1. How is a grayscale image represented on a computer? How about a color image?

Grayscale image are represent by a value between 0 and 255. It is represented by 2D array

Color image will be the same concept, but it has combination of RGB to represent color.

2. How are the files and folders in the MNIST\_SAMPLE dataset structured? Why?

Path/train/7/ … All Images

Path/train/3/ All Images

Path/valid/7/ All Images

Path/valid/3/ All Images

Images structure: 10001.png , 10002.png, 10003.png

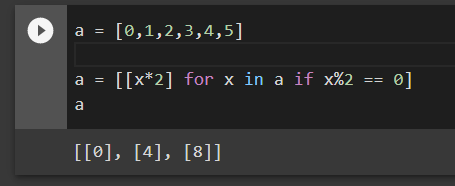
It is structured like this so that it is easy to fetch and process image and data

3. Explain how the “pixel similarity” approach to classifying digits works.

* We get the “average” picture of 3 and 7 as baseline by taking the average of the pixel in the training data
* “Subtract” each image from the mean, and see what is the distance ( using root mean square error, mean absolute error) of the each image from the mean ( through pixel value)
* See if the result is “closer” to 3 or 7 and predict

4. What is a list comprehension? Create one now that selects odd numbers from a  
list and doubles them.

List comprehension is the fast way to access and manipulate element of the list.

  
5. What is a rank-3 tensor?

It is simply a three dimensional tensors.  
6. What is the difference between tensor rank and shape? How do you get the rank  
from the shape?

Shape is the length of each axis

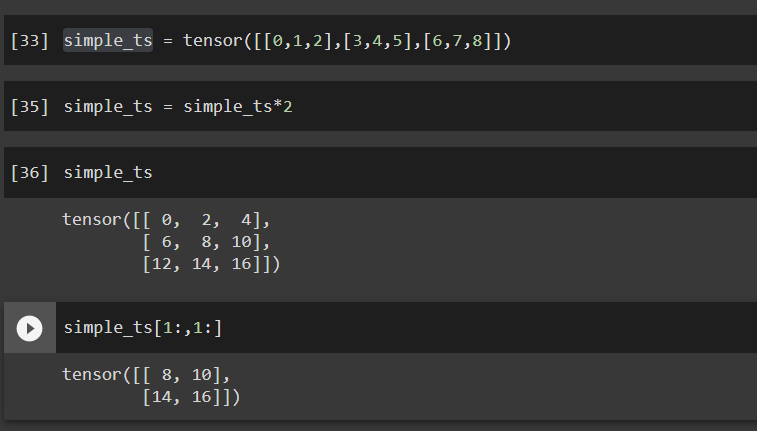
The length of the tensors shape is the rank

  
7. What are RMSE and L1 norm?

Root mean square error and mean absolute error

8. How can you apply a calculation on thousands of numbers at once, many thou‐  
sands of times faster than a Python loop?

Using numpy and Pytorch, it runs in optimized C, which is why it is very fast.

9. Create a 3×3 tensor or array containing the numbers from 1 to 9. Double it.  
Select the bottom-right four numbers.  


10. What is broadcasting?

Broadcasting: automatically expand the tensor with the smaller rank to have the same size as the one with larger rank  
11. Are metrics generally calculated using the training set or the validation set? Why?  
Metric is used for validation set, metric is for human to understand, and we want to see if the model perform well or not from the validation set.  
12. What is SGD?

Schochastic gradient descent:

Taking a step in the oppositite to the gradient to make the model parameters a little bit better  
13. Why does SGD use mini-batches?

To take an optimization step, we need to calculate the loss over one or more data  
items. How many should we use? We could calculate it for the whole dataset and take  
the average, or we could calculate it for a single data item. But neither of these is ideal.  
Calculating it for the whole dataset would take a long time. Calculating it for a single  
item would not use much information, so it would result in an imprecise and unsta‐  
ble gradient. You’d be going to the trouble of updating the weights, but taking into  
account only how that would improve the model’s performance on that single item.  
14. What are the seven steps in SGD for machine learning?

Step 1 : initilize the parameter

Step 2 : predict result

Step 3 : Calculate the loss

Step 4: Calculate the gradient

Step 5: Step the weights

Step 6: Repeat the process

Step 7: Stop  
15. How do we initialize the weights in a model?

We initilize the weight by randomly assign the number to the weight in the model. Specifically in the MNIST model, we assign weight to each pixel position  
16. What is loss?

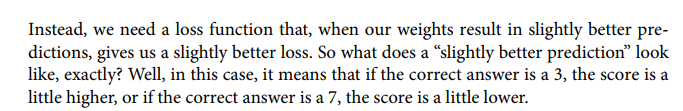
Loss is a value that represent how well ( or bad ) our model is doing  
17. Why can’t we always use a high learning rate?

