Federated Machine Learning

A research project

Karan Samani
1161081087
BSc. Computer Science (Hons.)
February 17, 2020

Abstract

Federated machine learning is the idea (from Google) of anonymised machine learning (or rather deep learning). It is a way to get a Neural Network trained on everyones data, but without having direct access to everyones data.

Traditionally, a Neural Network would require a lot of data from users to train a model that is fairly accurate. But with the federated approach, the users dont have to share their data with anyone else to obtain a better overall model. Instead, they train a model locally no their own data, and then send the weights and biases of the model (the original user data cannot be recreated with these weights and biases) to a server which then averages them and sends them to you all the users. Because the weight and biases are being sent, instead of the users actual data (like their images), privacy is maintained and essentially a model is trained using anonymised data from several users.

My project is based on implementing the way in which Google does this, and then implementing several more strategies proposed by Derek and comparing their outcome. At a high level, these strategies include discarding the weights of users in certain conditions or using a weighted average of their weights and biases.

Declaration

Acknowledgements

Thanks to Dr. Derek Bridge for dealing with me over the last few months, pushing me to my limits and giving support as and when required.

Thanks to the Alexander Baran-Harper's channel on YouTube where I learnt how to use use LaTeX.

Contents

List of Figures

1 Artificial Intelligence (dont know where to put this chapter

1.1 What is AI?

Artificial Intelligence can be described as the ability for a system to show "intelligence". Intelligence, as Dr. Derek Bridge put it, is the ability for a system to act autonomously and rationally when faced with disorder, uncertainty, imprecision and intractability. A subset of AI is called Machine Learning. ML is a broad field of AI application where explicit programming is not required for the system to recognise patterns and learn. The area of Neural Networks (and Deep Learning) is the one that will be explored in this project.

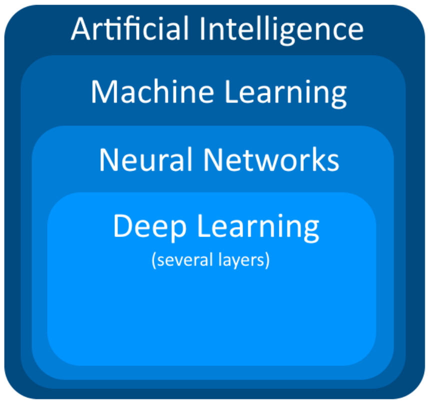

Figure 1: High level visual representation of what fields there are in the study of AI.

1.2 Neural Networks

The idea of neural networks is not new. In fact, it has been around for decades, but it only really took off in recent years with the emergence of better and more affordable hardware. The basic idea behind the workings of a neural network are quite straight forward. To start off, I must explain what the basic block for a neural network is, a perceptron.

A perceptron can be thought of as a node that takes in a weighted sum of its inputs, runs the value through a function (called the activation function) and outputs a value to be used later. The activation function below is a step function that outputs a 1 if the weighted sum is more than a certain value. The input labelled 1 in Figure 2 is used to represent a bias b that is always added to the weighted sum before inputting the value into the activation function. Note that the weighted sum can be written in a vectorised format.

$$w \cdot x \equiv \sum_j w_j x_j$$

$$\text{output} = \begin{cases} 0 & \text{if } w \cdot x \leq 0 \\ 1 & \text{if } w \cdot x > 0 \end{cases} \tag{1}$$

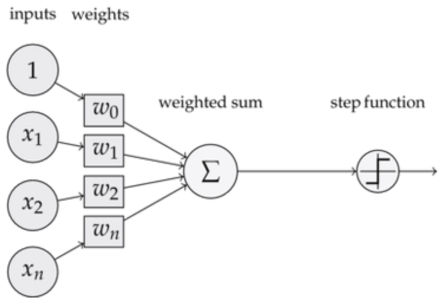

Figure 2: Major components of a perceptron.
(https://blog.knoldus.com/introduction-to-perceptron-neural-network/)

The activation function can be swapped out from the step function that was being used earlier to something else, for instance ReLU (what we will be using) which takes the max of either 0 or the weighted sum.

$$\text{output} \quad = \quad \max\{0, w \cdot x + b\} \tag{2}$$

In a neural network, there are layers of such perceptrons. Each layer connected to another in a different way. The basic stages are to feed data through to make predictions, compare them with the actual predictions and then use an algorithm called back propagation to update the values of the weights for every perceptron (referred to as nodes from now on). More information on this and the idea of neural networks can be found in the (online) book "Neural Networks and Deep Learning, Michael A. Nielsen"[1] and the book "Deep Learning with Python, Francois Chollet"[2]

1.2.1 Dense Layers

The way the nodes are connected to each other is called the architecture of the network. Densely connected layers of nodes can be seen in Figure 3. In these, all the nodes are connected to every node in the subsequent layer.

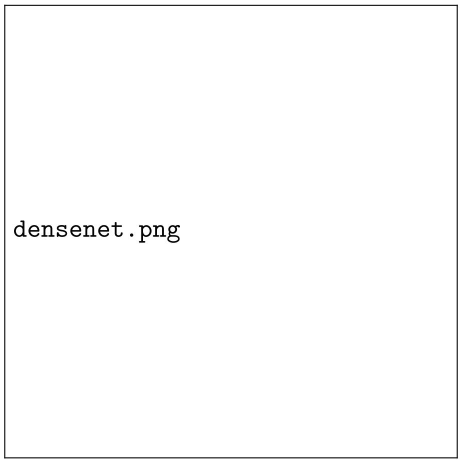

Figure 3: Basic neural network architecture[3].

