ILASR: Privacy-Preserving Incremental Learning for Automatic Speech Recognition at Production Scale

Gopinath Chennupati
Milind Rao
Gurpreet Chadha
Aaron Eakin
Amazon Alexa
USA

Anirudh Raju
Gautam Tiwari
Anit Kumar Sahu
Ariya Rastrow
Jasha Droppo
Amazon Alexa
USA

Andy Oberlin
Buddha Nandanoor
Prahalad Venkataramanan
Zheng Wu
Pankaj Sitpure
Amazon Alexa
USA

ABSTRACT

Incremental learning is one paradigm to enable model building and updating at scale with streaming data. For end-to-end automatic speech recognition (ASR) tasks, the absence of human annotated labels along with the need for privacy preserving policies for model building makes it a daunting challenge. Motivated by these challenges, in this paper we use a cloud based framework for production systems to demonstrate insights from privacy preserving incremental learning for automatic speech recognition (ILASR). By privacy preserving, we mean, usage of ephemeral data which are not human annotated. This system is a step forward for production level ASR models for incremental/continual learning that offers near real-time test-bed for experimentation in the cloud for end-to-end ASR, while adhering to privacy-preserving policies. We show that the proposed system can improve the production models significantly (3%) over a new time period of six months even in the absence of human annotated labels with varying levels of weak supervision and large batch sizes in incremental learning. This improvement is 20% over test sets with new words and phrases in the new time period. We demonstrate the effectiveness of model building in a privacy-preserving incremental fashion for ASR while further exploring the utility of having an effective teacher model and use of large batch sizes.

CCS CONCEPTS

• **Computing methodologies** → **Speech recognition**; **Neural networks**; *Semi-supervised learning settings*; • **Security and privacy** → **Privacy-preserving protocols**.

KEYWORDS

Incremental Learning, Automatic Speech Recognition, Privacy-preserving Machine Learning

KDD '22, August 14–18, 2022, Washington, DC, USA

ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539174

ACM Reference Format:
Gopinath Chennupati, Milind Rao, Gurpreet Chadha, Aaron Eakin, Anirudh Raju, Gautam Tiwari, Anit Kumar Sahu, Ariya Rastrow, Jasha Droppo, and Andy Oberlin, Buddha Nandanoor, Prahalad Venkataramanan, Zheng Wu, Pankaj Sitpure. 2022. ILASR: Privacy-Preserving Incremental Learning for Automatic Speech Recognition at Production Scale. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3534678.3539174

1 INTRODUCTION

Privacy preserving machine learning [1] has been at forefront, due to both increased interest in privacy and the potential susceptibility of deep neural networks to leaks and attacks. Federated Learning (FL) [44] is a machine learning technique that involves training models on edge devices, where data need not leave the device, and can be heterogeneous and non-identically and independently distributed (non-IID). In FL, multiple model updates from a number of participating devices are aggregated. In spite of raw data not leaving the edge device, FL has found to be susceptible to gradient inversion attacks [65, 66]. In response, various privacy-preserving mechanisms such as differential privacy and secure aggregation [19, 58] have been proposed to counter data leakage and conform to privacy preserving mechanisms. Moreover, the lack of labels for the data present in the participating entities, makes FL more challenging for applications such as automatic speech recognition (ASR). Most research in FL until now focuses on training models from scratch. In this work, we focus on privacy-preserving incremental learning (IL), in the context of end-to-end production model building at scale over extended time periods. Incremental learning [8, 62] has been extensively used to incrementally update models on the fly instead of training them from scratch. Incremental learning as such is not privacy-preserving.

Despite the above advances, to the best of our knowledge, few frameworks exist for privacy-preserving incremental training of end-to-end automatic speech recognition models. Prior work on federated learning for speech-based tasks [13, 16, 23] and end-to-end ASR [18, 26], focus on standard benchmarks[1] and not on large scale production data. Privacy-preserving IL on device for end-to-end ASR poses a number of challenges. Production-sized end-to-end ASR systems [11, 24] are expensive to train even in traditional

[1] e.g. LibriSpeech [47] is a small sized dataset (~ 1000 hours) recorded in a controlled environment

distributed setup, on-device training needs more work [7] to accommodate restrictive memory and computational constraints. Generating training labels i.e. speech transcripts, in near real-time, on the devices is another challenge. To alleviate unavailability of near real-time speech transcripts, teacher transcripts can be used in a semi-supervised and/or self-learning fashion. For example, consider the problem of improving models deployed in edge devices that run voice assistants. In such cases, the number of devices is in the millions, which results in a large scale of streaming data being generated. We propose to use large batch processing for the utterances being collected at the edge devices and sent to the cloud for processing, and the data is only stored ephemerally. However, deploying all or part of the above components on resource constrained speech devices (such as Alexa, Google Assistant and others) is challenging.

We build and use a cloud-based system, Incremental Learning for Automatic Speech Recognition (ILASR) to train and update production ready ASR models. ILASR automates the entire pipeline of incremental learning in a privacy-preserving manner. To enforce privacy-preserving aspects in the context of ASR, we enforce labelling of the utterances through pre-trained teacher models with no human annotations. ILASR processes an utterance once before updating the model, preserving the chronological order of data. To that end, the contributions of the paper are:

- A novel cloud-based IL system to train production ready ASR models in near real-time, with a large amount of streaming de-identified data, without having to manually transcribe or persist the audio.
- We provide new insights in terms of usage of large batch processing in ILASR that it does not have detrimental impact on test accuracy as compared to the contradicting findings [20, 33, 37, 41, 42, 52] (on CNN ImageNet). We could accommodate fixed learning rates and minimal hyper-parameter optimization [34] along with large batch training. With a monthly frequency of incremental model updates, we observe that the production models (converged on old data) improve in near real-time on new data belonging to a period of six months
- We empirically establish over six months of data that chronological vs randomized order of processing utterances does not produce any observable difference in performance.

We evaluate ILASR on three student recurrent neural network transducer (RNN-T) [24] architectures. The semi-supervised learning (SSL) approach produces machine transcripts using a larger teacher ASR model. The students are pre-trained on in-house de-identified data until 2020. Through training in ILASR, we observe an improvement of $3-7\%$ in word error rate (WER) over the pre-trained baselines when these students are trained incrementally on a new time period of six months in 2021. The improvement in WER is termed relative word error rate reduction (WERR). This increases to 20% on test sets with new words and phrases in 2021. Similarly, when the student models are trained incrementally each month, we observe WER improvements, as well the phenomenon where models get stale without further updates.

