Pima Indians Diabetes Database-Predict the onset of diabetes based on diagnostic measures

深度學習:基礎及應用

106753020 資科碩三 李俊毅
108753205 資科碩一 蔡宗諺
108753208 資科碩一 葉冠宏

INTRODUCTION

- Datasets is originally from the National Institute of Diabetes and Digestive and Kidney Diseases.
- The objective of the dataset is to diagnostically predict whether or not a patient has diabetes.
- All patients here are females at least 21 years old of Pima Indian heritage.

INTRODUCTION

- Datasets(Real number input, Binary output)

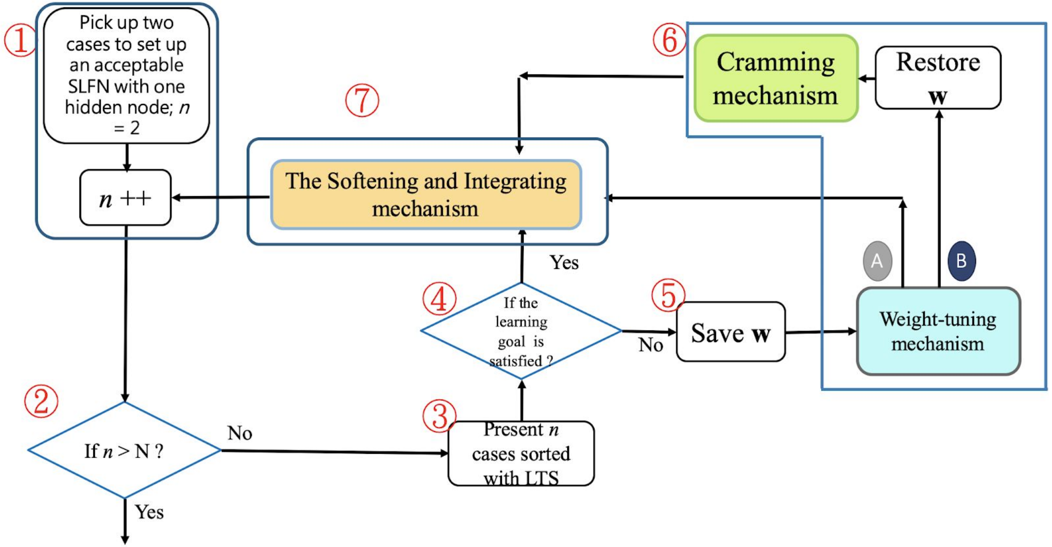
Variables	Description
Pregnancies	Number of times pregnant
Glucose	Plasma glucose concentration a 2 hours in an oral glucose tolerance test
BloodPressure	Diastolic blood pressure (mm Hg)
SkinThickness	Triceps skin fold thickness (mm)
Insulin	2-Hour serum insulin (mu U/ml)
BMI	Body mass index (weight in kg/(height in m)^2)
DiabetesPedigreeFunction	Diabetes pedigree function
Age	Age (years)

RELATED WORK

- Classification ability of single hidden layer feedforward neural networks.
- Sales forecasting using extreme learning machine with applications in fashion retailing.
- Single-layer learning revisited: a stepwise procedure for building and training a neural network.

METHODS

- Single Hidden Layer Feedforward Neural Network(SLFN)



METHODS

Data Preprocessing

1. Use MinMaxScaler to preprocess the data.
2. Train only two data set to get 100% prediction rate and initial weight.

$$X_{new} = \frac{X_i - \min(X)}{\max(x) - \min(X)}$$

METHODS

Literature Review

Least Trimmed Squares (LTS)

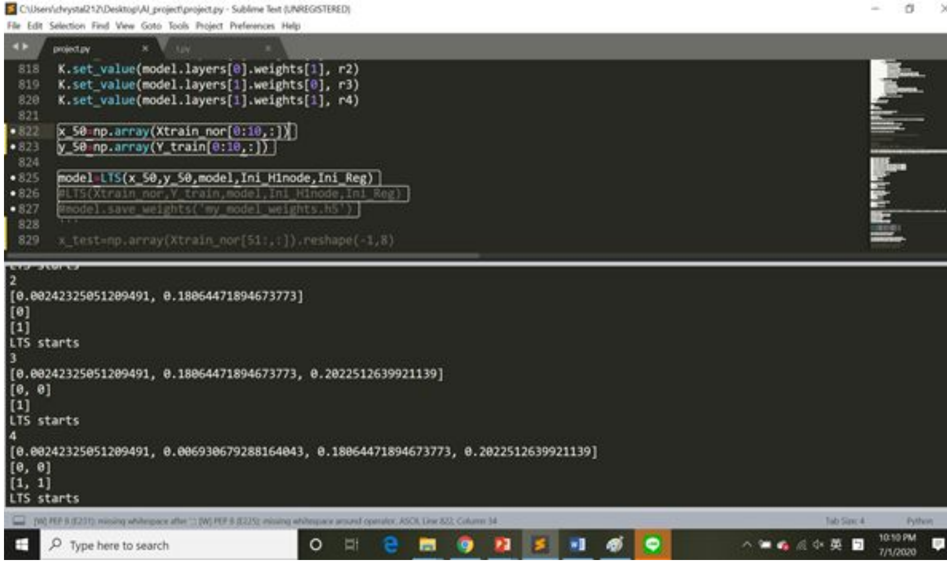
(Tsaih & Cheng, 2009; Tsaih, et al., 2018)

- Pick up the first n reference observations $\{(\mathbf{x}^c, y^c)\}$ which are sorted by all N reference observations' squared residuals in ascending order.

$$(e^{[1]})^2 \le (e^{[2]})^2 \le \ldots \le (e^{[n]})^2, \text{ in which } (e^c)^2 = (f(\mathbf{x}^c, \mathbf{w}) - y^c)^2$$

- Detect the potential outliers in (Tsaih & Cheng, 2009; Tsaih, et al., 2018)

- Accelerate the learning process and mimic the human learning



METHODS

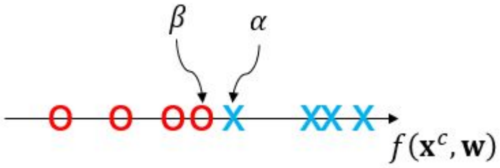
Literature Review

Condition L (Tsaih, 1993)

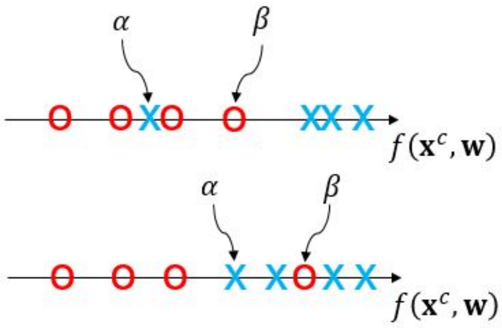
$y^c = 1$ all $c \in \mathbf{I}_1$; $y^c = -1$ all $c \in \mathbf{I}_2$

