TASK 21

SHERAZ BIN TAHIR

MAIL: BAIGS543@GMAIL.COM

Backpropagation in DL

Backpropagation is a fundamental algorithm used in training neural networks, particularly in deep learning. It is the method through which the network learns by adjusting its weights to minimize the error in predictions.

How Backpropagation Works

1. Forward Pass:

The input data is fed into the neural network, and a forward pass is performed to compute the output (predictions). This involves passing the input through each layer of the network and applying the associated weights and activation functions to produce the final output.

2. Loss Calculation:

The output is compared to the true labels (targets) using a loss function (e.g., Mean Squared Error for regression, Cross-Entropy for classification). The loss function quantifies how far off the network's predictions are from the actual results.

3. Backward Pass (Backpropagation):

Backpropagation starts with the calculation of the gradient of the loss function with respect to the output of the network. This gradient is then propagated backward through the network.

The key idea is to compute the gradient of the loss function with respect to each weight in the network. This involves applying the chain rule of calculus to calculate the derivative of the loss with respect to each weight, layer by layer, starting from the output layer and moving backward to the input layer.

4. Weight Update:

Once the gradients are calculated, the weights are updated using a method like Gradient Descent. In Gradient Descent, weights are adjusted in the opposite direction of the gradient to minimize the loss.

The weight update rule can be written as:

$$\mathbf{W}_{\text{New}} = \mathbf{W}_{\text{old}} - \eta \times \partial L/\partial W$$

where:

- W**new** is the updated weight,
- W**old** is the current weight,
- η is the learning rate (a small positive value),
- **∂L/∂W** is the gradient of the loss with respect to the weight.

5. Iteration:

The forward pass, loss calculation, backward pass, and weight update steps are repeated for many iterations (epochs) until the network's predictions converge to a satisfactory level of accuracy.

Importance of Backpropagation

- **Efficiency:** Backpropagation makes it feasible to train deep neural networks by efficiently calculating gradients.
- **Scalability:** The algorithm scales well with the complexity of the network, making it suitable for deep learning models with many layers.
- **Optimization:** It allows for the optimization of a network's parameters, enabling the model to learn from data and improve over time.

Challenges and Solutions

- **Vanishing/Exploding Gradients:** In very deep networks, gradients can become very small (vanishing) or very large (exploding), making training difficult. Solutions include techniques like gradient clipping, using different activation functions (e.g., ReLU), and initializing weights carefully.
- **Overfitting:** Backpropagation can lead to overfitting, where the model performs well on training data but poorly on unseen data. Regularization techniques such as dropout, L2 regularization, and data augmentation can help mitigate this.