# Introduction

List of projects for semester 2018/2019

ML Algorithms for Learning API

ML Pipeline Components for Learning API

# Implementation ML algorithms for LearningApi

*daenet* and *Frankfurt University of Applied Sciences* have implemented a simple Machine Learning framework in C# .NET Core called LearningAPI.

For more information see git repository: <https://github.com/UniversityOfAppliedSciencesFrankfurt/LearningApi>

The goal of projects described in this topic is implementation (extending of Learning API) of diverse algorithms. Every algorithm or module described here is a single project. Students can choose one of following projects as Software Engineering project or individual project.

## Implementation of Convolutional Network (ML 18/19-1.1)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | MATNR | Email | Path | State | Quality | Comment |
| TaimourYousuf  Dev Nair  Lakshmi Sudha Tanikella |  | [taimour.yousuf@hotmail.com](mailto:taimour.yousuf@hotmail.com)  [devan.nair@somaiya.edu](mailto:devan.nair@somaiya.edu)  lsudhatanikella@gmail.com |  |  |  |  |

The goal of this project is to implement Convolutional Network in C# .NET Core as module of *LearningAPI*. For this project two students can work as group. During design phase of the project, students will share tasks in repository.

For more information see: <https://github.com/cbovar/ConvNetSharp>

## Neural Network in .NET Core

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **MATNR** | **Email** | **Path** | **State** | **Quality** | **Comment** |
| **Firealem alemayehu** | 1225208 | Firealem.wosene@gmail.com |  |  |  |  |

Implementation of Deep Learning Neuronal Network Algorithm in .NET Core   
As reference implementation AForge library can be used.

## Evolutionary learning

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **MATNR** | **Email** | **Path** | **State** | **Quality** | **Comment** |
| Mohammad Nahin   Islam |  | sec.tabrez03@gmail.com |  |  |  |  |

## Support Vector Machine

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **MatNr** | **Mail** | **Path** | **State** | **Quality** | **Comment** |
| Tu Pham Thanh |  | [phamthanhtu261293@gmail.com](mailto:phamthanhtu261293@gmail.com) |  |  |  |  |

Students can also provide their own ideas to implement some other kind of algorithms. 

*References*

<http://www.aforgenet.com/framework/>

## Implement Naïve Bayes Inference Algorithm

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| Muhammad Usman |  | usmaan.awaan@gmail.com |  |  |  |

There are many different clustering algorithms. This is because the effectiveness of an algorithm depends to some extent on the characteristics of the data being clustered. The most popular algorithm is k-means clustering, which was already implemented on this university.   
Unfortunately, this algorithm is applicable only for numeric data items. In contrast, the clustering algorithm I’ll present in this article is based on a technique called Naive Bayes inference, which works with either categorical or numeric data.   
Based on [Learning API](https://github.com/UniversityOfAppliedSciencesFrankfurt/LearningApi.git), implement Naïve Bayes Inference algorithm. Additionally, figure out how to extend this algorithm to “Two Class Bayes Point Machine”.

*References*

Good source for implementation can be found here: <https://msdn.microsoft.com/magazine/jj991980>

## Deep belief network

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **MATNR** | **Email** | **Path** | **State** | **Quality** | **Comment** |
| Mohd Imranul Hoque  Limon  Akib Ibn Zahid | 1270224 | [limon1036@gmail.com](mailto:limon1036@gmail.com)  akibibnzahid@gmail.com |  |  |  |  |

Implement Deep belief network for LearningAPI. The goal is of the project is to implement the network for LearningAPI

*References* 

<https://en.wikipedia.org/wiki/Deep_belief_network>

<http://accord-framework.net/docs/html/T_Accord_Neuro_Networks_DeepBeliefNetwork.htm>

<https://github.com/primaryobjects/deep-learning>

## Haar Wavelet

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **MATNR** | **Email** | **Path** | **State** | **Quality** | **Comment** |
| Md. Suaib Islam | 1231014  1270224 | [suaib\_eee@outlook.com](mailto:suaib_eee@outlook.com) |  |  |  |  |
|  |  |  |  |  |  |  |

In [mathematics](https://en.wikipedia.org/wiki/Mathematics), a **wavelet series** is a representation of a [square-integrable](https://en.wikipedia.org/wiki/Square-integrable) ([real](https://en.wikipedia.org/wiki/Real_number)- or [complex](https://en.wikipedia.org/wiki/Complex_number)-valued) [function](https://en.wikipedia.org/wiki/Function_(mathematics)) by a certain [orthonormal](https://en.wikipedia.org/wiki/Orthonormal) [series](https://en.wikipedia.org/wiki/Series_(mathematics)) generated by a [wavelet](https://en.wikipedia.org/wiki/Wavelet). Nowadays, wavelet transformation is one of the most popular of the time-frequency-transformations

The goal of this project is to implement Wavelet transformation as module of [Learning API](https://github.com/UniversityOfAppliedSciencesFrankfurt/LearningApi.git). Implementation should provide various parameters and possibility to easily change existing transformation.

<https://en.wikipedia.org/wiki/Wavelet_transform>

<https://en.wikipedia.org/wiki/Haar_wavelet>

Haar Wavelet:   
<https://en.wikipedia.org/wiki/Haar_wavelet>

Haar Wavelet tutorial on pattern recognition:

<https://www.google.de/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwiWo9T21erQAhXI_ywKHSi7D0oQFggtMAA&url=https%3A%2F%2Fwww.researchgate.net%2Ffile.PostFileLoader.html%3Fid%3D54b4d915d685ccc6468b4652%26assetKey%3DAS%253A273675431940106%25401442260715381&usg=AFQjCNHQ-J_a6TJDoyhaaUfSfKTkr1MuoQ&sig2=sqCDtT7NZx7nYF1_KLmUgQ>

Example Discrete Haar Wavelet Transformation in C#:

<https://www.codeproject.com/articles/683663/discrete-haar-wavelet-transformation>

## Descision Trees

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **MATNR** | **Email** | **Path** | **State** | **Quality** | **Comment** |
| Sai Sudha | Murali | saisudhamurali02@gmail.com |  |  |  |  |

[Decision trees](http://en.wikipedia.org/wiki/Decision_tree_learning) are simple predictive models which map input attributes to a target value using simple conditional rules. Trees are commonly used in problems whose solutions must be readily understandable or explainable by humans, such as in [computer-aided diagnostics](http://en.wikipedia.org/wiki/Computer-aided_diagnosis) and [credit analysis](http://en.wikipedia.org/wiki/Consumer_credit_risk).

*References*: <http://crsouza.com/2012/01/04/decision-trees-in-c/>

## Ordinal regression

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** | **Comment** |
| Gideon Kwabla  Amewornu |  | gideonbenzene@gmail.com |  |  |  |  |

## Poisson regression

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** | **Comment** |
| Madhu  Bodige |  | madhuchacolate.shashi66@gmail.com |  |  |  |  |

## Fast forest quantile regression

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** | **Comment** |
| |  |  | | --- | --- | | Yash | Vyas | |  | [yash.vyas95@gmail.com](mailto:yash.vyas95@gmail.com) |  |  |  |  |

## Linear regression

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** | **Comment** |
| **Nalikala Srinivas** | 1063567 | srinivasnalikala@gmail.com | NalikalaSrinivas |  | 1 |  |

## Bayesian linear regression

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** | **Comment** |
| **Karthik**  **kasula** |  | karthikkasula1102@gmail.com |  |  |  |  |

## Neural network regression

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **MATNR** | **Email** | **Path** | **State** | **Quality** | **Comment** |
| ~~Asmade Kabore~~  Md.Toufi Zaman | 1253717 | [Asmade.kabore@gmail.com](mailto:Asmade.kabore@gmail.com)  md.toufiqzaman@gmail.com |  |  |  | DOES NOT MATH WITH EXCEL! |

## Decision forest regression

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| Sehrish fahim |  | sehrish.fahim91@gmail.com |  |  |  |

<https://www.codeproject.com/Articles/1217668/Sequence-Detection-With-a-Finite-State-Machine>

## Markov Model

Implement Markov Model as Markov Chain and Hidden Markov Model.

<https://en.wikipedia.org/wiki/Markov_model>

<https://github.com/chriscore/MarkovSharp/blob/master/MarkovSharp/GenericMarkov.cs>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **MATNR** | **Email** | **Path** | **State** | **Quality** | **Comment** |
| Md Equbal Hussain |  | [royalequbal@gmail.com](mailto:royalequbal@gmail.com) |  |  |  |  |
| Phuc Dinh Hoang |  | phucdh07@gmail.com |  |  |  |  |

## Recognition of handwriting signature

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **MATNR** | **Email** | **Path** | **State** | **Quality** | **Comment** |
| Prashant Pratik | 1271359 | [prash.pratik@gmail.com](mailto:prash.pratik@gmail.com) |  |  |  |  |

The K-Means algorithm has been extended on university with a function recognition algorithm compatible with LearningAPI.

Additionally, to algorithm, also an application ***MouseGestureRecognition*** has be created.

All required artefacts can be found here: <https://github.com/UniversityOfAppliedSciencesFrankfurt/LearningApi/tree/KMeans-branch/LearningApi>

Your task is to analyze existing application, which basically capture mouse pointer movements and collects coordinates. Set of coordinates is learning set of mouse points as array of two-dimensional vectors.

Your task is to analyze and document how feasible is recognition of signature (function) by using of two-dimensional mouse coordinates.

As next, you should extend existing application by adding a time as a third component in the vector.

Analyze and document algorithm with time approach and compare it with previous results.

## Kernel Logistic Rgression

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **MATNR** | **Email** | **Path** | **State** | **Quality** | **Comment** |
| Chandrima Biswas |  | Cbiswascse@gmail.com | Chandrima\_Biswas |  |  |  |

**Definition**

KLR is a kernel version of logistic regression, which is a well-known classification method in the field of machine learning. Basically kernel logistic Regression used to make binary prediction (False=0 or True=1) For example, KLR could predict if a person is male or female (male = 0, female = 1) based on predictor variables i.e. years of education (x1),annual income (x2).. KLR is an advanced variation of ordinary logistic regression.

**Difference between KLR and LR**

Logistic regression only applied for simple linearly separable data. On the other hand, Kernel Logistic regression can also manage the data which are non-linearly separable.

**Radial basis function (RBF)**

A kernel function called Radial basis function (RBF) used in KLR model which used to define the similarity between two vector or array. The RBF kernel function has a parameter called sigma. If the value of RBF(k) is more close to 0.0 indicate larger difference between two vector.

The equation for RBF is:

K(v1, v2) = exp( - || v1 - v2 ||^2 / (2 \* sigma^2) ).

**Methodology**

Suppose there are just four training data items:

td[0] = (2.0, 4.0, 0)

td[1] = (4.0, 1.0, 1)

td[2] = (5.0, 3.0, 0)

td[3] = (6.0, 7.0, 1)

Our goal is to predict the class label for x = (3.0, 5.0). Suppose the trained KLR model gave the alpha values and a bias of: alpha[0] = -0.3, alpha[1] = 0.4, alpha[2] = -0.2, alpha[3] =0.6, b = 0.1. and sigma value is 1.0.

The first step is to compute the RBF, similarity between the data item to predict each of the training items:

From the equation of RBF function a I have stated above:

|| td[0] - x ||^2 = (2-3)^2 + (4-5)^2

= 1 + 1

= 2

Next, divide the squared distance by 2 times sigma squared:

2 / (2 \* (1)^2) = 2/ 2 =1

Last, we take Euler’s number and raise it to the negative of the previous result:

K(td[0], x) = e^(1) = 0.3679

Similarly we have found the value for K(td[1], x), K(td[2], x) ,K(td[3], x)

K(td[0], x) = 0.3679

K(td[1], x) = 0.0002

K(td[2], x) = 0.0183

K(td[3], x) = 0.0015

Notice that at this point, x is most similar to td[0] and td[2], which both have class label 0. Now we have to calculate the sum of products of each K value and the associated alpha, and add the bias value:

z = (0.3679)(-0.3) + (0.0002)(0.4) + (0.0183)(-0.2) + (0.0015)(0.6) + 0.1

= -0.1120

Next calculate p = 1.0 / (1.0 + exp(-z)):

p = 1.0 / (1.0 + exp(0.1120))

= 0.4720

The probability denotes here as p. So the probability of the test data item has the class label=0.

Because if p is<0.5,the prediction is False (0) and if p is>0.5 ,the prediction is true(1)..

