Chapter 8

1. What problem does collaborative filtering solve?

It predicts items users might be interested in based on the ratings given to them by other users. This allows them to solve the problem of recommending new items to users.

2. How does it solve it?

It assigns *latent factors* to each user and item. Latent factors denote characteristics of the user/items that match and don't match, yielding to certain ratings. These factors are initially randomized, and then learned through SGD to reach the target values that the users have already rated. This fills in ratings using these latent factors for items those users have not interacted with.

3. Why might a collaborative filtering predictive model fail to be a very useful recommendation system?

Might not be useful if:

- Not enough recommendations from other users
- Not enough data about the present user to provide useful recommendations

4. What does a crosstab representation of collaborative filtering data look like?

Representation where users and items are the rows and columns respectively, and the created matrix is filled with the corresponding rating given to the item by the user.

5. Write the code to create a crosstab representation of the MovieLens data.

—

6. What is a latent factor? Why is it “latent”?

Latent factors denote characteristics of the user/items that match and don't match, yielding to certain ratings. These factors are initially randomized, and then learned through SGD to reach the target values that the users have already rated. This fills in ratings using these latent factors for items those users have not interacted with. They are considered 'latent' because they are not explicitly given to the model, but are learned without any meaning being explicitly assigned to them.

7. What is a dot product? Calculate a dot product manually using pure python with lists.

A dot product for a vector is when we multiply corresponding element of the two vectors and add them up.

```
a = list(range(5))
b = list(range(5,10))
dot_product = sum(i[0]*i[1] for i in zip(a,b))
```

8. What does pandas.DataFrame.merge do?

It combines to dataframes on a key column (rows match for the values of this column).

9. What is an embedding matrix?

The matrix of latent factors for all users that can be addressed for a subset of items in a computationally-unintensive and differentiable manner is an embedding matrix of user factors and vice-versa.

10. What is the relationship between an embedding and a matrix of one-hot encoded vectors?

An embedding is a computationally-unintensive way of performing the same operation as a matrix of one-hot encoded vectors.

11. Why do we need Embedding if we could use one-hot encoded vectors for the same thing?

Embedding is computationally more efficient than one-hot encoded vectors. Embedding

12. What does an embedding contain before we start training (assuming we're not using a pretained model)?

Embeddings are randomy initialized.

13. Create a class and use it.

```
class Name:
  def __init__(self, name):
    self.name = name

  def say(self, statement):
    return f'Hi, I am {name}. {statement}'

avi = Name('Avi')
avi.say('How are you?')
OUTPUT: "Hi, I am Avi. How are you?"
```

14. What does x[:,0] return?

`x[:,0]` returns the user-ids that have interacted with the items in the batch.

15. Rewrite the DotProduct class and train a model with it.

```
class DotProductBias(Module):
  def __init__(self, n_users, n_items, n_factors, y_range=(0,5.5)):
    self.user_factors = Embedding(n_users, n_factors)
    self.item_factors = Embedding(n_items, n_factors)
    self.user_bias = Embedding(n_users, 1)
    self.item_bias = Embedding(n_items, 1)
    self.y_range = y_range

  def forward(self, x):
    users = user_factors(x[:,0])
    items = item_factors(x[:,1])
    mmult = (users * items).sum(1)
    mmult += self.user_bias + self.item_bias
    return sigmoid_range(mmult, *self.y_range)
```

16. What is a good loss function to use for MovieLens? Why?

We use MSELoss() because we have numerical values to compare for loss.

17. What would happen if we used CrossEntropy loss with MovieLens? How would we need to change the model?

We would need to alter the model to output 5 predictions, the activations for the rating being 1 through 5.

18. What is the use of bias in a dot product model?

Bias allows for the fact that some movies are rated very good or very bad by everybody, or that some users tend to rate all movies higher or lower than average.

19. What is another name for weight decay?

L2 regularization

20. Write the equation for weight decay

```
loss_with_wd = loss + wd * (parameters ** 2).sum()
```

21. Write the equation for the gradient of weight decay. Why does it help reduce weights?

```
parameters.grad += 2 * wd * parameters
# because we decide wd, we can make wd = 2 * wd
parameters.grad += 2 * wd * parameters
```

We are essentially adding our weights to our loss. Because we are minimizing our loss, it will encourage the weights to be as small as possible.

22. Why does reducing weights lead to better generalization?

Letting the model learn high parameters allows it to create sharp surfaces, which might cause it to fit all the data points in the training set with an over-complex function. This leads to overfitting. Reducing the weights leads to smoother surfaces in the model, leading to better generalization.

23. What does argsort do in PyTorch?

`argsort` returns the indices that would sort an array.

24. Does sorting the movie biases give the same result as averaging overall movie ratings by movie? Why / why not?

No, it generates its popularity independent of latent factors. So you might enjoy a movie with high bias even if you don't enjoy that type of movie and vice-versa.

25. How do you print the names and details of the layers in a model?

`learn.model`

26. What is the "bootstrapping problem" in collaborative filtering?

That the model / system cannot make any recommendations or draw any inferences for users or items about which it has not yet gathered sufficient information. It's also called the cold start problem.

27. How could you deal with the bootstrapping problem for new users? For new movies?

You could solve this by coming up with an average embedding for a user or movie. Or select a particular user/movie to represent the average user/movie. Additionally, you could come up with some questions that could help initialize the embedding vectors for new users and movies.

28. How can feedback loops impact collaborative filtering systems?

The recommendations may suffer from representation bias where a small number of people influence the system heavily. E.g., highly enthusiastic anime fans who rate movies much more frequently than others may cause the system to recommend anime more often than expected (incl. to non-anime fans).

29. When using a neural network in collaborative filtering, why can we have different number of factors for movie and user?

In this case, we are not taking the dot product but instead concatenating the embedding matrices, so the number of factors can be different.

30. Why is there a nn.Sequential in the CollabNN model?

This allows us to pass our embedding matrix through linear layers and non-linearities to build a nueral network based solution.

31. What kind of model should be use if we want to add metadata about users and items, or information such as date and time, to a collaborative filter model?

Tabular model should be used.