1. Provide an example of where the bear classification model might work poorly in production, due to structural or style differences in the training data.  
   - The training data may not have all the possible instances of bears in real-life. When images are significantly different in structure and style from the training data, the algorithms may not able to recognize the image.
2. Where do text models currently have a major deficiency?  
   - While deep learning is able to generate context-appropriate text, such as replies to social media posts, and imitating a particular author’s style, it is not good at generating correct response. We don’t currently have a reliable way to combine a knowledge base of something with a deep learning for generating medically correct natural language responses. This can be very dangerous, because it is so easy to create content that appears to a layman to be compelling, but actually is entirely incorrect.
3. What are possible negative societal implications of text generation models?  
   - Text generation models can be used at massive scales to spread disinformation, create unrest, and encourage conflict.
4. In situations where a model might make mistakes, and those mistakes could be harmful, what is a good alternative to automating a process?   
   - Instead of using deep learning as an entirely automated process, we should use it as part of a process in which the model and a human user interact closely. This can potentially make humans orders of magnitude more productive than they would be with entirely manual methods, and actually result in more accurate processes than using a human alone. For instance, an automatic system can be used to identify potential stroke victims directly from CT scans, and send a high-priority alert to have those scans looked at quickly. There is only a three-hour window to treat strokes, so this fast feedback loop could save lives. At the same time, however, all scans could continue be sent to radiologists in the usual way, so there would be no reduction in human input.
5. What kind of tabular data is deep learning particularly good at?  
   - Deep learning is good at greatly increasing the variety of columns to include- for example, columns containing natural language (book titles, reviews, etc.), and high-cardinality categorical columns (i.e., something that contains a large number of discrete choices, such as zip code or product ID).
6. What’s a key downside of directly using a deep learning model for recommendation systems?  
   - It only tells what products a particular user might like, rather than what recommendations would be helpful for a user. Many kinds of recommendations for products a user might like may not be at all helpful – for instance, if the user is already familiar with the products, or if they are simply different packagings of products they have already purchased (such as a boxed set of novels, when they already have each of the items in the set).
7. What are the steps of the Drivetrain Approach?  
   - Defined Objective: Consider your objective.  
   - Levers: Think about what actions you can take to meet that objective.  
   - Data: What data you have (or can acquire) that can help.  
   - Models: Build a model that you can use to determine the best actions to take to get the best results in terms of your objective.
8. How do the steps of Drivetrain Approach map to a recommendation system?  
   - The objective of a recommendation engine is to drive additional sales by surprising and delighting the customers with recommendations of items they would not have purchased without the recommendation. The lever is the ranking of the recommendations. New data must be collected to generate recommendations that will cause new sales. This will require conducting many randomized experiments in order to collect data about a wide range of recommendations for a wide range of customers. This is a step that few organizations take; but without it, you don’t have the information you need to actually optimize recommendations based on your true objective (more sales!).
9. Create an image recognition model using data you curate, and deploy it on the web.  
   - Done <https://github.com/kevinangkajaya/ml_utility> module
10. What is “DataLoaders”?  
    - A thin class that just stores whatever DataLoader objects you pass to it, and makes them available as “train” and “valid”. DataLoaders provide the data for your model.
11. What four things do we need to tell “fastai” to create “DataLoaders”?  
    - What kinds of data we are working with  
    - How to get the list of items.  
    - How to label these items.  
    - How to create the validation set.
12. What does the “splitter” parameter to “DataBlock” do?  
    - Splitter is used to configure how we want to separate out data into training and validation sets.
13. How do we ensure a random split always gives the same validation set?  
    - By fixing the random seed – using the same starting point (called “seed”) each time.
14. What letters are often used to signify the independent and dependent variables?  
    - The independent variable is often referred to as “x” and the dependent variable is often referred to as “y”.
15. What’s the difference between the crop, pad, and squish resize approaches? When might you choose one over the others?  
    - Crop: Fit images to a square shape of the requested size, using the full width or height. This can result in losing some important details.  
    - Pad: Pad the images with zeros (black) for the part outside the original size.  
    - Squish/stretch: Fit images to a square shape by stretching width and height to the available space.  
    - Normally we don’t use any of these three, instead we normally select part of the image, and crop to just that part. On each epoch we randomly select a different part of each image. This means that our model can learn to focus on, and recognize, different features in our images. It also reflects how images work in the real world: different photos of the same thing may be framed in slightly different ways.
16. What is data augmentation? Why is it needed?  
