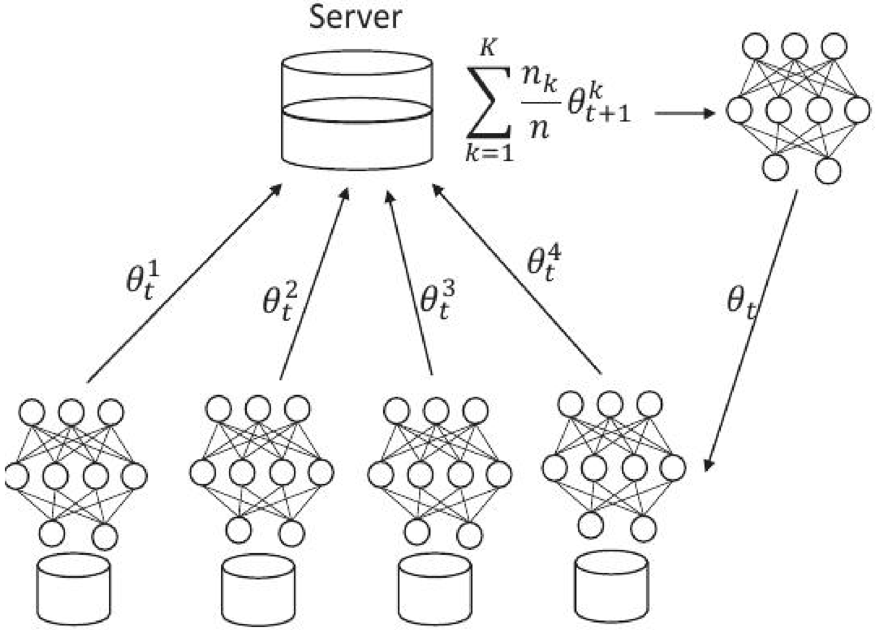
Multi-objective Evolutionary Federated Learning

https://arxiv.org/abs/1812.07478

胡超杰

2019-3-1

Federated learning



Algorithm 1 FederatedAveraging. K indicates the total numbers of clients; B is size of mini batch, E is equal to training iterations and η is the learning rate

Server:
Initialize θ_t
for each communication round $t = 1, 2, \ldots$ **do**
  Select $m = C \times K$ clients, $C \in (0, 1)$ clients
  Download θ_t to each client k
  for each client $k \in m$ **do**
    Wait **Client** k for synchronization
    $\theta_t = \sum_{k=1}^{m} \frac{n_k}{n} \theta^k$
  end for
end for
Client k**:**
$\theta^k = \theta_t$
for each iteration from 1 to E **do**
  for batch $b \in B$ **do**
    $\theta^k = \theta^k - \eta \nabla L_k(\theta^k, b)$
  end for
end for
return θ^k to server

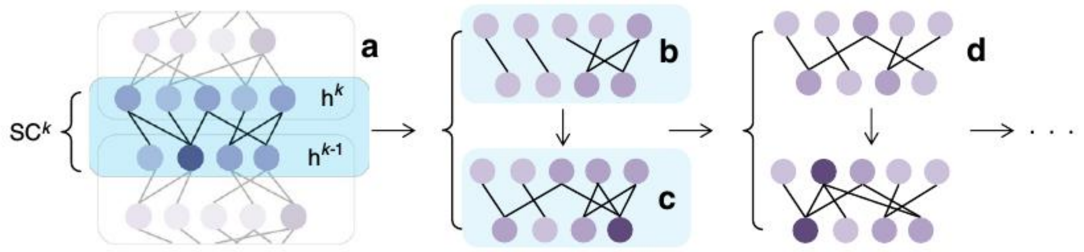
Multi-objective Evolutionary Federated Learning

- Multi-objective

(1)minimize the communication costs

(2)minimize the global model test errors

非结构化剪枝：Sparse evolutionary training(SET算法)



Algorithm 1: SET pseudocode

```
%Initialization;
initialize ANN model;
set ε and ζ;
for each bipartite fully-connected (FC) layer of the ANN do
    replace FC with a Sparse Connected (SC) layer having a Erdős-Rényi topology given by ε and Eq.1;
end
initialize training algorithm parameters;
%Training;
for each training epoch e do
    perform standard training procedure;
    perform weights update;
    for each bipartite SC layer of the ANN do
        remove a fraction ζ of the smallest positive weights;
        remove a fraction ζ of the largest negative weights;
        if e is not the last training epoch then
            add randomly new weights (connections) in the same amount as the ones removed previously;
        end
    end
end
```

全连接层首先使用ER随机图进行初始化

A:初始化后的连接
B:一个epoch后删掉一部分比较小的权重
C:随机初始化添加新的连接，保持总连接数不变
……
如果是最后一个epoch，则不添加新的连接

ER随机图构造算法思路：

（1）初始化：给定N个节点以及连边概率p~[0,1]