The paper is organized as follows: section 2 describes the essential concepts used in the paper; section 3 explains the proposed system; section 4 describes the experimental settings; section 5 presents the results; section 6 summarizes the related literature and finally, section 7 concludes and recommends future directions.

2 BACKGROUND

In this section we summarize the RNN-T architecture and large batch training with stochastic gradient descent (SGD).

2.1 RNN-T model architecture

Figure 1 shows the RNN-T [24] architecture used in real-time speech recognition. The model predicts the probability $P(\mathbf{y}|\mathbf{x})$ of labels $\mathbf{y} = (y_1, ..., y_U)$ given acoustic features $\mathbf{x} = (x_1, ..., x_T)$. It has an encoder, a prediction network, and a joint network. The encoder is analogous to an acoustic model that takes a sequence of acoustic input features and outputs encoded hidden representations. The prediction network corresponds to a language model that accepts the previous output label predictions, and maps them to hidden representations. The joint network is a feed forward DNN that takes both the encoder and prediction network outputs, and predicts the final output label probabilities with softmax normalization.

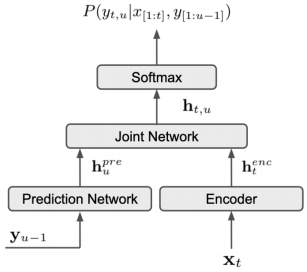


Figure 1: RNN-T ASR model architecture

2.2 Overview of learning with large batch size

When training with SGD, mini batches with a well crafted decaying learning rate schedule are commonly used as opposed to using large batches. Previous work in [33] has demonstrated a generalization drop when using large batches, thus recommending mini-batch SGD with decaying learning rate. However, recent advances in large batch training both with a linear scaling rule of the learning rate [22] and constant learning rate[53], large batch training has been shown to achieve similar performance as its mini-batch counterpart. A recurrent observation in the literature [20, 33, 37, 41, 42, 52] is that large batch training (for ImageNet, > 1000) results in test accuracy degradation. Despite the warm-up in [22], for ImageNet, the best accuracies are observed up to a large mini-batch of 8192 images.

In this paper, we deal with the challenges of 1) training with large batches in incremental learning and 2) semi-supervised learning to alleviate unavailability of human annotation and labels. For automatic speech recognition (ASR), with large batch sizes ($> 3e5$ utterances) using a fixed learning rate schedule, we observe better test accuracies, as opposed to the degradation in literature, while training with teacher transcripts for the incremental audio data.

3 ILASR: INCREMENTAL LEARNING FOR AUTOMATIC SPEECH RECOGNITION

This section describes the ILASR architecture and the corresponding incremental learning algorithm. ILASR offers large scale end-to-end ASR training with the ability to incrementally update the models in user-defined time windows. ILASR automates the whole life-cycle of data generation, sampling, labeling, model development, evaluation and deployment for audio data in near real-time.

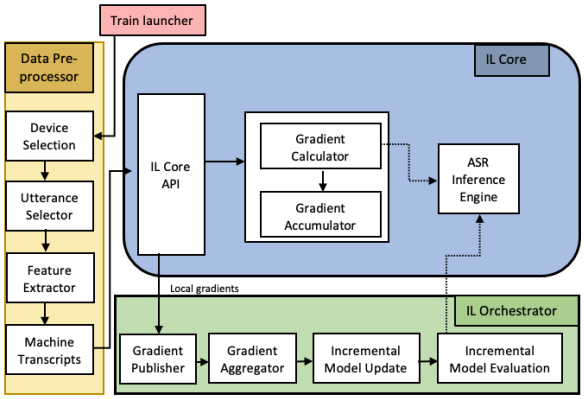


Figure 2: High-level skeleton of the ILASR architecture

3.1 ILASR Architecture

Figure 2 shows the architectural overview of ILASR system. The system comprises three primary components: (1) *Data preprocessor* – is a cloud runtime service that processes near real-time audio from device; (2) *IL Core* is responsible for model training, computing model updates and inference; and (3) *IL Orchestrator* aggregates the accumulated gradients, updates the model, performs evaluation and finalizes the model update based on the evaluation result.

Train launcher initiates the end-to-end ASR training in ILASR. The first step is *data preprocessing* to select a subset of devices and utterances to participate in the training loop. The selection could be random or based on heuristics aimed at improving the model in a particular way. Confidence scores obtained during inference are used [29, 30] coupled with heuristics such as presence of rare words or semantic tags and intents of interest. This selection can be extended to leverage weak signals from user feedback such as user indicating whether the action taken by the assistant is positive or negative or detecting friction such as repeated requests or cancellations. Acoustic features are extracted and augmented [48] for the selected utterances for training. *Machine transcripts* are generated using a teacher ASR model pre-trained using standard distributed training. The Conformer [25] based end-to-end ASR teacher model decodes the input audio (X) to produce machine transcripts (Y). These paired (X, Y) instances are used to train the model. The machine transcripts act as ground truth labels. ILASR produce transcriptions through secure automation without human intervention or review. The extracted features together with machine transcripts in this step are combined to train the student models using *IL Core*. The *IL Core* system has an application programming interface (API) that supports local gradient accumulation on each of the servers in the fleet, and an ASR inference engine. The IL Core API supports FedSGD and FedAvg [44] and can be extended to support other federated optimizers such as FedProx [51], FedMA [59], FedNova [60], and adaptive federated optimizer [50]. The *IL Orchestrator* coordinates training across the ILASR fleet. IL Orchestrator contains the gradient publisher, aggregator and updates the model incrementally. The gradient aggregator collects gradients from each of the IL Core instances, aggregates them and then applies them to the current model. Once the model update is done, the collected gradients are discarded and not stored in the system which helps with reducing the risk of gradient inversion attack. A periodic light-weight evaluation of the model ensures that the model is directionally improving. The global model is updated in a given round when the performance improves over that of the previous round. To reduce the probability of a model update resulting in worse performance, ILASR can be run in parallel with differing hyperparameters. In this scenario, one of the resulting models can be utilized should it result in improved performance. After a sufficient number of rounds, the final model is stored for the next model release after a detailed model validation step.

ILASR addresses security and privacy concerns with different levels of granularity. Since ILASR is a cloud-based system for privacy-preserving IL at scale, the audio encryption is two fold. In the first stage, TLS [15] encryption is applied on audio transmission followed by an application level key-master [40] encryption. Importantly, the audio is purged in a few minutes (≤ 10), within which the model updates are calculated.

