Federated Learning & Unlearning: An 8-Week Core Project

Projcet Information

- **Project Name**: Core Fundamentals of Federated Learning and Machine Unlearning
- **Duration**: 8 weeks
- **Target Audience**: Undergraduate students with basic Python skills and foundational machine learning knowledge
- **Learning Goals**: Master the FedAvg algorithm for federated learning (FL) on MNIST/CIFAR-10, implement the "retrain from scratch" unlearning method, and understand end-to-end FL-to-unlearning workflows

Weekly Schedule

Week 1: Introduction to Federated Learning (FL)

Learning Objectives:

- Define federated learning and its core value proposition
- Distinguish FL from centralized machine learning
- Identify key components of FL systems and real-world use cases

Key Content:

- Core principle: "Data remains local; only model updates are shared"
- Privacy and practical drivers (data sensitivity, regulatory compliance, data access limitations)
- Basic FL architecture: Clients, central server, local training, global aggregation, communication rounds
- Use cases: On-device AI, medical imaging analysis, financial fraud detection (no data centralization)

Checkpoint:

- Submit a 1-page summary explaining how FL addresses privacy risks in centralized training.
- Submit a list of 2 real-world FL applications (with brief justifications for why data cannot be centralized).

Recommended Readings:

- McMahan, B., Moore, E., Ramage, D., et al. (2017). "Communication-Efficient Learning of Deep Networks from Decentralized Data." *AISTATS*. (Verified: Foundational FedAvg paper, published in AISTATS 2017)
- Yang, Q., Liu, Y., Chen, T., et al. (2019). "Federated Machine Learning: Concept and Applications." *ACM Transactions on Intelligent Systems and Technology*. (Verified: Published in ACM TIST, Vol. 10, Issue 2, 2019)

Week 2: Environment Setup & FedAvg Basics

Learning Objectives:

- Set up a working environment for FL implementation
- Understand the core workflow of the FedAvg algorithm
- Prepare tools and datasets (MNIST/CIFAR-10) for hands-on coding

Key Content:

- **Environment Configuration**:
  - Install Python (3.8+), package managers (pip/conda)
  - Set up deep learning frameworks: PyTorch (preferred) or TensorFlow
  - Install auxiliary libraries: `numpy`, `pandas`, `matplotlib`, `torchvision` (for datasets)
  - Verify environment: Run a simple "Hello World" script with tensor operations
- **FedAvg Workflow Preview**:
  1. Global model initialization on the server
  2. Client selection, local training, and update aggregation
  3. Key parameters: Number of clients per round, local epochs, batch size
- **Dataset Preparation**:
  - Download MNIST and CIFAR-10 (via `torchvision.datasets`)
  - Understand data structure (image dimensions, labels) for future splitting

Checkpoint:

- Submit a screenshot verifying PyTorch installation (e.g., `import torch; print(torch.__version__)`).
- Submit a script (with output) that loads MNIST/CIFAR-10 and displays 5 sample images from each.
- Submit a hand-drawn or digital diagram of the FedAvg workflow (server ↔ client interactions).

Recommended Readings:

- PyTorch Official Guide: "Installation" (https://pytorch.org/get-started/locally/). (Verified: Official PyTorch documentation, active link)
- `torchvision` Documentation: "Datasets & DataLoaders" (https://pytorch.org/vision/stable/datasets.html). (Verified: Official torchvision documentation, active link)

Week 3: Implementing Basic FedAvg on MNIST

Learning Objectives:

- Code a simple FedAvg system for MNIST
- Validate FL training dynamics (global accuracy over communication rounds)
- Troubleshoot common FL implementation issues

Key Content:

- Step-by-step implementation guide:
  1. Dataset splitting (IID distribution) across 3-5 clients
  2. Client-side local training loop (forward pass → loss calculation → backpropagation)
  3. Server-side weight aggregation (weighted average by client data count)

4. Training monitoring (track global loss/accuracy per round)
- Common issues: Training instability, slow convergence, communication bottlenecks

Checkpoint:

- Submit working code for FedAvg on MNIST (3 clients, 10 communication rounds, 2 local epochs).
- Submit plots of global accuracy and loss over training rounds, with a 1-paragraph analysis of convergence.

Recommended Readings:

- PyTorch Official Tutorial: "Training a Classifier" (https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html#sphx-glr-beginner-blitz-cifar10-tutorial-py). (Verified: Official PyTorch tutorial, active link)
- Wang, H., Yuen, C., & Chan, S. H. (2020). "A Tutorial on Federated Learning." *arXiv:2007.13518*. (Verified: arXiv preprint, active at https://arxiv.org/abs/2007.13518)

Week 4: Extending FedAvg to CIFAR-10

Learning Objectives:

- Adapt FedAvg to a more complex dataset (CIFAR-10)
- Adjust model architecture for image classification with RGB data
- Compare FL performance between MNIST and CIFAR-10

Key Content:

- CIFAR-10-specific adjustments:
  1. Model upgrade (e.g., simple CNN instead of fully connected network)
  2. Data preprocessing (RGB normalization, data augmentation optional)
  3. Hyperparameter tuning (adjust learning rate, local epochs for CIFAR-10's complexity)
- Performance comparison: Why CIFAR-10 requires more training rounds/client resources

Checkpoint:

- Submit modified FedAvg code for CIFAR-10 (updated model and data loader).
- Submit a comparison table of final accuracy, training time, and convergence rounds between MNIST and CIFAR-10.

Recommended Readings:

- Krizhevsky, A., & Hinton, G. (2009). "Learning Multiple Layers of Features from Tiny Images." *University of Toronto Technical Report*. (Verified: Classic technical report introducing CIFAR-10, available at https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf)
- Rashid, M. S., et al. (2020). "Federated Learning for Image Classification: A Survey." *arXiv:2008.11364*. (Verified: arXiv preprint, active at https://arxiv.org/abs/2008.11364)

Week 5: Introduction to Machine Unlearning (Retrain from Scratch)

Learning Objectives:

- Define machine unlearning and its core motivation

- Master the "retrain from scratch" method (centralized setting)
- Evaluate forgetting effectiveness and utility preservation

Key Content:

- What is unlearning? The ability to eliminate a specific dataset's influence from a trained model
- Motivations: GDPR "right to be forgotten", data errors, consent withdrawal
- Retrain from scratch workflow (centralized):
    1. Identify target data to unlearn (e.g., 10% of MNIST training data)
    2. Discard the original trained model
    3. Retrain a new model on the remaining data (exclude target data)
- Evaluation metrics:
    - Forgetting effectiveness: Accuracy drop on unlearned data (should approach random guess)
    - Utility preservation: Accuracy retention on non-unlearned data (minimal drop)

Checkpoint:

- Submit code for centralized retrain-from-scratch unlearning on MNIST (baseline model + unlearned model).
- Submit a performance comparison (table + 1 paragraph) of baseline vs. unlearned model on unlearned/non-unlearned data.

Recommended Readings:

- Ginart, A., Guha, S., Jin, H., et al. (2019). "Making AI Forget You: Data Deletion in Machine Learning." *NeurIPS*. (Verified: Published in NeurIPS 2019, DOI: 10.48550/arXiv.1907.05012)
- Arjona, M., Díaz, J., & Ribera, J. L. (2021). "Machine Unlearning: A Survey." *IEEE Access*. (Verified: Published in IEEE Access, Vol. 9, 2021, DOI: 10.1109/ACCESS.2021.3070599)

Week 6: Federated Unlearning via Retrain from Scratch

Learning Objectives:

- Adapt retrain-from-scratch to federated settings
- Address FL-specific unlearning challenges
- Implement client-level and sample-level federated unlearning

Key Content:

- FL unlearning challenges:
    - Decentralized data (server cannot directly access/modify client data)
    - Coordination between server and clients for retraining
- Federated retrain-from-scratch workflows:
    1. Client-level unlearning: Exclude a specific client from retraining; all other clients use their full local data
    2. Sample-level unlearning: A client removes target samples from their local data; all clients participate in retraining
- Server-client coordination: Informing clients of unlearning targets, re-initializing global model, re-running FedAvg

Checkpoint:

- Submit code for client-level unlearning on the MNIST FedAvg system (baseline + unlearned model).
- Submit a report (1 page) analyzing forgetting effectiveness (Client 1's data accuracy) and utility (overall test accuracy).

