Statistical methods for big data in life sciences and health with R

Philippe Jacquet

Data Scientist at CADMOS

June 2018

CADMOS

- CADMOS: Center for Advanced Modeling Of Science

CADMOS

- CADMOS: Center for Advanced Modeling Of Science
- Free service for all UNIL, EPFL and UNIGE researchers

CADMOS

- CADMOS: Center for Advanced Modeling Of Science
- Free service for all UNIL, EPFL and UNIGE researchers
- Help with HPC parallel computing (code optimization) or Big Data analysis (pipeline development), including some computing power at SCITAS (the EPFL cluster)

CADMOS

- CADMOS: Center for Advanced Modeling Of Science
- Free service for all UNIL, EPFL and UNIGE researchers
- Help with HPC parallel computing (code optimization) or Big Data analysis (pipeline development), including some computing power at SCITAS (the EPFL cluster)
- Contact:
  UNIL: philippe.jacquet@unil.ch, etienne.orliac@unil.ch
  EPFL: vincent.keller@epfl.ch
  UNIGE: jean-luc.falcone@unige.ch

Outline

TODAY

13:30 – 15:00	Lecture on Neural Network
15:00 – 15:30	Coffee break
15:30 – 17:00	Practicals on Neural Network

Outline

TODAY

13:30 – 15:00	Lecture on Neural Network
15:00 – 15:30	Coffee break
15:30 – 17:00	Practicals on Neural Network

TOMORROW (Room 321 - Amphipôle)

09:00 – 10:30	Lecture on Decision Tree and Random Forest
10:30 – 11:00	Coffee break
11:00 – 12:30	Practicals on Decision Tree and Random Forest

Outline

If you have any question during the lecture

you are welcome to ask

Outline

We will analyse the same dataset (iris data)

with all the methods

Outline

And present some applications

in health science

The iris dataset

Iris Versicolor **Iris Setosa** **Iris Virginica**

The iris dataset

4 measurements on 50 flowers in 3 species

The iris dataset

Sepal length	Sepal width	Petal length	Petal width	Species
5.1	3.5	1.4	0.2	setosa
7.0	3.2	4.7	1.4	versicolor
6.3	3.3	6.0	2.5	virginica

The iris dataset

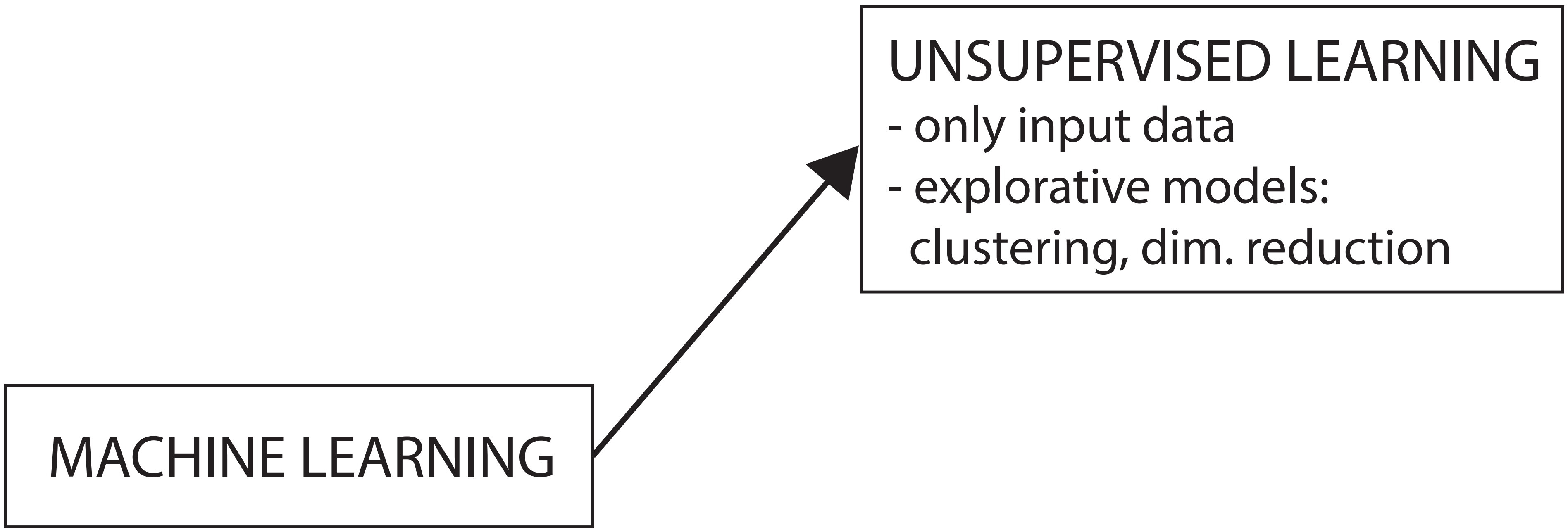
input				output
Sepal length	Sepal width	Petal length	Petal width	Species
5.1	3.5	1.4	0.2	setosa
7.0	3.2	4.7	1.4	versicolor
6.3	3.3	6.0	2.5	virginica

GOAL

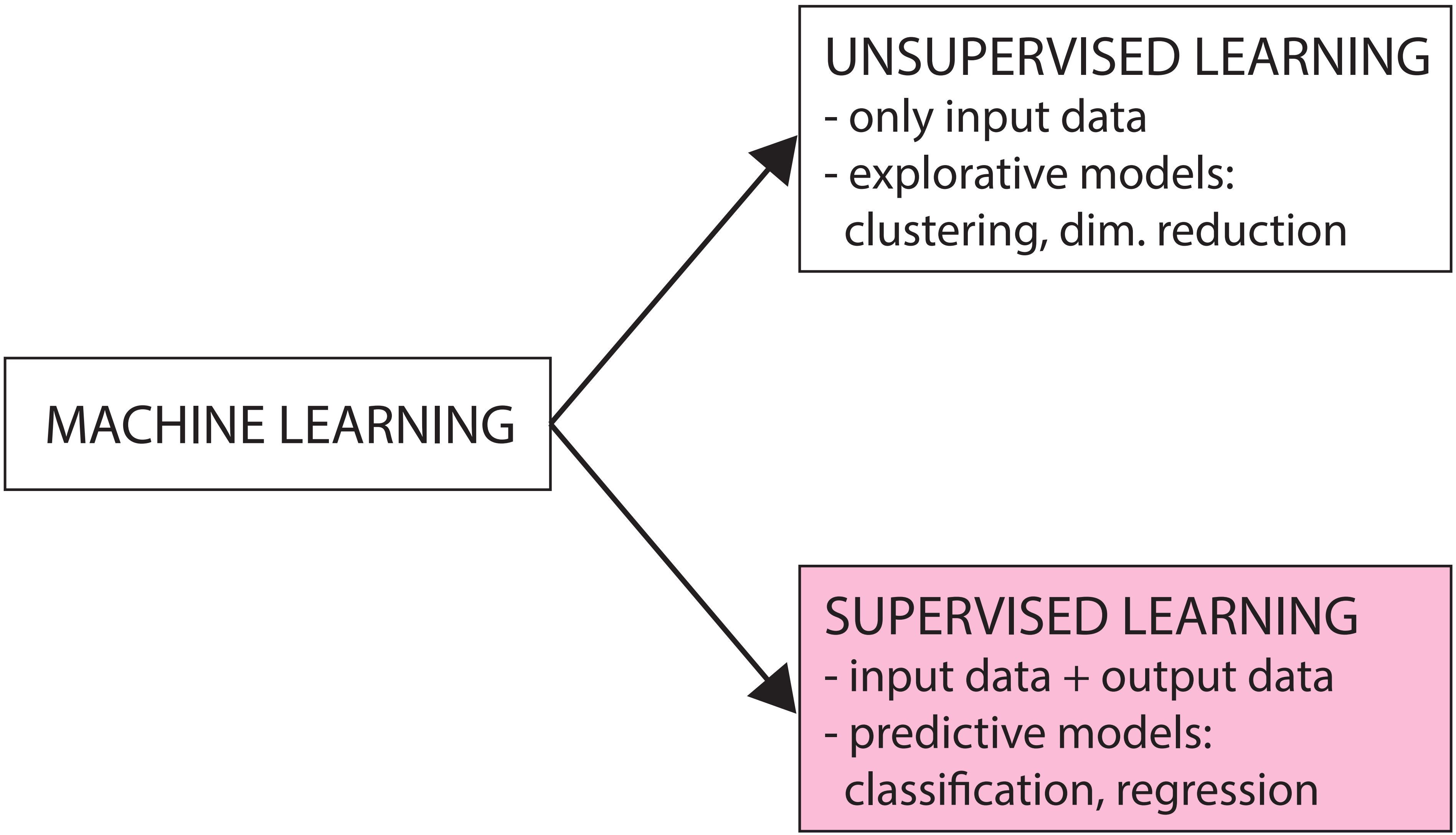
Given an input (4 measurements)

predict its output (species)

Machine learning

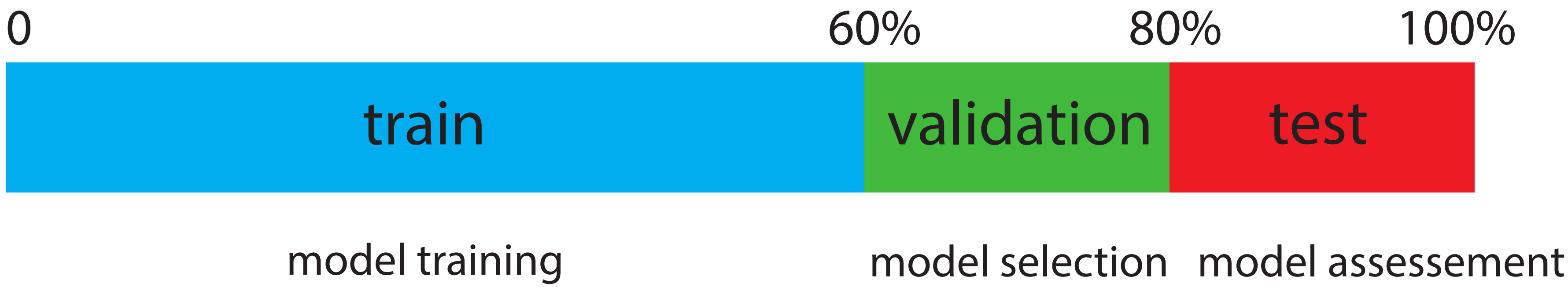


Machine learning



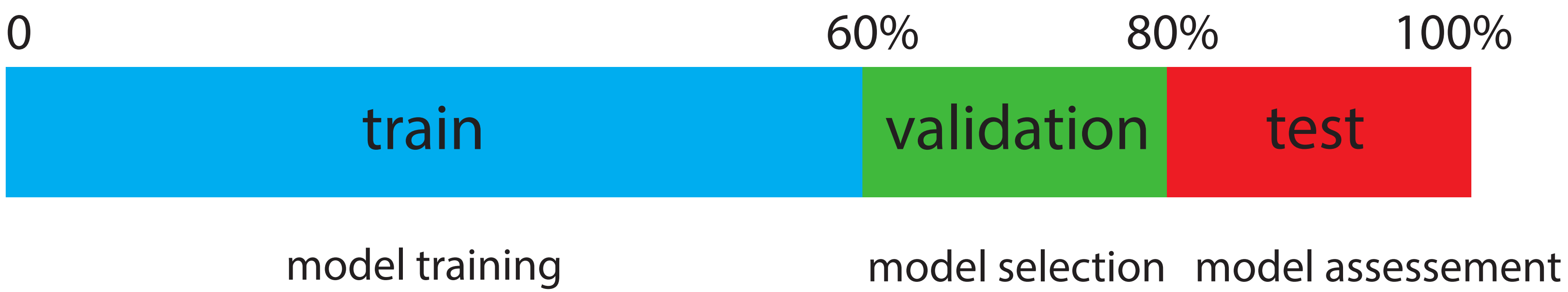
Supervised learning

Data splitting:



Supervised learning

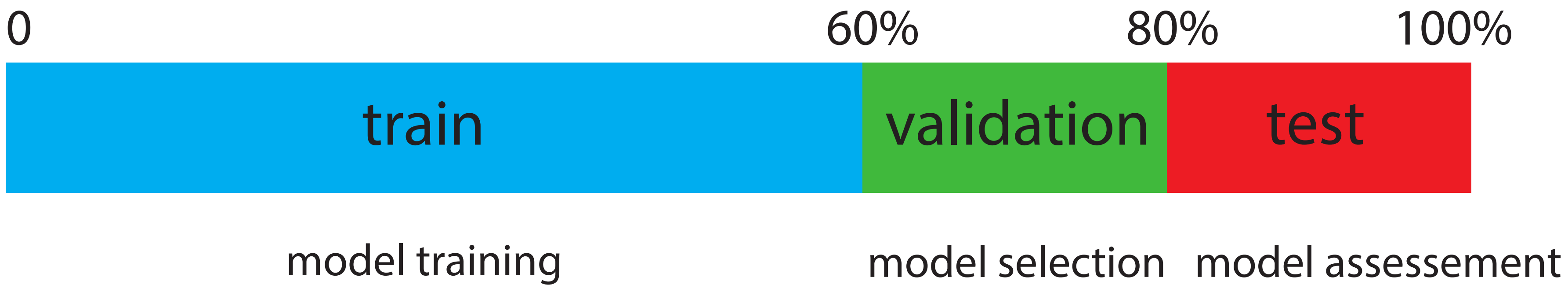
Data splitting:



