**1. What are the advantages of a CNN for image classification over a completely linked DNN?**

**Ans:** Advantages of CNN for Image Classification over completely linked DNN:

CNNs are designed to handle spatial information inherent in images more effectively than fully connected DNNs.

CNNs use shared weights and local connectivity, reducing the number of parameters and enabling the network to capture translation-invariant features.

CNNs can exploit hierarchical patterns in images through the use of convolutional and pooling layers, leading to better feature representation and improved classification performance.

CNNs are computationally efficient, as they exploit spatial locality and parameter sharing, making them suitable for processing large-scale image data.

**2. Consider a CNN with three convolutional layers, each of which has three kernels, a stride of two, and SAME padding. The bottom layer generates 100 function maps, the middle layer 200, and the top layer 400. RGB images with a size of 200 x 300 pixels are used as input. How many criteria does the CNN have in total? How much RAM would this network need when making a single instance prediction if we're using 32-bit floats? What if you were to practice on a batch of 50 images?**

**Ans:** Total parameters:

Bottom layer:

200×300×100×3×3

200×300×100×3×3

Middle layer:

100×150×200×3×3

100×150×200×3×3

Top layer:

50×75×400×3×3

50×75×400×3×3

Total parameters = sum of parameters in all layers

RAM usage for single instance prediction:

Sum of parameters multiplied by 32 bits

RAM usage for batch of 50 images:

Multiply RAM usage for single instance by 50

**3. What are five things you might do to fix the problem if your GPU runs out of memory while training a CNN?**

**Ans:**

Reduce batch size: Decrease the number of samples processed in each batch during training.

Reduce model size: Decrease the number of layers, neurons, or filters in the network architecture.

Use data augmentation: Generate synthetic training samples by applying transformations like rotation, flipping, or scaling to the input images.

Use mixed precision training: Use a combination of single and half precision floating-point formats to reduce memory usage without sacrificing model accuracy.

Utilize memory-efficient operations: Use memory-efficient operations and algorithms, such as gradient checkpointing or model parallelism, to reduce memory overhead during training.

**4. Why would you use a max pooling layer instead with a convolutional layer of the same stride?**

**Ans:** Max pooling layers are commonly used with convolutional layers to downsample feature maps and reduce spatial dimensions while retaining important features.

Max pooling layers help introduce translational invariance and reduce computational complexity by summarizing the most salient features within local regions of the input.

**5. When would a local response normalization layer be useful?**

**Ans:** Local response normalization layers are useful for introducing competition between neurons and enhancing the contrast of activation responses within local neighborhoods.

They help improve model generalization by normalizing the responses of neurons relative to their neighbors, making the network less sensitive to variations in input intensity and enhancing feature discrimination.

**6. In comparison to LeNet-5, what are the main innovations in AlexNet? What about GoogLeNet and ResNet's core innovations?**

**Ans:** AlexNet introduced the concept of deep convolutional neural networks for image classification tasks, along with innovations like ReLU activation, dropout regularization, and GPU acceleration.

GoogLeNet introduced the inception module, which enables efficient and effective feature extraction at multiple scales by using parallel convolutional operations with different kernel sizes.

ResNet introduced skip connections or residual connections, which allow the training of very deep networks by mitigating the vanishing gradient problem and promoting more effective gradient flow during training.

**7. On MNIST, build your own CNN and strive to achieve the best possible accuracy.**

**Ans:** Implement a CNN architecture using libraries like TensorFlow or PyTorch.

Design the network with convolutional, pooling, and fully connected layers.

Train the model on the MNIST dataset, optimizing for accuracy.

Experiment with different architectures, hyperparameters, and regularization techniques to achieve the best possible accuracy.

**8. Using Inception v3 to classify broad images. a.**

**Images of different animals can be downloaded. Load them in Python using the matplotlib.image.mpimg.imread() or scipy.misc.imread() functions, for example. Resize and/or crop them to 299 x 299 pixels, and make sure they only have three channels (RGB) and no transparency. The photos used to train the Inception model were preprocessed to have values ranging from -1.0 to 1.0, so make sure yours do as well.**

**Ans:** Load and preprocess images using Python libraries like matplotlib or scipy.

Resize and crop images to 299x299 pixels with three RGB channels and normalize pixel values to range from -1.0 to 1.0.

Use Inception v3 pre-trained model and fine-tune it for classifying broad images.

Evaluate the model's performance on the test set and fine-tune hyperparameters if necessary to improve accuracy.

**9. Large-scale image recognition using transfer learning.**

**a. Make a training set of at least 100 images for each class. You might, for example, identify your own photos based on their position (beach, mountain, area, etc.) or use an existing dataset, such as the flowers dataset or MIT's places dataset (requires registration, and it is huge).**

**b. Create a preprocessing phase that resizes and crops the image to 299 x 299 pixels while also adding some randomness for data augmentation.**

**c. Using the previously trained Inception v3 model, freeze all layers up to the bottleneck layer (the last layer before output layer) and replace output layer with appropriate number of outputs for your new classification task (e.g., the flowers dataset has five mutually exclusive classes so the output layer must have five neurons and use softmax activation function).**

**d. Separate the data into two sets: a training and a test set. The training set is used to train the model, and the test set is used to evaluate it.**

**Ans:** Create a training set with at least 100 images for each class, preprocess images, and perform data augmentation.

Use transfer learning with Inception v3 model by freezing layers up to the bottleneck layer and replacing the output layer with appropriate number of outputs for the new classification task.

Split the data into training and test sets, train the model on the training set, and evaluate its performance on the test set to assess classification accuracy.