


72

## Montagsmaler 2

---

■ Muad, Han, Xia, John

■ Mittwoch 13:00 - 13:45





—— contents ——

0  
1

Aufgabenverteilung

0  
2

Besonderheiten  
und Probleme

0  
3

Live-Demo

# week 1

Basis GUI und  
simples  
neuronales  
Netzwerk

Anfangs hatten wir mehrere GUI Vorschläge, aber haben uns letztendlich für Muads GUI entschieden.

Man kann mit dem GUI ein Bild zeichnen, als Images Speichern und es in einem Array umwandeln .

In der dieser Woche  
haben wir das GUI  
gemacht.

Außerdem haben wir uns  
alle zu neuronale  
Netzwerke und deep  
learning informiert.

## week 2

GUI mit Zeichenfläche, neuronales Netzwerk mit linearer Klassifizierung

- über Perzeptronen informiert
- klassifizierung backward/forward propagation entschieden
- wir orientieren uns dabei an übung 2 & 3
- gui erweiterten über überarbeiten

Weixia, John: basic mathematics matrix operations

Muad: GUI ,datareader

Han: forward propagation

## week 3

### Neuronales Netzwerk mit nonlinearer Klassifizierung

in dieser Woche haben John und Xia ein Trainingsset erstellt und den Datenreader überarbeitet.

Muad und Han haben Backwardpropagation und Runterskalierung gemacht.

Muad, Han, Weixia, John.

# week 4

GUI mit neuronalem  
Netzwerk verbinden

In dieser woche haben wir:

- weights read/write
- bias read/write
- backwardpropagation

-training mit  
Datareader verbinden

-runterskalieren

In dieser woche haben wir:

In dieser Woche haben wir  
die Bugs im NeuralNetwork  
gefixt,.

JavaDoc Kommentare  
geschrieben,  
das GUI mit dem  
NeuralNetwork  
verbunden

- und die Präsentation  
erstellt

# Besonderheiten und Probleme



The first problem is that the image is converted into binary encoding, the first time I made a mistake, because I turned the file itself into binary, not the content of the image

If the pixel of the entire picture is used as input, the amount of data is too large, so the suggestion of nils is runter skalierung, make the picture smaller

Because it is not clear how much the picture can be reduced, so have to check the information



# Besonderheiten und Probleme

输入到隐藏

$$\begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix}$$

2行是有2个节点在 hidden layer

3列是有3个节点在 input layer

With Bias Bias

$$H = \text{Sigmoid}(W_{ij} \cdot I_i + B_1)$$

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

$$O = \text{Sigmoid}(W \cdot H + B_2)$$

$$W_{ij} = W_{\text{hidden, input}}$$

$$\begin{bmatrix} w_{11} & w_{12} & w_{13} & b_1 \\ w_{21} & w_{22} & w_{23} & b_2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix} = \begin{bmatrix} \dots & +b_1 \\ \dots & +b_2 \end{bmatrix}$$

$$M.length = row$$

[

error:

$$\begin{bmatrix} o_1 \\ o_2 \\ o_3 \end{bmatrix} - \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

output      input answer

getOutput:

1	rectangle	1.2
0	circle	0.1
0	line	0.3

List<double> Error:

$$([ ], [ ], [ ] \dots)$$

only one hidden layer **may** be not enough

At first, there was a debate about how many hidden layers, the confusion is that it is one or two hidden layers

but **luckily** later we find that, with one hidden layer we could also get a good result

