PROCESSING MEASUREMENTS USING DEEP LEARNING

GIGIA APTSIAURI

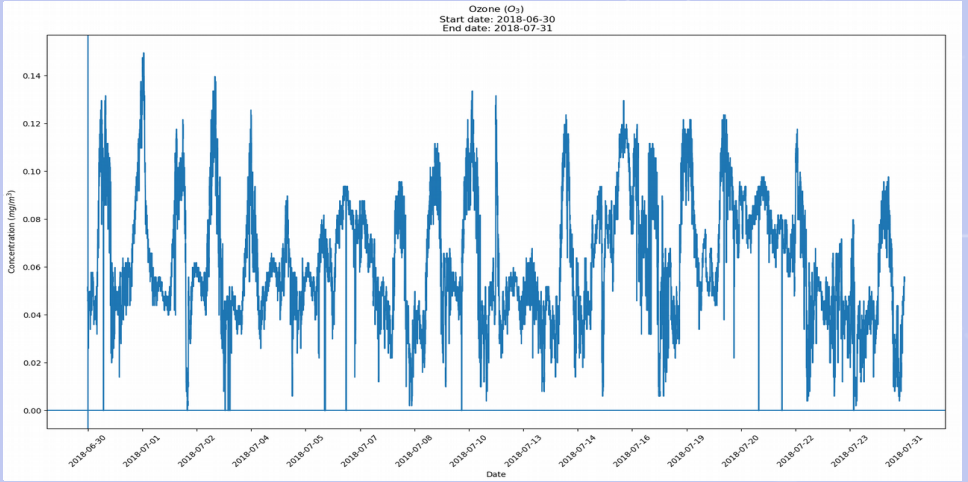
SALOME SHEKILADZE

SUPERVISORS – RAMAZ BOTCHORISHVILI, TINATIN DAVITASHVILI

Tbilisi State University

Georgian-German ScienceBRIDGE
CONNECTING PEOPLE AND KNOWLEDGE

HOW EVERYTHING HAS BEGUN?
WHAT IS OUR AIM?



- Started working at Tbilisi State SMART Atmosim Lab
- Monitor Tbilisi's air
- Solutions

Use device with high accuracy, cons – expensive, not compact.

Make sensors **SMARTER** *using machine learning*

Use cheap and compact sensors, cons – low accuracy.

INTERPOLATION AND EXTRAPOLATION

Our problem interpretation – Surface Approximation:

- First axis – Time
- Second axis – I Substance Measurement
- Third axis – II substance Measurement

  Aim:
- Predict II substance measurement knowing first and second axis values using extrapolation

RBF INTERPOLATION

Our problem interpretation – Surface Approximation:

Task:

$$\begin{bmatrix} \varphi_1(x_1) & \varphi_2(x_1) & \dots & \varphi_m(x_1) \\ \varphi_1(x_2) & \varphi_2(x_2) & & \varphi_m(x_2) \\ \vdots & & \ddots & \vdots \\ \varphi_1(x_m) & \varphi_2(x_m) & \dots & \varphi_m(x_m) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{bmatrix} = \begin{bmatrix} u(x_1) \\ u(x_2) \\ \vdots \\ u(x_m) \end{bmatrix} \qquad \varphi(d): \; d = \|x - x_i\| \quad x, x_i \in R^n$$

Solution with:

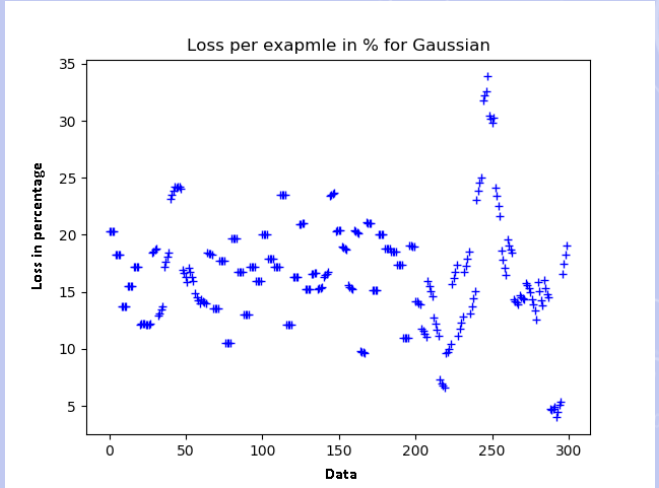
- Gaussian: $\varphi(d) = e^{-(\varepsilon d)^2}$
- Gielis Superformula:

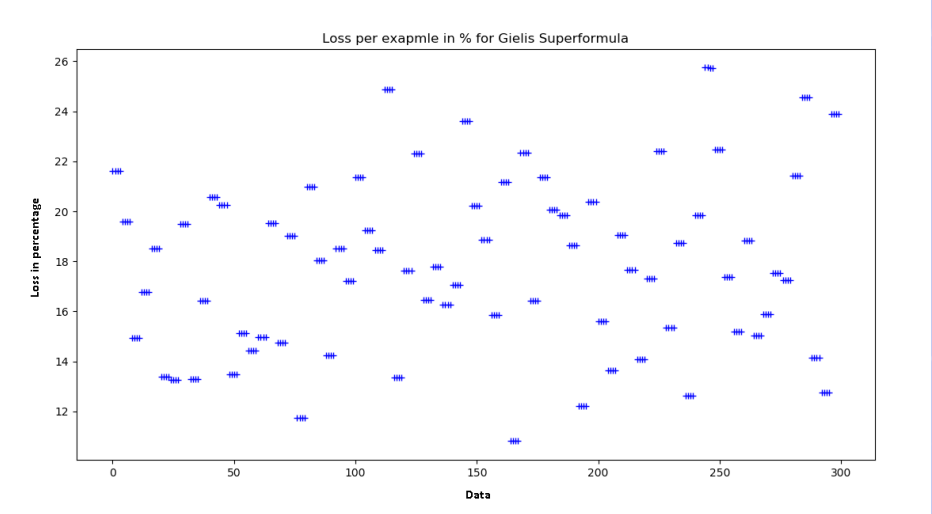
$$\theta(x, y, x_0, y_0) = \arccos \frac{x - x_0}{\sqrt{(x-x_0)^2 + (y-y_0)^2}}$$

$$r(\theta, n1, n2, n3, m, a, b) = \frac{1}{\left[\left(\frac{1}{a}\cos\left(\frac{m\theta}{4}\right)\right)^{n2} + \left(\frac{1}{b}\sin\left(\frac{m\theta}{4}\right)\right)^{n3}\right]^{\frac{1}{n1}}}$$

$$f(x, y, x_0, y_0) = \sqrt{(x - x_0)^2 + (y - y_0)^2}$$

$$\varphi(x, y) = \begin{cases} cos[f \cdot r] & , \; f \cdot r < \frac{\pi}{2} \\ 0 & , \; f \cdot r \geq \frac{\pi}{2} \\ 1 & , \; f = 0 \end{cases}$$

