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?

Answer:

The Self-driving car has many inputs:

1. Cameras(1 to N)
2. LiDAR
3. Speedometer
4. GPS

The output would be:

1. Steering
2. Braking
3. Acceleration

So we would need three outputs. Since we are dealing with images and other sensory data, we can use a Convolution Neural Network(CNN).

The Network will be designed as follows:

1. Input Layer: The input layer would take in the sensory inputs.
2. Convolution Layers: This layers would build more features from the list of camera inputs and regain more features from the images. There would be three to four layers of 32, 64, 128, 256 filters to extract the edges, shapes and other road and object characteristics from the visual and spatial data.
3. Neural Network: After the convolution layer, there would be a ANN to detect which actions to take in terms of braking, speeding or steering.
4. Output layer: The output layer would have three different output with softmax activation function to handle steering, braking and speeding. We can only do one of three things so Softmax would be the best activation function.

Of course, this only gives us which action to take. We can expand onto this by adding 3 more neural networks and combining them with Ensemble learning to make adequate decision on how much to steer, brake or speed.

For handling several weather and lighting conditions, we can :

1. Data Augmentation: We can simulate various weather conditions like rain, snow and storms and lighting conditions in virtual environments like Unity3D, Unreal Engine. And train on those data.
2. Dynamic adjustments: We can introduce various feedback loops within the network that adjust sensitivity based on real-time assessment of visibility and sensor-ability. For example, in case of poor visibility, the system could rely more on radar and GPS than visual inputs from camera and LiDAR.

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.

Answer:

For an ANN for MRIs and CT scans in case of identifying a disease, we can use a Convolution Neural Network(CNN).

The network would be as follows:

1. Input layer: The input would be an image of 512x512 or 1024x1024px image, depending on resolution with a processing step of normalization of images from 0 to 1.
2. Convolution Layers: The Convolution Layers would include multiple layers of varying kernels like 32, 64 and 128 to extract different features from the images like edges, shapes and patterns. Each convolution layer would use ReLU activation function to introduce non-linearity and batch normalization to speed up training and reduce overfitting.
3. Pooling Layers: Max Pooling layers would be interspersed between convolution layer to reduce dimensionality and to increase the field of view of higher layers, which helps in detecting features in larger regions of the image.
4. Dense Layer: After flattening the previous layers output, multiple layers of 1024 neurons each would integrate the high level features extracted from the convolutions layers. These layers would also use ReLU activation functions.
5. Output layer: The output layer would consist of neurons equal to the number of different stages of the disease. It would use a soft-max activation function to determine the stage of the disease.

Training Strategy:

1. Data Augmentation: to train the model, we can skew and transform various disease images to simulate different stages of disease and the patient condition to train the model.
2. Advanced techniques: We can use transfer learning methods, like using vast library of medical images to train the model on different diseases and then fine tune the model to learn the different stages of a disease.

For this multiclass classification CNN, we could use:

Cross-entropy loss function: This loss function is appropriate for multi-class classification problems. It is effective because it penalizes incorrect classifications heavily when the model is confident about its wrong predictions, which is crucial for medical diagnostics where mistakes can have serious consequences.