FL Introduction

Why the Global Model Cannot Fetch Data from Weights

- Weights Represent Patterns, Not Data
- Aggregated Updates
- Encryption and Privacy Mechanisms

Potential Risks of Extracting Data from Weights

- **Model Inversion Attacks:**
- Attackers may exploit the global model to reconstruct approximate representations of the training data. This involves using the model’s weights and outputs to infer what the input data might look like.
- Example: In image classification, attackers might reconstruct blurry versions of training images.

Potential Risks of Extracting Data from Weights

- **Gradient Leakage (or Data Leakage):**
- Gradients shared during training might encode information about the training data.
- Example: In scenarios where a single client contributes, it might be possible to reverse-engineer the data used to compute the gradients.
- Mitigation: **Secure aggregation** and **gradient clipping*** reduce this risk.

****Gradient Clipping*** *is a technique used in machine learning, particularly in gradient-based optimization methods (like stochastic gradient descent), to prevent the* ***exploding gradient problem****, where gradients become excessively large during backpropagation. This can destabilize training or lead to numerical errors.*

Potential Risks of Extracting Data from Weights

- **Adversarial Aggregation:**
- If the server or an adversary modifies the global model or manipulates aggregation, it could exploit vulnerabilities to infer data patterns from client updates.

Why Gradient Clipping is Important

- **Prevents Exploding Gradients**:
- Exploding gradients occur when gradients grow exponentially as they propagate backward, especially in deep networks or recurrent neural networks (RNNs).
- This leads to instability, where parameter values become too large or result in NaN (not a number) errors.
- **Stabilizes Training**:
- Ensures that the updates to model parameters are not overly large, leading to smoother convergence.

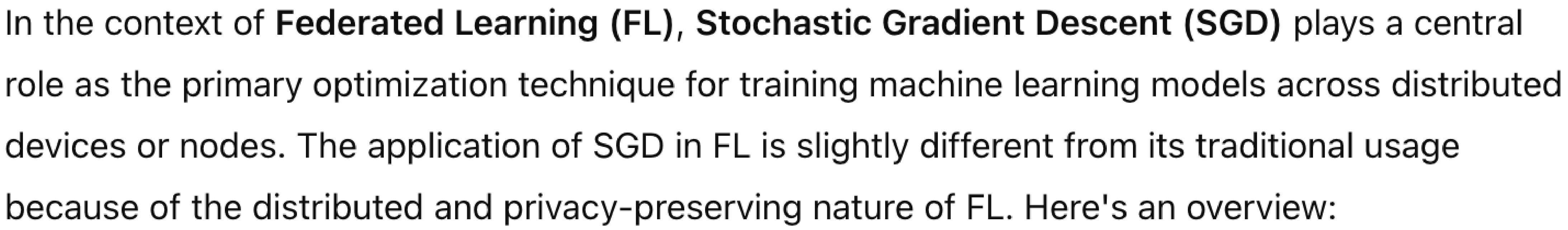
Why Gradient Clipping is Important

- **Improves Convergence**:
- Helps the model train faster and more effectively by avoiding unstable weight updates.
- **Enhances Robustness in Federated Learning**:
- In federated learning, where updates come from multiple clients, gradient clipping prevents any single client with anomalously large gradients from negatively impacting the global model.