（2）随机连边：

1.选择一对没有边相连的不同的节点。

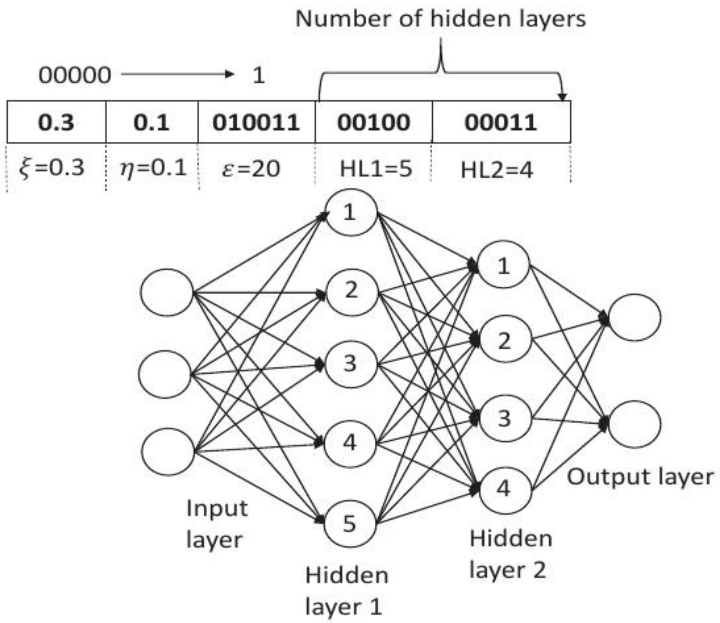
2.生成一个随机数 r~（0,1）。

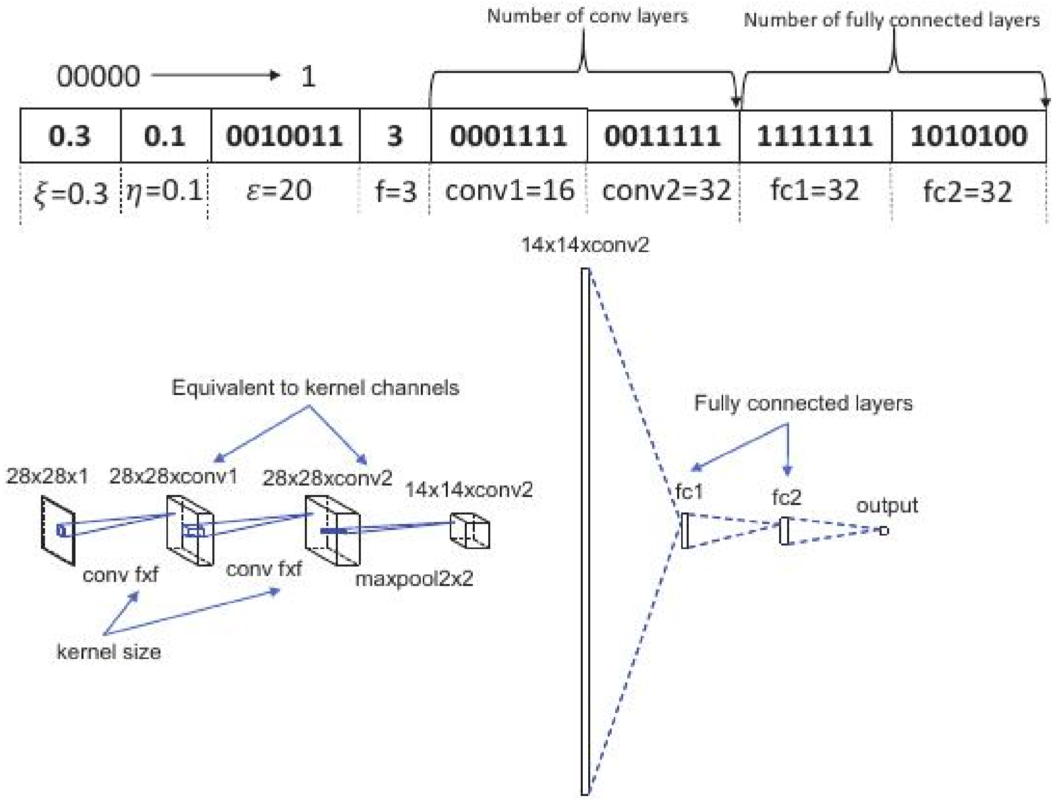
3.如果r < p，那么在这对节点之间添加一条边，否则就不添加。

4.重复1,2,3，直到所有的节点对都被选择。

结构化剪枝

神经网络的染色体表达





NSGA-II（基于遗传算法的多目标优化算法）

Algorithm 4 The modified SET FedAvg optimization. K indicates the total numbers of clients, k represents the k-th local client, B is the local mini-batch size, E is the number of local training iterations, η is the learning rate, Ω represents the number of connections, ε and ξ are both SET parameters introduced in **Algorithm 2**