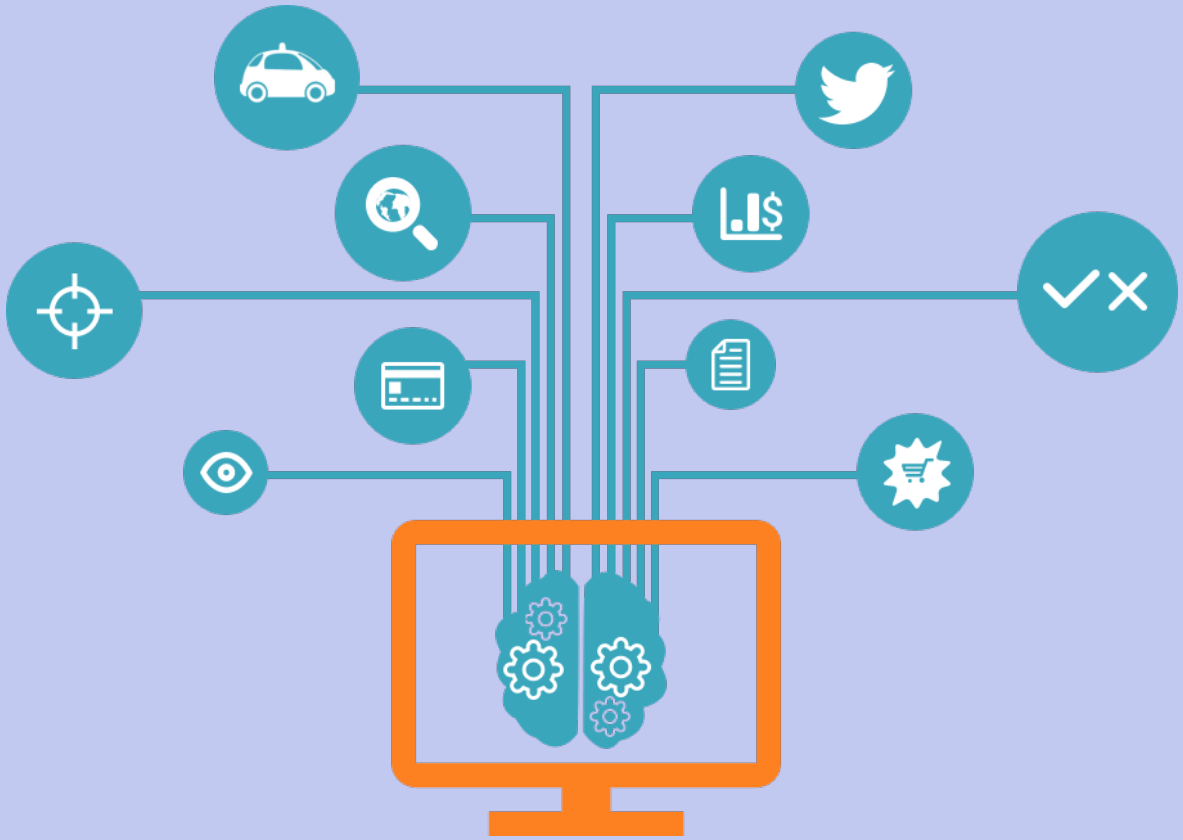




WHAT IS MACHINE LEARNING?

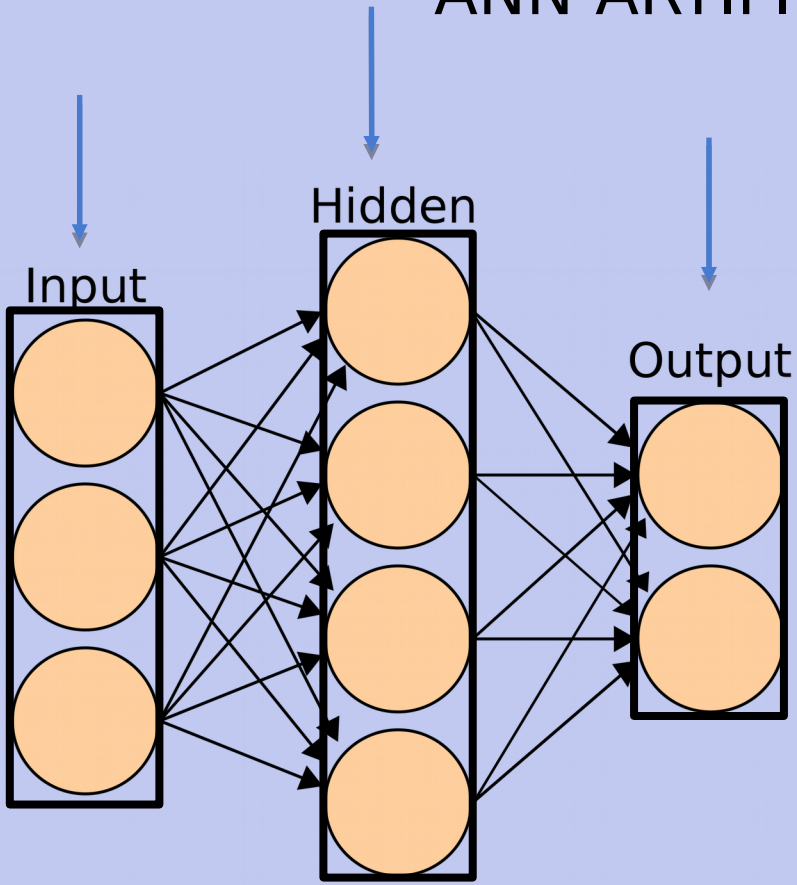
Field of study that gives computers the ability to learn without being explicitly programmed

