Class 1: Overview of Deep Learning

1. Introduction to Deep Learning

Deep learning is a subfield of **Artificial Intelligence (AI)** that leverages neural networks with multiple layers to learn complex patterns from data. It is at the core of major advancements in computer vision, natural language processing, and autonomous systems.

1.1. Key Characteristics of Deep Learning

- **Layered Structure:** Consists of an input layer, multiple hidden layers, and an output layer.
- **End-to-End Learning:** The model learns directly from raw data, minimizing manual feature engineering.
- **Data-Driven:** Requires large amounts of data for effective training and performance.
- **High Computational Power:** Deep learning models benefit from the use of GPUs and TPUs for faster training.

1.2. Common Applications of Deep Learning

- Image Recognition: Used in facial recognition systems, medical imaging, and surveillance systems.
- Natural Language Processing (NLP): Powers applications like chatbots, language translation, and sentiment analysis.
- Autonomous Systems: Plays a critical role in the development of self-driving cars, drones, and robotics.

2. What Are Neurons?

In the context of deep learning, **neurons** represent the fundamental units of computation. They are inspired by biological neurons and are designed to process and transmit information in neural networks.

2.1. Structure of a Neuron

- **Input:** Receives inputs from the previous layer or external sources.
- **Weights:** Each input has an associated weight that signifies its importance.
- **Bias:** An additional parameter that allows the model to shift the activation function curve.

- **Activation Function:** Determines whether the neuron's output should be activated or not.

2.2. Steps of Neuron Processing

1. Accepts inputs from other neurons or external data sources.
2. Multiplies each input by its corresponding weight.
3. Adds the bias term to the weighted sum.
4. Passes the sum through an activation function.
5. Outputs the result to the next layer or as final output.

3. Neural Networks

A **neural network** is a collection of interconnected neurons, organized into layers, that transform input data into meaningful outputs.

3.1. Components of a Neural Network

- **Input Layer:** Receives raw data or pre-processed data.
- **Hidden Layers:** Perform computations to extract complex patterns.
- **Output Layer:** Provides the final predictions or classifications.

3.2. Types of Neural Networks

- **Feedforward Neural Network (FNN):** The simplest form of neural network where connections do not form cycles.
- **Convolutional Neural Network (CNN):** Specialized in processing grid-like data such as images.
- **Recurrent Neural Network (RNN):** Designed for sequential data processing, suitable for NLP tasks.
- **Generative Adversarial Network (GAN):** Used for generating new data that resembles the input data.

4. Activation Functions

Activation functions introduce non-linearity into neural networks, enabling them to learn complex patterns. They play a crucial role in the training and prediction processes of neural networks.

4.1. Types of Activation Functions

- **Sigmoid:** Converts inputs to a range of [0, 1], suitable for binary classification tasks.

- **Tanh (Hyperbolic Tangent):** Scales values between [-1, 1], often used in recurrent networks.
- **ReLU (Rectified Linear Unit):** Outputs zero for negative inputs and linear values for positive inputs, helping to overcome vanishing gradient issues.
- **Leaky ReLU:** Allows a small, non-zero gradient for negative inputs to address dead neuron issues.
- **Softmax:** Converts logits into probabilities, suitable for multi-class classification.

5. Forward Propagation

Forward propagation is the process of passing inputs through the network layer by layer, generating outputs at each step until the final prediction is made.

5.1. Steps of Forward Propagation

1. Inputs are passed to the input layer.
2. Each hidden layer processes the inputs based on weights and biases.
3. Activation functions are applied to the outputs of each layer.
4. The final output is generated by the output layer.

5.2. Importance of Forward Propagation

- **Feature Extraction:** Each hidden layer extracts different features from the input data.
- **Prediction:** The network generates predictions based on learned patterns.
- **Error Calculation:** The output is used to calculate errors for backpropagation.

6. Backward Propagation

Backward propagation is the process of updating the weights of the network by calculating the gradient of the error with respect to each weight.

6.1. Steps of Backward Propagation

1. Calculate the error between the predicted and actual outputs.
2. Compute the gradient of the loss with respect to each weight.
3. Adjust weights based on the gradients to minimize the loss.

6.2. Importance of Backward Propagation

- **Model Optimization:** Helps minimize the error by adjusting weights in the network.
- **Learning Efficiency:** Enables the model to learn from errors and improve predictions.

Conclusion

Deep learning has transformed the fields of computer vision and natural language processing, allowing for significant advancements. By understanding the key concepts like neurons, neural networks, activation functions, forward propagation, and backward propagation, practitioners can build efficient models that solve complex problems.