- Model training: estimate the model parameters

Supervised learning

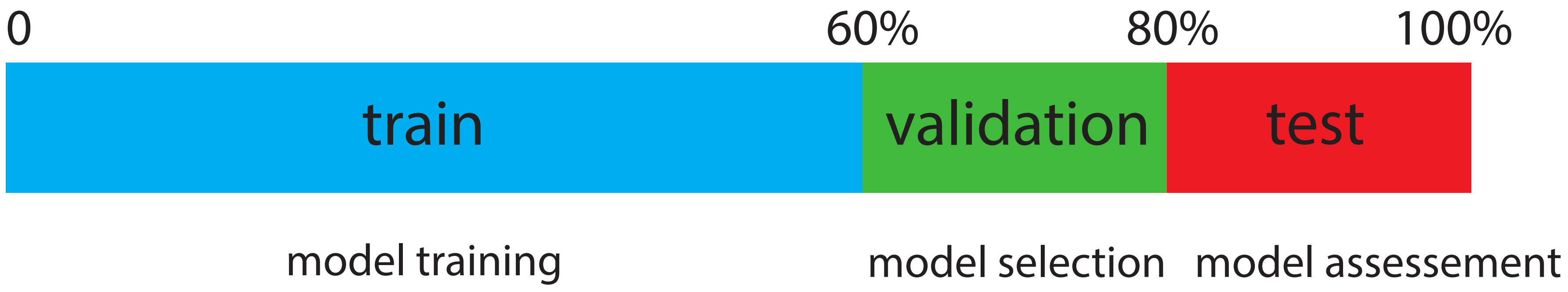
Data splitting:



- Model training: estimate the model parameters
- Model selection: estimate the performance of different models in order to choose the best one

Supervised learning

Data splitting:



- Model training: estimate the model parameters
- Model selection: estimate the performance of different models in order to choose the best one
- Model assessment: having chosen a final model, estimate its prediction accuracy on new data

EXAMPLE

Model 1= Neural Network

Supervised learning

Model 1= Neural Network

- Estimate the model parameters using the training set

Model 1= Neural Network

- Estimate the model parameters using the training set
- Measure the model accuracy based on the validation set

Model 2 = Decision Tree

Model 2 = Decision Tree

- Estimate the model parameters using the training set
- Measure the model accuracy based on the validation set

Model 3 = Random Forest

Model 3 = Random Forest

- Estimate the model parameters using the training set
- Measure the model accuracy based on the validation set

Model Selection

Model Selection

Select the best model based on its accuracy

Model Selection

Select the best model based on its accuracy

Neural Network = Decision Tree

Model Assessment

Model Assessment

Mesure best model accuracy on the test set

Model Assessment

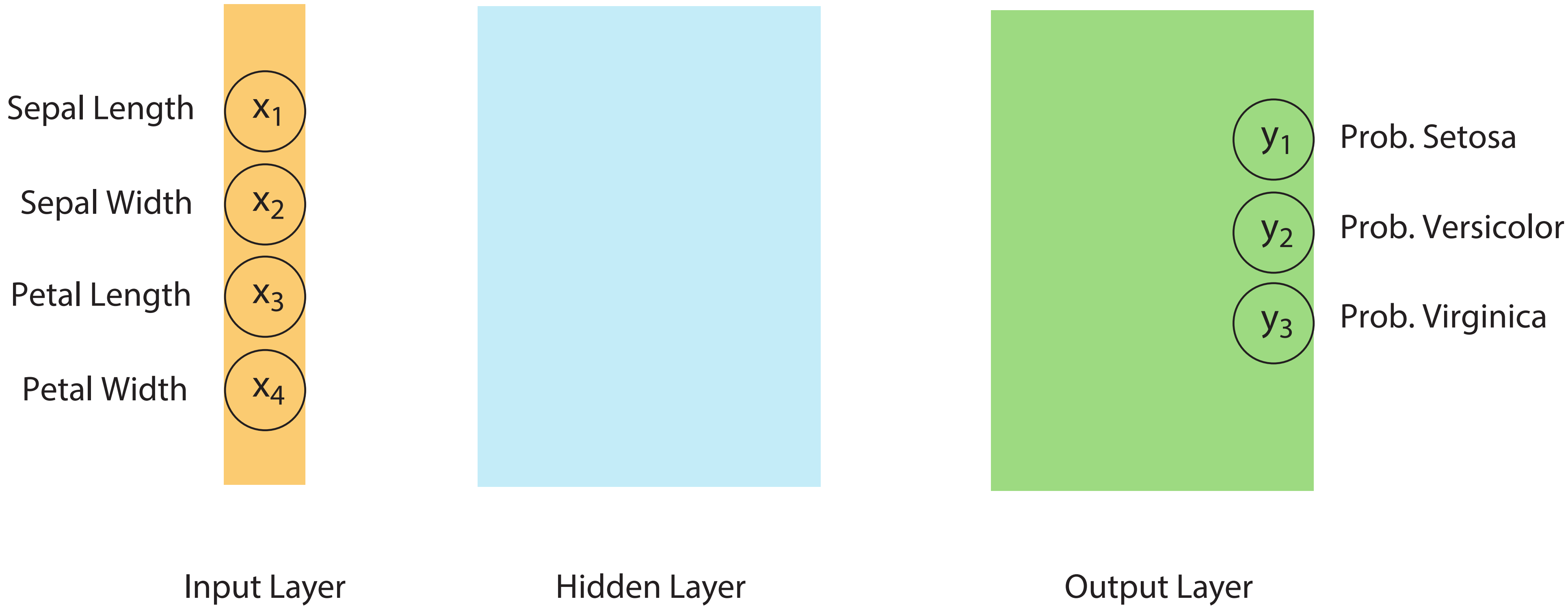
Mesure best model accuracy on the test set

THE END

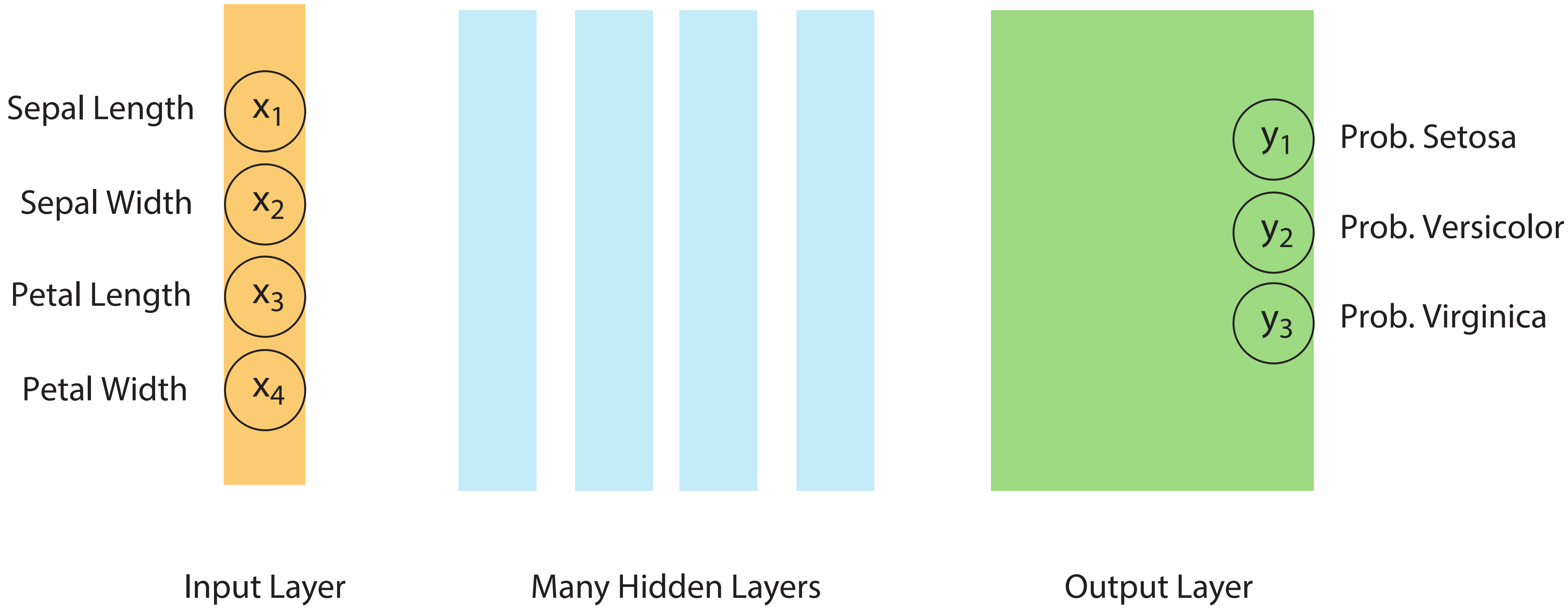
How to build a neural network

for the iris data ?

