Geodesy Division

A GEODETIC TOMOGRAPHY TO ARTIFICIAL INTELLIGENCE

Assoc. Prof. Dr. Orhan KURT

Kocaeli University, Karamürsel VS, TURKEY

5. Occupation: Artificial Intelligence Applications in Geodesy

27 November 2021 , 12:20 – 12:40 , Ankara / Turkey

OUTLINES of PRESENTATION

1. Hard Computing (**HC**) in science (and engineering).
2. Testing and analyzing.
   a) Hypothesis Test
3. Soft Computing (**SC**) (~Artificial Intelligence, **AI**) in science.
   a) Artificial Neural Network (**ANN**)
   b) Deep Learning (**DL**) and Machine Learning (**ML**)
   c) Genetic Algorithm
4. Some Comparative Examples in **HC** and **AI** (**SC**)

DATA MODELING AND ANALYZING IN SCIENCE - 1

$\hat{\mathbf{x}} = \mathbf{x} + \boldsymbol{\delta}$	An estimated vector of model parameters (u)
$\hat{\mathbf{y}} = \mathbf{y} + \boldsymbol{\epsilon}$	An estimated vector of observables (n)
$\mathbf{x}$	A vector of approximated values for the parameters (u)
$\mathbf{y}$	A vector of observed data (n)
$f = n - u$	Degree of freedom (Number of Conditions)

$f < 0$	Under-determined solution (by under some conditions, for example $\mathbf{x}^T \mathbf{x} \Rightarrow min$)
$f = 0$	Unique solution
$f > 0$	Over-determined solution (by using a criterion function)

HC - DATA MODELING AND ANALYZING IN SCIENCE - 2

Non-linear Functional Models

$\hat{\mathbf{y}} = \mathbf{\Phi}(\hat{\mathbf{x}})$	Modeling in parametric case (n)
$\hat{\mathbf{y}}_{out} = \phi^{m}(\hat{\mathbf{x}}, \hat{\mathbf{y}}_{inp})$	ANN (m: # of layers, ϕ: an activation function)
$\mathbf{\Psi}(\hat{\mathbf{x}}, \hat{\mathbf{y}}) = \mathbf{0}$	Modeling in combined case (m)
$\mathbf{\Gamma}(\hat{\mathbf{x}}) = \mathbf{0}$	Some conditions among the model parameters (c)
$\mathbf{\Theta}(\hat{\mathbf{y}}) = \mathbf{0}$	Modeling in conditional case (f)

Linear Functional Models via Taylor Expansions by the vectors x and y

$\boldsymbol{\epsilon} = \mathbf{A}\,\boldsymbol{\delta} - \boldsymbol{\ell}$	in the parametric case (n)
$\mathbf{A} = \left(\dfrac{\partial \mathbf{\Phi}(\hat{\mathbf{x}})}{\partial \hat{\mathbf{x}}}\right)$	Coefficient matrix of the parameters (n,u) in HC and SC
$\boldsymbol{\ell} = \mathbf{y} - \mathbf{\Phi}(\mathbf{x})$	The translated observable (n)

HC-DATA MODELING AND ANALYZING IN SCIENCE - 3

Stochastic Model (including physical and algebraic correlation)

$$\mathbf{\Sigma}_y = \begin{bmatrix} \sigma_{1,1} & \sigma_{1,2} & \cdots & \sigma_{1,n} \\ & \sigma_{2,2} & \cdots & \sigma_{2,n} \\ & & \cdots & \cdots \\ \text{sim.} & & & \sigma_{n,n} \end{bmatrix}$$ Variances-Covariance Matrix of Data

σ_0^2 , $\mathbf{P}=\sigma_0^2\ \mathbf{\Sigma}_y^{-1}$ A priory variance, Weight Matrix of Data (no in SC)

Least Squares Criterion Function and Its Solution

$\underset{\boldsymbol{\delta}}{\operatorname{argmin}}\{(\mathbf{A}\boldsymbol{\delta}-\boldsymbol{\ell})^T\mathbf{P}(\mathbf{A}\boldsymbol{\delta}-\boldsymbol{\ell})\}$ The Least Square (LS) Criterion for *f*>*0*

$\hat{\boldsymbol{\delta}}=(\mathbf{A}^T\mathbf{P}\,\mathbf{A})^{-1}\mathbf{A}^T\mathbf{P}\boldsymbol{\ell}$ The LS solution (*u*) (Learned part in ANN on **A**)

$\hat{\boldsymbol{\epsilon}}=\mathbf{A}\hat{\boldsymbol{\delta}}-\boldsymbol{\ell}$ Estimated residuals under LS (*n*)

$\hat{\sigma}_0^2=\sqrt{f^{-1}\ \hat{\boldsymbol{\epsilon}}^T\mathbf{P}\hat{\boldsymbol{\epsilon}}}$ *A posterior variance*

$\underset{\boldsymbol{\delta}}{\operatorname{argmin}}\{(\mathbf{A}\boldsymbol{\delta}-\boldsymbol{\ell})^T\mathbf{P}(\mathbf{A}\boldsymbol{\delta}-\boldsymbol{\ell})+\boldsymbol{\lambda}^T(\mathbf{B}\,\boldsymbol{\delta}-\boldsymbol{\omega})\}$ Regularization equations

HC (also in SC) – CRITERION FUNCTIONS (Lp-Norm) – 4 (LOOS or COST FUNCTION)

* **P** is very effective on the model parameter

L_2-Norm (LS) $\boldsymbol{\varepsilon}^T \mathbf{P} \boldsymbol{\varepsilon} \mapsto min$

* **P** is not effective on the model parameter (Combination, n=u)

L_1-Norm $\sum |\varepsilon| \mapsto min$ **(LAR, Least Absolute Res.)**

L_∞-Norm $max\{\boldsymbol{\varepsilon}\} \mapsto min$ **(MAR, Max. Absolute Res.)**

L_3-Norm (Skewness) $\sum \varepsilon^3 \mapsto min$

L_4-Norm (Kurtosis) $\sum \varepsilon^4 \mapsto min$

HC&SC - ANALYZING IN SCIENCE - 5

* In Geodesy, $\mathbf{P}$ includes stochastic behaviors (algebraic and physical correlations came from pre-processing). $\boldsymbol{\lambda}$ contains ***Lagrange multipliers***, and its dimension is equal to number of conditions.

$$\underset{\boldsymbol{\delta}}{\operatorname{argmin}}\{(\mathbf{A}\boldsymbol{\delta}-\boldsymbol{\ell})^T\mathbf{P}(\mathbf{A}\boldsymbol{\delta}-\boldsymbol{\ell})+\boldsymbol{\lambda}^T(\mathbf{B}\,\boldsymbol{\delta}-\boldsymbol{\omega})\}$$ in Geodesy (*by Null hypothesis*)

* In AI, $\mathbf{P}=\mathbf{I}$ (not effective in AI), and λ is a scalar regularization term (also a Lagrange multiplier) for $\boldsymbol{\delta}^T\boldsymbol{\delta} \rightarrow min$ condition.

$$\underset{\boldsymbol{\delta}}{\operatorname{argmin}}\{(\mathbf{A}\boldsymbol{\delta}-\boldsymbol{\ell})^T(\mathbf{A}\boldsymbol{\delta}-\boldsymbol{\ell})+\lambda\,\boldsymbol{\delta}^T\boldsymbol{\delta}\}$$ in AI (SC) (*by Null hypothesis*)

* The regularization term serves to overcame the inconsistency problem from the solution because of an over estimated ANN. This is the ***Levenberg-Marquat*** regularization.

$$(\mathbf{A}^T\mathbf{A}+\lambda\,\mathbf{I})\,\boldsymbol{\delta}=\mathbf{A}^T\boldsymbol{\ell}$$ to any optimization algorithm (***BFGS*** *in this study*)

(***BFGS*** *: Broyden–Fletcher–Goldfarb–Shanno Algorithm*)

* If $\mathbf{L}\neq\mathbf{I}$ is preferred in the regularization part of the criterion (loss or cost) function, then the regularization will be name as ***Tikhonow Regularization***.

* The relationship between the posterior variances with and without regularizing is following:

$$\Omega_\lambda=\Omega+R \qquad \Omega=(\mathbf{A}\boldsymbol{\delta}-\boldsymbol{\ell})^T(\mathbf{A}\boldsymbol{\delta}-\boldsymbol{\ell}) \text{ and } R=\boldsymbol{\delta}^T(\mathbf{A}^T\mathbf{A})\,\boldsymbol{\delta}$$

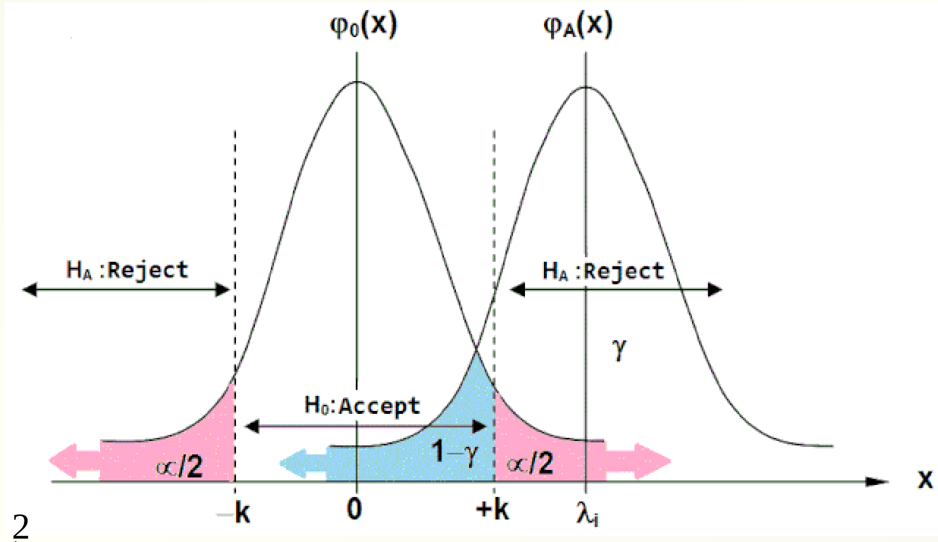
$$E\{\Omega\}=E\{\Omega_\lambda\} \text{ if } E\{R\}=0 \text{ for HC (also SC)}$$

HC (also in SC) – HYPOTHESIS TESTS - 6

$$\boldsymbol{\ell} \sim N(\mathbf{A}\boldsymbol{\delta}\,,\,\sigma_0^2\,\mathbf{P}^{-1})$$

*** Mathematical Model**

Functional Model	Stochastic Model
$E\{\boldsymbol{\ell}\}=\mathbf{A}\boldsymbol{\delta}$	$\mathbf{P}=\mathbf{I}$
$E\{\mathbf{B}\boldsymbol{\delta}+\mathbf{w}\}=\mathbf{0}$	Null Hypothesis (H_0)

