# Chapter 6:

# Deep Neural Network

**CHAPTER 6 Contents:**

[6.1. Deep Neural Network (DNN) 147](#_Toc125225604)

[6.2. Artificial Neural Network (ANN) 147](#_Toc125225605)

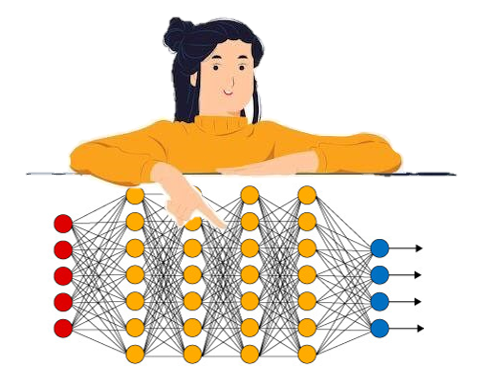
[6.3. Comparing Multilayer Neural Network with Regression 150](#_Toc125225606)

[6.4. Hyper-Parameter Tuning 152](#_Toc125225607)

[6.5. Evaluation DNNs 153](#_Toc125225608)

[6.6. Interpretability and Explainability in DNNs 156](#_Toc125225609)

[6.7. Difference between LIME and SHAP 159](#_Toc125225610)



Abstract: A particular form of network used in brain imaging, the neural network, is the topic of this chapter. Properties of neural nets are presented together with a detailed description of the structure of neurons and artificial neurons which comprise these networks. A neural net is a conceptual model based upon the human brain. It has an observed input layer which connects to a middle, hidden layer consisting of an unobserved number of synapses and neurons which leads to an observed output layer. There are different functions for measuring the degree or presence of activity between the neurons. The model can be trained in various ways but usually a design and test set approach is used where the model is given an input and allowed to develop unsupervised and eventually compared to the output (which is often a group) to assess fit using various measures such as mean absolute error and root mean square error. Tips are given on how to Interpret these fit indices. The learning rate, epoch and batch which control the amount of learning input data used are described. A worked example in R is at the end of the chapter looking at stress and anxiety levels in a child and its mother being used to predict the presence, or absence, of hyperactivity.

Keywords: Deep neural network; Artificial neural network; Multilayers; LIME; SHAP; Model fit

## 6.1. Deep Neural Network (DNN)

In 1981, Teach and Shortliffe emphasized the importance of explaining decisions in decision-making systems. In recent years, XAE has grown in popularity. Explanation and interpretation aims to reduce the opacity of black box models and to improve perception of reasoning in predictive and classification models. This reasoning precedes predictions. This makes it possible to promote the explanation and interpretation of transparency in AI/ML-based models; therefore, people will gain more confidence in such systems. First, artificial neural networks (ANNs) are analyzed in this chapter.

## 6.2. Neural Network

Neural networks or artificial neural networks (ANN) are a data processing system somewhat inspired by the human nervous system. They can also be considered an electrical model of the brain's nervous system. Like the learning mechanism, these models are also experiential. However, these models and problems employ different methods. This system uses smaller interconnected processors called neurons to create a network that is trained using training algorithms (Lawrence, J. 1993).

This technique uses the principles of the human brain and its structure to develop data processing strategies. The essence of the approach is reflected in the parallel processing of data. The procedure by which training is performed is the training algorithm. Through this process, the weights of the synapses are systematically changed in order to achieve the desired performance of the network.

Properties of neural network (Milan Milosavljević, 2019):

• non-linearity,

• input-output mapping,

• adaptability - the ability to change the strength of synaptic connections,

• evidentiary response,

• contextual information,

• failure resistance,

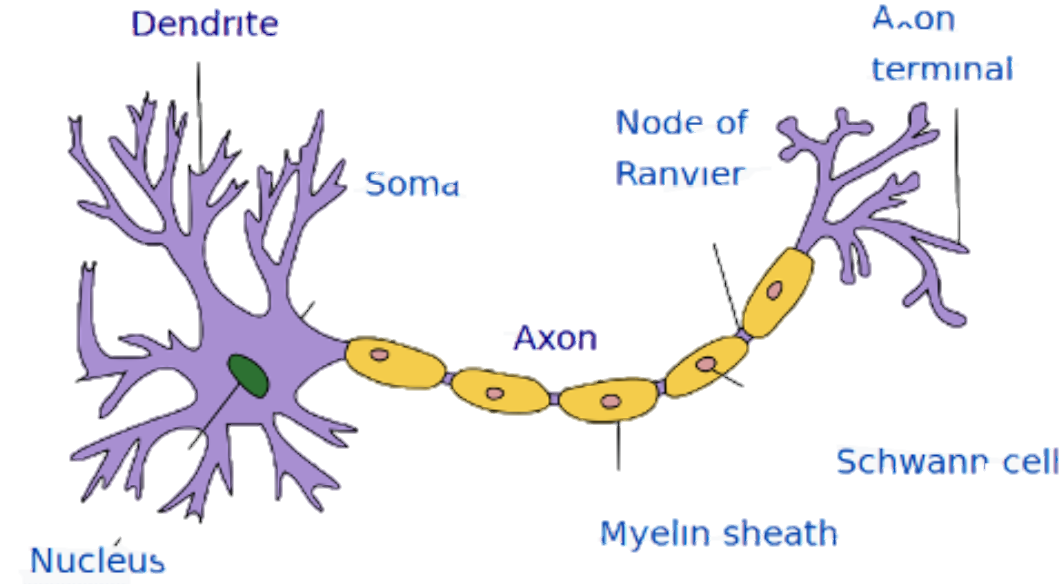
• possibility of realization in VLSI (Very Large Scale Integration) technology,

• uniformity of analysis and synthesis,

• neurobiological analogies.

## 6.3. Neuron

As mentioned earlier, the neural network in living organisms and the artificial neural network are both based on neurons. In their natural state, neurons have a biological structure comprised of dendrites, axons, and soma, as shown in the following figure.



1. Structure of neuron

Dendrites are the communication appendages of cells responsible for the communication and reception of signals from their surroundings, which are ultimately processed in the soma in the center of cells. The axon then sends the output signal to other neurons. Despite the greater complexity of the natural neuron model compared to this simple computational model, this model is the main basis of artificial neural networks (McCulloch, et al,1943).

In the mathematical model of the artificial neural cell, the processing unit or neuron can be shown as follows.

∑

W1

W2

Wn

Y

1. An artificial neuron structure

An artificial neuron is the basic element of a neural network. These neurons contain (Dragan Pamučar,2010):

• inputs-xi,

• synapses (input weighting factor) - wi,

• state of activation-z,

• output function-f,

• one output-o,

• Threshold-T.

Here, the inputs acting as dendrites are considered the features of a problem. In these models, input signals are multiplied by weight values (each input has its own weight). Thus, all inputs can be considered as input vectors, and the corresponding weights are the weight vector. The weights are usually adjusted during the training phase. Another value that may also be added to this set, the bias parameter, is also adjusted during training. Finally, the actual output is applied to the input based on the Φ activation function.

The simplest activation function is a binary function. If the total value of inputs exceeds the threshold value, the neuron spikes and the output function becomes 1, which is otherwise 0.

**(6.1)**

**A Network of Neurons:** The neural network is considered to have a network of neurons. A row of neurons is considered a layer, of which there are three in the neural network: an input layer, an output layer, and a hidden layer. The network can have different structures, and layers and neurons might vary in numbers.

Input Layer

Hidden Layer

Output Layer

1. A network of neurons

The neural network consists of (Dragan Pamučar,2010):

• The architecture (topology) of the network, i.e. the neuron connection scheme,

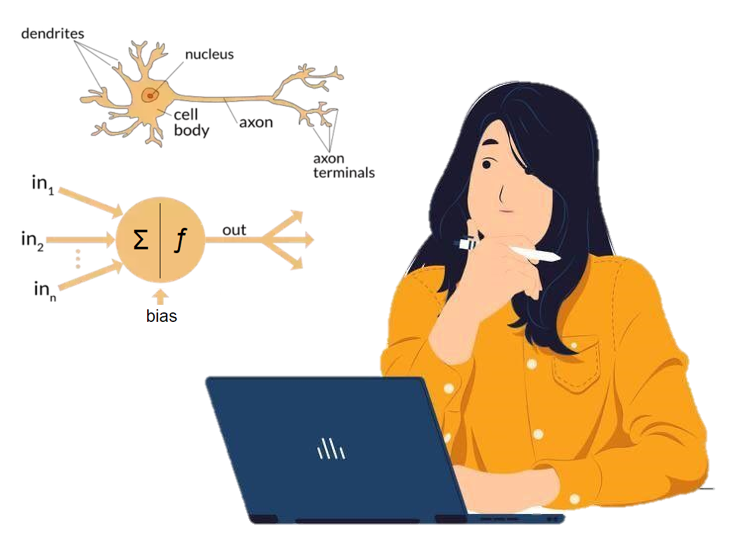
• Transmission functions of neurons

• Laws of learning.