Neural network for the iris data

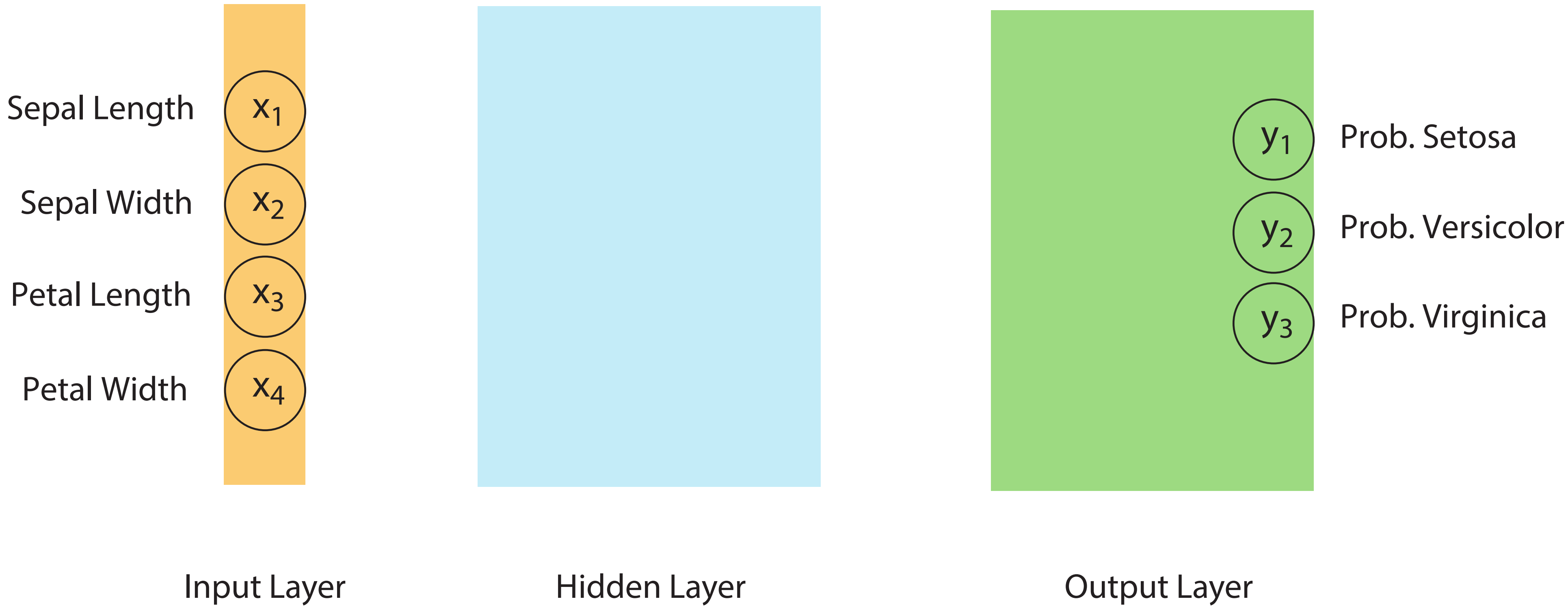


Neural network for the iris data

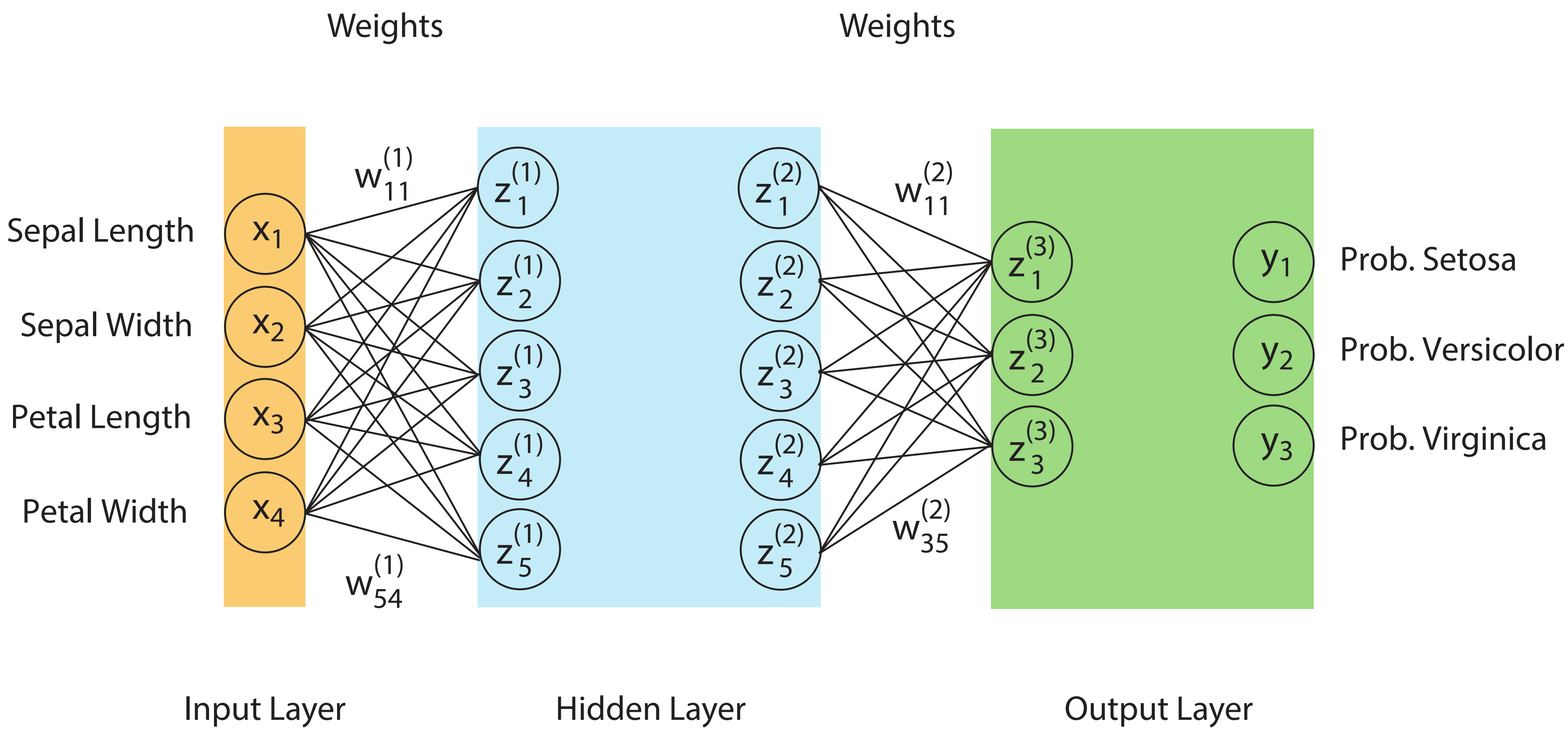


Deep Learning = More than 1 Hidden Layer

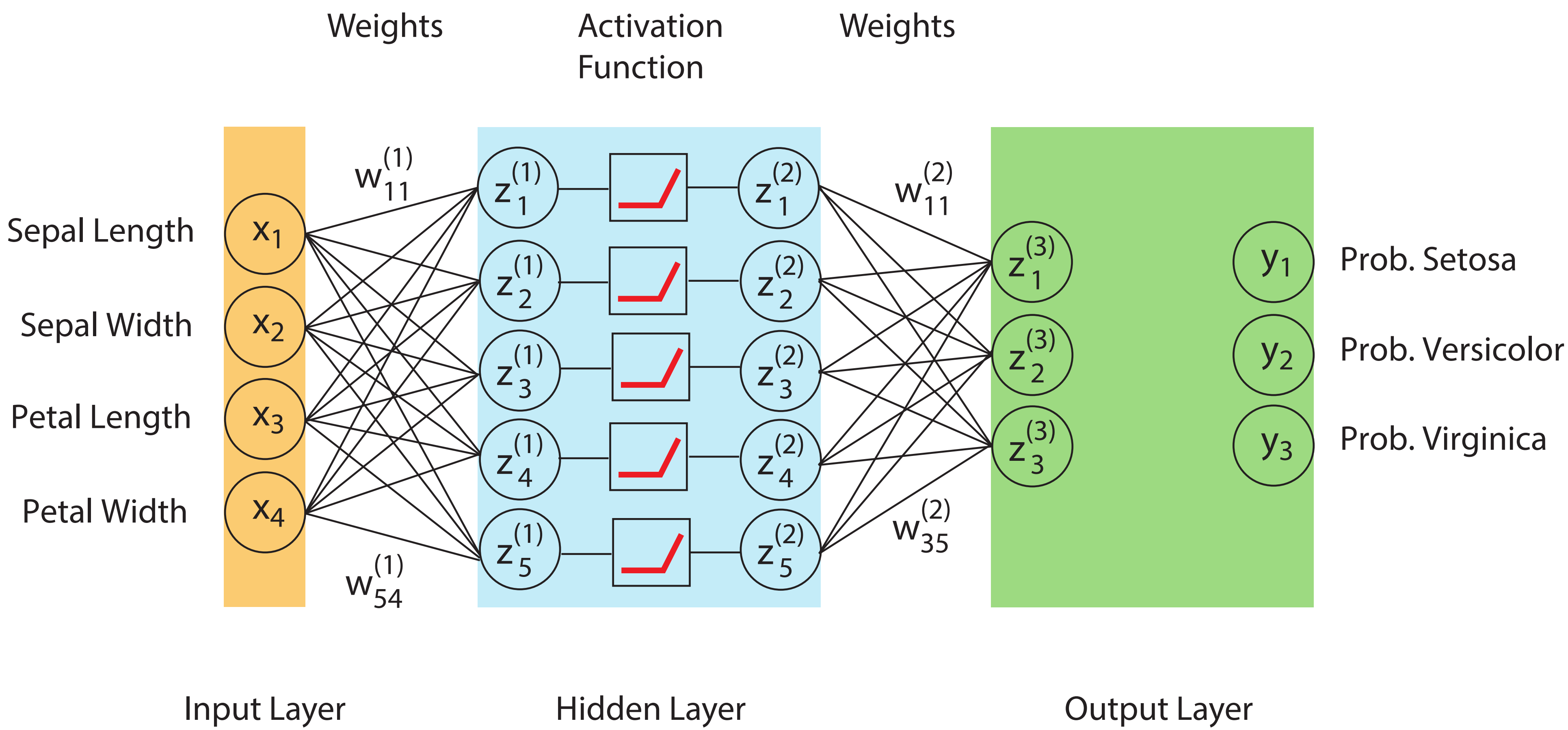
Neural network for the iris data



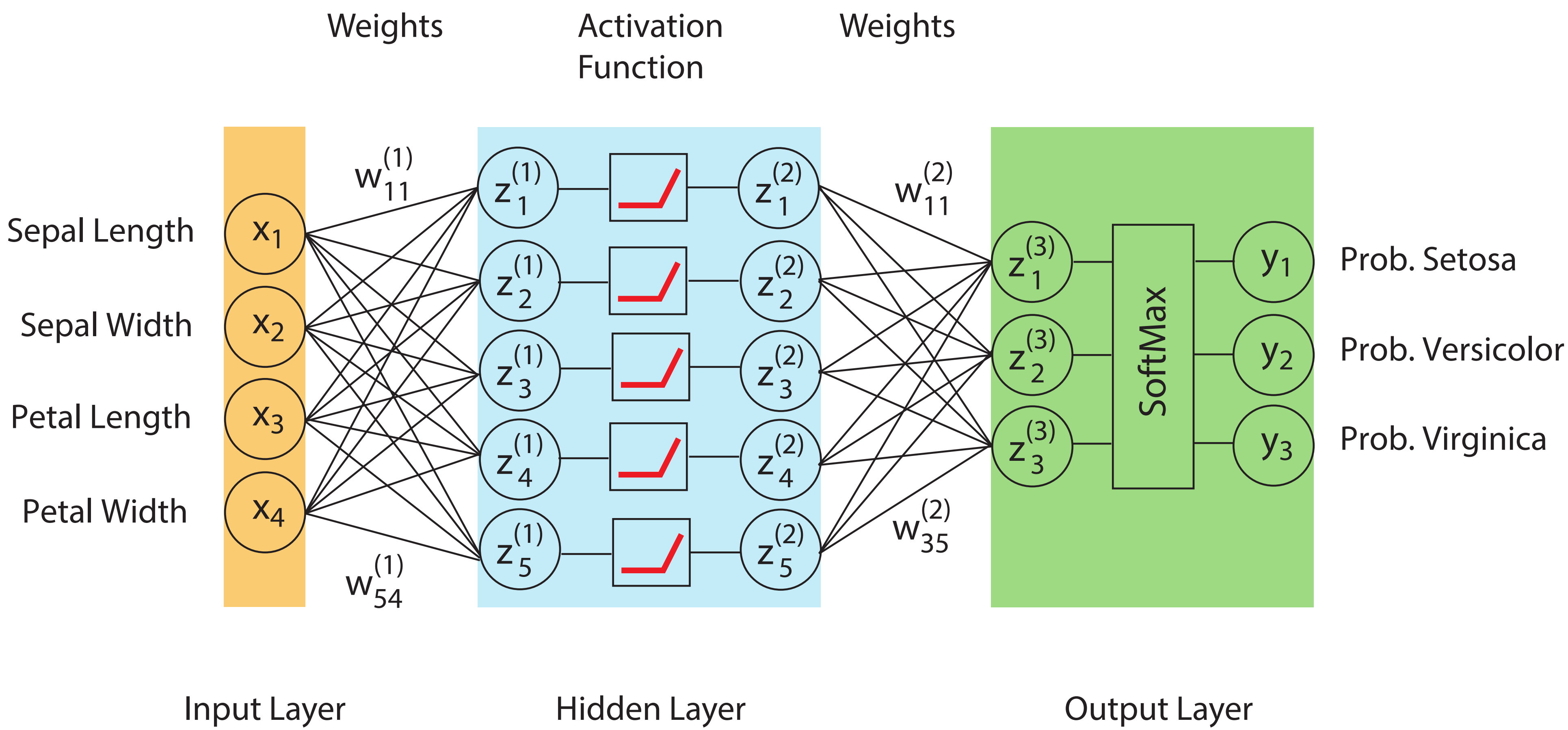
