

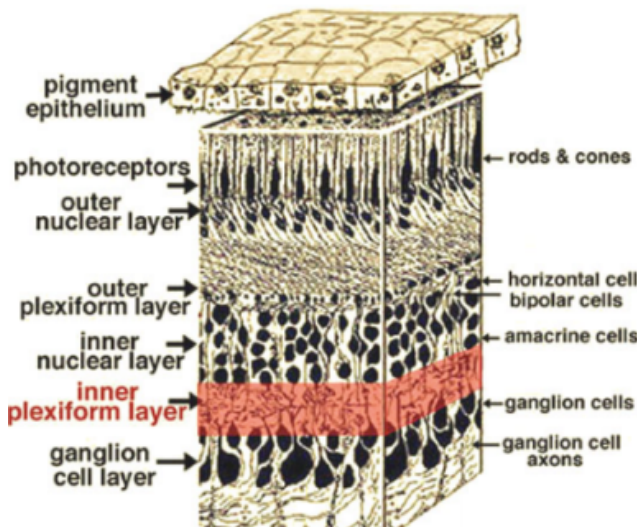
CNN kidolgozott tételek

1. Retina modellek, az emberi látás, és a CNN strukturális, és funkcionális kapcsolata

Motiváció: emberi látórendszer

- Néhány területen jobban teljesítünk, mint a gépek
- Nagy mennyiségű, strukturálatlan adat esetén
- Részben tanult képesség, összetett rendszer
- pl. kislány, olló, tűz

Retina felépítése

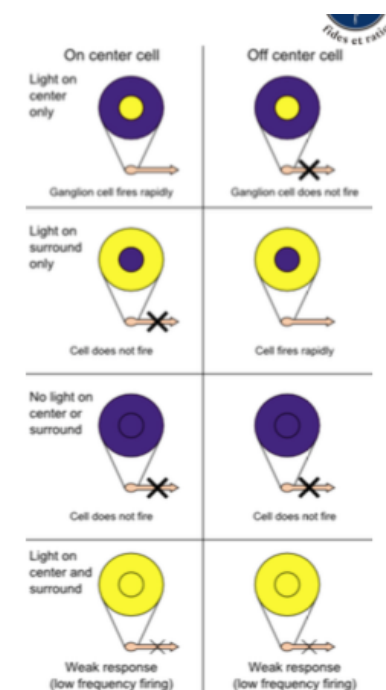


- Gerinces retina: Inverz retina= érzékelő sejtek a leghátó rétegben
- Csapok: 6 millió, jó fényviszonyok mellett
- Foveában, színlátásért felel
- Pácikák: 120 millió, rossz fényviszonyok mellett
- környéki látás, egy szín
- Nem 1-1 neuron kapcsolat, bonyolult rendszer
- Ganglion, amakrin, bipoláris, horizontális sejtek együttműködése
- 100x több fotoreceptor, mint ganglion sejt
- ezért nagymértékű tömörítésre van szükség
- Nem pixel szinten látunk, hanem területeket, változásokat érzékelünk
- különböző tulajdonságok alapján csoportosítunk
- hasonló területeket hajlamosak vagyunk összemérni

—> érzéki csalódások (face- vase illusion: öf.régiók összekapcsolása)

- OFF: fény hiányában tüzel
- ON: fény hatására tüzel
- Serkentő, gátló régiók
- Topografikus

- Roppant egyszerű struktúrák összekapcsolása



Folyamatok a látásban

Szakkádok:

- Gyors, szimultán szemmozgás, mely mindkét szemnél egyszerre történik, a homloklebeny irányításával
- Nem tudunk statikusan nézni a szemünk folyamatosan mozog
- —> biztosítja, hogy a fotoreceptorok folyamatos stimuláció alatt állnak
- Statikus képeket nem tudunk érzékelni
- vestibulo ocular reflex: a fej- test mozgását kiküszöböli a szemmozgás
- jobb, mint mozgó objektum követése, van visszacsatolás
 - bagoly

Retina modellezhető

- irányszelektivitás
- Az LGN-ben és a v1 ben léteznek olyan sejtek, amik egy megadott irányú és nagyságú egyesre adják a legnagyobb választ
- CNN templatek elve
 - pl. 3nál nem hosszabb detektáló CNN template
- Analóg számítás, állapot összeköttetés
 - analogikai számítógép: analóg és logikai számítások egyben