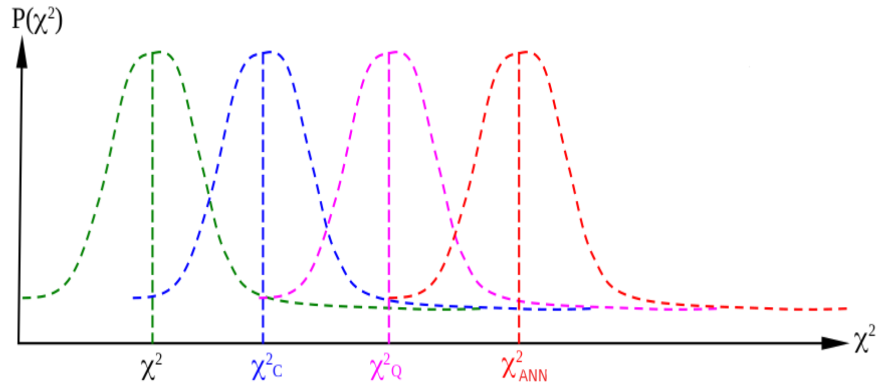


*** Non-central parameter**

$$R=(\mathbf{B}\boldsymbol{\delta}+\mathbf{w})^{\mathrm{T}}(\mathbf{B}\mathbf{Q}_{\delta}\mathbf{B}^{\mathrm{T}})^{-1}(\mathbf{B}\boldsymbol{\delta}+\mathbf{w})\ ,\ \lambda=R/\sigma_0^2$$

*** Power of test (1-γ) > %80 with α=%5, then**

$$\hat{\sigma}_0^{\,2}/\sigma_0^2 = 1+\lambda\ ,\ E\{\lambda\}=0$$



$$\chi^2=\Omega/\sigma_0^2$$

$\chi_C^2=\dfrac{\Omega_C}{\sigma_0^2}$ (Cubic) , $\chi_Q^2=\dfrac{\Omega_Q}{\sigma_0^2}$ (Quadratic) , $\chi_{ANN}^2=\dfrac{\Omega_{ANN}}{\sigma_0^2}$ (ANN)

*** A posterior variance**

$$\hat{\sigma}_0^{\,2} = \sigma_0^2 + R$$

* $E\{\lambda\}=0 \Leftrightarrow$ **HC or SC is okey**

- Mathematical model is fit to data
- No any outlier in data

HC - DATA MODELING AND ANALYZING IN SCIENCE – 7
SOME DEFINITIONS in SC (AI) and HC

Simulation, Optimization and Estimation

* Simulation is a modeling of any scientific problem If we do **not have any data** and restrictions.

* Optimization is a simulation technique that meets one or some of three expectations if we do **not have any data**. The expectations are **precision**, **reliability** and, **cost**.

* Estimation is a solution of **an observed scientific problem (data)** under **a criterion function** (*L_p-Norm estimation*).

Three main pitfalls in Soft Computing (SC) ≡ Artificial Intelligence (AI)

* Searching is made just only non-linear functional model from learning data (not testing data). How to separate data as the learning and testing part in a complicated problem.

* Do not use the stochastic part of the data.

* To overcome the over-estimated or under-estimated solution related to expertise level of SC users.

SC (AI) – ARTIFICIAL NEURAL NETWORK - 1

* One layer Feedforward Neural Network (FNN) is a mapping from $\mathbf{y}_1$ to $\mathbf{y}_2$ observations.

$\mathbf{y}_2 = \varphi(\mathbf{y}_1 , \mathbf{x})$ Feedforward Neural Network (FNN)

$\mathbf{y}_1 , \mathbf{y}_2$ ***Input layer*** $\left(\mathbf{y}_1 = [\varphi_1 , \lambda_1] \right)$ and ***Output Layer*** $(\mathbf{y}_2 = \mathbf{N})$

$\mathbf{x}$ FNN model parameters (named weights in ANN) to learn

* An ANN having n layers ($n \rightarrow \infty \Rightarrow$ Deep Learning (DL)) is represented as functions nested each other.

$$\mathbf{y}_2 = \boldsymbol{\varphi}^{(n)}(\mathbf{y}_1 , \mathbf{x}) = \varphi^{(n)}\left(\varphi^{(n-1)}\left(\ldots\varphi^{(2)}\left(\varphi^{(1)}\left(\mathbf{y}_1 , \mathbf{x}^{(1)}\right) , \mathbf{x}^{(2)}\right)\ldots , \mathbf{x}^{(n-1)}\right) , \mathbf{x}^{(n)}\right)$$

$$\hat{\boldsymbol{\varepsilon}} = \mathbf{y}_2 - \hat{\mathbf{y}}_2 = \mathbf{y}_2 - \boldsymbol{\varphi}^{(n)}(\mathbf{y}_1 , \hat{\mathbf{x}}) \quad \text{Residual vectors}$$

* If a DL process is carried out according to classified data, DL is named as a Machine Learning (ML) process. ML is a DL mechanism for the specific data (for example image processing, word processing, face recognition, etc.). So, the ML is faster than the DL.

SC (AI) – GENETIC ALGORITHM - 2

GA on Line Estimation ($y = \alpha x + \beta$)

Table. Com(12,2) = 66 genes for the line

α (genes)	β (genes)
-000.0100001100111110 1101	0110011.1001010110101011000
-000.0101101010101001 0011	00110100.101100010101010 1101
...	...
...	...
-0000.0100011101111111 011	00111111.1100110101000111101

* Create first genes
* Cross the genes for new candidates
* Search all candidates by using any genetic algorithm under an L_p-norm condition for the best (α, β) from mutations.

EXAMPLE – 1: ANN - Line Estimation – 3

- ✔ **i : for the iteration number (=100)**
- ✔ **j : for the data number (n=12),**
- ✔ **k: for the unknown number (u=2),**
- ✔ **η: the relaxing factor (for convergence rate),** *η= 0.01* **in this paper**

$$\Omega^{(i)} = \sum_j \left\{ \hat{y}_j^{(i)} - y_j^{(i)} \right\}^2 = \sum_j \left\{ \hat{\varepsilon}_j^{(i)} \right\}^2$$ **L_2-norm (LS)** $$\sigma^{(i)} = \sqrt{\Omega^{(i)} / (n-u)}$$

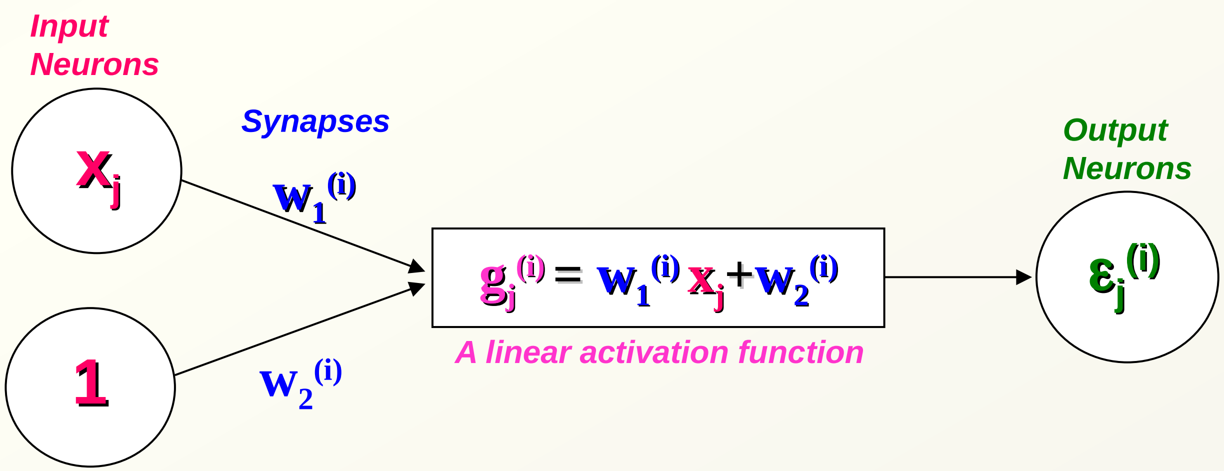
$$\Delta w_{k^{(i)}} = -\eta \;.\; \underset{w_k}{grad} \left\{ \Omega^{(i)} \right\}$$

$$w_k^{(i+1)} = w_k^{(i)} + \Delta w_k^{(i)}$$

$$g_j^{(i)} = w_1^{(i)} x_j + w_2^{(i)}$$

$$\hat{\varepsilon}_j^{(i)} = w_1^{(i)} x_j + w_2^{(i)} - y_j = \hat{\alpha}\; x_j + \hat{\beta} - y_j$$

$$\hat{\varepsilon}_j^{(i)} = g_j^{(i)} - y_j$$



(Kurt, 2019).