3.2 ILASR: Incremental Learning

Algorithm 1 ILASR incremental learning algorithm

Require: $\mathcal{K}$ servers, $\mathcal{L}$ loss function, N number of local steps per round, $\mathcal{B}$ local batch size, (η) learning rate, P_k^r recent utterances pulled by server k in round r, $\mathcal{D}_{eval}$ eval set and $\mathcal{D}_{ht}$ past transcribed data if used for rehearsal training.

Ensure: $w_{\mathcal{G}}^r$ incrementally updated global model and wer_r word error rate on the eval set after r rounds

Init. $w_{\mathcal{G}}^0$// start training with a pre-trained model
$wer_0 = asr\_inference\_engine(\mathcal{D}_{eval}, w_{\mathcal{G}}^{\mathcal{T}})$
for each round $r = 1, 2, \ldots$ **do**
  for each server $k \in$ ILASR Fleet **in parallel do**
    $w_k^r = w_{\mathcal{G}}^{r-1}$
    $\mathcal{D}_{ssl} \leftarrow$ (filter P_k^r based on utterance selection criteria and generate machine transcript, refer algorithm 2)
    $\mathcal{D}_{train} \leftarrow$ (mix $\mathcal{D}_{ssl}$ and $\mathcal{D}_{ht}$ if $\mathcal{D}_{ht}$ is used for rehearsal, else just $\mathcal{D}_{ssl}$)
    $\mathcal{D}_{train} \leftarrow$ (split $\mathcal{D}_{train}$ into N batches of size $\mathcal{B}$)
    for each batch b_i from b_1 to b_N **do**
      $w_k^r \leftarrow \text{optimizer}_k.\text{update}(\eta, \nabla\mathcal{L}(w; b_i))$
    end for
  end for
  $w_{\mathcal{G}}^r \leftarrow \frac{1}{\mathcal{K}} \sum_{k=1}^{\mathcal{K}} w_k^r$
  $wer_r \leftarrow asr\_inference\_engine(\mathcal{D}_{eval}, w_{\mathcal{G}}^r)$
  $w_{\mathcal{G}}^r = w_{\mathcal{G}}^{r-1}$ if $wer_r > wer_{r-1}$ // Revert to the previous model if not a better model.
end for

Algorithm 1 shows the incremental learning policy in ILASR framework. The new model obtained in each round is used only if it performs better than the model from the previous round. Parallel runs of the algorithm with differing hyper parameters to train an ensemble of incrementally updated models can ensure that there is at least one model that performs better than the model from the previous round. Another interesting consideration is the effect of catastrophic forgetfulness [17, 21, 43] in incremental learning of ILASR framework, where the previous learned behaviour of a model is forgotten with new updates. This can be mitigated with the rehearsal [2] of training on a subset of annotated historical data along with the new data.

We describe the SSL data generation method in algorithm 2. We randomly sample a subset of the audio in near real-time, to prepare a data pool ($\mathcal{P}$), and calculate target number of utterances ($\mathcal{U}$) to be sampled from $\mathcal{P}$, where each of the utterances include a pre-calculated confidence value [57]. For each confidence bin, for example confidence in (600,700] where confidence is evaluated on a scale from 0 to 1000, utterances are filtered to conform to the confidence criterion. The randomly sampled utterances from above are set to get target number of utterances and sent to IL core for training, which are deleted as soon as the model takes a pass over it for the first time. Additional criteria such as presence of rare words, presence of desired semantic tags can also be utilized.

Algorithm 2 SSL data selection procedure

Require: τ list of utterance confidence bins
Ensure: $\mathcal{X}$ data set
1: $\mathcal{P} = random\_sample()$ // prepare a random pool of data
2: $\mathcal{P} = teacher\_decode(\mathcal{P})$ // generate machine transcripts
3: $\mathcal{U} = calc\_utterance\_confidence(\mathcal{P})$.
4: $\mathcal{Q} = [\ ]$ // a bin for each confidence range
5: **for** $c \in \tau$ **do**
6: $\quad \mathcal{Q}[c] = filter\_utterances(\mathcal{U}, c)$
7: $\quad \mathcal{X} = select\_utterances(\mathcal{Q})$ // can include additional criteria like presence of rare words or desired semantic tags
8: **end for**

4 EXPERIMENTS

We describe the datasets, model configurations and experimental settings used in this paper, to provide insights and study privacy-preserving incremental learning through ILASR.

4.1 Datasets

All speech data used for training and evaluation are de-identified.
Train sets The audio streams are prepared into offline training datasets. The following training datasets are used for experimentation:
Pre-training datasets: A 480k-hour pre-training dataset is utilized for building pre-training models. This pre-trained model is used as a starting point for incremental training with the ILASR system. This comprises two datasets:

(1) *120K-hour HT*: Human-transcribed (HT) data from 2020 and previous years
(2) *360K-hour SSL*: Machine-transcribed data in 2020

Incremental training dataset: We consider the end of 2020 as the start date for incremental training of ASR models.

(1) *180K-hour ILASR SSL*: Machine-transcribed data is generated over a period of six months in 2021 (Jan to June) and is used for near real-time training of the ILASR system.

Test sets: We evaluate the models on in-house human transcribed (HT) test sets.
General: Includes three HT datasets from different time ranges representing the general use case. It comprises a 37-hour test set from 2021, a 10-hour test set from 2020 and a 96-hour test set from $2018 - 2019$.
Rare: Includes three HT datasets from different time ranges, where the transcriptions contain at least one rare word. Rare words are those in the long-tail of the vocabulary determined by word frequency. This includes a 44-hour test set from 2021, a 44-hour test set from 2020, and a 27-hour test set from $2018 - 2019$.
Delta: This consists of a 22-hour HT test set that records a change in frequency of words in 2021 over 2020. The transcriptions are filtered based on 1-gram, 2-gram and 3-grams that are 5x more frequent in 2021 than 2020. This test set captures changes in the data distribution and is very relevant to measure the impact of incremental learning with ILASR.
Messaging: Includes two HT datasets that comprise of messaging and communications domain data. It includes a 2.7-hour HT test from 2020 and a 45.5-hour HT test set from $2018 - 2019$.
Monthly datasets (2021): We use six monthly test sets from Jan to June 2021 to evaluate the incremental learning setup of ILASR. Each of these datasets are refered to as (Jan, Feb,$\cdots$,June) and each month has on average 70-hours of data. We further report results on 3-month datasets $Jan - Mar$ including data from Jan, Feb, Mar and $Apr - Jun$ including data from Apr, May, June.

