A Privacy-Preserving Deep Learning Framework for Genomic Data

Delica Leboe-McGowan, Md. Momin Aziz, Noman Mohammed

Department of Computer Science, University of Manitoba

Motivation

- Human analysts cannot process large biomedical datasets.
- Deep learning, a form of artificial intelligence, finds complex patterns in large datasets, but users must ensure that outsourcing analysis will not violate health privacy rights.
- Genomic profiles are an example of sensitive personal information that could help guide medical diagnosis and treatment.

Related Work

- Several privacy-preserving deep learning frameworks have been developed.
- SecureNN achieves fast runtimes with secret sharing (Wagh et al., 2019)[1], a method that splits data into fragments that individually have no meaning.
- Researchers have developed secure computation protocols for many types of genome analysis (e.g., Zhang et al., 2015)[2] except deep learning.

Contributions

- Preserving input privacy while training a highly accurate deep learning model
- Affirming the viability of secret sharing frameworks for medical applications
- Demonstrating that SecureNN's exact computation of costly *non-linear* operations may not always be necessary to achieve high accuracy

Test Problem

- The 2019 iDASH Privacy & Security Workshop challenged participants to diagnose breast cancer from genomic data compiled by The Cancer Genome Atlas (TCGA).
- The TCGA's methods distill many gene activity measures into 17 814 numeric values[3].
- An accurate deep learning model must find reliable, informative patterns across these features, without knowing any donor's private gene expression profile.

Secret Sharing Scheme

- The feature values are concealed from the deep learning model through a process called additive secret sharing.
- For this type of secret sharing, a data owner randomly selects two or more numbers such that their sum equals the secret value that must be protected.

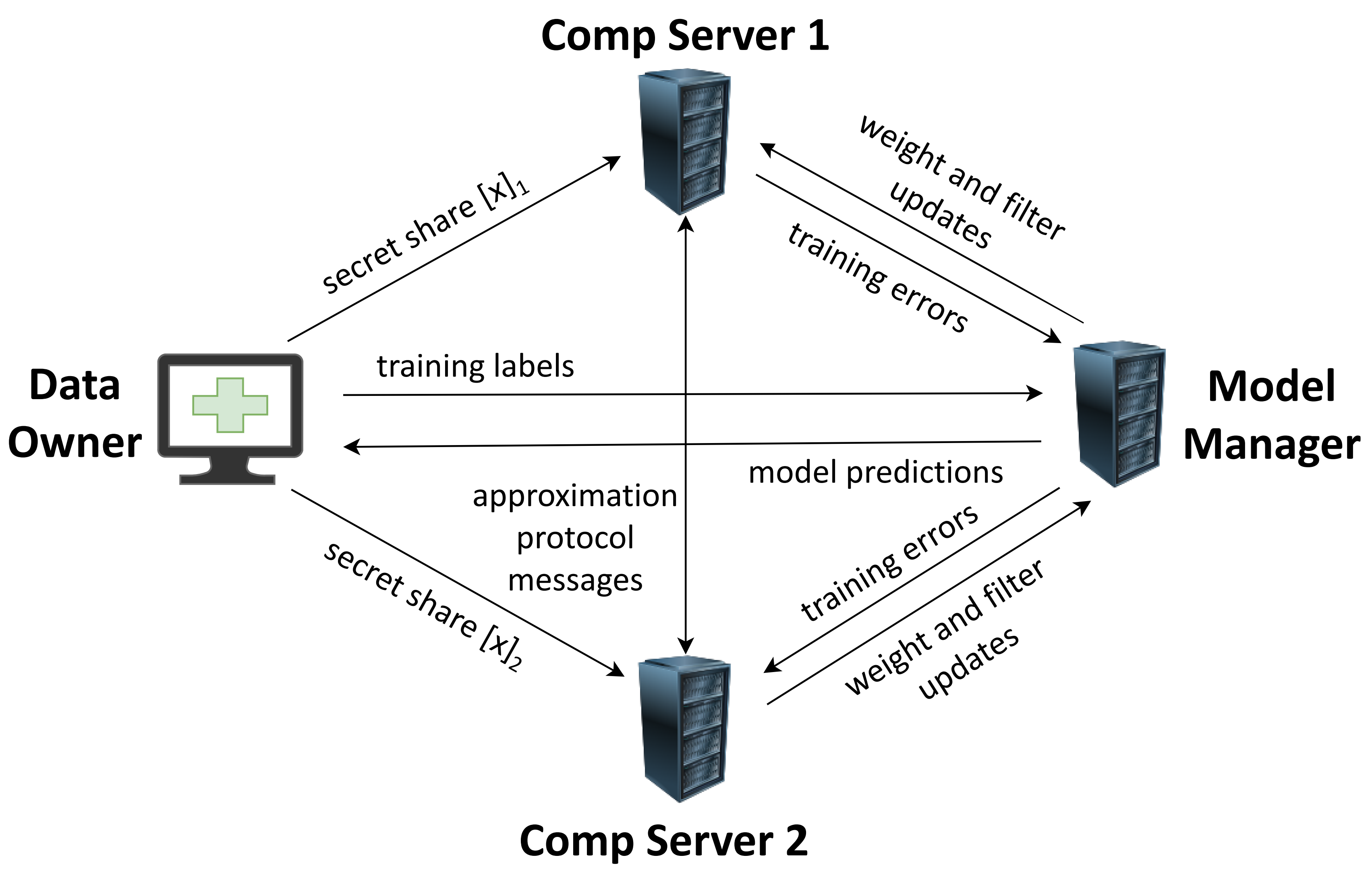
$$\underbrace{x}_{\text{Original Data Value}} = \underbrace{[x]_1}_{\text{Secret Share 1}} + \underbrace{[x]_2}_{\text{Secret Share 2}}$$

