100 JAAR YEARS 1913 - 2013 IMPACT

NEURAL NETWORKS

Jim Senft
Sander Stuut
Bart van den Elshout
Bart Lammers

Erasmus Universiteit Rotterdam

LECTURE

Part 1: Biological neural networks

Part 2: Artificial neural networks

Part 3: Real life applications

Part 4: Data preparation

Part 5: Method application & Pay-off analysis

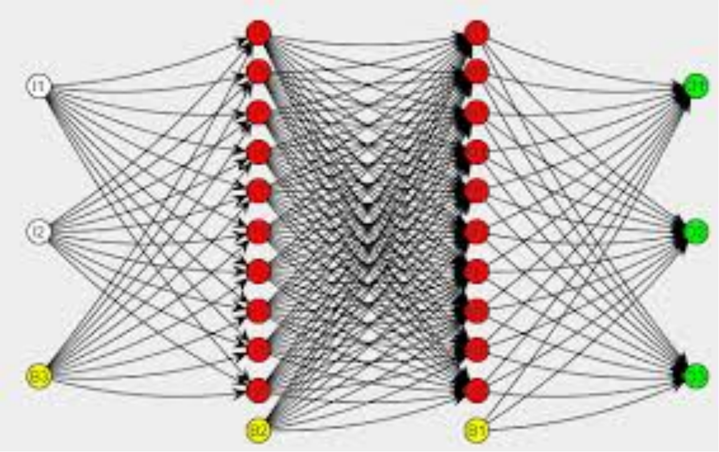
PART 1

Biological Neural Networks

BIOLOGICAL NEURAL NETWORKS

- Biological

- Artificial



BIOLOGICAL NEURAL NETWORK

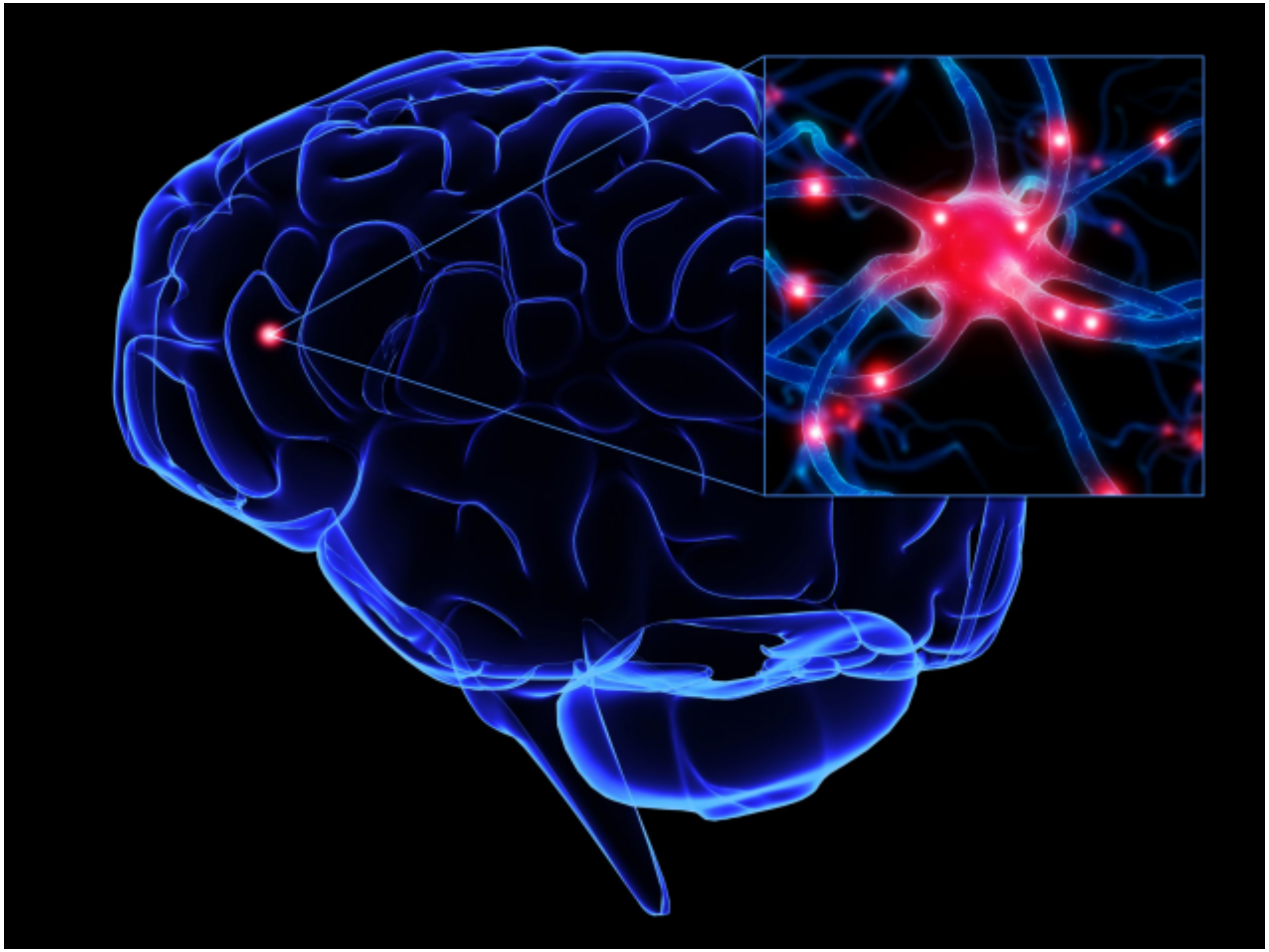
- Artificial neural networks are based on biological neural networks.

- By gaining an understanding of the basic principles underlying a biological neural network, it is much easier to grasp the concept of an artificial neural network.

- Properties of biological neural networks:
  - intuitive interaction with data
  - pattern recognition
  - learning capability

- Application of same properties in artificial neural networks.

NEURON



SYNAPSE

- A synapse is a structure that permits a neuron to pass a signal to another neuron or cell.

SYNAPTIC PATHWAYS

- Information travels from neuron to neuron through synaptic pathways.
- Data travels several complex routes through the network, which may be back and forth, resulting in an interpretation of the data, as well as learning.

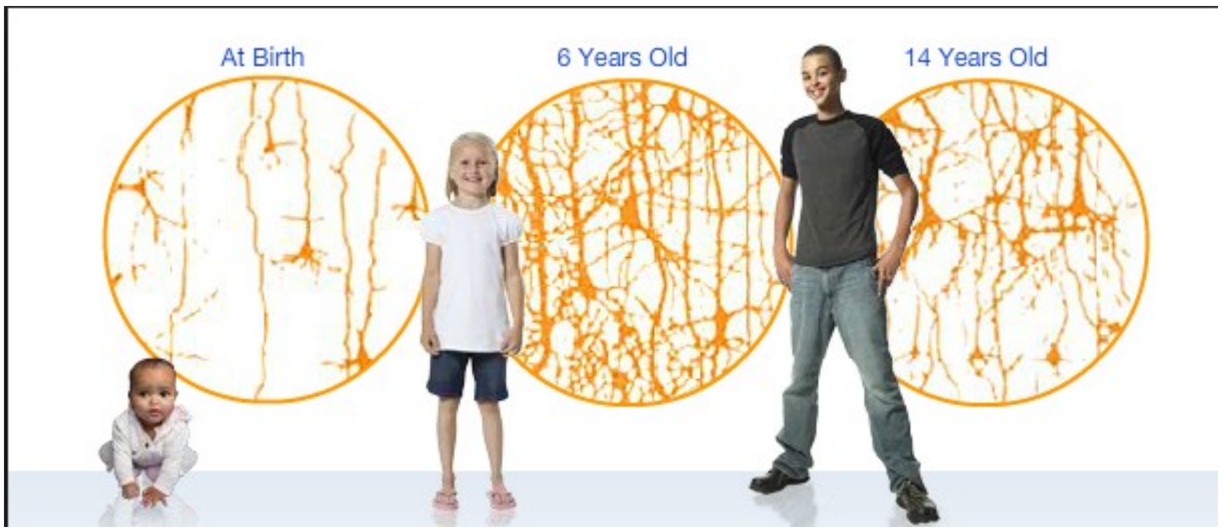
SYNAPTIC PATHWAYS

- In the grand scheme, these synaptic pathways allow organisms to learn and adapt.

- “How” this works is unknown. However, we understand the basic principle of it, which gives rise to our abilities such as learning.

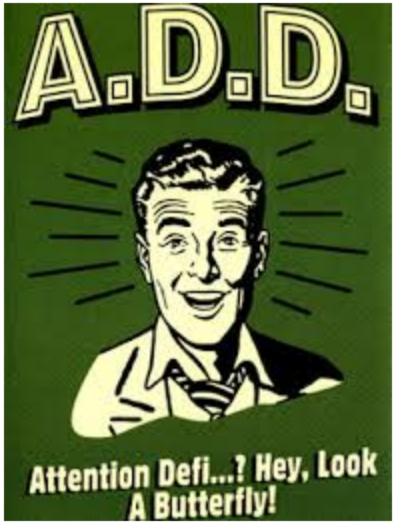
EXAMPLE 1: LEARNING IN YOUNG ORGANISMS

- In early life, children develop huge amounts of synaptic pathways.
- This is beneficial, as it prepares an organism for many different possible environments (hence, many different sorts of data to interact with).
  For this reason, young organisms have greater learning ability.
- As organisms grow older, synaptic pathways are cut off.
  This increases the efficiency of the network.



EXAMPLE 2: LATENT INHIBITION IN BIOLOGICAL NEURAL NETWORKS

- Latent inhibition – ability to filter irrelevant data



- ADD – many synapses, low latent inhibition
- Result: preferred output not achieved, due to lack of proper “data reduction”

CONCEPT OF A NEURAL NETWORK

- Three conceptual phases:

Input – Processing – Output

1. Data goes in. (sensory experience, etc.)
2. Data is processed. (synaptic activity)
3. Data comes out. (knowledge, understanding)

CONCEPT OF A NEURAL NETWORK

- Three conceptual phases:

Input – Processing – Output

1. Data goes in.	(sensory experience, etc.)
2. Data is processed.	(synaptic activity)
3. Data comes out.	(knowledge, understanding)

The input is known.
The output is known.
The processing is a “black box”.

Whatever happens in phase 2, it enables biological systems to interact intuitively with data.

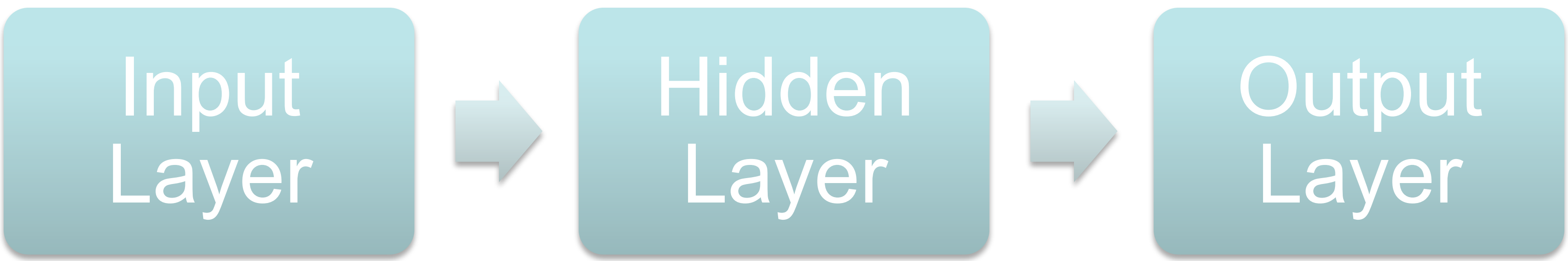
PREMISE OF AN ARTIFICIAL NEURAL NETWORK

- Biological neural networks enable organisms to interact intuitively with data. Hence they are able to learn.
- Ability to recognize patterns.
- Pattern recognition leads to predictive capability.
- Successful concept for how to interact with data intuitively.
- When applied to software or other systems, the benefits might be similar.

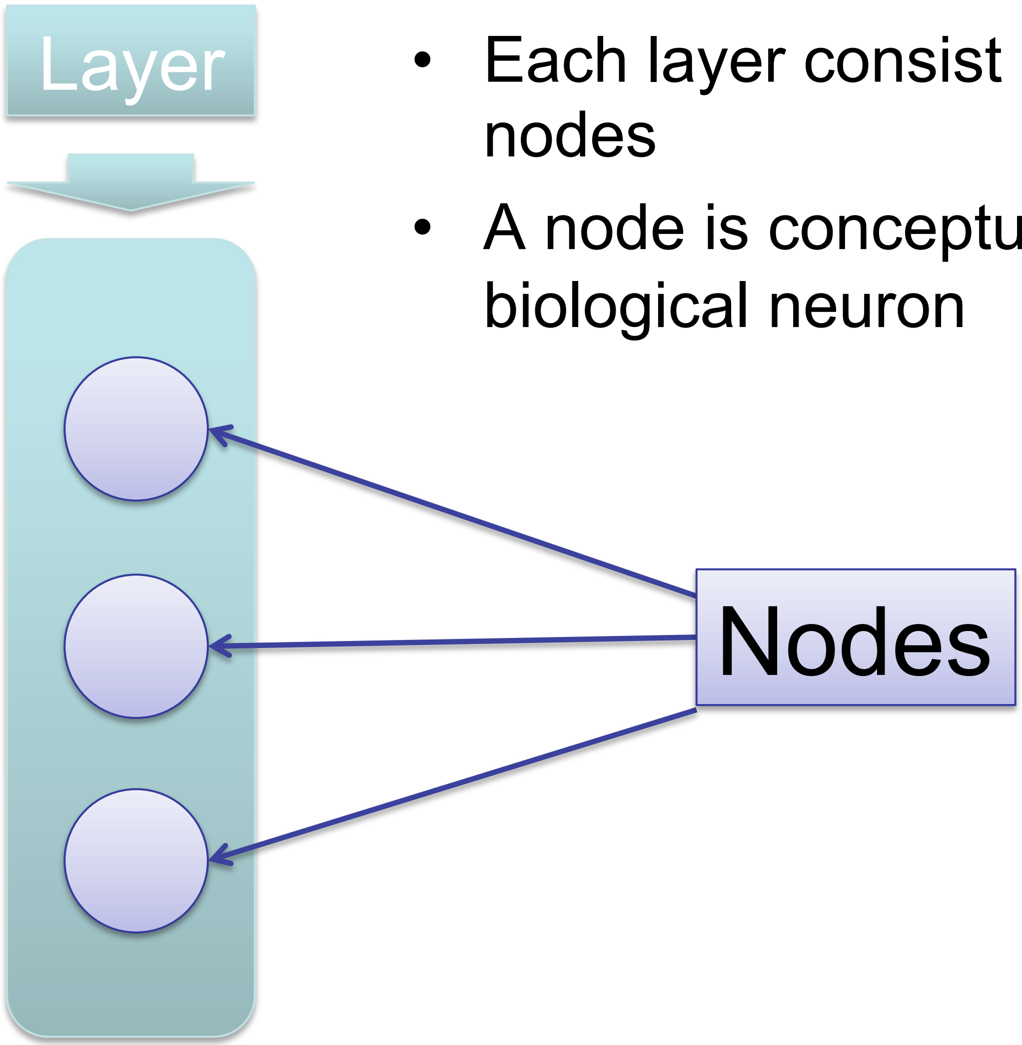
PART 2

Artificial Neural Networks

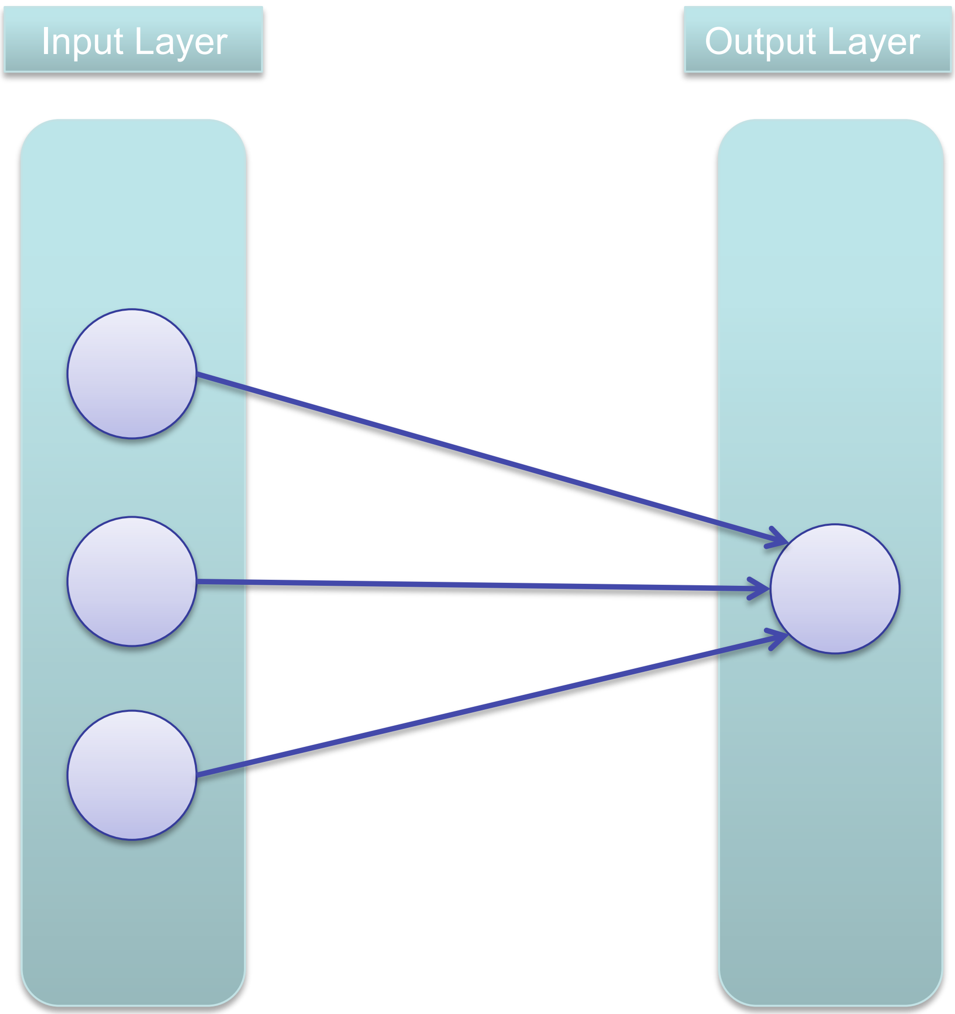
Artificial Neural Network model



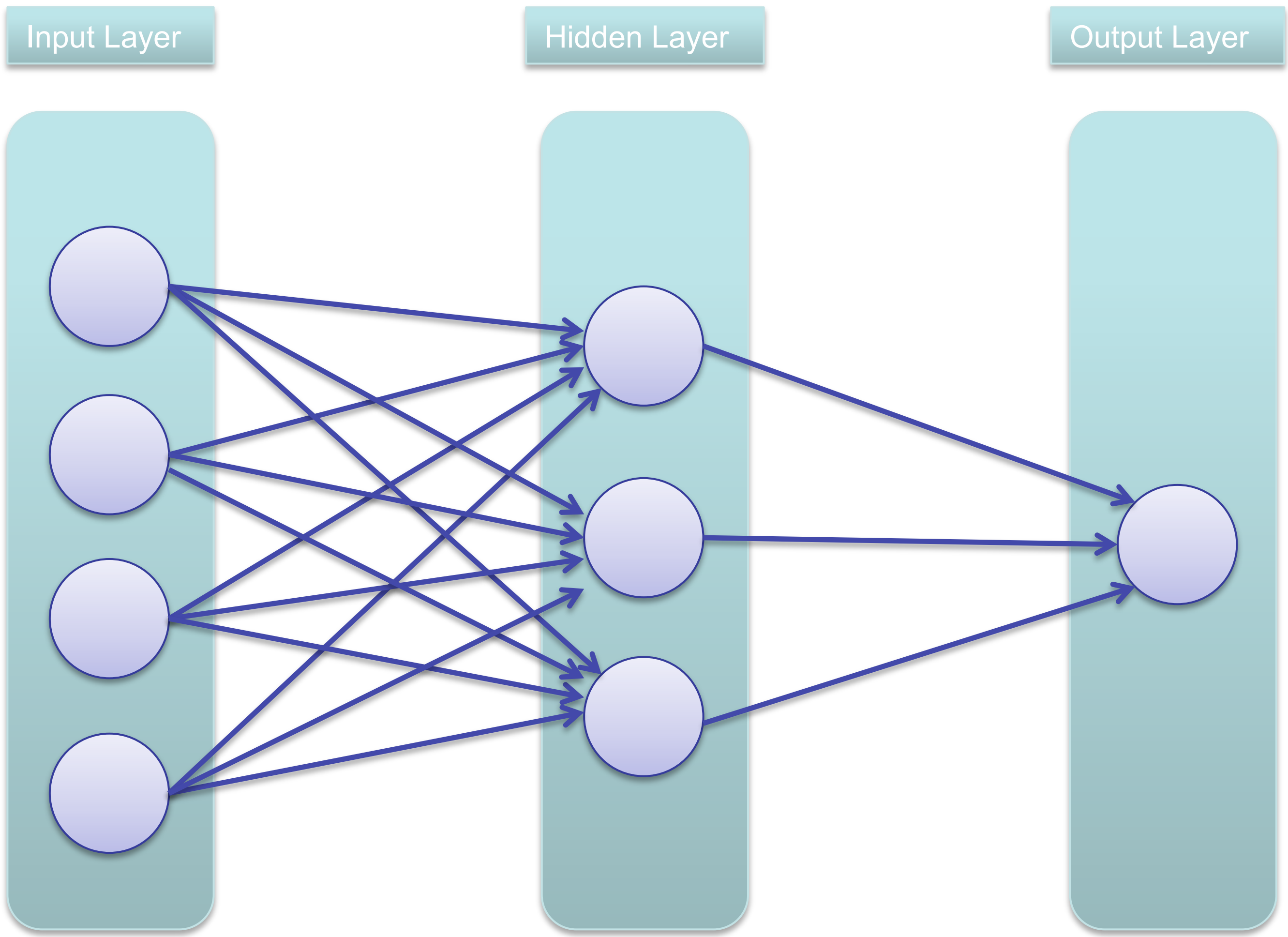
- Each layer consist of a number of nodes
- A node is conceptually equivalent to a biological neuron