Neural network for the iris data



Neural network for the iris data



Neural network for the iris data



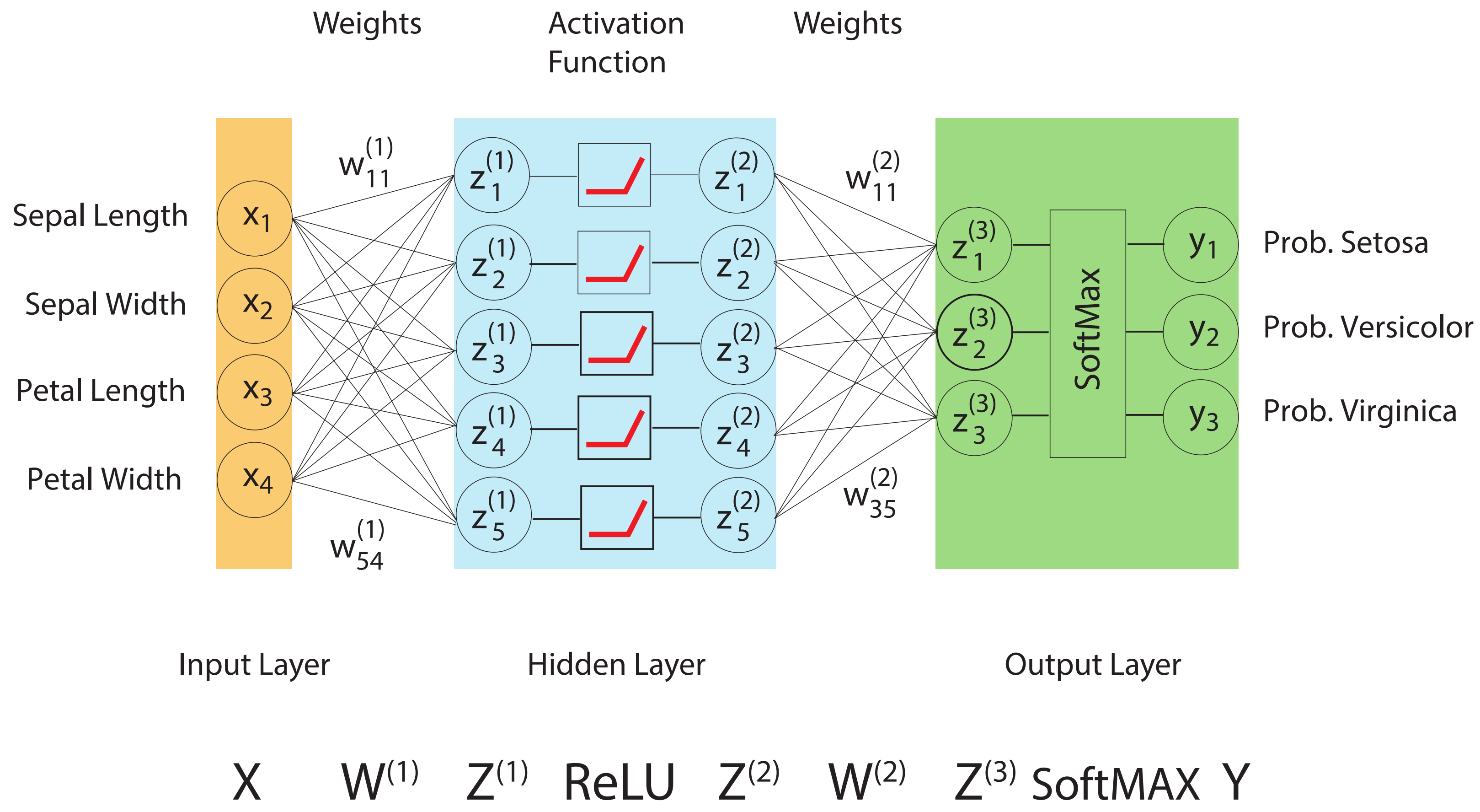
Neural network for the iris data

What is

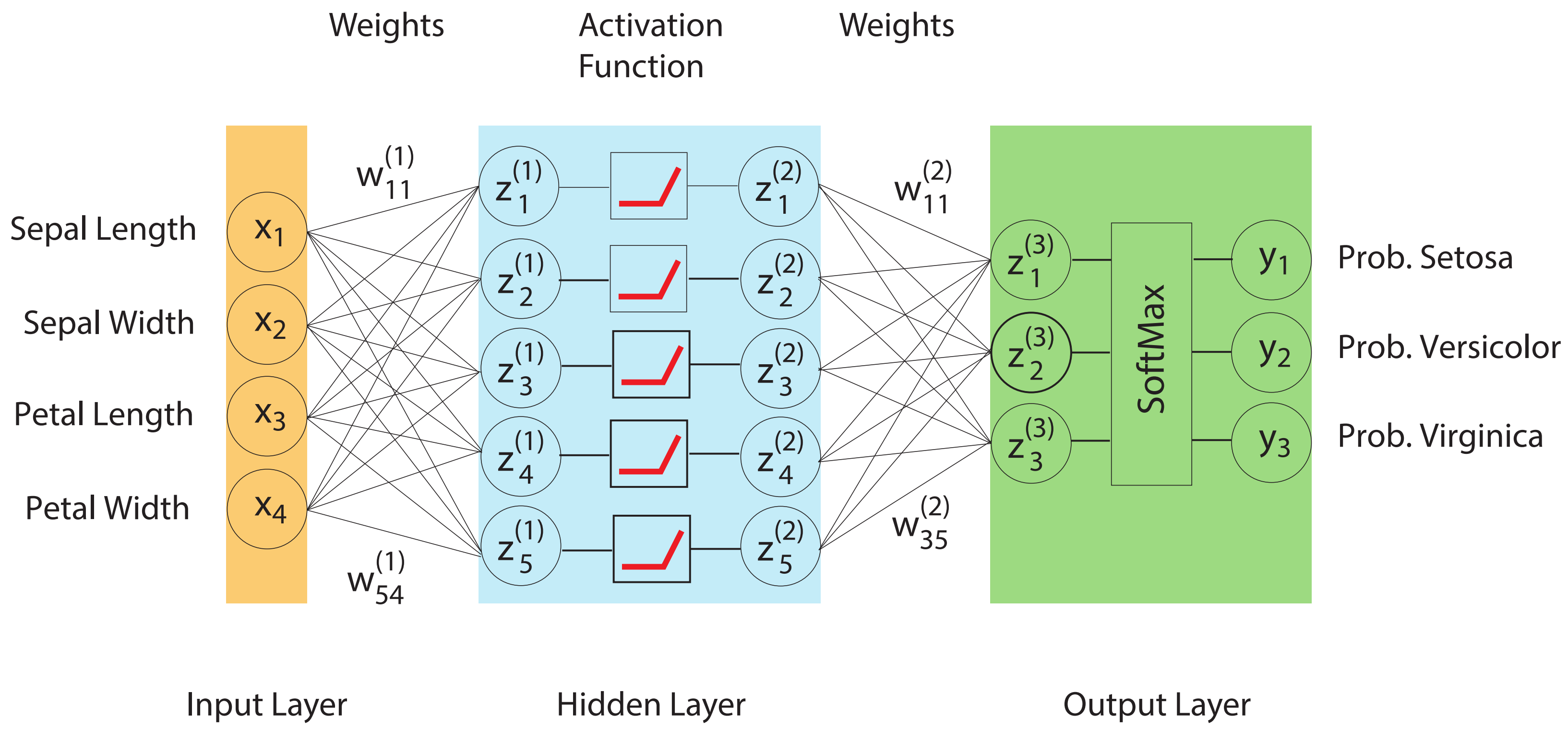
the mathematical formula

for this neural network ?