## Time-Series Regression

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **MATNR** | **Email** | **Path** | **State** | **Quality** | **Comment** |
| Md. Riajul Islam | 1270305 | riajulrony@yahoo.com |  |  |  |  |

Time-series regression is used to make predictions based on historical time data. It demonstrates how to perform this kind of analysis using rolling-window data combined with a neural network, which can lead to more accurate results. The idea is best explained by an example. The demo program analyzes the number of airline passengers who traveled each month between a certain time (years) intervals. The demo data has been included from different open sources on internet. The demo creates a neural network with four input nodes, 12 hidden processing nodes and a single output node. The number of input nodes corresponds to the number of predictors in the rolling window. The number of neural network hidden nodes must also be determined by trial and error, which is always true for neural networks. There’s just one output node because time series regression predicts one time unit ahead.

References : <https://msdn.microsoft.com/en-us/magazine/mt826350.aspx>

## Reinforcement Learning (Q algorithm)

Reinforcement Learning Algorithm in Learning API.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | MATNR | Email | Path | State | Quality | Comment |
| Aniket Sathe | 1230866 | Satheau692@gmail.com | Git:  https://daenet.visualstudio.com/\_git/SE-2017-2018?path=%2FSE%2FProjects%2FAniket%20Sathe%2FReinforcement%20Learning&version=GBAniketSathe | In progress  (Not passed yet) |  | Needs refactoring |

<https://sourceforge.net/projects/dotrl/>

## Implement [Anomaly Detection](https://github.com/UniversityOfAppliedSciencesFrankfurt/anomalydetection) with LearningApi (K-Mean Algorithm)

We have already implemented K-Mean algorithm in .Net core. You task will be integrate with LearningApi

Git: <https://github.com/UniversityOfAppliedSciencesFrankfurt/anomalydetection>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **MatNr** | **Email** | **Path** | **State** | **Quality** |
| Pretom Kumar Saha |  | pretomksaha@yahoo.com | <https://github.com/UniversityOfAppliedSciencesFrankfurt/se-dystsys-2018-2019-softwareengineering/tree/pretom_saha/Visual%20work/Project_2018/AnomalyDetectionK> |  |  |

## Thompson Sampling using C#

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **MatNr** | **Email** | **Path** | **State** | **Quality** |
| Pretom Kumar Saha |  | pretomksaha@yahoo.com | <https://github.com/UniversityOfAppliedSciencesFrankfurt/se-dystsys-2018-2019-softwareengineering/tree/pretom_saha/Visual%20work/Project_2018/AnomalyDetectionK> |  |  |

Thompson sampling is a clever algorithm that can help you estimate the relative effectiveness of a number of different possible choices

Reference:  (n.d.). Retrieved from <https://msdn.microsoft.com/magazine/mt829274?fbclid=IwAR1s1w1ZQMphMFlB2YQ-LOtL3JLA0WRoAxOK9P5Jr8bI1N6BD9-KWjxSEjw>

## Implement Gabor filter C#

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **MatNr** | **Email** | **Path** | **State** | **Quality** |
|  |  |  |  |  |  |

# Pipeline and other LearningAPI related projects

Pipeline components are not ML algorithms. They are processing components in the middle of pipeline. They can be used to load data, to filter data or to do some pre-calculations.

In this topic, all projects will be implemented by a single student (NO GROUP WORK), because pipeline modules are usually simpler task than implementation of a whole algorithm.

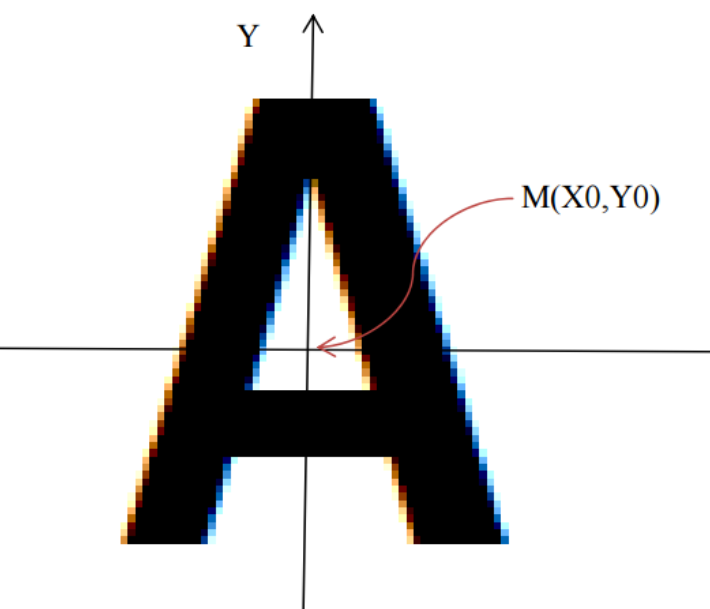
We have provided here many intersting ideas. However, we encourage you to provide you own ideas if you think, you could contribute on something more intersting.



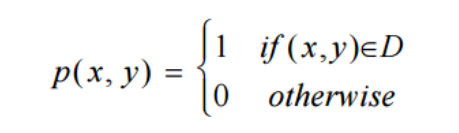
## Dimension Reduction by Ring Projection Pipeline Module

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | MATNR | Email | Path | State | Quality | Comment |
| Nhat Quang Bui |  | [buinhatquang31011996@gmail.com](mailto:buinhatquang31011996@gmail.com) |  |  |  |  |

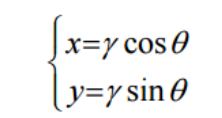
Implement Ring-Projected algorithm for Dimensionality Reduction as a module for Learning API. The module **RingProjectionModule** gets a double[][] array, which holds the coordinates of recognizing pattern.



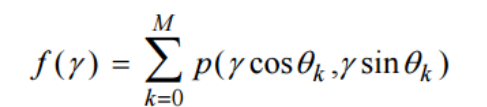
Algorithm should travers through all black points and transform them from



Into following form:



That means, module will calculate γ and θ for every black point. Result of a module calculation is a function f.



That means, module transforms 2-dimensional function into one dimensional function. Resulting type of module is double[].

1. Implement ring Algorithm.
2. Test algorithm with 2D pictures. Provide as result pictures and resulting functions
3. Test algorithm with 3D hologram triangles. Provide as result pictures and resulting functions

References:

<https://www.google.de/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=0ahUKEwjx3f_eoaPRAhVDKcAKHYrnBX4QFgg2MAI&url=https%3A%2F%2Fwww.researchgate.net%2Ffile.PostFileLoader.html%3Fid%3D54b4d915d685ccc6468b4652%26assetKey%3DAS%253A273675431940106%25401442260715381&usg=AFQjCNHQ-J_a6TJDoyhaaUfSfKTkr1MuoQ>

## Implement one Algorithm for dimensionality reduction

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | MATNR | Email | Path | State | Quality | Comment |
| TrinhTran |  | trinhtran2151995@gmail.com |  |  |  |  |

<https://en.wikipedia.org/wiki/Dimensionality_reduction>

## Gaussian and Mean Filters for noise removal in .NET Core

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** | **Comment** |
| Harshal Vaze |  | [harshalvaze11@gmail.com](mailto:harshalvaze11@gmail.com) |  |  |  |  |

Implement algorithm for removal of the noise from images based on Gaussian and Mean Filter. Algorithm should be implemented in .NET Core and MUST BE aligned to [Learning API](https://github.com/UniversityOfAppliedSciencesFrankfurt/LearningApi.git) open source project.    
Provide number of different images and apply both filters separately and in combination. At the end describe and explain results.

*References*:

<http://tech.it168.com/KnowledgeBase/Articles/4/6/5/46580ec2f2e98e8ab34ebddb650c3428.htm>

<https://www.cs.auckland.ac.nz/courses/compsci373s1c/PatricesLectures/Gaussian%20Filtering_1up.pdf>

<http://www.aforgenet.com/framework/>

<http://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm>

## Implement Drawing Module for LearningAPI

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **MATNR** | **Email** | **Path** | **State** | **Quality** | **Comment** |
| Sahiti  Medichetti |  | sahiti437@gmail.com |  |  |  |  |
| Anagha Sathaye | 1230895 | sathayeanagha@gmail.com | Git:  https://daenet.visualstudio.com/\_git/SE-2017-2018?path=%2FSE%2FProjects%2FAnagha%20Sathaye&version=GBAnaghaSathaye | Mostly done.  Not passed yet |  | Needs some adjustments. Uses absolute paths |

[Learning API](https://github.com/UniversityOfAppliedSciencesFrankfurt/LearningApi.git) provides a concept of chaining of different learning modules in a single processing pipeline.   
In this project student has to implement a pipeline which with receive some data (typically double[][]) and draws this data as HTML page.

The pipeline module should not change the data at any point of processing. Same inputted data MUST be returned to the caller.

Here is the example of the method, which should be implemented:

Data:

12.21, 15.23   
…   
123.4, 78.73

public double[] Run(double[] rawData, IContext ctx)   
{

     TODO: WritePointToHTML(rawData)

}

Where:

rawData is the vector of multiple scalars.

## Implement Image Binarizer

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **MATNR** | **Email** | **Path** | **State** | **Quality** | **Comment** |
| Basumitra Dutta |  | basumitradutta2015@gmail.com |  |  |  |  |

Image Binarizer is already implemented as a class in LearningApi: *$LearningApi\src\Helpers\ImageBinarizer\Binarizer.cs*.

It is used to create 1/0 sequence from image. Any specified image is loaded, binarized and saved as binarized output. The goal of this project is to improve current implementation. On of improvements is automatically selection of optimal values for R,G,B thresholds, between 0 and 1.

One possible optimization is calculation of mean value of the image of each R,G and B components and choosing that values as thresholds. You can also try different approaches.

Output of the project should be a LarningAPI pipeline component. It should be a well-documented code published as nuget package, which can also be published as .NET tool.

At the end of project binarizer needs to be a pipeline component. Additionally an exe application should be done and published as .NET tool.

## Implement Center Module for LearningApi

## Implement Center module for Learning API

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **MATNR** | **Email** | **Path** | **State** | **Quality** | **Comment** |
| Valerian Scherr |  | v.scherr@gmx.de |  |  |  |  |

The goal of this project is to calculate and match center of data. It can be any picture or number. During training process of an algorithm, data in dataset might not be centered. For example, consider following vector:

0000011100000

In training dataset the same vector might look as shown in next examples:

0001110000000  
0000000111000  
0011100000000

As shown above, the sequence ‘111’ is not centered in dataset. Your task is to find algorithm to center the data. Note that training vectors might be multidimensional. Following example shows digit ‘7’.

|  |  |
| --- | --- |
| 0000000000 0111111100 0000011000 0000110000 0001100000 0011000000 0110000000 | 0000000000 0001111111 0000000110 0000001100 0000011000 0000110000 0001100000 |

Use MNIST dataset of digits: <http://yann.lecun.com/exdb/mnist/>

## Soebel Edge detection

Implement SOEBEL edge detection algorithm

<https://en.wikipedia.org/wiki/Sobel_operator>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **MATNR** | **Email** | **Path** | **State** | **Quality** | **Comment** |
| **Abhishek sharma** | 1274291 | Abhisharmaec0982@gmail.com | **Abhishek sharma** | 1274291 | Abhisharmaec0982@gmail.com | **Abhishek sharma** |
|  |  |  |  |  |  |  |
| **Rahul Krishna** | 1224694 | Rahulcbr250r@gmail.com |  |  |  |  |

## Bayesian Network

Implement Bayesian Network

<https://en.wikipedia.org/wiki/Bayesian_network>

<https://dslpitt.org/genie/wiki/C_Sharp_Tutorial_1:_Creating_a_Bayesian_Network>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **MATNR** | **Email** | **Path** | **State** | **Quality** | **Comment** |
| Quang Trung Tran |  | tranquangtrung.vgu@gmail.com |  |  |  |  |

## Image Edge Detection

The types of edge detection are: Laplacian, Laplacian of Gaussian, Sobel, Prewitt and Kirsch

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| Anilkumar   |  |  | | --- | --- | | Madhu | Bodige (Laplacian of Gaussian) | | Pujar | anilpujar06@gmail.com  bodigemadhushashi@gmail.com |  | Working on Laplacian detection |  |

<https://code.msdn.microsoft.com/windowsapps/Image-Edge-Detection-5c5a0dc2>

<https://en.wikipedia.org/wiki/Laplace_operator>

[https://en.wikipedia.org/wiki/Blob\_detection#The\_Laplacian\_of\_Gaussian](https://en.wikipedia.org/wiki/Blob_detection)

<https://en.wikipedia.org/wiki/Sobel_operator>

<https://en.wikipedia.org/wiki/Prewitt_operator>

<https://en.wikipedia.org/wiki/Kirsch_operator>

## Blob Detection

The types of edge detection are: Laplacian, Laplacian of Gaussian, Sobel, Prewitt and Kirsch

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **MATNR** | **Email** | **Path** | **State** | **Quality** | **Comment** |
| lakshmi sudha | tanikella | lsudhatanikella@gmail.com |  |  |  |  |

<https://en.wikipedia.org/wiki/Blob_detection>

<https://www.codeproject.com/Questions/738734/blob-detection-in-csharp>

[www.dreamincode.net/forums/topic/219038-c%23-blob-detection/](http://www.dreamincode.net/forums/topic/219038-c%23-blob-detection/)

## Implement Euclidian Color Filter in .NET Core

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| Sven Eisenbach |  | [Sven-Eisenbach3@hotmail.de](mailto:Sven-Eisenbach3@hotmail.de) |  |  |  |

Implement algorithm for Euclidian Color Filtering. Algorithm should be implemented in .NET Core and MUST BE aligned to [Learning API](https://github.com/UniversityOfAppliedSciencesFrankfurt/LearningApi.git) open source project.    
Additionally, provide a simple Windows Forms application, which loads an image and performs Gaussian noise removal by changing of various parameters.

