**Federated Learning with Deep Neural Networks for Text**

DATA 590 Project Proposal

University of Washington, Autumn 2019

Arjun Singh

Joel Stremmel

**Executive Summary / Introduction**

While machine learning on large datasets is a dominant paradigm, there are a number of drawbacks to centrally aggregating data, the most significant one being privacy. Federated Learning aims to address this and has shown promise for [text completion tasks on mobile devices.](https://arxiv.org/pdf/1811.03604.pdf) The [Tensorflow Federated API](https://github.com/tensorflow/federated) provides methods to train Federated models and conduct Federated Learning experiments on data grouped by clients but never aggregated. Through our research partnership with Google, we aim to build on the existing body of Federated Learning experiments with a particular focus on enhancing text models for Natural Language Understanding tasks, such as Next Word Prediction, which falls in the realm of auto-regression and Next Sentence Prediction, which can be categorized in the realm of auto-encoding.

**Problem Statement**

Federated Learning aims to train machine learning models in a distributed fashion without centralizing data but instead updating and passing model parameters from a central server to distributed entities and back to perform stochastic gradient descent. McMahan et al. propose the Federated Averaging algorithm in [Communication-efficient learning of deep networks from decentralized data.](https://arxiv.org/abs/1602.05629) This algorithm and associated experiments in the paper yield promising results. However, they are currently trained on RNNs and LSTMs, without making use of the extremely powerful concept of [Attention in NLP](https://arxiv.org/pdf/1706.03762.pdf).

Our goal is to replicate the existing network architectures for Federated Averaging, stress testing their limits within our simulated environment in terms of compute, memory and power resources. We subsequently want to apply Attention based models for the currently established tasks, such as Next Word Prediction, and if successful in our pursuits, introduce the powerful Attention based model, BERT, and/or its pruned versions, into the realm of Federated Learning.

**Proposed Solutions**

We have three proposed solutions for the problem posed. Although we will aim to deliver all three, having more than one allows us to be more flexible with our target, depending on how long it takes to familiarize ourselves with NLP and FL concepts, the current TFF codebase, and the required duration of training and experimentation with various models:

1. Replicate the results from the current LSTM model published as part of the [GBoard paper](https://arxiv.org/pdf/1811.03604.pdf), that makes use of Federated Averaging for training a model for the task of Next Word prediction. The major goal is to assess what parts of this 1.4 million parameter model could be replaced or omitted, while trying to diminish the complexity/size of the network, without compromising on metrics like accuracy.
2. As an immediate next step in the enhancement of the aforementioned model, we aim to leverage Transformer models with Federated Averaging, for the task of Next Word prediction. These models are based on the paper, [“Attention Is All You Need”.](https://arxiv.org/pdf/1706.03762.pdf) Transformers, unlike many models based on Attention, can be Auto-Regressive and can be used for language modeling, as in NWP tasks. A [PyTorch implementation](https://github.com/pytorch/examples/blob/master/word_language_model/model.py) shared on [their website](https://pytorch.org/docs/stable/nn.html#torch.nn.Transformer) shall be used for reference.
3. Finally, if the above experiments are successful and if we have enough time, we’d like to assess the usage of pre-trained text deep neural networks such as [BERT](https://arxiv.org/abs/1810.04805) for “Federated Fine-Tuning”. This is a concept that hasn’t yet been explored, to the best of our understanding. As interesting and novel as this idea is, it is also equally hard to materialize due to the power/memory/compute constraints of personal mobile devices, the predominant target of Federated Learning, at the time of this writing. Well aware of these limitations, we also aim to explore light versions of BERT, such as [ALBERT](https://arxiv.org/pdf/1909.11942.pdf).

**Risks**

Major risk with BERT/ALBERT models: the task for which we’re training the first two of the three models listed above is Next Word Prediction (Language Modeling), which is best addressed by auto-regressive models. On the other hand, BERT and the light versions that succeeded it, are all auto-encoding based models. This implies that we will need to amend our task from NWP to something more along the lines of NSP (Next Sentence Prediction), a task on which BERT-type models have held the SOTA (state-of-the-art) scores for a long period of time since their advent.

**Schedule**

In the two and a half months, that we have to complete our project, we aim to tackle our three proposed solutions in order. Tentatively we aim to do the following:

January:

* Familiarize ourselves with the Stack Overflow data and the TFF dataset API for model training
* Replicate LSTM results from the GBoard paper using the Federated Averaging algorithm and work to reduce size and complexity of the LSTM without compromising much on accuracy

February:

* Identify main contributions from reducing LSTM size for Federated Averaging and finalize experiments with these methods
* Create a Transformer model with Federated Averaging
* Assess viability of BERT and ALBERT for federated fine tuning

March:

* Compile experimental results
* Create project deliverables
* March 11, 2020 is our final capstone presentation