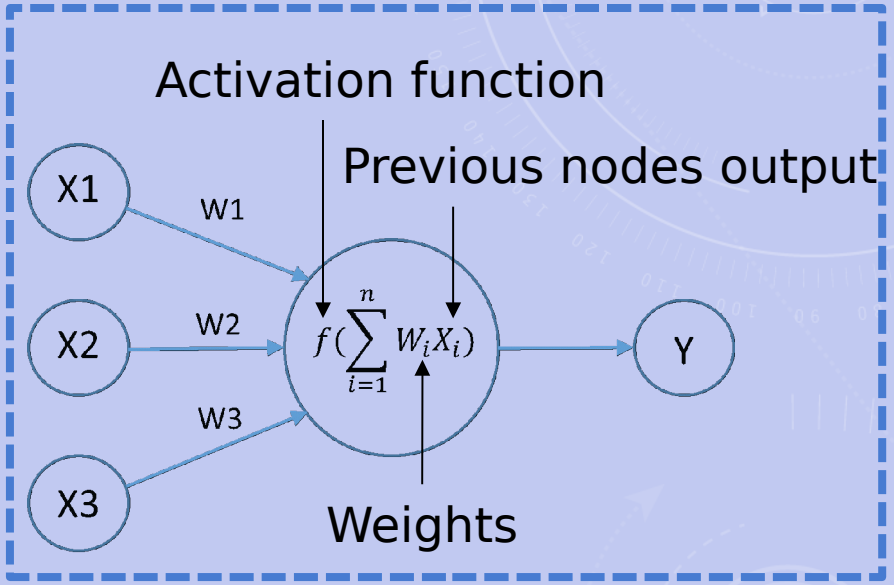
Arthur Samuel



FEW WORDS ABOUT ANN

ANN-ARTIFICIAL NEURAL NETWORK

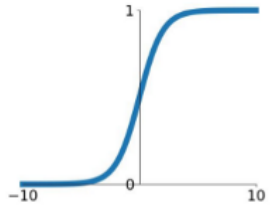




ACTIVATION FUNCTIONS

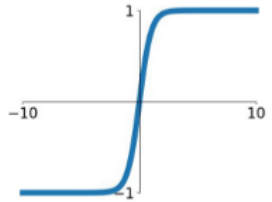
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



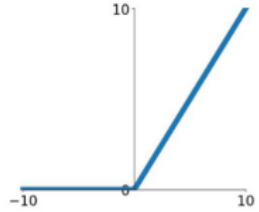
tanh

$$\tanh(x)$$



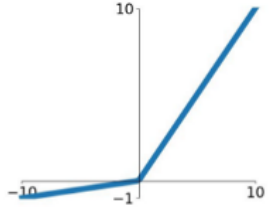
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

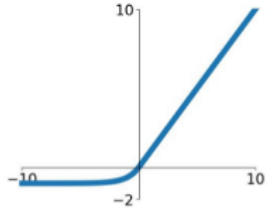


Maxout

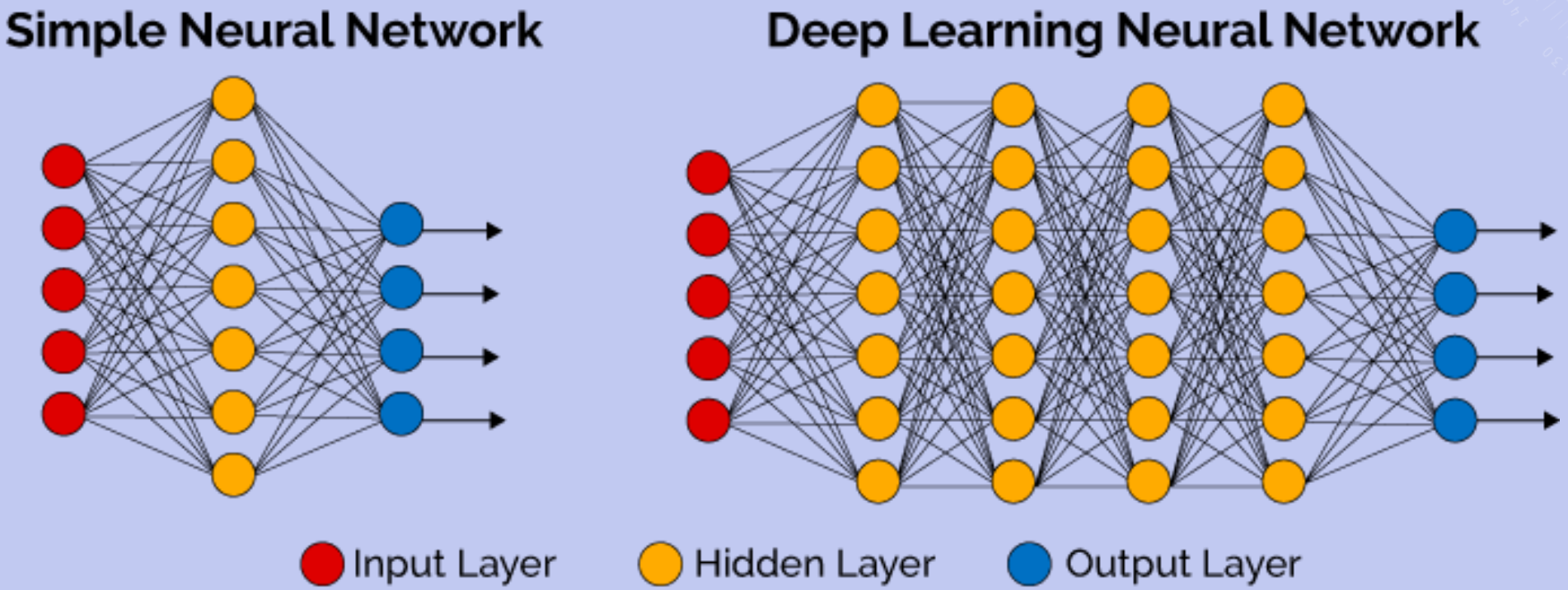
$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

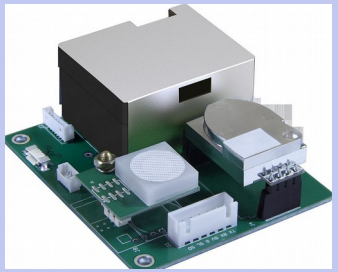
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