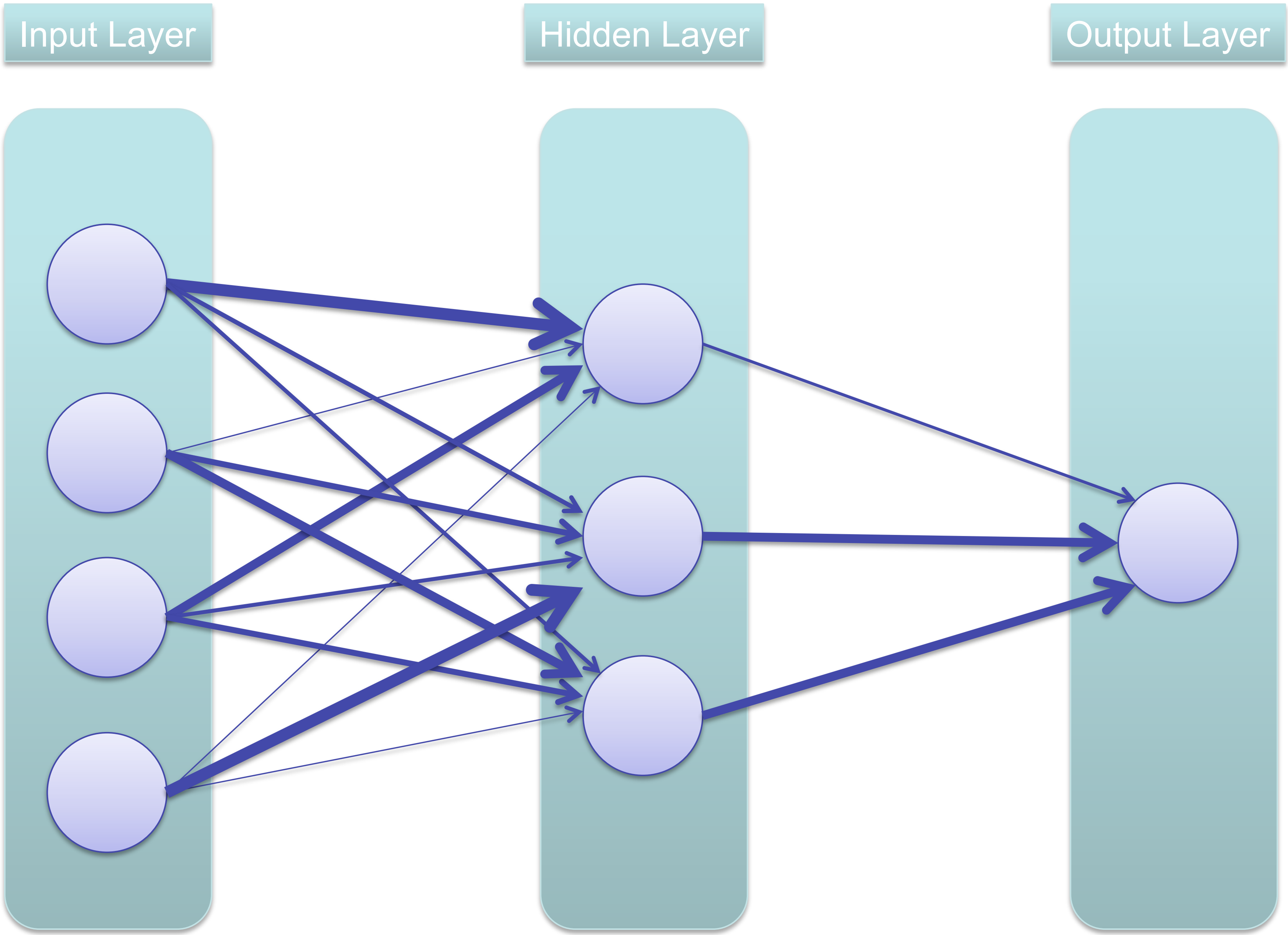


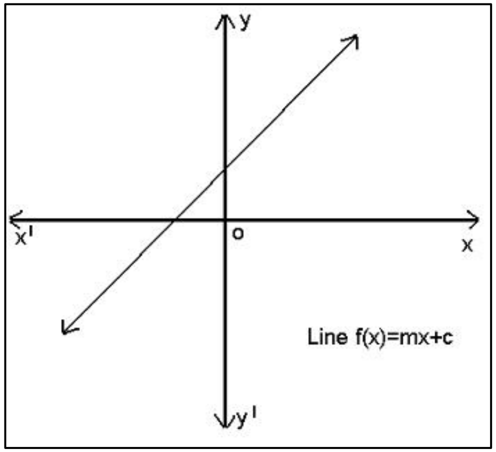
Single-layer Artificial Neural Network
Perceptron



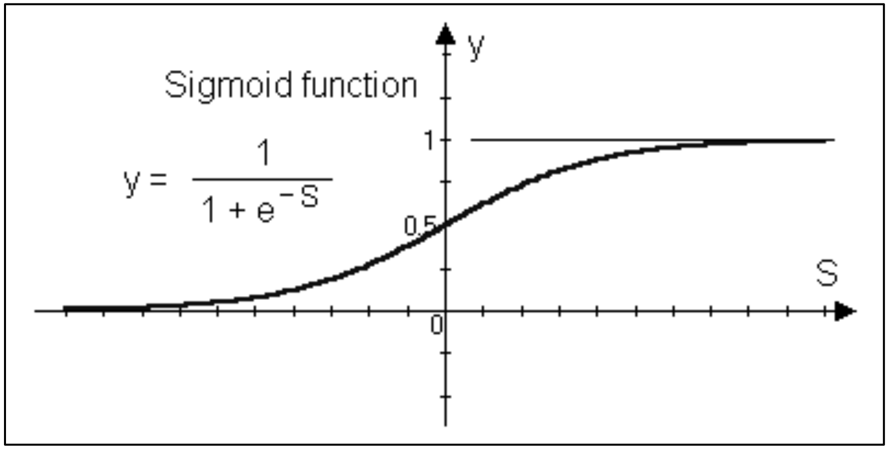
Multilayer Artificial Neural Network
Multilayer Perceptron



