Privacy-Preserving Deep Learning via Weight Transmission

Le Trieu Phong and Tran Thi Phuong

汇报人：卢睿博，李卓翰

Privacy-preserving SGD for distributed trainers via weight transmission

通过权重传输实现的针对分布式训练者的隐私保护随机梯度下降算法

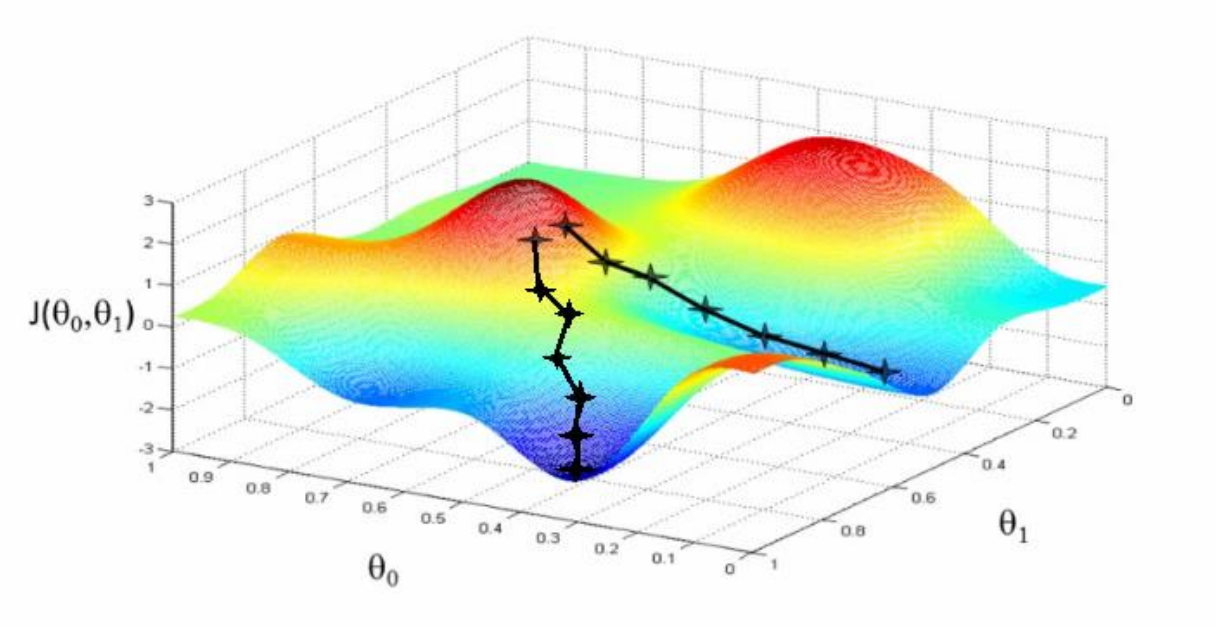
算法： SGD 随机梯度下降
目的： 隐私保护
环境： 分布式神经网路 & 深度学习

目录 Content

- 介绍 Introduction
- 系统 Systems
  - SNT & FNT
- 定理及证明 Theorem & Proof
- 应用预定义 Hedge
- 实验 Experiment
- 总结 Conclusion

介绍 Introduction

GD 最速下降法 – Gradient Descent



公式:

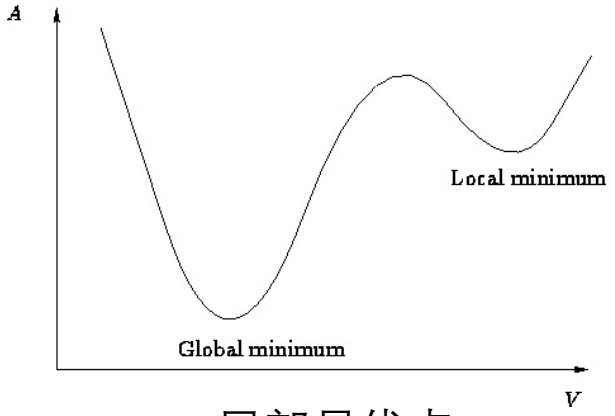
$$x_{t+1} = x_t - \eta_t \nabla f(x_t)$$

缺点:

