Week 12 - Extra Credit

During the Week 12 lecture, we discussed different types of Neural Networks. For extra credit, I would like for you to write up a summary of one type of Neural Network in your own words - how does it work, how is it used, any gotchas in using it. Add a link to an IPython notebook in your Week 12 folder to your post. The IPython notebook should have an example of the Neural Network being trained and used.

CNN NEURAL NEYWORK.

What is a CNN?
A Convolutional Neural Network (CNN) is a deep learning model that is particularly useful for image categorization and pattern recognition applications. Unlike standard neural networks, CNNs employ convolutional layers to scan and recognize features (such as edges, textures, patterns) in input data, particularly images.

CNNs typically include the following layers:

1. Convolutional Layers: These apply learnable filters (kernels) to the input image. Each filter activates specific features, such as edges, textures, or patterns. The outcome is a feature map that emphasizes the presence of specific visual features.

2. Activation Function (usually ReLU): After convolution, the output is sent through a non-linear function (e.g. ReLU) to incorporate non-linearity into the model, allowing it to learn complicated patterns.

3. Pooling Layers: Pooling minimizes the spatial dimensions (width and height) of feature maps. The most popular is Max Pooling, which keeps the most significant characteristic in a certain location. Pooling minimizes the amount of parameters, hence preventing overfitting and increasing computation performance.

4. Fully Connected Layers (Dense Layers): After downsampling by convolution and pooling, feature maps are flattened and passed into dense layers. These function as a decision-making layer.

5. Output Layer: Classification tasks commonly utilize a softmax activation function to forecast class probabilities.

CNNs are extremely effective and widely utilized in:

1.Image classification (e.g., identifying objects, people, and animals in photographs)
2. Medical Image Analysis (e.g., tumor identification and retinal scan analysis)
3. Face Recognition
4.Self-driving cars (e.g., pedestrian detection and lane markings)
5.Satellite Image Analysis
6.Document Analysis (for example, handwritten digit recognition)

Gotchas in Using CNNs: -

Overfitting: Overfitting is a potential drawback of using CNNs due to their ability to predict complicated patterns on little datasets. This can be minimized using strategies like as data augmentation, dropout, and transfer learning.

Computational requirements: Training CNNs on large datasets can be computationally intensive. GPUs are generally required for efficient training.

Interpretability: CNNs can be "black boxes"; understanding why they made a certain decision might be difficult without supplementary tools such as Grad-CAM or saliency maps.

CNNs are sensitive to the size and format of their input pictures. Preprocessing and normalization are necessary.