EXAMPLE – 1: ANN - Line Estimation – 4

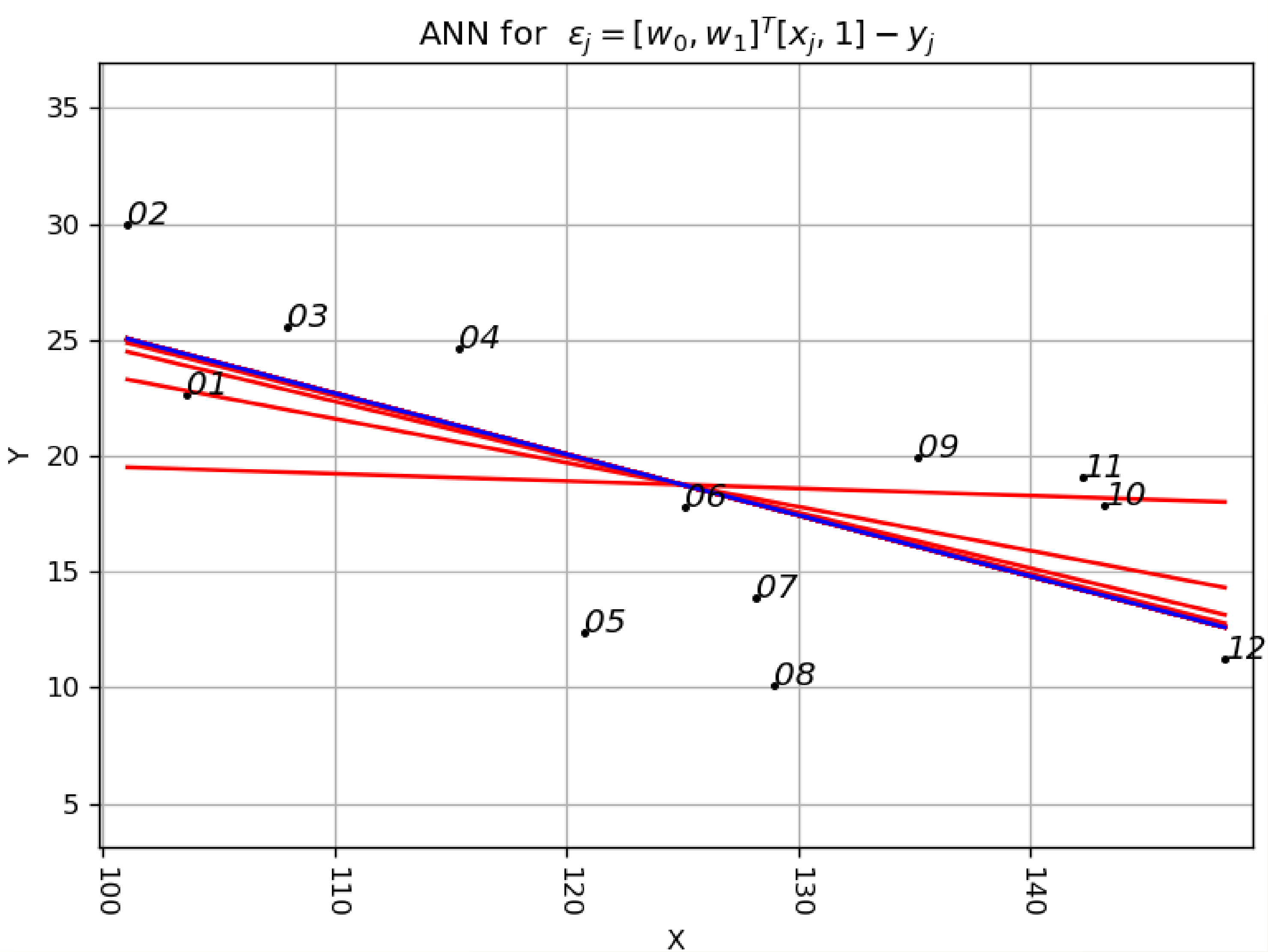


Figure 0. ANN line estimation (blue line) and ANN iterations (red lines) for Simulated 2D data (from Table 1).

(Kurt, 2019).

EXAMPLE – 1: SIMULATED DATA - Line Estimation – 5

Table 1. A simulated line data with coordinate errors and weights.

i	Data (unit)		Errors (unit)		Weights	
	$X=x+\varepsilon_x$	$y=\alpha X+\beta-\varepsilon_y$	ε_x	ε_y	p_x	p_y
01	103.6115	22.6026	3.6115	1.4946	0.08	0.45
02	101.0472	29.9443	-3.4982	-5.2061	0.08	0.04
03	107.9535	25.5708	-1.1374	-2.5591	0.77	0.15
04	115.3465	24.6284	1.7101	-3.4650	0.34	0.08
05	120.7853	12.3584	2.6035	7.4453	0.15	0.02
06	125.0892	17.7569	2.3619	0.9708	0.18	1.06
07	128.1874	13.8566	0.9146	4.0966	1.20	0.06
08	128.9239	10.1226	-2.8942	7.6464	0.12	0.02
09	135.1791	19.9339	-1.1845	-3.7287	0.71	0.07
10	143.1969	17.8771	2.2878	-3.6764	0.19	0.07
11	142.2737	19.1036	-3.1808	-4.6720	0.10	0.05
12	148.4057	11.2449	-1.5943	1.6536	0.39	0.37
Outliers	***-50.0000***	***+20.0000***	*N(0.00;2.42)*	*N(0.00;4.39)*		

(Kurt, 2019).

EXAMPLE – 1: HC vs SC (AI) - Line Estimation – 6

Figure 1a. (Weighted and P=I) line estimation.
REAL:f(x,y) = y -0.25 x - 50 = 0

```
P = I

σ_ols =    4.80684446
         | -0.26267781 |
x_ols =| 51.58473253 |   Hard Comp.

 Weighted LS
 σ_wls =    1.08013012
          | -0.25890396 |
 x_wls =| 50.59713781 |   Hard Comp.

 σ_tls =    1.05618002
          | -0.26895668 |
 x_tls =| 51.99881571 |   Hard Comp.

 σ_com =    1.05652798
          | -0.26473923 |
 x_com =| 51.46738173 |   Hard Comp.
```

```
P = I

σ_ann =    4.80684446    iter = 100
         | -0.26267707 |    η = 0.01
x_ann =| 51.58464036 |   SOFT Comp.

σ_L1 = 3.79770839  29.('03', '12')
         | -0.35414390 |
x_L1  =| 63.80187385 |   Hard Comp.

σ_inf = 6.60052418 50.('06', '12')
         | -0.27928720 |
x_inf =| 52.69271242 |   Hard Comp.

Combination(12,2)=66
```

(Kurt, 2019).

EXAMPLE – 1: HC vs SC (AI) - Line Estimation – 7

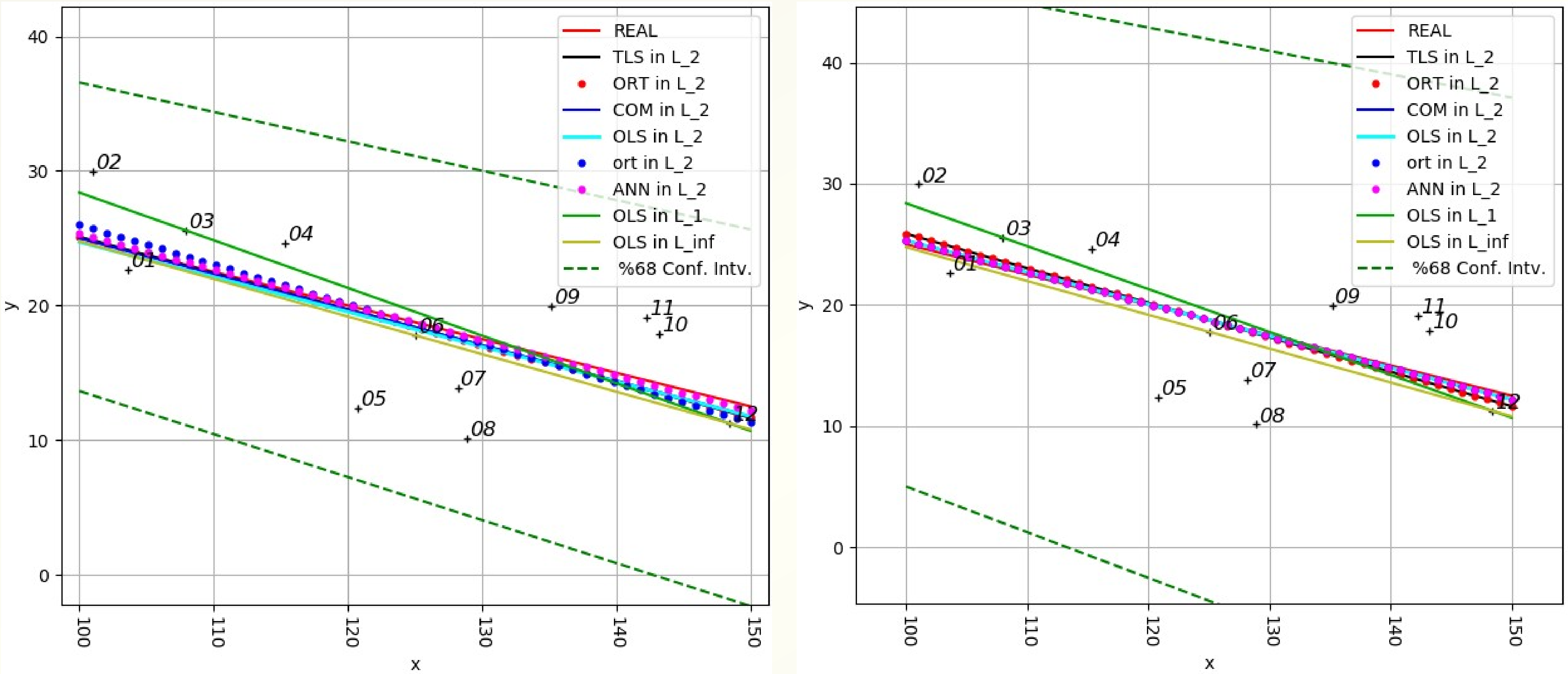
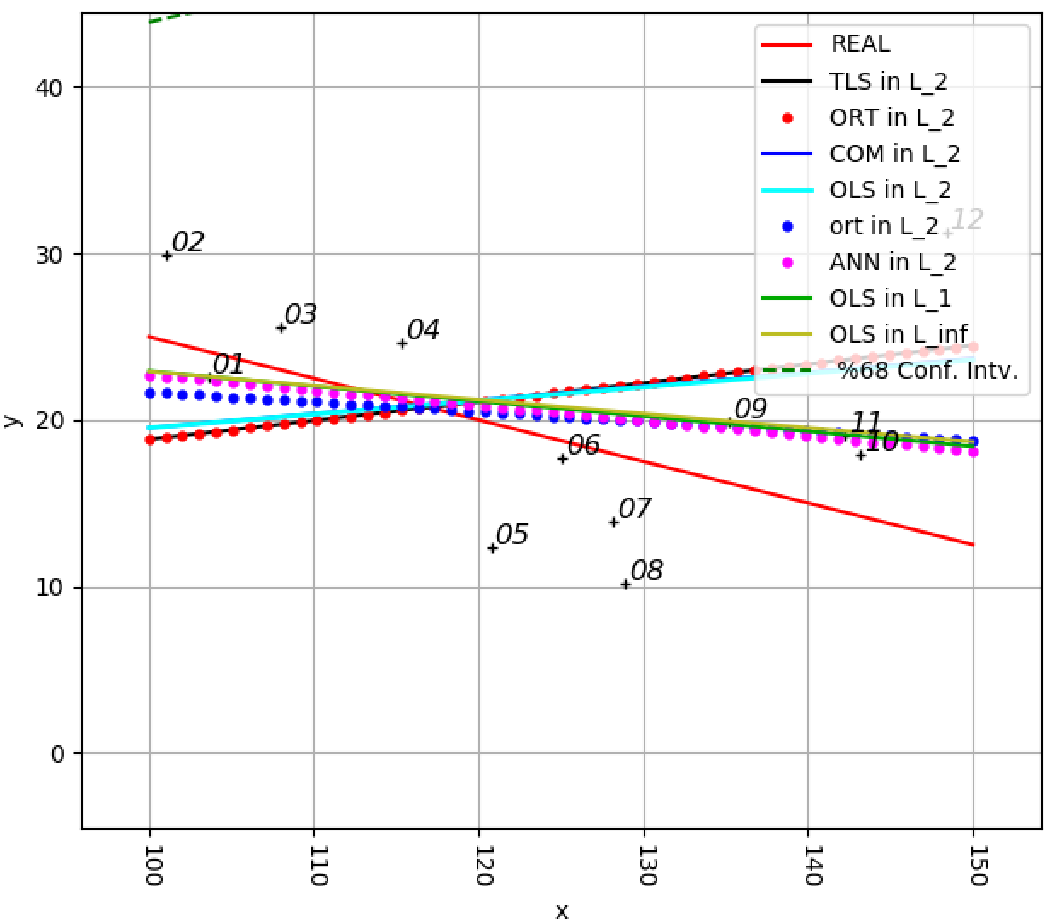


Figure 1. Simulated 2D data (Table 1). (a) Weighted L-p norm line estimations from Table 1 (b) Equal Weighted L-p norm line estimations (P=I)

(Kurt, 2019).

