Implementing ALEXNET WITH CIFAR100 dataset

Name - Aloukik Aditya

Subject - Machine Learning

Student_id - 1115290

ASSIGNMENT 3 Part 1(Using weights, tranfer learning)

```
from __future__ import print_function
import tensorflow.compat.v1 as tf
from keras.models import Model
import keras
import pandas as pd
from tensorflow.keras.callbacks import TensorBoard
from tensorflow import keras
from keras.datasets import cifar100
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation, Flatten, Input
from keras.layers import Conv2D, MaxPooling2D, ZeroPadding2D
from keras.optimizers import SGD
from keras.regularizers import l2
from keras.callbacks import Callback, LearningRateScheduler, TensorBoard, ModelCheckpoint
from keras.preprocessing.image import ImageDataGenerator
from keras.utils import print_summary, to_categorical
from keras import backend as K
import numpy as np
import matplotlib
import json
from matplotlib import pyplot as plt
from keras.models import Sequential
from keras.optimizers import Adam
from keras.callbacks import ModelCheckpoint
from keras.models import load_model
from keras.layers import Lambda, Conv2D, MaxPooling2D, Dropout, Dense, Flatten, Activation
```

```
import sys
import os
import cv2
from PIL import Image
import numpy as np
from skimage.transform import resize
from skimage import data, io, filters
from skimage.transform import rescale
import time
from keras.layers.normalization import BatchNormalization
from sklearn.model_selection import StratifiedShuffleSplit
```

The default version of TensorFlow in Colab will soon switch to TensorFlow 2.x.
We recommend you upgrade now or ensure your notebook will continue to use TensorFlow 1.x via the `%tensorflow_version 1.x` magic: more info.

```
Using TensorFlow backend
```

```
NAME = "Alexnet_cifar100"
NUM_CLASSES = 100
num_classes = 100
```

```
(x_train, y_train), (x_test, y_test) = cifar100.load_data()
```

```
x_train = np.array([cv2.resize(img, (224, 224), interpolation=cv2.INTER_CUBIC) for img in x_train])#-----------------------changing
x_test = np.array([cv2.resize(img, (224, 224), interpolation=cv2.INTER_CUBIC) for img in x_test])#--------- orignal data iamges are :
                                                                                                  #- there are 50000 testing images so
```

```
print(x_train.shape)
print(x_test.shape)#----printing shape of values
print(y_train.shape)
```

```
(50000, 224, 224, 3)
(10000, 224, 224, 3)
(50000, 1)
```

```
y_train = to_categorical(y_train, NUM_CLASSES)
y_test = to_categorical(y_test, NUM_CLASSES)
```