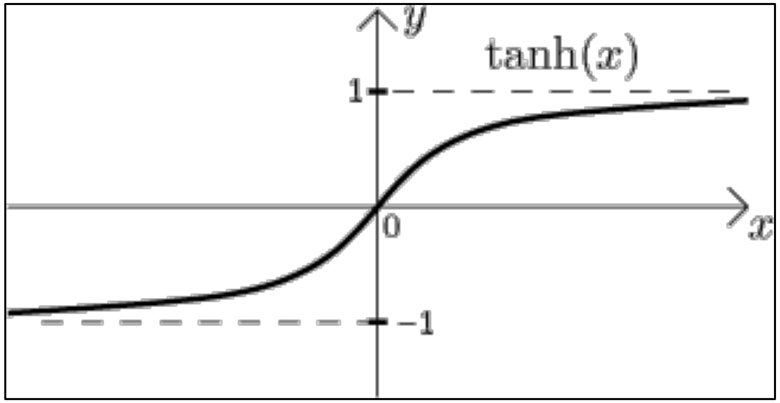

Input Layer
Hidden Layer
Output Layer



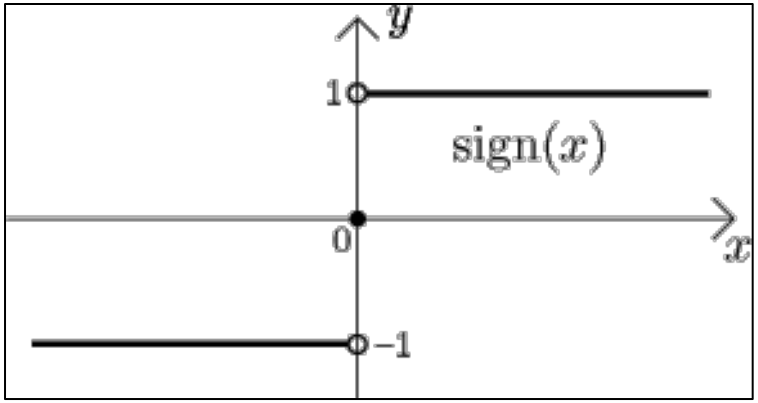
Linear function



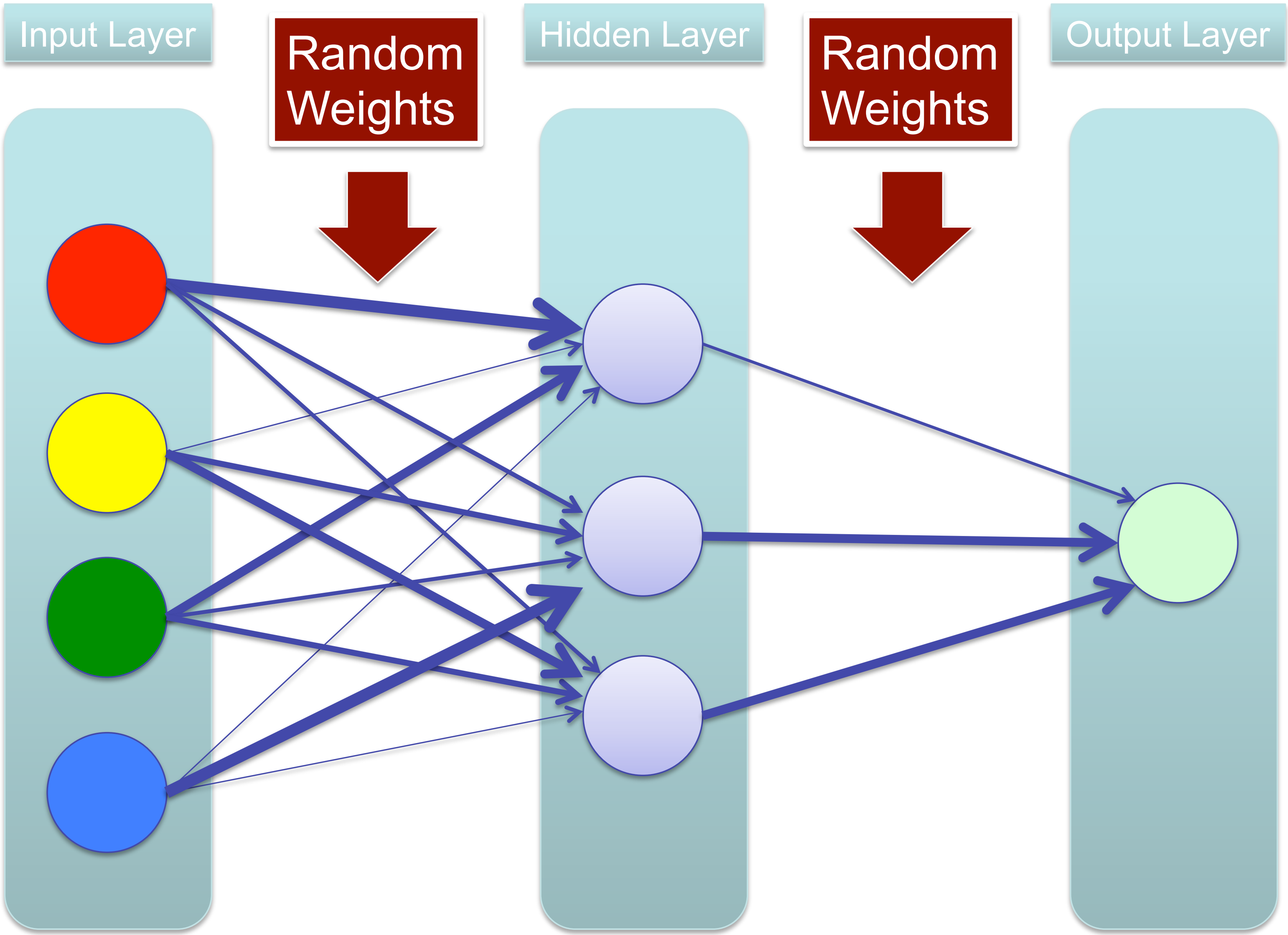
Sigmoid function

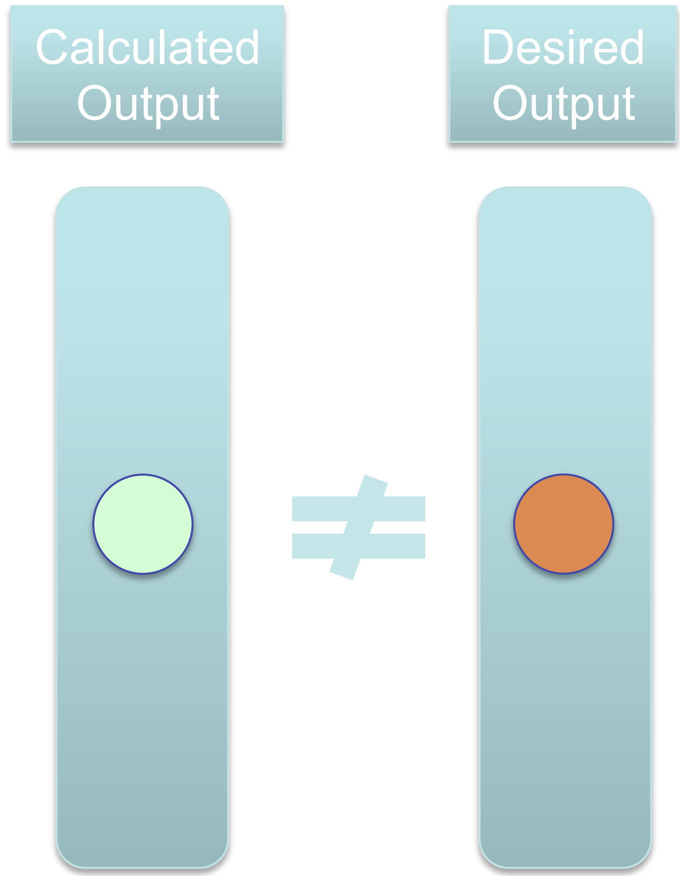


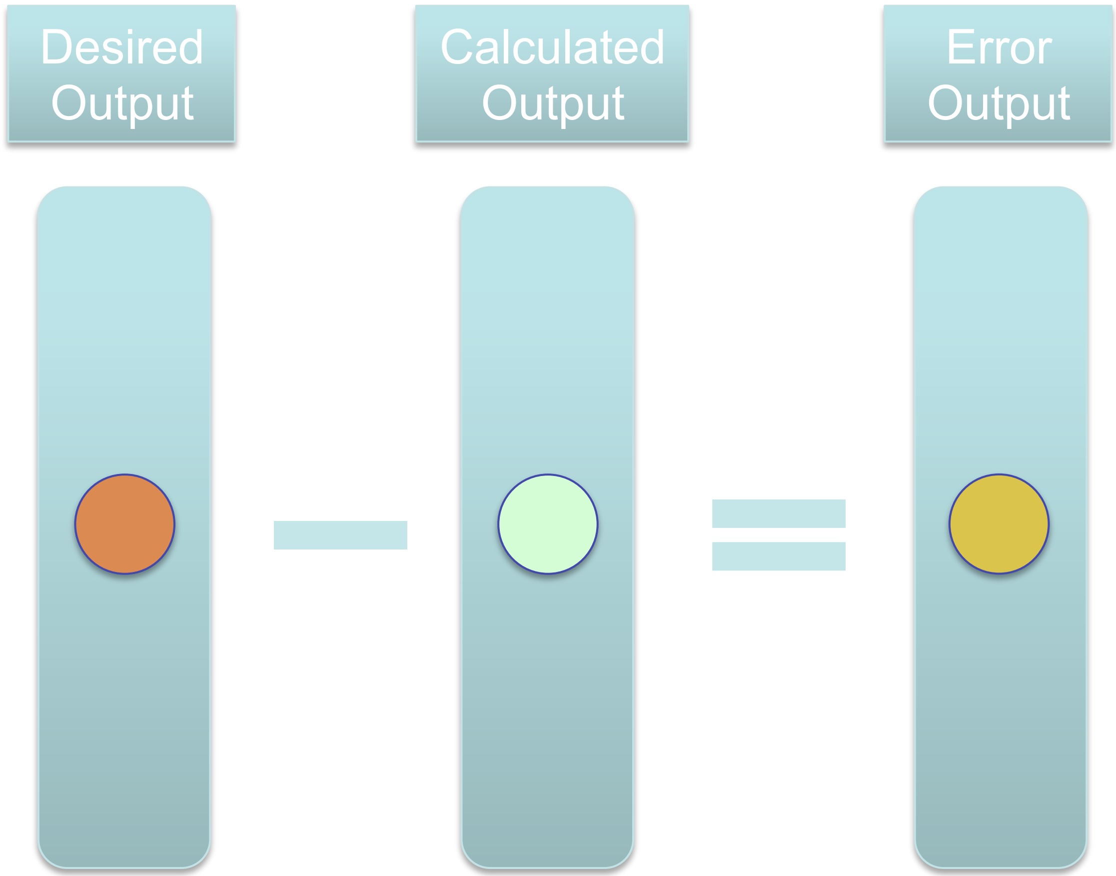
Tanh function



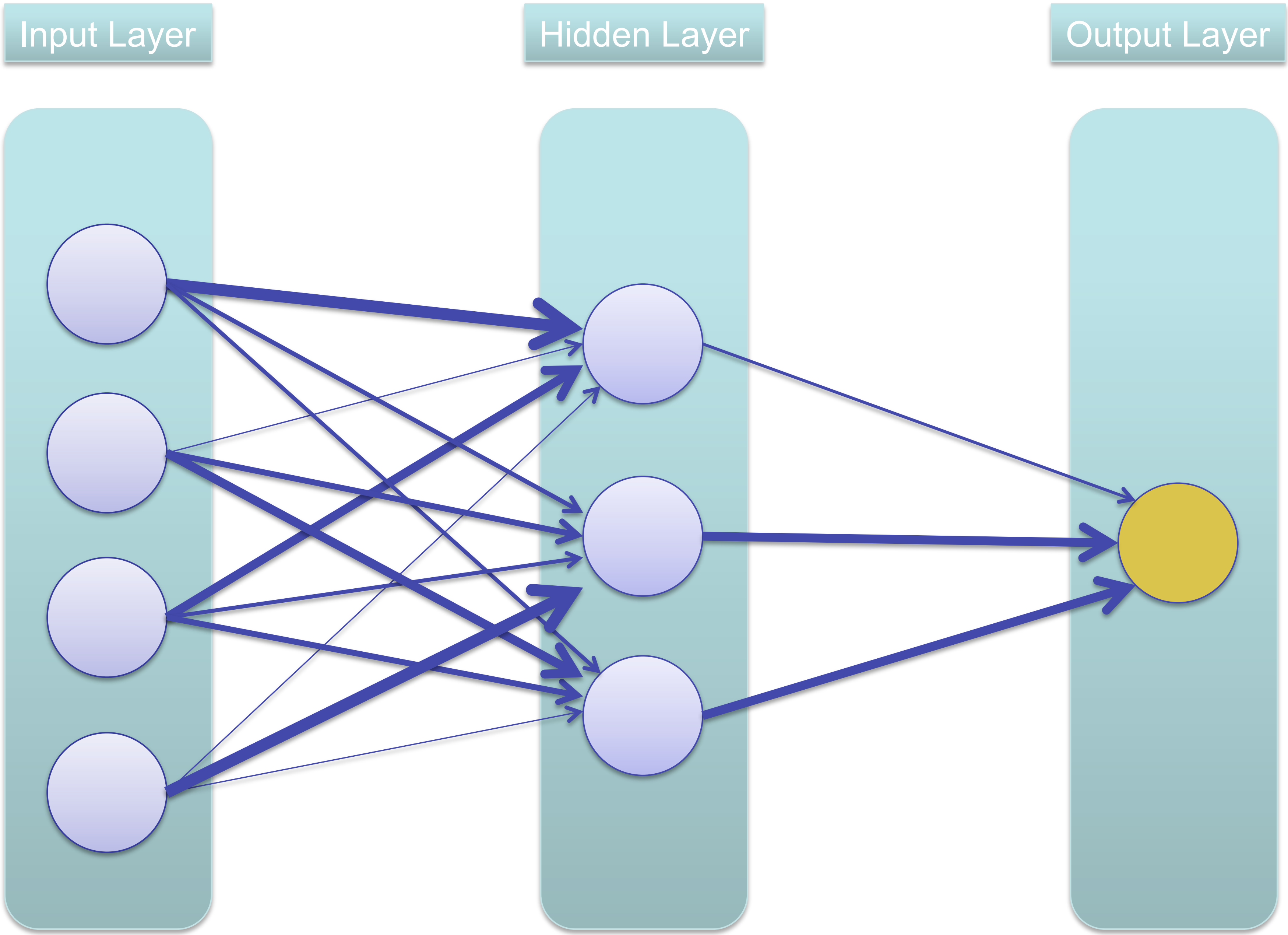
Sign function

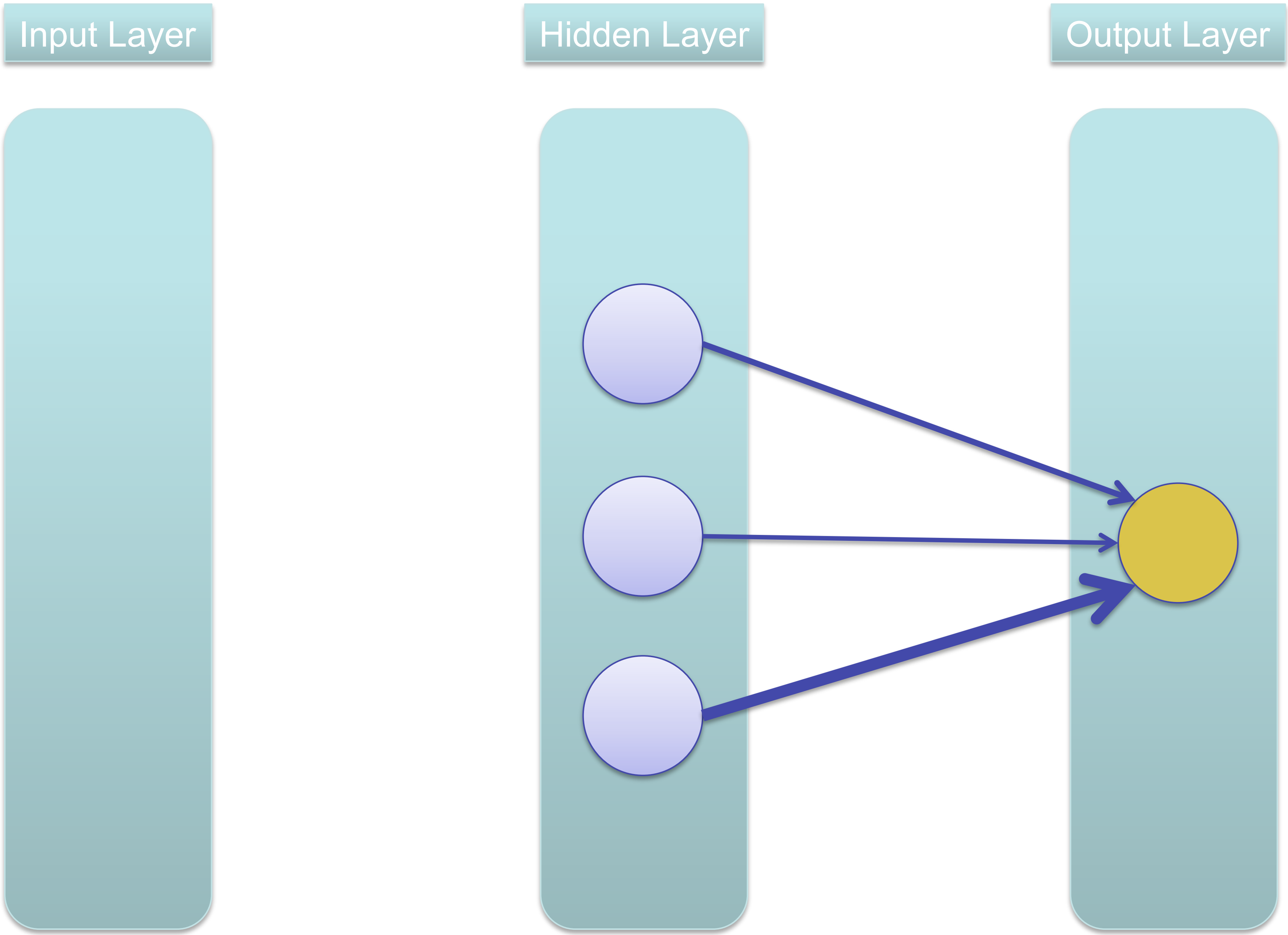

Input Layer
Random Weights
Hidden Layer
Random Weights
Output Layer

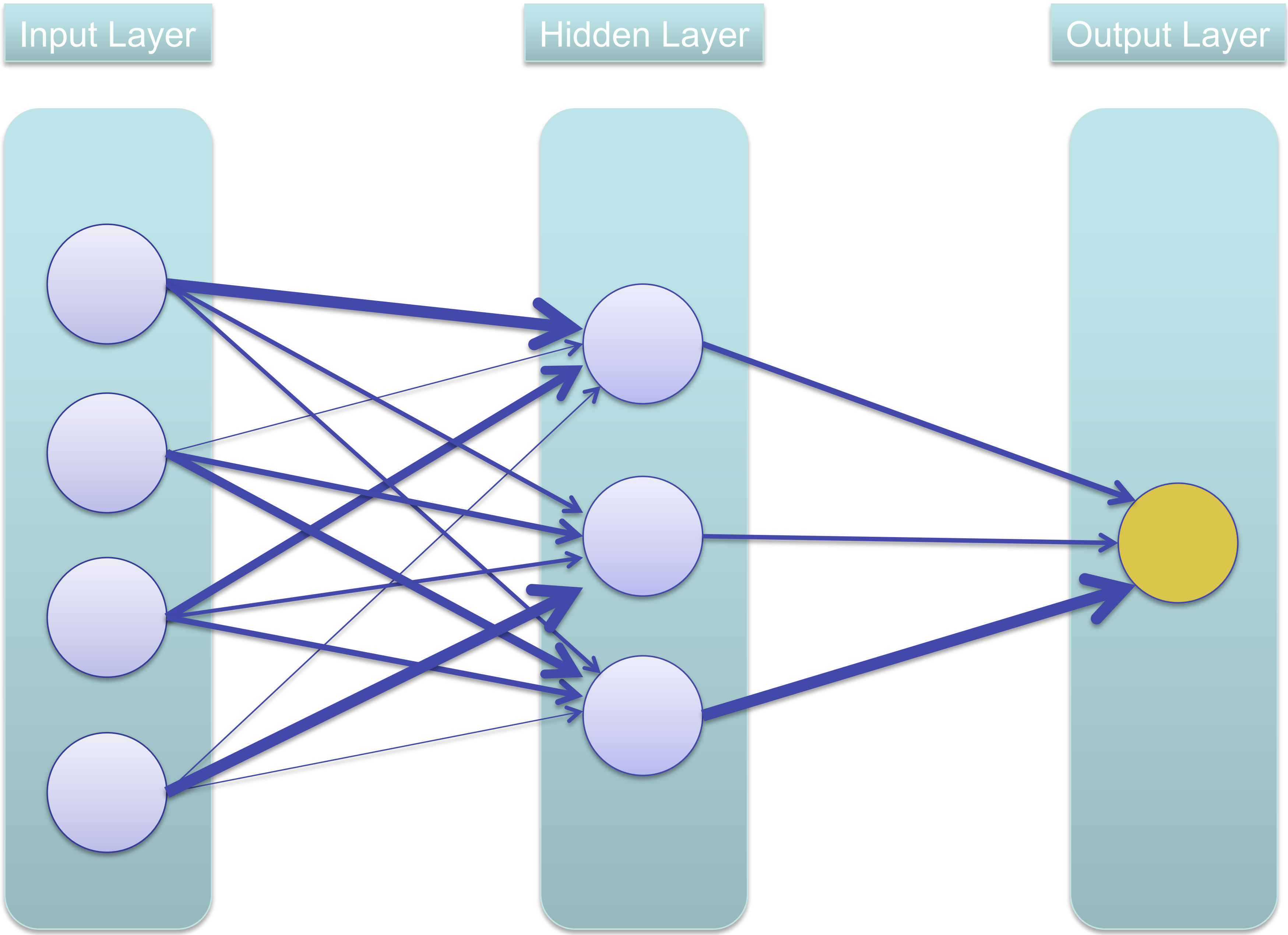

Calculated Output
Desired Output

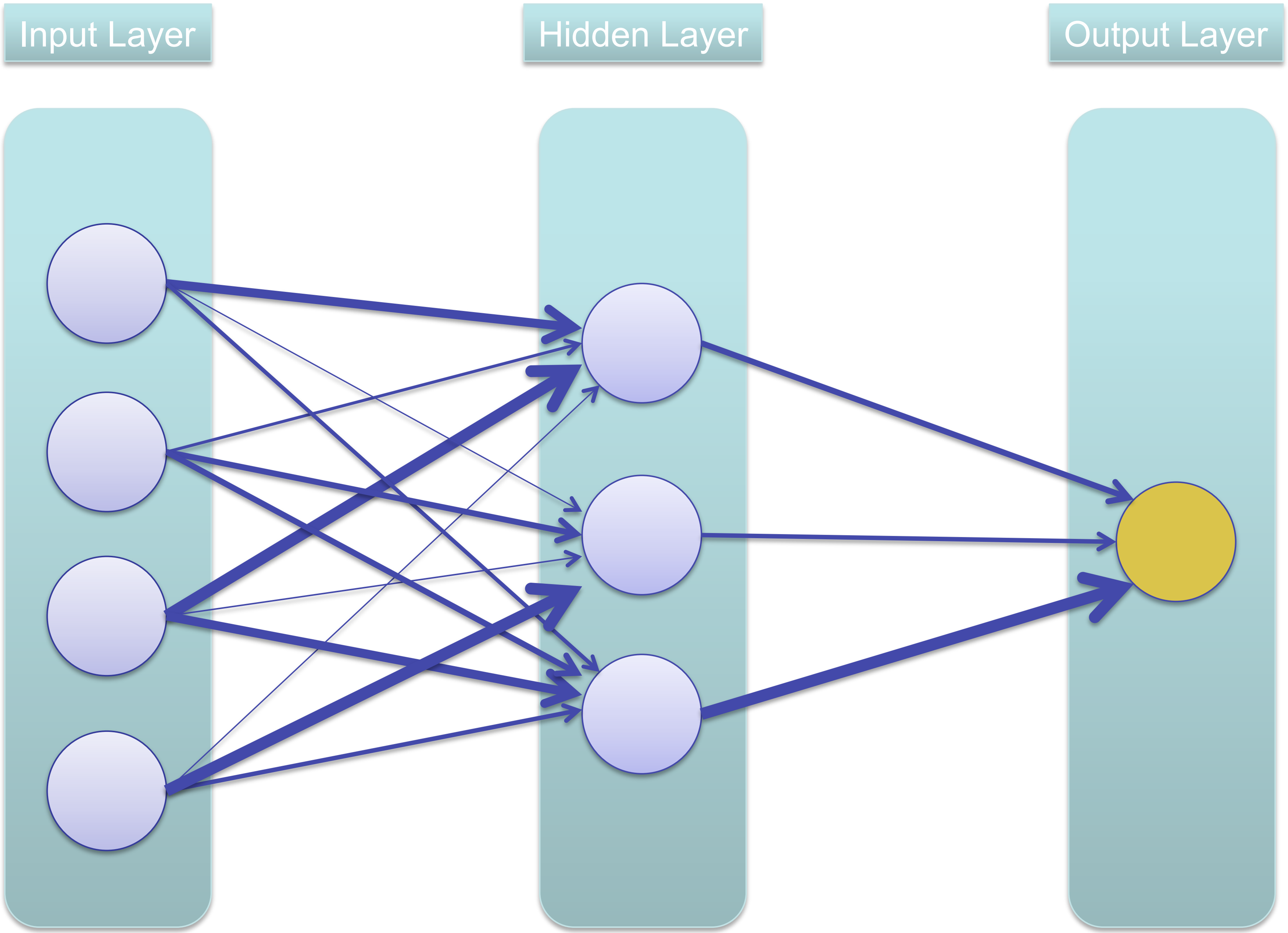


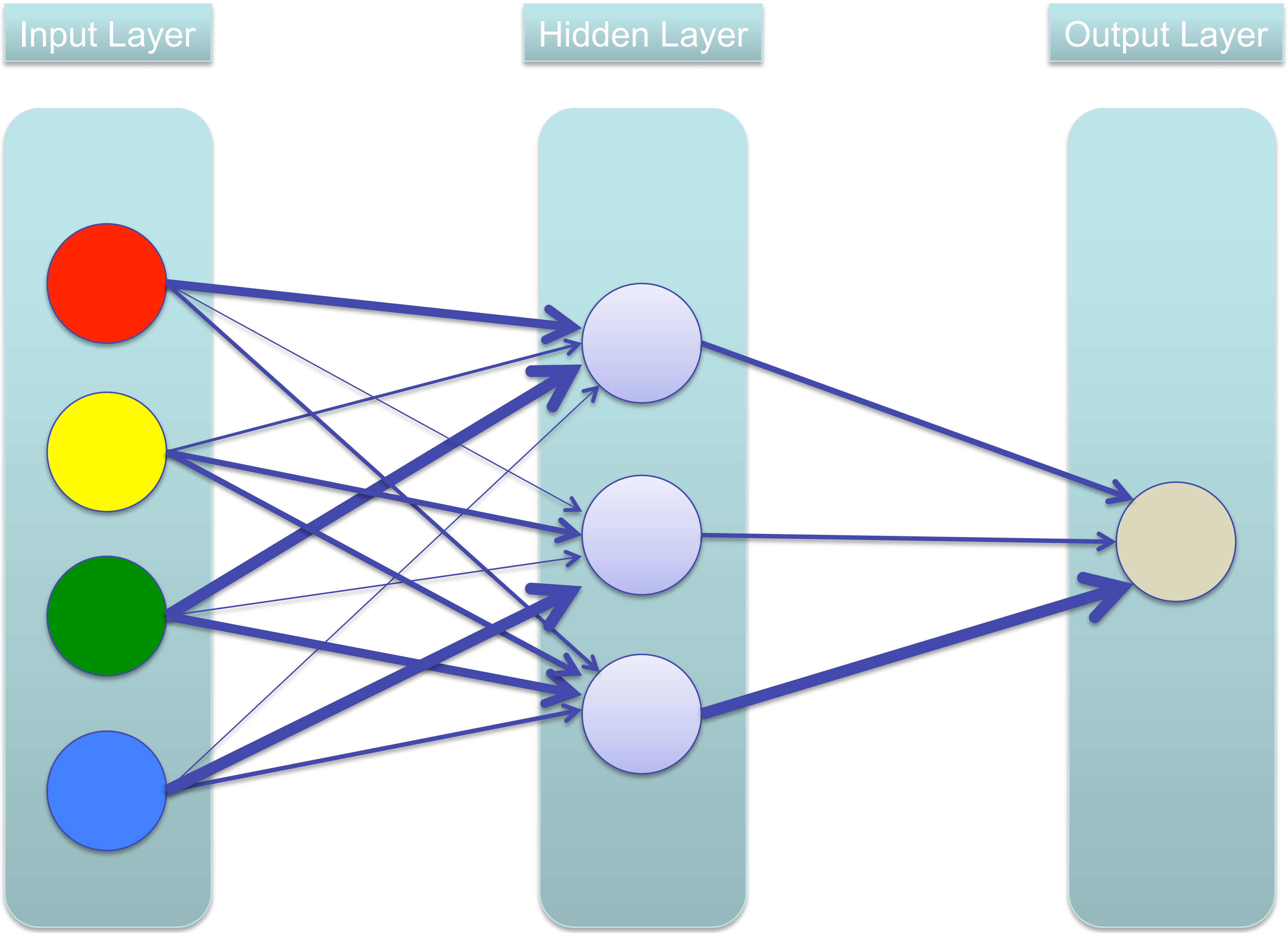
Then we apply backpropagation

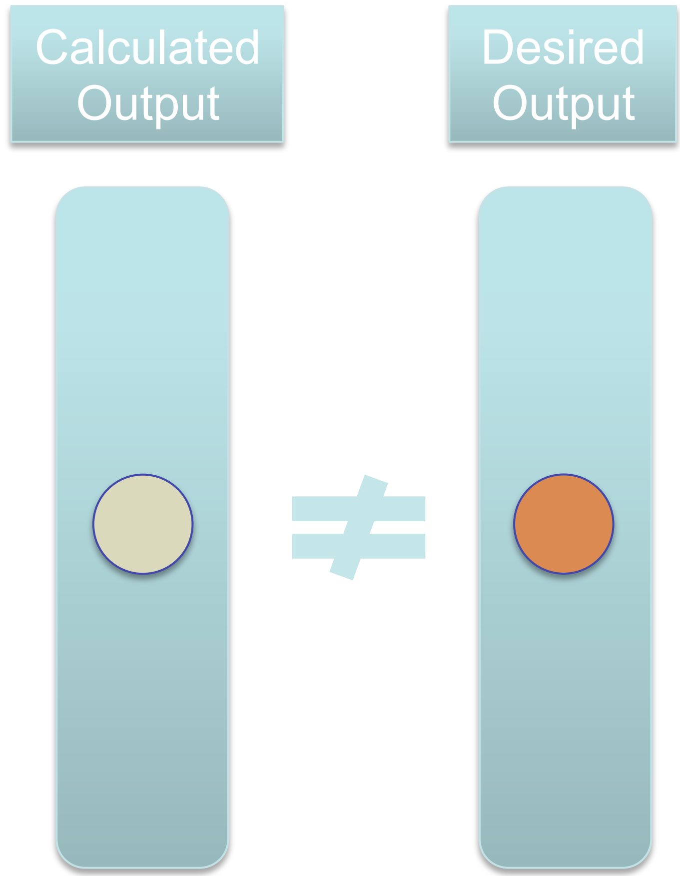

Input Layer
Hidden Layer
Output Layer

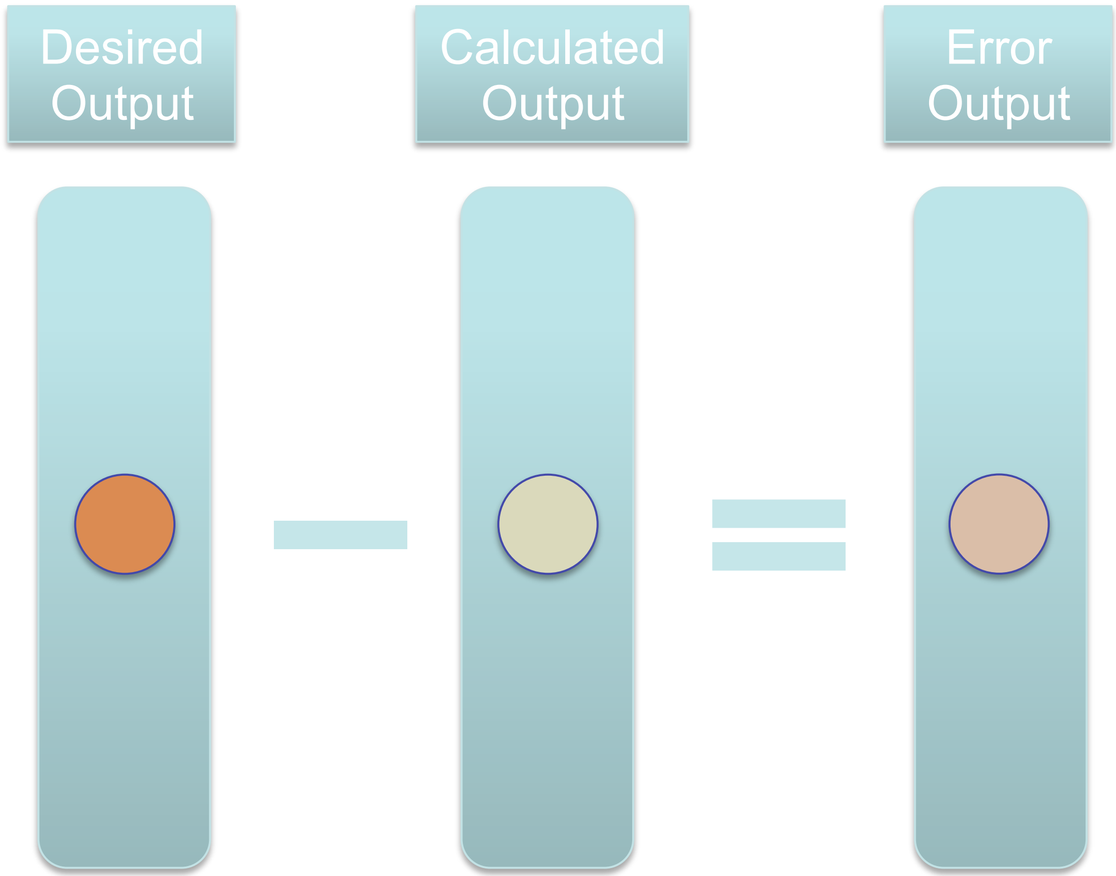

Input Layer
Hidden Layer
Output Layer

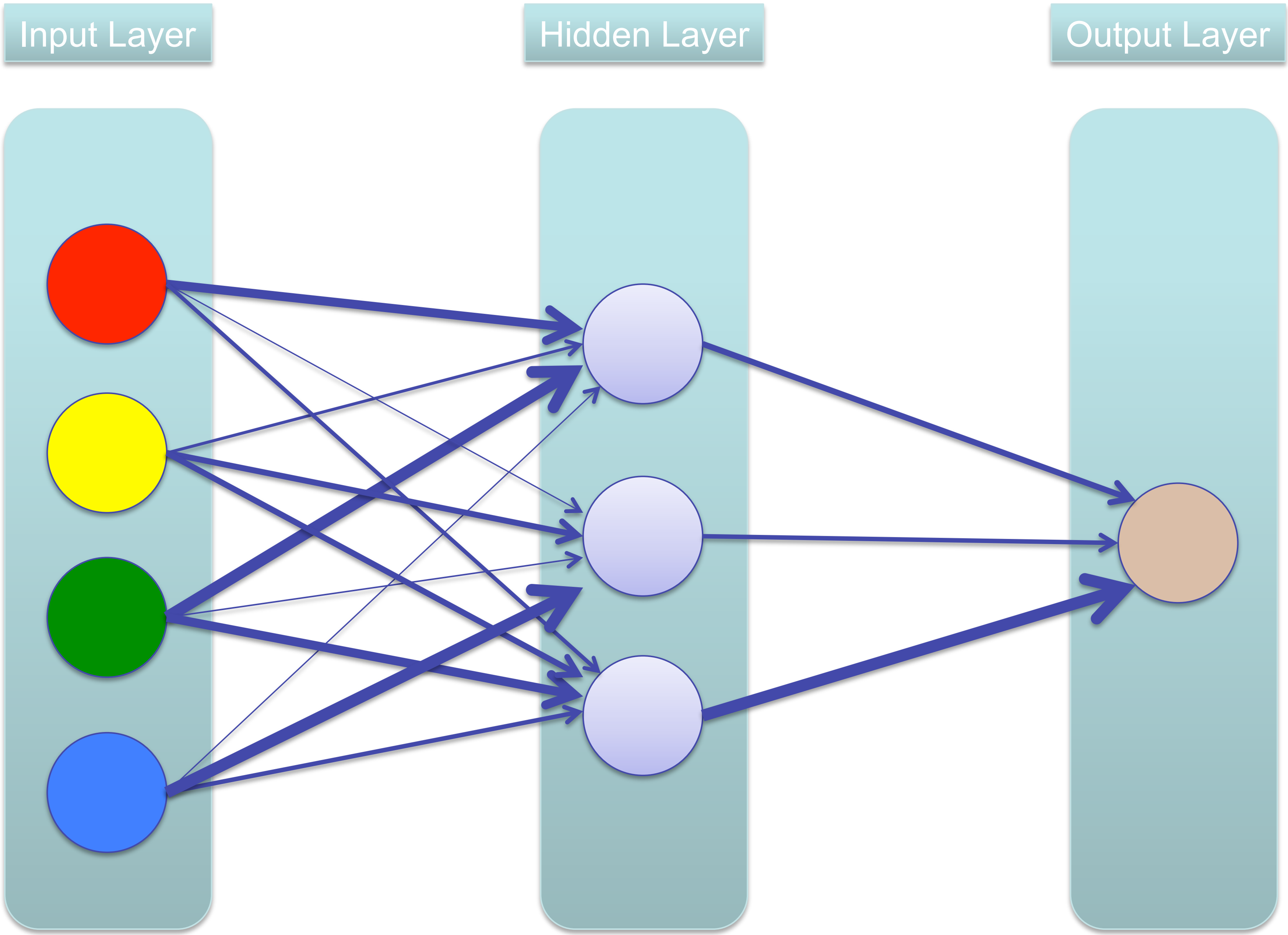

Input Layer
Hidden Layer
Output Layer

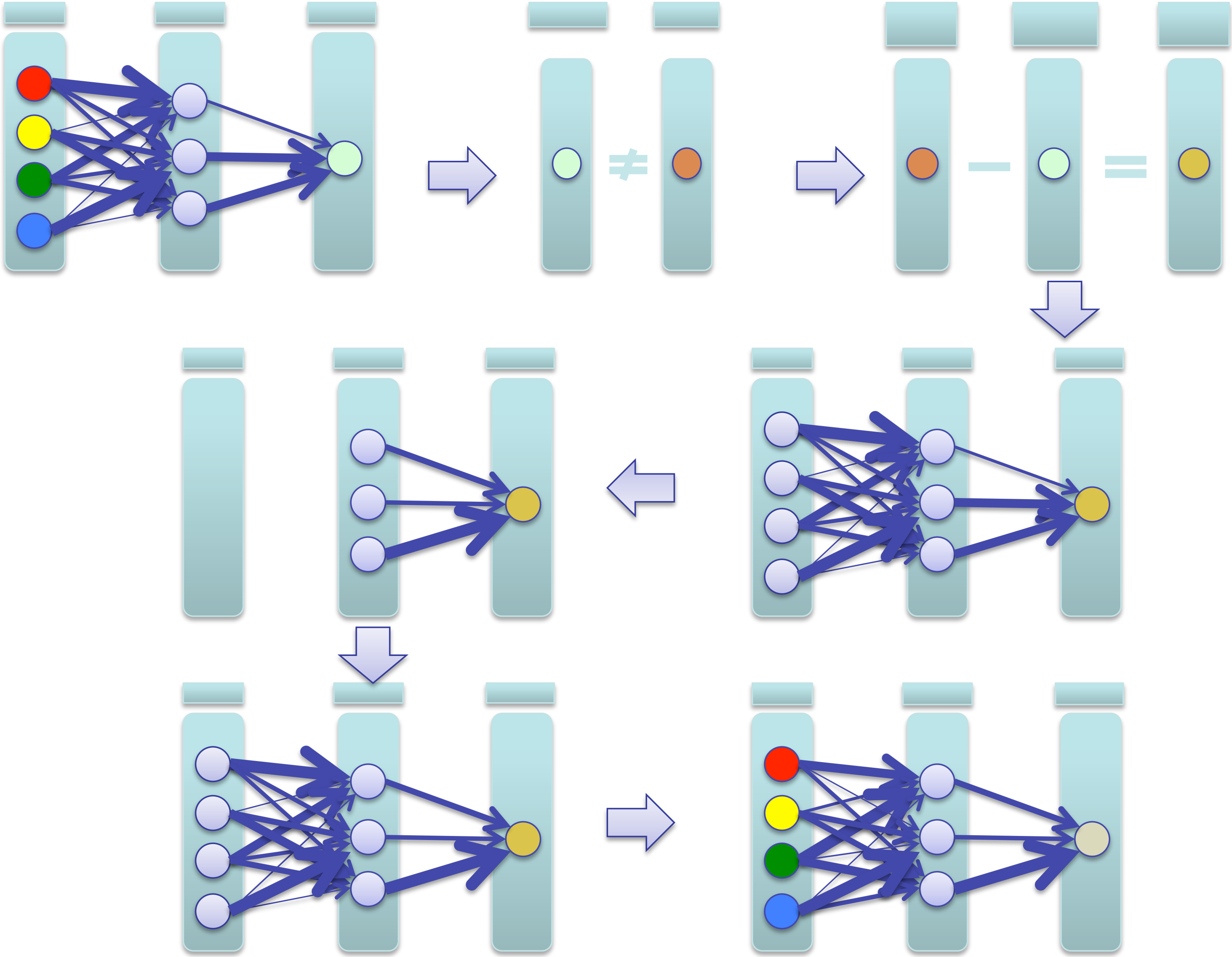

Input Layer
Hidden Layer
Output Layer

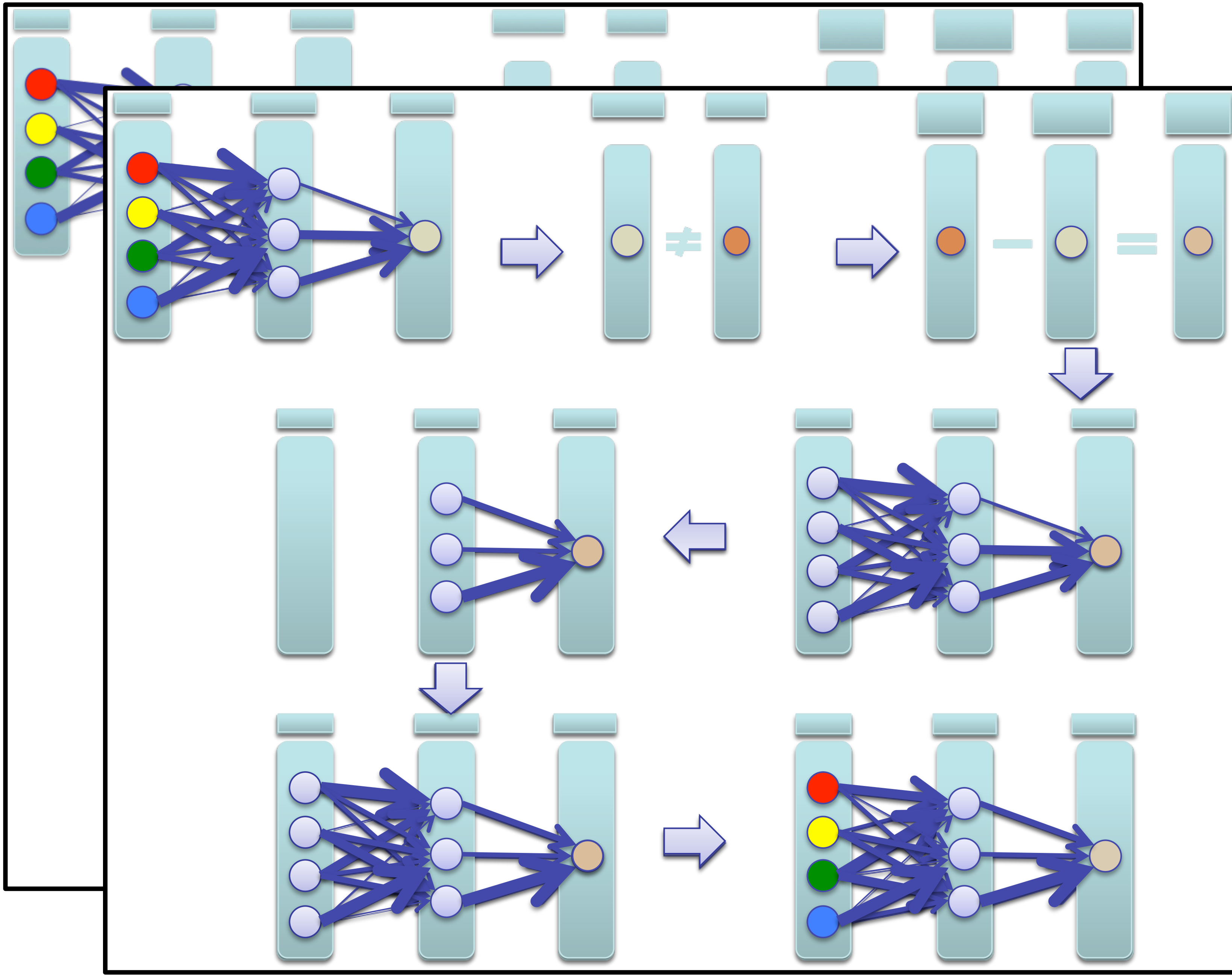

Input Layer
Hidden Layer
Output Layer

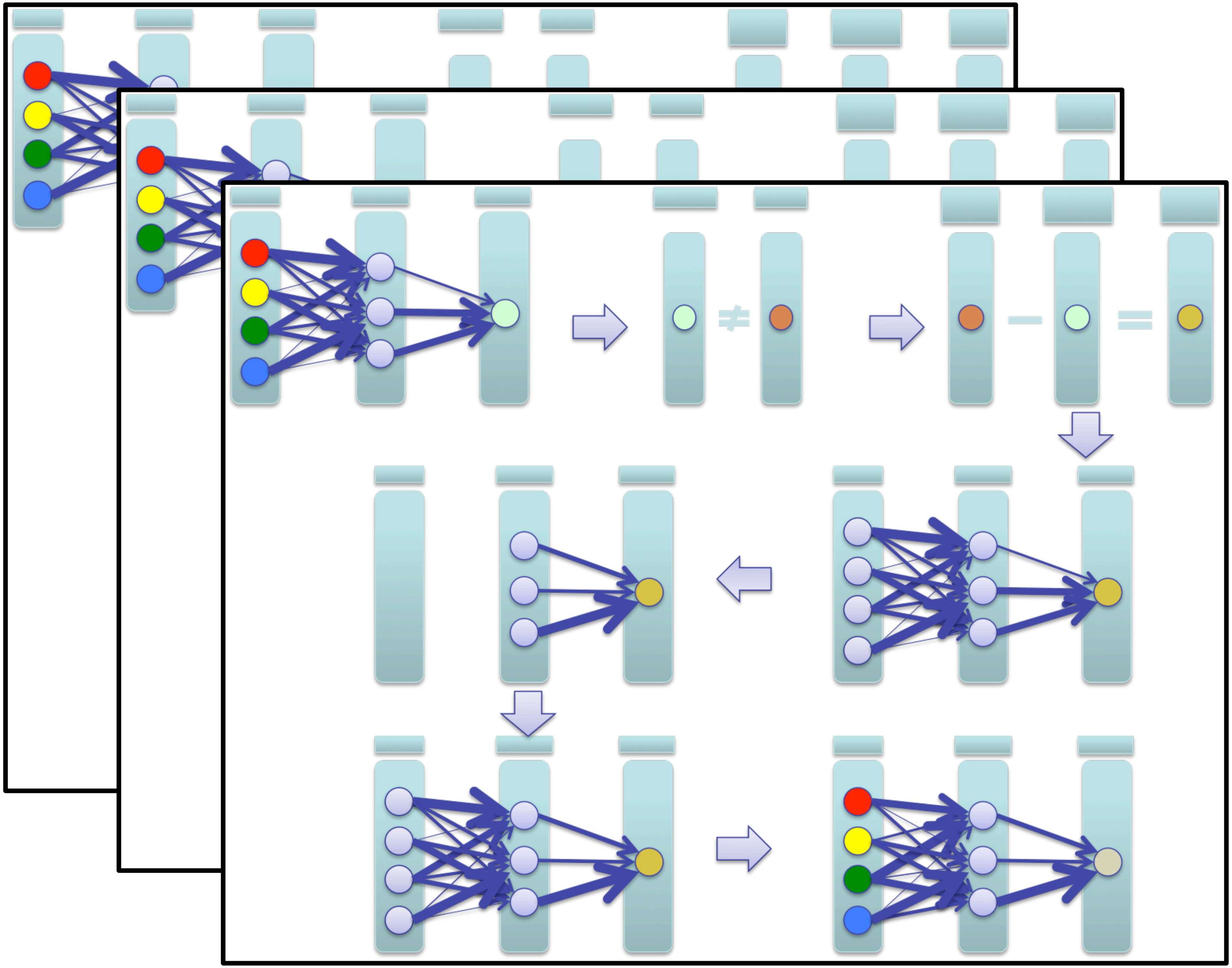

Calculated Output
Desired Output
≠

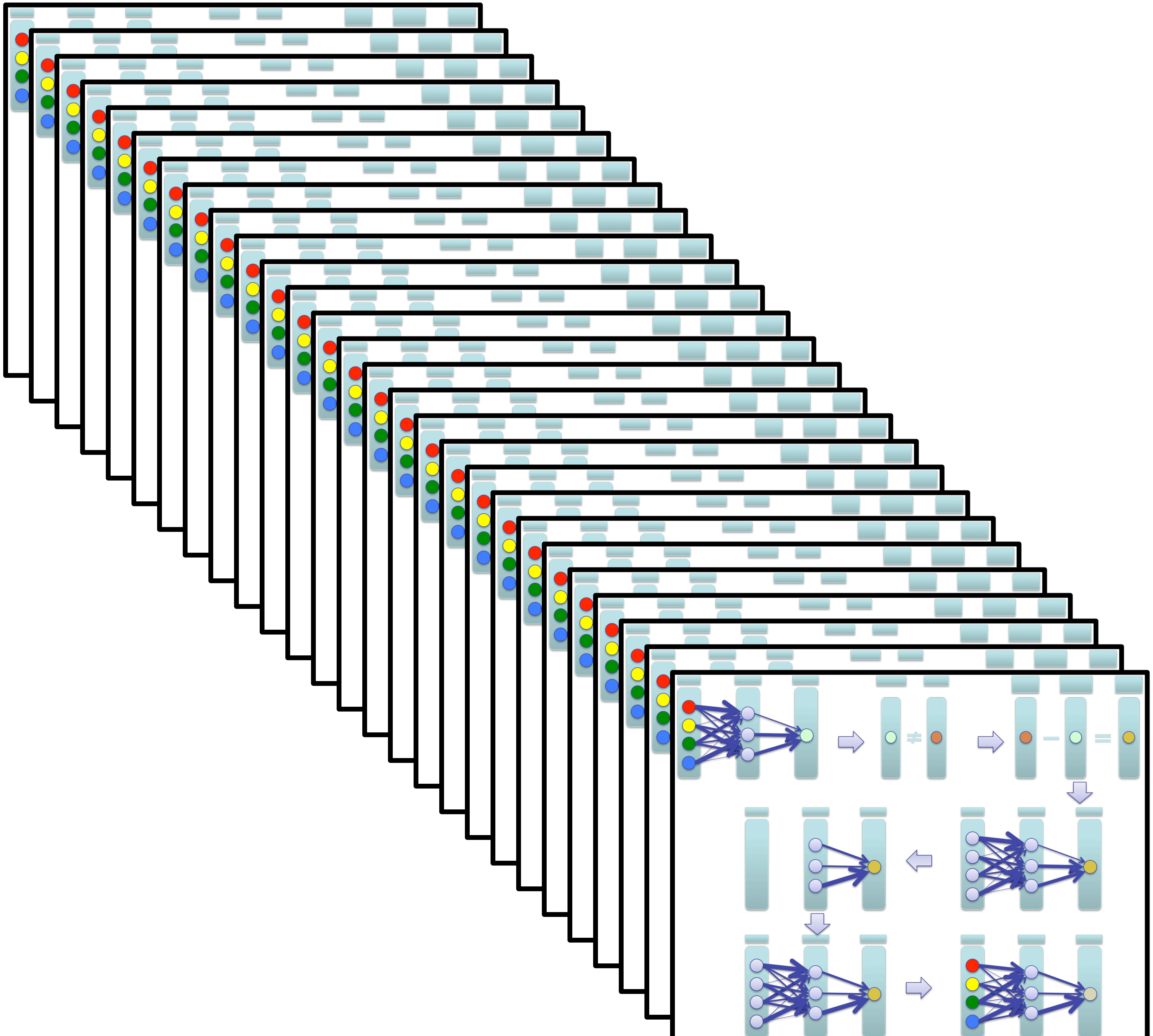

Desired Output
Calculated Output
Error Output

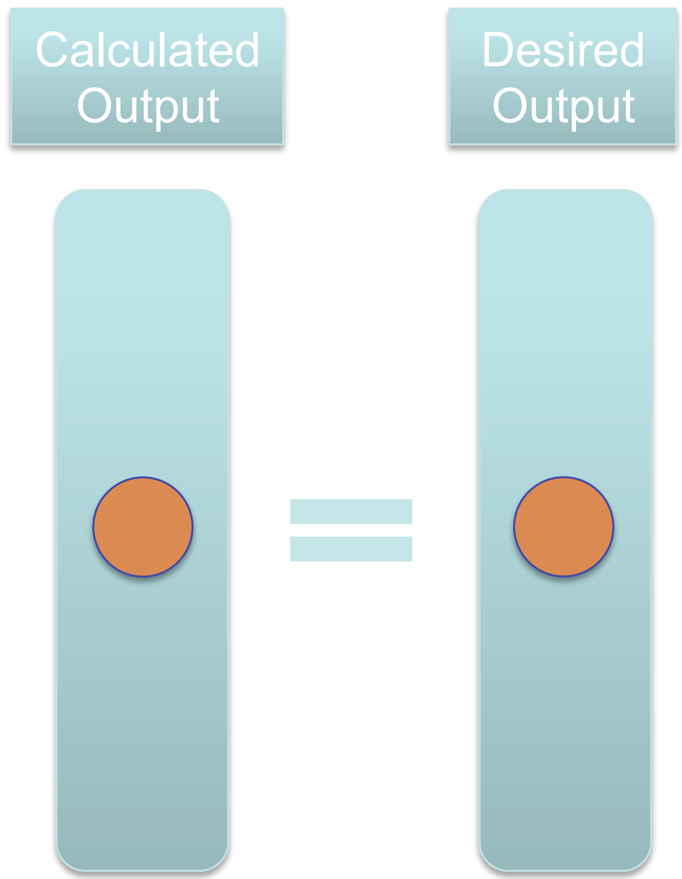

Input Layer
Hidden Layer
Output Layer



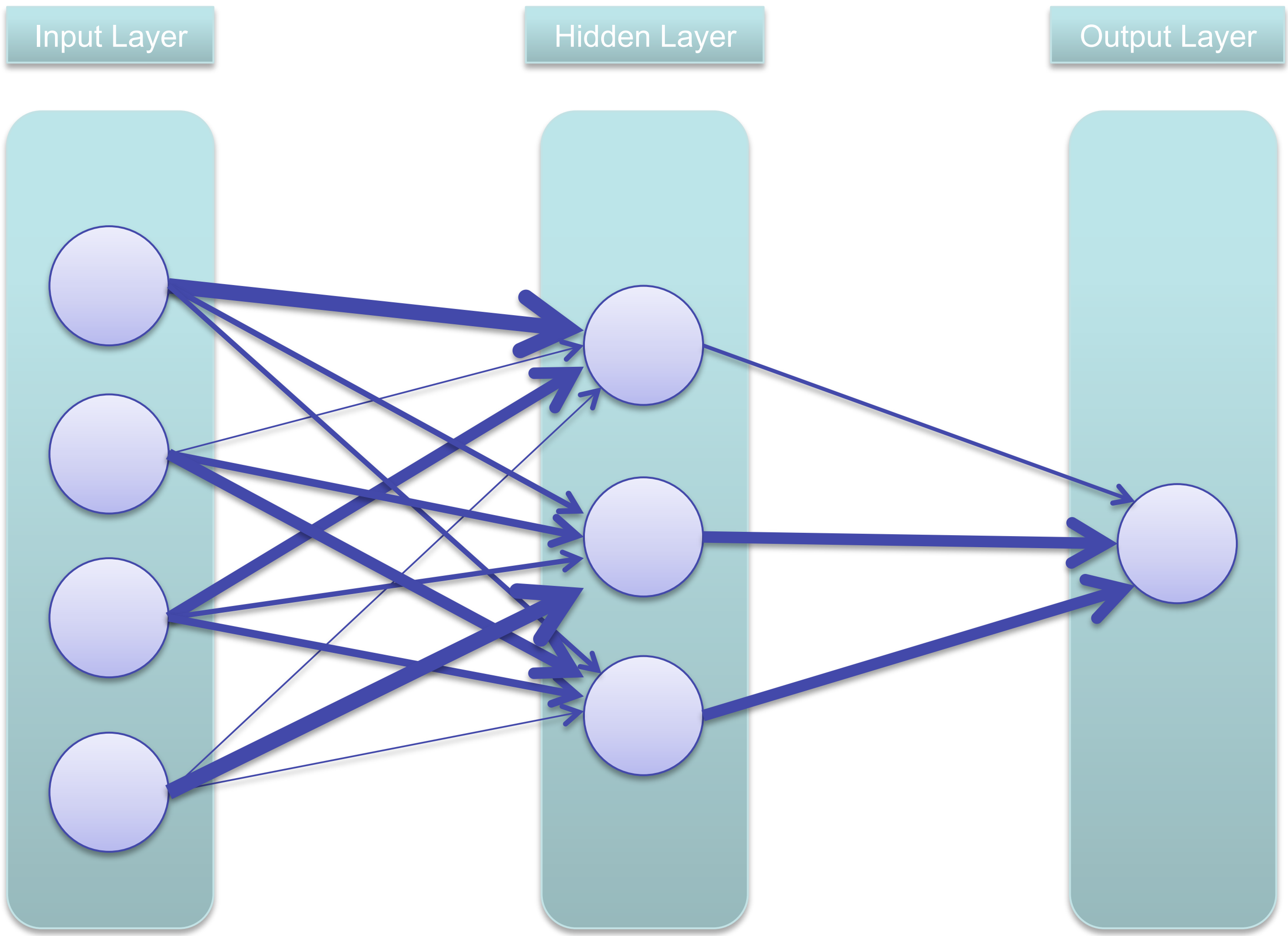







Calculated Output
Desired Output

All weights are good and the network has trained itself



PART 3

Real Life Applications

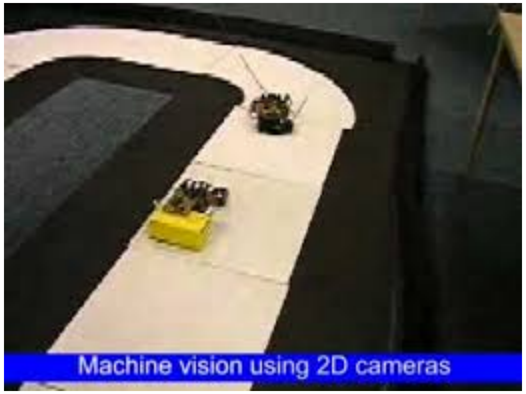
REAL LIFE APPLICATIONS?

- Predictive models
- Artificial intelligence
- Robotics
- Etc.

- VIC: Vision guided Intelligent Car

  Input: 2D camera + human driving observation

  Output: Correct driving technique



BUSINESS INTELLIGENCE

- We set the conditions, such that the neural network will learn from the data in such a way that our desired output is achieved.

