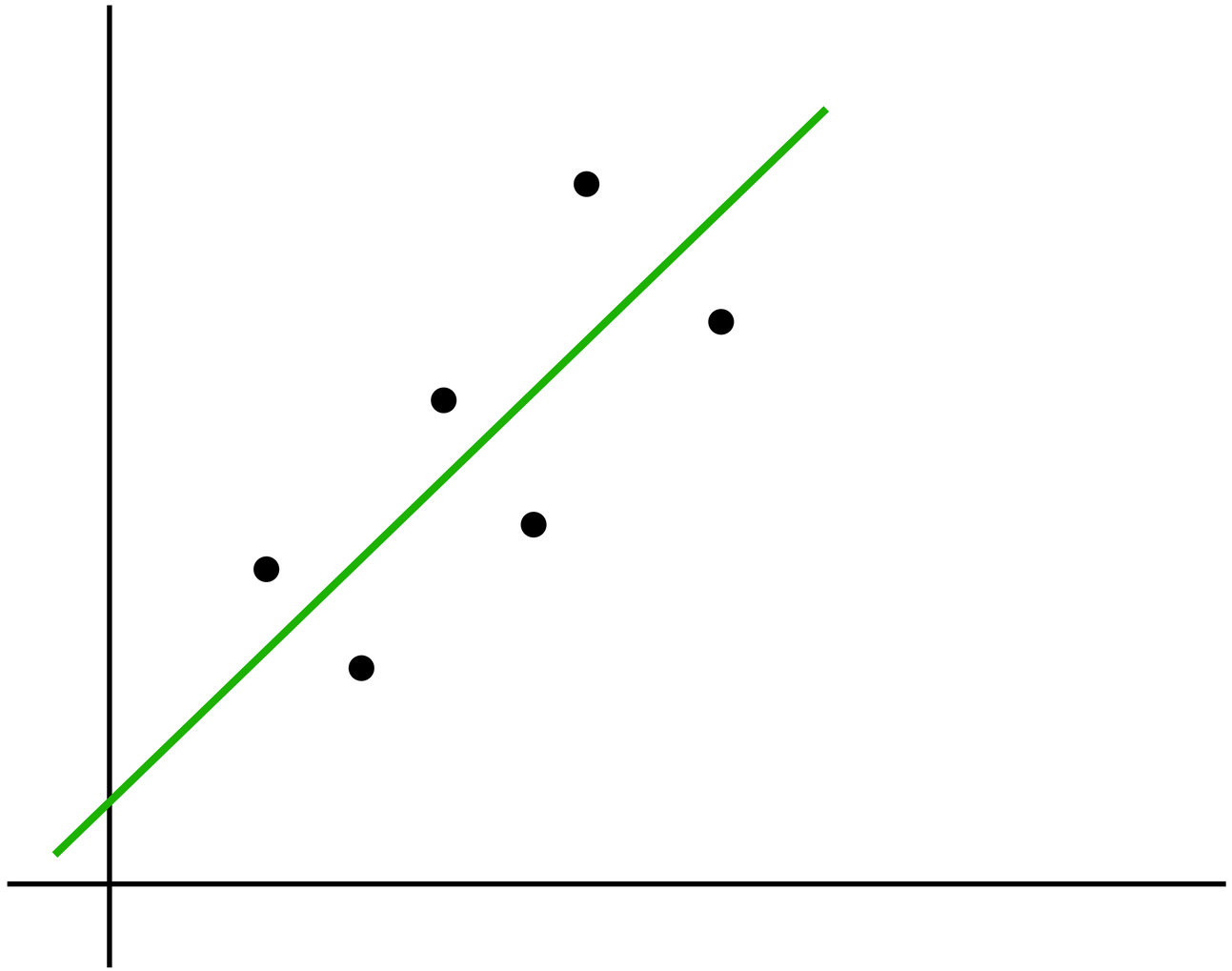
Neural Networks

Introduction to Neural Networks

Rahul Singh
rsingh@arrsingh.com

Simple Linear Regression

$$\hat{y} = \beta_0 + \beta_1 x$$

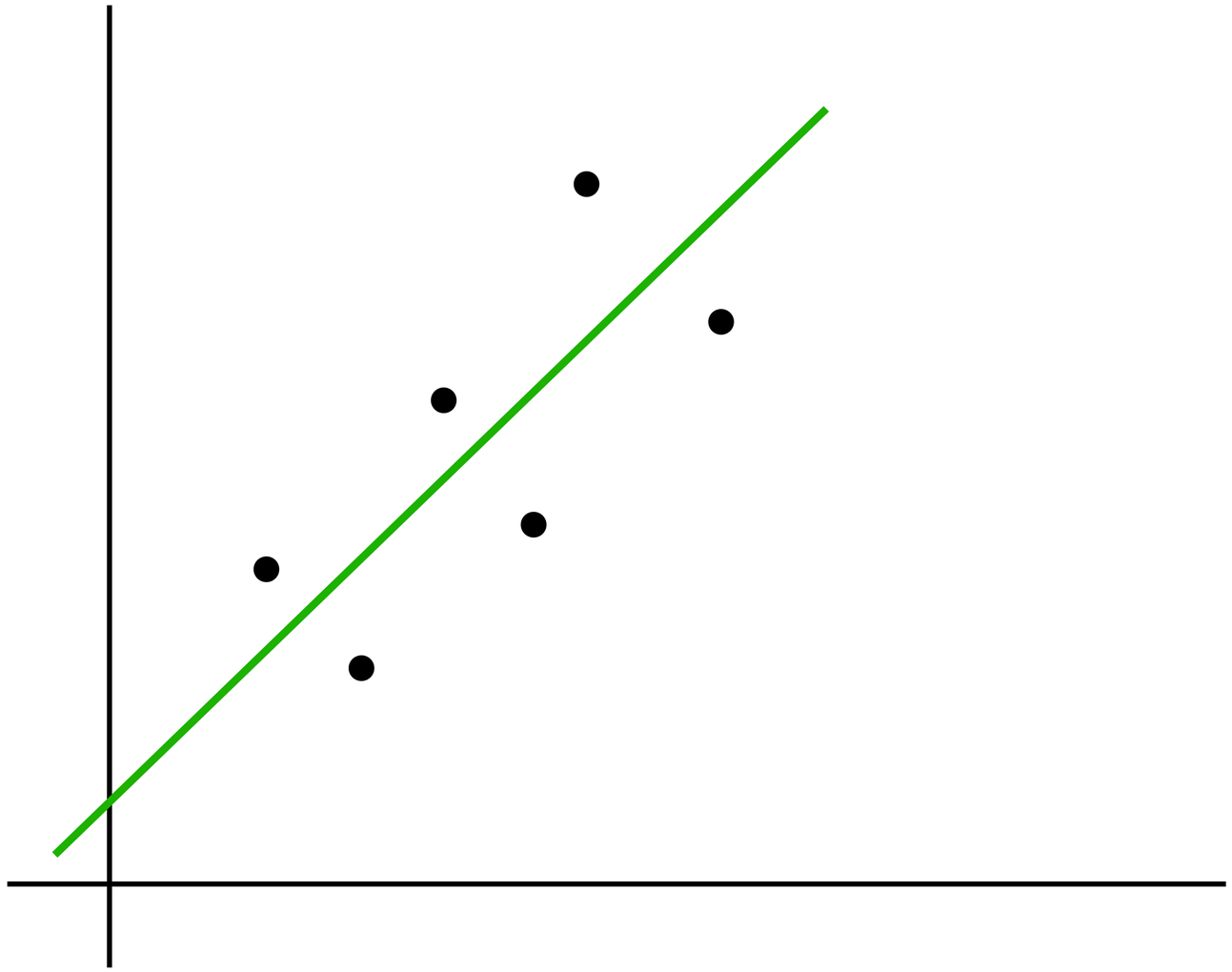


Data is synthetic and not plotted to scale

Simple Linear Regression

1 independent variable

$$\hat{y} = \beta_0 + \beta_1 x$$



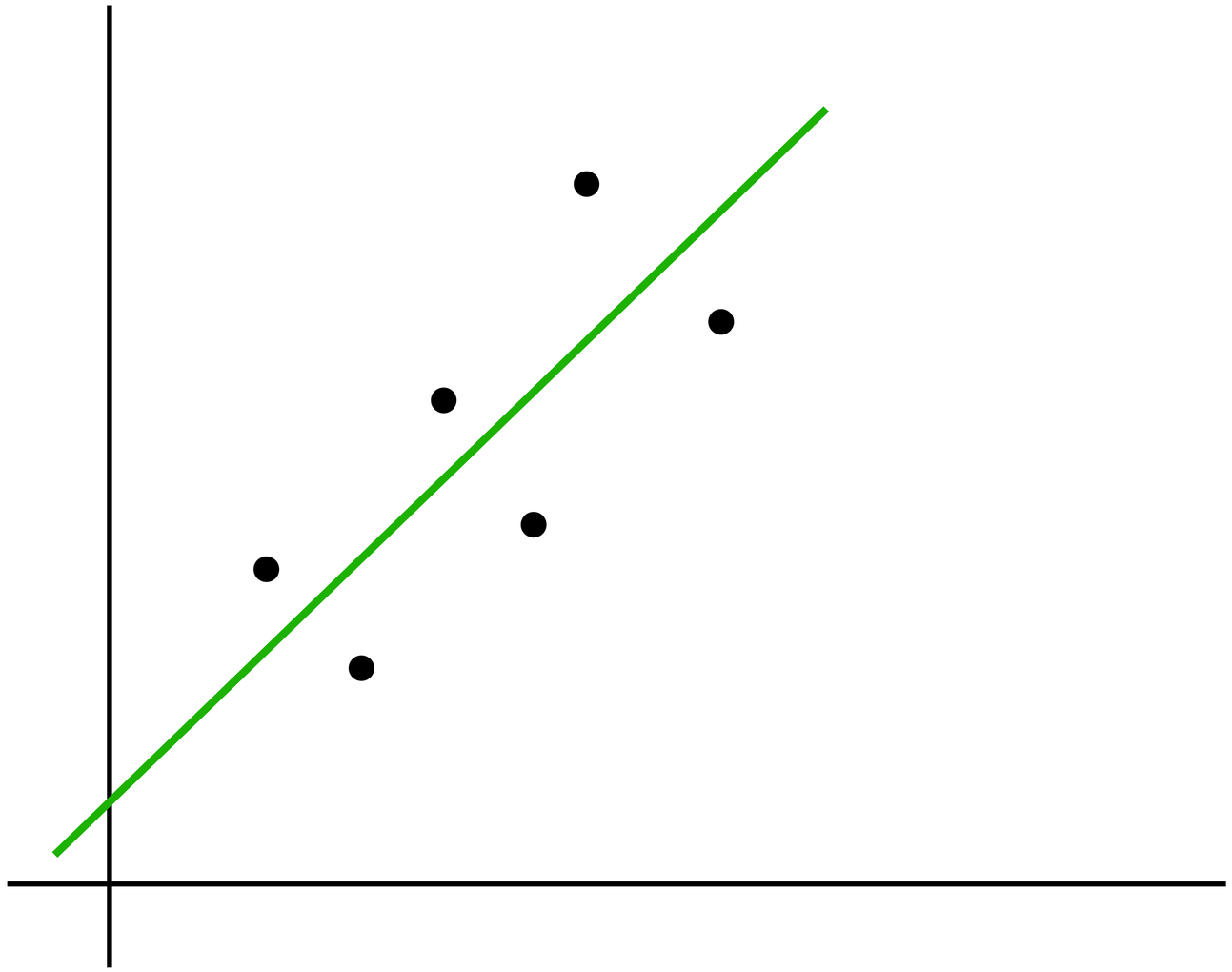
Data is synthetic and not plotted to scale

Simple Linear Regression

1 dependent variable

1 independent variable

$$\hat{y} = \beta_0 + \beta_1 x$$



Data is synthetic and not plotted to scale

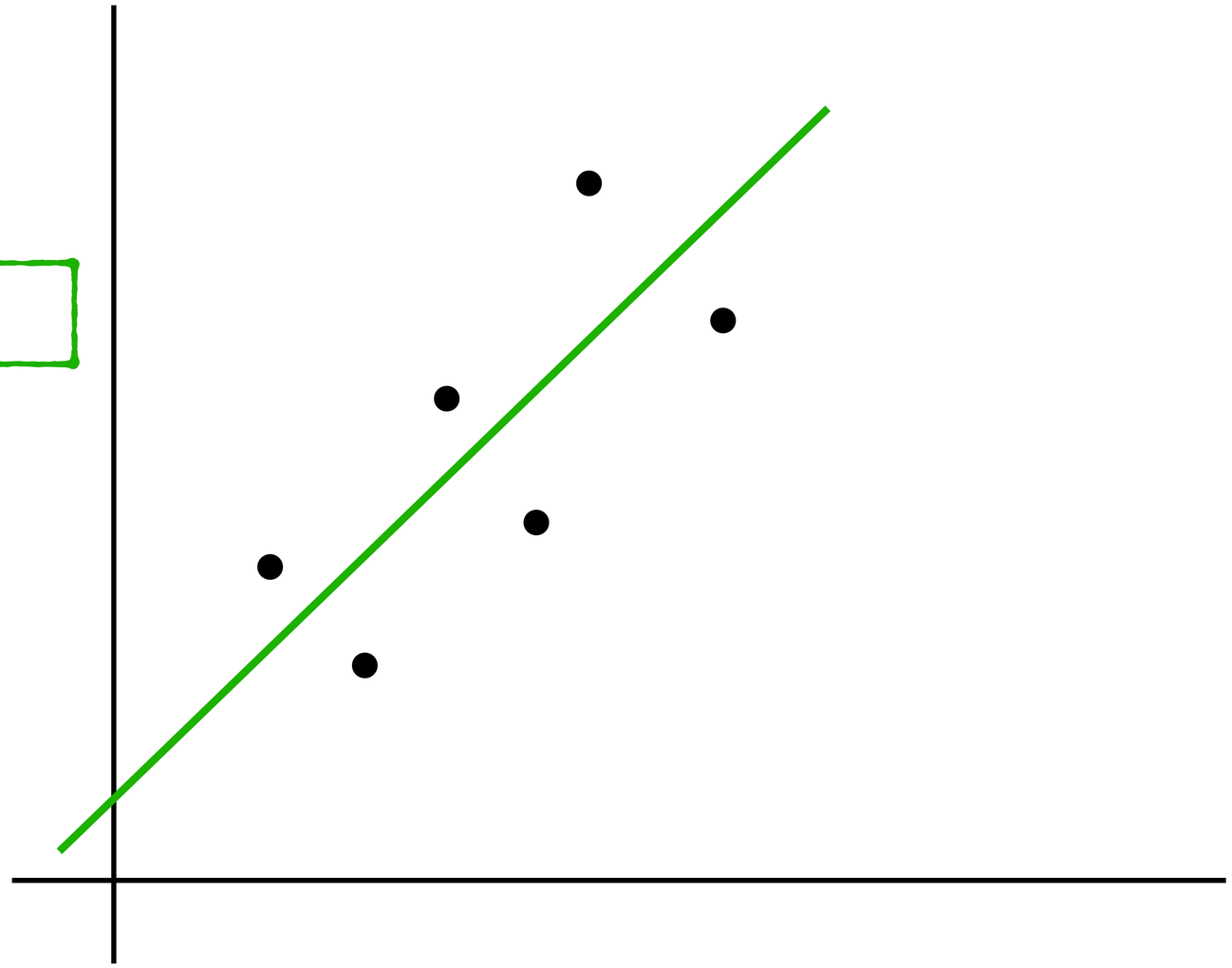
Simple Linear Regression

1 dependent variable

1 independent variable

$$\hat{y} = \beta_0 + \beta_1 x$$

2 Parameters - β_0 and β_1



Data is synthetic and not plotted to scale

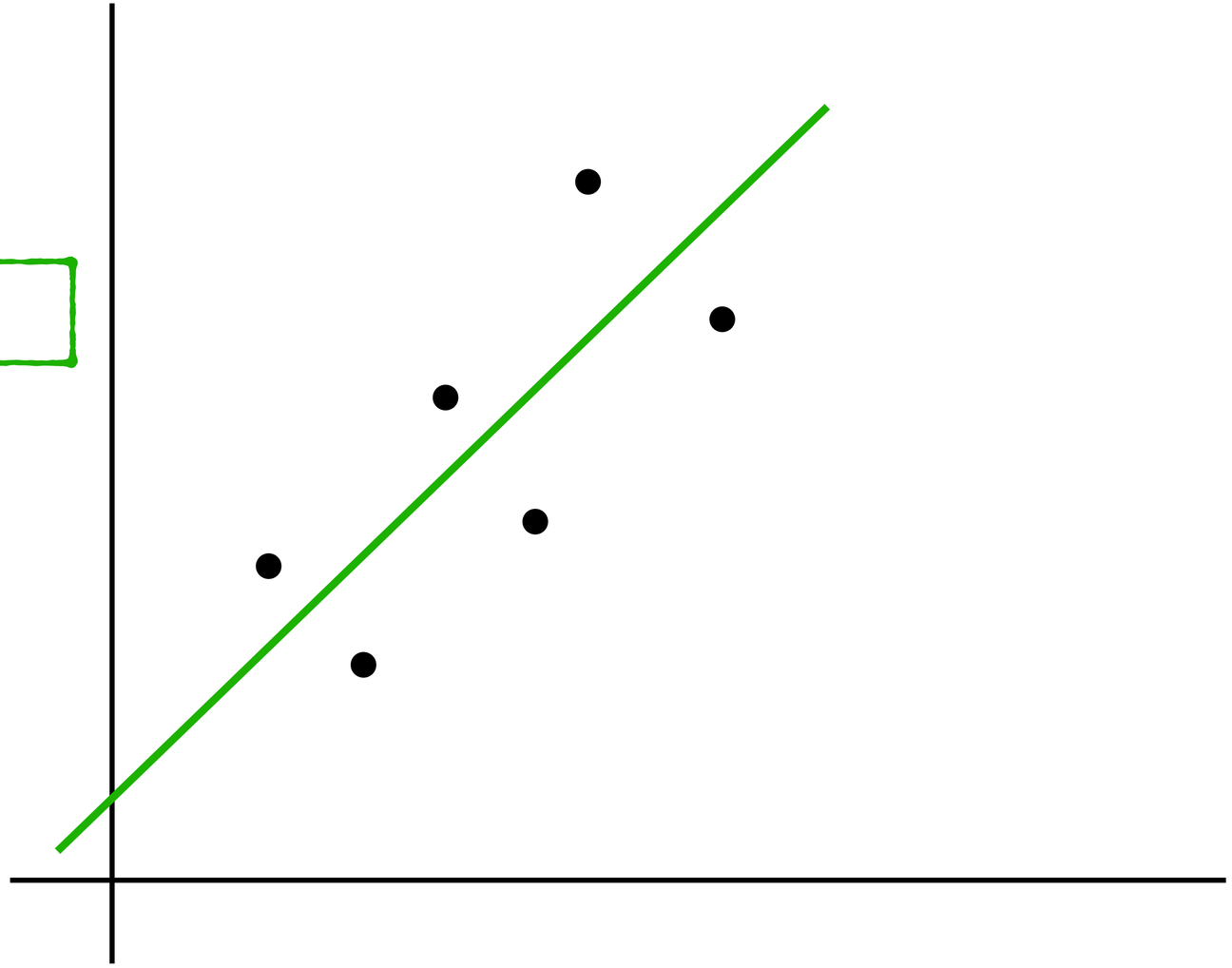
Simple Linear Regression

1 dependent variable

1 independent variable

$$\hat{y} = \beta_0 + \beta_1 x$$

2 Parameters - β_0 and β_1



The Problem Statement:

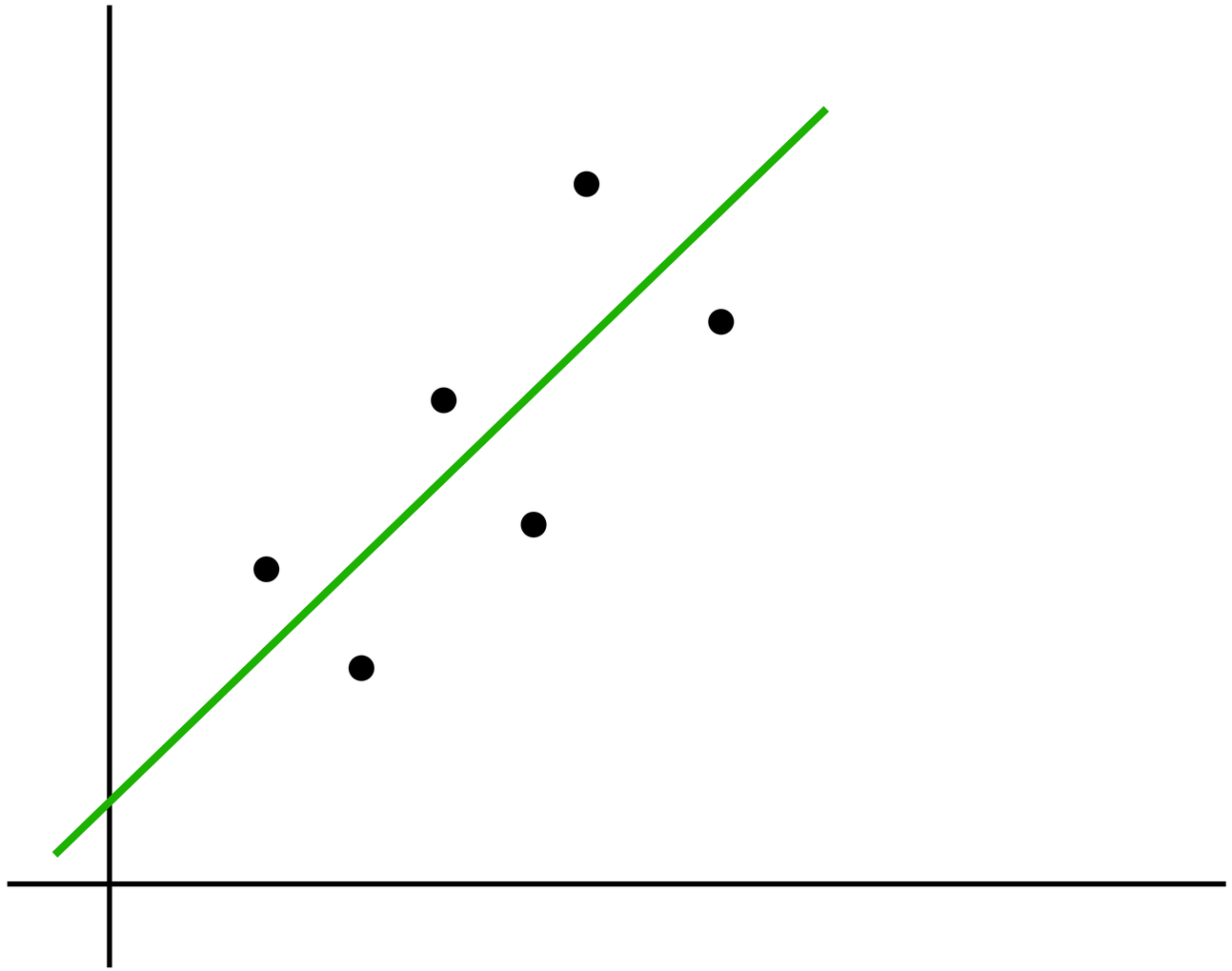
Simple Linear Regression: Find the values of β_0 and β_1 such that the Mean Squared Error (MSE) is minimized.

Data is synthetic and not plotted to scale

Simple Linear Regression

$$\hat{y} = \beta_0 + \beta_1 x$$

Let's rename the parameters…



Data is synthetic and not plotted to scale

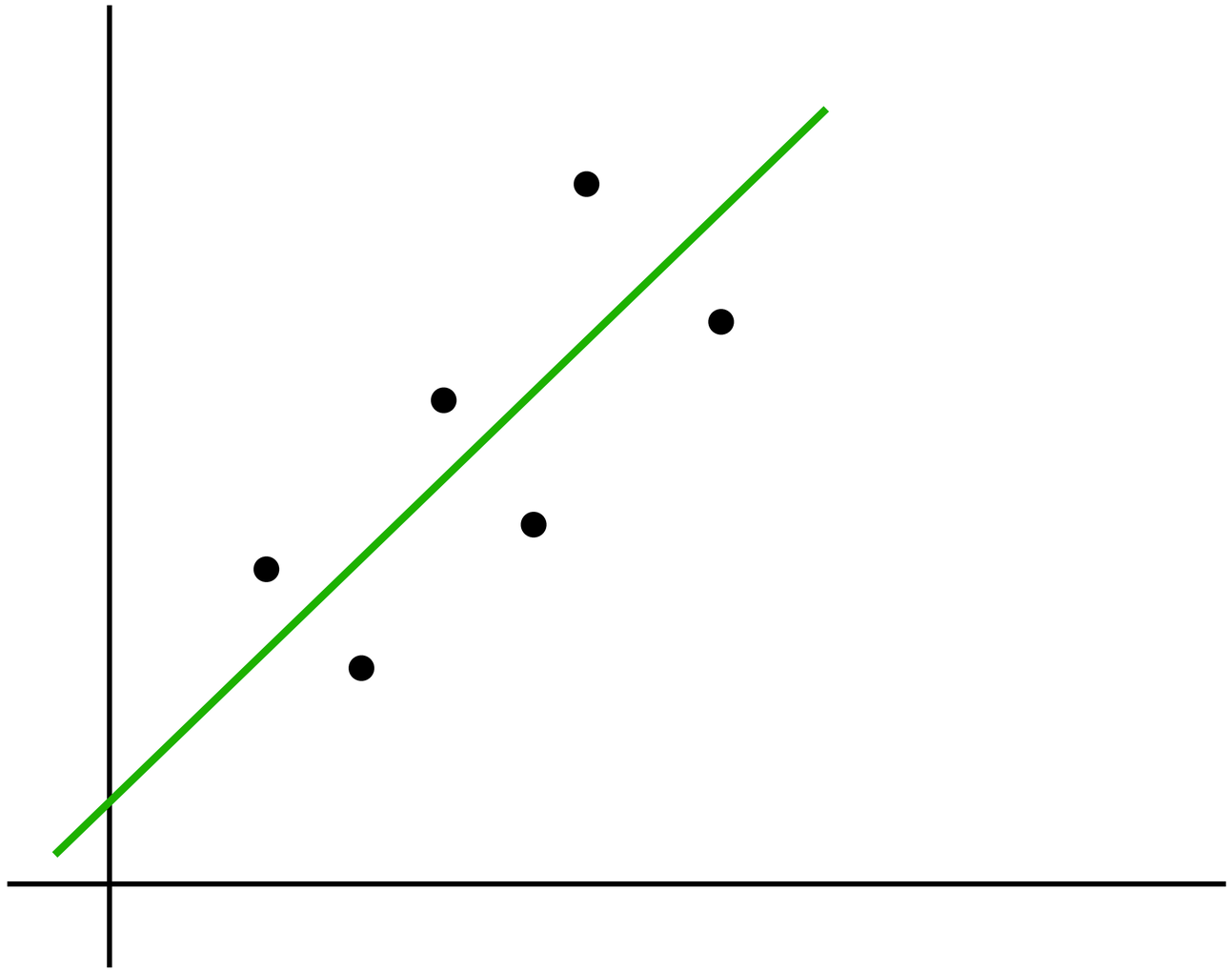
Simple Linear Regression

$$\hat{y} = \beta_0 + \beta_1 x$$

Let's rename the parameters…

Rename β_0 to β

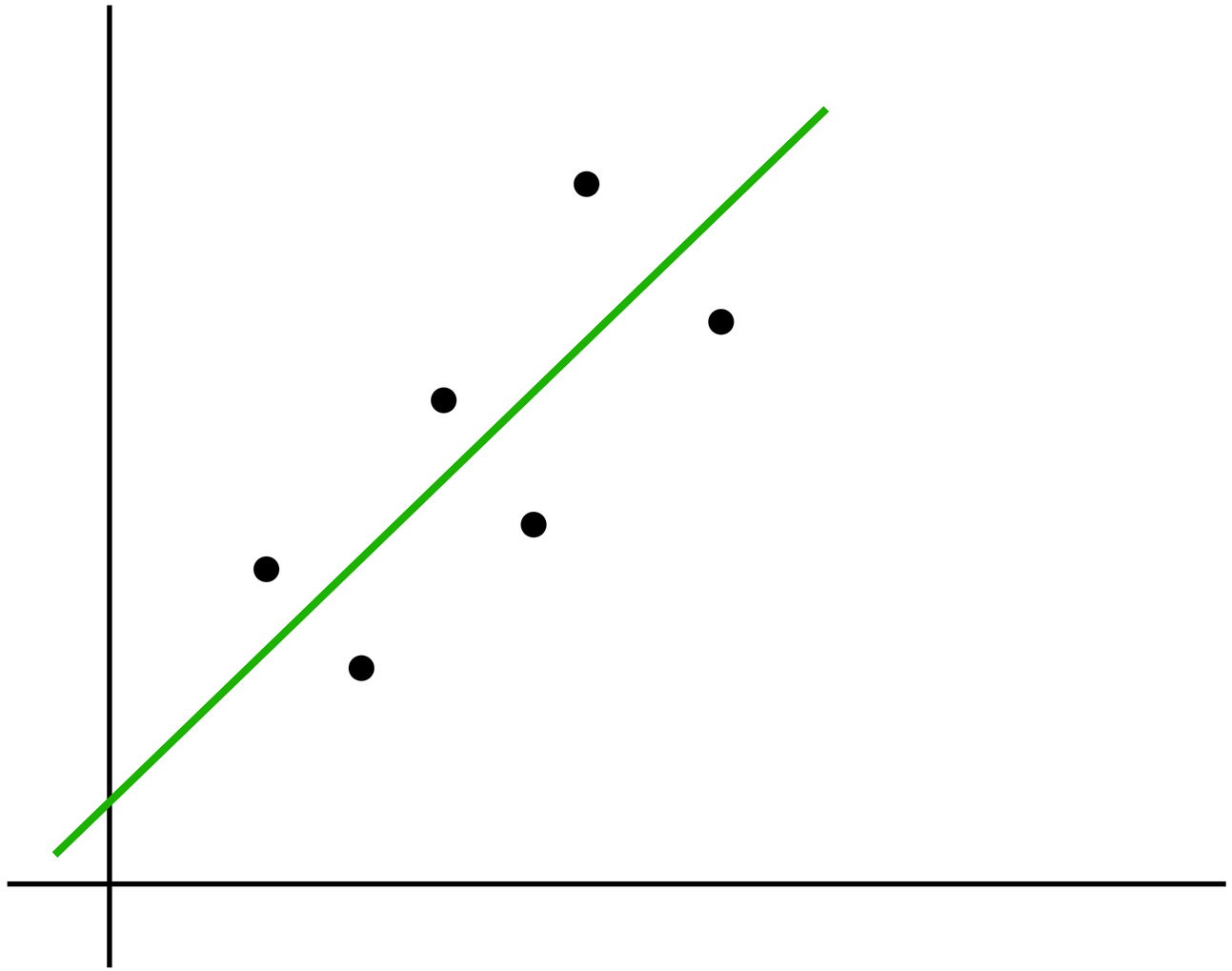
Rename β_1 to w_1



Data is synthetic and not plotted to scale

Simple Linear Regression

$$\hat{y} = \beta + w_1 x$$

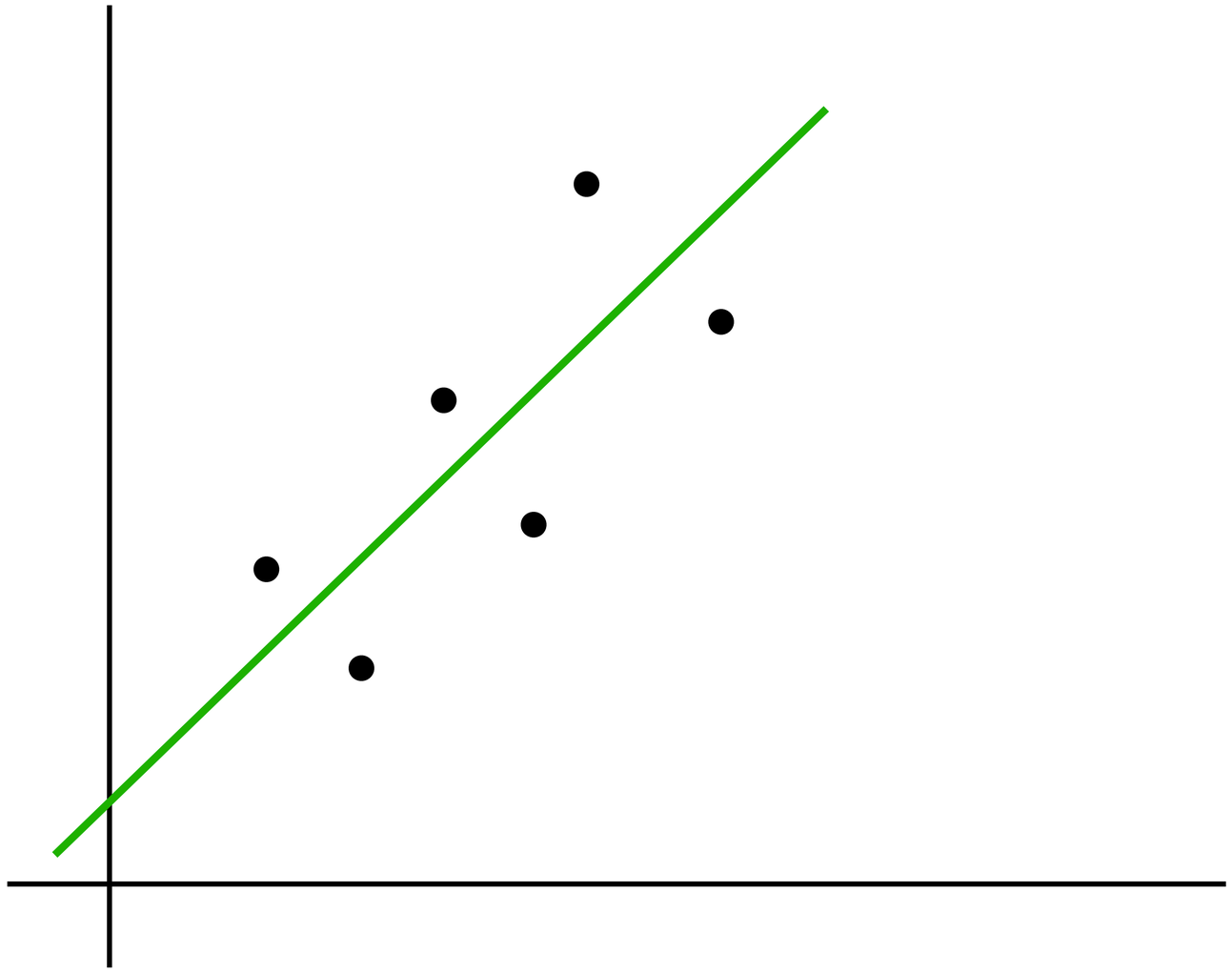


Data is synthetic and not plotted to scale

Simple Linear Regression

1 independent variable

$$\hat{y} = \beta + w_1 x$$



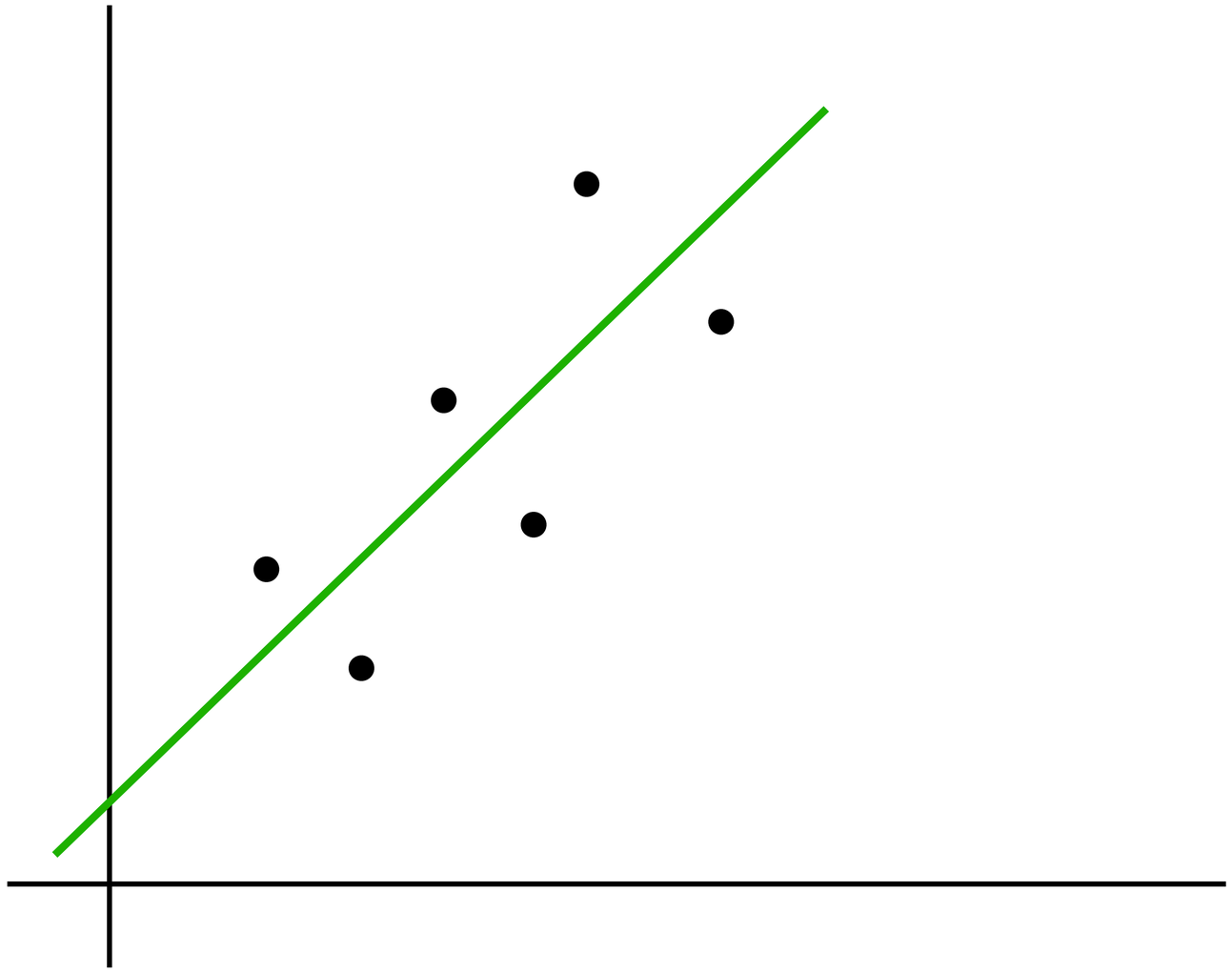
Data is synthetic and not plotted to scale

