Neural Network! That's Great ! What is that?

- A neural network consists of nodes and connections between the nodes.
- The numbers along each connection represent parameter values that were estimated when this neural network was fit to the data.
- For now, just know that these parameter estimates are analogous to the slope and intercept values that we solve for when we fit a straight line to data.
- Likewise, a neural network starts out with unknown parameter values that are estimated when we fit the neural network to a dataset using a method called backpropagation.
- We've already fit this neural network to this specific dataset, and that means we have already estimated these parameters.
- You may have noticed that some of the nodes have curved lines inside of them. these bent or curved lines are the building blocks for fitting a squiggle to data.
- This specific curved line is called soft plus, which sounds like a brand of toilet paper.
- **Formula** : $\text{softplus}(x) = \ln(1 + e^x)$
- We also could've used this bent line, called ReLU, which is short for rectified linear unit.
- **Formula** : $\text{ReLU}(x) = max(x, 0)$
- or, we could use a sigmoid shape, or any other bent or curved line.
- The curved or bent lines are called activation functions.
- When you build a neural network you have to decide which activation function, or functions, you want to use.
- When most people teach neural networks they use the sigmoid activation function.
- However, in practice, it is much more common to use the ReLU activation function, or the soft plus activation function. so we'll use the soft plus activation function
- Neural networks are usually much fancier and have more than one input node, more than one output node, different layers of nodes between the input and output nodes, and a spider web of connections between each layer of nodes.
- These layers of nodes between the input and output nodes are called hidden layers. when you build a neural network one of the first things you do is decide how many hidden layers you want and how many nodes go into each hidden layer.

Input to Hidden Layer:

For dosage = 0:

To the first hidden node:
$\text{SoftPlus}(-34.4 \times 0 + 2.14)$
Calculation:
$\text{SoftPlus}(2.14) = \log(1 + e^{2.14}) \approx 2.25$

Constructing the Blue Curve:

Dosage 0.1:

Calculation:
$\text{SoftPlus}(-34.4 \times 0.1 + 2.14)$
$y \approx 0.24$
Continue for $x = 0.2, 0.3, \ldots, 1$ to build the blue curve.

Scaling the Blue Curve:

Scale y-values by -1.3:
For $y = 2.25$:
$-2.93 = 2.25 \times -1.3$

Input to Second Hidden Node:

For dosage = 0:

To the second hidden node:
$\text{SoftPlus}(-2.52 \times 0 + 1.29)$
Calculation:
$\text{SoftPlus}(1.29) \approx 1.53$

Constructing the Orange Curve:

Calculate for other dosage values similarly.

Scaling the Orange Curve:

Scale y-values by 2.28.

Combining Curves:

Add blue and orange curves to create the green squiggle.
Adjust with a bias (subtract 0.58).

Final Prediction:

For dosage = 0.5:

Plug into the neural network to find y.
Calculation yields:
$y \approx 1.03$
indicating effectiveness.

Conclusion :

- Neural networks can fit complex data patterns using layers and nodes.
- This simple example illustrates the fundamental mechanics of neural networks and their capability to learn from data.