*References*:

<http://www.aforgenet.com/framework/docs/html/67fa83b5-dede-8d3a-8d3b-b7a6b9859538.htm>

<http://www.aforgenet.com/framework/>

1. Validate and improve Tests of existing agorithms

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| Saboor Abdul  Hafiz Maaz Ahmed  Shruti mathure  Jayashree Regoti | K3.1  SJ90 | Saboorabdul3333@gmail.com  [hafiz.ahmed@stud.fra-uas.de](mailto:hafiz.ahmed@stud.fra-uas.de)  shruti.mathure@stud.fra-uas.de  jayashree.regoti@stud.fra-uas.de |  |  |  |

Run existing tests, check if they are correct and add missing ones. You will dig into all projects in learning API, read documentation and understand all unit tests. You can also restructure existing tests also provide additional tests. You will have to understand algorithm, think about new tests (experiments), find appropriate data and implement tests.  
All tests in all projects must run and must successfully pass

This project can be shared up to 5 students. Note that every student will have a set of her/his own algorithms to be validated and improved. Project is worked out in the public github repo.

1. Fix namespaces

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
|  |  |  |  |  |  |

Fix namespaces in LearningApi so they are in a unified pattern and meet convention standards.

Use following code analyzers:  
<https://docs.microsoft.com/en-us/visualstudio/code-quality/install-fxcop-analyzers?view=vs-2019>

1. Increase .Net Core version to 3.0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
|  |  |  |  |  |  |

Increase .Net Core version of each existing project to 3.0.

1. Clean up code of existing projects

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** |  | **Email** | **Path** | **State** | **Quality** |
|  |  |  |  |  |  |  |

Check existing code and ensure it meets coding convention standards and does not contain commented code, code violations etc.

Use following code analyzers:  
<https://docs.microsoft.com/en-us/visualstudio/code-quality/install-fxcop-analyzers?view=vs-2019>

**Migrate old projects to LearningApi repository**

Add projects from previous semesters to LearningApi. Existing well done projects are usually after completion migrated to the LearningAPI repository on github. Your task is to review the project code, tests and project documentation. Usually some improvements need to be done. You should identify gaps in the project and fix them. Finally, you should publish the project in the public LearningAPI repository in the common open source project quality.

Todos:

* Check naming convention
* Make sure that unit tests are running and passing
* Provide more unit tests by using different input dataset(s). Find somewhere some dataset (i.e.: CSV file) and implement unit test.
* Make sure that documentation is understandable. If not improve it.
* Provide examples, which easy demonstrate how to use the algorithm.  
  Example:   
  You have to make following documentation in a high quality: <https://github.com/UniversityOfAppliedSciencesFrankfurt/LearningApi/blob/master/GaussianMeanFilter.md>  
  You can use as a source student’s documentation (PDF/DOCX). See the associated paper (documentation) related to this algorithm.  
  <https://github.com/UniversityOfAppliedSciencesFrankfurt/se-dystsys-2018-2019-softwareengineering/blob/Asmade-Kabore/MyMachineLearningProject/Documentation/documentation_final_version.pdf>

You will get access to given project after assignment.

1. Migrate ML 18/19-2.1.Kernel Logistic Regression to LearningAPI repository

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
|  |  |  |  |  |  |

1. Migrate ML18/19-2.8 Decision Trees to LeaningApi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
|  |  |  |  |  |  |

1. Migrate ML18/19-3.2.Implement one Algorithm for dimensionality reduction to LeaningApi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
|  |  |  |  |  |  |

1. Migrate ML18/19-5.1 Migration of Python Hello-Sample to .NET Core to LearningApi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
|  |  |  |  |  |  |

1. Migrate ML 18/19-2.3. Implement Anomaly Detection with LearningApi (K-Mean Algorithm) to LearningApi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
|  |  |  |  |  |  |

1. Migrate ML18/19-4.3. Testing of RBM for Recommendation Engine to LearningApi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
|  |  |  |  |  |  |

1. Migrate ML18/19-3.11 Implement Euclidian Color Filter in .NET Core to LearningApi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| Deepali Ashok Betkar |  | deepali.betkar@stud.fra-uas.de |  |  |  |

1. Migrate ML18/19-3.6. Implement Center Module for LearningApi to LearningApi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
|  |  |  |  |  |  |

1. Migrate ML18/19-1.16 13.2 Markov Chain Model to LearningApi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
|  |  |  |  |  |  |

1. Migrate ML 18/19-3.7 Soebel edge detection  to LearningApi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
|  |  |  |  |  |  |

1. Migrate ML 18/19-2.1. Implementation of convolutional Network to LearningApi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
|  |  |  |  |  |  |

1. Migrate ML18/19-3.3 Gaussian and Mean Filters for noise removal in .NET Core to LearningApi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
|  |  |  |  |  |  |

1. Migrate ML18/19-3.1 Dimension Reduction by Ring Projection Pipeline Module to LearningApi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
|  |  |  |  |  |  |

1. Migrate ML18/19-2.4 Support Vector Machine to LearningApi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
|  |  |  |  |  |  |

1. Migrate 18.2 Implement Image Binarizer(ML 18/19-2.5) to LearningApi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| Arinze Okechukwu Okpagu  Uchechukwu John Emenike | SEIV-X | arinze.okpagu@stud.fra-uas.de  jonemenike2000@gmail.com |  |  |  |

1. Implement Gabor filter C#

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **MatNr** | **Email** | **Path** | **State** | **Quality** |
|  |  |  |  |  |  |

# Research on existing algorithms and Quality Improvement projects

These projects focus more research than programming. If you are not well skilled developer, these projects might be more suitable to you. You will have to do some programming in any case, but your task will not be implementation of algorithm or module. Your task will be to apply algorithm on some meaningful data source.

## (NOT OFFERED IN THIS SEMESTER)Image Recognition for HoloLens

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
|  |  |  |  |  |  |

Current implementation of Deep Belief algorithm, which has been done on University of Applied Sciences can be used for learning of patterns. In this project, students must apply existing algorithm and provide and validate number of tests, which demonstrate how algorithm recognize images.

In the first part of project, student must use Deep Belief implementation in Learning API.

In the second part of algorithm project, the code must be implemented, which capture image on HoloLens and uses Deep Belief algorithm to recognize trained object.

## (NOT OFFERED IN THIS SEMESTER)Physical World Recognition Algorithm for HoloLens

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
|  |  |  |  |  |  |

Current implementation of Deep Belief algorithm, which has been done on University of Applied Sciences can be used for learning of patterns. In this project, students must apply existing algorithm and provide and validate number of tests, which demonstrate how algorithm recognize different patterns provided as vertices.

Your task is to implement .NET Core solution which uses Deep Belief algorithm to learn pattern provided as **list of vertices**. Note that list of vertices provided by HoloLens is different than image. However same technic can be used for both.

In the second part of algorithm project, the code must be implemented, which capture vertices on HoloLens and uses trained model Deep Belief algorithm to recognize trained object.

Trained objects are usually objects in a room. That might be table, chair, wall, TV, Flowers, People or anything else. Vertices will not help us to recognize persons, but it can help to recognize human as a person.

Algorithm should be able to recognize some physical object in room. HoloLens will generate a mesh for all surfaces in the room. Algorithm will be previously trained to recognize some object (i.e. chair, table, TV, surface, HoloLens, SmartPhone, some number, some symbol etc.) by using of some algorithm. After training algorithm will load data provided by HoloLens and find wanted object.

You will not be required to generate mesh directly from HoloLens. Mesh data will be provided to you as binary file.

*References*:

You do not have to implemented anything described under following references. They describe, technology behind creating of vertex data, which will be used as input in your project.

<https://developer.microsoft.com/en-us/windows/holographic/spatial_mapping>

[https://developer.microsoft.com/en-us/windows/holographic/Spatial\_mapping\_in\_DirectX.html#set\_up\_your\_app\_to\_use\_the\_spatialperception\_capability](https://developer.microsoft.com/en-us/windows/holographic/Spatial_mapping_in_DirectX.html)

## Testing of RBM for Recommendation Engine

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| **Sai Durga Dasu** |  | [saidurga606@gmail.com](mailto:saidurga606@gmail.com) |  |  |  |

RBM Algorithm is already a part of LearningAPI. This algorithm can already be used as recommendation engine. The goal of this project is to try movie recommendation (or something else) by using of existing algorithm. This algorithm has been implemented as a part of a thesis on Frankfurt University of Applied Sciences.

For recommendation MovieLens dataset has been used which can be found at https://grouplens.org/datasets/movielens/. This dataset contains 500,100 anonymous ratings of 3898 movies by 3,257 users. This data contains User Id, Movie Id, Ratings and timestamps. The ratings are between 1 ~ 5. Ratings are made on a 5-star scale, which means only integers but no decimal or rational number are used. Each user has at least 20 ratings. Complete dataset has not been explained here. Only the information needed for this thesis is explained. There are 18 genres in total for all the movies.

## Create HelloWorld Tutorial for LearningAPI

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| Anusha | Ashok Reddy | anushaashokreddy@gmail.com |  |  |  |

The goal of this project is to create Hello World tutorial for LearningAPI. The tutorial should be written as Markdown document on Git and it should contain a solution ‘HelloLearningAPI’ with tree projects: HelloAlgorithm, HelloPipelineComponent and HelloLearningAPIUnitTests.

Results of the project will be published on public git repository.

Use following coding/naming convention:

<https://github.com/ktaranov/naming-convention/blob/master/C%23%20Coding%20Standards%20and%20Naming%20Conventions.md>

## Implement and Test Save and Load in LearningAPI algorithms

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| Muhammad  Khubaib |  | m.khubaib18@gmail.com |  | Working on Linear Regression Save/Load |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| Aamir Muhammad |  | mirmuhammad91@gmail.com |  | Gaussian and mean filter for noise removal |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| Sardar Eyaan Ahmed |  | eyaan1@outlook.com |  | Working on Logistic Regression Save/Load |  |

aa

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| **Ehtasham Ul Hassan** |  |  |  |  |  |

***This project can be taken by multiple students.***

Most of algorithms in LearnignAPI have never been tested for Save and Load features. These features provide support for saving (persistence) of trained models to a file and reading of persisted (saved) model from file. With this feature trained models can be shared as a file.

