Natural Language Processing and Tabular Data sets in Federated Continual Learning

A usability study of FCL in domains beyond Image classification

Taneshwar Pranav Parankusam

Supervisor(s): Bart Cox, Jérémie Decouchant

EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 25, 2023

Name of the student: Taneshwar Pranav Parankusam
Final project course: CSE3000 Research Project
Thesis committee: Jérémie Decouchant, Bart Cox, Qing Wang

Abstract

Federated Continual Learning (FCL) is a emerging field with strong roots in Image classification. However, limited research has been done on its potential in Natural Language Processing and Tabular datasets. With recent developments in A.I. with language models and the widespread use of mobile devices, it becomes relevant to consider FCL's capabilities in dynamic environments. Our paper discusses and evaluates the applicability of FCL methods between the domains of Natural Language Processing Tabular Data with Image Processing as a baseline. We use Long-Short Term Memory (LSTM) models, DNN's and LeNet-5 as models for Sentiment analysis, Tabular classification and Image classification. Through our experiments, we evaluate the average accuracy and backwards transfer of EWC, GEM, their federated variants and the state-of-the-art FCL method FedWEIT. With these methods, sentiment analysis and tabular classification tasks show that image classification reached over 17% higher average accuracy and achieved a 99.5% average increase in Knowledge Transfer between tasks. Furthermore, we observe that non-federated continual learning methods on average reach higher accuracies than their federated counterparts as well as the state-of-the-art methods.

1 Introduction

In the modern world, the increased usage of mobile devices and Internet of things (IoT) has led to vast amounts of data being generated and consumed on a daily basis worldwide. The abundance of rich data and increased computing power provides a highly suitable environment to train advanced machine learning models such as speech recognition, image classification and text generation. However, in real-world scenarios such data is often sensitive and private in nature leading to legal and ethical restrictions [1]. Moreover, the ever changing nature of data and the ability to adapt to various tasks is a requirement for A.I. in dynamic environments. To address these problems, advances in machine learning has led to the rise of Federated Learning (FL) and Continual Learning (CL).

Federated learning aims to solve the problem of privacy while enabling a model to be trained on a much larger data set [1]. It is a emerging machine learning paradigm where a multitude of decentralized local models used by clients are aggregated to form a global model in a centralized server. Only model parameter values obtained by local training are communicated to the server whereas the training data is not, removing the need share sensitive and private information.

Continual Learning is the process of learning from a sequence of tasks from a stream of data. This field aims to solve the problem of *catastrophic forgetting* which occurs in Deep Neural Networks (DNN's) when tasks arriving sequentially [2]. Older tasks face large drops in accuracy as newer tasks are being trained which results in the network "forgetting".

Combining these two concepts we get **Federated Continual Learning** (FCL) . The local decentralized models are trained on a private sequence of tasks after which they are aggregated together to form a global model. Current state-of-the-art architectures used for FCL include FedWeit architecture [3], which uses the FedAvg aggregation method [1], and derivative method known as FedKNOW[4]. Applications of FCL have been seen with respect to image classification but limited research has observed its effects on natural language processing (NLP) tasks and tabular datasets.

This research paper aims to explore the usability of FCL in relation to natural language processing tasks and tabular data sets. Natural language processing involves processing human language and extracting data from it. This includes tasks such as sentiment analysis, speech recognition and auto completion [5]. Applying FCL to real-world scenarios such as for voice recognition systems like Siri and auto completion keyboards can greatly benefit from the use of FCL based methods to train their models. Furthermore, FCL can also be applied to models that use tabular datasets such as patient records in hospitals. In order for accurate prediction of medical conditions, large amounts of sensitive patient data is required which can be addressed by FCL based architectures.

Our paper will explore the research question *How can FCL (Federated Continual Learning) work in other tasks (NLP or tabular data) apart from Image classification?* The main contributions from this paper are summarized below:

- Integration of various ML models with FCL methods for NLP and Tabular tasks.
- Comparing average accuracies of NLP, Tabular and Image domains.
- Comparing the stability and plasticity properties of ML models in the Federated Continual Learning setting.
- Evaluating overall FCL method performance with respect to the domains.

Subsequent sections are organized as follows. Section 2 we introduce the federated and continual learning settings as well as their respective challenges and the models to be used in the paper. In Section 3 we discuss the FCL setting as well as the state-of-the-art in Natural Language Processing and Tabular Tasks. Our experimental methodology is described in Section 4, with a detailed account of our experimental setup and results in Section 5. An analysis and discussion is presented in Section 7. Conclusions obtained from our discussion as well as future explorations are detailed in Section 8.

2 Background

In this section we provide an intuitive explanation of the background knowledge. First, we describe the Federated setting in 2.1 and its challenges in 2.2. We then go on to detail the continual setting in 2.3 and its respective challenges in 2.4.

Subsection 2.5 provides a detailed explanation of the functionality of the LSTM, one of the models used in our experimentation.

2.1 The Federated setting

The typical federated setting considers a centralized global model with multiple local clients running a cloned model [6]. Local models are trained to either maximize or minimize their local objective function (typically empirical risk [6] using optimization algorithms such as SGD(Stochastic Gradient Descent), batch gradient descent and mini-batch gradient descent. Clients model parameters are aggregated together based on the following global objective function:

$$\min_{\theta \in W} F(\theta) \text{ where } F(\theta) = \sum_{i \in P} p_i f_i(\theta) \qquad \text{(eq.1)}$$

In this equation W is the set of all local models and their parameters, while P represents the partitions of the dataset, where i represents a client. The condition on p_i is that $\sum_{i \in P} p_i = 1$ and p_i here is a weight corresponding to is the size of the relative size of the partition to the total as used by FedAvg [1].