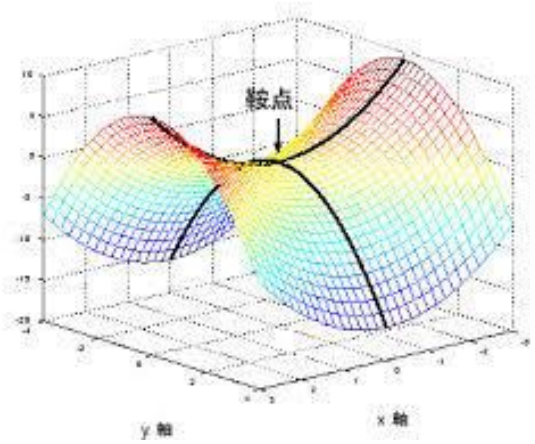
- 精确导数-算的慢
- 无法逃离鞍点和局部最优点

SGD 随机最速下降法Stochastic Gradient Descent

导数为0



局部最优点



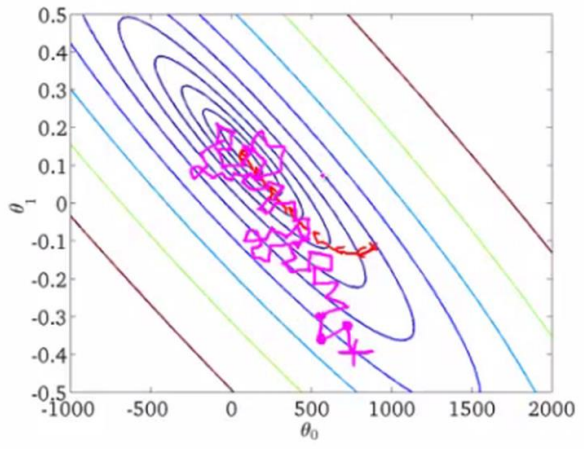
鞍点

$$G_t = \frac{\delta J(W_{t-1}, X_t, Y_t)}{\delta W}$$

$$W_t = W_{t-1} - \alpha(t) \cdot G_t$$

不必准确

$$E[G_t] = \nabla f(X_t)$$



导数可以包含噪声，所以算得很快

大量的理论工作说明，只要噪声不离谱，其实（至少在f是凸函数的情况下），SGD都能够很好地收敛

SGD有非常多的优良性质．它能够自动逃离鞍点，自动逃离比较差的局部最优点，而且，最后找到的答案还具有很强的一般性（generalization），即能够在自己之前没有见过但是服从同样分布的数据集上表现非常好

隐私保护 Privacy-preserving

诚实但好奇 Honest-but-curious

也称为半诚实模型 Semi-Honest Model

诚实Honest：

结点会诚实的将他的工作全部完成

好奇Curious：

结点会将所有他的数据保存下来并利用他们进行推测攻击

共谋 Collusion

Trainer 和 server 是一伙的

server

保密性 Secrecy：

不能泄露数据集的任何信息 data leakage

数据集不能离开属于他的训练者

权值传输： Weight Transmission

不暴露梯度，并引入一个难解问题，利用权值的递归性达到分布式训练

对称加密： Symmetric Encryption

利用对称加密不将明文暴露给curious

TLS/SSL

防止被监听信道

区别 difference

1. 能使用所有的激活函数- 意味着不会使用隐私保护友好的近似
    1. 常用的多项式近似法对于激活函数的选择是有要求的，并且还在讨论中
2. SGD的梯度(gradient)不会被发送，只发送权值参数(weight parameter)
3. 健壮性：在只有一个trainer可信，server和其他trainer共谋的情况下也能保证不泄露数据
4. 准确性：能保证具有与基于数据集的并集的原SGD相同的准确率

与其他论文比较

TABLE I

PRIVACY-PRESERVING DISTRIBUTED DEEP LEARNING SYSTEMS

Paper	Use of cryptography	Activation function	Security against honest-but-curious server	Security against collusion	Trainer transmission
[2]	Transport Layer Security (TLS)	Any	No	No	(Parts of) Gradients
[4]	Additively homomorphic encryption & TLS	Any	Yes	No	(Encrypted) Gradients
This paper	Transport Layer Security (TLS)	Any	Yes	Yes	Weights