EXAMPLE – 1: HC vs SC (AI) - Line Estimation – 8



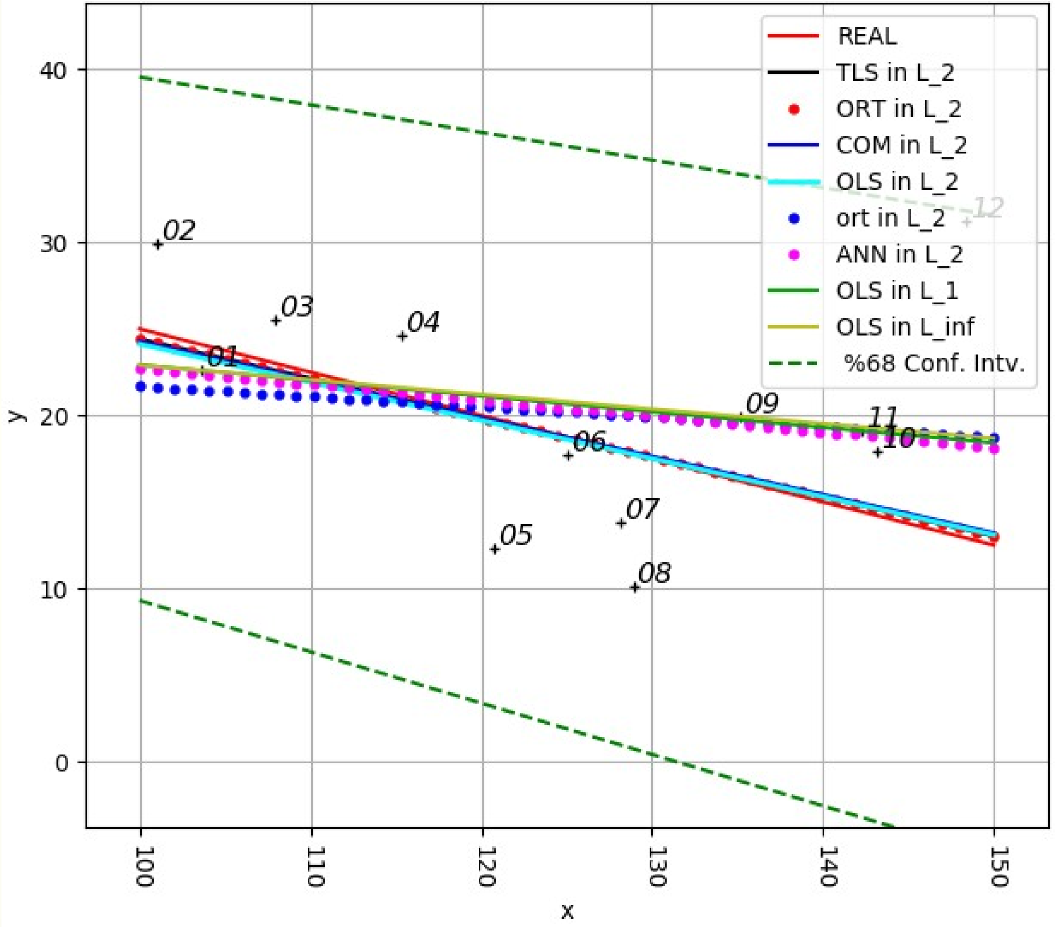
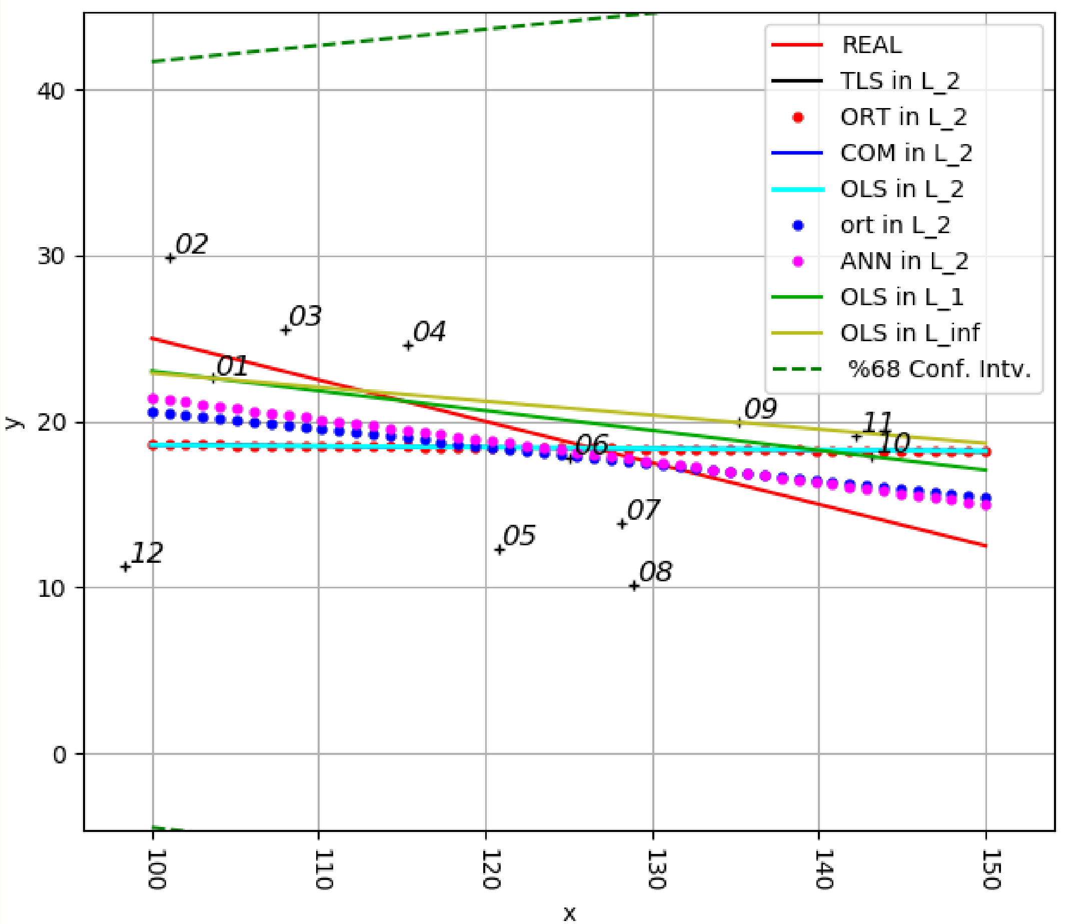


Figure 2. (a) L-p norm estimations with biased y_{12} component of point *12* ($y_{12}=y_{12}+20$) (b) Robust L-p norm estimations (by $p_{y12}=0.001$).

(Kurt, 2019).

EXAMPLE – 1: HC vs SC (AI) - Line Estimation – 9



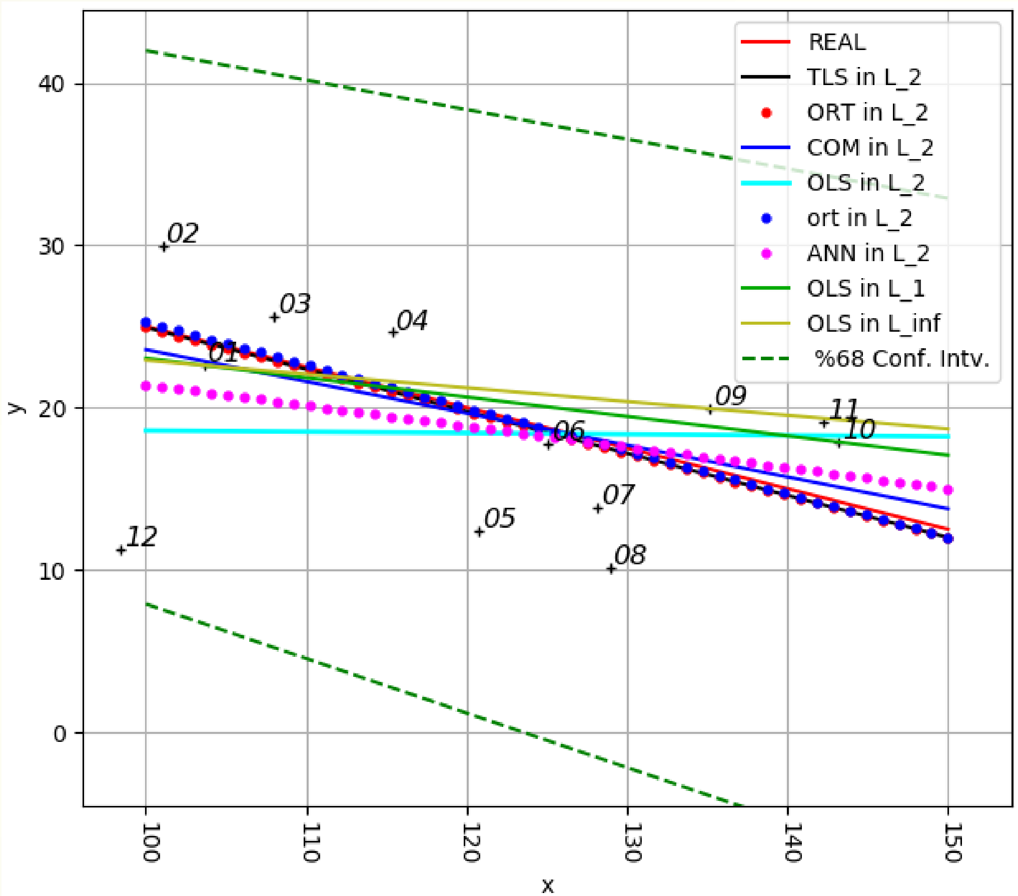


***Figure 3.** (a) L-p norm estimations with biased x_{12} component of point 12 ($x_{12}=x_{12}-50$) (b) Robust L-p norm estimations (by $p_{x12}=0.001$).*

(Kurt, 2019).