2.2 Challenges in Federated Learning

There are several core challenges to federated learning due to its distributed nature including high communication cost, system heterogeneity and i.i.d data distributions

Communication challenges encompass the amount of data being transferred between local and global models as updates as well as the amount of time it takes. Model size is one problem as larger models require more model parameters to be updated on each communication round. This scales enormously with a large number of connected clients as is today with many modern networks [6]. Another consideration is the total number of communication rounds between a client and the central server as each training round of training adds proportionally more costs. Depending on the type of network whether it is asynchronous or synchronous can greatly affect the time between communication rounds as the server has to potentially wait for all clients to send updates. Latency greatly affects this waiting time as certain clients maybe on low bandwidth channels while others are not.

System Heterogeneity describes the variation in system characteristics[6] across the network such as latency, computational power, etc. This also includes the communication method either synchronous or asynchronous. Synchronous communication ensures that the global model is synchronized with all local models ensuring that all communication is handled uniformly and aids to its simplicity to implement. This deviates from the asynchronous setting which must consider faulty devices, major delays between devices and limited participation by client devices in the training rounds.

I.I.D data means that each random variable, or data reading, is independent from other samples and follows the same probability distributions. This assumption regarding data simplifies the training process by assuming convergence via the Central Limit Theorem and the Law of Large Numbers [6]. However it also reduces performance in real-world scenarios where data is typically non i.i.d causing certain federated algorithms to diverge such as FedAvg. In this research, the focus will be on i.i.d data and future works can consider the effect of non i.i.d on NLP and Tabular Tasks in FCL.

2.3 The Continual setting

Unlike the federated setting where multiple clients focus on completing the same task, the continual setting focuses on one model learning from a sequence of tasks. One aim of a continual learning model is to learn from a dynamic data distribution [2] .

One aim of continual learning is to learn from tasks dynamically. Though each task contains a static distribution, the dynamic nature of incoming tasks results in the overall dynamic distribution for the model to learn [6]. Such dynamic data distributions during run-time mean that models will suffer from accuracy losses as they are typically compiled for a specific static model [7]. Cases where each task has the same label but has a different data distribution is known as Domain-Incremental Learning and for tasks with the same data distribution as well it is known as Instance-Incremental Learning. We will be focusing on Task-Incremental Learning meaning each task contains a disjoint subset of classes.

2.4 Challenges in Continual learning setting

With sequential data, catastrophic forgetting is a major problem as models will overwrite previous knowledge of certain tasks with new knowledge of the current task [8]. In the Continual learning setting, training data from previous tasks is no longer available as tasks come sequentially. This problem has two aspects to it known as learning plasticity and memory stability. Learning plasticity relates to the ability of the model to adapt to new data while memory stability relates to a models ability to retain old knowledge [2]. As one would suspect, these two aspects are inversely proportional to each other and need to be balanced to ensure optimal model performance. There are multiple metrics used to measure the stability-plasticity trade-off with each metric measuring one aspect. With regards to memory stability, the most common metrics are the backward transfer [9] and the forgetting measure [10]. Likewise for measuring learning plasticity, commonly used metrics are the forward transfer [9] and the intransience measure [10]. Countless models have been proposed to counteract the effects of catastrophic forgetting including Elastic Weight Consolidation (EWC) [11] and Gradient Episodic Memory (GEM) [9]. EWC reduces the effect of catastrophic forgetting by selectively slowing down the gradients of the weights of other tasks. GEM on the other hand keeps a subset of the training data from previous tasks as *episodic memory* to retrain the model. These models are the baselines for state-of-the-art FCL models such as FedKNOW [4] and FedWeit[3].

2.5 Models

The **LSTM** or **Long Short-term Memory** is a specialized recurrent neural network which aims to mitigate the vanish-

ing/exploding gradient problem [12]. The structure of a single LSTM cell is described in the figure 1. The σ represents the Sigmoid function and the $tanh$ represents the hyperbolic tangent function [13]. The S(t-1) represents the short-term memory from the previous iteration while the H(t-1) represents the long-term memory. Note that the long-term memory also acts as the output to the entire LSTM cell and contains the final value. A multitude of these cells are used in a single layer similar to the neurons in the multi-layer perceptron.

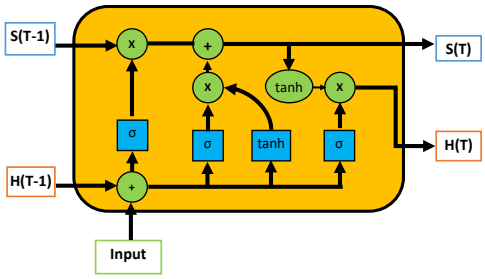

Figure 1: LSTM Cell construction

3 Related Work

This section details the latest developments in the Federated Continual Learning (subsection 3.1) including the state-of-the-art. Additionally, an overview of the various tasks and approaches used in Natural Language Processing Tasks (subsection 3.2) and Tabular Tasks (subsection 3.3) is given.

3.1 Federated Continual Learning

Federated Continual Learning approaches continual learning tasks in a federated way. Each client has a local model which is trained on a sequence of tasks after which the global model aggregates the clients local models.