The architecture of an artificial neural network is represented by the specific arrangement and connection of neurons in the form of a network. By architecture, neural networks differ according to the number of neural layers. Usually, each layer receives inputs from the previous layer and sends its outputs to the next layer. The first layer is called input, the last is output, and the other layers are called hidden layers. One of the most common neural network architectures is a three-layer network. The first layer (input) is the only layer that receives signals from the environment. The first layer transmits signals to the next layer (hidden layer) which processes this data and extracts features and patterns from the received signals. Data that is considered important is sent to the output layer, the last layer of the network. The final processing results are obtained at the outputs of the neurons of the third layer. More complex neural networks can have multiple hidden layers, feedback loops and time delay elements.

Learning neural networks boils down to learning from examples, which should be as many as possible so that the network can behave more precisely.

## 6.4. Artificial Neural Network (ANN)

Some human inventions have been inspired by nature. In fact, nature is full of inspiration, and those who are more prepared than others will benefit from intuition.

A neural network is a conceptual model based on the human brain, which consists of nearly 10 billion neurons and 60 trillion synapses (Shepherd, & Koch ,1990).

The human brain is considered to be a very complicated non-linear parallel information processing system.

Therefore, scientists have tried to mimic the behavioral pattern of a neuron and design an artificial neuron called a perceptron. This chapter does not intend to review the history of the perceptron but aims to address its application to explainable artificial psychology presented in this book. The idea proposed by Pitts and McCulloch in 1943 was used as the basis for artificial neural networks (ANNs).

An ANN contains a large number of very simple interconnected processors called neurons. These artificial neurons serve as biological neurons. They receive inputs and provide outputs.

In ANNs, neurons are connected by links, each of which is given a numerical weight. These weights serve as long-term memory in ANNs. They represent the strength or importance of each input neuron. With ANNs, learning means nothing more than adjusting these weights in an iterative process.

Table 12 draws a brief comparison between a biological neural network and an artificial neural network.

1. a brief comparison between a biological neural network and an artificial neural network

|  |  |
| --- | --- |
| Biological Neural Network | ANNs |
| Soma | Neuron |
| Dendrite | Input |
| Axon | Output |
| Synapse | Weight |

The following figure demonstrates an artificial neuron as a computing element of ANNs.

Y

X1

X2

Xn

.

.

.

Input signals

Weight

Y

Y

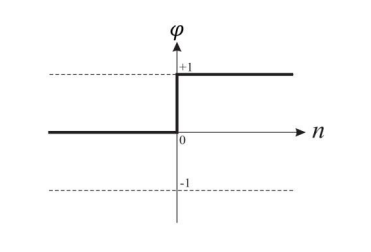
1. an artificial neuron’s elements

As already mentioned, the main idea of ANNs in 1943 was based on the seed presented by Pitts and McCulloch. They briefly pointed out that the weighted sum of the inputs would be calculated by a mathematical formula called neuron and its output would be compared to a threshold (θ). If the final value is less than the threshold, the neuron output is -1; however, it is +1 if the final value is greater than or equal to the threshold. There is an activation function in these sets, which is computed as follows:

**(6.2)**

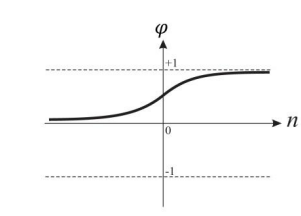
**(6.3)**

Nevertheless, the sign function is not the only activation or transfer function. There are several active functions, a limited number of which are practically applicable.

**(6.4)**

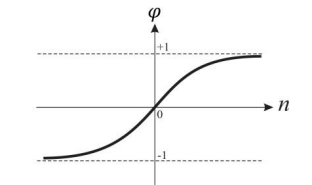
1. Threshold Function

This is a widely-used function in an artificial neural network. The following are two examples of these functions, which are widely used since they are derivable, ascending, and continuous.

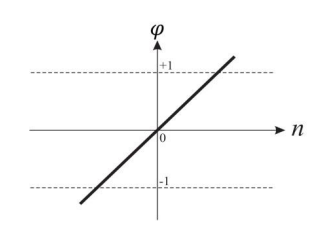
**(6.5)**

1. Sigmoid function-1

**(6.6)**

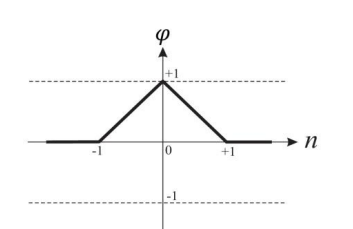


1. Sigmoid function-2

This function uses the simple equation for a linear approximation. Its diagram is as follows:

1. Linear Activity Function

**(6.7)**



1. Triangular Activity Function

The level and sign functions are also known as hard limit functions, which are commonly used for pattern recognition and classification. In addition, the sigmoid function can take any input value from -∞ to +∞ and convert it into a value between 0 and 1. This activation function is typically used in backpropagation networks.

In 1958, Frank Rosenblatt presented a training algorithm that can be considered the first training method for a simple ANN.

He named this simple single-neuron network the perceptron, which consisted of a single neuron with tunable synaptic weights and a hard limiter. This model was based on the idea of Pitts and McCulloch. Over time, this single neuron ANN evolved and multilayer neural networks (MLNNs) emerged in scientific fields.

Classification is very common in psychological studies and this purpose fully justifies and proves the application of explainable AI through the use of MLNNs in the psychological sciences.

The Multilayer Perceptron Neural Network (MLPNN) is a MLNN in which input signals (i.e. input data) propagate in the forward direction layer by layer. This model consists of an input layer, a middle or hidden layer, and an output layer.

In MLPNNs, each layer has its own specific function. The input signals from the outside world are fed into the input layer and then redistributed to all the neurons of the hidden layer. The input layer either lacks computational neurons or rarely has any; therefore, no processing takes place in this layer. The output layer receives the output signals from the hidden layer and creates the output pattern of the entire network.

The neurons of the hidden (middle) layer recognize some features and the weights of the neurons are considered as features for the hidden layer, which is called hidden because it hides the output of interest. In other words, the hidden layer determines its desired output. It is possible to increase the number of hidden layers. Deep learning is achieved when the network has more than one hidden layer. The number of neurons in each hidden layer can range from 10 to 1000 layers. Although it is possible to increase the number of hidden layers arbitrarily, this increases the computational load.

 Learning is very important in MLNNs and Deep Learning. There are more than a hundred different algorithms for network learning. The most common algorithm is the backpropagation method, introduced by Ho and Bryson (1969). Learning in a multi-layer network resembles a perceptron. The network calculates the output pattern. If there is an error (i.e., a difference between arbitrary and real output patterns), the weights are adjusted to reduce this error.

The goal of this learning algorithm is to minimize the sum of squared errors. A network is said to have converged if the sum of the squared errors in the learning sets or in the epoch is sufficiently small.

Furthermore, learning in an MLNN does not exactly emulate the behavior of a biological neuron, for these neurons do not have backpropagation mechanisms.

## 6.5. Types of training

There are three types of training (Anđelković, 2005):

• **Supervised training** - input data and expected output data are presented to the network,

• **Training by evaluation** - the network is not presented with the expected output data, but after some time is presented with the evaluation of the previous work and

• **Self-organization** - only input is presented to the network.

Initially, neural networks were used to model the nervous systems of living organisms. Today, neural networks are applied for:

• shape recognition,

• Handwriting recognition,

• Speech recognition,

• Financial and economic models,

• Predicting price movements on the market,

• Systems management,

• Management of production processes,

• Analysis of electric circuits,

• Psychiatric assessments,

• Data compression,

• Oil research,

• Criminological research,

• Analysis of medical tests,

• Examination of EEG and ECG signals,

• finding the optimal solution,

• Managing robots,

• analyzing data during pyrolysis and spectroscopy,

• In biocomputer systems,

• Weather forecast in other areas as well.

Neural networks have been successfully applied in supervised and unsupervised learning. Neural network techniques belong to the so-called nonlinear techniques that can model complex functions. Generally, they are applied in the fields of prediction, classification or control in a number of fields.

A key feature of neural networks is related to learning the relationship between input and output parameters through the network training process.

Network training is used to learn behavioral patterns and the main goal of training is to find a set of weights between neurons that determine the global minimum of the "error function". This procedure includes decisions in terms of the number of training iterations, that is, the point at which training stops ( Anđelković, 2005).