EXAMPLE – 2. Surface Estimation in SC - 10

Data – 1 (An under-determinate problem, n < u)

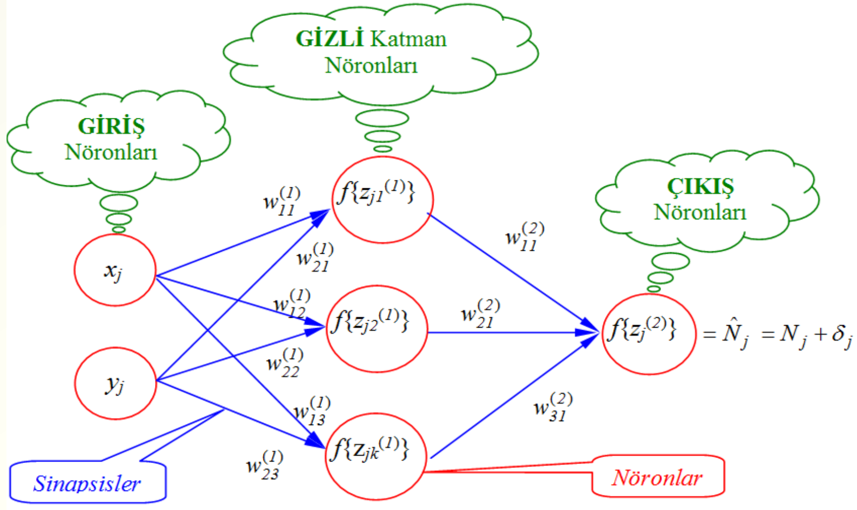
```
      Sleep  Study
     | 4.00  5.50 |
     | 4.50  1.00 |
 X =| 9.00  2.50 |
     | 6.00  2.00 |
      Score
     | 70.00 |
     | 89.00 |
 y =| 85.00 |
     | 75.00 |
```

FNN (Feedforward Neural Network)

BFGS : Broyden–Fletcher–Goldfarb–Shanno Algorithm

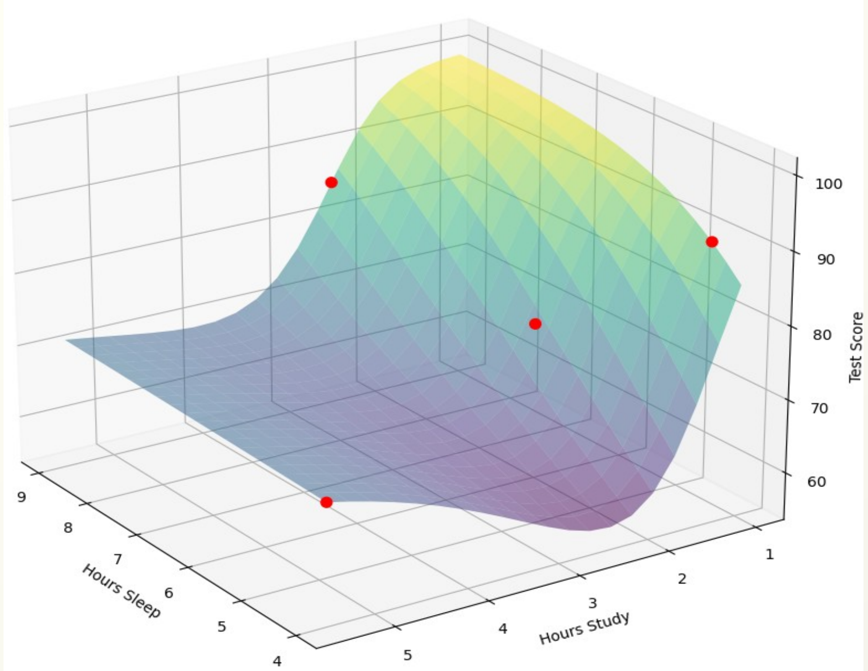
```
BFGS Optimization terminated successfully.
         Current function value: 0.000000
         Iterations: 78
         Function evaluations: 85
         Gradient evaluations: 85

... Layer - 1 :
              |  0.02015775 -0.49117146  3.41151511 | Sleep
 Δ1 = W1 =|  5.46771181 -2.93076455-10.19775336 | Study

... Layer - 2 :
              |  1.09268399 |
 Δ2 = W2 =| -5.88632537 |
              |  6.78904251 |

... Control :
                            |  0.00002267727423 |
                            |  0.00000309987661 |
 ε = y - f(f(X Δ1) Δ2) =| -0.00001482144064 |
                            | -0.00000455565339 |
```

(Welch, 2014-2015).



İleri Beslemeli YSA

FNN by back-propagating



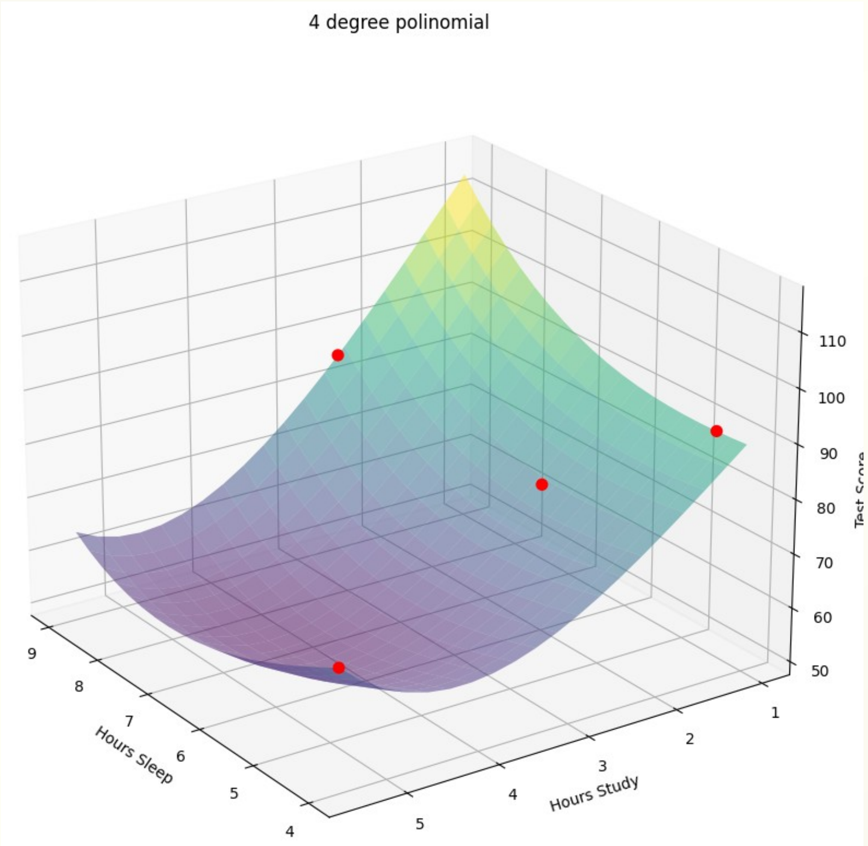
EXAMPLE – 2. Surface Estimation in HC – 11

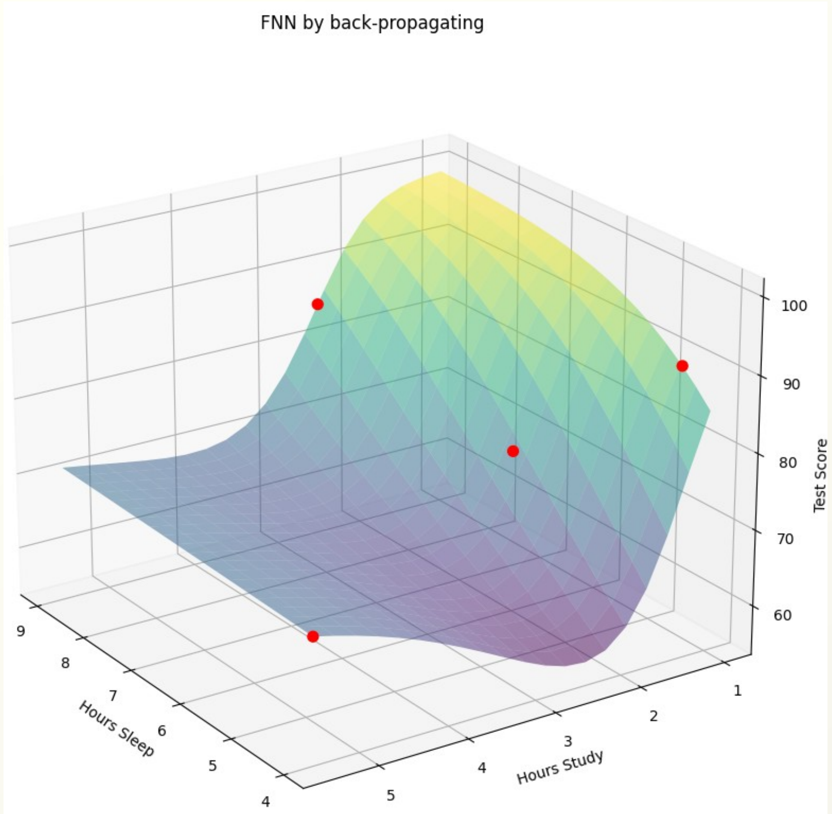
A polynomial function in d degrees $z_k = \sum_{i=0}^{d} \sum_{j=0}^{d} x_k^i \, y_k^j \quad \text{for } i+j \le d \ \wedge \ k=1, 2, \ldots n$

... Mathematics Model: A Δ = z

$$
\begin{array}{cccccccccccccccc}
 & \Delta_{00} & \Delta_{01} & \Delta_{02} & \Delta_{03} & \Delta_{04} & \Delta_{10} & \Delta_{11} & \Delta_{12} & \Delta_{13} & \Delta_{20} & \Delta_{21} & \Delta_{22} & \Delta_{30} & \Delta_{31} & \Delta_{40} \\
A= & \multicolumn{15}{c}{\left|\begin{array}{ccccccccccccccc}
1.00 & 1.00 & 1.00 & 1.00 & 1.00 & 0.44 & 0.44 & 0.44 & 0.44 & 0.20 & 0.20 & 0.20 & 0.09 & 0.09 & 0.04 \\
1.00 & 0.18 & 0.03 & 0.01 & 0.00 & 0.50 & 0.09 & 0.02 & 0.00 & 0.25 & 0.05 & 0.01 & 0.12 & 0.02 & 0.06 \\
1.00 & 0.45 & 0.21 & 0.09 & 0.04 & 1.00 & 0.45 & 0.21 & 0.09 & 1.00 & 0.45 & 0.21 & 1.00 & 0.45 & 1.00 \\
1.00 & 0.36 & 0.13 & 0.05 & 0.02 & 0.67 & 0.24 & 0.09 & 0.03 & 0.44 & 0.16 & 0.06 & 0.30 & 0.11 & 0.20
\end{array}\right|}
\end{array}
$$

$$
z = \left| \begin{array}{c} 0.70000000 \\ 0.89000000 \\ 0.85000000 \\ 0.75000000 \end{array} \right|
$$