4.2 Model details

Features: The audio features are 64 dimensional log-mel filter-bank energies [46] computed over a 25ms window, with a 10ms shift. The features computed on 3 consecutive 10ms frames are stacked and sub-sampled to result in 192 dimensional features at a 30ms frame rate, and are provided as input to the ASR model. The ground truth transcripts are tokenized to 2500 sub-word units using a uni-gram language model [35].
Models: *Teacher models*: Teacher models are used to generate SSL machine transcripts. We have three teacher models available: $T3$ is a teacher model (a conventional RNN-HMM hybrid ASR system[6]) that is trained on 100K-hours of data until 2019 only. The machine-transcripts from $T3$ are utilized to bootstrap and provide transcripts for the more recent 360K-hour SSL pre-training dataset. The 480K-hour pre-training dataset, including the 360K-hour SSL dataset based on $T3$ and the 120K-hour HT dataset, is utilized to train two updated teacher models: (1) $T1$: A larger conformer based ASR architecture [25] trained on 480K-hours. $T1$ has 122M parameters, an encoder with 17×512 LSTM layers, 8 attention heads with 32 dimensional convolution kernel. The prediction network uses 2×1024 LSTM layers. (2) $T2$ is a conventional RNN-HMM hybrid ASR system [6] and is trained on the same 480K-hour dataset. Finally, the student models for all experiments in the paper are trained on SSL datasets that use the most recent $T1$ teacher model.

In section 5.1.3, for the purpose of ablations comparing various teachers, we train student models on SSL datasets that are based on $T2$ and $T3$.

Student models: The student models are based on different LSTM based RNN-T architectures. These vary in the number of encoder layers and the feature frame rates. Two student models are described as follows. *rnnt_60m* contains 60M parameters with 5×1024 LSTM encoder, 2×1024 LSTM prediction network and a feed-forward joint network with *tanh* activation. The input embeddings of the prediction network are 512 dimensional. SpecAugment [48] is used on the audio features. *rnnt_90m* contains 90M parameters with 8×1024 LSTM layer encoder, a prediction network of size 2×1024, and a feed-forward joint network with *tanh* activation. The input embeddings of the prediction network use 512 dimensional embeddings and a 2500 sub-word tokenizer from a uni-gram language model. SpecAugment is used on the audio features. The encoder uses an LSTM based time-reduced [54] RNN multi-layer (for speed of training and inference) with feature frame rate set to 3 layers. Each of these feature frame layers have 1536 units and the LSTM projection with a size of 512.

The models *rnnt_90m* and *rnnt_60m* are pre-trained on both the HT data of $120K$ hours and $340K$ hours of SSL data generated using the teacher ($T1$) decoded labels. The human transcribed data used in the pre-training utilizes data up to the end of 2020, while the SSL data is in 2020. For our experiments in this paper, we further train the above pre-trained RNN-T student models using a total amount of $180k$ hours of SSL data (teacher generated labels) available in a time-window of 6 months in 2021.

Training details: We use the following parameters to train both the teacher and student models. The system is run on a fleet consisting of 200 nodes. We adopt a learning rate schedule of warm-up where $lr = 1e-7$ for the first 3000 steps, followed by constant learning rate of $5e-4$ till $50k$ steps, then exponential decay ($lr = 1e-5$) from $50k$ to $750k$ steps with Adam optimizer (hyperparameters are $\beta_1 = 0.9$, $\beta_2 = 0.99$).

We experiment with multiple large batch sizes ($9k$, $18k$, $73k$, $147k$, $215k$, $307k$) through gradient accumulations. Note that these accumulations have an implicit effect of changing the gradient values due to the summation of gradients across a large batch. We process large batches without altering the lr schedule while accumulating the gradients. The performance of these models is measured in terms of relative word error rate reduction (WERR) over the corresponding baselines. WER is the ratio of edit distance to sequence length, where edit distance is the length of the shortest sequence of insert, delete and substitution operation on transforming a predicted sequence to target.

5 RESULTS & DISCUSSION

In this section, we analyze the performance of incremental learning in ILASR. In particular, we analyze the performance of incremental learning in ILASR in terms of relative word error rate reduction (WERR) in comparison with the initial pre-trained student models as baselines.

From Table 1, we see that ILASR improves a strongly trained base model by up to 3% on test sets in 2021 which climbs to 20% on the delta dataset that consists of new or trending words and phrases. At the same time, performance on older general and tail test sets do not see much degradation.

Table 1: Relative % WER improvements from the initial model when trained with the ILASR system

Time	Test-set	ILASR	
		replay	no replay
2021	Rare	0.72%	0.66%
	Delta	20.10%	23.99%
	General	1.23%	0.41%
	Jan-Mar	1.25%	1.50%
	Apr-Jun	2.73%	3.09%
2020	Rare	0.62%	0.62%
	General	0.00%	-0.72%
	Message	-0.83%	-2.04%
2018-2019	Rare	-0.63%	-0.63%
	General	-1.21%	-2.6%
	Message	-2.82%	-3.42%

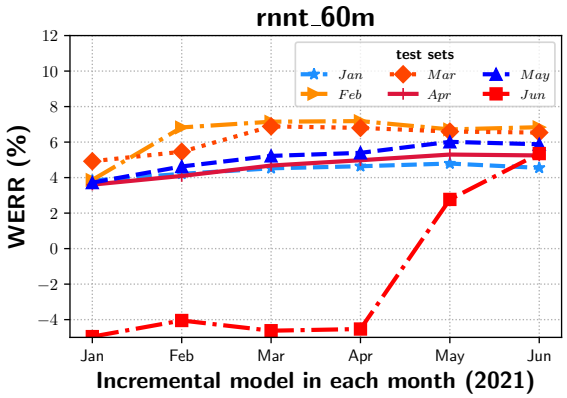
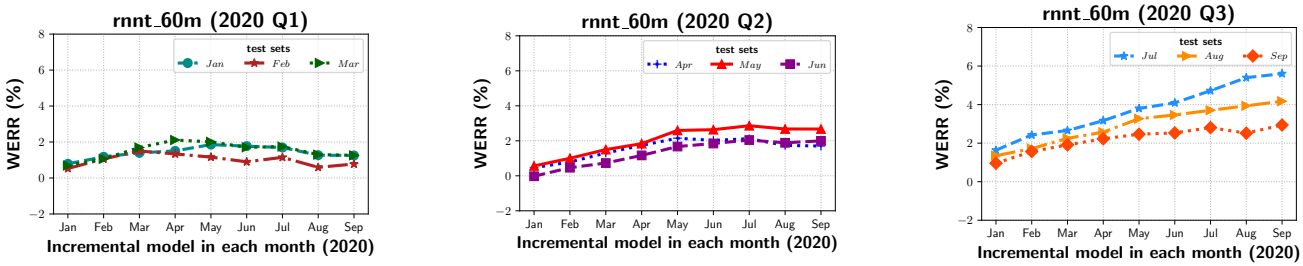


Figure 3: Monthly WERR (%) for incremental learning in ILASR for *rnnt_60m* on six of the monthly test sets (*Jan–Jun*) when measured relative to the starting model versus the one trained incrementally in each month.