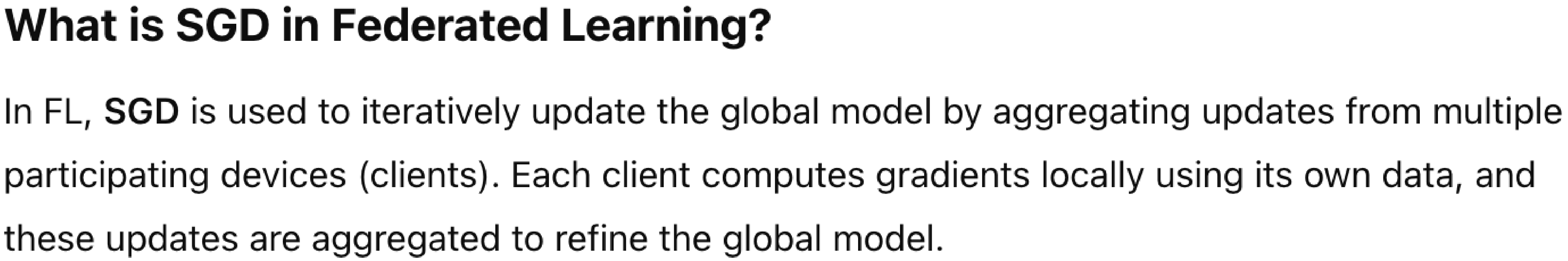
Types of Gradient Clipping

- **Global Norm Clipping**:
- Scales gradients based on the norm of all gradients across the model.
- **Per-Layer Clipping**:
- Applies different clipping thresholds for gradients at different layers of the network.
- **Element-Wise Clipping**:
- Clips each individual gradient element to lie within a specified range.

Stochastic Gradient Descent (SGD)



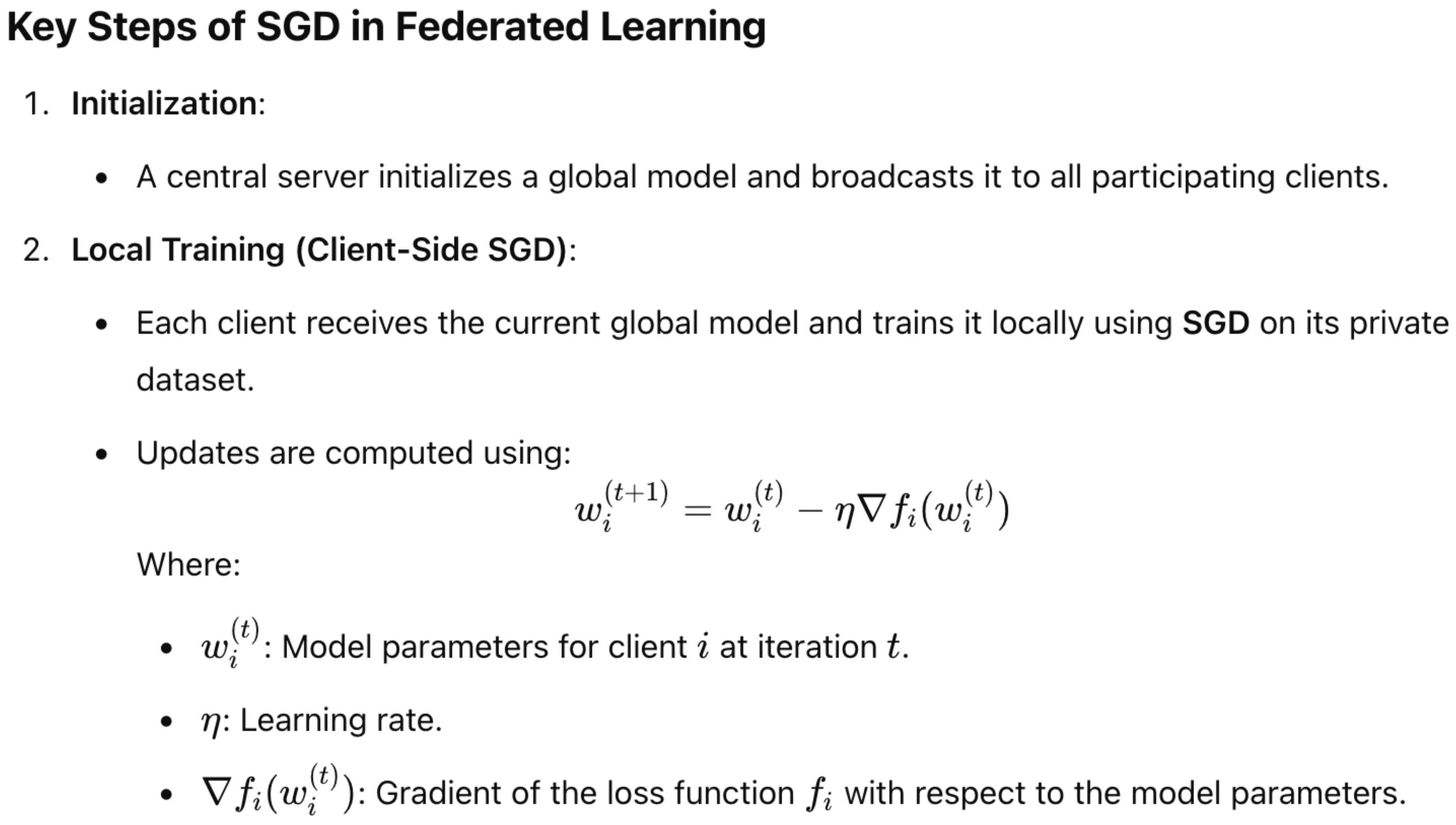
In the context of **Federated Learning (FL)**, **Stochastic Gradient Descent (SGD)** plays a central role as the primary optimization technique for training machine learning models across distributed devices or nodes. The application of SGD in FL is slightly different from its traditional usage because of the distributed and privacy-preserving nature of FL. Here's an overview:



What is SGD in Federated Learning?

In FL, **SGD** is used to iteratively update the global model by aggregating updates from multiple participating devices (clients). Each client computes gradients locally using its own data, and these updates are aggregated to refine the global model.

Stochastic Gradient Descent (SGD)



Key Steps of SGD in Federated Learning

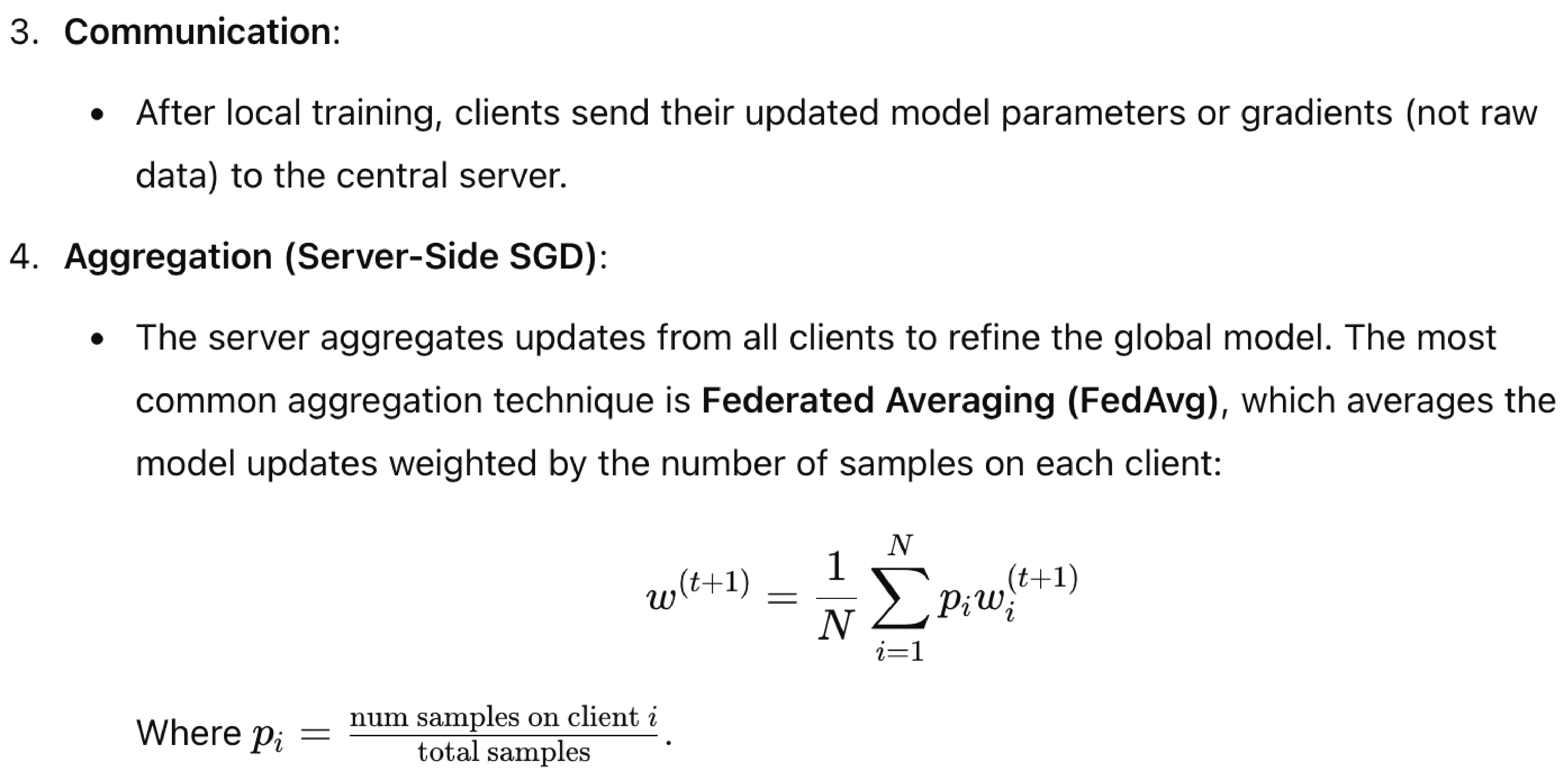
1. **Initialization**:
   - A central server initializes a global model and broadcasts it to all participating clients.
2. **Local Training (Client-Side SGD)**:
   - Each client receives the current global model and trains it locally using **SGD** on its private dataset.
   - Updates are computed using:

$$w_i^{(t+1)} = w_i^{(t)} - \eta \nabla f_i(w_i^{(t)})$$

     Where:

     - $w_i^{(t)}$: Model parameters for client i at iteration t.
     - η: Learning rate.
     - $\nabla f_i(w_i^{(t)})$: Gradient of the loss function f_i with respect to the model parameters.

Stochastic Gradient Descent (SGD)

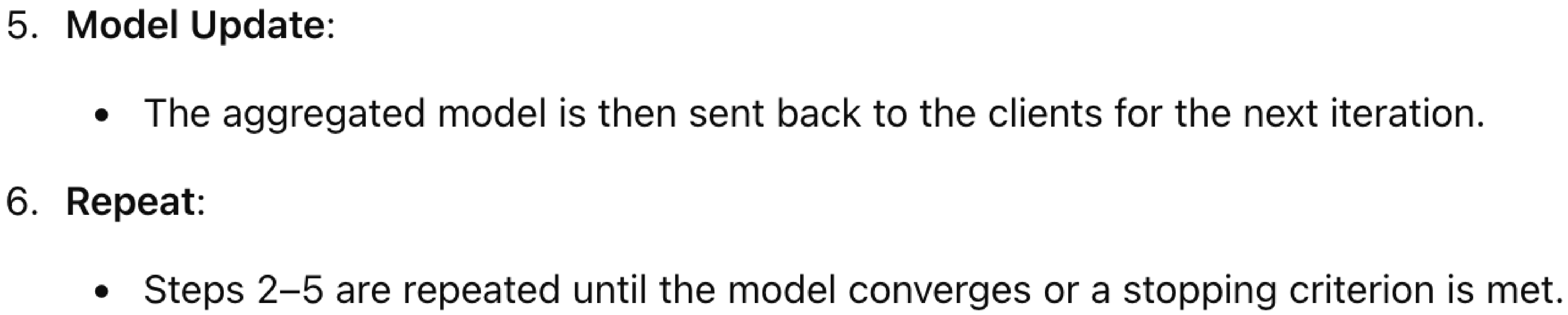


3. **Communication**:
    - After local training, clients send their updated model parameters or gradients (not raw data) to the central server.
4. **Aggregation (Server-Side SGD)**:
    - The server aggregates updates from all clients to refine the global model. The most common aggregation technique is **Federated Averaging (FedAvg)**, which averages the model updates weighted by the number of samples on each client:

$$w^{(t+1)} = \frac{1}{N} \sum_{i=1}^{N} p_i w_i^{(t+1)}$$

Where $p_i = \frac{\text{num samples on client } i}{\text{total samples}}$.