Simple Linear Regression

1 dependent variable

1 independent variable

$$\hat{y} = \beta + w_1 x$$



Data is synthetic and not plotted to scale

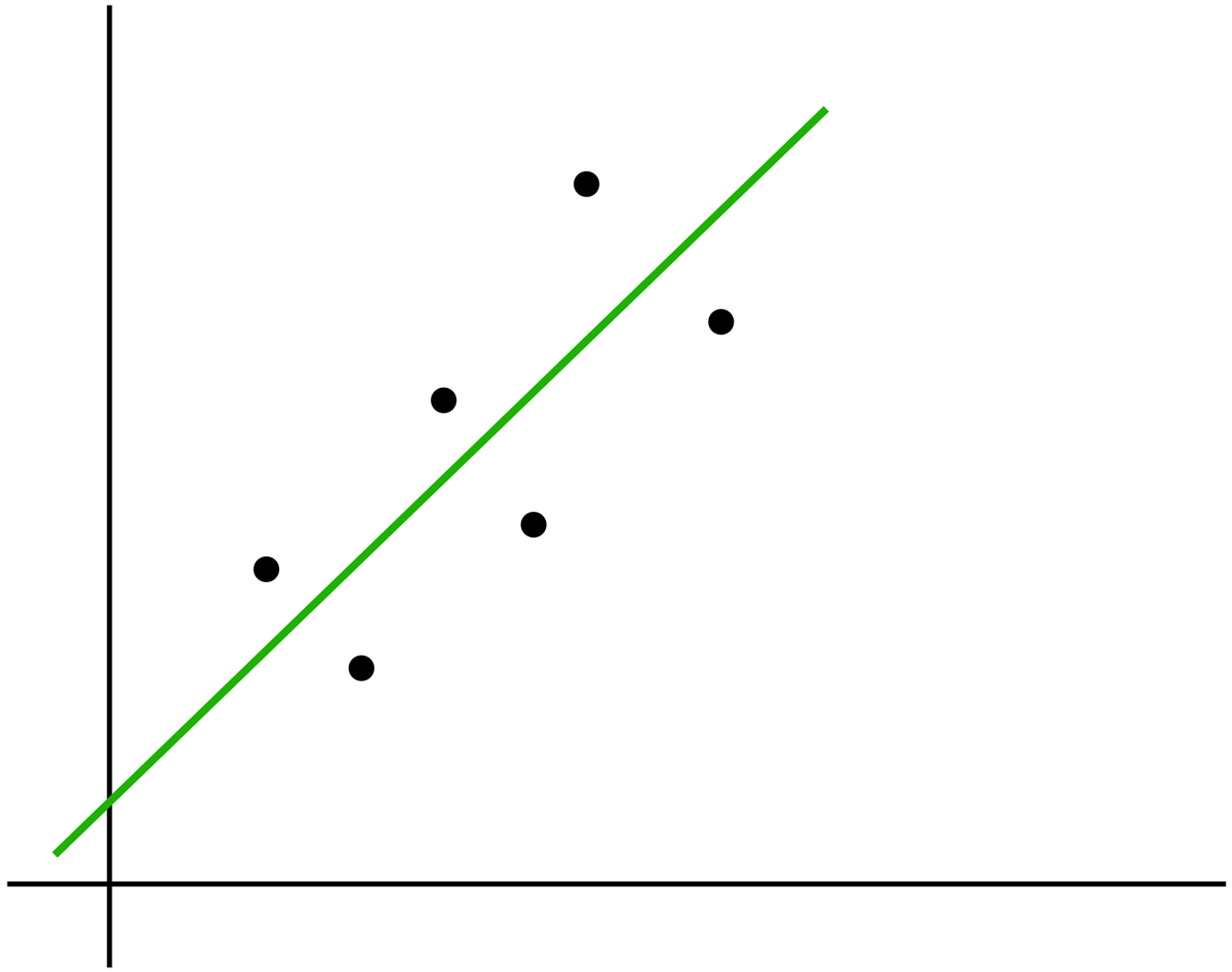
Simple Linear Regression

1 dependent variable

1 independent variable

$$\hat{y} = \beta + w_1 x$$

2 Parameters - β and w_1



Data is synthetic and not plotted to scale

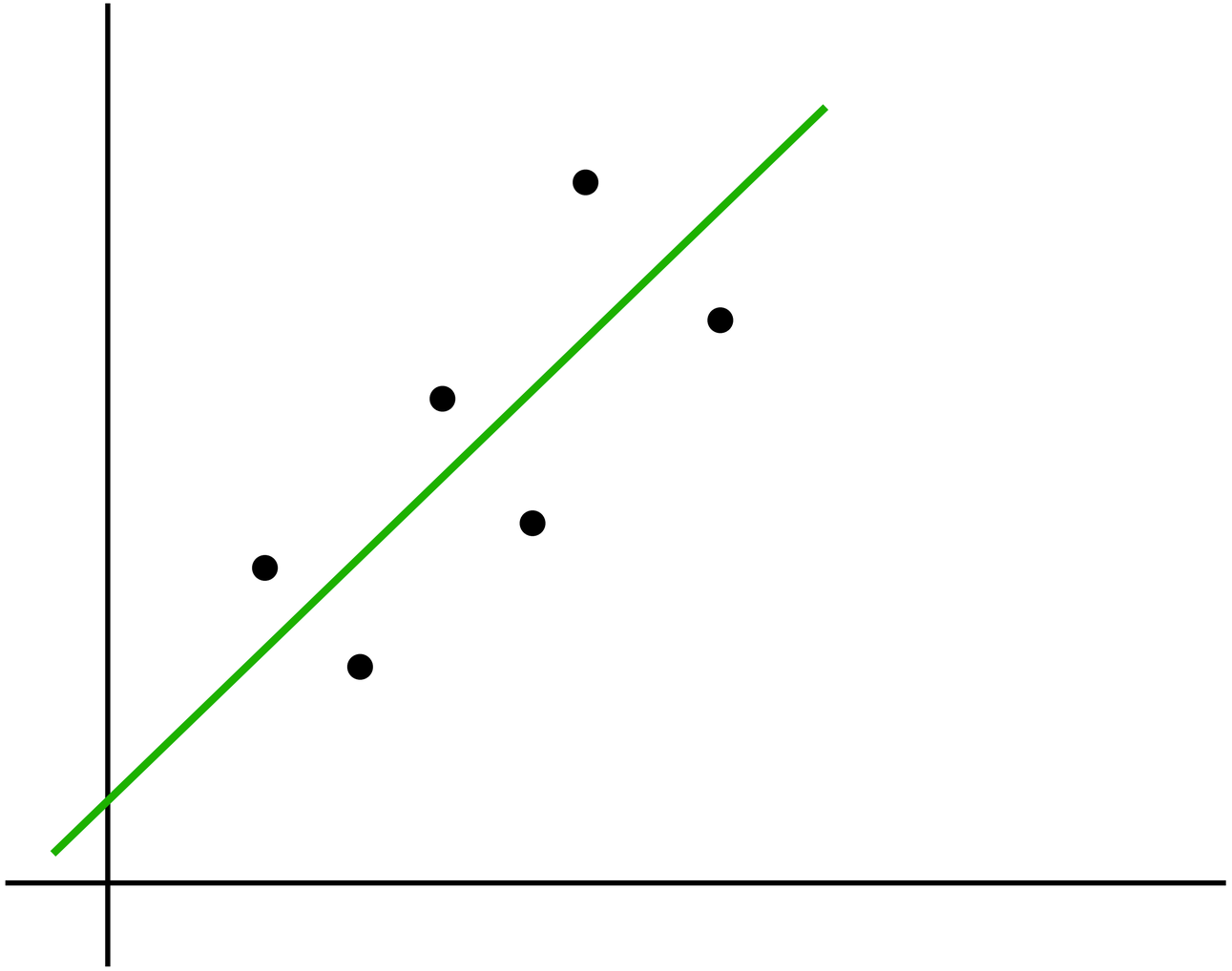
Simple Linear Regression

1 dependent variable

1 independent variable

$$\hat{y} = \beta + w_1 x$$

2 Parameters - β and w_1



The Problem Statement:

Simple Linear Regression: Find the values of β and w_1 such that the Mean Squared Error (MSE) is minimized.

Data is synthetic and not plotted to scale

Simple Linear Regression

$$\hat{y} = \beta + w_1 x$$

This is a function that takes input x and produces an output $\hat{y}$

Data is synthetic and not plotted to scale

Simple Linear Regression

$$\hat{y} = \beta + w_1 x$$

This is a function that takes input x and produces an output $\hat{y}$

We can represent it as a graph structure

Not to be confused with Graph Neural Networks. Here we're talking simply about a visual representation that resembles a graph

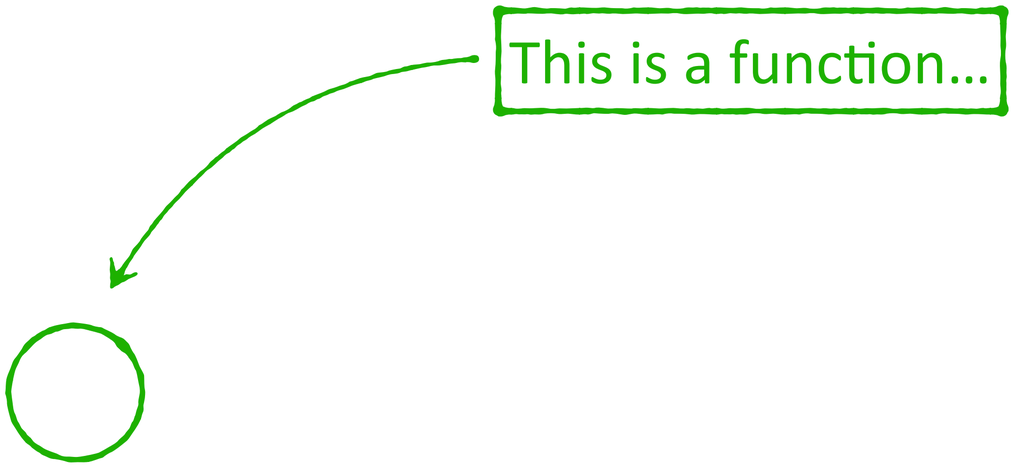
Data is synthetic and not plotted to scale

Neural Networks

$$\hat{y} = \beta + w_1 x$$

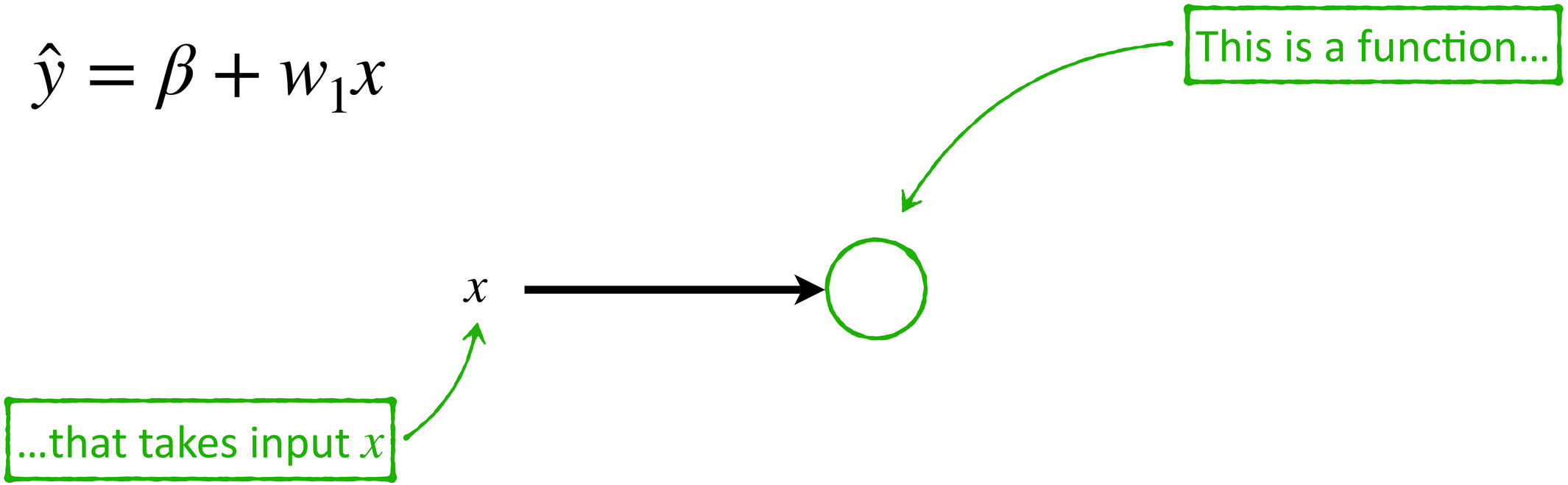
Neural Networks

$$\hat{y} = \beta + w_1 x$$



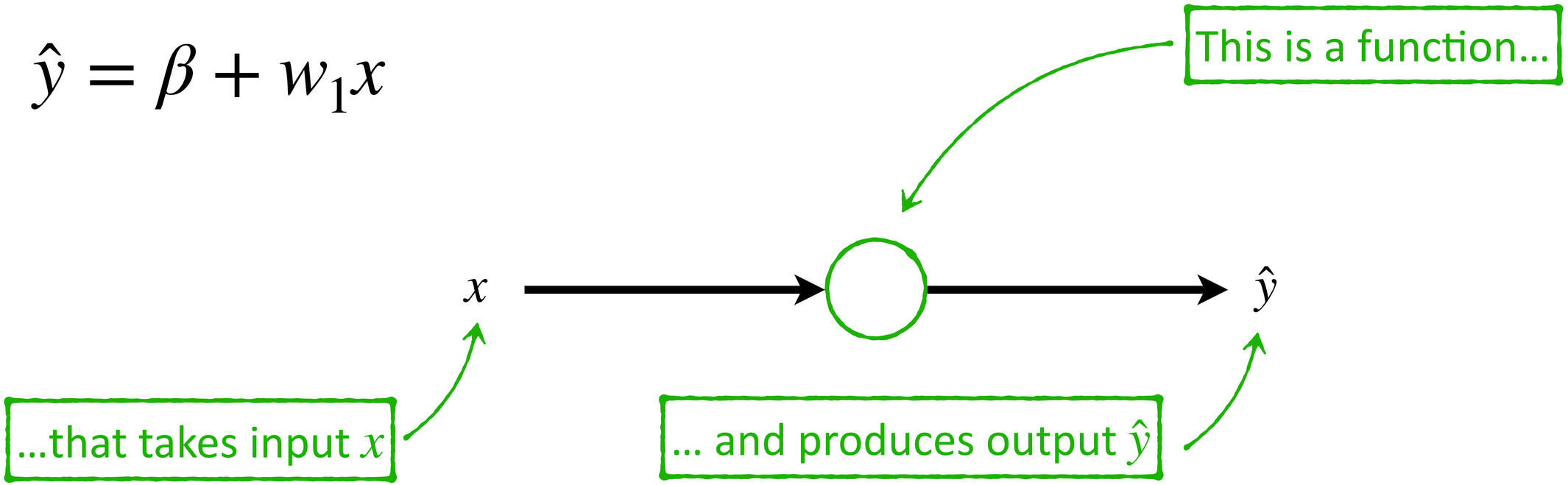
Neural Networks

$$\hat{y} = \beta + w_1 x$$



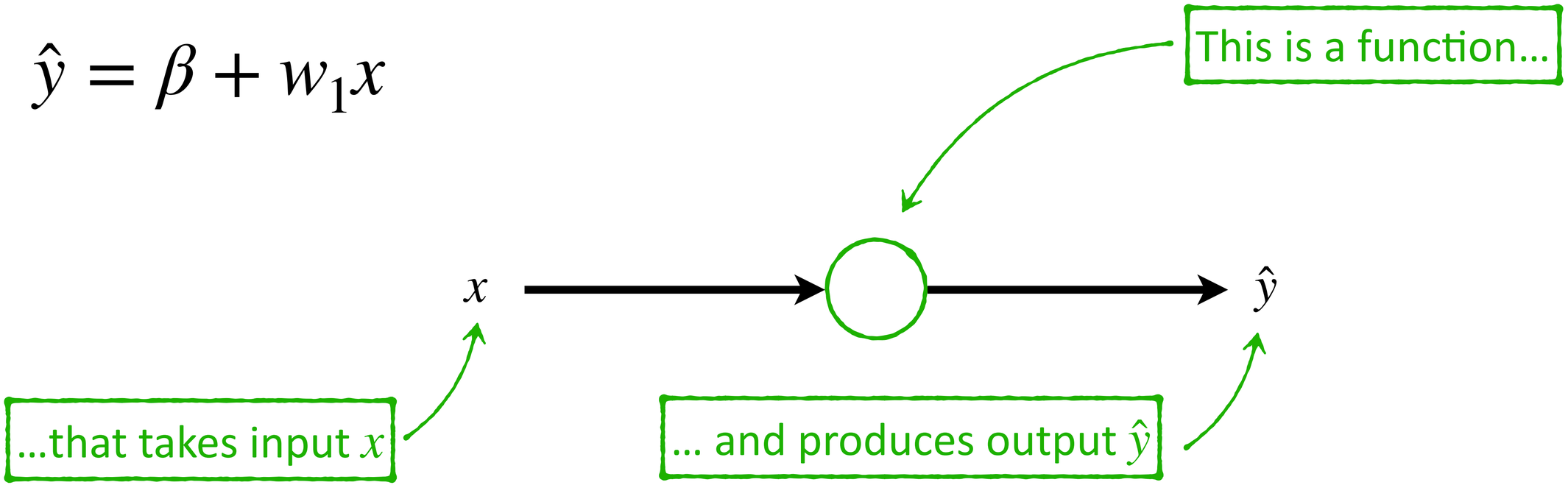
Neural Networks

$$\hat{y} = \beta + w_1 x$$



Neural Networks

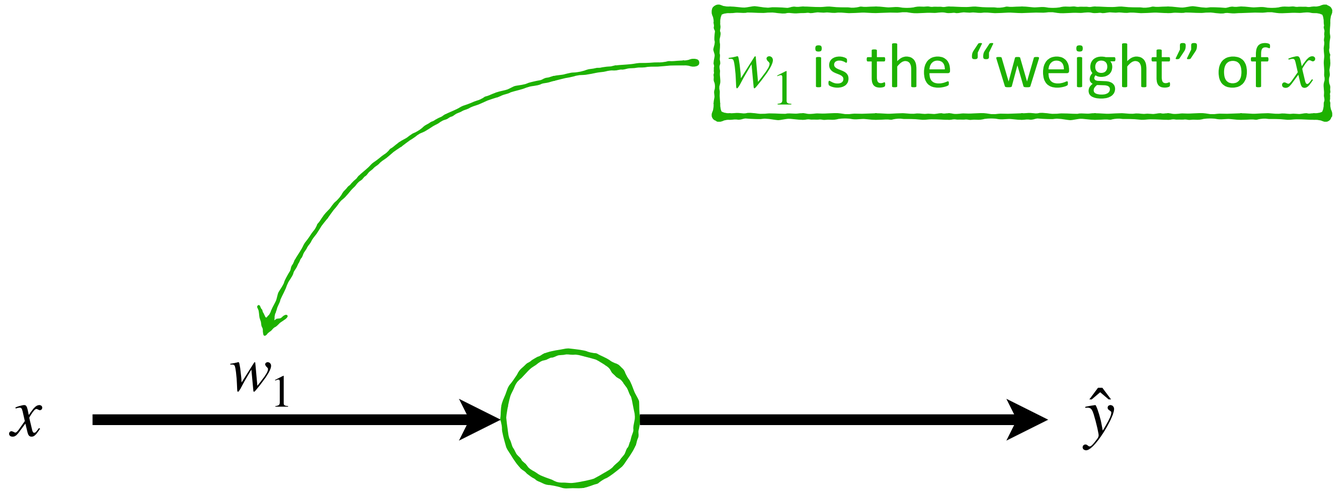
$$\hat{y} = \beta + w_1 x$$



Lets specify the parameters β and w_1

Neural Networks

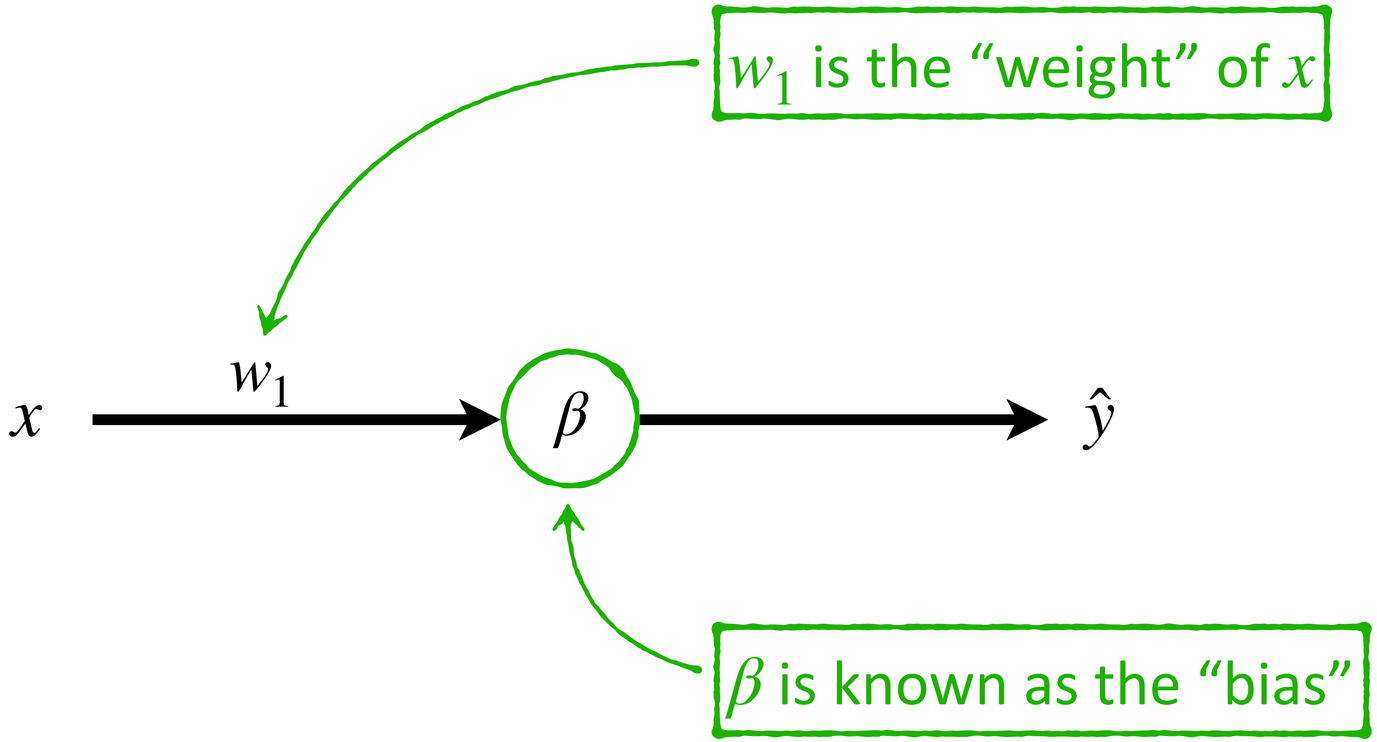
$$\hat{y} = \beta + w_1 x$$



Lets specify the parameters β and w_1

Neural Networks

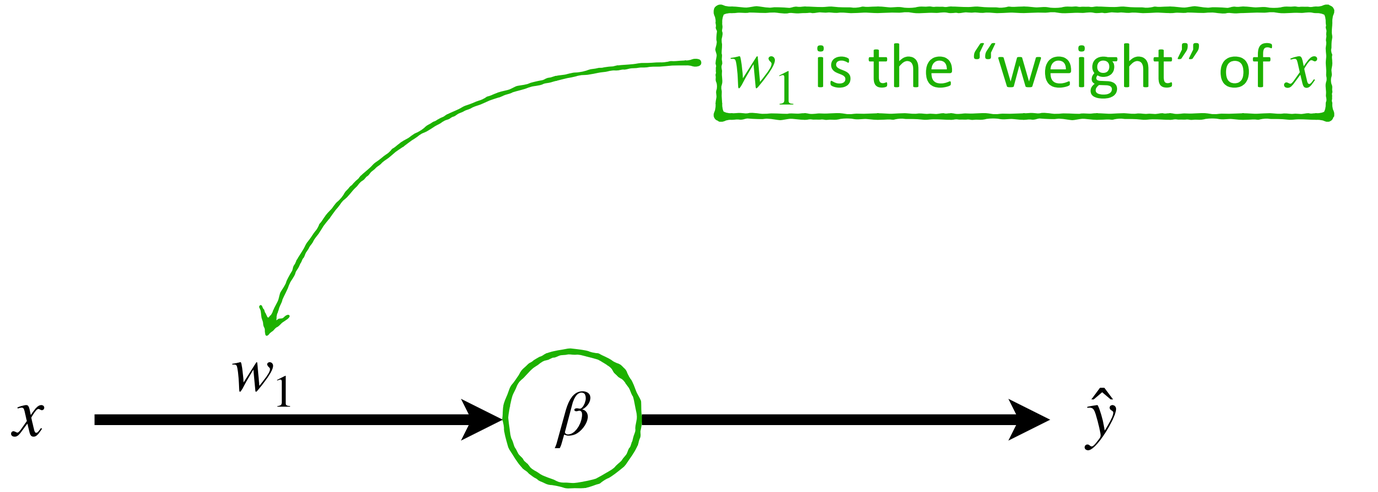
$$\hat{y} = \beta + w_1 x$$



Lets specify the parameters β and w_1

Neural Networks

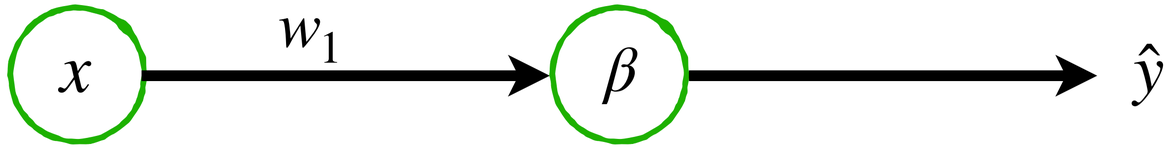
$$\hat{y} = \beta + w_1 x$$



Lets add a circle around the input for consistency

Neural Networks

$$\hat{y} = \beta + w_1 x$$



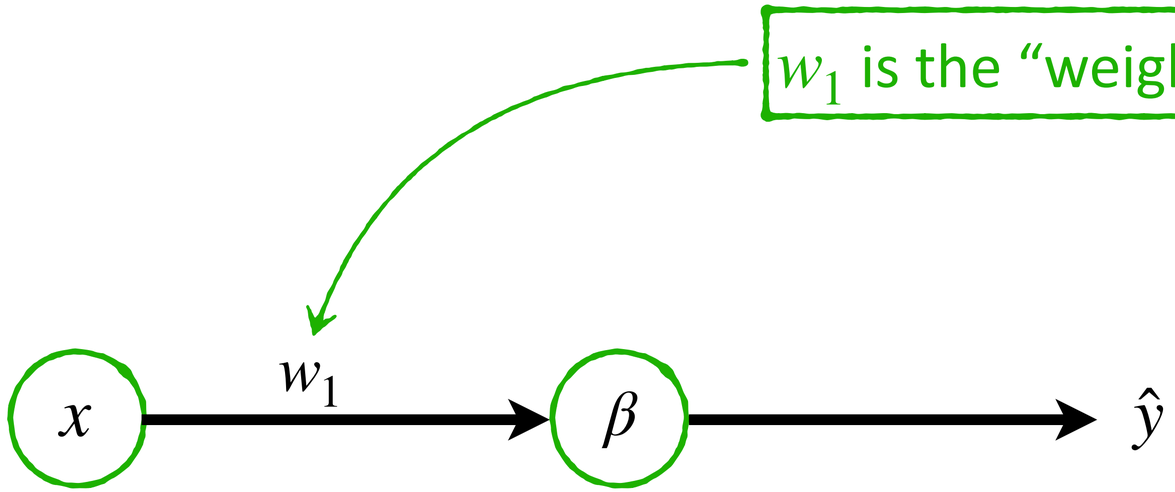
w_1 is the "weight" of x

Lets add a circle around the input for consistency

β is known as the "bias"

Neural Networks

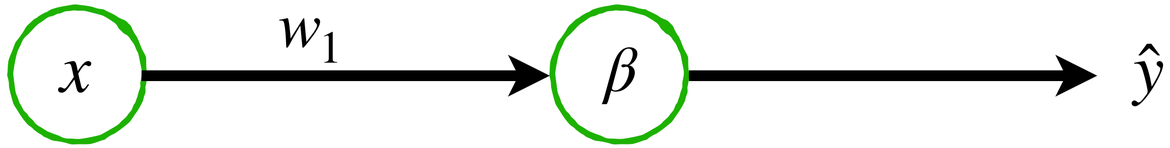
$$\hat{y} = \beta + w_1 x$$



Multiply the input (x) with the Weight (w_1) and add the Bias (β) to compute the output ($\hat{y}$)

Neural Networks

$$\hat{y} = \beta + w_1 x$$



This is the simplest possible neural network

Multiply the input (x) with the Weight (w_1) and add the Bias (β) to compute the output ($\hat{y}$)

Neural Networks

$$\hat{y} = \beta + w_1 x_1 + w_2 x_2$$

Neural Networks

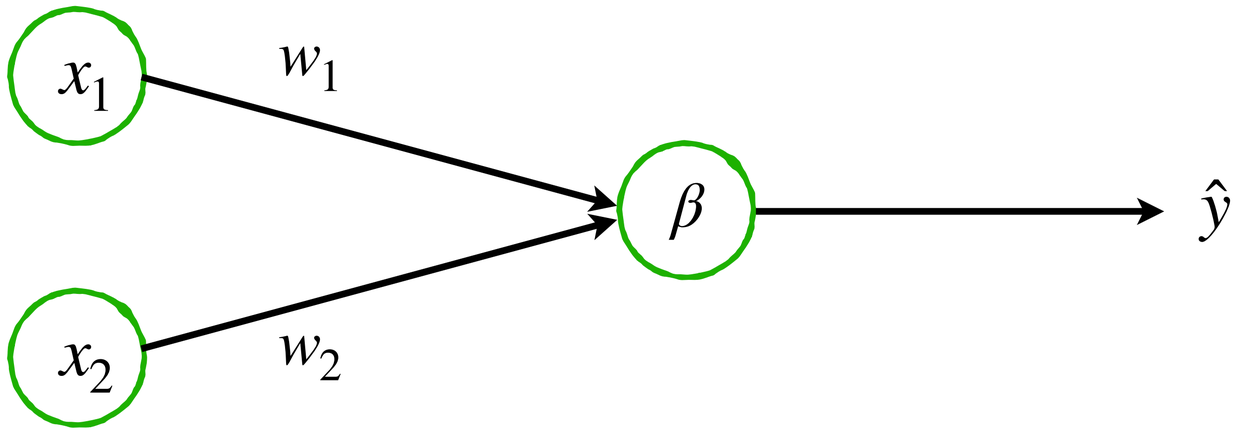
$$\hat{y} = \beta + w_1 x_1 + w_2 x_2$$

Let’s add another input variable (x_2) with its own weight (w_2)

Neural Networks

$$\hat{y} = \beta + w_1 x_1 + w_2 x_2$$

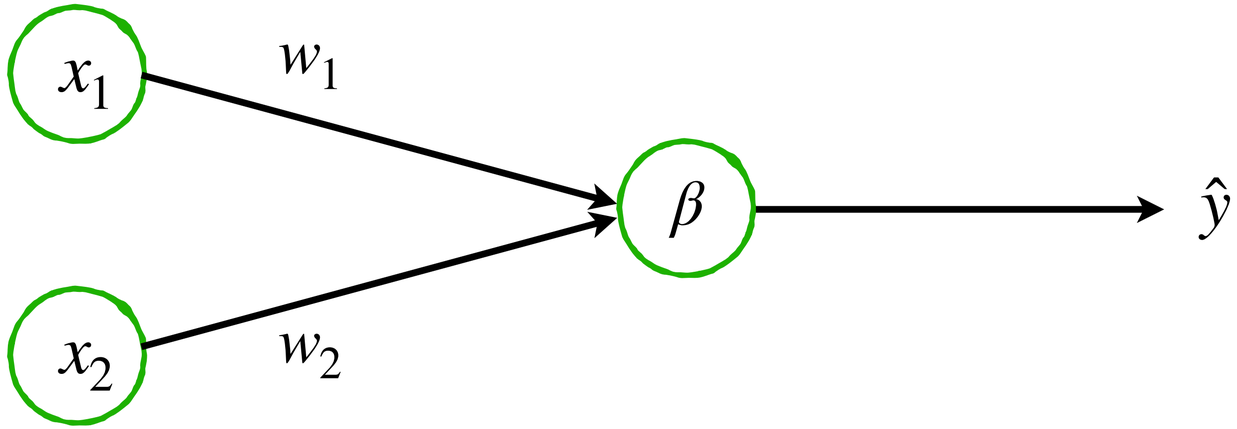
Let's add another input variable (x_2) with its own weight (w_2)



Neural Networks

$$\hat{y} = \beta + w_1 x_1 + w_2 x_2$$

Let's add another input variable (x_2) with its own weight (w_2)



Multiply the inputs (x_1, x_2) with the Weights (w_1, w_2)
and add the Bias (β) to compute the output ($\hat{y}$)

Neural Networks

$$\hat{y} = \beta + w_1 x_1 + w_2 x_2 + w_3 x_3$$

Neural Networks

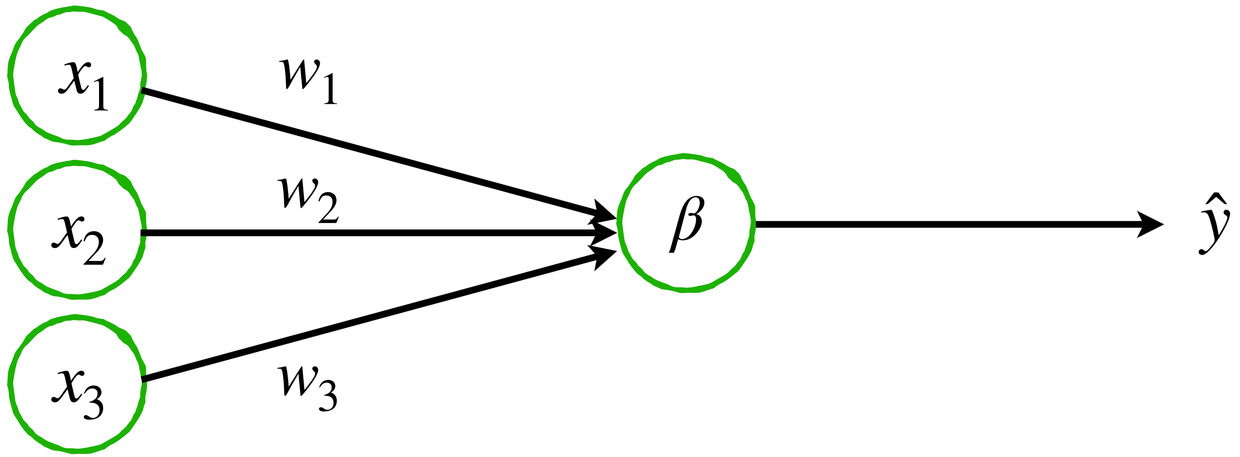
$$\hat{y} = \beta + w_1 x_1 + w_2 x_2 + w_3 x_3$$

Let's add a third input variable (x_3) with its own parameter (w_3)

Neural Networks

$$\hat{y} = \beta + w_1 x_1 + w_2 x_2 + w_3 x_3$$

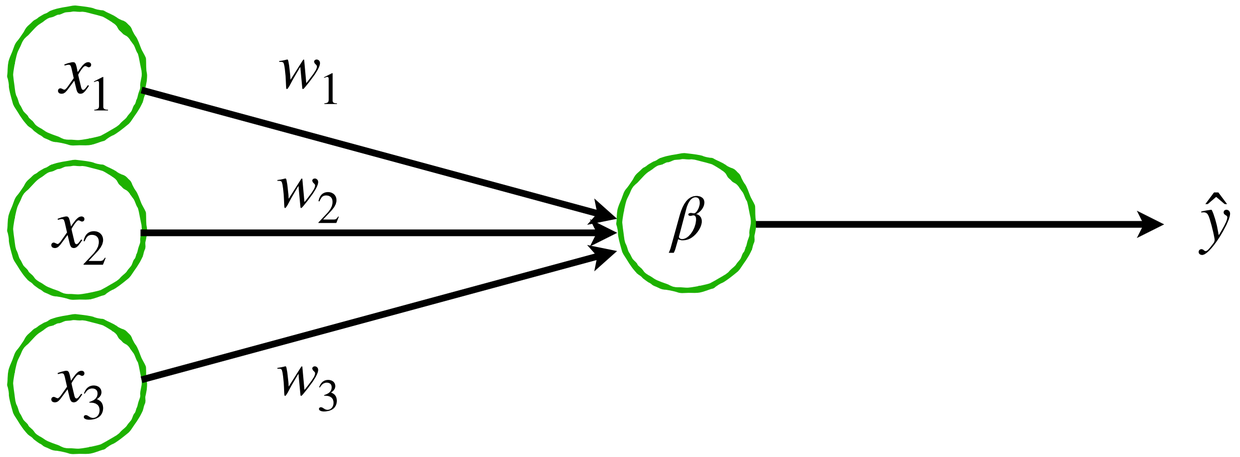
Let's add a third input variable (x_3) with its own parameter (w_3)



Neural Networks

$$\hat{y} = \beta + w_1 x_1 + w_2 x_2 + w_3 x_3$$

Let's add a third input variable (x_3) with its own parameter (w_3)

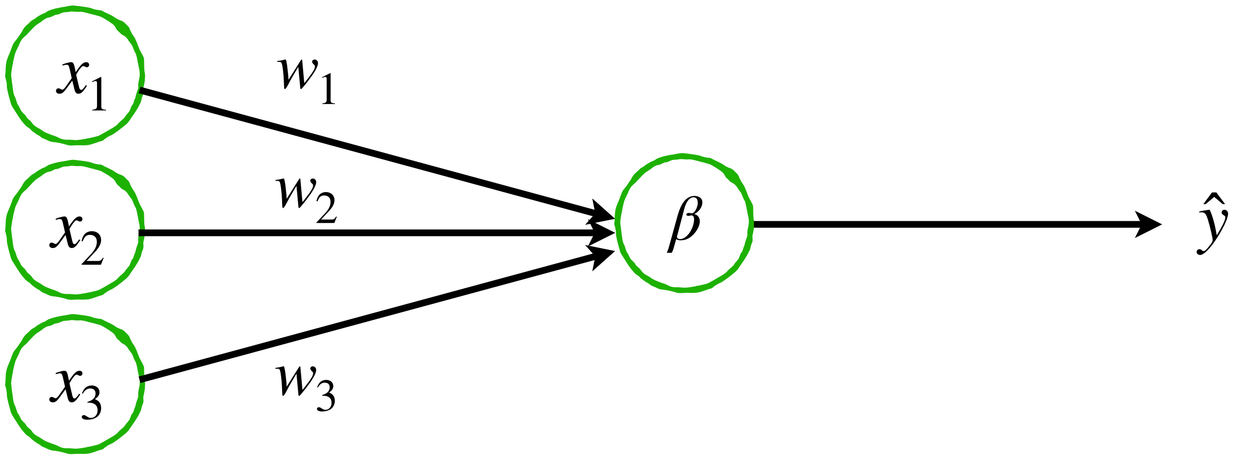


Multiply the inputs (x_1, x_2, x_3) with the Weights (w_1, w_2, w_3)
and add the Bias (β) to compute the output ($\hat{y}$)