Neural network for the iris data



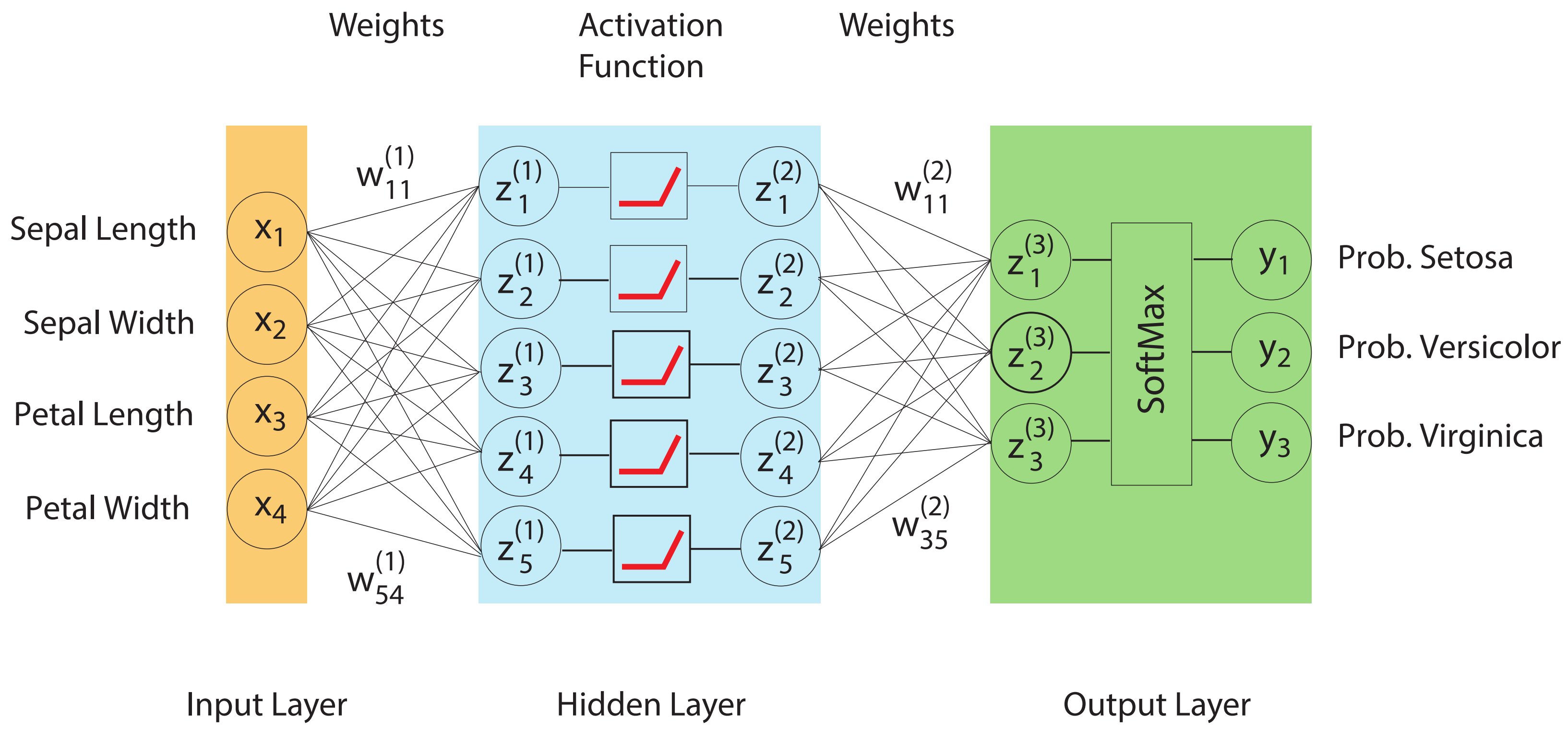
Neural network for the iris data



$W^{(1)}$ $Z^{(1)}$ ReLU $Z^{(2)}$ $W^{(2)}$ $Z^{(3)}$ SoftMAX Y

X

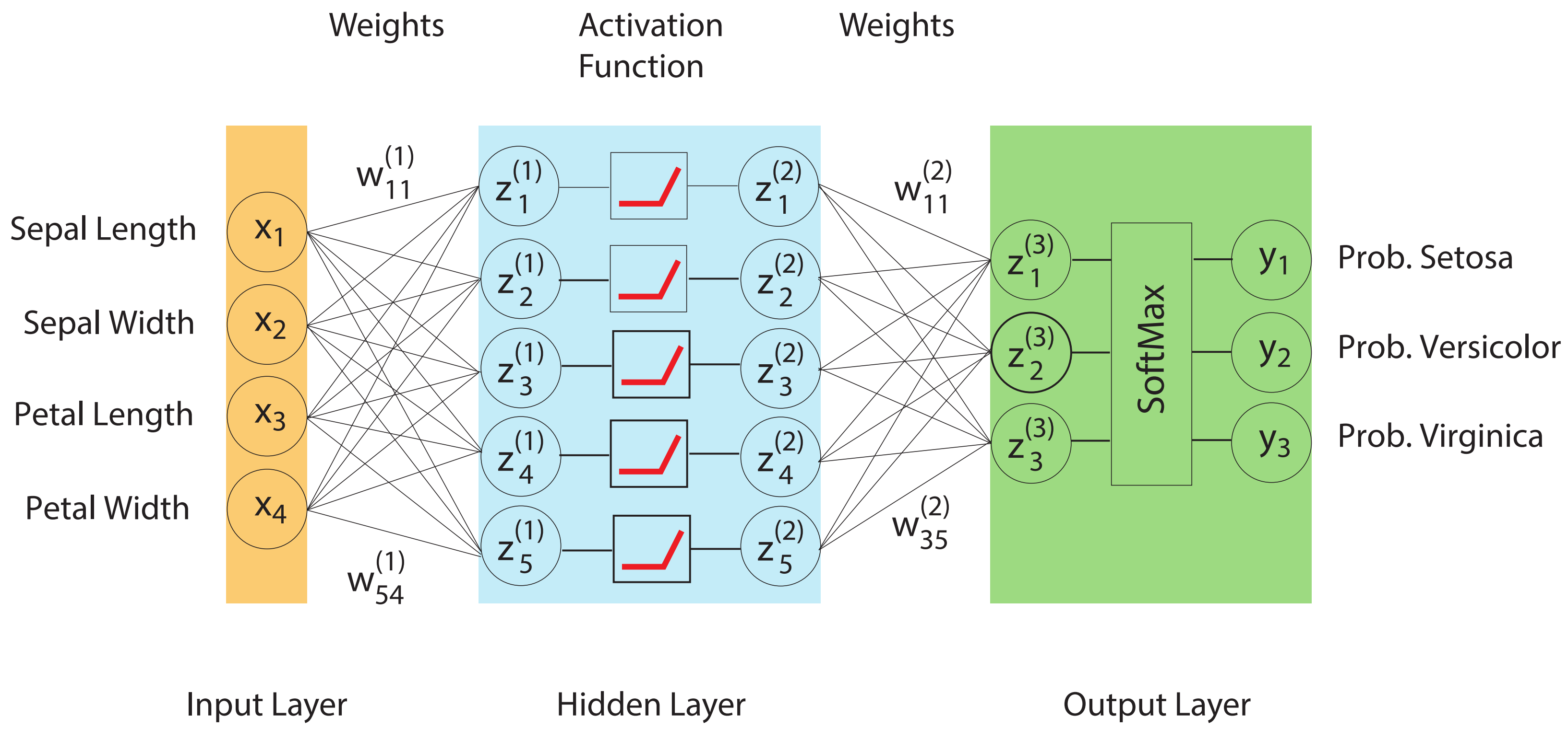
Neural network for the iris data



$Z^{(1)}$ ReLU $Z^{(2)}$ $W^{(2)}$ $Z^{(3)}$ SoftMAX Y

$$W^{(1)}X$$

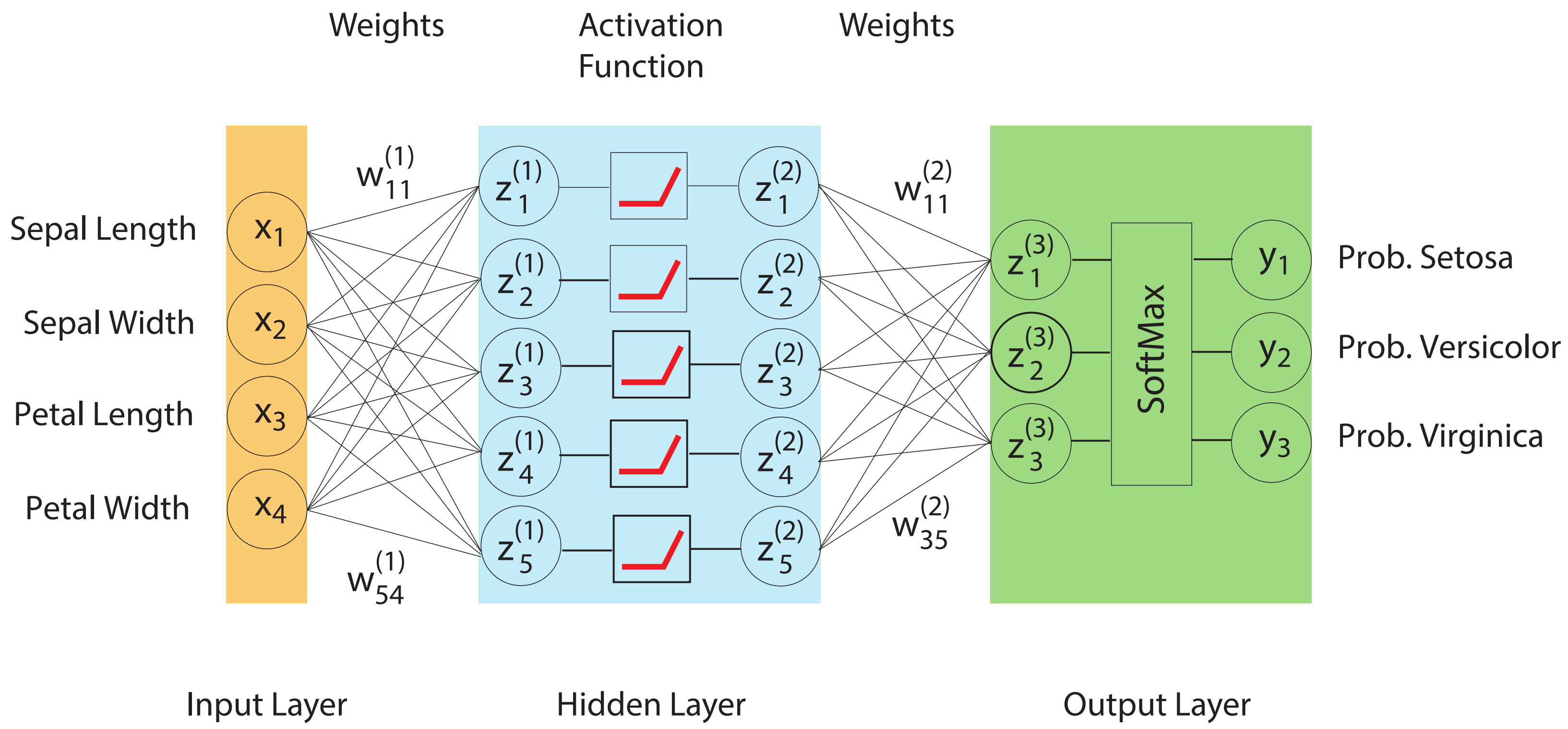
Neural network for the iris data



ReLU $Z^{(2)}$ $W^{(2)}$ $Z^{(3)}$ SoftMAX Y

$$\overbrace{W^{(1)}X}^{Z^{(1)}}$$

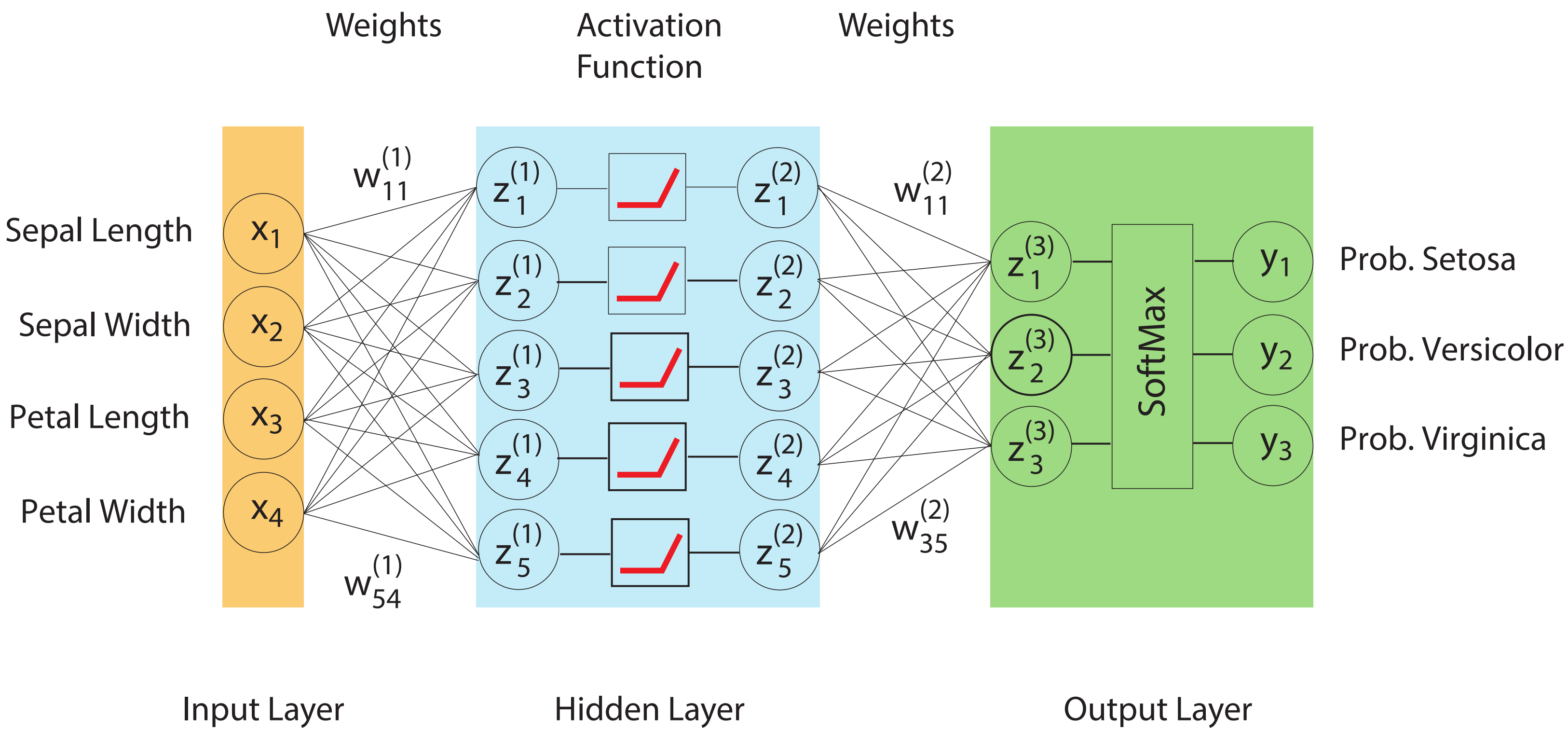
Neural network for the iris data



$Z^{(2)}$ $W^{(2)}$ $Z^{(3)}$ SoftMAX Y

$$\text{ReLU}(\overbrace{W^{(1)}X}^{Z^{(1)}})$$

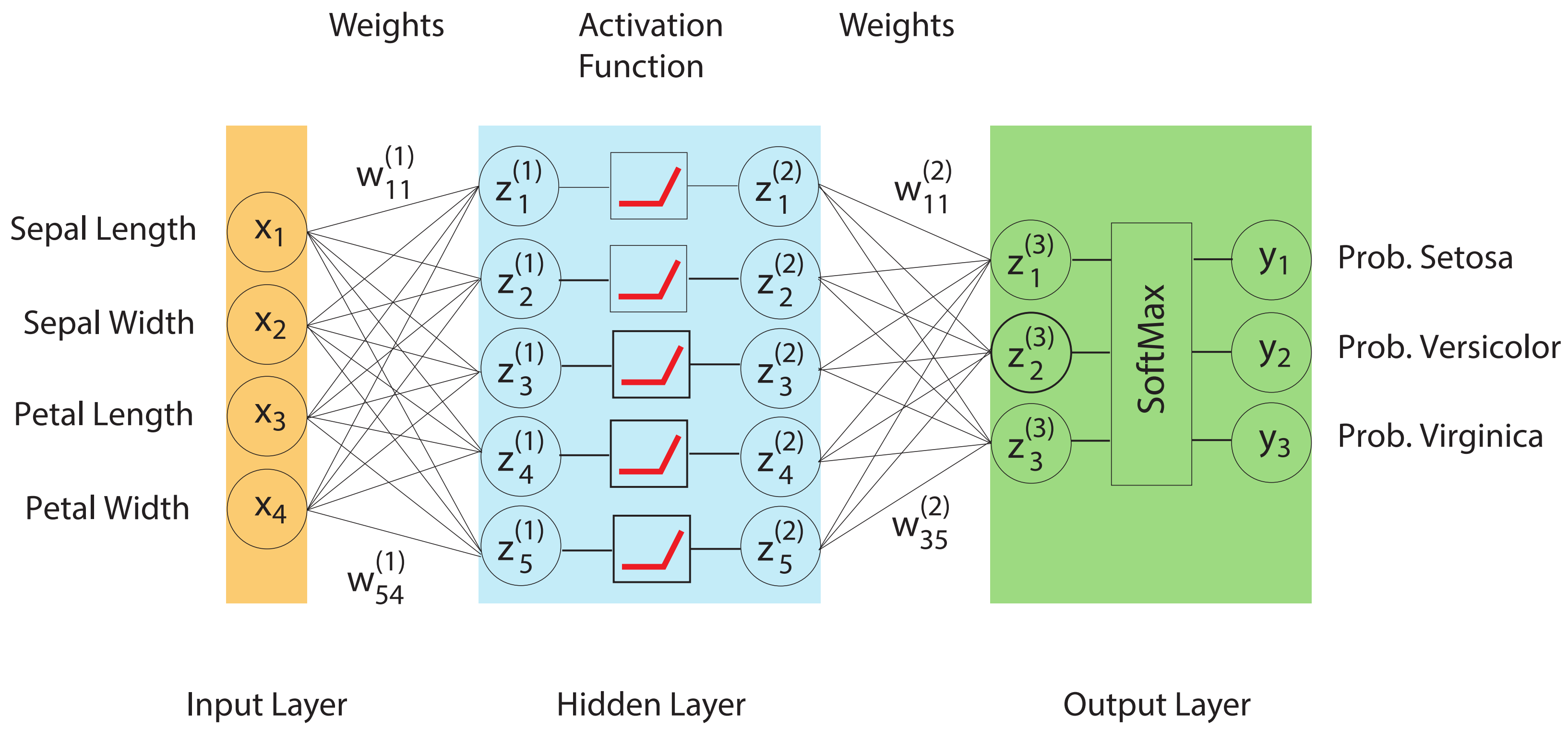
Neural network for the iris data



$W^{(2)}$ $Z^{(3)}$ SoftMAX Y

$$\underbrace{\mathrm{ReLU}(\overbrace{W^{(1)}X}^{Z^{(1)}})}_{Z^{(2)}}$$

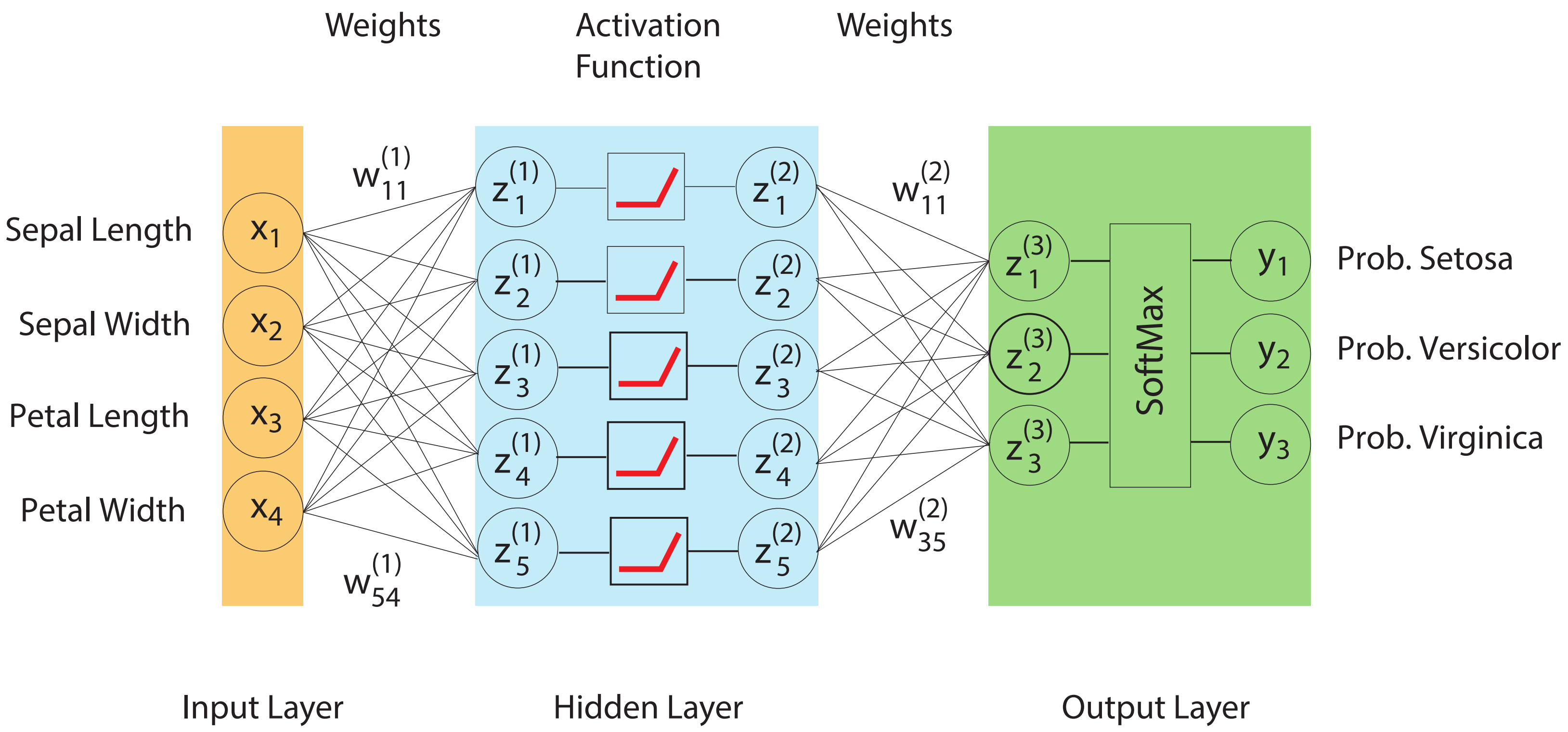
Neural network for the iris data



$Z^{(3)}$ SoftMAX Y

$$W^{(2)} \underbrace{\text{ReLU}(\overbrace{W^{(1)}X}^{Z^{(1)}})}_{Z^{(2)}}$$

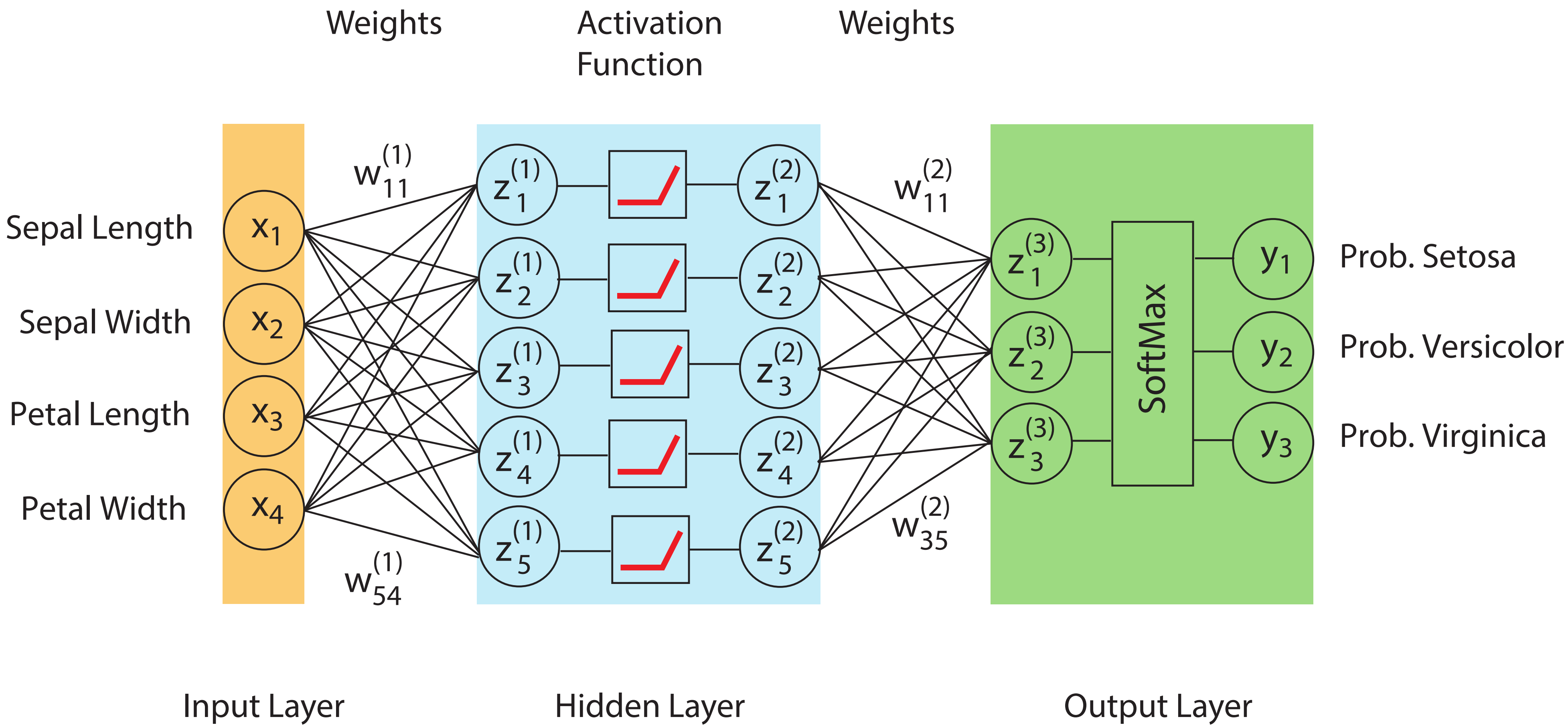
Neural network for the iris data



$$Y = \text{SoftMax}(W^{(2)} \underbrace{\text{ReLU}(\overbrace{W^{(1)}X}^{Z^{(1)}})}_{Z^{(2)}})$$

In summary

Neural network for the iris data



$$Y = f(X) = \text{SoftMax}(W^{(2)}\text{ReLU}(W^{(1)}X))$$

Forward propagation

In forward propagation, we go from X to Y by applying a series of functional transformations :

$$Y = \text{SoftMax}(W^{(2)}\text{ReLU}(W^{(1)}X))$$

Forward propagation

In forward propagation, we go from X to Y by applying a series of functional transformations :

$$Y = \text{SoftMax}(W^{(2)}\text{ReLU}(W^{(1)}X))$$

Questions:

- How do we determine the weights $W^{(1)}$ and $W^{(2)}$?

Forward propagation

In forward propagation, we go from X to Y by applying a series of functional transformations :

$$Y = \text{SoftMax}(W^{(2)}\text{ReLU}(W^{(1)}X))$$

Questions:

- How do we determine the weights $W^{(1)}$ and $W^{(2)}$?
- Why do we need the activation function ReLU ?

Forward propagation

