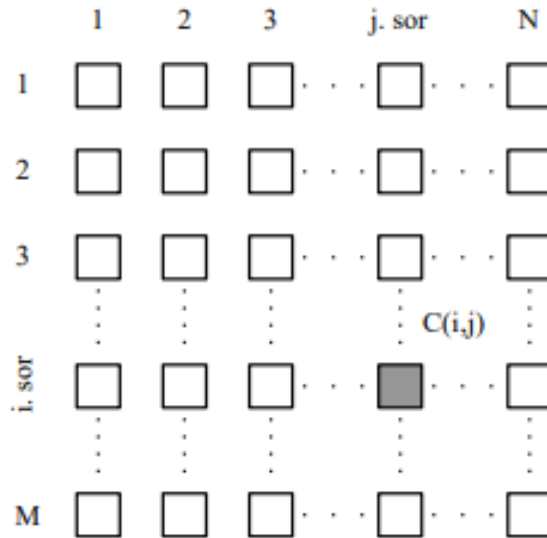


1. A Celluláris Neurális Hálózatok és implementációi

A Celluláris Neurális Hálózatok elméletét 1988-ban publikálta L. O. Chua és L. Yang [6], melyet azóta már számos feladat megoldására felhasználtak. Ilyen feladat típusok voltak a különböző tér-idő dinamikák számítása, vagy képfeldolgozási feladatok megoldása.

1.1.A CNN hálózatok elmélete

A CNN egy $M \times N$ -es téglalap alakú rács rácspontjaiban elhelyezkedő teljesen azonos, analóg módon működő $C(i, j)$ ($i = 1, 2, \dots, M$ és $j = 1, 2, \dots, N$) processzáló elemekből, úgynevezett cellákból épül fel, ahogy ez az 1.1. ábrán is látható [4].

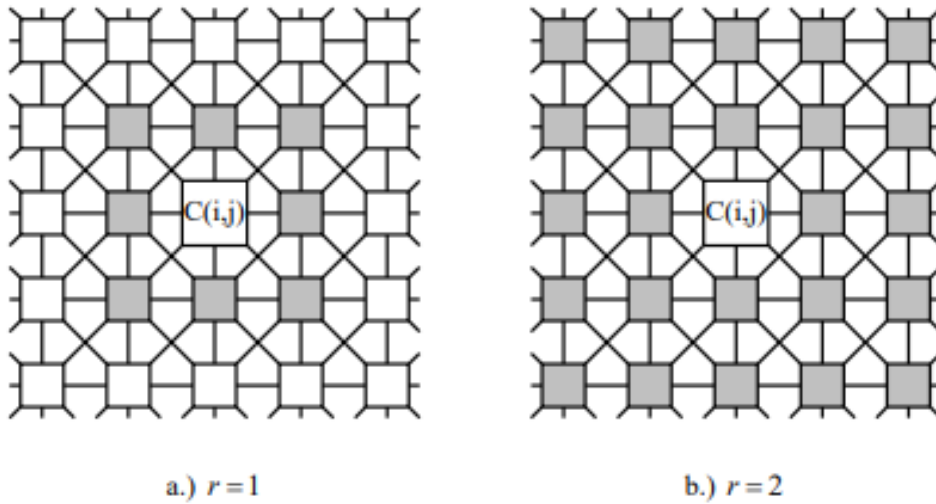


1.1. ábra. A Celluláris Neurális Hálózat felépítése

Egy $C(i, j)$ cella környezetét $S_r(i, j)$ -vel jelöljük, és azokat a cellákat tartalmazza, amelyekre teljesül az alábbi:

$$S_r(i, j) = \left\{ C(k, l) \mid \max_{1 \leq k \leq M, 1 \leq l \leq N} \{ |k - i|, |l - j| \} \leq r \right\}, \quad r, i, j, k, M, N \in \mathbb{Z}^+ \quad (1.1)$$

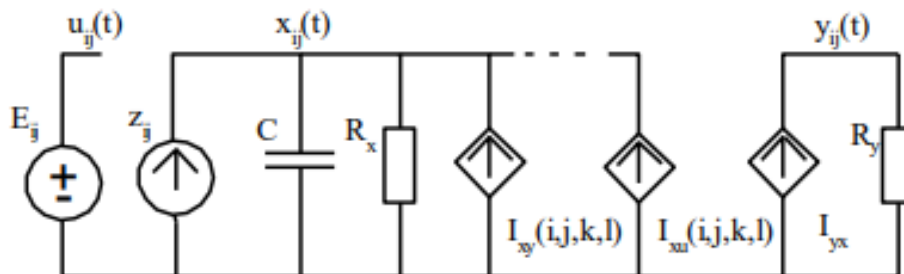
A $C(i, j)$ cella egy $S_r(i, j)$ környezetét gyakran a cella $(2r+1) \times (2r+1)$ -es környezetének nevezzük. Az egyes cellák csak a környezetükben lévő cellákkal tudnak kommunikálni, programozható összeköttetések, úgynevezett szinapszisok révén. A $C(i, j)$ cella 3×3 -as ($r = 1$) és 5×5 -ös ($r = 2$) környezete, és a cellák közötti összeköttetések az 1.2. ábrán láthatóak.