Federated Continual Learning approaches continual learning tasks in a federated way. Each client has a local model which is trained on a sequence of tasks after which the global model aggregates the clients local models. The naive approaches aims to use the simple Fed Avg algorithm with existing continual learning techniques as discussed in [3]. There are limitations to this approach namely inter-client interference where the gradients from clients who trained on different tasks results in accuracy loss for certain tasks.

FedWeit

Federated weighted inter-client transfer or FedWeit is widely considered a state-of-the-art server-side solution for the task-incremental learning setting as noted by [4], [6]. It approaches FCL by adopting additional model parameters through decomposition which are then learned during the training process. The model parameters are decomposed into 3 subgroups, global parameters θ_g, local base parameters B and task-adaptive parameters A as well as an additional vector mask $m$to transform the local base parameters. For each client c the local parameters for a task t are given as follows eq.2

$$\theta_c = B_c^{(t)} \odot m_c^{(t)} + A_c^{(t)} \sum_{i \in C-c} \sum_{t \in T} \alpha_{i,j} A_i^{(t)} \qquad \text{(eq.2)}$$

A regularization based loss function ensures that the parameters B_c, m_c, A_c and α_c are learned. A sparsity inducing regularization function Ω is used to reduce the total amount of data that needs to be communicated the centralized server at a time. The local client parameters are aggregated together using the FedAvg equation described in eq.1.

3.2 Natural Language processing tasks

Natural language processing (NLP) concerns the understanding of human language and processing it. There are a number of language processing tasks including Speech recognition, speech tagging, named entity recognition, sentiment analysis and natural language generation [5]. Speech recognition converts voice data into text-data. A similar process of converting text to speech or TTS is explored by FedSpeech [14] which employees its own aggregation algorithm through the use of masks. The FedSpeech algorithm is tailored specifically to text-to-speech conversion and certain techniques explored by FedSpeech are only applicable in this scenario.

Speech tagging aims to understand the grammatical structures present in a sentence and tags each word with its corresponding grammatical word. For example, the word ”ran” in ”I ran” is a verb and is thus tagged as such by the model. Similar to speech tagging, Named entity recognition is another task where words/phrases are recognized as specific objects and correlations between objects are understood and classified as objects. For example, in the sentence ”The red car was parked” can be classified as a car object with the color attribute being red. Additionally, Natural Language generation is another well known task such as next word prediction or generation of whole texts as popularly shown with GPT-4 [15]. Finally, there is semantic analysis which aims to extract the meaning behind sentences such as whether a sentence is happy, sad, angry, neutral.

3.3 Tabular Data based Tasks

Apart from computer vision and natural language processing, a majority of fields make use of tabular data such as the medical and financial industries. Tabular data heterogeneous features where each column represents a feature and a row representing a data point. Each feature can have different data types such as images, text,categorical values, geographical coordinates, etc. Common tasks in tabular data include classification, regression (or prediction), generation and imputation [16] and have been explored in both the federated and continual learning setting. Horizontal federated learning is one such variant where the rows in different clients have different sample spaces but have the same feature space [17].

Current state-of-the-art methods in the FCL such as Continual Horizontal Federated Learning (CHFL) [18] focus more on addressing problems relating to unique heterogeneous data rather than addressing the applicability of weighted FCL algorithms on tabular data.

4 Methodology

Within this section, we provide a comprehensive overview of the FCL and CL algorithms used in our experimentation (subsection 4.1). The metrics (subsection 4.2) and a description of how domains will be compared (subsection 4.3) is provided. Additionally, we talk about task specific details for each domain in subsections 4.4, 4.5 and 4.6.

4.1 Algorithms

In order to validate the performance of FCL models in the Natural Language Processing setting and the Tabular setting, a comparison has been made to Image classification as it is the baseline for a majority of FCL algorithms. As both NLP and Tabular tasks have been explored in the CL setting, baselines are drawn with reference to existing literature. This baseline is compared with FCL methods and their performance in tasks. Each Task is tested with the following CL and FCL frameworks:

Continual Learning
• EWC
• GEM
Federated Continual Learning
• FedAvg + EWC
• FedAvg + GEM
• FedWeit

Figure 2: FCL and CL methods used in experiments

4.2 Metrics

The metrics used to compare these frameworks will be average accuracy [2] and backward transfer[9]. For the below metrics, we make the assumption that the test sets of each task are available.

Average Accuracy

Average accuracy (ACC) is metric used to measure the general performance of a model during incremental learning. It is measured for the *i*th task where $a_{i,j}$ represents accuracy of task i when evaluated on the test set of the *j*th task. Then the ACC is given by the following equation:

$$ACC = \frac{1}{j}\sum_{i=1}^{j} a_{i,j} \qquad \text{(eq.3)}$$

Backward Transfer

Backward transfer measures the memory stability of the model during training and observes the effect of the current task on past tasks. It measures the impact that task *j* has on a past task *i*. A negative value for backward transfer indicates that the DNN is forgetting previously learned knowledge about past tasks. The equation for BT is described below as well:

$$FT = \frac{1}{j-1}\sum_{i=1}^{j-1} a_{i,j} - a_{i,i} \qquad \text{(eq.4)}$$

4.3 Comparison of Domains

In order to perform a fair evaluation of the given domains, each task is a form of classification, as it provides the simplest and most effective basis for comparison. Furthermore, considering that we are in the federated continual learning setting, we consider the task-incremental learning paradigm where each task contains a disjoint subset of classes. This means that a class C along with its training data is only present in a single task T and not in other tasks. Additionally, the total number of classes used in the experiment is given by the lowest number of classes present in all the datasets. For example, if the first dataset contains 4 classes and the second dataset has 5 classes, only 4 classes from each dataset are used for their respective experiments. This is done to ensure that tasks are more comparable for data analysis.