- What we control:
  - input layer
  - output layer
  - amount of hidden layers
  - amount of nodes within hidden layer

- What we do not control:
  - whatever happens in the hidden layer

- Therefore: the inner workings of an artificial neural network in data mining, are difficult, if not impossible to interpret.

NEURAL NETWORKS IN BUSINESS INTELLIGENCE

- Benefits:
  - advanced pattern recognition
  - relatively accurate predictions

- Drawbacks:
  - difficult to interpret the inner workings
  - the bigger the network, the longer it takes to compute
  - sensitivity to overfitting

PART 4

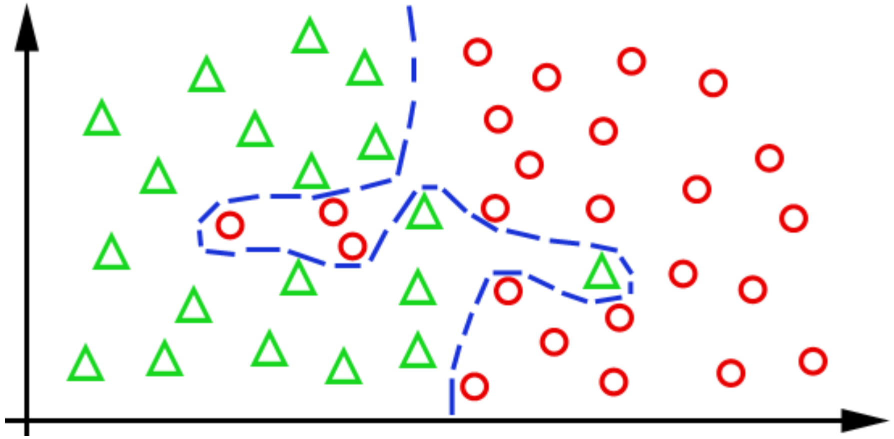
Data Preparation

DATA PREPARATION

A weakness of the neural network is that it can easily overfit the data.

Overfitting has a negative effect on the predictive capabilities of the neural networks because it will predict the noise in the data instead of the overall patterns.

It is important to limit the number of training epochs and not to overtrain the model.



Data cleaning and dimensionality Reduction can also decrease the chance of overfitting.

DATA PREPARATION

1. Cleaning data – removing outliers

2. Normalisation

3. Dimension reduction
    - Principal Component Analysis
    - Correlation Analysis

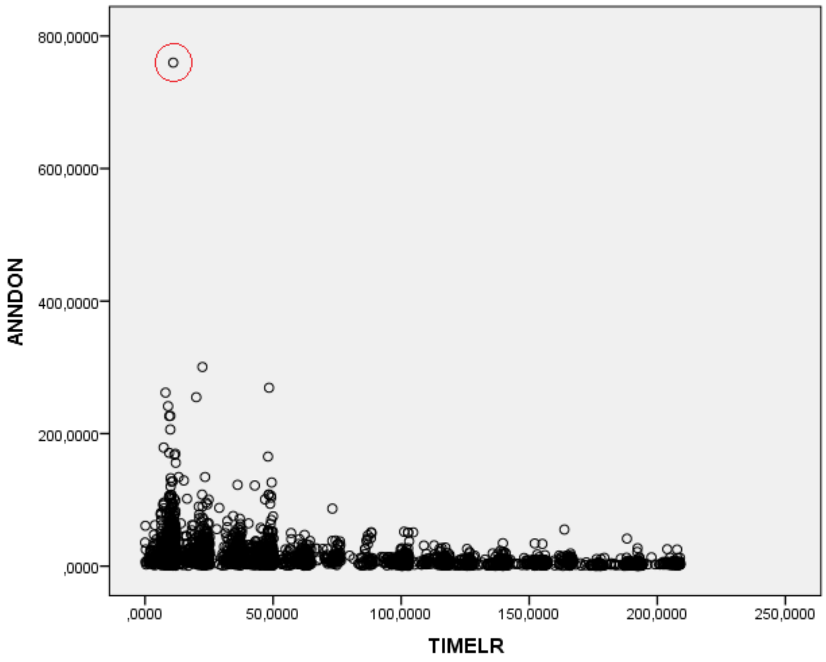
4. Rescaling

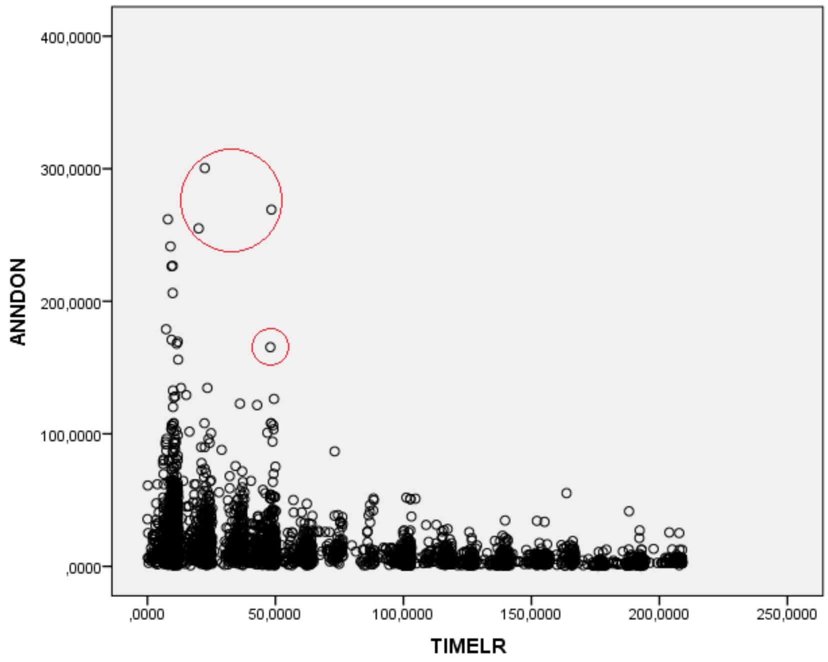
DATA CLEANING

Removing outliers

1-dimensional through sorting

2-dimensional through scatter plot





NORMALISATION

Because the variables do not have the same scale, normalization is required.

For instance:

- TIMELR: [0,0 – 209,0]
- FRQRES: [0,0 – 1,0]

$$\frac{X - \mu}{\sigma}$$

The normalized values of the variables are computed by subtracting the mean and deviding the result by the standard deviation.

DIMENSION REDUCTION

Principal Component Analysis

- Helps us to summarize the different dimension of the data in factors