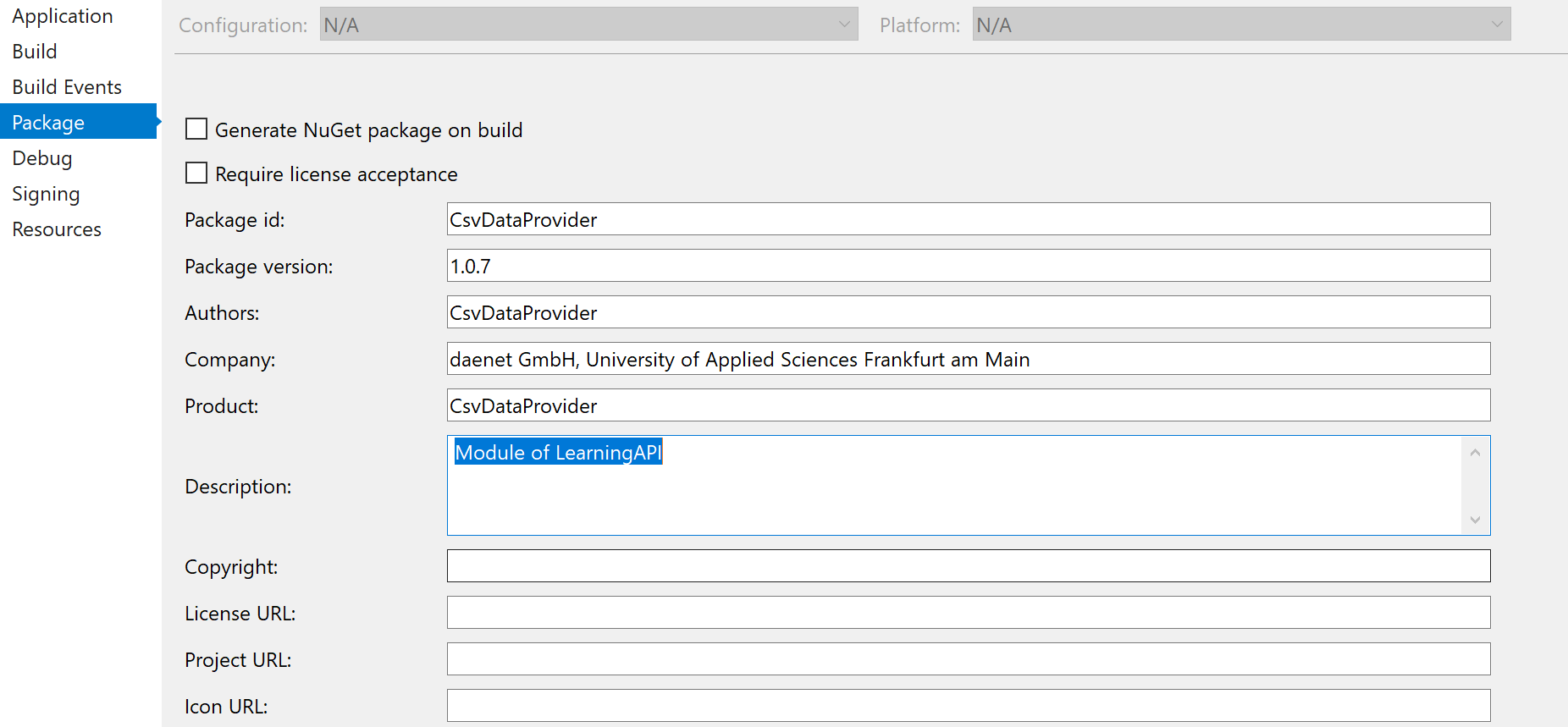
The goal of this project(s) is to identify such algorithm in LearningAPI and to implement few UnitTests, which invoke methods Save() and Load(). Unit tests should fully implement training (as already done in all algorithms), then to save trained model, to load it and finally t compare results of loaded and trained models.

Results of these project(s) will be published on git public repository.

## Create and document all NUGET packages of LearningAPI algorithms

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
|  |  |  |  |  |  |

The goal of this project is to create NUGET packages of all LearningAPI projects.  
You will have to step through all algorithms implemented in LearningAPI and build NUGET packages with corresponding package description as shown at following picture.



Before building of packages UnitTests have to be successfully executed.  
At the end of the project packages will be documented on public Git and published to NUGET public repository.

## Implement batched data processing for LearningAPI

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| Vasuki Muthusamy |  | [vasukimuthusamy@gmail.com](mailto:vasukimuthusamy@gmail.com) |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
|  |  |  |  |  |  |

***This project can be taken by multiple students.***

The goal of this project is to implement and document batched processing for existing algorithms in the LearnigAPI, which do not support it.

Currently, K-Means algorithm supports batching only. Following URL shows how batching can be mplemented: <https://github.com/UniversityOfAppliedSciencesFrankfurt/LearningApi/blob/master/WorkingWithBatches.md>

## Function recognition K-Means prediction test

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
|  |  |  |  |  |  |

K-Means is one of the most famous unsupervised clustering algorithm used in machine learning. You will be provided with the implementation of K-Means in C# along with an extension, also in C#, for function recognition. You will also be provided with the MNIST and Semeion datasets which contain binary images of handwritten digits.

Problem statement: The current implementation of the K-Means function recognition extension is very sensitive to outliers in the training data and the aim is to implement an algorithm that detects these outliers. For example, if the same or similar hand-written digit is not centered same way, algorithm will not be able to train accurately.

After fully understanding the current implementation of K-Means and K-Means function recognition extension, the tasks will be as following:

1 - Go manually through any of these datasets and form an outlier-free training set. Go through many of images and select only images, which are not outliners. If possible, you can correct outliners.

Use corrected/selected dataset to train algorithm.

The clustering results of the following task will be used as a reference for the next task.

Optionally Implement an algorithm, based on what was learned from task 1, to detect automatically outliers in these training datasets and any dataset in general.

Train algorithm and use data to check how prediction works with selected dataset.

# Hierarchical Temporal Memory

Before you start with project you should read about HTM. Quick review of Hierarchical Temporal Memory (HTM): <https://en.wikipedia.org/wiki/Hierarchical_temporal_memory>

Projects in this section are all related to HTM. You can chose one of projects, which are not blocked and start with code **ML 19/20**. No any of projects in this section will require you to implement any algorithm. But in all projects you will have to implement experimental code, which helps research certain tasks. Experimental code must be provided in a form of unit tests.

The focus of projects in this section is research. That means, you will have to write code which implement required experiments by using some data. When writing code for your experiment you will most likely have to change various of parameters in experiments and document results as diagrams, tables or similar.

Diagrams can be created by using Excel. Unite Test which implements code for experiment must be deterministic. It should always use same set of input data, provided set of input arguments and deterministic output. Excel should point to output files generated by your experiment.

Your final documentation must point to experiment UnitTest, required input data and final output.

## Migration of Python Hello-Sample to .NET Core

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| Ikechukwu Joseph  Okoye |  | okoyeijoseph@gmail.com |  |  |  |

Following samples should be implemented as UnitTests in C# .NET Core implementation of Spatial Pooler.  
  
[hello\_sp.py](https://github.com/numenta/nupic/blob/master/examples/sp/hello_sp.py)

<https://github.com/numenta/nupic/tree/master/examples/sp>

UnitTests must be well documented and various tests with different parameters should be executed.

After implementation of Unit Tests, some research must be done in dependence on different topologies of the Spatial Pooler.

Input/Columns should take following topologies: I={x,y}, C={u,w}. Result should show how initialization and learning computation behaves for different values of input and column topologies. Results should be shown in form of **plotly** diagrams in Python (examples will be provided). C# UnitTest should generate file with result data and Python code will lot the file and create the graph.

1. Performance of initialization of Spatial Pooler which includes execution of following code.

var parameters = GetDefaultParams();

parameters.setInputDimensions(new int[] { 32, 32 });

parameters.setColumnDimensions(new int[] { 64, 64 });

var sp = new SpatialPooler();

var mem = new Connections();

parameters.apply(mem);

//Stopwatch.start

sp.init(mem);  
//Stopwatch.stop

sw.WriteLine($”{stopwatch.totalms}”);

2. Performance of execution of compute method

var parameters = GetDefaultParams();

parameters.setInputDimensions(new int[] { 32, 32 });

parameters.setColumnDimensions(new int[] { 64, 64 });

var sp = new SpatialPooler();

var mem = new Connections();

parameters.apply(mem);

sp.init(mem);

int[] activeArray = new int[64 \* 64];

int[] inputVector = Helpers.GetRandomVector(32 \* 32, parameters.Get<Random>(KEY.RANDOM));

//Stopwatch.start

sp.compute(mem, inputVector, activeArray, true);  
//Stopwatch.stop

3. Tests robustness of the HTM algorithm against noise in input. First show graphically (propose appropriate graph), that different inputs produce different active columns in output. Also show that same inputs produce same active columns.

As next add some noise to input vector. Noise should be added in percent by flipping specified number of bits in input vector. Then add simultaneously more and more noise to different topologies and provide appropriate diagrams, which explain how much noise can be added to input vector and keep a stable result. Compare this with error math equation ( ) of HTM.

To calculate change of output (change in active columns) use hamming distance as implemented in method GetHammingDistance.

All tests and diagrams must be briefly documented.

Note results produced by C# code must be identical with results produced by existing Python code.

## Migration of Python Spatial Pooler Sample to .NET Core Part 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| Thanh Quang Nguyen |  | thanhquang0912@gmail.com |  |  |  |

This project is related to **#Part 1** of the sample of Python file [sp\_tutorial.py](https://github.com/numenta/nupic/blob/master/examples/sp/sp_tutorial.py) . Sample shown below should be implemented as UnitTests in C# .NET Core implementation of Spatial Pooler.  
  
In SpatialPooler algorithm every column connects to a subset of the input vector (specified by both the potentialRadius and potentialPct). The overlap score for a column is the number of connections to the input that become active when presented with a vector. When learning is 'on' in the SP, the active connections are reinforced, whereas those inactive are depressed (according to parameters synPermActiveInc and synPermInactiveDec. In order for the SP to create a sparse representation of the input, it will select a small percentage (usually 2%) of its most active columns, ie. columns with the largest overlap score. In this **first part (YOUR PROJECT)**, you will have to create a histogram showing the overlap scores of the Spatial Pooler (SP) after feeding it with a random binary input. As well, the histogram will show the scores of those columns that are chosen to build the sparse representation of the input.  
That means, as in Python example already implemented, your C# code will do same as shown in example below.

var parameters = GetDefaultParams();

parameters.setInputDimensions(new int[] { 32, 32 });

parameters.setColumnDimensions(new int[] { 64, 64 });

var sp = new SpatialPooler();

var mem = new Connections();

parameters.apply(mem);

sp.init(mem);

int[] activeArray = new int[64 \* 64];

int[] inputVector = Helpers.GetRandomVector(32 \* 32,  
 parameters.Get<Random>(KEY.RANDOM));

sp.compute(mem, inputVector, activeArray, true);

overlaps = sp.getOverlaps()

WriteArrayToFile(overlaps, “overlaps.txt”);// Implement this method

foreach i in activeCols.nonzero() // PSEUDO CODE

activeColsScores.append(overlaps[i])

WriteArrayToFile(activeColsScores, “winners.txt”);

To show histogram, your code will have to be written as output to text file. Here is an example of two created files:

1,2,3,3,4,5,5,5,7,8,8,9,10,10,10,10 // overlaps.txt  
0,0,3,3,4,5,5,5,7,9,9,9,10,10,11,12 // winners.txt

Once you have created files, you can use (run) already finished Python code (to be provided), which loads created files and create histogram.

Following example shows how to run Python code from command prompt:

python ./OverlapTest/create-histogram.py graph1 28 overlaps ./OverlapTest/overlaps.txt winners ./OverlapTest/winners.txt

Your task is to create diagrams, which show how activated columns in **activeArray** are distributed for different topologies. Example above shows topologies 32x32 and 64x64.

All code should be implemented in well documented UnitTests. To save time, you can run Python code (optionally) from UnitTest directly.

All tests and diagrams must be briefly documented.

Note results produced by C# code must be identical with results produced by existing Python code.

## Approval of stabilization of similar images

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| Anil Kumar  Gupta |  | ganilkumar226@gmail.com |  |  |  |

By using of spatial pooler, native image data will be loaded from JPEG/PNG image and binarized by using of LeraningAPI ImageBinarizer. Your task is show how selected active columns (returned in activeArray after invoking of method compute) change in dependence of image change.

You should think about various test, which can be performed. For example, you could take a single image and add noise to it, by flipping some ‘1’ and ‘0’ for specified percentage. Try this experiment for multiple images (>20).

Then try to use same original images and move it to slightly different position by on Y and Y axis. Note how active columns (activeArray) are changing.

The idea of this project is to research and document how spatial pooler generates same output for slightly different changes of input (in this case images).

Additionally, try to add some noise to image by flipping a number of 1 and 0. For example flip 10% of 1 bits in 0 bits and vice versa. Document how stable is the result.

Also try to use same image moved in position X, to see how output changes in that case.

