1. After each stride-2 conv, why do we double the number of filters?

Ans: Double the number of filters after each stride-2 conv is a common practice in deep learning. This is because it helps the model learn more complex features at each layer and helps reduce the dimensionality of the input. By doubling the number of filters, the model can more effectively detect features across the entire image. Additionally, doubling the number of filters helps to reduce the total number of parameters in the model, which can reduce the risk of overfitting.

2. Why do we use a larger kernel with MNIST (with simple cnn) in the first conv?

Ans: We use a larger kernel with MNIST in the first convolutional layer because it helps to capture the large, broad features of the images in the dataset. Larger kernels have a larger receptive field, which helps to pick up more abstract features. This allows for better feature extraction and ultimately better classification performance.

3. What data is saved by ActivationStats for each layer?

Ans: ActivationStats saves the activation mean and standard deviation for each layer in a neural network. It also stores the layer name, the number of parameters, the number of neurons, and the activation shape.

4. How do we get a learner's callback after they've completed training?

Ans: A learner's callback after they have completed training can be achieved by setting up a callback URL in the training platform. This URL should be configured to trigger a POST request to the desired endpoint upon completion of training. The endpoint should then process the data returned in the POST request to determine the learner's results and take any necessary action.

5. What are the drawbacks of activations above zero?

Ans The drawbacks of activations above zero include potential saturation of neurons, making them unable to learn, and possible instability of the network due to large gradients. Additionally, activations above zero can lead to slower training, as the gradients become smaller as the activation values increase.

6. Draw up the benefits and drawbacks of practicing in larger batches?

Ans:

Benefits:

• Larger batches allow for increased data and more diverse training data sets.

• Larger batches can result in faster learning of complex patterns.

• The use of larger batches can reduce overfitting due to the sheer number of examples to train on.

Drawbacks:

• Larger batches can cause longer training times due to increased computation time.

• The use of larger batches can result in increased chances of overfitting due to a lack of generalization.

• Larger batches can lead to increased memory requirements due to the increased amount of data.

7. Why should we avoid starting training with a high learning rate?

Ans: Starting with a high learning rate can cause the weights to update too quickly and the model to diverge. This can result in the model overfitting to the training data and performing poorly on unseen test data. It is better to start with a low learning rate and gradually increase it as the model begins to converge.

8. What are the pros of studying with a high rate of learning?

Ans: Pros of studying with a high rate of learning include:

* Improved accuracy and performance of algorithms
* Increased ability to recognize and classify objects
* Faster development and deployment of computer vision systems
* Ability to identify and track objects more accurately
* Improved understanding of how vision works in different environments
* Enhanced ability to identify and respond to changes in the environment
* Increased ability to recognize patterns and identify trends in data sets

9. Why do we want to end the training with a low learning rate?

Ans: We want to end the training with a low learning rate because it helps the model learn more slowly and with finer precision, allowing it to hone in on the best parameters for the model. This helps prevent the model from overfitting or underfitting the data, which can lead to poor performance.