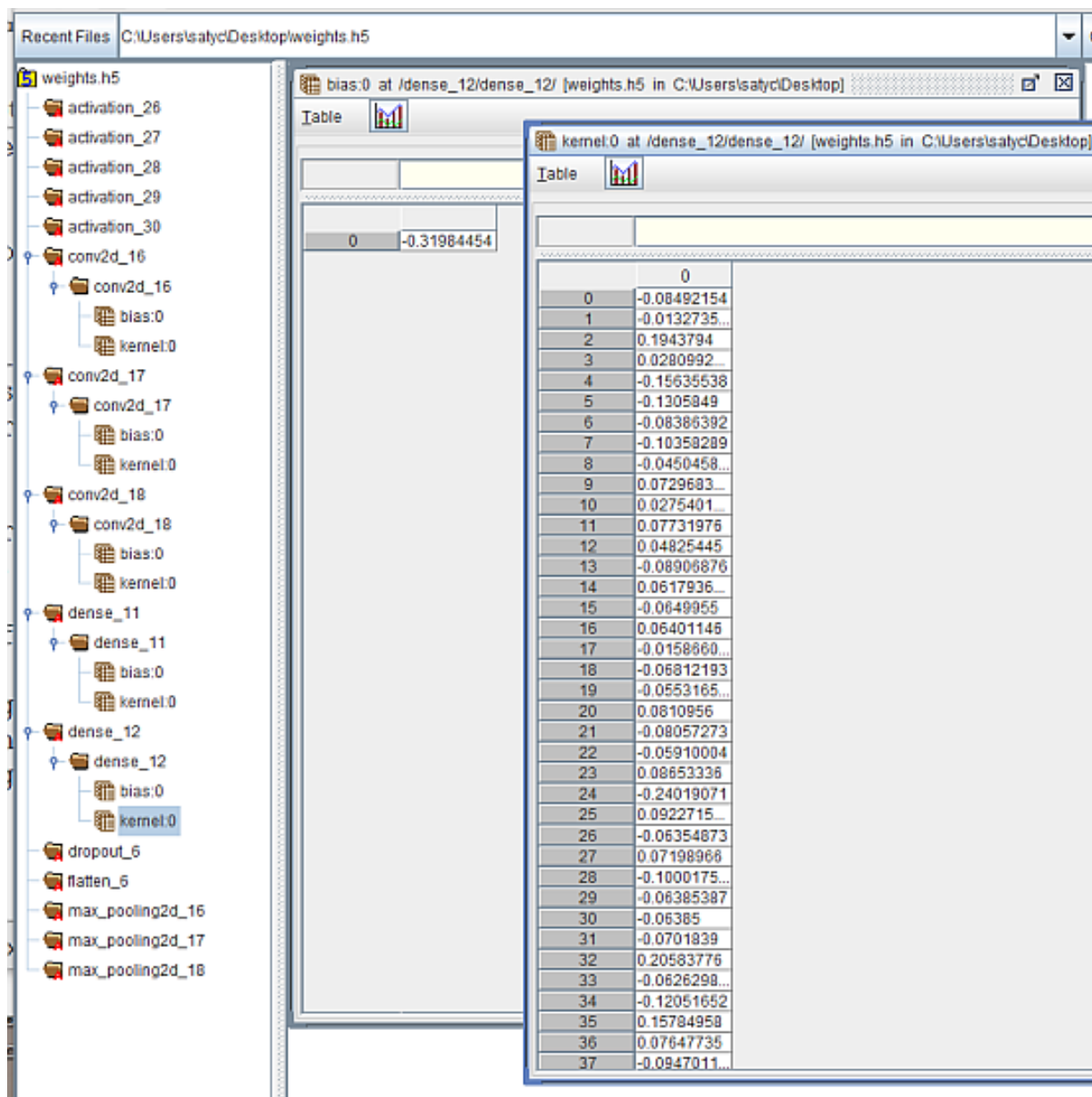
Recent Files: weights.h5

weights.h5

- activation_5
- activation_6
- activation_7
- activation_8
- activation_9
- conv2d_3
- conv2d_4
- conv2d_5
- dense_2
 - bias:0
 - kernel:0
- dense_3
 - bias:0
 - kernel:0
- dense_4
 - bias:0
 - kernel:0
- dropout_1
- flatten_1
- max_pooling2d_3
- max_pooling2d_4
- max_pooling2d_5
- top_level_model_weights