1.2. ábra. Egy $C(i, j)$ cella $r = 1$ és $r = 2$ sugarú környezete

A $C(i, j)$ cellát reguláris cellának nevezzük, ha a cella $S_r(i, j)$ környezetében minden $C(i, j) \in S_r(i, j)$ cella létezik. Ha ez a feltétel nem teljesül, akkor a $C(i, j)$ cellát határ cellának nevezzük [4].

Az 1.1. ábrán látható hálózat rácspontjaiban elhelyezkedő cellák egyszerű analóg áramköri elemekből épülnek fel. Minden egyes $C(i, j)$ cella rendelkezik egy állapotváltozóval, egy bemenettel és egy kimenettel. Az állapotváltozó értékében folytonos. A $C(i, j)$ cella a bemenetén keresztül figyeli a szomszédok bemeneteit és kimeneteit, melyek a saját állapotának megváltoztatásához szükségesek. A kimeneten keresztül pedig értesítheti szomszédjait a saját aktuális állapotáról. Egy ilyen analóg $C(i, j)$ cella kapcsolási rajza látható az 1.3. ábrán [6].



1.3. ábra. Egy analóg $C(i, j)$ cella felépítése

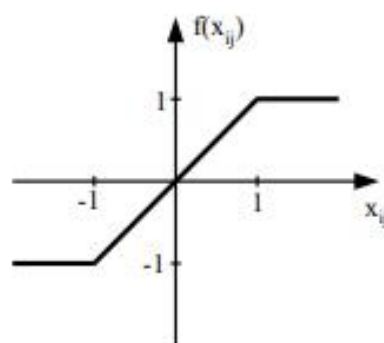
Az 1.3. ábra bal oldalán látható E_{ij} a cella bemenete, innen kapja a cella a külvilág felől érkező $u_{ij}(t)$ jeleket. A feszültségforrás mellett egy z_{ij} áramforrás is található, mely a

cellán belüli konstans eltolás vagy bias. A $C(i, j)$ cella $x_{ij}(t)$ állapotváltozója a C kondenzátor árama, melyet az áramforrás forrásárama, az R_x ellenálláson folyó áram és a feszültségvezérelt áramforrások forrásáramai ($I_{xu}(i, j, k, l)$ és $I_{xy}(i, j, k, l)$) határozzák meg. A vezérelt források áramát a szomszédos cellák bemenetén és kimenetén lévő, valamint a $C(i, j)$ cella saját be- és kimenetének megfelelően súlyozott értékei határozzák meg, az alábbi egyenletek alapján:

$$I_{xu} = \frac{I}{R_B(i, j, k, l)} u_{kl}, I_{xu}, u_{kl} \in \mathbb{R}, R_B \in \mathbb{R} \times \mathbb{R}, i, j, k, l \in \mathbb{Z}^+ \quad (1.2)$$

$$I_{xy} = \frac{I}{R_A(i, j, k, l)} y_{kl}, I_{xy}, y_{kl} \in \mathbb{R}, R_A \in \mathbb{R} \times \mathbb{R}, i, j, k, l \in \mathbb{Z}^+ \quad (1.3)$$

ahol az $R_B(i, j, k, l)$ és az $R_A(i, j, k, l)$ a $C(i, j)$ és a $C(k, l)$ cella között lévő összeköttetés súlya. Az 1.3. ábra jobb oldalán látható egységnyi ellenálláson eső $y_{ij}(t)$ feszültség a $C(i, j)$ cella kimenete, melyet a tőle balra eső nemlineáris, úgynevezett szigmoid karakterisztikával (1.4. ábra) rendelkező feszültségvezérelt áramgenerátor I_{xy} árama hoz létre az R_y ellenálláson.



1.4. ábra. A szigmoid karakterisztika

Ebből adódik, hogy egy $M \times N$ -es CNN hálózatot két egyenlettel a $C(i, j)$ cella ($i = 1, 2, \dots, M, j = 1, 2, \dots, N$) állapotegyenletével és egy kimeneti egyenlettel lehet leírni. Az állapotegyenlet Kirchhoff csomóponti törvénye alapján az (1.4) egyenletben megadott differenciálegyenlettel lehet leírni.