The most famous training algorithm is known as back propagation. During the training phase, the training data is in the input layer. In this algorithm, each node in the hidden layer receives input from each node in the input layer which is multiplied by the appropriate weights and then summed. The output from the hidden node is a nonlinear transformation of the resulting sum. Similarly, each node in the output layer receives input from all nodes from the hidden layer which are multiplied by the appropriate weights and then summed.

Then the resulting output values are compared with the target output values. The target output values are those values that the neural network is trying to teach.

Then, the error between the obtained output values and the target values is calculated and fed back through the hidden layer. This procedure is called the backward procedure through the hidden layer. The error is used to correct the strength of the connection between the nodes, the weights between the input and hidden layers, and in this way the hidden and output layers are updated.

## 6.6. Usage of neural network

Neural network models can be used for:

• **Classifications** - if the object should be associated with one of the existing, predefined groups or classes and

• **Predictions** - if an object that is not in the set of existing classes is predicted.

Before using software that includes neural network techniques, it is necessary to determine the reliability of the results obtained. Reliability is determined through procedures:

• **Validation** (This procedure provides an answer to the question: "Is the appropriate product made?")

• **Verifications** (This procedure answers the question: "Is the product made in the right way?")

Validation is a process in which the degree of accuracy of the model representation is determined in relation to the purpose of using the model (AIAA G-077-1998 2002).

Verification provides an evaluation of the accuracy of the model. Also, the verification process analyzes the test results and provides results that measure the reliability of the system. Verification is the process of determining whether the model implementation accurately represents the conceptual description of the model as its solution.

Verification and validation processes include model evaluation using real data. In order for the neural network to be successfully applied in real situations, it is necessary to evaluate the test results.

Verification ensures the complete specification and ensures that there are no errors in the implementation of the model. However, verification does not ensure that the model solves the problem, and that it accurately represents the real processes that occur.

There are several approaches for obtaining information about the quality of information obtained through data analysis models (model validation):

• Use of statistical indicators,

• Division of the data set into two parts: a part for training and testing

• Consulting with experts and reviewing the results obtained to determine whether the discovered behavior patterns make sense in the specific use case

This procedure is carried out in order to determine the fulfillment of the following characteristics of the created model:

• Accuracy,

• Reliability and

• Usefulness.

## 6.7. The Artificial Neural Network Structure

The arrangement of neurons and layers in the network represents the neural network’s structure. An artificial neural network includes layers of one or several neurons. Since it is not involved in computations, the first layer is not considered a neuronal layer. The parameters of artificial neural networks and neuron behavior will be discussed in the following. In terms of the number of layers, there are two types of neural networks (Minsky, Papert, 1969).

1. Single-Layer Networks

This network includes a layer of neurons with weighted connections where n denotes the number of inputs and m is the number of outputs.

Input Layer

Output Layer

1. Single-Layer Neural Network

2. Multilayer Networks

These artificial neural networks include one or several hidden layers between input and output layers. Compared to single-layer networks, multilayer networks can solve more complex problems but are much more difficult to train (Camuñas-Mesa,2019).

Input Layer

Hidden Layer

Output Layer

1. Multilayer Neural Network

Therefore, the number of layers is an important efficiency parameter of neural networks. Selecting too few layers may render the network unable to provide a good solution, whereas selecting too many layers may significantly increase network complexity and training time. The number of neurons in each layer is another important parameter requiring careful selection since selecting the wrong number of neurons has the same consequences as the number of layers mentioned earlier (Ma, S., & Ji, C. 1999; Camuñas-Mesa,2019)

## 6.9. Modeling an Artificial Neural Network

As mentioned earlier, the elements of an artificial neural network include activation functions, weights, and biases. Generally, since the network's activation function is assumed to be unchangeable and constant, the variables are weights and biases. Therefore, the neural network can be generally considered a function with w and b parameters as follows:

ANN

X

Y

1. Modelling an ANN

**(6.8)**

Now, to use a neural network to model a process (any process, for example, the diagnosis of a mental disorder by a psychologist can be considered a process), we have:

ANN

X

Diagnosis Process

Y

1. Diagnosis process and ANN

Here, Y is the output and is the output the network aims to model in the shared input. Ideally, and modeling should be accurate. Nevertheless, natural and real problems are known to have many parameters that cannot be modeled completely. For example, is the parameter of the moon’s size and its angle to the earth relevant to determining a person's mood or activity? Despite its apparent irrelevance to the problem, just as the moon's gravity affects the tides of oceans and seas, it also influences the amount of water in the human body and can affect the human body and mood. Nonetheless, these parameters are ignored without knowledge of their existence or application and measurement in the model. Thus, never occurs in normal problems. In this case, the goal is to minimize the error between the two outputs:

ANN

X

Diagnosis Process

Y

1. Minimizing error in ANN

Therefore, this is an optimization problem. To find the best model, the bias and weight values should be determined such that the value of *e* or its equivalent, *e2*or *MSE*, is minimized. Hence, the problem is as follows:

**(6.9)**

Since , we have:

**(6.10)**

Here, we are after the correct *W, B* values, respectively weights and biases.

Now, if the input is more than one number and the data covers several patients, the sum of errors, which creates the *MSE* problem, should be considered:

**(6.11)**

Note that here, all inputs have the same value; if they do not, it is possible to apply weights to them.

Therefore, this is an optimization problem that requires two general optimization methods (Agatonovic-Kustrin,2000):

### 6.9.1. Classical Optimization Methods

Although this classification includes numerous methods, one of the best-known is arguably the gradient descent method. It is among the most widely-used regression and classification methods and algorithms. The backpropagation method also uses this method and depends on the first-order derivative of the optimization function to determine how to change weights to minimize the function. Its advantages include ease of calculation, explanation, and implementation, but it is very memory intensive and could fall into a local minimum.

There are several measures used in the validation process of neural networks. The most commonly used measures are:

* mean absolute error - Mean Absolute Error (MAE),
* Root Mean Square Error (RMSE),
* relative error-Relative Error (MARE),
* correlation coefficient (R2).
* Root Mean Square Error (RMSE):
* RMSE (Harper, G. 2011)

**(6.12)**

Where ti is the computed output provided by the network, oi is the desired (actual) output for case (input vector) i, and n is the number of cases in the sample. The error is averaged according to the number of output variables and in relation to the number of cases in the sample on which it is calculated.

In addition to the mentioned measure, the classification rate is also used in the classification problem as a measure of network validation. The classification rate shows the percentage or proportion of correctly classified cases.

### 6.9.2. Intelligent Optimization Methods

These methods use intelligent algorithms such as a genetic algorithm, PSO, ISA, and DE. Meta-heuristic algorithms can be used to determine the weights of edges and biases in neural networks.

## 6.10. Types of Data in Machine Learning Algorithms

Data is very important in machine learning algorithms, which actually extract relationships from existing data. In machine learning algorithms, data can be divided into three categories.

1. **Training Data**

Used for building the model, these data have known outputs to the model and are called learning or training data.

1. **Test Data**

These data are used for model evaluation and their class status is unknown to the model. . After applying the data, the class status is compared to the predicted class status to evaluate the model's efficiency.

1. **Validation Data**

This data is used for testing various trained models in supervised learning. They are taken from training data and used for model evaluation before being returned to the model. Testing data is used with the assumption that the algorithm has used a series of questions for training, including questions and correct answers, and to test whether training was completed. In this case, validation tests are the midterms that show progress in education. Sometimes, however, this data is not considered and is only divided into test and training groups (Bonaccorso, 2017).

Train

Train

Validation

Test

Train

1. Types of Data in machine-learning algorithms

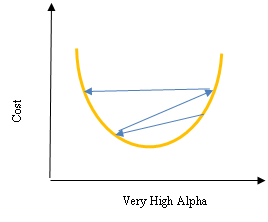
## 6.11. Basic concepts

**Learning Rate**:

As one of the most important hyperparameters in neural networks, learning rate helps with proper adjustment of the neural network. It indicates the magnitude of each step in each iteration for network training until the cost function is minimized. This concept is usually denoted by alpha, which varies between 0 and 1. Finding the optimal operational learning rate is difficult. If too high, that is, the steps are large; the algorithm operates faster with the risk of losing the minimum value for the cost function, since the weights are updated quickly. If low, the calculation will be slower and produce results later, and if too low, there is a possibility of getting stuck in the local minimum. Thus, its size, which should be between the two states, depends on the data and network structure. This concept is shown in the following figure (Bengio, Y. 2012).

Cost

Very Low Alpha



1. The impact of Alpha in cost function

Meanwhile, concepts such as epoch, batch size, or iteration in machine learning and deep learning are used when the dataset is very large and all the data cannot be entered into the model at once. Hence, the data must be divided (Wang, X 2017).

