Background

完全基于MPC的ML预测和训练一直是关注的热点，之前的SecureML、ABY3等论文一直在尝试这个难题。FALCON使用和ABY3同样的三方计算下的Replicated Secret Sharing方案来提升系统的计算和传输性能。和之前的方案不同的是，FALCON中完全使用算术秘密分享，但是使用了两个不同的环来处理线性计算（大环，$\mathbb{Z}_L, L = 32$）和非线性计算（小环，$\mathbb{Z}_{37}$）。首先，作者总结了现存方案，并和FALCON做了比较。

	Framework	Inference	Training	Semi-honest	Malicious	Linear	Convolution	Exact ReLU	Maxpool	Batch-Norm	HE	GC	SS	LAN	WAN	MNIST	CIFAR-10	Tiny ImageNet	From [4]	From [5]	From [6]	LeNet	AlexNet	VGG-16
		Private Capability		Threat Model		Supported Layers					Techniques Used			LAN/ WAN		Evaluation Dataset			Network Architectures					
		Theoretical Metrics												Evaluation Metrics										
2PC	MiniONN [6]	●	○	●	○	●	●	●	●	○	●	●	●	●	○	●	●	○	●	○	●	○	○	○
	Chameleon [5]	●	○	●	○	●	●	●	●	○	○	●	●	●	●	●	●	○	●	●	●	○	◐	○
	EzPC [7]	●	○	●	○	●	●	●	●	○	○	●	●	●	●	●	●	○	●	○	●	○	○	○
	Gazelle [8]	●	○	●	○	●	●	●	●	○	●	●	●	●	○	●	●	○	●	●	●	○	○	○
	SecureML [4]	●	●	●	○	●	●	●	●	○	●	●	●	●	◐	●	○	○	●	○	○	○	○	○
	XONN [9]	●	○	●	●	●	●	●	●	◐	○	●	●	●	○	●	●	○	●	●	●	○	○	◐
	Delphi [10]	●	○	●	○	●	●	●	●	○	●	●	●	●	○	○	●	○	○	○	●	○	○	◐
3PC	ABY^3 [11]	●	●	●	●	●	●	●	●	○	○	●	●	●	◐	●	○	○	●	●	●	○	○	○
	SecureNN [12]	●	●	●	◐	●	●	●	●	◐	○	○	●	●	●	●	○	○	●	●	●	○	○	○
	CryptFlow [13]	●	○	●	●	●	●	●	●	○	○	○	●	◐	○	●	●	●	●	●	●	●	○	◐
	QuantizedNN [14]*	●	○	●	●	●	●	●	●	◐	◐	○	●	●	●	○	○	●	○	○	○	○	○	◐
	ASTRA [15]	●	●	●	◐	●	●	●	●	○	○	●	●	●	●	●	○	○	●	○	○	○	○	○
	BLAZE [16]	●	●	●	●	●	●	●	●	○	○	●	●	●	●	◐	○	○	●	○	○	○	○	○
	Falcon (This Work)	●	●	●	●	●	●	●	●	●	○	○	●	●	●	●	●	◐	●	●	●	●	●	●
4PC	FLASH [17]	●	●	●	●	●	●	●	●	○	○	○	●	●	●	●	◐	○	●	◐	○	○	○	○
	Trident [18]	●	●	●	●	●	●	●	●	○	○	●	●	●	●	●	○	○	●	○	○	○	○	○

进一步，作者分别提出了针对各层的3PC下的计算协议。

基本原语

Multiplication：加法和数乘方法比较简单。针对线性层的乘法，包括矩阵乘法和卷积，则使用和ABY3相同的乘法和截断方法。乘法本身需要一次交互，截断则需要预处理生成的关联随机数。

Reconstruction：秘密恢复则需要每一方向下一方发送对方缺失的秘密分享。

Select Shares Π_{SS}: 给定$([x]^L, [y]^L, [b]^2)$，如果$b = 0$则返回$[x]^L$，否则返回$[y]^L$。本文首先生成关联随机数$[c]^2$和$[c]^L$。然后公开$b \oplus c = e$。如果$e = 1$，令$[d]^L = [1 - c]^L$；否则，令$[d]^L = [c]^L$。最后，计算$[z]^L = [(y - x) \cdot d]^L + [x]^L$。

