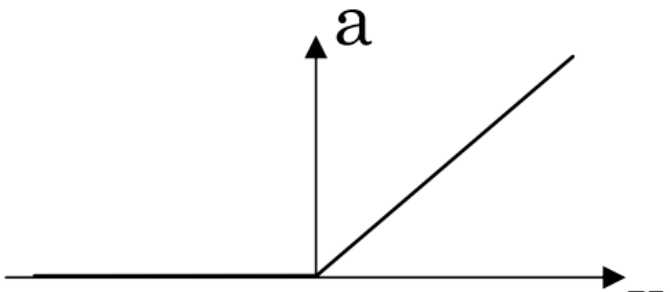
Neural Network

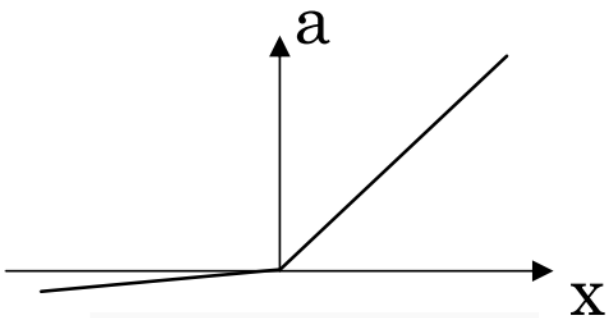
Activation Function

- Rectified Linear Unit(ReLU)



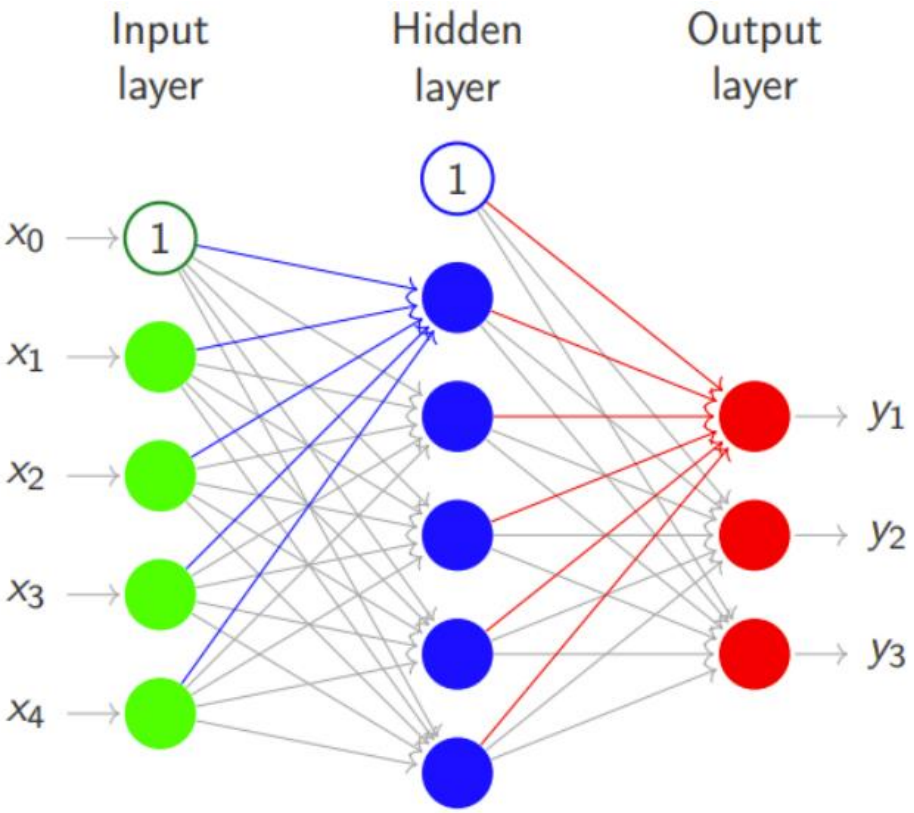
$$f(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ x & \text{for} \quad x \geq 0 \end{cases}$$

- Leaky ReLU



$$f(x) = \begin{cases} \alpha x & \text{for} \quad x < 0 \\ x & \text{for} \quad x \geq 0 \end{cases}$$

Neural Network



Output of a neural network can be visualised graphically as *forward propagation of information*.