Additionally , a crucial assumption that is made is that all testing data is available for the algorithm which ensures that the backward transfer and average accuracy can be calculated. Without this assumption, limited information regarding the memory-stability and model accuracy deduced. Due to the inverse relationship between memory-stability and learning plasticity, calculating backward transfer allows us to infer plasticity proprieties as well. Each metric is implemented for domain and its corresponding tasks.

Finally, each model is written with 2 variants, a non-decomposed variant and a decomposed variant. The decomposed variant is implemented for the FedWeit algorithm as each parameter must be separated as outlined in eq.2. The non-decomposed variant is used for the other CL and FCL algorithms.

4.4 Natural Language Processing

In order to provide a fair evaluation of FCL in an NLP setting and the typical image classification setting, we evaluate FCL frameworks on semantic analysis tasks. Semantic analysis is a task which classifies sentences to abstract meaning. This provides direct parallels towards the image classification setting as both aim to solve classification problems. For this task we use emotions dataset [19] which consists of 16,000 sentences and provides 6 different classes of emotions including sadness, joy, fear, anger, surprise and love.

For most NLP tasks, we cannot use words directly and they must be converted to numerical values. The input vector is given by size $|V|$ represents the size of the vocabulary V and each word is encoded with one-hot encoding, meaning that each word is assigned a single position in the vector V [20]. The LSTM model is prepended with an additional embedding layer which transforms the input vector into word embedding. these word embedding encode the lexicographical structure of the sentences which is then further passed through the LSTM model.

In the FCL setting, we must consider that the dataset must be split into various tasks, and for each task there must be a corresponding test set to evaluate the performance of the task. We use task-increment learning where tasks consist disjoint classes and data. Data and classes is split evenly so that each task has similar number of data points.

4.5 Tabular Data

Tabular data consists of heterogeneous data where each column is a feature (possibly of a different data type) and each row is an entry corresponding to the task. For the tasks in the research, we use the Forest Cover Type [21] dataset from the University of California Irvine which aims to classify geographical sites into 7 different forest coverages. The dataset consists of 54 different geographical attributes as well as 581012 rows in total.

As our classification model for the task, we make use of an Deep Neural Network (DNN) with ReLU activation and Softmax function for multi-class classification. CNN's and RNN's are not suitable for this dataset as it provides no spatial correlation between features and data points have no temporal/sequential dependence on each other, thus the use of simple DNN's.

4.6 Image classification

Image classification is the baseline task which is typically used in the Federated Continual Learning setting to evaluate the performance of algorithms and models. For this classification task, the CiFAR100 dataset [22] consists of a total of 100 classes each with 500, 32x32 pixel images providing sufficient training material. Considering image classifications prominence in FCL, it acts as a baseline comparison against natural language processing tasks and tabular tasks.

Typically for Image classification, convolutional neural networks (CNN's) are used as they can extract patterns between neighbouring pixels to grasp deeper patterns in the data. For this task, we use of LeNet-5 [23], a CNN based architecture described in [3] due to its small size. This reduces the communication overhead as well as the training time in comparison to more complex and deep convolutional neural network such as ResNet-18 [24] used in [3].

5 Experimental setup and Results

A detailed description regarding the experimental setup is given in this section. A general overview is given in subsection 5.1 and specific model parameters are discussed in subsection 5.2. The experimental results as well as their analysis is detailed in 5.3.

5.1 Implementation overview

The FCL algorithms and the ML models were implemented in python using the PyTorch library. The FedKNOW repository [4] acted as the basis for code development. Each algorithm is modified to include the metrics mentioned in section 4.2 as well as the LSTM, DNN and LeNet implementations for their corresponding tasks. Experiments were run on a Nvidia GeForce RTX 2060 Mobile GPU and an Intel Core i7 10th gen Mobile processor.

For each domain described in sections 4.4, 4.5, 4.6 a total of 6 classes were selected from their respective datasets. Tasks require a minimum of 2 classes for classification problems, thus leading to a total of 3 tasks per experiment. In the federated setting, 5 clients were chosen each with 1 epoch of training and a batch size of 1000 samples. Larger batch sizes and number of clients were not chosen due to computational limitations.

5.2 Model parameters

In this section we elaborate on the structure of the models to be used for each domain including the LSTM, DNN and LeNet model. The decomposed variants mentioned in section 4.3 consist of the same structure as their non-decomposed counterparts. The LeNet-5 uses the same architecture described in [3] and is detailed in the appendix B

Layers	Output shape	Parameters
Embedding Layer	(batch, vocab size, 128)	size of vocabulary 1350279 words, 128 embedding features
LSTM Layer	(batch, 128, 128, 1)	128 embedding features, 128 output features, 1 layer
Linear Layer	(batch, 128, 6)	6 output classes with log softmax function

Table 1: Overview of model structure of an LSTM with layer specific details.