Neural Networks

$$\hat{y} = \beta + w_1 x_1 + w_2 x_2 + w_3 x_3$$

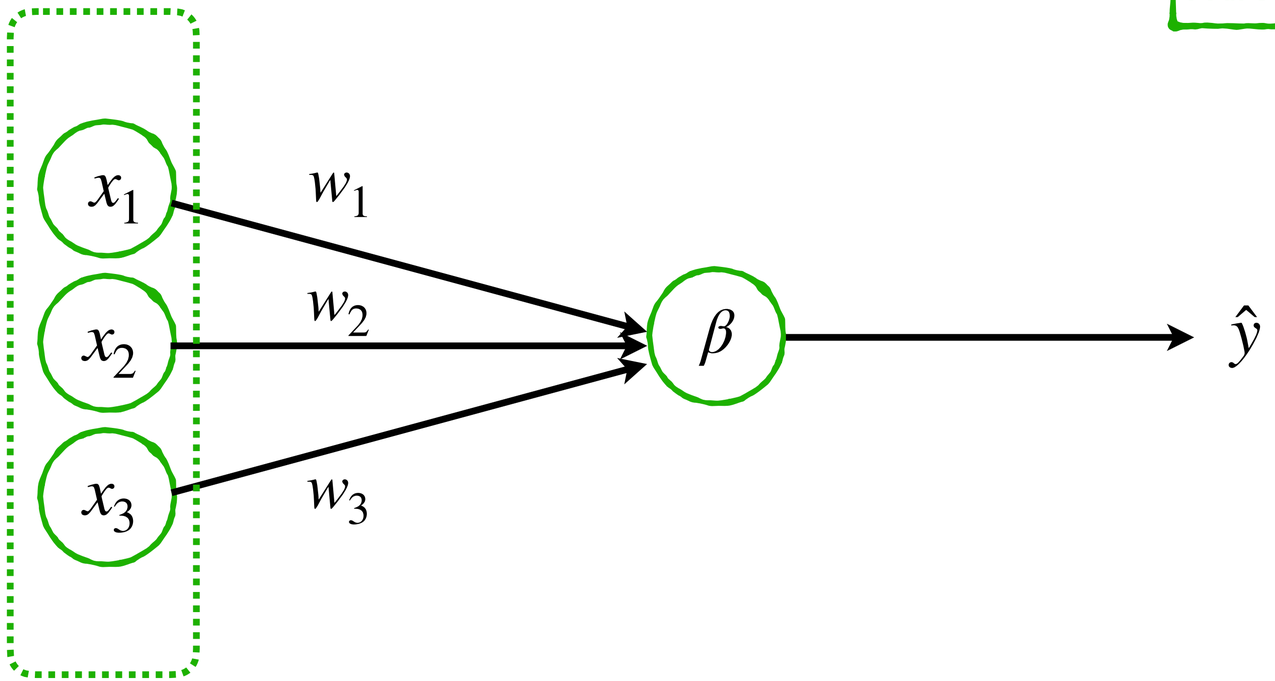
All these neural networks have two Layers…



Neural Networks

$$\hat{y} = \beta + w_1 x_1 + w_2 x_2 + w_3 x_3$$

All these neural networks have two Layers…

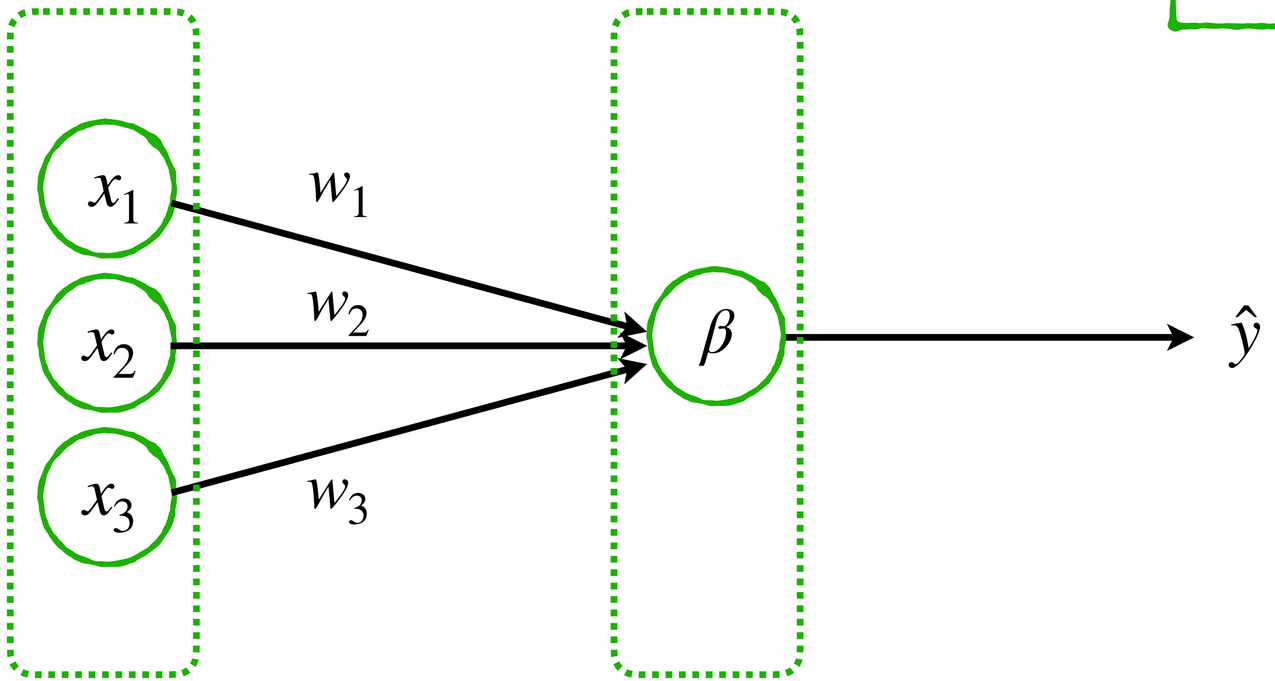


1 Input Layer

Neural Networks

$$\hat{y} = \beta + w_1 x_1 + w_2 x_2 + w_3 x_3$$

All these neural networks have two Layers…



1 Input Layer

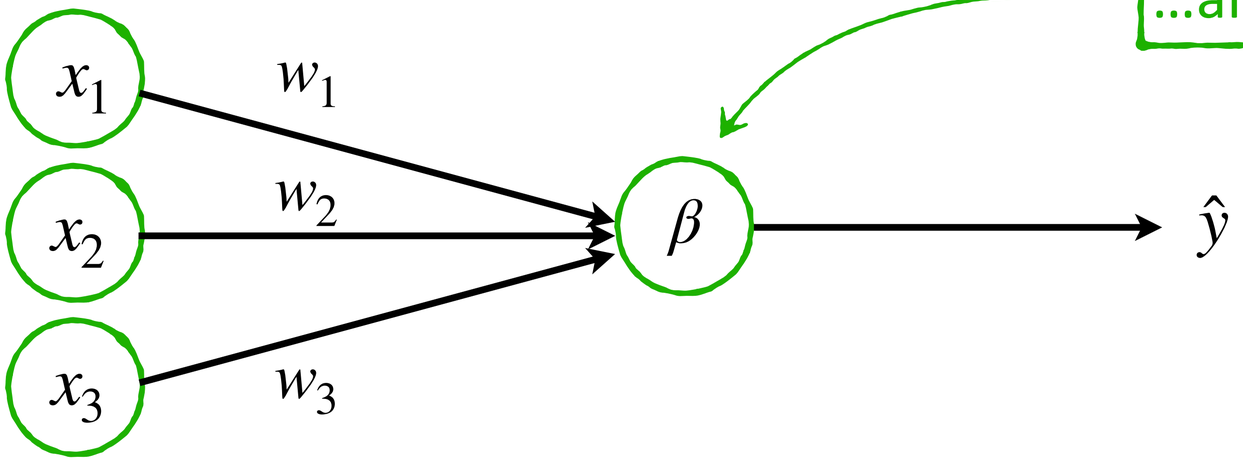
1 Output Layer

Neural Networks

$$\hat{y} = \beta + w_1 x_1 + w_2 x_2 + w_3 x_3$$

All these neural networks have two Layers…

…and one activation function

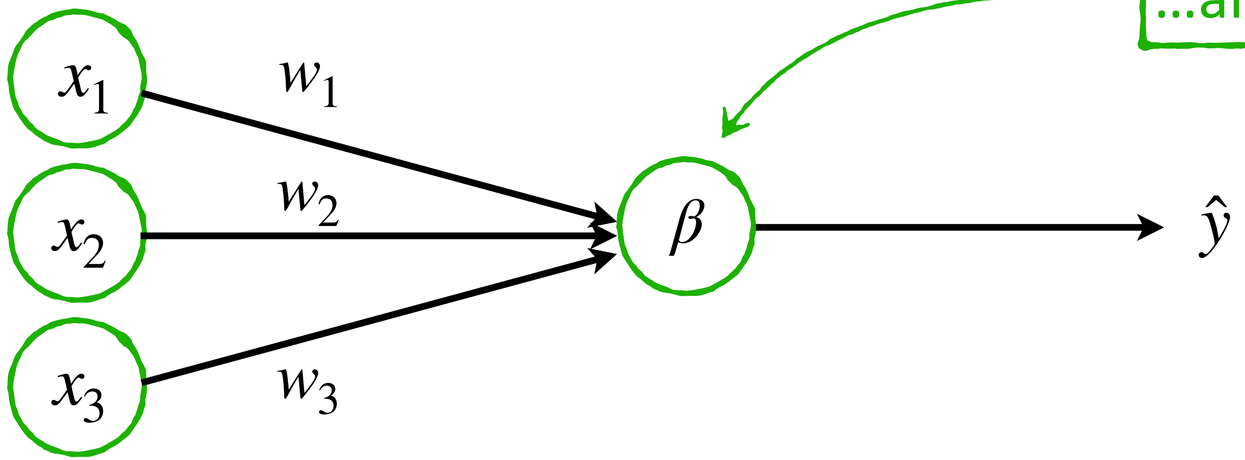


Neural Networks

$$\hat{y} = \beta + w_1 x_1 + w_2 x_2 + w_3 x_3$$

All these neural networks have two Layers…

…and one activation function



In this example the Activation Function is a linear combination of the inputs

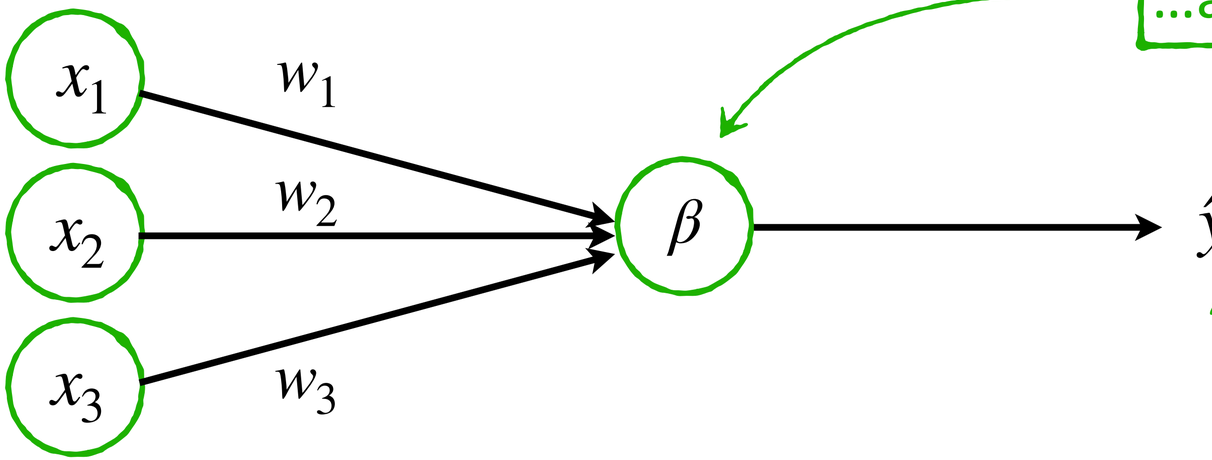
$$\hat{y} = \beta + \sum_{n=1}^{n} w_i x_i$$

Neural Networks

$$\hat{y} = \beta + w_1 x_1 + w_2 x_2 + w_3 x_3$$

All these neural networks have two Layers…

…and one activation function



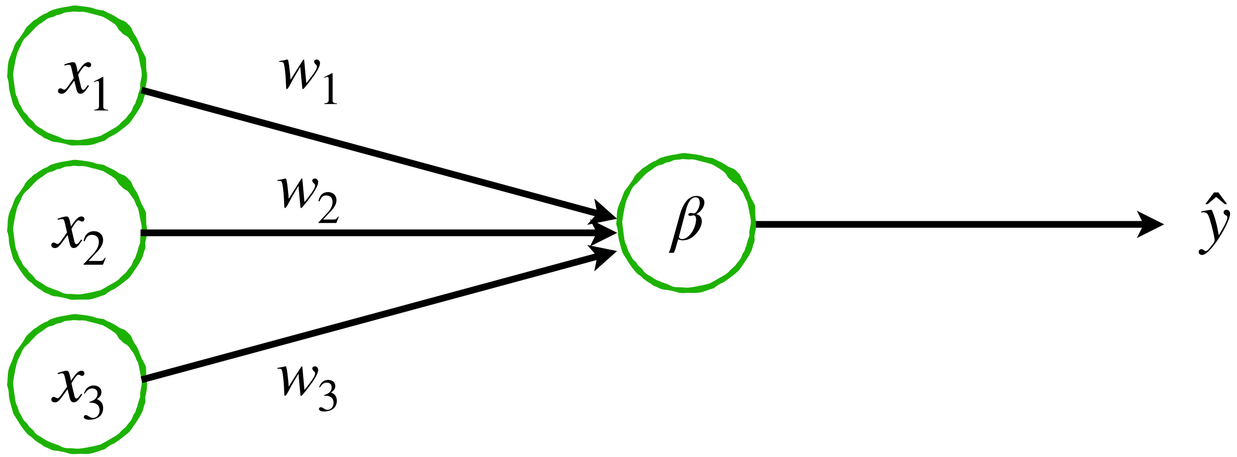
$\hat{y}$ is a continuous value

In this example the Activation Function is a linear combination of the inputs

$$\hat{y} = \beta + \sum_{n=1}^{n} w_i x_i$$

Neural Networks

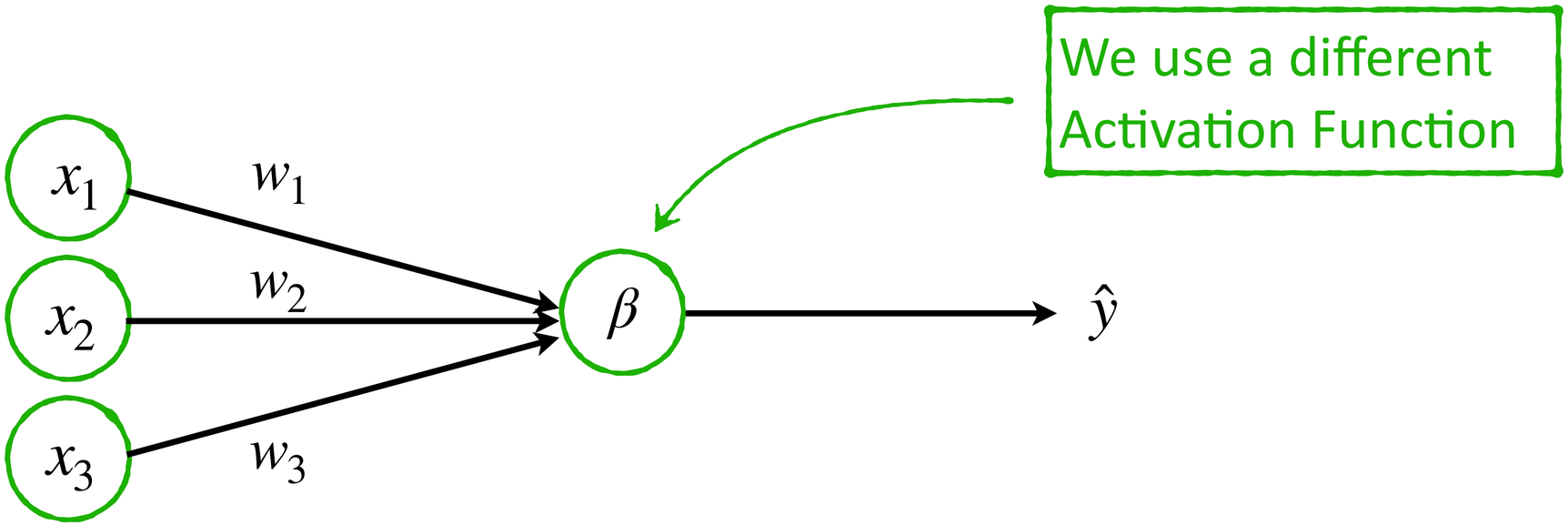
$$\hat{y} = \beta + w_1 x_1 + w_2 x_2 + w_3 x_3$$



What if we want to do Binary Classification instead of Linear Regression?

Neural Networks

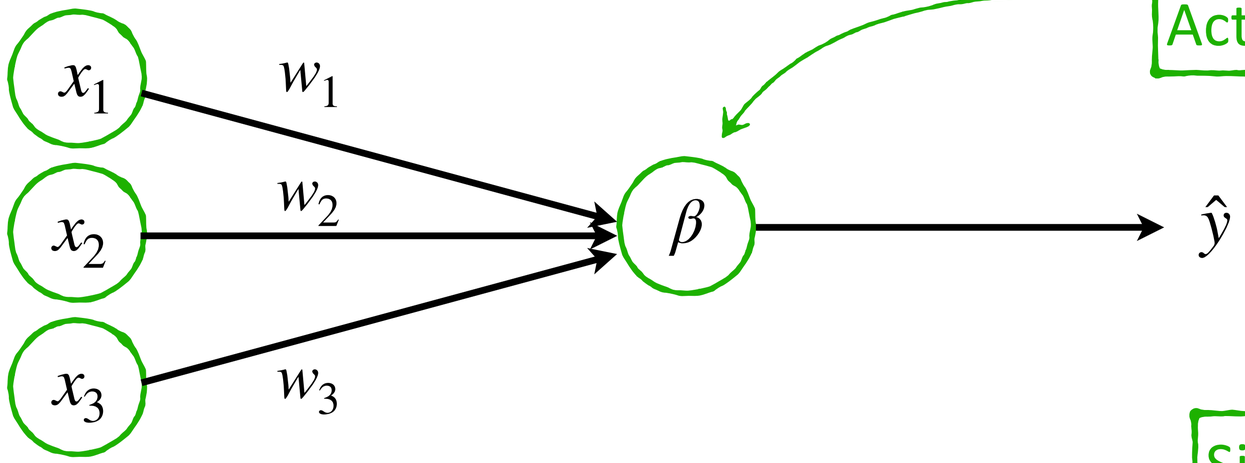
$$\hat{y} = \beta + w_1 x_1 + w_2 x_2 + w_3 x_3$$



What if we want to do Binary Classification instead of Linear Regression?

Neural Networks

$$\hat{y} = \beta + w_1 x_1 + w_2 x_2 + w_3 x_3$$



We use a different Activation Function

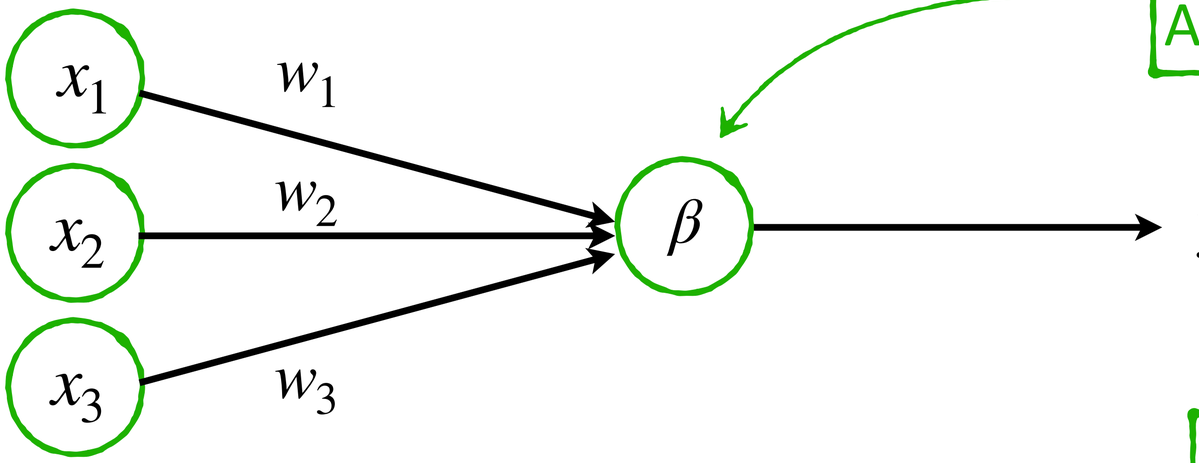
Sigmoid Activation Function

$$\frac{1}{1 + e^{-x}}$$

What if we want to do Binary Classification instead of Linear Regression?

Neural Networks

$$\hat{y} = \beta + w_1 x_1 + w_2 x_2 + w_3 x_3$$



We use a different Activation Function

$\hat{y}$ is a probability between 0 and 1

Sigmoid Activation Function

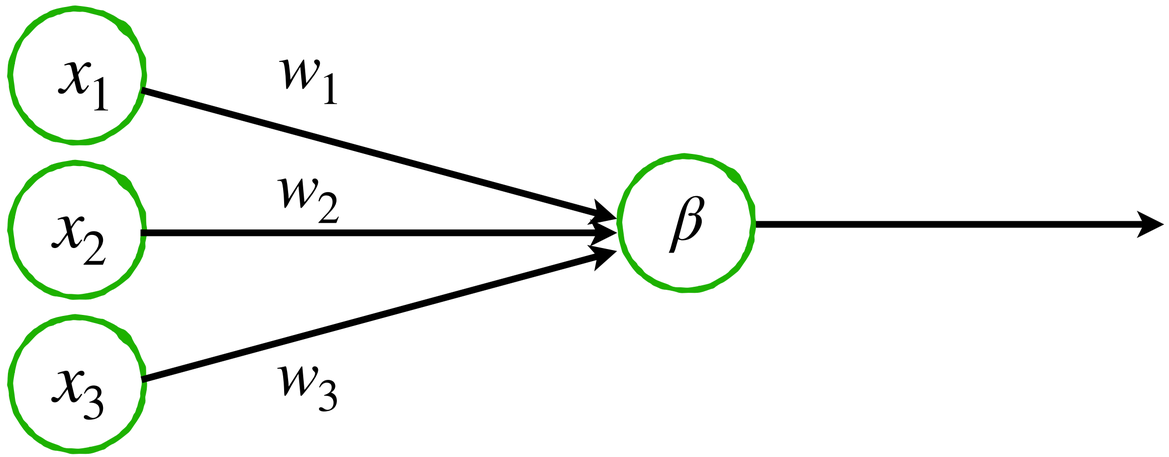
$$\frac{1}{1 + e^{-x}}$$

What if we want to do Binary Classification instead of Linear Regression?

Neural Networks

$$\hat{y} = \beta + w_1 x_1 + w_2 x_2 + w_3 x_3$$

Add a threshold for binary classification



$$\hat{y} \quad \begin{cases} true, & if\ \hat{y} \geq 0.5 \\ false, & if\ \hat{y} < 0.5 \end{cases}$$

Sigmoid Activation Function

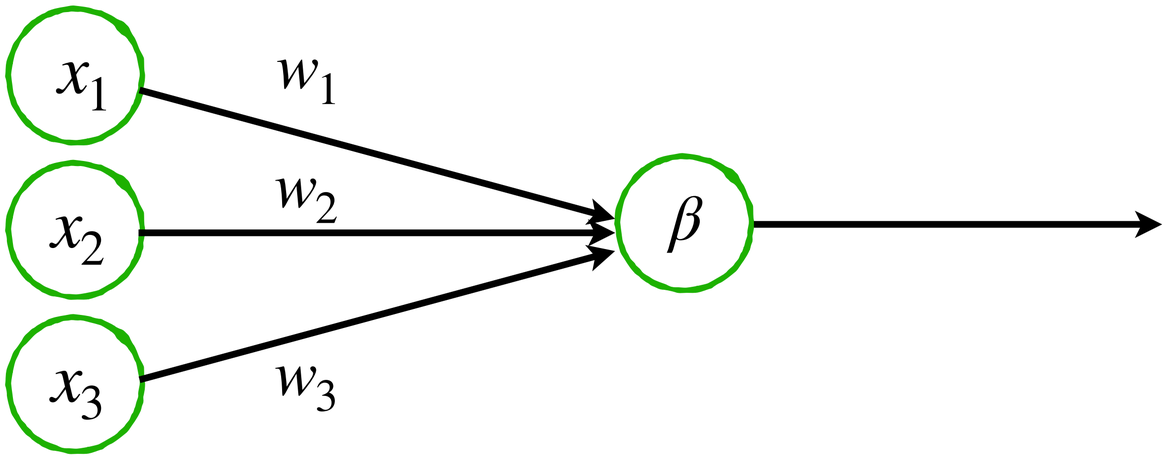
$$\frac{1}{1 + e^{-x}}$$

What if we want to do Binary Classification instead of Linear Regression?

Neural Networks

$$\hat{y} = \beta + w_1 x_1 + w_2 x_2 + w_3 x_3$$

Add a threshold for binary classification



$$\hat{y} \quad \begin{cases} true, & if\ \hat{y} \geq 0.5 \\ false, & if\ \hat{y} < 0.5 \end{cases}$$

Sigmoid Activation Function

$$\frac{1}{1 + e^{-x}}$$

Logistic Regression

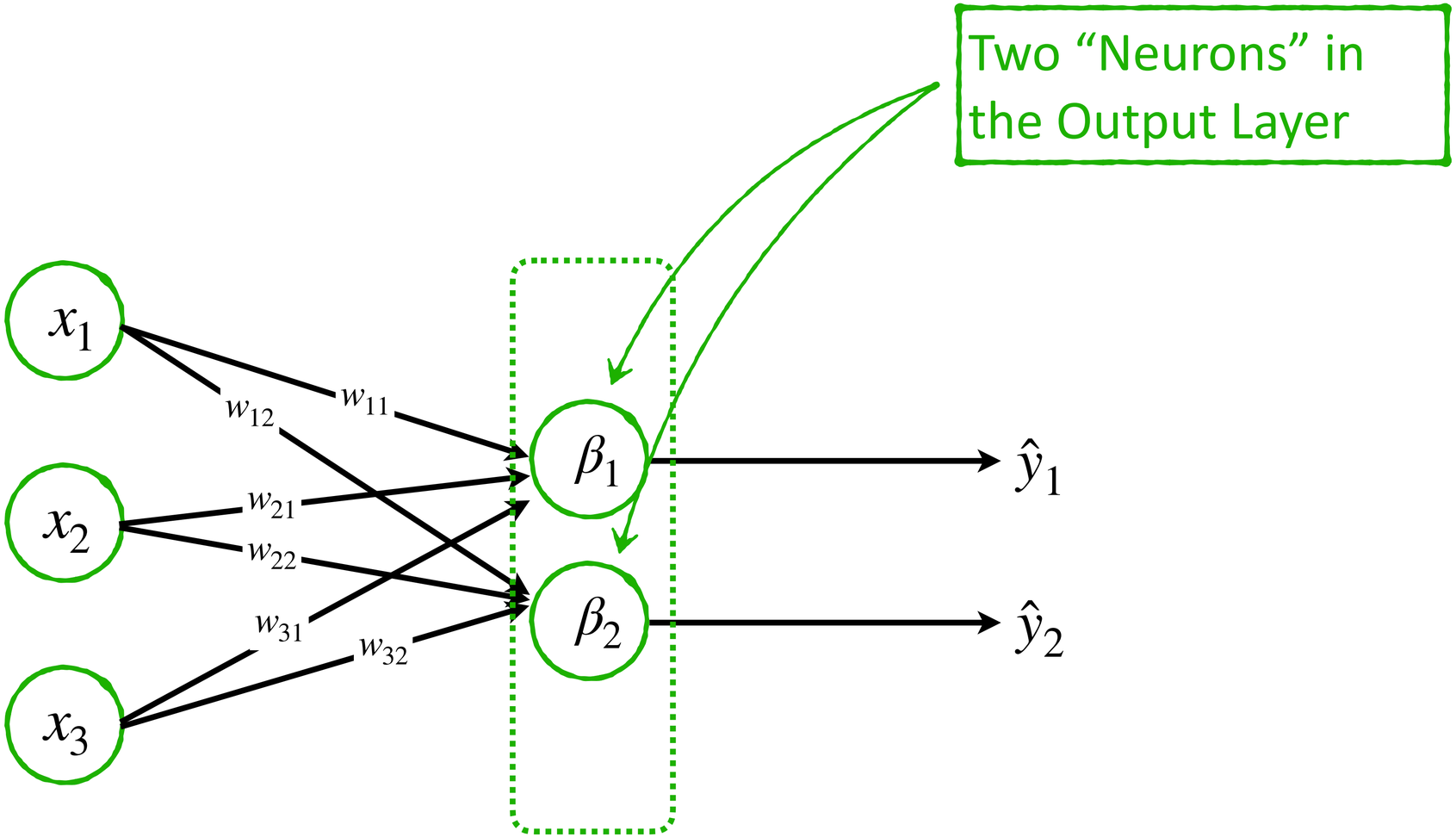
Neural Networks

$$\hat{y} = \beta + w_1 x_1 + w_2 x_2 + w_3 x_3$$

How do we do Multi-Class Classification?

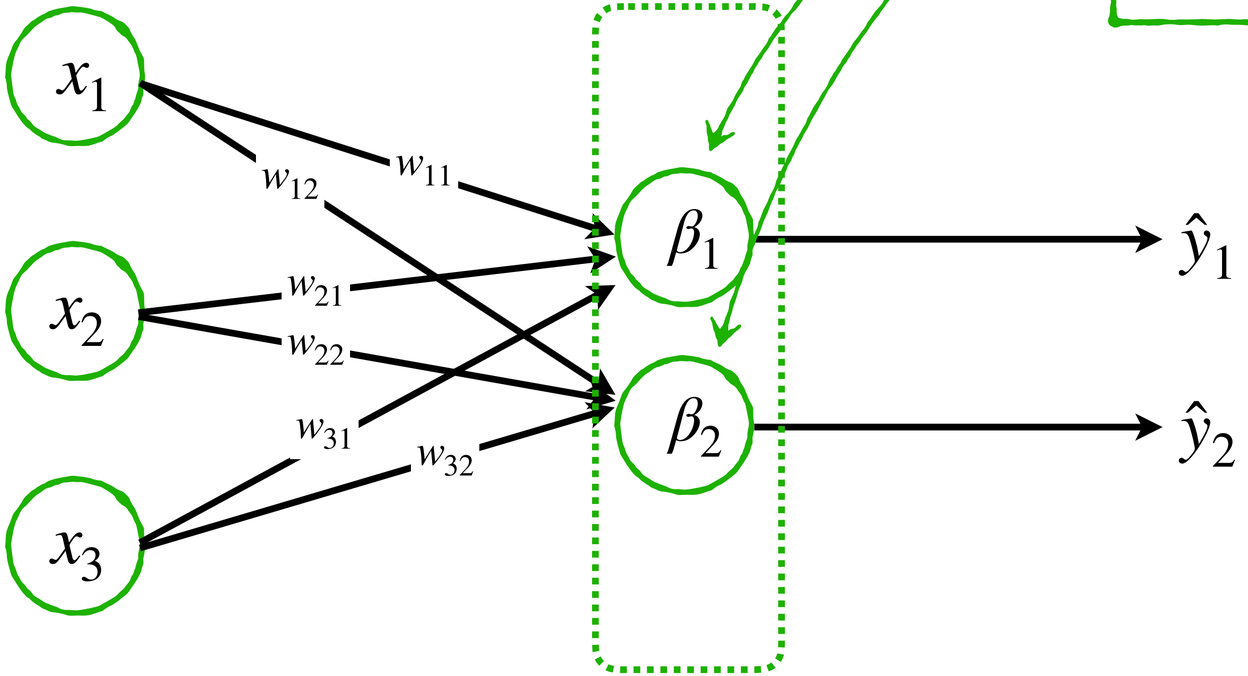
We simply add more outputs to the output layer

Neural Networks



Multi Class Classification with two outputs

Neural Networks



Two “Neurons” in the Output Layer

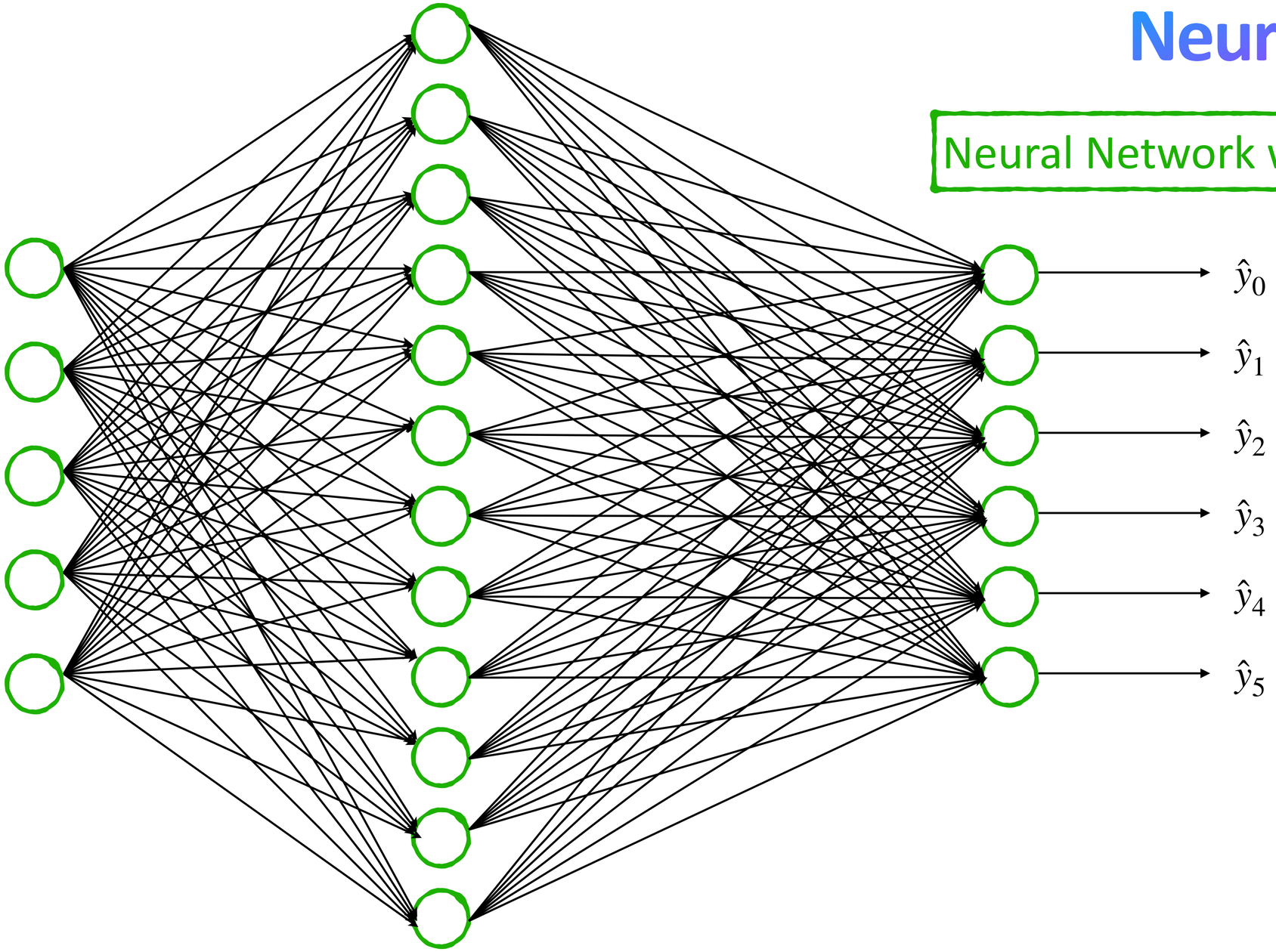
Each with its own Bias β_1 and β_2

Multi Class Classification with two outputs

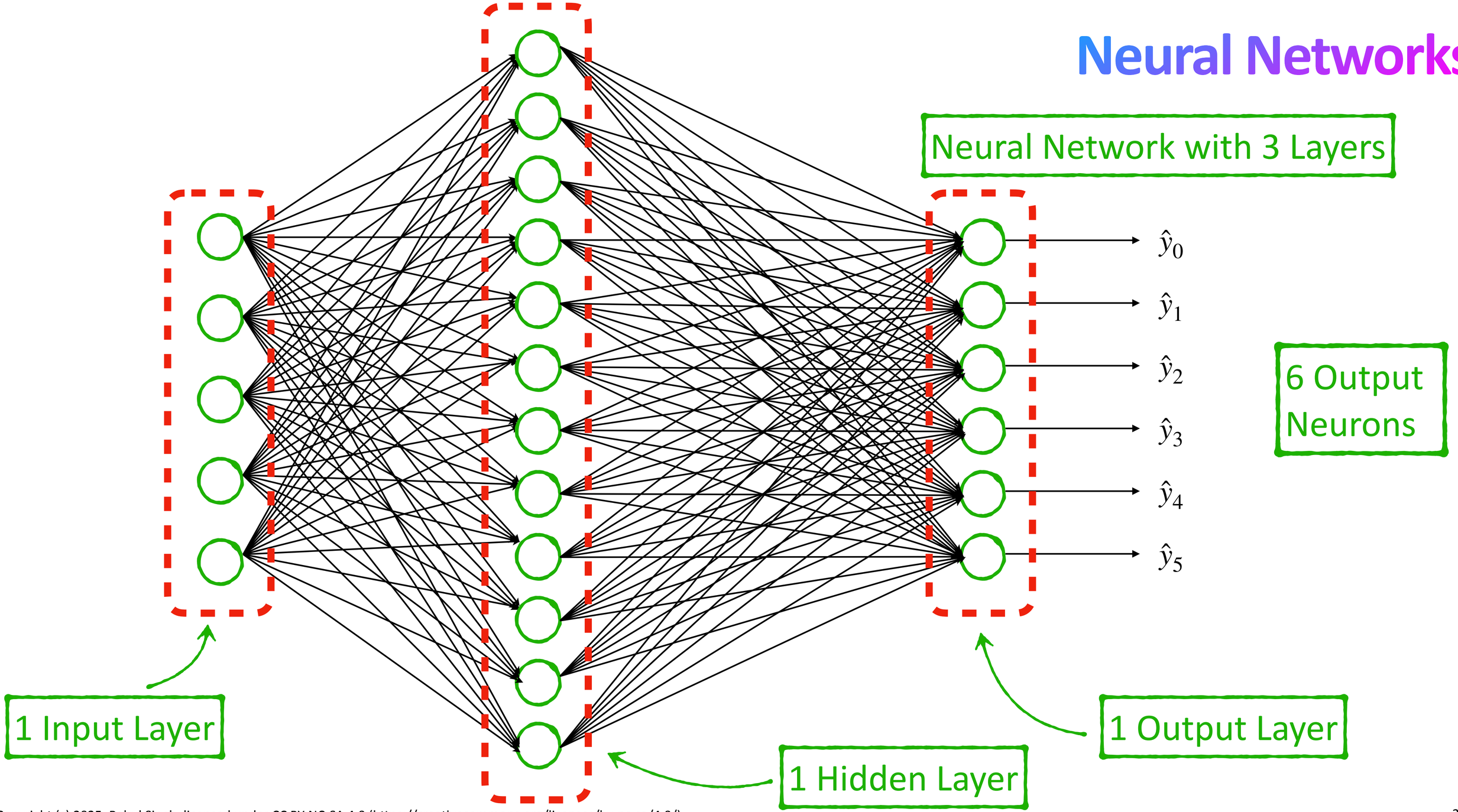
Neural Networks

We can add as many outputs as we need…
We can add as many layers as well need…

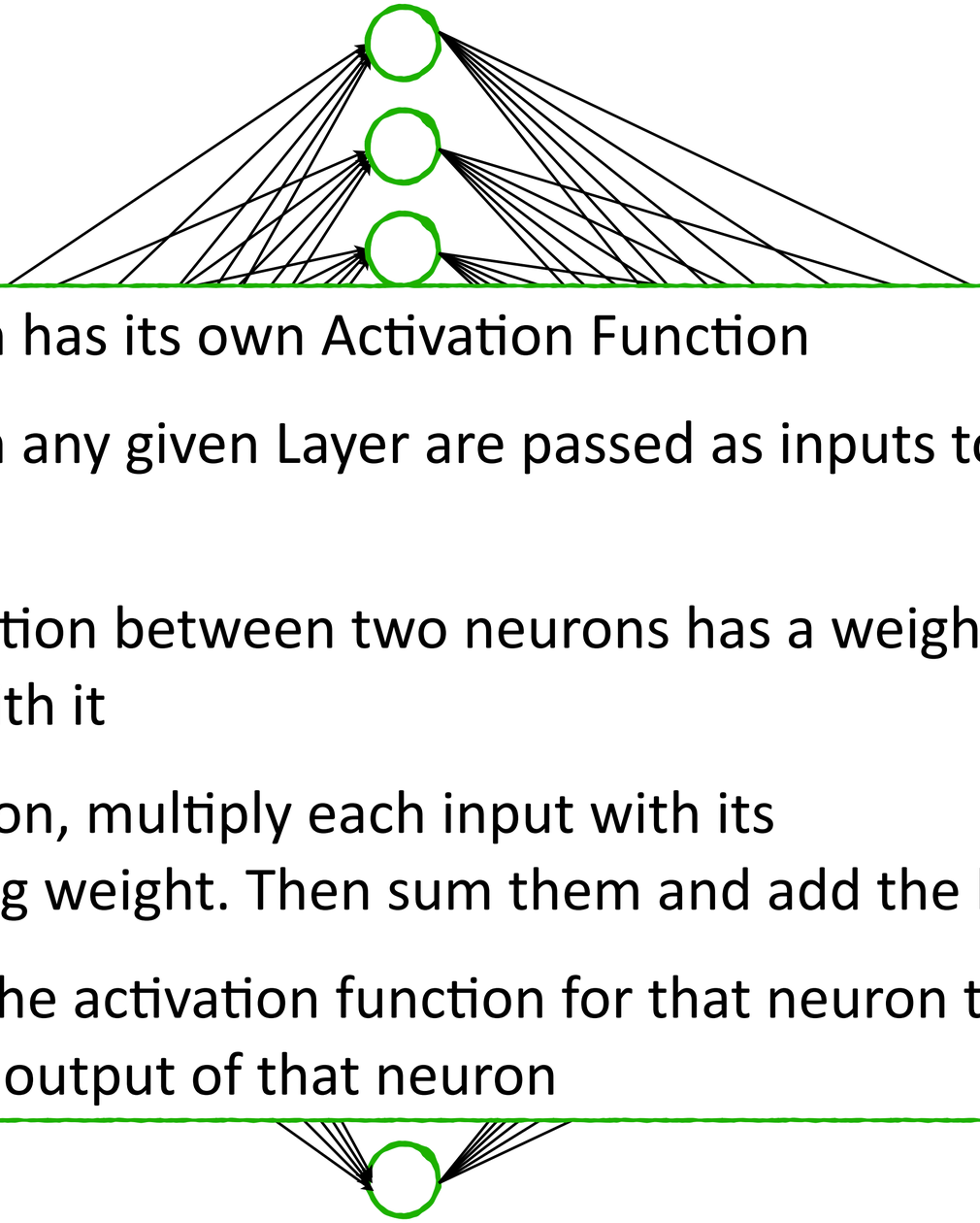
Neural Networks



Neural Networks



Neural Networks



Neural Network with 3 Layers

- Every Neuron has its own Activation Function
- Outputs from any given Layer are passed as inputs to the next layer
- Every connection between two neurons has a weight associated with it
- For any Neuron, multiply each input with its corresponding weight. Then sum them and add the bias.
- Pass that to the activation function for that neuron to compute the output of that neuron

