Cooper Tush

Dr. Sathyanarayanan Aakur

CS-4783-65788

October 31st, 2022

Machine Learning Assignment 3 Question 2

1.

Changing the learning rate of the optimizer will result in smaller or larger updates on the parameters. In other words, we change the learning rate to tell the model how aggressive we want to be with updating the total number of all the parameters.

A bigger learning rate will result in smaller loss and more accurate data. I tested this with learning rates of 0.1 and 0.0001 and got better results with 0.1, Yet, we must be careful as a learning rate that is too big can make the model converge too quickly resulting in poor performances.

2.

Batch size controls the accuracy of the estimate of the error gradient when training any kind of neural network (including CNNs). A larger catch size will increase the accuracy of our model/data.

To explain how I got this answer, I tested it with two values for the batch size 16, 32 and 64. The larger value of 64 resulted in the accuracy being larger.

3.

Model: "sequential"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

conv2d (Conv2D) (None, 28, 28, 6) 456

max\_pooling2d (MaxPooling2D (None, 14, 14, 6) 0

)

conv2d\_1 (Conv2D) (None, 10, 10, 16) 2416

max\_pooling2d\_1 (MaxPooling (None, 5, 5, 16) 0

2D)

conv2d\_2 (Conv2D) (None, 1, 1, 120) 48120

flatten (Flatten) (None, 120) 0

dense (Dense) (None, 84) 10164

dense\_1 (Dense) (None, 10) 850

=================================================================

Total params: 62,006

Trainable params: 62,006

Non-trainable params: 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

I found two test sets with interesting results. Both had learning rates of 0.1 but different batch sizes. The first one had a batch size of 32 which gave it high accuracy and low loss, but high val\_loss and low val\_accuracy:

Epoch 1/25

1563/1563 [==============================] - 30s 18ms/step - loss: 1.8892 - accuracy: 0.3175 - val\_loss: 1.7455 - val\_accuracy: 0.3748

Epoch 2/25

1563/1563 [==============================] - 26s 17ms/step - loss: 1.5738 - accuracy: 0.4296 - val\_loss: 1.5368 - val\_accuracy: 0.4555

Epoch 3/25

1563/1563 [==============================] - 28s 18ms/step - loss: 1.4462 - accuracy: 0.4856 - val\_loss: 1.4827 - val\_accuracy: 0.4772

Epoch 4/25

1563/1563 [==============================] - 26s 17ms/step - loss: 1.3518 - accuracy: 0.5193 - val\_loss: 1.3529 - val\_accuracy: 0.5226

Epoch 5/25

1563/1563 [==============================] - 27s 17ms/step - loss: 1.2789 - accuracy: 0.5506 - val\_loss: 1.4829 - val\_accuracy: 0.5093

Epoch 6/25

1563/1563 [==============================] - 27s 17ms/step - loss: 1.2158 - accuracy: 0.5714 - val\_loss: 1.3302 - val\_accuracy: 0.5379

Epoch 7/25

1563/1563 [==============================] - 26s 17ms/step - loss: 1.1744 - accuracy: 0.5880 - val\_loss: 1.2898 - val\_accuracy: 0.5559

Epoch 8/25

1563/1563 [==============================] - 27s 17ms/step - loss: 1.1288 - accuracy: 0.6027 - val\_loss: 1.2818 - val\_accuracy: 0.5639

Epoch 9/25

1563/1563 [==============================] - 27s 17ms/step - loss: 1.0931 - accuracy: 0.6137 - val\_loss: 1.2965 - val\_accuracy: 0.5613

Epoch 10/25

1563/1563 [==============================] - 27s 17ms/step - loss: 1.0604 - accuracy: 0.6266 - val\_loss: 1.3208 - val\_accuracy: 0.5624

Epoch 11/25

1563/1563 [==============================] - 27s 18ms/step - loss: 1.0336 - accuracy: 0.6365 - val\_loss: 1.3294 - val\_accuracy: 0.5526

Epoch 12/25

1563/1563 [==============================] - 26s 17ms/step - loss: 0.9981 - accuracy: 0.6489 - val\_loss: 1.3507 - val\_accuracy: 0.5511

Epoch 13/25

1563/1563 [==============================] - 27s 17ms/step - loss: 0.9717 - accuracy: 0.6573 - val\_loss: 1.3391 - val\_accuracy: 0.5594

Epoch 14/25

1563/1563 [==============================] - 27s 17ms/step - loss: 0.9529 - accuracy: 0.6641 - val\_loss: 1.4003 - val\_accuracy: 0.5500

Epoch 15/25

1563/1563 [==============================] - 27s 17ms/step - loss: 0.9309 - accuracy: 0.6696 - val\_loss: 1.3711 - val\_accuracy: 0.5603

Epoch 16/25

1563/1563 [==============================] - 27s 17ms/step - loss: 0.9076 - accuracy: 0.6783 - val\_loss: 1.3585 - val\_accuracy: 0.5681

Epoch 17/25

1563/1563 [==============================] - 27s 17ms/step - loss: 0.8857 - accuracy: 0.6869 - val\_loss: 1.4036 - val\_accuracy: 0.5582

Epoch 18/25

1563/1563 [==============================] - 27s 17ms/step - loss: 0.8675 - accuracy: 0.6939 - val\_loss: 1.4620 - val\_accuracy: 0.5600

Epoch 19/25

1563/1563 [==============================] - 27s 17ms/step - loss: 0.8509 - accuracy: 0.7007 - val\_loss: 1.5031 - val\_accuracy: 0.5611

Epoch 20/25

1563/1563 [==============================] - 27s 17ms/step - loss: 0.8375 - accuracy: 0.7044 - val\_loss: 1.5452 - val\_accuracy: 0.5446

Epoch 21/25

1563/1563 [==============================] - 27s 17ms/step - loss: 0.8208 - accuracy: 0.7106 - val\_loss: 1.5085 - val\_accuracy: 0.5510

Epoch 22/25

1563/1563 [==============================] - 27s 17ms/step - loss: 0.7966 - accuracy: 0.7205 - val\_loss: 1.5177 - val\_accuracy: 0.5606

Epoch 23/25

1563/1563 [==============================] - 27s 18ms/step - loss: 0.7996 - accuracy: 0.7210 - val\_loss: 1.4858 - val\_accuracy: 0.5621

Epoch 24/25

1563/1563 [==============================] - 28s 18ms/step - loss: 0.7851 - accuracy: 0.7258 - val\_loss: 1.5997 - val\_accuracy: 0.5568

Epoch 25/25

1563/1563 [==============================] - 27s 17ms/step - loss: 0.7704 - accuracy: 0.7295 - val\_loss: 1.6674 - val\_accuracy: 0.5471

313/313 [==============================] - 2s 7ms/step - loss: 1.6674 - accuracy: 0.5471

LeNet Model Test Accuracy: 0.5471000075340271

Meanwhile, the other test set which had a batch size of 16 had the exact opposite, high loss and low accuracy but low val\_loss and high val\_accuracy:

