# Efficient Data Selection for Split Neural Networks

SplitNN [1] is a distributed, privacy-preserving training paradigm that partitions a model at a cut layer into a client-side subnetwork and a server-side subnetwork. Compared with Federated Learning (FL) [2], this design lowers on-device compute by keeping only a small portion of the model on each client. However, SplitNN's communication and computation still scale with (i) the size of client activations at the cut layer, (ii) the placement of the cut, and (iii) the number of clients per round [2–4]. Devising a subset-selection technique for SplitNN to potentially overcome the computation and the communication constraints seem natural. Unfortunately, most existing selection schemes either require full-model losses/gradients [5] or rely on proxy models [6, 7], neither of which is directly available or desirable in SplitNN where clients do not see the server head.

We propose a simple, generic framework that makes loss/gradient-based subset selection feasible in SplitNN by equipping each client with a lightweight auxiliary prediction head attached to its cut-layer output. This head produces local class-probability estimates and a client-local loss, enabling the client to score its own activations and select an informative subset to transmit. Only the selected activations are sent to the server; the server completes the forward/backward pass on the global head and returns gradients solely for the selected samples. Client subnetworks are updated with these returned gradients as in vanilla Split Learning (SL), while auxiliary heads are updated locally using cross-entropy on client data. Clients proceed sequentially (SL) or concurrently (SplitFed-style variants [8, 9]) without exposing raw data.

For client $k$ with local data $D_k = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ and client model $c_k$, cut-layer activations are $A_{[n]} = \{c_k(\mathbf{x}_i)\}_{i=1}^n$. Given budget $m \leq n$, we seek a subset $s = \{s_j\}_{j=1}^m \subseteq A_{[n]}$ corresponding to data $d \subseteq D_k$ that preserves training quality: $\min_{|d|=m} \mathbb{E}_{(\mathbf{x},y)}[\ell(\mathbf{x}, y; D_k)] - \mathbb{E}_{(\mathbf{x},y)}[\ell(\mathbf{x}, y; d)]$ . Using the auxiliary head $g_k$ with probabilities $P(\hat{y} \mid \mathbf{x}; g_k)$, we instantiate standard uncertainty-based criteria: *least confidence* $f_{\text{conf}}(\mathbf{x}; g_k) = 1 - \max_{\hat{y}} P(\hat{y} \mid \mathbf{x}; g_k)$, and *entropy* $f_{\text{ent}}(\mathbf{x}; g_k) = -\sum_{\hat{y}} P(\hat{y} \mid \mathbf{x}; g_k) \log P(\hat{y} \mid \mathbf{x}; g_k)$, selecting the top-$m$ by score (random sampling is a baseline). To curb outliers, we optionally drop the top 5% highest-uncertainty items before selection. The approach is *criterion-agnostic*: any client-computable loss/gradient surrogate can plug in. Let $s_k$ be the selected activations for client $k$. The server optimizes; $\min_{\mathbf{w}} H_k(\mathbf{w}) = \frac{1}{|s_k|} \sum_{x \in s_k} \ell(x; \mathbf{w})$, updates $\mathbf{w}$ via minibatch SGD, and returns gradients through the cut to update $c_k$. In parallel, the auxiliary head $g_k$ is updated locally by $\min_{\mathbf{g}_k} F_k(\mathbf{g}_k) = \frac{1}{|d_k|} \sum_{(\mathbf{x},y) \in d_k} \ell(\mathbf{x}, y; \mathbf{g}_k)$. Client models are synchronized across clients as in SL.

We validate our algorithm on CIFAR-10 dataset. By Communicating only 50% of the cut-layer activations per client per round, our activation-level subset selection improves test accuracy over vanilla SplitNN on both IID and non-IID data distributions. On IID data, our method attains 82.10% vs. 81.85% for SplitNN. On non-IID data, it reaches 81.84% vs. 80.82%. These gains demonstrate that we can halve the volume of intermediate activations transmitted to the server while maintaining or improving generalization.

**Conclusion:** We propose a novel framework for subset-selection in SplitNN that uses the partial model at the client-side to rank the informative samples in the local dataset. An auxiliary network appended to the client-side model generates pseudo-predictions locally which are utilized in computing a subset-selection measure. Extensive experimentation and empirical results corroborates that the proposed framework efficiently selects the subset of data and reduces the computation and communication burden between the server and the clients.

# References


[1] P Vepakomma, O Gupta, T Swedish, and R Raskar. Split learning for health: Distributed deep learning without sharing raw patient data. *NIPS*, 2018.

[2] B McMahan, E Moore, D Ramage, S Hampson, and Blaise A y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[3] J Konečnỳ, HB McMahan, FX Yu, P Richtárik, AT Suresh, and Dave B. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[4] Xing Chen, Jingtao Li, and Chaitali Chakrabarti. Communication and computation reduction for split learning using asynchronous training, 2021.

[5] O Sener and S Savarese. Active learning for convolutional neural networks: A core-set approach. *ICLR*, 2018.

[6] David D. Lewis and J Catlett. Heterogeneous uncertainty sampling for supervised learning. In William W. Cohen and Haym Hirsh, editors, *Machine Learning Proceedings 1994*, pages 148–156. Morgan Kaufmann, San Francisco (CA), 1994. ISBN 978-1-55860-335-6.

[7] C Coleman, C Yeh, S Mussmann, B Mirzasoleiman, P Bailis, P Liang, J Leskovec, and M Zaharia. Selection via proxy: Efficient data selection for deep learning. *ICLR*, 2020.

[8] C Thapa, M A P Chamikara, and S Camtepe. Splitfed: When federated learning meets split learning. *arXiv preprint:2004.12088*, 2020.

[9] DJ Han, HI Bhatti, J Lee, and J Moon. Accelerating federated learning with split learning on locally generated losses. In *Workshop on FL*. ICML, 2021.