Fed-Squares: Use of Squares-Based Tool for analysis of Federated Learning Simulations

Thiago V. M. Souza*
Centro de Informática - CIN - UFPE

ABSTRACT

Federated Learning (FL) is a new paradigm on Machine Learning (ML) that permits train an ML model in a distributed way. The training process is executed at the edge-preserving of the user's privacy since the data never leaves the local device that differs from the standard paradigm centralized, where the users sent data to a central server. The FL scenario deals with a more heterogeneous scenario related to data distribution between clients that are not analyzed by an engineer as in a centralized paradigm. In general, statistical techniques such as accuracy, recall, precision, log loss, and the confusion matrix, in visualization, are used to compare the models and interpret if the training process occurs well, and the model converged to a defined scenario. However, these techniques only give an overall idea about how the data is used and threaded by the model, given similar comparison values between two models with very different characteristics. The SQUARES [6] technique permits a more precise evaluation of the model at the instance level, which allows the analysis, data biases, inspection, outliers, and how the model responds while training to the tested samples. This paper shows a Square prototype's development and evaluation of an FL image classification model's different scenarios. In this more challenging scenario, we expect to help through the visualization finds some insights and exciting inferences and cases that are impossible only with the standards visualizations and metrics before mentioned and most found on ML benchmarks.

Index Terms: Visualization—Visualization design and Machine Learning evaluation methods—Squares; Machine Learning—Deep Learning—Federated Learning

1 INTRODUCTION

Understand the machine learning (ML) algorithms and how we interpret your processing of learning is a vast area of research and development. The data visualization (DV) field helps the researchers and engineers get some level of interpretability of the results and scenarios generated by these models, where in general, they process the data as a black box. In the last years, the Deep Learning (DL) algorithms improve the results in many tasks threaded by the area as image classification, object detection, and others, near or surpassing a human-level accuracy. Many data visualization techniques measure the trained model's results and help understand the training process and model performance. Many of these techniques gauge only global aspects of the model accuracy and train, at a global or class level, with statistical measures plotted on chart lines or confusion matrix, found on the majority of libraries available for machine learning. However, these evaluation types do not allow the direct analysis of the data samples individually and how the model responds to each one in a comfortable and integrated way. The SQUARES [6] Visualization proposes this process that can help quickly discover patterns, bias, and problems on train and test datasets. The technique evaluates a standard centralized ML paradigm, where one machine center all the data for training and test. In this paper, we evaluate the squares on a Federate Learning (FL) scenario, a new paradigm on ML that preserves the users' data privacy training the model at the edge, with the user's local data. With this more heterogeneous scenario, we developed a SQUARES prototype to perform some analysis on an FL environment generated by the use of the benchmark LEAF [2] using the ten classes filtered dataset FEMINIST based on the EMNIST dataset [3]. Some discussions and quick evaluations are achieved by the tool compared to the other techniques available at the benchmark itself.

The objective of this solution are listed below:

- Improves the inspection of the model generated by the training process and evaluates the components' impact in a heterogeneous environment in the model test.
- Provide better Monitoring of client and server test accuracy
- Provide better Monitoring of client and server test score for each class and tested sample
- Follow details about the model and dataset at the instance level

Combining this information allows identifying inappropriate sample patterns, outliers in the test databases, and the better inspection of the model trained.

The paper is organized as follows, in section 2 will be commented on some works related to our solution from the data visualization. Section 3 described the background of some areas and technologies fundamental to understand our solution. Section 4 presents the proposed solution FED-SQUARES and the LEAF benchmark to generate the FL simulation and model. Section 5 describes the experiments, environment, dataset chosen the FEMNIST, and the evaluation tasks performed. In section 7, present the results and discussions. Section 7 describes the conclusion of this paper and the future steps to overcome some limitations.