# Besonderheiten und Probleme

gradient descent

$$E = (t - o)^2$$

$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial (\sum_n (t_n - o_n)^2)}{\partial w_{jk}} \quad t \text{ is const}$$

$$= -2(t_k - o_k) \cdot \frac{\partial o}{\partial w_{jk}}$$

$$= -2(t_k - o_k) \frac{\partial}{\partial w_{jk}} \text{sigmoid}(\sum_j w_{jk} \cdot o_j)$$

$$= -2(t_k - o_k) \cdot \text{sigmoid}(\sum_j w_{jk} \cdot o_j) (1 - \text{sigmoid}(\sum_j w_{jk} \cdot o_j)) o_j$$

$$w_{jk}^{\text{new}} = \text{old } w_{jk} - \alpha \cdot \frac{\partial E}{\partial w_{jk}}$$

$$w_{\text{old}} - w_{\text{new}} = \alpha \frac{\partial E}{\partial w_{jk}} = E_1 \cdot \text{sigmoid}(\sum_j w_{jk} \cdot o_j) (1 - \text{sigmoid}(\sum_j w_{jk} \cdot o_j)) o_j$$

$$= \begin{bmatrix} E_1 \cdot S_1 (1 - S_1) \\ E_2 \cdot S_2 (1 - S_2) \\ E_k \cdot S_k (1 - S_k) \end{bmatrix} \cdot \begin{pmatrix} o_1 & o_2 & o_j \end{pmatrix}$$

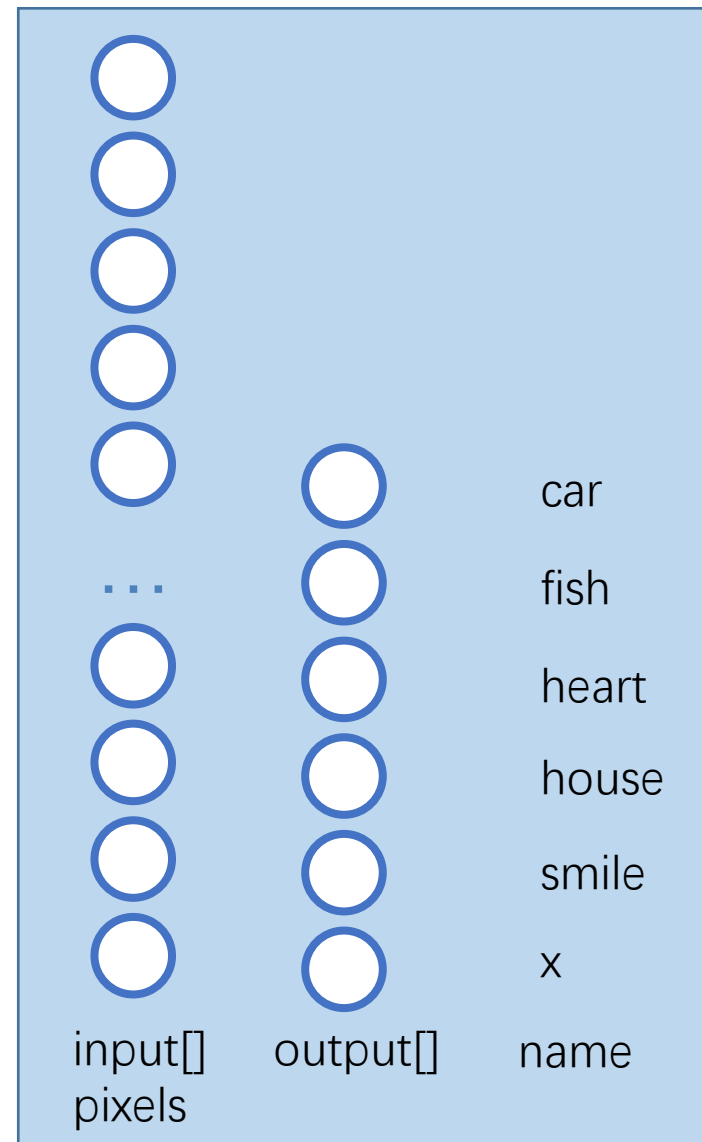
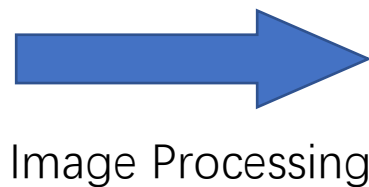
$$\Delta W = I_r \times E \times (O \times (1 - O)) \cdot H^T$$