$$C \frac{dx_y(t)}{dt} = -\frac{1}{R_x} x_y(t) + \sum_{C(k,l) \in S_r(i,j)} A(i,j,k,l) y_k(t) + \sum_{C(k,l) \in S_r(i,j)} B(i,j,k,l) u_k(t) + z_y,$$

$$x_y(t), y_k(t), u_k(t), z_y, t, C, R_x \in \mathbb{R}, A, B \in \mathbb{R} \times \mathbb{R}, \text{ és } i, j, k, l \in \mathbb{Z}^+ \quad (1.4)$$

ahol az előbbieknél megfelelően az $x_y(t)$ az állapot, az $y_k(t)$ a kimenet, az $u_k(t)$ a bemenet, és a z_y pedig a bias érték. Az (1.4) egyenletben két mátrix is található, amelyek a cellák között lévő összeköttetések súlyát határozzák meg, azaz:

$$A(i,j,k,l) = \frac{1}{R_A(i,j,k,l)}, A, R_A \in \mathbb{R} \times \mathbb{R}, i, j, k, l \in \mathbb{Z}^+ \quad (1.5)$$

$$B(i,j,k,l) = \frac{1}{R_B(i,j,k,l)}, B, R_B \in \mathbb{R} \times \mathbb{R}, i, j, k, l \in \mathbb{Z}^+ \quad (1.6)$$

ahol az $A(i,j,k,l)$ mátrix az úgynevezett visszacsatoló, míg a $B(i,j,k,l)$ mátrix az úgynevezett előrecsatoló szinaptikus operátor vagy template [6].

A kimeneti egyenlet az (1.7) egyenletben megadott nemlineáris karakterisztika, mely lényegében az 1.4. ábrán látható szigmoid karakterisztikát írja le [6].

$$y_y = f(x_y) = \frac{1}{2} |x_y + 1| - \frac{1}{2} |x_y - 1|, y_y, x_y \in \mathbb{R}, i, j \in \mathbb{Z}^+ \quad (1.7)$$

A legtöbb esetben az 1.3. ábrán látható C kondenzátor és R ellenállás egységnyi nagyságú, így az (1.4) egyenlet felírható az alábbi alakban is:

$$\dot{x}_y = -x_y + \sum_{C(k,l) \in S_r(i,j)} A(i,j,k,l) y_k + \sum_{C(k,l) \in S_r(i,j)} B(i,j,k,l) u_k + z_y, \\ x_y, y_k, u_k, z_y \in \mathbb{R}, A, B \in \mathbb{R} \times \mathbb{R}, \text{ és } i, j, k, l \in \mathbb{Z}^+ \quad (1.8)$$

Az u_k bemenet értéke rendszerint egy $M \times N$ -es szürkeskálás kép adott pixelének intenzitása, a -1 és a +1 tartományra normalizálva, azaz $-1 \leq u_k \leq +1$, ahol a -1 jelenti a fehér pixel intenzitás, míg a +1 a fekete pixel intenzitás értéket. Álló képek esetében az u_k értéke független az időtől, míg mozgó képek, mint például videó esetében időfüggő. A többi változó, azaz az x_y , az y_k és a z_y is definiálhatók, mint kép pixel intenzitás értékek.

Az (1.4) egyenletből látszik, hogy minden egyes cella esetében az új állapotértéket a cella $S_r(i,j)$ környezetébe tartozó cellák be- és kimenetei alapján kell kiszámítani. A határcellák esetében azonban az $S_r(i,j)$ környezetnek van olyan része,

amely kívül esik a CNN $M \times N$ -es struktúráján. Ezekon a pontokon úgynevezett virtuális cellákat kell definiálnunk, amelyek u_M és y_M értékeit az úgynevezett határfeltételek alapján lehet meghatározni [4]. A határfeltételeknek alapvetően három típusát különböztetjük meg:

- A Dirichlet vagy Fix határfeltétel, mely esetében a virtuális cellák bemenete: $u_{k,0} = E_1$, $u_{k,N+1} = E_2$, $u_{0,l} = E_1$, $u_{M+1,l} = E_2$, $l = 0, \dots, N+1$ és $k = 1, \dots, M$ és $k, l, M, N \in \mathbb{Z}^+$, $u, E_1, E_2 \in \mathbb{R}$,
kimenete: $y_{k,0} = E_3$, $y_{k,N+1} = E_4$, $y_{0,k} = E_3$, $y_{M+1,k} = E_4$, $l = 0, \dots, N+1$ és $k = 1, \dots, M$ és $k, l, M, N \in \mathbb{Z}^+$, $y, E_3, E_4 \in \mathbb{R}$.
- A Neumann vagy Zero-flux határfeltétel, mely esetében a virtuális cellák bemenete: $u_{k,0} = u_{k,1}$, $u_{k,N+1} = u_{k,N}$, $u_{0,l} = u_{1,l}$, $u_{M+1,l} = u_{M,l}$ $l = 0, \dots, N+1$ és $k = 1, \dots, M$, és $k, l, M, N \in \mathbb{Z}^+$, $u \in \mathbb{R}$,
kimenete: $y_{k,0} = y_{k,1}$, $y_{k,N+1} = y_{k,N}$, $y_{0,l} = y_{1,l}$, $y_{M+1,l} = y_{M,l}$ $l = 0, \dots, N+1$ és $k = 1, \dots, M$, és $k, l, M, N \in \mathbb{Z}^+$, $y \in \mathbb{R}$.
- A periódikus vagy toroid határfeltétel, mely esetében a határcellák bemenete: $u_{k,l} = u_{k,N}$, $u_{1,l} = u_{M,l}$, $l = 1, \dots, N$ és $k = 1, \dots, M$, és $k, l, M, N \in \mathbb{Z}^+$, $u \in \mathbb{R}$,
kimenete: $y_{k,l} = y_{k,N}$, $y_{1,l} = y_{M,l}$, $l = 1, \dots, N$ és $k = 1, \dots, M$, és $k, l, M, N \in \mathbb{Z}^+$, $y \in \mathbb{R}$.