In forward propagation, we go from X to Y by applying a series of functional transformations :

$$Y = \text{SoftMax}(W^{(2)}\text{ReLU}(W^{(1)}X))$$

Questions:

- How do we determine the weights $W^{(1)}$ and $W^{(2)}$?
- Why do we need the activation function ReLU ?
- Why do we use the SoftMax function at the end ?

Weight matrices

How do we determine

the weight values ?

Weight matrices

We want to choose the weights to make

the best predictions possible

Weight matrices

We will use an algorithm

to find these weights

Weight matrices

This algorithm needs

some initial weight values

Weight matrices

How do we choose

the initial weight values ?

Weight matrices

The naive choice

Weight matrices

We set all the weights to zero

Weight matrices

All the predictions are the same:

P=1/3 for the three species

Weight matrices

The weights are updated by the same amount

and thus cannot get unequal values

Weight matrices

The network cannot learn

Weight matrices

A good choice

Weight matrices

Random initialisation around 0

Weight matrices

We choose the weights according to

a normal distribution $\mathcal{N}(0, \sigma^2)$

Weight matrices

We choose the weights according to

a normal distribution $\mathcal{N}(0, \sigma^2)$

The value of σ^2 affects the performance

(speed and accuracy)

Weight matrices

Two common choices

$$\sigma^2 = \frac{1}{N_{in}} \quad \text{(Xavier)}$$

$$\sigma^2 = \frac{2}{N_{in} + N_{out}} \quad \text{(Glorot \& Bengio)}$$

N_{in} = the number of nodes going in the weight matrix W

N_{out} = the number of nodes going out the weight matrix W

Weight matrices

Probabilistic framework

Linear model:

$$Y = W\,X$$

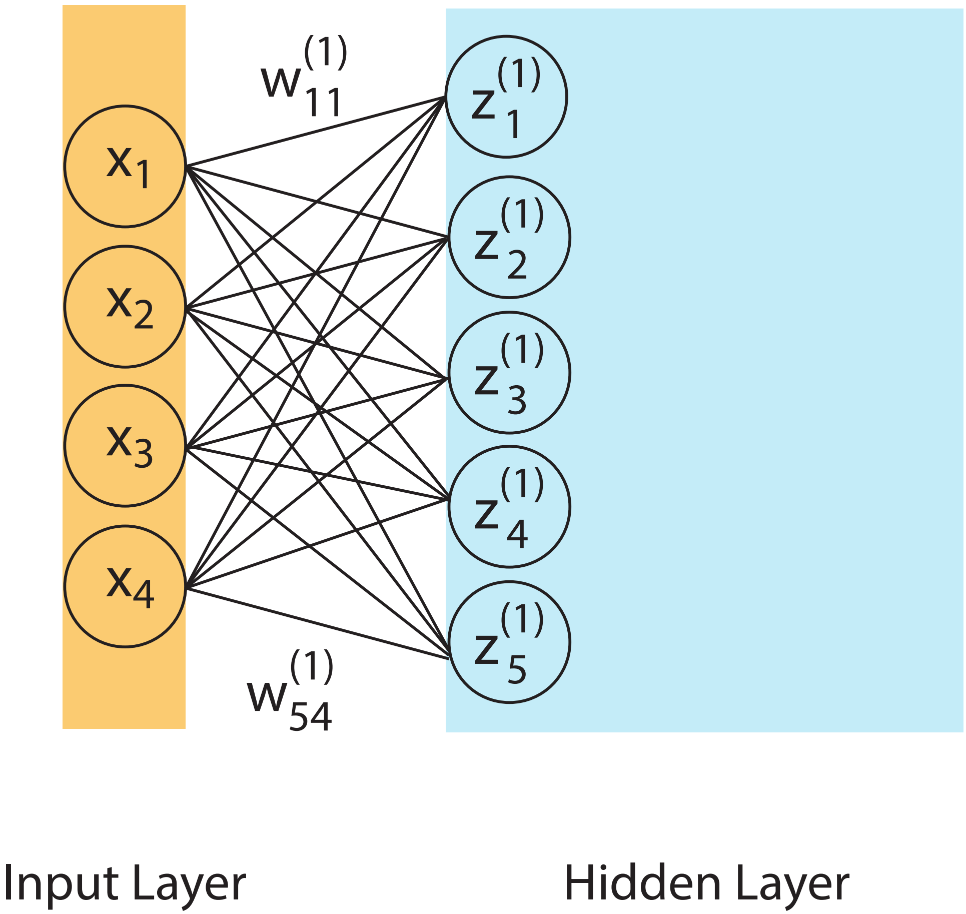
$$Y_i = \sum_{j=1}^{N_{\rm in}} W_{ij}\,X_j\,, \quad i = 1, \ldots, N_{\rm out}$$

Assumptions: $X_j \sim \mathcal{N}(0, \sigma_X^2)$ and $W_{ij} \sim \mathcal{N}(0, \sigma^2)$

The condition $V(Y_i) = V(X_j)$ leads to the values for σ^2:
Xavier and Glorot (when including backpropagation)

