ASSIGNMENT - 7

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

Ans: Covariate Shift:

Imagine you train a model to recognize cats using pictures of fluffy house cats.

Covariate shift occurs when the real world throws you something different, like hairless sphinx cats.

The model's training data (fluffy cats) has a different distribution of features (furriness) compared to the new data (hairless cats).

This mismatch confuses the model, leading to poor performance on unseen data.

Impact : As a large language model, covariate shift can happen when the way people use language changes over time. For example, new words (e.g. Simp or Rizz) or phrases might not be in my training data, affecting my ability to understand them correctly.

2. What is the process of BATCH NORMALIZATION?

Ans: Batch Normalization:

* Imagine you're training a team of people (the model) on a task.
* Some people might get data (inputs) that's naturally harder or easier (different scales).
* Batch normalization evens the playing field by standardizing the data each "mini-batch" of training examples receives.
* This helps the model learn faster and avoid getting stuck in bad local minima.

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

Ans: LeNet Architecture (Simple terms):

LeNet is like a tiered detective agency solving an image crime (classification).

The first tier (convolution) gathers clues (features) from the image (pixels).

The next tier (pooling) summarizes the most important clues from each area.

This clue summarization continues through more tiers, getting higher-level details.

Finally, the "boss" (fully connected layer) uses all the clues to decide the criminal (image class).

Conv -> Pool -> Conv -> Pool -> Fully Connected -> Output (Class)

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

Ans: AlexNet Architecture (Simple terms):

* AlexNet is like a deeper, more specialized detective agency.
* It has more convolutional and pooling layers, extracting finer details from the image.
* It also uses ReLU (Rectified Linear Unit) activation functions, making the model more efficient at learning.

Conv -> ReLU -> Pool -> Conv -> ReLU -> Pool ->

Conv -> ReLU -> Conv -> ReLU -> Pool -> Fully Connected -> Output (Class)

5. Describe the vanishing gradient problem.

Ans: Vanishing Gradient Problem:

* Imagine training a long line of people (neurons) to pass a message (gradient) back.
* The message might get weaker (vanish) as it's passed through many layers.
* This makes it hard for the early layers (far from the output) to learn.

6. What is NORMALIZATION OF LOCAL RESPONSE?

Ans: Normalization of Local Response (NLR):

* NLR is a technique to address vanishing gradients in earlier layers of convolutional neural networks (CNNs).
* It compares a neuron's activation to the activity of its neighbors in the same feature map.
* By normalizing based on this local activity, NLR helps prevent dominant activations and improves gradient flow.

7. In AlexNet, what WEIGHT REGULARIZATION was used?

Ans: Weight Regularization in AlexNet:

* Weight regularization is like adding a "don't overfit" rule to the detective training.
* AlexNet used L2 regularization, which penalizes large weights in the model.
* This discourages the model from relying too heavily on specific features and encourages it to learn more generalizable patterns.

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

Ans: VGGNet Architecture (Simple terms):

* VGGNet is like a very deep detective agency, relying heavily on convolutional layers.
* It stacks many small convolutional filters (3x3) one after another, extracting lower-level features efficiently.
* This allows VGGNet to achieve good performance without needing as many complex operations as AlexNet.

Many Conv (3x3) layers -> Pool -> Many Conv (3x3) layers -> Pool -> Fully Connected -> Output (Class)

9. Describe VGGNET CONFIGURATIONS.

Ans: VGGNet Configurations:

* VGGNet comes in different versions (A, B, C, D, E) with varying depths (number of convolutional layers).
* VGG-16 and VGG-19 are popular configurations, with 16 and 19 convolutional layers, respectively.

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

Ans: Regularization methods in VGGNet:

* VGGNet primarily uses weight decay (L2 regularization) to prevent overfitting.
* By penalizing large weights, the model is encouraged to learn more generalizable features.
* Additionally, VGGNet's architecture with smaller convolutional filters helps reduce model complexity and overfitting.