Table: kernel:0 at /dense_2/dense_2/ [weights.h5 in /home/phil/Downloads]

	0	1	2	3	4	5	6	7	8	9
0	0.052021...	-0.00749...	-0.05614...	-0.01131...	0.052585...	0.0966447...	0.085006...	0.005574...	0.010971...	-0.01528...
1	0.041888...	-0.01808...	-0.02155...	-0.01316...	3.341031...	-0.05941...	0.069555...	0.013432...	0.050234...	-0.00968...
2	0.043320...	-0.01257...	0.025668...	-0.00378...	0.069260...	0.007241...	0.065156...	0.091384...	0.053139...	-0.01798...
3	-0.02386...	-0.00728...	-0.11954...	7.005441...	-0.05469...	-0.03913...	-0.03737...	0.002974...	0.114472...	0.002638...
4	0.098632...	-0.01998...	0.155003...	1.286101...	0.074765...	0.136139...	0.125143...	0.043894...	0.106056...	0.004931...
5	0.169093...	-0.01717...	0.0943474...	0.004475...	0.077984...	0.064134...	0.102245...	0.113269...	0.093802...	-0.01102...
6	0.180969...	0.002503...	0.155584...	-1.87772...	0.143561...	0.143591...	0.158296...	0.192671...	0.083493...	0.007414...
7	9.509647...	0.005586...	0.082764...	-0.00518...	0.029444...	0.010464...	0.010993...	9.843013...	0.0662728...	-0.00572...
8	0.114421...	-0.00917...	0.117652...	-0.00163...	0.103600...	0.0778603...	0.083012...	0.115040...	0.107925...	-0.02035...
9	0.081560...	0.011086...	0.0664612...	0.006226...	0.056582...	0.025615...	0.115764...	0.090563...	0.040127...	0.004113...
10	0.026002...	0.003617...	0.078838...	-0.00869...	0.036101...	0.096434...	0.083974...	0.107321...	0.073918...	-4.96935...
11	0.063919...	-0.01857...	0.008068...	0.002973...	-0.05840...	-0.07651...	0.042101...	-0.00806...	0.056483...	-0.02236...
12	0.194189...	-0.01212...	0.154249...	0.006767...	0.096458...	0.100609...	0.211552...	0.129744...	0.04979...	0.010573...
13	-0.11002...	-0.01661...	-0.22599...	-0.00115...	-0.15405...	-0.12181...	-0.18492...	-0.13021...	0.164190...	-0.01344...
14	0.071349...	-0.02912...	0.235914...	-0.01676...	0.075480...	0.089702...	0.076437...	0.168160...	0.055715...	-0.00202...
15	-0.02944...	0.002056...	-0.00134...	-0.00137...	0.018306...	0.011074...	-0.02162...	0.03526...	-0.02411...	-0.00902...
16	0.043417...	0.017111...	2.199718...	-0.00725...	0.013237...	0.033093...	0.040820...	0.062495...	0.04260...	-0.01848...
17	0.076908...	-8.83821...	0.102128...	-0.00614...	0.036058...	0.016648...	-0.01785...	0.017553...	0.023876...	0.014202...
18	0.0388086...	7.057679...	0.017414...	0.008104...	0.051099...	0.026411...	0.065304...	0.014017...	0.016745...	0.002144...
19	-0.02723...	-0.00998...	-0.03103...	-0.01174...	-0.01825...	-0.00115...	0.017548...	9.08833E-4...	2.69655...	-0.00788...
20	0.067682...	-0.00274...	-0.00801...	-0.01178...	0.023712...	-0.01022...	0.055008...	5.638173...	0.094717...	0.001531...
21	0.005856...	0.017986...	0.022106...	-0.00977...	0.055145...	0.090758...	0.037706...	0.052342...	0.086400...	0.009426...
22	-0.02692...	0.002603...	-0.01054...	-0.00171...	0.009260...	-0.01162...	-0.00554...	-0.00356...	0.007211...	0.006219...
23	-0.01500...	-0.01862...	-0.01033...	-0.01216...	-0.00525...	0.009381...	0.007444...	0.004431...	0.008585...	-0.01108...
24	0.112639...	-0.01175...	0.137983...	-0.01429...	0.090710...	0.097866...	0.095094...	0.078589...	0.087356...	-0.00793...
25	0.072094...	-0.01649...	0.090218...	-0.02423...	0.008126...	0.065671...	0.081761...	0.057128...	0.019645...	-0.00885...
26	-0.10071...	-0.01803...	-0.04650...	0.011301...	-0.01104...	-0.03291...	-0.04758...	-0.06895...	0.100033...	-0.01068...
27	0.075875...	0.002206...	0.101294...	-0.00641...	0.084364...	0.061996...	0.146238...	0.111792...	0.018023...	0.011896...
28	0.084363...	0.012950...	-0.00575...	-0.01181...	0.044226...	0.019513...	0.104320...	0.022646...	0.020951...	-0.01519...
29	0.132289...	-1.62234...	0.101736...	-0.00998...	0.098554...	0.099923...	0.093485...	0.1051601...	0.090571...	-4.43595...
30	0.122533...	0.002729...	0.048812...	-0.01579...	0.110678...	0.159399...	0.107490...	0.057683...	-0.00584...	-0.0172924...
