CSE517 Optimization for Machine Learning Project

Hardik Garg, 2019040 | Tathagat Pal, 2019211

Problem Statement: To implement at least one of the optimizers (Adam's Optimizer) in federated setting using PySyft and PyTorch.

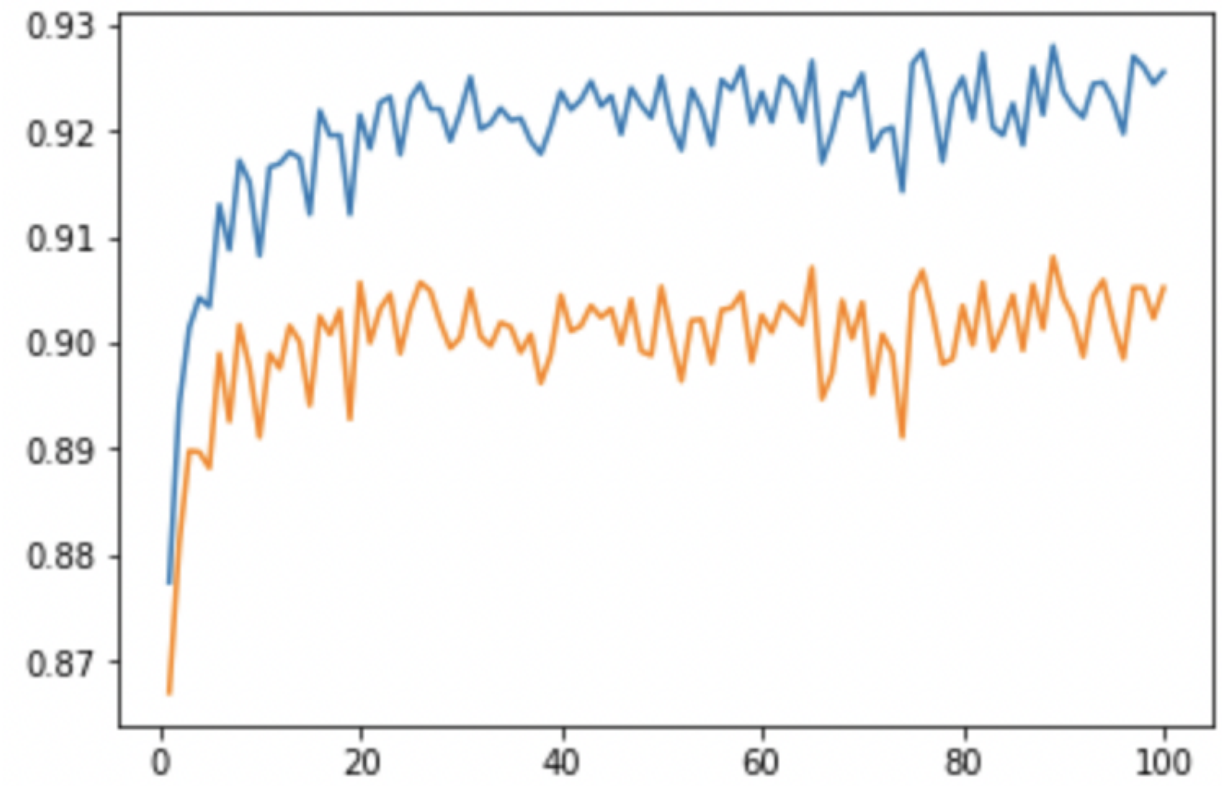
Hyperparameter Tuning:

In [3]:
```
#Class to define FL parameters
class Arguments():
    def __init__(self):
        self.images = 60000
        self.clients = 5
        self.rounds = 100
        self.epochs = 5
        self.local_batches = 128
        self.lr = 0.01
        self.C = 0.9
        self.drop_rate = 0.1
        self.torch_seed = 0
        self.log_interval = 100
        self.iid = 'iid'
        self.split_size = int(self.images / self.clients)
        self.samples = self.split_size / self.images
        self.use_cuda = False
        self.save_model = False

args = Arguments()

use_cuda = args.use_cuda and torch.cuda.is_available()
device = torch.device("cpu")
kwargs = {'num_workers': 1, 'pin_memory': True} if use_cuda else {}
```

We used the Fashion MNIST dataset. The batch size was 128 and there were 100 rounds (5 epochs within each round).

We selected 5 clients among which the data was divided into for local training.
Learning rate = 0.01
C = 0.9 (C*100 = the % of clients to be selected for training)
We have shown the results for only iid data, but have also implemented non-iid. If we write 'noniid' in place of 'iid', the model will be trained for non-iid data.

We implemented the FedAvg algorithm (using PySyft and PyTorch) for Adam's Optimizers on IID dataset. The metric used for evaluation is Accuracy. In the **Accuracy Plot**, the blue line depicts the **training accuracy**, whereas the orange line shows the testing accuracy. Also, it is clearly evident from the plot that the model is converging after around 30 rounds with an accuracy of ~92%. The x-axis and the y-axis represent the number of rounds and accuracy/100 respectively.



Note: We have followed a course on Federated Learning available on Udemy.