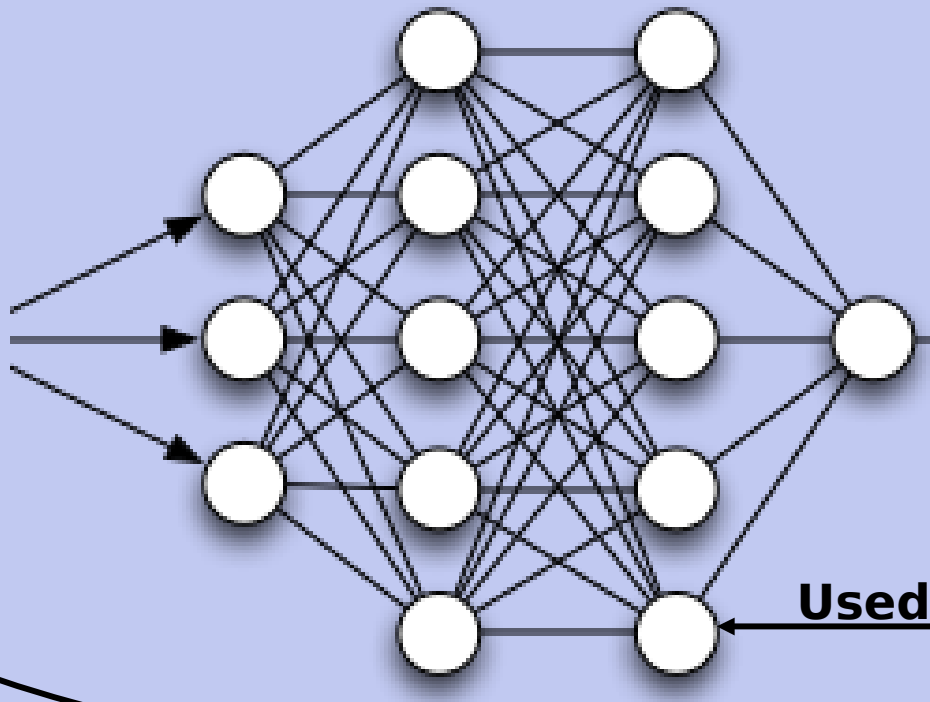


DEEP LEARNING



USE OF ANN FOR OUR TASK



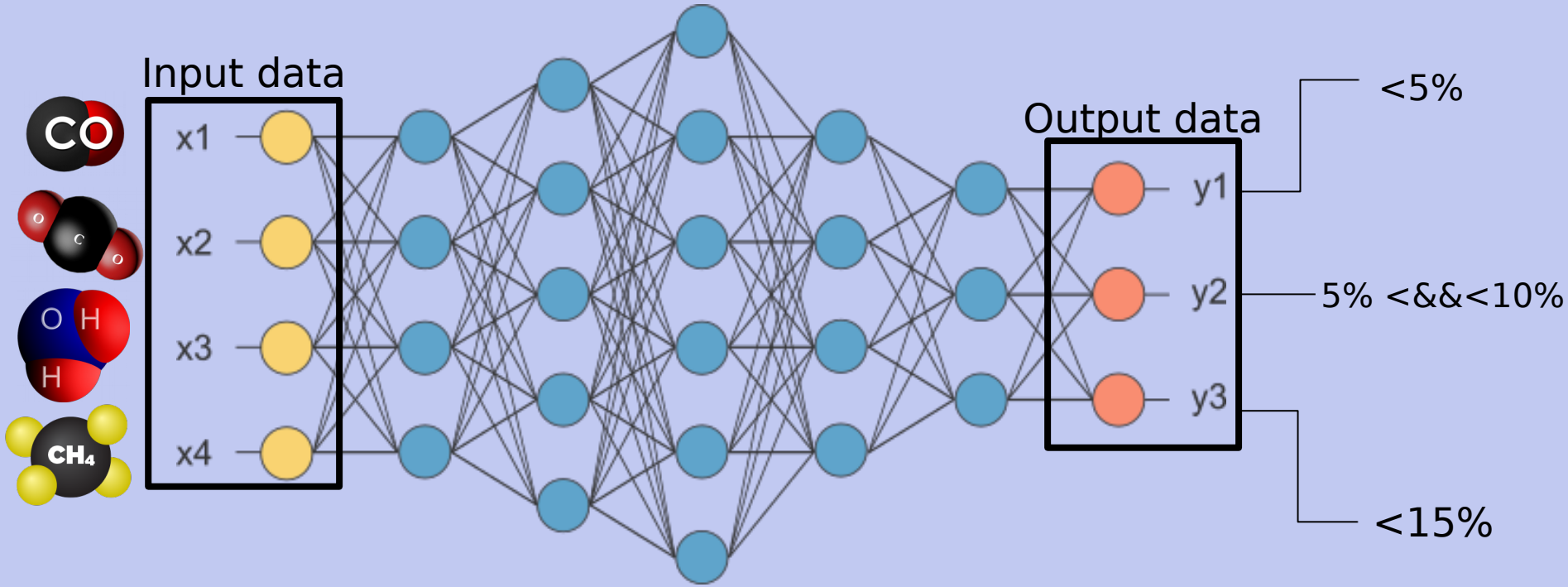


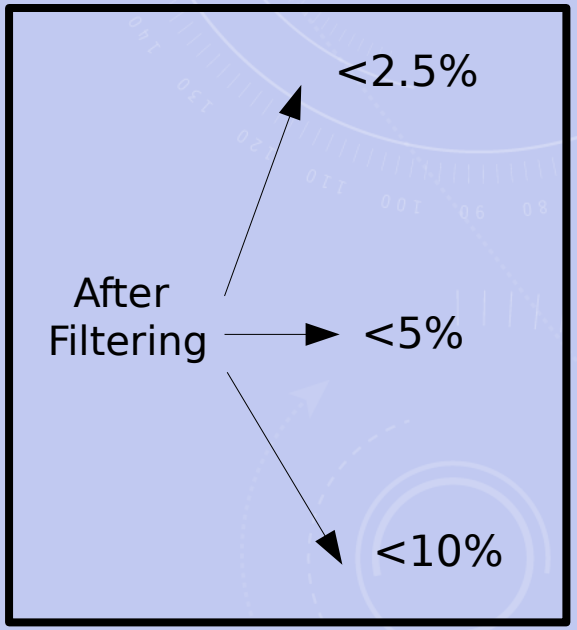
OUTPUT AIM **Min(|OUTPUT-CORRECT DATA|^2)**

