

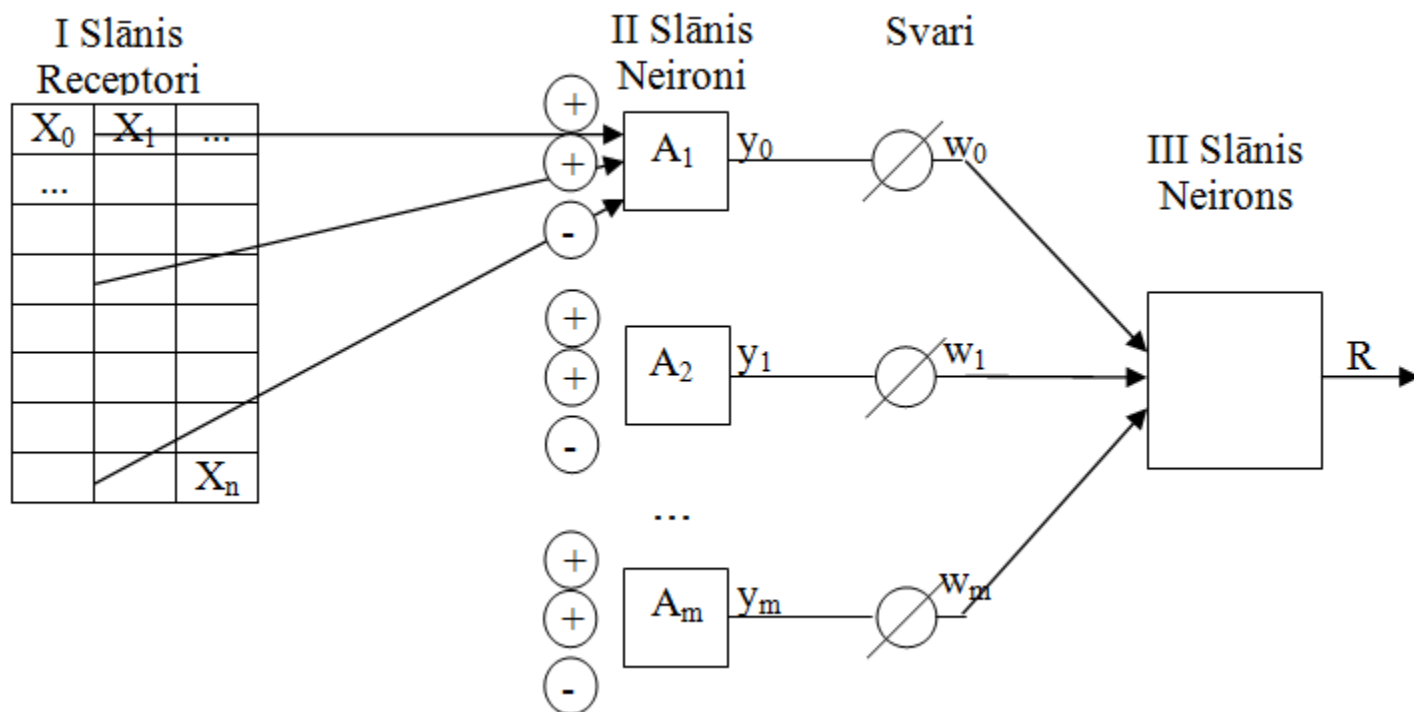
# Neironu tīkli

# Kas ir neironu tīkls?

---

- Neironu tīkla struktūra
- Receptoru slānis
- Neironu slānis
- Reaģējošais slānis
- Neirona tīkla apmācība
- Neironu tīkla uzlabošana

# Neironu tīkla struktūra



# Receptoru slānis

- **Receptoru slānis** kalpo lai iegūt informāciju no objekta. Slānis sastāv no receptoriem, kuriem ir divi stāvokļi: ierosināts ( $x_i=1$ ) vai neierosināts ( $x_i=0$ ). Mūsu gadījumā vai noteiktais pikselis ir melns (1) vai balts (0).

