Terminology:

NN – Neural network

Project – is a NN Description, which contains NN Structure Object, NN Training Object. Can contain multiple NN Models.

NN Description – contains general information about NN and chose NN Paradigm.

NN Structure Object – structure of NN. Amount of input, hidden, output layers and these connection.

NN Training Object - contains information about input fields, which has to be filled out in order to train a NN. It also contains information about output fields in order to perform testing.

NN Model – trained NN. Parameters were taken from NN Training Object, values are adjusted by creator.

Project initialization from existing paradigm.

Every project describes particular problem. User can choose some developed NN paradigm, that is exists in N2Sky. This paradigm should correspond to project problematic and helps to analyze and solve it.

User has to go to project creation page view on N2Sky and follow the next steps:

1. Create a NN Description.

User has to fill out mandatory information about a project:

* Project name
* Project description

The next step is choosing developed paradigm, which is available on N2Sky. Lets assume, that user choose the Backpropagation paradigm. After choosing it user will get predefined fields. Values of these fields can be adjusted in order to create desired network.

For Backpropagation case following fields will appear on the page:

* Network type will be filled out automatically. Cannot be adjusted.
* Problem type will be filled out automatically. Cannot be adjusted.
* Domain description will be filled out automatically. Can be adjusted.
* Propagation Type will be filled out automatically. Can be choose from drop down menu. Lets assume, that user choose “feedforward” type.
* Learning type will be filled out automatically. Can be changed.
* Creator information will be filled automatically. Adjustment is not possible.

1. Create a NN Structure Object.

After submitting NN Description user will be redirected to creation of NN Structure Object page view. For our Backpropagation network case default values will be filled out. User can adjust next fields:

* Number of inputs
* Number of hidden layers
* Number of outputs
* Connection type

Visual representation of NN with adjusted values will be shown.

1. Create NN training object.

Since user choose Backpropagation paradigm the training object namely input fields will be automatically created. User can adjust a default values, which later will be user be creation of NN Model.

For Backpropagation following fields will appear on the page.

Input Parameters:

* Bias Input
* Bias Hidden
* Activation function
* Momentum
* Epochs
* Learning Gate
* Threshold

Output Parameters

* Net output

As soon as these three steps are completed, the project considers to be initiated.

1. Create a NN Models

Since project is initialized user can create multiple NN Models and train them.

User will be redirected to creation of NN Models page view where of current project where the fields from NN Training Object will appear. User can adjust a default values and train the NN in order to create a NN Model.

For NN Model user has to give a NN Model name and adjust a default input values:

* Bias Input – 1
* Bias Hidden – 1
* Activation function – sigmoid (choose from dropdown menu)
* Momentum – 0.1
* Epochs - 1000
* Learning Gate – 0.4
* Threshold - 0.00001

As soon as user submitted the form above NN will be trained and NN Model will be created. User will see the training process of every epoch, the errors and weights.

User can create multiple NN Models with different input values.

1. Test a Model

User has to choose a trained NN Model to perform tests. User will b redirected to Model Tests page view. The output fields with a default values from NN Model will be represented on the page.

User can perform multiple tests on his model. User can decide either to keep the NN Model or to delete it. If user decide to keep the NN Model he can publish it so that other users can perform tests of copy following model and adjust some parameters.