Question 1 (30 points):

1. **Please explain the pros and cons of Instance-Based Learning and Model-Based Learning respectively. (7 points)**

Answer:

Instance-Based Learning

Advantages
• Training is very fast.
• Learn complex target functions. • Do not lose information.

Disadvantages
• Slow at query time.
• Easily fooled by irrelevant attributes.
• For large training sets, requires large memory.

Model-Based Learning

Advantages
• Safe to plan exploration and can train from simulated experiences.
• It tends to have higher sample, meaning it requires less data to learn a policy.
• Can determine some degree of model uncertainty, so that you can gauge how confident you should be about the resulting decision process.

Disadvantages
• Agent only as good as the model learnt. Also, sometimes this becomes a bottleneck, as the model becomes surprisingly tricky to learn.
• Computationally more complex than model-free methods.
• There are two different sources of approximation error in model-based approach, whereas in model-free approach there's only one.

2. **Explain what is Distance Weighted kNN. (5 points)**

Answer:

Since KNN predictions are based on the intuitive assumption that objects close in distance are potentially similar, it makes good sense to discriminate between the K nearest neighbors when making predictions, i.e., let the closest points among the K nearest neighbors have more say in affecting the outcome of the query point.

This can be achieved by introducing a set of weights W, one for each nearest neighbor, defined by the relative closeness of each neighbor with respect to the query point.

Thus:

$$W(x, p_i) = \frac{\exp(-D(x, p_i))}{\sum_{i=1}^{K} \exp(-D(x, p_i)}$$

where *D (x, pi)* is the distance between the query point x and the ith case pi of the example sample. It is clear that the weights defined in this manner above will satisfy:

$$\sum_{i=1}^{K} W(x_o, x_i) = 1$$

Thus, for regression problems, we have:

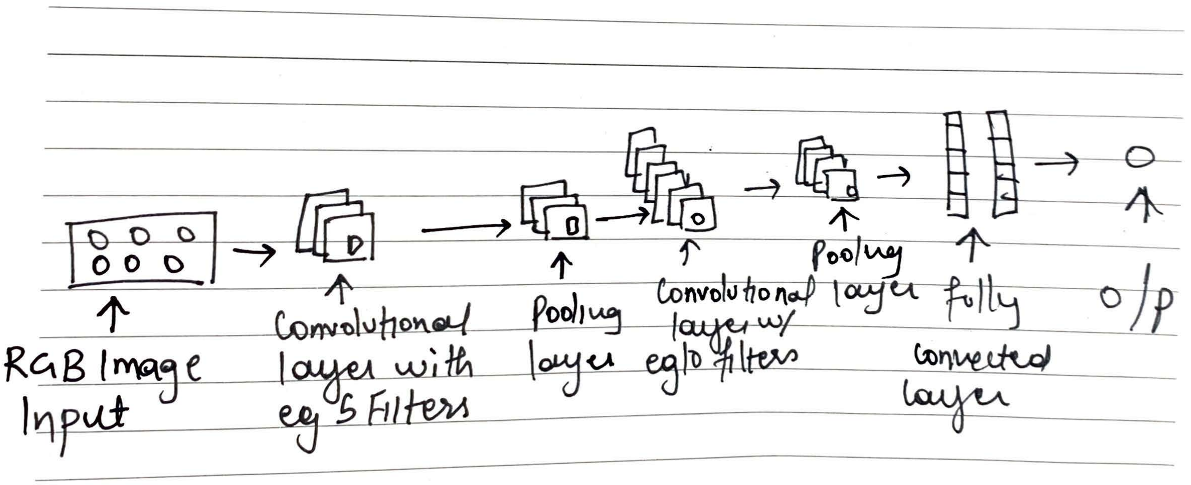
$$y = \sum_{i=1}^{K} W(x_o, x_i) y_i$$

For classification problems, the maximum of the above equation is taken for each class variables.
It is clear from the above discussion that when k>1, one can naturally define the standard deviation for predictions in regression tasks using:

$$error\ bar = \mp \sqrt{\frac{1}{K-1} \sum_{i=1}^{K} (y - y_i)^2}$$

3. **Please draw the diagram of Convolutional Neural Networks (CNN). Then explain the functionality of each layer of CNN. Name several latest algorithms of CNN (e.g., AlexNet). (10 points)**

Answer:


RGB Image Input
Convolutional layer with eg 5 Filters
Pooling layer
Convolutional layer w/ eg 10 filters
Pooling layer
fully connected layer
o/p

Convolutional Neural Networks are used to do image recognition, image classification, face recognition, etc.

Basically, in CNN we take in an image, processes it and then classifies it under certain categories.

Each input image will pass it through a series of convolution layers with filters (Kernels), Pooling, fully connected layers and apply Softmax function to classify an object with probabilistic values between 0 and 1.

Convolutional Layer:

Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel. Convolution of an image with different filters can perform operations such as edge detection, blur and sharpen by applying filters.

Non-Linearity (ReLU):

ReLU stands for Rectified Linear Unit for a non-linear operation. The output is ƒ(x) = max (0, x). ReLU's purpose is to introduce non-linearity in our Convolutional network. Since, the real-world data would want our network to learn would be non-negative linear values.

Pooling Layer:
Pooling layers section would reduce the number of parameters when the images are too large. Spatial pooling also called subsampling or down-sampling which reduces the dimensionality of each map but retains important information. Spatial pooling can be of different types:

- Max Pooling
- Average Pooling
- Sum Pooling

Max pooling takes the largest element from the rectified feature map. Taking the largest element could also take the average pooling. Sum of all elements in the feature map call as sum pooling.

Fully Connected Layer:
The layer we call as FC layer is where we flattened our matrix into vector and feed it into a fully connected layer like a neural network. Finally, we have an activation function such as softmax or sigmoid to classify the outputs as cat, dog, car, truck etc.

List of CNN Algorithms –

1. **LeNet**
2. **AlexNet**
3. **VGGNet**
4. **GoogLeNet**
5. **ResNet**
6. **ZFNet**

4. **When training deep networks using Backpropagation, one difficulty is so-called "diffusion of gradient", i.e., the error will attenuate as it propagates to early layers. Please explain how to address this problem. (8 points)**

   Answer:

   The solution is to use other activation functions instead of sigmoid, such as **ReLU**, which doesn't cause a small derivative.

   Residual networks are another solution, as they provide residual connections straight to earlier layers. This residual connection doesn't go through activation functions that squashes the derivatives, resulting in a higher overall derivative.

   Batch normalization layers can also resolve the issue. We know that the problem arises when a large input space is mapped to a small one, causing the derivatives to disappear. Batch normalization reduces this problem by simply normalizing the input so that it doesn't reach the outer edges of the sigmoid function. It normalizes the input so that most of it falls in the region where the derivative isn't too small.