Federated Learning

Prithvi Kannan

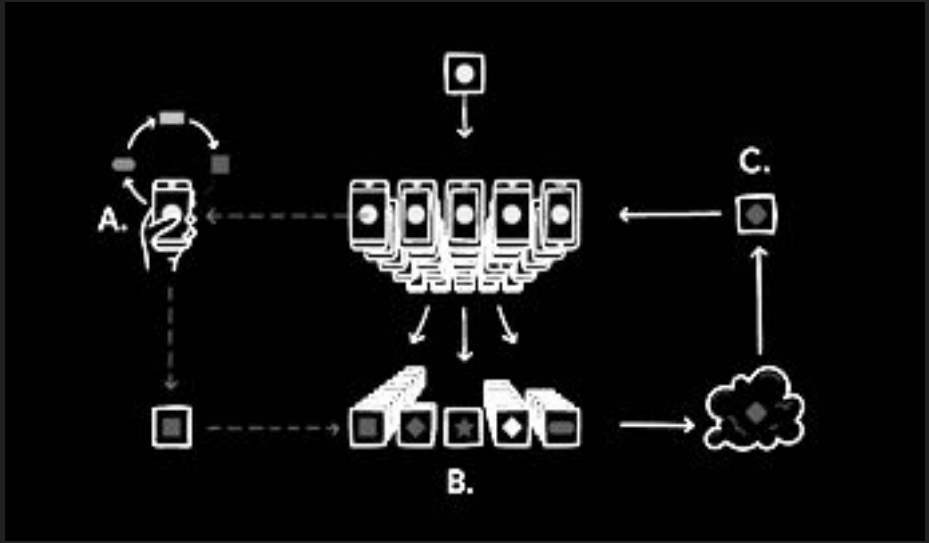
Problems With Traditional Machine Learning Today

- Data centers are vulnerable to breaches and failures
- Centralized models cannot be personalized to the user

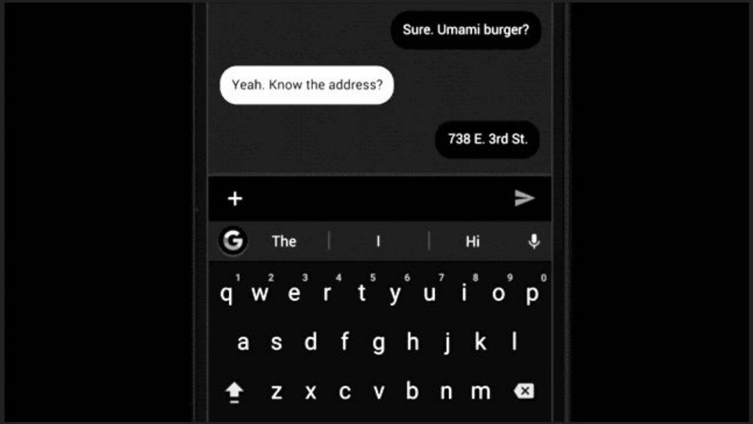
Federated Learning is *decentralized* and *does not store data*

How Federated Learning works

1. Downloads the master model
2. Trains on local data as used
3. Package encrypted changes and send to cloud
4. Update master model



Case Study: Google GBoard



Sure. Umami burger?

Yeah. Know the address?

738 E. 3rd St.

- GBoard is Google's keyboard that offers relevant suggestions
- User's don't want their keystrokes to be recorded by Google, but still want personalized results
- GBoard downloads the model, trains on user data locally, and then sends updates back to the cloud once the phone is idle
- Gives user more accurate models and doesn't compromise security

Applications to Ads

- Advertisers want to show the most relevant ads, but users don’t want their browsing habits to be stored by Facebook, Google, etc.
- Federated learning allows users’ data to never leave their device but still train models

Applications to Healthcare

- Drug research requires lots of data
- Patient data is the most sensitive and needs to be confidential
- Using federated learning, a network of hospitals can help train researcher's models without needing to centralize and upload patient data

Applications to Self Driving Cars

- Cars produce gigabytes of video and sensor data per minute but data storage and streaming can be expensive
- Autonomous algorithms require tons of data to learn
- Federated learning limits volume of data transfer/storage and accelerates training with onboard calculations