$\hat{y}_0$

$\hat{y}_1$

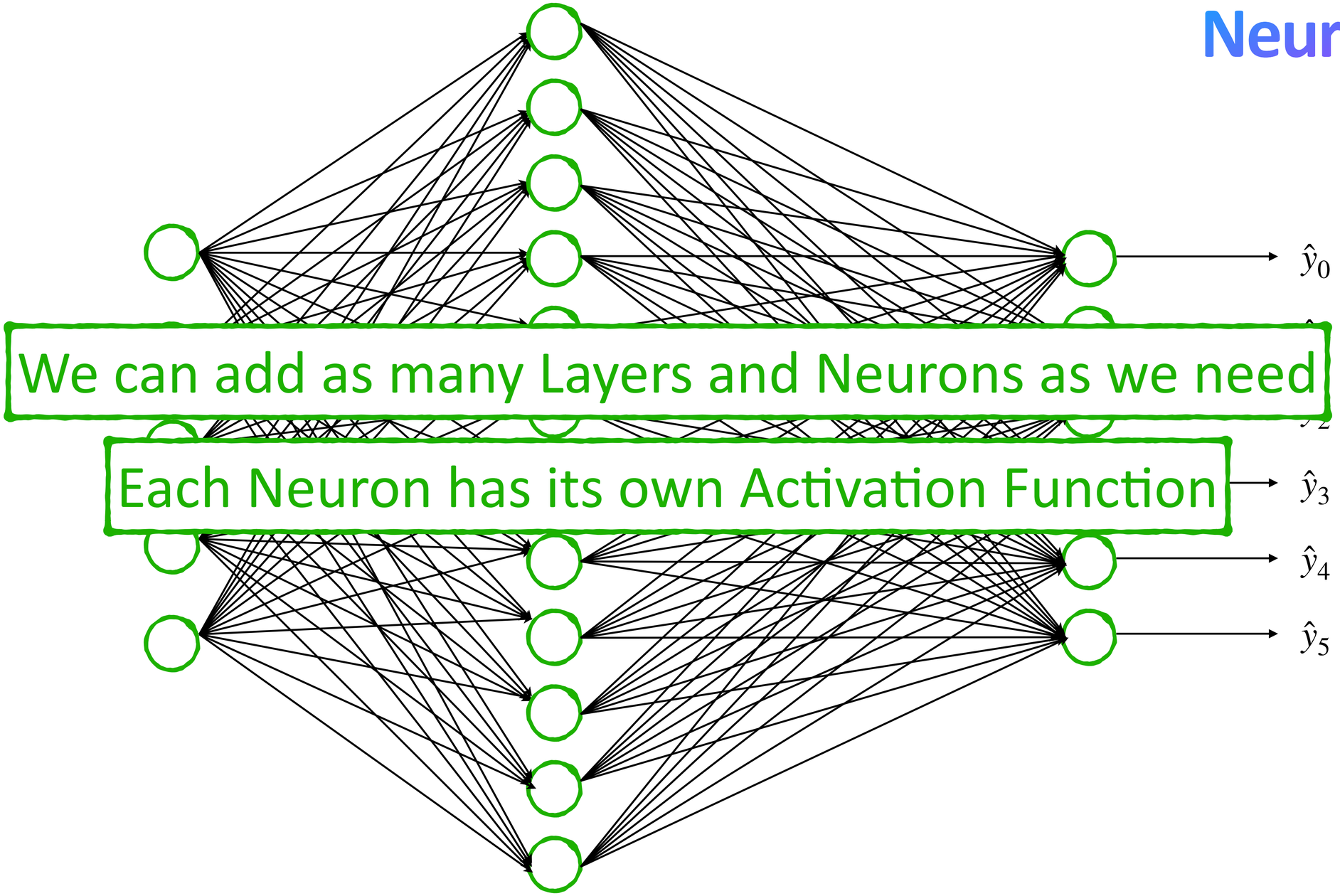
$\hat{y}_2$

$\hat{y}_3$

$\hat{y}_4$

$\hat{y}_5$

Neural Networks



We can add as many Layers and Neurons as we need

Each Neuron has its own Activation Function

Neural Networks

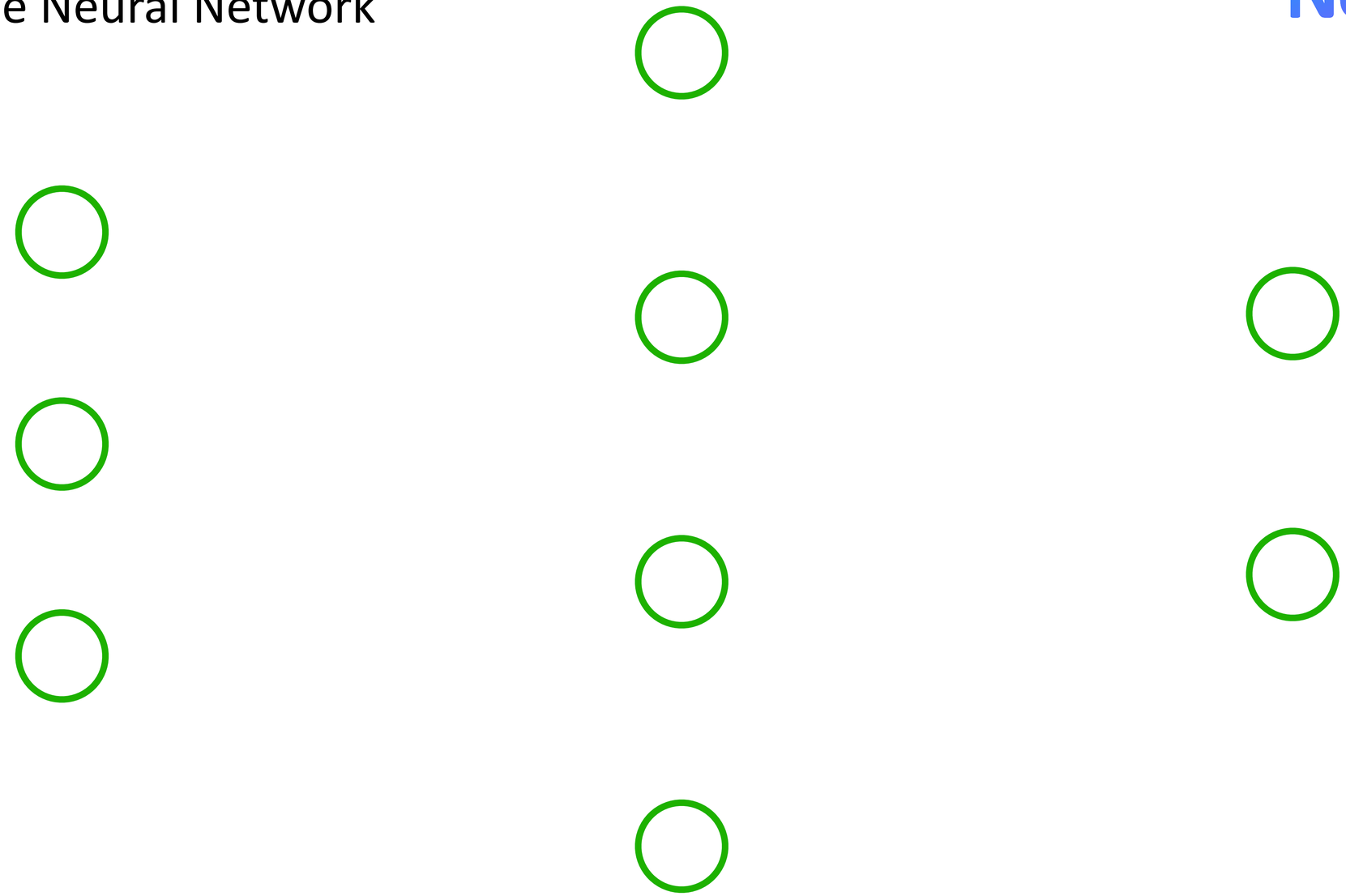
Question: How do we compute the inputs and outputs of each neuron?

- For any Neuron, multiply each input with its corresponding weight. Then sum them and add the bias.
- Pass that to the activation function for that neuron to compute the output of that neuron

Lets walk through how this works in practice…

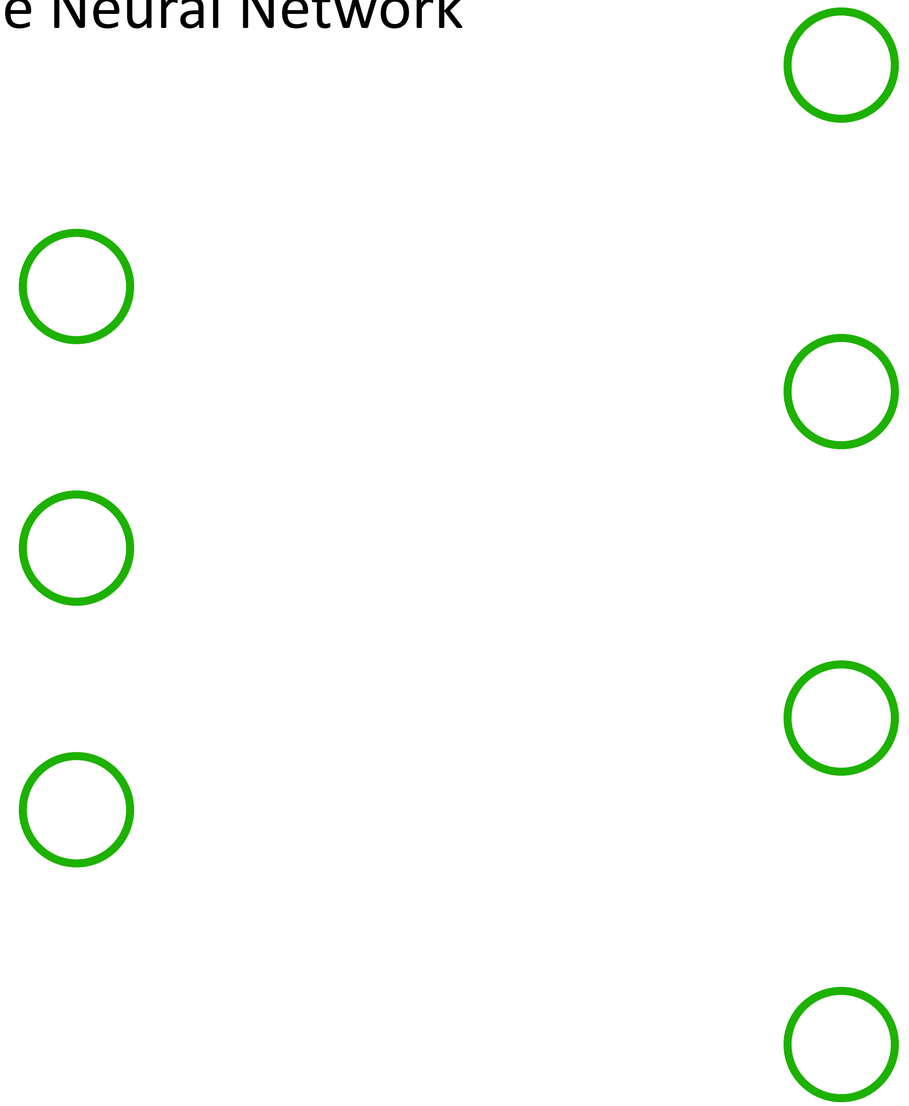
Neural Networks

A Simple Neural Network



Neural Networks

A Simple Neural Network

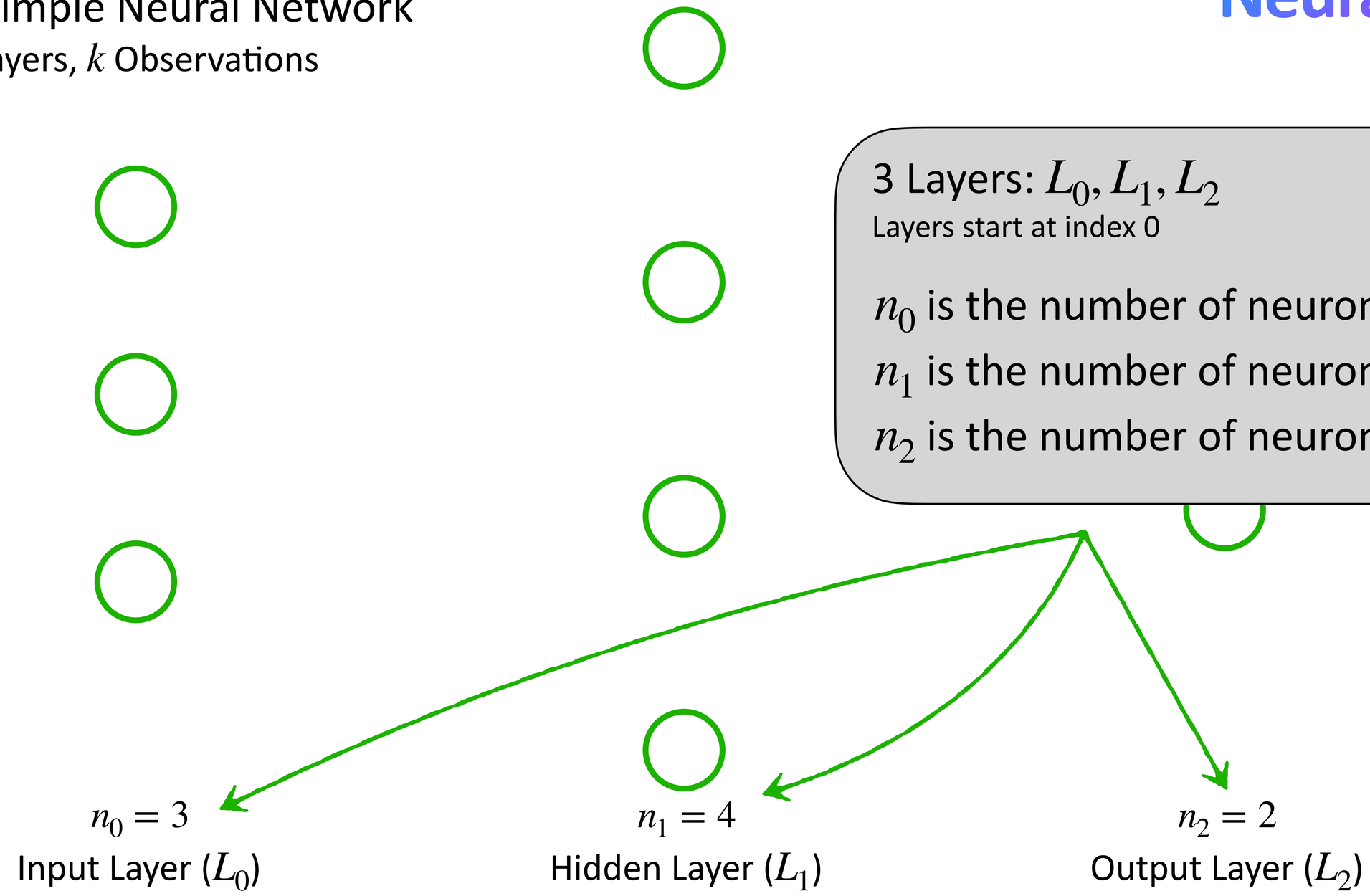


Lets develop a standard notation

Neural Networks

A Simple Neural Network

3 Layers, k Observations



Neural Networks

A Simple Neural Network

3 Layers, k Observations

The training data has k observations

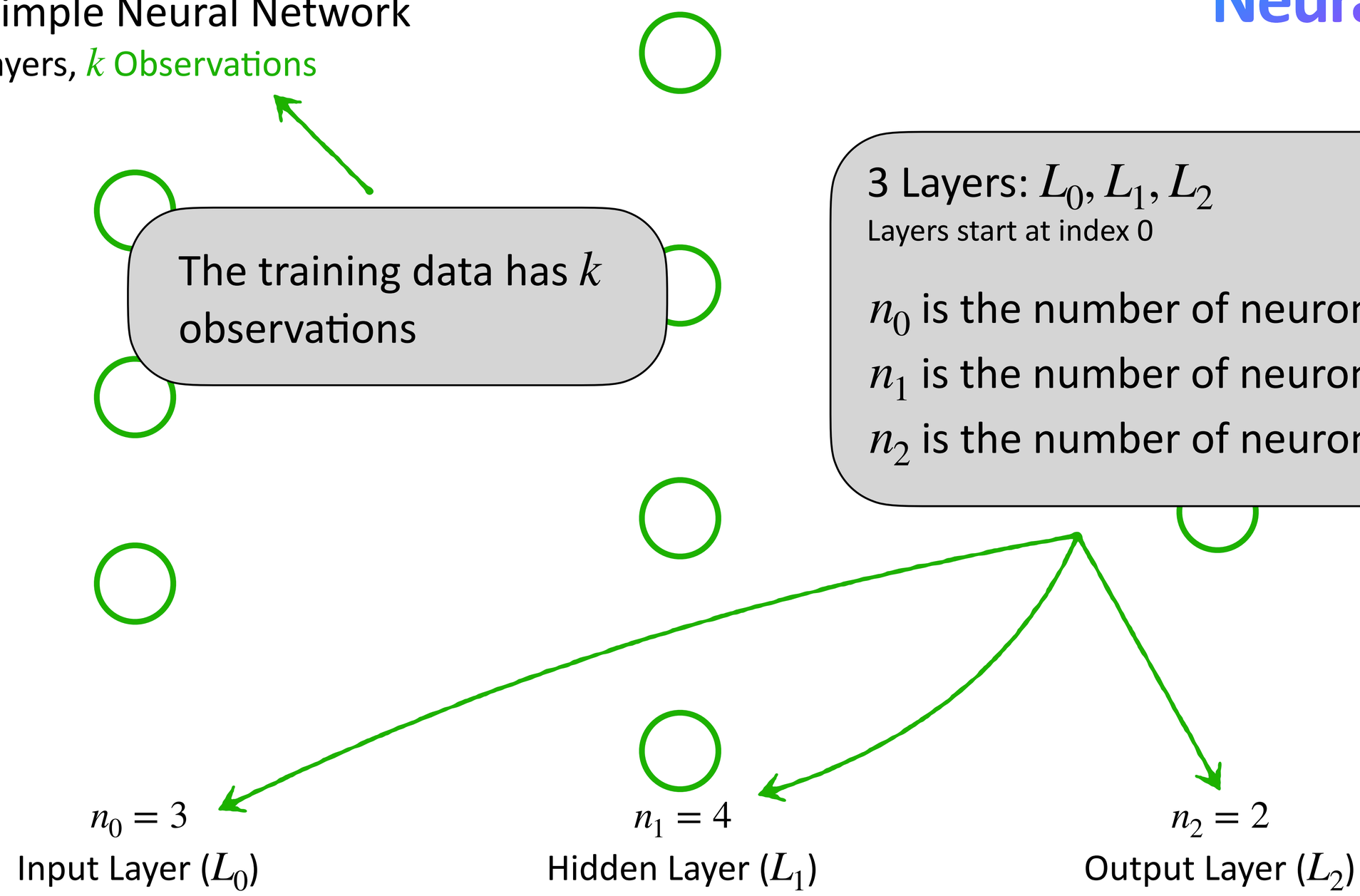
3 Layers: L_0, L_1, L_2

Layers start at index 0

n_0 is the number of neurons in L_0

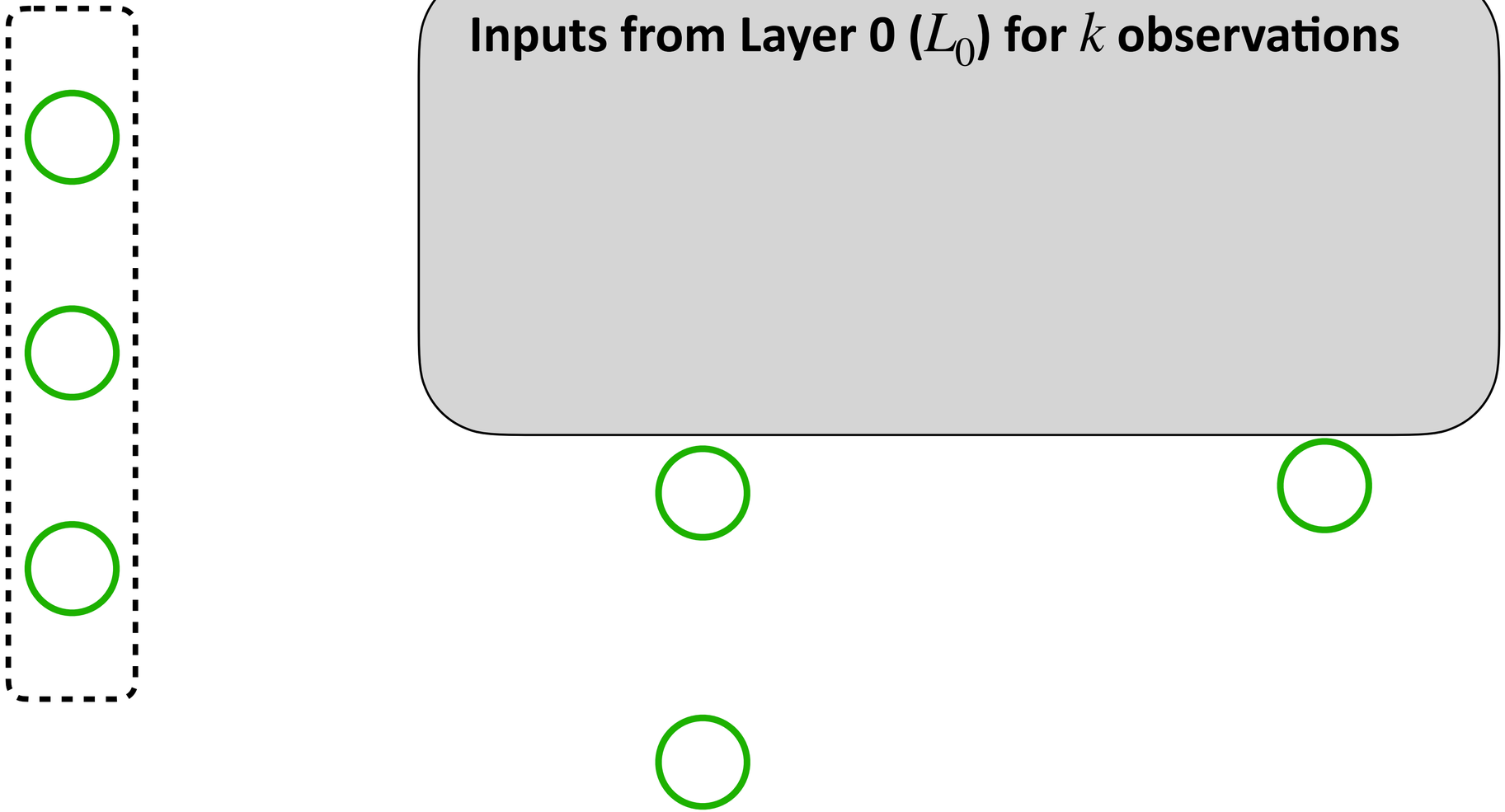
n_1 is the number of neurons in L_1

n_2 is the number of neurons in L_2



A Simple Neural Network

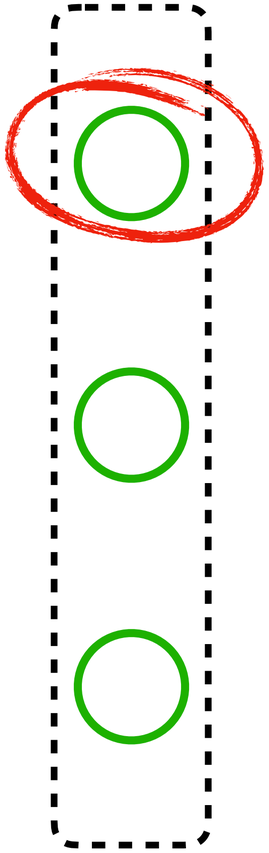
3 Layers, k Observations



Neural Networks

A Simple Neural Network

3 Layers, k Observations



Inputs from Layer 0 (L_0) for k observations

Neuron 1, in Layer 0 (L_0) with k observations

$$x_{11}, x_{12}, x_{13} \ldots x_{1k}$$

$n_0 = 3$

Input Layer (L_0)

$n_1 = 4$

Hidden Layer (L_1)

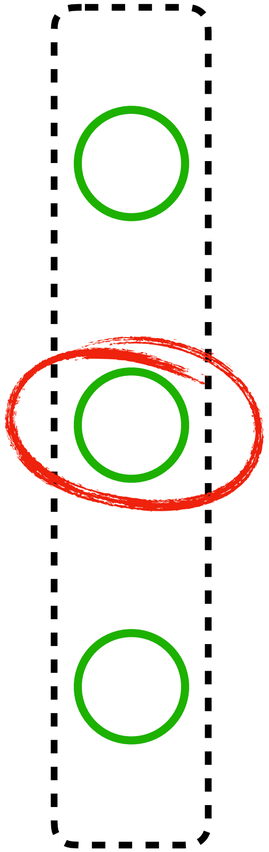
$n_2 = 2$

Output Layer (L_2)

Neural Networks

A Simple Neural Network

3 Layers, k Observations



Inputs from Layer 0 (L_0) for k observations

Neuron 2, in Layer 0 (L_0) with k observations

$$x_{21}, x_{22}, x_{23} \ldots x_{2k}$$

$n_0 = 3$

Input Layer (L_0)

$n_1 = 4$

Hidden Layer (L_1)

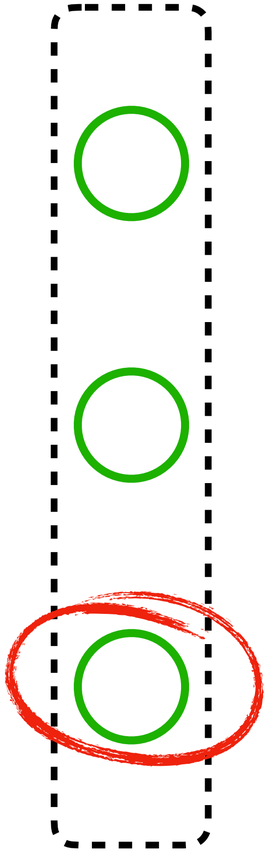
$n_2 = 2$

Output Layer (L_2)

Neural Networks

A Simple Neural Network

3 Layers, k Observations



Inputs from Layer 0 (L_0) for k observations

Neuron 3, in Layer 0 (L_0) with k observations

$$x_{31}, x_{32}, x_{33} \ldots x_{3k}$$

$n_0 = 3$

Input Layer (L_0)

$n_1 = 4$

Hidden Layer (L_1)

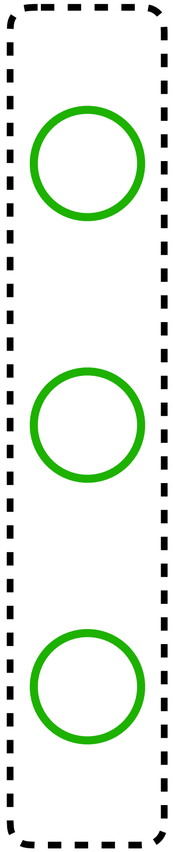
$n_2 = 2$

Output Layer (L_2)

Neural Networks

A Simple Neural Network

3 Layers, k Observations



Inputs from Layer 0 (L_0) for k observations

Layer 0 can be represented by a $n_0 \times k$ matrix

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} \ldots x_{1k} \\ x_{21} & x_{22} & x_{23} \ldots x_{2k} \\ x_{31} & x_{32} & x_{33} \ldots x_{3k} \end{bmatrix}$$

$$3 \times k$$

This is the matrix of inputs (L_0)

$n_0 = 3$

Input Layer (L_0)

$n_1 = 4$

Hidden Layer (L_1)

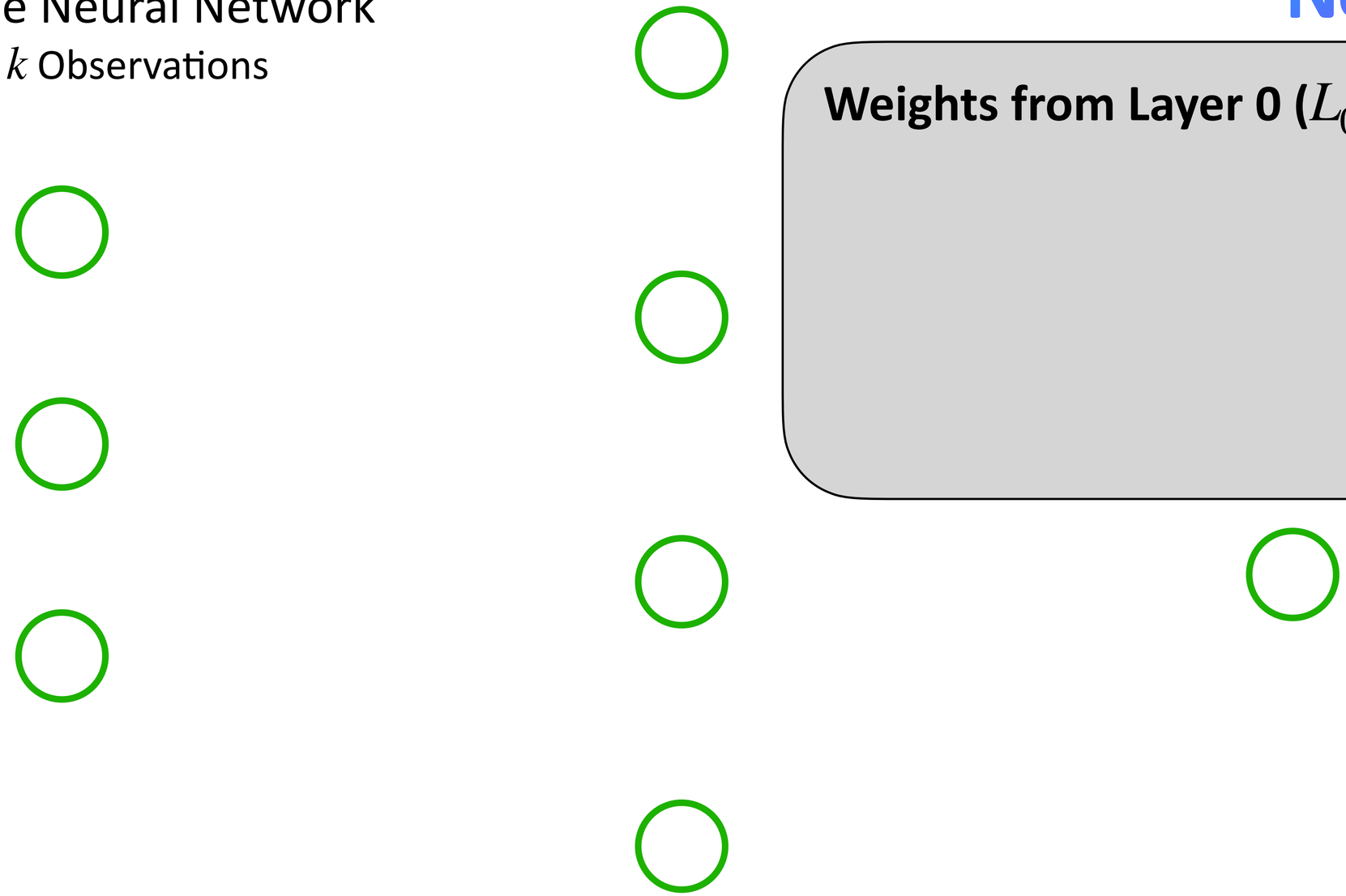
$n_2 = 2$

Output Layer (L_2)

Neural Networks

A Simple Neural Network

3 Layers, k Observations



$n_0 = 3$

Input Layer (L_0)

$n_1 = 4$

Hidden Layer (L_1)

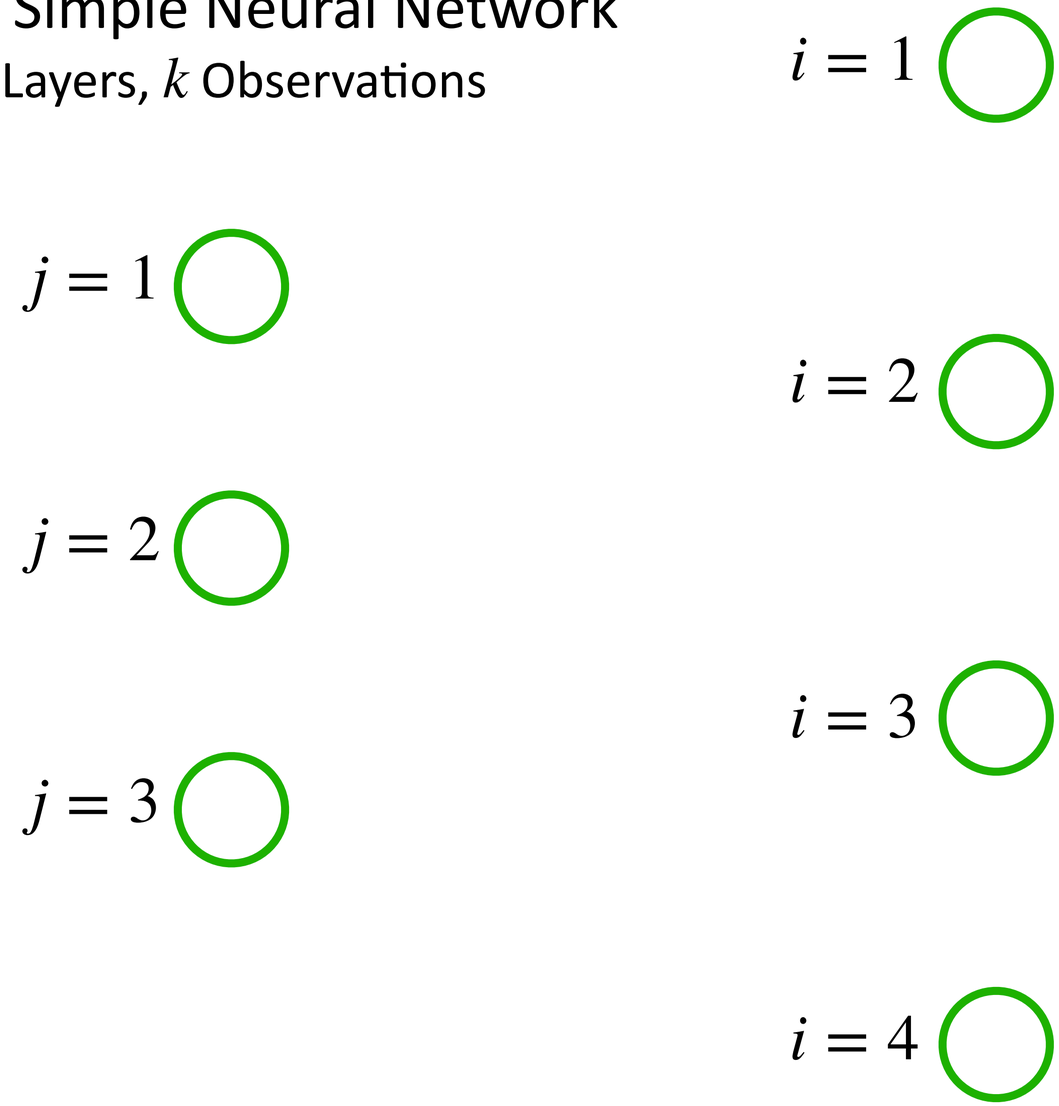
$n_2 = 2$

Output Layer (L_2)

