Interview Questions

Question: Imagine you are tasked with designing an ANN-based system for a self-driving car that needs to process real-time data from multiple sensors to navigate through urban environments. Describe the architecture of the neural network you would propose. Include details on the type of layers, the number of nodes in each layer, and the activation functions you would use. How would you ensure that the network can effectively handle varying lighting and weather conditions?

1. Input Layer:
    - The input layer will receive real-time data from multiple sensors such as cameras, LiDAR, radar, and GPS.
    - Each sensor's data will be pre-processed and fed into the network.
2. Convolutional Layers:
    - Use convolutional layers to extract hierarchical features from visual sensor data (e.g., images from cameras).
    - Apply ReLU activation functions to introduce non-linearity.
3. Pooling Layers:
    - Add pooling layers to reduce spatial dimensions while retaining important information.
4. Fully Connected Layers:
    - Incorporate dense (fully connected) layers to process features from convolutional layers.
    - Use ReLU activation functions in these layers as well.
5. Output Layer:
    - The output layer will produce control outputs for steering, acceleration, and braking.
    - Apply appropriate activation functions based on the specific control task (e.g., linear for steering, sigmoid for throttle/brake).

Number of Nodes in Each Layer:

- Input Layer:
    - Number of nodes depends on the dimensions of sensor data (e.g., image resolution, GPS coordinates).
- Convolutional Layers:
    - Typically, start with 32 to 128 nodes per layer, increasing in depth for complex feature extraction.
- Pooling Layers:
    - No neurons in pooling layers as they perform down-sampling.

- Fully Connected Layers:
  - Experiment with 128 to 512 nodes per layer, balancing model complexity and performance.
- Output Layer:
  - Number of nodes corresponds to the number of control outputs (e.g., steering angle, throttle, brake).

Activation Functions:

- Convolutional and Fully Connected Layers:
  - Use ReLU activation functions for hidden layers to introduce non-linearity and improve model training.
- Output Layer:
  - Apply appropriate activation functions based on the control task (e.g., linear activation for steering, sigmoid for throttle/brake).

Handling Varying Lighting and Weather Conditions:

- Data Augmentation:
  - Apply techniques like random brightness adjustment, rotation, and flipping to generate diverse training data, helping the network generalize better to varying conditions.
- Transfer Learning:
  - Utilize pre-trained models on large datasets (e.g., ImageNet) to bootstrap learning and adapt to different visual environments.
- Regularization:
  - Implement dropout or L2 regularization to prevent overfitting and enhance model robustness to noise in sensor data.

By implementing this ANN architecture with appropriate activation functions and handling techniques for varying conditions, the self-driving car system can effectively navigate urban environments, adapting to changing lighting and weather conditions for safe and reliable operation.

Question: You are involved in creating an ANN model to assist in diagnosing diseases from complex medical images, such as MRIs or CT scans. Outline the design of your ANN, specifying the types of layers and activation functions that would be most suitable for this task. How would you train your network to differentiate between very subtle variations in medical images that indicate different stages of a disease? Describe the loss function you would choose and the rationale behind this choice, considering the critical nature of accurate medical diagnostics.

Step-by-step

1. Input Layer
    a. This will take images from CT scan or MRI
2. Convolutional layer
    a. In here we will extract feature from the images using learnable filters.
    b. Use the Rectified Linear Unit (ReLU) activation function for introducing nonlinearity.
3. Pooling Layer
    a. Reduce spatial dimensions while retaining essential information.
4. Dense Layer:
    a. Aggregate features learned from convolutional layers.
5. Output Layer:
    a. Consist of multiple classes or probability of disease so SoftMax computation is used.

Training for Subtle variation in Images:

1. Data Augmentation
    a. Apply transformations to generate diverse training samples
2. Transfer Learning
    a. Start with pre trained Convolutional Neural Network and fine-tunre on medical image data
3. Regularization
    a. User dropout or L2 regularization to prevent overfittion

I would choose Cross Entropy Loss function because it works well with classification algorithm.

From the abovementioned solution I think we can make an Effective ANN model that can effectively differentiate subtle variations in medical images, enabling accurate disease diagnosis.