- analóg bemenet, digitális kimenet
 - A CNN architektúra képes a különböző térbeli struktúrák, tulajdonságok detektálására egyszerű és elegáns utasítások (egyetlen template) által
 - különösen jól teljesít képfeldolgozási feladatokban-> bionikus szemüveg
-

Platformok

pl:

- BI-I v2
 - 128x128-as képek feldolgozása 10000 FPS-sel
- Eyeris
- Q-eye

2. Egydimenziós celluláris automata, szabályok csoportosítása

Cellularitás – a rész és az egész

Kiemelkedés (Emergence): komplex mintázatok megjelenése 'viszonylag' egyszerű interakcióknak köszönhetően

Hangyák - hangyavár, városok

Mintagenerálás: érdekes, globális struktúrák kialakulása

Celluláris automata

A "játéktér":

- N-dimenziós végtelen rács (a gyakorlatban néha véges)
- Minden cellának diszkrét, előre meghatározott állapota lehet (általában bináris)
 - Minden cella csak a szomszédos cellákkal kommunikál
 - Az állapotok szinkronizált (minden mező egyszerre frissül), memóriamentes frissítése

Mintha egy négyzetrácsos lapon játszanánk megadott szabályok szerint

CA a gyakorlatban

- Sejtek
 - Celluláris struktúra és minta generálás
 - Tumor komplexitás - Közlekedési dugók - Erdőtűz
 - Populációdinamika - Zene, komponálás (Xenakis)
-

A CAk története

A formális logika százada:

- Boole (1845) - logika
- Gödel (1930) – számozás, rendszerek Turing (1936) – Turing teljesség
- Stanislaw Ulam – kristályrácsok szerkezete
- Neumann János – Önmásoló rendszerek
 - Létezik-e önmagát építő robot, automata (self replicating systems)?
 - 200000 darab 29 állapotú sejt, ami végtelen másolatot készít önmagából
 - Tárolt program Von Neumann – mesterséges élet
- 1940 Neumann Ulam - complexity Conway(1970): 'Game of Life'
- 2000 S. Wolfram/Chua nonlinear dynamics

Celluláris számítás:

- Neumann Celluláris Automata
 - Bináris/kétállapotú (digitális sejtek)
 - Élet/mesterséges
 - NP teljes problémák
-

CA- Majority Voting

- Egy egyszerű szabályrendszer:

- Két állapotú sejtek, mindenki lemásolja a többséget (8 szomszédjából, azt amelyikből több van, egyenlőség esetén megmarad a saját színe)
- Populációs dinamika, vírusok járványok, nyelvek, mém-ek

1D bináris CA

Egyszerű struktúra:

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

$\wedge 8$

$2^8 = 512$ szabály

- A szabályok táblázatba foglalhatóak, leírhatóak kettes számrendszerbeli számokként
- Mindegy szabályhoz létezik egy kód

Gyakorlati hasznuk

- Univerzális szabályok (Turing teljes R110)
- Kaotikus jelenségek (Rule 86, 135, 149, 30)
- Nagyon jó pszuedo-random generátor (rule 30- Mathematica)

Szabályok

- Nem függetlenek
- Komplementer (240-15)
- Bal-jobb ekvivalencia (240-170)
- Bal-jobb komplement (240-85)
- Vierergruppe transformation (Chua 2004)

- 88 független szabály

Csoportosítás:

- Komplex

Wolfram:	Chua:
- Homogén	• Period1
- Stabil vagy periódikus - Kaotikus	• Period2
- Komplex	• Period3
	• Bernoulli
	• Complex bernoulli (bilateral)
	• Hyper bernoulli (Non bilateral)

CA bemenettel

- Új, összetett viselkedési formák
- Bemenet és állapot logikai függvénye (pl.: VAGY)
- Konstans input
- Periódikus input
- Tetszőleges input

Különbségek a CNN és a CA között

CNN- en van input; itt csak state CNN állapota folytonos; itt diszkrét

3. Kétdimenziós celluláris automata, Game Of Life

CA: 2. tételből

2Ds bináris CAk leírás

9 elemtől függ			1 kimenet
C(-1,-1)	C(0,-1)	C(1,-1)	B
C(-1,0)	C(0,0)	C(1,0)	
C(-1,1)	C(0,1)	C(1,1)	

Game of life

- Conway (1937): Elv - élet kialakulás, megmaradás (1970): 2-nél kevesebb szomszéd → meghal
- 3-nál több szomszéd → meghal
- 2 vagy 3 szomszéd → életben marad
- 3 szomszéd esetén élet keletkezik
- Turing teljes architektúra:
- Minden algoritmus megvalósítható ezen architektúrán
- Minden algoritmikusan megoldható probléma megoldható a Game of Life-val