for each population $i \in R$ **do**
  Globally initialize θ_t^i with a Erdos Rnyi topology given by ε and equation (5)
  for each communication round $t = 1, 2, \ldots$ **do**
    Select $m = C \times K$ clients, $C \in (0, 1)$ clients
    $\Omega_t = 0$
    for each client $k \in m$ **do**
      for each local epoch e from 1 to E **do**
        for batch $b \in B$ **do**
          $\theta_e^k = \theta_t^i - \eta \nabla \ell(\theta_t^i; b)$
        end for
        remove a fraction of ξ smallest values in θ^k
      end for
      $\theta_{t+1}^i = \theta_t^i + \frac{n_k}{n}\theta^k$
      $\Omega^k = f(\theta^k)$ (calculate the number of weight parameters)
      $\Omega_t = \Omega_t + \frac{n_k}{n}\Omega^k$
    end for
  end for
  Evaluate test accuracy through θ^i and test dataset
  Calculate test error as objective one f_i^1
  Set Ω_t as objective two f_i^2
end for
return f^1 and f^2

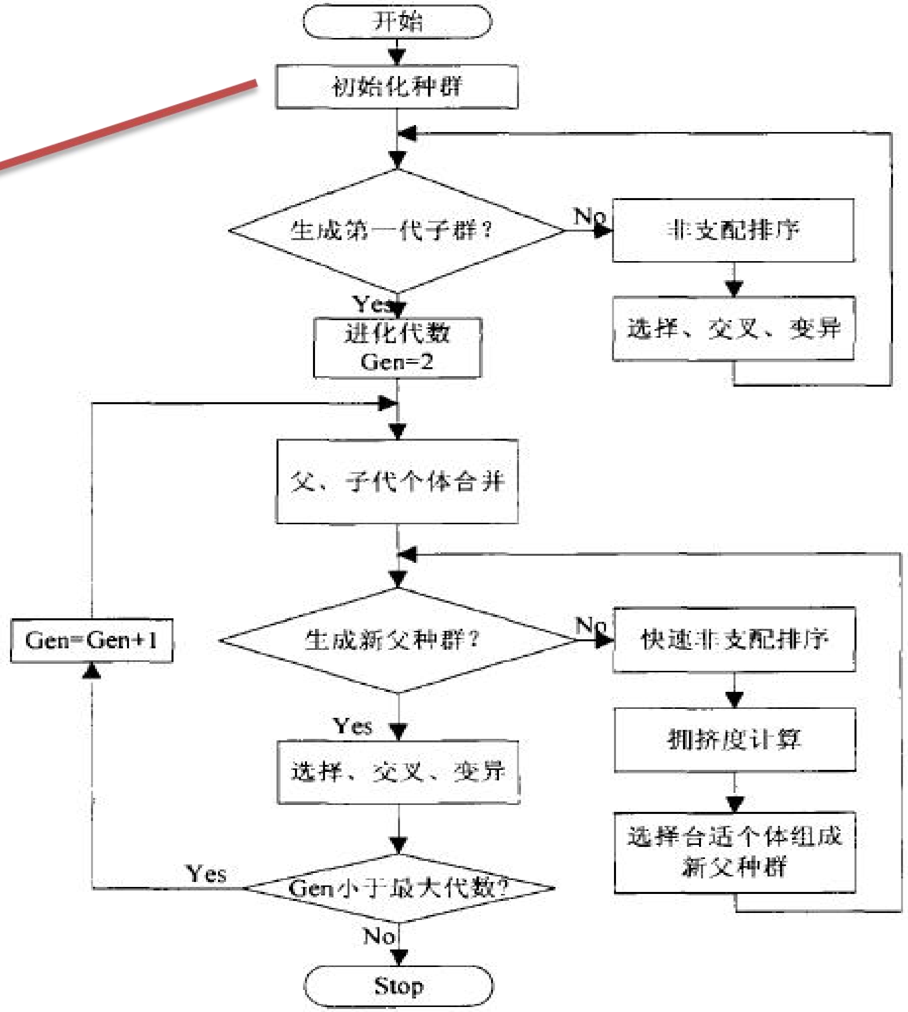
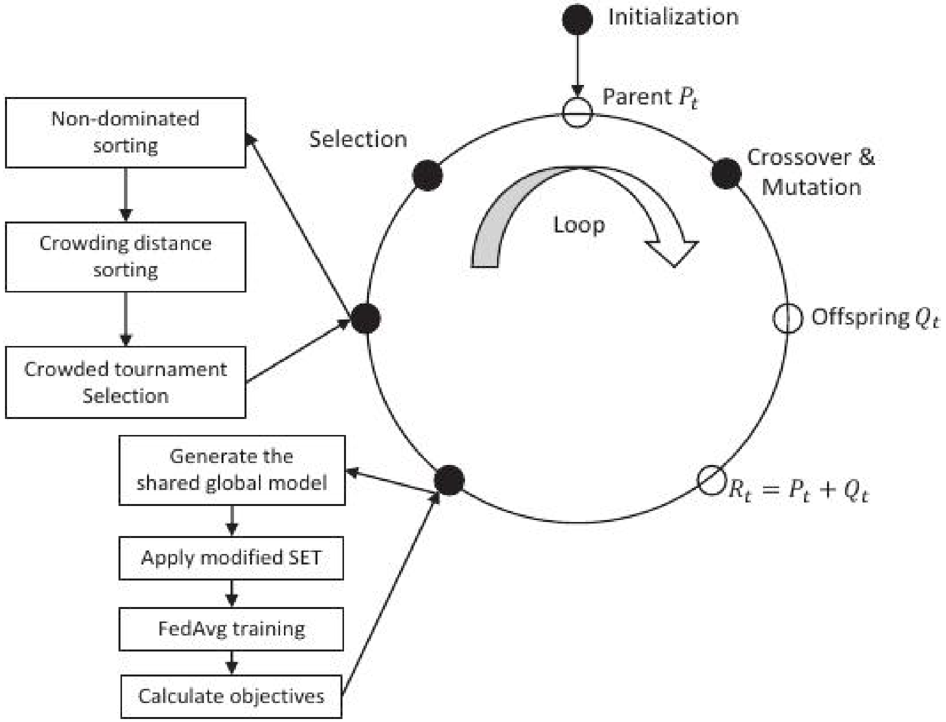


