Brainstorming - Carbon Efficient Federated Learning on Large language models as workload

Goal:

Reduce carbon emissions during the training of large language models by leveraging carbon intensity variations in a federated learning system.

Federated learning offers several advantages that align with our objectives:

1. **Distributed Nature:** The decentralized approach enhances scalability and provides an opportunity to optimize cost by exploiting the cost variations across multiple locations.
2. **Limited Communication:** Reduces the need for extensive data transfer, thus minimizing network overhead and preserving bandwidth.
3. **Asynchronous Learning:** Enables clients to update the model at different times, providing flexibility in the training process.

We divide the problem statement into 4 major parts:

- Develop an oracle (from our model) that accurately predicts carbon intensity for any given time of day.
- Create a reliable prediction model for the duration of a training round. - average training times
- Formulate a model that accurately estimates the probability of completing a training round within the predicted time frame while keeping the carbon emissions in a favorable range.

Based on the probability outcome from the model, we can perform the following operations:

a) In case of high probability: we train the clients in the FL round given the probability of completing that round is high (higher than 60%) with carbon emissions being favorable.
b) Uncertain (probability closer to 50%): In this case, we're not certain if we will be able to complete the particular round with the low carbon emissions thus we reduce the overall round time by maybe reducing the batch size etc.
c) In case of low probability: we pause the training or move the clients to the places where the carbon intensities are low.

Process:

Round 0: The initial round selects a client randomly from all the available clients. The randomly selected client trains and sends its model parameters to the server. The server now has an initial set of parameters, which it sends to all the other clients.

Round 1: The client manager on the server selects “n” clients from the N available clients. The selection of clients is based on the carbon intensity values, which represent the carbon emissions of the clients for training in that specific round. These carbon intensity values vary across different locations and times of the day, week, month, and year. Our goal is to reduce carbon emissions in each round. To achieve this, we want to calculate the probability of completing a training round with favorable carbon intensity values.

To do this, we need to calculate the carbon intensity at a given time of day based on weather forecasts, energy sources, and time of day. Additionally, we need to know the training time for that round, which we predict using LSTMs and historical data. Thus, to determine the overall carbon emission in each round, we use the following formula:

Total Carbon Emissions in a Round=Training Time × Carbon Intensity

The probability distribution function should look like this:

$P(T \leq t, C \leq c) = FT(t) \times FC(c)$

where FT(t) is the CDF of the training time and FC(c) is the CDF of the carbon intensity.

Where:

- $P(T \leq t, C \leq c)$ represents the probability that the training round can be completed within a certain time t and with a carbon intensity lower than a certain value c.
- T is the random variable representing the training time.
- C is the random variable representing the carbon intensity.

Carbon forecasting for the next round of training:

We can use CarbonCast (https://dl.acm.org/doi/pdf/10.1145/3563357.3564079) to do the carbon forecasting for the next round of the FL training. We can use the equation method which requires the first tier electricity forecasts from the grid and the carbon emission factor for various sources of the electricity. The average carbon intensity per unit of electricity generated in a region is the weighted average of carbon emitted by each source due to electricity generated by them. Mathematically, the average carbon intensity (in $g/kW\,\hbar$) of a region at any time is as follows:

$$(Carbon\ Intensity)_{avg} = \frac{\Sigma(E_i \times CEF_i)}{\Sigma E_i} \quad -(1)$$

where E_i is the electricity generated (MW) by a source i and CEF_i is the carbon emission factor (g/kWh) of that source.

To predict the carbon forecast for the next day, we can run inference on the CarbonCast model at 00:00:00 each day by providing electricity generation data from the previous day and weather forecasts for the current day. Using this information, along with the carbon emission factors from different sources of energy, CarbonCast can predict the carbon emissions for the current day. These carbon emission forecasts can then be used to accurately estimate the carbon emissions for each hour of the day. When mapped to the FL round, we can accurately predict the carbon emissions for a federated learning round.

Following are the sources which provide information about the electricity generation and weather forecasts.

- US ISO electricity generation by source: EIA hourly grid monitor
- European regions electricity generation by source: ENTSOE
- Australian regions electricity generation by source: OpenNEM
- Weather forecasts: GFS weather forecast archive
- Day-ahead solar/wind Forecasts:
- CISO: OASIS
- European regions: ENTSOE
- Carbon emissions factors: Carbon emissions factors

Our goal is to predict the carbon emissions for the next round of FL training. We can use existing techniques to forecast carbon emissions for the next day and use these forecasts to predict the carbon emissions for the next round of training. To forecast the carbon intensities, we can use the analytical equation which utilizes the electricity generation forecast and carbon emission factors. The carbon emission factors are standard, and we can obtain them from this source: Carbon emissions factors. For the electricity generation forecast, we will train a data-driven model on the weather forecast data and historical electricity generation data. I have acquired the historical electricity generation data, and the next step is to obtain the weather forecasts. The breakdown for the tasks is as follows:

1. **Obtain the weather forecasts from the GFS weather forecast archive** (I'm working on it right now)
    - Alternatively, we can use historical data to forecast weather using a data-driven model or use the weather forecasts available for the next 50 years.

2. **Train a data-driven model to generate the electricity generation forecasts.**
    - Inputs to the data-driven model:
        1. Weather forecasts (or historical data) (Step 1) - next task
        2. Historical data for electricity generation (we have obtained this data from the EIA hourly grid monitor)
3. **Use the analytical equation to predict carbon emissions based on energy data and carbon emission factors.**