New N2Sky Architecture

2.3. New N2Sky Architecture

Contemporary User Interface

User-centered design is a fundamental requirement for N2Sky. Looking back on past experience with an application it was identified the real users capabilities and needs. N2Sky was moved from complex expert system to system that is understandable for every interested user without having deep knowledge in neural network field. The goal was to save and gain current functionality of an application and decrease visual complexity of it.

N2Sky today is a cross-platform handy application with a responsive design. The frontend is written on ReactJS framework and it is convertible to React-Native framework. The application is accessible as well from desktop computer, as from mobile devices and other devices with any operational system. Further backend has microservices architecture to support scalability. Every microservice is developed on NodeJS server, which remains to be efficient and lightweight. This architecture let users freely and easily to work with an application without interruptions or waiting until it completely loaded.

The central concept of application is to support Software as a Service (SaaS) and Platform as a Service (PaaS) distribution. N2Sky consist of two modules:

* Administration module
* Main application module.

Administration module allows the system administrator control an environment. The module supports OpenStack and Cloudify monitoring and managing throw application dashboard. It is also contains custom monitoring and alerting management system, which can be installed to any server within N2Sky user interface. Administration module implements PaaS. It is fully configurable and wrapped into open source project in order to make module accessible to third-party applications.

Main application module is the central module of N2Sky. Within this module users can use, train and test existing neural networks, reuse neural network paradigm or even create and deploy own neural network on N2Sky environment. Module services are supporting SaaS distribution. Experts can use an application directly throw N2Sky API or they can integrate N2Sky services to their own application.

In order to make N2Sky user interface as well understandable for arbitrary users as professional for advances users it was decided to separate the user roles with a different way to interact with an application.

Contributor is an arbitrary user. This is does not has to have a deep knowledge in neural network field or know any programming language. The main goal of arbitrary user is to study neural networks within N2Sky. Contributor has access only to own dashboard and other public available dashboards on main application module. Contributor can perform sematic search for available neural network paradigms and use it. He can train neural network objects and test them. This user also can share his trained neural network making it public.

Developer is an expert user, which has enough knowledge and experience to create his own neural network. This user can create neural network paradigm using ViNNSL schema and publish it on N2Sky. Developer can deploy neural network as well on N2Sky environment as on his own environment and provide training and testing endpoints. The goal of developer to study how his network behaves with a different neural network structure, input parameters and training data that was provided by other users.

