(tensorflow) C:\Users\FACULTY OF SCIENCE>python

Python 3.7.7 (default, May 6 2020, 11:45:54) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32

Type "help", "copyright", "credits" or "license" for more information.

>>> import pathlib as pb

>>> import pathlib as pb

>>> import tensorflow as tf

>>> import numpy as np

>>> from tensorflow import feature\_column

>>> from tensorflow.keras import layers

>>> pip install sklearn

>>> import sklearn as sk

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

>>> import pathlib as pb

>>> dataset\_url='http://storage.googleapis.com/download.tensorflow.org/data/petfinder-mini.zip'

>>> csv\_file='datasets/petfinder-mini/petfinder-mini.csv'

>>> tf.keras.utils.get\_file('petfinder\_mini.zip',dataset\_url,extract=True, cache\_dir='.')

Downloading data from http://storage.googleapis.com/download.tensorflow.org/data/petfinder-mini.zip

1671168/1668792 [==============================] - 1s 1us/step

'.\\datasets\\petfinder\_mini.zip'

>>> dataframe=pd.read\_csv(csv\_file)

>>> dataframe.head()

Type Age Breed1 ... Description PhotoAmt AdoptionSpeed

0 Cat 3 Tabby ... Nibble is a 3+ month old ball of cuteness. He ... 1 2

1 Cat 1 Domestic Medium Hair ... I just found it alone yesterday near my apartm... 2 0

2 Dog 1 Mixed Breed ... Their pregnant mother was dumped by her irresp... 7 3

3 Dog 4 Mixed Breed ... Good guard dog, very alert, active, obedience ... 8 2

4 Dog 1 Mixed Breed ... This handsome yet cute boy is up for adoption.... 3 2

[5 rows x 15 columns]

>>>

>>> init\_op=global\_variables\_initializer()

>>> print(tf.add(4,6))

tf.Tensor(10, shape=(), dtype=int32)

>>> tensor\_id =np.array([1.3,3,1,4.0,23.99])

>>> print(tensor\_id)

[ 1.3 3. 1. 4. 23.99]

>>> print(tensor\_id[2])

1.0

>>> tensor\_2d =np.array([(1.3,3,1,4.0,23.99),(1,2,3,4,5),(6,7,8,9,0),(3,4,5,6,7)])

>>> print(tensor\_2d)

[[ 1.3 3. 1. 4. 23.99]

[ 1. 2. 3. 4. 5. ]

[ 6. 7. 8. 9. 0. ]

[ 3. 4. 5. 6. 7. ]]

>>> tensor\_2d[3][2]

5.0

>>> matrix1 = np.array([(2,2,2),(2,2,2),(2,2,2)],dtype='int32')

>>> matrix2 = np.array([(1,1,1),(1,1,1),(1,1,1)],dtype='int32')

>>> print (matrix1)

[[2 2 2]

[2 2 2]

[2 2 2]]

>>> print (matrix2)

[[1 1 1]

[1 1 1]

[1 1 1]]

>>> matrix1 = tf.constant(matrix1)

>>> matrix2 = tf.constant(matrix2)

>>> matrix\_product = tf.matmul(matrix1, matrix2)

>>> matrix\_sum = tf.add(matrix1,matrix2)

>>> matrix\_3 = np.array([(2,7,2),(1,4,2),(9,0,2)],dtype='float32')

>>> print (matrix\_3)

[[2. 7. 2.]

[1. 4. 2.]

[9. 0. 2.]]

>>> tf.print(matrix\_sum)

[[3 3 3]

[3 3 3]

[3 3 3]]

>>> forrtl: error (200): program aborting due to control-C event

Image PC Routine Line Source

libifcoremd.dll 00007FFE51183B58 Unknown Unknown Unknown

KERNELBASE.dll 00007FFE81146634 Unknown Unknown Unknown

KERNEL32.DLL 00007FFE82508102 Unknown Unknown Unknown

ntdll.dll 00007FFE845DC5B4 Unknown Unknown Unknown

>>> dataframe['target']=np.where(dataframe['AdoptionSpeed']==4,0,1)

>>> dataframe=dataframe.drop(columns=['AdoptionSpeed','Description'])