Double-click (or enter) to edit

```
x_train = x_train.astype(np.float32)#---------------------normalizaing the data#
x_test = x_test.astype(np.float32)
x_train = x_train / 255
x_test = x_test / 255
```

```
#print(x_train.shape)
#print(y_train.shape)
#print(x_test.shape)
#print(y_test.shape)#-----------------------verifying the shapes again

#x = []
#for i in range(len(x_train)):

  # img = cv2.resize(x_train[i], (224, 224))
  # x.append(img)


print(type(x_test))
print(type(x_test))
```

```
<class 'numpy.ndarray'>
```

```
def Alexnet(n_class=100, init_lr = 0.01):#--------------Starting the main architecture of ALEXNET

    # Convolution layer 1
    inputs = Input((224,224,3))
    conv1 = Conv2D(filters=96, input_shape=(224,224,3), kernel_size=(11,11),activation = 'relu',strides=(4,4), padding='same')(inputs
    conv1 = Dropout(0.2)(conv1)

    pool1 = MaxPooling2D(pool_size=(2,2), strides=(2,2), padding='same')(conv1)

    # Convolution layer 2
    conv2 = Conv2D(filters=256,kernel_size=(5,5),activation ='relu',strides=(1,1), padding='same')(pool1)
    conv2 = Dropout(0.2)(conv2)
    pool2 = MaxPooling2D(pool_size=(2,2), strides=(2,2), padding='same')(conv2)

    # Convolution layer 3
    conv3 = Conv2D(filters=384,kernel_size=(3,3),activation = 'relu',strides=(4,4), padding='same')(pool2)
    conv3 = Dropout(0.2)(conv3)
    pool3 = MaxPooling2D(pool_size=(2,2), strides=(2,2), padding='same')(conv3)

    # Convolution layer 4
    conv4 = Conv2D(filters=384,kernel_size=(3,3),activation = 'relu',strides=(4,4), padding='same')(pool3)
    conv4 = Dropout(0.2)(conv4)

    # Convolution layer 5
    conv5 = Conv2D(filters=256,kernel_size=(3,3),activation = 'relu',strides=(4,4), padding='same')(conv4)
    conv5 = Dropout(0.2)(conv5)

    pool5 = MaxPooling2D(pool_size=(2,2), strides=(2,2), padding='same')(conv5)

    conv6 = Flatten() (pool5)
    # FCC Layer 1
    conv6 = Dense(4096) (conv6)
    conv6 = Activation('relu') (conv6)
    conv6 = Dropout(0.5)(conv6)
    # FCC layer 2
    conv7 = Dense(4096) (conv6)
    conv7 = Activation('relu') (conv7)
    conv7 = Dropout(0.5)(conv7)


    # FCC layer 3 #---------------------------There are total 8 layers (Alexnet)
    conv8 = Dense(n_class)(conv7)
    conv8 = Activation('softmax')(conv8)

    model = Model(input=inputs, output=conv8)


    model.compile(loss="categorical_crossentropy", optimizer= Adam(lr=init_lr) , metrics=['accuracy'] )


    return model




model = Alexnet()#---------------Getting model
model.summary()
#model.load_weights('Alexnet_weights.h5', by_name=True)
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:66: The name tf.get_default_g

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:541: The name tf.placeholder

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:4432: The name tf.random_uni

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:148: The name tf.placeholder_

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:3733: calling dropout (from t
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 - keep_prob`.
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:4267: The name tf.nn.max_pool

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/optimizers.py:793: The name tf.train.Optimizer is deprecate

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:3576: The name tf.log is depr

Model: "model_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
input_1 (InputLayer)         (None, 224, 224, 3)       0         
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 56, 56, 96)        34944     
_________________________________________________________________
dropout_1 (Dropout)          (None, 56, 56, 96)        0         
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 28, 28, 96)        0         
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 28, 28, 256)       614656    
_________________________________________________________________
dropout_2 (Dropout)          (None, 28, 28, 256)       0         
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 14, 14, 256)       0         
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 4, 4, 384)         885120    
_________________________________________________________________
dropout_3 (Dropout)          (None, 4, 4, 384)         0         
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 2, 2, 384)         0         
