

Machine learning is a generic term for the "artificial" generation of knowledge from experience: An artificial system learns from examples and can generalise these after the learning phase has ended.

So far, FlexFinance offers the use of results generated in neuronal networks for:

- Credit risk adjusted pricing
- Risk provisioning
- Early warning system

There is a two-step application of neuronal networks covered by FlexFinance:

1. Training of the neuronal network at portfolio level
2. Application of the trained neuronal network to an individual deal

Actually, machine learning provided by FlexFinance is focussed on the calculation of Probability of Default (PD), Loss Given Default (LGD) and Expected Credit Lossess (ECL). PD and LGD can be used as input into a method to derive the credit spread (CS).

The following topics are covered for each neuronal network:

- Input parameters (data considered during the training of the neuronal net)
- Setup and training of the neuronal net
- Output, e.g. early warnings over the course of time

Input parameters for the neuronal network are organised in data marts. Data is key and entails

- Customer DNA (profession, number of children, property owned etc.)
- History of deals and related payments (e.g. DPD, open payments at specific reference dates)
- Macroeconomic parameters
- Current contractual deal data (derivation of exposure at default)

For the neuronal networks,

- **Parameters** for calculation can be defined

The screenshot shows the 'Set & Train Deep Learning Network' interface. It features a teal header with a search bar and navigation icons. Below the header, there are tabs for 'Setting' and 'Chart'. The 'Setting' tab is active, displaying a table of parameters for the neural network setup. The table has columns for 'Label', 'Value', and 'Description'. A 'Start training of neural net' button is visible in the top right corner of the settings area. The footer of the interface indicates '© FERNBACH Financial Software'.

Label	Value	Description
Weight Initialization	Xavier	Weights must be initialized from scratch.
Epochs	20	An epoch is defined as a full pass of the data set.
BatchSize	1000	Number of training data sets used for one pass.
Minibatch Size	32	A minibatch refers to the number of examples used at a time, when computing gradients and parameter updates.
Learning Rate	0.001	The "speed" of learning.
Hidden Layers Activation Function	Relu	Activation function used in Hidden Layers.
Output Layer Activation Functions	Softmax	Activation function used in Output Layer.
Regularization	L2	Regularization methods can help to avoid overfitting during training.
Updater	Nesterov	The optimization algorithm is how updates are made, given the gradient.
Gradient Normalization	On	When training a neural network, it can sometimes be helpful to apply gradient normalization, to avoid the gradients being too large (the so-called exploding gradient problem, common in recurrent neural networks) or too small.
Number of Hidden Layers	3	Define the depth of the neural network.
Number of Neurons per hidden Layer	13	Width of hidden layers. Defaults to number of features (input layer size).

Figure: Parameter settings of the neuronal network

- **Training** of the neuronal net can be started for the current posting date, using the latest run of data feed for the underlying data marts. For each training iteration, a test against the training

group is performed. The test compares the prediction made on the basis of the training portfolio with the real figure in the test portfolio. The graph below shows the training process: For the training of neuronal networks, the portfolio will be split into two groups of deals:

- Training
- Testing

Beside others, the number of iterations "Training ..> Testing Training Testing ..." can be configured as parameters for the neuronal network.



Figure: Model score versus iteration

- **Influence of individual attributes (weight in %) can be viewed** as result of the training of the neuronal network. It can be seen which parameters and which weighting have been included in the ECL calculation for the entire portfolio of assets.



Figure: Training of the neuronal network for ECL calculation

Machine learning uses the [performance database](#) as the basis for the early warning system.