2 RELATED WORKS

2.1 Data Visualization Techniques

The growing interest in machine learning techniques and the recent advances in the area generate many initiatives in data visualization research to understand how these models work and evaluate which one is the best for a defined task though a fair measure and data visualization technique. Initiatives, as presented in Boxer [**?**, 4] a solution that allows a comparison between different machine learning models, analyzing different subsets of a common training and test dataset, the platform offer a large type of data visualization (line charts, parallel coordinates, data histograms, and others to compare the models) for analysis of the scenarios, given the freedom to the user composes an interface selecting the visualizations that better to our specific analysis. Another example is the ModelTracker [1] that overall model performance while enabling direct data inspection. Boxes represent user labeled examples, and color indicates the label given. The technique places the test examples at the top and train examples at the bottom. According to the model's prediction scores, examples are laid out horizontally, with low scoring examples to

*e-mail: tvms@cin.ufpe.br

the left and high scoring examples on the right side. Users can interact directly with ModelTracker to reveal additional information and inspect examples. The SQUARES [6] it is another approach to visualize and compare the performance of models, comparing results and give information at the instance level, showing the data and model statistics information, developed by the same team from the ModelTracker, but with an interface much more straightforward with a similar proposal. All the previous works evaluate their solutions with centralized ML models; in our approach, we will adapt the SQUARES solution that will be described in the Background section to evaluate the Federated Learning model.

3 Background

3.1 Federated Learning

Federated Learning is a new paradigm of machine learning, where the design pillars are data privacy and computation at the edge. Different from the traditional centralized paradigm where the data is sent from the local devices to the model owner server to be trained, in this scenario, the server sent the model itself to train in the local devices with local data that never leaves the host. After training, the local model is sent again to the server to be aggregated by specific functions as FEDAVG [5] that performs a weighted average of the model weights generating one new global model to be sent to new selected clients. The training process still occurs through these rounds/cycles of training until it achieves a stop criteria. Unlike the centralized scenario, the federated is much more heterogeneous and deals with Non-iid data distribution between the clients; it means that there is no grant that each device will have a balanced amount of data between classes. Also, the features can be unrelated to the problem domain, biased, or in some instances, the data has a style related to the origin, user, or region. in this way, it is essential to use advanced data visualization technique to interpret, debug and evaluate the federated models.

3.2 LEAF - Federated Learning Benchmark

To execute the Federated Learning simulation, we used a benchmark for simulations of federated learning model training and evaluation, called LEAF [2]; with this benchmark is possible to extract train information and test a machine learning in a federated way. The benchmark provides different federated datasets subdivided in client formats, where each client has some part of the entire dataset. In this way, the simulation occurs in only one machine, but the virtual server sent the models to the virtual clients that hold a part of the entire dataset, and the train occurs locally. The benchmark also comes with different machine learning models to be evaluated. The framework allows modifications in the entire structure to test new scenarios and techniques. The benchmark also generates data related to test/train accuracy and loss between rounds for each client and the entire framework and information about the processing consumption on each client and dataset. Some visualizations are available, but no one at the instance and score level, linking data inspection and the test and training data.

3.3 SQUARES

The original SQUARES [6] tool is divided into three main visualization areas, as can we see on Fig. 1. On the top, we can see a stacked horizontal histogram for each class of the model where the bars on the left side of each class vertical axis correspond to the false-negative proportion of samples predicted to the corresponding class. The slashed bars at the right side correspond to the false positives, and the clean colored bars the true positives. Vertically each bar position corresponds to a proportion of samples classified with the level of score according to the left axis on the screen representing the ranges of score prediction. At the top of each class, axis exists a chart line miniature with all score curves from positive cases based on the parallel coordinates. A bar with information about the tested model (accuracy, precision, recall, False Positives/Negatives, and True Positive rates and number of classes) and dataset (number of samples showed or not) composes the middle of the interface. Also, the tool permits group samples at the parallel coordinate chart, though stacks/bars (more general group), strips (samples with a certainty level of similarity) or squares (more detailed level of granularity, where each square represents a sample). At the bottom of the interface, each tested sample's information is showed at a table, with an image/feature visualization, prediction, ground truth, correctness result, and score prediction for each sample class. Examples in the table can be selected, and automatically the system plots a line at the parallel coordinate chart corresponding to the scores inferred for each class by the model on the selected samples. The system permits to select the modes stacks/bars, strips, or squares at the top chart and highlight the corresponding examples at the table that belongs to this selection, and the plotting score lines at the chart.