... Solution:

$$
\Delta = \left| \begin{array}{r} 1.0478 \\ -0.4895 \\ -0.1069 \\ 0.1839 \\ 0.3083 \\ -0.0379 \\ -0.3965 \\ -0.0919 \\ 0.0784 \\ -0.1482 \\ -0.1855 \\ -0.0173 \\ 0.0825 \\ 0.0273 \\ 0.3872 \end{array} \right|
$$

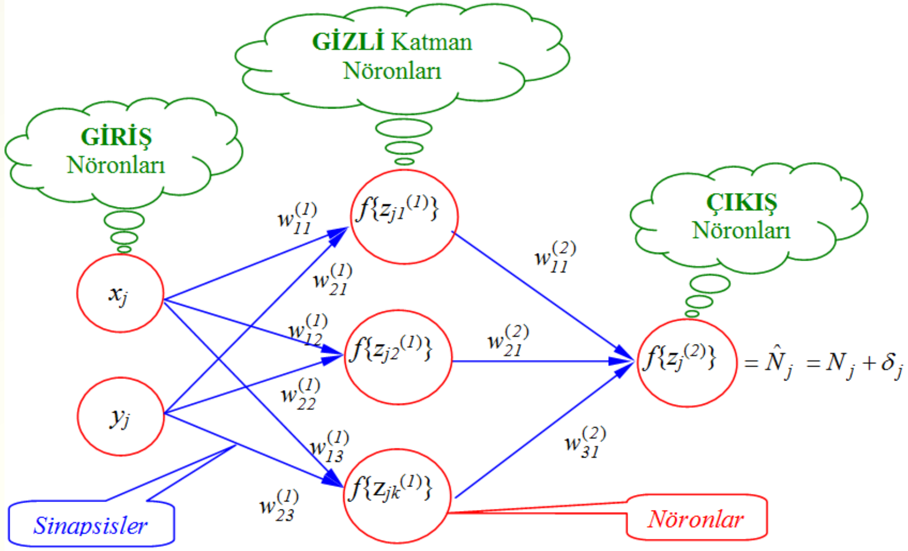
... Control :

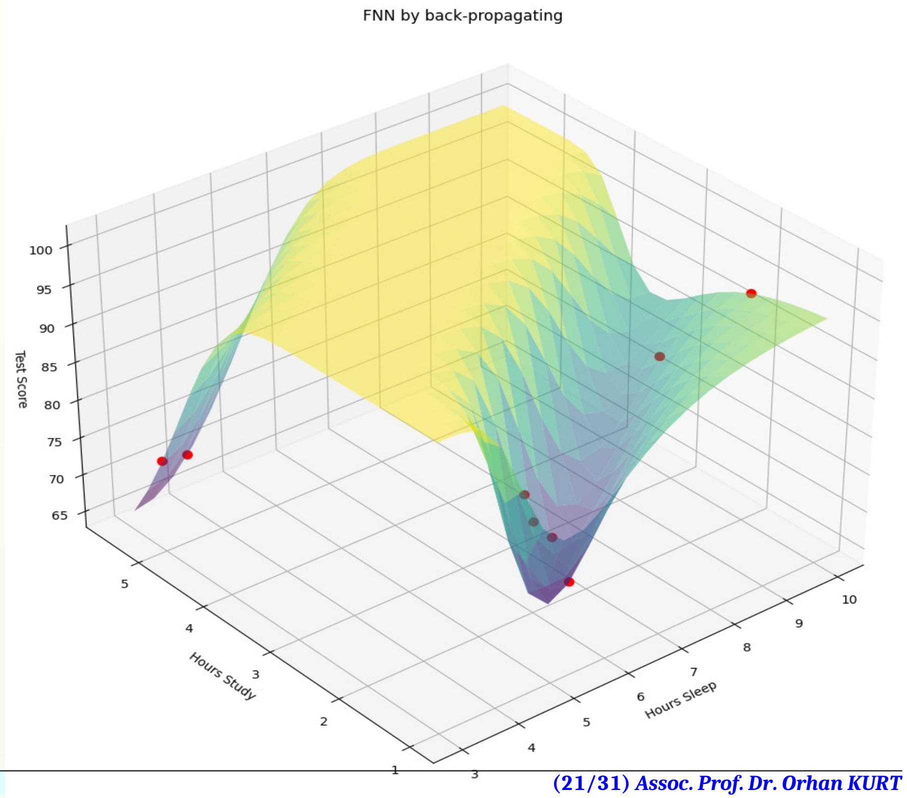
$$
\varepsilon = z - A\,\Delta = \left| \begin{array}{r} 0.00000000000000 \\ 0.00000000000000 \\ 0.00000000000000 \\ -0.00000000000000 \end{array} \right|
$$

EXAMPLE – 2. Surface Estimation in SC – 12

Data – 2 (FNN without regularizing) *(Welch, 2014-2015).*

```
      Sleep  Study            Score
     | 3.00  5.00 |         | 75.00 |
     | 5.00  1.00 |         | 82.00 |
     |10.00  2.00 |         | 93.00 |
     | 6.00  1.50 |         | 70.00 |
 X = | 4.00  5.50 |     y = | 70.00 |
     | 4.50  1.00 |         | 89.00 |
     | 9.00  2.50 |         | 85.00 |
     | 6.00  2.00 |         | 75.00 |
 ... BFGS Optimization
Optimization terminated successfully.
         Current function value: 0.000000
         Iterations: 246
         Function evaluations: 270
         Gradient evaluations: 270

... Layer - 1 :
            |  9.73214731   6.92750911-11.19661513 |
 Δ1 = W1 =|-11.00488932 -7.02906322 -4.79567257 |
... Layer - 2 :
            |-55.16463540 |
 Δ2 = W2 =| 58.29805996 |
            |840.48159098 |

... Control :
                          |  0.00000234694056 |
                          |  0.00000311145797 |
                          |  0.00000199420600 |
                          | -0.00000337434455 |
 ε = y - f(f(X Δ1) Δ2) =|  0.00000188477530 |
                          | -0.00000443547947 |
                          | -0.00000139174774 |
                          | -0.00000116449878 |
```

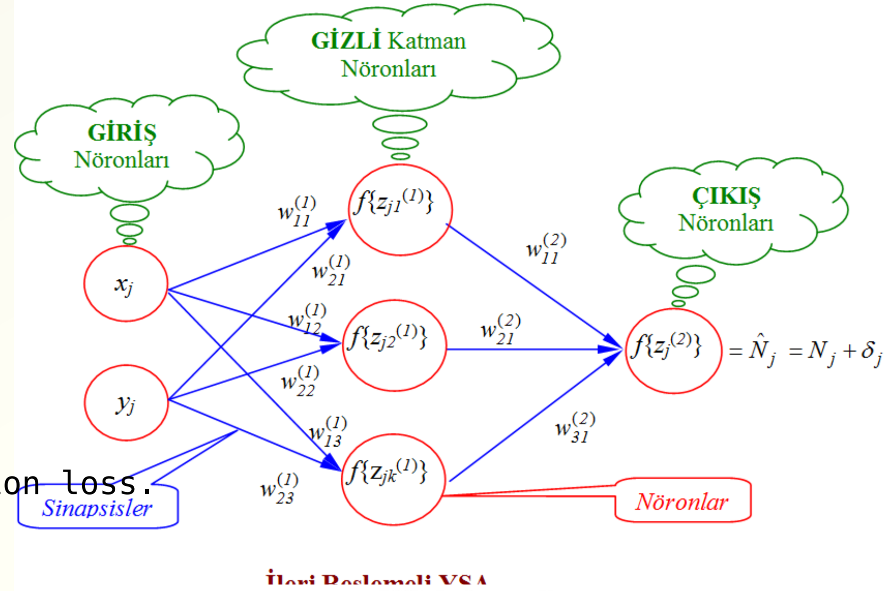


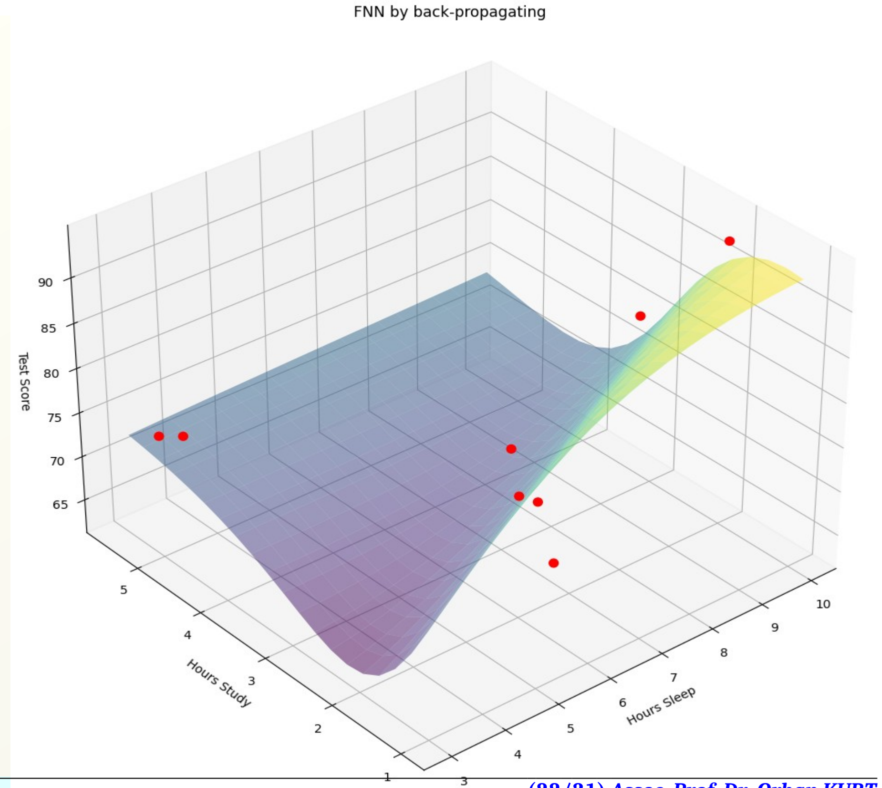


EXAMPLE – 2. Surface Estimation in SC – 13

Data – 2 (FNN with regularizing) ***(Welch, 2014-2015).***

```
       Sleep  Study               Score
     |  3.00  5.00 |            | 75.00 |
     |  5.00  1.00 |            | 82.00 |
     | 10.00  2.00 |            | 93.00 |
     |  6.00  1.50 |            | 70.00 |
 X = |  4.00  5.50 |        y = | 70.00 |
     |  4.50  1.00 |            | 89.00 |
     |  9.00  2.50 |            | 85.00 |
     |  6.00  2.00 |            | 75.00 |
... BFGS Optimization
Warning: Desired error not necessarily achieved due to precision loss.
         Current function value: 0.001446
         Iterations: 223
         Function evaluations: 483
         Gradient evaluations: 471
... Regularization Term : 0.0000001
... Layer - 1 :
             | -1.79889239 -3.02697246  0.37579130 |
 Δ1 = W1 =|  5.68464984  7.73096560  0.12919199 |
... Layer - 2 :
             |  9.23397032 |
 Δ2 = W2 =| -9.37323446 |
             |  2.07742501 |

... Control :
                          |  0.02895182579661 |
                          | -0.01755123391647 |
                          |  0.02074779703263 |
                          | -0.11628544253054 |
 ε = y - f(f(X Δ1) Δ2) =| -0.02977545021417 |
                          |  0.07512387812293 |
                          |  0.03515808505381 |
                          |  0.00646494875537 |
```