Catastrophic forgetting is one of the issues incremental learning needs to circumvent in order to have consistent performance across both old and new data. In Table 1, we compare the performance of *replay* based incremental learning, where a sub-sampled portion of 120K-hour human-transcribed data is also consumed in model training while the *no replay* counterpart does not involve that. As demonstrated in Table 1, *replay* based training tends to outperform its *no replay* counterpart on older test sets as expected from IL literature.

Next, we evaluate the incrementally trained ILASR models on fine-grained test sets that are prepared in each of the six months (Jan-Jun) of 2021, see Figure 3. For all the evaluations in Figure 3, we report the WERR in each month relative to the initial pre-trained model (for example, WERR in *May* is the relative difference between the WERs of *May* model and the pre-trained). The results show incremental improvements in performance on all the six monthly test sets from month to month in the ILASR training. This suggests



(a) 2020$Q1$ **(b)** 2020$Q2$ **(c)** 2020$Q3$

Figure 4: For the *rnnt_60m*, the pre-trained model is trained on the data available until 12/2019. Training this pre-trained model in incremental mode for the next nine months (*Jan – Sep*) in 2020. The x-axis shows the monthly incremental model, where the model from previous month is fine-tuned on the data in current month; y-axis shows the relative WER in each month w.r.t the initial pre-trained model. Each of the curves represent the test set of the corresponding month.

that the incremental training helps in capturing the new trends in time periods while the model is adapting to the incremental changes in the data. It is also noteworthy that the incremental improvement does not come at the cost of catastrophic forgetting. More interestingly, the models trained with the data until *May*/*June* degrade the performance on *June* test-set, which improves after the model is trained on the data available from *May*/*June*. This clearly suggests the adaptive nature of capturing the shifts in data in the new time periods in ILASR.

To further strengthen the incremental learning claims, we analyze the incremental learning patterns for a longer duration in the time-periods between *Jan – Sep* in 2020. Figure 4 shows the learning patterns on a quarterly basis for the first three quarters (Q1–Q3) of 2020. In 2020$Q1$ and $Q2$, the WERR improves initially and then decreases as the incremental model training progresses on a month-over-month basis. The degradation (whilst better than the baseline) is a demonstration of forgetting as newer updates are prioritized over the months old test sets. Consequently, in 2020$Q3$, the performance improves without any downward trends, which is due to the fact that the models keep learning month over month while the test sets also belong to the same time-periods. These trends suggest that the proposed techniques help in incrementally improving the performance even in longer time-periods while limiting the regressions on the older eval data.

Next, we explore several design choices which play a key role in the performance of ILASR and share our insights in terms of the design choices.

5.1 Design Choices: ILASR

We explore the following design choices in the context of the ILASR framework: 1) effect of large batch sizes on performance of the student models; 2) temporal effects on processing the data in ILASR; 3) analyze the importance of different teacher models in ILASR.

5.1.1 Training is robust to large batch sizes. We use large batches in ILASR via gradient accumulations. As the effective batch size increases, the number of optimization or update steps reduces as the same amount of data is processed. Larger batch sizes would require fewer optimization steps and vice versa for the same amount of data. Use of large batches accelerates the training (shown in [64]),

Table 2: Effect of large batches on the relative improvement in performance (in terms of WERR, %) of all the three models when fine-tuned in ILASR.

Time	Test-set	effective batch size					
		9k	18k	73k	147k	215k	307k
		rnnt_60m					
2021	Jan–Mar	2.74%	1.49%	2.37%	2.37%	2.24%	2.24%
2018–2019	Rare	3.58%	3.58%	3.72%	3.65%	3.78%	3.72%
	Message	6.46%	6.05%	9.46%	9.46%	9.28%	9.37%
	General	15.14%	15.12%	14.70%	14.56%	14.41%	14.26%

which is similar in ILASR. The reason large batch sizes is relevant in the ILASR system is that there are limitations about how quickly gradients can be aggregated and the global model distributed to the servers in the fleet. Hence, a limited number of update steps can take place in a time period compared to GPU-based offline distributed training. Moreover, as data arrives in a streaming fashion and is not persisted, it needs to be consumed as and when it arrives, in near real-time. For each of the limited number of updates, a large amount of streaming data is available.

We explore the trade-off between large batches and model performance. Table 2 shows the effect of large batches on performance of a student models trained in ILASR. The performance (WERR) is relative to the corresponding pre-trained student model. This baseline is weaker, hence improvements are larger. We find that increasing the batch multiplier (effective batch size) has insignificant effect on WER. As batch sizes increase from 9K to 300K utterances, the difference in the accuracies is insignificant.

More importantly, this finding is in contrast to the test accuracy degradation effects reported in literature [20, 22, 33, 37, 41, 42, 52, 56] with the use of large batches. We observe that such degradation is not evident for model training in ILASR. Although, the attempts in the literature have no strong mathematical justification, Goyal et al. [22] reasoned the performance degradation to optimization issues, thereby using warm-up to mitigate the degradation. Similarly, in our case, we attribute the gains and/or no performance degradation to the following factor. The initialized models are pre-trained that have converged on the data from a previous time period as opposed to random initialization in the large batch training in

Table 3: Impact of the temporal order (chronological versus random) of processing the training data in ILASR for both with and without replay of the human transcriptions.

Time	Test-set	Chrono vs. random	
		replay	no replay
2021	Rare	-0.62%	-1.16%
	Delta	-1.68%	-0.73%
	General	0.15%	1.47%
	Jan-Mar	-0.53%	-0.24%
	Apr-Jun	-0.47%	0.29%
2020	Rare	-0.56%	-0.90%
	General	-0.55%	1.61%
	Message	0.35%	0.48%
2019-2019	Rare	-0.46%	-1.26%
	General	0.32%	0.67%
	Message	-0.11%	-0.87%

literature, usually, these models are trained from scratch (despite the few initial epochs in warm-up) in the literature.