XOR with a public bit: 对于一比特在环内的秘密分享$[x]^m$和一个公开的比特值b，要求$[y]^m = [x \oplus b]^m$，可以计算$y = x + b - 2bx$。因为b是公开的，该计算不需要交互。

Evaluating $[(-1)^\beta \cdot x]^m$: 给定$[x]^m$和$[\beta]^m$，计算$[(1 - 2\beta) \cdot x]^m$即可。

核心模块

鉴于乘法等计算之前的博文已经介绍过很多次，这次仅介绍有关非线性计算的部分，即激活函数部分的计算。

Private Compare

本文的比较方案和其他方案，例如ABY3等不同。在本文构建的Private Compare中，做比较之前，三方参与者已经得到了秘密分享值得比特分解，而且每一比特的分解是在$\mathbb{Z}_p$上的。本文的目标则是比较$x \geq r$，其中r是一个公开的数。鉴于x的比特分解已知，r是公开数，比较则可以和SecureNN中类似按比特比较。算法如下：

Algorithm 1 Private Compare $\Pi_{\mathsf{PC}}(P_1, P_2, P_3)$:

Input: P_1, P_2, P_3 hold secret sharing of bits of x in $\mathbb{Z}_p$.
Output: All parties get shares of the bit $(x \geq r) \in \mathbb{Z}_2$.
Common Randomness: P_1, P_2, P_3 hold a public ℓ bit integer r, shares of a random bit in two rings $[\![\beta]\!]^2$ and $[\![\beta]\!]^p$ and shares of a random, secret integer $m \in \mathbb{Z}_p^*$.

1: **for** $i = \{\ell - 1, \ell - 2, \ldots, 0\}$ **do**
2: Compute shares of $u[i] = (-1)^{\beta}(x[i] - r[i])$
3: Compute shares of $w[i] = x[i] \oplus r[i]$
4: Compute shares of $c[i] = u[i] + 1 + \sum_{k=i+1}^{\ell} w[k]$
5: **end for**
6: Compute and reveal $d := [\![m]\!]^p \cdot \prod_{i=0}^{\ell-1} c[i] \pmod p$
7: Let $\beta' = 1$ if $(d \neq 0)$ and 0 otherwise.
8: **return** Shares of $\beta' \oplus \beta \in \mathbb{Z}_2$

其中，β起到茫化作用，step 1-5中setp 2计算乘法需要一次交互，其余为本地计算；step 6需要$\log_2 \ell + 1$次交互。当$d = 0$时，即说明$c[i]$中有一项为0。假设$c[t] = 0$，$\beta = 0$，那么$x[t] - r[t] = -1$；对于$i > t$，$w[i] = 0$恒成立；如此$c[t] = 0$。而对于$i < t$，由于$w[t] = 1$，$c[i] \geq 1$恒成立。所以在x和r最高不相同的比特位置，有$r[t] = 1$，$x[t] = 0$。那么$d = 0$时，则有$x < r$。故而$\beta' = 0$。否则当$d \neq 0$时有$\beta' = 1$。

Wrap Function

在$\mathbb{Z}_L$中，wrap_2定义如下

$$\text{wrap}_2(a_1, a_2, L) = \begin{cases} 0, a_1 + a_2 < L \\ 1, otherwise \end{cases}$$

对于三个数，有
$$
{\rm wrap}*{3e}(a_1, a_2, a_3, L)=\begin{cases}*
0, \sum{i=1}^3 a_i < L\
1, L\le \sum_{i=1}^3 a_i < 2L\
2, 2L \le sum_{i=1}^3 a_i < 3L

\end{cases}
$进一步，定义{\rm wrap}(a_1, a_2, a_3, L)={\rm wrap}_{3e}(a_1, a_2, a_3, L) \mod{2}$。

为了计算wrap_3，参与方首先生成随机数$[x]^L$，并生成x的每一比特在$\mathbb{Z}_p$中的算术分享$[x[i]]^p$，并生成$\alpha = \text{wrap}_3(x_1, x_2, x_3, L)$d的比特分享$[\alpha]^2$。进一步，对于秘密值$a$，有：

$$r = a + x - \eta \cdot L \ (1)$$
$$r = r_1 + r_2 + r_3 - \delta_e \cdot L \ (2)$$
$$r_i = a_i + x_i - \beta_i \cdot L \ (3)$$
$$x = x_1 + x_2 + x_3 - \alpha_e \cdot L \ (4)$$
$$a = a_1 + a_2 + a_3 - \theta_e \cdot L \ (5)$$

$(1) - (2) - (3) + (4) + (5)$有，

$$\theta_e = \beta_1 + \beta_2 + \beta_3 + \delta_e - \eta - \alpha_e,$$

上式 $\mod 2$则得到$\theta = \beta_1 + \beta_2 + \beta_3 + \delta - \eta - \alpha$。需要注意的是(3)式包含三个式子。算法如下

 image-20211030203330261

在已知关联随机数的情况下，只需要Step4 交互计算（调用Private Compare）。

ReLU

对于$\text{ReLU}(a)$，关键在于计算a最高有效位。本文的一个重要发现对于$a = a_1 + a_2 + a_3 \mod L$，有

$$\mathrm{MSB}(a) = \mathrm{MSB}(a_1) + \mathrm{MSB}(a_2) + \mathrm{MSB}(a_3) + c \mod 2$$

其中，c是三个a_i的低$\ell - 1$位对于最高位的进位。即$c = \mathrm{wrap}_3(2a_1, 2a_2, 2a_3, L)$。

如此，$\mathrm{DReLU}(a) = \mathrm{MSB}(a_1) \oplus \mathrm{MSB}(a_2) \oplus \mathrm{MSB}(a_3) \oplus \mathrm{wrap}_3(2a_1, 2a_2, 2a_3, L) \oplus 1$。得到$\mathrm{DReLU}$之后，计算$\Pi_{\mathrm{SS}}$杰克得到激活函数结果。协议如下：

Maxpool

池化层在协议层面和SecureNN一样，不同的是比较部分的计算调用本文构造的方案。

Division & BatchNorm

本文除法利用近似计算。本文首先计算除数的指数，即计算$2^{\alpha} \leq x < 2^{\alpha+1}$中的$\alpha$。算法如下：

Step4中，$c = 1$表示$x - 2^{2^i + \alpha} \geq 0$，则需要将当前指数加入最终结果。

得到α之后，进行如下计算

在该近似算法中，除数需要满足$b \in [0.5, 1)$。本文采用的方法在于提取α之后，将近似用的常数2.9142和1都扩大$2^{\alpha+1}$从而使得$b \in [0.5, 1)$。最后结果截断乘法造成的scaling factor膨胀。

BatchNorm的算法如下：

除了计算均值和方差，剩下的部分和做除法类似，都是采用了近似算法。

实验效果

本文实验颇多，在此列举一下部分实验结果。

预测开销

	Framework	Threat Model	LAN/ WAN	Network-A		Network-B		Network-C	
				Time	Comm.	Time	Comm.	Time	Comm.
2PC	SecureML [4]	Semi-honest	LAN	4.88	-	-	-	-	-
	DeepSecure [45]	Semi-honest	LAN	-	-	9.67	791	-	-
	EzPC [7]	Semi-honest	LAN	0.7	76	0.6	70	5.1	501
	Gazelle [8]	Semi-honest	LAN	0.09	0.5	0.29	0.8	1.16	70
	MiniONN [6]	Semi-honest	LAN	1.04	15.8	1.28	47.6	9.32	657.5
	XONN [9]	Semi-honest	LAN	0.13	4.29	0.16	38.3	0.15	32.1
3PC	Chameleon [5]	Semi-honest	LAN	-	-	2.7	12.9	-	-
	ABY^3 [11]	Semi-honest	LAN	0.008	0.5	0.01	5.2	-	-
	SecureNN [12]	Semi-honest	LAN	0.043	2.1	0.076	4.05	0.13	8.86
	FALCON	Semi-honest	LAN	0.011	0.012	0.009	0.049	0.042	0.51
		Malicious	LAN	0.021	0.31	0.022	0.52	0.089	3.37
	SecureNN [12]	Semi-honest	WAN	2.43	2.1	3.06	4.05	3.93	8.86
	FALCON	Semi-honest	WAN	0.99	0.012	0.76	0.049	3.0	0.5
		Malicious	WAN	2.33	0.31	1.7	0.52	7.8	3.37
4PC	FLASH [17]	Malicious	LAN	0.029	-	-	-	-	-
	FLASH [17]	Malicious	WAN	12.6	-	-	-	-	-

Table 2. Comparison of inference time of various frameworks for different networks using MNIST dataset. All runtimes are reported in seconds and communication in MB. ABY^3 and XONN do no implement their maliciously secure versions. 2-party (2PC) protocols are presented here solely for the sake of comprehensive evaluation of the literature.

