Assignment 3:

Sajjad Zangiabadi

Question 1:

- Gradient Accumulation gives a memory-efficient technique for employing higher effective batch sizes in deep learning training. With this technique, the training data is systematically divided into smaller mini batches & processed one after each other. Following each mini batch, the gradients are accumulated across several iterations without the model's internal parameters (weights) being updated. Only when a specific number of mini batches have been passed through the network will the weight update occur.
- This method works very well when:
  - Your hardware (such as GPU RAM) is constrained; thus, you would like to use a bigger batch size.
  - You're working with complex models or high-resolution photos requiring additional RAM.
  - For example, if you want to use 32 images in one batch, but your hardware crashes once you go beyond 8. In that case, you can use batches of 8 images and update weights once every 4 batches.
- Its implementation with pytorch

````
# loop through batches
for inputs, labels in data_loader:
  # forward pass
  preds = model(inputs)
  loss = criterion(preds, labels)
  # backward pass
  loss.backward()
  # condition the weight update on the index of the running batch
  if (index + 1) % accum_iter == 0:
     # weights update
     optimizer.step()
     optimizer.zero_grad()
````

Question 3:

- The advantages of pooling layers
  - **Dimensionality Reduction:** By reducing the feature maps' dimensionality, pooling layers help the network perform less computation and learn fewer parameters.
  - **Robustness to Variations**: They increase the model's resistance to changes in the input image's feature positions.
  - **Preventing Overfitting:** Adding pooling layers also has the effect of preventing overfitting.
  - **Enhanced Performance and Quicker Training Times**: In a CNN model, pooling layers can result in heightened performance and quicker training periods.
- The drawbacks for pooling layers
  - **Information Loss:** Some information from the input feature maps is lost during the pooling layers' process, which may be crucial for the subsequent classification or regression operation.
  - **Hyper-parameter tunning**: The performance of the network may be affected by and may need to be adjusted due to the introduction of hyperparameters by pooling, such as the pooling window size and stride.

- Comparing Maximum and Average Pooling
    - **Max Pooling** selects the maximum element from the region of the feature map covered by the filter. Consequently, the feature map resulting from the max-pooling layer will contain the most noticeable features of the previous feature map.
    - **Average Pooling** computes the average of the elements present in the region of feature map covered by the filter.
    - Thus, average pooling gives the average of the features present in a patch, while max pooling gives the most obvious feature in a specific patch of the feature map. The most distinguishing characteristics of an object, such as its edges and corners, are frequently retained by max pooling. On the other hand, average pooling encourages the network to identify the object's entire extent.

- In convolutional neural networks, pooling layers are frequently employed. Indeed, a CNN model architecture often consists of many convolution and pooling layers placed one after the other.

Question 4:

- **Transfer learning** is a machine learning technique that a pre-trained model is used as the foundation for a different but related problem. This technique allows us to utilize the knowledge extracted from vast datasets to address new, smaller datasets with limited labeled data.
- Start with a CNN model that has been extensively trained on a vast dataset such as ImageNet, comprising millions of labeled images. This training enables the model to learn low-level features like edges, lines, and shapes. To implement transfer learning, we retain the pre-trained model and freeze the weights (parameters) in the initial convolutional layers. These layers are responsible for capturing generic image features applicable across diverse tasks. We append additional trainable layers top of the frozen convolutional layers. These new layers serve as a classifier specifically customized for the new, smaller dataset and task. By fine-tuning solely these final layers, the model adapts the learned features to the new problem.
- This method has a number of benefits, especially for tasks with limited data:
    - **Reduced Training Time:** Compared to training the complete network from scratch, training the last layers of the pre-trained model is considerably quicker because it already has a solid basis.
    - **Enhanced Performance:** Even with a smaller dataset, the model can achieve higher accuracy on the new job by utilizing the pre-trained features.
    - **Data Efficiency:** By using transfer learning, we can efficiently train models on datasets that would not be enough to train a CNN from the beginning.

- Popular pre-trained models' examples are as follows:
    - **VGG**: such as VGG16 and VGG19, were utilized in the ILSVRC-2014 competition and are revered for their simplistic yet deep architecture. They are predominantly employed in tasks involving image classification and localization.
    - **ResNet**: revolutionized deep neural networks by introducing the concept of "skip connections" to combat the issue of vanishing gradients. This model is commonly utilized for object detection and fine-grain classification tasks.
    - **Inception**: develop the "network in network" concept to reduce computational expenses. It is frequently applied in image classification tasks.
    - **MobileNet**: is specifically designed for mobile applications, providing efficiency in speed and size without reducing performance. It is ideal for real-time tasks on low-end hardware, such as mobile or embedded devices.

Question 5:

- **True**: Face Validation is the process of validating if a person's facial features match an existing reference face; this is often used for activities like device unlocking or admission into restricted areas. Face recognition, on the other hand, is the process of identifying a person by comparing their face features to a database or a group of faces. This technique is usually used in tracking systems or photo tagging applications where multiple people need to be identified.
- **I am not sure**, but I think training a face validation system that must have one reference for each person would be a good idea for this given data, and for this purpose, use neural networks like Siamese. For the face recognition task, I think we should have more data – images – for each person.
- **True**: The network usually learns to recognize low-level features like edges and color gradients in the first few layers (like layer 1). Deeper inside the network (such as layer 4), the network starts to identify higher-level, more complicated properties. Depending on the depth of the network and the complexity of the dataset, these might be individual items or even entire objects.