X : $f(\mathbf{x}^c, \mathbf{w})$, all $c \in \mathbf{I}_1$

O : $f(\mathbf{x}^c, \mathbf{w})$, all $c \in \mathbf{I}_2$



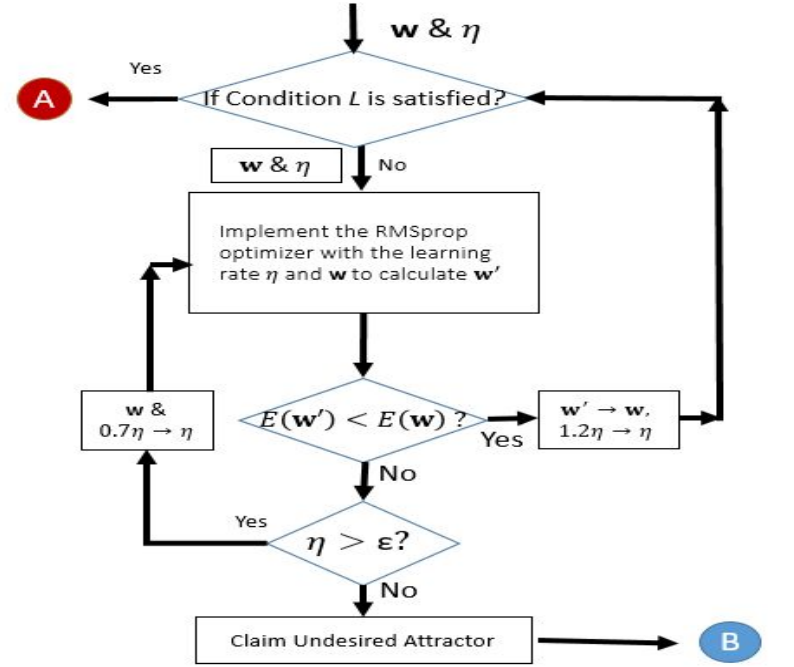
$\alpha > \beta$
Condition L : True

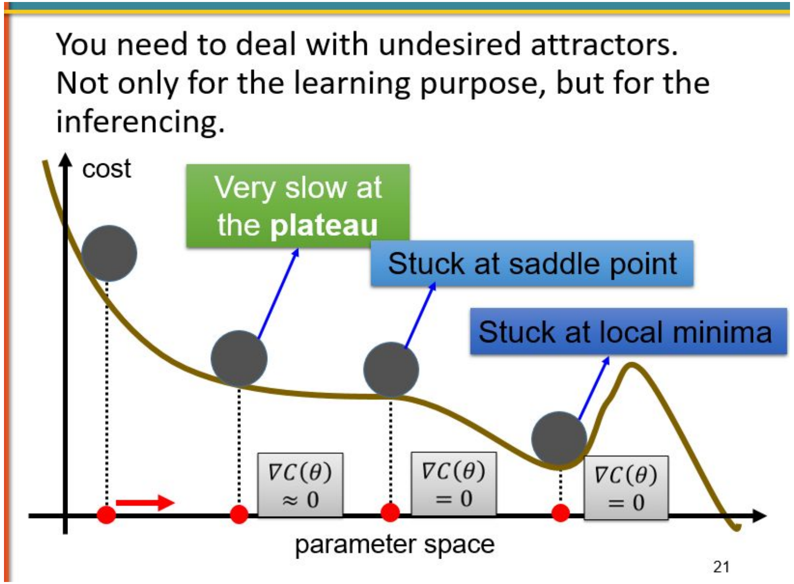


$\alpha < \beta$
Condition L : False

METHODS

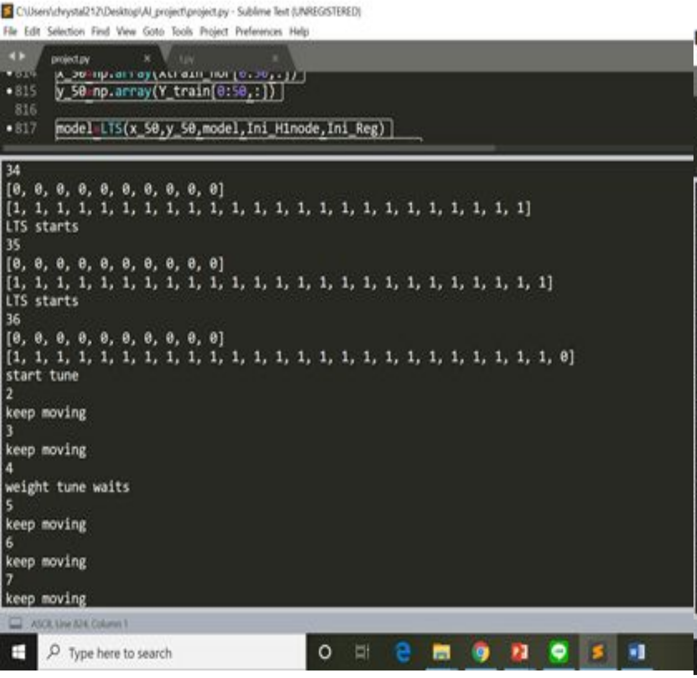
- Weight-tuning mechanism

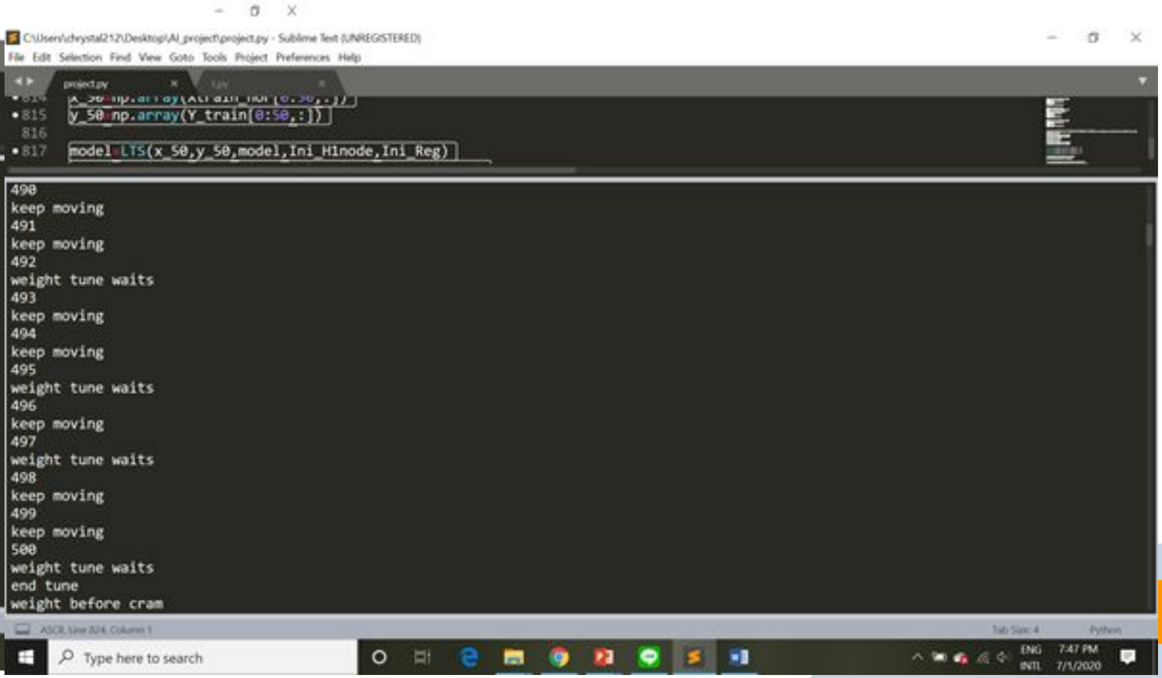




METHODS

- Weight-tuning mechanism



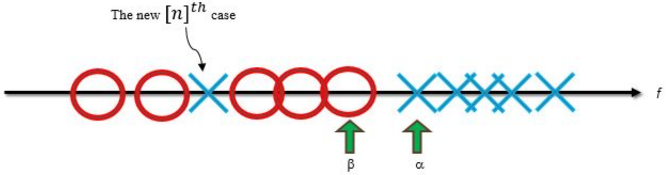


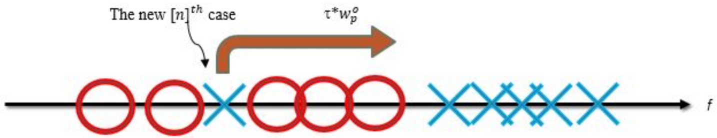
METHODS

Cramming mechanism

using the LTS