4. CNN dinamika, mint elemi utasítás

A standard CNN architektúra cellák NxM-es reguláris hálózata

$C(i,j)$ $(i,j) \rightarrow i = 1, 2, \dots, N$ $j = 1, 2, \dots, M$

M és N nem feltétlenül egyenlőek

CNN esetén egyetlen szabály: standard CNN dinamika

$x' = -x + Ay + Bu + z$

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

> kell egy nemlinearitás, amely a kimenetet határozza meg: $y_{ij} = f(x_{ij}) = 0.5 * (x_{ij} + 1) - 0.5 * (x_{ij} - 1)$

Dinamika megoldása

- meg kell adni:
 - dinamika $\rightarrow$ értelmezési tartomány, értékkészlet
 - $(-1, 1)$
 - hamis- fehér fekete-igaz
 - bemenet $\rightarrow$ konstans/ periodikus/ időben változó
 - boundary condition (peremfeltétel)
 - csatolás
 - kezdeti érték

$\rightarrow$ szimulációhoz: futási idő, nem laikus megoldás

- állapot: folytonos értékek, elméletben nem korlátos intervallum, gyakorlatban korlátos

Boundary condition

1. Fixed (Dirichlet) boundary conditions:

Külső virtuális cellákba konstans értéket töltünk

2, Zero flux (Neumann)

A virtuális cellák a 'legközelebbi' valós cella értékét veszik fel

3, Periodic (Toroidal)

Áthuzalozással végtelen, periódikus képet hoznak létre.

Cellák szomszédsága

szomszédság, szomszédos cellák: $S_r(i,j)$

$S_r(i,j)$ Sphere of influence of cell $C(i,j)$ Szomszédsági sugár: r

Azon cellák, melyekre: $S_r(i,j) = \{C(k,l) | \max(1 \leq k \leq M, 1 \leq l \leq N) \{ |k-i|, |l-j| \} \leq r \}$

Template típusok

- serkentő $\rightarrow$ minden $A +$
- gátló $\rightarrow$ minden $A -$

- zero feedback—> minden $A=0$
- zero input—> minden $B=0$
- uncoupled—> csak az A középső eleme nem 0

5. CNN-UM, algoritmusok megvalósítása CNN-UM-en

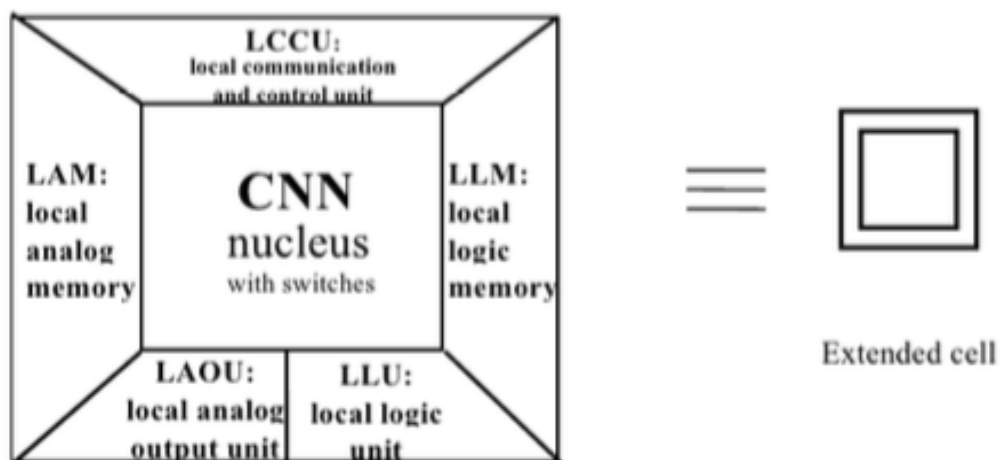
CNN univerzális gép:

- Algoritmus tervezéséhez (Turing teljes), feladatok megvalósításához (milyen feladat megoldására hatékony)
- Nem minden algoritmus valósítható meg egyetlen template-tel Minden algoritmus megvalósítható template-ek sorozataként.
- Egyéb kiegészítő infrastruktúrára van szükség az algoritmusok megvalósításához
- El kell tárolnunk az egyes utasítások eredményeit és később azokat vissza kell tudnunk tölteni és kombinálni más eredményekkel
- Memóriára van szükség:
 - lokális memória, elég a köztes eredmények tárolásához
 - globális memória: Az utasítás sorozat leírásához, minden cellához eljut