$X_0$	$X_1$	...
...		
		$X_n$

# Neironu slānis

- **Neironu slānis (Asociatīvais elements A)** kalpo iegūtas informācijas apstrādei. Tas sastāv no neironiem, kuriem ir 3 ieejas (divas pozitīvas un viena negatīva) un viena izeja. Katra ieeja ir pievienota nejaušam receptoram ar + vai - saiti. Neirona izejas vērtība var būt 1 vai 0:

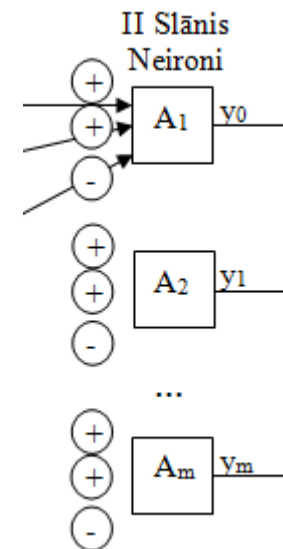
$$\phi_j = \begin{cases} 1, & \text{ja } \sum_{i=1}^n S_{ij} \cdot X_i - \theta \geq 0 \\ 0, & \text{ja } \sum_{i=1}^n S_{ij} \cdot X_i - \theta < 0 \end{cases}$$

kur  $\theta$  ir sliekšņa vērtība,  $S_{ij}$  – matrica, kas apraksta neirona saites ar receptoriem:

$S_{ij} \in [-1; 0; +1]$ ,  $i$  - receptora numurs,  $j$  – neirona numurs.

- $S$  matricas ne-nulles elementu skaitam ir jābūt vienādam ar neirona ieeju skaitu:

$$\sum_{i=1}^n |S_{i,j}| = 1$$

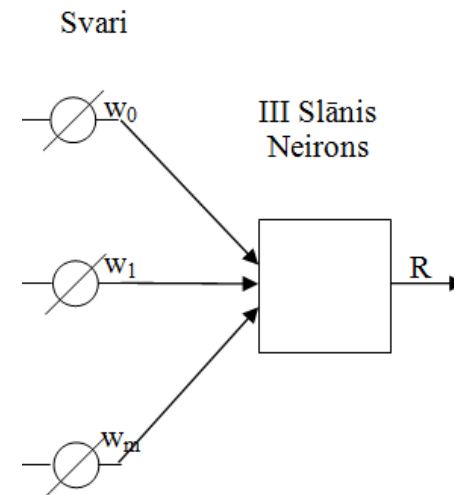


# Reaģējošais slānis

- **Reaģējošais slānis** kalpo rezultāta iegūšanai. Katra neirona izejai tiek pievienoti svari, kas ietekmē tā rezultātu un tā svarīgumu. Svaru sākotnēja vērtība ir nejauša, pēc tām svara vērtības var tikt koriģētas. Perceptrona izejas vērtību iegūst pēc formulas:

$$R = \begin{cases} 1, & \text{ja } \sum_{j=1}^n W_j \cdot \phi_j \geq 0 \\ 0, & \text{ja } \sum_{j=1}^n W_j \cdot \phi_j < 0 \end{cases}$$

kur  $\phi_j$  ir neironu svāri.



# Neironu tīkla apmācība

- **Parametriskā adaptācija**

- Parametriskā adaptācija notiek perceptrona apmācības laikā. Svari tiek koriģēti sekojošajā veidā:

$$W_j^{n+1} = \begin{cases} W_j^{n+1} + \varphi_j(1 - 2R), & \text{if errors exists} \\ W_j^n, & \text{if no errors exists} \end{cases}$$

kur n ir apmācības iterācijas numurs

- Iterācijas beidzās, kad nebūs kļūdu, vai kļūdu skaits būs vienāds katrā iterācijā, vai iterāciju skaits pārsniedz kādu noteiktu vērtību.

# Neironu tīkla uzlabošana

- **Strukturālā adaptācija**

- Strukturālās adaptācijas gadījumā tiek mainīta (nejauši, no jauna ģenerēta) matrica tā, lai pēc apmācības būtu pēc iespējas mazāk kļūdu.