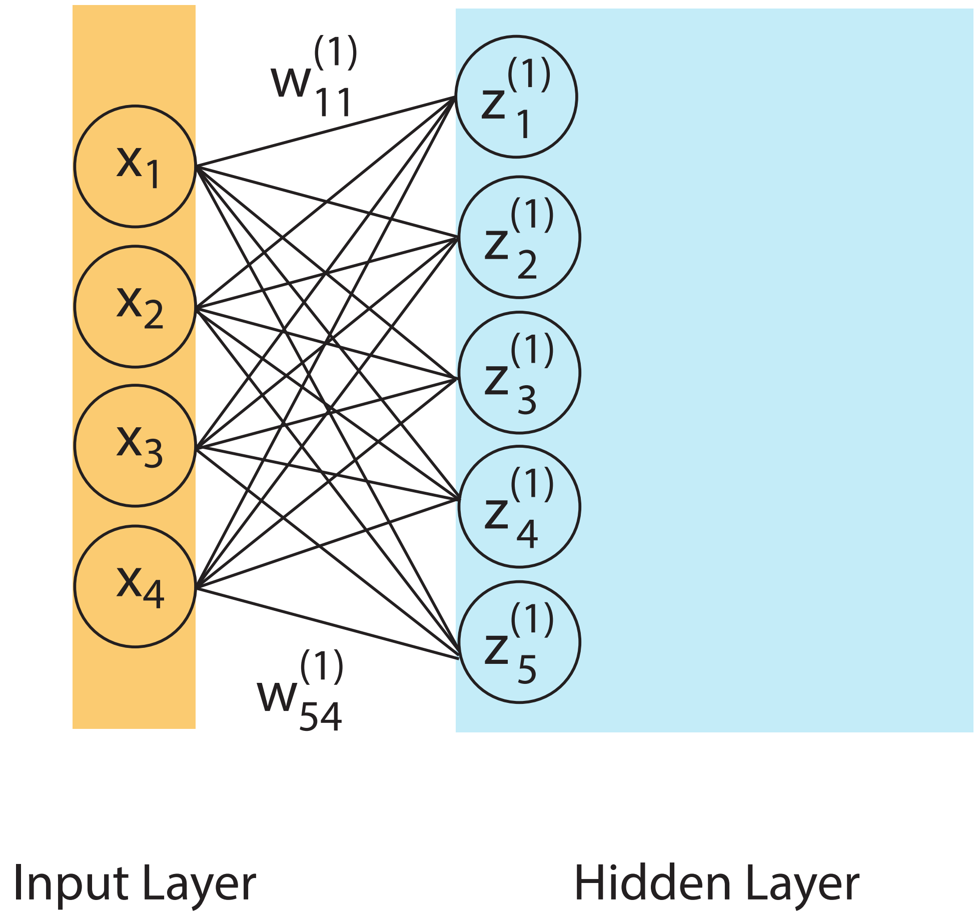
Weight matrices

EXAMPLE



Weight matrices

EXAMPLE

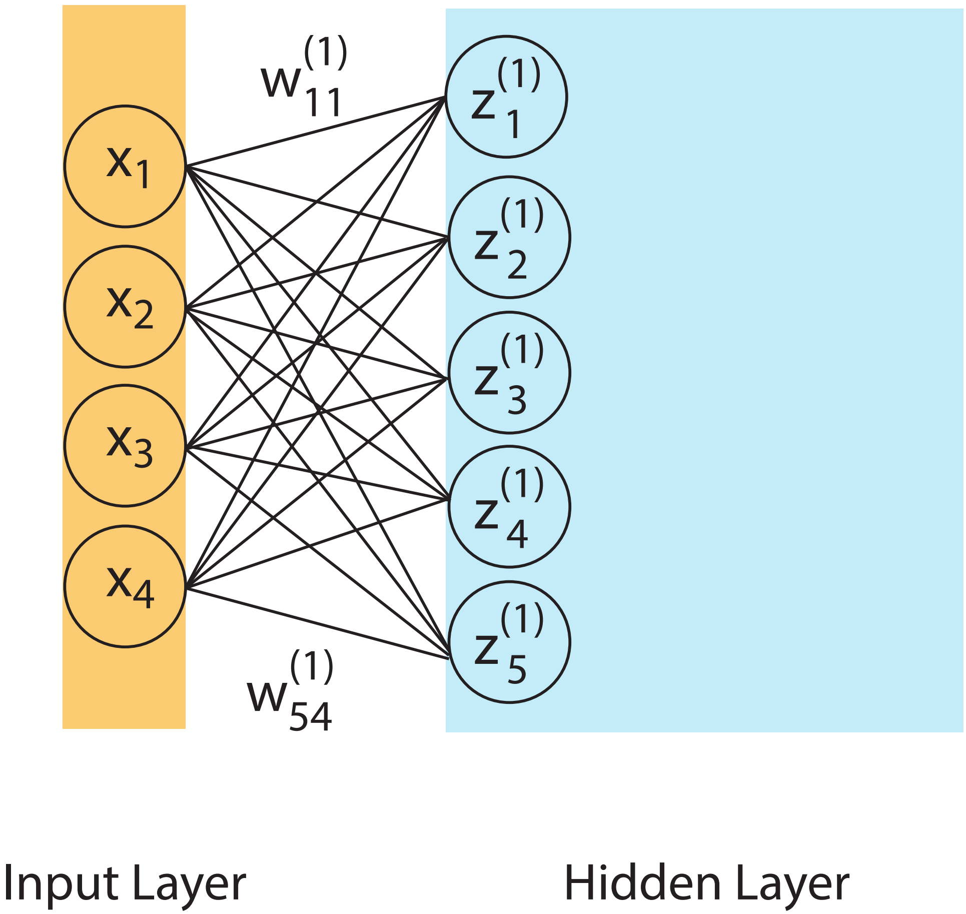


$N_{in} = 4$

$N_{out} = 5$

Weight matrices

EXAMPLE



$N_{in} = 4$

$N_{out} = 5$

$$\begin{pmatrix} 0.5 & 0.1 & -0.2 & -0.4 \\ -0.4 & 1.0 & 0.5 & 1.0 \\ -0.2 & -0.2 & -0.5 & -0.1 \\ 0.2 & 0.7 & 0.3 & 0.2 \\ 0.6 & 0.6 & 0.1 & -0.4 \end{pmatrix}$$

STEP 1

$$Y = \text{SoftMax}(W^{(2)}\text{ReLU}(W^{(1)}X))$$

STEP 1

$$Y = \text{SoftMax}(W^{(2)}\text{ReLU}(W^{(1)}X))$$

input

output

Sepal length	Sepal width	Petal length	Petal width	Species
5.1	3.5	1.4	0.2	setosa
7.0	3.2	4.7	1.4	versicolor
6.3	3.3	6.0	2.5	virginica

STEP 1

$$Y = \text{SoftMax}(W^{(2)}\text{ReLU}(W^{(1)}X))$$

Matrix multiplication ($\sigma = 0.5$):

$$W^{(1)}\,X = \begin{pmatrix} 0.5 & 0.1 & -0.2 & -0.4 \\ -0.4 & 1.0 & 0.5 & 1.0 \\ -0.2 & -0.2 & -0.5 & -0.1 \\ 0.2 & 0.7 & 0.3 & 0.2 \\ 0.6 & 0.6 & 0.1 & -0.4 \end{pmatrix} \begin{pmatrix} 5.1 \\ 3.5 \\ 1.4 \\ 0.2 \end{pmatrix} = \begin{pmatrix} 5.4 \\ 2.2 \\ 2.9 \\ -1.7 \\ -2.4 \end{pmatrix}$$

Activation function

Why do we need

the activation function ReLU ?

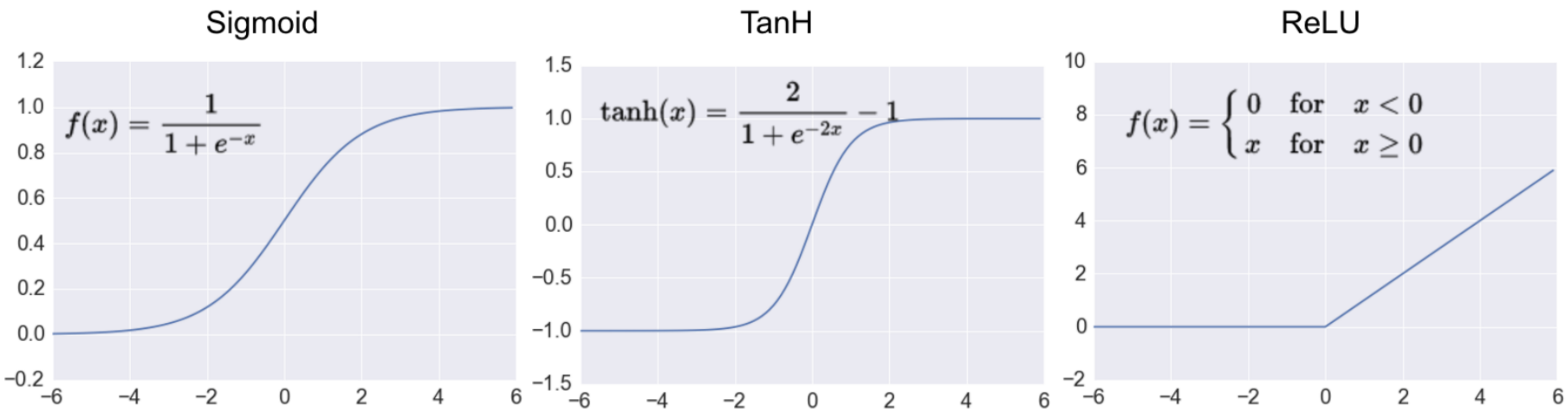
Activation function

A neural network without any non-linear activation function would simply be a linear regression model

Activation function

A neural network without any non-linear activation function would simply be a linear regression model

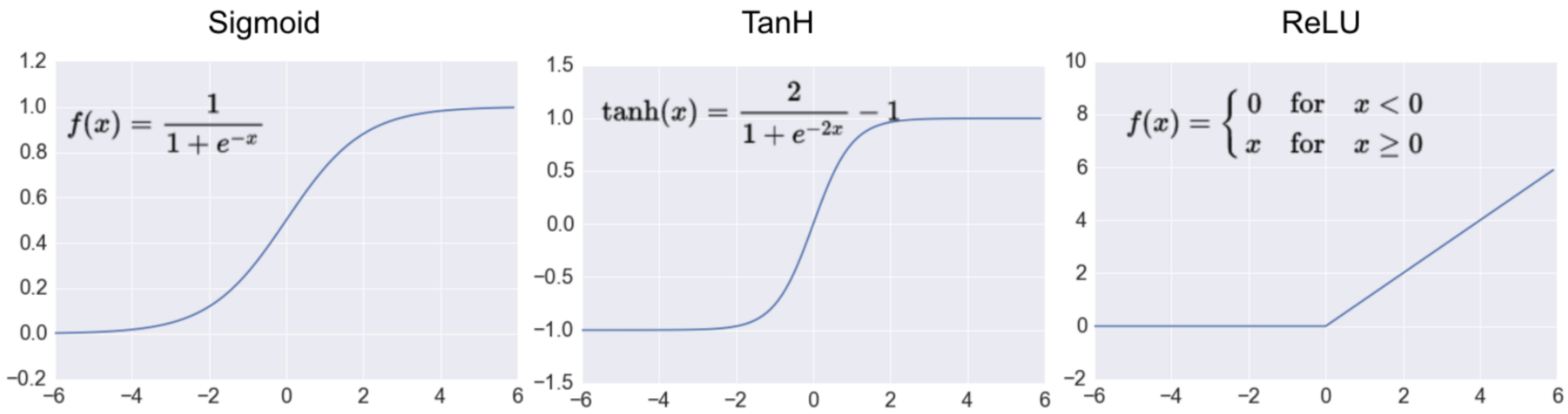
There are several possible non-linear activation functions:



Activation function

A neural network without any non-linear activation function would simply be a linear regression model

There are several possible non-linear activation functions:



ReLU is very simple and very popular in deep neural networks

STEP 2

$$Y = \mathrm{SoftMax}(W^{(2)}\mathrm{ReLU}(W^{(1)}X))$$

STEP 2

$$Y = \text{SoftMax}(W^{(2)}\text{ReLU}(W^{(1)}X))$$

Activation function (ReLU):

$$ReLU(W^{(1)}\ X) = ReLU(\begin{pmatrix} 5.4 \\ 2.2 \\ 2.9 \\ -1.7 \\ -2.4 \end{pmatrix}) = \begin{pmatrix} 5.4 \\ 2.2 \\ 2.9 \\ 0 \\ 0 \end{pmatrix}$$

STEP 3

$$Y = \mathrm{SoftMax}(W^{(2)}\mathrm{ReLU}(W^{(1)}X))$$

STEP 3

$$Y = \mathrm{SoftMax}(W^{(2)}\mathrm{ReLU}(W^{(1)}X))$$

Matrix multiplication ($\sigma = 0.5$):

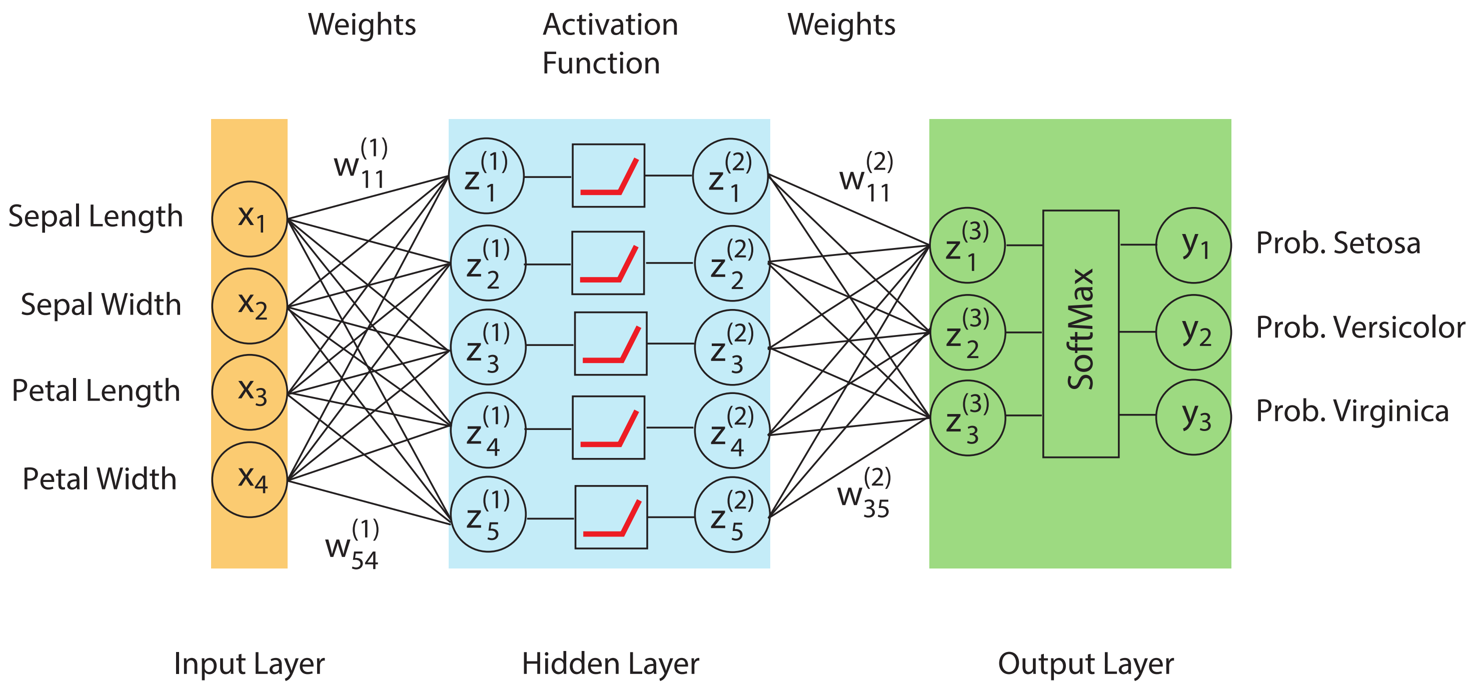
$$W^{(2)}\mathrm{ReLU}(W^{(1)}X) =$$

$$\begin{pmatrix} 0.6 & 0.1 & 0.9 & -0.2 & -0.5 \\ 0.3 & -0.3 & 0.3 & -0.9 & -0.9 \\ 0.3 & 0.2 & 0.4 & -1.0 & 0.6 \end{pmatrix} \begin{pmatrix} 5.4 \\ 2.2 \\ 2.9 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1.9 \\ 0.3 \\ 2.9 \end{pmatrix}$$