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 1, 1, 384)         1327488   
```

```
conv2d_4 (Conv2D)            (None, 1, 1, 384)         1327488
_________________________________________________________________
dropout_4 (Dropout)          (None, 1, 1, 384)         0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 1, 1, 256)         884992
_________________________________________________________________
dropout_5 (Dropout)          (None, 1, 1, 256)         0
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 1, 1, 256)         0
_________________________________________________________________
flatten_1 (Flatten)          (None, 256)               0
_________________________________________________________________
dense_1 (Dense)              (None, 4096)              1052672
_________________________________________________________________
activation_1 (Activation)    (None, 4096)              0
_________________________________________________________________
dropout_6 (Dropout)          (None, 4096)              0
_________________________________________________________________
dense_2 (Dense)              (None, 4096)              16781312
_________________________________________________________________
activation_2 (Activation)    (None, 4096)              0
_________________________________________________________________
dropout_7 (Dropout)          (None, 4096)              0
_________________________________________________________________
dense_3 (Dense)              (None, 100)               409700
_________________________________________________________________
activation_3 (Activation)    (None, 100)               0
=================================================================
Total params: 21,990,884
Trainable params: 21,990,884
Non-trainable params: 0
_________________________________________________________________
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:45: UserWarning: Update your `Model` call to the Keras 2 API: `Mode
```

```
tensorboard = TensorBoard(log_dir="logs/{}".format(NAME))#------------Tensorboard will be used as log files
checkpoint = ModelCheckpoint('best_model_simple.h5',  #--- only the best model will be forwarded in next epoch
                             monitor='val_loss',
                             verbose=0,
                             save_best_only= True,
                             mode='auto')


model.compile(loss='categorical_crossentropy', #-------------------Model will compile from here
              optimizer=Adam(lr=1.0e-4),
              metrics = ['accuracy'])
average_accuracy = 0
d =0



for i in range(3):
  d=0

  model.load_weights('Alexnet_weights.h5', by_name=True)
  model_details = model.fit(x_train, y_train,#--------------------Putting our cifar100 data in alexnet
                      batch_size = 128, # number of samples
                      epochs = 30, # number of epochs
                      validation_data= (x_test, y_test),
                      callbacks=[tensorboard],
                      verbose=1)
  d=0

  d = model_details.history['val_acc']
  print(d)
  average_accuracy += d[29]
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:190: The name tf.get_default_

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:197: The name tf.ConfigProto 

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:203: The name tf.Session is d

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:207: The name tf.global_varia

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:216: The name tf.is_variable_

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:223: The name tf.variables_in

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow_core/python/ops/math_grad.py:1424: where (from tensorf
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:1033: The name tf.assign_add 

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:1020: The name tf.assign is d

Train on 50000 samples, validate on 10000 samples
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/callbacks.py:1122: The name tf.summary.merge_all is depreca

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/callbacks.py:1125: The name tf.summary.FileWriter is deprec

Epoch 1/30
50000/50000 [==============================] - 53s 1ms/step - loss: 3.3036 - acc: 0.1909 - val_loss: 3.3373 - val_acc: 0.2379