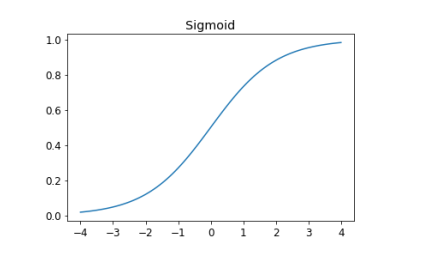
High learning rate can actually increases the loss, because it jump too far from the optimize point( Recall the parabola graph)  
18. What is a gradient?

Derivative  
19. Do you need to know how to calculate gradients yourself? For understanding, yes  
20. Why can’t we use accuracy as a loss function?

The small change in the weight will often not change the accuracy at all. Therefore, accuracy is not a good input for the loss function

This is why

  
21. Draw the sigmoid function. What is special about its shape?



The sigmoid value will always be between 0 and 1. It increases from 0 to 1  
  
22. What is the difference between a loss function and a metric?

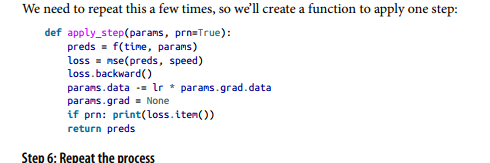
* Loss function: use it to change the weight and improve the model
* Metrics: Use for human to evaluate the model  
  23. What is the function to calculate new weights using a learning rate?
* Train\_epoch(model) for question 25 as well:
  + Loop through the data set
    - Calculate the gradient(take in the data , apply the model)
    - Step
    - Zero out the gradient

24. What does the DataLoader class do?

* It takes any Python collection and turn it into the iterator over many batches, shuffle can be included

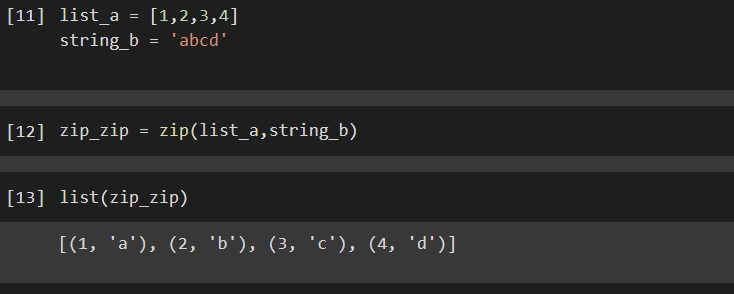
25. Write pseudocode showing the basic steps taken in each epoch for SGD.

(Question 23)



26. Create a function that, if passed two arguments [1,2,3,4] and 'abcd', returns  
[(1, 'a'), (2, 'b'), (3, 'c'), (4, 'd')]. What is special about that out‐  
put data structure?

It is the collection that can be put into the data loader



27. What does view do in PyTorch?

* View is a pytorch method that changes the shape of the tensor, but not changing its content

28. What are the bias parameters in a neural network? Why do we need them?

In the neural network, the w in the equation y = w\*x + b. w is the weight, and b is the bias

The function weights\*pixel is not flexible enough, if pixel = 0, the function will always return 0. That’s why we need to have bias.

29. What does the @ operator do in Python?

Matrix multiplication

30. What does the backward method do?

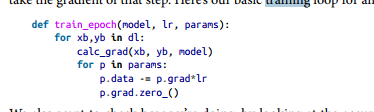
Take the derivative.  
31. Why do we have to zero the gradients?

The backward() will add gradient of loss to any current gradient, we have to zero them out to have no effect on them when we call the function again.

32. What information do we have to pass to Learner?

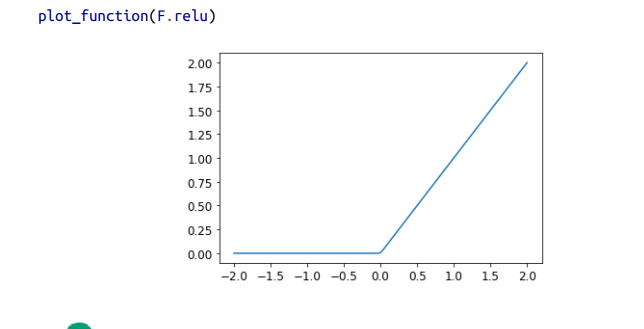
* The Data Loader
* The model
* Optimization function
* Loss function
* Any metrics to print

33. Show Python or pseudocode for the basic steps of a training loop.

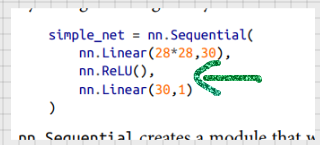
  
34. What is ReLU? Draw a plot of it for values from -2 to +2.

Rectified Linear Unit

Replace any negative value with zero, if it is positive, keep it the same

  
35. What is an activation function?

Is the nonlinearity layer ReLU

  
36. What’s the difference between F.relu and nn.ReLU?

The same  
37. The universal approximation theorem shows that any function can be approxi‐  
mated as closely as needed using just one nonlinearity. So why do we normally  
use more?

Performance, we do not need to use as many parameters.

Smaller matrices, with more layer will get better result than big matrix with less layer