**Epoch:** Epoch is when all the data has moved forward or backward once along the network. The epoch number is a hyperparameter related to forward and backward movement in the network with higher numbers meaning greater accuracy but also longer training time and possible network overfitting. A period is an opportunity for each sample to change the internal parameters. The number of epochs is usually considered high so that the model continues until the error is minimized. (The number can be 0 to infinity and even greater than 100 or 1000). If the epoch is too big to be given to the model at once, it will be divided into small batches. (Wang, X 2017).

**Batch**: An epoch consists of one or several batches (batch size is also a hyperparameter that determines the number of samples). With this method, the model receives the data in batches instead of all at once.

**Iteration**: Iteration is the number of batches needed to complete an epoch; that is, the number of batches is the number of iterations in each epoch. (Wang, X 2017).

**Network Operation Modes**

**Overfitting:** It is one of the modes where machine learning underperforms. In overfitting, the model uses too much detail and noise from the training data, which negatively affects the model and its future predictions. In this case, model error is good for training data but poor for the test and new data, causing a generalization problem. The overfitting model is typically a non-linear and non-parametric one. The following figure clearly illustrates the concept of overfitting.

1. Illustration of overfitting

**Underfitting**: Like overfitting, it provides an inappropriate model that should be avoided. This model is inappropriate for training data and test data. In this case, different learning models should be tested to find the best model for the problem, which the following figure clearly shows (Brownlee, 2016).

1. Illustration of Underfitting

**Good Fit**: Statistically, fit means the quality with which the model can approximate and model the target function. Machine learning algorithms include methods for this purpose, such as finding residual errors in supervised algorithms. Ideally, the model should be somewhere between overfitting and underfitting, which is a challenging goal (Brownlee, 2016).

1. Illustration of good fit

A model error can be measured for analysis. Ideally, training and testing data errors while training the network should be reduced together. Now, if training takes too long, the model pays too much attention to details and noise, which causes overfitting and increases test data error (Brownlee, 2016).

Epoch:

Error

Train

Test

Underfitting:

Fit

Overfitting:

1. Impact of overfitting and Underfitting on Error

**Cross-validation**: As explained earlier, the model needs to estimate and check the parameters for performance evaluation (e.g., to avoid overfitting). In machine learning algorithms, the generalization feature, which is very important for inputs, is accomplished by cross-validation. Moreover, too many network parameters introduce difficulty in the performance evaluation of the model and necessitate cross-validation. Cross-validation is easy to implement and understand, has less bias than other methods, and has different methods, such as the holdout method, Leave-One-Out method, Leave-P-Out method, and k-Fold method (Russell, S. J. 2010 ).

## 6.12. Types of Artificial Neural Networks

There are various artificial neural networks for different applications all inspired by the human nervous system. A close examination of learning in the human brain reveals a similar process via the weakening or strengthening of the connections between brain cells or neurons. In mathematics, these changes are known as weights. Some neural networks will be discussed in the following.

**Multilayer Perceptron (MLP) Neural Network**

Considered one of the most basic artificial neural networks, it typically consists of an input layer, one or several hidden layers, and an output layer. This neural network considers the network behavior of humans and is also known as the feed-forward network. Here, neurons receive, process, and transfer signals between each other, which ultimately produces a result. This network uses the mean square error (MSE) as an efficiency index (Marques, et al, 2014)

X1

X2

Output layer

Hidden layer

Input layer

Y

1. Perceptron network

**Feed-Forward Neural Network**

This is also one of the older neural networks with several main rules, such as the interconnectivity of all nodes, activation from the input layer to the output layer, and a hidden layer between input and output.

Deep feed-forward neural networks have more than one hidden layer, which is very time-consuming and impractical.

**Radial Basis Function Neural Network**

Despite its structural similarity to MLP, this network has different neuron processing. It employs the logistic function for the activation function. Due to its easier setup, this network provides faster training and is suitable for classification. However, it underperforms with regression.

**Recurrent Neural Networks**

In these networks, the neurons have a recurrent state. Basically, recurrent neural networks are used when the “context” is important and when decisions from previous iterations or samples can affect current examples.

**Deep Learning**

Generally, a higher number of layers and neurons in the artificial neural network make the network deeper and the model more complex. Having over three layers in the artificial neural network creates a deep network used for deep learning. In fact, deep learning is an artificial neural network with a large number of hidden layers. Nonetheless, the difference between deep learning and neural networks lies in the fact that deep learning is the learning mode, whereas neural networks are machine learning algorithms (Dangeti, P. 2017 ).

## 6.13. Comparing Multilayer Neural Network with Regression

If an artificial neural network has no hidden layers but has a sigmoid activation layer, the neural network is equal to a logistic regression. If it has a linear activation function, it is equal to a linear regression.

An artificial neural network designed for output layer classification has a class label.

 In recent years, neural networks have been redesigned in deep learning format, often leading to better results (Goodfellow, Bengio, & Courville, 2016).

The aim of machine learning (ML) is to develop computing algorithms or statistical models that can automatically infer hidden patterns from data. In the latest technological wave of ML and AI, deep learning approaches aim to develop an end-to-end mechanism that can directly map input raw features to outputs and place a multi-layer network structure between them to capture inputs from hidden patterns.

There are various deep learning (DL) strategies. This chapter addresses only the deep feedforward neural network.

As previously mentioned, ANNs aim to mimic how the human brain works. From a mathematical point of view, an artificial neuron is viewed as a non-linear transformation unit that captures the weighted set of all inputs and feeds the result to an activation function (e.g. sigmoid, rectifier (relu) and hyperbolic tangent).

The simplest ANN is a feedforward neural network that stacks neurons layer by layer in a forward pattern. The first layer is the input layer. Each unit gets one dimension of the data vector. The last layer is an output layer that outputs the probability. It also assigns an individual to different classes. The middle layer is the hidden layer. There are usually several hidden layers, which is why a network is called a deep network.

The input features are considered as an input layer when designing deep artificial neural networks (DANNs). The hidden layer is mainly selected as dense or fully connected, and each neuron is assigned to another neuron in previous or next layers. The network is then trained to find the data pattern. This process results in a model that can be used to predict scores or classes (here classes) in the new untrained features. In every training process there is an epoch. An epoch is the number of times a model has updated itself. Updating means calculating the error and back-propagating its gradient in previous layers. The class- weight is a very important parameter in the network because it reduces the relative weights of false negatives in the total error.

It is important to manipulate this parameter, because researchers often have to deal with false positives. Therefore, this ratio must be chosen carefully. The ratio is usually chosen as 1:4 because it is usually acceptable (Bailey et al, 2021).

After the network has been trained, its appropriateness should be tested. For this purpose, it is imperative to use the data that has not been used to train the network. From a psychological point of view, testing all models would be the test of generalization to the untrained.

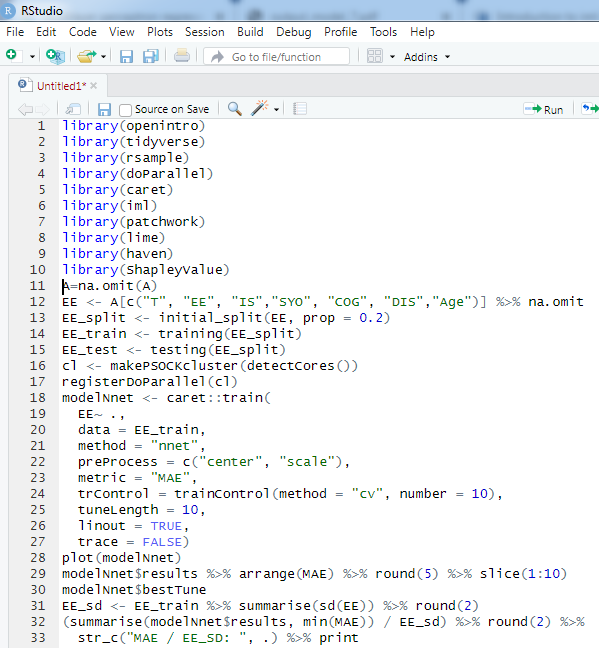
To test a model, the prototype is divided into two parts: 70% training and 30% testing.