Neural Networks

A Simple Neural Network

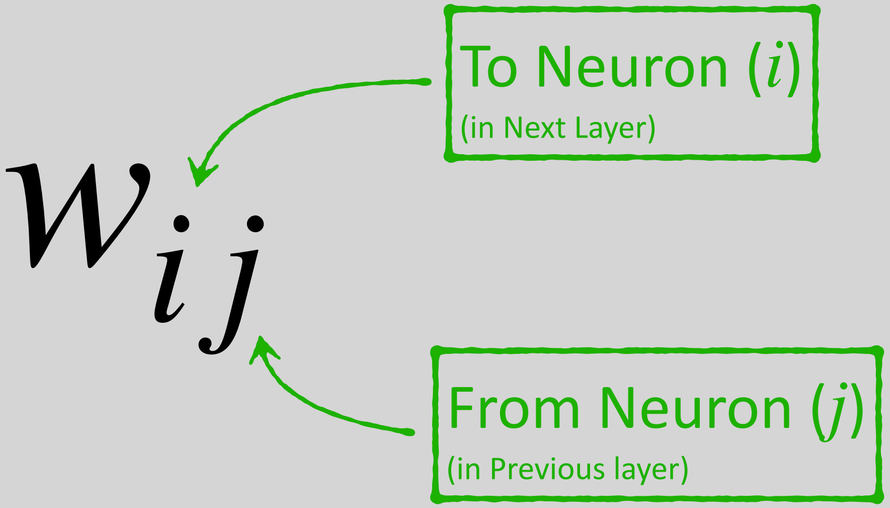
3 Layers, k Observations



Weights from Layer 0 (L_0) to Layer 1 (L_1)

The first subscript (i) represents the neuron in the destination layer

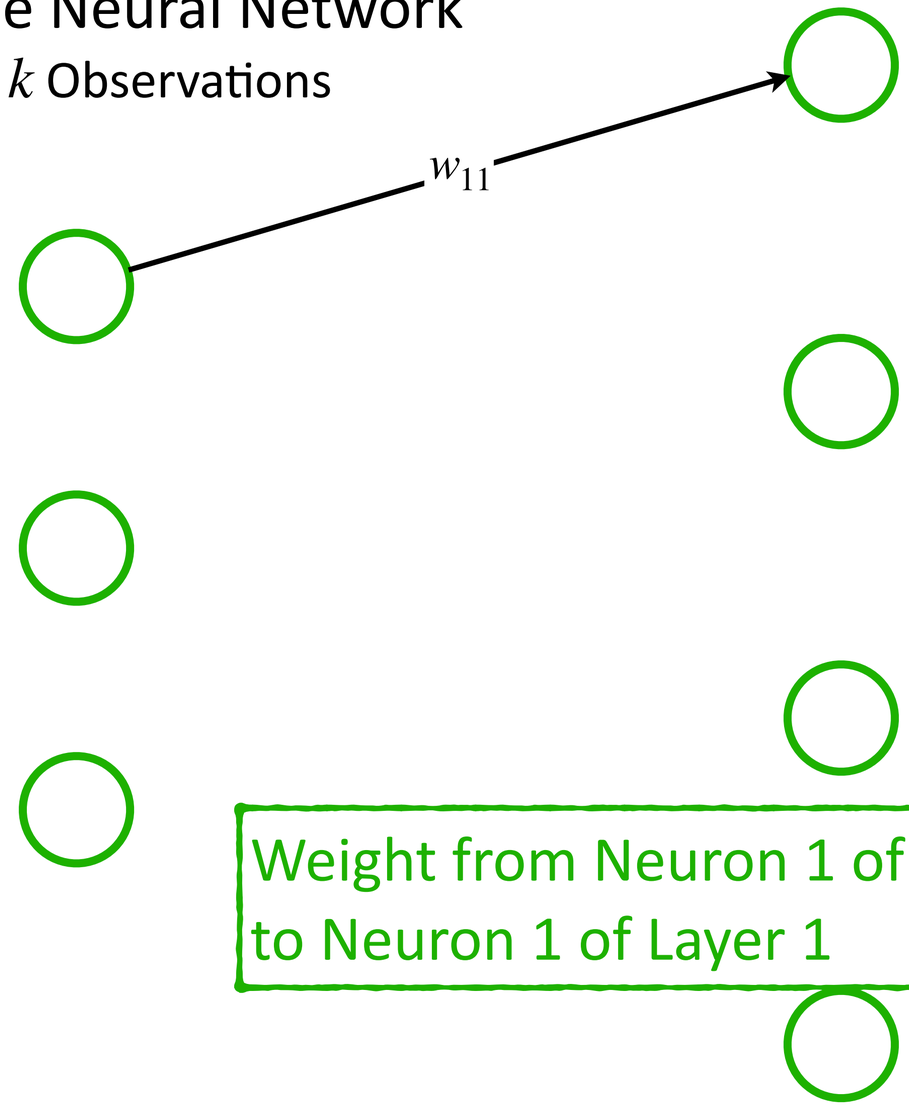
The second subscript (j) represents the neuron in the source layer



Neural Networks

A Simple Neural Network

3 Layers, k Observations



Weights from Layer 0 (L_0) to Layer 1 (L_1)

$$w_{11}, w_{12}, w_{13}$$

w_{11}

To Neuron (i)
(in Next Layer)

From Neuron (j)
(in Previous layer)

Weight from Neuron 1 of Layer 0 to Neuron 1 of Layer 1

$n_0 = 3$
Input Layer (L_0)

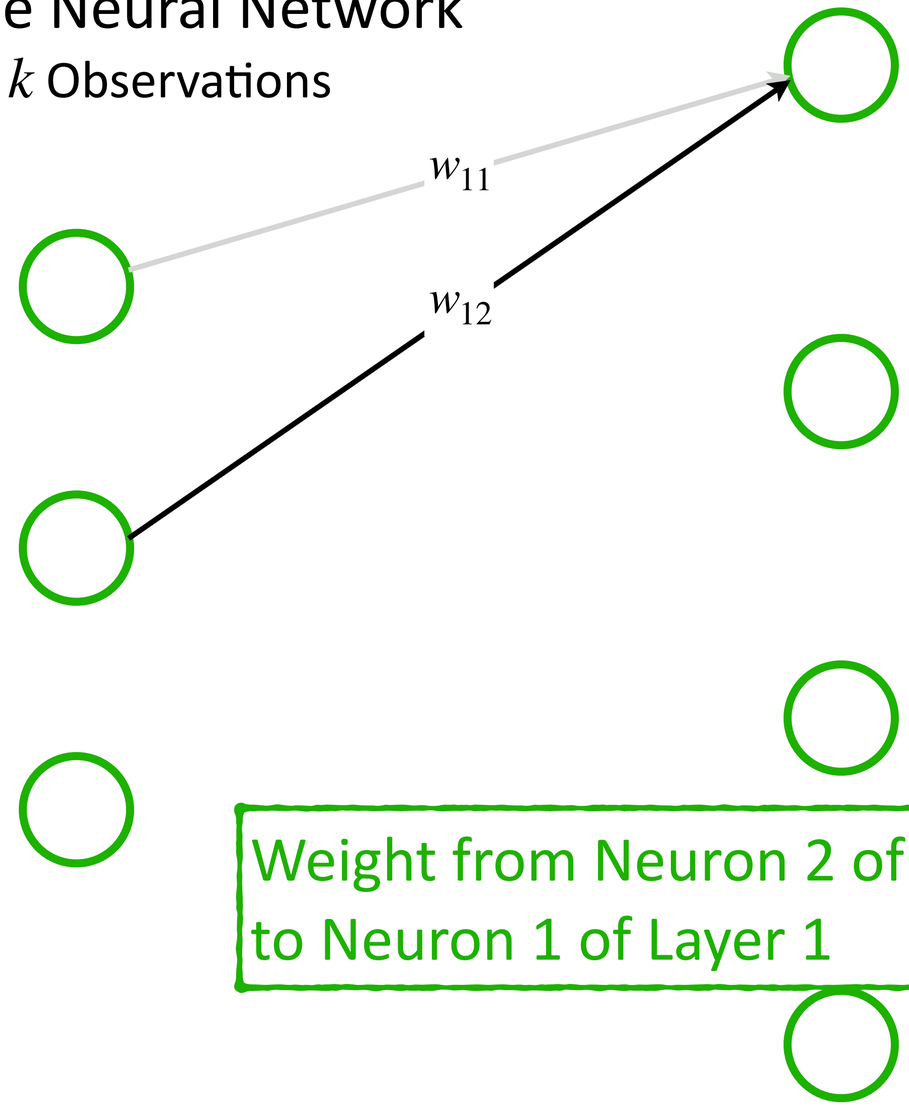
$n_1 = 4$
Hidden Layer (L_1)

$n_2 = 2$
Output Layer (L_2)

Neural Networks

A Simple Neural Network

3 Layers, k Observations



Weights from Layer 0 (L_0) to Layer 1 (L_1)

$$w_{11}, w_{12}, w_{13}$$

w_{12}

To Neuron (i)
(in Next Layer)

From Neuron (j)
(in Previous layer)

Weight from Neuron 2 of Layer 0
to Neuron 1 of Layer 1

$n_0 = 3$
Input Layer (L_0)

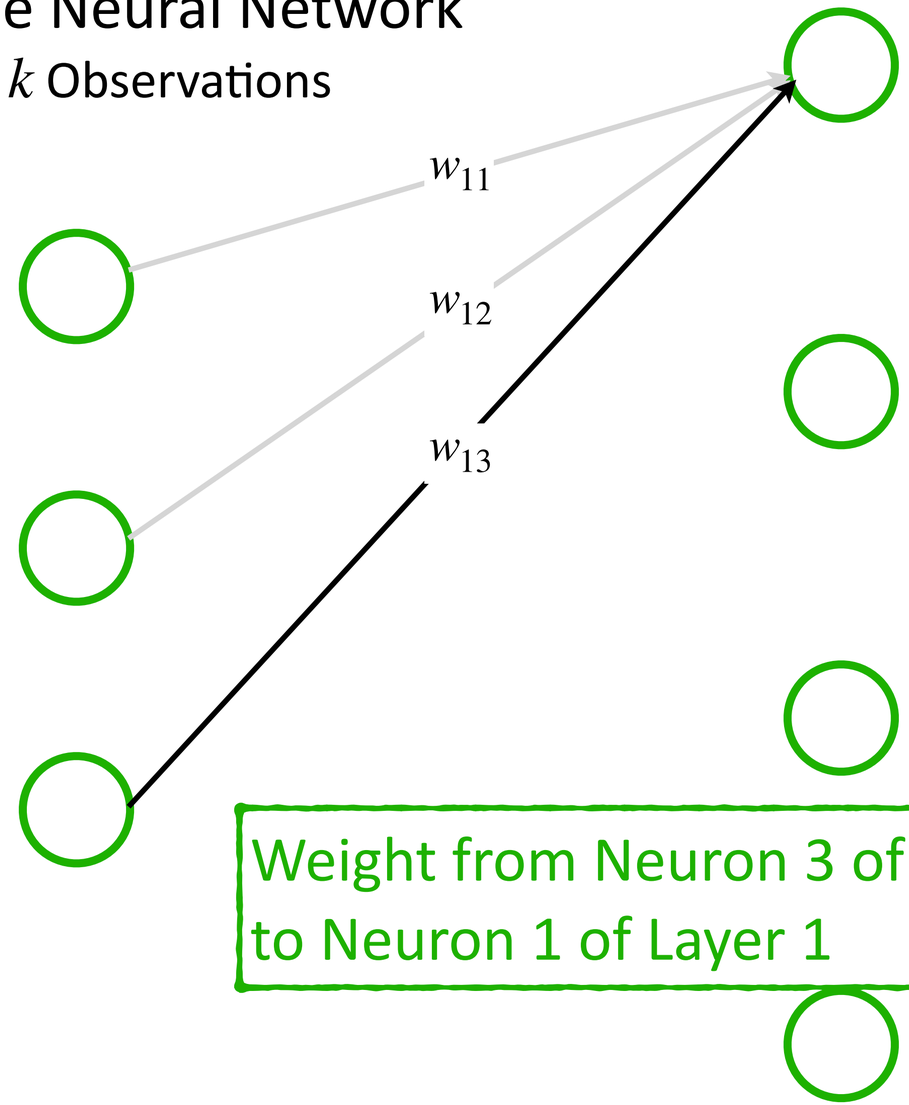
$n_1 = 4$
Hidden Layer (L_1)

$n_2 = 2$
Output Layer (L_2)

Neural Networks

A Simple Neural Network

3 Layers, k Observations



Weights from Layer 0 (L_0) to Layer 1 (L_1)

$$w_{11}, w_{12}, w_{13}$$

To Neuron (i)
(in Next Layer)

$$w_{13}$$

From Neuron (j)
(in Previous layer)

Weight from Neuron 3 of Layer 0 to Neuron 1 of Layer 1

$n_0 = 3$
Input Layer (L_0)

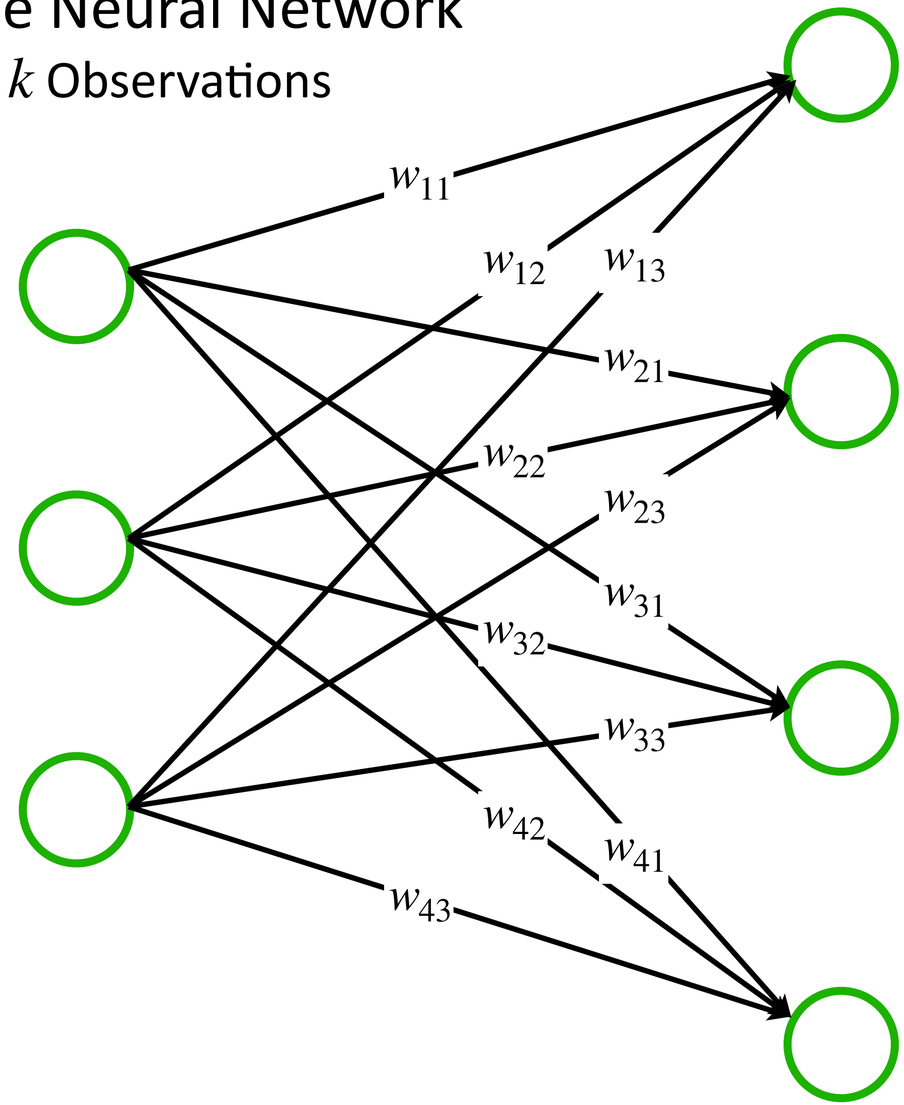
$n_1 = 4$
Hidden Layer (L_1)

$n_2 = 2$
Output Layer (L_2)

Neural Networks

A Simple Neural Network

3 Layers, k Observations



Weights from Layer 1 (L_1) to Layer 2 (L_2)

$$W_1 = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \\ w_{41} & w_{42} & w_{43} \end{bmatrix}$$

$$4 \times 3$$

W_1 Is a $(n_1 \times n_0)$ Matrix of weights **from** Layer L_0 to Layer L_1

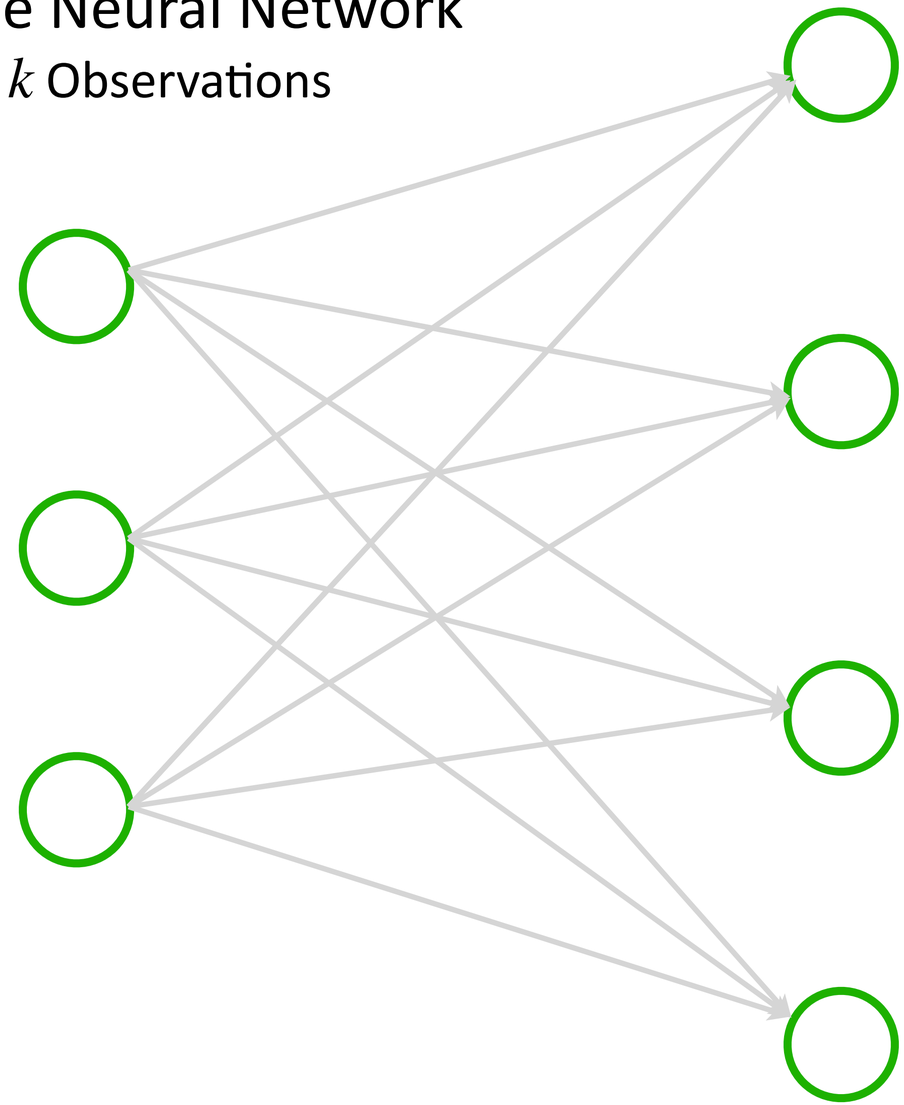
$n_0 = 3$
Input Layer (L_0)

$n_1 = 4$
Hidden Layer (L_1)

$n_2 = 2$
Output Layer (L_2)

A Simple Neural Network

3 Layers, k Observations



Biases in Layer 0 (L_0)

Layer 0 is an Input Layer and there are no biases

$n_0 = 3$

Input Layer (L_0)

$n_1 = 4$

Hidden Layer (L_1)

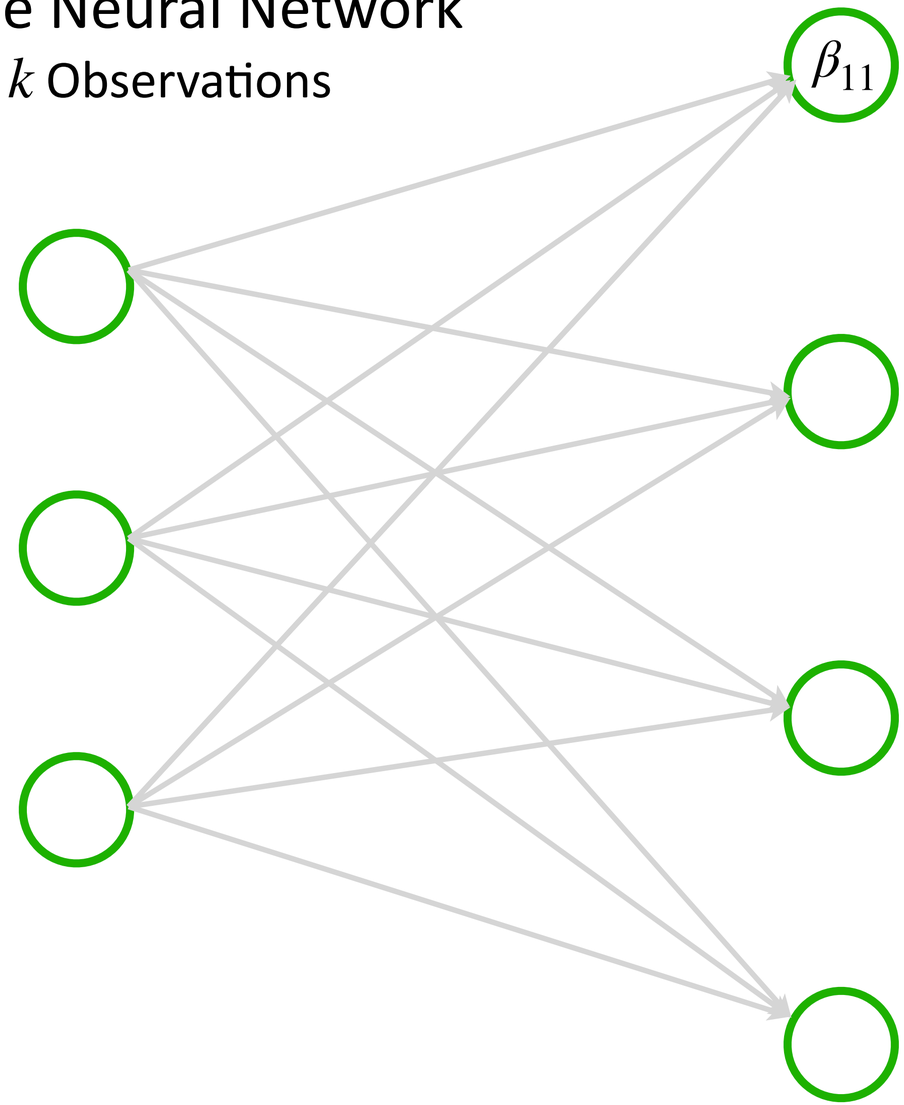
$n_2 = 2$

Output Layer (L_2)

Neural Networks

A Simple Neural Network

3 Layers, k Observations



$n_0 = 3$

Input Layer (L_0)

$n_1 = 4$

Hidden Layer (L_1)

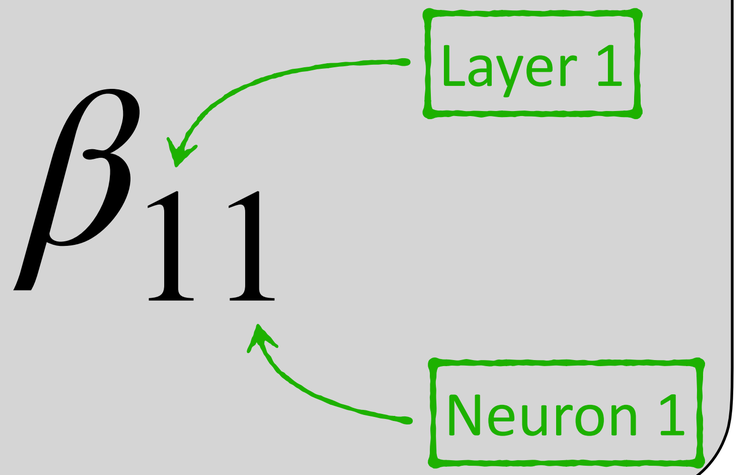
$n_2 = 2$

Output Layer (L_2)

Biases in Layer 1 (L_1)

Bias associated with Neuron 1 in Layer 1

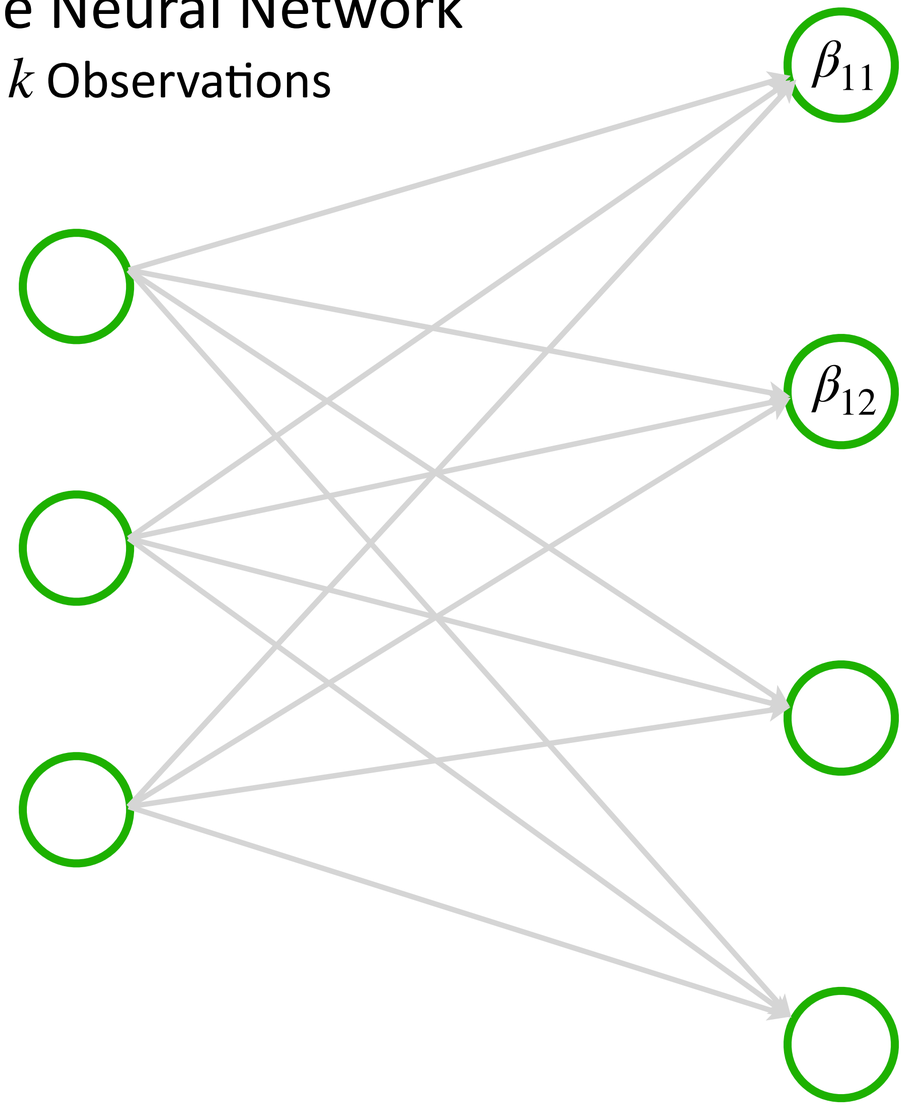
β_{11}



Neural Networks

A Simple Neural Network

3 Layers, k Observations



$n_0 = 3$

Input Layer (L_0)

$n_1 = 4$

Hidden Layer (L_1)

$n_2 = 2$

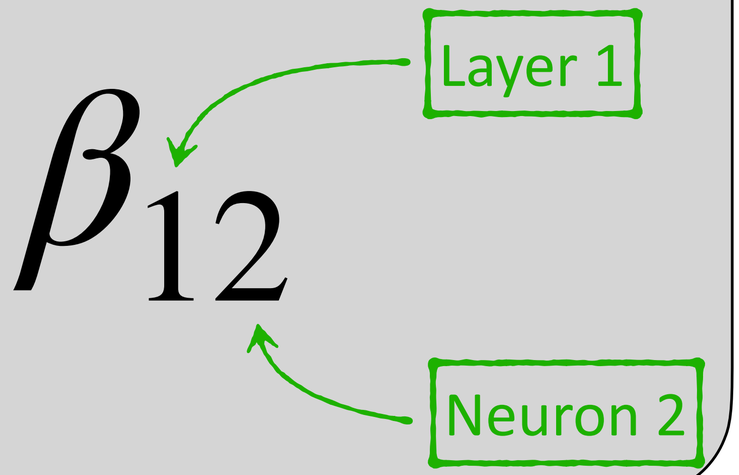
Output Layer (L_2)

Biases in Layer 1 (L_1)

Bias associated with Neuron 1 in Layer 1

β_{11}

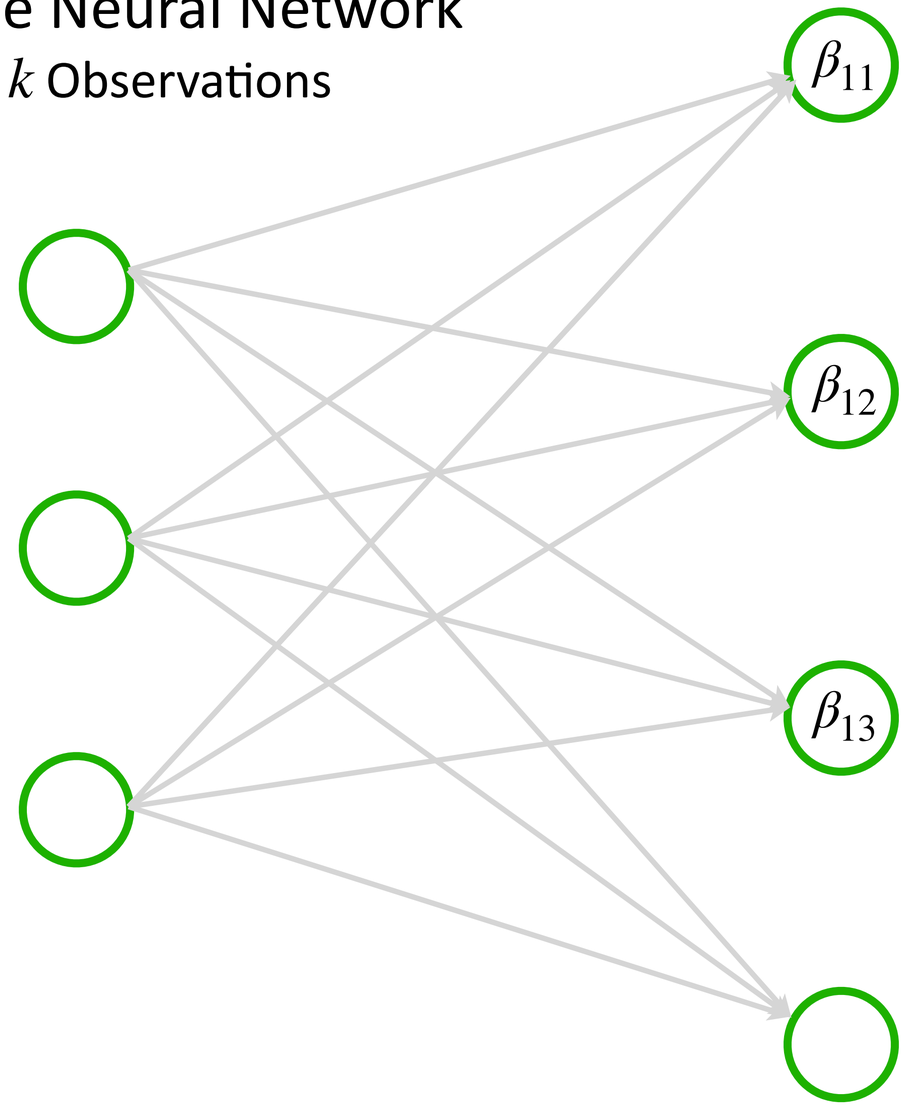
β_{12}



Neural Networks

A Simple Neural Network

3 Layers, k Observations



$n_0 = 3$

Input Layer (L_0)

$n_1 = 4$

Hidden Layer (L_1)

$n_2 = 2$

Output Layer (L_2)

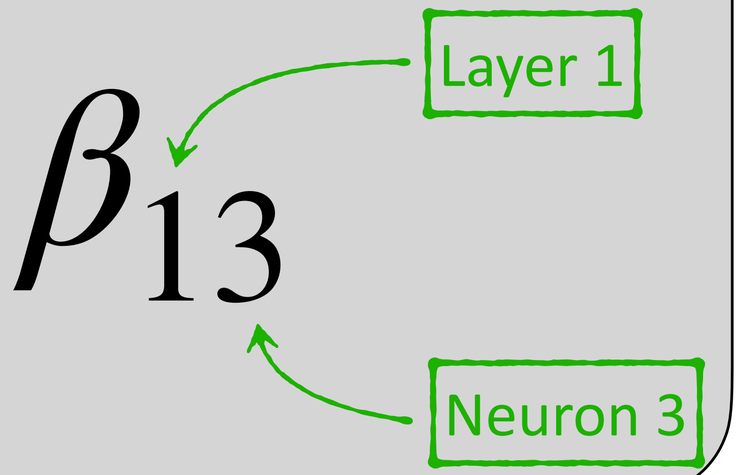
Biases in Layer 1 (L_1)

Bias associated with Neuron 1 in Layer 1

β_{11}

β_{12}

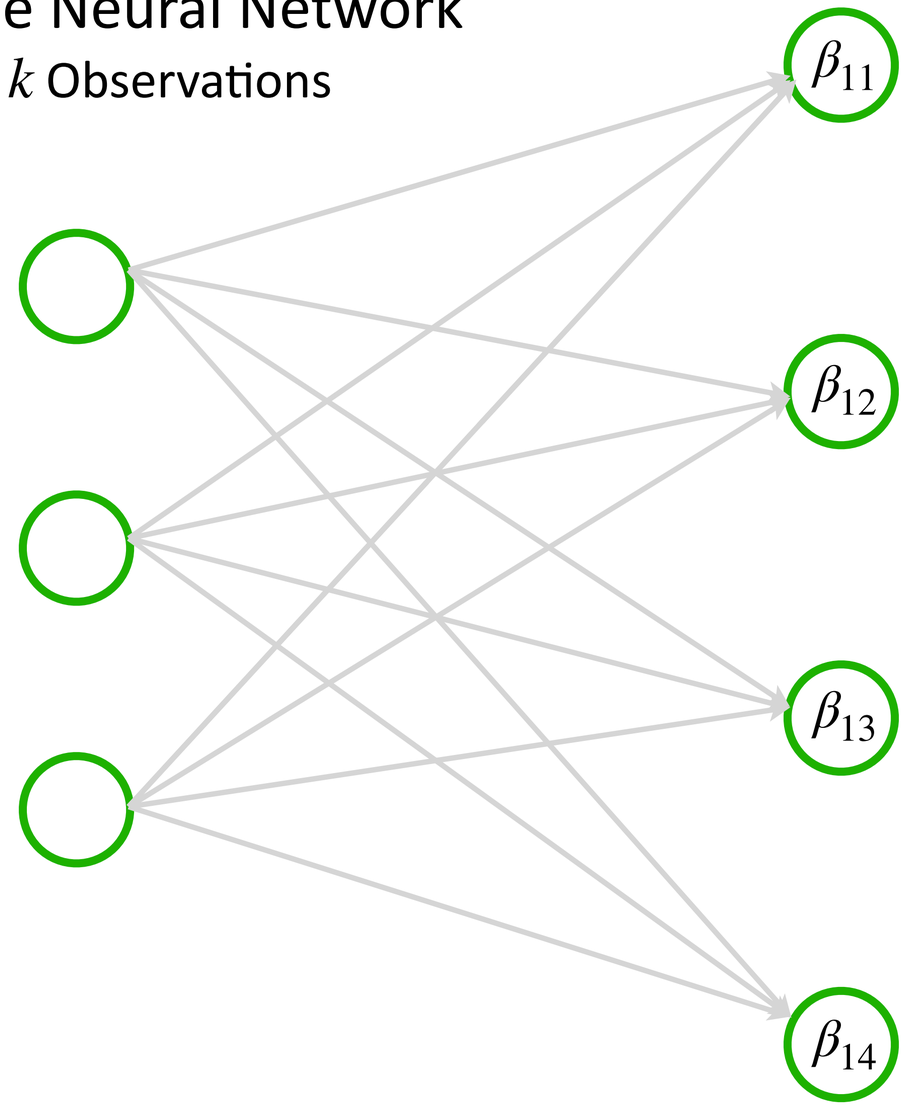
β_{13}



Neural Networks

A Simple Neural Network

3 Layers, k Observations



$n_0 = 3$

Input Layer (L_0)

$n_1 = 4$

Hidden Layer (L_1)

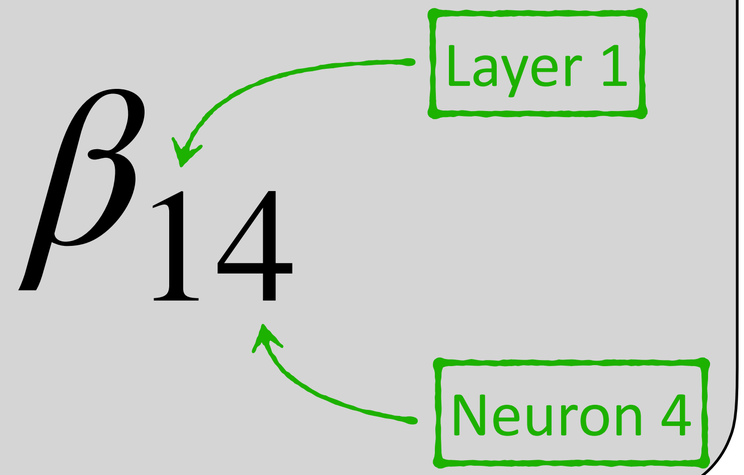
$n_2 = 2$

Output Layer (L_2)

Biases in Layer 1 (L_1)