- **Komiteju metodes**

- Vienlīdzības balsošanas algoritms
- Svērtas balsošanas algoritms
- Ansambla balsošanas algoritms



# Kā ieprogrammēt Neironu tīklu?

---

- Lietotāju saskarne
- Globālie mainīgie
- Neironu tīkla veidošana un skaitīšana
- Neironu tīkla apmācība
- Neironu tīkla uzlabošana\*

# Lietotāju saskarne.

- Receptorus ( $X_i$ ), neironus ( $Y_i$ ), svarus ( $W_i$ ) un saites starp receptoriem un neironiem ( $S_{ij}$ ) var attēlot ar StringGrid tabulām
- Attēlus lietotājam var parādīt izmantojot Image vai SpeedButton

Form1

Open Training set (TS)

Open Exam Set (ES)

Teta =

Result:

Info:  
Neural Network was trained  
in 13 iterations

	X
0	0
1	0
2	1
3	1
4	0
5	0
6	0
7	1
8	0
9	0
10	1
11	0
12	1
13	0
14	0
15	0
16	0
17	1
18	1
19	0
20	0
21	0
22	0

S	y0	y1	y2	y3	y4	y5	y6
x0	0	0	0	0	0	0	0
x1	0	0	0	0	0	0	0
x2	0	0	0	0	0	0	0
x3	0	0	0	0	0	0	0
x4	0	0	0	0	0	0	-1
x5	0	0	0	0	0	0	0
x6	0	0	0	0	0	-1	0
x7	0	0	0	0	0	0	0
x8	0	0	1	0	0	0	0
x9	0	0	0	0	0	-1	0
x10	0	0	0	0	0	0	0
x11	0	0	0	0	0	0	0
x12	0	0	0	0	0	-1	0
x13	0	-1	0	0	0	0	0
x14	0	1	0	0	-1	0	0
x15	0	0	0	1	0	0	0
x16	0	0	0	0	0	0	0
x17	1	0	0	0	0	0	0
x18	0	0	0	0	0	0	0
x19	0	0	0	0	0	0	0
x20	0	0	0	0	0	0	0
x21	0	0	0	0	0	1	0