An example: data (loan application)

Approved or not

ID	Age	Has_Job	Own_House	Credit_Rating	Class
1	young	false	false	fair	**No**
2	young	false	false	good	**No**
3	young	true	false	good	**Yes**
4	young	true	true	fair	**Yes**
5	young	false	false	fair	**No**
6	middle	false	false	fair	**No**
7	middle	false	false	good	**No**
8	middle	true	true	good	**Yes**
9	middle	false	true	excellent	**Yes**
10	middle	false	true	excellent	**Yes**
11	old	false	true	excellent	**Yes**
12	old	false	true	good	**Yes**
13	old	true	false	good	**Yes**
14	old	true	false	excellent	**Yes**
15	old	false	false	fair	**No**

An example: the learning task

- Learn a classification model from the data
- Use the model to classify future loan applications into
  - Yes (approved) and
  - No (not approved)
- What is the class for following case/instance?

Age	Has_Job	Own_house	Credit-Rating	Class
young	false	false	good	?

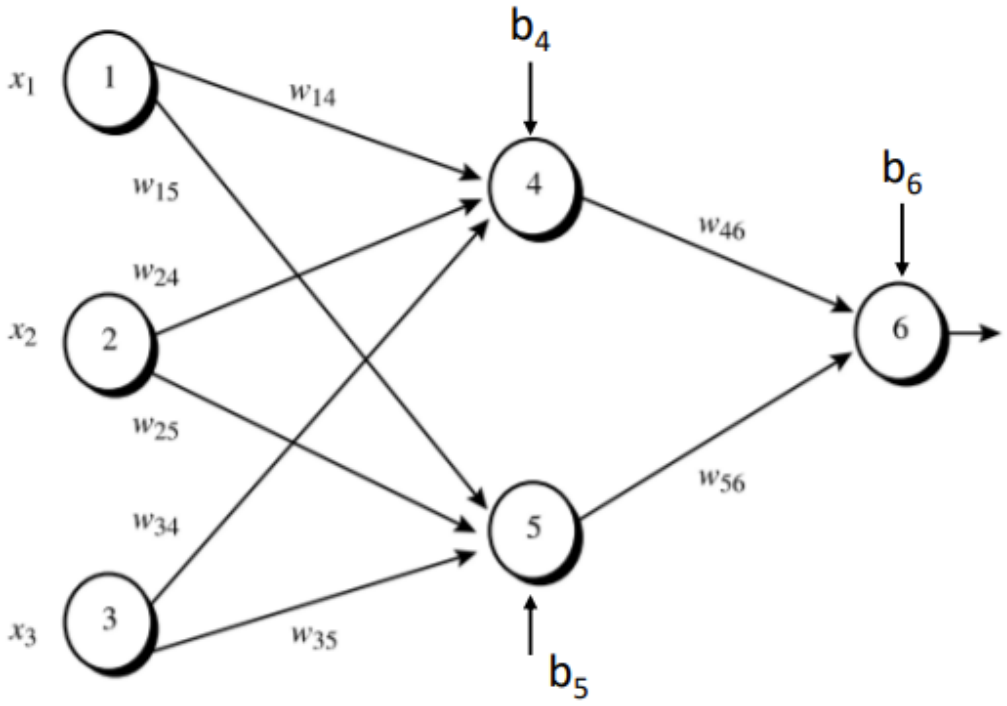
Example

- Number of input neurons (Input Layer)=2
- Number of hidden neurons (Hidden Layer)=3
- Number of output neurons (output Layer)=1