Extended CNN cell in the CNN-UM

Folytonos időben, folytonos állapotváltozóval, az ismert nemlineáris dinamika

+Lokális memória



LAOU

Local Analog Output Unit: analóg memória

Bemenetet, kimenetet kezeli a szűrkeskálás bemenetekre, kimenetekre, állapotokra

Kombinálja (összeadás, kivonás) a lokálisan tárolt adatokat egy kimeneti vagy bemeneti képpé

LLU

Local Logic Unit

Bemenetet, kimenetet kezeli a bináris (fekete/fehér) bemenetekre, kimenetekre, állapotokra

Kombinálja (logikai műveletek: AND, OR...) a lokálisan tárolt adatokat egy kimeneti vagy bemeneti képpé

LCCU

Local Communication and Control Unit.

Communication Unit: Fogadja az utasításokat a Global Analog Programming Unit (GAPU)-tól és beállítja a szükséges paramétereket

GAPU

Négy további részből áll:

- Analóg program regiszter
- Logikai program regiszter
- Switch configuration regiszter - Global analog control unit

Globálisan kezeli az utasításokat, ezek elindítását, ütemezését (nincs közös órajel), elküldi az utasítások paramétereit és fogadja/feldolgozza a kimeneteket.

Az Alpha nyelv

- Standard nyelv, ami leírja az algoritmikus lépéseket

Flow-chart, Diagram

- Könnyebb megérteni és átlátni az algoritmikus lépéseket Komplex eljárásoknál nagy helyet igényel, nehéz átlátni
- Kevésbé tömör leírás
- Objektum számlálás
- whole pill or an image

Szimulátor

- MatCNN
- Mit kell megadnunk:

1. Kezdeti érték

2. Bemenet
3. Boundary condition
4. Futási idő
5. Numerikus megoldás

Szimulátor inicializálása: `cnn_setenv;`

Kép betöltése: `INPUT1 = lbmp2cnn('piclib/BABOON.bmp');`

Bemenet betöltése: `mCNN.INPUT1 = INPUT1;`

Kezdeti állapot betöltése: `mCNN.STATE = INPUT1;`

Peremfeltételek megadása: `mCNN.Boundary = 2`

-1 és 1 között constans (az adott színnek megfelelő)

2 – zero flux

3 — periódikus

6. CNN dinamika kvalitatív elmélete: korlátosság, megoldhatóság, stabilitás

A dinamika korlátossága

Egy áramkörre mindig teljesül a gyakorlatban, mivel végés eneriga áll rendelkezésre.

Azonban meg kell vizsgálnunk a dinamikát, nehogy egy nem várt korlátba ütközzünk

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

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

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

Megoldás egyértelmű létezése

cél: megoldja a dinamika egy adott feladatát, de legyen általános is

- differenciálegyenletnek lehet végtelen sok megoldása, függnek a kezdeti feltételtől, amivel szűkíteni lehet a hasznos megoldások számát, de nem elég

Áramköri szinten

Lineáris elemekből épített áramkör használatával mindig egyértelműen létezik megoldás.

Azonban ezek számításra nem igazán alkalmasak. Egy döntés/osztályozás pont akkor érdekes, ha nemlineáris.

Nemlineáris elemek esetén nem biztosított a megoldás egyértelmű létezése – (már 2 elemű nemlineáris áramkör esetén sem)

Picard–Lindelöf tétel /Cauchy-Lipschitz/ (Émile Picard, Ernst Lindelöf)

Adott a következő kezdeti érték feladat:

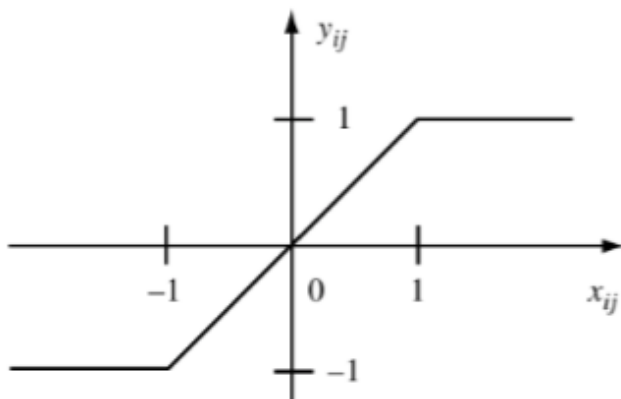
$$y'(t) = f(t, y(t)), \quad y(t_0) = y_0, \quad t \in [t_0 - \epsilon, t_0 + \epsilon]$$