1.2.A CNN template-ek bemutatása

Az (1.8) egyenletben bemutatott CNN állapotegyenletet kiegészíthetjük két újabb template-el is. Az egyik az állapottól (C template), a másik pedig mindhárom változótól (D template) függ. Ebben az esetben a CNN az alábbi állapotegyenlettel írható le:

$$\begin{aligned} \dot{x}_y = & -x_y + \sum_{C(k,l) \in S_y(i,j)} A(i,j,k,l)y_M + \sum_{C(k,l) \in S_y(i,j)} B(i,j,k,l)u_M + \sum_{C(k,l) \in S_y(i,j)} C(i,j,k,l)x_M + \\ & + \sum_{C(k,l) \in S_y(i,j)} \hat{D}(i,j,k,l)(u_M, x_M, y_M) + z_y, \\ x_y(t), x_M(t), y_M(t), u_M(t), z_y \in \mathbb{R}, A, B, C, \hat{D} \in \mathbb{R} \times \mathbb{R}, \text{ és } i, j, k, l \in \mathbb{Z}^+ \end{aligned} \quad (1.9)$$

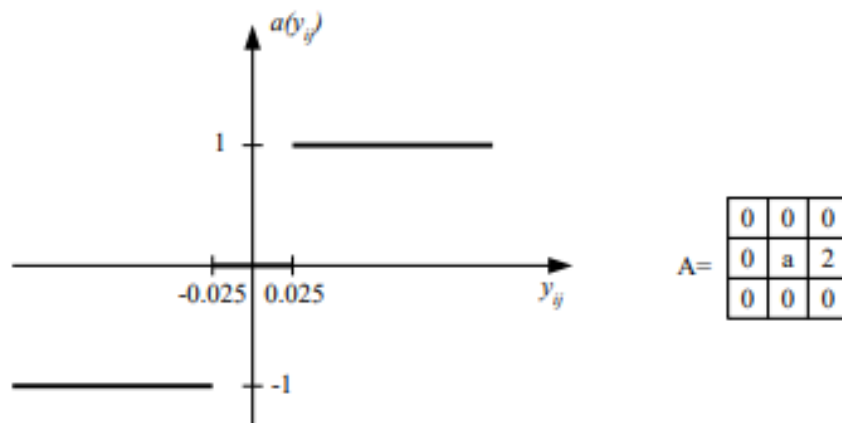
Az (1.9) egyenletben szereplő $A(i, j, k, l)$, a $B(i, j, k, l)$, a z_{ij} illetve a $C(i, j, k, l)$ és $D(i, j, k, l)$ template-ek jelentik a CNN struktúra programját. A template-eket az alábbi három szempont szerint csoportosíthatjuk:

- helyfüggő és helyfüggetlen template-ek,
- idő variáns és idő invariáns template-ek,
- lineáris és nemlineáris template-ek.

Helyfüggő template-ek esetében az egyes template értékek függenek az (i, j) pozíciótól, míg az idővariáns template-ek esetében az egyes template értékek időfüggők. A lineáris template-ek kizárólag konkrét számértékeket tartalmaznak. A nemlineáris template-ek az aktuális cella valamely változójának (bemenet, kimenet, állapot), vagy pedig az aktuális és a szomszédos cella valamely változóinak (pl. különbségének) nemlineáris függvényei. Tehát lineáris template-ek esetén az aktuális template értékkel súlyozni kell a hozzá tartozó cella állapotát, bemenetét, vagy kimenetét az (1.9) egyenlet alapján. A nemlineáris template-ek esetén azonban a nemlinearitás alapján meghatározott nemlineáris template érték már a hozzá tartozó cella állapotának, bemenetének, vagy kimenetének template-el súlyozott értékét jelenti [4].

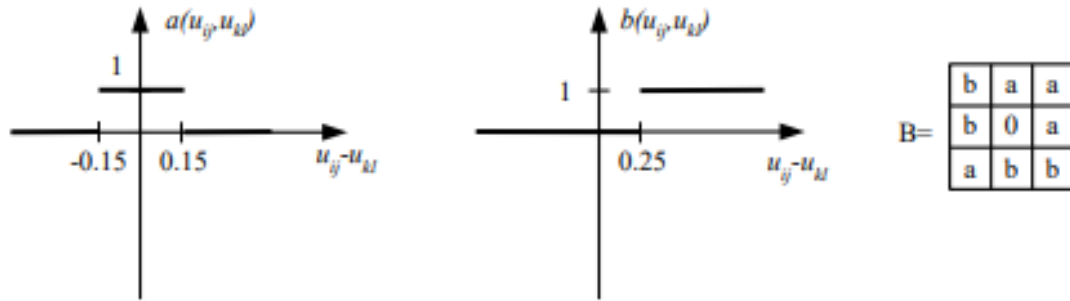
A nemlineáris template-ek alkalmazására számos példát találhatunk a CNN Template Library 3.1-es verziójában [4],[5]. Nézzünk most meg néhányat ezek közül.