Y	W
0	0
1	1
2	0
3	0
4	1
5	0
6	0
7	0
8	1
9	0
10	0
11	1
12	1
13	1
14	1
15	1
16	1
17	1
18	1
19	0
20	0
21	1
22	1

# Globālie mainīgie.

type

vec60i = array[0..59] of integer; //receptoriem

vec40i = array[0..39] of integer; //neironiem

vec40f = array[0..39] of Single; //svariem

strct = array[0..59,0..39] of integer; //saitem

var

x: vec60i; //receptori

y: vec40i; //neironi

w: vec40f; //svari

s: strct; //saites starp neironiem un receptoriem

r: integer; //rezultats

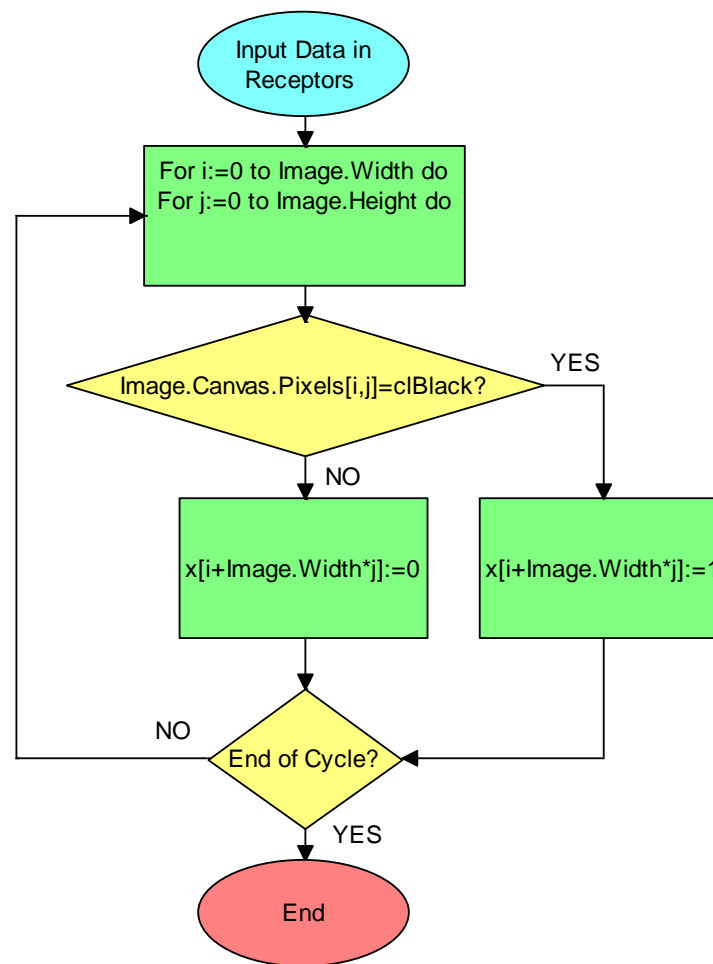
teta: real; //teta, lai adaptēt svarus

currentClass: integer; //lai zināt, kurai klasei pieder objekts (tas ir nepieciešams apmācībai)

# Neironu tīkla veidošana un skaitīšana.

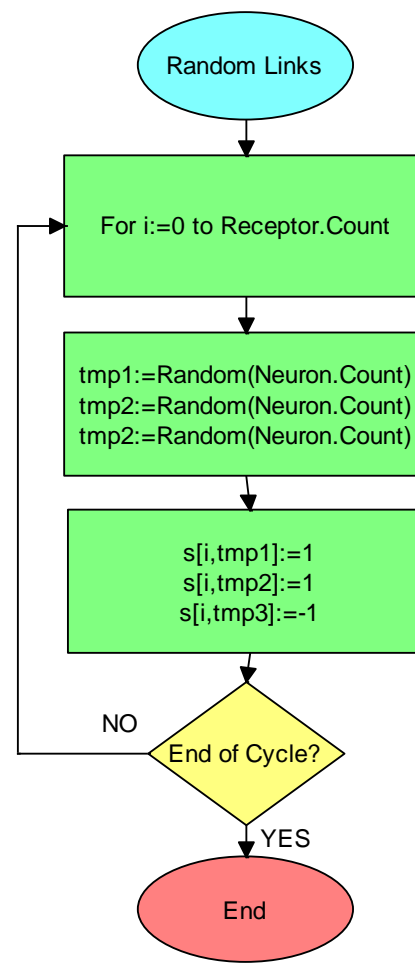
## Datu ievade

- procedure NN\_Input\_X();  
//Ievadīt datus no konkrēta attēla iekšā  $X_i$  receptoros (mainīgajā  $x[i]$ ). Datus no  $x[i]$  pēc tam var izvadīt tabulā StringGrid.



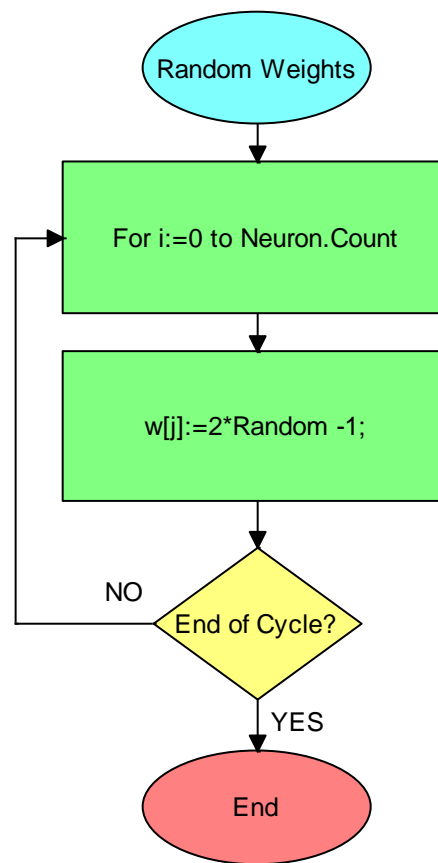
# Neironu tīkla veidošana un skaitīšana. Saites.