Tfh:

f Lipschitz folytonos y-ban és folytonos t-ben

- A kezdetiérték feladatnak egyértelműen létezik megoldása a megadott halmazon
- Ekkor a kezdeti érték feladatnak egyértelműen létezik megoldása a megadott halmazra
- Felt 1: szinaptikus operátorok lineárisak és memóriamentesek: A és B skalárral szorzás:
Konstans, időben nem változó template értékeket használunk
- Felt 2: A bemenet $u_{ij}(t)$ és a bias $z_{ij}(t)$ folytonos függvények: A bemenet és a Bias is konstans
Felt 3: A nemlinearitás (kimenet: $y=f(x)$) Lipschitz folytonos A általunk használt nemlinearitás valóban Lipschitz folytonos

Stabilitás



-BIBO stabilitás—> bounded input, output

-structural stability

-Nyquist stability

-Asymptotic stability (Ljapunov)

Ljapunov stabilitás:

A dinamika tart valahova

Mérnökileg azt szeretnénk, hogy egy adott T idő elteltével mindig az elvárt megoldást olvassuk ki (lehetőleg minél pontosabban).

Differenciálegyenletek esetén ez nem megvalósítható, de elvárhatjuk, hogy az idő múlásával egyre közelebb kerüljünk a megoldáshoz.

Ljapunov fgv vizsgálható

Korlátos, monoton függvény, ami kapcsolódik a dinamikához Energifüggvény

$V(x) > 0$

$dV(x)/dx < 0$ (asszimptótikusan stabil, egyenlőség esetén stabil egyensúlyi pont)

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

Lássuk be, hogy valóban Ljapunov függvény

Ha A centrálisan szimmetrikus, akkor a dinamika asszimptótikusan stabil.

Elég, ha A előjelesen centrálisan szimmetrikus. Ezt most nem bizonyítjuk

7. Template tervezés, Bináris függvények tervezése, DP-plot

EDDIG: ismertük a templatet, láttuk, hogy működnek, libraryből töltöttük be őket

MOST: Ha megértettük a dinamikát → Meg kell oldanunk az inverz problémát:

Ismerve a bemenetet és az elvárt kimenetet, meg tudjuk adni a tényleges dinamikát, ami az adott funkcionalitással bír?

Ha igen, akkor tudunk számítógépet tervezni

Template tervezési módszerek:

- Intuitív template tervezés
- Adatbázis alapú tervezés, tanító algoritmusok (genetic algorithm) Sosem lehetünk teljesen biztosak, hogy az elért funkció általános. :(
- Direkt tervezés, Direct design: Megértve a dinamikát, meg tudjuk oldani a hálózatot leíró differenciál egyenletet. Ennek segítségével meg tudjuk mondani, hogy milyen funkcióval bír egy template.

Ezáltal megoldhatjuk az inverz problémát is, és generálhatunk template-eket egy bemenet, kimenet párra.

NxM-es hálózat esetén:

NxM dimenziós diffegyenlet – Ha van visszacsatolás, mindegyik elem mindegyikkel összefügg (SOK!)

Elég bonyolult egy ilyen diffegyenletrendszer megoldása

Feladat és template típusok

Az alábbi nem propagáló feladatosztályokat azonosíthatjuk:

- Lokális mintafelismerés
- Pixel változás felismerés
- Térbeli bináris logikai függvények, két bemenettel és egy kimenettel

DE megoldása Driving point plottal—>fixpontokat keresünk

nem szükséges numerikus megoldás

A template tervezés lépései

- 1, A feladat globális leírása
- 2, A lokális szabályok és formulák megállíptása
- 3, A legegyszerűbb template formula meghatározása
- 4, Igazságtáblázat elkészítése
- 5, Az egyenlőtlenségrendszer megoldása
- 6, Template optimalizáció
- 4+5, Driving-point plot