- The original data is a linear combination of the factors

- Some correlation between variables in de dataset is required for PCA

PRINCIPAL COMPONENT ANALYSIS

1	2,963	42,323	42,323	2,963	42,323	42,323
2	1,392	19,890	62,212	1,392	19,890	62,212
3	1,121	16,021	78,233	1,121	16,021	78,233
4	,955	13,640	91,873	,955	13,640	91,873
5	,321	4,590	96,463	,321	4,590	96,463
6	,164	2,345	98,809	,164	2,345	98,809
7	,083	1,191	100,000	,083	1,191	100,000

Extraction Method: Principal Component Analysis.

Component Matrix[a]

	Component						
	1	2	3	4	5	6	7
zAVGDON	,929	,204	,074	,139	,148	-,020	-,221
zANNDON	,884	,269	-,214	-,064	,157	-,221	,149
zFRQRES	,720	-,607	-,099	,028	,188	,249	,072
zTIMELR	-,685	,583	-,014	,128	,403	,105	,023
zLSTDON	,567	,719	,210	,112	-,256	,187	,065
zTIMECL	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

PRINCIPAL COMPONENT ANALYSIS

1	2,963	42,323	42,323	2,963	42,323	42,323
2	1,392	19,890	62,212	1,392	19,890	62,212
3	1,121	16,021	78,233	1,121	16,021	78,233
4	,955	13,640	91,873	,955	13,640	91,873
5	,321	4,590	96,463	,321	4,590	96,463
6	,164	2,345	98,809	,164	2,345	98,809
7	,083	1,191	100,000	,083	1,191	100,000

Extraction Method: Principal Component Analysis.

Component Matrix[a]

	Component						
	1	2	3	4	5	6	7
zAVGDON	,929	,204	,074	,139	,148	-,020	-,221
zANNDON	,884	,269	-,214	-,064	,157	-,221	,149
zFRQRES	,720	-,607	-,099	,028	,188	,249	,072
zTIMELR	-,685	,583	-,014	,128	,403	,105	,023
zLSTDON	,567	,719	,210	,112	-,256	,187	,065
zTIMECL	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

PRINCIPAL COMPONENT ANALYSIS

1	2,963	42,323	42,323	2,963	42,323	42,323
2	1,392	19,890	62,212	1,392	19,890	62,212
3	1,121	16,021	78,233	1,121	16,021	78,233
4	,955	13,640	91,873	,955	13,640	91,873
5	,321	4,590	96,463	,321	4,590	96,463
6	,164	2,345	98,809	,164	2,345	98,809
7	,083	1,191	100,000	,083	1,191	100,000

Extraction Method: Principal Component Analysis.