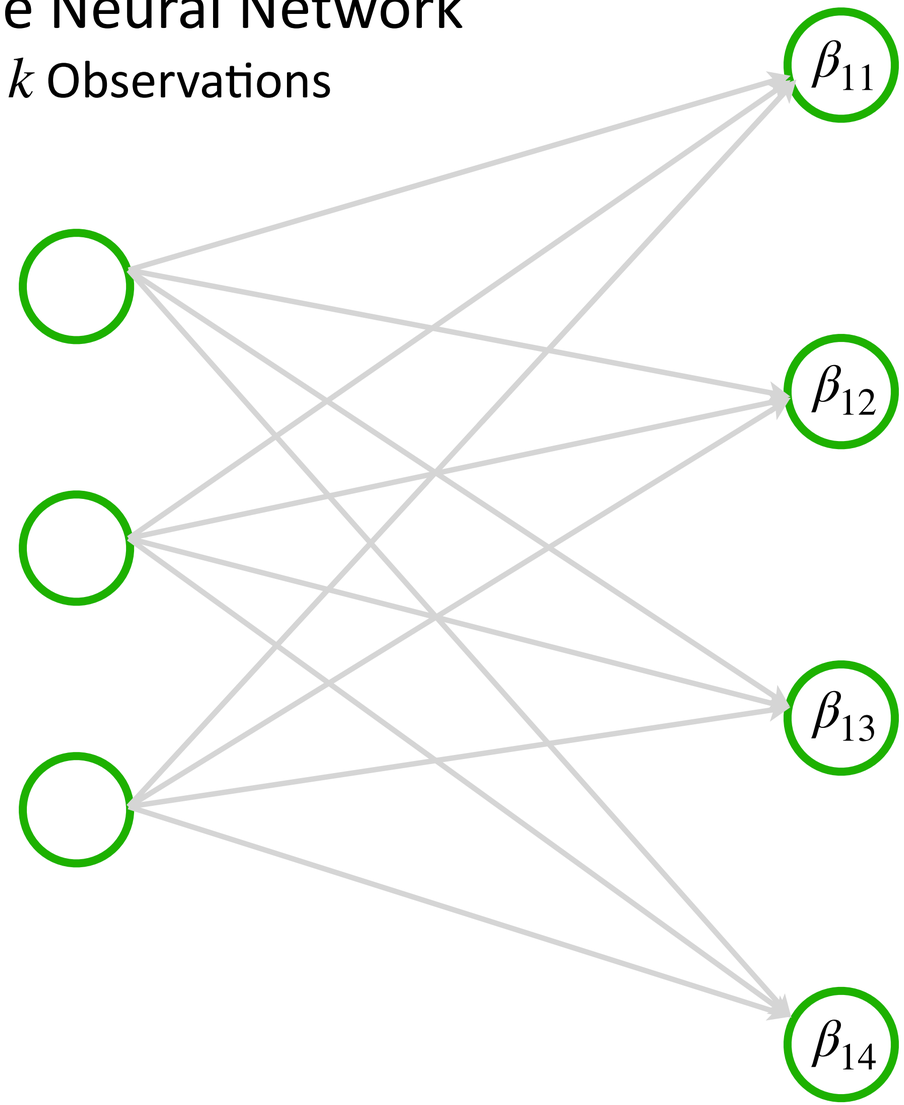
Bias associated with Neuron 1 in Layer 1

β_{11}
β_{12}
β_{13}
β_{14}



A Simple Neural Network

3 Layers, k Observations



$n_0 = 3$

Input Layer (L_0)

$n_1 = 4$

Hidden Layer (L_1)

$n_2 = 2$

Output Layer (L_2)

Biases in Layer 1 (L_1)

Bias associated with Neuron 1 in Layer 1

$$\beta_1 = \begin{bmatrix} \beta_{11} \\ \beta_{12} \\ \beta_{13} \\ \beta_{14} \end{bmatrix}$$

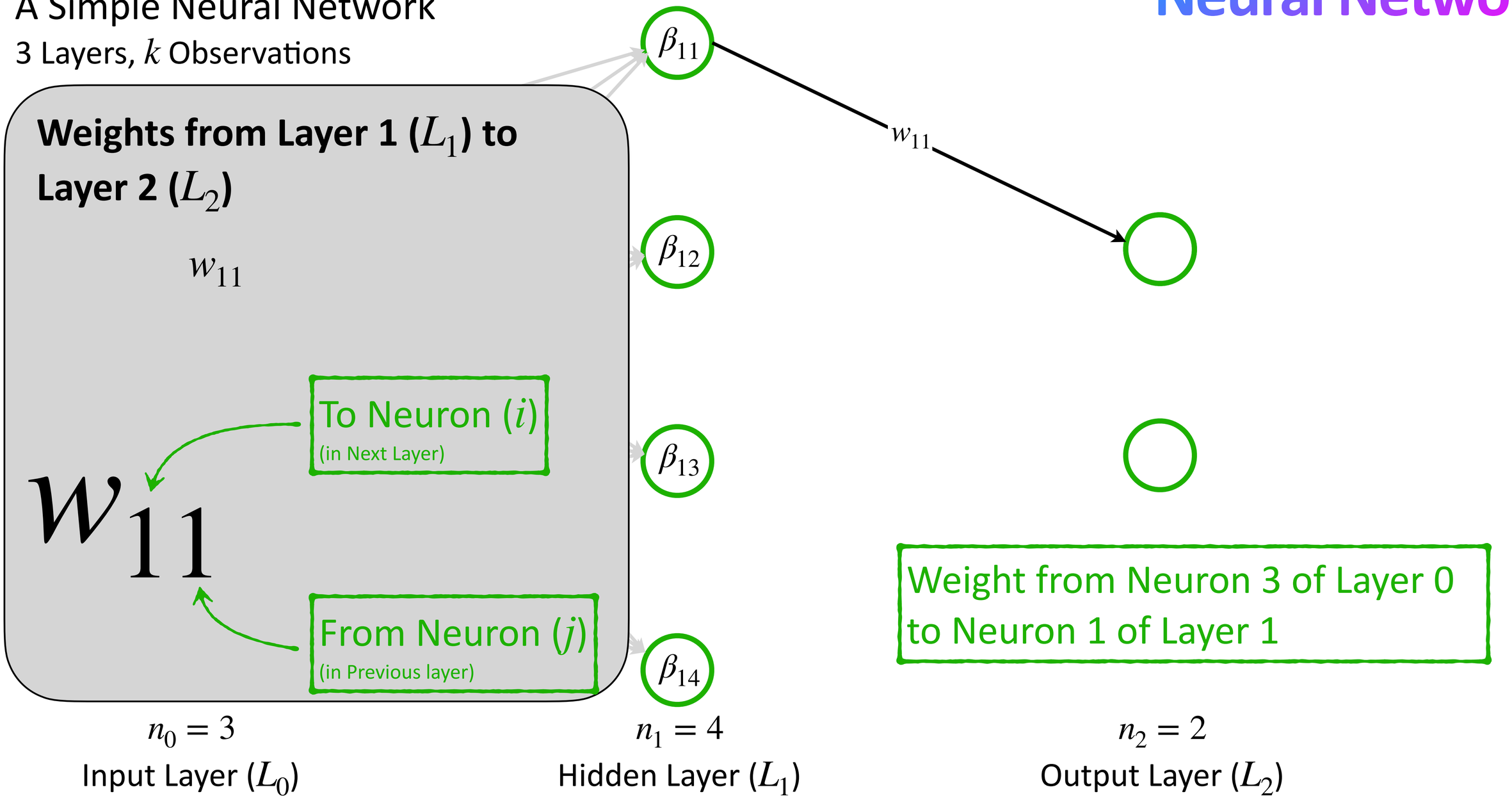
4×1

β_1 Is a $(n_1 \times 1)$ Vector of Biases in Layer L_1

Neural Networks

A Simple Neural Network

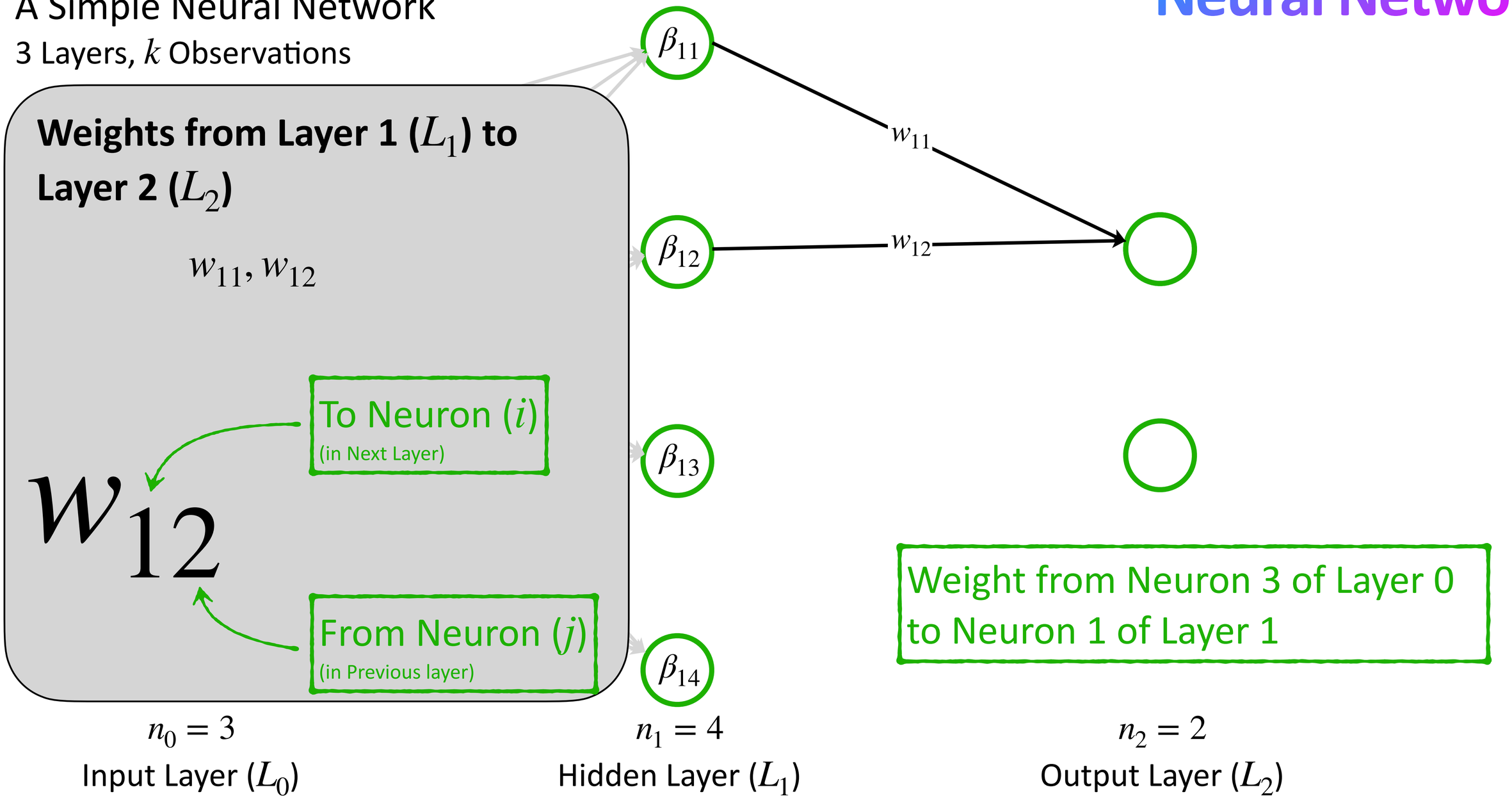
3 Layers, k Observations



Neural Networks

A Simple Neural Network

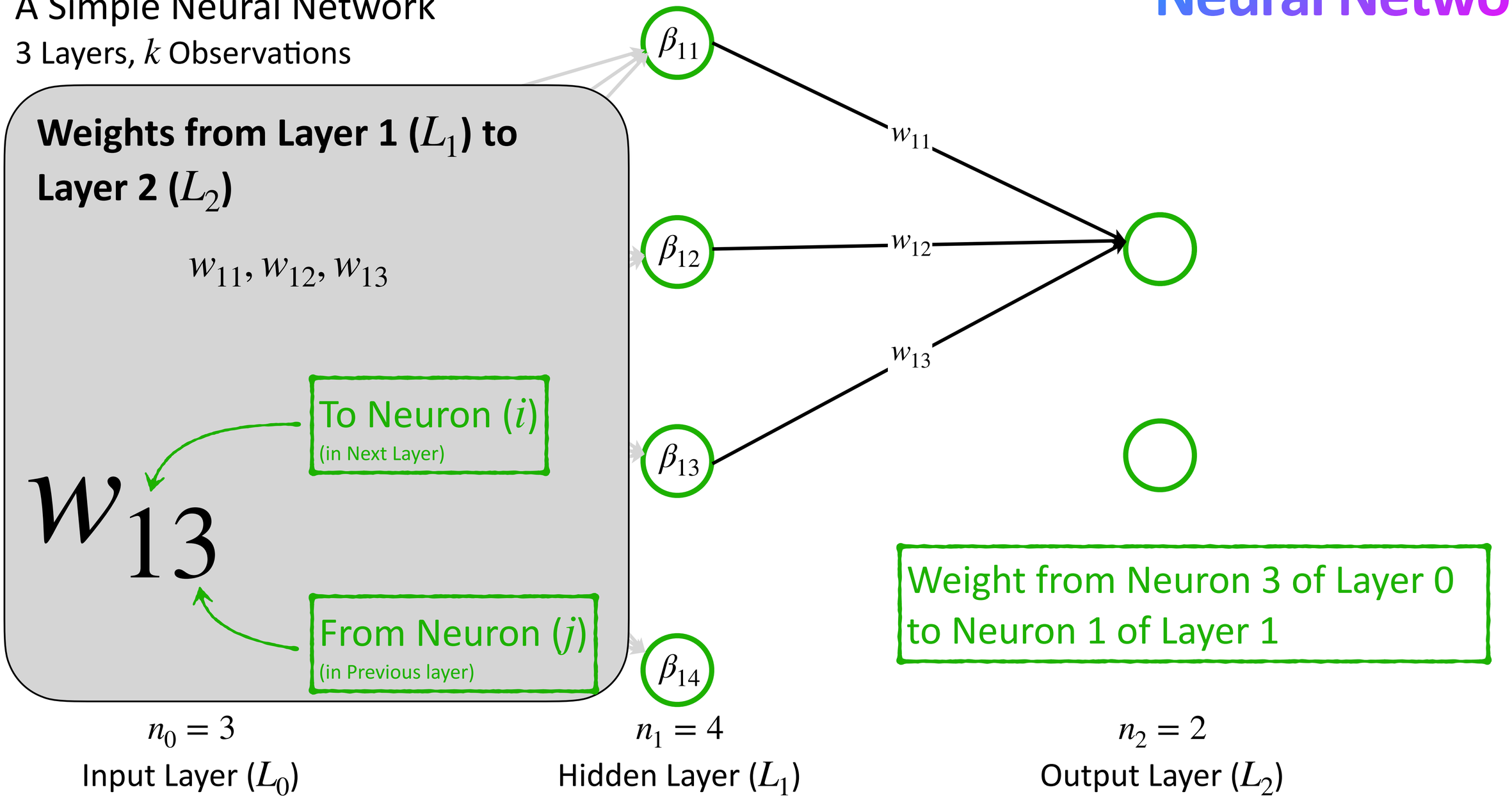
3 Layers, k Observations



Neural Networks

A Simple Neural Network

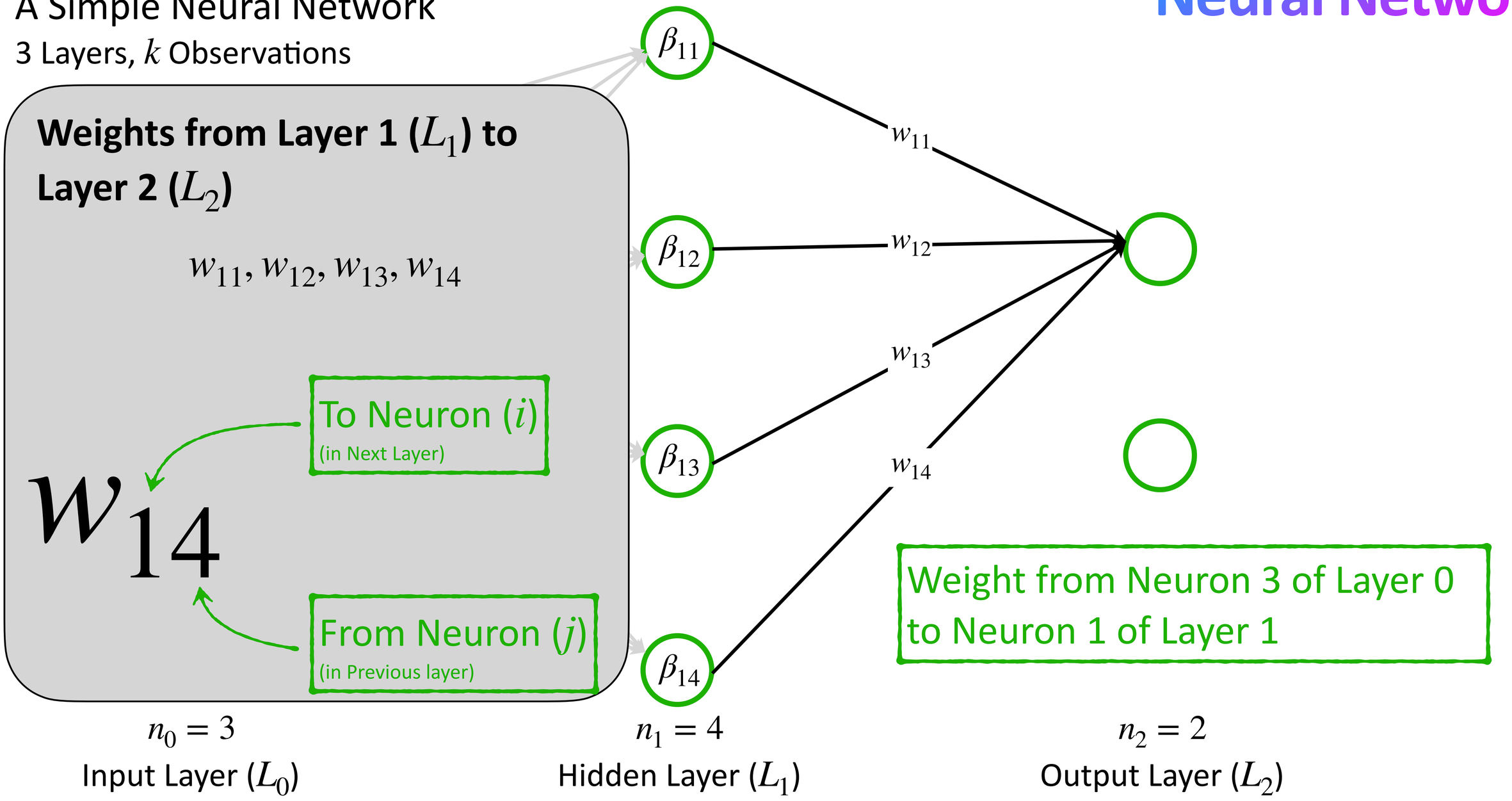
3 Layers, k Observations



Neural Networks

A Simple Neural Network

3 Layers, k Observations



Neural Networks

A Simple Neural Network

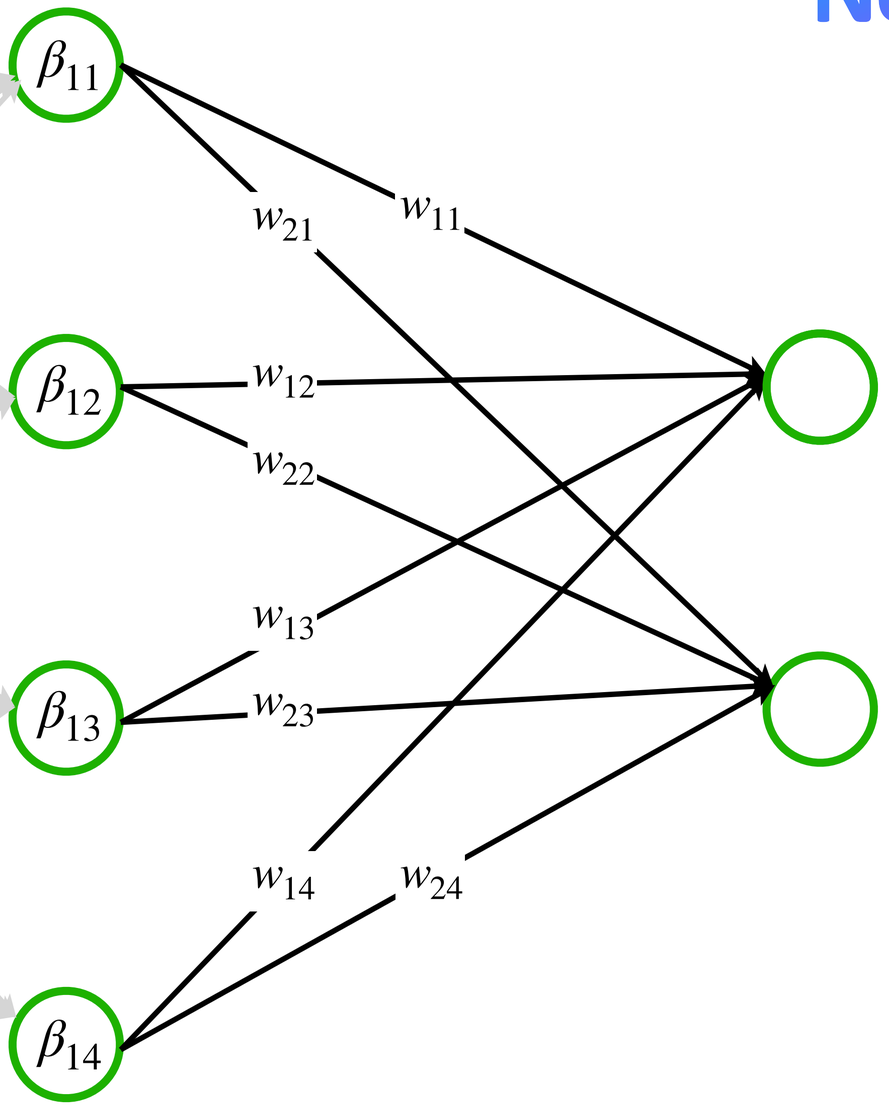
3 Layers, k Observations

Weights from Layer 1 (L_1) to Layer 2 (L_2)

$$W_2 = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \end{bmatrix}$$

$$2 \times 4$$

W_2 Is a $(n_2 \times n_1)$ Matrix of weights **from** Layer L_0 to Layer L_1



$n_0 = 3$

Input Layer (L_0)

$n_1 = 4$

Hidden Layer (L_1)

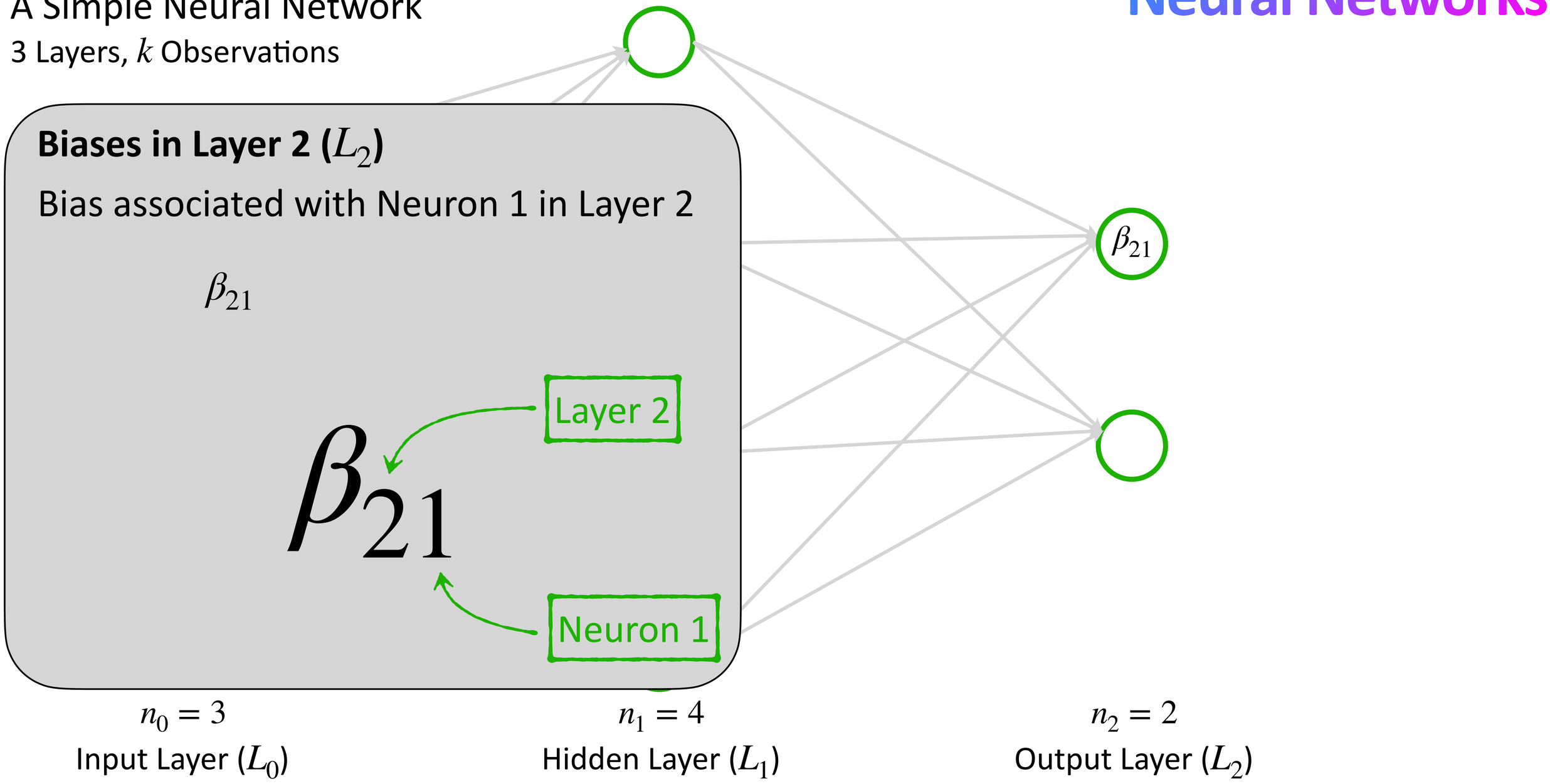
$n_2 = 2$

Output Layer (L_2)

Neural Networks

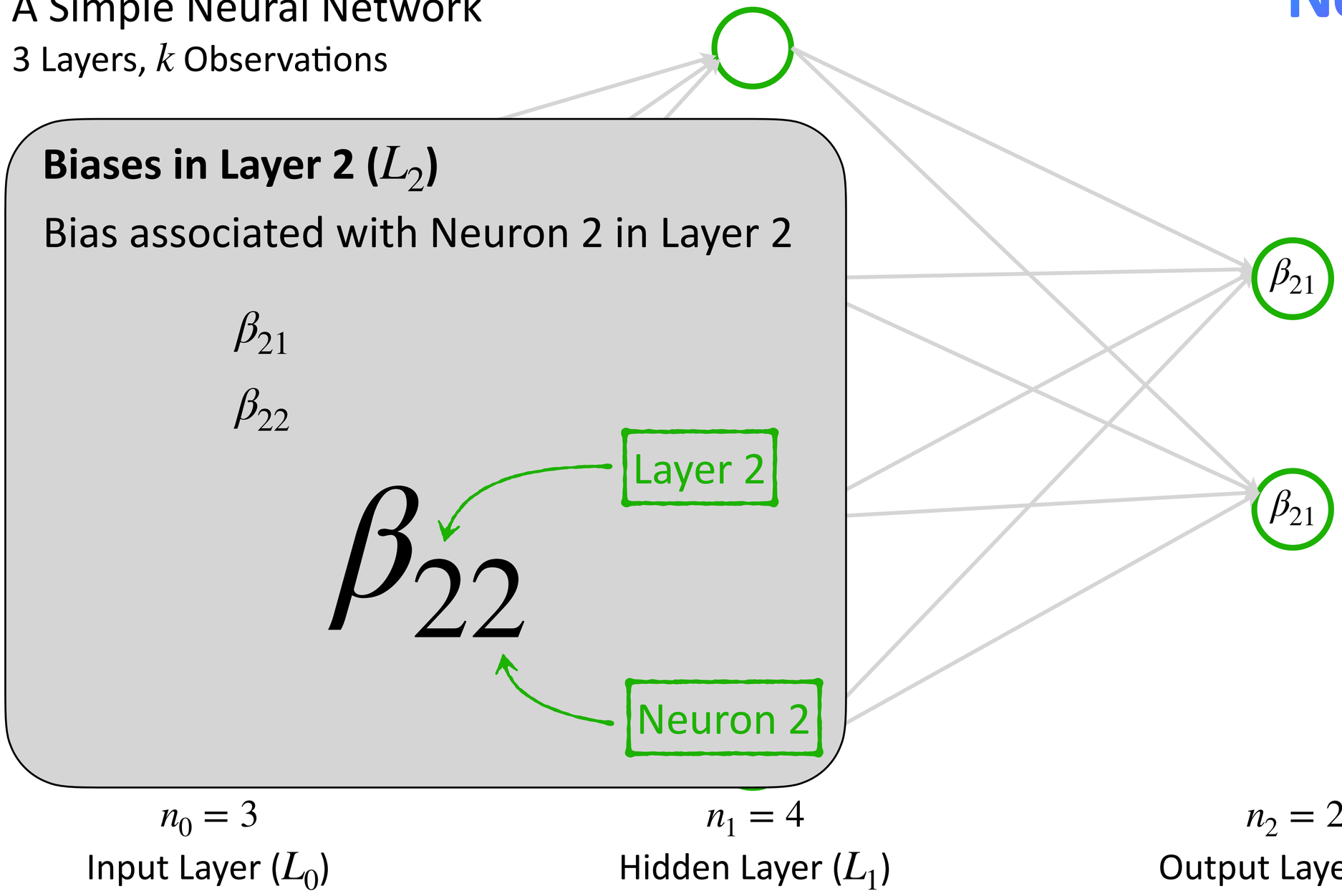
A Simple Neural Network

3 Layers, k Observations



A Simple Neural Network

3 Layers, k Observations



$n_0 = 3$

Input Layer (L_0)

$n_1 = 4$

Hidden Layer (L_1)

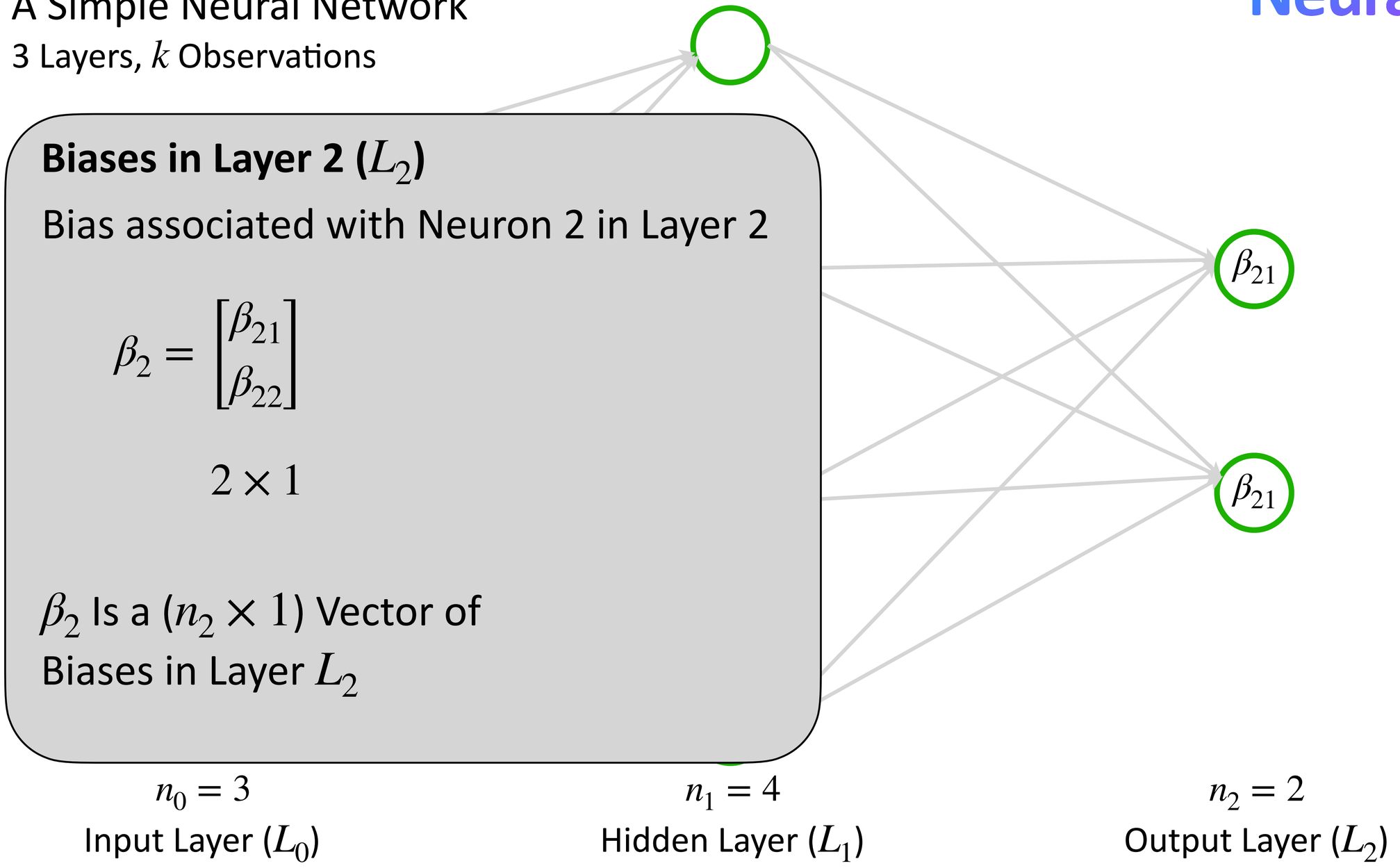
$n_2 = 2$

Output Layer (L_2)

Neural Networks

A Simple Neural Network

3 Layers, k Observations



Biases in Layer 2 (L_2)

Bias associated with Neuron 2 in Layer 2

$$\beta_2 = \begin{bmatrix} \beta_{21} \\ \beta_{22} \end{bmatrix}$$

$$2 \times 1$$

β_2 Is a $(n_2 \times 1)$ Vector of Biases in Layer L_2

$n_0 = 3$
Input Layer (L_0)

$n_1 = 4$
Hidden Layer (L_1)

$n_2 = 2$
Output Layer (L_2)

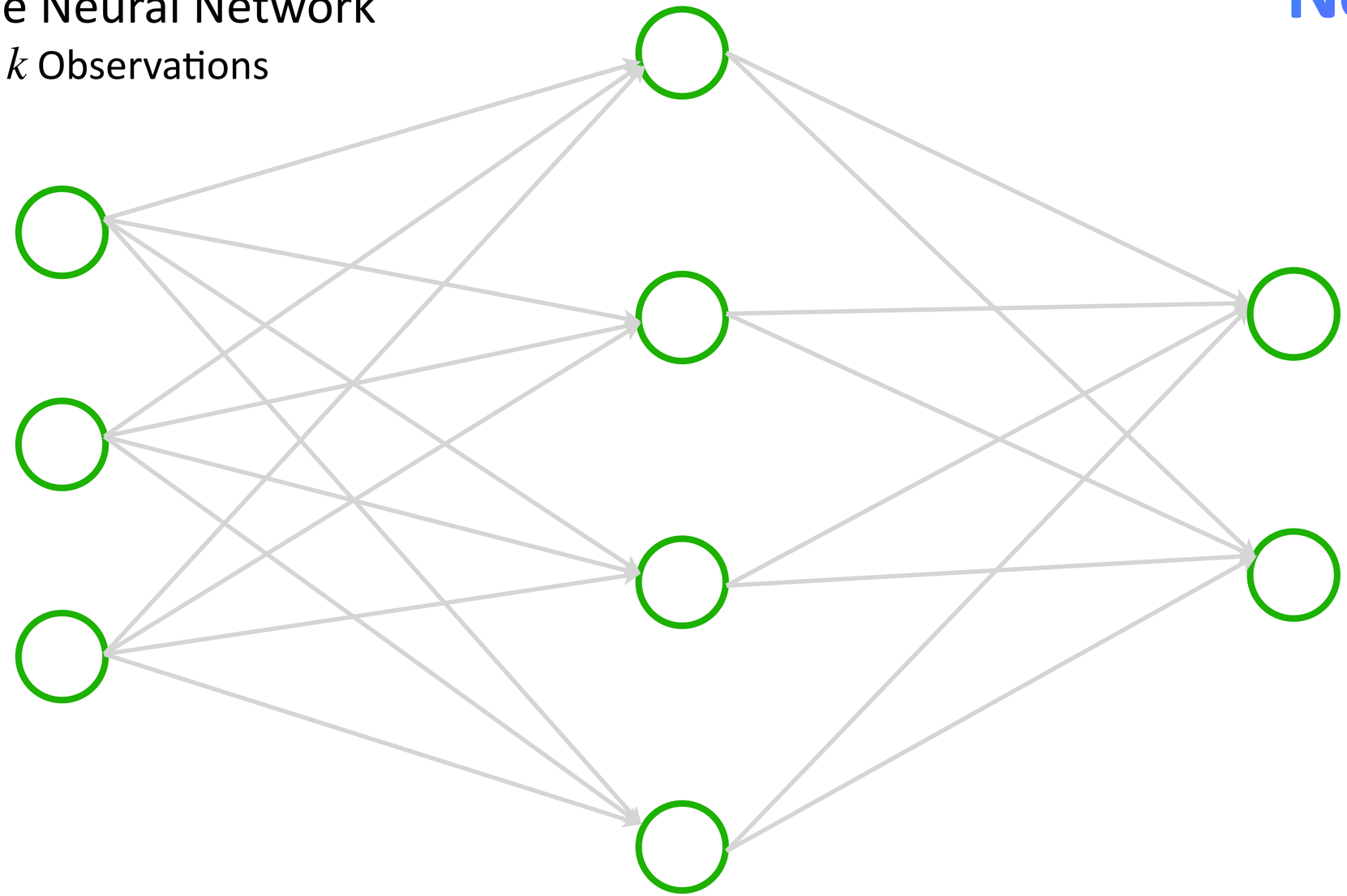
Neural Networks

Lets put it all together…

Neural Networks

A Simple Neural Network

3 Layers, k Observations



$n_0 = 3$

Input Layer (L_0)

$n_1 = 4$

Hidden Layer (L_1)