The mathematical problem, the first theoretical formula, was organized into two papers

# Model



Image 450\*450



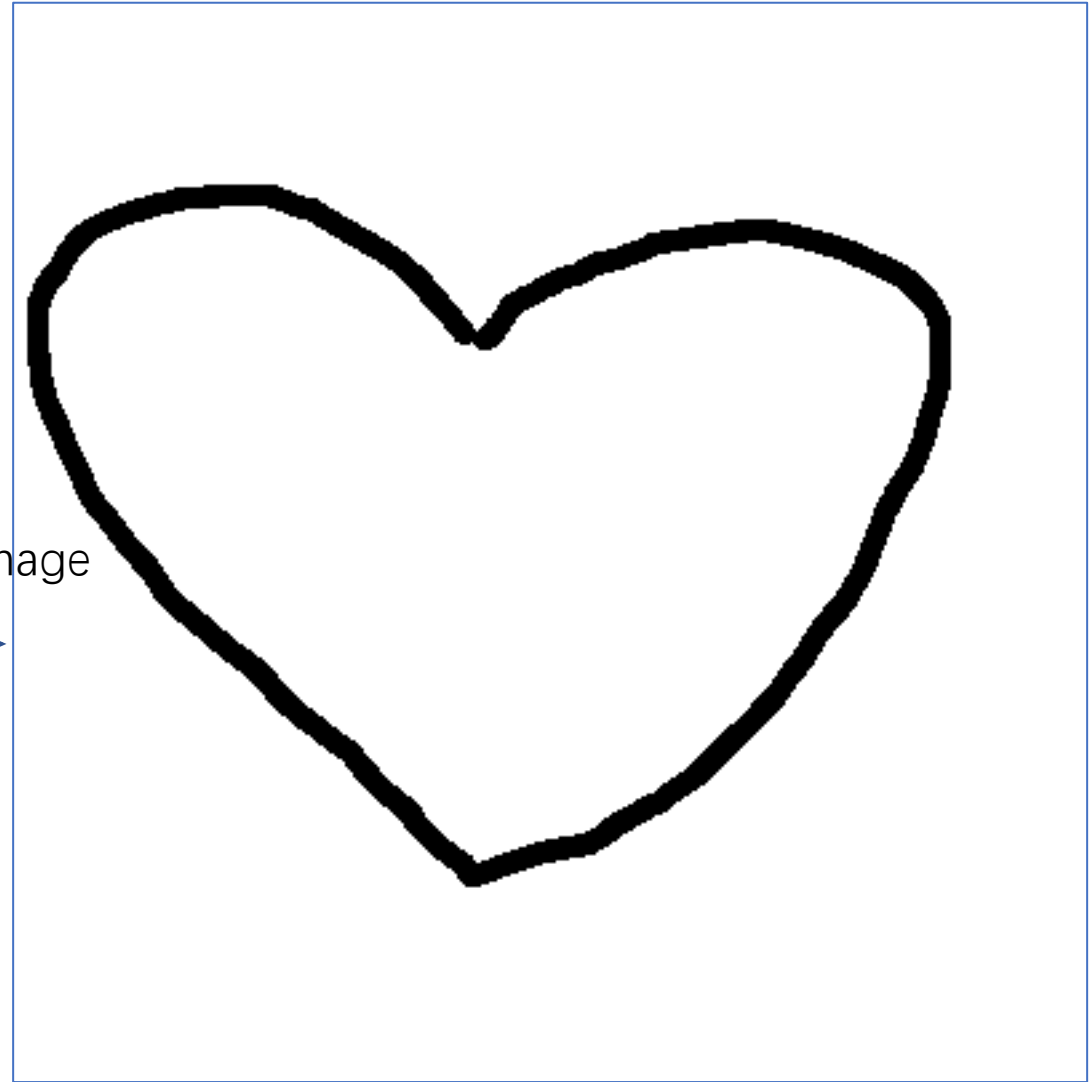
Data Type : Data