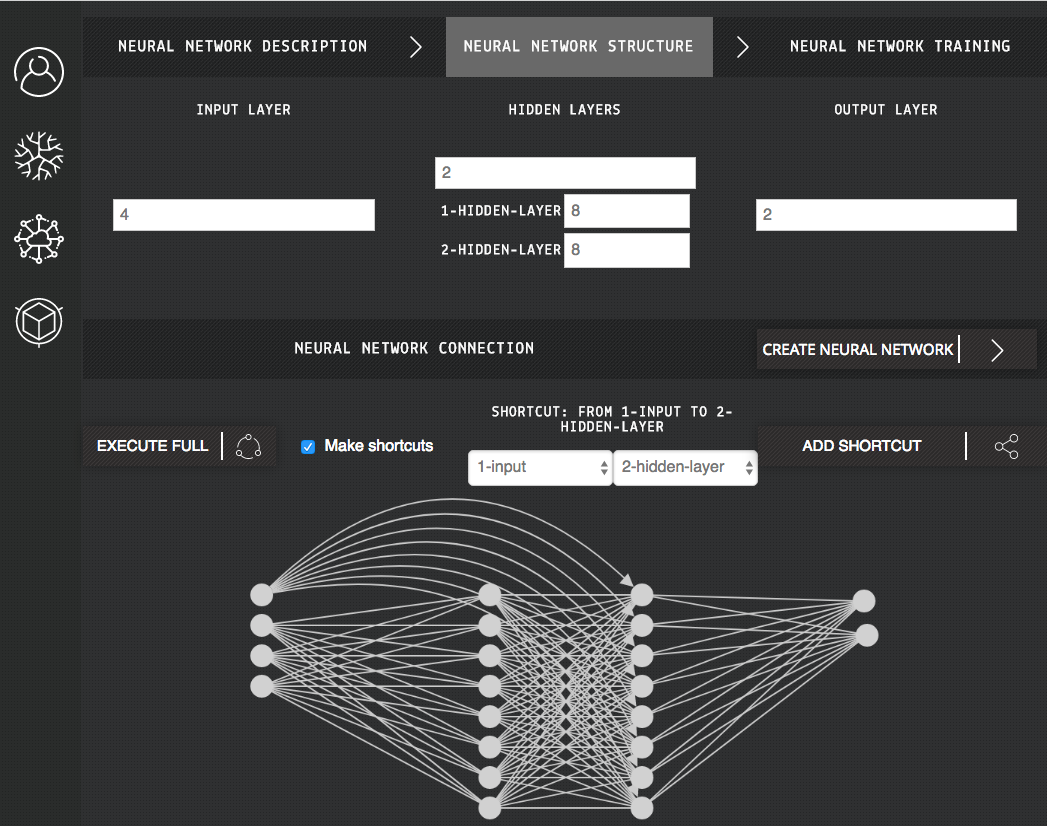
Use Cases

Solving any problem as a Contributor and as a Developer is differs. When Contributor needs some easy step-by-step workflow, the Developer wants to perform procedure fast without destructions. There are only few pre-requirements, which will remain the same: to be authenticated within N2Sky and create first project from N2Sky dashboard, which describe your problem field.

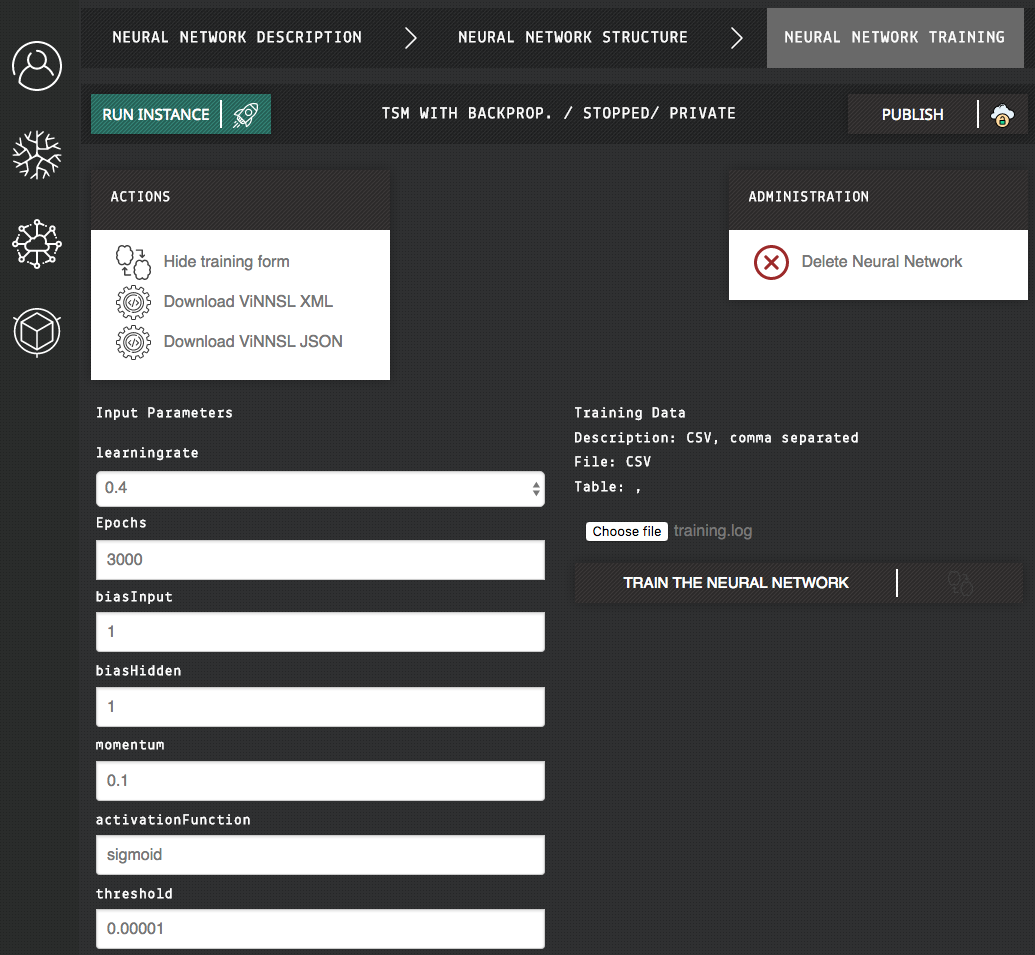
Use case: Solving Problem with N2Sky as a Contributor

After creating a project you have few possibilities to find a neural network which corresponds your needs: create neural network from paradigm or perform sematic search in order to find already available neural networks.