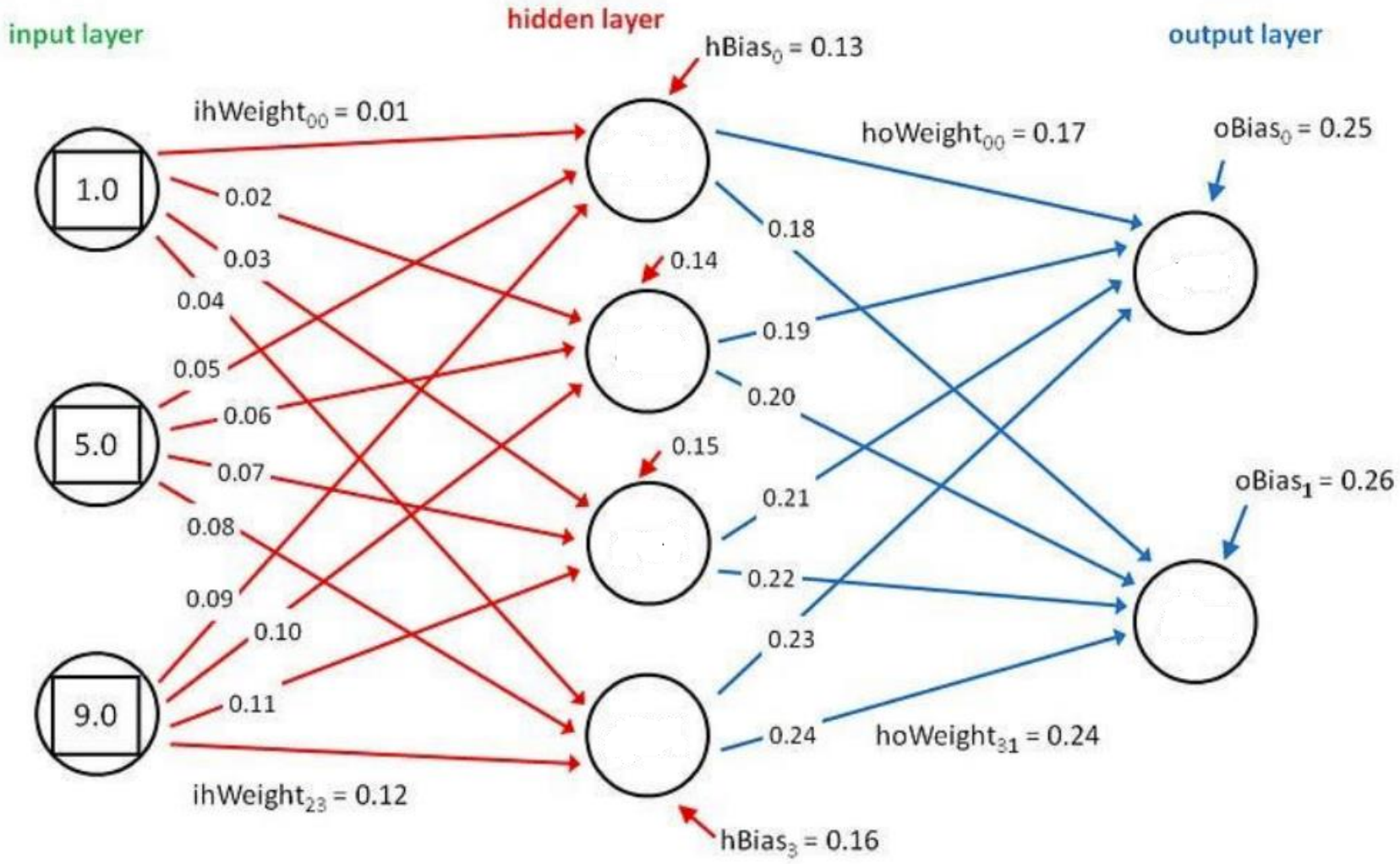
- Activation function for both hidden and output layer is sigmoid

Example

x1	x2	x3	w14	w15	w24	w25	w34	w35	w46	w56	b4	b5	b6
1	0	1	0.2	-0.3	0.4	0.1	-0.5	0.2	-0.3	-0.2	-0.4	0.2	0.1



Example


input layer
hidden layer
output layer
hBias0 = 0.13
ihWeight00 = 0.01
hoWeight00 = 0.17
oBias0 = 0.25
1.0
0.02
0.03
0.04
0.18
0.14
0.19
0.05
0.20
5.0
0.06
0.15
0.07
oBias1 = 0.26
0.21
0.08
0.22
0.09
0.10
0.23
9.0
0.11
0.24
hoWeight31 = 0.24
ihWeight23 = 0.12
hBias3 = 0.16

One HOT Encoding