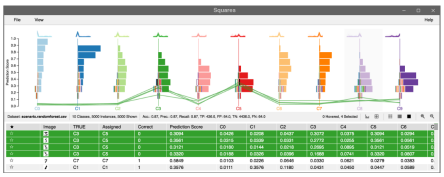

Figure 1: Original SQUARES interface, showing the results of an analysis of an FL model for a test sample example.

In the original paper, this environment was used to evaluate two different machine learning techniques, comparing the proposed visualization with the consolidated confusion matrix for some tasks tested with many subjects. Tasks are proposed to evaluate the speed of the analysis and ease of interpretation of the model's responses and comparisons. The SQUARES surpass the conventional method on the experiments with the users.

4 Proposed Solution - FED-SQUARES

As mentioned, the FED-SQUARES prototype is based on the SQUARES visualization that is a private solution. We used a set of tools to develop the available prototype, including D3 for the charts, Tabulator, for building the table and Javascript, CSS, and HTML. In this prototype, the main aspects of understanding the model at an instance level were developed. For this, we prioritize some visualizations and interactions from the original tool adapted for an FL scenario. These visualizations give us simple insights but turn possible to make a comparison between the trained models at an instance level in an interactive way, where it is possible to analyze scores prediction for each class, miss-labeled samples, corrupted samples, model bias, certainty/accuracy level of the model for each class and compare the models at overall statistics and class score inference level.

For our purpose, the SQUARES solution was reproduced, keeping the top parallel coordinate and horizontal stacked histogram chart with bar/stacks pattern to represent the samples' proportions on each score bin and respective axis class. The general pieces of information about the model as the number of class, accuracy, dataset information, and sample visualization quickly help analyze the scenario and is localized at the top of the interface. The parallel coordinate chart and the stacked histogram bars representing the False Positive/Negatives and true positives proportions and scores of the overall tested samples are in the middle section of the interface, and at the bottom a is showed a table with the classification and data information of each sample recovered by each client and used for the test. In this table, some formations are related only to a federated scenario. All the fields presented on him are described below:

- **Client:** The name of client where the sample come from

- **Round:** The number of train rounds executed before test the sample (optional)
- **Hierarchy:** Parameter related to the simulation tool used LEAF (optional)
- **Num_samples:** Number of samples from the client used for test.
- **Set:** Name of the set used for evaluation from the client. Can be: (test, train, eval)
- **Features:** Path to the raw feature used for test on the simulation, in this case, a path for the image.
- **Accuracy:** Overall accuracy for the evaluated samples from the specific client that the sample belongs.
- **Loss:** Overall test loss for the evaluated samples from the specific client that the sample belongs.
- **Ground_truth:** The ground truth/correct classification label for the specific sample.
- **Prediction:** Class predicted by the federated model for the specific sample.
- **Cn:** Prediction score attributed by the federated model for the specific sample to class n. (n can be a number at 0 and the max number of classes supported by the model).

It is possible to sort the rows in the table by each column, selecting the column header. The row can be selected, and a line with a color correspondent to the predicted class is plotted at the parallel coordinate chart representing the scores attributed to the sample by the model on each class, which turn possible evaluate if there is some confusion of the model in this specific example. Also, the image/feature tested is showed for inspection on the left top corner of the screen, which turns possible a visual inspection of the sample and an improved analysis. The chart turns possible to compare the different models at a score and instance level and the same model trained in different steps on time.