[2] 为第七次讲的文章-接下来会说到
[4] 为几乎同样作者用同态加密实现的文章

Amine Boulemtafes, Abdelouahid Derhab, Yacine Challal. A review of privacy-preserving techniques for deep learning. Neurocomputing, Elsevier, 2020, 384, pp.21-45. ff10.1016/j.neucom.2019.11.041ff. ffhal-02921443

回顾 Review

Gradient or Weight

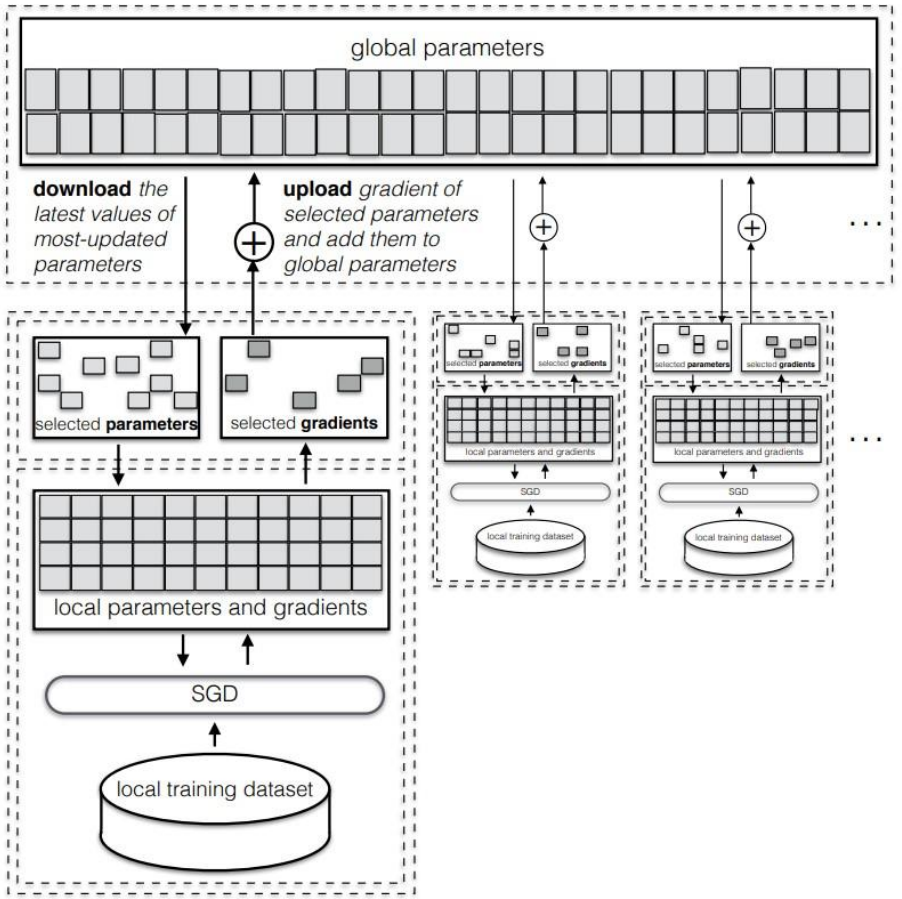
$$W_{server} := W_{server} - \alpha \cdot G_{local}^{selective}$$

$G_{local}^{selective}$=本地计算的梯度选择的一部分(1%-10%)