Following code shows how image data should be loaded in initialized spatial pooler and trained by invoking of *compute()* method.

var parameters = GetDefaultParams();

parameters.setInputDimensions(new int[] { 32, 32 });

parameters.setColumnDimensions(new int[] { 64, 64 });

var sp = new SpatialPooler();

var mem = new Connections();

parameters.apply(mem);

sp.init(mem);

int[] activeArray = new int[64 \* 64];

int[] inputVector = READIMAGEDATA()// TODO. Implement this method

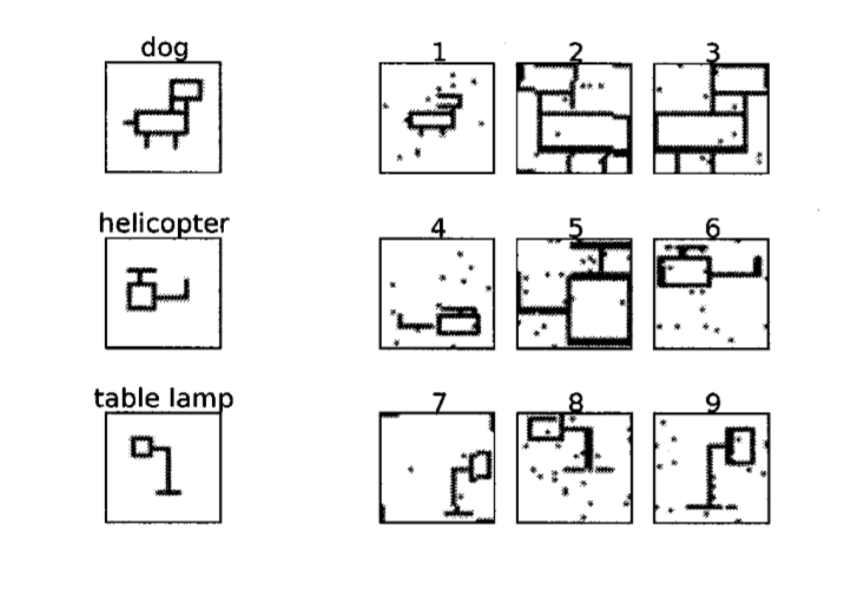
for(int n=0;n<??)

sp.compute(mem, inputVector, activeArray, true);

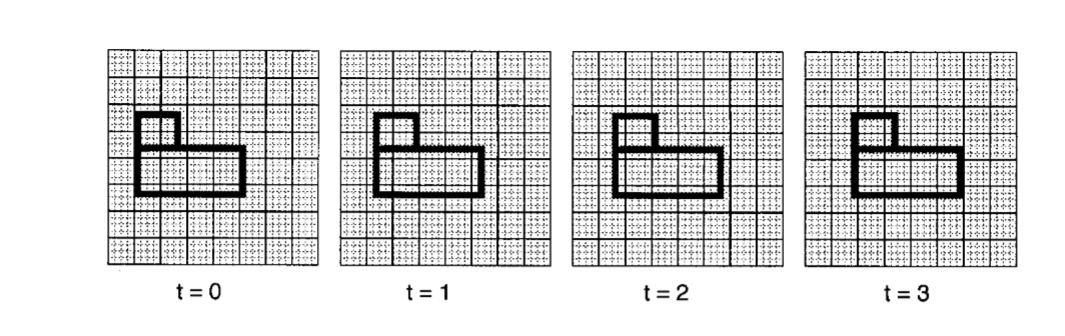
noiseVector1p = makeSomeNoise(inputVector, 0.01);  
sp.compute(mem, inputVector, activeArray1p, false);

noiseVector5p = makeSomeNoise(inputVector, 0.05);  
sp.compute(mem, inputVector5P, activeArray5p, false);

In your experiment you can follow ideas shown at following image:



You should also use following example, which sows moving of the simple object.



You do not have to use same image. Feel free to create some new image. I.e.: smiley. Then move image from left to right and see how result is changed. Result has to be well documented.

## Improving of quality of .NET Core Spatial Pooler

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
|  |  |  |  |  | In progress |

In this project student must improve the quality of existing implementation of Spatial Pooler. Spatial Pooler is a component of Hierarchical Temporal Memory NeocortexAPI.

Following tasks should be done:

1. .NET Naming Convention must be applied.
2. All warnings must be removed
3. All methods and classes inside of Spatial Pooler must be documented
4. All UnitTests must be documented
5. Spatial Pooler documentation has to be written. Documentation can be aligned to Spatial Pooler existing documentation in Python.

## Implementation of Scalar encoder in HTM

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| Syed Hassnat Alamgir |  | [syedhassnat@hotmail.com](mailto:syedhassnat@hotmail.com) |  |  |  |

See this URL to understand theory of encoders:

<https://pdfs.semanticscholar.org/1455/2052a0ee729539298e4f8eb42292c0eb1622.pdf>

Scalar encoder is a component (class), which encodes specified number to bit sequence.  
Encoder is defined by **minval** and a **maxval** which determine the value range.   
Endoder is also defined by number of bits **n** in encoded bit sequence and **w** number of bits set to   
  
The encoding will have **n** total bits with **w** on bits (1’s) to encode values in range **minVal** - **maxVal**. Values are put into buckets.   
  
There are **(n-w)+1** buckets that each represent an equally-sized value range between **minval** and **maxval**. The smallest bucket is represented with the first w bits on and the rest off. The next larger bucket is represented by shifting the on bits to the right by one position. In this way, adjacent buckets have the most overlap which helps to capture the semantics of scalar values.

Following tables shows all parameters, which defines encoder:

|  |  |
| --- | --- |
| n | The number of bits that are set to encode a single value |
| w | The "width" of the output signal. Note: w must be odd to avoid centering problems. |
| maxVal | The upper bound of the input signal |
| minVal | The minimum value of the input signal. |
| periodic | If true, then the input value "wraps around" such that minval = maxval For a periodic value, the input must be strictly less than maxval, otherwise maxval is a true upper bound. |

There are three mutually exclusive parameters that determine the overall size of  
of the output. Only one of these should be specified by initialization of encoder (constructor):

|  |  |
| --- | --- |
| radius |  |
| resolution | Two inputs separated by more than the radius have non-overlapping representations. Two inputs separated by less than the radius will  in general overlap in at least some of their bits. You can think  of this as the radius of the input. |
| radius | Two inputs separated by greater than, or equal to the resolution are guaranteed to have different representations. |

Example:   
A scalar encoder with a range from 0 to 100 with n=12 and w=3 will produce the following encodings:

1   becomes 111000000000  
7   becomes 111000000000  
15 becomes 011100000000  
36 becomes 000111000000

For more information look here:

[http://nupic.docs.numenta.org/1.0.3/api/algorithms/encoders.html#](http://nupic.docs.numenta.org/1.0.3/api/algorithms/encoders.html)

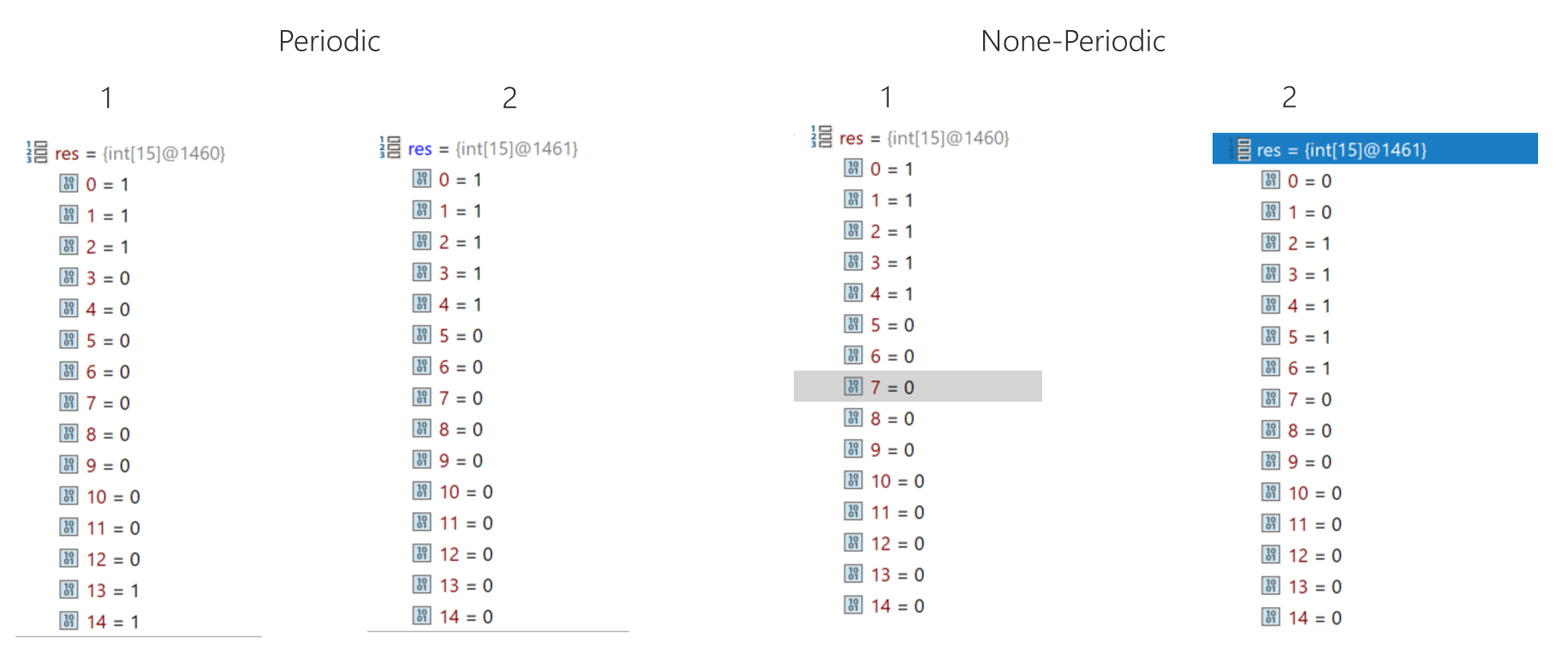
<http://nupic.docs.numenta.org/1.0.3/api/algorithms/encoders.html#scalar-encoders>

<https://www.youtube.com/watch?v=V3Yqtpytif0>

Please note that encoder (also all other encoders defined in this document) MUST implement following abstract class:

public abstract class EncoderBase<T> : EncoderBase

Encoding sample:



## C# Implementation of DateTime encoder for HTM

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| Ramesh Kumar | 2018/2019 | rkumar.panchani@gmail.com |  |  |  |
|  |  |  |  |  |  |
| Moin Abbas  Qureshi | 2019/2020 | [moinabbasqureshi@gmail.com](mailto:moinabbasqureshi@gmail.com) |  |  |  |

For more information look here:

[http://nupic.docs.numenta.org/1.0.3/api/algorithms/encoders.html#](http://nupic.docs.numenta.org/1.0.3/api/algorithms/encoders.html)

<https://www.youtube.com/watch?v=PTYlge2K1G8&index=7&list=PL3yXMgtrZmDqhsFQzwUC9V8MeeVOQ7eZ9>

## C# Implementation of Category encoder for HTM

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| Matih Ullah |  | matihjanjua11@gmail.com |  |  |  |

Encodes a list of discrete categories (described by strings), that aren’t related to each other

For more information look here:

[http://nupic.docs.numenta.org/1.0.3/api/algorithms/encoders.html#](http://nupic.docs.numenta.org/1.0.3/api/algorithms/encoders.html)

## Learning progress experiment in Spatial Pooler

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| |  |  | | --- | --- | | Akib Ibn  Zahid |  | |  | [akibibnzahid@gmail.com](mailto:akibibnzahid@gmail.com) |  |  |  |

Spatial Pooler columns connects to a subset of the input vector (specified by both the potentialRadius and potentialPct). During this process overlap score of columns is created. The overlap score for a column is the number of connections to the input that become active when presented with a vector. This is a random process driven during initialization of Spatial Pooler algorithm. When learning is activated in the SP, the active connections are reinforced, whereas those inactive are depressed (according to parameters synPermActiveInc and synPermInactiveDec. In order for the SP to create a sparse representation of the input, it will select only a small percentage (usually 2%) of its most active columns, ie. columns with the largest overlap score.

Following code shows how to setup Spatial Pooler.

[TestMethod]

public void StableOutputOnSameInputTest()

{

var parameters = GetDefaultParams();

parameters.setInputDimensions(new int[] { 32, 32 });

parameters.setColumnDimensions(new int[] { 64, 64 });

parameters.setNumActiveColumnsPerInhArea(0.02 \* 64 \* 64);

var sp = new SpatialPooler();

var mem = new Connections();

parameters.apply(mem);

sp.init(mem);

int[] activeArray = new int[64 \* 64];

int[] inputVector = Helpers.GetRandomVector(32 \* 32,   
 parameters.Get<Random>(KEY.RANDOM));

for (int i = 0; i < 10; i++)

{

sp.compute(mem, inputVector, activeArray, true);

var activeCols = ArrayUtils.IndexWhere(activeArray, (el) => el == 1);

var str = Helpers.StringifyVector(activeCols);

Debug.WriteLine(str);

}

}

Your task is to create diagrams, which show how activated columns in **activeArray** (output of compute cycle) change in every cycle. Results have to be shown as diagrams created with **plotly** in Python (you will be assisted in this part of task). For example, C# code will generate text output, which will be loaded by Python code to create diagram with **plotly**. Python code might be provided and idelly should be invoked directly from unit test.

In this task you should use many different topologies of Spatial Pooler and Inputs vectors. Example above shows topologies 32x32 and 64x64.

All tests and diagrams must be briefly documented.