Used for training

Correct data changed with NumPy uniform distribution

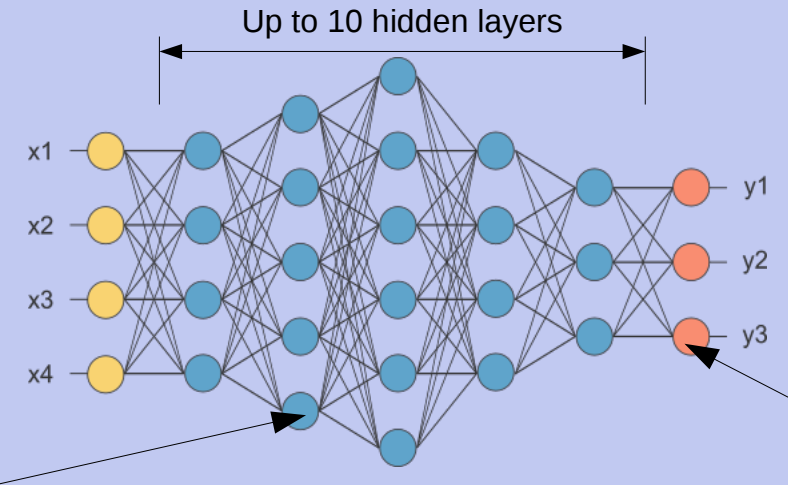
MODEL EXAMPLE AND FILTER DATA





MLP BUILT-IN CLASSIFIERS IN PYTHON

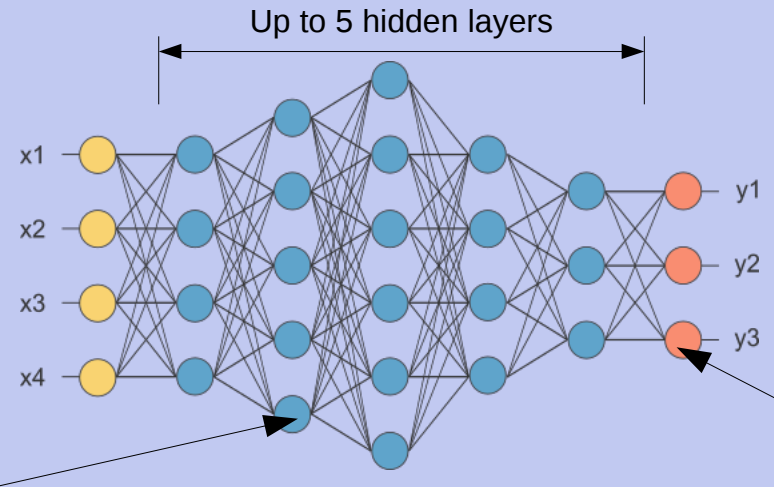
Result → **Average accuracy 59%**



MLP BUILT-IN CLASSIFIERS IN PYTHON

Keras

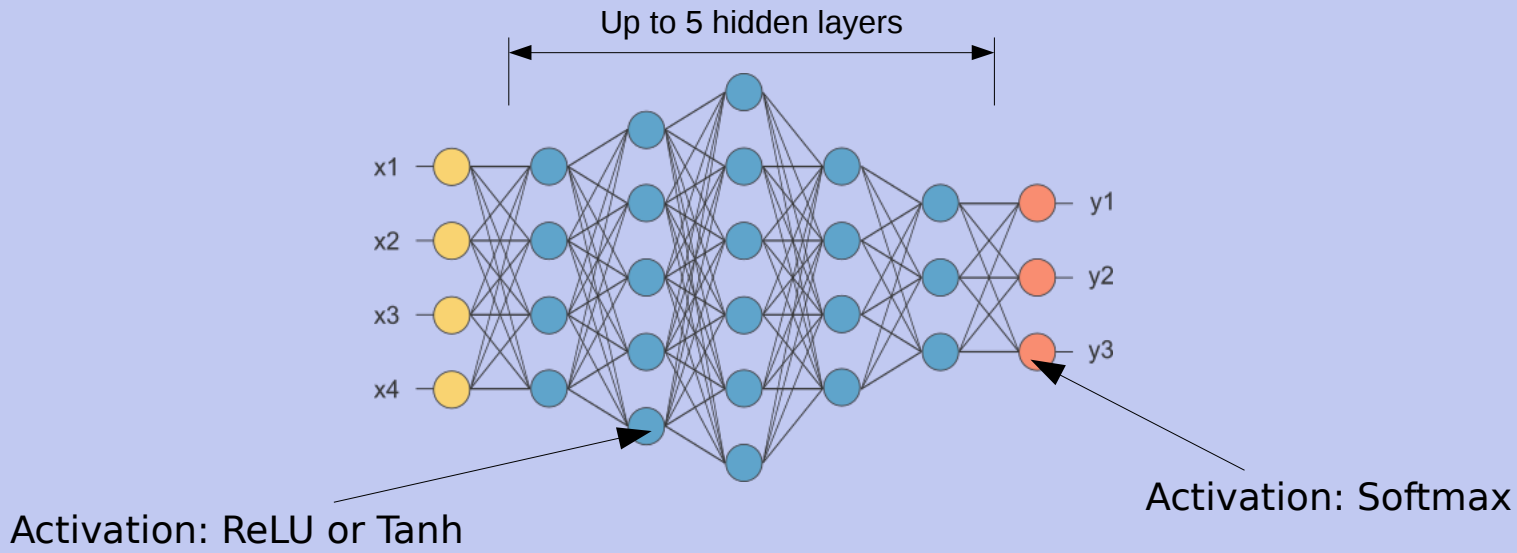
Result → **Average accuracy 68%**



MLP BUILT-IN CLASSIFIERS IN PYTHON

PYTORCH

Result → **Average accuracy 69%**

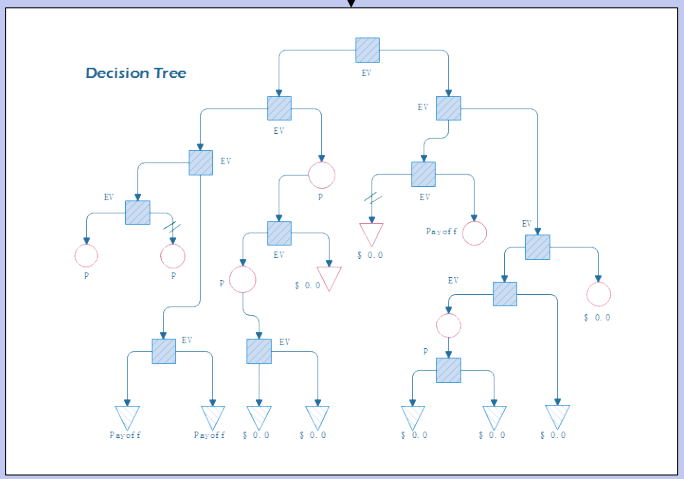


IS THIS PROBLEM INTERPRETATION CORRECT ?