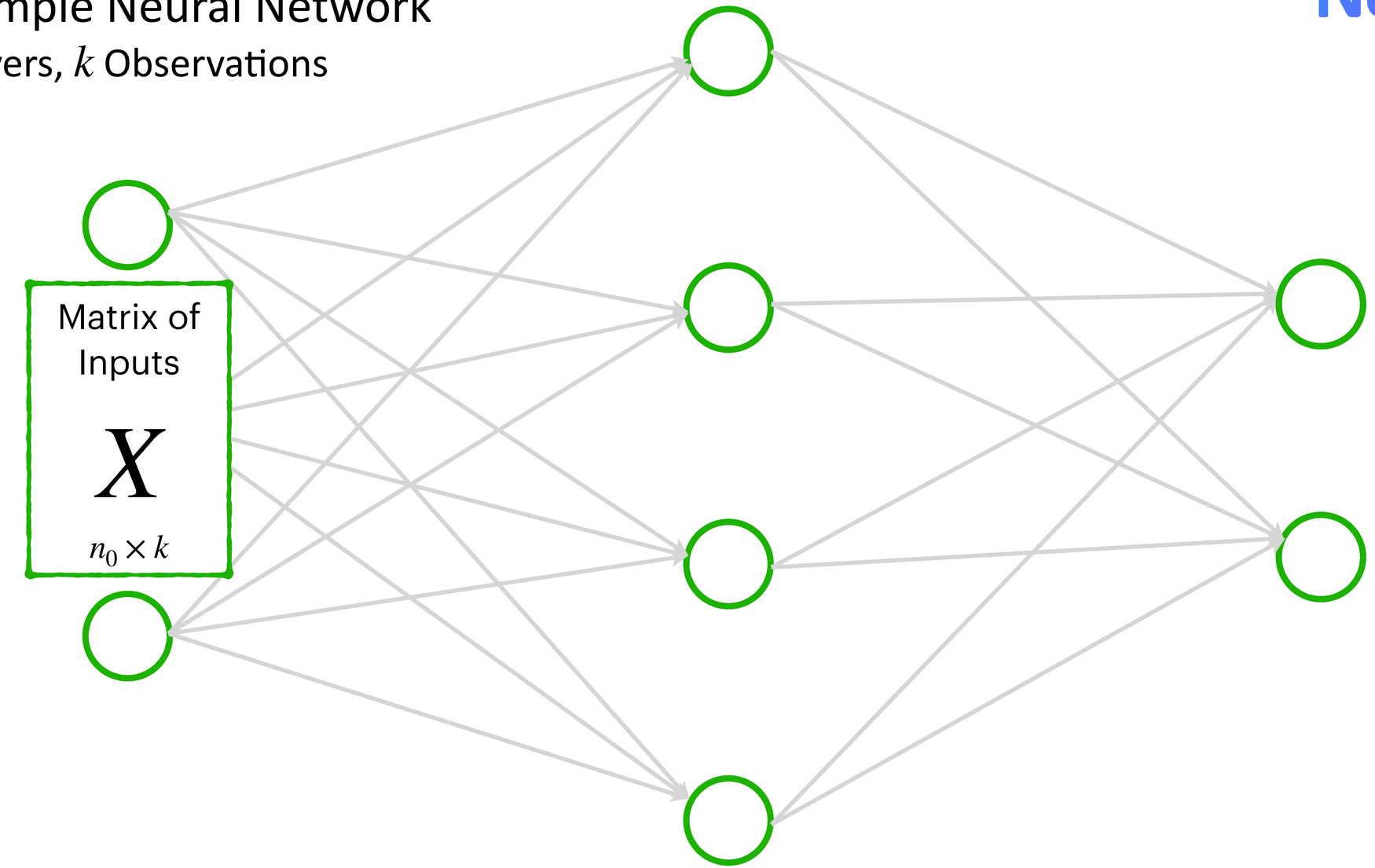
$n_2 = 2$

Output Layer (L_2)

Neural Networks

A Simple Neural Network

3 Layers, k Observations



$n_0 = 3$

Input Layer (L_0)

$n_1 = 4$

Hidden Layer (L_1)

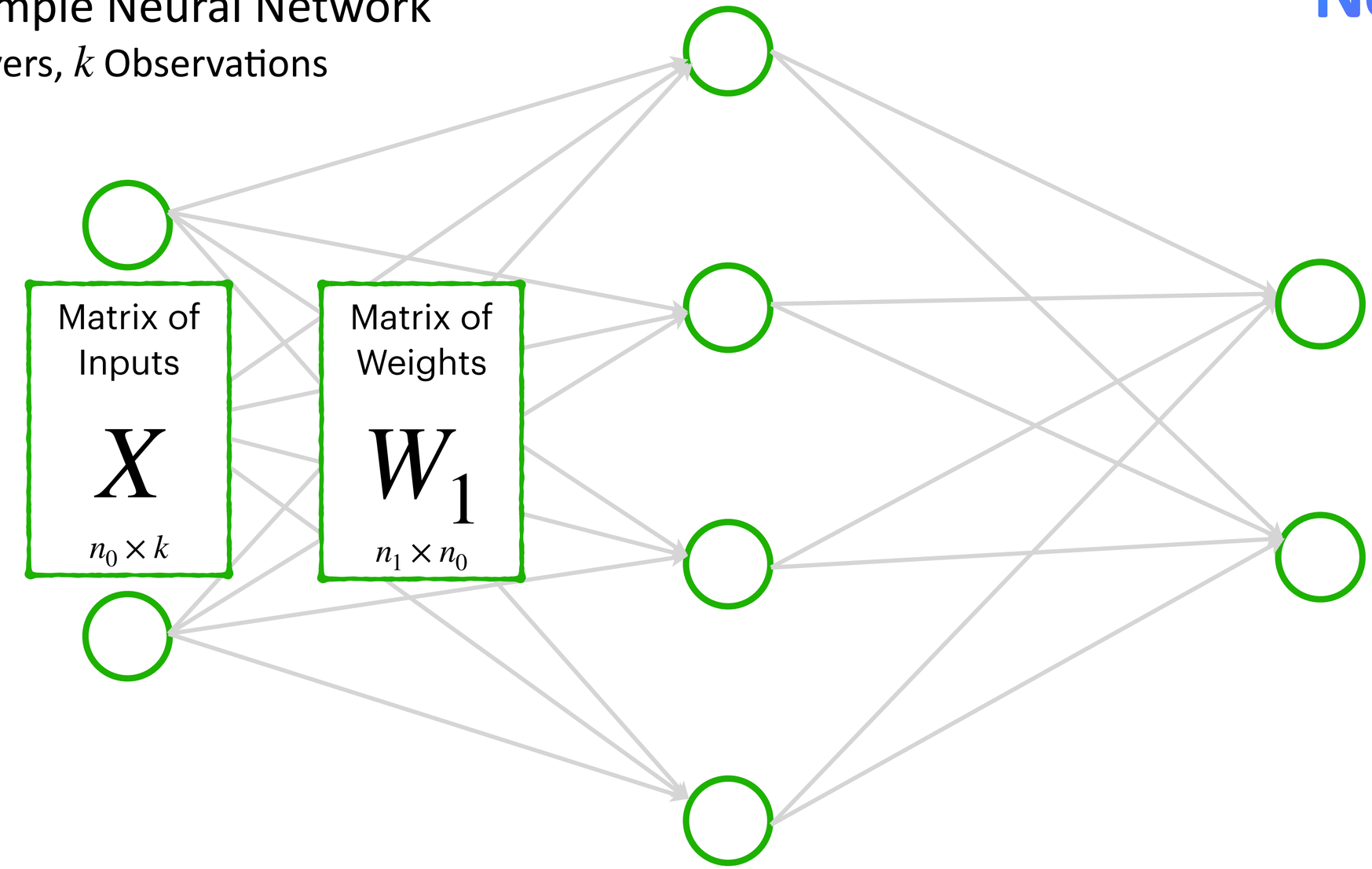
$n_2 = 2$

Output Layer (L_2)

Neural Networks

A Simple Neural Network

3 Layers, k Observations



$n_0 = 3$

Input Layer (L_0)

$n_1 = 4$

Hidden Layer (L_1)

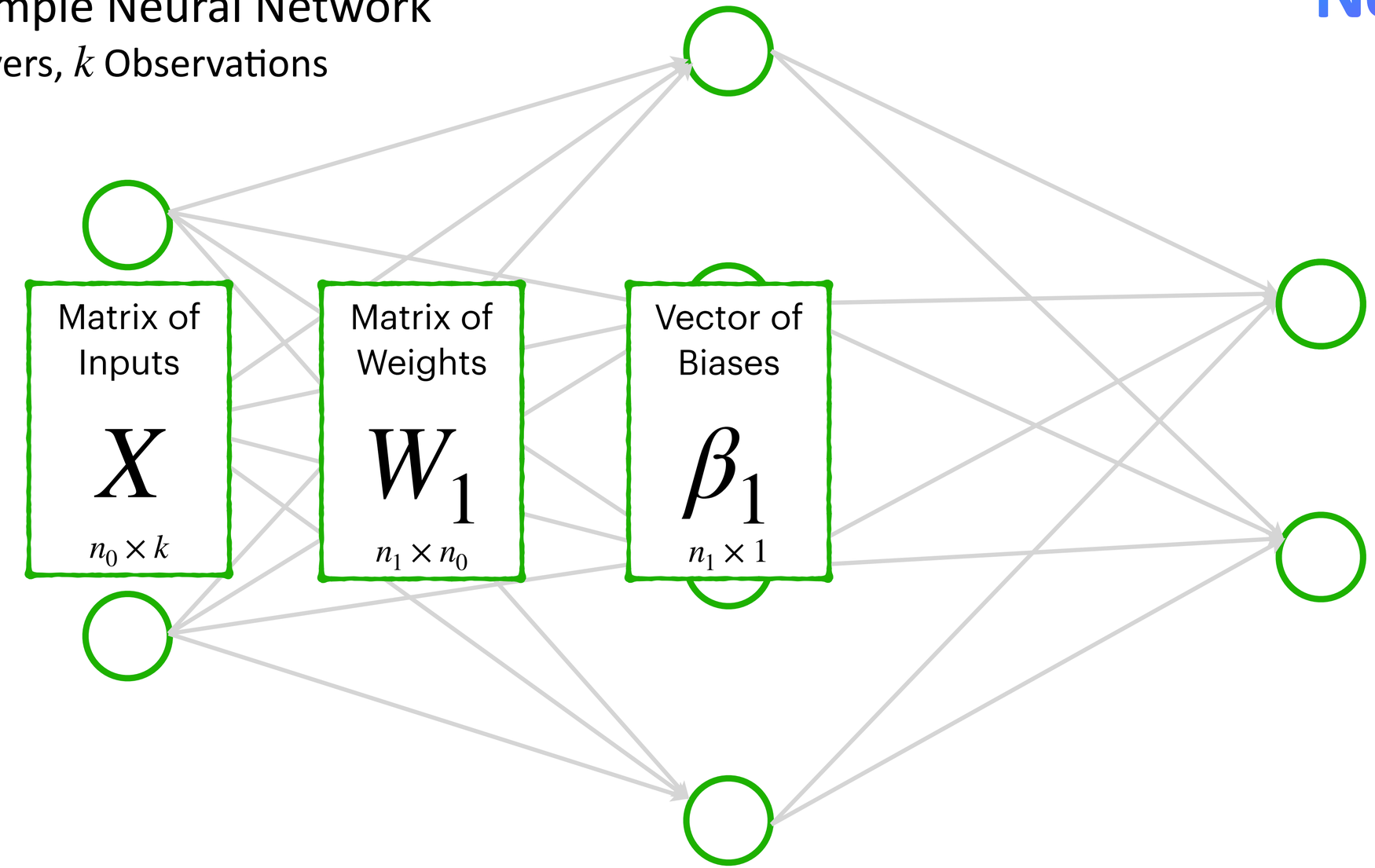
$n_2 = 2$

Output Layer (L_2)

Neural Networks

A Simple Neural Network

3 Layers, k Observations



$n_0 = 3$

Input Layer (L_0)

$n_1 = 4$

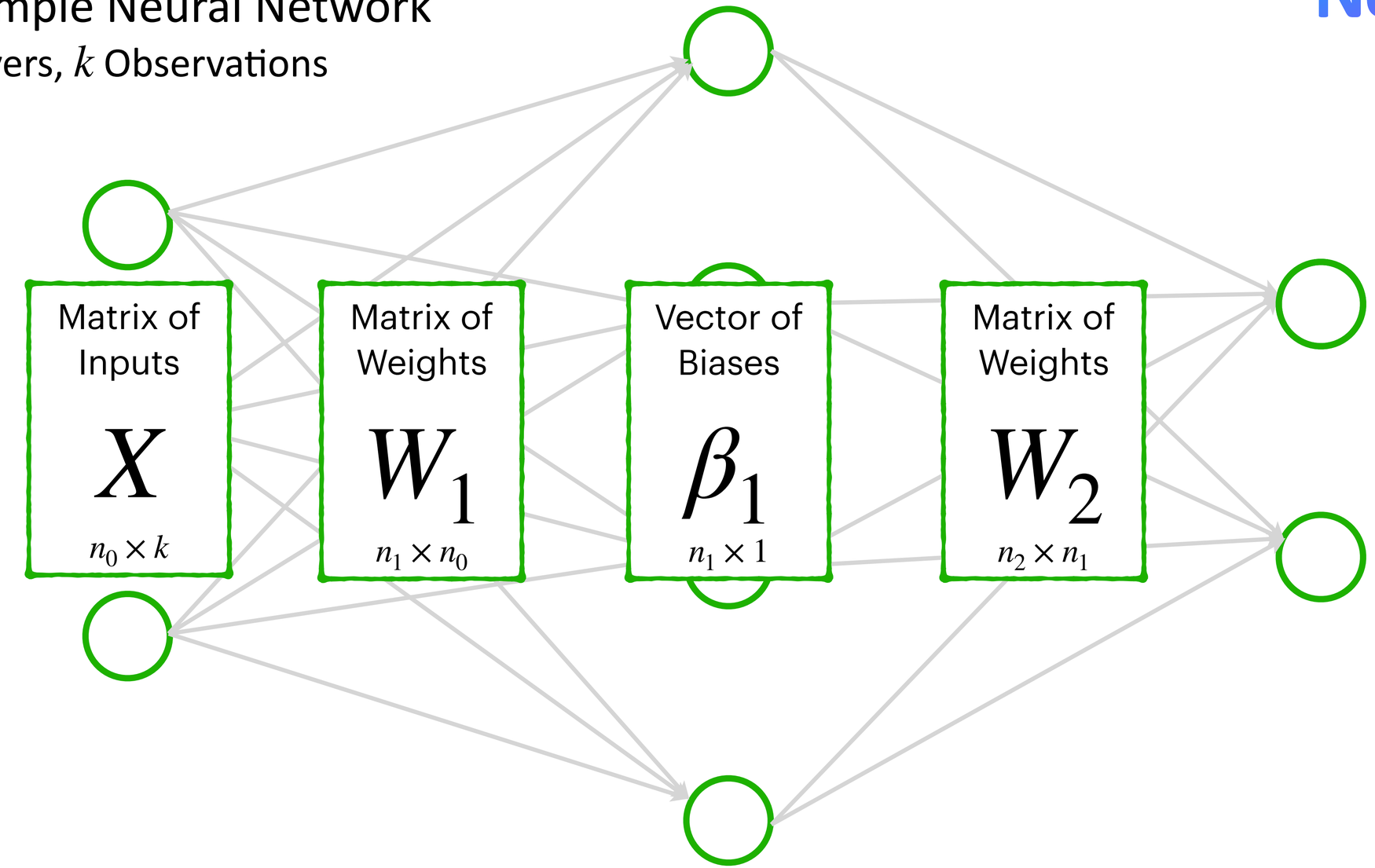
Hidden Layer (L_1)

$n_2 = 2$

Output Layer (L_2)

A Simple Neural Network

3 Layers, k Observations



$n_0 = 3$

Input Layer (L_0)

$n_1 = 4$

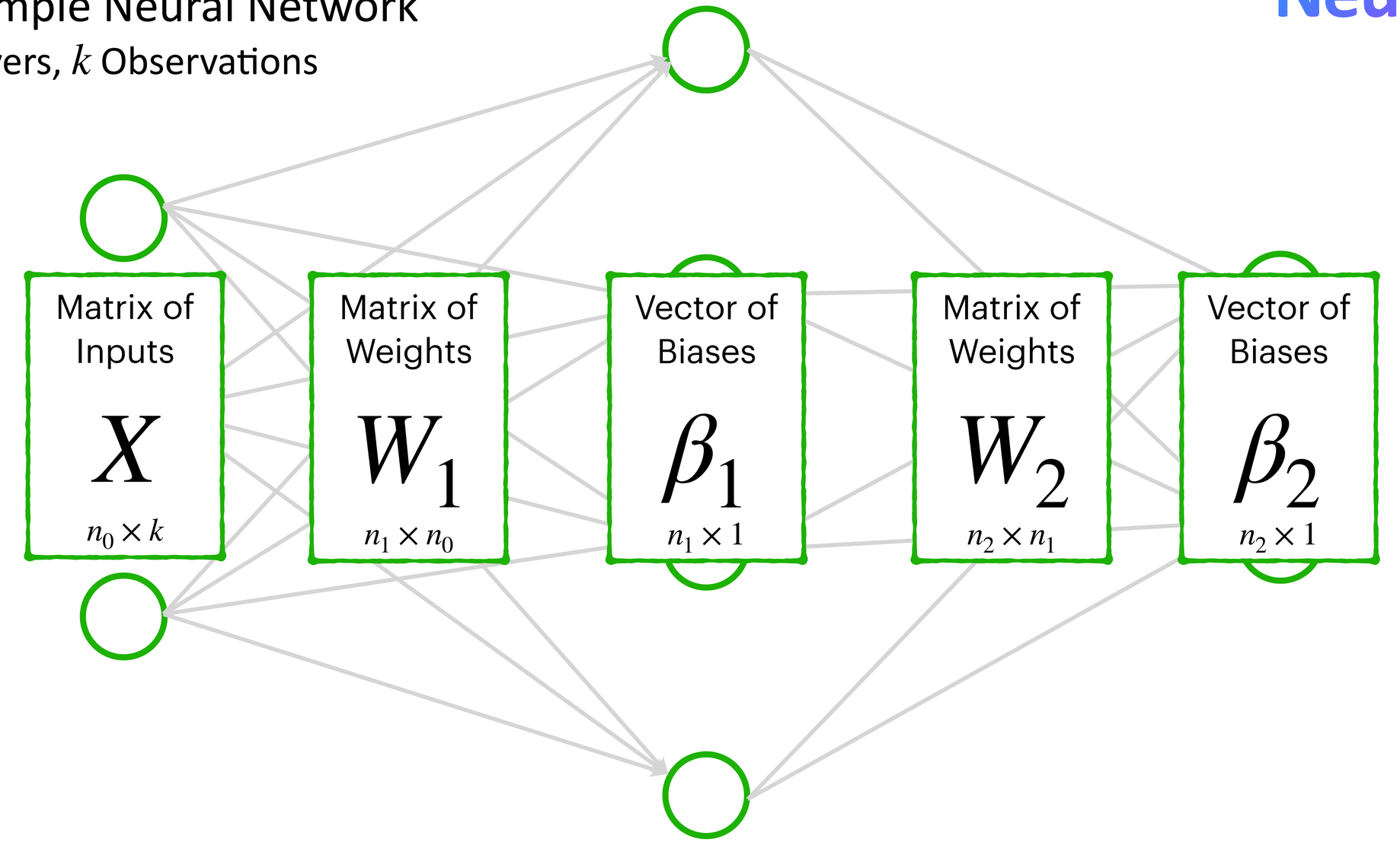
Hidden Layer (L_1)

$n_2 = 2$

Output Layer (L_2)

A Simple Neural Network

3 Layers, k Observations



$n_0 = 3$

Input Layer (L_0)

$n_1 = 4$

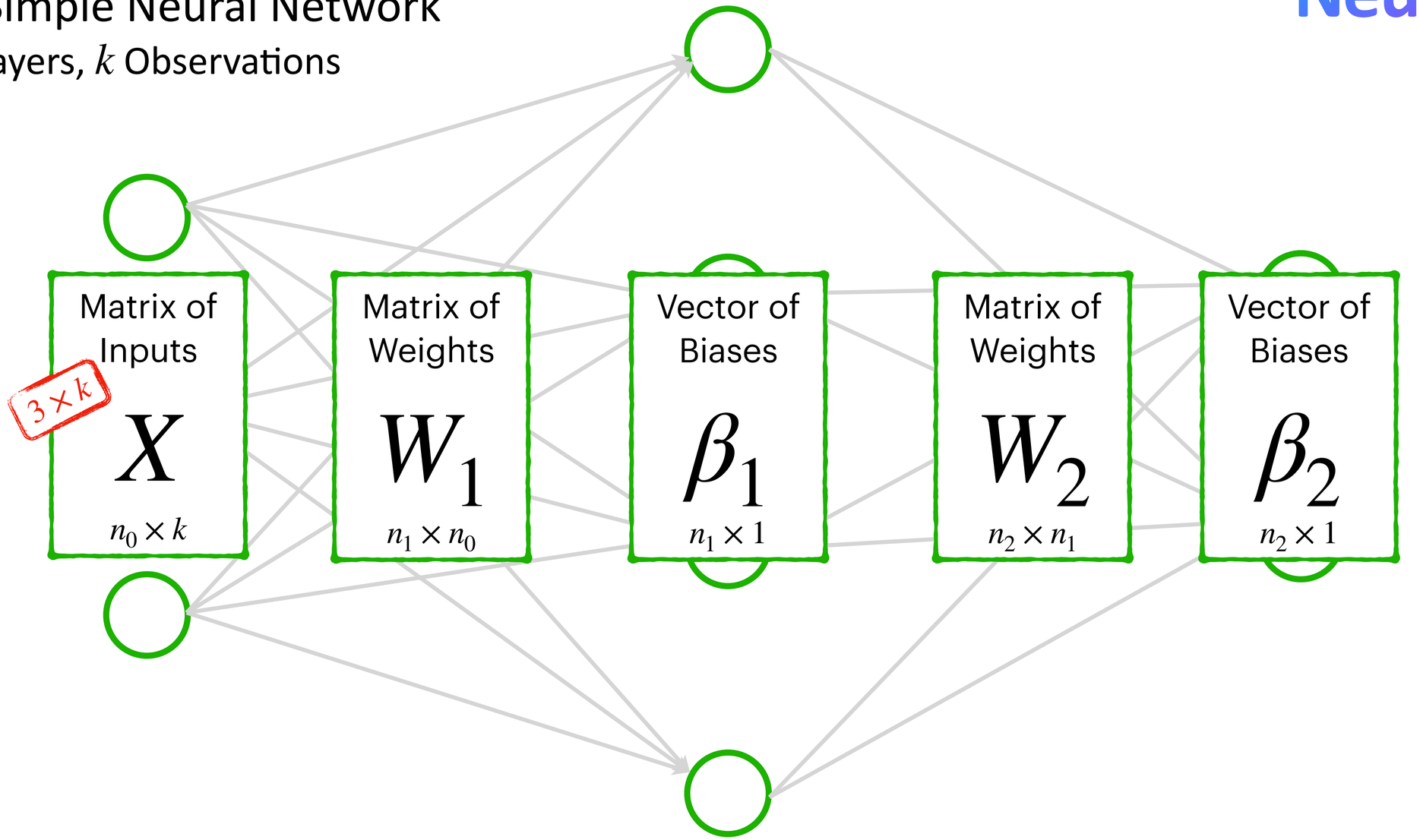
Hidden Layer (L_1)

$n_2 = 2$

Output Layer (L_2)

A Simple Neural Network

3 Layers, k Observations



$n_0 = 3$

Input Layer (L_0)

$n_1 = 4$

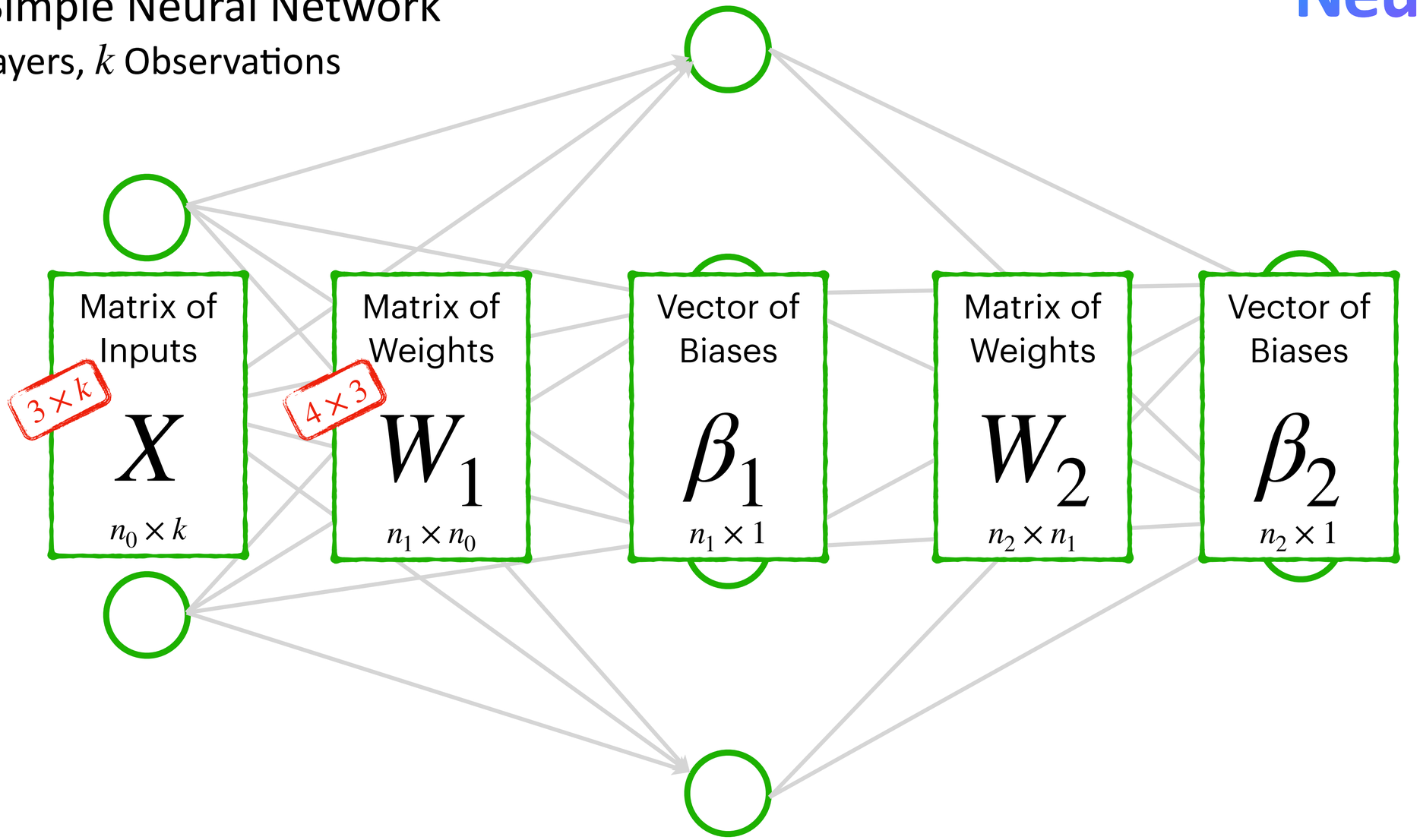
Hidden Layer (L_1)

$n_2 = 2$

Output Layer (L_2)

A Simple Neural Network

3 Layers, k Observations



$n_0 = 3$

Input Layer (L_0)

$n_1 = 4$

Hidden Layer (L_1)

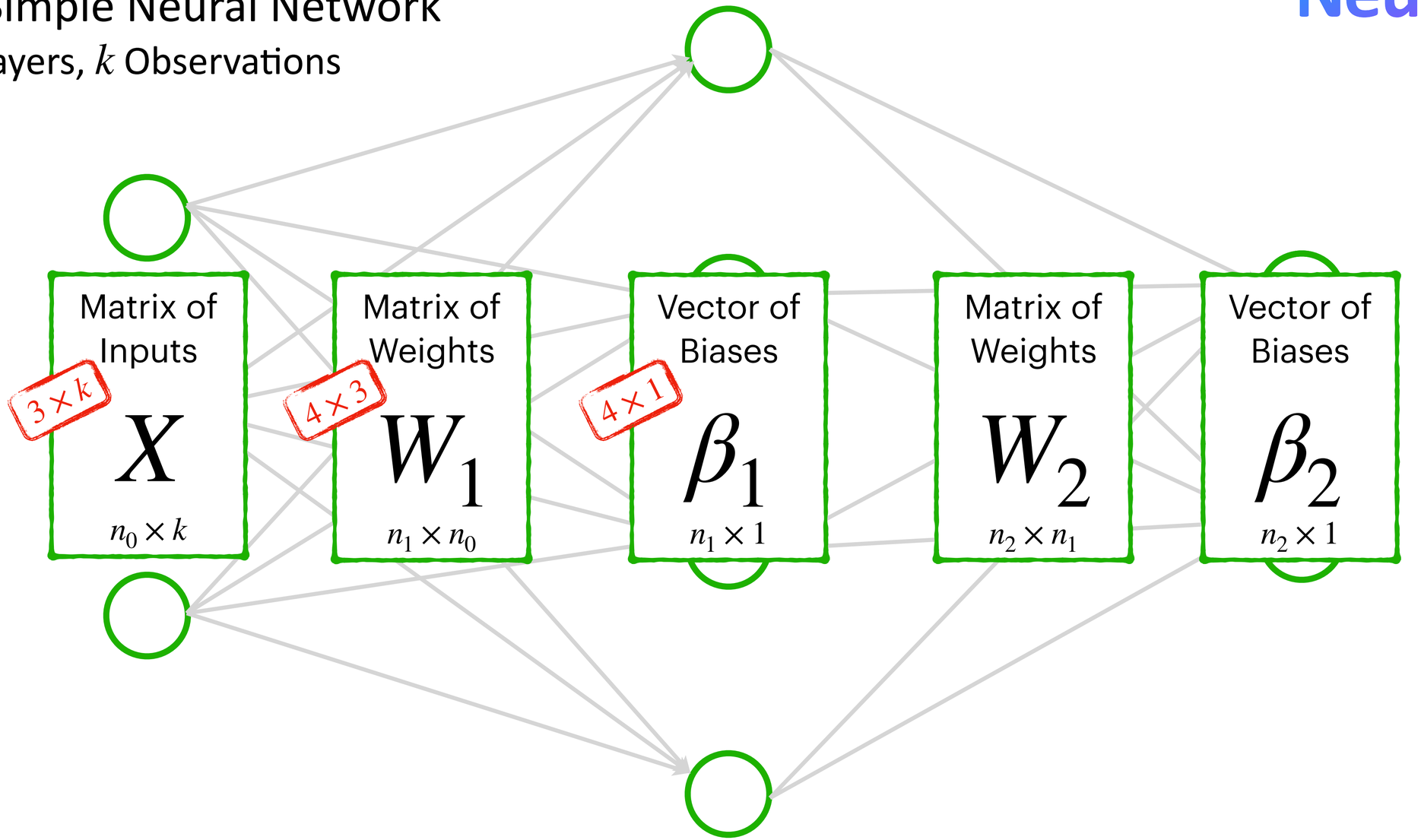
$n_2 = 2$

Output Layer (L_2)

Neural Networks

A Simple Neural Network

3 Layers, k Observations



$n_0 = 3$

Input Layer (L_0)

$n_1 = 4$

Hidden Layer (L_1)

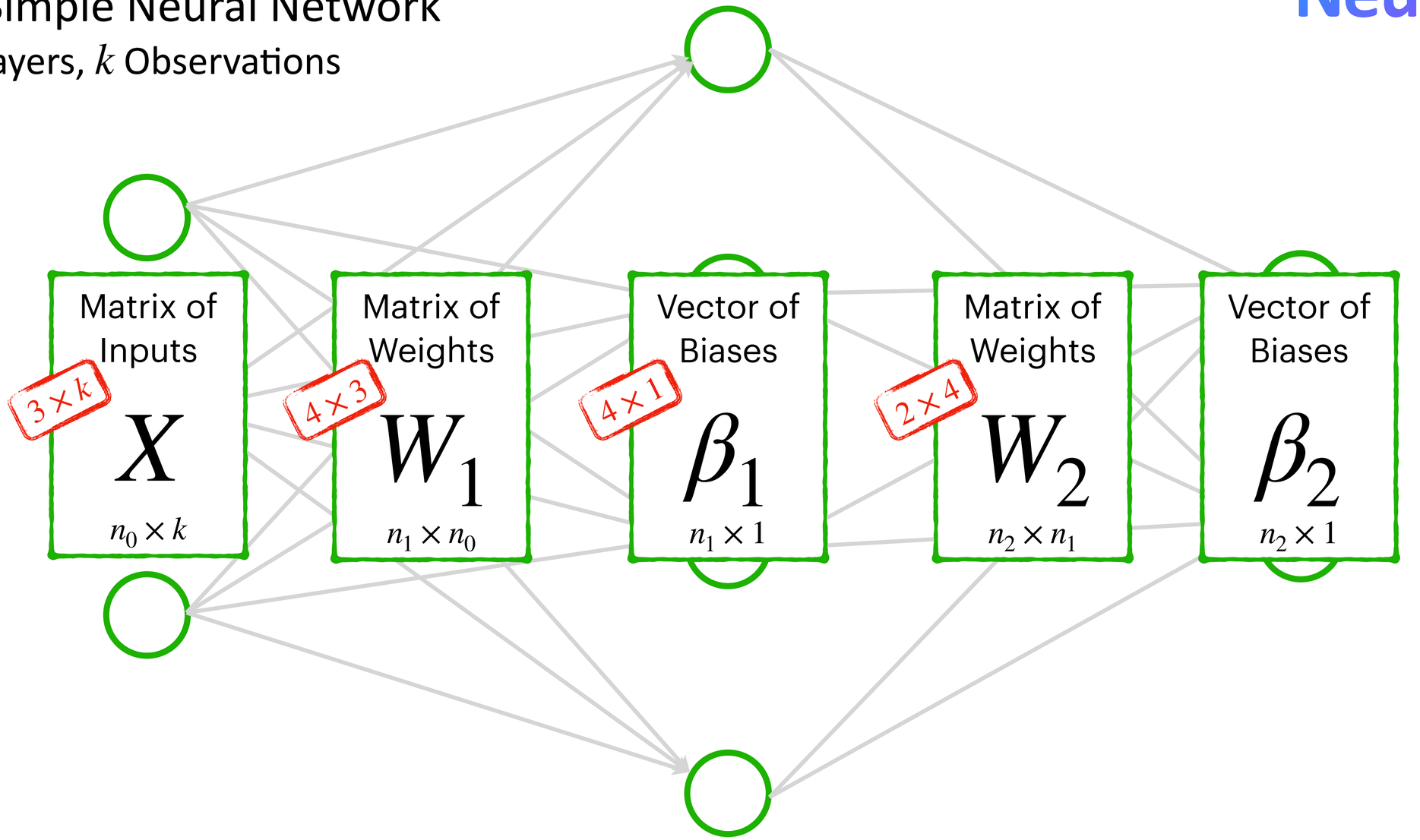
$n_2 = 2$

Output Layer (L_2)

Neural Networks

A Simple Neural Network

3 Layers, k Observations



$n_0 = 3$

Input Layer (L_0)

$n_1 = 4$

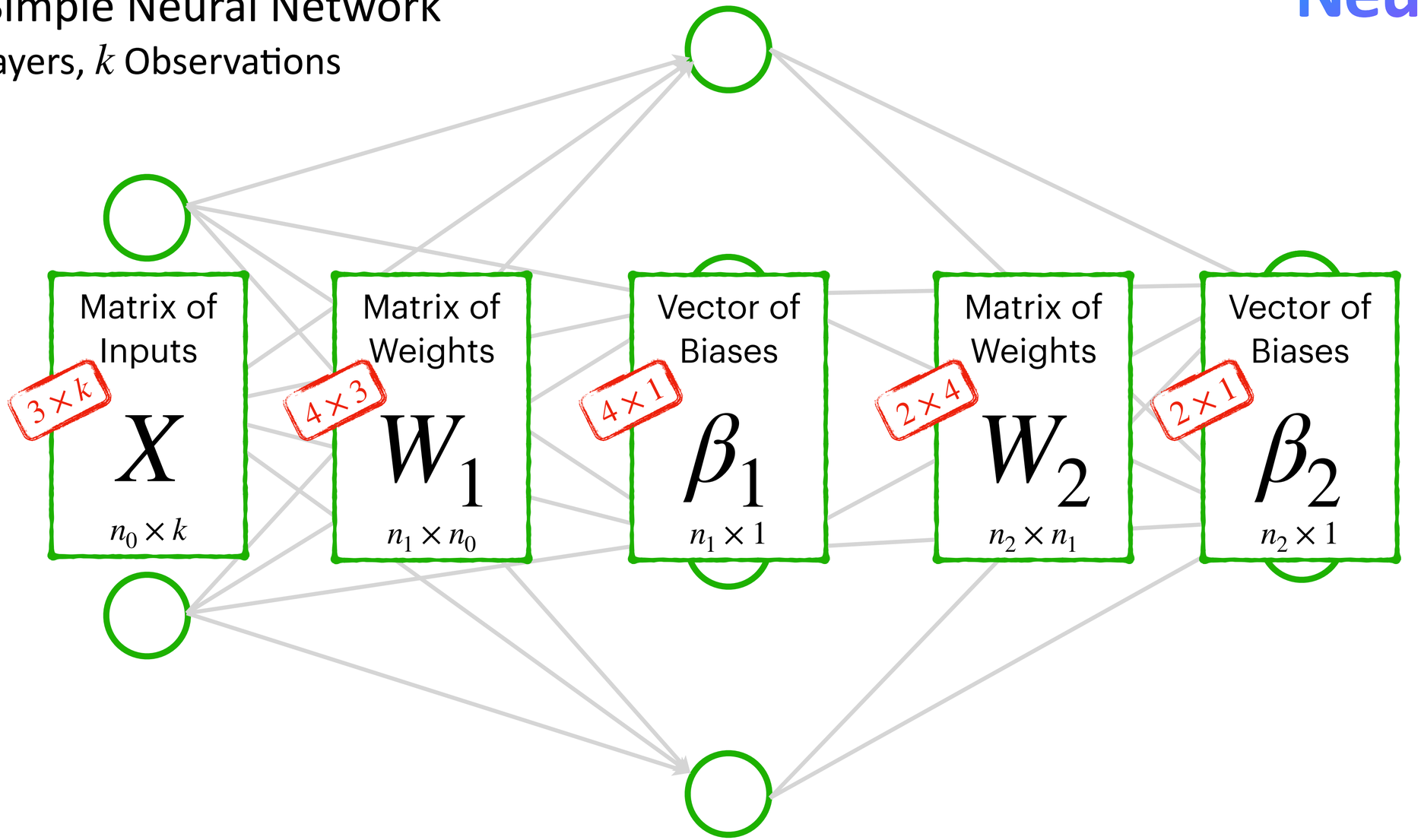
Hidden Layer (L_1)

$n_2 = 2$

Output Layer (L_2)

A Simple Neural Network

3 Layers, k Observations



$n_0 = 3$

Input Layer (L_0)

$n_1 = 4$

Hidden Layer (L_1)

