Chapter 6

1. How could multi-label classification improve the usability of the bear classifier?

Allow for the prediction of no bears present in the image. Otherwise, a multi-class classifier will always predict a class of bear that it thinks is most likely.

2. How do we encode the dependent variable in a multi-label classification problem?

As a one-hot encoded vector.

3. How do you access the rows and columns of a DataFrame as if it was a matrix?

To access the m-th row and the n-th column: `df.iloc[m,n]`

4. How do you get a column by name from a DataFrame

`df['column_name']`

5. What is the difference between a dataset and DataLoader?

`Dataset`: tuple of independent and dependent variable for a *single* item

`DataLoader`: iterator that provides a stream of mini-batches, where each mini-batch is a tuple of a batch of independent variables and dependent variables (but doesn't have to be; you can pass any collection to a `DataLoader` to be split into mini-batches)

6. What does a Datasets object normally contain?

A training `Dataset` and a validation `Dataset`

7. What does a DataLoaders object normally contain?

A training `DataLoader` and a validation `DataLoader`

8. What does lambda do in Python?

`Lambda` are shortcuts for writing one-line functions.

9. What are the methods to customise how the independent and dependent variables are created with the data block API?

`get_x` is used to specify how the independent variables are created

`get_y` is used to specify how the dependent variables are created

10. Why is softmax not an appropriate output activation function when using a one hot encoded target?

One-hot encoded vectors are used when the target label has more than one positive labels, aka multi-label classifications. However, `softmax` makes it such that the model will select one class over the others, making it ideal for training a classifier when each item has one definite label, but not an approproiate activation function for multi-label classification.

11. Why is nll_loss not an appropriate loss function when using a one hot encoded target?

`nll_loss` can only output and select one class over the others. Hence, it can't be used for multi-label classification where one-hot ecoded targets are used.

12. What is the difference between nn.BCELoss and nn.BCEWithLogitsLoss?

`nn.BCELoss` does not include the initial sigmoid. It assumes that the appropriate activation function (ie. the sigmoid) has already been applied to the predictions. `nn.BCEWithLogitsLos`, on the other hand, does both the sigmoid and cross entropy in a single function.

13. Why can't we use regular accuracy in a multi-label problem?

Regular accuracy when the final class predicted (and only one of it), is the one with the highest activation. Multi-label however requires comparing the sigmoid of the activations with a threshold as multiple classes can be applicable to an item.

14. When is it okay to tune an hyper-parameter on the validation set?

It is okay to do so when the relationship between the hyper-parameter and the metric being observed is smooth. With such a smooth relationship, we would not be picking an inappropriate outlier.

15. How is y_range implemented in fastai?

`y_range` is implemented using `sigmoid_range` in fastai:

```
def sigmoid_range(x, low, high):
  return x.sigmoid() * (high - low) + low
```

16. What is a regression problem? What loss function should you use for such a problem?

A problem where the dependent variable are continuous values. They use `nn.MSELoss` or Mean Squared Error loss function.

17. What do you need to do to make sure the fastai library applies the same data augmentation to your input images and your target point coordinates?

Use `blocks = (ImageBlock, PointBlock)` in the `DataBlock`. This makes sure that fastai automatically and correctly applies data augmentation to coordinates.