Framework	Threat Model	LAN/WAN	LeNet (MNIST)		AlexNet (CIFAR-10)		VGG16 (CIFAR-10)		AlexNet (ImageNet)		VGG16 (ImageNet)	
			Time	Comm.	Time	Comm.	Time	Comm.	Time	Comm.	Time	Comm.
FALCON	Semi-honest	LAN	0.047	0.74	0.043	1.35	0.79	13.51	1.81	19.21	3.15	52.56
	Malicious	LAN	0.12	5.69	0.14	8.85	2.89	90.1	6.7	130.0	12.04*	395.7*
	Semi-honest	WAN	3.06	0.74	0.13	1.35	1.27	13.51	2.43	19.21	4.67	52.56
	Malicious	WAN	7.87	5.69	0.41	8.85	4.7	90.1	8.68	130.0	37.6*	395.7*

Table 3. Comparison of inference time of various frameworks over popular benchmarking network architectures from the machine learning domain. All runtimes are reported in seconds and communication in MB. * indicate non-amortized numbers.

训练开销

Framework	Threat Model	LAN/ WAN	Network-A		Network-B		Network-C	
			Time	Comm.	Time	Comm.	Time	Comm.
SecureML [4]*	Semi-honest	LAN	81.7	-	-	-	-	-
SecureML [4]	Semi-honest	LAN	7.02	-	-	-	-	-
ABY^3 [11]	Semi-honest	LAN	0.75	0.031	-	-	-	-
SecureNN [12]	Semi-honest	LAN	1.03	0.11	-	-	17.4	30.6
FALCON	Semi-honest	LAN	0.17	0.016	0.42	0.056	3.71	0.54
	Malicious	LAN	0.56	0.088	1.17	0.32	11.9	3.29
SecureML [4]*	Semi-honest	WAN	4336	-	-	-	-	-
SecureNN [12]	Semi-honest	WAN	7.83	0.11	-	-	53.98	30.6
FALCON	Semi-honest	WAN	3.76	0.016	3.4	56.14	14.8	0.54
	Malicious	WAN	8.01	0.088	7.5	0.32	39.32	3.29
Batch Size, Epochs			128, 15		128, 15		128, 15	

Table 4. Comparison of training time of various frameworks over popular benchmarking network architectures from the security domain. All runtimes are reported in hours and communication in TB. * correspond to 2PC numbers. ABY^3 does not implement their maliciously secure protocols.

Framework	Threat Model	LAN/ WAN	LeNet		AlexNet (CIFAR-10)		VGG16 (CIFAR-10)		AlexNet (ImageNet)		VGG16 (ImageNet)	
			Time	Comm.	Time	Comm.	Time	Comm.	Time	Comm.	Time	Comm.
FALCON	Semi-honest	LAN	6.05×10^0	0.81	7.89×10^1	7.24	8.43×10^2	45.9	1.23×10^4	222.9	5.19×10^3	156.0
	Malicious	LAN	1.22×10^1	4.82	2.82×10^2	43.4	3.05×10^3	185.3	4.63×10^4	1598	1.95×10^4	1012
	Semi-honest	WAN	1.85×10^1	0.81	2.33×10^2	7.24	2.09×10^3	45.9	1.54×10^4	222.9	6.89×10^3	156.0
	Malicious	WAN	5.20×10^1	4.82	7.24×10^2	43.4	5.26×10^3	185.3	5.71×10^4	1598	2.47×10^4	1012
Batch Size, Epochs			128, 15		128, 90		128, 25		128, 90		128, 25	

Table 5. Comparison of training time of various frameworks over popular benchmarking network architectures from the machine learning domain. All runtimes are reported in hours and communication in TB.

总结

本文在三方下基于算术电路提出了一种训练和预测的框架，在线计算的效率提升了很多。但是，关于预计算的关联随机数生成还是没有给出新的高效方案，只能用已有方案来做。