Component Matrix[a]

	Component						
	1	2	3	4	5	6	7
zAVGDON	,929	,204	,074	,139	,148	-,020	-,221
zANNDON	,884	,269	-,214	-,064	,157	-,221	,149
zFRQRES	,720	-,607	-,099	,028	,188	,249	,072
zTIMELR	-,685	,583	-,014	,128	,403	,105	,023
zLSTDON	,567	,719	,210	,112	-,256	,187	,065
zTIMECL	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

PRINCIPAL COMPONENT ANALYSIS

1	2,963	42,323	42,323	2,963	42,323	42,323
2	1,392	19,890	62,212	1,392	19,890	62,212
3	1,121	16,021	78,233	1,121	16,021	78,233
4	,955	13,640	91,873	,955	13,640	91,873
5	,321	4,590	96,463	,321	4,590	96,463
6	,164	2,345	98,809	,164	2,345	98,809
7	,083	1,191	100,000	,083	1,191	100,000

Extraction Method: Principal Component Analysis.

Component Matrix[a]

	Component						
	1	2	3	4	5	6	7
zAVGDON	,929	,204	,074	,139	,148	-,020	-,221
zANNDON	,884	,269	-,214	-,064	,157	-,221	,149
zFRQRES	,720	-,607	-,099	,028	,188	,249	,072
zTIMELR	-,685	,583	-,014	,128	,403	,105	,023
zLSTDON	,567	,719	,210	,112	-,256	,187	,065
zTIMECL	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

PRINCIPAL COMPONENT ANALYSIS

1	2,963	42,323	42,323	2,963	42,323	42,323
2	1,392	19,890	62,212	1,392	19,890	62,212
3	1,121	16,021	78,233	1,121	16,021	78,233
4	,955	13,640	91,873	,955	13,640	91,873
5	,321	4,590	96,463	,321	4,590	96,463
6	,164	2,345	98,809	,164	2,345	98,809
7	,083	1,191	100,000	,083	1,191	100,000

Extraction Method: Principal Component Analysis.

Component Matrix[a]

	Component						
	1	2	3	4	5	6	7
zAVGDON	,929	,204	,074	,139	,148	-,020	-,221
zANNDON	,884	,269	-,214	-,064	,157	-,221	,149
zFRQRES	,720	-,607	-,099	,028	,188	,249	,072
zTIMELR	-,685	,583	-,014	,128	,403	,105	,023
zLSTDON	,567	,719	,210	,112	-,256	,187	,065
zTIMECL	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

PRINCIPAL COMPONENT ANALYSIS

By trial and error, we discovered that the Neural Network has the highest predictive accuracy with all seven factors.

Thus, we continued our with seven initial variables.

(Not the factors!)

CORRELATION ANALYSIS

zTIMELR	Pearson Correlation	1	-,064**	-,739**	-,115**	-,448**	-,039*	-,410**
	Sig. (2-tailed)		,000	,000	,000	,000	,013	,000
	N	4025	4025	4025	4025	4025	4025	4025
zTIMECL	Pearson Correlation	-,064**	1	,073**	,065**	,094**	,044**	-,225**
	Sig. (2-tailed)	,000		,000	,000	,000	,005	,000
	N	4025	4025	4025	4025	4025	4025	4025
zFRQRES	Pearson Correlation	-,739**	,073**	1	-,017	,549**	-,043**	,478**
	Sig. (2-tailed)	,000	,000		,287	,000	,007	,000
	N	4025	4025	4025	4025	4025	4025	4025
zMEDTOR	Pearson Correlation	-,115**	,065**	-,017	1	,043**	,097**	,023
	Sig. (2-tailed)	,000	,000	,287		,006	,000	,142
	N	4025	4025	4025	4025	4025	4025	4025
zAVGDON	Pearson Correlation	-,448**	,094**	,549**	,043**	1	,648**	,847**
	Sig. (2-tailed)	,000	,000	,000	,006		,000	,000
	N	4025	4025	4025	4025	4025	4025	4025
zLSTDON	Pearson Correlation	-,039*	,044**	-,043**	,097**	,648**	1	,571**
	Sig. (2-tailed)	,013	,005	,007	,000	,000		,000
	N	4025	4025	4025	4025	4025	4025	4025
zANNDON	Pearson Correlation	-,410**	-,225**	,478**	,023	,847**	,571**	1
	Sig. (2-tailed)	,000	,000	,000	,142	,000	,000	
	N	4025	4025	4025	4025	4025	4025	4025

CORRELATION ANALYSIS

zTIMELR	Pearson Correlation	1	-,064**	-,739**	-,115**	-,448**	-,039*	-,410**
	Sig. (2-tailed)		,000	,000	,000	,000	,013	,000
	N	4025	4025	4025	4025	4025	4025	4025
zTIMECL	Pearson Correlation	-,064**	1	,073**	,065**	,094**	,044**	-,225**
	Sig. (2-tailed)	,000		,000	,000	,000	,005	,000
	N	4025	4025	4025	4025	4025	4025	4025
zFRQRES	Pearson Correlation	-,739**	,073**	1	-,017	,549**	-,043**	,478**
	Sig. (2-tailed)	,000	,000		,287	,000	,007	,000
	N	4025	4025	4025	4025	4025	4025	4025
zMEDTOR	Pearson Correlation	-,115**	,065**	-,017	1	,043**	,097**	,023
	Sig. (2-tailed)	,000	,000	,287		,006	,000	,142
	N	4025	4025	4025	4025	4025	4025	4025
zAVGDON	Pearson Correlation	-,448**	,094**	,549**	,043**	1	,648**	,847**
	Sig. (2-tailed)	,000	,000	,000	,006		,000	,000
	N	4025	4025	4025	4025	4025	4025	4025
zLSTDON	Pearson Correlation	-,039*	,044**	-,043**	,097**	,648**	1	,571**
	Sig. (2-tailed)	,013	,005	,007	,000	,000		,000
	N	4025	4025	4025	4025	4025	4025	4025
zANNDON	Pearson Correlation	-,410**	-,225**	,478**	,023	,847**	,571**	1
	Sig. (2-tailed)	,000	,000	,000	,142	,000	,000	
	N	4025	4025	4025	4025	4025	4025	4025

CORRELATION ANALYSIS

We observe four high correlations:

- zAVGDON → zANNDON
- zTIMELR → zFREQRES
- zAVGDON → zLSTDON
- zFREQRES → zAVGDON

CORRELATION ANALYSIS

By trial and error, we discovered that the neural network produces the highest predictive accuracy when we leave variable zAVGDON out.

This is in line with our expectations because AVGDON contains the the least information about the respondent, in comparison to ANNDON.

RESCALING

The neural networks performs best when all variables are on a scale of [-1, 1].

This is because of the functions that are used in the nodes of the hidden layer.

RESCALING

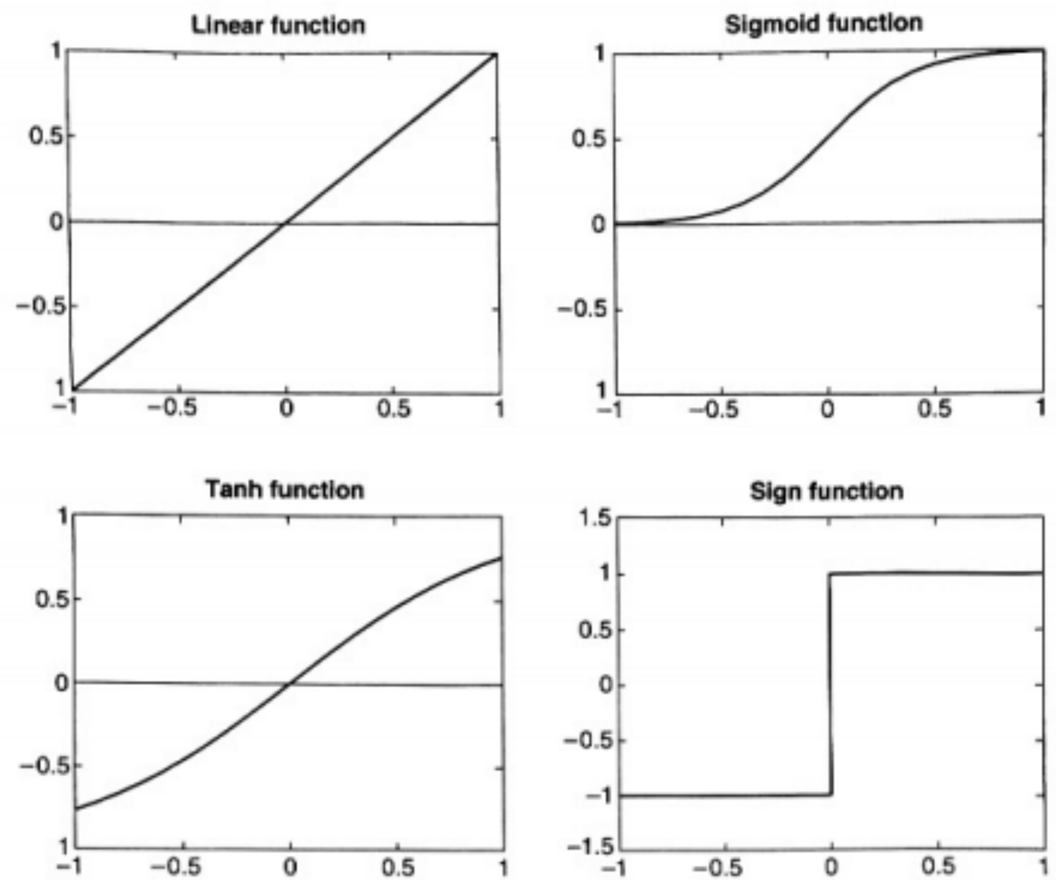


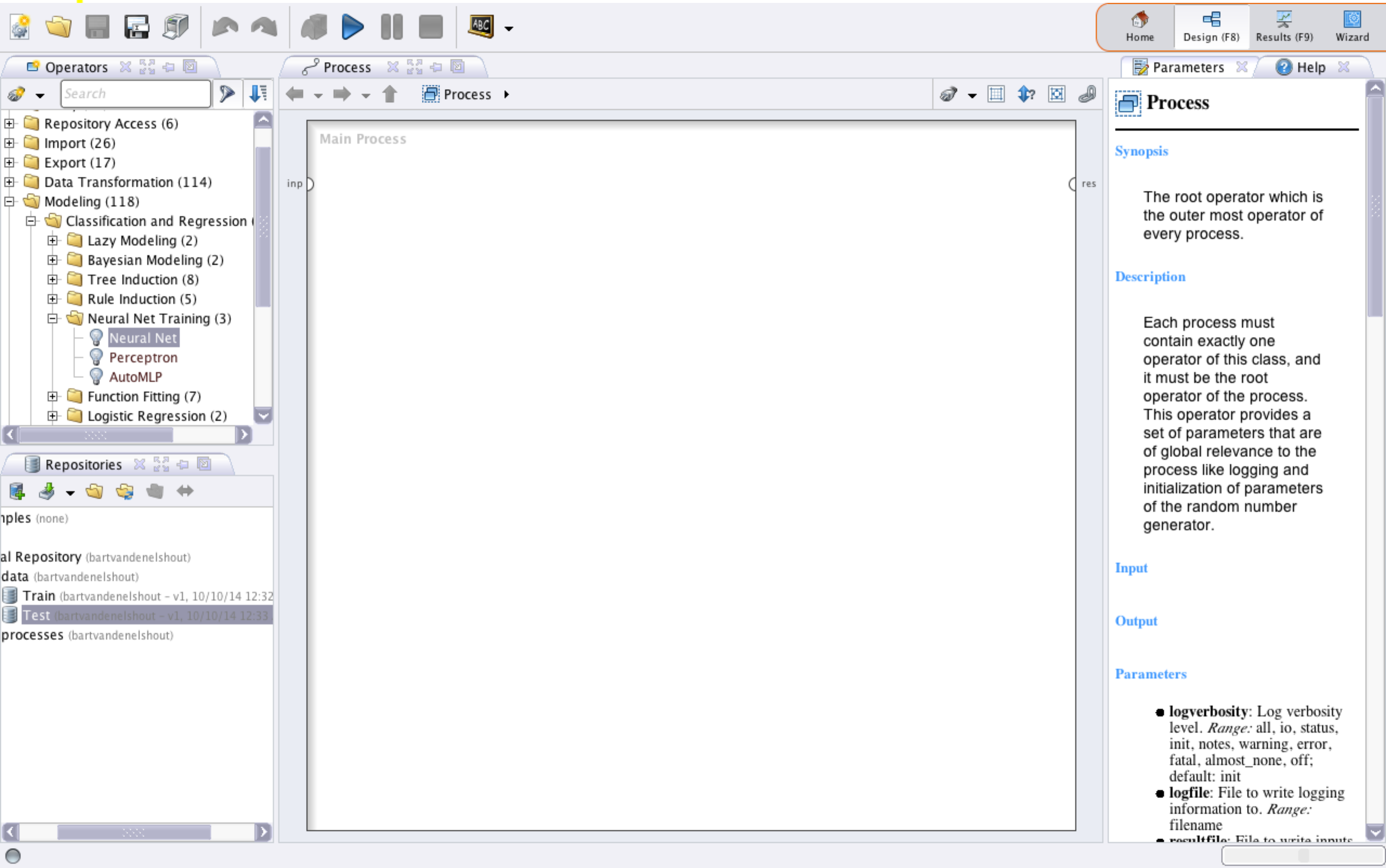
Figure 5.18. Types of activation functions in artificial neural networks.

