1. What is the COVARIATE SHIFT Issue, and how does it affect you?

Ans: Covariate shift is an issue arising when training and test data are drawn from different distributions. It occurs when the distributions of the input variables in the training and test data are different or when the joint distributions of the input and output variables in the training and test data are different. This can lead to poor performance when a model trained on the training data is applied to the test data. It affects you by reducing the accuracy of your model when it is applied to data outside the training set. This can be especially problematic when working with high-dimensional data, as it is often more difficult to detect the differences between training and test data distributions.

2. What is the process of BATCH NORMALIZATION?

Ans: Batch normalization is a technique used to normalize data within a mini-batch by subtracting the mean and dividing by the standard deviation. It is used to reduce the internal covariate shift of the model which can help improve the model’s accuracy. Batch normalization also helps reduce the amount of computational resources required for training by reducing the number of parameters to be adjusted. The process consists of adding two parameters to the model: a learnable scale and a learnable shift. These parameters are learned during the training process and enable the model to adjust itself to the distribution of the data. Batch normalization can help improve the accuracy and convergence of the model, reduce overfitting, and speed up training time.

3. Using our own terms and diagrams, explain LENET ARCHITECTURE.

Ans: A LeNet architecture is a type of deep neural network that consists of multiple convolutional layers, pooling layers, and fully connected layers. It was developed by Yann LeCun in 1989 and is the foundation of modern convolutional neural networks.

The architecture of LeNet includes the following layers:

1. Convolutional Layer: This layer applies a set of convolution filters to the input image, each filter produces a feature map.
2. Pooling Layer: This layer reduces the spatial size of the feature map by performing down sampling.
3. Fully Connected Layer: This layer consists of neurons that are fully connected to all the neurons in the previous layer.
4. Output Layer: This layer performs the classification task by transforming the output of the fully connected layer into a probability distribution over the classes.

4. Using our own terms and diagrams, explain ALEXNET ARCHITECTURE.

Ans: AlexNet is a convolutional neural network architecture used in deep learning. It is the first deep learning model to win the ImageNet competition in 2012.

The AlexNet architecture consists of five convolutional layers and three fully-connected layers. The convolutional layers apply convolution operations to the input image to extract features at different levels of abstraction. The fully-connected layers are used for classification.

AlexNet architecture consists of:

* Convolutional Layers:
  + The first convolutional layer has 96 filters with a size of 11x11 and stride of 4.
  + The second convolutional layer has 256 filters with a size of 5x5 and stride of 1.
  + The third convolutional layer has 384 filters with a size of 3x3 and stride of 1.
  + The fourth convolutional layer has 384 filters with a size of 3x3 and stride of 1.
  + The fifth convolutional layer has 256 filters with a size of 3x3 and stride of 1.
* Max-Pooling Layers:
  + The first max-pooling layer has a pool size of 3x3 and stride of 2.
  + The second max-pooling layer has a pool size of 3x3 and stride of 2.
* Fully-Connected Layers:
  + The first fully-connected layer has 4096 neurons.
  + The second fully-connected layer has 4096 neurons.
  + The third fully-connected layer has 1000 neurons.
* Dropout Layer: This layer is used for regularization purposes and prevents overfitting.
* Softmax Layer: This layer is used for classification and produces a probability distribution for the classes.

5. Describe the vanishing gradient problem.

Ans: The vanishing gradient problem is a phenomenon in which the gradients of a neural network's weights, when backpropagated, become so small that they virtually vanish. This is due to the fact that the gradients are multiplied together in the backpropagation process which causes the gradients to shrink exponentially as they propagate backwards through the layers of the network. As a result, the weights of earlier layers in the network are not updated effectively and the network is not able to learn effectively. This can lead to poor performance and slow training of the network.

6. What is NORMALIZATION OF LOCAL RESPONSE?

Ans: Normalization of local response (NLR) is a technique used to normalize the responses of a convolutional neural network (CNN) across different image regions. This technique helps to make the prediction of a model more robust and accurate. NLR works by normalizing the input data, which helps to reduce the effects of local variation in the input. The normalized responses are then used to make predictions. NLR can be used to improve the accuracy of object recognition, facial recognition, and other computer vision tasks.

7. In AlexNet, what WEIGHT REGULARIZATION was used?

Ans: AlexNet used a form of weight regularization known as dropout regularization. This technique randomly sets neurons in the network to zero during training, reducing the number of weights and neurons that are used in the network. This helps reduce overfitting and improves generalization.

8. Using our own terms and diagrams, explain VGGNET ARCHITECTURE.

Ans: VGGNet (Visual Geometry Group Network) is a popular convolutional neural network architecture that was developed by Oxford's Visual Geometry Group in 2014. VGGNet is a deep convolutional neural network composed of 16 layers, including 13 convolutional layers and 3 fully-connected layers. The architecture consists of alternating convolutional layers with a 3x3 filter size and a 2x2 max pooling layer followed by a fully connected layer with a ReLU (Rectified Linear Unit) activation.

The architecture of VGGNet consists of a series of convolutional layers with small filter sizes of 3x3 pixels, followed by a max pooling layer. This allows the network to capture fine-grained features while also reducing the computational complexity. The number of convolutional layers increases as the network goes deeper, and the number of filters increases with each layer. The fully-connected layers at the end of the network combine the features extracted by the convolutional layers to classify the input image.

9. Describe VGGNET CONFIGURATIONS.

Ans: VGGNet was developed by the Visual Geometry Group (VGG) at Oxford, and is a deep convolutional neural network (CNN) comprised of 16 layers. It is one of the most advanced and influential architectures in the field of computer vision. VGGNet configurations are characterized as having only 3x3 convolutional filters and a deep stack of convolutional layers that are followed by a few fully connected layers. It is also known to have a lot of parameters and can be quite slow to train. The main configurations of VGGNet are VGG-16 and VGG-19, the two models that were trained on the ImageNet dataset for the ILSVRC-2014 competition. VGG-16 has 16 layers, 13 convolutional layers and 3 fully connected layers, while VGG-19 has 19 layers, 16 convolutional layers and 3 fully connected layers. Both models use small 3x3 filters, but differ in the number of convolutional layers. VGG-16 uses more layers with smaller strides, while VGG-19 uses fewer layers with larger strides. VGG-16 and VGG-19 both use the same input size, 224 x 224. The output size of VGG-16 is 1000, while the output size of VGG-19 is 1000.

10. What regularization methods are used in VGGNET to prevent overfitting?

Ans: Regularization methods used in VGGNet to prevent overfitting include weight decay, dropout, and batch normalization. Weight decay is a form of regularization where a penalty is applied to the weights in a model to reduce their values and help reduce overfitting. Dropout is a technique where nodes in a neural network are randomly ignored during training, which can help reduce overfitting. Batch normalization is a technique where the inputs to each layer in a neural network are normalized, which helps reduce overfitting.