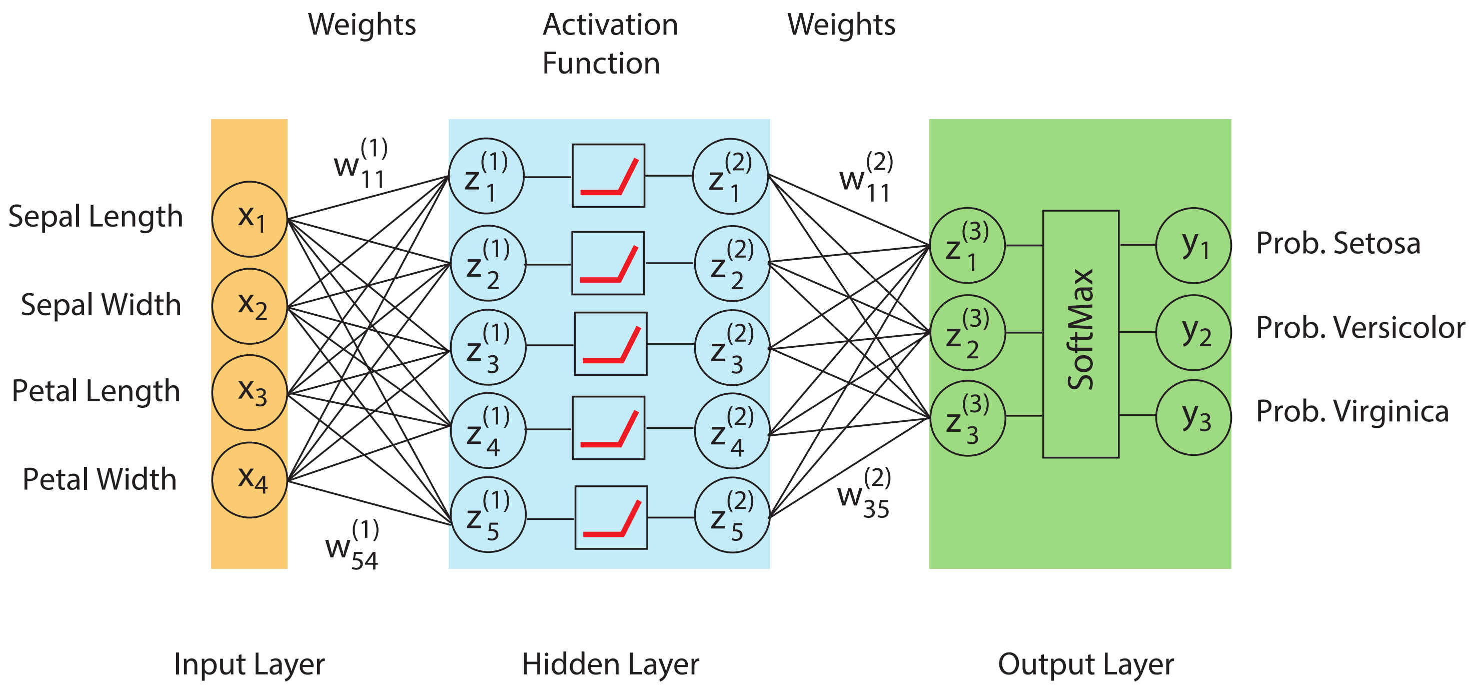
Why do we use the

SoftMax function ?

SoftMax function



SoftMax function

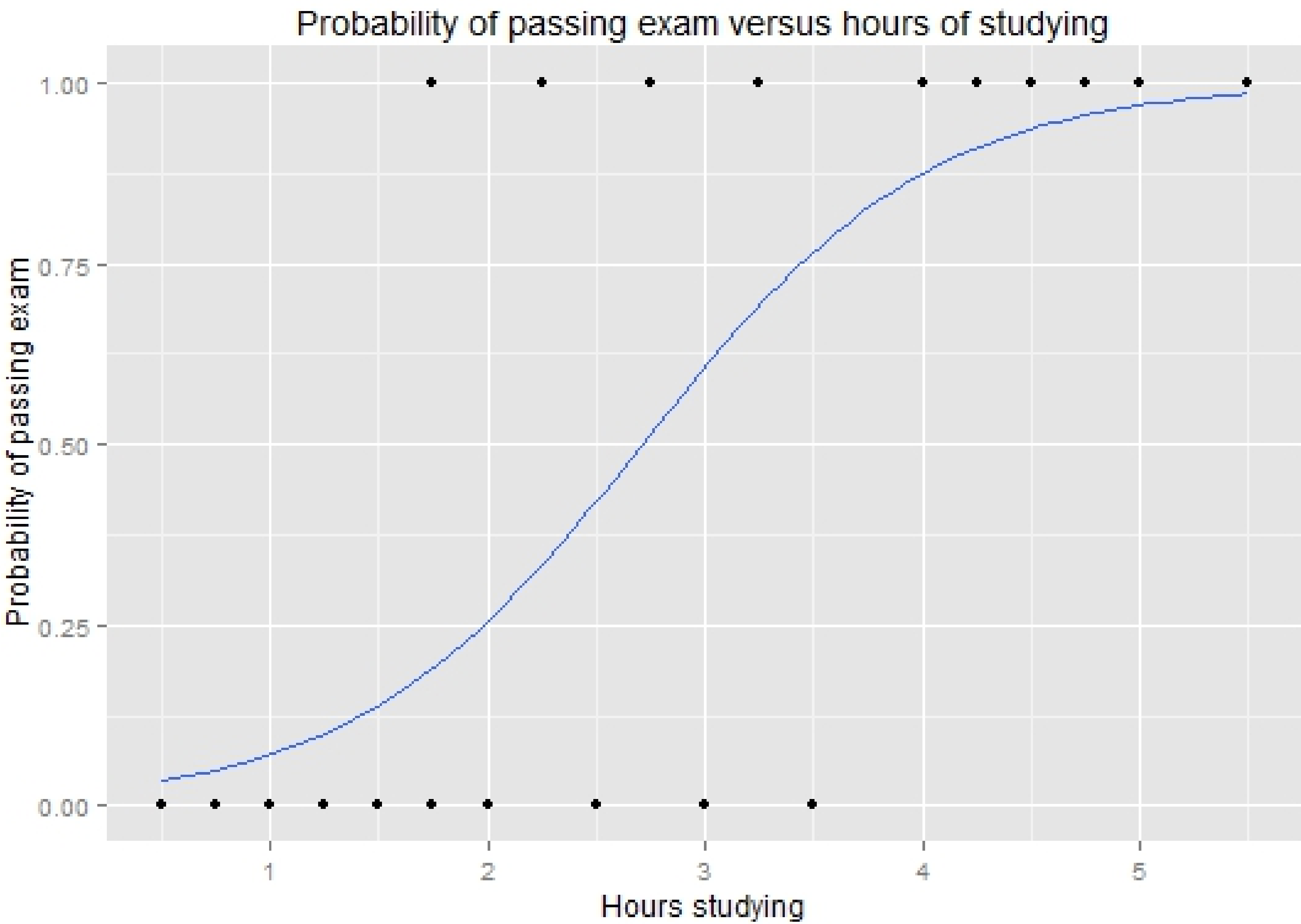


The output $Z^{(3)}$ are arbitrary real values and we want to convert them into probabilities:

$$\text{SoftMax}(Z^{(3)})_i = \frac{e^{z_i^{(3)}}}{e^{z_1^{(3)}} + e^{z_2^{(3)}} + e^{z_3^{(3)}}} = \text{Prob. Class i} \in (0, 1)$$

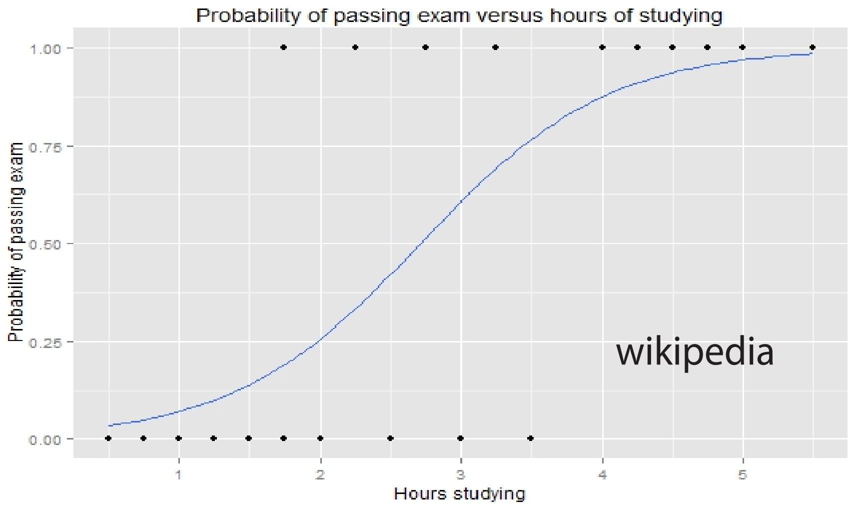
Binary classification

SoftMax function



SoftMax function

Binary classification:



Empirical observation - logistic regression:

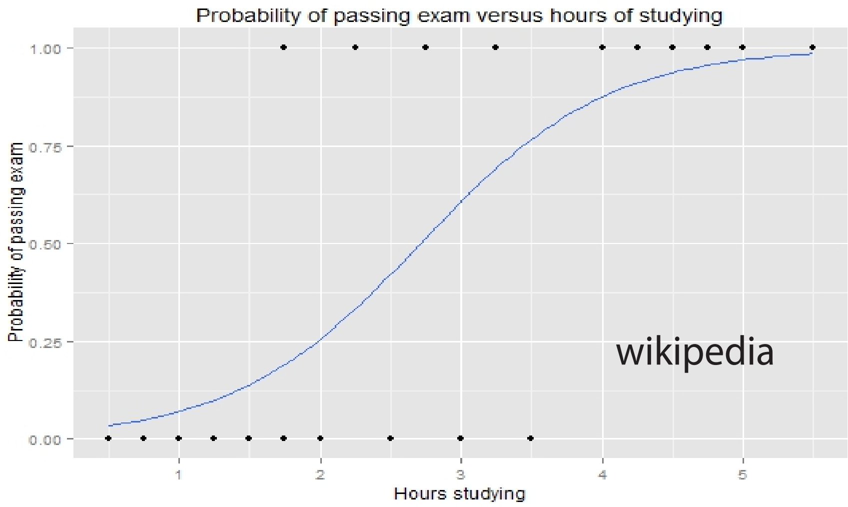
$$p(x) = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}$$

Motivate - sigmoid function (1 output neuron):

$$S(x) = \frac{e^x}{1+e^x}$$

SoftMax function

Binary classification:



Empirical observation - logistic regression:

$$p(x) = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}}$$

Motivate - sigmoid function (1 output neuron):

$$S(x) = \frac{e^x}{1 + e^x}$$

Multinomial logistic regression:

$$P(X)_\ell = \frac{e^{\beta_\ell X}}{\sum_{k=1}^{K} e^{\beta_k X}}$$

SoftMax function (K classes):

$$\text{SoftMax}(X)_\ell = \frac{e^{X_\ell}}{\sum_{k=1}^{K} e^{X_k}}$$

STEP 4

$$Y = \mathrm{SoftMax}(W^{(2)}\mathrm{ReLU}(W^{(1)}X))$$

STEP 4

$$Y = \mathrm{SoftMax}(W^{(2)}\mathrm{ReLU}(W^{(1)}X))$$

SoftMax function:

$$\mathrm{SoftMax}(W^{(2)}\mathrm{ReLU}(W^{(1)}X)) = \mathrm{SoftMax}(\begin{pmatrix} 1.9 \\ 0.3 \\ 2.9 \end{pmatrix}) = \begin{pmatrix} 0.25 \\ 0.05 \\ 0.70 \end{pmatrix}$$

STEP 4

$$Y = \text{SoftMax}(W^{(2)}\text{ReLU}(W^{(1)}X))$$

SoftMax function:

$$\text{SoftMax}(W^{(2)}\text{ReLU}(W^{(1)}X)) = \text{SoftMax}(\begin{pmatrix} 1.9 \\ 0.3 \\ 2.9 \end{pmatrix}) = \begin{pmatrix} 0.25 \\ 0.05 \\ 0.70 \end{pmatrix}$$

STEP 4

$$Y = \text{SoftMax}(W^{(2)}\text{ReLU}(W^{(1)}X))$$

SoftMax function:

$$\text{SoftMax}(W^{(2)}\text{ReLU}(W^{(1)}X)) = \text{SoftMax}(\begin{pmatrix} 1.9 \\ 0.3 \\ 2.9 \end{pmatrix}) = \begin{pmatrix} 0.25 \\ 0.05 \\ 0.70 \end{pmatrix}$$

Prediction: Virginica

STEP 4

$$Y = \text{SoftMax}(W^{(2)}\text{ReLU}(W^{(1)}X))$$

SoftMax function:

$$\text{SoftMax}(W^{(2)}\text{ReLU}(W^{(1)}X)) = \text{SoftMax}(\begin{pmatrix} 1.9 \\ 0.3 \\ 2.9 \end{pmatrix}) = \begin{pmatrix} 0.25 \\ 0.05 \\ 0.70 \end{pmatrix}$$

Prediction: Virginica This is wrong! It is Setosa

Network training using gradient descent

What to do when

the prediction is wrong ?