When we encounter with a new case that cannot be learnt well with the current SLFN and the weight-tuning mechanism, an extra hidden node is used to cram this case.



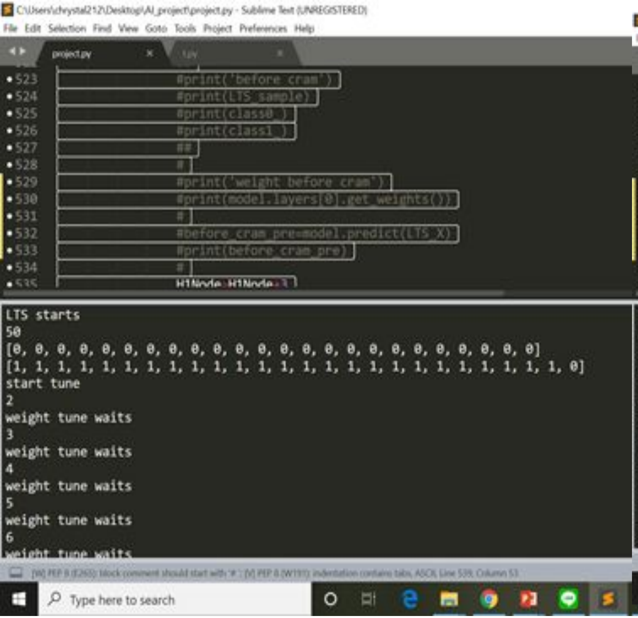


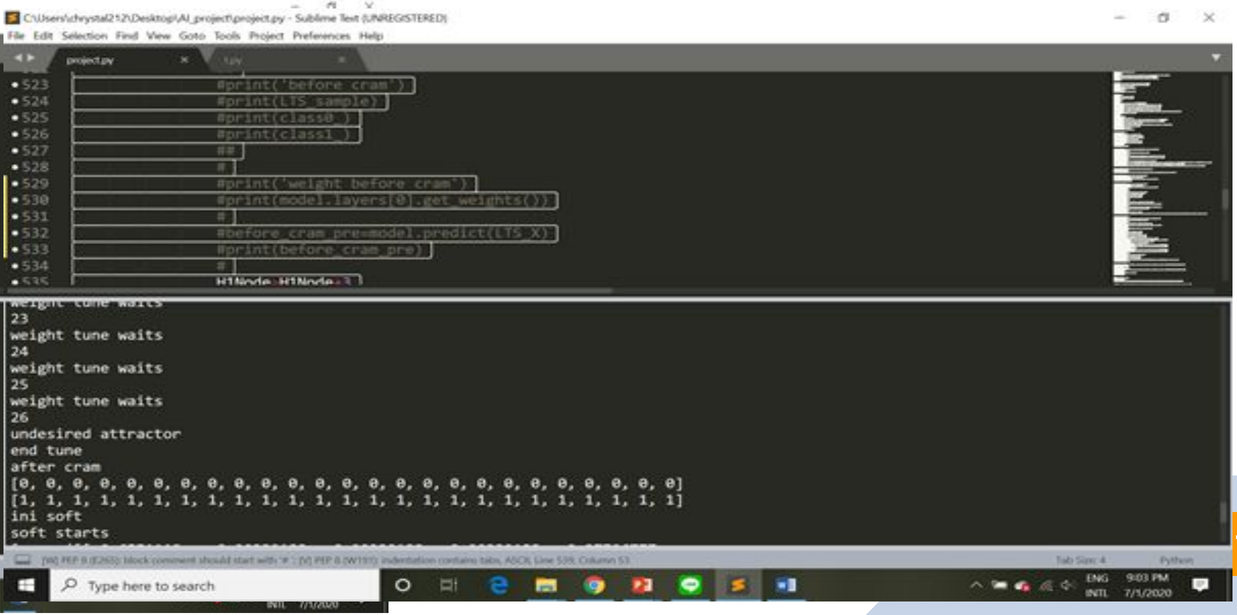
- Via adding an extra hidden node, we would like to pull this new case into the right place with other cases staying still at the same place (for instance, in the above scenario).
- set up the τ value that renders $w_p^o \tau > \max_{u \in \mathbf{I}_2(n)} \sum_{i=1}^{p-1} w_i^o a_i^u > \sum_{i=1}^{p-1} w_i^o a_i^{[n]}$ be true.