EXAMPLE – 2. Surface Estimation in HC – 14

****** Exaple 10 - 4. Degree Polynomial *******

... Mathematics Model: A Δ = z

$$z = \Delta_{00} + \Delta_{01}y + \Delta_{02}y^2 + \Delta_{03}y^3 + \Delta_{04}y^4 + \Delta_{10}x^1 + \Delta_{11}x^1y^1 + \Delta_{12}x^1y^2 + \Delta_{13}x^1y^3 + \Delta_{20}x^2 + \Delta_{21}x^2y^1 + \Delta_{22}x^2y^2 + \Delta_{30}x^3 + \Delta_{31}x^3y^1 + \Delta_{40}x^4$$

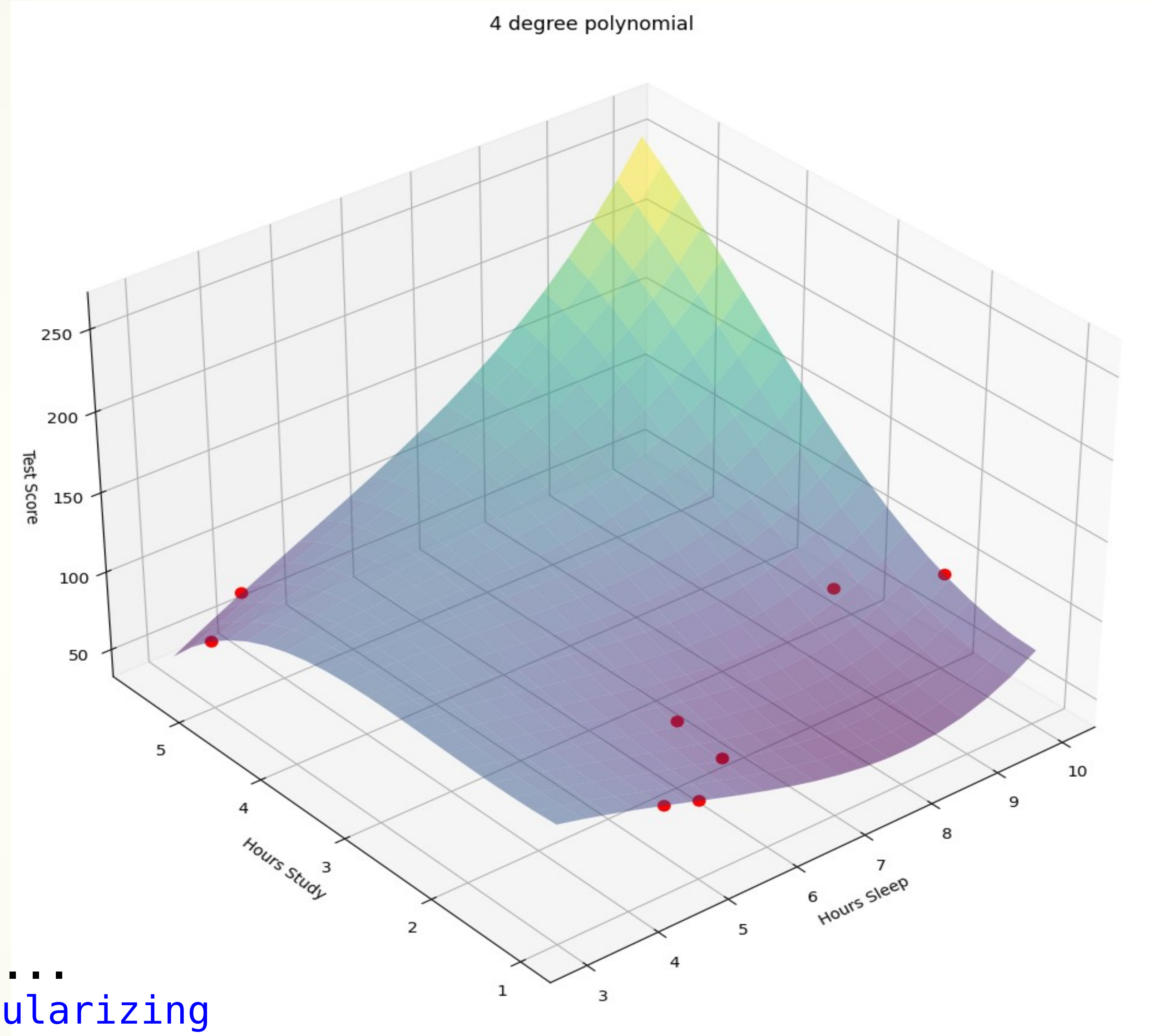
n=8 u= 15

... Solution:

$$\Delta = \begin{vmatrix} 1.3355 \\ -0.9245 \\ 1.1053 \\ -0.3627 \\ -1.5537 \\ 0.1257 \\ 0.3301 \\ 1.8791 \\ 1.1345 \\ -2.0904 \\ -0.3926 \\ 0.8363 \\ -0.9843 \\ -0.4054 \\ 2.5348 \end{vmatrix}$$

... Control :

$$\varepsilon = z - A\,\Delta = \begin{vmatrix} -0.00000000000044 \\ 0.00000000000002 \\ 0.00000000000043 \\ 0.00000000000012 \\ -0.00000000000014 \\ -0.00000000000037 \\ -0.00000000000040 \\ 0.00000000000035 \end{vmatrix}$$



... A posteriory σ = sqrt(ε'ε) ...

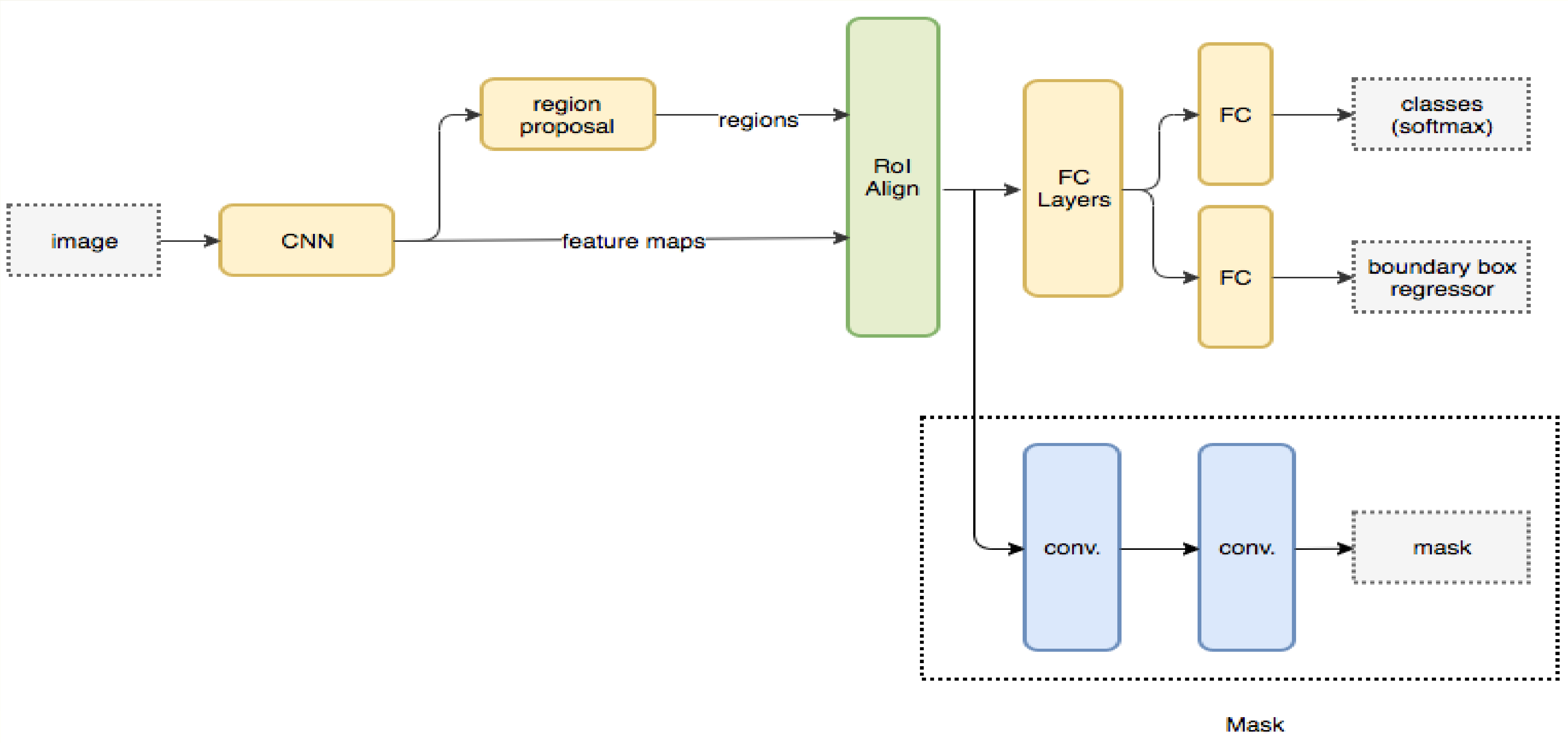
σ1 = 0.000000 without regularizing
σ2 = 0.022907 with regularizing
σP = 0.000000 4. polynomial

EXAMPLE – 3. Feature Extraction in SC – 15

<---- Classification

(Sinanoğlu, 2020) Object Masking -->

EXAMPLE – 3. Feature Extraction in SC – 16



Faster R-CNN *(Sinanoğlu, 2020)*

CONCLUSIONS

- Artificial Intelligence (AI≡SC) are based on Artificial Neural Networks (ANN). ANN methods, which should be preferred in the solution of problems in which we don't establish the functional relationships between data and model parameters (named as weights in ANN), give different results according to the expertise degree of the user. The interpretation of the calculated results may be able to vary depending on the expert, as well.
- The model parameters (the ANN weights) are not scholastically weights due to being able to have negative values.
- ANN solution resolution is weaker than the solution of the WLS (Weighted Least Squares) since the input data in ANN are admitted to have uncorrelated and equal scholastic weights.
- We should prefer the LS (≡HC) solution if we can set up the functional relationship between the data and model parameters. Otherwise, we may prefer ANN (AI≡SC).
- The most powerful side of LS is the ease of interpretation of the results. But, because the LS is sensitive to data errors, it should be strengthened by the M-Estimation (Robust Estimation) for the data having some outliers.
- If the calculation is soft, the results are soft. When using in engineering problems, careful analysis should be made for the results and they should be verified with other sources.