- procedure  
NN\_Random\_Links();  
//Nejauši salikt saites starp  
neironiem un receptoriem  
(mainīgajā  $s[i,j]$ ). Datus no  
 $s[i,j]$  pēc tam var izvadīt  
tabulā StringGrid.



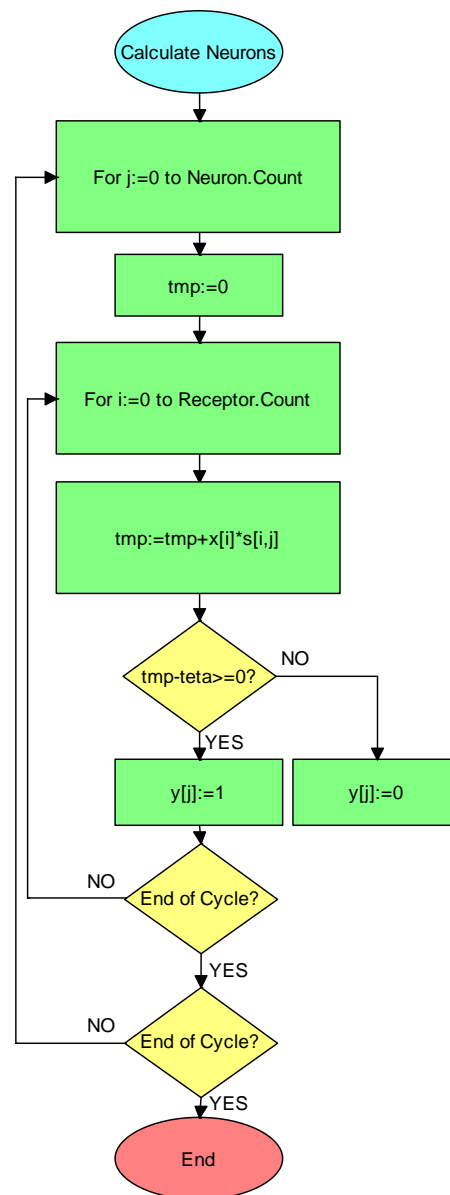
# Neironu tīkla veidošana un skaitīšana. Svari.

- procedure  
NN\_Random\_W();  
//Nejauši salikt svarus  
(mainīgajā w[i]). Pēc tam var  
izvadīt datus tabulā  
StringGrid.



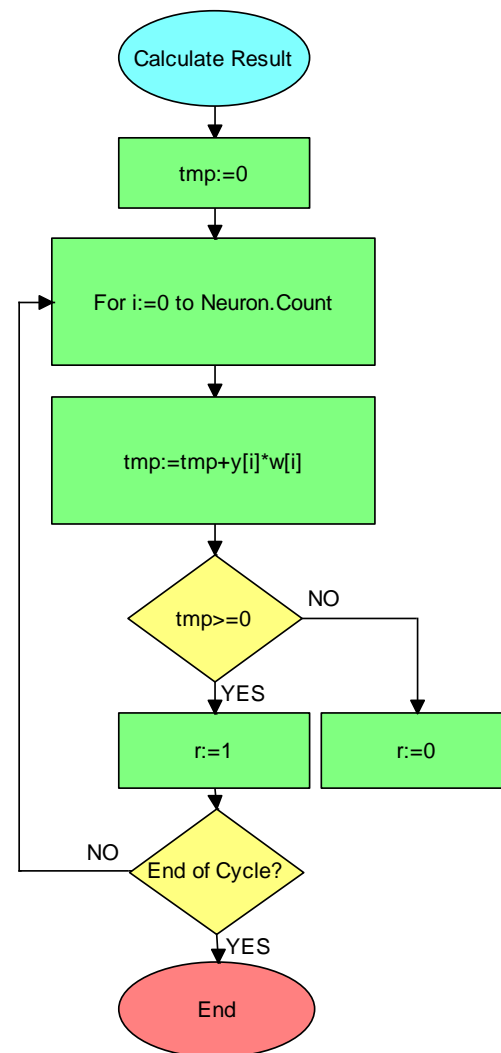
# Neironu tīkla veidošana un skaitīšana. Neironi.