节省加密时间，但这样做可能会造成本地数据的泄露，

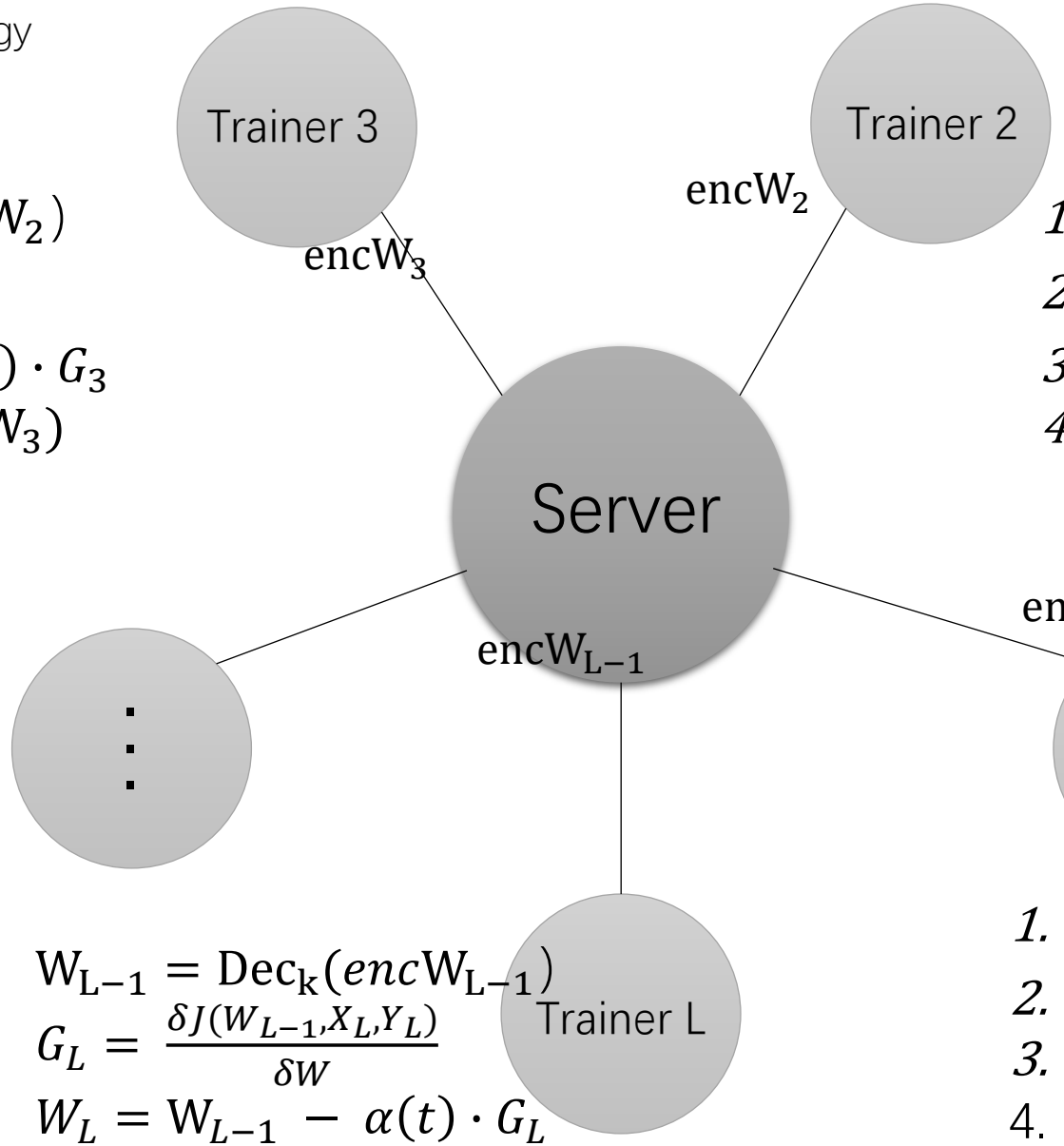
尽管在Sec.7 中文章作者又利用了差分隐私增加拉普拉斯噪声的方法，但是**数据保密性和差分隐私是正交**的，所以仍可能造成泄露

[4] 中利用同态加密的方法，但由于仍是梯度传输，在面对共谋时仍然可能造成泄露



R. Shokri and V. Shmatikov, “Privacy-preserving deep learning,” in Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur., I. Ray, N. Li, and C. Kruegel, Eds., Oct. 2015, pp. 1310–1321.
L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, “Privacypreserving deep learning via additively homomorphic encryption,” IEEE Trans. Inf. Forensics Security, vol. 13, no. 5, pp. 1333–1345, May 2018.