- If a server has access to only one of these secret shares ($[x]_1$ or $[x]_2$), it is impossible to know the exact value of the secret x.

Protocol Summary

The proposed framework requires four parties:

- *The Data Owner* (e.g., the hospital that collected the tissue sample) is the only party that should have access to a patient's gene expression profile.
- *The Model Manager* server stores the filters and weights that define how the deep learning model processes data.
- *The Two Comp Servers* compute the outputs produced by each layer of the model. For privacy, each only receives one additive secret share from the Data Owner.



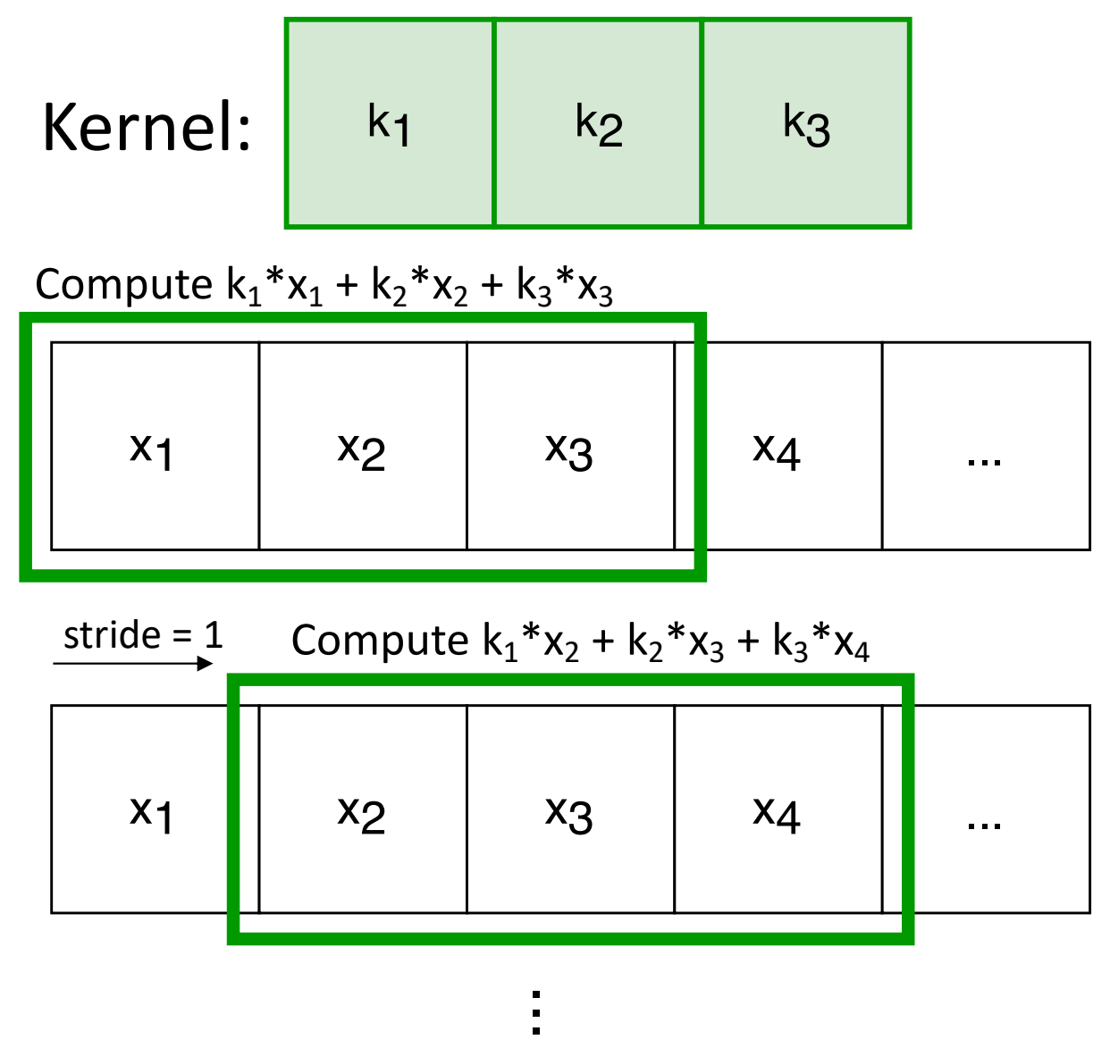
Deep Learning Model

- Convolutional Neural Networks (CNNs) are a popular deep learning option for classification tasks.
- CNNs reduce thousands of feature values into a single output by using four key operations—convolutions, batch normalization, ReLUs, and fully-connected layers.

Convolutions

- Give the model information about a particular region of data
- They are *linear* operations: $\text{conv}(x) = \text{conv}([x]_1) + \text{conv}([x]_2)$.
- The linearity allows Comp Servers to separately compute the convolutions of their secret shares.

1D Convolutions (with 1x3 Kernel)


Kernel:
k1
k2
k3
Compute $k_1*x_1 + k_2*x_2 + k_3*x_3$
x1
x2
x3
x4
...
stride = 1
Compute $k_1*x_2 + k_2*x_3 + k_3*x_4$
x1
x2
x3
x4
...

ReLUs (Rectified Linear Units)

$$\text{ReLU}(x) = \begin{cases} x, x \geq 0 \\ 0, x < 0 \end{cases}$$

- Given only one secret share, it is impossible to know whether x is positive or negative.
- The Comp Servers exchange the signs of their data, since they can determine the ReLU output immediately if both shares have the same sign.
- If the signs are different, Server 1 determines the ReLU output by using the mean of its data to guess whether its share likely has a larger magnitude than Server 2's value.