- procedure  
NN\_Calculate\_Y();  
//Saskaitīt neironus  
(mainīgajā  $y[i]$ ). Rezultātu var  
izvadīt tabulā StringGrid.



# Neironu tīkla veidošana un skaitīšana. Rezultāts.

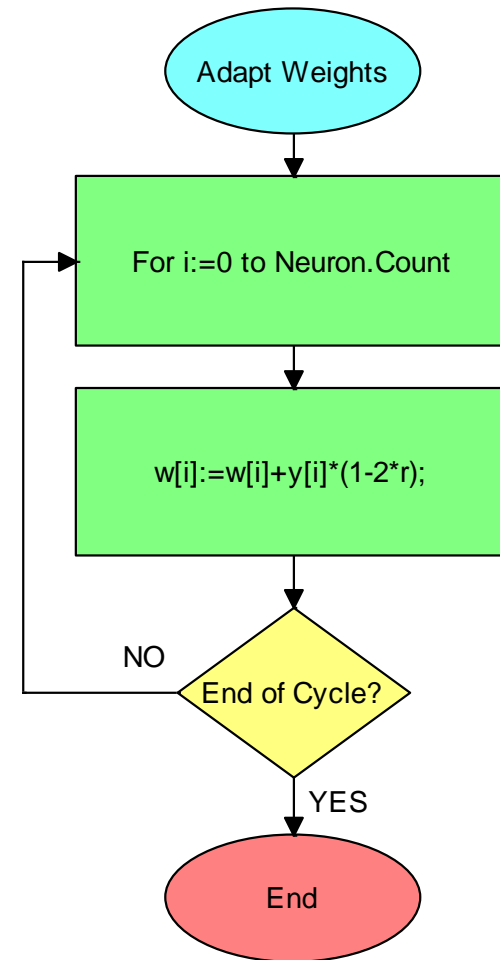
- procedure  
NN\_Calculate\_Result();  
//Skaitīt rezultātu (mainīgais  
r). Datus var izvadīt  
Edit.Text.





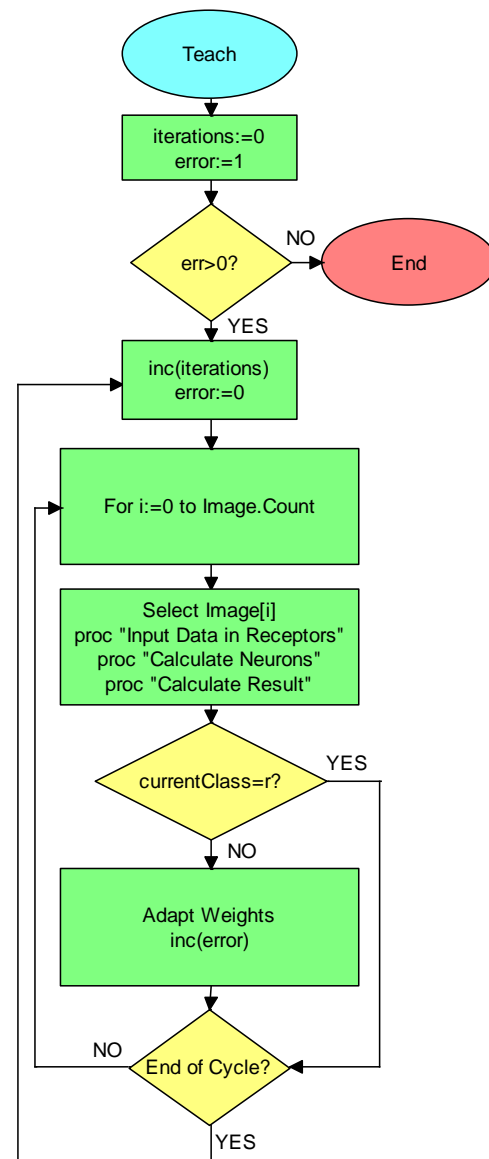
# Neironu tīkla apmācība. 1 solis.