5 Experiments

To evaluate our proposed solution, we generated the data to analyze with the LEAF benchmark using a Convolution neural network (CNN) that's already implemented in the tool. The benchmark does not save information about the output layer as the score given for each sample tested, so we modified the benchmark to save this information is in a CSV file together with the other information's presented in the previous section in the table description; the CSV is also converted for JSON to be loaded by the table component. The benchmark comes with pre-established scenarios to test, including the number of total clients and dataset formats. It is possible to define the portion of clients to be used on the training but not the number of classes, so we develop the filter of classes and change the output layer size.

The chosen dataset selected for the simulation was the FEMNIST image set of handwritten digits and characters manuscripts prepared for federated tests and based on EMNIST [3]. This dataset contains a total of 80,5263 of samples distributed between 3,550 users. The images are labeled in 62 classes, divided into 10 numeric digits, 26 upper case letters, and 26 lower case letters. For the experiments, the dataset was filtered, as mentioned. It was selected only 10% of the total clients, filtering these clients only to have 10 classes, the numeric digits for fast evaluation of our solution. 350 clients compose the filtered dataset used in our experiments, and the test set is composed of 4056 images that correspond to the sum of 10% of each local dataset that belongs to the selected clients. The datasets between the clients are unbalanced, and some do not have examples of all classes. The experiments were executed on an HP Z840 workstation with a processor Intel Xeon E5-2609 @ 1.70 GHz x 16, 32 GB of RAM, and an NVIDIA RTX 2080 ti with 11GB.

We made four train executions with the standard LEAF CNN with the output layer modified for ten classes. The learning rate used was 0.001, batch size 10, and one local epoch for all executions. In the first execution, the model was trained by 500 rounds, the second 1000 rounds, and the third by 3000 rounds. The model was trained, selecting 2 clients per round with the same random seed to train reproducibility and continuity in all scenarios. With these scenarios, we can analyze the model's evolution through the rounds, observing the confusion between classes and the certainty related to class prediction scores. In the fourth execution, the second experiment was executed again with 1000 rounds but with a different random seed, which affects clients selected for the train. Since the dataset between clients is unbalanced, a different training order will generate a different model to compare with the first generated by the second execution.

The results expected is a growing on the accuracy of the model with more rounds of training, turning possible analyze, outliers, and miss-labeled examples on a precise model, and also with the fourth execution perceive the difference of the models with a similar accuracy, which is not possible with a confusion matrix or experimented in a centralized environment.

6 Results

It is possible to observe the results of the three first executions increasing the number of rounds on Fig. 2 where the interface of FED-SQUARES is presented, and the expected scenario happens, the training with 500 rounds achieves 48% of accuracy presenting in the chart a low level of certainty in the prediction with a maximum score of 0.3 for some samples with a high rate of false positives represented by the grey bars at the right side of the axis and false negatives at the left side, and small colored bars that represent true positives at the right. Without a direct visualization of the scores, the model can be considered by some tasks reasonable since made 41% of accuracy, but looking at the score prediction is much easier to perceive the low level of the model quality.

Figure 2: Visualizations with FED-SQUARES. **Top:** model trained with 500 rounds. **Middle:** model trained with 1000 rounds **Bottom:** model trained with 3000 rounds

The execution with 1000 rounds the overall accuracy increases to 79%, and the distribution of the scores prediction increases, achieving answers near 100% of certainty. This model, like the first-mentioned model, with this level of accuracy, is possible that an engineer or customer be satisfied by the model only looking for the overall accuracy or confusion matrix. However, we see a considerable portion of false negatives in class 8 and false positives in class five, looking at the scores. If these classes are critical to the task, it will be a risk to use this model. It can be argued that with a confusion matrix, this scenario can be analyzed, but image a model with this level of accuracy. However, in this case, low scores for each prediction, the FED-SQUARE visualization, can indicate a problem on the model in calculating the scores or a highly intra-class similarity. Without the analysis of the output scores, these cases are not perceived.

In the third execution with 3000 rounds, the model's accuracy increases to 91%. At this level, the rate of false positives and negatives is low, as can be observed. Also, each answer's certainty level is very high, the majority near to 100%. The model can be useful, and the wrong cases can be analyzed to verify if incorrect predictions or outliers and miss labeled examples are existents since these are common cases found in large datasets labeled by humans.

