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

**Ans:** Doubling the number of filters after each stride-2 convolution helps to increase the capacity and richness of the feature representation in deeper layers of the network. Since the spatial dimensions of the feature maps are reduced by half after each stride-2 convolution, doubling the number of filters helps to maintain the expressive power of the network.

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

**Ans:** Using a larger kernel with MNIST in the first convolution helps capture more complex patterns and structures in the input images. MNIST images are relatively simple compared to natural images, so using a larger kernel allows the network to extract more meaningful features from the input data.

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

**Ans:** ActivationStats saves the following data for each layer during training: mean, std (standard deviation), mean of absolute values, pct zero, and histogram of activations. These statistics provide insights into the behavior and distribution of activations within each layer, which can be useful for debugging, monitoring training progress, and diagnosing potential issues such as vanishing or exploding gradients.

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

**Ans:** To get a learner's callback after they've completed training, you can use the after\_fit callback, which is executed after the training loop has completed. You can define a custom callback function to perform any desired actions or computations based on the learner's training results or final state.

**5. What are the drawbacks of activations above zero?**

**Ans:** Drawbacks of activations above zero include the potential for gradient saturation, which can lead to vanishing or exploding gradients during training. Additionally, activations above zero may introduce non-linearities that could potentially distort the representation of the input data and affect the overall performance of the neural network.

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

**Ans:** Benefits of practicing in larger batches include increased computational efficiency, better utilization of hardware resources, and potentially faster convergence during training. However, drawbacks include increased memory requirements, potential generalization issues, and sensitivity to hyperparameters such as learning rate and weight decay.

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

**Ans:** Starting training with a high learning rate can lead to unstable training dynamics, rapid divergence, or oscillation of the loss function. This can result in poor convergence, slow learning progress, or failure to find a good local minimum of the loss function.

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

**Ans:** Pros of studying with a high learning rate include faster initial progress in reducing the loss function, potentially faster convergence towards a good local minimum, and improved exploration of the parameter space during optimization.

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

**Ans:** Ending the training with a low learning rate helps to fine-tune the model parameters towards a more optimal solution and improve the generalization performance of the model. It allows the model to make smaller, more precise adjustments to the parameter values while minimizing the risk of overfitting or destabilizing the training process.