SNT Server-aided Network Topology



1. $\mathrm{W}_2 = \mathrm{Dec}_k(encW_2)$
2. $G_3 = \frac{\delta J(W_2, X_3, Y_3)}{\delta W}$
3. $W_3 = \mathrm{W}_2 - \alpha(t) \cdot G_3$
4. $\mathrm{encW}_3 = \mathrm{Enc}_k(\mathrm{W}_3)$

1. $\mathrm{W}_1 = \mathrm{Dec}_k(encW_1)$
2. $G_2 = \frac{\delta J(W_1, X_2, Y_2)}{\delta W}$
3. $W_2 = \mathrm{W}_1 - \alpha(t) \cdot G_2$
4. $\mathrm{encW}_2 = \mathrm{Enc}_k(\mathrm{W}_2)$

1. $\mathrm{W}_{L-1} = \mathrm{Dec}_k(encW_{L-1})$
2. $G_L = \frac{\delta J(W_{L-1}, X_L, Y_L)}{\delta W}$
3. $W_L = \mathrm{W}_{L-1} - \alpha(t) \cdot G_L$
4. $\mathrm{encW}_L = \mathrm{Enc}_k(\mathrm{W}_L)$

1. $G_1 = \frac{\delta J(W_0, X_1, Y_1)}{\delta W}$
2. $W_1 = \mathrm{W}_0 - \alpha(t) \cdot G_1$
3. $\mathrm{encW}_1 = \mathrm{Enc}_k(\mathrm{W}_1)$
4. 多次迭代

符号	意义
Server	Honest-but-Curious
$\mathrm{Dataset}_i$	每个人掌握的数据集
(X,Y)	Dataset的子集
K	Trainer的共同密钥
$\mathrm{Enc}_k(W)$	用k密钥对W进行加密
encW	加密后的密文
$\mathrm{Dec}_k(c)$	用k密钥对c进行解密
W_0	随机选择
PS	大部分都是vector