To analyze the outliers or mislabeled examples in the previous scenario the FED-SQUARES visualization allows sort the samples by each column in the table. Sorting by the ground truth is much mo easy to search for wrong predictions and verify trough the image visualization if the image is corrupted, miss labeled, or in fact, a wrong prediction. As shown on Fig. 3, where the left-column example, a correct prediction is made, the digit is legible and showed at the left corner, and the line at the chart point to a high level of certainty. In the right-column case, a wrong prediction is shown. The image labeled as 0 was classified as 1; inspecting this sample noticeable the character's deformation that confuses the model. The thin aspect of the digit increases the score of class 1 causing confusion with the correct class. Examples like this can be explored and found quickly with this interface.

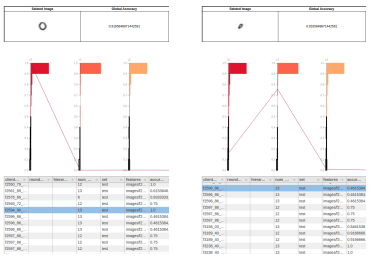

Figure 3: Inspection of samples predictions with FED-SQUARES. **Left-column:** correct prediction on class 0. **Right-column:** Wrong prediction on class 0 due to the image corruption and deformed digit

On Fig. 4, we can see the different scenarios of score prediction distribution to each class, comparing the two models trained with the same parameters but selecting clients in different orders. They both have similar overall accuracy, the first in the top with 79% of accuracy, and the second with 78%. Using FED-SQUARES is easily perceive the difference between the two models supported by the stacked parallel chart. Is easy to perceive that the first model has a considerable rate of false negatives in class 9, and the second model presents an overall reduction in score in the majority of the class except on the class 7 and has less false negatives in class 8 that the first model. Perhaps the accuracy models are very similar the models are very different, compared with a confusion matrix, this analysis is much more intuitive and visual, and don't there is no strict need to analyze each bin of number because the visualization turns easy to found the patterns.

Figure 4: Models with similar accuracy but very different according to FED-SQUARES. **TOP:** Model trained with 1000 rounds. **Bottom:** Model trained with the same configuration and rounds number, but with a different random seed.

7 Conclusions and Future Works

As shown in the original paper [6], the foundations of SQUARES are adequate to analyze centered learning scenarios. However, we evaluate the technique's main ideas and foundations with our experiments, building a prototype, and applying it in a federated model. The results are very similar in terms of the interpretability of the final model generated. They can contribute to understanding federated learning environments and the final model results as expected, improving the monitoring of server and client test accuracy and prediction score at the overall model and instance level. It was quickly possible to find deformed examples and similar models according to accuracy but very different at the score prediction and instance level. We did not make a subjective evaluation to certify the results, but some examples are shown. For future works will be interesting to associate in formations of the training dataset from the clients and use the tool to follow each client's impact on the overall model; also, a subjective evaluation of the model presented on the SQUARES paper is important to definitively validate our solution.

References

[1] S. Amershi, M. Chickering, S. Drucker, B. Lee, P. Simard, and J. Suh. Modeltracker: Redesigning performance analysis tools for machine learning. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI 2015)*. ACM - Association for Computing Machinery, April 2015.

[2] S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konečný, H. B. McMahan, V. Smith, and A. Talwalkar. Leaf: A benchmark for federated settings, 2019.

[3] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik. Emnist: an extension of mnist to handwritten letters, 2017.

[4] M. Gleicher, A. Barve, X. Yu, and F. Heimerl. Boxer: Interactive comparison of classifier results, 2020.

[5] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data, 2017.

[6] D. Ren, S. Amershi, B. Lee, J. Suh, and J. D. Williams. Squares: Supporting interactive performance analysis for multiclass classifiers. vol. 23, pp. 61–70, 2017.