$n_2 = 2$

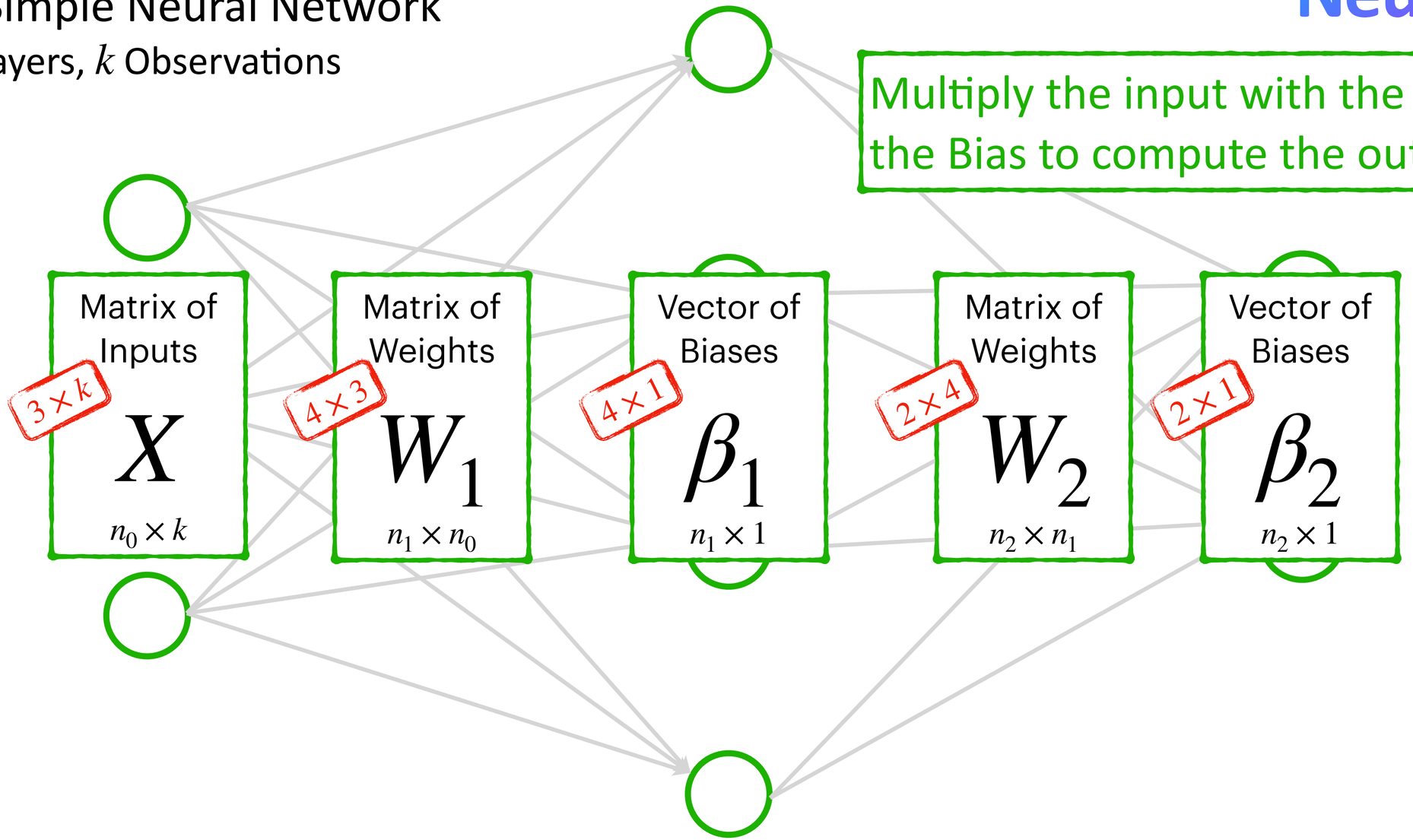
Output Layer (L_2)

Neural Networks

A Simple Neural Network

3 Layers, k Observations

Multiply the input with the Weight and add the Bias to compute the output at each layer



$n_0 = 3$
Input Layer (L_0)

$n_1 = 4$
Hidden Layer (L_1)

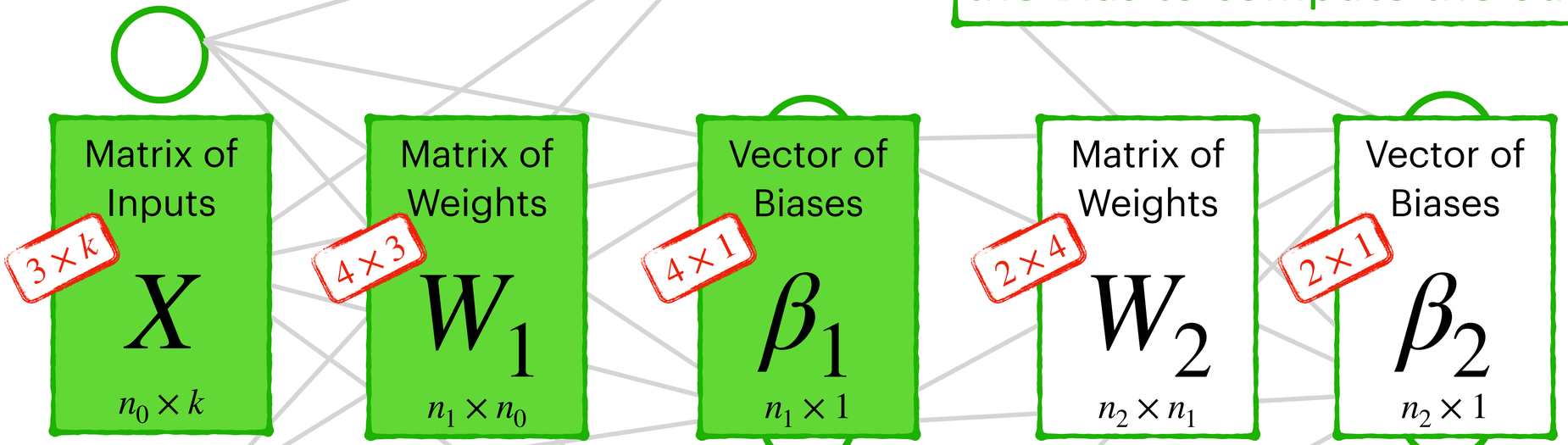
$n_2 = 2$
Output Layer (L_2)

Neural Networks

A Simple Neural Network

3 Layers, k Observations

Multiply the input with the Weight and add the Bias to compute the output at each layer



Output from Layer 1 (L_1)

$$Z_1 = f(W_1 X + \beta_1)$$

$f(g)$ is the activation function in Layer 1 (L_1)

$n_0 = 3$

Input Layer (L_0)

$n_1 = 4$

Hidden Layer (L_1)

$n_2 = 2$

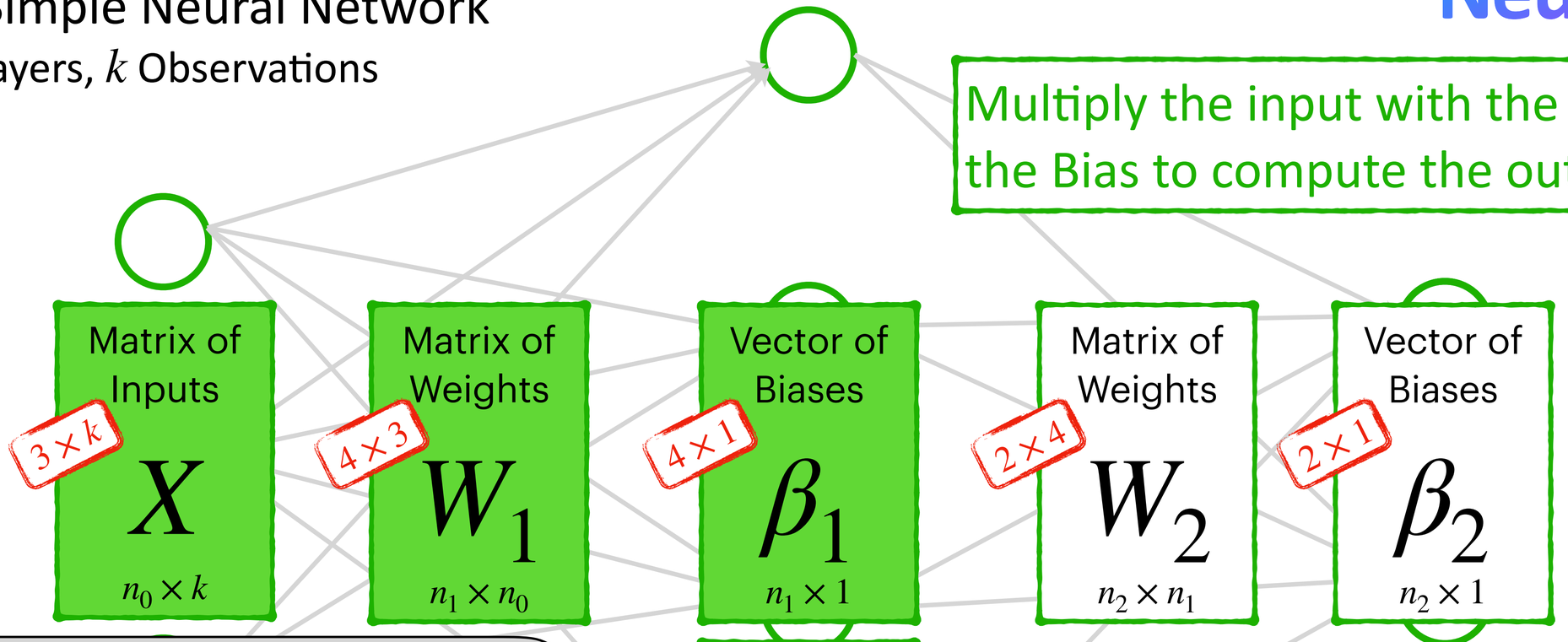
Output Layer (L_2)

Neural Networks

A Simple Neural Network

3 Layers, k Observations

Multiply the input with the Weight and add the Bias to compute the output at each layer



Output from Layer 1 (L_1)

$$Z_1 = f(W_1 X + \beta_1)$$

$f(g)$ is the activation function in Layer 1 (L_1)

$n_0 = 3$

Input Layer (L_0)

$n_1 = 4$

Hidden Layer (L_1)

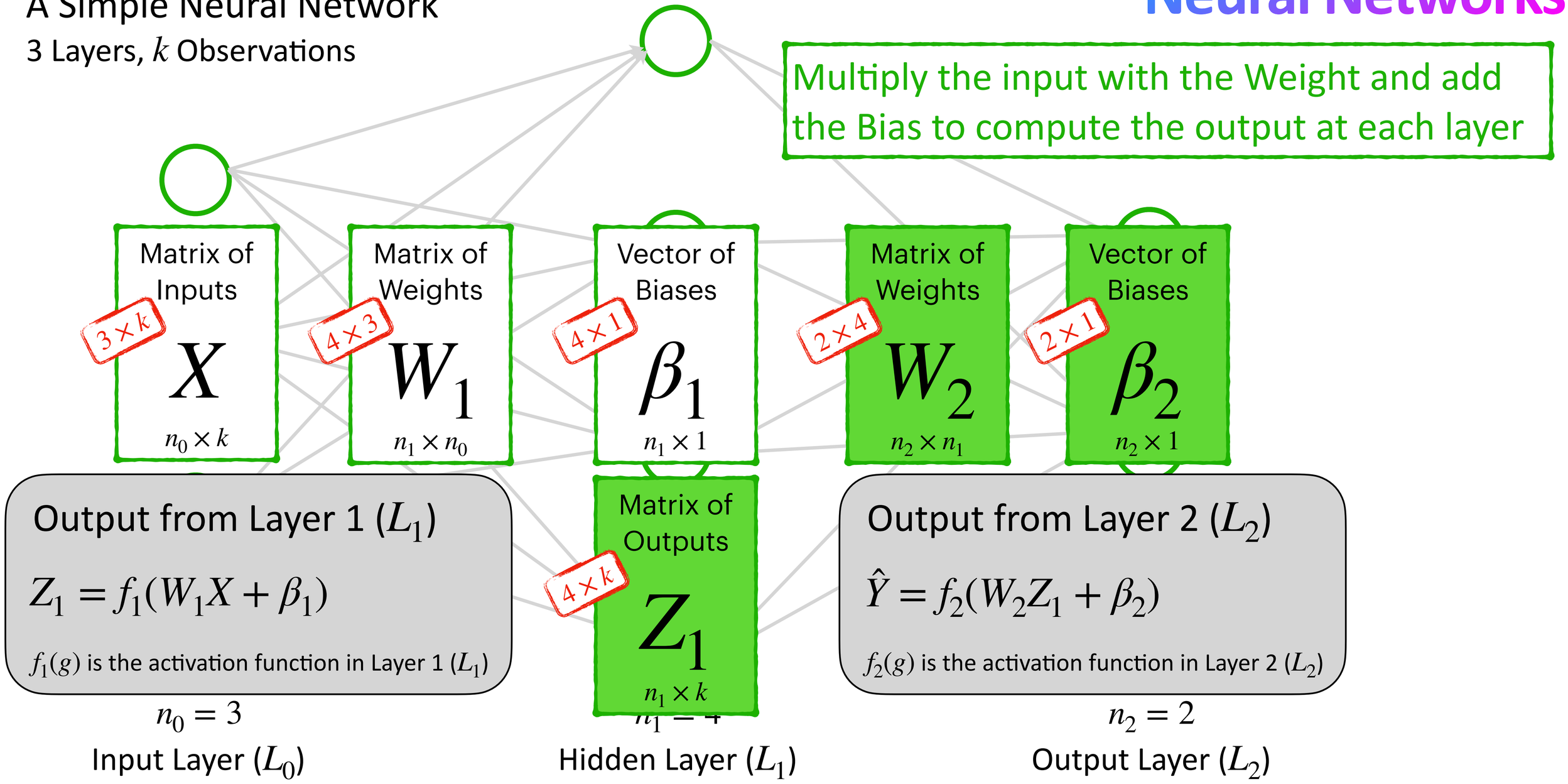
$n_2 = 2$

Output Layer (L_2)

Neural Networks

A Simple Neural Network

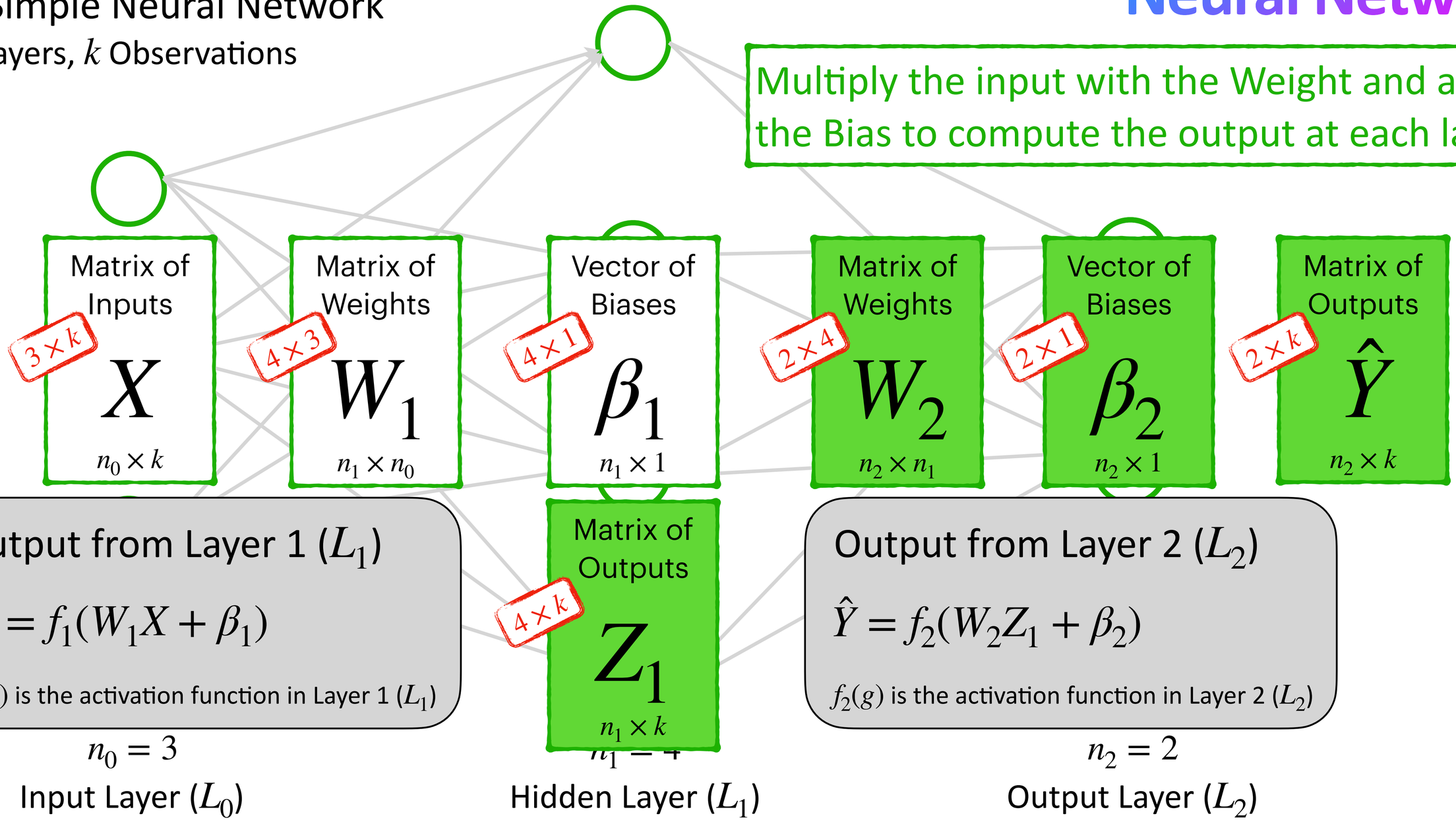
3 Layers, k Observations



Neural Networks

A Simple Neural Network

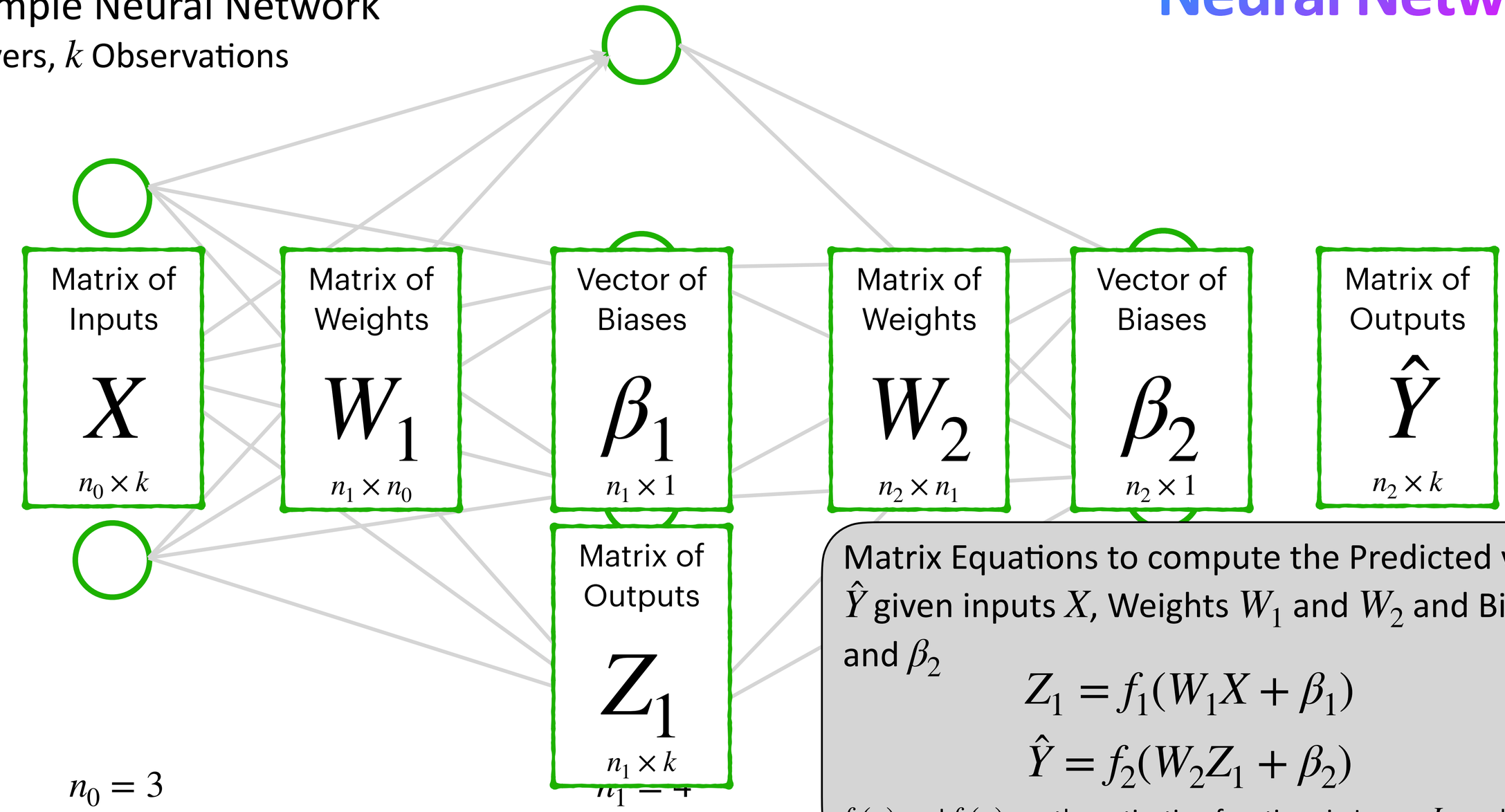
3 Layers, k Observations



Neural Networks

A Simple Neural Network

3 Layers, k Observations



$n_0 = 3$

Input Layer (L_0)

$n_1 = 4$

Hidden Layer (L_1)

Matrix Equations to compute the Predicted values $\hat{Y}$ given inputs X, Weights W_1 and W_2 and Biases β_1 and β_2

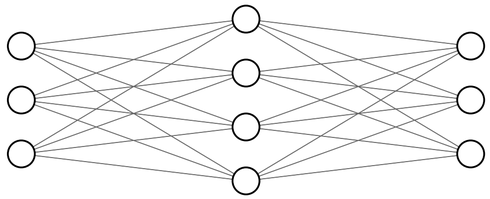
$$Z_1 = f_1(W_1 X + \beta_1)$$

$$\hat{Y} = f_2(W_2 Z_1 + \beta_2)$$

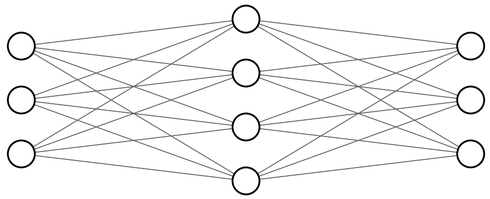
$f_1(g)$ and $f_2(g)$ are the activation functions in Layers L_1 and L_2

Neural Networks

Neural Networks



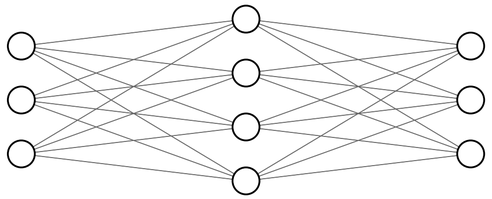
Neural Networks



3 Layers

2 Matrix Equations

Neural Networks



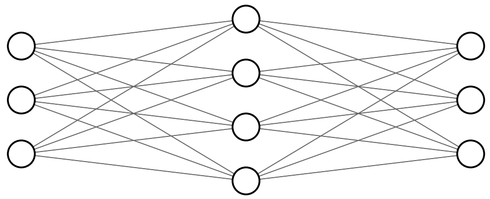
3 Layers

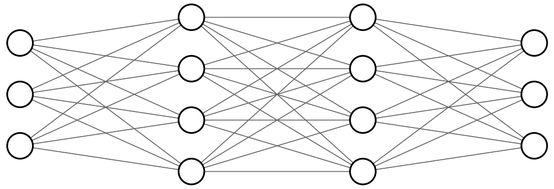
2 Matrix Equations

$$Z_1 = f(W_1 X + \beta_1)$$

$$\hat{Y} = f(W_2 Z_1 + \beta_2)$$

Neural Networks





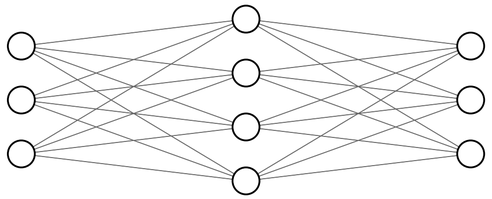
3 Layers

2 Matrix Equations

$$Z_1 = f(W_1 X + \beta_1)$$

$$\hat{Y} = f(W_2 Z_1 + \beta_2)$$

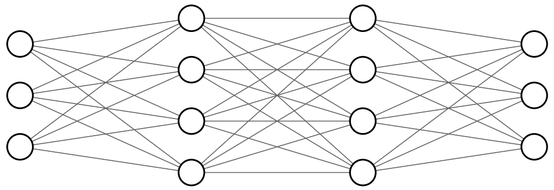
Neural Networks



3 Layers

2 Matrix Equations

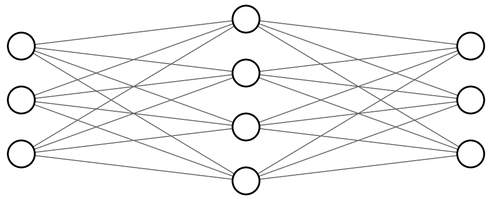
$$Z_1 = f(W_1 X + \beta_1)$$
$$\hat{Y} = f(W_2 Z_1 + \beta_2)$$



4 Layers

3 Matrix Equations

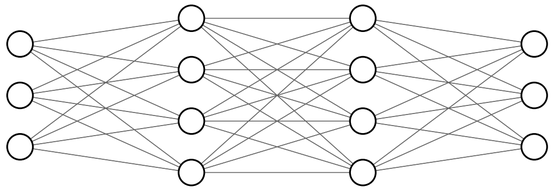
Neural Networks



3 Layers

2 Matrix Equations

$$Z_1 = f(W_1 X + \beta_1)$$
$$\hat{Y} = f(W_2 Z_1 + \beta_2)$$

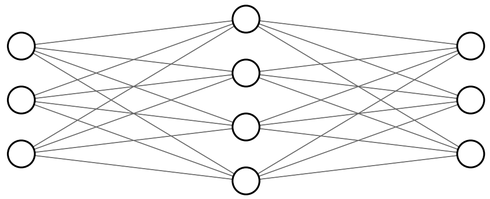


4 Layers

3 Matrix Equations

$$Z_1 = f(W_1 X + \beta_1)$$
$$Z_2 = f(W_2 Z_1 + \beta_2)$$
$$\hat{Y} = f(W_3 Z_2 + \beta_3)$$

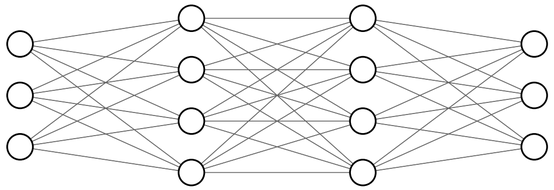
Neural Networks



3 Layers

2 Matrix Equations

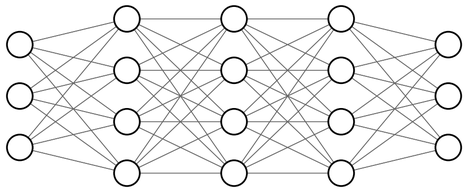
$$Z_1 = f(W_1 X + \beta_1)$$
$$\hat{Y} = f(W_2 Z_1 + \beta_2)$$



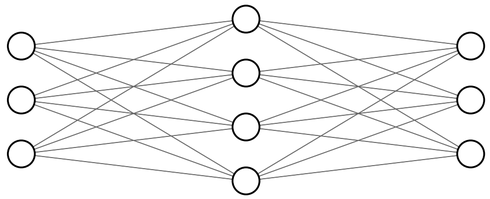
4 Layers

3 Matrix Equations

$$Z_1 = f(W_1 X + \beta_1)$$
$$Z_2 = f(W_2 Z_1 + \beta_2)$$
$$\hat{Y} = f(W_3 Z_2 + \beta_3)$$

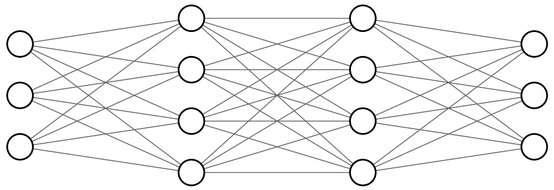


Neural Networks



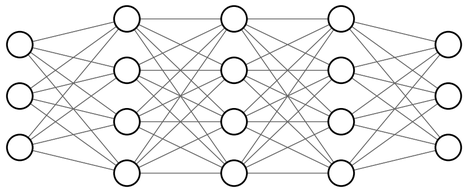
3 Layers
2 Matrix Equations

$$Z_1 = f(W_1 X + \beta_1)$$
$$\hat{Y} = f(W_2 Z_1 + \beta_2)$$



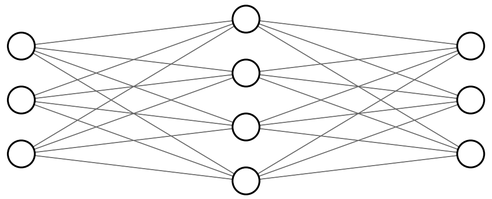
4 Layers
3 Matrix Equations

$$Z_1 = f(W_1 X + \beta_1)$$
$$Z_2 = f(W_2 Z_1 + \beta_2)$$
$$\hat{Y} = f(W_3 Z_2 + \beta_3)$$



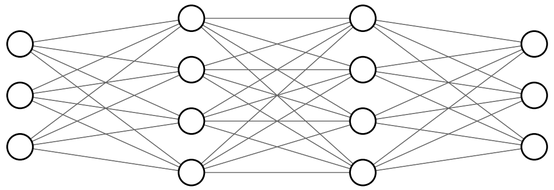
5 Layers
4 Matrix Equations

Neural Networks



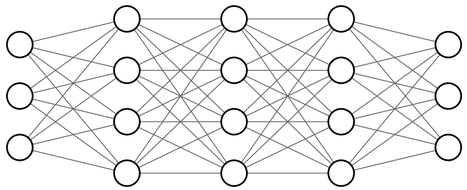
3 Layers
2 Matrix Equations

$$Z_1 = f(W_1 X + \beta_1)$$
$$\hat{Y} = f(W_2 Z_1 + \beta_2)$$



4 Layers
3 Matrix Equations

$$Z_1 = f(W_1 X + \beta_1)$$
$$Z_2 = f(W_2 Z_1 + \beta_2)$$
$$\hat{Y} = f(W_3 Z_2 + \beta_3)$$

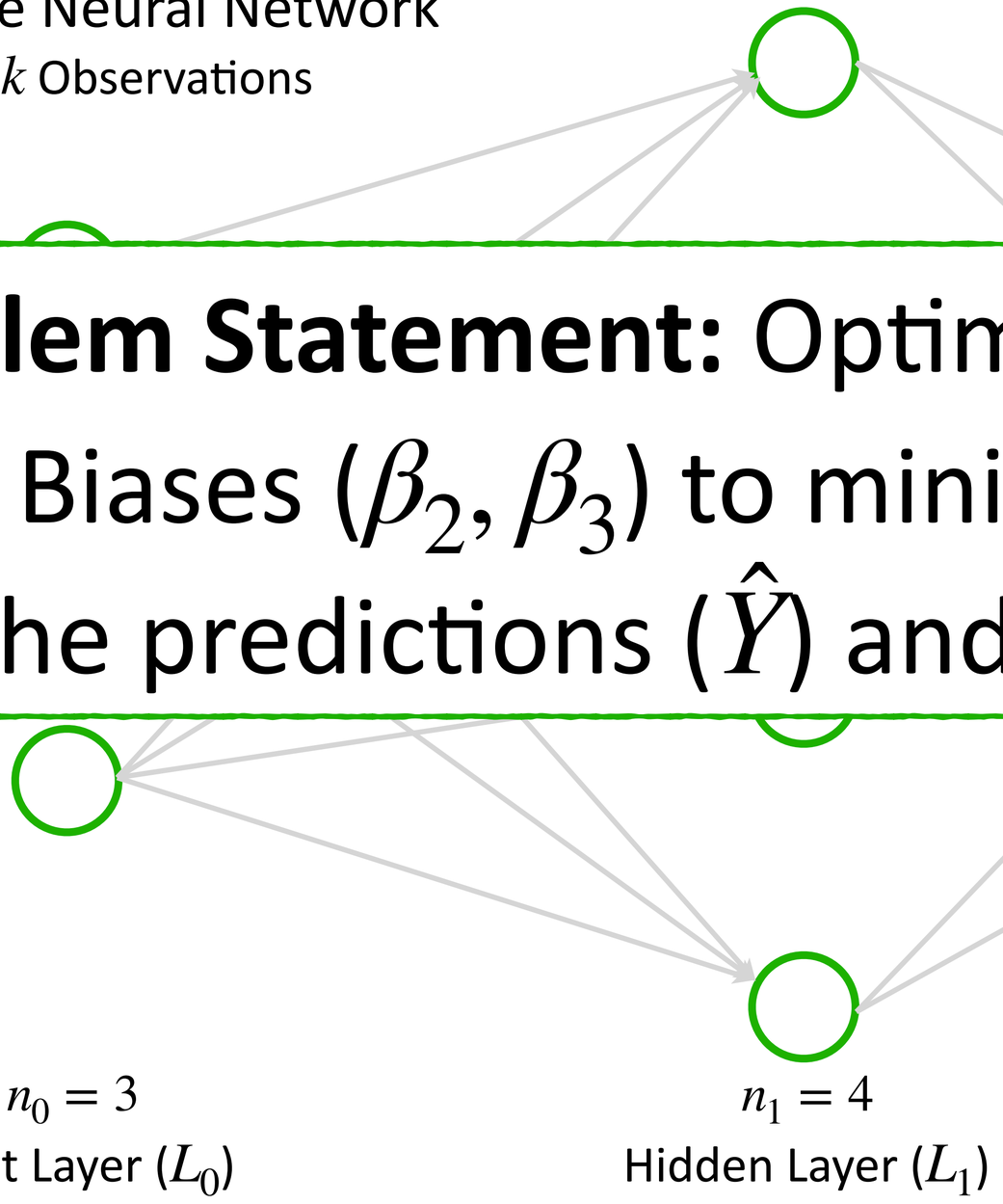


5 Layers
4 Matrix Equations

$$Z_1 = f(W_1 X + \beta_1)$$
$$Z_2 = f(W_2 Z_1 + \beta_2)$$
$$Z_3 = f(W_3 Z_2 + \beta_3)$$
$$\hat{Y} = f(W_4 Z_3 + \beta_4)$$

A Simple Neural Network

3 Layers, k Observations



Problem Statement: Optimize the weights (W_1, W_2) and Biases (β_2, β_3) to minimize the error between the predictions ($\hat{Y}$) and the observations (Y)

$n_0 = 3$

Input Layer (L_0)

$n_1 = 4$

Hidden Layer (L_1)

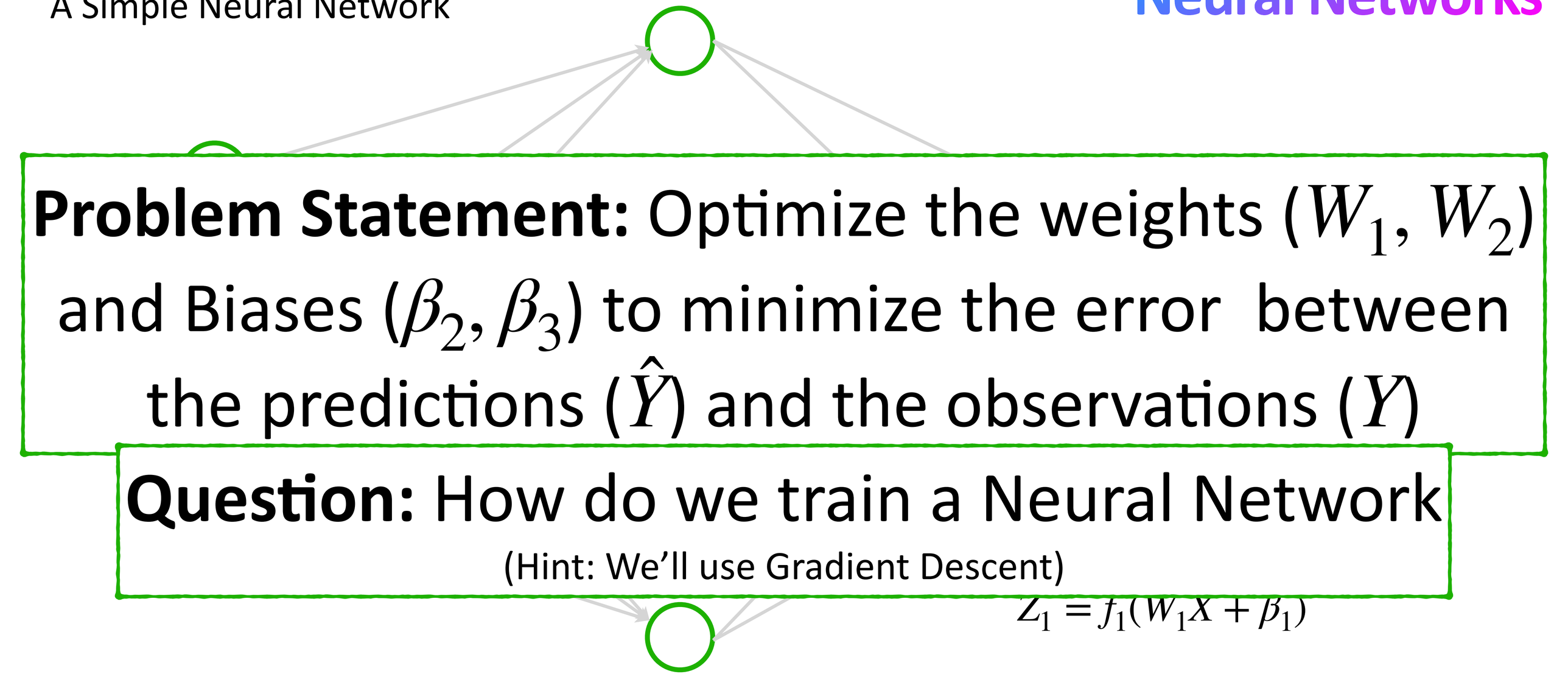
Matrix Equations to compute the Predicted values $\hat{Y}$ given inputs X, Weights W_1 and W_2 and Biases β_1 and β_2

$$Z_1 = f_1(W_1 X + \beta_1)$$

$$\hat{Y} = f_2(W_2 Z_1 + \beta_2)$$

$f_1(g)$ and $f_2(g)$ are the activation functions in Layers L_1 and L_2

A Simple Neural Network



Problem Statement: Optimize the weights (W_1, W_2) and Biases (β_2, β_3) to minimize the error between the predictions ($\hat{Y}$) and the observations (Y)

Question: How do we train a Neural Network

(Hint: We'll use Gradient Descent)

$$Z_1 = f_1(W_1 X + \beta_1)$$

Related Tutorials & Textbooks

Forward and Back Propagation in Neural Networks

A deep dive into how Neural Networks are trained using Gradient Descent. Output predictions, are compared to observations to calculate loss and Backward propagation then computes gradients by working backward through the network

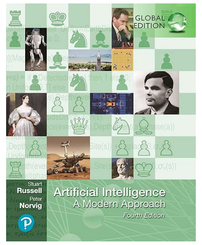
Multiple Regression

Multiple regression extends the two dimensional linear model introduced in Simple Linear Regression to k + 1 dimensions with one dependent variable, k independent variables and k+1 parameters.

Gradient Descent for Multiple Regression

Gradient Descent algorithm for multiple regression and how it can be used to optimize k + 1 parameters for a Linear model in multiple dimensions.

Recommended Textbooks



Artificial Intelligence: A Modern Approach
by Peter Norvig, Stuart Russell

For a complete list of tutorials see:
https://arrsingh.com/ai-tutorials