### 6.13.1. Practical example using R

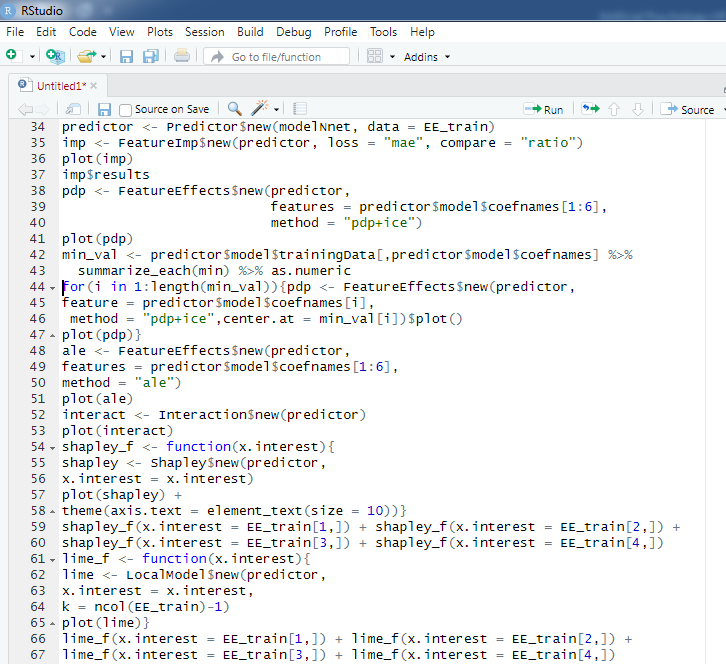
**Example1**

In this research, an artificial psychologist using DNN artificial neural network tries to predict Emotional competence (EE) based on 2 other variables. This model shows the application of DNN regression. The R codes for DNN is in listing 1. In this example, in order to ensure the results, in addition to dividing the main sample into two training sample and the test sample, the researcher also performs Cross-Validation with K=10. He also standardizes the predictor variables.

1. R codes for DNN implement



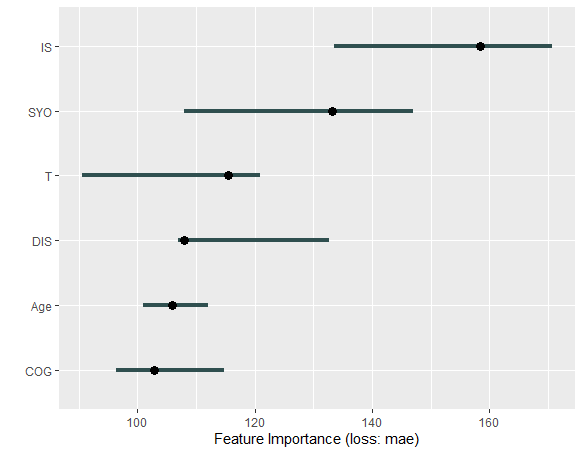
1. (continued)R codes for DNN implement



|  |  |
| --- | --- |
| Variable label | variable name |
| IS | Internalized Shame |
| T | Childhood Trauma |
| COG | Cognitive Flexibility |
| DIS | Distress Tolerance |
| SYO | Alexithymia |
| EE | Emotional Competence |
| Age | Age |

1. Abbreviated name and description of variables

Table 1 shows the variable label and their names. In this analysis the cost function is the loss function and has been used by MAE. . Figure 6.22 shows the median importance of the features .



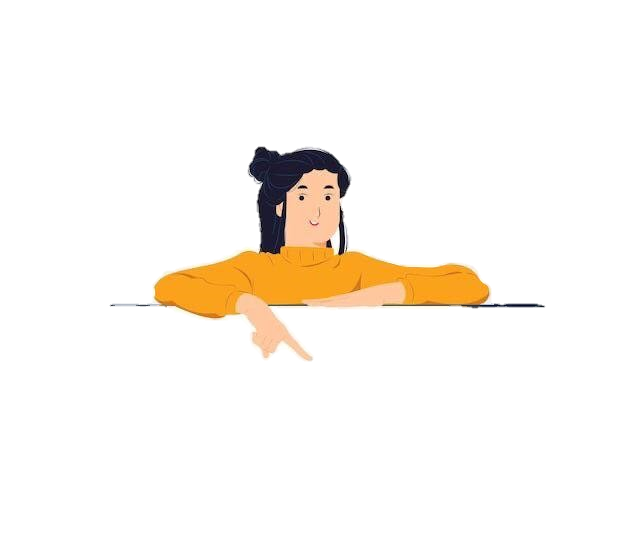
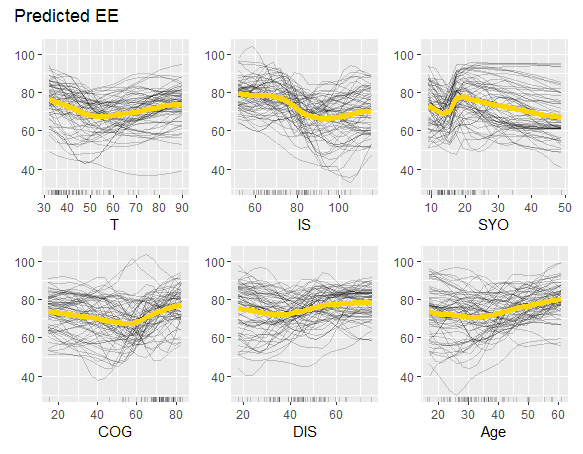
1. The median importance

This plot shows that the IS variable has the most relative importance in predicting EE and COG has the least importance.

The ------- chart provides summary information about the behavior of the model.

There are different models for interpreting features (variables). One of these methods proposed by Friedman in 2001 is the Partial Dependence Plot (PDP), which shows the Marginal Effect of variables on the dependent variable(s) in the trained model (listing 2). PD shows how much the predicted values change for a change in the value of a variable in the model.

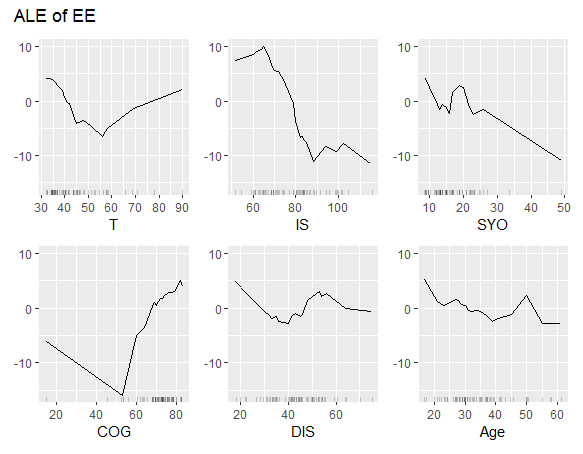
In 2015, Goldstein et al. introduced the Individual Conditional Expectation (ICE) plot. This plot shows how the predictions of a fitted model change if only the variable in question changes compared to the rest of the variables. One condition is that the rest of the variables are fixed based on the value of a sample.



1. PDP and ICE plot

Listing 2 shows the R codes for drawing PDP and Rice. The results are shown in the plot 6.23. In this PDP diagram, they are drawn as yellow curves and ICE curves are drawn in black, and the distributions of each variable are shown as short lines on the -X axis. It can be seen that descending lines like IS and T show a negative relationship with EE and a variable like SYO shows a non-linear relationship.

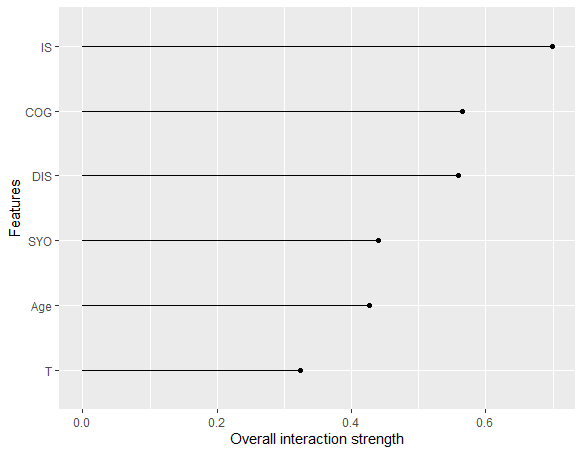
In 2016, Alpey introduced a new plot called Accumulated Local Effects (ALE). The ALE Plot calculates the average effect of predictor variables on model predictions. It is faster and more unbiased than PDP. This chart includes the interaction of variables even if they are correlated. The ALE plot related to the DNN regression model is shown in Figure 6.24.



1. ALE plot

Figure 6.24 indicates that predictor variables have non-linear relationships with EE, which was not easily observed in PDP.