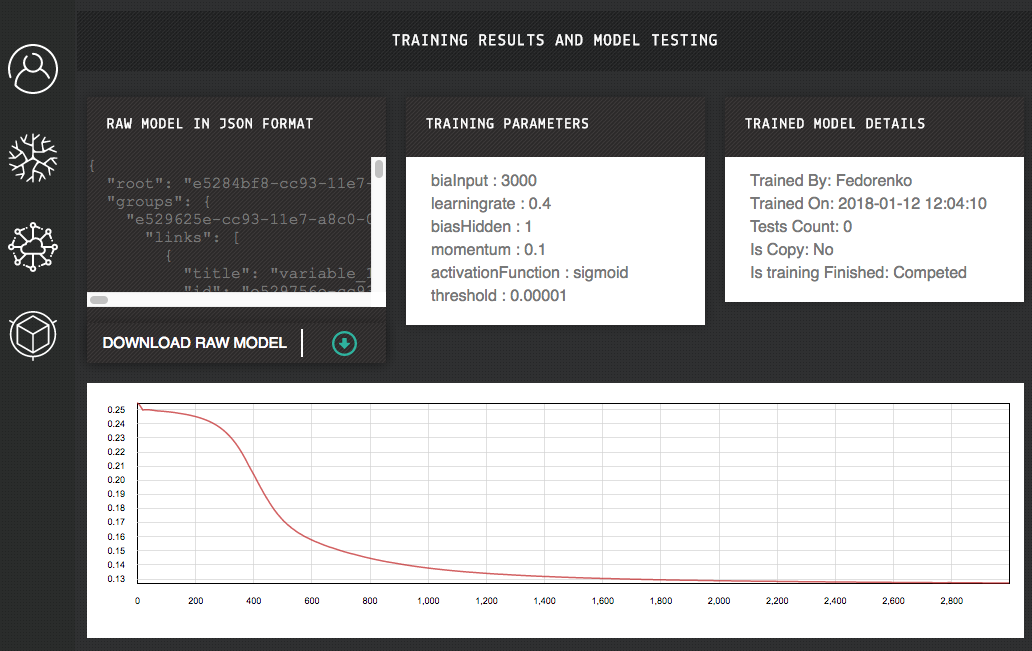
To create neural network from paradigm you will need to choose wanted paradigm, in case of [SCREENSHOT] is Backpropagation, fill out network description form and set the network structure [SCREENSHOT]. User can add shortcut connections and make some customized network. He does not have to worry about correctness of the chose structure because an application will help him to finish workflow correctly.



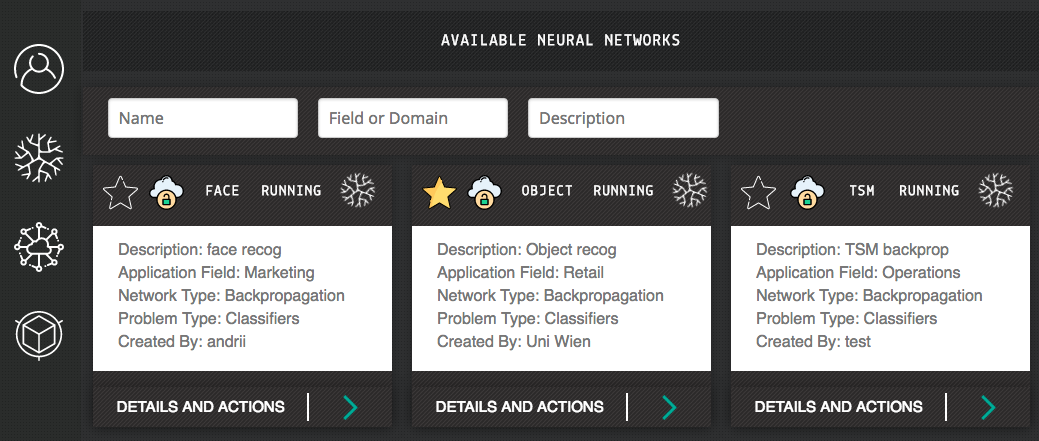
After creating neural network user will be redirected to training view [SCREENSHOT]. From here he can run instance on N2Sky, publish neural network to make it available for rest users, download ViNNSL schema format in case if he want to study it and most importantly the user can perform training. Since neural network was created from paradigm default parameters for training are set, which helps user in further actions.



Training could take a while, but user will not be blocked. Contributor can observe every training evaluation. After training is completed user can perform testing[SCREENSHOT]. It is also possible to publish trained model so that other users could test it with own data.



It is also possible to reuse already existing neural network. The user can in neural network repertory [SCREENSHOT] and copy needed neural network to his own project. User can perform training and testing on copied networks.



Use case: Solving Problem with N2Sky as a Developer

Developer is an export, he can download ViNNSL schema template and fill it up and customize his neural network as he wish. After creating the project user can just upload his ViNNSL formatted paradigm and deploy.