1. Példa: A Majority Vote-Taker eljárás, mely segítségével eldönthető, hogy egy bináris bemeneti képen a kép soraiban található fekete és fehér pixelek száma között milyen reláció áll fenn. Az eljárás olyan $A(i, j, k, l)$ template-et alkalmaz, mely tartalmaz olyan értéket, amely az y_{ij} kimenet 1.5. ábrán látható nemlineáris függvényeként van meghatározva ($A(i, j, k, l) = a(y_{ij})$).



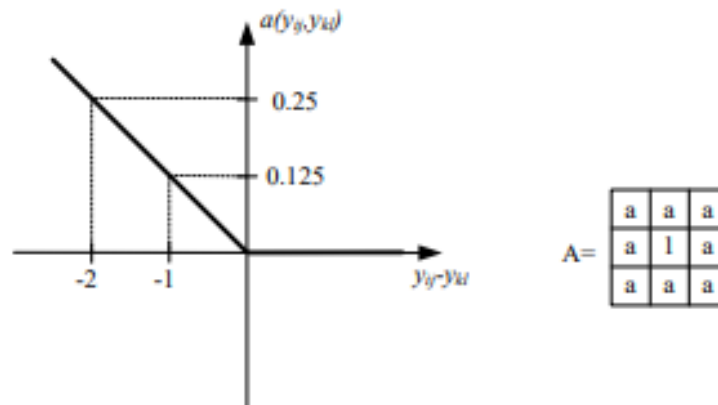
1.5. ábra. Majority Vote-Taker eljárásban használt nemlinearitás és template

2. **Példa:** A Grayscale Line Detector eljárás, mely egy bináris bemeneti képen meghatározza az egymással 30° -os szöget bezáró egyeneseket. Ennél az eljárásnál az $A(i, j, k, l)$ template megfelelő értékei az aktuális cella és a szomszédos cella kimenetének (y_{ij}, y_{kl}) különbségének nemlineáris függvénye által definiáltak ($A(i, j, k, l) = a(y_{ij}, y_{kl})$). Továbbá a $B(i, j, k, l)$ template értékei az aktuális cella és a szomszédos cella bemenetének (u_{ij}, u_{kl}) különbségének nemlineáris függvénye által meghatározottak ($B(i, j, k, l) = b(u_{ij}, u_{kl})$). Az eljáráshoz tartozó nemlinearitások és a template az 1.6. ábrán láthatóak.



1.6. ábra. A Grayscale Line Detector eljárásban használt nemlinearitások és template

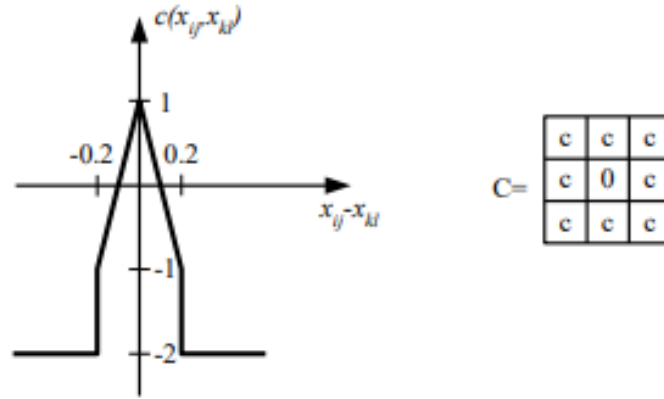
3. **Példa:** A Global Maximum Finder eljárás, mely meghatározza a bemeneti szürkeskálás képen a maximális intenzitás értéket. Ennél az eljárásnál az $A(i, j, k, l)$ template bizonyos értékei az aktuális cella és a szomszédos cella kimenetének (y_{ij}, y_{kl}) különbségének nemlineáris függvényei ($A(i, j, k, l) = a(y_{ij}, y_{kl})$), amely nemlinearitás és a hozzá tartozó template az 1.7. ábrán láthatóak.



1.7. ábra. A Global Maximum Finder eljáráshoz tartozó nemlinearitás és template

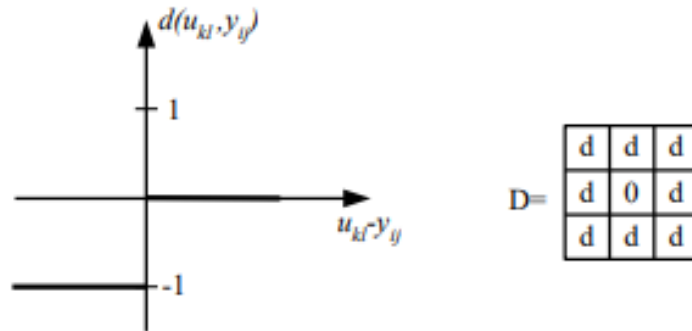
4. **Példa:** Az Extreme eljárás, mely a bemeneti szürkeskálás képen meghatározza egy adott küszöbértéknél kisebb gradiens értékek pozícióját. Ennél az eljárásnál

alkalmazott $C(i, j, k, l)$ template tartalmaz olyan értékeket, amelyek az aktuális cella, és a szomszédos cella állapotának (x_{ij}, x_{kl}) különbségének nemlineáris függvényei ($C(i, j, k, l) = c(x_{ij}, x_{kl})$). A nemlinearitás és a hozzá tartozó template az 1.8. ábrán láthatók.