1.2.2 Convolutional Layers

These are more complicated than the previously mentioned dense layers. Without going into too much unnecessary details, the basic idea is that the layers in this architecture try to find patterns in the input data and then output the presence of them.

A convolutional layer has a kernel or a window that is used to look over the input data and recognise patterns localised in that window. Every layer has a "depth" number of sub-layers which represent the patterns that the layer is trying to learn and the subsequent layers are connected to these. For instance, with the depth being 3, the convolutional layer looks for 3 patterns in that layer. Figure 4 does a good job of giving an idea about how this process works. But the main concept remains the same. There are underlying weights that are being tweaked to learn patterns.

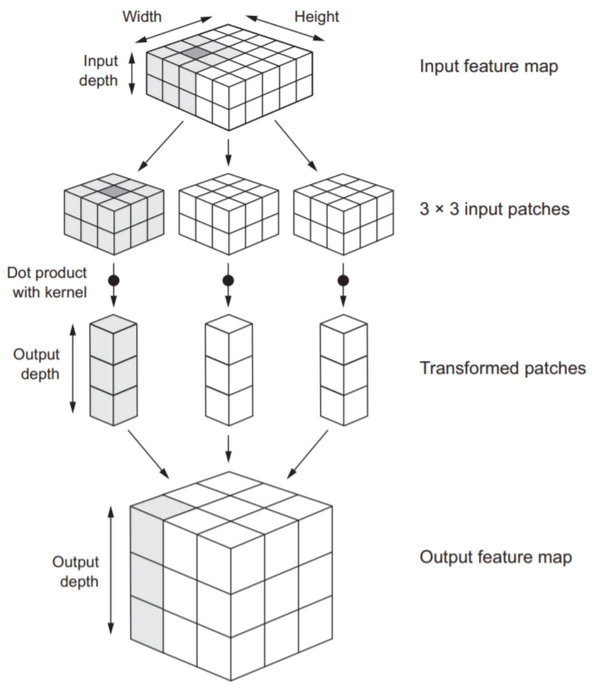

Figure 4: How a convolution works[2, (Page 125, Chapter 5)].

At the end of a series of convolutional layers, there is a series of dense layers that are used to give the predicted output(s).

2 Background

2.1 Introduction

Modern day edge devices have a wealth of data on them and have more than enough computation power to run complex calculations on them with ease. These devices can range from a personal computer to a smart phone. In a world where data is power, access to the data on these devices is highly advantageous.

Machine Learning, a branch of Artificial Intelligence (AI), is based on the idea of creating a model that recognises patterns from data to be able to solve prediction problems. To be able to do so effectively, one must have access to a lot of data. More data equates to a higher probability of having a more robust and better overall model. If a model can look and learn from more data, the chances are that it can generalise well, and that is the ideal goal.

The solution to having well trained models seems simple now. Just use the vast amounts data from the edge devices to train a model, correct? To do so, the users must give their data to a server so that the server can then train the model on the data that was just provided to it by the edge users. This trained model can then be used by everyone to predict unseen samples. But what if the users don't want to share their data but still want the benefits of having a model trained on everyone's data?

2.2 Motivation

There are numerous instances where people may not want to share their personal data. For instance, for the training of a model that deals with predictive text, the input data would require essentially everything that a user may type into their device. It is pretty obvious to see why some people may not want to share the messages and other content that they type on their devices. It is a clear invasion of their privacy.

Another application could be training on images for classification purposes. People may not want to share images, which may include sensitive images, that they have stored on their devices with a third party. This can be extended to an even more sensitive topic of medical imaging where people may not want to share something like the X-Rays of their bodies.

In general, people are sceptical of sharing personal data. But we still want to train a model that has had exposure to as much data as possible. Is there a compromise?

2.3 Federated Machine Learning

This was the motivation behind the idea of Federated Machine Learning. The idea of privacy, the idea of not having to share your data with someone else and yet still have the benefits of having a model that has exposure to their data.

Traditional ML where everyone sends their data to a server, a central user, and the model is trained on everyone's data on the server. In Federated ML, an identical copy of the model is sent to every participating edge user. (Section 1)

3 Literature Review

How to train Google doing it Privacy Federated Runing on edge too Differential

3.1 What has been done, not mine but others

3.2 Google doing it

3.3 Privary

3.4 Differntial privacy

4 How to train

5 running on edge

6 Core Design

6.1 building an desinging this

6.2 implemenatation

6.3 Global

6.4 google implementation

6.5 my implementation

7 exnteded ideas

7.1 Central Server

7.1.1 Weighted Average

7.1.2 Excluding based on std dev

7.1.3 Local only

7.2 Peer to Peer

7.2.1 Weighted Average

7.2.2 Excluding based on std dev

7.2.3 Local only

7.2.4 testing

8 Datasets used

9 testing other weights on global data

10 tensorflow

11 modily

h5 files

12 each dataset output experimentation

artificial users

For each dataset show the otuput we got

Bibliography

[1] M. A. Nielsen, "Neural networks and deep learning," 2015.

[2] F. Chollet, *Deep Learning with Python*, 1st. USA: Manning Publications Co., 2017, ISBN: 1617294438.

[3] F. Bre, J. M. Gimenez, and V. D. Fachinotti, "Prediction of wind pressure coefficients on building surfaces using artificial neural networks," *Energy and Buildings*, vol. 158, pp. 1429–1441, 2018.