dmlc

XGBoost

Result → **Accuracy 70%**



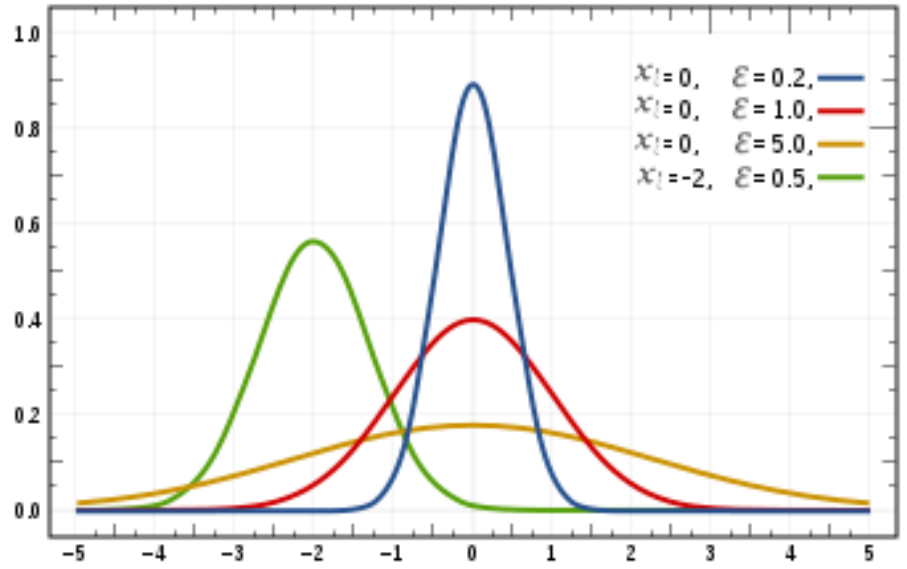
With use of ANN accuracy should be at least 70%

RBFNN – WHY THERE IS NOT BUILT-IN MODEL?

Gaussian

$\varphi(d): \quad d = \|x - x_i\| \quad x, x_i \in R^n$

$\varphi(d) = e^{-(\varepsilon d)^2}$



RBFNN – TESTED MODELS

For update weights in RBFNN we used:

- ADAM (Gradient descent)
- SGD (Stochastic Gradient Descent)

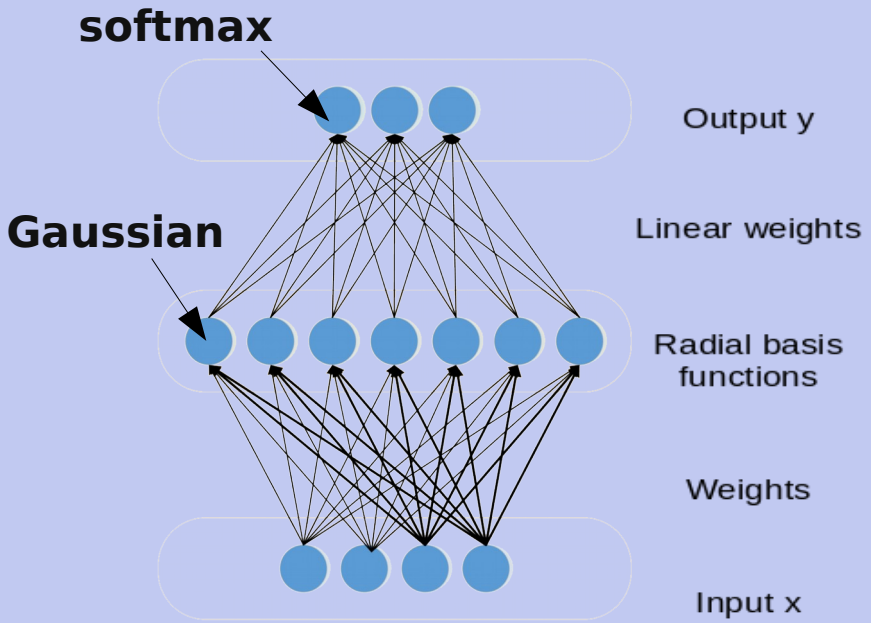
For learning-rate update we used:

- ReplaceItOnPlatoue (that divides learning parameter by factor)

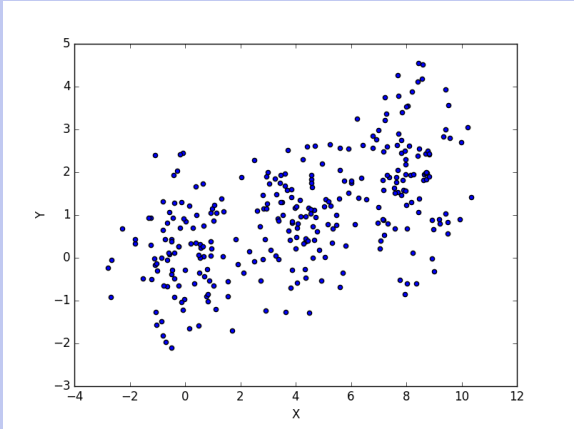
For measuring misfit we used :

- Cross entropy loss (for classification)
- MSE LOSS (Mean squared error loss (for regression))

RBFNN - RADIAL BASIS FUNCTION NEURAL NETWORK



Kmeans(for clustering)