- procedure  
NN\_Adapt\_W(); //Svaru  
adaptācija. Pēc būtības tas  
ir apmācības viens solis,  
jeb viena iterācija

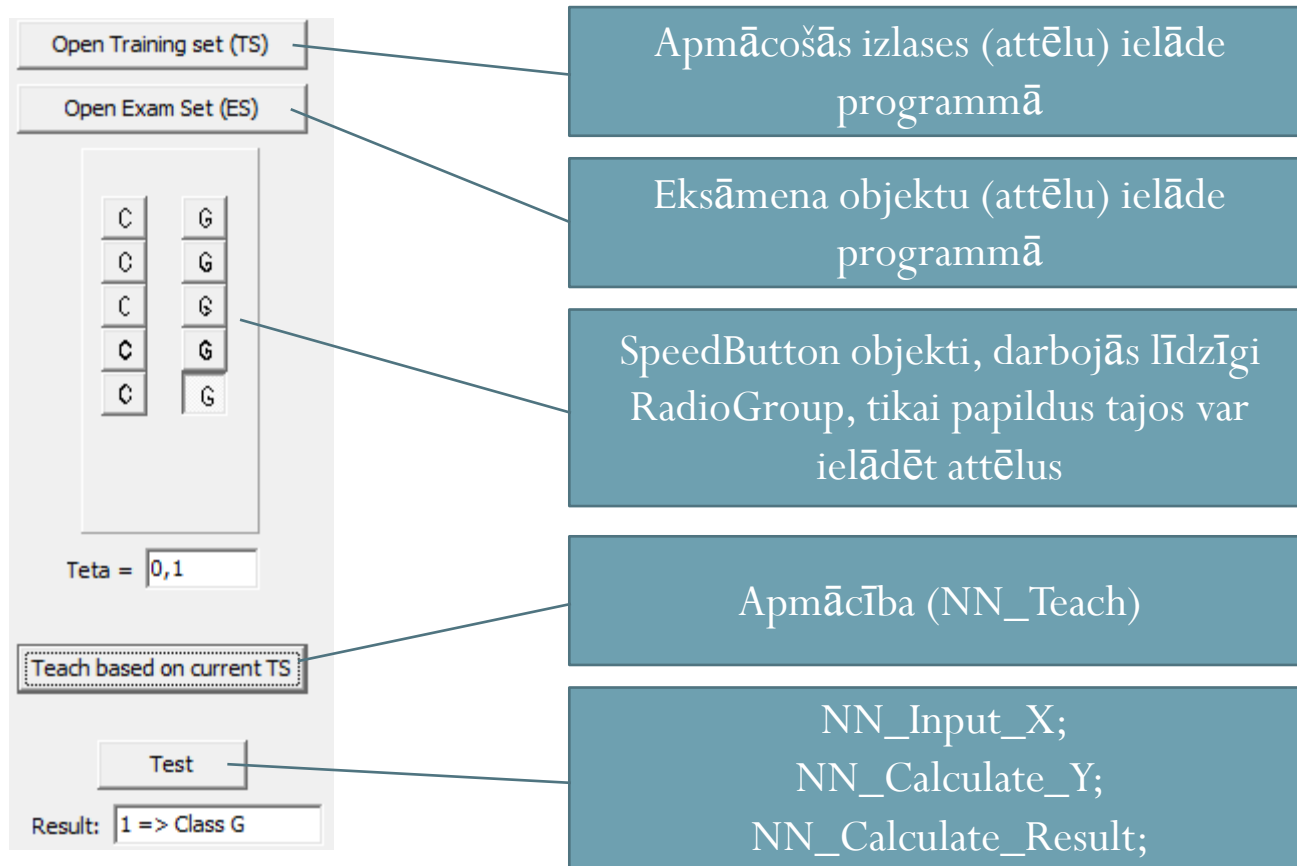


# Neirona tīkla apmācība. Apmācošās izlases apstrāde.

- procedure NN\_Teach();  
// Apmācība. Izieņ cauri visiem objektiem (attēliem) apmācošā izlasē un adaptē visus svarus, izmantojot NN\_Adapt\_W(), lai galu galā neironu tīkls spētu pareizi atpazīt apmācošu izlasi.