References

Akyılmaz, O., 2005. ***Applications of Soft Computing Methods in Geodesy***, İTÜ-FBE, Ph.D. Thesis, https://polen.itu.edu.tr/bitstream/11527/12127/1/2851.pdf, (accessed at 4 Aug 2019).

Arslan, O.; Kurt, O.; Konak, H. 2007. ***Suggestions on Applications of Artificial Neural Networks in Geodesy***, *11. Türkiye Harita Bilimsel ve Teknik Kurultayı*, 2–6 April 2007, Ankara, Turkey, https://www.researchgate.net/publication/229039355, (Accessed at 1 Nov 2021).

Hekimoğlu, Ş., 1996. ***Robust Estimation Methods***, Lecture notes (in Turkish), Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul (not pressed).

Flohic, J.L., 1999, ***Genetic Algorithms Tutorials (in C)***, http://www-cs-students.stanford.edu/~jl/Essays/ga.html, (accessed at 04 Aug 2019).

Guillaume, S., 2001, ***Designing Fuzzy Inference Systems from Data: An Interpretability - Oriented Review***, IEEE Transsactions on Fuzzu Systems, Vol. 9, No. 3, June 2001, https://pdfs.semanticscholar.org/fdf1/0b28242cf1e814052911e9c77b5a3d34248a.pdf, (accessed at 4 Aug 2019).

Jain, A.K., Mao, J., Mohiuddin, K., 1996. ***Artifical Neural Networks,*** IEEE Computer Special Issue on Neural Computing, March 1996. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=485891, (Accessed at 4 Aug 2019).

Karray F.O. and Silva, C., 2004. ***Soft Computing and Intelligent Systems Design***, Pearson Education Limited 2004, SBN 0 321 11617 8.

Kiusalaas, J. 2013. ***Numerical Methods in Engineering with Python 3***, Cambridge, www.cambridge.org/9780521852876, (accessed at 04 Aug 2019).

Koch, K.R., 1999. ***Parameter Estimation and Hypothesis Testing in Linear Models***, Springer-Verlag Berlin Heidelberg New York, ISBN-540-65257-4. https://link.springer.com/book/10.1007/978-3-662-03976-2, (accessed at 4 Aug 2019).

Konak, H., Dilaver, A., Kurt, O., 1999. ***Fuzzy Logic Approach at Outlier Detection Process***, TMMOB-HKMO, 7. Türkiye Harita Bilimsel ve Teknik Kurultayı, s.213-227, Ankara, 1-5 Mart 1999, https://www.researchgate.net/publication/281107919, (accessed at 4 Aug 2019).

Krenker, A., Bešter, J. and Kos, A., 2011. ***Introduction to the Artificial Neural Networks, ArtificialNeural Networks - Methodological Advances and Biomedical Applications***, Prof. Kenji Suzuki (Ed.), ISBN: 978-953-307-243-2, InTech, Available from: http://www.intechopen.com/books/artificial-neural-networks-methodological-advances-and-biomedical-applications/introduction-to-the-artificial-neural-networks, (accessed at 4 Aug 2019).

Kurt, O., Arslan, O., Konak, H. 2007. ***Polynomial Height Transformation, 11. Türkiye Harita Bilimsel ve Teknik Kurultayı, 2–6 April 2007, Ankara, Turkey***, https://www.researchgate.net/publication/280944644, (accessed at 1 Nov 2019).

Kurt, O., Arslan, O., Konak, H. 2007. ***Double Stage Semi Dynamic Polynomial Surface Fitting For Modeling The Local Geoid***, International Earthquake Symposium Kocaeli 2007, 22-26 October 2007, 265-274, https://www.researchgate.net/publication/280979070, (accessed at 1 Nov 2019).

Kurt, O., 2010. ***Numerical Analysis***, Lecture notes (in Turkish), Kocaeli University, Kocaeli. https://orhankurt.jimdo.com/undergraduate/spring-bahar/muh208-numerical-analysis/, (accessed at 04 Aug 2019).

Kurt, O., 2011. ***Analysis of Geodetic Data***, Lecture Notes (in Turkish), Kocaeli University, https://orhankurt.jimdo.com/graduate/autumn/jjm513-estimation-of-geodetic-quantities-and-hypothesis-testing-in-linear-models/, (accessed at 4 Aug 2019).

Kurt, O., 2013. ***Advanced Adjustment***, Lecture Notes (in Turkish), Kocaeli University, https://orhankurt.jimdo.com/undergraduate/autumn-g%C3%Bcz/hrt409-advanced-topics-in-adjustment/, (accessed at 4 Aug 2019).

Kurt, O., 2014. ***Spectral Analysis of Keplerian Orbit Elements of GNSS Satellites***, 2014 22nd Signal Processing and Communications Applications Conference (SIU), DOI: https://ieeexplore.ieee.org/document/6830671, (accessed at 1 Nov 2021).

Kurt, O., 2016. ***Height Transformation as a New Tool to Check GNSS and Leveling Networks***, Türkiye Ulusal Jeodezi Komisyonu (TUJK) Çalıştayı – 2016, Yıldız Teknik Üniversitesi, İstanbul, https://www.researchgate.net/publication/309727153.

Kurt, O., 2016. ***Selection of Best Fit Height Transformation Method with Uncertainty Data***, HKMO-Mühendislik Ölçmeleri STB Komisyonu, 8. Ulusal Mühendislik Ölçmeleri Sempozyumu, Yıldız Teknik Üniversitesi, İstanbul, Turkey, https://www.researchgate.net/publication/309375350.

Kurt, O., 2018, **On Non-Linearity in Non-Linear Least Squares**, Chapter 4 in Optimization Algorithms - Examples, Edited by Jan Valdman, IntechOpen, https://www.intechopen.com/books/optimization-algorithms-examples (accessed at 1 Jun 2019) .

Kurt, O., 2019, ***Data Modeling and Analyzing***, Lecture notes, https://orhankurt.jimdofree.com/graduate/autumn/jjm615-data-modeling-and-analyzing/, (accessed at 1 Nov 2021).

Kurt, O., 2019, ***On L_p Norm Estimator from Regression Analysis to Soft Computing***, Conference: IESKO 2019 VI. International Earthquake Symposium Kocaeli, Kocaeli/Turkey, https://www.researchgate.net/publication/337199449 (accessed at 1 Jun 2019).

Mangano, S., 1995, ***Genetic Algorithm***, http://www.javamath.com/snucode/translateit.pdf, (accessed at 4 Aug 2019).

Michael Nielsen (2015), ***Neural Networks and Deep Learning,*** Determination Press, http://static.latexstudio.net/article/2018/0912/neuralnetworksanddeeplearning.pdf, (accessed at 4 Aug 2019).

Press, W.H., Flannery, B.P., Teukoslsky, S.A., 2002. ***Numerical Recipes in C***, Second Edition, Cambridge University Press, Cambridge UK., http://numerical.recipes/, (accessed at 04 Aug 2019).

Sinanoğlu, N., 2020. ***Object Extraction from Satellite Images,*** Undergraduate Thesis, Geomatics Engineering Department at Kocaeli University.

Stoltz, E., 2018. ***Evolution of a salesman: A complete genetic algorithm tutorial for Python***, https://towardsdatascience.com/evolution-of-a-salesman-a-complete-genetic-algorithm-tutorial-for-python-6fe5d2b3ca35, (accessed at 04 Aug 2019).

Wang, C., 2015. ***A Study of Membership Functions on Mamdani-Type Fuzzy Inference System for Industrial Decision-Making***, Theses, and Dissertations. Paper 1665. https://preserve.lehigh.edu/cgi/viewcontent.cgi?article=2665&context=etd, (Accessed: 04 Aug 2019)

Whitley, D., 1994, ***A Genetic Algorithm Tutorial***, Statistics and Computing, (1994) 4, 65-85, https://link.springer.com/content/pdf/10.1007%2FBF00175354.pdf, (accessed at 4 Aug 2019).

Some URL References for GNU Software used in the Presentation

Code::Blocks, A free C/C++ and Fortran IDE, https://www.codeblocks.org, (accessed at 01 Nov 2021).

Drawing, A drawing application for the GNOME desktop, https://maoschanz.github.io/drawing/, (accessed at 01 Nov 2021).

Geany - A fast and lightweight IDE, https://www.geany.org/, (accessed at 01 Nov 2020).

LibreOffice, Free and Open Source Office Tools, https://www.libreoffice.org/, (accessed at 01 Nov 2021).

Linux Mint, It is one of alternatives to Microsoft Windows OS and Apple MacOS, https://linuxmint.com/, (accessed at 01 Nov 2021).

Pix, An image viewer and browser, https://github.com/linuxmint/pix, (accessed at 01 Nov 2021).

Python, https://www.python.org/, (accessed at 01 Nov 2021).

Feasible Tubes for ANN Python Codes and Examples

Stephen Welch (2014), http://www.welchlabs.com/blog, ***Neural Networks Demystified [Part 1: Data and Architecture]***, https://www.youtube.com/watch?v=bxe2T-V8XRs, (accessed at 01 Nov 2021).

Stephen Welch (2014), http://www.welchlabs.com/blog, ***Neural Networks Demystified [Part 2: Forward Propagation]***, https://www.youtube.com/watch?v=UJwK6jAStmg, (accessed at 01 Nov 2021).

Stephen Welch (2014), http://www.welchlabs.com/blog, ***Neural Networks Demystified [Part 3: Gradient Descent]***, https://www.youtube.com/watch?v=5u0jaA3qAGk, (accessed at 01 Nov 2021).

Stephen Welch (2014), http://www.welchlabs.com/blog, ***Neural Networks Demystified [Part 4: Backpropagation]***, https://www.youtube.com/watch?v=GlcnxUlrtek, (accessed at 01 Nov 2021).

Stephen Welch (2014), http://www.welchlabs.com/blog, ***Neural Networks Demystified [Part 5: Numerical Gradient Checking]***, https://www.youtube.com/watch?v=pHMzNW8Agq4, (accessed at 01 Nov 2021).

Stephen Welch (2014), http://www.welchlabs.com/blog, ***Neural Networks Demystified [Part 6: Training],*** https://www.youtube.com/watch?v=9KM9Td6RVgQ, (accessed at 01 Nov 2021).

Stephen Welch (2015), http://www.welchlabs.com/blog, ***Neural Networks Demystified [Part 7: Overfitting, Testing, and Regularization]***, https://www.youtube.com/watch?v=S4ZUwgesjS8, (accessed at 01 Nov 2021).