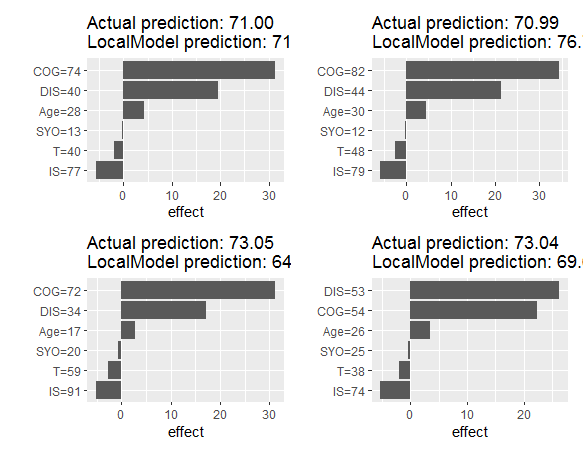
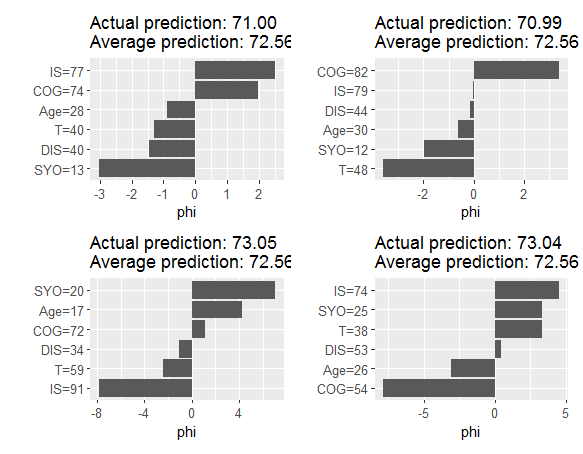
Figure 6.25 shows feature interactions. This diagram shows a way to show the interaction strength between 2 features. This value was given by Friedman and varies between 0 and 1. Zero (0) represents no interaction between two variables and 1 (one) is for when all the effect on the observed values is due to the interaction.



1. Interaction plot of the variables

As the interaction plot in Figure 6.25 shows, the two variables IS and COG have the most interaction with other variables and T the least.

As we discussed, LIME and Shapley Value are used for local interpretation. LIME is not used in this example because the package used in this case does not use LIME, but we will discuss it in the next example. We use the Shapley Value in this example. The artificial psychologist should keep in mind the importance when using Black-box ML in Interpretability and explainability of findings and results. The Shapley Value was developed in 1953 based on game theory, which actually shows the relative contribution of predictor variables in predictions. In the IML package available in R, this indicator is based on a Monte Carlo sampling approximation by Strumbelj et al. in 2014.



1. Interaction plot of the variables

Figure 6.26 shows the average predicted values compared to the actual prediction. Using the Shapley Value, it is possible to compare actual and average predictions for a subset of data and even for a single data point.

## 6.14. Hyper-Parameter Tuning

The model training process involves the use of parameters. Depending on the data and the type of problem, parameters are obtained from different data sets.

However, hyper parameters are determined by the involvement of a researcher. For example, they refer to the number of hidden layers in a DNN or the number of trees in a random forest. In general, this process aims to optimize models by reducing errors or cost functions and improving accuracy. Two approaches should be considered by artificial psychology when determining hyper parameters: 1) a manual approach; 2) an automated approach. The former is based on prior knowledge. Sometimes an artificial psychologist knows that better results are obtained when hyper parameters are determined by the knowledge-based model. However, the second approach does not require prior knowledge. There are various parameters in DNNs. They can be manipulated directly. This changes the model architecture or the learning rate (Yang & Shami, 2020).

Hyper-parameters include the number of neurons in a hidden layer, the number of epochs, the number of hidden layers, the number of mini-batches, the activation function, the cost function, the learning rate, and regularization methods (Koutsoukas et al., 2017).

The first approach consists of some recommendations and approximation techniques that can be used to determine hyper-parameters and observe their impact on the model.

When determining the number of neurons in a hidden layer, one solution is to use half of the input data (Heaton, 2015). A better solution is to employ the hyper-parameter tuning technique. To do this, the training data is split into two halves:

1) Training set and 2) validation set.

The training set is utilized to train the model while the validation set is only used to check the model accuracy.

In practice, 50% of the data goes into the validation set and the rest goes into the training set, so both sets can contain the same amount of data. Therefore, the accuracy analysis is performed on the validation set rather than the test set, preventing overfitting. The optimal model is determined by hyper parameter tuning in the validation set.

The random search cross-validation can be used in an automated approach. This method can be employed to resolve the opacity in parameter regularization when there is no prior knowledge. In this method, a grid of value ranges is determined for hyper-parameters, from which some values are then randomly selected and evaluated. In this method, the most important arguments control the number of iterations (n-iter) and the number of different combinations tested (usually it is considered 100). Undoubtedly, increasing the number of these two arguments will reduce overfitting; however, a long time is required to fit the model.

Therefore, the best post-training hyperparameters can usually be obtained when random search CV is implemented. This is considered a major achievement as it helps access a smaller range of hyperparameter values. All resulting combinations are then analyzed using grid search CV instead of random sampling. The CV random search is used to reduce the space of hyper-parameters, while CV grid search is used to determine the best parameters.

After the model has been trained and validated, a new sample is used to check generalizability. In other words, 30% of the prototype that was not used for model training is now used.

Sometimes the data may not be enough (i.e. the sample is small). In this case, cross- validation can be used. To do this, a single data set is divided into two or more approximately equal data sets. They are called folds and are represented as "k", which is usually taken as 10. In this case, the algorithm trains the model in k-1 folds and then tests it on the remaining fold, called the holdout. The number of folds is usually considered to be 10 in cross-validation.

## 6.15. Evaluation DNNs

The confusion matrix is used to evaluate DNNs with the purpose of classifying the target variable. This matrix helps determine the classification and misclassification rates in the model. Conventionally, the effectiveness and performance are determined by its accuracy. There are some components in the confusion matrix.

1. Confusion matrix layout. ‘1’ for positive class

|  |  |
| --- | --- |
| TN | FP |
| FN | TP |

1 0

0

1

TN: True Negative, FP, False Positive

FN: False Negative, TP, True Positive

The following four metrics are employed to evaluate the performance of a classification model:

1. Accuracy: This metric indicates the total percentage of individuals classified correctly.

**(6.13)**

1. Precision: This metric indicates what percentage of individuals having the predicted feature actually have that feature.

**(6.14)**

1. Recall: This metric indicates what percentage of individuals that actually have that feature possess the predicted feature correctly.

**(6.15)**

1. F-Measure or F-Score: This metric is a combination of precision and recall, and R indicates the importance of recall over precision. If B=1, then they are equally important.

**(6.16)**

The Receiver Operating Characteristic (ROC) curve is another tool for analyzing the performance of classification models. The area under curve (AUC) of ROC is measured for performance analysis. In the ROC curve, the true positive rate (TPR) is drawn against the false positive rate (FPR).

One of the most common metrics is R2, the proportion of variance accounted for by the model. Higher R2 values signify higher accuracy. When the residual variance (i.e., “sum of squared residuals” in the formula below) is zero, the model makes perfect predictions and R2=1. If the sum of the residuals is equal to the total variance (in the denominator), the model is useless, predicting the mean is equally accurate, and R2=0. Other metrics focus on the average residual size (instead of residual proportion as in R2), with smaller residuals signifying higher prediction accuracy. The most common is the mean absolute error (MAE); negative signs of residuals are removed) and the root means square error (RMSE; residuals are squared).

**(6.17)**

**(6.18)**

## 6.16. Interpretability and Explainability in DNNs

First, it is essential to distinguish interpretation from explanation. In fact, interpretation refers to a model’s ability to interpret the input-output relationship, while explanation denotes the ability to explain the model output in human language.

As a model becomes more complex, it becomes increasingly difficult to introspect and understand how that model makes a particular prediction. There are many methods to explain black box models. These methods are usually adopted in terms of their scope, model dependency or independence, and runtime.

An artificial psychologist knows that a DNN is a black box model. Although it has accurate inputs and outputs, it must analyze these results to determine where to find the meaning of features and how to interpret and explain the results. LIME is a new modern tool for the model-agnostic technique. It can be applied to any model to make the results interpretable and justifiable. LIME was first introduced in 2016 by Marco Tulio Riberio, Sameer Singh and Carlos Guestrin and stands for locally interpretable model-agnostic explanations. They proposed this method in a paper entitled “Why Should I Trust You? Explaining the Predictions of Any Classifier?”

According to the results reported by the developers of this method:

1) The explanations of predictions should be understandable; they should be explainable in every aspect.

2) It should be possible to make individual predictions, something which is called the local fidelity.

3) The explaining model should be applicable to all models, something which is considered model-agnostic.

4) The model should be able to provide a general explanation, which means the global perspective.

Therefore, it can be concluded that explanation is very important because we need to trust that the model predictions are correct (trust), the model behavior is correctly perceived (control), the signs of explanation are sufficient and data meet the expectations (prediction evaluation), and how it is possible to improve and enhance classification (improvement and enhancement).