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/callbacks.py:1265: The name tf.Summary is deprecated. Pleas

Epoch 2/30
50000/50000 [==============================] - 50s 994us/step - loss: 3.1522 - acc: 0.2183 - val_loss: 3.1266 - val_acc: 0.2584
Epoch 3/30
50000/50000 [==============================] - 50s 993us/step - loss: 3.0214 - acc: 0.2443 - val_loss: 3.0605 - val_acc: 0.2750
Epoch 4/30
50000/50000 [==============================] - 50s 993us/step - loss: 2.8982 - acc: 0.2668 - val_loss: 2.9616 - val_acc: 0.2978
Epoch 5/30
50000/50000 [==============================] - 50s 992us/step - loss: 2.7748 - acc: 0.2917 - val_loss: 2.9164 - val_acc: 0.3110
Epoch 6/30
50000/50000 [==============================] - 50s 994us/step - loss: 2.6690 - acc: 0.3097 - val_loss: 2.8037 - val_acc: 0.3294
Epoch 7/30
50000/50000 [==============================] - 50s 993us/step - loss: 2.5571 - acc: 0.3352 - val_loss: 2.7327 - val_acc: 0.3293
Epoch 8/30
50000/50000 [==============================] - 50s 995us/step - loss: 2.4433 - acc: 0.3576 - val_loss: 2.6550 - val_acc: 0.3504
```

```
Epoch 9/30
50000/50000 [==============================] - 50s 994us/step - loss: 2.3558 - acc: 0.3751 - val_loss: 2.6282 - val_acc: 0.3501
Epoch 10/30
50000/50000 [==============================] - 50s 994us/step - loss: 2.2528 - acc: 0.3980 - val_loss: 2.5282 - val_acc: 0.3666
Epoch 11/30
50000/50000 [==============================] - 50s 994us/step - loss: 2.1656 - acc: 0.4168 - val_loss: 2.5427 - val_acc: 0.3648
Epoch 12/30
50000/50000 [==============================] - 50s 992us/step - loss: 2.0944 - acc: 0.4296 - val_loss: 2.4936 - val_acc: 0.3748
Epoch 13/30
50000/50000 [==============================] - 50s 998us/step - loss: 2.0139 - acc: 0.4504 - val_loss: 2.4784 - val_acc: 0.3744
Epoch 14/30
50000/50000 [==============================] - 50s 997us/step - loss: 1.9406 - acc: 0.4631 - val_loss: 2.4149 - val_acc: 0.3808
Epoch 15/30
50000/50000 [==============================] - 50s 998us/step - loss: 1.8757 - acc: 0.4775 - val_loss: 2.4356 - val_acc: 0.3767
Epoch 16/30
50000/50000 [==============================] - 50s 997us/step - loss: 1.7996 - acc: 0.4987 - val_loss: 2.3725 - val_acc: 0.3928
Epoch 17/30
50000/50000 [==============================] - 50s 998us/step - loss: 1.7362 - acc: 0.5110 - val_loss: 2.4093 - val_acc: 0.3871
Epoch 18/30
50000/50000 [==============================] - 50s 992us/step - loss: 1.6710 - acc: 0.5271 - val_loss: 2.3922 - val_acc: 0.3911
Epoch 19/30
50000/50000 [==============================] - 50s 992us/step - loss: 1.6218 - acc: 0.5396 - val_loss: 2.4236 - val_acc: 0.3838
Epoch 20/30
50000/50000 [==============================] - 50s 992us/step - loss: 1.5606 - acc: 0.5539 - val_loss: 2.3525 - val_acc: 0.3971
Epoch 21/30
50000/50000 [==============================] - 49s 985us/step - loss: 1.4993 - acc: 0.5663 - val_loss: 2.3549 - val_acc: 0.3972
Epoch 22/30
50000/50000 [==============================] - 48s 951us/step - loss: 1.4488 - acc: 0.5776 - val_loss: 2.4117 - val_acc: 0.3888
Epoch 23/30
50000/50000 [==============================] - 47s 944us/step - loss: 1.4027 - acc: 0.5908 - val_loss: 2.3545 - val_acc: 0.3940
Epoch 24/30
50000/50000 [==============================] - 47s 946us/step - loss: 1.3489 - acc: 0.6047 - val_loss: 2.3784 - val_acc: 0.3924
Epoch 25/30
50000/50000 [==============================] - 47s 947us/step - loss: 1.3166 - acc: 0.6118 - val_loss: 2.3676 - val_acc: 0.4015
Epoch 26/30
50000/50000 [==============================] - 47s 943us/step - loss: 1.2767 - acc: 0.6251 - val_loss: 2.4451 - val_acc: 0.3873
Epoch 27/30
50000/50000 [==============================] - 47s 947us/step - loss: 1.2439 - acc: 0.6327 - val_loss: 2.3641 - val_acc: 0.4024
Epoch 28/30
50000/50000 [==============================] - 47s 948us/step - loss: 1.1940 - acc: 0.6461 - val_loss: 2.4141 - val_acc: 0.3907
Epoch 29/30
50000/50000 [==============================] - 47s 948us/step - loss: 1.1674 - acc: 0.6543 - val_loss: 2.4262 - val_acc: 0.3952
```

```
Epoch 30/30
50000/50000 [==============================] - 47s 944us/step - loss: 1.1268 - acc: 0.6633 - val_loss: 2.4167 - val_acc: 0.3988
[0.2379, 0.2584, 0.275, 0.2978, 0.311, 0.3294, 0.3293, 0.3504, 0.3501, 0.3666, 0.3648, 0.3748, 0.3744, 0.3808, 0.3767, 0.3928, 
Train on 50000 samples, validate on 10000 samples
Epoch 1/30
