# ModelStore—A Lightweight System for Storing Neural Network Models

I. Kastanis, M. Höchemer, E. Ntavelis, P. Purwar, P. A. E. Schmid

*ModelStore is an easy to use framework for storing and managing neural network models. It offers extensibility, portability, traceability, and reproducibility. Minimal effort is required for storing a model making it an optimal solution for integration into existing machine learning based pipelines. It enables users to keep track of multiple models with practical overviews of the most important parameters, metrics, and results. It is a fundamental building block for any life-long learning application by managing model specific data upon which automatic decision support systems can be developed.*

Training machine learning models is currently a laborious and manual task. The storage and management of the different instances of models is often left at the hands of the expert operator and typically lacks standardization and structure. While this works for brief periods of time, in the long run it becomes intractable to retrieve information about the model origins, its performance and training parameters. The often-used file and folder naming schemes fail to capture sufficient information about the model, are cumbersome to extend and do not provide informative overviews of the trained models.

ModelStore overcomes the aforementioned issues by standardizing the storage of models by managing not only the instances of the trained models but also their linked data.

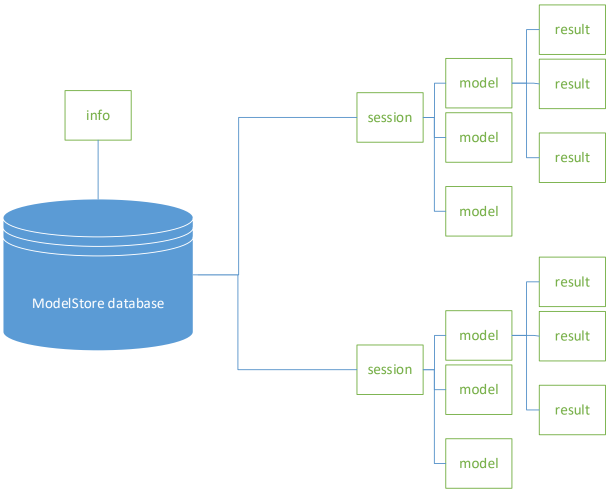

*Figure 1: ModelStore database architecture.*

The ModelStore consists of multiple layers. In the lowest level is a SQLite [1] database with the architecture shown in Figure 1. On top of this a Python layer, encapsulating the most common machine learning frameworks (keras, tensorflow and Pytorch), offers functionality to push and pull models into the database with simple to use functions overcoming the need to write lengthy queries. Finally, a C# layer with graphical user interfaces can be used for deployment purposes.

The modular architecture allows users to easily extend the framework to fit their specific requirements. In practice SQLite queries are the common interface that allows implementation in any programming language. The SQLite database is self-contained and cross-platform. Porting means copying a single file. The framework generates a unique identifier for each model by hashing it. In this manner each model can be identified, and its origins traced, even in the case of stray models that are no longer inside the database. To be able to reproduce a model the source code that was used for training is referenced in ModelStore by integrating the version control software Git [2]. The framework is capable of automatically inserting the latest commit ID in the database, which can then be used to retrieve the complete source code that was used for generating the model.

ModelStore has been integrated in CSEM's Visard [3] platform that enables the design and execution of complete automation systems according to Industry 4.0 design principles. Small batch highly customized production lines can utilize Visard's advanced machine learning capabilities to efficiently introduce product variations and decrease integration time.

ModelStore is a framework designed for both development and deployment purposes. It is an invaluable tool during development, in particular for large teams of developers facilitating effortless knowledge exchange in the area of machine learning. Unused models can be stripped from a database for the purposes of deployment, while at the same retaining all the descriptive information required.

Life-long learning approaches present the true potential of machine learning. They enable dynamic solutions that can adapt to changing conditions. Such methods pave the road towards more advanced types of artificial intelligence. Currently such adaptive machine learning solutions that can incrementally learn on dynamic environments are rare, and mainly restricted in research environments. ModelStore offers a foundation for gathering models as well as related descriptive data in a systematic and efficient manner. Upon this foundation decision support systems can be built that determine the right time to train a new model and assess its performance in comparison with other instances of that model, while at same time being able to trace all of these steps.

[1] https://www.sqlite.org

[2] https://git-scm.com/

[3] M. Höchemer, I. Kastanis, P. A. Schmid, "VISARD —Vision Automation Robotics Designer", CSEM Scientific and Technical Report (2016) 95.