Reflection Questions

Q1. Where did training fail due to structure?

Training failed in the deeper sigmoid network. As depth increased, the model's training accuracy stopped improving and loss decreased very slowly. This happened because the deep fully connected structure combined with sigmoid activation caused vanishing gradients. In the image task, the dense network also struggled compared to CNN because it ignored spatial structure and had too many unnecessary parameters.

Q2.Where did optimizer matter more than activation?

Optimizer mattered more in the CNN training. When the architecture and activation were fixed, switching from SGD to Adam significantly improved convergence speed and final loss. The activation function remained the same (ReLU), but optimizer choice changed the training behavior drastically. This shows optimizer had stronger influence than activation in that case.

Q3.Where did activation matter more than depth?

Activation mattered more in deeper networks. A 10-layer ReLU network trained properly, while a 10-layer sigmoid network showed degraded performance. Even though both had same depth, the activation determined whether gradients could flow effectively. Hence, activation choice had more impact than depth alone.

Q4.What causes gradient shrinkage?

Gradient shrinkage happens due to repeated multiplication of small derivatives during backpropagation. In sigmoid activation, the derivative is always less than 1 and often close to zero. In deep networks, gradients are multiplied across many layers, causing them to decrease exponentially. As a result, early layers receive almost no updates.

Q5. Why does CNN generalize better than dense?

CNN generalizes better because it uses parameter sharing and local receptive fields. Instead of connecting every input to every neuron, convolution applies the same filter across spatial locations. This reduces parameters and helps the model learn meaningful patterns instead of memorizing noise. Dense networks have more parameters and overfit more easily.

Q6.Why does dropout reduce overfitting?

Dropout randomly disables neurons during training. This prevents neurons from becoming overly dependent on specific features. It forces the network to learn more robust and distributed representations. As a result, the gap between training and validation accuracy reduces.

Q7.When does depth hurt test performance?

Depth hurts test performance when the model becomes too complex for the dataset size. More layers increase parameters and model capacity, which can lead to overfitting. If validation accuracy decreases while training accuracy increases, it indicates depth is harming generalization.

Q8.Did validation always predict test performance correctly?

In most cases, validation performance closely matched test performance. However, small differences may occur due to random data split. Validation generally provided a good estimate, but it is not always perfectly identical to test results.