50000/50000 [==============================] - 47s 947us/step - loss: 3.3025 - acc: 0.1896 - val_loss: 3.3509 - val_acc: 0.2321
Epoch 2/30
50000/50000 [==============================] - 47s 946us/step - loss: 3.1805 - acc: 0.2094 - val_loss: 3.2353 - val_acc: 0.2547
Epoch 3/30
50000/50000 [==============================] - 47s 945us/step - loss: 3.0844 - acc: 0.2327 - val_loss: 3.1950 - val_acc: 0.2635
Epoch 4/30
50000/50000 [==============================] - 47s 943us/step - loss: 2.9968 - acc: 0.2445 - val_loss: 3.1239 - val_acc: 0.2742
Epoch 5/30
50000/50000 [==============================] - 47s 942us/step - loss: 2.9173 - acc: 0.2640 - val_loss: 3.0091 - val_acc: 0.2835
Epoch 6/30
50000/50000 [==============================] - 47s 943us/step - loss: 2.8410 - acc: 0.2786 - val_loss: 2.9128 - val_acc: 0.3049
Epoch 7/30
50000/50000 [==============================] - 47s 938us/step - loss: 2.7597 - acc: 0.2927 - val_loss: 2.8844 - val_acc: 0.3103
Epoch 8/30
50000/50000 [==============================] - 47s 941us/step - loss: 2.6720 - acc: 0.3117 - val_loss: 2.8226 - val_acc: 0.3219
Epoch 9/30
50000/50000 [==============================] - 47s 943us/step - loss: 2.5867 - acc: 0.3284 - val_loss: 2.7621 - val_acc: 0.3354
Epoch 10/30
50000/50000 [==============================] - 47s 949us/step - loss: 2.5002 - acc: 0.3452 - val_loss: 2.7049 - val_acc: 0.3393
Epoch 11/30
50000/50000 [==============================] - 47s 949us/step - loss: 2.4232 - acc: 0.3636 - val_loss: 2.6519 - val_acc: 0.3448
Epoch 12/30
50000/50000 [==============================] - 48s 951us/step - loss: 2.3424 - acc: 0.3775 - val_loss: 2.6182 - val_acc: 0.3536
Epoch 13/30
50000/50000 [==============================] - 47s 942us/step - loss: 2.2754 - acc: 0.3913 - val_loss: 2.5765 - val_acc: 0.3590
Epoch 14/30
50000/50000 [==============================] - 47s 943us/step - loss: 2.1888 - acc: 0.4091 - val_loss: 2.5478 - val_acc: 0.3653
Epoch 15/30
50000/50000 [==============================] - 47s 946us/step - loss: 2.1145 - acc: 0.4262 - val_loss: 2.5341 - val_acc: 0.3679
Epoch 16/30
50000/50000 [==============================] - 47s 942us/step - loss: 2.0500 - acc: 0.4394 - val_loss: 2.4634 - val_acc: 0.3757
Epoch 17/30
50000/50000 [==============================] - 47s 944us/step - loss: 1.9823 - acc: 0.4544 - val_loss: 2.4598 - val_acc: 0.3740
Epoch 18/30
50000/50000 [==============================] - 47s 946us/step - loss: 1.9196 - acc: 0.4692 - val_loss: 2.4377 - val_acc: 0.3826
Epoch 19/30
50000/50000 [==============================] - 47s 948us/step - loss: 1.8424 - acc: 0.4858 - val_loss: 2.4412 - val_acc: 0.3767
```

```
Epoch 20/30
50000/50000 [==============================] - 48s 951us/step - loss: 1.7964 - acc: 0.4966 - val_loss: 2.4067 - val_acc: 0.3847
Epoch 21/30
50000/50000 [==============================] - 48s 958us/step - loss: 1.7338 - acc: 0.5093 - val_loss: 2.4424 - val_acc: 0.3760
Epoch 22/30
50000/50000 [==============================] - 48s 958us/step - loss: 1.6763 - acc: 0.5266 - val_loss: 2.3957 - val_acc: 0.3911
Epoch 23/30
50000/50000 [==============================] - 48s 952us/step - loss: 1.6158 - acc: 0.5400 - val_loss: 2.3867 - val_acc: 0.3925
Epoch 24/30
50000/50000 [==============================] - 48s 955us/step - loss: 1.5747 - acc: 0.5507 - val_loss: 2.3660 - val_acc: 0.3954
Epoch 25/30
50000/50000 [==============================] - 47s 948us/step - loss: 1.5209 - acc: 0.5618 - val_loss: 2.4179 - val_acc: 0.3866
Epoch 26/30
50000/50000 [==============================] - 47s 945us/step - loss: 1.4731 - acc: 0.5742 - val_loss: 2.4059 - val_acc: 0.3895
Epoch 27/30
50000/50000 [==============================] - 47s 943us/step - loss: 1.4256 - acc: 0.5862 - val_loss: 2.4506 - val_acc: 0.3877
Epoch 28/30
50000/50000 [==============================] - 47s 944us/step - loss: 1.3828 - acc: 0.5980 - val_loss: 2.3831 - val_acc: 0.4000
Epoch 29/30
50000/50000 [==============================] - 47s 946us/step - loss: 1.3341 - acc: 0.6095 - val_loss: 2.4475 - val_acc: 0.3885
Epoch 30/30
50000/50000 [==============================] - 47s 946us/step - loss: 1.3086 - acc: 0.6163 - val_loss: 2.4135 - val_acc: 0.3938
[0.2321, 0.2547, 0.2635, 0.2742, 0.2835, 0.3049, 0.3103, 0.3219, 0.3354, 0.3393, 0.3448, 0.3536, 0.359, 0.3653, 0.3679, 0.3757,
Train on 50000 samples, validate on 10000 samples
Epoch 1/30