A hyper-parameter

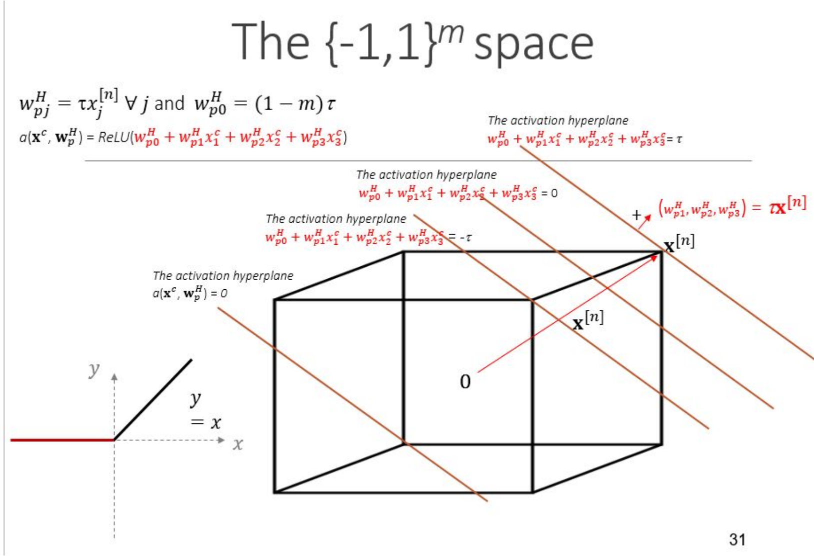
METHODS

- Cramming mechanism

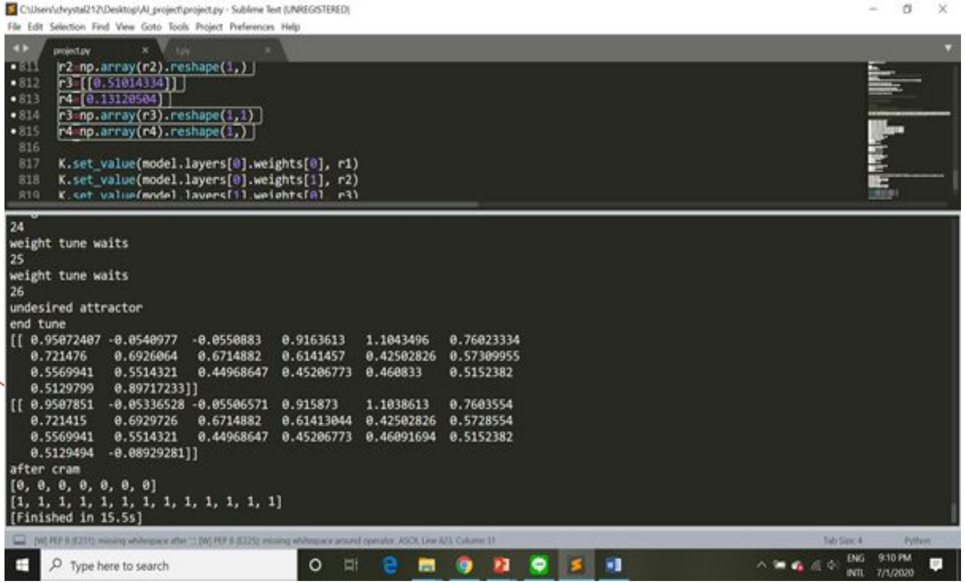




METHODS



The {-1,1}^{m} space
$w_{pj}^H = \tau x_j^{[n]}\ \forall j$ and $w_{p0}^H = (1-m)\tau$
$a(\mathbf{x}^c, \mathbf{w}_p^H) = ReLU(w_{p0}^H + w_{p1}^H x_1^c + w_{p2}^H x_2^c + w_{p3}^H x_3^c)$
The activation hyperplane
$w_{p0}^H + w_{p1}^H x_1^c + w_{p2}^H x_2^c + w_{p3}^H x_3^c = \tau$
The activation hyperplane
$w_{p0}^H + w_{p1}^H x_1^c + w_{p2}^H x_2^c + w_{p3}^H x_3^c = 0$
The activation hyperplane
$w_{p0}^H + w_{p1}^H x_1^c + w_{p2}^H x_2^c + w_{p3}^H x_3^c = -\tau$
The activation hyperplane
$a(\mathbf{x}^c, \mathbf{w}_p^H) = 0$
$(w_{p1}^H, w_{p2}^H, w_{p3}^H) = \tau\mathbf{x}^{[n]}$
+
$\mathbf{x}^{[n]}$
$\mathbf{x}^{[n]}$
0
y
$y = x$
x
31

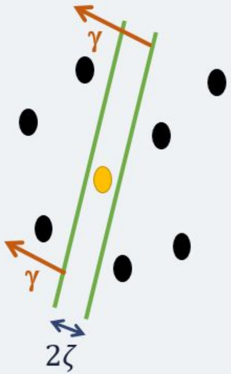


C:\Users\chrystal212\Desktop\AI_project\project.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
project.py
r2=np.array(r2).reshape(1,)
r3=[[0.51014334]]
r4=[0.13120504]
r3=np.array(r3).reshape(1,1)
r4=np.array(r4).reshape(1,)
K.set_value(model.layers[0].weights[0], r1)
K.set_value(model.layers[0].weights[1], r2)
24
weight tune waits
25
weight tune waits
26
undesired attractor
end tune
[[0.95072407 -0.0540977 -0.0550883 0.9163613 1.1043496 0.76023334
0.721476 0.6926064 0.6714882 0.6141457 0.42502826 0.57309955
0.5569941 0.5514321 0.44968647 0.45206773 0.460833 0.5152382
0.5129799 0.89717233]]
[[0.9507851 -0.05336528 -0.05506571 0.915873 1.1038613 0.7603554
0.721415 0.6929726 0.6714882 0.61413044 0.42502826 0.5728554
0.5569941 0.5514321 0.44968647 0.45206773 0.46091694 0.5152382
0.5129494 -0.08929281]]
after cram
[0, 0, 0, 0, 0, 0, 0]
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
[Finished in 15.5s]
Type here to search

METHODS

The Cramming Mechanism

Step 1: Let ζ be a given small number. Find an m-vector $\boldsymbol{\gamma}$ of length one such that $\gamma^{\mathrm{T}}(\mathbf{x}^c-\mathbf{x}^{[n]}) \neq 0$ all $c \in \mathrm{I}(n)-\{[n]\}$ AND $(\zeta+\gamma^{\mathrm{T}}(\mathbf{x}^c-\mathbf{x}^{[n]}))*(\zeta-\gamma^{\mathrm{T}}(\mathbf{x}^c-\mathbf{x}^{[n]})) < 0$ all $c \in \mathrm{I}(n)-\{[n]\}$.