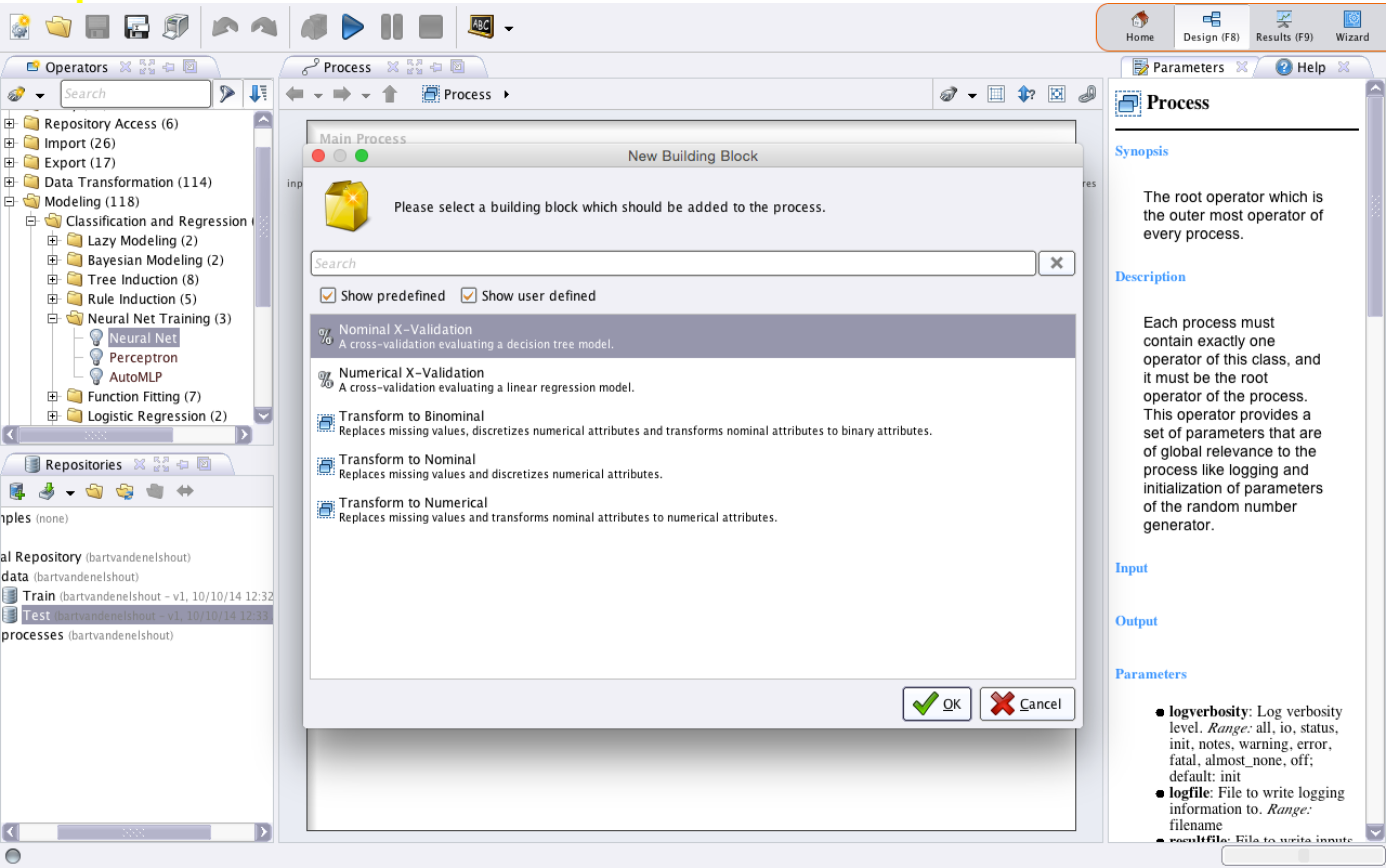
PART 5

Method application & Pay-off analysis

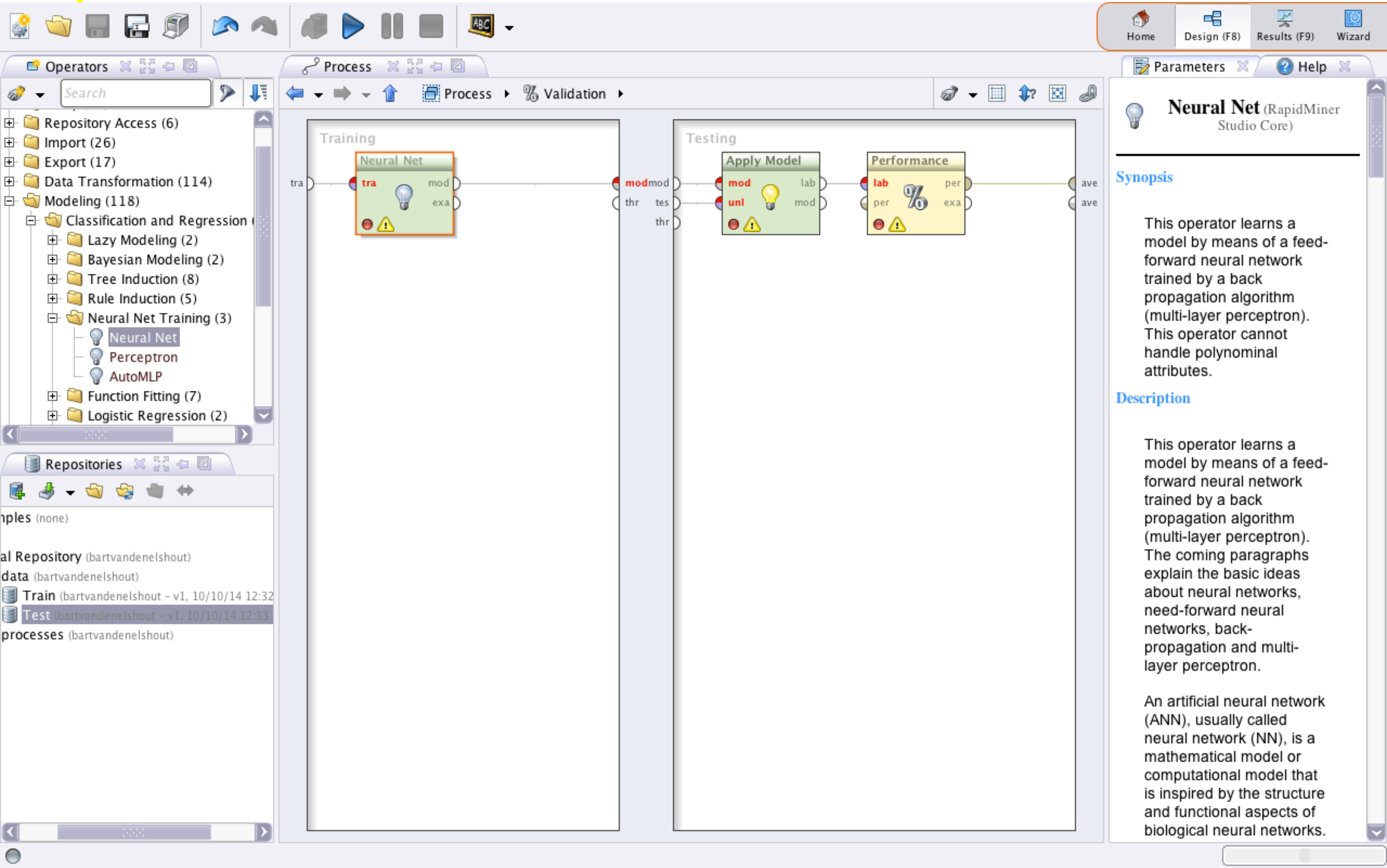
METHOD APPLICATION

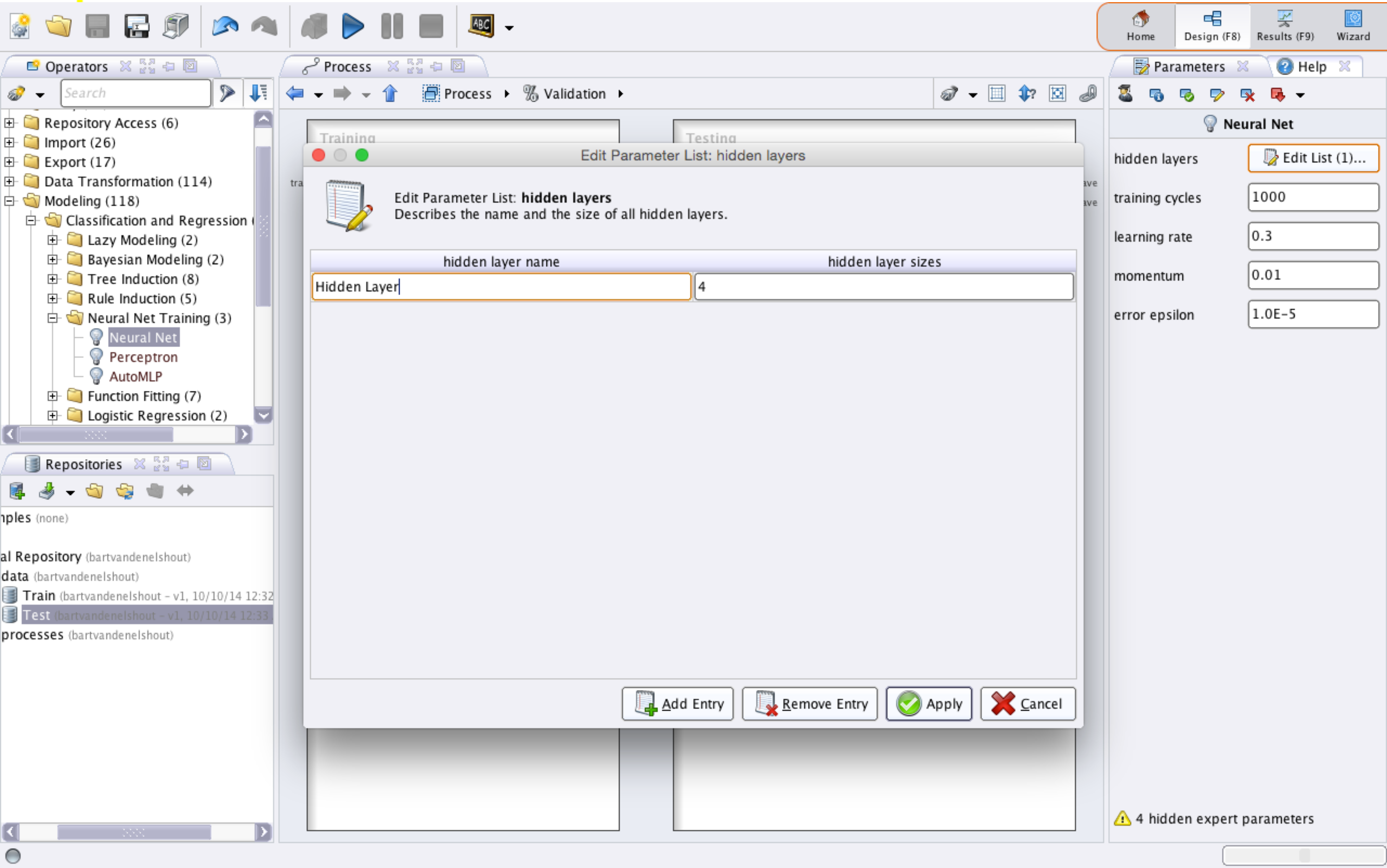
Training the data

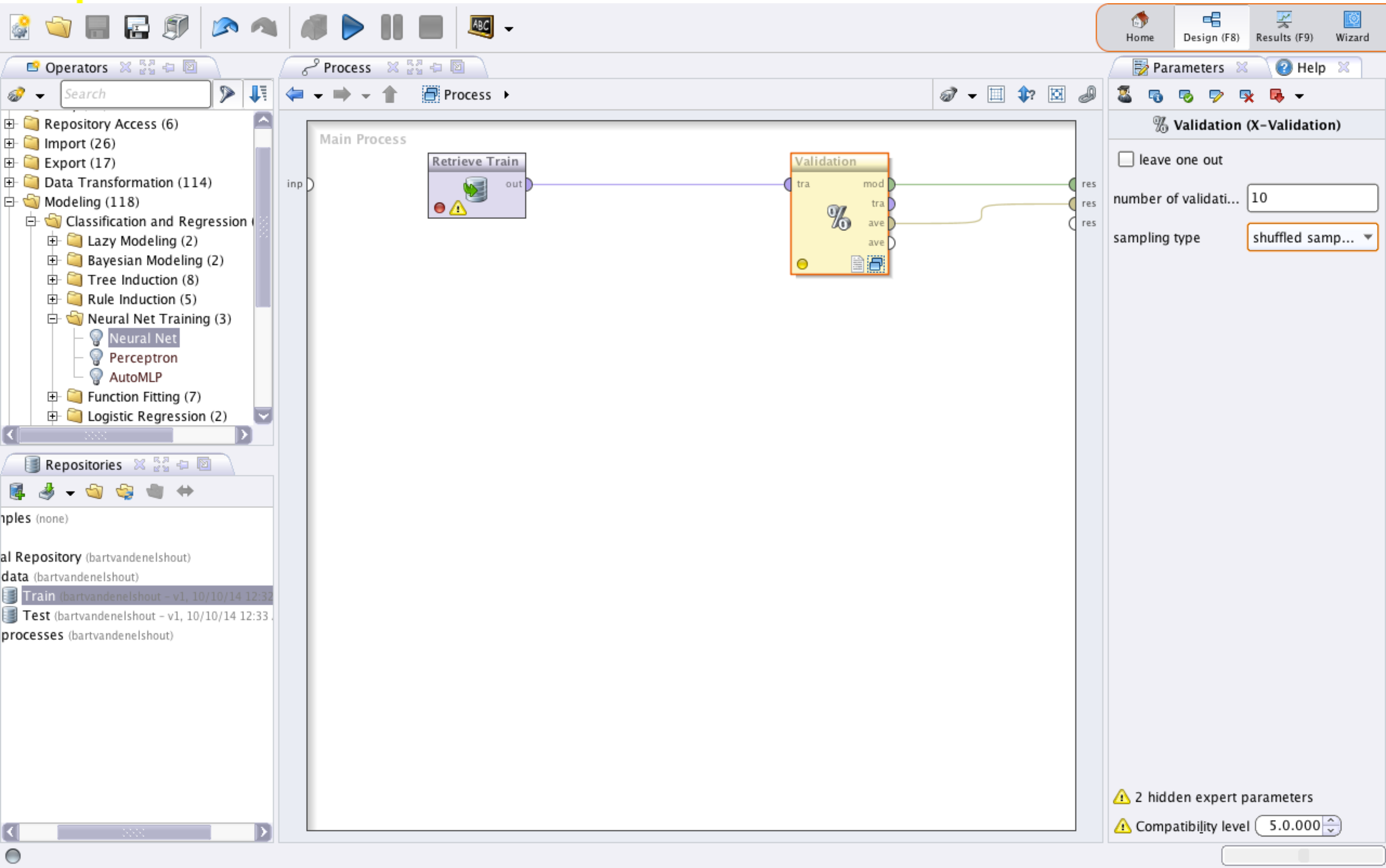

Home
Design (F8)
Results (F9)
Wizard
Operators
Process
Parameters
Help
Repository Access (6)
Import (26)
Export (17)
Data Transformation (114)
Modeling (118)
Classification and Regression
Lazy Modeling (2)
Bayesian Modeling (2)
Tree Induction (8)
Rule Induction (5)
Neural Net Training (3)
Neural Net
Perceptron
AutoMLP
Function Fitting (7)
Logistic Regression (2)
Repositories
Train (bartvandenelshout - v1, 10/10/14 12:32
Test (bartvandenelshout - v1, 10/10/14 12:33
processes (bartvandenelshout)
Main Process
inp
res
Process
Synopsis
The root operator which is the outer most operator of every process.
Description
Each process must contain exactly one operator of this class, and it must be the root operator of the process. This operator provides a set of parameters that are of global relevance to the process like logging and initialization of parameters of the random number generator.
Input
Output
Parameters
logverbosity: Log verbosity level. Range: all, io, status, init, notes, warning, error, fatal, almost_none, off; default: init
logfile: File to write logging information to. Range: filename

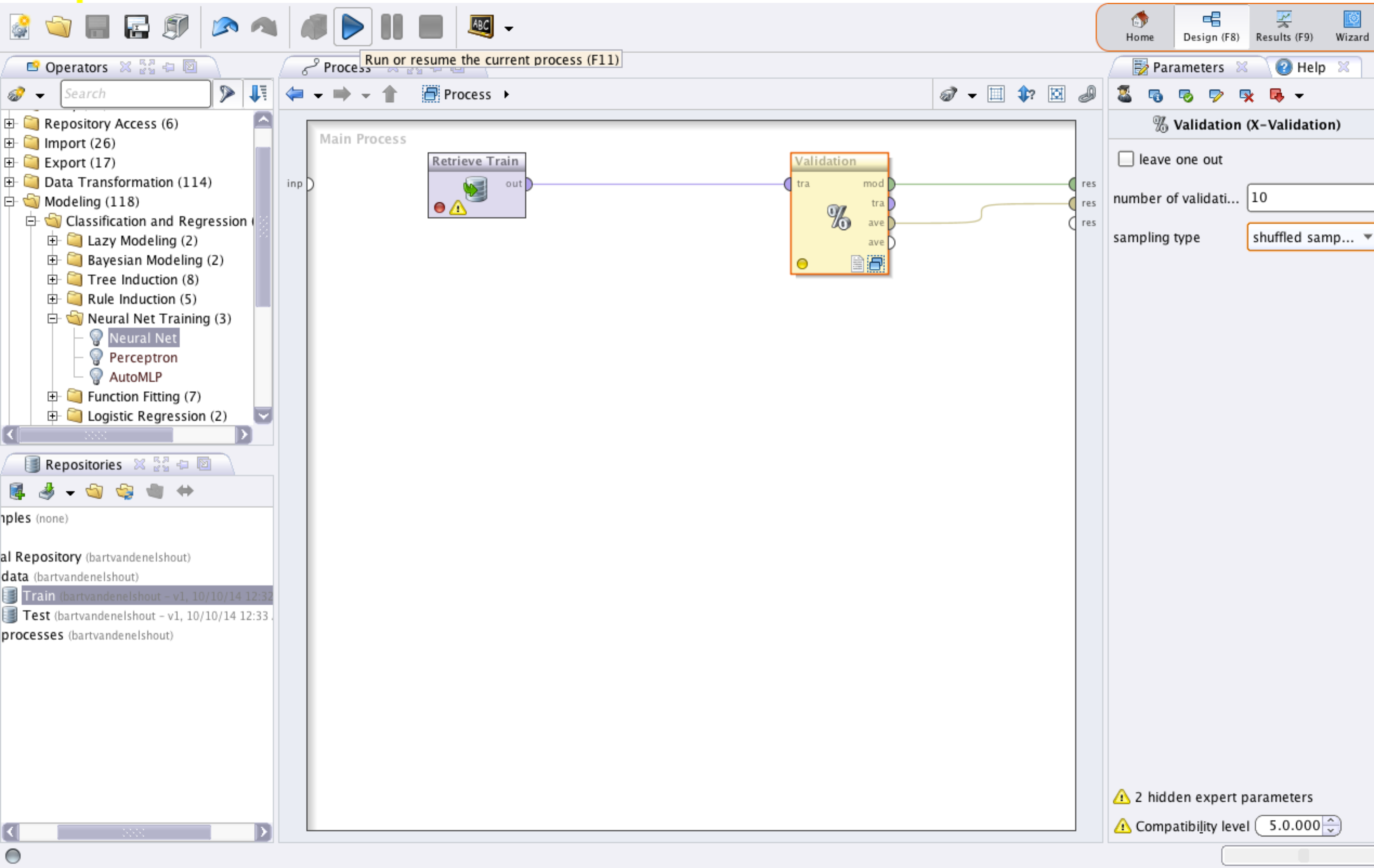


Home
Design (F8)
Results (F9)
Wizard
Operators
Process
Parameters
Help
Repository Access (6)
Import (26)
Export (17)
Data Transformation (114)
Modeling (118)
Classification and Regression
Lazy Modeling (2)
Bayesian Modeling (2)
Tree Induction (8)
Rule Induction (5)
Neural Net Training (3)
Neural Net
Perceptron
AutoMLP
Function Fitting (7)
Logistic Regression (2)
Repositories
Main Process
New Building Block
Please select a building block which should be added to the process.
Search
Show predefined
Show user defined
Nominal X-Validation
A cross-validation evaluating a decision tree model.
Numerical X-Validation
A cross-validation evaluating a linear regression model.
Transform to Binominal
Replaces missing values, discretizes numerical attributes and transforms nominal attributes to binary attributes.
Transform to Nominal
Replaces missing values and discretizes numerical attributes.
Transform to Numerical
Replaces missing values and transforms nominal attributes to numerical attributes.
OK
Cancel
Process
Synopsis
The root operator which is the outer most operator of every process.
Description
Each process must contain exactly one operator of this class, and it must be the root operator of the process. This operator provides a set of parameters that are of global relevance to the process like logging and initialization of parameters of the random number generator.
Input
Output
Parameters
logverbosity: Log verbosity level. Range: all, io, status, init, notes, warning, error, fatal, almost_none, off; default: init
logfile: File to write logging information to. Range: filename


Home
Design (F8)
Results (F9)
Wizard
Operators
Process
Parameters
Help
Search
Process
Validation
Repository Access (6)
Import (26)
Export (17)
Data Transformation (114)
Modeling (118)
Classification and Regression
Lazy Modeling (2)
Bayesian Modeling (2)
Tree Induction (8)
Rule Induction (5)
Neural Net Training (3)
Neural Net
Perceptron
AutoMLP
Function Fitting (7)
Logistic Regression (2)
Repositories
nples (none)
al Repository (bartvandenelshout)
data (bartvandenelshout)
Train (bartvandenelshout - v1, 10/10/14 12:32
Test (bartvandenelshout - v1, 10/10/14 12:33
processes (bartvandenelshout)
Training
Neural Net
tra
mod
exa
Testing
Apply Model
mod
unl
lab
Performance
lab
per
exa
ave
thr
tes
Neural Net (RapidMiner Studio Core)
Synopsis
This operator learns a model by means of a feed-forward neural network trained by a back propagation algorithm (multi-layer perceptron). This operator cannot handle polynominal attributes.
Description
This operator learns a model by means of a feed-forward neural network trained by a back propagation algorithm (multi-layer perceptron). The coming paragraphs explain the basic ideas about neural networks, need-forward neural networks, back-propagation and multi-layer perceptron.
An artificial neural network (ANN), usually called neural network (NN), is a mathematical model or computational model that is inspired by the structure and functional aspects of biological neural networks.

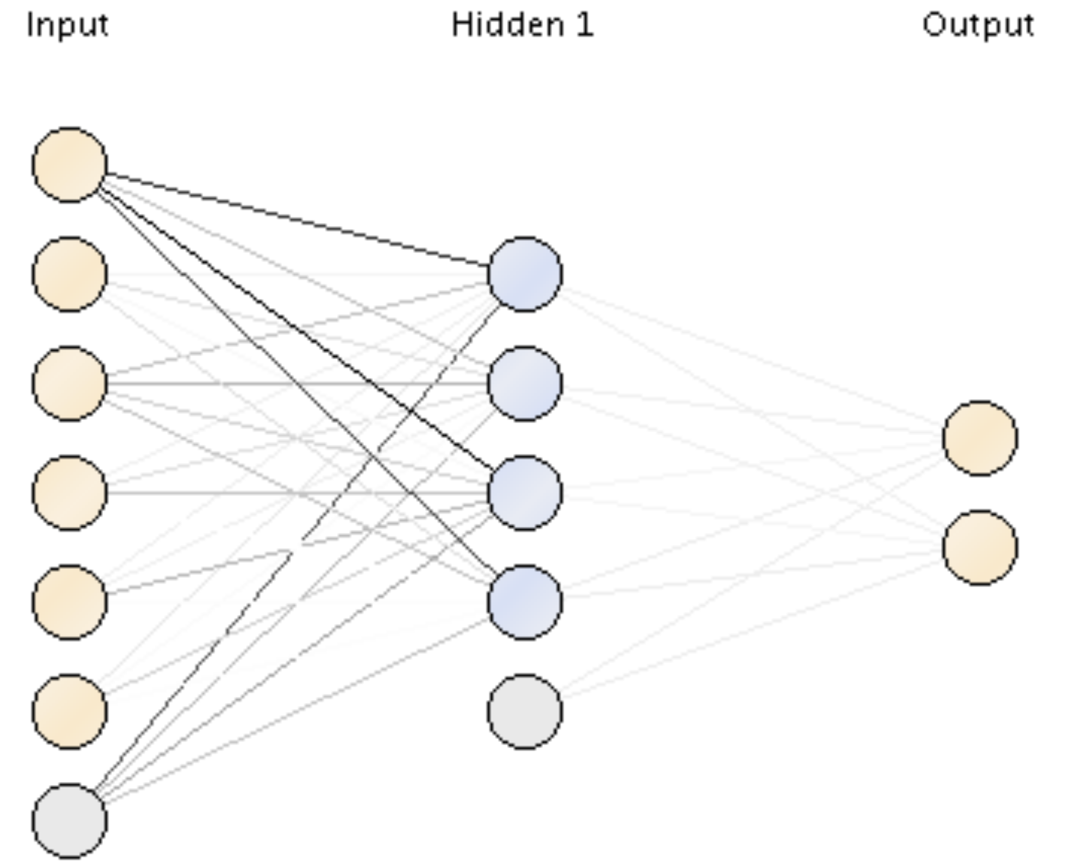

Home
Design (F8)
Results (F9)
Wizard
Operators
Process
Parameters
Help
Search
Process
Validation
Repository Access (6)
Import (26)
Export (17)
Data Transformation (114)
Modeling (118)
Classification and Regression
Lazy Modeling (2)
Bayesian Modeling (2)
Tree Induction (8)
Rule Induction (5)
Neural Net Training (3)
Neural Net
Perceptron
AutoMLP
Function Fitting (7)
Logistic Regression (2)
Repositories
Training
Testing
Edit Parameter List: hidden layers
Describes the name and the size of all hidden layers.
hidden layer name
hidden layer sizes
Hidden Layer
4
Add Entry
Remove Entry
Apply
Cancel
Neural Net
hidden layers
Edit List (1)...
training cycles
1000
learning rate
0.3
momentum
0.01
error epsilon
1.0E-5
4 hidden expert parameters