## Draw diagrams which show inhibition process of HTM

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| Muhammad Sadat Shaukat |  | sadatshaukat@gmail.com |  |  |  |

HTM algorithm has internally a process of selecting of neighborhood cells and inhibiting them. Your task in this project is to graphically represent how inhibition is distributed by specific cell overlaps.

For every input vector, algorithm will create a random set of connections from cells to the input. During learning process, in every cycle, different set of cells will be chosen from set of cells, which are connected to input. Inhibition process will take a care that only a small subset of cells of connected cells will be chosen as winner.

You should implement code, which runs algorithm and writes result to some file to create diagram from it. You can use some simple Python script to create plotly diagram from files.

Following code shows how to run algorithm:

/// <summary>

/// Generates result of inhibition

/// </summary>

[TestMethod]

public void SPInhibitionTest()

{

var parameters = GetDefaultParams();

parameters.setInputDimensions(new int[] { 1000 });

parameters.setColumnDimensions(new int[] { 2048 });

parameters.setNumActiveColumnsPerInhArea(0.02 \* 2048);

parameters.setGlobalInhibition(true);

var sp = new SpatialPooler();

var mem = new Connections();

parameters.apply(mem);

sp.init(mem);

int[] inputVector = Helpers.GetRandomVector(1000, parameters.Get<Random>(KEY.RANDOM));

int[] activeArray = new int[2048];

for (int i = 0; i < 10; i++)

{

var overlaps = sp.calculateOverlap(mem, inputVector);

var strOverlaps = Helpers.StringifyVector(overlaps);

var inhibitions = sp.inhibitColumns(mem, ArrayUtils.toDoubleArray(overlaps));

var strInhibitions = Helpers.StringifyVector(inhibitions);

sp.compute(mem, inputVector, activeArray, true);

var activeCols = ArrayUtils.IndexWhere(activeArray, (el) => el == 1);

var strActiveCols = Helpers.StringifyVector(activeCols);

Debug.WriteLine(strOverlaps);

Debug.WriteLine(strInhibitions);

Debug.WriteLine(strActiveCols);

}

}

Last 3 lines of code prints out calculated overlaps, inhibited columns and active columns. Later two should be identical. You should change the code to write results to file and then to invoke already existing Python code, which will create diagram from specified file.

*Remarks:*

Overlap is defined as the number of synapses of a single column, which are connected to active input bits. Synapses connected to inactive input bits are not included in overlap.

Analog to overlap, Percentage Overlap is a ratio between number of column synapses connected to active (set on ‘1’) input bits and number of column synapses to input bits. For example, if column is connected to 1000 input bits and 100 of 1000 connected inputs bits is set on ‘1’ (means column overlap is 100) then percentage overlap ratio is 100/1000 = 0.1.

## Migration of Python Spatial Pooler Sample to .NET Core Parts 2a, 2b, and 3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| Asmade Kabore |  | [asmade.kabore@gmail.com](mailto:asmade.kabore@gmail.com) |  |  |  |

This project is related to **#Part 2a, 2b and 3** of the sample of Python file [sp\_tutorial.py](https://github.com/numenta/nupic/blob/master/examples/sp/sp_tutorial.py) .

Read project *Migration of Python Spatial Pooler Sample to .NET Core Part* 1 and implement Parts 2a, 2b and 3 of Python example.

This project is identical to “*Migration of Python Spatial Pooler Sample to .NET Core Part* 1”.

## Image classification with Spatial Pooler

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| khushnood yousaf(left)  Mandeep Singh |  | [khushnoodyousafsanwal@gmail.com](mailto:khushnoodyousafsanwal@gmail.com)y  mandeepsingh.sherry@gmail.com |  |  |  |

See project “Approval of stabilization of similar images”.

<https://pdfs.semanticscholar.org/be51/8b8fa749d51c4fb75a19614b6a05fcb5d9d0.pdf>

<https://ieeexplore.ieee.org/document/6460663>

<http://publications.lib.chalmers.se/records/fulltext/157497.pdf>

<https://www.chi.uni-hannover.de/uploads/tx_tkpublikationen/masterarbeit_muders_2011.pdf>

## Implement different inhibition in Spatial Pooler

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| Vadlamani Raghuram  Trung Luu |  | [raghuram.vadlamani1@gmail.com](mailto:raghuram.vadlamani1@gmail.com)  baotrung1309@gmail.com |  |  |  |

Inhibition of columns is implemented in NeoCortexApi\SpatialPooler.cs in following method:

public virtual int[] inhibitColumnsLocal(Connections c, double[] overlaps, double density)

Your task is to change existing inhibition algorithm and to document changes of accuracy of Spatial Pooler.

To do this test robustness of the HTM algorithm against noise in input by comparing existing algorithm versus new one. First show graphically (propose appropriate graph), that different inputs produce different active columns in output. Also show that same inputs produce same active columns.

As next add some noise to input vector. Noise should be added in percent by flipping specified number of bits in input vector. Then add simultaneously more and more noise to different topologies and provide appropriate diagrams, which explain how much noise can be added to input vector and keep a stable result. Compare this with error math equation ( ) of HTM.

To calculate change of output (change in active columns) use hamming distance as implemented in method GetHammingDistance.

All tests and diagrams must be briefly documented.

1. Implement Geo-Spatial encoder

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| Ravikumar Solanki  Gouthaami ramesh  Lakshmi Alekya Chittem | A19WISE | ravikumar.solanki@stud.fra-uas.de  gouthaami.sukardoreramesh@stud.fra-uas.de  lakshmi.chittem@stud.fra-uas.de |  |  |  |



See video for more details.

<https://youtu.be/KxxHo-FtKRo>

1. Improving of implementation of the Scalar encoder in HTM

Scalar Encoder

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| Kobra Ranjbartazehabadi  Soheil Nasiri | StarWars | [kobra.ranjbartazehabadi@stud.fra-aus.de](mailto:kobra.ranjbartazehabadi@stud.fra-aus.de) |  |  |  |

Category Encoder

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| syed faizan khursheed  Muhammad Rahim syed  Muhammad basaier | netriders | faizan.khursheed50@yahoo.com  syedm.1995@gmail.com  muhammadbasaier8@gmail.com |  |  |  |

DateTime encoder

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| Sahana Prasad  Komala Balakrishna  Prajwal Nagaraja | Group3 | sahana30p@gmail.com  komalagowda0196@gmail.com  prajwal.nagaraja@stud.fra-uas.de |  |  |  |

See this URL to understand theory of encoders:

<https://pdfs.semanticscholar.org/1455/2052a0ee729539298e4f8eb42292c0eb1622.pdf>

The goal is to show up how encoder rally works by using of various intersting examples.  
Examples should use various values for W, N, Radius, Resolution, Periodic or not periodic.

* + - 1. Review Project
      2. Create more UnitTests
         1. Provide human readable and meaningful inputs
         2. Implement asserts to specified input.
      3. Provide more examples
      4. Provide a great documentation
      5. Document the encoder on git hub. Examples: <https://nupic.docs.numenta.org/1.0.3/api/algorithms/encoders.html>

1. Improving of implementation of DateTime encoder in HTM

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| Syed Hassnat Alamgir |  | [syedhassnat@hotmail.com](mailto:syedhassnat@hotmail.com) |  |  |  |

See this URL to understand theory of encoders:

<https://pdfs.semanticscholar.org/1455/2052a0ee729539298e4f8eb42292c0eb1622.pdf>

1. Investigation of Sequence Learning of SP/TM layer

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| Muhammad Mubashir Ali Khan  DIAB ELMEHDI  Ghulam Mustafa  abdul samad  Treesa Maria Thomas | GT4 | ali.mubashir2993@yahoo.com  el.diab@stud.fra-uas.de  ghulam95mustafa@gmail.com |  |  |  |

Starting point for this project is SimpleSequenceExperiment (scalar encoder), CategorySequenceExperiment (Category Encoder) Unit Test. Note for date time encoder, unit tests is not available.   
With this unit test(s) an input sequence can be tested with various number of parameters. This project should deliver information about sequence learning behavior of SP/TM layer in dependence on various parameters.

Provide an input sequence (i.e.: [1,2,3,..,N]) and follow learning process in the trace like this one:

-------------- 0 ---------------

W: 37, 105, 192, 233, 285, 302, 354, 505, 522, 583, 607, 682, 712, 777, 913, 2953, 2998,

P: 184, 243, 283, 305, 344, 559, 569, 619, 681, 696, 935, 951, 973, 1001, 1016, 1139, 1147, 2973,

Item length: 18 Items: 381

cnt:0 0 0 = bits 2

cnt:2 2 1 = bits 7

cnt:42 42 1 = bits 11

cnt:62 62 1 = bits 17

cnt:102 102 1 = bits 18

[ 9, 0, 0, \* 1 \*, 1, 2, ]

Current Input: 0 | Predicted Input: 1

Match 0

Item length: 18 Items: 382

cnt:0 0 0 = bits 2

cnt:2 2 1 = bits 7

cnt:42 42 1 = bits 11

cnt:62 62 1 = bits 17

cnt:102 102 1 = bits 18

[ 9, 0, 0, \* 1 \*, 1, 2, ]

-------------- 1 ---------------

W: 184, 243, 283, 305, 344, 559, 569, 619, 681, 696, 935, 951, 973, 1001, 1016, 1139, 1147, 2973,

P: 361, 631, 702, 823, 863, 1108, 1190, 1259, 1347, 1380, 1427, 1469, 1538,

Item length: 13 Items: 383

cnt:4 4 2 = bits 7

cnt:44 44 2 = bits 10

cnt:64 64 2 = bits 13

[ 0, 1, 1, \* 2 \*, 2, 3, ]

Current Input: 1 | Predicted Input: 2

Match 1

Item length: 13 Items: 384

cnt:4 4 2 = bits 7

cnt:44 44 2 = bits 10

cnt:64 64 2 = bits 13

[ 0, 1, 1, \* 2 \*, 2, 3, ]

-------------- 2 ---------------

W: 361, 519, 569, 619, 631, 702, 823, 863, 951, 973, 1001, 1108, 1142, 1190, 1259, 1347, 1380, 1427, 1469, 1538,

P: 621, 715, 723, 816, 879, 986, 1043, 1075, 1199, 1233, 1338, 1387, 1411, 1442, 1689, 1719, 1764,

Item length: 17 Items: 385

cnt:4 4 2 = bits 3

cnt:6 6 3 = bits 6

cnt:44 44 2 = bits 7

cnt:46 46 3 = bits 14

cnt:66 66 3 = bits 17

[ 1, 2, 2, \* 3 \*, 3, 4, ]

Current Input: 2 | Predicted Input: 3

Match 2

Item length: 17 Items: 386

cnt:4 4 2 = bits 3

cnt:6 6 3 = bits 6

cnt:44 44 2 = bits 7

cnt:46 46 3 = bits 14

cnt:66 66 3 = bits 17

[ 1, 2, 2, \* 3 \*, 3, 4, ]

-------------- 3 ---------------

*Note, the meaning of this trace will be explained.*

Your task is to vary parameters and sequences (cells per column, column dimensions, input bits, W, N, etc.) and to document and figure out how they influence learning process.

Here are some questions of research interest:

How many iterations are required to learn the sequence by a given set of parameters?

How W, N and other encoder parameters influence learning process?

Is the learned sequence encoded in cells and columns stable during learning process?