50000/50000 [==============================] - 47s 944us/step - loss: 3.3050 - acc: 0.1902 - val_loss: 3.3159 - val_acc: 0.2311
Epoch 2/30
50000/50000 [==============================] - 47s 945us/step - loss: 3.1833 - acc: 0.2118 - val_loss: 3.2252 - val_acc: 0.2517
Epoch 3/30
50000/50000 [==============================] - 47s 942us/step - loss: 3.0878 - acc: 0.2312 - val_loss: 3.1660 - val_acc: 0.2667
Epoch 4/30
50000/50000 [==============================] - 47s 944us/step - loss: 3.0056 - acc: 0.2454 - val_loss: 3.1195 - val_acc: 0.2649
Epoch 5/30
50000/50000 [==============================] - 47s 943us/step - loss: 2.9119 - acc: 0.2619 - val_loss: 2.9887 - val_acc: 0.2891
Epoch 6/30
50000/50000 [==============================] - 47s 943us/step - loss: 2.8351 - acc: 0.2781 - val_loss: 2.9641 - val_acc: 0.2965
Epoch 7/30
50000/50000 [==============================] - 47s 944us/step - loss: 2.7419 - acc: 0.2987 - val_loss: 2.8598 - val_acc: 0.3194
Epoch 8/30
50000/50000 [==============================] - 47s 942us/step - loss: 2.6544 - acc: 0.3164 - val_loss: 2.7806 - val_acc: 0.3257
Epoch 9/30
50000/50000 [==============================] - 47s 942us/step - loss: 2.5712 - acc: 0.3323 - val_loss: 2.7404 - val_acc: 0.3339
```

```
Epoch 10/30
50000/50000 [==============================] - 47s 942us/step - loss: 2.4878 - acc: 0.3475 - val_loss: 2.6476 - val_acc: 0.3531
Epoch 11/30
50000/50000 [==============================] - 47s 931us/step - loss: 2.3901 - acc: 0.3704 - val_loss: 2.6872 - val_acc: 0.3413
Epoch 12/30
50000/50000 [==============================] - 47s 930us/step - loss: 2.3164 - acc: 0.3830 - val_loss: 2.5994 - val_acc: 0.3538
Epoch 13/30
50000/50000 [==============================] - 47s 930us/step - loss: 2.2327 - acc: 0.4017 - val_loss: 2.5335 - val_acc: 0.3668
Epoch 14/30
50000/50000 [==============================] - 47s 932us/step - loss: 2.1558 - acc: 0.4141 - val_loss: 2.4656 - val_acc: 0.3808
Epoch 15/30
50000/50000 [==============================] - 46s 929us/step - loss: 2.0782 - acc: 0.4353 - val_loss: 2.4806 - val_acc: 0.3739
Epoch 16/30
50000/50000 [==============================] - 46s 922us/step - loss: 2.0180 - acc: 0.4468 - val_loss: 2.4726 - val_acc: 0.3760
Epoch 17/30
50000/50000 [==============================] - 46s 928us/step - loss: 1.9423 - acc: 0.4666 - val_loss: 2.4036 - val_acc: 0.3922
Epoch 18/30
50000/50000 [==============================] - 46s 924us/step - loss: 1.8742 - acc: 0.4822 - val_loss: 2.4378 - val_acc: 0.3840
Epoch 19/30
50000/50000 [==============================] - 46s 921us/step - loss: 1.8031 - acc: 0.4957 - val_loss: 2.3986 - val_acc: 0.3936
Epoch 20/30
50000/50000 [==============================] - 46s 917us/step - loss: 1.7425 - acc: 0.5094 - val_loss: 2.3580 - val_acc: 0.4000
Epoch 21/30
50000/50000 [==============================] - 46s 918us/step - loss: 1.6805 - acc: 0.5233 - val_loss: 2.3373 - val_acc: 0.3983
Epoch 22/30
50000/50000 [==============================] - 46s 926us/step - loss: 1.6263 - acc: 0.5378 - val_loss: 2.3391 - val_acc: 0.4041
Epoch 23/30
50000/50000 [==============================] - 46s 921us/step - loss: 1.5720 - acc: 0.5522 - val_loss: 2.4015 - val_acc: 0.3905
Epoch 24/30
50000/50000 [==============================] - 46s 922us/step - loss: 1.5173 - acc: 0.5647 - val_loss: 2.3716 - val_acc: 0.3956
Epoch 25/30
50000/50000 [==============================] - 46s 924us/step - loss: 1.4771 - acc: 0.5753 - val_loss: 2.3439 - val_acc: 0.3996
Epoch 26/30
50000/50000 [==============================] - 46s 926us/step - loss: 1.4325 - acc: 0.5865 - val_loss: 2.3619 - val_acc: 0.4028
Epoch 27/30
50000/50000 [==============================] - 46s 919us/step - loss: 1.3747 - acc: 0.6005 - val_loss: 2.3393 - val_acc: 0.4046
Epoch 28/30
50000/50000 [==============================] - 46s 919us/step - loss: 1.3457 - acc: 0.6079 - val_loss: 2.3616 - val_acc: 0.4043
Epoch 29/30
50000/50000 [==============================] - 47s 933us/step - loss: 1.3073 - acc: 0.6174 - val_loss: 2.3767 - val_acc: 0.3992
Epoch 30/30
50000/50000 [==============================] - 47s 937us/step - loss: 1.2561 - acc: 0.6308 - val_loss: 2.4188 - val_acc: 0.4000
```

```
     [0.2311, 0.2517, 0.2667, 0.2649, 0.2891, 0.2965, 0.3194, 0.3257, 0.3339, 0.3531, 0.3413, 0.3538, 0.3668, 0.3808, 0.3739, 0.376,
```

```
final_average_accuracy = average_accuracy/3.
print(final_average_accuracy)
```