The LSTM model structure is given in Table 1. An embedding layer is used to extract patterns in sentences and phrases and encode them in an embedding vector. The vocabulary size is given by the number of unique words present in the dataset and is determined when pre-processing of the dataset. 128 embedding features and hidden layers are used, to reduce communication costs with limited loss to performance. This value was selected as state-of-the art models such as the models in [25] use values as small as 150 and 300 for significant results.

For the DNN, the structure, described in Table 2 consists of multiple layers with ReLU activation. Due to the limited input size for this tabular dataset, a larger depth (multiple layers) can be used effectively learn the underlying distribution.

5.3 Results

This section provides an overview of the results obtained on in the FCL and CL algorithm in domains of natural language processing, tabular data and image processing. Tabular data

Layers	Output shape	Parameters
Linear layer	(batch, 54, 500)	54 input features, 500 hidden features
2 x Linear layer	(batch, 500, 500)	500 hidden input features, ReLU activation
Linear Layer	(batch, 500, 6)	6 output classes with log softmax function

Table 2: Overview of model structure of an DNN with layer specific details.

is described in section 5.3, nlp is discussed in section 5.3 and the image processing is described in section 5.3.

Tabular Results

In Figure 3, it is evident that each algorithm experiences a significant decline in accuracy when the task changes every 30 rounds. The accuracy decrease correlates proportionally to the number of tasks that have been learnt. This relationship is further explained by the relative accuracy between each task shown in Table 4, which is approximately given by $\frac{1}{T}$ where T is the number of tasks learnt by the model.

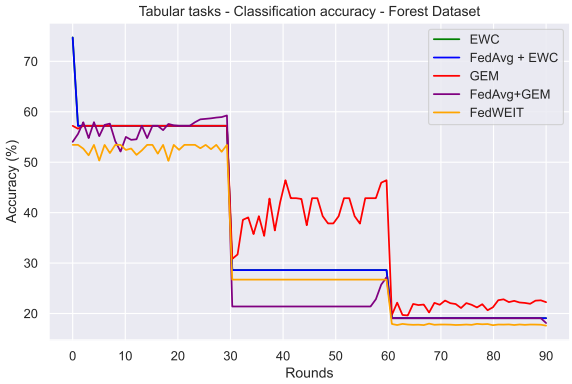

Figure 3: Accuracy obtained on Forest dataset on EWC, FedAvg + EWC, GEM, FedAvg + GEM and FedWEIT.

Furthermore, in Figure 3 we observe that the accuracy for each task stays relatively constant for the majority of the CL and FCL algorithms. The exception being GEM which shows a progressive increase in accuracy as the rounds progress. Constant accuracy implies that the DNN model's loss function has been optimized within the first round and thus shows limited fluctuation with an increase in the number of rounds.

Moreover, GEM stands out by achieving an overall higher average accuracy in Task 2 and Task 3. This is in stark contrast to its federated counterpart, which displays the lowest overall performance in Task 2. EWC and FedAvg+EWC yield nearly

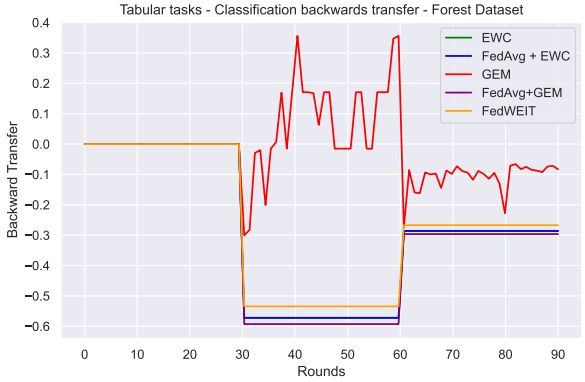

Figure 4: Backward Transfer on Forest dataset on EWC, FedAvg + EWC, GEM, FedAvg + GEM and FedWEIT.

identical results, while FedWeit exhibits the lowest overall accuracy. However, it is worth noting that the FedWEIT algorithm demonstrates lower negative backward transfer, indicating superior memory stability compared to other algorithms, except for GEM.

	Average Backwards Transfer		
Domain	**Task 1**	**Task 2**	**Task 3**
NLP	0	-0.140	-0.02
Tabular	0	-0.441	-0.248
Image	0	-0.05	-0.02

Table 3: Table comparing the average backward transfer of all FCL and CL methods from their respective domains

NLP results

As evident in Figure 5, the accuracy from sentiment analysis exhibits minimal variability between algorithms on the initial two tasks. In relation to this, purely continual learning methods including EWC and GEM have shown to have a lower accuracy in earlier tasks than their federated counter parts. Additionally, it is clear that FedWEIT shows the fastest convergence of its loss function, resulting in limited change throughout the rounds for each task. The regularization based methods including EWC and FedWEIT show limited variance in their accuracy during training due to the penalization term in the loss function preventing the gradients from deviating.

Regarding memory-stability, we observe that EWC, GEM and their federated variants consistently maintain backward transfer values close to 0, implying on average positive knowledge transfer and stable memory. Similar to the tabular results in Figure 4, the arrival of a new task causes large drops in memory-stability and shows a significant increase in knowledge loss. As opposed to the aforementioned tabular results, we observe that backward transfer values for most algorithms (other than FedWEIT) revert back to their original values in the second task and increase notably in the final task.