    - Data augmentation refers to creating random variations of our input data such that they appear different, but do not actually change the meaning of the data. Examples of common data augmentation techniques for images are rotation, flipping, perspective warping, brightness changes and contrast changes.
17. What is the difference between “item\_tfms” and “batch\_tfms”?  
    - item\_tfms: Apply changes to each image individually.  
    - batch\_tfms: Apply augmentations to an entire batch using the GPU, which will save a lot of time. Good to use when the images are all the same size.
18. What is a confusion matrix?   
    - A visualization to show mistakes the model makes.
19. What does “export” save?  
    - Save a model (which consist of “architecture” and “parameters”) in order to load a model you can be sure that have the matching architecture and parameters. This method even saves the definition of how to create the “DataLoaders”, otherwise you would have to redefine how to transform your data in order to user your model in production.
20. What is it called when we use a model for getting predictions, instead of training?  
    - Inference is when we use a model for getting predictions instead of training.
21. What are IPython widgets?  
    - IPython widgets (ipwidgets) are GUI components that bring together JavaScript and Python functionality in a web browser, and can be created and used within a Jupyter notebook. For instance, the image cleaner at the earlier part of chapter 2.
22. When might you want to use CPU for deployment? When might GPU be better?  
    - For deployment, we always want to use CPU. GPUs are only useful when they do lots of identical work in parallel. If you’re doing image classification, then you’ll normally be classifying just one user’s image at a time, and there isn’t normally enough work to do in a single image to keep a GPU busy for long enough for it to be very efficient. So, a CPU will often be more cost-effective.  
    - GPU is better only during training images, which means we should use GPU during the development (training) computers.
23. What are the downsides of deploying your app to a server, instead of to a client (or edge) device such as a phone or PC?  
    - Deploying to client (or edge) devices tend to require a lot of extra steps and boilerplate, and do not always support all the PyTorch and fastai layers that your model might use. In addition, the work you do will depend on what kind of mobile devices you are targeting for deployment – you might need to do some work to run on iOS devices, different work to run on newer Android devices, different work for older Android devices, etc.  
    - On the other hand, while deploying to a server has some advantages, such as easier initial installation as you only need to deploy a small GUI application, which connects to the server that do the heavy lifting, and upgrades of core logic can happen on server instead of needing to be distributed to all of the users, the downsides are your application will require a network connection and there will be some latency each time the model is called, also if your application uses sensitive data then your users may be concerned about an approach which sends that data to a remote server
24. What are three examples of problems that could occur when rolling out a bear warning system in practice?  
    - Training requires video data instead of images.  
    - Handling nighttime images which may not appear in dataset.  
    - Dealing with low-resolution camera images.  
    -Ensuring results are returned fast enough to be useful in practice.  
    - Recognizing bears in positions that are rarely seen in photos that people post online (for example from behind, partially covered by bushes, or when a long way away from the camera).
25. What is “out-of-domain data”?  
    - Data that our model sees in production which is very different from what it saw during training.
26. What is “domain shift”?  
    - When the type of data that our model sees changes over time. For instance, an insurance company may use a deep learning model as part of its pricing and risk algorithm, but over time the types of customers that the company attracts and the types of risks they represent, may change so much that the original training data is no longer relevant.
27. What are the three steps in the deployment process?  
    - Manual process: Deep learning model approach running in parallel but not being used directly to drive any actions. The humans involved in the manual process should look at the deep learning outputs and check whether they make sense.   
    - Limited scope deployment: Limit the scope of the model and have it carefully supervised by people. For instance, do a small geographically and time-constrained trial of the model-driven approach.  
    - Gradual expansion: Gradually increase the scope of the rollout. Ensure that you have really good reporting systems in place, to make sure that you are aware of any significant changes to the actions being taken compared to your manual process.

Further Research

1. Consider how the Drivetrain Approach maps to a project or problem you’re interested in.  
   - Objective: Distinguish items for faster checkout in cashier, to prevent delay of manually identifying items, without needing to use barcode.  
   - Levers: The items images from all sides.  
   - Data: Self-captured images of items.  
   - Models: Camera scanner to identify item.
2. When might it be best to avoid certain types of data augmentation?  
   - When the augmentation image result are no longer showing the important part that define what image it is. For example, when data augmentation crops a large portion of an image, or when the images become too blurry or distorted to infer.
3. For a project you’re interested in applying deep learning to, consider the thought experiment “What would happen if it went really, really well?”  
   - Cashiers will no longer mislabel items during the transactions (for example, mistaken Coca cola zero as regular coca cola).  
   - Staffs no longer need to spend time managing barcode; instead everything can be directly identified from the item itself.
4. Start a blog, and write your first blog post. For instance, write about what you think deep learning might be useful in a domain you’re interested in.

Answer: <https://forums.fast.ai/t/fastbook-chapter-2-questionnaire-solutions-wiki/66392>