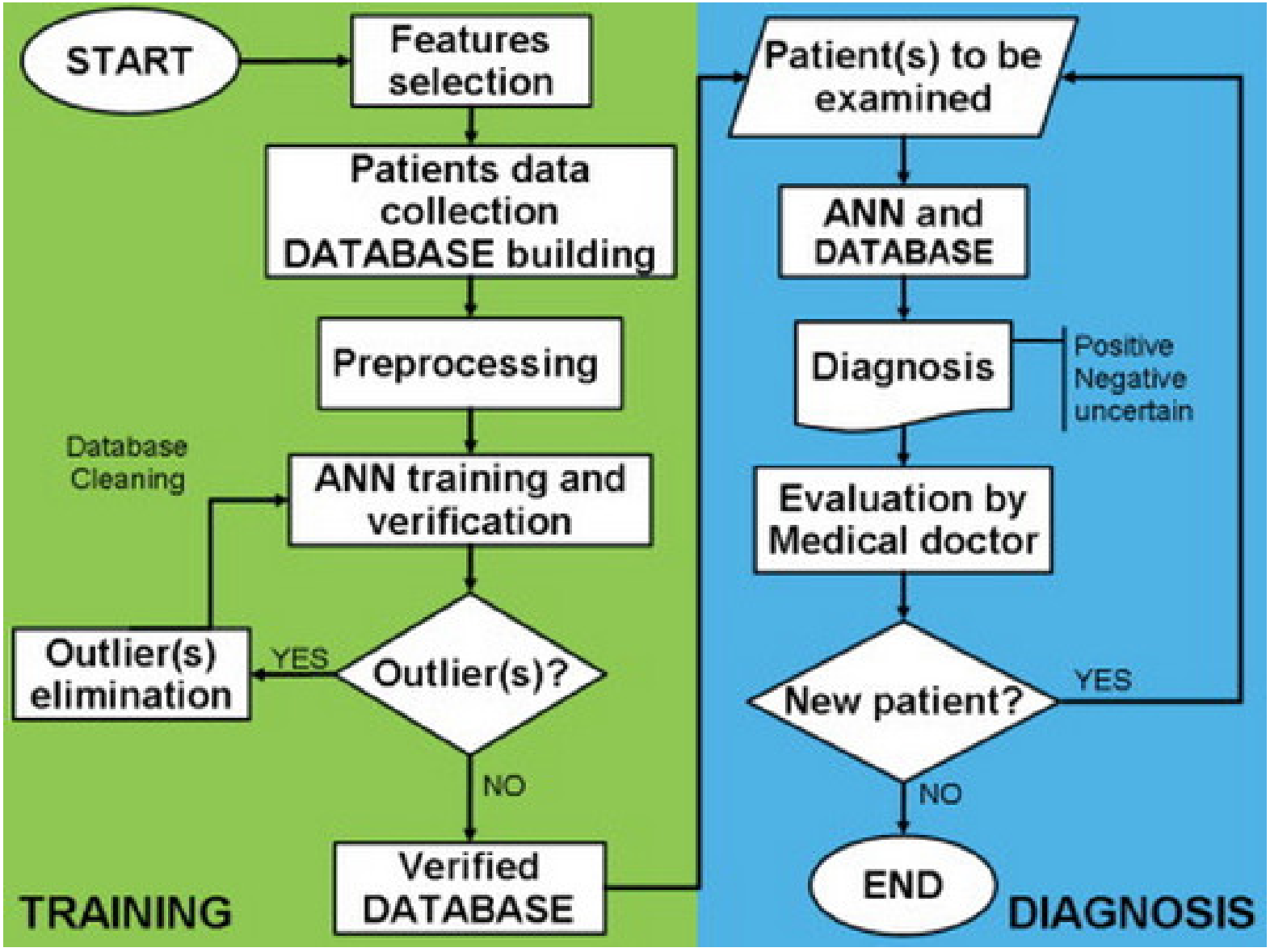
Aplikace vícevrstvých sítí

- Diagnostika v medicíně
- Predikce časových řad
- ALVINN
- Rozpoznávání směrovacích čísel
- Komprese dat

Zdroj: Fundamentals of Artificial Neural Networks. Mohamad H. Hassoun, The MIT Press

Diagnostika pomocí neuronových sítí



Diagnóza srdečního infarktu

- infarkt není jednoduché diagnostikovat
- expert (údajně) pozná infarkt ve zhruba 88% případů, planý poplach nastává ve zhruba 29%.
- neuronové sítě to počátkem devadesátých let zvládly lépe

Organizační dynamika:

- Třívrstvá síť 41 – 10 – 10 – 1
- Vstupy jsou různé atributy pacientů přijatých na kardiologii:
  - např. věk, pohlaví, nevolnost, zvracení, dušnost, cukrovka, vysoký tlak, ...
  - binární vstupy jako např. přítomnost cukrovky nebo pohlaví jsou kódovány pomocí 0 a 1
  - ostatní jsou normalizovány do intervalu $[0, 1]$

Aktivní dynamika:

- aktivační funkce: standardní logistické sigmoidy $\sigma(\xi) = \frac{1}{1+e^{-\xi}}$

Diagnóza srdečního infarktu

Adaptivní dynamika: trénink:

- tréninková množina obsahovala 356 pacientů, 236 bez infarktu, 120 s infarktem
- gradientní sestup na náhodně vybrané polovině pacientů (půl s a půl bez infarktu)
- očekávané výstupy 1 nebo 0 podle toho, zda pacient měl či neměl infarkt

výsledky:

- testováno na 178 pacientech, kteří nebyli součástí tréninkové množiny (60 s infarktem, 118 bez)
- 92% korektní identifikace infarktu (oproti 88% u expertů)
- 4% falešný poplach (oproti 29% u expertů)

Existuje velké množství podobných aplikací neuronových sítí v diagnostice, viz. např. `http://jab.zsf.jcu.cz//11_2/havel.pdf`

Predikce časových řad

Např. se jedná o předvídání vývoje počasí, hodnot akcií, měnových kurzů, počtu zákazníků apod.