FNT Fully-connected Network Topology

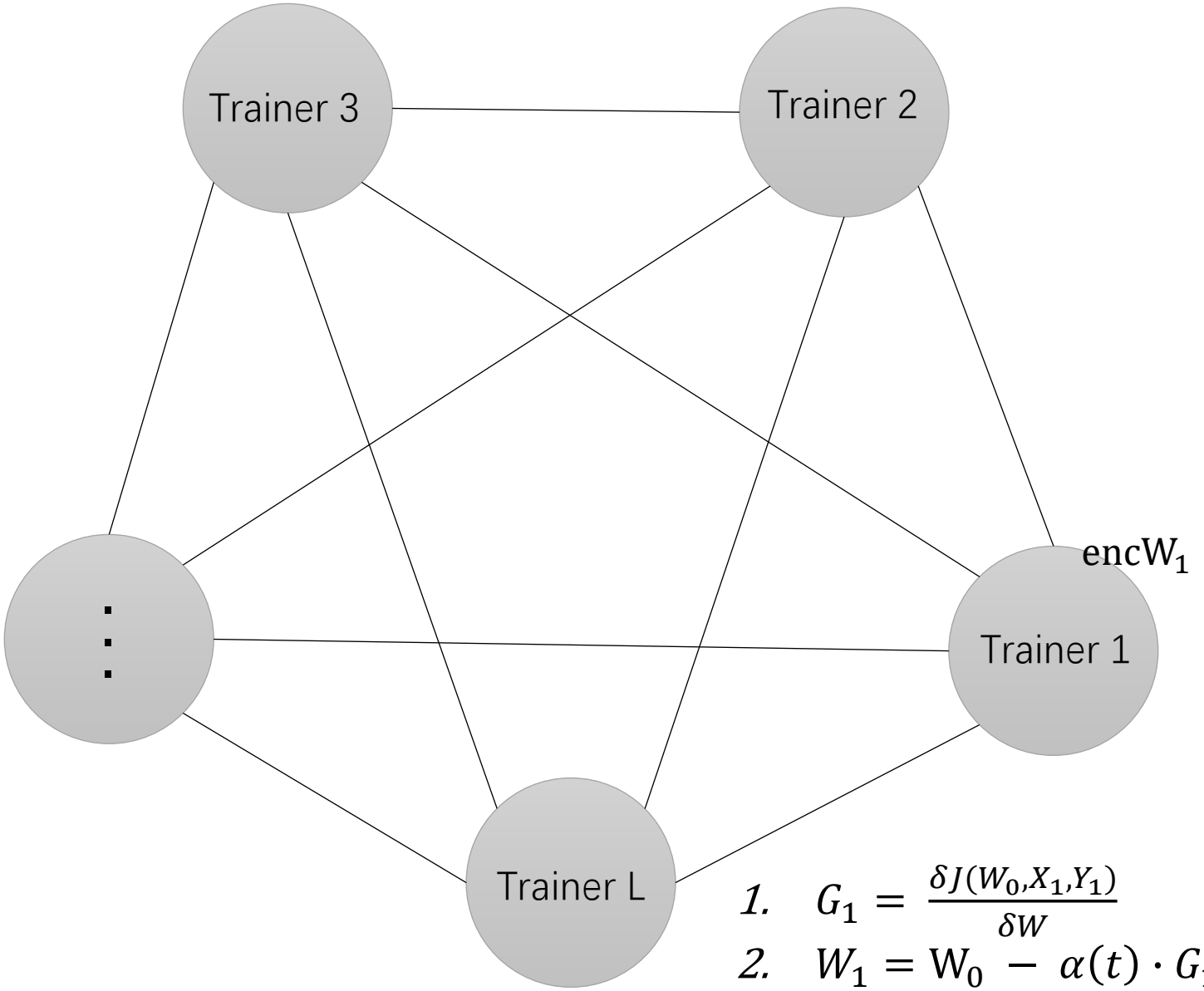
Wo 由第一个trainer随机选择的初始权重

Dataset(i) : 根据分布不同可以是整个数据集或者随机选择的一部分

K：trainer 间共享的密钥，但是不能告诉 server –FNT 就不需要了

工作模式：可以按顺序也可以随机发送

为什么一定要全连接呢?
1. 提高可靠性
2. 可以适应后一种随机发送的工作模式
3. 可以进行**匿名传输**



1. $G_1 = \frac{\delta J(W_0, X_1, Y_1)}{\delta W}$
2. $W_1 = \mathrm{W}_0 - \alpha(t) \cdot G_1$
3. $\mathrm{encW}_1 = \mathrm{Enc}_k(\mathrm{W}_1)$

定理及证明 Theorem & Proof

Theorem 1/4: 面对honest-but-curious server的安全

1. 利用对称加密使server只能看到密文ciphertext
2. 加密要求抵抗**选择明文攻击**

Theorem 2: 面对极端共谋collusion的安全性

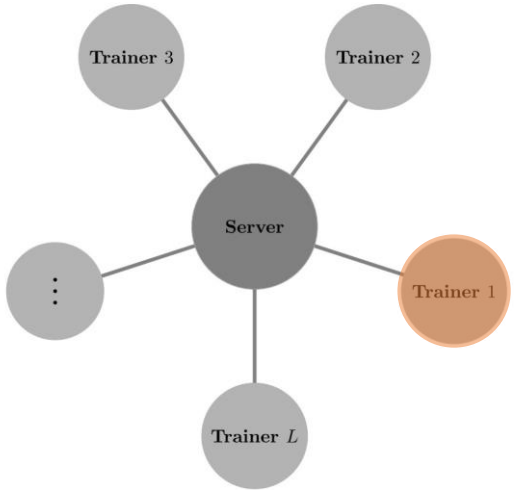
只有trainer1可信，trainer2-l,server都不可信的情况下，坏人也不能计算出trainer1的数据集信息，除非他们解决**非线性**问题或**子集和**问题

- $(X_i, Y_i)\ (1 \le i \le n)$是Dataset1洗牌后的一小部分
- α_i 可以是相同的，也可以是trainer1自己选择的
- $n = |Dataset1|/batch\_size \gg 1$

$$W^{(init)} - W^{(final)} = \alpha_1 G_1 + \cdots + \alpha_n G_n$$

$$G_1 \leftarrow \frac{\delta J(W_0, X_1, Y_1)}{\delta W}$$
$$W_1 \leftarrow W_0 - \alpha_1 G_1$$
$$\vdots$$
$$G_i \leftarrow \frac{\delta J(W_{i-1}, X_i, Y_i)}{\delta W}$$
$$W_i \leftarrow W_{i-1} - \alpha_i G_i$$
$$\vdots$$
$$G_n \leftarrow \frac{\delta J(W_{n-1}, X_n, Y_n)}{\delta W}$$
$$W_n \leftarrow W_{n-1} - \alpha_n G_n$$

$$\begin{aligned} W^{(final)} &= W_n \\ &= W_{n-1} - \alpha_n G_n \\ &= W_{n-2} - \alpha_{n-1} G_{n-1} - \alpha_n G_n \\ &\vdots \\ &= W_0 - (\alpha_1 G_1 + \cdots + \alpha_n G_n) \\ &= W^{(init)} - (\alpha_1 G_1 + \cdots + \alpha_n G_n). \end{aligned}$$