Step 2: Let $p+3 \rightarrow p$ and add three new hidden nodes p-2th, p-1th and pth to the existing SLFN with

- $w^H_{p-2,0} = \zeta-\gamma^{\mathrm{T}}\mathbf{x}^{[n]}$, $\mathbf{w}^H_{p-2} = \boldsymbol{\gamma}$,
- $w^H_{p-1,0} = -\gamma^{\mathrm{T}}\mathbf{x}^{[n]}$, $\mathbf{w}^H_{p-1} = \boldsymbol{\gamma}$,
- $w^H_{p0} = -\zeta-\gamma^{\mathrm{T}}\mathbf{x}^{[n]}$, $\mathbf{w}^H_{p} = \boldsymbol{\gamma}$, and
- $w^o_{p-2} = -0.5w^o_{p-1} = w^o_p = 1.1(\max_{u \in \mathbf{I}_2(n)} \sum_{i=1}^{p-3} w_i^o a_i^u - \sum_{i=1}^{p-3} w_i^o a_i^{[n]})/\zeta$

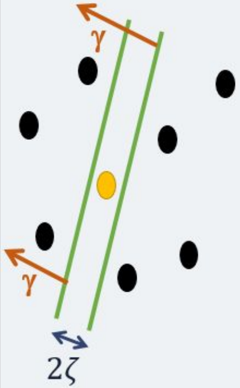
τ = 1.1

Case of $y^{[n]} = 1.0$

37

The Cramming Mechanism

Step 1: Let ζ be a given small number. Find an m-vector $\boldsymbol{\gamma}$ of length one such that $\gamma^{\mathrm{T}}(\mathbf{x}^c-\mathbf{x}^{[n]}) \neq 0$ all $c \in \mathrm{I}(n)-\{[n]\}$ AND $(\zeta+\gamma^{\mathrm{T}}(\mathbf{x}^c-\mathbf{x}^{[n]}))*(\zeta-\gamma^{\mathrm{T}}(\mathbf{x}^c-\mathbf{x}^{[n]})) < 0$ all $c \in \mathrm{I}(n)-\{[n]\}$.



Step 2: Let $p+3 \rightarrow p$ and add four new hidden nodes p-2th, p-1th and pth to the existing SLFN with

- $w^H_{p-2,0} = \zeta-\gamma^{\mathrm{T}}\mathbf{x}^{[n]}$, $\mathbf{w}^H_{p-2} = \boldsymbol{\gamma}$,
- $w^H_{p-1,0} = -\gamma^{\mathrm{T}}\mathbf{x}^{[n]}$, $\mathbf{w}^H_{p-1} = \boldsymbol{\gamma}$,
- $w^H_{p0} = -\zeta-\gamma^{\mathrm{T}}\mathbf{x}^{[n]}$, $\mathbf{w}^H_{p} = \boldsymbol{\gamma}$, and
- $w^o_{p-2} = -0.5w^o_{p-1} = w^o_p = 1.1(\min_{u \in \mathbf{I}_1(n)} \sum_{i=1}^{p-3} w_i^o a_i^u - \sum_{i=1}^{p-3} w_i^o a_i^{[n]})/\zeta$

τ = 1.1

Case of $y^{[n]} = -1.0$

38

METHODS

Contribution of three extra hidden nodes to the output

- $\mathbf{x}^c$: *whose* $\boldsymbol{\gamma}^T(\mathbf{x}^c - \mathbf{x}^{[n]})$ is either greater than ζ or less than $-\zeta$
- $\Delta f = w_p^o$ *[ReLU($\zeta+\boldsymbol{\gamma}^T(\mathbf{x}^c-\mathbf{x}^{[n]})$) - 2ReLU($\boldsymbol{\gamma}^T(\mathbf{x}^c-\mathbf{x}^{[n]})$) + ReLU($-\zeta+\boldsymbol{\gamma}^T(\mathbf{x}^c-\mathbf{x}^{[n]})$)]

$\boldsymbol{\gamma}^T(\mathbf{x}^c-\mathbf{x}^{[n]})$	$(p-2)^{th}$	$-2(p-1)^{th}$	p^{th}	Δf
-2ζ	0	0	0	0
$-\zeta$	0	0	0	0
-0.7ζ	0.3ζ	0	0	$0.3\zeta w_p^o$
-0.5ζ	0.5ζ	0	0	$0.5\zeta w_p^o$
-0.3ζ	0.7ζ	0	0	$0.7\zeta w_p^o$
0	ζ	0	0	ζw_p^o
0.3ζ	1.3ζ	-0.6ζ	0	$0.7\zeta w_p^o$
0.5ζ	1.5ζ	-1.0ζ	0	$0.5\zeta w_p^o$
0.7ζ	1.7ζ	-1.4ζ	0	$0.3\zeta w_p^o$
ζ	2ζ	-2ζ	0	0
2ζ	3ζ	-4ζ	ζ	0

44

The algorithm for creating an *m*-vector γ of length one such that $\gamma^T(\mathbf{x}^c-\mathbf{x}^{[n]}) \neq 0$ all $c \in \mathbf{I}(n)-\{[n]\}$

- Assume $\mathrm{x}^i \neq \mathrm{x}^j$ when $i \neq j$.

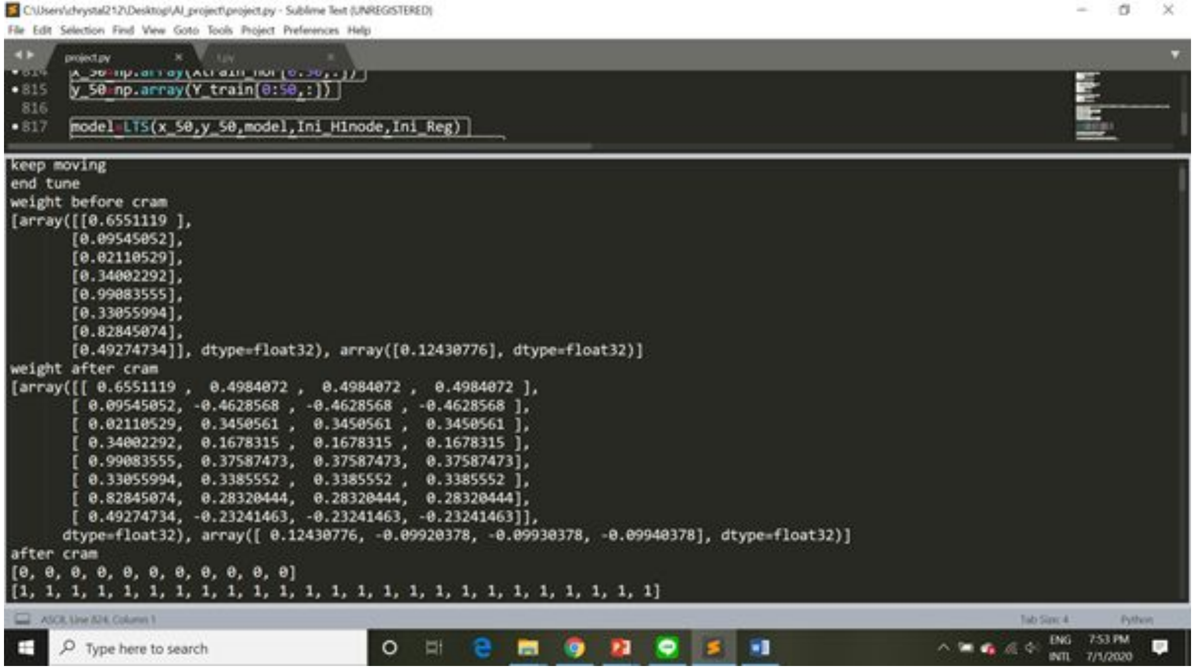
Step 1: Set $\beta_1 = 1$ and let $k = 2$.