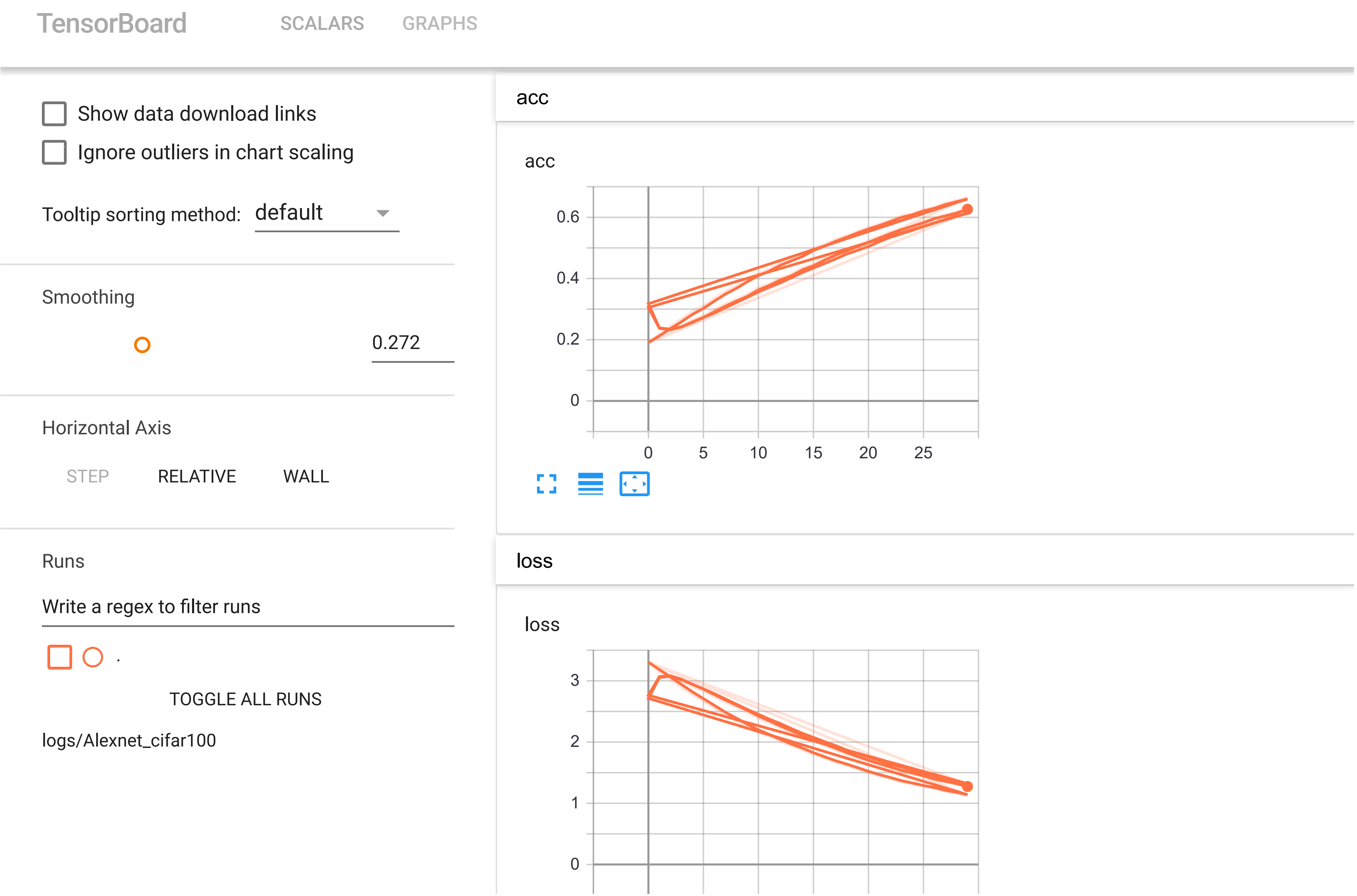
```
0.39753333333333335
```

```
print(d[29])
```

```
0.4
```

```
%load_ext tensorboard
%tensorboard --logdir logs/Alexnet_cifar100
```

```
The tensorboard extension is already loaded. To reload it, use:
  %reload_ext tensorboard
Reusing TensorBoard on port 6006 (pid 2753), started 0:33:15 ago. (Use '!kill 2753' to kill it.)
```