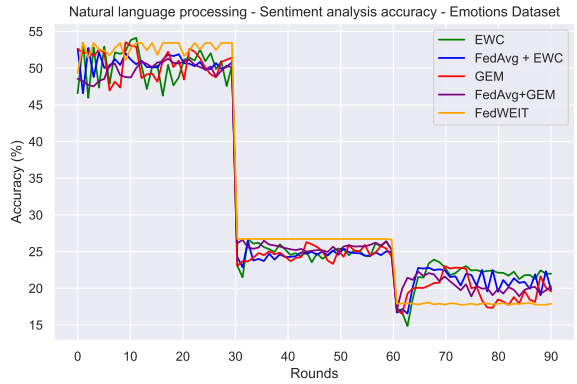

Figure 5: Accuracy obtained on Emotions dataset on EWC, FedAvg + EWC, GEM, FedAvg + GEM and FedWEIT.

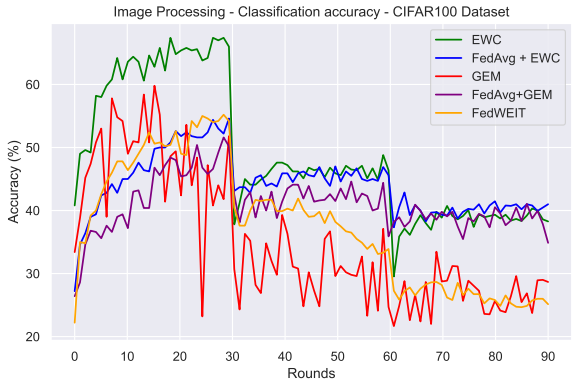

Figure 7: Accuracy obtained on Image dataset on EWC, FedAvg + EWC, GEM, FedAvg + GEM and FedWEIT.

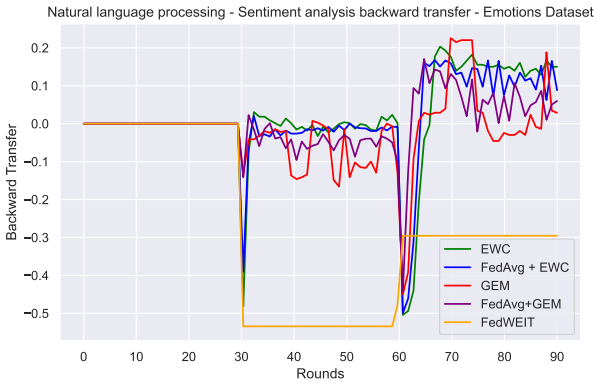

Figure 6: Accuracy obtained on Emotions dataset on EWC, FedAvg + EWC, GEM, FedAvg + GEM and FedWEIT.

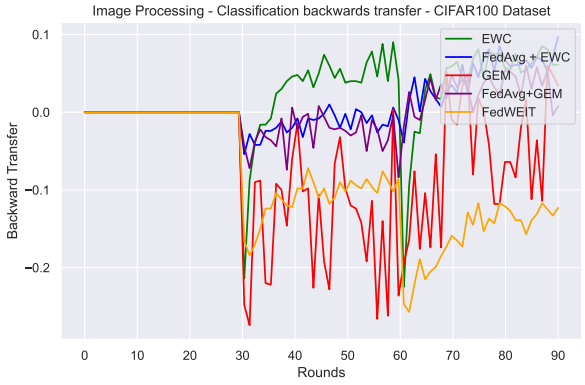

Figure 8: Backward Transfer obtained on Image dataset on EWC, FedAvg + EWC, GEM, FedAvg + GEM and FedWEIT.

Image results

In Image classification, the accuracy of all CL and FCL algorithms are above 20%. Unlike both the NLP and Tabular setting, none of the methods converge to a stable accuracy for every task but instead show a linear trend. The GEM algorithm shows a progressive downward decrease in accuracy throughout the learning process. In contrast, the other algorithms show convergence to higher accuracy values for each task. Additionally, more noise is present in image classification tasks than in natural language processing and tabular tasks.

The backward transfer of each algorithm gives insight into the large variations found in their respective accuracies. EWC and FedAvg + EWC exhibit the greatest memory-stability, maintaining an average value greater than 0.01 while other variants range from 0 to -0.2. FedWEIT experiences forgetting, which, though not catastrophic, leads significant accuracy losses across tasks when compared to the competitors.

6 Responsible Research

With recent developments in NLP and Big Data, privacy has grown to be more and more of a concern with large scale text-generation models like GPT-4 [15] requiring enormous amounts of training data. Naturally, data must collected and used in accordance with local privacy laws and regulations which have strict guidelines on collecting and storing personal data. The application of FCL based algorithms, which is explored by our research, reduces such privacy concerns as client data is kept decentralized. However, the federated approach comes with its own set of risks as instead parameters are shared which can be malicious used to manipulate the global models [26].

Our research follows responsible and ethical practices to ensure the reproducible, reliable and valid results from our experimentation. The datasets used are all open source and developed in published studies with strict data collection methodologies. No sensitive private information is used during experimentation. The code developed during experimentation as well as its corresponding documentation and results are made public on the TU Delft repository. The hyper pa-

	Mean Accuracy per Task			Relative Accuracy (RA) per Task		
Domain	**Task 1**	**Task 2**	**Task 3**	**RA Task 1**	**RA Task 2**	**RA Task 3**
NLP	50.84	25.33	20.05	1.0	0.50	0.39
Tabular	56.43	29.12	19.34	1.0	0.52	0.34
Image	49.00	40.40	34.11	1.0	0.83	0.71

Table 4: Mean accuracy for the forest dataset across all algorithms for each task. Each tasks accuracy is compared to the initial task.

rameters used are detailed further in the appendix section 9 so readers may reproduce similar results during experimentation.