定理及证明 Theorem & Proof

Theorem 3/5 : SNT的准确性证明

证明分布式进行W的计算，和在并集上进行运算得到的结果的准确性是相同的
PS：去掉了加解密过程

Initialize W randomly
for mini-batch (X, Y) in $Dataset_1$:
$G \leftarrow \frac{\delta J(W,X,Y)}{\delta W}$
$W \leftarrow W - \alpha(t) \cdot G$
endfor
$\vdots$
for mini-batch (X, Y) in $Dataset_L$:
$G \leftarrow \frac{\delta J(W,X,Y)}{\delta W}$
$W \leftarrow W - \alpha(t) \cdot G$
endfor
	Initialize W randomly
	Set $CombinedDataSet = Dataset_1 \cup \cdots \cup Dataset_L$
	for mini-batch (X, Y) in $CombinedDataSet$:
	$G \leftarrow \frac{\delta J(W,X,Y)}{\delta W}$
	$W \leftarrow W - \alpha(t) \cdot G$
	endfor

Fig. 3. Pseudocodes for the proof of Theorem 3.

附加考虑 additional consideration

1. 在极端共谋下的实验数据:

使未知数的数量远大于等式的数量

$$V_1 = \alpha_1 G_1^{(1)} + \cdots + \alpha_n G_n^{(1)}$$
$$\vdots$$
$$V_k = \alpha_1 G_1^{(k)} + \cdots + \alpha_n G_n^{(k)}$$

TABLE III

THE NUMBER OF UNKNOWNS AND EQUATIONS IN (10)

Dataset (in Section V)	**#unknowns (10)** (= #data items in local dataset)	**#equations in (10)** (= #central epochs k)
Pima	30	20
Breast Cancer	19	5
Banknote Authentication	39	1
Adult Income	1628	120
Skin/NonSkin	9802	20
Credit Card Fraud Detection	12816	30
MNIST	10,000	639 (MLP), 193 (CNN)
CIFAR-10	10,000	50 (CNN), 100 (ResNet)
CIFAR-100	10,000	100 (ResNet)

*In all cases, #unknowns $>$ #equations. If each unknown is a real vector of dimension d, then the number of variables in (10) becomes $d\times$ #unknowns.

2. 扩张数据集的大小，增加训练效果

例如图片：利用旋转，裁剪，翻转等方法，过拟合后面会解决

3. 使用SGD的进化版本

例如： RMSProp [27] or Adam [28],

附加限制 additional hedge

达到完美的，丝毫不泄露是不可能的

1. Dalenius desideratum – 完全的随机数传输

1. 增加差分隐私 – 添加噪声
    1. 在准确性和**隐私性**之间找到均衡

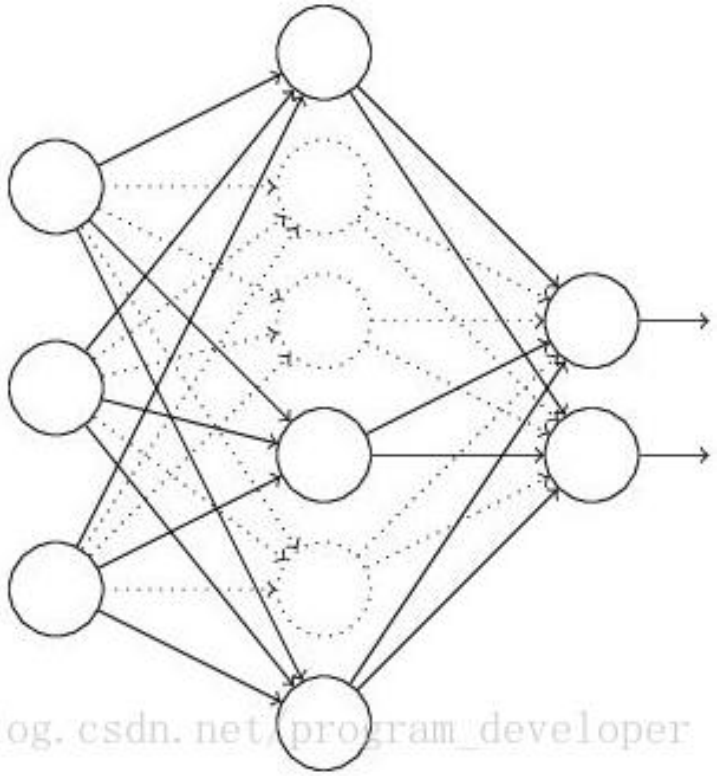
2. 利用**Dropout** 的方法增强差分隐私
    1. 降低结点间依赖性

