Improving Unit Tests

Homeostatic Plasticity Controller

NeoCortex API

- Inspired by the natural functioning of human brain, the Hierarchical Temporal Memory Cortical Learning Algorithm (HTM CLA) uses spatial pattern recognition and temporal sequence learning for intelligent operations such as classification. This behaviour of human brain is captured and modelled in the NeoCortexApi [1].
- This model of human brain can be implemented and used to perform many intelligent tasks. To train such a model HTM CLA can be utilized, HTM CLA consist of two algorithms: Spatial Pooler and Temporal Memory.
- NeoCortexApi is today built as .NET Standard 2.2 library. It contains all artifacts required to run Spatial Pooler, Temoral Memory and few encoders.

Spatial Pooler

- The neocortex is a set of layers of the mammalian cerebral cortex involved in higher-order brain functions such as sensory perception, cognition, generation of motor commands, spatial reasoning and language.
- Hierarchical temporal memory (HTM) Spatial Pooler (SP) algorithm provides a theoretical framework that models several key computational principles of the neocortex.
- Homeostatic Plasticity Controller and SP have also been modelled and implemented in NeoCortex API.

Homeostatic Plasticity Controller

- Spatial Pooler algorithm is augmented with the process of boosting and inhibition of mini-columns by using Homeostatic Plasticity Controller in the NeoCortex.