Stochastic Gradient Descent (SGD)



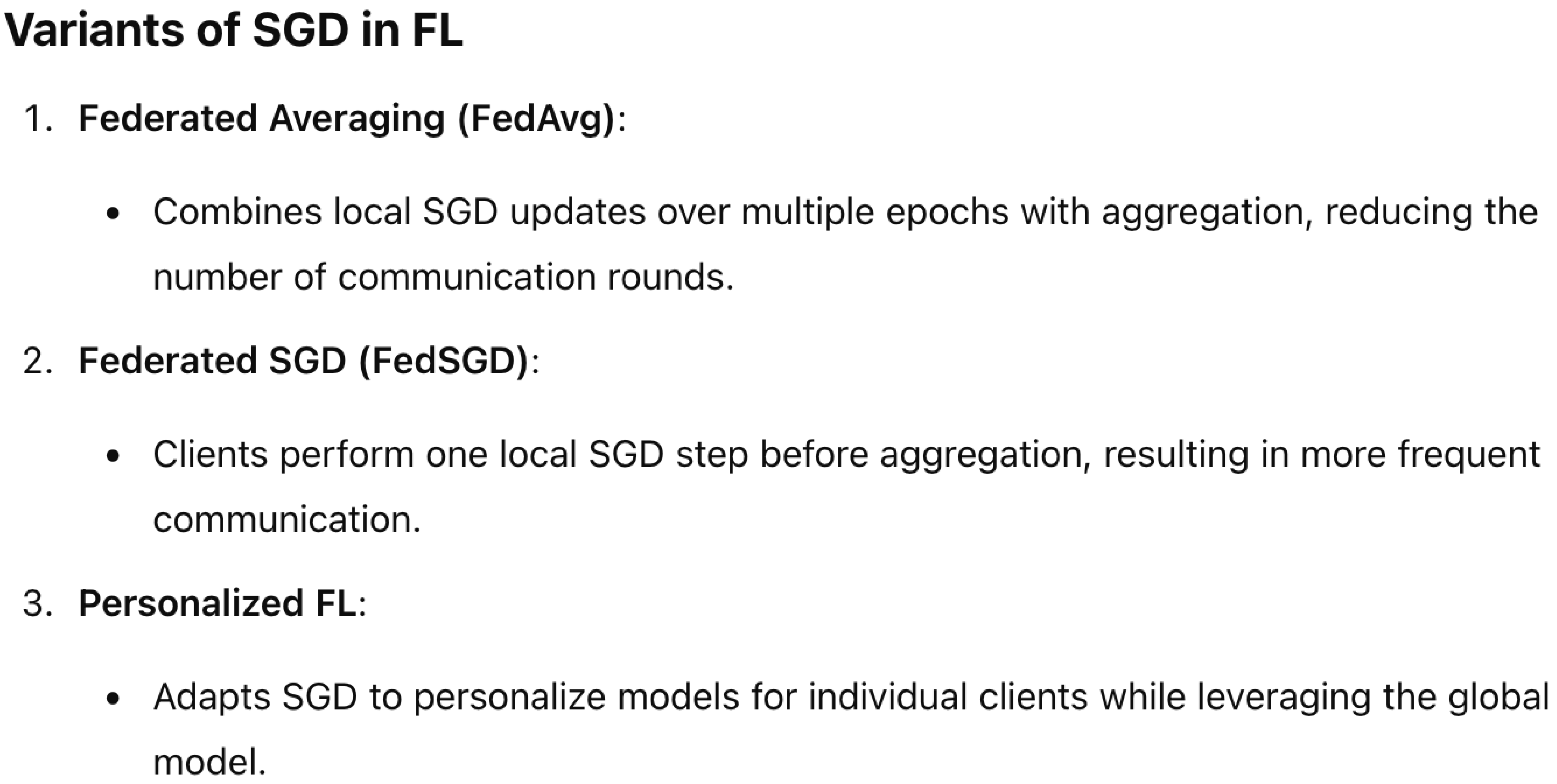
5. **Model Update**:

    - The aggregated model is then sent back to the clients for the next iteration.

6. **Repeat**:

    - Steps 2–5 are repeated until the model converges or a stopping criterion is met.

Stochastic Gradient Descent (SGD)



Variants of SGD in FL

1. **Federated Averaging (FedAvg)**:
   - Combines local SGD updates over multiple epochs with aggregation, reducing the number of communication rounds.
2. **Federated SGD (FedSGD)**:
   - Clients perform one local SGD step before aggregation, resulting in more frequent communication.
3. **Personalized FL**:
   - Adapts SGD to personalize models for individual clients while leveraging the global model.