## Image Processing (step 1)



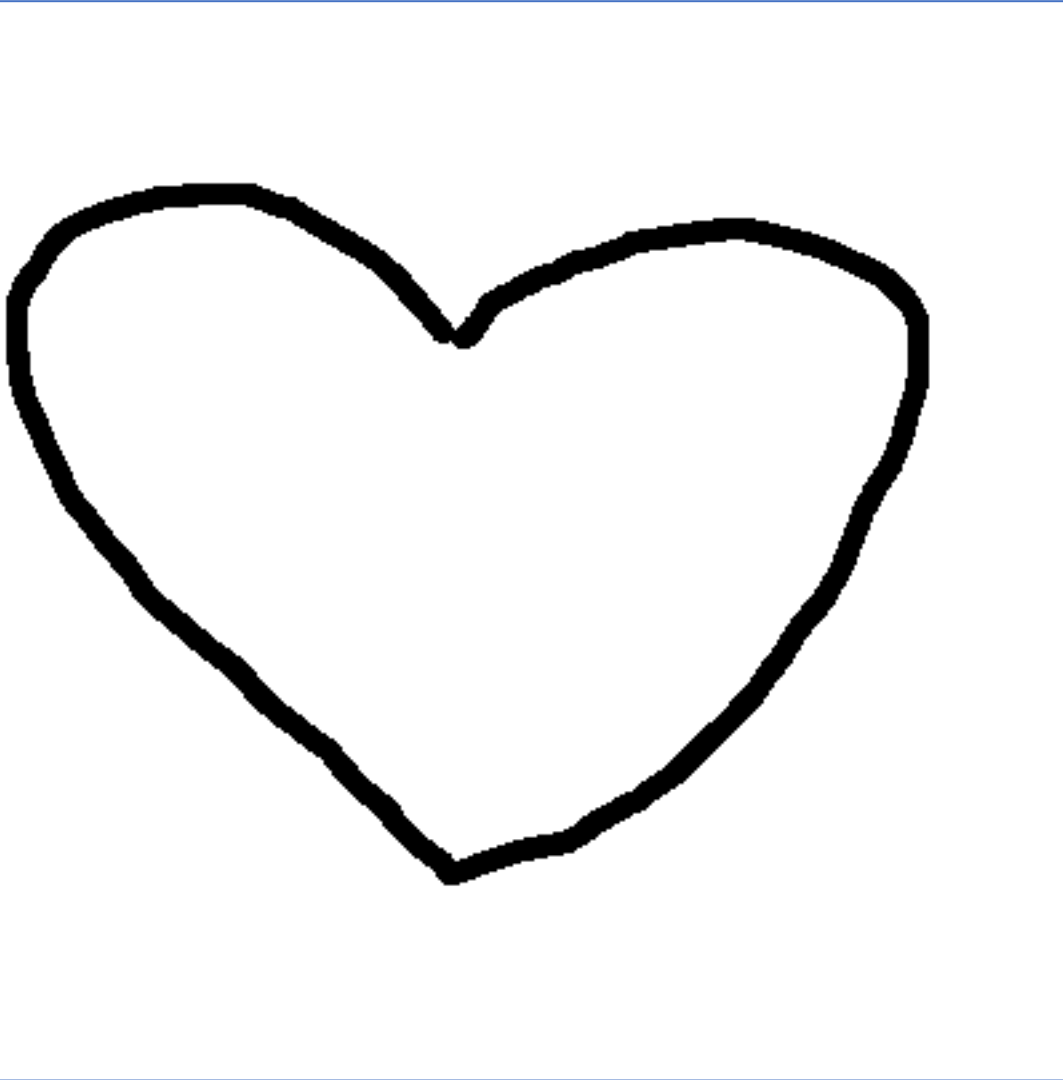
Image 450\*450

Convert to binary image



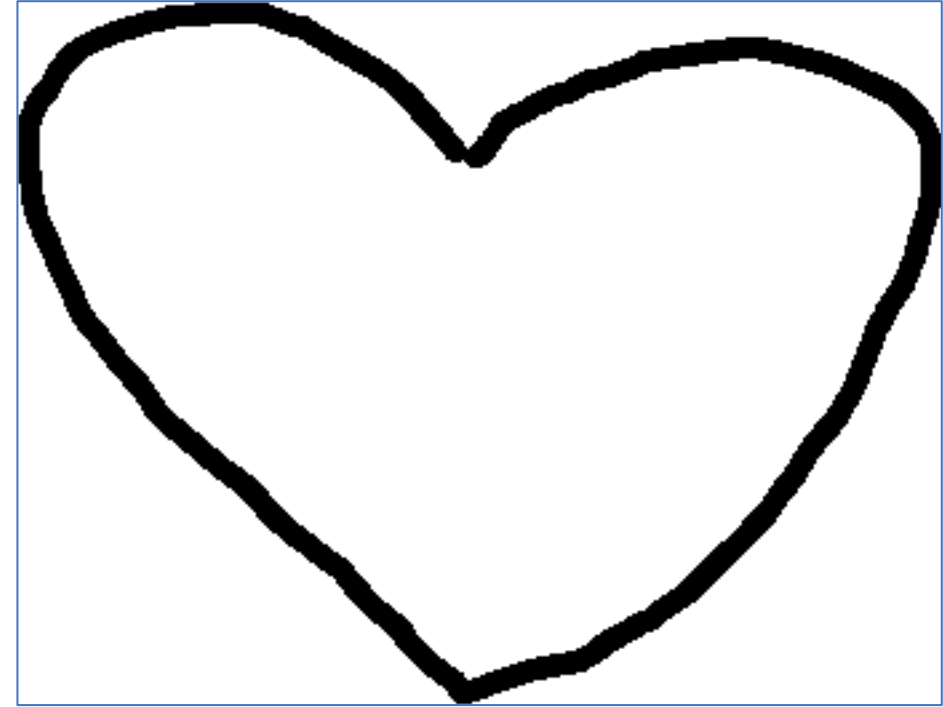
Binary image

