Homomorphic Encryption allows addition and multiplication to be done on encrypted data. Neural networks can be implemented using addition, multiplication and a non-linearity – a acticatioon function.
The most commonly used activatoin fucntions are :

- Sigmoid Function
- Relu Function
- TanH Function

Lets look into Sigmoid Function:

f(x) = 1/(1+e^-x)

Now remenber when we said, a homomorphic function can only contain addition and multiplication, well the sigmoid function contains a division, and an exponential, which is unfortunately, not compatiabe with homomorphic functions. So is there a way to form the sigmoid function based entirely on addition and multiplication?

Turns out, there is. Taylor series expanson helps us to approximate all functions in terms of higher degree polynomials of that variable.
The taylor series expansion of the sigmoid function turns out to be as follows:

1/2 + (1/4)x – (1/48)x^3 + (1/480)x^5

f(x) = x if x > 0
      = 0 if x <= 0

As you see Relu has the if-else statement which, while very simple for us and to code, is a problem when needed to be expressed as a polynomial. The researchers for this paper came up with a work-around. We see the derievative of Relu seems similar to that if ssigmoid function over a large interval. Lets look at them.

We have the derievative of a function. How do we get to the actual function? We integrate that function. So if we you lost track of what s happening, here,

We have derievative of Relu which is actually the sigmoid function .
Now to approximate the Relu function , we integrate it's derievative ie. We integrate the sigmoid function .

We have two forms of sigmoid function with us, the original sigmoid function and one is the approximatated one we got using Taylor Series expansion. Which one do we use?