5.1.2 Impact of chronologically ordered data. One important aspect of IL is the data being processed in time as is available, chronologically. We analyze the effect of processing order (chronological vs random) for the six months in 2021. Note, random order is same as shuffling the data in regular distributed training of deep models. Chronological data is not IID across time as utterances have a correlation with the time of day (for example, requests to snooze alarms in the morning or turning smart lights on after sundown). We found that there is no difference in performance of processing the data chronologically as compared to randomly as depicted in Table 3. Moreover, in both the cases of chronological and randomized, the improvements over initial baselines are clearly evident (see Table 1).

Table 4: Performance (in terms of WERR, %) of the RNN-HMM hybrid ASR teacher ($T2$) and bidirectional RNN-HMM hybrid ASR ($T3$) based teacher models with respect to the Conformer teacher ($T1$). The negative (-) sign represents that $T1$ performs worse while the rest shows that $T1$ is the best performing teacher model.

Time	Test-set	$T1$ vs $T3$	$T1$ vs $T2$
2021	Jan–Mar	16.63%	0.14%
2018–2019	Rare	8.75%	12.02%
	Message	7.34%	14.92%
	General	−0.89%	20.51%

5.1.3 Ablations with teachers and students. We experiment with three different teacher models that are trained for different time ranges with different architectures. This experiment helps us explore the importance of keeping an updated and more effective teacher. The three teachers are: $T1$ is the Conformer based that is explained earlier in section 4.2; $T2$ is a RNN-HMM conventional hybrid model [6]; $T3$ is a bidirectional RNN-HMM conventional hybrid ASR model. $T1$ and $T2$ are trained on the same amount of data until the end of 2020 while $T3$ is trained on the data (a total of $\sim 100k$ hours of HT data) available till the end of 2019.

Table 4 compares the performance of the teacher models. On an average, $T1$ is better than the rest of the two teachers, $T1 > T2 > T3$ on new data reflecting the importance of keeping the teacher model up-to-date. Conformer based teacher, $T1$ is better than the rest of the remaining two teachers. The relative performance differences, when measured on the four standard test sets are, $T1$ is better than $T2$ and $T3$ with 11.85% and 7.96% WERR, respectively.

Table 5: The performance (in terms of WERR) of the student models when trained with the machine transcripts generated from each of the three different teacher models.

Time	Test-set	*rnnt_90m*			*rnnt_60m*		
		$T1$	$T2$	$T3$	$T1$	$T2$	$T3$
2021	Jan–Mar	10.5%	8.54%	7.41%	7.78%	6.27%	2.87%
2018–2019	Rare	5.48%	5.71%	5.60%	4.21%	3.89%	3.58%
	Message	4.12%	4.88%	7.12%	3.07%	3.74%	6.05%
	General	8.25%	7.88%	7.30%	5.07%	5.03%	6.55%

Table 5 shows the WERR of two student models (*rnnt_90m* and *rnnt_60m*) when trained using the machine transcripts generated from the three teacher models. We observe that both the students are similar in terms of performance. On an average, for *rnnt_90m*, $T1$ based training is better than $T2$ and $T3$, with 4.66% and 3.25% WERR, respectively. For *rnnt_60m*, $T1$ is better than $T2$ and $T3$ with 5.96 and 2.18% relative WERR improvement respectively. The improvements are larger than in Table 1 as these experiments were done with 3 months of data using a weaker baseline. In fact, both the students have same order of performance as the teachers, that is $T1 > T2 > T3$ even after training in IL on new data. More important, the magnitude of improvement in student (true for both the student models) training is not of same scale as the difference in teachers. For example, Conformer based teacher ($T1$) is better than $T3$ by 7.96%, whereas *rnnt_90m* student trained with Conformer transcripts ($T1$) is 3.25% better than the one trained with $T3$ transcripts. This suggests that better teacher models result in improving the student performance but the difference (same student trained with different teacher models) is narrower. In other words, a significantly better teacher model can have a limited impact in improving students models in ILASR.

6 RELATED WORK

SGD gradients mini and large: Stochastic gradient descent (SGD) drives the training of neural nets with mini batches. Large mini batches [22, 27, 53, 64] reduce the number of updates with a large step size. Simply increasing the batch size reduces the test accuracy [33] as the gradients get integrated. Test set accuracy can be improved with large batches that are proportional to the learning rate. This simple linear scaling is inefficient, which necessitates a warm-up phase [22]. Instead of decaying the learning rate, increasing the batch size during training [53] helps to reduce the communication steps to update the model and improves the test accuracy. Federated averaging [44] (FedAvg) follows a similar strategy of

synchronously updating the gradients. Thus, the centralized model simply aggregates the updates from various clients. Therefore, we apply these large synchronous batch updates (as in [53]) to the model in federated settings (similar designs were proposed in [5]) both in federated SGD and averaging algorithms. Considering the negative effects of large-batch training on test accuracy in literature [9, 20, 33, 37, 41, 42, 52], a post-local SGD was proposed [38], inspired from the FedAvg, where they adopt a warm-up [22] based mini-batch SGD training for initial training before launching FedAvg. Similarly, distributed SGD for speech [55] and large scale training with million hours of speech [49] have helped accelerate the production for ASR models.

Semi-supervised Learning in ASR: The semi-supervised learning described in [31, 32] employed auto-encoders to extract speech and text features from unpaired text and speech data. Semi-supervised ASR with filter-bank features [39] use deep contextualized acoustic representations with small amounts of labeled data. The weak distillation of audio-text pairs resulted from the unsupervised techniques in [36] helped in improving the end to end ASR. The semi-supervised approaches in [61] combined the data augmentation through spectral augment [48] and consistency regularization to improve the performance. Dropout offer the power of ensembles, the semi-supervised dropout attempts in [14] improved the pseudo label accuracy and model performance in ASR. Recently, the work in [63] employed contrastive semi-supervised learning with pseudo-labeling in transcribing the video content. In this paper, we use the pseudo-labels generated from a teacher model in federated setting with large batch sizes.