1.8. ábra. Az Extreme eljárásához tartozó nemlinearitás és template

5. **Példa:** A Grayscale Erosion eljárás, mely a bemeneti szürkeskálás képen eróziót hajt végre. Ennél az eljárásnál alkalmazott $D(i, j, k, l)$ template megfelelő értékei a szomszédos cella bemenetének és az aktuális cella kimenetének (u_{kl}, y_{ij}) különbségének 1.9. ábrán látható nemlineáris függvénye ($D(i, j, k, l) = d(u_{kl}, y_{ij})$).



1.9. ábra. Grayscale Erosion eljárásához tartozó nemlinearitás és template

1.3.A CNN különböző megvalósításai

A CNN-nek alapvetően négy különböző implementációja van. Az első az analóg/kevert jelű VLSI megvalósítás (Ace400, Ace4K, Ace16k, Xenon, Eye-Ris), amelynek előnye a nagy számítási teljesítmény (akár 30 TerraOP/mp), azonban csak kis számítási pontossággal (7-8 bit) rendelkezik, valamint érzékeny a feszültség és hőmérsékletváltozásra, továbbá csak lineáris template-ek használhatók. A második

Neighbourhood of cells

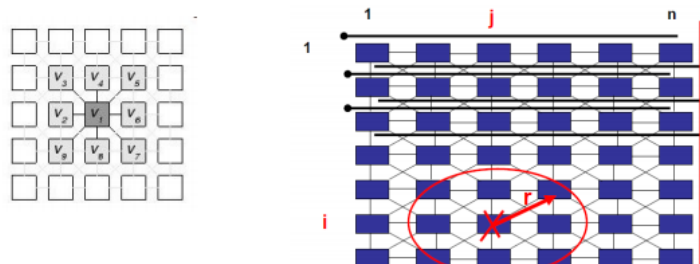
Neighbourhood is given by a set: $Sr(i,j)$

$Sr(i,j)$ Sphere of influence of cell $C(i,j)$

Neighbourhood radius: r

Those cells are interacting indirectly where:

$$Sr(i,j) = \{C(k,l) \mid \max 1 \leq k \leq M, 1 \leq l \leq N \mid |k-i|, |l-j| \leq r\}$$



Boundary conditions:

1. Fixed (Dirichlet) boundary conditions: Virtual External cells with constants values
2. Zero flux (Neumann) This boundary condition usually applies to the case where there is no input The virtual cells have the value of the closest real cell
3. Periodic (Toroidal) We can have an infinit, periodic grid

Some mathematics

Description of two-dimensional binary cellular automata:

$2^9=512$ rows in a binary table

512 possible states

2^{512} lehetséges kitöltési mód. -> ennyi fele szabály lehet

$C(-1,-1)$	$C(0,-1)$	$C(1,-1)$
$C(-1,0)$	$C(0,0)$	$C(1,0)$
$C(-1,1)$	$C(0,1)$	$C(1,1)$

$C(-1,-1)$	$C(-1,0)$	$C(-1,1)$	$C(0,-1)$	$C(0,0)$	$C(0,1)$	$C(1,-1)$	$C(1,0)$	$C(1,1)$	B
0	0	0	0	0	0	0	0	0	B0
0	0	0	0	0	0	0	0	1	B1
...	...	...	...	...	...	...	...	...	...
1	1	1	1	1	1	1	1	0	B510
1	1	1	1	1	1	1	1	1	B511



One-dimensional binary CA

$2^8=256$ rules

Description of CA rules



Rules can be organized by binary numbers as they are written into binary tables

For every rule we will have a unique code

We have altogether $2^8=256$ rules

C(-1)	C(0)	C(1)	240	15	204	110
0	0	0	0	1	0	0
0	0	1	0	1	0	1
0	1	0	0	1	1	1
0	1	1	0	1	1	1
1	0	0	1	0	0	0
1	0	1	1	0	0	1
1	1	0	1	0	1	1
1	1	1	1	0	1	0

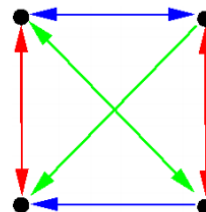
- Universal rules (Turing complete R110)
- Chaotic rules (Rule 86,135,149,30)
- Pseudo-random number generators (rule 30- Mathematica)

Do we really have 256 rules?

Are all rules independent?

Can we apply rules on the rules?