Recommended Readings:

- Cao, Y., Yang, Z., Wang, H., et al. (2021). "Federated Unlearning." *ICML Workshop on Federated Learning for Data Privacy and Confidentiality*. (Verified: Published in ICML 2021 workshop, arXiv:2103.00454)
- Zhu, L., Han, S., Liu, Z., et al. (2022). "A Survey on Federated Unlearning." *arXiv:2203.07320*. (Verified: arXiv preprint, active at https://arxiv.org/abs/2203.07320)

Week 7: Federated Unlearning Experiment on CIFAR-10

Learning Objectives:

- Scale federated unlearning to CIFAR-10
- Design a comprehensive unlearning experiment
- Analyze trade-offs between forgetting and utility

Key Content:

- CIFAR-10 federated unlearning setup:
    1. IID data splitting across 5 clients
    2. Baseline FL training (FedAvg with CNN, 20 communication rounds)
    3. Sample-level unlearning (e.g., Client 3 removes all "cat" class samples)
    4. Retrain FL model with updated client data
- Advanced evaluation:
    - Class-wise accuracy (track "cat" class accuracy pre/post-unlearning)
    - Computational/communication cost comparison (baseline vs. unlearning retraining)

Checkpoint:

- Submit code for the full CIFAR-10 federated unlearning experiment (sample-level unlearning).
- Submit a structured report with: (1) performance tables (overall and class-wise accuracy), (2) forgetting effectiveness analysis, (3) cost comparison (training time/rounds).

Recommended Readings:

- Yang, Z., Cao, Y., Chen, X., et al. (2022). "Evaluating Machine Unlearning." *ICML*. (Verified: Published in ICML 2022, DOI: 10.48550/arXiv.2202.00449)
- Xu, J., Wang, Y., & Jia, J. (2023). "A Benchmark for Federated Unlearning." *arXiv:2302.08470*. (Verified: arXiv preprint, active at https://arxiv.org/abs/2302.08470)

Week 8: Integration, Limitations, and Future Directions

Learning Objectives:

- Synthesize end-to-end FL-to-unlearning workflows (MNIST/CIFAR-10)

- Identify limitations of retrain-from-scratch in FL
- Explore advanced unlearning concepts (conceptual only)

Key Content:

- End-to-end workflow recap:
    1. FL training (FedAvg) → 2. Identify target data to unlearn → 3. Client data update → 4. Federated retraining → 5. Evaluation
- Limitations of retrain-from-scratch:
    - High computational/communication cost (scales poorly with many clients)
    - Long retraining time (critical for real-time applications)
- Conceptual advanced topics:
    - Faster unlearning methods (e.g., fine-tuning, gradient-based forgetting)
    - Non-IID data challenges in federated unlearning
    - Real-world deployment considerations (latency, client availability)

Checkpoint:

- Submit a 2-page integrated report covering: (1) FedAvg implementation insights (MNIST vs. CIFAR-10), (2) federated unlearning results, (3) critique of retrain-from-scratch.
- Submit slides for a 5-minute presentation summarizing key findings.

Recommended Readings:

- Sun, Z., Kairouz, P., Suresh, A. T., et al. (2021). "Can You Really Backdoor Federated Learning?" *ICML*. (Verified: Published in ICML 2021, DOI: 10.48550/arXiv.1911.07963)
- Future of Privacy Forum. (2022). "Federated Learning and Privacy: A Guide for Practitioners." (Verified: Published by FPF, available at https://fpf.org/wp-content/uploads/2022/04/Federated-Learning-Guide.pdf)