## Image Processing (step 2&3)



Binary image

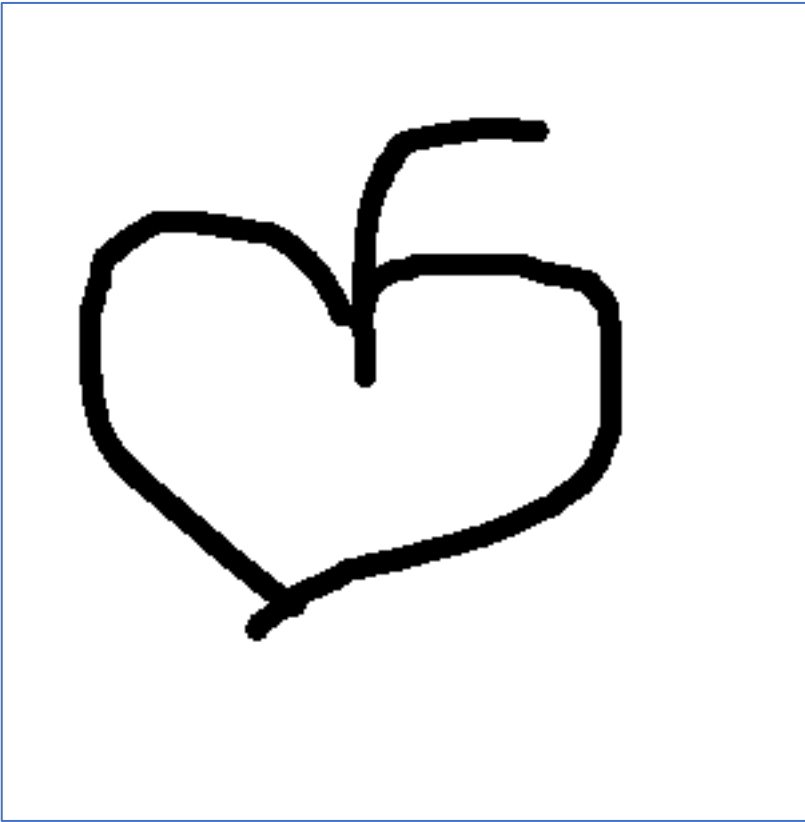
Cut out the white side



Scale to size 50\*50



## Research: Scale, to find the best size



30\*30

