Information Technology Course
Module Software Engineering
by Damir Dobric / Andreas Pech

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

# Implement Temporal Memory Learning sample with Serialization

**Hung Bui Tran Hai**
hung.buitranhai@stud.fra-uas

**Tanu Agarwal**
tanu.agarwal@stud.fra-uas.de

# INTRODUCTION

- In this project, serialization with Temporal Memory is implemented in Multi Sequence Learning.
- The state after prediction can be stored using the serialization technique.
- Additionally, prediction result comparison is done between the normal and serialized predictor.
- Therefore, the outline of this project can mainly be described in two parts as follows:
  - Implementing Serialize(), and Deserialize() methods in the Predictor class to serialize and deserialize predictor instances containing Connections, CortexLayer, and HtmClassifier properties.
  - Comparing predictions made using the standard and serialized predictor to validate Serialization and Deserialization functionality.

# METHODS

- **Serialization** involves transforming an object into a format suitable for streaming, enabling it to be stored in a file, database, or memory. The object is serialized to a stream, which carries not just the data, but information about the object's type.

- **Deserialization** is the counterpart of Serialization and involves retrieving a serialized object for future use. Essentially, it restores the object's state by configuring properties, fields, and other relevant attributes.

# METHODS

- To implement serialization with TM for the MSL project, we need at first make the “Predictor” which is the output of MSL become serializable.
- The Predictor class inherits the interface ISerializable which defines serialization methods.
- The “Predictor” which is the output of MSL must be serializable. Thus, Serialize() and Deserialize() methods are implemented in the Predictor class.
- However, methods for the serialization of objects which are in the Predictor layers are required.

# METHODS

- Multi Sequence Learning (MSL) experiment demonstrates how to learn two sequences and how to use the prediction mechanism through the RunMultiSequenceSerializationExperiment() method.
- For testing purposes, we defined two instances of class Predictor i.e., "predictor" for normal predictor and "serializedPredictor“ results after serialization and deserialization of Predictor.
- We compare the prediction output from the "predictor" and "serializedPredictor" instances. The same prediction (5, 4, 2) was made by both the normal predictor and the serialized predictor with the same accuracy.
- This proves that the serialization approach appears to be correct since the output from both predictors is the same.

# RESULT

- Implementation of serialization with Temporal Memory in the Multi Sequence Learning (MSL) project is successful. The methods (Serialize(), Deserialize(), Save(), and Load()) are implemented in the Predictor class.

- Methods for serialization and deserialization of Predictor properties are also implemented which are HtmClassifer.Serialize(), HtmClassifier.Deserialize(), CortexLayer.Serialize(), …

- An MSL example has been made to validate the use of Serialization in the MSL project.

- Predictions were executed using both the standard predictor and the serialized predictor. The results of predictions from both predictors were scrutinized and found to be identical.

# DISCUSSION

- Serialization with Temporal Memory in Multi Sequence Learning is implemented successfully.
- The properties of the Predictor class and the Predictor instance are successfully serialized.
- The project is now serializable, and we can store the trained model and use it for the upcoming training.
- By comparing predictions made using the standard predictor and the serialized predictor, this project proves that the serialization approach appears to be promising and reliable since the objects are serialized and deserialized accurately.
- However, it is realized that the implementation is not totally complete as serialization for the scalar encoder is not successful yet. Further, research is necessary to identify how to do serialization of scalar encoder with this approach.

# Thanks for your attention!