图 3.2 NSGA-II 基本流程

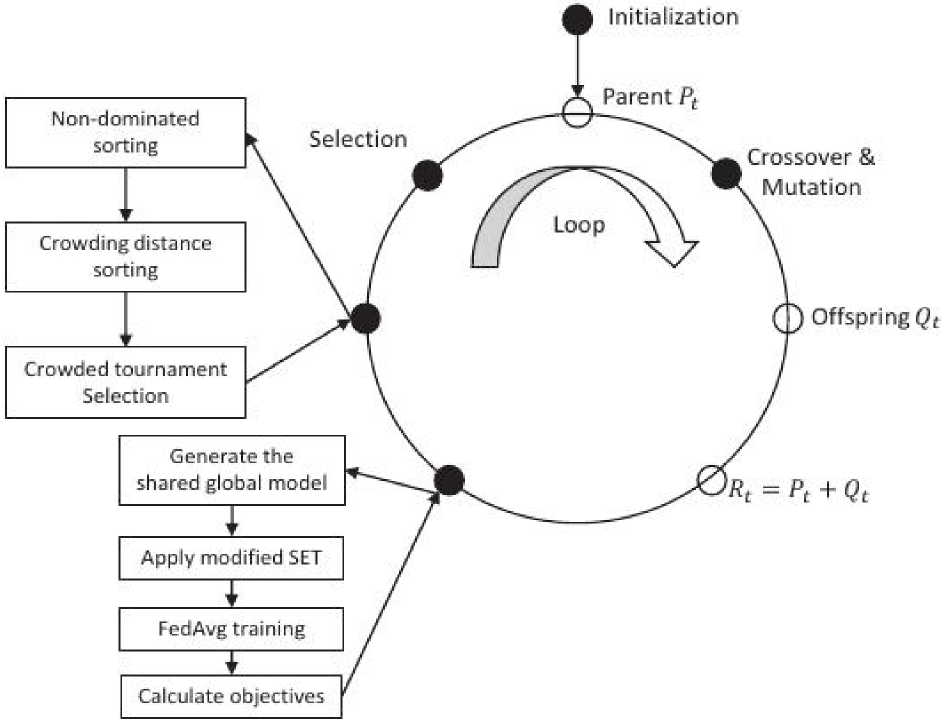
NSGA-II



Algorithm 5 Multi-objective evolutionary optimization

```
Randomly generate parent solutions P_t where |P_t| = M
for each generation t = 1, 2, ... do
    Generate offspring |Q_t| = M through crossover and mutation
    R_t = P_t + Q_t
    Evaluate f_t^1 and f_t^2 by Algorithm 4
    f ← (f_t^1, f_t^2)
    for each solution in R_t do
        Do non-dominated sorting and calculate crowding distance on f
        Select high-ranking solutions from R_t
        Let P_t = R_t
    end for
end for
```

NSGA-II



通过多目标的遗传演化算法得到较优的超参数以及网络结构

但是计算量偏大

在数据非独立同分布时效果一般

thanks