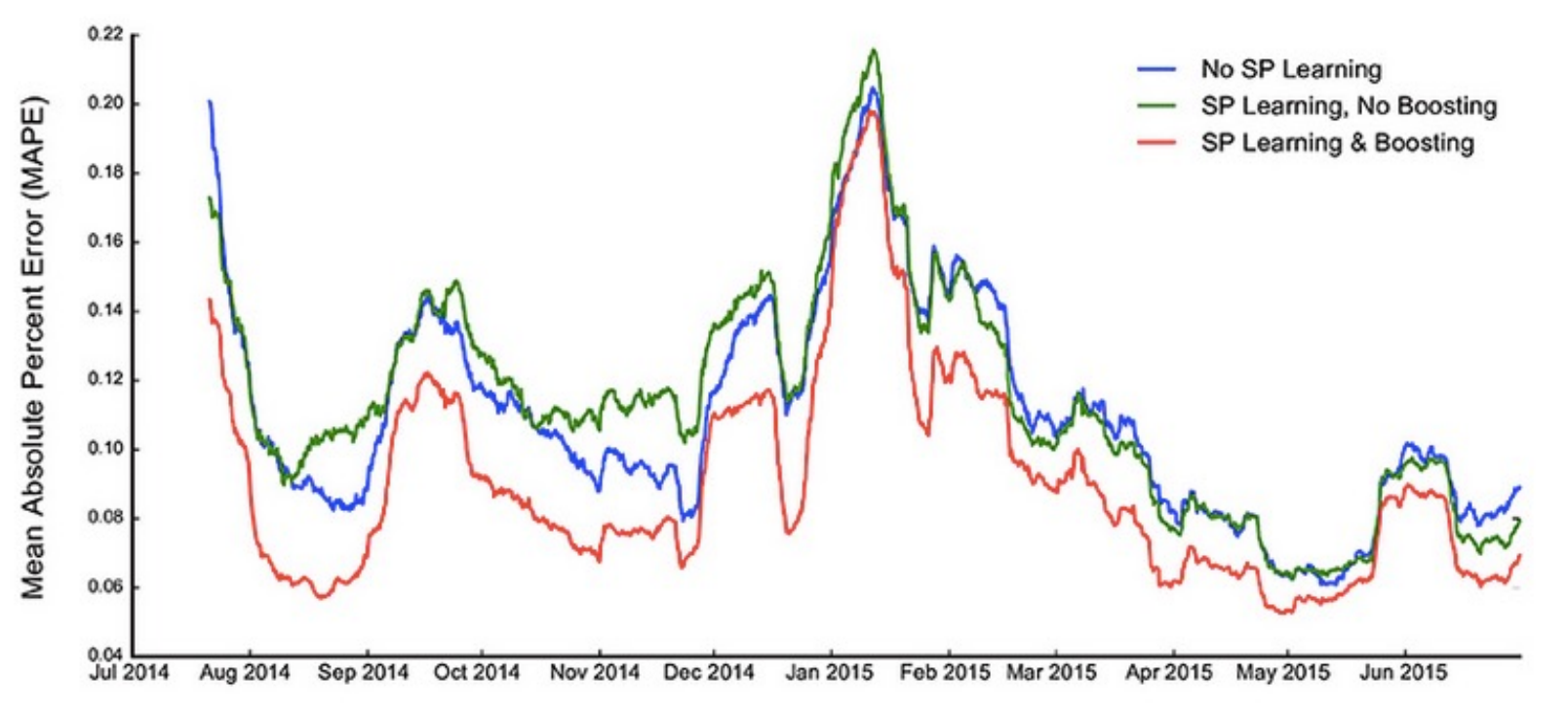

Figure 1: Comparision of error rates demonstrating impact of HPC. Source [2]

Unit Test Class

- As part of this project, a new Unit Test Class: HomeostaticPlasticityControllerTests was created inside namespace UnitTestsProject of the Neocortex API.
- This new Unit Test Class embodies a total of 14 Unit Tests covering wide variety of variations possible for the HomeostaticPlasticityController class.
- In certain Unit Tests where the arguments are defined at compile-time, an attribute called DataRow can be used to run multiple test cases with variety of data all of which test the same underlying code structure under different scenarios. This feature allows more rigorous testing without compromising readability of Unit Tests.

Unit Tests Summary

- CalcArraySimilarityTest
  - The method CalcArraySimilarity is designed to compare two arrays and calculate similarity among those arrays on a scale of 0.0 to 1.0. The Unit Test designed to test this method has 3 DataRow attributes, each one tests a unique case. The test CalcArraySimilarityOld2Test is designed along similar lines.
- GetHashTest
  - GetHash method is designed to take in an array of Integer values, converting each value into array of Bytes. Further SHA256 algorithm is used to get the encoded version of the array. The GetHashTest TestMethod tests this method for 2 different arrays of varying length.
- TraceStateTest
  - TraceState method of HPC class can be utilized to keep track of the no. of stable cycles achieved during the training phase. The Tests for this method validate the behaviour of the method TraceState under different circumstances.

Unit Tests Summary

- DeserializeTest
  - HPC class provides both the functions, Serialization and Deserialization which wrap the mentioned System.IO Classes. Using these functions, user can Serialize and Deserialize the HPC objects with ease. A HPC object goes through complete cycle in this Unit Test.
- ComputeTest
  - HPC Compute method is called when compute method of SpatialPooler class is executed. This method returns a Boolean value indicating if stable state is achieved in that particular compute cycle. Unit Tests for this methods cover both states: stable and unstable.
- EqualsTest
  - Equals methods in the HPC class was designed to compare two HPC objects and return a Boolean value indicating if the two object were identical. A total of 7 different assertions are made in the Unit Test designed for this method.

Conclusion

- Completeness of a Unit Test project can be judged from the code coverage metric achieved after the Unit Tests are run.
- An impressive 93% of code coverage has been achieved at the end of this project.
- This ensures that majority of the code base in the HPC class has been executed and validated at-least once when the tests inside Unit Test Class HomeostaticPlasticityControllerTests is run.
- There are is a wide variety of tools available in .NET environment to calculate the code coverage metric, most popular among them are two open-source projects: Coverlet [3] and ReportGenerator [4].

References

1. NeoCortex API: https://github.com/ddobric/neocortexapi
2. Figure 1 : https://www.frontiersin.org/articles/10.3389/fncom.2017.00111/full
3. Coverlet
4. ReportGenerator