7 Discussion

The results from section 5.3 provide a basis for comparison of the various domains. The tasks for each of the three domains and their average accuracy are described in Table 4. It is evident that the algorithms obtained maximum accuracy on the first task in the NLP setting. However the mean task accuracy is higher for Image Classification. Image classification performed on average 17.7% than sentiment analysis and 28.4% better than tabular classification. Notably, the direct inverse proportionality observed in both sentiment analysis and tabular classification as described in 5.3 is not present in the image classification setting. The relative accuracy follows a more linear trend implying that memory stability and learning plasticity of CL and FCL algorithms are more suited to spatially correlated data such as images.

Moreover, NLP and tabular tasks converge quickly to a local minima within the initial rounds of the task, preventing the further loss reduction. This is in stark contrast to the image classification setting. It can be concluded from that NLP and tabular tasks are not as robust as image classification tasks towards negative knowledge transfer. This is further demonstrated in the average backward transfer for the different tasks. The image classification task shows a 99.5% decrease in knowledge loss in comparison to sentiment analysis and a 99.8% decrease in comparison to tabular classification. This is directly correlated with its average accuracy and thus also relates to the convergence of the loss function. We can conclude that FCL algorithms show greater effectiveness in the image processing domain than in natural language processing and tabular data domain.

Furthermore, we notice that the Federated variants of continual learning methods including FedAvg+EWC and FedAvg+GEM show more consistent and less varied data. This can be attributed to the increased number of clients who each train on similar samples from I.I.D data. Averaging across multiple clients not only reduces the noise from the training data but also converges the distribution faster. Additionally, we observe that the state-of-the-art FCL algorithm FedWEIT yields substandard performance in comparison to other regularization-based and rehearsal based FCL methods. This can be attributed to a decomposition of parameters described in eq.2 which drastically raises the number of model parameters that need to be trained. This is particularly relevant in the image processing scenario where a considerable number of rounds are required per task to optimize model parameters

8 Conclusions and Future work

We demonstrated the effectiveness of Federated Continual Learning algorithms in Natural Language Processing tasks and Tabular tasks, specifically sentiment analysis and tabular classification. An analytical comparison between baseline continual learning algorithms, their federated counterparts and the state-of-the art indicate that FCL methods perform better in Image classification than in NLP or Tabular tasks. This performance is directly correlated to their memory-stability, where LSTM and DNN models exhibit much lower stability but higher plasticity which contrasts with a CNN models who demonstrate greater stability and limited plasticity.

Furthermore, we have discover that non-federated continual learning methods achieve an overall accuracy greater than their federated counterparts in all domains. This is due to the statistical averaging of federated methods on clients and their varying model parameters. The state-of-the art model FedWEIT suffers in all domains due to the drastic increase in model parameters caused by parameter decomposition leading to lower accuracy and knowledge retention.

Lastly, in this work we have utilized relatively shallow models with at most 5 layers for the various tasks due to computational and time constraints. Extensions to this work include deeper models which can be used to achieve higher accuracy at the cost of increased training time. Large language models such as BERT and transformers [27] can be used for NLP tasks, and deeper and more advanced DNN's can be used for Tabular tasks. Such models can be adapted for a larger range of tasks including natural language generation, speech tagging and named entity recognition for NLP. The applicability of these FCL models in the Horizontal Federated Learning setting for tabular data can also analysed to provide a more comprehensive evaluation.

References

[1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. Google Inc., 2017.

[2] L. Wang, X. Zhang, H. Su, J. Zhu, Fellow, and IEEE, "A comprehensive survey of continual learning theory, method and application," IEEE. ArXiv, 2023.

[3] J. Yoon, W. Jeong, G. Lee, E. Yang, and S. J. Hwang, "Federated continual learning with weighted inter-client transfer," in *Proceedings of the 38th International Conference on Machine Learning*. Korea Advanced Institute of Science and Technology (KAIST), 2021.

[4] Y. Luopan, R. Han, Q. Zhang, C. H. Liu, and G. Wang, "Fedknow: Federated continual learning with signature task knowledge integration at edge," Beijing Institute of Technology. ArXiv, 2022.

[5] IBM, "What is natural language processing (nlp)?" 2023.

[6] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," 2020.

[7] T. S. Abdelrahman and K. L. Ma, "Evaluation of dynamic data distributions on numa shared memory multiprocessors," in *International Conference on Parallel and Distributed Processing Techniques and Applications*. The University of Toronto, 1996.

[8] S. kar, G. Castellucci, S. Filice, S. Malmasi, and O. Rokhlenko, "Preventing catastrophic forgetting in continual learning of new natural language tasks," in *Conference on Knowledge Discovery and Data Mining*. Amazon, 2022.

[9] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," in *Advances in Neural Information Processing Systems 31 (NeurIPS 2018)*. Conference on Neural Information Processing Systems, 2018.

[10] A. Chaudhry, P. K. Dokania, T. Ajanthan, and P. H. S. Torr, "Riemannian walk for incremental learning: Understanding forgetting and intransigence," in *Proceedings of 15th European Conference on Computer Vision*, 2018.

[11] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, and K. Milan, "Overcoming catastrophic forgetting in neural networks," 2017.