Home
Design (F8)
Results (F9)
Wizard
Operators
Process
Parameters
Help
Search
Repository Access (6)
Import (26)
Export (17)
Data Transformation (114)
Modeling (118)
Classification and Regression
Lazy Modeling (2)
Bayesian Modeling (2)
Tree Induction (8)
Rule Induction (5)
Neural Net Training (3)
Neural Net
Perceptron
AutoMLP
Function Fitting (7)
Logistic Regression (2)
Repositories
Main Process
Retrieve Train
Validation
inp
out
tra
mod
ave
res
Validation (X-Validation)
leave one out
number of validati...
10
sampling type
shuffled samp...
2 hidden expert parameters
Compatibility level
5.0.000
al Repository (bartvandenelshout)
data (bartvandenelshout)
Train (bartvandenelshout – v1, 10/10/14 12:32
Test (bartvandenelshout – v1, 10/10/14 12:33
processes (bartvandenelshout)


Home
Design (F8)
Results (F9)
Wizard
Operators
Search
Repository Access (6)
Import (26)
Export (17)
Data Transformation (114)
Modeling (118)
Classification and Regression
Lazy Modeling (2)
Bayesian Modeling (2)
Tree Induction (8)
Rule Induction (5)
Neural Net Training (3)
Neural Net
Perceptron
AutoMLP
Function Fitting (7)
Logistic Regression (2)
Repositories
nples (none)
al Repository (bartvandenelshout)
data (bartvandenelshout)
Train (bartvandenelshout – v1, 10/10/14 12:32
Test (bartvandenelshout – v1, 10/10/14 12:33
processes (bartvandenelshout)
Process
Run or resume the current process (F11)
Main Process
Retrieve Train
out
inp
Validation
tra
mod
tra
ave
ave
res
res
res
Parameters
Help
Validation (X-Validation)
leave one out
number of validati...
10
sampling type
shuffled samp...
2 hidden expert parameters
Compatibility level
5.0.000

accuracy: 74.99% +/− 2.15% (mikro: 74.99%)

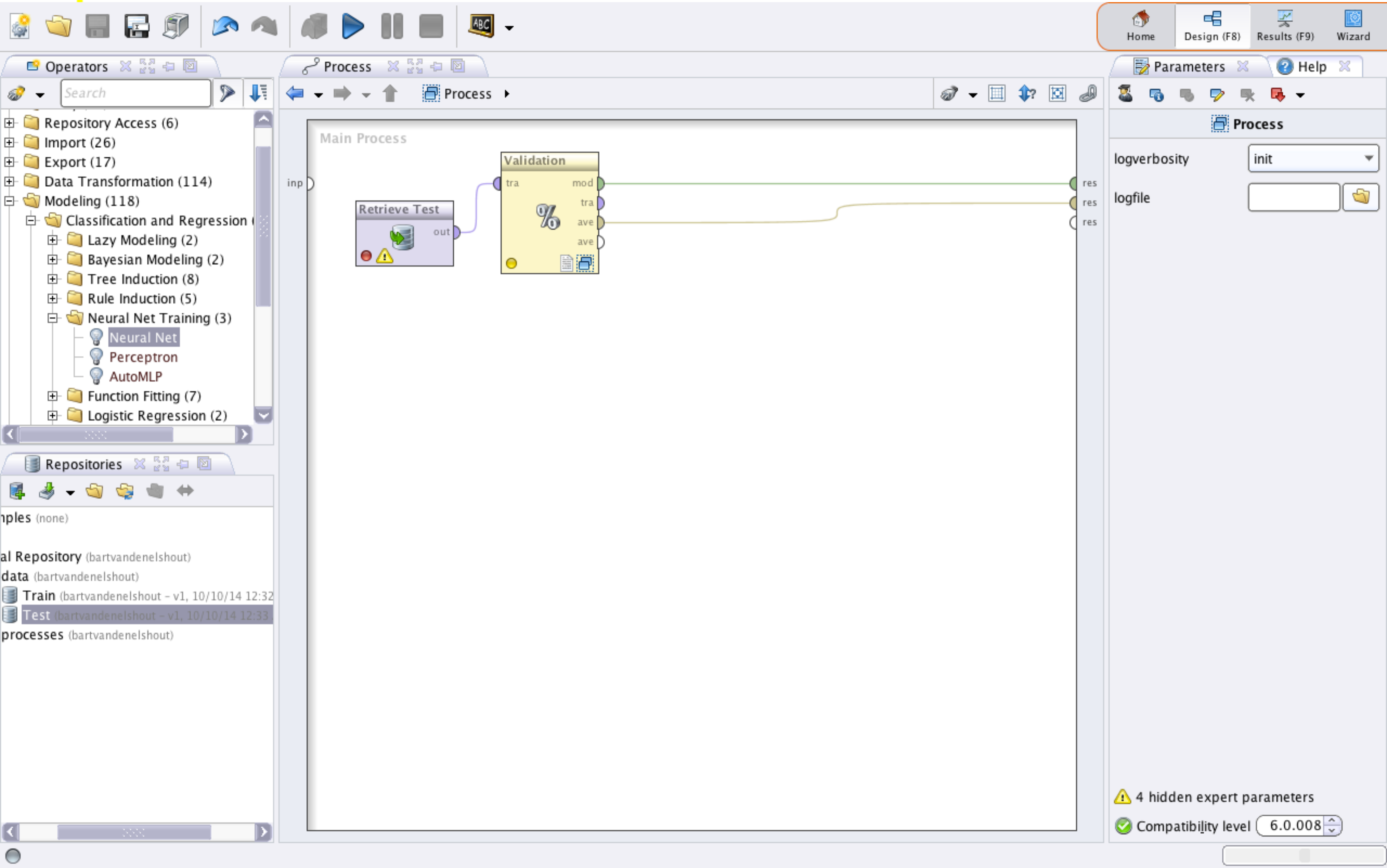
	true 0.0	true 1.0	class precision
pred. 0.0	1981	466	80.96%
pred. 1.0	466	814	63.59%
class recall	80.96%	63.59%	

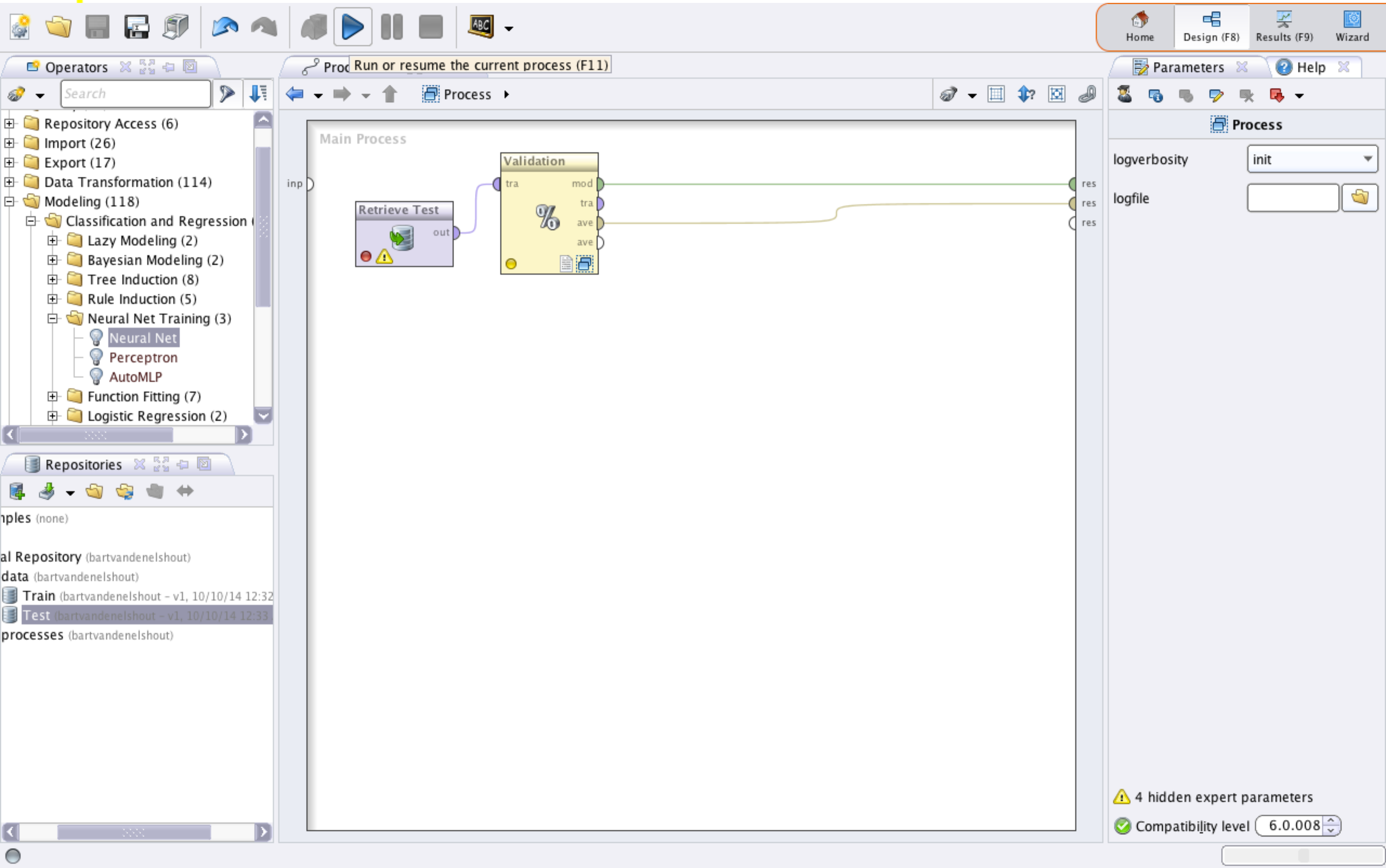


precision: 64.13% +/− 4.77% (mikro: 63.59%) (positive class: 1.0)

	true 0.0	true 1.0	class precision
pred. 0.0	1981	466	80.96%
pred. 1.0	466	814	63.59%
class recall	80.96%	63.59%	

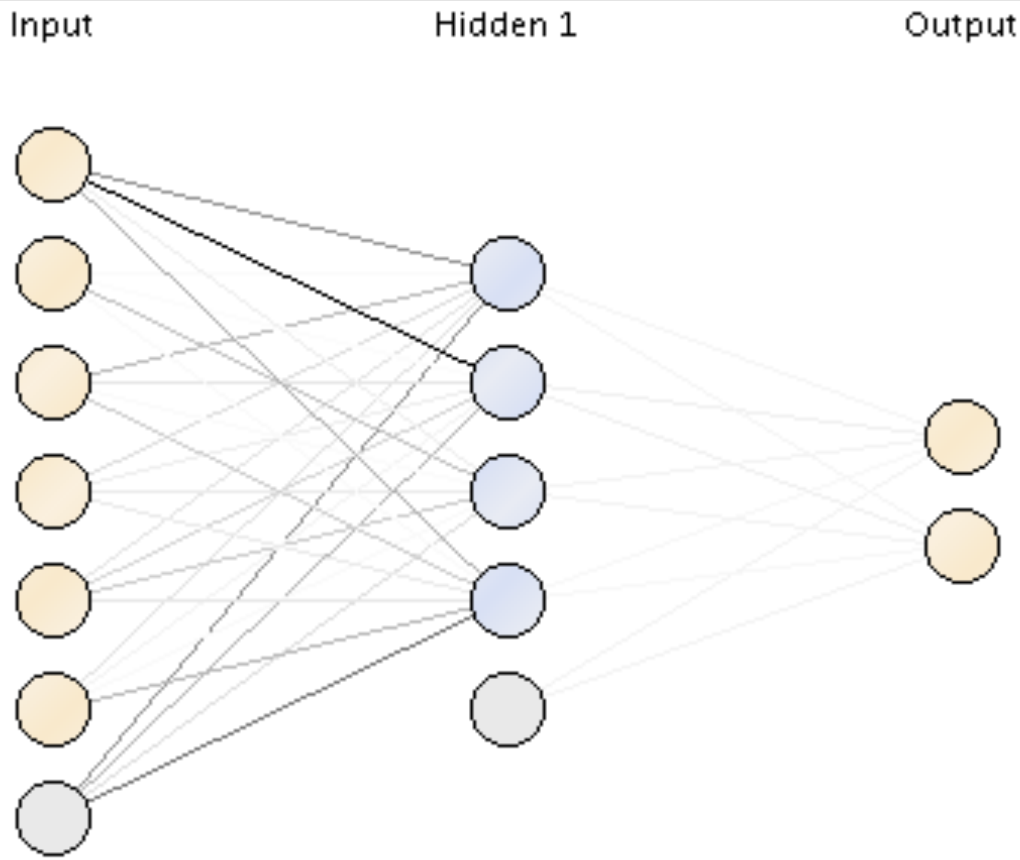
Testing the data


Home
Design (F8)
Results (F9)
Wizard
Operators
Process
Parameters
Help
Repository Access (6)
Import (26)
Export (17)
Data Transformation (114)
Modeling (118)
Classification and Regression
Lazy Modeling (2)
Bayesian Modeling (2)
Tree Induction (8)
Rule Induction (5)
Neural Net Training (3)
Neural Net
Perceptron
AutoMLP
Function Fitting (7)
Logistic Regression (2)
Main Process
Validation
Retrieve Test
logverbosity
init
logfile
4 hidden expert parameters
Compatibility level
6.0.008
Repositories
Train (bartvandenelshout - v1, 10/10/14 12:32
Test (bartvandenelshout - v1, 10/10/14 12:33
processes (bartvandenelshout)


Home
Design (F8)
Results (F9)
Wizard
Operators
Run or resume the current process (F11)
Process
Parameters
Help
Repository Access (6)
Import (26)
Export (17)
Data Transformation (114)
Modeling (118)
Classification and Regression
Lazy Modeling (2)
Bayesian Modeling (2)
Tree Induction (8)
Rule Induction (5)
Neural Net Training (3)
Neural Net
Perceptron
AutoMLP
Function Fitting (7)
Logistic Regression (2)
Main Process
Validation
Retrieve Test
logverbosity
init
logfile
Repositories
Train
Test
processes (bartvandenelshout)
4 hidden expert parameters
Compatibility level
6.0.008

accuracy: 73.42% +/- 2.18% (mikro: 73.42%)

	true 0.0	true 1.0	class precision
pred. 0.0	2117	564	78.96%
pred. 1.0	489	792	61.83%
class recall	81.24%	58.41%	



precision: 62.92% +/- 5.91% (mikro: 61.83%) (positive class: 1.0)

	true 0.0	true 1.0	class precision
pred. 0.0	2117	564	78.96%
pred. 1.0	489	792	61.83%
class recall	81.24%	58.41%	

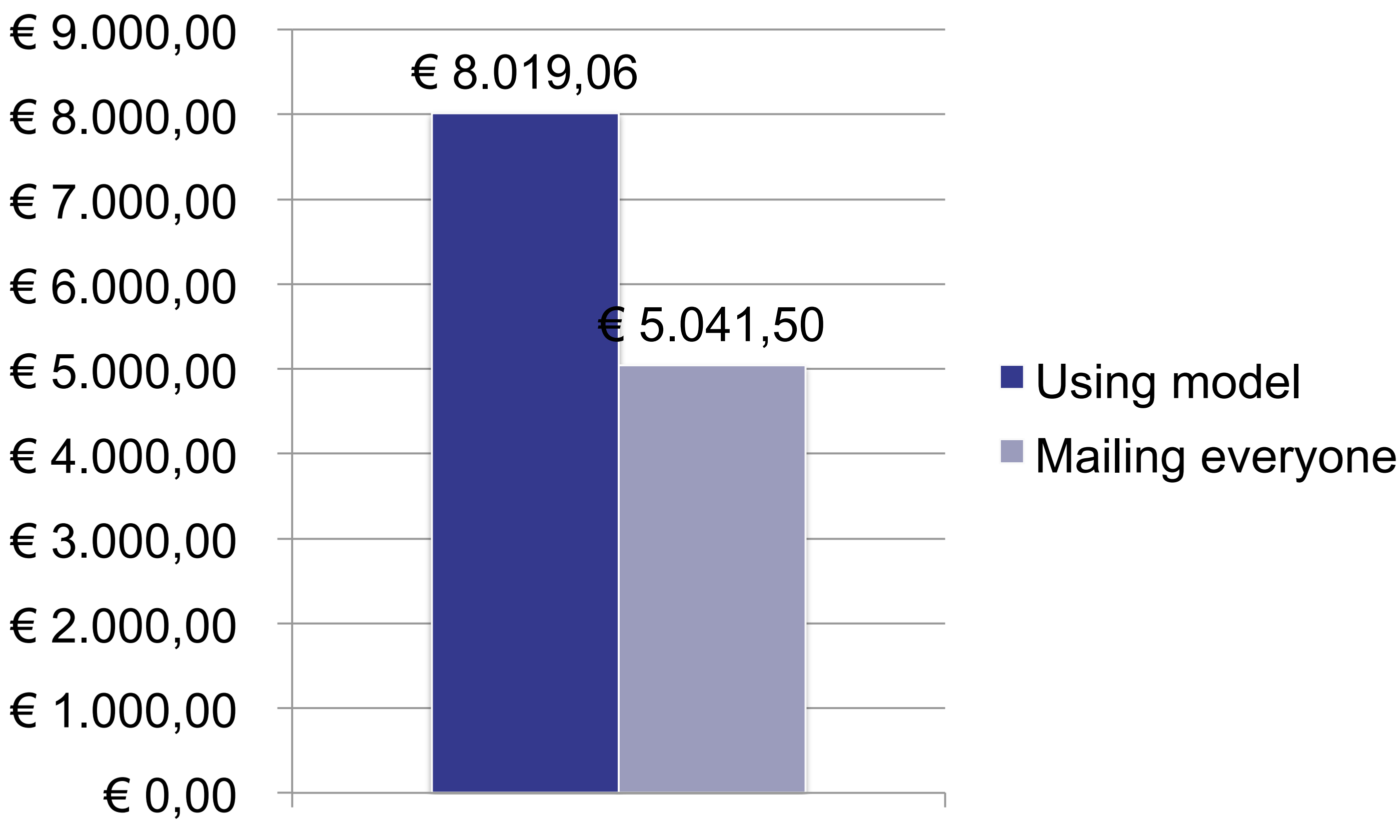
Pay-off analysis

Predicted amount of future donators:	1281
Average annual donation*:	7.01 €
Estimated costs per mailing**:	0.75 €

Projected revenue: 733 x € 7.01 =	8,979.81 €
Costs: 733 x € 0.75 =	960.75 € _
Projected “profit” =	8,019.06 €

* calculated by averaging the annual donation

** 0.64 € per stamp, + paper and writing costs


€ 9.000,00
€ 8.000,00
€ 7.000,00
€ 6.000,00
€ 5.000,00
€ 4.000,00
€ 3.000,00
€ 2.000,00
€ 1.000,00
€ 0,00
€ 8.019,06
€ 5.041,50
Using model
Mailing everyone