31	0.078387...	0.014032...	0.074184...	0.003891...	0.006597...	0.035638...	0.051106...	0.025777...	0.086554...	0.013620...
32	0.227095...	0.010810...	0.212541...	0.003946...	0.235137...	0.161414...	0.165654...	0.236827...	-0.06345...	0.017385...
33	-0.05099...	-1.40770...	-0.14831...	-0.01182...	-0.00448...	-0.03757...	0.012322...	0.011110...	0.046658...	0.007642...
34	-0.04263...	-0.00765...	-0.07065...	-0.01982...	-0.04412...	0.011182...	-0.0212357...	-0.0138489...	0.025868...	0.002409...
35	0.074696...	0.002798...	0.092945...	0.009503...	0.039355...	0.050419...	0.066226...	0.026892...	0.071123...	0.011375...
36	0.0253177...	-0.02100...	0.014608...	-0.01064...	0.01064...	0.016155...	0.047432...	0.0576001...	0.074355...	-0.01040...
37	0.026418...	0.001804...	0.109489...	0.013259...	0.057567...	0.070535...	0.067205...	0.037604...	0.045517...	-0.00352...
38	-0.00651...	0.003037...	0.025000...	-0.00584...	0.049733...	-0.00800...	0.066241...	-0.01841...	0.102907...	-0.01698...
39	0.006736...	0.009192...	0.018639...	0.004087...	-0.01754...	0.020867...	0.042304...	-0.00430...	0.020098...	0.010391...
40	0.002365...	0.004232...	0.001561...	0.013684...	-0.01030...	-8.18085...	4.197542...	-0.00698...	-0.00578...	-0.00909...
41	0.023037...	-0.01776...	0.021389...	0.009177...	-0.00255...	0.016507...	0.049054...	0.017229...	0.064895...	-0.00252...
42	0.0188111...	0.015352...	-5.11927...	0.012263...	-0.01764...	-0.04291...	0.009149...	0.002890...	0.026684...	-0.00303...
43	0.013417...	0.013320...	-0.02048...	0.014836...	0.008376...	-0.00516...	0.015999...	0.011242...	0.001588...	-0.01632...
44	0.098095...	-0.00508...	0.110264...	-0.00176...	0.026322...	0.028521...	0.057063...	0.058677...	0.006404...	-0.00362...
45	-0.0428349...	-0.01491...	-0.17983...	0.002864...	-0.11472...	-0.04427...	-0.05507...	-0.15537...	0.111217...	-0.00173...
46	0.026964...	-0.01202...	9.938073...	-0.00896...	0.012781...	0.037446...	0.004494...	0.001701...	0.043456...	-0.00186...
47	0.041910...	0.001631...	0.124578...	0.014705...	0.058115...	0.096422...	0.091869...	0.022599...	0.071387...	8.176383...
48	-0.01217...	-0.01813...	0.040789...	0.007664...	-0.01680...	0.041908...	0.055408...	-0.01965...	0.029143...	0.010291...
49	0.053703...	-0.02094...	0.041199...	-6.96480...	-0.00750...	-0.04991...	0.011919...	0.038487...	0.084800...	0.003821...
50	-0.02751...	0.008061...	0.002250...	-0.00705...	-0.00876...	-0.00726...	-0.04723...	0.054909...	0.047530...	0.004317...
51	-0.13941...	0.005701...	-0.08471...	-0.01571...	-0.00100...	0.019279...	-0.10527...	-0.12101...	0.092038...	-0.00670...
52	0.105512...	-0.00482...	0.089708...	0.001690...	0.018365...	0.104729...	0.138647...	0.158386...	-0.00774...	0.016854...
53	0.057103...	0.004404...	0.071331...	-0.01959...	0.098100...	0.001866...	0.029175...	0.050493...	0.045183...	-0.01708...
54	0.026573...	-0.00935...	0.071484...	-0.01309...	0.022622...	0.026472...	0.043088...	0.064859...	0.2378176...	-0.01566...
55	0.001454...	-0.00364...	-0.01282...	-0.01037...	-0.01600...	1.536309...	0.005190...	0.009318...	-0.02511...	-0.00492...
56	-0.00797...	0.006837...	0.003001...	0.00978...	-0.00976...	-0.00891...	0.002334...	0.002260...	-0.00124...	-0.00440...
57	0.057843...	0.005993...	0.051280...	0.010005...	0.018742...	0.007295...	0.055669...	0.035843...	0.1216092...	-0.00615...
58	-0.00937...	-0.01053...	0.001489...	-0.01224...	-0.01923...	0.004217...	0.009779...	0.004155...	-0.00990...	0.017796...
59	-0.096141...	-0.01511...	-0.09976...	0.010510...	-0.02862...	-0.03957...	9.650993...	0.008480...	0.030485...	-0.00228...
60	-3.52374...	0.014613...	-0.02678...	-0.01815...	0.002758...	0.003940...	0.009840...	0.012824...	-0.01967...	0.006924...
61	-0.0270141...	0.00562...	0.028536...	0.010410...	0.051620...	-0.042631...	0.004260...	0.024265...	0.026643...	-0.00402...