[12] S. Horchereiter and J. Schmidhuber, "Long short-term memory," 1997.

[13] A. Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network."

[14] Z. Kian, Y. Ren, M. Lei, and Z. Zhao, "Fedspeech: Federated text-to-speech with continual learning," in *International Joint Conferences on Artificial Intelligence*. Zhejiang University, 2021.

[15] OpenAI, "Gpt-4 technical report," 2023.

[16] V. Borisov, T. Leemann, K. Seßler, J. Haug, M. Pawelczyk, and G. Kasneci, "Deep neural networks and tabular data: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2022.

[17] H. Wu, Z. Zhao, L. Y. Chen, and A. V. Moorsel, "Federated learning for tabular data: Exploring potential risk to privacy," in *2022 IEEE 33rd International Symposium on Software Reliability Engineering (ISSRE)*. Los Alamitos, CA, USA: IEEE Computer Society, nov 2022, pp. 193–204.

[18] J. Mori, I. Teranishi, and R. Furukawa, "Continual horizontal federated learning for heterogeneous data," *2022 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2022.

[19] E. Saravia, H.-C. T. Liu, Y.-H. Huang, J. Wu, and Y.-S. Chen, "CARER: Contextualized affect representations for emotion recognition," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 3687–3697.

[20] E. Sezerer and S. Tekir, "A survey on neural word embeddings," 2021.

[21] J. Blackard, "Covertype," UCI Machine Learning Repository, 1998.

[22] A. Krizhevsky, V. Nair, and G. Hinton, "Learning multiple layers of features from tiny images," 2009.

[23] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, jun 2016, pp. 770–778.

[25] S. Wang and J. Jiang, "Learning natural language inference with LSTM," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 1442–1451.

[26] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," in *Future Generation Computer Systems*. Elsevier, 2020.

[27] P. Yand, D. Li, and P. Li, "Continual learning for natural language generations with transformer calibration," Baidu Research. Association for Computational Linguistics, 2022.

A Accuracy Tables

	Avg Accuracy per Task			Relative Accuracy per Task		
Method	**Task 1**	**Task 2**	**Task 3**	**RA Task 1**	**RA Task 2**	**RA Task 3**
EWC	57.82	28.61	19.07	1.0	0.50	0.33
FedEWC	57.82	28.61	19.07	1.0	0.50	0.33
GEM	57.20	40.06	21.67	1.0	0.70	0.38
FedGEM	56.61	21.60	19.07	1.0	0.38	0.34
FedWEIT	52.71	26.71	17.81	1.0	0.51	0.34
Average	56.43	29.12	19.34	1	0.52	0.34

Table 5: Mean accuracy for the forest dataset across all algorithms for each task. Each tasks accuracy is compared to the initial task.

	Avg Accuracy per Task			Relative Accuracy per Task		
Method	**Task 1**	**Task 2**	**Task 3**	**RA Task 1**	**RA Task 2**	**RA Task 3**
EWC	50.35	24.96	21.53	1.0	0.50	0.43
FedEWC	50.67	24.58	20.89	1.0	0.49	0.41
GEM	50.68	24.72	19.70	1.0	0.49	0.39
FedGEM	49.79	25.65	20.24	1.0	0.52	0.41
FedWEIT	52.72	26.71	17.87	1.0	0.51	0.34
Average	50.84	25.33	20.05	1	0.50	0.39

Table 6: Mean accuracy for the Emotions dataset across all algorithms for each task. Each tasks accuracy is compared to the initial task.

	Avg Accuracy per Task			Relative Accuracy per Task		
Method	**Task 1**	**Task 2**	**Task 3**	**RA Task 1**	**RA Task 2**	**RA Task 3**
EWC	61.46	45.69	38.37	1.0	0.74	0.62
FedEWC	46.53	450.9	40.13	1.0	0.97	0.86
GEM	47.39	30.84	26.61	1.0	0.65	0.56
FedGEM	42.06	41.96	38.87	1.0	0.99	0.92
FedWEIT	47.54	38.43	26.55	1.0	0.81	0.56
Average	48.99	40.40	34.11	1	0.84	0.71

Table 7: Mean accuracy for the CiFAR100 dataset across all algorithms for each task. Each tasks accuracy is compared to the initial task.

B Model Structure

Layers	**Output shape**	**Parameters**
Convolution Layer 1	(batch size, 20, 32, 32)	20 channels, stride 1, padding 2 and kernel size 5x5
Max Pool Layer 1	(batch size, 20, 16, 16)	2 stride, padding 1 and kernel size 3x3
Convolution Layer 2	(batch size, 50, 16, 16)	50 channels, stride 1, padding 2, kernel size 5x5
Max Pool Layer 2	(batch size, 50, 8, 8)	2 stride, padding 1, kernel size 3x3
Linear Layer 1	(batch size, 800, 800)	800 inputs, 800 outputs
Linear Layer 2	(batch size, 800, 500)	800 inputs, 500 outputs
Linear Layer 3	(batch size, 500, num classes)	500 inputs and output is the number of classes

Table 8: LeNet-5 architecture used for Image classifcation

Hyper Parameter	value
Number of classes	6
Number of clients	5
Number of rounds	30
Number of tasks	3
Batch Size	1000
Learning Rate	0.01
Training Data split	0.9
Testing Data split	0.1
local epochs	1

Table 9: Hyper Parameters used during training