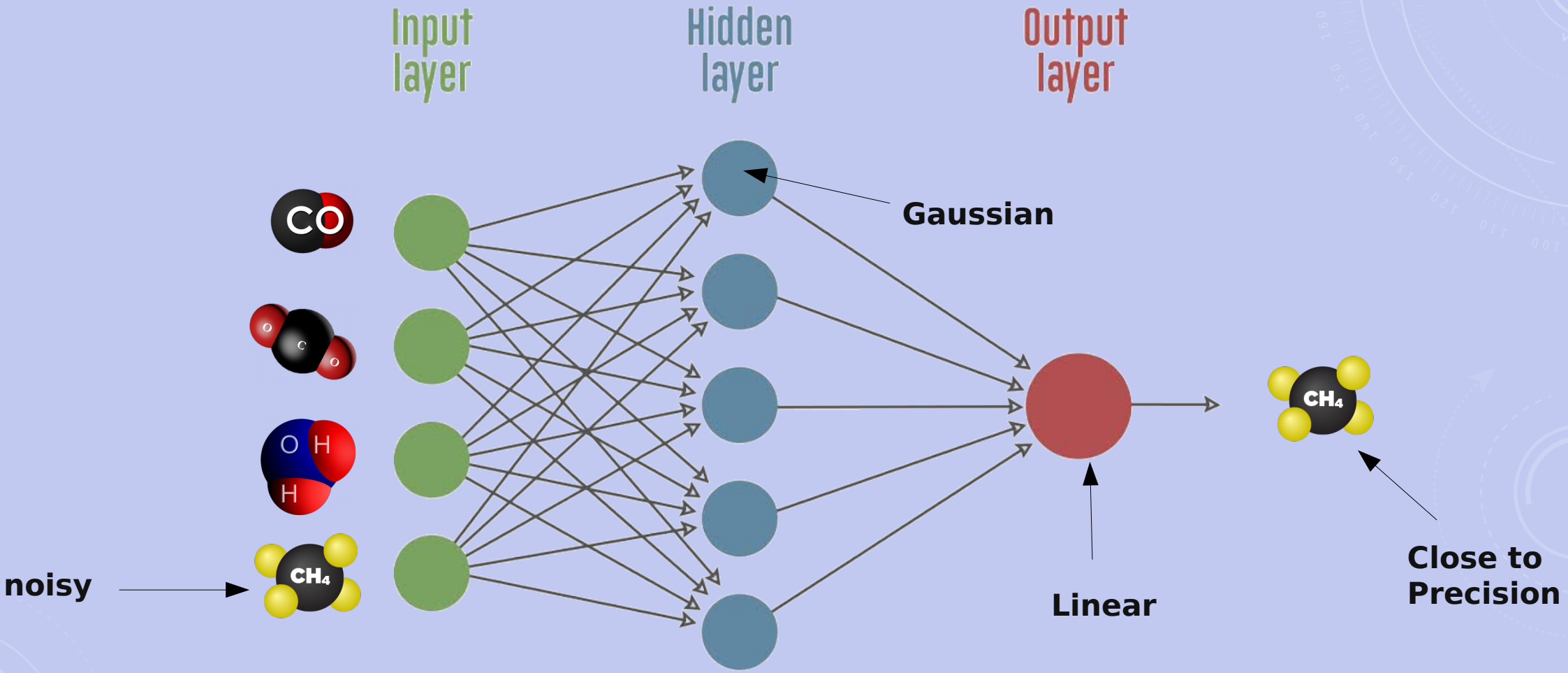


Results

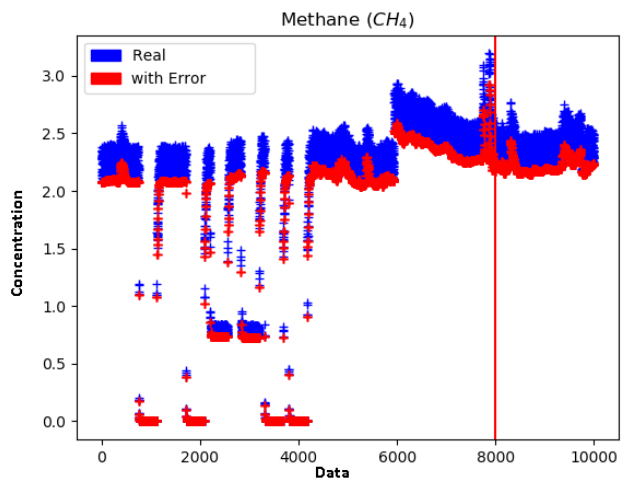
Accuracy 89%
With specially selected parameters