# Vēlreiz par lietotāju saskarni.



# Neironu tīkla uzlabošana

- **Strukturālā adaptācija** - strukturālās adaptācijas gadījumā tiek mainīta (nejauši, no jauna ģenerēta) matrica  $S[i,j]$  tā, lai pēc apmācības būtu pēc iespējas mazāk kļūdu.
  - Mainīt nejauši saites, izmantojot tādu pašu struktūru  $[2(+), 1(-)]$
  - Mainīt saišu struktūru  $[2(+), 1(-)]$  uz  $[2(+), 2(-)]$  vai...

# Neironu tīkla uzlabošana

- Komiteju metodes
  - Tiek apmācīti vairāki neironu tīkli (piemēram 5). Lēmums par to, kurai klasei pieder objekts tiek pieņemts, balstoties uz visiem 5 neironu tīkliem.
  - Komiteju metodes vislabāk strādā, kad tīkliem nepietika iterāciju apmācībai, un tie izdod kļūdas jau uz apmācošās izlases. Tad var iet runa par konkrēta neironu tīkla ticamību un to svaru balsošanā.

# Neironu tīkla uzlabošana

- Vienlīdzīgās balsošanas algoritms

$$R = \begin{cases} 1, & \text{ja } \sum_{i=1}^T R_i \geq \frac{T}{2} \\ 0, & \text{ja } \sum_{i=1}^T R_i < \frac{T}{2} \end{cases}$$

- Piemēram: 5 tīkli izdod jaunam objektam rezultātu 1 1 1 0 0, izmantojot formulu iegūsim:

$$1 + 1 + 1 + 0 + 0 \text{ un } \frac{5}{2} \Rightarrow 3 > 2.5 \Rightarrow R = 1$$

- Tātad sanāk, ka objekts pieder klasei 1.

# Neironu tīkla uzlabošana

- Svērtas balsošanas algoritms

$$R = \begin{cases} 1, & \text{ja } \sum_{i=1}^T z_i R_i \geq \frac{1}{2} \\ 0, & \text{ja } \sum_{i=1}^T z_i R_i < \frac{1}{2} \end{cases}$$

$$z_i = \frac{q_i}{\sum_{i=1}^T q_i}$$

Tīkla drošums

- Piemēram: 5 tīkli izdod rezultātu 11100, bet 1. tīkls apmācošā izlasē pareizi apmācīja 4 objektus no 10 ( $q_1=0.4$ ), bet 2. tīkls tikai 2 objektus no 10  $q_2=0.2$ , pārējiem kļūdu nav:  $q_3=q_4=q_5=1$ .  $\sum q_i=3,6$   
 $1 \cdot \frac{0.4}{3.6} + 1 \cdot \frac{0.2}{3.6} + 1 \cdot \frac{1}{3.6} + 0 \cdot \frac{1}{3.6} + 0 \cdot \frac{1}{3.6} = 0.111 + 0.056 + 0.278 = 0.445$   
 $0.445 < \frac{1}{2} \Rightarrow R = 0$
- Sanāk, ka objekts pieder klasei 0.

# Neironu tīkla uzlabošana

- **Ansamblā balsošanas algoritms**

$$R = \begin{cases} 1, & \text{ja } \frac{1}{T} \sum_{i=1}^T R_i \geq \frac{1}{2} \\ 0, & \text{ja } \frac{1}{T} \sum_{i=1}^T R_i < \frac{1}{2} \end{cases}$$

- Tas der tīkliem, kas izdod rezultātu R diapazonā [0..1]  
(perceptrons rezultātu nosaka izmantojot loģistisko funkciju)