Step 2: Let $C_k \equiv \{c: c \in \mathbf{I}(n)-\{[n]\}$ AND $x_j^c = x_j^{[n]}$ all $j = 1, ..., k\}$. Considering β_k as the unknown and β_j, $j = 1, ..., k-1$, as previously determined, set β_k = the smallest integer that is greater than or equal to 1 and $\sum_{j=1}^{k} \beta_j \, (x_j^c - x_j^{[n]}) \neq 0$ all $c \in \mathbf{I}(n)-\{[n]\}-C_k$.

Step 3: $k+1 \rightarrow k$. If $k \leq m$, go to Step 2;

Step 4: Set $\gamma_j = \dfrac{\beta_j}{\sqrt{\sum_{j=1}^{m} \beta_j^{\,2}}}$ $\forall\, j = 1, ..., m$ and STOP.

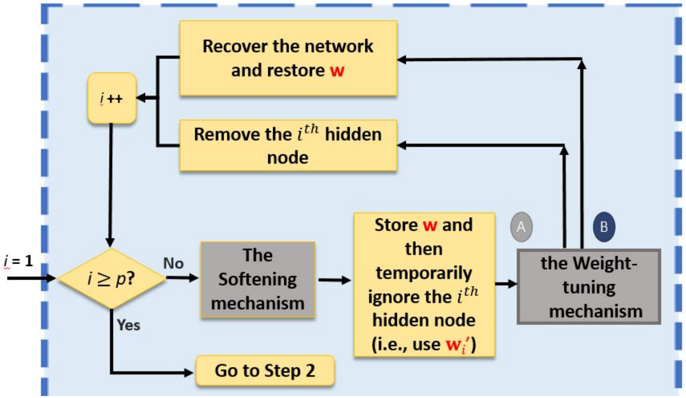
METHODS


C:\Users\chrystal212\Desktop\AI_project\project.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
project.py
815 y_50=np.array(Y_train[0:50,:])
816
817 model=LTS(x_50,y_50,model,Ini_H1node,Ini_Reg)
keep moving
end tune
weight before cram
[array([[0.6551119],
[0.09545052],
[0.02110529],
[0.34002292],
[0.99083555],
[0.33055994],
[0.82845074],
[0.49274734]], dtype=float32), array([0.12430776], dtype=float32)]
weight after cram
[array([[0.6551119 , 0.4984072 , 0.4984072 , 0.4984072],
[0.09545052, -0.4628568 , -0.4628568 , -0.4628568],
[0.02110529, 0.3450561 , 0.3450561 , 0.3450561],
[0.34002292, 0.1678315 , 0.1678315 , 0.1678315],
[0.99083555, 0.37587473, 0.37587473, 0.37587473],
[0.33055994, 0.3385552 , 0.3385552 , 0.3385552],
[0.82845074, 0.28320444, 0.28320444, 0.28320444],
[0.49274734, -0.23241463, -0.23241463, -0.23241463]],
dtype=float32), array([0.12430776, -0.09920378, -0.09930378, -0.09940378], dtype=float32)]
after cram
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
Type here to search
ENG INTL
7:53 PM
7/1/2020

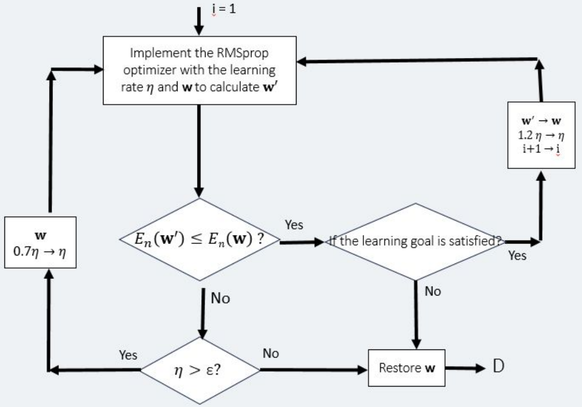
METHODS

- The softening and integrating mechanism

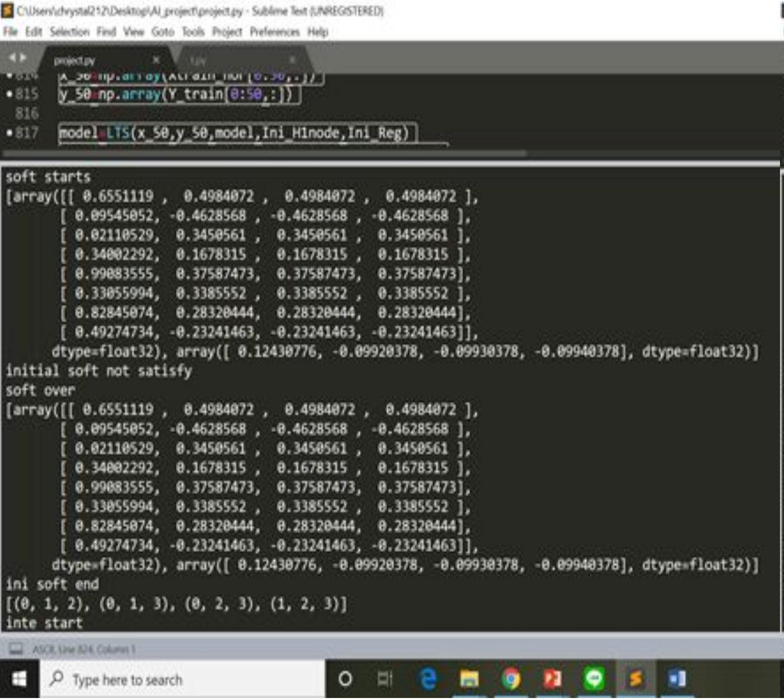
The Softening and Integrating mechanism (I) (sequentially check all hidden nodes)