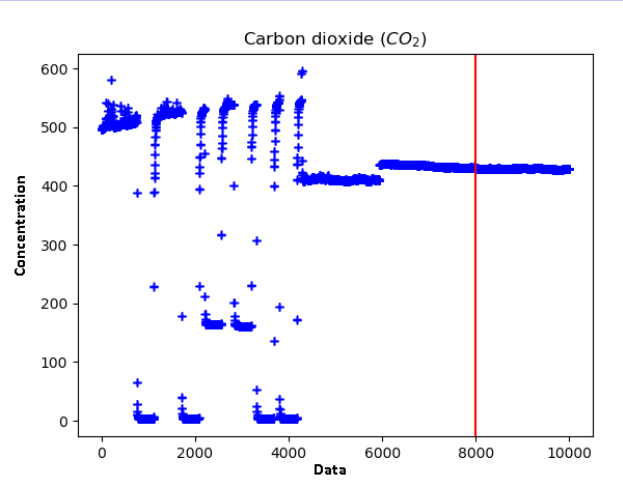
Average accuracy 78%

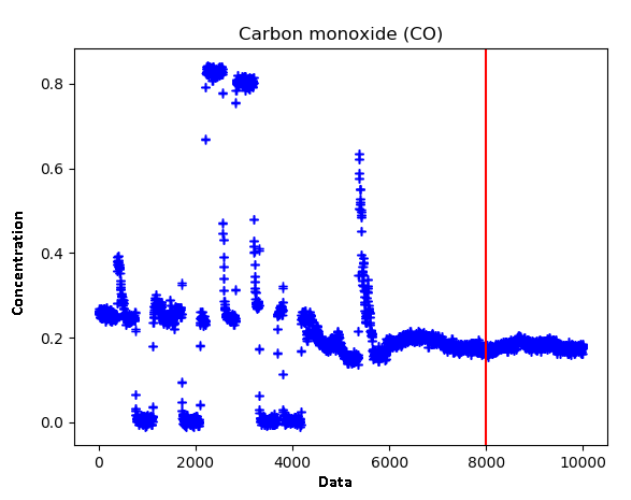
RBFNN – FROM CLASSIFICATION TO REGRESSION

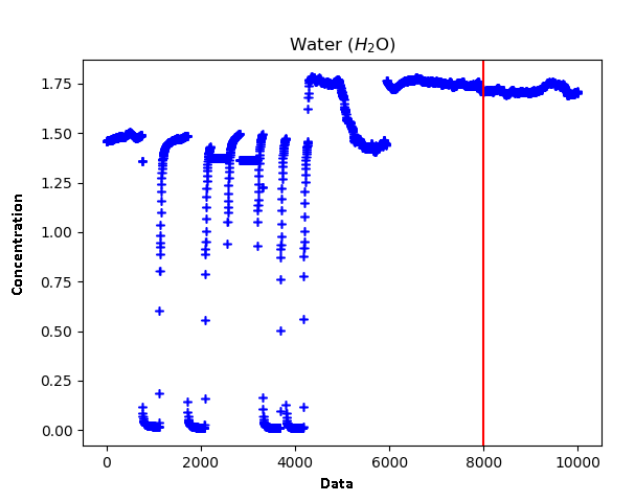


RBFNN REGRESSION MODEL DATA WE USED





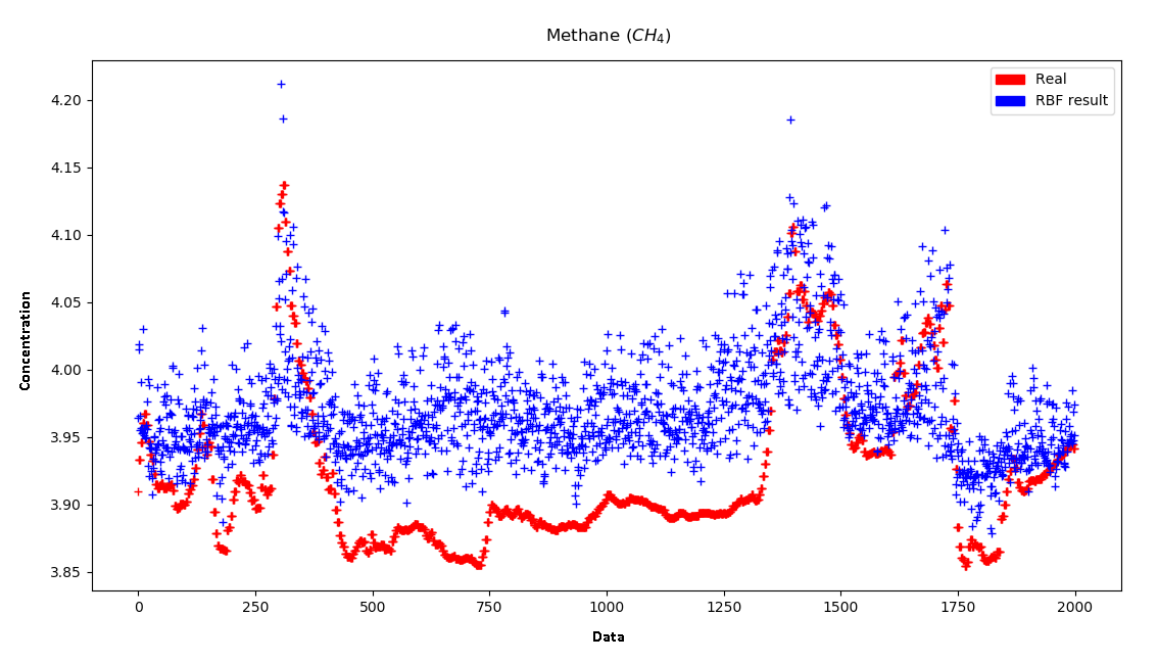


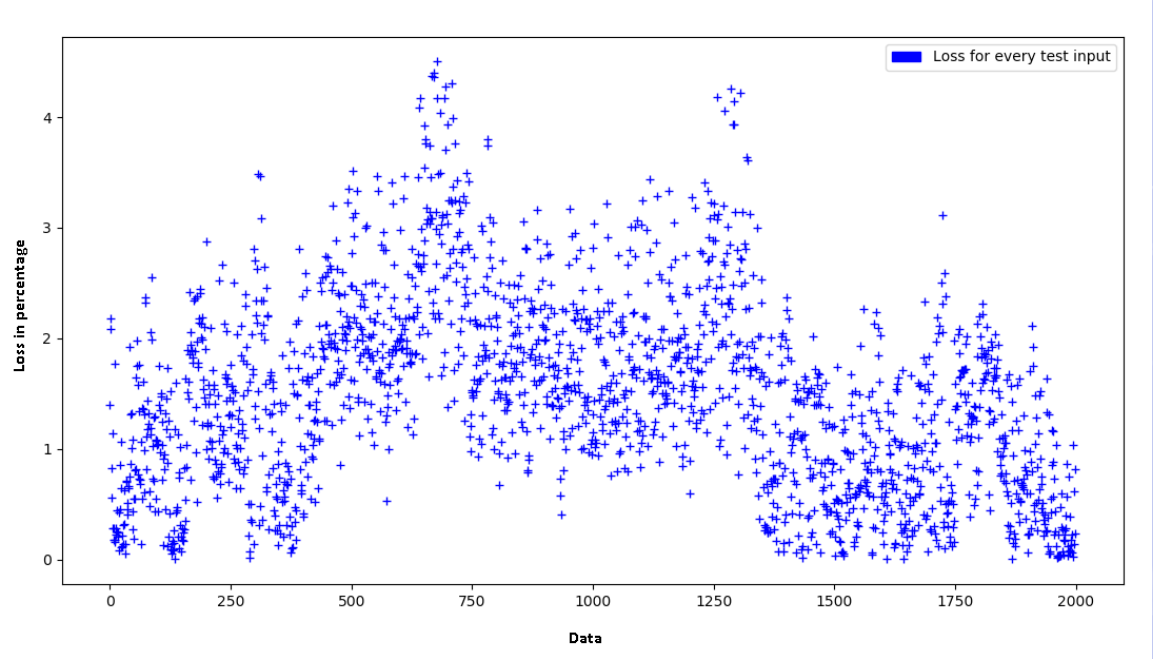


Data proportion → **8000-train 2000-test**

RESULTS

With special selected parameters





FUTURE PLANS

- **Choose appropriate shape parameters for activation functions in RBF neural network**
- **Prepare model for usage in SMART Lab**

Thank You for Attention