…

Example: Learned encoded cells for double number 6.0 looks in some experiment like shown below:

6

5500, 5501, 5502, 5503, 5504, 5505, 5506, 5507, 5508, 5509, 5510, 5511, 5512, 5513, 55

5500, 5501, 5502, 5503, 5504, 5505, 5506, 5507, 5508, 5509, 5510, 5511, 5512, 5513, 55

5500, 5501, 5502, 5503, 5504, 5505, 5506, 5507, 5508, 5509, 5510, 5511, 5512, 5513, 55

5500, 5501, 5502, 5503, 5504, 5505, 5506, 5507, 5508, 5509, 5510, 5511, 5512, 5513, 55

5100, 5101, 5102, 5103, 5104, 5105, 5106, 5107, 5108, 5109, 5110, 5111, 5112, 5113, 51

5100, 5101, 5102, 5103, 5104, 5105, 5106, 5107, 5108, 5109, 5110, 5111, 5112, 5113, 51

5100, 5101, 5102, 5103, 5104, 5105, 5106, 5107, 5108, 5109, 5110, 5111, 5112, 5113, 51

5100, 5101, 5102, 5103, 5104, 5105, 5106, 5107, 5108, 5109, 5110, 5111, 5112, 5113, 51

5100, 5101, 5102, 5103, 5104, 5105, 5106, 5107, 5108, 5109, 5110, 5111, 5112, 5113, 51

5100, 5101, 5102, 5103, 5104, 5105, 5106, 5107, 5108, 5109, 5110, 5111, 5112, 5113, 51

5100, 5101, 5102, 5103, 5104, 5105, 5106, 5107, 5108, 5109, 5110, 5111, 5112, 5113, 51

5100, 5101, 5102, 5103, 5104, 5105, 5106, 5107, 5108, 5109, 5110, 5111, 5112, 5113, 51

5100, 5101, 5102, 5103, 5104, 5105, 5106, 5107, 5108, 5109, 5110, 5111, 5112, 5113, 51

5100, 5101, 5102, 5103, 5104, 5105, 5106, 5107, 5108, 5109, 5110, 5111, 5112, 5113, 51

5140, 5343, 5413, 5543, 6093, 6107, 6280, 6480, 6585, 6780, 7277, 7641, 8196, 8203,

5140, 5343, 5413, 5543, 6093, 6107, 6280, 6480, 6585, 6780, 7277, 7641, 8196, 8203,

5140, 5343, 5413, 5543, 6093, 6107, 6280, 6480, 6585, 6780, 7277, 7641, 8196, 8203,

5140, 5343, 5413, 5543, 6093, 6107, 6280, 6480, 6585, 6780, 7277, 7641, 8196, 8203,

5140, 5343, 5413, 5543, 6093, 6107, 6280, 6480, 6585, 6780, 7277, 7641, 8196, 8203,

5140, 5343, 5413, 5543, 6093, 6107, 6280, 6480, 6585, 6780, 7277, 7641, 8196, 8203,

5140, 5343, 5413, 5543, 6093, 6107, 6280, 6480, 6585, 6780, 7277, 7641, 8196, 8203,

5140, 5343, 5413, 5543, 6093, 6107, 6280, 6480, 6585, 6780, 7277, 7641, 8196, 8203,

5140, 5343, 5413, 5543, 6093, 6107, 6280, 6480, 6585, 6780, 7277, 7641, 8196, 8203,

As you see encoded representation changes for every set of learning iterations (cycle). You should investigate how stability can be achieved by choosing different parameters. Ideally, encoded representation remain stable after a number of cycles.

To be able to research in this experiment, you will have to understand the background of Spatial Pooler, Temporal Pooler and the HTMClassifier used in this experiment.

All results must be documented in form of many diagrams. Provide also comments and discussion.

Useful URLs:

<https://discourse.numenta.org/t/unusual-prediction-behaviour-in-htm-js/5980/5>

<http://www.paulscode.com/demos/htm.js/demos/piano/index.htm>

This project can be done by multiple students. Groups should perform described analysis based on Scalar encoder.

Try to play with different sequences. We are looking for influence of different configuration parameters. For example:

1. Quality of sequence learning when number of encoder input bits is higher or less than number of columns. The quality can be measured by the number of cycles (repeatings) of the sequence needed to fully learn the sequence.
2. How good is learning when number of cells per column changes.
3. Execution time of experiment in dependence on number of input bits, number of columns and number of cells per column.
4. Execution time of experiment in dependence on number of input vectors.
5. Validate if there is any influence of the pattern of data in the sequence to quality or speed of learning. For example, following sequence might be (more) difficult to learn: 1,2,3,1,5,3,1,6,3.  
   The reason for this is that the digit 1 has different followers 1->2, 1->5 and 1->6.  
   Validate and describe how such sequence can be learned in comparison to sequence 1,2,3,4,5,6,7,8,9.
6. Capture number of required cycles to reach stable accuracy of 100% by the same input sequence Stable accuracy is when for example 100% accuracy remains for 10 cycles. Music Experiment demonstrates exactly how to exist experiment with stable accuracy.
7. Investigate learning when using global vs. local inhibition.

### Useful findings

Following finding can be useful during investigation.

#### Experiment configuration:

See PowerConsumptionExperiment.cs

*Spatial Pooler:*

Parameters p = Parameters.getAllDefaultParameters();

p.Set(KEY.RANDOM, new ThreadSafeRandom(42));

p.Set(KEY.COLUMN\_DIMENSIONS, new int[] { 2048 });

p.Set(KEY.INPUT\_DIMENSIONS, new int[] { inputBits });

p.Set(KEY.CELLS\_PER\_COLUMN, 10);

p.Set(KEY.GLOBAL\_INHIBITION, true);

// N of 40 (40= 0.02\*2048 columns) active cells required to activate the segment.

p.setNumActiveColumnsPerInhArea(0.02 \* 2048);

// Activation threshold is 10 active cells of 40 cells in inhibition area.

p.setActivationThreshold(10 );

p.setInhibitionRadius(15);

*Encoder*

Note in all experiments the last column (value) was learned. Date and time were omitted.

##### *Sequence 1:*

7/2/2010 0:00,1.0

7/2/2010 1:00,2.0

7/2/2010 2:00,3.0

7/2/2010 3:00,4.0

7/2/2010 4:00,5.0

7/2/2010 5:00,6.0

7/2/2010 6:00,7.0

7/2/2010 7:00,8.0

7/2/2010 8:00,9.0

7/2/2010 9:00,10.0

*Result:  
Enters learned state with accuracy of 100% in 21 cycles.*

##### *Sequence 2:*

7/2/2010 0:00,1.0

7/2/2010 1:00,2.0

7/2/2010 2:00,3.0

7/2/2010 3:00,4.0

7/2/2010 4:00,5.0

7/2/2010 5:00,6.0

7/2/2010 6:00,7.0

7/2/2010 7:00,8.0

7/2/2010 8:00,9.0

7/2/2010 9:00,10.0

7/2/2010 0:00,11.0

7/2/2010 1:00,12.0

7/2/2010 2:00,13.0

7/2/2010 3:00,14.0

7/2/2010 4:00,15.0

7/2/2010 5:00,16.0

7/2/2010 6:00,17.0

7/2/2010 7:00,18.0

7/2/2010 8:00,19.0

7/2/2010 9:00,20.0

*Result:  
Enters learned state with accuracy of 100% in 13 cycles.*

1. Implement CLA Classifier

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| Sami Patwary  Sami Patwary  Meem Shahrin  Kazi Md Wahidul Gani  Riyad-Ul- Islam  Mou Saha  Anik Saha | Super5 | samipatwary@gmail.com  samipatwary@gmail.com  shahrinanushua@gmail.com |  |  |  |

CLA Classifier is an implementation of classification algorithm in JAVA. Your task is to learn about this classifier and to provide an implementation in .NET Core.

<https://github.com/numenta/htm.java/blob/master/src/main/java/org/numenta/nupic/algorithms/CLAClassifier.java>

<https://github.com/UniversityOfAppliedSciencesFrankfurt/se-cloud-2019-2020/blob/master/Source/HTM/NeoCortexApi/Network/HtmClassifier.cs>

1. Implement SDR Classifier

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| Jagdeep Singh  Sanjana Sharma  Vishakha Choudhary Babulal  chetan chadha  maria majid  sidra hussain | A1920 | jagdeep.singh@stud.fra-uas.de  sanjana.sharma@stud.fra-uas.de  [vishakha.babulal@stud.fra-uas.de](mailto:vishakha.babulal@stud.fra-uas.de)  chetan.chadha@stud.fra-uas.de  mariakhan.mm92@gmail.com  sidra.hussain2633@gmail.com |  |  |  |

CLA Classifier is an implementation of classification algorithm in JAVA. Your task is to learn about this classifier and to provide an implementation in .NET Core.

<https://github.com/numenta/htm.java/blob/master/src/main/java/org/numenta/nupic/algorithms/CLAClassifier.java>

<https://github.com/UniversityOfAppliedSciencesFrankfurt/se-cloud-2019-2020/blob/master/Source/HTM/NeoCortexApi/Network/HtmClassifier.cs>

1. Performance Spatial Pooler Global vs. Local Inhibition

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| Thanh Quang Nguyen  Quang Trung Tran | Group A1 | thanhquang0912@gmail.com  tranquangtrung.vgu@gmail.com |  |  |  |

In this project you are required to create experiments, which compare learning performance of Spatial Pooler, when configured for Local and Global inhibition. You have to first understand the meaning of both inhibition algorithms and then decide which input data and set of input arguments need to be used. You should measure the speed of learning in dependence on multiple input parameters for both algorithms. Finally compere one algorithm vs. another one.

1. Serialization of Spatial Pooler

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| Anik Biswas  Lakshmi Mounika Kolisetty  Manash Chakraborty  Raghavendra Yarlagadda | GroupSW | aniklog@gmail.com  lakshmi.kolisetty@stud.fra-uas.de  Manash. [Chakraborty@stud.fra-uas.de](mailto:Chakraborty@stud.fra-uas.de)  ryarlagadda404@gmail.com |  |  |  |

Current implementation of Spatial Pooler does not support serialization. The idea is to provide serialization capabilities for algorithms. Following must be fulfilled:

var alg1 = New Algorithm();  
alg1.Train(dataSubset1);  
alg1.Persist(fileName);

var alg2 = ALGORITHM.Load(fileName);  
alg2.Train(dataSubset2);

Persist method saves the full state of algorithm to a file. Method load loads the algorithm state from file. The goal is to be able to provide continuous learning.

EXAMPLE

public class ABC{

public string Name {get;set}

public string EMail{get;set

}

ABC x;

if(File.Exists("filename.json") == false)

{

string name = Console.ReadLine();

string eMail = Console.ReadLine();

x = new ABC();

x.Name = name;

x.EMail = email;

// SERIALIZATION

JSonConvert.Serialize(x, "filename.json");

}

else

{

// DESERIALIZATION

x = JsonConvert.Load("filename.json");

}

{

"Name": "My Name",

"EMail": "abc@bl.com",

}

<https://www.newtonsoft.com/json/help/html/SerializingJSON.htm>

1. Serialization of Temporal Memory

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| Yousaf Safdar  Nizam Jan  Ali Firouzi | Black Panthers | safdar@stud.fra-uas.de  [njan@stud.fra-uas.de](mailto:njan@stud.fra-uas.de)  [ali.firouzi.ac@gmail.com](mailto:ali.firouzi.ac@gmail.com) |  |  |  |

Current implementation of TM does not support serialization. The idea is to provide serialization. Following must be fulfilled:

var alg1 = New Algorithm();  
alg1.Train(dataSubset1);  
alg1.Persist(fileName);

var alg2 = ALGORITHM.Load(fileName);  
alg2.Train(dataSubset2);

Persist method saves the full state of algorithm to a file. Method load loads the algorithm state from file. The goal is to be able to provide continuous learning.

You can use following unit test to train algorithm, which will be serialized.

public void MusicNotesExperiment()

Take also a look on following test:

public void SerializationTest()

It demonstrates how to work with JSON serializer.

1. Validate Memorizing capabilities of SpatialPooler

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| Thanh Toan Truong  Hoang Hai Pham | HTGroup | [noath2302@gmail.com](mailto:noath2302@gmail.com)  [hhaipham21@gmail.com](mailto:hhaipham21@gmail.com) |  |  |  |

Validate if learning sequence does matter

1. Create many inputVectors
2. Train SP on InputVectors: N\*I1, N\*I2,..,N\*In  
   Pseudo code:  
     
   foreach inputVector in inputVectors

for i=0 to N  
 sp.compute(inputVector, activeColumns, true)

1. Use variable number of iterations N and follow how SP enters stable state for every vector.
2. Test Prediction

After training of all vectors, memorize activecolumns (output of SP)  
Run SP.compute with false and check if active columns are as expected.