8. Parciális diffegyeneltek megoldása CNN-en

ODE- Ordinary differential equation

Egy változónk van és egy függvényt keresünk:

$$F(t, y, y', y'', y''', \dots) = 0$$

A megoldás $y=y(t)$ függvény

$$F(t, y, y', y'', y''', \dots) = 0$$

A diffegyenlet: $y' = ky$

A megoldása: $y = Ce^{(kt)}$

A diffegyenlet: $y'' - y = 0$

A megoldása: $y = c_1 e^{(t)} + c_2 e^{(-t)}$

PDE- Partial differential equation

Többváltozós verziója az ODE-knek:

$$F(x_1, x_2, \dots, x_n, \frac{\partial u}{\partial x_1}, \frac{\partial u}{\partial x_2}, \dots, \frac{\partial u}{\partial x_n}, \frac{\partial^2 u}{\partial x_1^2}, \frac{\partial^2 u}{\partial x_1 x_2}, \dots) = 0$$

A megoldás $y=y(x_1, x_2, \dots, x_n)$ függvény

A diffegyenlet: $x \frac{\partial u}{\partial x_1} + y \frac{\partial u}{\partial y} = 0$

A megoldása: $u(x, y) = x^2 + y^2$

A diffegyenlet: $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$

A megoldása: $u(x, y) = x^2 - y^2$

Local PDE:

- a rendszer dinamikája bármely pontban csak lokális tulajdonságoktól függ

Global PDE (Sapiro et. al):

- Létezik olyan pont, ahol a rendszer dinamikája globális (távoli) tulajdonságoktól függ

Miért fontosak a PDEk?

A fizikai világunk nagyon jó és tömör leírói.

PDE példák

		Gradient operator
Wave equation	$\frac{\partial^2 u}{\partial t^2} = c^2 \Delta u$	$\nabla = \frac{\partial}{\partial x} \mathbf{i} + \frac{\partial}{\partial y} \mathbf{j} + \dots$

Heat equation	$\frac{\partial u}{\partial t} = c^2 \Delta u$
---------------	--

Laplacian

$$\Delta = \nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \dots$$

Wave equation: egy hullám terjedése egy adott közegben (áramlástan, elektromágneses hullámok)

PDE megoldása

Egy megoldás ellenőrzése könnyű, ugyanúgy mint ODE esetén. Ki tudjuk számolni a függvény és a derivált értékét minden pontban.

A megoldás megkeresése azonban rendkívül bonyolult.

ODE-k esetében több módszer is létezik (próbafüggvény, SÉ SV...).

Nem létezik általános eljárás PDE megoldására

Különböző osztályokra vannak analitikus megoldási módszerek.

PDE-ket általában hagyományos számítógépen numerikusan közelítjük

Diszkretizáció

Térben és időben folytonos, diszkretizálni kell

Térbeli diszkretizáció:

Mintavételezés adott pontokban (pixelek)

Diszkretizáció CNN-en

Egy diszkretizációt el tudunk kerülni – a CNN folytonos időben működik, így csupán a többi parciális derivált alapján (általában térben) kell diszkretizálnunk.

Közelítenünk kell a második deriváltat:

$$f''(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$
$$\frac{f(x+h) - 2f(x) + f(x-h)}{h^2} = \frac{\frac{f(x+h)-f(x)}{h} - \frac{f(x)-f(x-h)}{h}}{h}$$

Heat equation - gyakorlati alkalmazás

A B template-et nem használtuk fel, ezáltal időben változó bemenettel akár melegítés során is vizsgálhatjuk a hőterjedést

Nem egyenletes hőterjedés: heterogén CNN, különböző A template-ek különböző pontokban

Nemlineáris diffusion equation?

A PDE-k egy jelentős osztályát egyszerűen szimulálhatjuk, azonban számos olyan egyenlet van, ahol a parciális deriváltak nemlineáris kombinációjára lenne szükség.

$$\begin{aligned} \frac{d}{dt} v_{x_{ij}}(t) = & -v_{x_{ij}}(t) + \sum_{kl \in N_r} A_{ij,kl} v_{y_{kl}}(t) + \sum_{kl \in N_r} B_{ij,kl} v_{u_{kl}}(t) + I_{ij} \\ & + \sum_{kl \in N_r} \hat{D}_{ij,kl}(\Delta v) \end{aligned}$$

Az extra tag esetében D template elemei függvények az argumentuma pedig a bemenet a state és a kimenet

A PDE-k egy jelentős osztályát egyszerűen szimulálhatjuk.

Minden nemlineáris template megvalósítható lineáris template-ek sorozataként

A nemlineáris template-ek használatával azonban az algoritmus A template-ek tervezése is egyszerűbb

Ténylegesn architektúrákon (ugyan nem mindegyiken) is van lehetőség nemlineáris template-ek megvalósítására: pl Ace16k