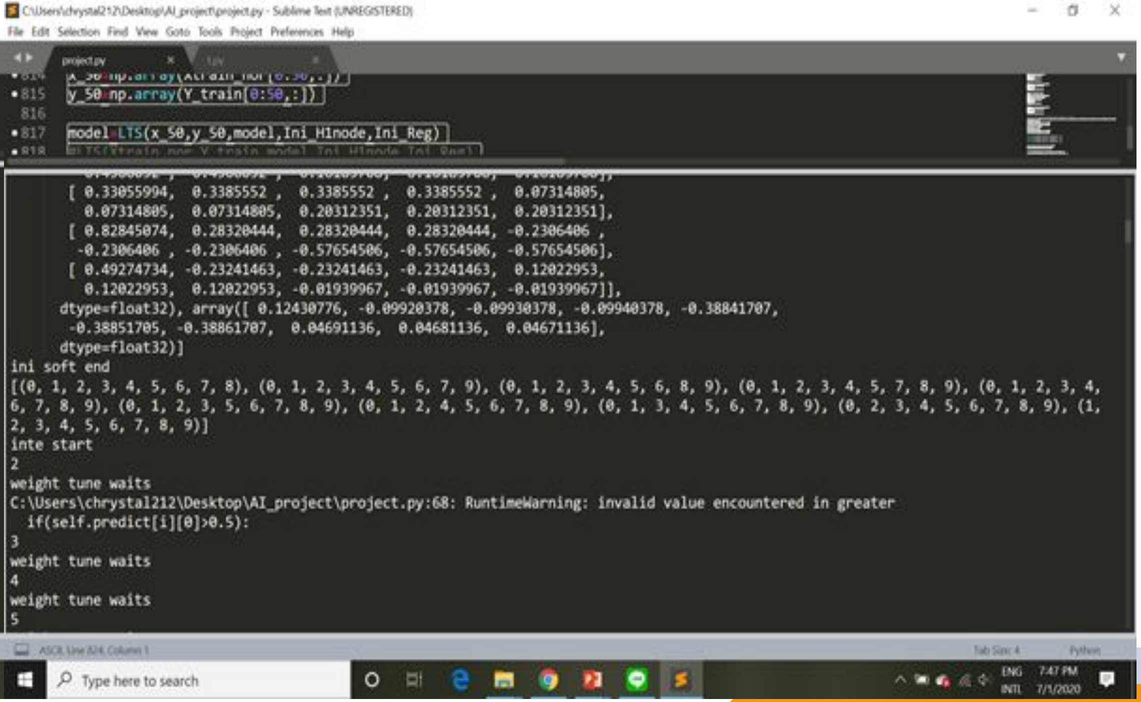
The softening mechanism with the regularization term



METHODS



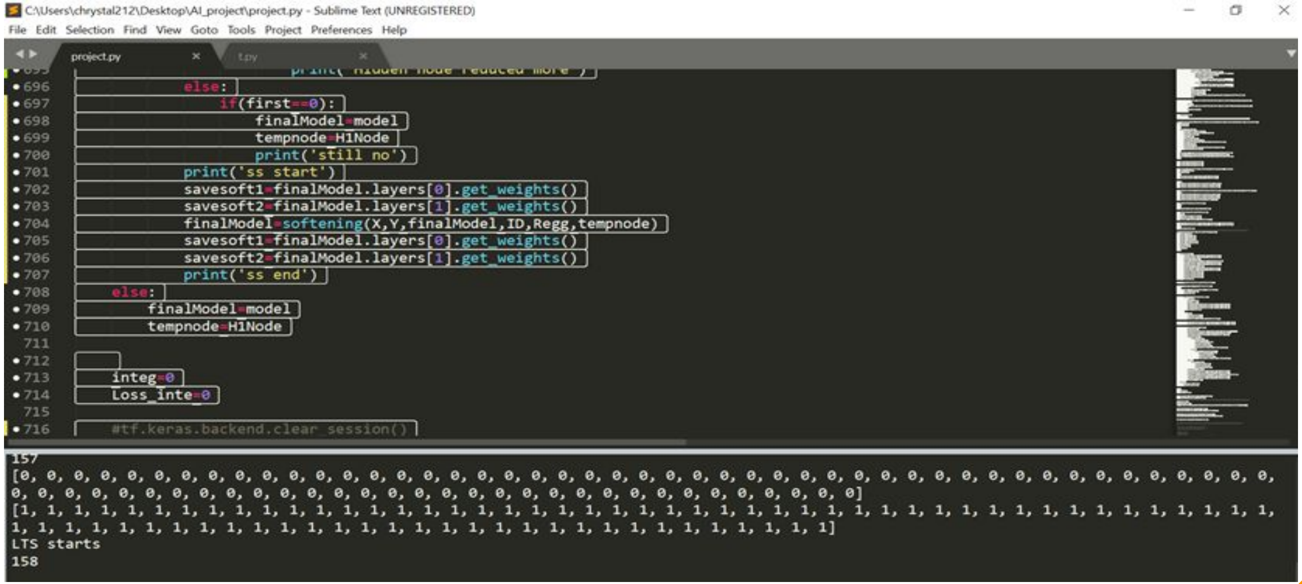
C:\Users\chrystal212\Desktop\AI_project\project.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
project.py
815 y_50=np.array(Y_train[0:50,:])
816
817 model=LTS(x_50,y_50,model,Ini_H1node,Ini_Reg)
soft starts
[array([[0.6551119 , 0.4984072 , 0.4984072 , 0.4984072],
[0.09545052, -0.4628568 , -0.4628568 , -0.4628568],
[0.02110529, 0.3450561 , 0.3450561 , 0.3450561],
[0.34002292, 0.1678315 , 0.1678315 , 0.1678315],
[0.99083555, 0.37587473, 0.37587473, 0.37587473],
[0.33055994, 0.3385552 , 0.3385552 , 0.3385552],
[0.82845074, 0.28320444, 0.28320444, 0.28320444],
[0.49274734, -0.23241463, -0.23241463, -0.23241463]],
dtype=float32), array([0.12430776, -0.09920378, -0.09930378, -0.09940378], dtype=float32)]
initial soft not satisfy
soft over
[array([[0.6551119 , 0.4984072 , 0.4984072 , 0.4984072],
[0.09545052, -0.4628568 , -0.4628568 , -0.4628568],
[0.02110529, 0.3450561 , 0.3450561 , 0.3450561],
[0.34002292, 0.1678315 , 0.1678315 , 0.1678315],
[0.99083555, 0.37587473, 0.37587473, 0.37587473],
[0.33055994, 0.3385552 , 0.3385552 , 0.3385552],
[0.82845074, 0.28320444, 0.28320444, 0.28320444],
[0.49274734, -0.23241463, -0.23241463, -0.23241463]],
dtype=float32), array([0.12430776, -0.09920378, -0.09930378, -0.09940378], dtype=float32)]
ini soft end
[(0, 1, 2), (0, 1, 3), (0, 2, 3), (1, 2, 3)]
inte start
Type here to search