3. 每个结点利用归一化，Dropout 等方法防止过拟合

4. 匿名传输 – 增强隐私性
    1. 隐藏数据源/使数据源可能来自集合中的随机元素

5. 每个结点发现问题就可以进行防御性进攻指（1，2，3）

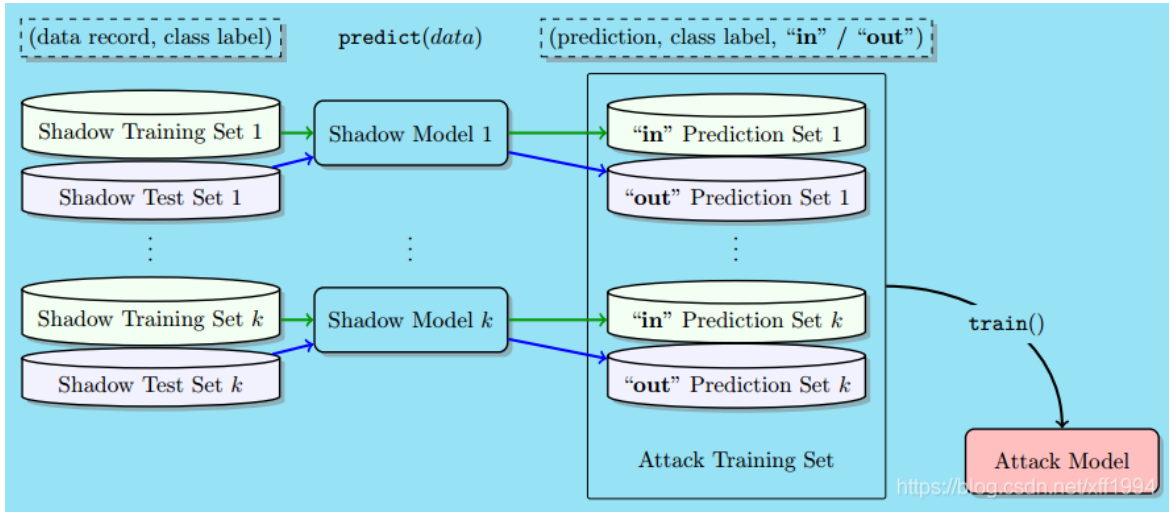
6. 当面临实际的攻击时，它可以这样解决：目的是找到数据源
    1. 综合前1,2,4的方法，不让攻击者找到数据源
    2. 当面临成员推理攻击时



dropout

成员推理攻击 member inference attack

目的：给出一条样本，可以推断该样本是否在模型的训练数据集中——即便对模型的参数、结构知之甚少，该攻击仍然有效



构建一个二分类模型，以X,Y为参数，判断该项是否在数据集中
利用影子模型(shadow model)构建与原目标模型训练集相似的数据集

Model-based synthesis： 直观上，如果目标模型以很高的概率给出了某条record的类别，那么该record与目标模型训练集中的数据应该是十分相似的。所以，可以用目标模型本身来构建影子模型的训练数据：

Statistics-based synthesis： 攻击者知道目标模型训练数据的分布信息，比如feature的边缘分布，那么可以直接由分布生成数据。

Noisy real data： 攻击者也许可以获得和目标模型训练集数据相似的数据，可以认为是目标模型训练集的噪声版本，直接利用之