>>> train, test = train\_test\_split(dataframe, test\_size =0.2)

>>> train, val = train\_test\_split(train, test\_size =0.2)

>>> print(len(train), 'train example')

7383 train example

>>> print(len(val), 'validation example')

1846 validation example

>>> print(len(test), 'test example')

2308 test example

>>> def df\_to\_dataset(dataframe,shuffle=True, batch\_size=32):

... dataframe=dataframe.copy()

... labels = dataframe.pop('target')

... ds = tf.data.Dataset.from\_tensor\_slices((dict(dataframe), labels))

... if shuffle:

... ds=ds.shuffle(buffer\_size=len(dataframe))

... ds=ds.batch(batch\_size)

... return ds

...

>>> batch\_size=5

>>> train\_ds=df\_to\_dataset(train, batch\_size=5)

>>> val\_ds=df\_to\_dataset(val, shuffle=False, batch\_size=5)

>>> test\_ds=df\_to\_dataset(test, shuffle=False, batch\_size=5)

>>> for feature\_batch, label\_batch in train\_ds.take(1):

... print('Every feature:', list(feature\_batch.keys()))

... print('A batch of age:', feature\_batch['Age'])

... print('A batch of targets:', label\_batch)

...

Every feature: ['Type', 'Age', 'Breed1', 'Gender', 'Color1', 'Color2', 'MaturitySize', 'FurLength', 'Vaccinated', 'Sterilized', 'Health', 'Fee', 'PhotoAmt']

A batch of age: tf.Tensor([2 2 4 5 5], shape=(5,), dtype=int32)

A batch of targets: tf.Tensor([1 1 1 1 1], shape=(5,), dtype=int32)

>>> example\_batch=next(iter(train\_ds))[0]

>>> def demo(feature\_column):

... feature\_layer=layers.DenseFratures(feature\_column)

... print(feature\_layer(example\_batch).numpy())

...

>>> photo\_count=feature\_column.numeric\_column('PhotoAmt')

>>> demo(photo\_count)

[[5.]

[5.]

[1.]

[2.]

[8.]]

>>> age=feature\_column.numeric\_column('Age')

>>> age\_buckets=feature\_column.bucketized\_column(age, boundaries=[1,3,5])

>>> demo(age\_buckets)

[[0. 0. 0. 1.]

[0. 0. 0. 1.]

[0. 1. 0. 0.]

[0. 1. 0. 0.]

[0. 0. 0. 1.]]

>>> animal\_type=feature\_column.categorical\_column\_with\_vocabulary\_list('Type', ['Cat','Dog'])

>>> animal\_type\_one\_hot=feature\_column.indicator\_column(animal\_type)

>>> demo(animal\_type\_one\_hot)

[[0. 1.]

[0. 1.]

[0. 1.]

[1. 0.]

[0. 1.]]

>>> breed1=feature\_column.categorical\_column\_with\_vocabulary\_list('Breed1', dataframe.Breed1.unique())

>>> breed1\_embedding=feature\_column.embedding\_column(breed1, dimension=8)

>>> demo(breed1\_embedding)

[[-0.3354375 0.26329488 0.2907019 0.01901461 -0.10193539 0.2896002

0.33973992 -0.5723305 ]

[-0.4052237 -0.01893171 -0.23466891 -0.24360067 -0.5415307 -0.34551075

-0.02385106 0.16807243]

[-0.4052237 -0.01893171 -0.23466891 -0.24360067 -0.5415307 -0.34551075

-0.02385106 0.16807243]

[ 0.314629 -0.21907958 -0.59133685 -0.04529675 -0.35798317 -0.12787533

0.2516734 -0.35414383]

[-0.4052237 -0.01893171 -0.23466891 -0.24360067 -0.5415307 -0.34551075

-0.02385106 0.16807243]]

>>> breed1\_hashed=feature\_column.categorical\_column\_with\_hash\_bucket('Breed1', hash\_bucket\_size=10)

>>> demo(feature\_column.indicator\_column(breed1\_hashed))

[[0. 0. 0. 0. 0. 1. 0. 0. 0. 0.]

[0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]

[0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]

[0. 0. 0. 0. 1. 0. 0. 0. 0. 0.]

