Hessian Network : an open platform for distributed intelligence over encrypted data

(Last Version - Revised and Released By LJP and KSH & BEA)

Abstract:

We propose a framework and platform that simulates a data and intelligence market place where users can provide models or data and receive a token as a reward for their work,the platform stipulates a Proof Of Work model abstracted from the "nonce finding algorithm present in crypto-currencies and turns into an evaluable problem for all .*

This paper is highly condensed and acts as an introducing reading for the community to understand the combinations and usability of both technologies .

Outline of the paper content

1. **Introduction**
    a. Problem Formulation

2. **Structure Preserving Encryption for Data**
    a. Homomorphic Encryption schemes and their feasibility
    b. Alternative Processes
        i. Multi Party Computation and feasibility

3. **Inference on encrypted data**
    a. Inference & Intelligence
        i. Mathematical Requirements for Evaluation
        ii. Feasibility
        iii. Extended models based on statistical methods

4. **Conclusion**

Problem Formulation :

- *“The world’s most valuable resource is no longer oil, but data”* , The Economist

It’s a fact the technologies that you use everyday are powered by data in different ways,social networks use your data to present a virtual presence of yourself,companies broker in your data in non-regulated ways,the websites you use everyday present you some sort of content based on an analysis of the data you provided them ,investors,hedge funds leverage data to make profit in the markets and so on .

With the advent of mobile devices, the amount of data that is generated has exploded exponentially. There are concerns about the vast amount of personal data send across to servers and used for generating money without the actual producer of data not benefiting. The emergence of Machine Learning had made things even more interesting. Most of the machine learning algorithms available now works by training models in a highly centralised environment, this means that user’s data need to be send to a server before actually used for learning.

Data is costly to collect,clean,manage and analyse this is why it’s valuable,it’s own value is backed by the insight it can provide ,such as price predictions,product offerings and such .

At the same time data can be useful to the open public,the insight that can be extracted from data is proportional to the people working on it , the idea of swarm-intelligence was previously discussed but omitted because data is to valuable to share to the public .

Encryption is essentially a mapping from a set A to a set B ,that’s how it works essentially it takes a usable input X and turns it into an unusable input Y but verifiable .Homomorphism is essentially a structure preserving mapping between two algebraic-structures of the same type (two rings,two groups …) such as operations on structures remains equivalent between the two parts ,in simpler terms If I multiply two encrypted numbers the decrypted result is always equal to the multiplication between the numbers but decrypted .

The possibility of the latter opens a new door to the problem formulated at first can we open the data to the public without revealing it's content .

I- Structure Preserving Encryption schemes and their feasibility

- **Homomorphic Encryption**

Homomorphic encryption schemes are cryptographic constructions which enable to securely perform operations on encrypted data without ever decrypting them. More precisely, a (group) homomorphic encryption scheme over a group

$$G(*)$$

satisfies that given two encryptions c1 = Ek(m1) and c2 = Ek(m2), where m1, m2 $\in$ G and k is the encryption key, one can efficiently compute Ek(m1 $*$ m2) without decrypting c1 and c2. Homomorphic encryption schemes are widely used in many interesting applications, such as private information retrieval, electronic voting , multiparty computation , and cloud computing etc. Generally, fully homomorphic encryption schemes that support two operations over the underlying algebraic structure, i.e., addition and multiplication, will benefit more problems with different notions of security and cost. The possibilities of homomorphic encryption were first explored by *Rivest, Adleman, and Dertouzos* ,as the **Unpadded RSA** and **ElGamal** are both **partially** homomorphic , the recent research mainly pioneered *Craig Gentry* who introduced the first lattice-based fully homomorphic scheme .

The main issue with the known fully (or partially) homomorphic encryption schemes is the high computational complexity and large communication cost required for their execution,but this issue can be fixed by using bootstrapping as Gentry showed .

A Fully Homomorphic scheme unlike Partial permits to construct arbitrary computations using circuits to do the latter two primary operations are needed addition and multiplication a partial scheme only permits one over two encrypted inputs,

Research into the subject showed multiple fully homomorphic schemes that are both feasible in time and computational requirements,thus the use of the current implementations is possible,but the development team is currently reviewing its own implementations of the schemes to make it compatible with the platform evolution .

A homomorphic scheme provides a very powerful option the possibility of operating on the private domain alas a few performance wise problems can be met when these operations are non arithmetic or non total such example comes often in deep learning the Sigmoid function which is a non-linear bounded function use the exponential,a function that itself can't be calculated precisely but often approximated algorithmically ,this and other types of calculations that are met in deep learning can be error prone but it's proven that 8-bit numbers are enough for stable models thus the current methods of approximation are good enough to be used although in some schemes as the one proposed by Gentry the small errors can interfere with the result when repeated often alas the bootstrapping method mentioned above "refreshes" the circuits to reduce this error .

Microsoft,SEAL library used in the *"CryptoNets"* paper shows that when CryptoNets are applied to the **MNIST** dataset, an accuracy of **99%** can be achieved with a throughput of 58982 predictions per hour on a single PC, and a latency of 250 seconds. Note that a single prediction takes 250 seconds to complete, however, the same process can make 4096 predictions simultaneously with no added cost. Therefore, over an hour, our implementation can make 58982 predictions on average.

