Spring 2021

Home

Announcements

Modules

Discussions

Grades

Quizzes

People

UW Libraries

Info and Help

Zoom

UW Resources

Assignment 5: ConvNets (Part 2) A+

New Attempt

Due May 11 by 5:59pm   **Points** 6   **Submitting** a text entry box or a file upload

Submission

✓ **Submitted!**

May 11 at 1:34pm

Submission Details

Download HW05.txt

Grade: 6 (6 pts possible)

Graded Anonymously: no

Comments:
No Comments

Question 5 (1 point)

Suppose we define a convolutional network as shown below.

```python
# python mnist-cnn.py sgd
# python mnist-cnn.py rmsprop
# python mnist-cnn.py adam

import sys
from tensorflow.keras import callbacks, datasets, layers, models, optimizers

optimizer_name = sys.argv[1]
model_name = "mnist-" + optimizer_name

(x_train, y_train), (x_test, y_test) = datasets.mnist.load_data()
x_train = x_train.reshape((x_train.shape[0], x_train.shape[1], x_train.shape[2],
                           1)).astype("float32") / 255.0
x_test = x_test.reshape((x_test.shape[0], x_test.shape[1], x_test.shape[2],
                         1)).astype("float32") / 255.0

model = models.Sequential()
model.add(layers.Conv2D(filters = 16, kernel_size = (3, 3), activation = "relu",
                        input_shape = (28,28,1)))
model.add(layers.Conv2D(filters = 32, kernel_size = (3, 3), activation = "relu"))
model.add(layers.MaxPooling2D(pool_size = (2, 2)))
model.add(layers.Dropout(0.25))
model.add(layers.Flatten())
model.add(layers.Dense(units = 128, activation = "relu"))
model.add(layers.Dropout(0.50))
model.add(layers.Dense(y_train.max() + 1, activation = "softmax"))
model.compile(loss = "sparse_categorical_crossentropy", optimizer = optimizer_name,
              metrics = ["accuracy"])

model.fit(x_train, y_train, batch_size = 128, epochs = 16, validation_split = 0.1,
          callbacks = [ callbacks.EarlyStopping(monitor = "val_accuracy", patience = 8,
                                                restore_best_weights = True) ])
model.summary()
model.save(model_name)

model = models.load_model(model_name)
model.evaluate(x_test, y_test)
```

a) What is the output shape of the second Conv2D() layer?

b) How many parameters are there in the second Conv2D() layer?

c) Assuming your model uses single precision (32-bit floating point numbers) for parameters, what will be the size of the second Conv2D() layer in bytes?

d) When we look at the size of mnist-optimizer/variables/variables.data-00000-of-00001, we see the following file sizes for the 3 optimizers:

```
sgd: 2,388,926
rmsprop: 4,774,349
adam: 7,159,495
```

Why would the size of a model trained with optimizer="rmsprop" be about twice the size of the model trained with optimizer="sgd"?

Hint: Check the RMSprop slide from the first night.

Model 5 (5 points)

Please navigate to the following URL and click the "I understand and agree" button to accept the invitation for this Kaggle task:

https://www.kaggle.com/c/ml530-2021-sp-imagenet/data

Activate the conda environment on your VM:

```
conda activate py37_tensorflow
```

Download the data and create tensors for the images:

```
kaggle competitions download ml530-2021-sp-imagenet
wget https://www.cross-entropy.net/ML530/imagenet-tensors.py.txt
python imagenet-tensors.py.txt
```

Download and run the sample off-the-shelf prediction script [note the validation accuracy and compare it to the validation accuracy for the imagenet-transfer.py.txt script (which you'll run next)]:

```
wget https://www.cross-entropy.net/ML530/tiny_imagenet_names.txt
wget https://www.cross-entropy.net/ML530/imagenet-off-the-shelf.py.txt
python imagenet-off-the-shelf.py.txt
```

Download and run the sample fine-tuning script:

```
wget https://www.cross-entropy.net/ML530/imagenet-transfer.py.txt
python imagenet-transfer.py.txt
```

Note: This script will take around 120 minutes to complete. Make sure you have enough hours available on the VM to complete training. Send email to sadm_rudy514@uw.edu if you need additional hours. If you'd like, you can try changing SIZE_INDEX = 0 to SIZE_INDEX = 4 and EfficientNetB0 to EfficientNetB4.

Upload your predictions to kaggle:

```
kaggle competitions submit ml530-2021-sp-imagenet -f predictions.csv -m "imagenet submission"
kaggle competitions leaderboard ml530-2021-sp-imagenet -s
```

‹ Previous   Next ›