[0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]]

>>> crossed\_feature=feature\_column.crossed\_column([age\_buckets, animal\_type], hash\_bucket\_size=10)

>>> crossed\_feature=feature\_column.crossed\_column([age\_buckets, animal\_type], hash\_bucket\_size=2)

>>> demo(feature\_column.indicator\_column(crossed\_feature))

OverflowError: Python int too large to convert to C long

The above exception was the direct cause of the following exception:

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

File "<stdin>", line 3, in demo

File "C:\ProgramData\Anaconda3\envs\tensorflow\lib\site-packages\tensorflow\_core\python\keras\engine\base\_layer.py", line 822, in \_\_call\_\_

outputs = self.call(cast\_inputs, \*args, \*\*kwargs)

File "C:\ProgramData\Anaconda3\envs\tensorflow\lib\site-packages\tensorflow\_core\python\feature\_column\dense\_features.py", line 135, in call

self.\_state\_manager)

File "C:\ProgramData\Anaconda3\envs\tensorflow\lib\site-packages\tensorflow\_core\python\feature\_column\feature\_column\_v2.py", line 4351, in get\_dense\_tensor

return transformation\_cache.get(self, state\_manager)

File "C:\ProgramData\Anaconda3\envs\tensorflow\lib\site-packages\tensorflow\_core\python\feature\_column\feature\_column\_v2.py", line 2615, in get

transformed = column.transform\_feature(self, state\_manager)

File "C:\ProgramData\Anaconda3\envs\tensorflow\lib\site-packages\tensorflow\_core\python\feature\_column\feature\_column\_v2.py", line 4290, in transform\_feature

transformation\_cache, state\_manager)

File "C:\ProgramData\Anaconda3\envs\tensorflow\lib\site-packages\tensorflow\_core\python\feature\_column\feature\_column\_v2.py", line 4142, in get\_sparse\_tensors

transformation\_cache.get(self, state\_manager), None)

File "C:\ProgramData\Anaconda3\envs\tensorflow\lib\site-packages\tensorflow\_core\python\feature\_column\feature\_column\_v2.py", line 2615, in get

transformed = column.transform\_feature(self, state\_manager)

File "C:\ProgramData\Anaconda3\envs\tensorflow\lib\site-packages\tensorflow\_core\python\feature\_column\feature\_column\_v2.py", line 4103, in transform\_feature

hash\_key=self.hash\_key)

File "C:\ProgramData\Anaconda3\envs\tensorflow\lib\site-packages\tensorflow\_core\python\ops\sparse\_ops.py", line 598, in sparse\_cross\_hashed

name=name)

File "C:\ProgramData\Anaconda3\envs\tensorflow\lib\site-packages\tensorflow\_core\python\ops\sparse\_ops.py", line 651, in \_sparse\_cross\_internal

name=name)

File "C:\ProgramData\Anaconda3\envs\tensorflow\lib\site-packages\tensorflow\_core\python\ops\gen\_sparse\_ops.py", line 1048, in sparse\_cross

internal\_type)

SystemError: <built-in function TFE\_Py\_FastPathExecute> returned a result with an error set

>>>

>>> feature\_columns=[]

>>> for header in ['PhotoAmt', 'Fee','Age']:

... feature\_columns.append(feature\_column.numeric\_column(header))

...

>>> age=feature\_column.numeric\_column('Age')

>>>

>>> age\_buckets=feature\_column.bucketized\_column(age, boundaries=[1,2,3,4,5])

>>> feature\_columns.append(age\_buckets)

>>> demo(age\_buckets)

[[0. 0. 0. 0. 0. 1.]

[0. 0. 0. 0. 0. 1.]

[0. 0. 1. 0. 0. 0.]

[0. 1. 0. 0. 0. 0.]

[0. 0. 0. 0. 0. 1.]]

>>> indicator\_column\_names=['Type','Color1','Color2','Gender','MaturitySize', 'FurLength','Vaccinated','Sterilized','Health']

>>> for col\_name in indicator\_column\_names:

... categorical\_column= feature\_column.categorical\_column\_with\_vocabulary\_list(col\_name, dataframe[col\_name].unique())

... indicator\_column=feature\_column.indicator\_column(categorical\_column)

... feature\_columns.append(indicator\_column)

...