Unsupervised Learning in ASR: A related area is the training of representation, foundation, or upstream models from scratch using large volumes of unlabelled data. This model can then be fine-tuned for downstream use cases such as ASR, speaker recognition, among others. This paradigm is contrasted with the use case of incremental updates to a pre-trained ASR model presented in this work. A comprehensive survey of such methods for speech representation learning are in [45]. The upstream model is trained with a *pretext task* such as a generative approach to predict or reconstruct the input given a limited view (eg past data, masking) such as autoregressive predictive coding [12]. In a contrastive approach, a representation is learned that is close to a positive sample and further away from negative samples; wav2vec 2.0 [3] is an exemplar where the representation is trained to be close to a quantized target vector. Finally, in predictive approaches [4, 10, 28], the pretext task is to predict for masked input timeframes, a distribution over a discrete vocabulary such as clustered log-mel features. ASR models pre-trained using these techniques can be updated using ILASR.

7 CONCLUSIONS

We proposed the ILASR framework for privacy preserving incremental learning of end-to-end automatic speech recognition systems. ILASR is a big step forward for production level ASR systems, especially for automatic incremental updates of these systems. In this study of near-real time training with ILASR, we learned that even the converged production level ASR models: 1) can be improved significantly in an incremental fashion with 3% general improvements that can go up to 20% on test sets with new words or phrases; 2) training with large batches arising as a result of communication constraints does not result in degradation; 3) memory replay training is effective at mitigating catastrophic forgetting on older test sets; 4) there is no significant impact of chronological versus random processing of data in IL for speech recognition over a period of six months; and finally; 5) having a significant improvement in teacher models used to generate machine transcripts does not translate to the same scale of improvements in students.

In the future, we will explore the utility of noisy students for iterative self-learning instead of relying on teacher models in ILASR. Real-time resource-constrained on-device speech recognition is still a hard challenge. Here, we plan to further explore different directions such as finding the best hyper parameters [34], controlling leaky gradients [66], stopping gradient inversion and data leakage attacks [58], personalizing ASR depending on the device context, and using smaller teacher models or self-labelling that can be run on device. Approximate gradient computation techniques may be required with severe compute resource limitations. Further, exploring methods of integrating weak supervision information from inferred or explicit user feedback from a session of interactions as well as externally updated language models are avenues of further research.

ACKNOWLEDGMENTS

We thank Kishore Nandury, Fred Weber, and Anand Mohan for discussions related to production ASR and utterance selection heuristics. Valentin Mendelev assisted with delta testset construction to measure the impact of IL on new data. We thank Bach Bui, Ehry MacRostie, Chul Lee, Nikko Strom, and Shehzad Mevawalla for helpful discussions, review and support. We are indebted to the Alexa Speech Recognition group for comments, dataset construction and training infrastructure development.

REFERENCES

[1] Al-Rubaie, M. and Chang, J. M. Privacy-preserving machine learning: Threats and solutions. *IEEE Security & Privacy*, 17(2):49–58, 2019.
[2] Anthony, R. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2):123–146, 1995.
[3] Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460, 2020.
[4] Baevski, A., Hsu, W.-N., Xu, Q., Babu, A., Gu, J., and Auli, M. Data2vec: A general framework for self-supervised learning in speech, vision and language. *arXiv preprint arXiv:2202.03555*, 2022.
[5] Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečnỳ, J., Mazzocchi, S., McMahan, H. B., et al. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*, 2019.
[6] Bourlard, H. A. and Morgan, N. *Connectionist speech recognition: a hybrid approach*, volume 247. Springer Science & Business Media, 2012.
[7] Cai, H., Gan, C., Zhu, L., and Han, S. Tinytl: Reduce memory, not parameters for efficient on-device learning. *Advances in Neural Information Processing Systems*, 33:11285–11297, 2020.
[8] Castro, F. M., Marín-Jiménez, M. J., Guil, N., Schmid, C., and Alahari, K. End-to-end incremental learning. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 233–248, 2018.
[9] Chen, K. and Huo, Q. Scalable training of deep learning machines by incremental block training with intra-block parallel optimization and blockwise model-update filtering. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5880–5884, 2016. doi: 10.1109/ICASSP.2016.7472805.
[10] Chen, S., Wang, C., Chen, Z., Wu, Y., Liu, S., Chen, Z., Li, J., Kanda, N., Yoshioka, T., Xiao, X., et al. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *arXiv preprint arXiv:2110.13900*, 2021.
[11] Chiu, C.-C., Sainath, T. N., Wu, Y., Prabhavalkar, R., Nguyen, P., Chen, Z., Kannan, A., Weiss, R. J., Rao, K., Gonina, E., et al. State-of-the-art speech recognition with

sequence-to-sequence models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4774–4778. IEEE, 2018.

[12] Chung, Y.-A. and Glass, J. Generative pre-training for speech with autoregressive predictive coding. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3497–3501. IEEE, 2020.

[13] Cui, X., Lu, S., and Kingsbury, B. Federated acoustic modeling for automatic speech recognition. *CoRR*, abs/2102.04429, 2021.

[14] Dey, S., Motlicek, P., Bui, T., and Dernoncourt, F. Exploiting semi-supervised training through a dropout regularization in end-to-end speech recognition. *arXiv preprint arXiv:1908.05227*, 2019.

[15] Dierks, T. and Rescorla, E. The transport layer security (tls) protocol version 1.2. 2008.

[16] Dimitriadis, D., Kumatani, K., Gmyr, R., Gaur, Y., and Eskimez, S. E. A federated approach in training acoustic models. In *Proc. Interspeech*, 2020.

[17] French, R. M. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

[18] Gao, Y., Parcollet, T., Fernandez-Marques, J., de Gusmao, P. P. B., Beutel, D. J., and Lane, N. D. End-to-end speech recognition from federated acoustic models, 2021.

[19] Geyer, R. C., Klein, T., and Nabi, M. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

[20] Golmant, N., Vemuri, N., Yao, Z., Feinberg, V., Gholami, A., Rothauge, K., Mahoney, M. W., and Gonzalez, J. On the computational inefficiency of large batch sizes for stochastic gradient descent. *arXiv preprint arXiv:1811.12941*, 2018.

[21] Goodfellow, I. J., Mirza, M., Xiao, D., Courville, A., and Bengio, Y. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013.

[22] Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[23] Granqvist, F., Seigel, M., van Dalen, R., Cahill, A., Shum, S., and Paulik, M. Improving on-device speaker verification using federated learning with privacy, 2020.

[24] Graves, A. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*, 2012.

[25] Gulati, A., Qin, J., Chiu, C.-C., Parmar, N., Zhang, Y., Yu, J., Han, W., Wang, S., Zhang, Z., Wu, Y., et al. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*, 2020.