C:\Users\chrystal212\Desktop\AI_project\project.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
project.py
815 y_50=np.array(Y_train[0:50,:])
816
817 model=LTS(x_50,y_50,model,Ini_H1node,Ini_Reg)
[0.33055994, 0.3385552 , 0.3385552 , 0.3385552 , 0.07314805,
0.07314805, 0.07314805, 0.20312351, 0.20312351, 0.20312351],
[0.82845074, 0.28320444, 0.28320444, 0.28320444, -0.2306406 ,
-0.2306406 , -0.2306406 , -0.57654506, -0.57654506, -0.57654506],
[0.49274734, -0.23241463, -0.23241463, -0.23241463, 0.12022953,
0.12022953, 0.12022953, -0.01939967, -0.01939967, -0.01939967]],
dtype=float32), array([0.12430776, -0.09920378, -0.09930378, -0.09940378, -0.38841707,
-0.38851705, -0.38861707, 0.04691136, 0.04681136, 0.04671136],
dtype=float32)]
ini soft end
[(0, 1, 2, 3, 4, 5, 6, 7, 8), (0, 1, 2, 3, 4, 5, 6, 7, 9), (0, 1, 2, 3, 4, 5, 6, 8, 9), (0, 1, 2, 3, 4, 5, 7, 8, 9), (0, 1, 2, 3, 4, 6, 7, 8, 9), (0, 1, 2, 3, 5, 6, 7, 8, 9), (0, 1, 2, 4, 5, 6, 7, 8, 9), (0, 1, 3, 4, 5, 6, 7, 8, 9), (0, 2, 3, 4, 5, 6, 7, 8, 9), (1, 2, 3, 4, 5, 6, 7, 8, 9)]
inte start
2
weight tune waits
C:\Users\chrystal212\Desktop\AI_project\project.py:68: RuntimeWarning: invalid value encountered in greater
if(self.predict[i][0]>0.5):
3
weight tune waits
4
weight tune waits
5
Type here to search
7:47 PM
7/1/2020

EXPERIMENTAL RESULTS

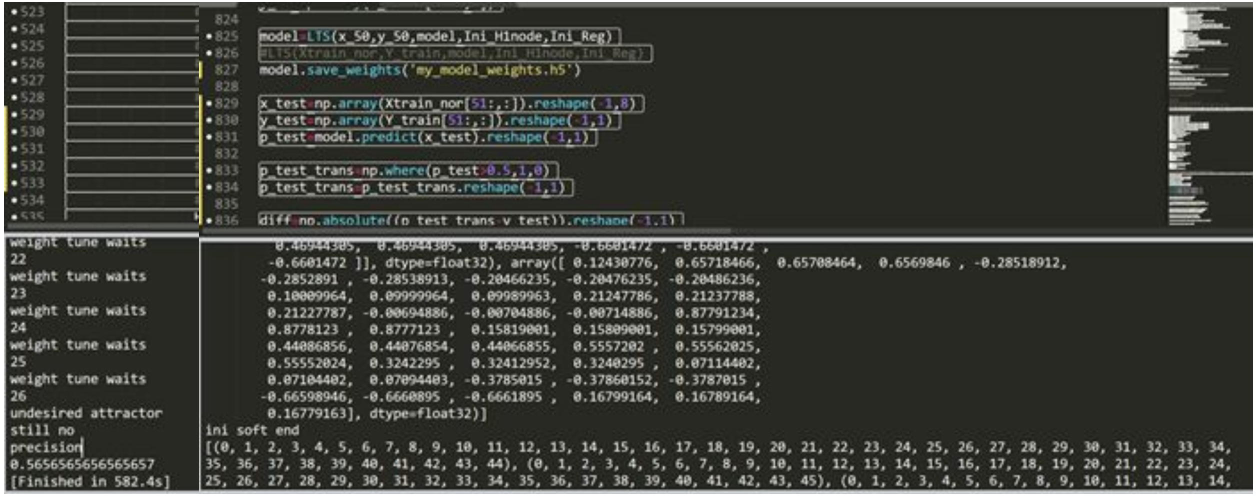
Cramming mechanism and classification works very well in our training dataset.



EXPERIMENTAL RESULTS

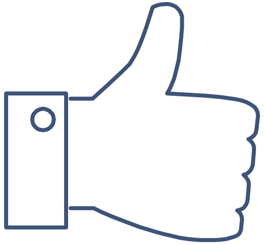
We first train only on 50 dataset, and use the next 100 data as test dataset. Yet, the prediction rate is not good.

1. It may arise from overfitting due to overfitting from the cramming process. Even though we have softening and integrating mechanism to try to penalize high weight and reduce the amount of hidden nodes as many as possible, the mechanism doesn't kill any nodes in most of the time due to the quality of the data.



REFERENCE

- S. Knerr, L. Personnaz and G. Dreyfus, “Single-layer learning revisited: a stepwise procedure for building and training a neural network”, Neurocomputing. Heidelberg : Springer Berlin Heidelberg, 1990.
- Guang-Bin Huang, Yan-Qiu Chen and H. A. Babri, "Classification ability of single hidden layer feedforward neural networks," in *IEEE Transactions on Neural Networks*, vol. 11, no. 3, pp. 799-801, May 2000.
- Knerr, Stefan, Léon Personnaz, and Gérard Dreyfus. "Single-layer learning revisited: a stepwise procedure for building and training a neural network." *Neurocomputing*. Springer, Berlin, Heidelberg, 1990. 41-50.
- Sun, Zhan-Li, et al. "Sales forecasting using extreme learning machine with applications in fashion retailing." *Decision Support Systems* 46.1 (2008): 411-419.
- https://food.ndtv.com/health/100-growth-in-diabetes-patients-in-india-in-the-last-15-years-1292926
- https://www.kaggle.com/uciml/pima-indians-diabetes-database
- 國立政治大學 資訊管理學系 蔡瑞煌 特聘教授



THANKS!