Vierergruppe transformation



Complementation (235-40)

Left-Right transformation (235-249)

Left-Right Complementation(235-96)

Vierergruppe transformation (Chua 2004)

88 independent rules

4 rules in every group: $4(8 - R_0)$

CAs with input

New complex behaviors due to input

What is the connection between the input and the state variables? (Or is the input just another state variable?)

Usually a logical function (pl.: AND, XOR)

Constant inputs, Periodic inputs, Arbitrary inputs

Differences between CAs and CNNs

CNN: both input and state are present, CA: just state, no input

CNN: continuous states CA: discrete states

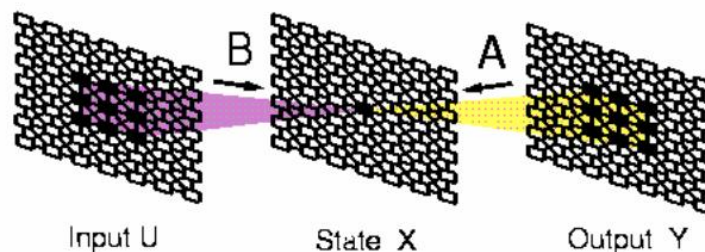
CNN state equation



1 State equation

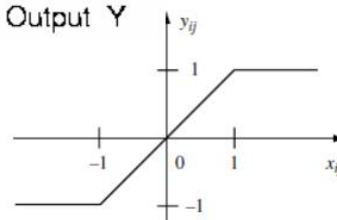
$$\dot{x}_{ij} = -x_{ij} + \sum_{C(k,l) \in S_r(i,j)} A(i,j;k,l)y_{kl} + \sum_{C(k,l) \in S_r(i,j)} B(i,j;k,l)u_{kl} + z_{ij}$$

$$\mathcal{C}(A, B, z)$$



2 Output equation

$$y_{ij} = f(x_{ij}) = \frac{1}{2}|x_{ij} + 1| - \frac{1}{2}|x_{ij} - 1|$$



Space variant template: coupling depends on the cell position, Can be used well in certain well-defined applications

Space invariant template: The same A and B coupling matrices/templates are used for every cell

Template types:

- Excitatory: the elements in the A template are positive
- Inhibitory: the elements in the A template are negative
- Zero-Feedback: The elements in the A template are zeros
- Zero-input, autonomous: the elements in the B template are zeros
- Uncoupled: Only the middle element of the A template is not zero

Circuit implementation



Necessary independent elements:

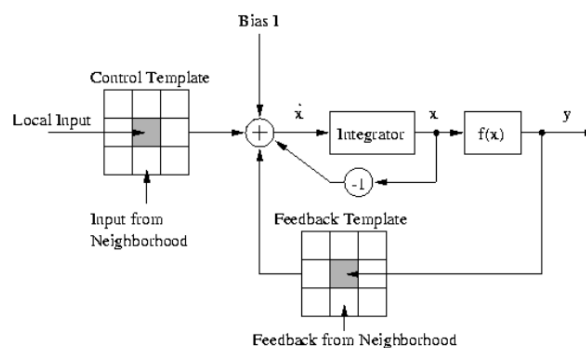
- Internal voltage source

- Bias

- Feed forward loop

- Feedback loop

- Output



Connection to convolutional networks

Both networks implement convolution+non/linearity

Good for pattern matching

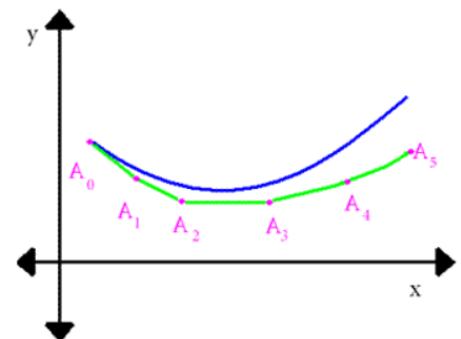
The feed forward template implements convolution on the input.

The cells are globally connected thanks to feedback in one layer.

Euler method (or Runge Kutta):

We have to approximate a function at a given time by its derivative.

We have to define a step interval for this calculation (how frequently we have to calculate the derivative).



(Nyquist-Shanon theorem: Every bounded (continuous), periodic function $x(t)$ can be restored perfectly from its observations, if the sampling frequency is at least two times higher, than the highest frequency in the signal.)

Scalability: The possibility of speeding up computation:

- Discrete system: higher frequencies
- Continuous systems: higher voltage source

Synchronization

Speed of convergence: We can optimize the propagation time of our operations by choosing the proper template

Bizonyítsa be, hogy a CNN dinamika korlátos!