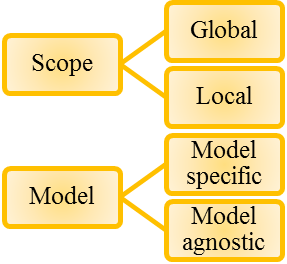
**ML MODEL BLACK BOX**

**Y**

Evaluation metrics

Interpretation

1. Explainable and interpretable of ML models



1. Two main types of Explainable and interpretable approach

Interpretability

Liner regression

Decision trees

K-nearest neighbors

Deep neural network

Support vector machines

Random forest

Accuracy

1. Reverse relationship between interpretability and accuracy of ML models

**Information**

Interpretability methods

**Extract**

**Learn**

Black box model

**Capture**

World

Data

Humans



1. The conceptual frame work to capture knowledge from the world using models

In LIME, it is desirable that artificial psychology perceives the relationship between the features of a specific sample and the predictions of that model by training a more explainable model such as a linear model driven from samples that are slightly different from the original inputs.

An explanation can be extracted from the coefficients of features that exceed a threshold in a linear model. The intuitive logic behind this reasoning is that these features of the linear model have the greatest importance in explaining and predicting that model. As a result, these local examples can be employed to argue the contributions of every feature in the resultant explanation and prediction. In other words, LIME creates the new datasets that include perturbed samples. Hence, LIME trains an interpretable model on these new data. This model is then weighted through the proximity of sampled instances taken from the analysis samples.

A trainable model might be a good approximation of local predictions for machine learning; however, it may not necessarily have a global fit.

Another new method of explaining the predictions of a deep neural network is to use shapley additive explanations abbreviated to SHAP.

In fact, SHAP is a novel method for making individual predictions obtained from a complicated model. The goal of SHAP is to calculate the share of each feature in prediction to determine the effect of each input. The explanation of the technique of SHAP is rooted in the cooperative game theory, the principles of which are used to calculate shapley values.

As the game theory aims to analyze how the coalitions of some players affect the results, SHAP uses the same method to determine how the features contribute to the model outputs. In the game theory, specific players have key roles in the results. Likewise, some features play major roles in the model prediction in artificial neural networks; therefore, they are more important.

As mentioned previously, shapley values are calculated in SHAP. This is the average metric for all marginal contributions for all coalitions. In SHAP, the feature-related data act as the members of a coalition in an instance, and the explanations of shapely values are expressed as a form of an additive approach to a linear model.

Not only do shapley values determine the relevance of a feature, but they also indicate whether that feature has a positive or negative effect on a prediction.

## 6.17. Difference between LIME and SHAP

In LIME, a set of important features is obtained. However, those features are not exactly linked to their roles in the model output. In other words, LIME cannot determine the attribution of the exact amount of stress in predicting the pain severity in fibromyalgia. It can merely indicate whether a feature is important in predicting the pain severity of patients with fibromyalgia.

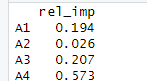
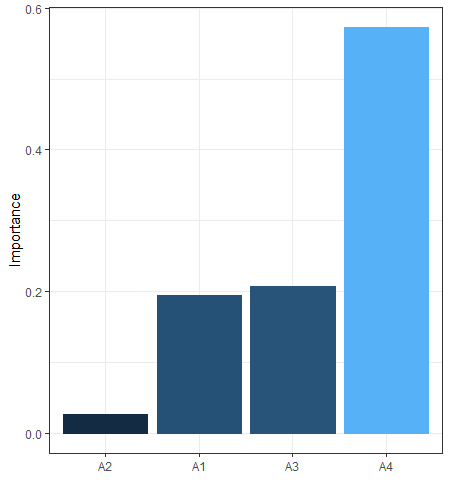
However, an output is attributed to a combination of measured features in SHAP. In fact, SHAP can be considered the statistical attribution of a model output to a set of inputs. For instance, 4 degrees of the total pain severity are predicted out of 10 degrees and then attributed to a 20-degree stress severity in these patients.

### 6.17.1. Practical example using R

**Example 2**

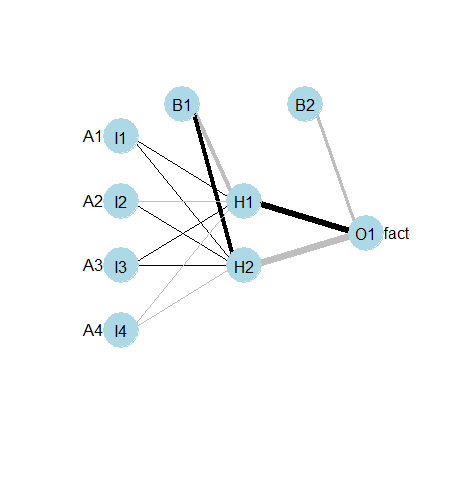
An artificial psychologist seeks to predict the probability of group membership using 4 emotional variables (A1, A2, A3, and A4). He uses a Multilayer Perceptron Artificial Neural Network.

Since there is one hidden layer in the model this example is not classified as deep learning. For this purpose, he examined 194 10-12-year-old elementary school boys, 97 of whom had high hyperactivity symptoms and 97 who had low hyperactivity symptoms. He measures a child's stress level = A1, child's anxiety = A2, mother's stress = A3, mother's anxiety = A4. Then he considers 80% of the total sample as the training sample and 20% as the test sample group. He also considers the execution times once for convenience. Also, he uses the CV method to validate the model and, if necessary, changes the hyper parameter of the model to find the best model. He has used the grid search method for tuning his model. Also, size and decay must be changed in the model. Size is the number of hidden layer nodes and the decay parameter controls the amount of decay to change the weights. The weights are changed in back-propagation models. In this case, it is said that the model learns. The decay parameter takes values from 0.001 to 0.1 depending on the model. He has considered the amount of decay in the range of 0 to 0.1. Weights in an artificial neural network are similar to regression coefficients. Using the Garson algorithm, the relative importance of each variable in the prediction is determined. The process of Pooling and Scaling of all the weights related to a variable in the model creates a Single Value that is in the range of 0 to 1 to assess the relative importance of that variable in prediction.



1. Importance plot of MLP

Accordingly, the graph from the Garson algorithm shows in order of decreasing importance A4, A3, A1 and A2 are the most important predictor variables. (rel-imp column in. (Figure 6.31).



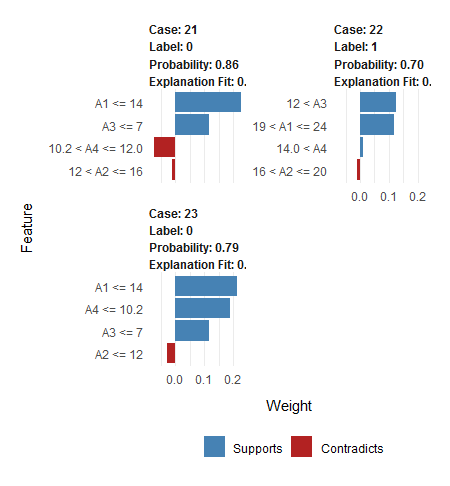
1. Net plot of MLP

Figure 6.32 shows the inputs in the net plot are A1 to A4 which has a hidden back layer with two neurons. Fact or Factor is the output variable (group membership). He used LIME to explain the model.

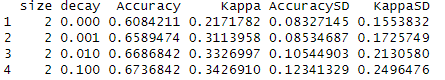
LIME is an interpretability agnostic method. Agnostic, which is related to the Black Box concept, refers to the fact that it can be used to generate insight into a process that is unknown or cannot be known. This method is used to interpret complex models. This method creates a qualitative link between input and response variables (group membership). In the LIME algorithm, the explain () function is used, which is used for new observations (such as children number 21 to 23). In this diagram, Case refers to the person's number in the data set, and Label refers to the observed value of the person on the target variable. Probability refers to the predicted probability for that label, and the explanation fit index measures the quality of the model used in the explanation.

The features (variables) marked in blue in the plot are the variables that support their label, and the length of the bar in the diagram indicates the weight ratio of a characteristic. A facetted heat map style plot shows the feature combinations for cases 21 to 23.

The individual number can be seen on the horizontal axis (X) and the features that have been categorized can be seen on the vertical axis (Y). The results of running the model based on size and decay are shown in the Figure 6.34).

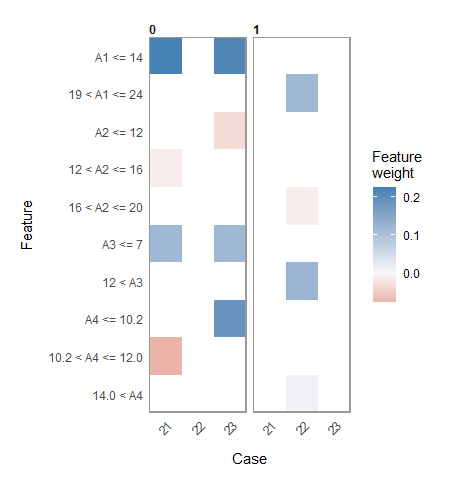


1. LIME for case 21-23



1. Size and decay of the model

As can be seen, with a size of 2.0 and decay equal to 0.1, the accuracy of the model is maximised and equal to 0.674.