>>> breed1=feature\_column.categorical\_column\_with\_vocabulary\_list('Breed1', dataframe.Breed1.unique())

>>> breed1\_embedding=feature\_column.embedding\_column(breed1, dimension=8)

>>> feature\_columns.append(breed1\_embedding)

>>> age\_type\_feature=feature\_column.crossed\_column([age\_buckets, animal\_type], hash\_bucket\_size=100)

>>> feature\_columns.append(feature\_column.indicator\_column(age\_type\_feature))

>>> feature\_layer=tf.keras.layers.DenseFeatures(feature\_columns)

>>> batch\_size=32

>>> train\_ds=df\_to\_dataset(train, batch\_size=32)

>>> val\_ds=df\_to\_dataset(val, shuffle=False, batch\_size=32)

>>> test\_ds=df\_to\_dataset(test, shuffle=False, batch\_size=32)

>>> model=tf.keras.Sequential([

... feature\_layer,

... layers.Dense(128, activation='relu'),

... layers.Dense(128, activation='relu'),

... layers.Dropout(.1),

... layers.Dense(1)

... ])

>>> model.fit(train\_ds, validation\_data=val\_ds, epochs=10)

WARNING:tensorflow:From C:\ProgramData\Anaconda3\envs\tensorflow\lib\site-packages\tensorflow\_core\python\feature\_column\feature\_column\_v2.py:4322: CrossedColumn.\_num\_buckets (from tensorflow.python.feature\_column.feature\_column\_v2) is deprecated and will be removed in a future version.

Instructions for updating:

The old \_FeatureColumn APIs are being deprecated. Please use the new FeatureColumn APIs instead.

Train for 231 steps, validate for 58 steps

Epoch 1/10

231/231 [==============================] - 5s 21ms/step - loss: 0.6887 - accuracy: 0.6892 - val\_loss: 0.5705 - val\_accuracy: 0.7546

Epoch 2/10

231/231 [==============================] - 1s 5ms/step - loss: 0.5666 - accuracy: 0.7135 - val\_loss: 0.6212 - val\_accuracy: 0.6674

Epoch 3/10

231/231 [==============================] - 1s 5ms/step - loss: 0.5317 - accuracy: 0.7242 - val\_loss: 0.5020 - val\_accuracy: 0.7497

Epoch 4/10

231/231 [==============================] - 1s 5ms/step - loss: 0.5088 - accuracy: 0.7298 - val\_loss: 0.5015 - val\_accuracy: 0.7470

Epoch 5/10

231/231 [==============================] - 1s 5ms/step - loss: 0.4986 - accuracy: 0.7301 - val\_loss: 0.5012 - val\_accuracy: 0.7459

Epoch 6/10

231/231 [==============================] - 1s 5ms/step - loss: 0.4927 - accuracy: 0.7326 - val\_loss: 0.5107 - val\_accuracy: 0.7096

Epoch 7/10

231/231 [==============================] - 1s 5ms/step - loss: 0.4871 - accuracy: 0.7422 - val\_loss: 0.5008 - val\_accuracy: 0.7226

Epoch 8/10

231/231 [==============================] - 1s 5ms/step - loss: 0.4843 - accuracy: 0.7489 - val\_loss: 0.4970 - val\_accuracy: 0.7378

Epoch 9/10

231/231 [==============================] - 1s 5ms/step - loss: 0.4773 - accuracy: 0.7502 - val\_loss: 0.4982 - val\_accuracy: 0.7308

Epoch 10/10

231/231 [==============================] - 1s 5ms/step - loss: 0.4741 - accuracy: 0.7464 - val\_loss: 0.4963 - val\_accuracy: 0.7492

<tensorflow.python.keras.callbacks.History object at 0x000001A7F7B48E88>

>>> loss, accuracy=model.evaluate(test\_ds)

73/73 [==============================] - 0s 2ms/step - loss: 0.4899 - accuracy: 0.7621

>>> print('Accuracy', accurary)

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

NameError: name 'accurary' is not defined

>>> print('Accuracy', accuracy)

Accuracy 0.7621317

>>>