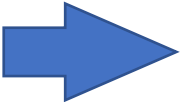


40\*40



50\*50

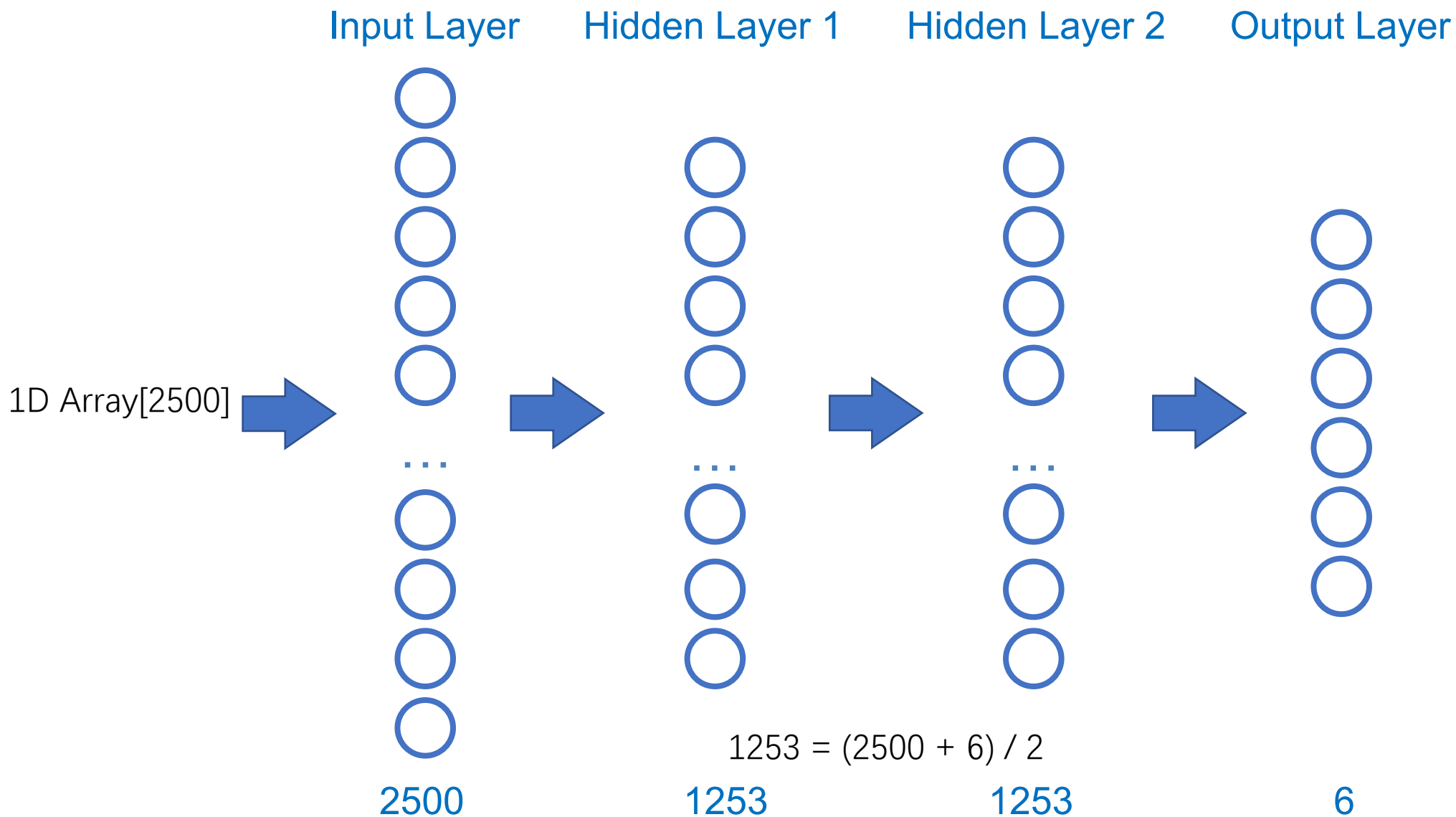
50\*50 contains most informations

[illegible]