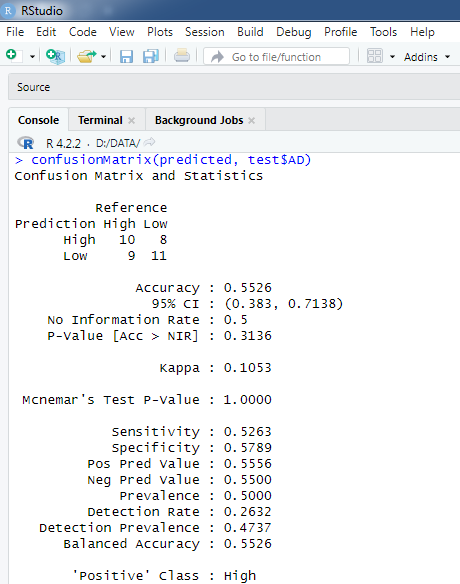


1. Overall LIME for the cases and feature weight

Figure 6.35 is a facetted heatmap-style visualization of all case-feature combinations for 3 selected cases. The case numbers are shown in the horizontal axis and categorized features are shown on the vertical axis.

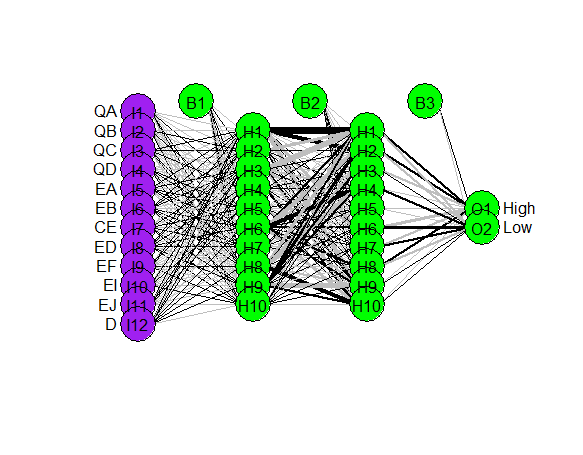
**Example 3**

An artificial cyber psychologist tries to predict the tendency to cyber addiction (low-high) based on 13 early maladaptive schemas that he thinks are effective based on the background of the research. Since the criterion variable or Target is binary, he is trying to predict group membership (categorization). He fits a Multilayer Perceptron Artificial Neural Network (MLP) which uses 2 hidden layers. He considers each hidden layer to consist of 10 neurons. The number of hidden layers and their number of neurons is a Hyper Parameter. By increasing the number of hidden layers and the number of neurons, the accuracy of the prediction model (Classification Regression) increases, but the model becomes complicated and its execution time becomes longer. The R codes are in the Listing. In this model, high group is considered as code 1. First, the quantitative predictor variables (which are all quantitative in this example) are standardized, and the Cyber addiction variable, marked with the AD symbol, is set as the Criterion or Target variable. To implement the model, the researcher will need the necessary specifications in NE.net.. Of course, these specifications can be changed. A Confusion matrix is used to check classification accuracy. The R output of this analysis is shown in figure number 6.37.



1. The R output of deep MLP classifier

In this figure, it can be seen that the accuracy of the model is equal to 0.55, the p-value is equal to 0.31, the kappa coefficient is equal to 0.10, and the sensitivity is equal to 0.53 with the specificity equal to 0.58, which is indicative of low model accuracy. .



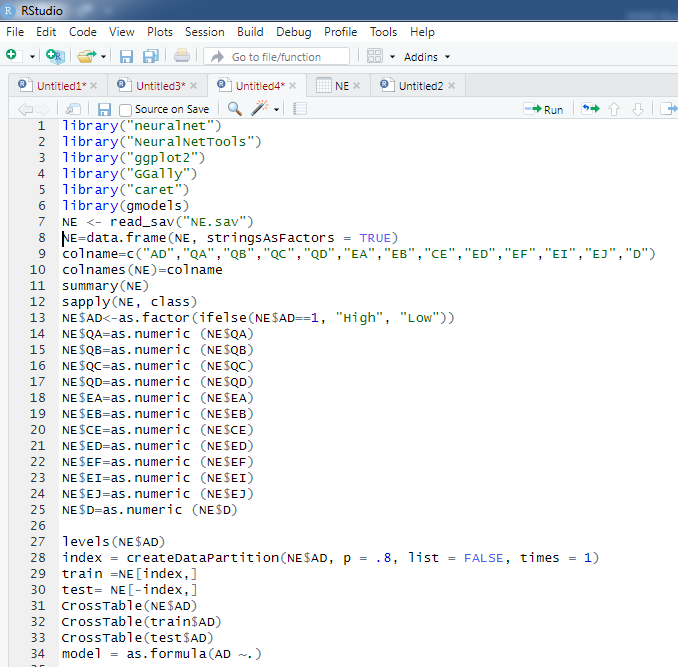
1. Net plot

Figure 6.37 shows the Network plot. The number of neurons in the input layer is equal to the number of predictor variables, i.e. 12 variables (the same 12 primary maladaptive schemas that the researcher identified among the primary maladaptive schemas most related to cyber addiction). The number of hidden layers is 2 layers each having 10 neurons, and finally we have the Output layer, which is the tendency to cyber addiction (high or low).

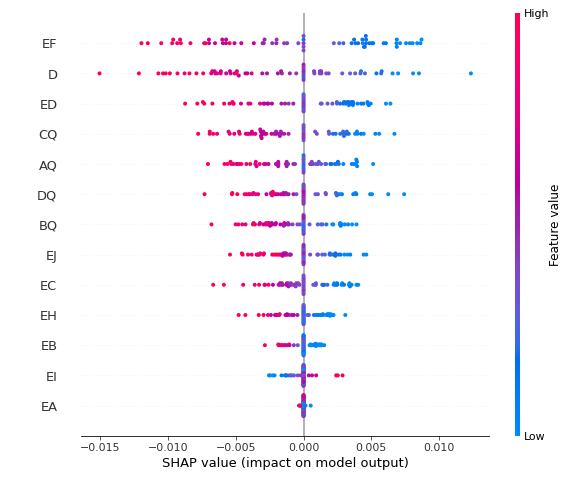
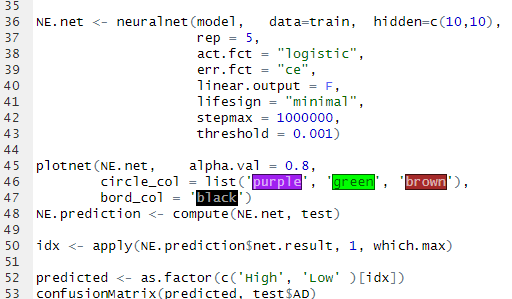
As discussed in detail in this chapter. Interpretability and explainability are two very important features that should be taken into account when using these models. The SHAP algorithm has been used for this purpose. In this figure, it can be seen that each point is based on the test sample. In the next example, we will examine this issue in a clearer way.

As mentioned, the psychologist uses a multilayer perceptron neural network with 2 hidden layers for classification. For this purpose, out of 194 participants, he selects 155 people (80 percent) as a training sample and 39 people (around 20 percent) as a test sample (listing 3).

1. R codes for deep MLP

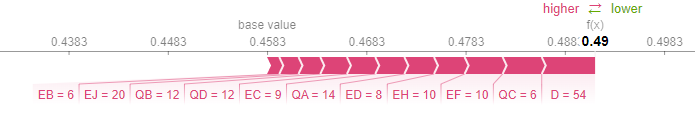


1. (continued) R codes for deep MLP



1. SHAP values plot of deep MLP classifier

Each point of every row is a record of the test dataset. The features are sorted from the most important (top) to the less important. We can see that EF is the most important feature. The higher and more positive the SHAP value of this feature, the more influential is the impact on classification accuracy of the target. The more negative this value, the less importance the variable has on classification accuracy.



1. The force plot of the model

A very useful plot we can draw is called a force plot. The force plot is an explanation of feature importance based on the SHAP values. The force plot shows the influence of each feature on the current prediction. Values in red can be considered as the values that have a positive influence on the prediction pushing the values higher than the average value across all the cases without any variables in the model (baseline) whereas values in green have a negative influence on the prediction pushing the prediction lower than the baseline prediction.

