CS7150.37338.202130 Homework 1

Eda Aydin Oktay, Aishwarya Vantipuli

TOTAL POINTS

3.65 / 4

QUESTION 1

1 Question 1 1 / 1

✓ **- 0 pts Correct**

- 0.1 pts An unclear conclusion

- 0.25 pts Unclear explanation for more than one channel.

- 0.1 pts Unclear conclusion for 1 channel case

- 0.1 pts Not dependent on the layers

Need specify when each case

QUESTION 2

Question 2 2 pts

2.1 Part a) 0.65 / 1

- 0 pts Correct

- 0.05 pts Neurons should be connected

- 0.05 pts Figure must include the dimension size of each layer

- 0.5 pts Missing training procedure

- 0.05 pts Missing optimizer

✓ **- 0.05 pts Missing learning rate**

✓ **- 0.3 pts Mising figure of architecture**

- 0.15 pts Architecutre must have 4 layerd

2.2 Part b) 1 / 1

✓ **- 0 pts Correct**

- 0.05 pts Missing axis labels

- 0.25 pts Missing Training Accuracy versus Epochs

QUESTION 3

3 Question 3 1 / 1

✓ **- 0 pts Correct**

- 0.05 pts State conclusion

CS 7150: Deep Learning — Spring 2021— Paul Hand

HW 1
Due: Friday February 12, 2020 at 5:00 PM Eastern time via Gradescope

Names: [Aishwarya Vantipuli, Eda Aydin Oktay]

You will submit this homework in groups of 2. You may consult any and all resources. Note that some of these questions are somewhat vague by design. Part of your task is to make reasonable decisions in interpreting the questions. Your responses should convey understanding, be written with an appropriate amount of precision, and be succinct. Where possible, you should make precise statements. For questions that require coding, you may either type your results with figures into this tex file, or you may append a pdf of output of a Jupyter notebook that is organized similarly. You may use code available on the internet as a starting point.

Question 1. *Is a* 1×1 *convolution operation the same as scaling the input by a single scalar constant? Explain. If the answer is sometimes yes, then make sure to explain when it is and when it isn't.*

Response:

Given an input image: H x W x C *(when C = 1)*, applying 1x1xC *(C=1)* convoulution is just multiply the each pixel with single scalar constant which results in H x W x C *(C=1)*.

When there are multiple input channels, 1 x 1 x C convolution (C > 1) acts as a single neuron in a fully connected layer and will take element wise product of input C and 1x1xC conv and apply ReLU which results in HxWxC. This is helpful when we want to add non linearity to network by keeping output channels same as input channels.

When there are multiple input channels(C) and we want to shrink them to some c, 1x1xC conv is applied c times which results in HxWxc

1x1 conv is useful to change (increase/decrease) dimensionality of channels or keeps same.

1 Question 1 1 / 1

✓ **- 0 pts Correct**

- 0.1 pts An unclear conclusion

- 0.25 pts Unclear explanation for more than one channel.

- 0.1 pts Unclear conclusion for 1 channel case

- 0.1 pts Not dependent on the layers

Need specify when each case

Question 2. *In this problem, you will train a classifier on the MNIST dataset. You can find this dataset in TorchVision. Train a fully-connected neural network with 2 hidden layers and ReLU activations.*

(a) Clearly convey the architecture (by providing a figure) and training procedure you used.

Response:

Building a Fully Connected Neural network with 2 hidden layers:

First, a sequential model is used with Keras TensorFlow background. Next, the first layer is hidden with 512 nodes.Each node will receive an element from each input vector (784 inputs) and apply some weight and bias to it.

Nonlinear activation function ReLU is applied to the output which checks each output and determined whether a neuron is activated/fired based on positive or negative sign.Dropout is added to prevent overfitting.

In second hidden layer, each of 512 nodes receive input from 512 nodes of previous layer.

Final layer is has 10 nodes , which represent classes of MNIST; is fully connected to previous layer containing 512 nodes. Softmax Activation is used which determines the probability of input belonging to one of the classes.

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_3 (Dense)              (None, 512)               401920
_________________________________________________________________
activation_3 (Activation)    (None, 512)               0
_________________________________________________________________
dropout_2 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_4 (Dense)              (None, 512)               262656
_________________________________________________________________
activation_4 (Activation)    (None, 512)               0
_________________________________________________________________
dropout_3 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_5 (Dense)              (None, 10)                5130
_________________________________________________________________
activation_5 (Activation)    (None, 10)                0
=================================================================
Total params: 669,706
Trainable params: 669,706
Non-trainable params: 0
```

Figure 1: Network Architechture

Training

Model is compiled using categorical cross entropy loss and with Adams optimizer. Trained with batch size of 128 in 30 epochs. The observed test accuracy is 0.98.

2.1 Part a) 0.65 / 1

- **0 pts** Correct
- **0.05 pts** Neurons should be connected
- **0.05 pts** Figure must include the dimension size of each layer
- **0.5 pts** Missing training procedure
- **0.05 pts** Missing optimizer

✓ - **0.05 pts** **Missing learning rate**

✓ - **0.3 pts** **Mising figure of architecture**

- **0.15 pts** Architecutre must have 4 layerd