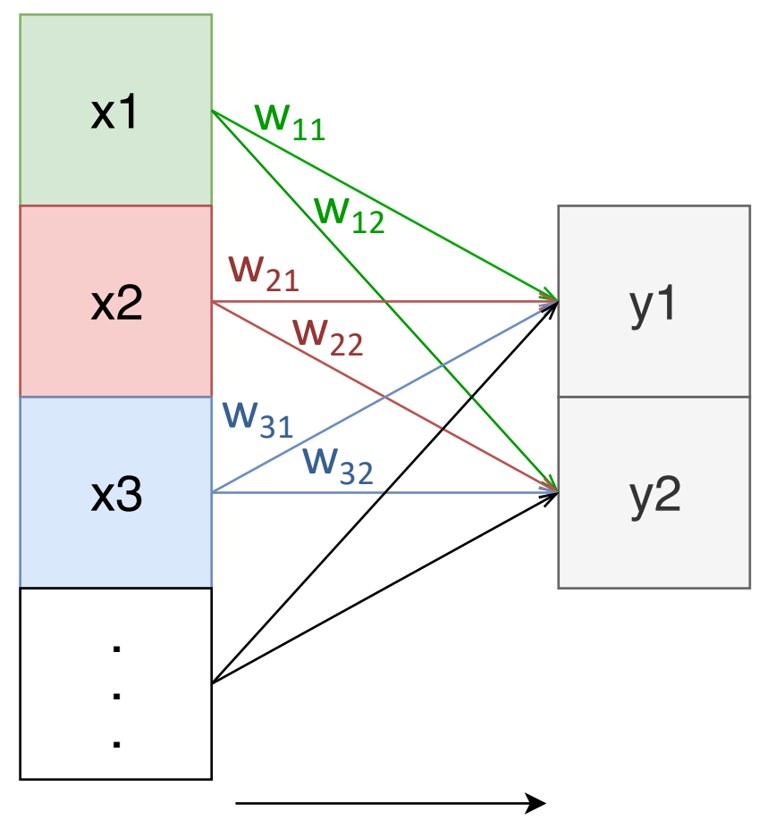
Batch Normalization

$$x_{\text{norm}} = \frac{x - \mu}{\sigma}$$

- Prevents outputs in the CNN from growing too large, but requires the mean value μ and the standard deviation σ to preserve relative distribution of features
- Finding μ is a linear operation (i.e., $\mu = \mu_1 + \mu_2$, where μ_n is the mean for secret share $[x]_n$), whereas the calculation of σ is *non-linear* ($\sigma \neq \sigma_1 + \sigma_2$).
- The Comp Servers send the standard deviations of their data shares to each other and take the average value, which is then divided by a constant to approximate σ.

Fully-Connected Layer

- These are critical for helping a CNN condense its layer outputs into smaller sets of features.
- Multiplication with a weights matrix ensures every output in the preceding layer affects every output in the following layer.
- Like convolutions, this matrix multiplication is a linear operation that is simple to implement in a secret shared setting.



multiply input vector X
by weights matrix W

Model Performance

- The TCGA's breast cancer dataset (BC-TCGA) was used to test implementations of the same CNN with and without the privacy-preserving protocols.
- This dataset included 98 records (49 positive & 49 negative; 68 to train & 30 to test).
- The training algorithm processed data 34 records at a time with a 0.01 learning rate.
- On a desktop computer with a 3.6 GHz Intel CPU, we tested the secure and unsecure versions of the CNN for different numbers of training epochs.

Average Runtimes and Accuracies across Five Independent Trials for Various Amounts of Network Training

Number of Training Epochs	Unsecure Training Runtime	Unsecure Test Accuracy	Secure Training Runtime (with 250 ms network delay)	Secure Test Accuracy
1	2 min 48 sec	74.0%	3 min 3 sec	70.7%
2	5 min 36 sec	91.3%	6 min 3 sec	88.7%
5	14 min 15 sec	98.7%	16 min 1 sec	99.3%
10	29 min 8 sec	98.7%	30 min 47 sec	99.3%

Conclusion

Our deep learning model achieves 99% accuracy after just over 30 minutes of training, demonstrating that this additive secret sharing approach can be a viable option for securely analyzing gene expression profiles.

Acknowledgements

This Undergraduate Research Award (URA) project was completed as part of a summer research term funded by The Office of the Vice President, Research and International, and the U of M Student Union (UMSU).

References

1. Wagh, S., Gupta, D., and Chandran, N. (2019). SecureNN: 3-party secure computation for neural network training. *PoPETs*, (3).

2. Zhang, Y., Blanton, M., & Almashaqbeh, G. (2015). Secure distributed genome analysis for GWAS and sequence comparison computation. *Proceedings of the 4th iDASH Privacy Workshop: Critical Assessment of Data Privacy and Protection (CADPP) Challenge*.

3. The Cancer Genome Atlas Network. (2012). Comprehensive molecular portraits of human breast tumours. *Nature*, 490, 61-70.

Contact: Delica Leboe-McGowan (leboemcd@myumanitoba.ca)