Q2. (2 points)

Sample:



what1.jpg



what2.jpg



```
model.add(Conv2D(32, (3, 3), input_shape=input_shape))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(32, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(64, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())
model.add(Dense(64))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(1))
model.add(Activation('sigmoid'))
model.compile(loss='binary_crossentropy',
              optimizer='rmsprop',
              metrics=['accuracy'])

#from keras.models import load_model
model.load_weights('/content/drive/My Drive/Colab Notebooks/cats-vs-dogs/weights.h5')

from keras.preprocessing import image

myPic = '/content/drive/My Drive/Colab Notebooks/cats-vs-dogs/data/live/what1.jpg'
#myPic = '/content/drive/My Drive/Colab Notebooks/cats-vs-dogs/data/live/doglike-cat.jpg'

test_image= image.load_img(myPic, target_size = (img_width, img_height))
test_image = image.img_to_array(test_image)
test_image = test_image.reshape(input_shape)
test_image = numpy.expand_dims(test_image, axis = 0)
result = model.predict(test_image,verbose=0)
print(result[0][0])

myPic2 = '/content/drive/My Drive/Colab Notebooks/cats-vs-dogs/data/live/what2.jpg'
#myPic2 = '/content/drive/My Drive/Colab Notebooks/cats-vs-dogs/data/live/corgi.jpg'
test_image2= image.load_img(myPic2, target_size = (img_width, img_height))
test_image2 = image.img_to_array(test_image2)
test_image2 = test_image2.reshape(input_shape)
test_image2 = numpy.expand_dims(test_image2, axis = 0)
result = model.predict(test_image2,verbose=0)
print(result[0][0])

0.0
1.0
```

- +1: two images of cat (0.5) and dog (0.5)
- +1: screenshot with correct predictions of cat (0.5) and dog (0.5)
- 0.5 if code is not included. Code should be in the screenshot to show classification results corresponding to pictures.

[Saty: either code, or just the part showing the classification, is fine!]

Note:

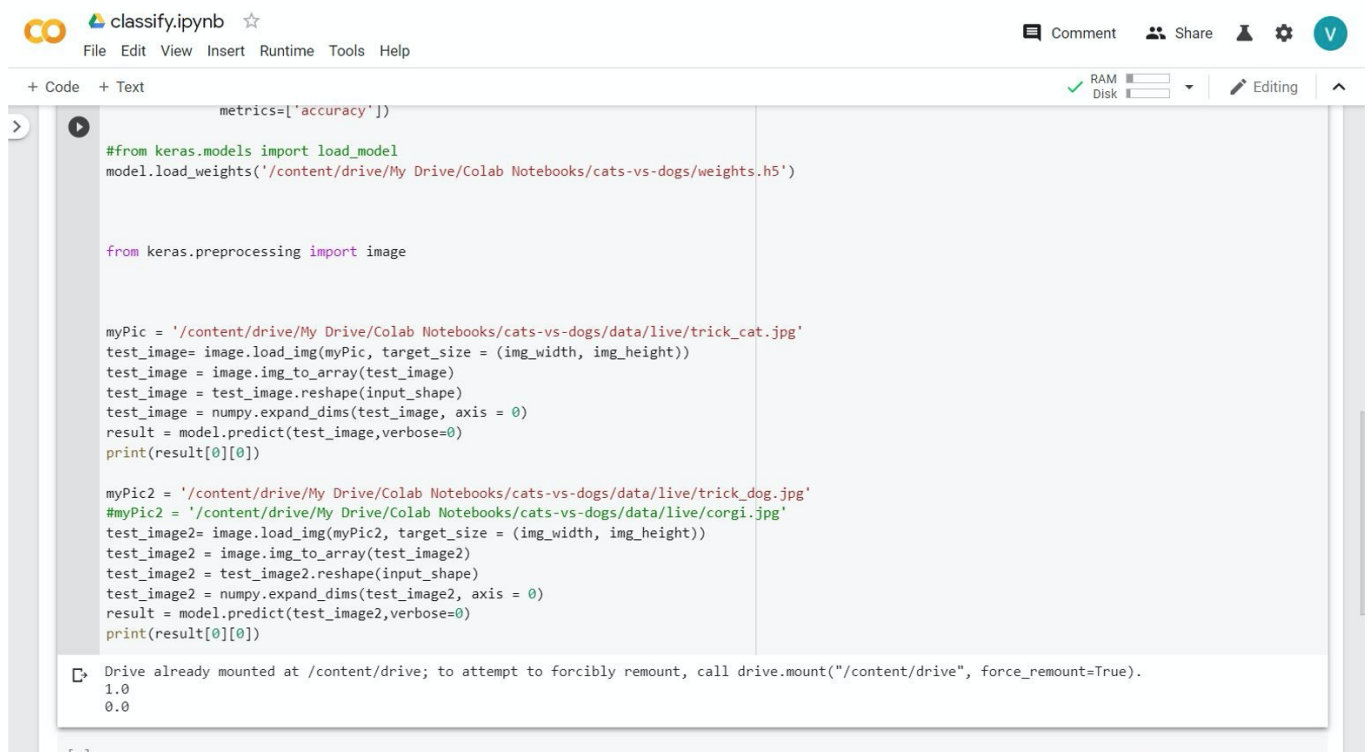
the values with a 20% range are acceptable. E.g. for classification 1, accuracy > 0.8 should be OK

Q3. (2 points)

+1: two images of 'trick' cat (0.5) and 'trick' dog (0.5) - any breed
+1: screenshot with misclassification of of cat (0.5) and dog (0.5)
-0.5 if code is not included. Code should be in the screenshot to show classification results corresponding to pictures.

[Saty: either code, or just the part showing the classification, is fine!]

Node: the values with a 20% range are acceptable. E.g. for classification 1, accuracy > 0.8 should be OK



```
metrics=['accuracy'])

#from keras.models import load_model
model.load_weights('/content/drive/My Drive/Colab Notebooks/cats-vs-dogs/weights.h5')

from keras.preprocessing import image

myPic = '/content/drive/My Drive/Colab Notebooks/cats-vs-dogs/data/live/trick_cat.jpg'
test_image= image.load_img(myPic, target_size = (img_width, img_height))
test_image = image.img_to_array(test_image)
test_image = test_image.reshape(input_shape)
test_image = numpy.expand_dims(test_image, axis = 0)
result = model.predict(test_image,verbose=0)
print(result[0][0])

myPic2 = '/content/drive/My Drive/Colab Notebooks/cats-vs-dogs/data/live/trick_dog.jpg'
#myPic2 = '/content/drive/My Drive/Colab Notebooks/cats-vs-dogs/data/live/corgi.jpg'
test_image2= image.load_img(myPic2, target_size = (img_width, img_height))
test_image2 = image.img_to_array(test_image2)
test_image2 = test_image2.reshape(input_shape)
test_image2 = numpy.expand_dims(test_image2, axis = 0)
result = model.predict(test_image2,verbose=0)
print(result[0][0])
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

```
1.0
0.0
```

Bonus: (1 point)

chihuahua Dog, correct muffin IDing, mis IDing miss any one of them -1

Prediction doesn't necessarily have to be the first label.

[Saty: for the misIDing of the muffin, any dog breed is ok, for the label]