```
Epoch 1/5
469/469 [==============================] - 8s 15ms/step - loss: 0.4368 - accuracy: 0.8693
Epoch 2/5
469/469 [==============================] - 7s 15ms/step - loss: 0.1045 - accuracy: 0.9683
Epoch 3/5
469/469 [==============================] - 7s 15ms/step - loss: 0.0760 - accuracy: 0.9761
Epoch 4/5
469/469 [==============================] - 7s 15ms/step - loss: 0.0546 - accuracy: 0.9827
Epoch 5/5
469/469 [==============================] - 7s 15ms/step - loss: 0.0408 - accuracy: 0.9868
313/313 [==============================] - 1s 3ms/step - loss: 0.0626 - accuracy: 0.9815
Test score: 0.062604434788227 08
Test accuracy: 0.9815000295639038
```

Figure 2: Training fully connected neural network

(b) Plot training loss, training accuracy, and test accuracy vs. iteration count.

Response:

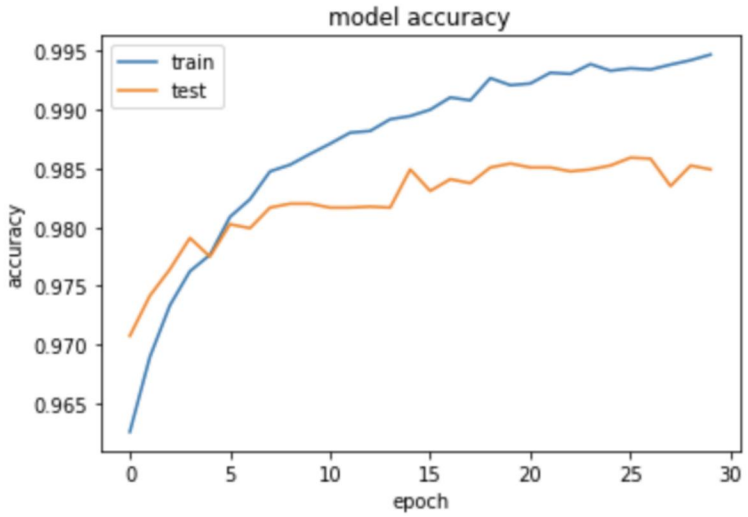

Figure 3: Accuracy of test and training sets

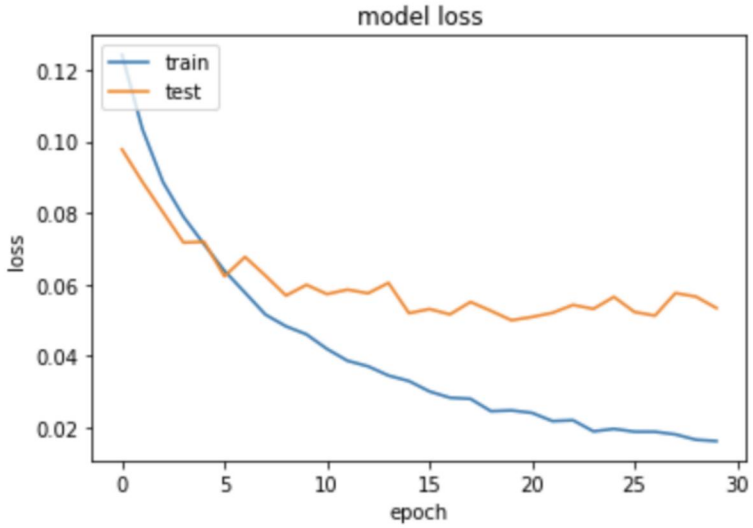

Figure 4: Loss for of test and training sets

Question 3. *Download an AlexNet and a ResNet101 model that have been trained on ImageNet. You can find these models in TorchVision. Using a camera, take a picture of an object that belongs to one of the ImageNet classes. Use both models to classify the image. Output the top 5 predicted classes and their corresponding probabilities, according to each model. Were the models correct?*

Response:

Figure 5: A corn image from corn class of ImageNet Dataset

AlexNet Prediction Probabilities of Top 5 Classes

(”987: ’corn’,”, 0.7743),
(”998: ’ear, spike, capitulum’,”, 0.2232),

2.2 Part b) 1 / 1

✓ **- 0 pts** **Correct**

- 0.05 pts Missing axis labels

- 0.25 pts Missing Training Accuracy versus Epochs

(”582: ’grocery store, grocery, food market, market’,”, 0.0004),
(”905: ’window shade’,”, 0.00017924334679264575),
(”939: ’zucchini, courgette’,”, 0.0001)

ResNet Prediction Probabilities of Top 5 Classes

’ear, spike, capitulum’, 0.527),
(’corn’, 0.468),
(’hamper’, 0.0007),
(’grocery store, grocery, food market, market’, 0.0006),
(’pineapple, ananas’, 0.0003)

AlexNet predicted the image correctly as corn with a probability of 0.77. ResNet has predicted the image as ear with 0.52 chance, which was not correct.(top 1). However, corn was in top 2 with 0.46 i.e. It is interesting to find food products in top 5 predictions in both models. Still considering the complexity of the problem both models predicted the object well in the top 2 choices.

3 Question 3 **1 / 1**

✓ **- 0 pts** **Correct**

- 0.05 pts State conclusion