1: black

1D Array[2500]

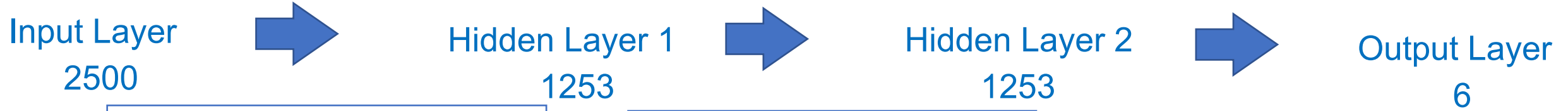
# Mathematischer Hintergrund



(actually the model with only 1 hidden layer was decided)



# Mathematischer Hintergrund: Forward



$$W^{ih}_{1253,2500} * X^{in}_{2500} + B^{ih}_{1253} = H^1$$

$$W^{hh}_{1253,1253} * H^1 + B^{hh}_{1253} = H^2$$

$$W^{ho}_{6,1253} * H^2 + B^{ho}_6 = O_{6,1}$$

$$\begin{bmatrix} w_{1,1} & \cdots & w_{1,2500} \\ \vdots & \ddots & \vdots \\ w_{1253,1} & \cdots & w_{1253,2500} \end{bmatrix} * \begin{bmatrix} x_1 \\ \vdots \\ x_{2500} \end{bmatrix} + \begin{bmatrix} b^1_1 \\ \vdots \\ b^1_{1253} \end{bmatrix} = \begin{bmatrix} h^1_1 \\ \vdots \\ h^1_{1253} \end{bmatrix}$$

weight matrix      input      bias      hidden 1

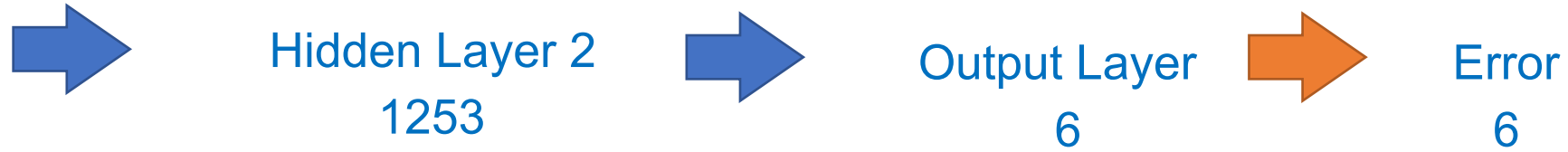
$$\begin{bmatrix} w_{1,1} & \cdots & w_{1,1253} \\ \vdots & \ddots & \vdots \\ w_{1253,1} & \cdots & w_{1253,1253} \end{bmatrix} * \begin{bmatrix} h^1_1 \\ \vdots \\ h^1_{1253} \end{bmatrix} + \begin{bmatrix} b^2_1 \\ \vdots \\ b^2_{1253} \end{bmatrix} = \begin{bmatrix} h^2_1 \\ \vdots \\ h^2_{1253} \end{bmatrix}$$

weight matrix      hidden1      bias      hidden 2

$$\begin{bmatrix} w_{1,1} & \cdots & w_{6,1253} \\ \vdots & \ddots & \vdots \\ w_{1253,1} & \cdots & w_{1253,1253} \end{bmatrix} * \begin{bmatrix} h^2_1 \\ \vdots \\ h^2_{1253} \end{bmatrix} + \begin{bmatrix} b^{ho}_1 \\ \vdots \\ b^{ho}_{1253} \end{bmatrix} = \begin{bmatrix} o_1 \\ \vdots \\ o_6 \end{bmatrix}$$

weight matrix      hidden 2      bias      output

# Mathematischer Hintergrund: Error



$$\begin{bmatrix} e_1 \\ \dots \\ e_6 \end{bmatrix} = \begin{bmatrix} o_1 \\ \dots \\ o_6 \end{bmatrix} - \begin{bmatrix} t_1 \\ \dots \\ t_6 \end{bmatrix}$$