1. Test same 1-4 when training vectors after each other: N\*(I1,I2,..,In)

for i=0 to N  
 foreach inputVector in inputVectors

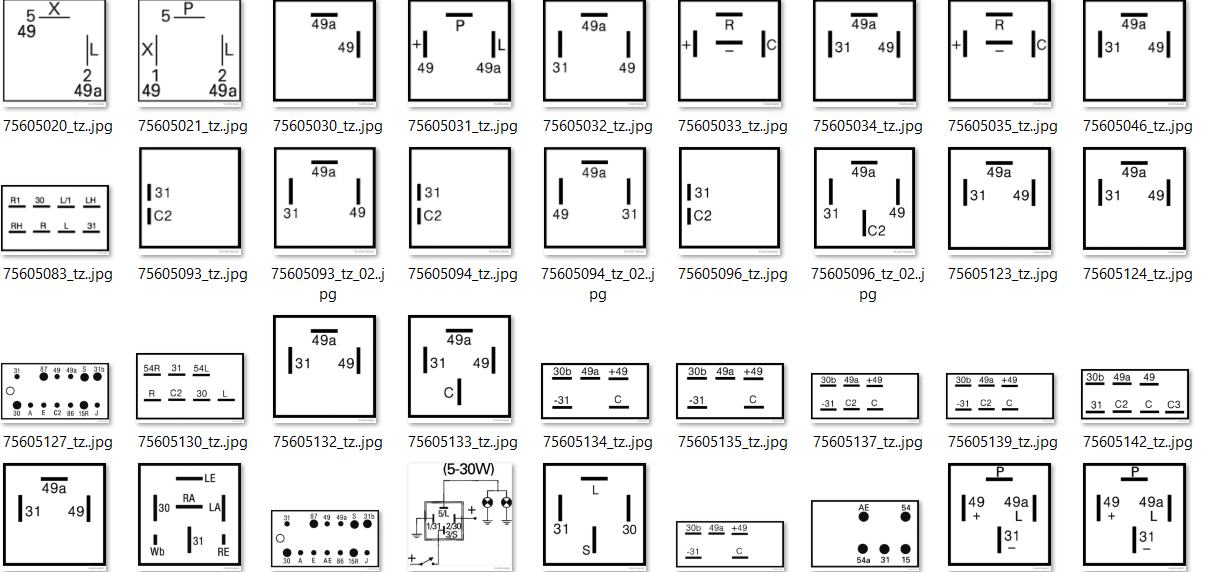
sp.compute(inputVector, activeColumns, true)  
  
If there is a difference then this might be an indicator for cognitive concentration capability.

To calculate learning of SP use technique of *NoiseTest* method in *SpatialPoolerResearchTests*

1. Schema Image Classification

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| Md Mahbub Ul Alam  Tanvir Hasan  Md Mijanur Rahaman  Abu Syeem Syeem  Mir Mehedi Hasan Rayhan  Md Zahidul Islam |  | imleon2010@gmail.com  tanvir.bsmrstu@gmail.com  mahedi.siu@gmail.com  mijan039@gmail.com  syeem007@gmail.com  jahidkaysar@gmail.com | Trojan Horse |  |  |

In this project, you should train the Spatial Pooler (with or without Temporal Memory) to classify images shown below.



Some already existing experiments (see LearningimageDoubleShiftStble in SpatialPoolerResearchTests.cs) demonstrate how to train images with SP. In this project you should calculate the hamming distance between

1. Investigate SpatialPooler noise robustnes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
| Phuoc Sang Nguyen  Duy Thanh Nguyen | VNGroup | phuocsangnguyen97@gmail.com  ngthanhduy7@gmail.com |  |  |  |

Motivation in this work is to analyze how encoding of input pattern itself influences the robustness of SDR against noise when using Spatial Pooler.

In this work you should provide different (every student inside of the group use own data) input data. For example some power consumption curve or any other kind of input.  
As next SP is trained with the ideal curve (data). Then data data is obfuscated with noise (see NoiseTest in SpatialPoolerResearchTests.cs).

It should be validated if the ideal curve is predicted with SP with provided the input with the noise.

Follow same approach as shown in the test named above.

For more information look on Influence of input sparsity to “Hierarchical Temporal Memory Spatial Pooler noise robustness v2.docx”

1. Template Project

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Matriculation Number** | **Email** | **Path** | **State** | **Quality** |
|  |  |  |  |  |  |

# Individual Projects

## Improving of quality of LearningApi (IP 18/19-1)

Within this individual project you will learn how to improve quality of software projects, which are in code-complete state. Such projects are typically declared as done. This means, that code, unit-tests and documentation are completed. However, usually deliverables of projects in this state is typically not market ready.

In this project the quality of LearningAPI framework must be increased. Your task is to review documentation and current state of all algorithms. You will compare your results with already existing reviews and make decisions how to improve the quality.

On the end of the project, the high or higher-quality documentation, code and unit tests have to be committed in the master branch of LearningAPI on git-hub.

Documentation should be written in Markdown.

## Implementation of Service Bus actor model NET

During last semester, Service Bus Actor Model framework has been implemented to support parallel implementation of various machine learning algorithms.

The goal of this project is to finalize, document and publish existing implementation. Current implementation require more unit tests, some initialization(setup) code good documentation and more examples. Finally, the project will be published to github.

## Validate Memorizing capabilities of SpatialPooler

### Validate if learning sequence does matter

1. Create many inputVectors
2. Train SP on InputVectors: N\*I1, N\*I2,..,N\*In
3. Use variable number of iterations N and follow how SP enters stable state for every vector.
4. Test Prediction

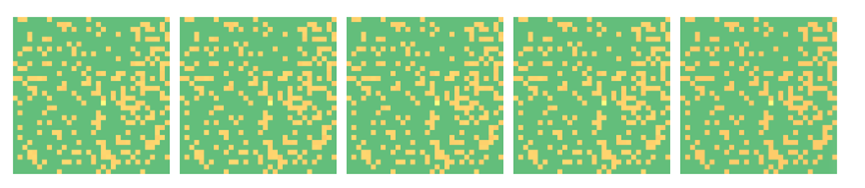
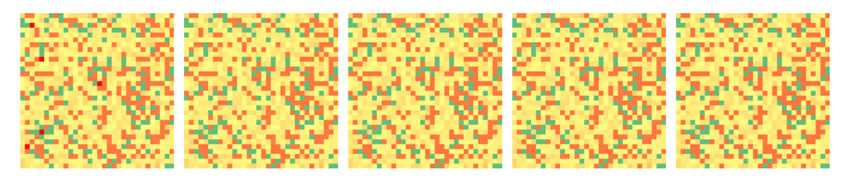
After training of all vectors, memorize activecolumns (output)  
Run SP.compute with false and check if active columns are as expected.

1. Test same 1-4 when training vectors after each other: N\*(I1,I2,..,In)  
   If there is a difference then this might be an indicator for cognitive concentration capability.

To calculate learning of SP use results of *NoiseTest* method in *SpatialPoolerResearchTests*.

### Boosting and Headmap

Compare how inhibition and boosting work from iteration to iteration and show heatmaps. Use already existing helper methods to create heatmaps. Note that higher heat (energy) means better stability of trained active columns.



### Capacity

Train SP to many input vectors with some noise and approve capacity. Validate if stability of learned patterns (vectors) change with increased number of input vectors.

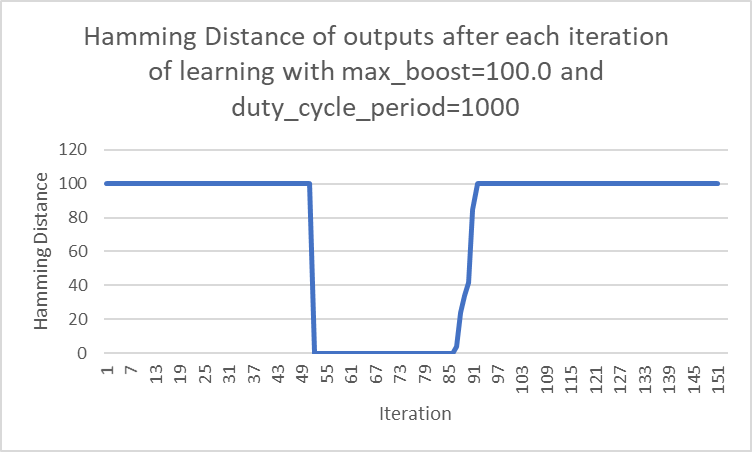
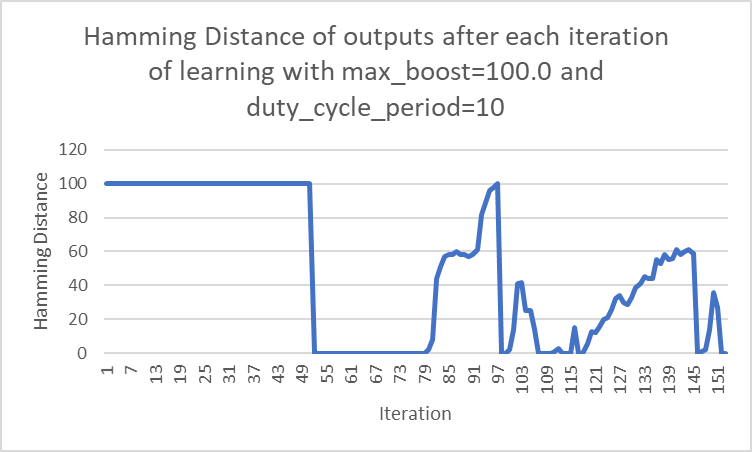
Related project:  
<https://github.com/UniversityOfAppliedSciencesFrankfurt/se-dystsys-2018-2019-softwareengineering/issues/139>

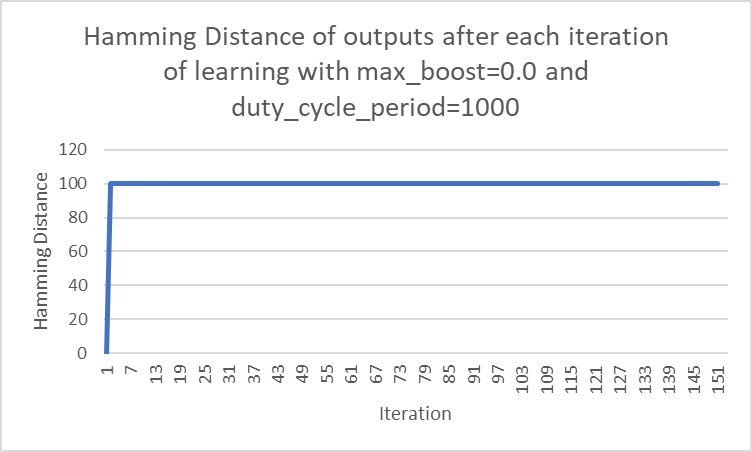
## Improve HTM boosting algorithm

Right now, Spatial Pooler provides a boosting algorithm, which bumps up columns. It makes sure that all of columns become active in learning process.  
This behavior

Boosting is an implementation of homeostatic plasticity mechanism of the brain. This mechanism regulates excitability of neurons relative to their activity in the area (possibly in cortical column). HTM CLA provides a boosting mechanism as computational representation of plasticity. The current boosting implementation, makes sure that columns get uniformly activated (at least used), but it does not produce good learning results. First results show, that current implementation of boosting algorithm in some cases leads to instability of the learned spatial pattern. As shown in figures below, boosting enforces instability, which is not wanted behavior. How do we understand stability? After learning of some pattern is done (usually 2-3 learning steps), Spatial Pooler produces the code in set of active columns. For example, by 1000 columns, learning process will activate columns 1,2,3 and 5. (usually 2% of columns, depending on configuration).

Once SP has learned the pattern, active columns will not change when that pattern appears again. IN our stability tests, we have trained same pattern over and over again. After every training step, weh have compared hamming distance (overlap) between last and previous step. If there is no any change in output in last and previous stem, we consider the system as stable.





Stability of learning process in dependence on boosting settings. In this experiment MNIST image 28x28 was used to iteratively learn Spatial Pooler. X-axis shows iteration cycle and Y – axis shows hamming distance (overlap) between column output in iteration k and column output in previous iteration k-1. Figure on top shows that SP learns very quickly (usually 1-3 iteration steps) and gets stable. After number of iterations defined by duty\_cycle period boosting algorithm gets activated and SP start forgetting and gets instable. After a while depending on dyty\_cycle SP will get stable again (figure in the middle – iteration cycle >85). With complete deactivation of boosting (figure on bottom) SP learns and remains stable.

Interestingly, according to some papers, boosting in L4 (layer in brain) seems to be only active in new-born species. This means that boosting biologically uniformly activates columns by learning unsupervised patterns from environment.

In this project, Spatial Pooler should be redesigned to distinguish between *new-born* and *learning* stage. Right now, SP does not include a concept of new-born state.

## Persistence of Spatial Pooler

Implement persistence feature of spatial pooler and temporal memory algorithm inside of HTT CLA. Learned model has to be saved and loaded. Currently there are already 3 implementations for persistence:

* Table Storage
* Blob Storage
* File Systems

In this project Blob Storage persistence provider must be improved. Right now, persistence work by using of JSON serialization. Because JSON occupies to much memory, current implementation is not efficient.   
The goal of this project id to find out some other format (i.e.: some binary format) to store ond load data.

## Docker and Kubernetes support for HTM

The goal of this project is conternerize current implementation of HTM algorithm. Final version of HTM should run in docker container in Azure Kubernetes.

## Song identification HTM

<https://www.semanticscholar.org/paper/Song-Identification-Using-the-Numenta-Platform-for-Schey/e64ed1dfc77d640e4a697c0c8e04d9a6ac63e62c>

# Thesis

## Parallel implementation of CNN with actor model (TH 18/19-1)

## Parallel implementation of RBM with actor model (TH 18/19-2)

## Parallel implementation of K-Means with actor model (TH 18/19-3)