Deep Learning Course – L 6 - Detailed Notes

📉 Gradient Descent and Learning Rate in Neural Networks

Gradient Descent is one of the most essential optimization algorithms used in machine learning and deep learning. In this lecture, we'll explore **how gradient descent works** and the **critical role of the learning rate** in optimizing a model's parameters.

🔧 What Is Gradient Descent?

Objective:
To **minimize the loss function** by adjusting a model's parameters (weights and biases), thus improving performance.

🧠 How It Works:

1. Start with random values for weights and biases.
2. Compute the **gradient** (slope) of the loss function with respect to the parameters.
3. **Update parameters** in the direction that reduces the loss (opposite to the gradient).
4. Repeat until the model **converges** to an optimal solution.

💎 **Key Takeaway:** Gradient descent gradually improves model accuracy by minimizing error.

□ Gradient Descent Formula

$$w = w - \alpha \frac{\partial J(w)}{\partial w}$$

- **w**: Weight (parameter)
- **α (alpha)**: Learning rate – controls the step size
- **∂J(w)/∂w**: Gradient of the loss function with respect to w

👉 If the gradient is **positive**, the weight **decreases**.
👉 If the gradient is **negative**, the weight **increases**.

💎 **Key Takeaway:** The gradient directs how much and which way to update the weights.

📊 Visualizing Gradient Descent

- Plot the **loss function J(w)** on the Y-axis and the **parameter w** on the X-axis.
- The lowest point on the curve represents the **global minimum**.

Example:

- If weight starts on the left side of the curve, it moves right.
- If on the right, it moves left—always heading downhill to the minimum.

💎 **Key Takeaway:** Gradient descent follows the path of steepest descent toward the minimum.

✅ Convergence of Gradient Descent

Convergence happens when the gradient ≈ 0, meaning updates no longer improve the model.

When Gradient = 0:

w=w−0⇒w=ww = w - 0 \Rightarrow w = ww=w−0⇒w=w

💎 **Key Takeaway:** A zero gradient means the algorithm has reached the minimum point.

🚀 Learning Rate (α): Controlling Step Size

Definition: The **learning rate** determines how large a step gradient descent takes during each update.

✅ Small Learning Rate:

- **Pros**: More stable convergence
- **Cons**: Slower training

⚠️ Large Learning Rate:

- **Pros**: Faster convergence
- **Cons**: Risk of overshooting or diverging

Examples:

- α=0.1\alpha = 0.1α=0.1: Moderate step size
- α=2\alpha = 2α=2: Risky, may overshoot

💎 **Key Takeaway:** The learning rate must be carefully tuned to avoid unstable training.

Global vs Local Minima

- **Global Minimum**: Lowest point in the loss curve (ideal solution).
- **Local Minimum**: A low point that is **not the global minimum**.

🧠 Challenge:

Gradient descent can get **stuck in a local minimum** if the loss function is non-convex.

🔧 Solutions:

- Use **momentum**, **RMSProp**, or **Adam Optimizer** to escape local minima.

💎 **Key Takeaway:** Choosing the right optimization strategy helps reach the best solution.

💡 Gradient Descent in Neural Networks

Step-by-Step Process:

1. Initialize weights and biases randomly.
2. Calculate the loss for current parameters.
3. Compute gradients using backpropagation.
4. Update parameters using the gradient descent rule.
5. Repeat until convergence.

💎 **Key Takeaway:** This iterative process forms the backbone of training neural networks.

Practical Example: Optimizing a Single Weight

Scenario:

You want to minimize loss for a single parameter **w**.

Steps:

1. Randomly initialize **w**
2. Compute ∂J(w)∂w\frac{\partial J(w)}{\partial w}∂w∂J(w)
3. Update:

   w=w−α·∂J(w)∂ww = w - \alpha \cdot \frac{\partial J(w)}{\partial w}w=w−α·∂w∂J(w)

4. Repeat until convergence

💎 **Key Takeaway:** This logic extends to optimizing all weights in a complex network.

🎯 Learning Rate as a Hyperparameter

Definition:
A **hyperparameter** is set before training and **not learned** from data.

Learning Rate's Role:

- Controls convergence speed
- Prevents divergence

Typical Values:
0.001, 0.01, 0.1

💎 **Key Takeaway:** The learning rate is a critical hyperparameter that impacts training success.

🧠 Final Thoughts

Gradient descent is a **powerful optimization algorithm** essential for training models. The **learning rate** controls the optimization path, influencing whether your model converges effectively or fails.

By mastering these concepts, you lay a strong foundation for building and training neural networks.

👉 **Next Up:** We'll dive into **advanced optimization techniques** like Momentum, Adam, and RMSProp used in deep learning frameworks.