☐ Show data download links

☐ Ignore outliers in chart scaling

Tooltip sorting method: default

Smoothing 0.272

Horizontal Axis

Runs

Write a regex to filter runs

logs/Alexnet_cifar100

```
hist_df = pd.DataFrame(model_details.history)
hist_csv_file = 'history.csv'
with open(hist_csv_file, mode='w') as f:
    hist_df.to_csv(f)
```

```
!jupyter nbconvert --to latex Alexnet Cifar100.ipynb
```

```
[NbConvertApp] WARNING | pattern u'Alexnet' matched no files
[NbConvertApp] WARNING | pattern u'Cifar100.ipynb' matched no files
This application is used to convert notebook files (*.ipynb) to various other
formats.

WARNING: THE COMMANDLINE INTERFACE MAY CHANGE IN FUTURE RELEASES.

Options
-------

Arguments that take values are actually convenience aliases to full
Configurables, whose aliases are listed on the help line. For more information
on full configurables, see '--help-all'.

--execute
    Execute the notebook prior to export.
--allow-errors
    Continue notebook execution even if one of the cells throws an error and include the error message in the cell output (the 
--no-input
    Exclude input cells and output prompts from converted document.
    This mode is ideal for generating code-free reports.
--stdout
    Write notebook output to stdout instead of files.
--stdin
    read a single notebook file from stdin. Write the resulting notebook with default basename 'notebook.*'
--inplace
    Run nbconvert in place, overwriting the existing notebook (only
    relevant when converting to notebook format)
-y
    Answer yes to any questions instead of prompting.
--clear-output
    Clear output of current file and save in place,
    overwriting the existing notebook.
--debug
    set log level to logging.DEBUG (maximize logging output)
--no-prompt
    Exclude input and output prompts from converted document.
--generate-config
    generate default config file
--nbformat=<Enum> (NotebookExporter.nbformat_version)
    Default: 4
    Choices: [1, 2, 3, 4]
```

```
      Choices: [1, 2, 3, 4]
      The nbformat version to write. Use this to downgrade notebooks.
    --output-dir=<Unicode> (FilesWriter.build_directory)
      Default: ''
      Directory to write output(s) to. Defaults to output to the directory of each
      notebook. To recover previous default behaviour (outputting to the current
      working directory) use . as the flag value.
    --writer=<DottedObjectName> (NbConvertApp.writer_class)
      Default: 'FilesWriter'
      Writer class used to write the  results of the conversion
    --log-level=<Enum> (Application.log_level)
      Default: 30
      Choices: (0, 10, 20, 30, 40, 50, 'DEBUG', 'INFO', 'WARN', 'ERROR', 'CRITICAL')
      Set the log level by value or name.
    --reveal-prefix=<Unicode> (SlidesExporter.reveal_url_prefix)
      Default: u''
      The URL prefix for reveal.js (version 3.x). This defaults to the reveal CDN,
      but can be any url pointing to a copy  of reveal.js.
      For speaker notes to work, this must be a relative path to a local  copy of
      reveal.js: e.g., "reveal.js".
      If a relative path is given, it must be a subdirectory of the current
      directory (from which the server is run).
      See the usage documentation
      (https://nbconvert.readthedocs.io/en/latest/usage.html#reveal-js-html-
      slideshow) for more details.
    --to=<Unicode> (NbConvertApp.export_format)
      Default: 'html'
      The export format to be used, either one of the built-in formats
      ['asciidoc', 'custom', 'html', 'latex', 'markdown', 'notebook', 'pdf',
      'python', 'rst', 'script', 'slides'] or a dotted object name that represents
      the import path for an `Exporter` class
    --template=<Unicode> (TemplateExporter.template_file)
      Default: u''
      Name of the template file to use
    --output=<Unicode> (NbConvertApp.output_base)
      Default: ''
      overwrite base name use for output files. can only be used when converting
      one notebook at a time.
    --post=<DottedOrNone> (NbConvertApp.postprocessor_class)
      Default: u''
      PostProcessor class used to write the results of the conversion
    --config=<Unicode> (JupyterApp.config_file)
      Default: u''
```

```
        Full path of a config file.

    To see all available configurables, use `--help-all`

    Examples
    --------

        The simplest way to use nbconvert is

        > jupyter nbconvert mynotebook.ipynb

        which will convert mynotebook.ipynb to the default format (probably HTML).

        You can specify the export format with `--to`.
        Options include ['asciidoc', 'custom', 'html', 'latex', 'markdown', 'notebook', 'pdf', 'python', 'rst', 'script', 'slides']

        > jupyter nbconvert --to latex mynotebook.ipynb

        Both HTML and LaTeX support multiple output templates. LaTeX includes
        'base', 'article' and 'report'.  HTML includes 'basic' and 'full'. You
        can specify the flavor of the format used.

        > jupyter nbconvert --to html --template basic mynotebook.ipynb

        You can also pipe the output to stdout, rather than a file

        > jupyter nbconvert mynotebook.ipynb --stdout

        PDF is generated via latex

        > jupyter nbconvert mynotebook.ipynb --to pdf

        You can get (and serve) a Reveal.js-powered slideshow

        > jupyter nbconvert myslides.ipynb --to slides --post serve

        Multiple notebooks can be given at the command line in a couple of
        different ways:

        > jupyter nbconvert notebook*.ipynb
        > jupyter nbconvert notebook1.ipynb notebook2.ipynb
        or you can specify the notebooks list in a config file, containing::

            c.NbConvertApp.notebooks = ["my_notebook.ipynb"]

        > jupyter nbconvert --config mycfg.py
```

New Section