[26] Guliani, D., Beaufays, F., and Motta, G. Training speech recognition models with federated learning: A quality/cost framework. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3080–3084. IEEE, 2021.

[27] Hoffer, E., Hubara, I., and Soudry, D. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. *arXiv preprint arXiv:1705.08741*, 2017.

[28] Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R., and Mohamed, A. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021.

[29] Jiang, H. Confidence measures for speech recognition: A survey. *Speech communication*, 45(4):455–470, 2005.

[30] Kalgaonkar, K., Liu, C., Gong, Y., and Yao, K. Estimating confidence scores on asr results using recurrent neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4999–5003. IEEE, 2015.

[31] Karita, S., Watanabe, S., Iwata, T., Ogawa, A., and Delcroix, M. Semi-supervised end-to-end speech recognition. In *Interspeech*, pp. 2–6, 2018.

[32] Karita, S., Watanabe, S., Iwata, T., Delcroix, M., Ogawa, A., and Nakatani, T. Semi-supervised end-to-end speech recognition using text-to-speech and autoencoders. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6166–6170. IEEE, 2019.

[33] Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

[34] Khodak, M., Li, T., Li, L., Balcan, M., Smith, V., and Talwalkar, A. Weight sharing for hyperparameter optimization in federated learning. In *Int. Workshop on Federated Learning for User Privacy and Data Confidentiality in Conjunction with ICML 2020*, 2020.

[35] Kudo, T. Subword regularization: Improving neural network translation models with multiple subword candidates. In *ACL*, 2018.

[36] Li, B., Sainath, T. N., Pang, R., and Wu, Z. Semi-supervised training for end-to-end models via weak distillation. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2837–2841. IEEE, 2019.

[37] Li, M., Zhang, T., Chen, Y., and Smola, A. J. Efficient mini-batch training for stochastic optimization. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pp. 661–670. ACM, 2014.

[38] Lin, T., Stich, S. U., and Jaggi, M. Don't use large mini-batches, use local sgd. *ArXiv*, abs/1808.07217, 2020.

[39] Ling, S., Liu, Y., Salazar, J., and Kirchhoff, K. Deep contextualized acoustic representations for semi-supervised speech recognition. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6429–6433. IEEE, 2020.

[40] Mahajan, P. and Sachdeva, A. A study of encryption algorithms aes, des and rsa for security. *Global Journal of Computer Science and Technology*, 2013.

[41] Masters, D. and Luschi, C. Revisiting small batch training for deep neural networks. *arXiv preprint arXiv:1804.07612*, 2018.

[42] McCandlish, S., Kaplan, J., Amodei, D., and Team, O. D. An empirical model of large-batch training. *arXiv preprint arXiv:1812.06162*, 2018.

[43] McCloskey, M. and Cohen, N. J. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pp. 109–165. Elsevier, 1989.

[44] McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pp. 1273–1282. PMLR, 2017.

[45] Mohamed, A., Lee, H.-y., Borgholt, L., Havtorn, J. D., Edin, J., Igel, C., Kirchhoff, K., Li, S.-W., Livescu, K., Maaløe, L., et al. Self-supervised speech representation learning: A review. *arXiv preprint arXiv:2205.10643*, 2022.

[46] Nadeu Camprubí, C., Hernando Pericás, F. J., and Gorricho Moreno, M. On the decorrelation of filter-bank energies in speech recognition. In *EUROSPEECH'95: 4th European Conference on Speech Communication and Technology: Madrid, Spain: 18-21 September 1995*, pp. 1381–1384, 1995.

[47] Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 5206–5210. IEEE, 2015.

[48] Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D., and Le, Q. V. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*, 2019.

[49] Parthasarathi, S. H. K. and Strom, N. Lessons from building acoustic models with a million hours of speech. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6670–6674. IEEE, 2019.

[50] Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečnỳ, J., Kumar, S., and McMahan, H. B. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

[51] Sahu, A. K., Li, T., Sanjabi, M., Zaheer, M., Talwalkar, A., and Smith, V. On the convergence of federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 3:3, 2018.

[52] Shallue, C. J., Lee, J., Antognini, J., Sohl-Dickstein, J., Frostig, R., and Dahl, G. E. Measuring the effects of data parallelism on neural network training. *arXiv preprint arXiv:1811.03600*, 2018.

[53] Smith, S. L., Kindermans, P.-J., Ying, C., and Le, Q. V. Don't decay the learning rate, increase the batch size, 2018.

[54] Soltau, H., Liao, H., and Sak, H. Reducing the computational complexity for whole word models. *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 63–68, 2017.

[55] Strom, N. Scalable distributed dnn training using commodity gpu cloud computing. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[56] Sun, P., Feng, W., Han, R., Yan, S., and Wen, Y. Optimizing network performance for distributed dnn training on gpu clusters: Imagenet/alexnet training in 1.5 minutes, 2019.

[57] Swarup, P., Maas, R., Garimella, S., Mallidi, S. H., and Hoffmeister, B. Improving asr confidence scores for alexa using acoustic and hypothesis embeddings. In *Interspeech*, pp. 2175–2179, 2019.

[58] Triastcyn, A. and Faltings, B. Federated generative privacy. *IEEE Intelligent Systems*, 35(4):50–57, 2020.

[59] Wang, H., Yurochkin, M., Sun, Y., Papailiopoulos, D., and Khazaeni, Y. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440*, 2020.

[60] Wang, J., Liu, Q., Liang, H., Joshi, G., and Poor, H. V. Tackling the objective inconsistency problem in heterogeneous federated optimization. *arXiv preprint arXiv:2007.07481*, 2020.

[61] Weninger, F., Mana, F., Gemello, R., Andrés-Ferrer, J., and Zhan, P. Semi-supervised learning with data augmentation for end-to-end asr. *arXiv preprint arXiv:2007.13876*, 2020.

[62] Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., and Fu, Y. Large scale incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 374–382, 2019.

[63] Xiao, A., Fuegen, C., and Mohamed, A. Contrastive semi-supervised learning for asr. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3870–3874. IEEE, 2021.

[64] You, Y., Gitman, I., and Ginsburg, B. Scaling sgd batch size to 32k for imagenet training. *arXiv preprint arXiv:1708.03888*, 6:12, 2017.

[65] Zhao, B., Mopuri, K. R., and Bilen, H. idlg: Improved deep leakage from gradients. *arXiv preprint arXiv:2001.02610*, 2020.

[66] Zhu, L., Liu, Z., and Han, S. Deep leakage from gradients. *Advances in Neural Information Processing Systems*, 32, 2019.