Error      Output      Target



$$E_{hidden} = W^T_{HO} + Eou_{tput}$$
$$\begin{bmatrix} e_1 \\ \dots \\ e_6 \end{bmatrix}$$

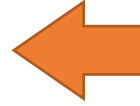
Error

# Mathematischer Hintergrund: Backword

Input Layer  
2500



Hidden Layer 1  
1253



Hidden Layer 2  
1253



Output Layer  
6

Calculate Error donation:

$$E_{hidden} = W_{HO}^T + E_{ou_{tput}}$$

Calculate Weight Difference:

$$\Delta W = Lr * E * (O' * (1 - O')) \cdot HT$$

Lr: learning rate

$$O' = \text{sigmoid}(O)$$

$$\Delta W_{ho} = \begin{bmatrix} w_{1,1} & \cdots & w_{1,2500} \\ \vdots & \ddots & \vdots \\ w_{1253,1} & \cdots & w_{1253,2500} \end{bmatrix} = Lr * \begin{bmatrix} e^o_1 \\ \dots \\ e^o_6 \end{bmatrix} + \begin{bmatrix} o'_1 \\ \dots \\ o'_6 \end{bmatrix} * (1 - \begin{bmatrix} o'_1 \\ \dots \\ o'_6 \end{bmatrix}) \cdot [h^2_1 \quad \dots \quad h^2_{1253}]$$

$$\Delta W_{hh} = \begin{bmatrix} w_{1,1} & \cdots & w_{1,2500} \\ \vdots & \ddots & \vdots \\ w_{1253,1} & \cdots & w_{1253,2500} \end{bmatrix} = Lr * \begin{bmatrix} e^{h21} \\ \dots \\ e^{h26} \end{bmatrix} + \begin{bmatrix} h^{2'}_1 \\ \dots \\ h^{2'}_6 \end{bmatrix} * (1 - \begin{bmatrix} h^{2'}_1 \\ \dots \\ h^{2'}_6 \end{bmatrix}) \cdot [h^1_1 \quad \dots \quad h^1_{1253}]$$

$$\Delta W_{ih} = \begin{bmatrix} w_{1,1} & \cdots & w_{1,2500} \\ \vdots & \ddots & \vdots \\ w_{1253,1} & \cdots & w_{1253,2500} \end{bmatrix} = Lr * \begin{bmatrix} e^{h11} \\ \dots \\ e^{h16} \end{bmatrix} + \begin{bmatrix} h^{1'}_1 \\ \dots \\ h^{1'}_6 \end{bmatrix} * (1 - \begin{bmatrix} h^{1'}_1 \\ \dots \\ h^{1'}_6 \end{bmatrix}) \cdot [x_1 \quad \dots \quad x_{2500}]$$

$$\Delta B = Lr * \begin{bmatrix} e_1 \\ \dots \\ e_6 \end{bmatrix} + \begin{bmatrix} o'_1 \\ \dots \\ o'_6 \end{bmatrix} * (1 - \begin{bmatrix} o'_1 \\ \dots \\ o'_6 \end{bmatrix})$$

Update Bias:

$$B = B + \Delta B$$

Update Weight:

$$W = W + \Delta W$$

# Structure

Class	Describe
Data	a data structure
Data Reader	operations with data and images, image processing
GUI	Functions to pain with convey and pen
Matrix	mathematic functions
Neural Network	The neural network, with input layer, output layer, hidden layers, error, forward and backward propergation Provides functions, like train and test