We name 3 implementations that are favorable for the current use case :

- *Brakerski-Gentry-Vaikuntanathan* (BGV) Scheme

- *Paillier* Scheme
- *YASHE* (Yet Another Somewhat Homomorphic Encryption) Scheme
- *AONO* Scheme

- **Alternative Processes and Multi Party Computation**

Homomorphic encryption and secure multi party computation are equal processes where the goal is to compute over a set of encrypted inputs but when Homomorphic Encryption is more computationally expensive and interaction is cheaper , a MPC scheme requires a more significant interaction but less computation since the work is divided by multiple parties ,MPC is used currently as a service provided by multiple companies .MPC is different than HE ,in a MPC an input is divided into shares using a secret sharing scheme,it splits an input into n shares in such a way that if anyone sees less than the n shares, then nothing at all is revealed about the value; yet, by seeing all three shares, the value can easily be reconstructed.

Recovering the value becomes only an operation of reordering those shares ,for example company A shares a dataset D to N users each has a part of it after training a model each user sends the respective weights learned the final model is reconstructed using these weights and dataset D is never revealed as the network is obscure since each participant can't know the other one in a simple manner,an attack against this scheme is only possible if all the N users are malicious and coordinated .

The scheme hessian network will choose will depend on the multiple benchmarks and releases of the platform,this is a very important part of the platform since the concealing of the data and weights is both required a unified process of doing both in a correct and accurate manner .
Inference on encrypted data :

II- Inference and intelligence :

When we talk about artificial intelligence we talk about machine learning or deep learning the first is based on statistical analysis and observation,the second is based on learning approximate mapping and probabilistic functions over a dataset . Deep Learning is based on a simple idea we try to learn a function f(x) = y by approximating f using z(x) such as:

$$\varepsilon(x) = F(x) - Z(x) \sim \varepsilon$$

this is done by optimizing a loss function J .

Deep Learning methods are mainly operations on matrices multiplication and addition at their core ,*backpropagation* is then used to optimize the a loss function that evaluate the predictions with their truth,let's see a very simple example .

Logistic Regression is a classification "algorithm" ,we have a dataset X and it's truth value Y the dataset is composed from 2 classes $X' \,\&\, X''$ to separate between them we need to learn a function that give Xi will decide if it's of class 1 or class 2.

- *Input*
- *Layers* $\ddot{Y} = W * x + \beta$ *where* W, β *are respectively the weights and bias of the network*
- *Ouput or prediction*

The *nonlinearity* of the separation is introduced using the sigmoid function :

$$S(x) = 1/(1 + e^{-X})$$

b. Feasibilty

Inference on encrypted data becomes a simple process but error-prone since the nonlinearities will need to be approximated using algorithms as an example using Taylor-Maclaurin ,a function

$\varsigma\ (x)$ *can be approximated to a polynomial sum i.e* : $e^x \sim \sum_{0}^{n} (x^i) + \varepsilon$

But as shown by Microsoft Research ,Cryptonets were able to achieve 99% accuracy on MNIST (a standard dataset for evaluation deep architectures) this breakthrough was convincing enough for us to start our research and present an solution that rewards users alike .

III. Extending Models to Metamodels

Using *ensembling techniques* such as B*ayesian optimal classifier* ,B*ayesian averaging,Bagging or Stacking* , the user submitted models can be “unified” to create a predictive model with better accuracy the process is similar to what *swarm-intelligence* or *hive mind ideas* described An ensemble is a supervised learning algorithm, because it can be trained and then used to make predictions. The trained ensemble, therefore, represents a single hypothesis. This hypothesis, however, is not necessarily contained within the hypothesis space of the models from which it is built. Thus, ensembles can be shown to have more flexibility in the functions they can represent. This flexibility can, in theory, enable them to overfit the training data more than a single model would, but in practice, some ensemble techniques (especially bagging) tend to reduce problems related to over-fitting of the training data.

Empirically, ensembles tend to yield better results when there is a significant diversity among the models. Many ensemble methods, therefore, seek to promote diversity among the models they combine. Although perhaps non-intuitive, more random algorithms (like random decision trees) can be used to produce a stronger ensemble than very deliberate algorithms (like entropy-reducing decision trees).[] Using a variety of strong learning algorithms, however, has been shown to be more effective than using techniques that attempt to *dumb-down* the models in order to promote diversity.

This process is meta-heuristic and is used extensively in competitions and real applications it permits to extend the capabilities of predictive model by averaging on multiple ones such as a robust random forest can balance the overfitting problem caused by a polynomial logistic regression .
This step is aimed towards companies and users alike ,a company can select top-k submitted models and ensemble them into a more powerful one thus effectively creating a superior intelligence .

4. Conclusion:

We presented here 3 main components of the Hessian Network the fourth components smart-contracts will be discussed in the next paper that we plan to release,these introduction were dense and clear and written to be read and understood by any type of readers for more informations you are always welcome to contact us directly .

The paper didn’t introduce the Hessian Token mechanism that rewards the parties of the network because it will be presented in a future paper,in this one we tried to condense multiple fields and domains into a unified resource that explains each part of the Hessian Network .

Our goal is to provide a new way of sharing and using data without violating privacy and security of humans

Ressources :

- **CryptoNets paper .**
- **Gentry Scheme.**
- **Homomorphic Encryption.**
- **Multi Party Computation .**
- **Deep Learning .**