The bounds of CNN dynamics



Upper and lower bounds:

$$|x_{ij}(0)| \leq 1, \quad |u_{ij}(t)| \leq 1, \quad |z_{ij}(t)| \leq z_{\max}$$

$$|x_{ij}(t)| \leq x_{\max} \quad \forall t \geq 0, \quad 1 \leq i \leq M, \quad 1 \leq j \leq N$$

$$x_{\max} = 1 + z_{\max} + \max_{\substack{1 \leq i \leq M \\ 1 \leq j \leq N}} \left[\sum_{C(k,l) \in S_r(i,j)} (|A(i, j; k, l)| + |B(i, j; k, l)|) \right]$$

A lemma for CNN stability



If A is centrally symmetric than the feedback template for the whole network $\hat{A}$ is symmetric

CNN stability



A Ljapunov function for CNN dynamics:

$$V(\mathbf{x}) = -\frac{1}{2} \mathbf{y}^T \hat{A} \mathbf{y} + \sum_{i=1}^n \left[\int_{\theta}^{y_i} f^{-1}(v) dv \right] - \mathbf{y}^T \hat{B} \mathbf{u} - \mathbf{y}^T \mathbf{z}$$

Let's prove that this is a Ljapunov function

If A is centrally symmetric, than the dynamics are asymptotically stable

It is enough if A is centrally sign symmetric. We will not prove this during the course.

(BIZ) If A is centrally symmetric than the feedback template for the whole network ($A^\wedge$) is symmetric

- sorfolytonosan (és az egyszerűség kedvéért a boundary condition-ról most feledkezzünk meg)

-tegyük fel, hogy van egy $N \times M$ -es mátrixunk,

-akkor a teljes rendszert egy $(N \times M) \times (N \times M)$ es nagy mátrix írja le, ami megadja, hogy melyik elem melyikkel van összekötve (ebben nagy sok elem nulla lesz...)

-igazából minden sorban csak 9 elem lesz nem nulla, mert minden elem csak 9 másikkal van összekötve -önmagát is beleszámolva-

-Ha a template centrálisan szimmetrikus, akkor így néz ki:

c d e

b a b

e d c

a nagy $A^\wedge$ mátrixban:

- a főátlóban lesznek a értékek, mert minden elem önmagával van összekötve.

- és minden elem össze van kötve az eggyel korábbival és későbbivel b értékkel (mert sorfolytonos a leírás és a szomszédok pont egyel korábban/később vannak a sorban) tehát bab van mindenhol a főátlónál.

-Ha N elemből állt egy sor, akkor a felette lévő elem a nagy $A^\wedge$ mátrixban pont N-nel van korábban, az alsó N-el később. ezekre d az összeköttetés a szomszédokra pedig c és d.

Complete Stability Theorem 1

Any $M \times N$ space-invariant CNN of arbitrary neighborhood size with *constant inputs and constant threshold* is *completely stable* if the following three hypotheses are satisfied:

1. The A template is symmetric:

$$A(i,j;k,l) = A(k,l;i,j) \quad (8.8)$$

2. The nonlinear function $y_i = f(x_i)$ is *differentiable, bounded*, and

$$f'(x_{i0}) > 0, \text{ for all } -\infty < x_{i0} < \infty \quad (8.9)$$

3. All equilibrium points are *isolated*.⁴

Proof. Consider the CNN state equation (2.8) from Chapter 2 for *constant* input $\mathbf{u}$ and threshold $\mathbf{z}$:

$$\dot{\mathbf{x}} = -\mathbf{x} + \hat{\mathbf{A}} \mathbf{y} + \hat{\mathbf{B}} \mathbf{u} + \mathbf{z} \quad (8.10)$$

$$y_i = f(x_i), \quad i = 1, 2, \dots, n = MN \quad (8.11)$$

Here, $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ are $n \times n$ matrices whose nonzero entries are the synaptic weights $A(i,j;k,l)$ and $B(i,j;k,l)$, respectively. Observe that hypothesis (8) and space invariance imply that

$$\hat{\mathbf{A}} = \hat{\mathbf{A}}^T \quad (8.12)$$

independent of the packing scheme.

Now, hypothesis (9) implies that $f(\bullet)$ is a one-to-one (injective) function and therefore has an *inverse* function

$$x_i = f^{-1}(y_i) \quad (8.13)$$

defined for all y_i over the range of $f(x_i)$, $x_i \in (-\infty, \infty)$. Define the *scalar function*

$$V(\mathbf{x}) = -\frac{1}{2} \mathbf{y}^T \hat{\mathbf{A}} \mathbf{y} + \sum_{i=1}^n \left[\int_{\theta}^{y_i} f^{-1}(v) dv \right] - \mathbf{y}^T \hat{\mathbf{B}} \mathbf{u} - \mathbf{y}^T \mathbf{z} \quad (8.14)$$

where θ is any number such that $f(-\infty) < \theta < f(\infty)$.⁵

A scalar function $V(\mathbf{x})$ is called a *Lyapunov function* if its time derivative along any trajectory is non-positive, i.e.,

⁴ An equilibrium point $\mathbf{x}_0$ of $\dot{\mathbf{x}} = f(\mathbf{x})$ is said to be *isolated* if, and only if, there are no other equilibrium points in a sufficiently small neighborhood of $\mathbf{x}_0$.

⁵ In the nonlinearity f we have been using, we can choose $\theta=0$, since $f(-\infty) = -1$, $f(\infty) = 1$. For the sake of generality, the hypothesis on f does not require that the values of f lie between -1 and 1 .