(hrubá) matematická formulace:

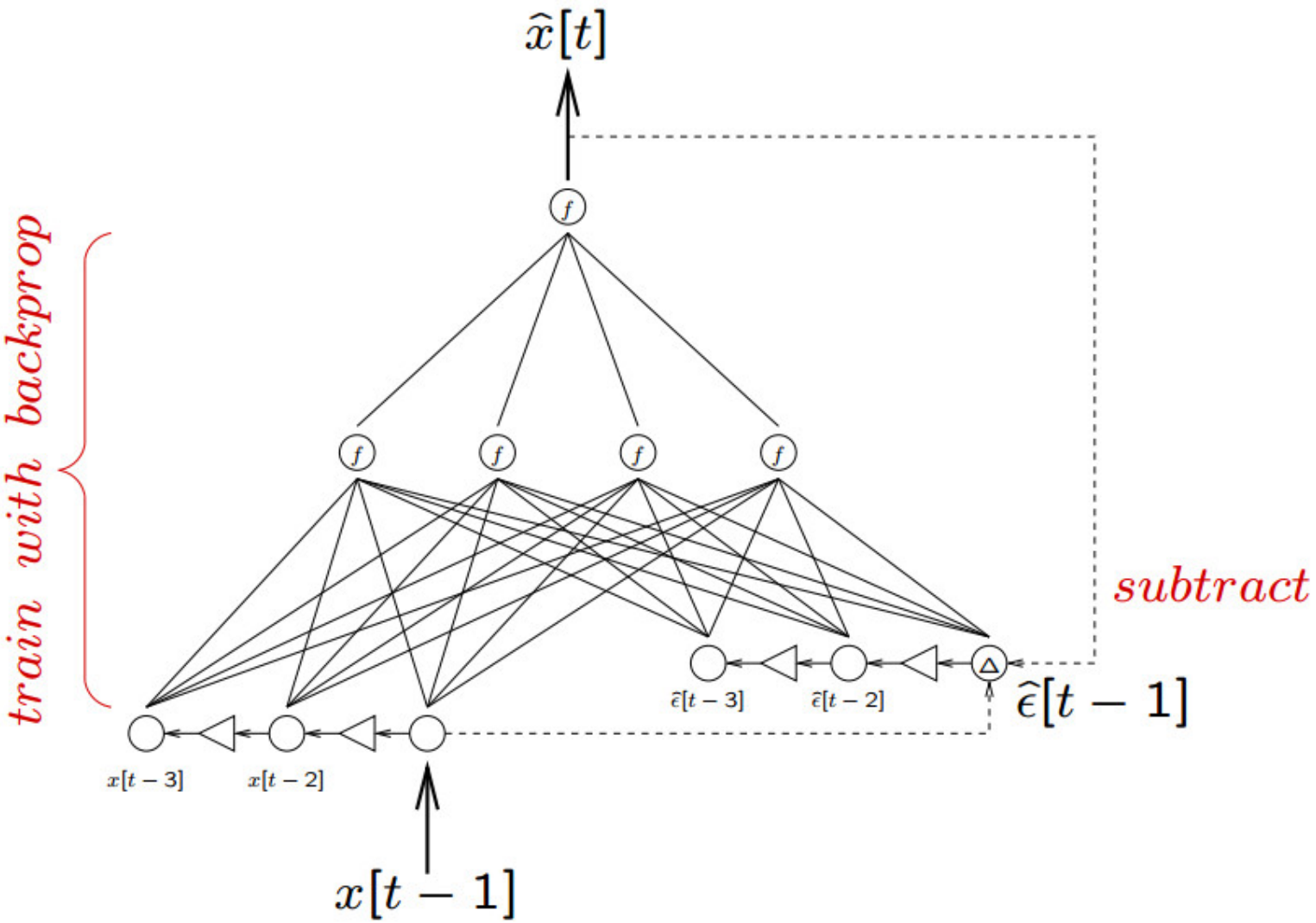
- Mějme numerickou řadu $x[1], x[2], x[3], \ldots$
  (např. hodnota koruny vůči euru po jednotlivých dnech)
- Chceme odhadnout $x[\ell + 1]$ na základě $x[\ell], x[\ell - 1], \ldots, x[\ell - k]$

Uvažme síť s k vstupními a jedním výstupním neuronem. Tuto síť natrénujeme na části řady tak, že jí budeme předkládat $x[\ell], x[\ell - 1], \ldots, x[\ell - k]$ na vstupy a $x[\ell + 1]$ očekávaný výstup.
(V případě měnových kurzů použijeme historii vývoje cen za minulé období.)

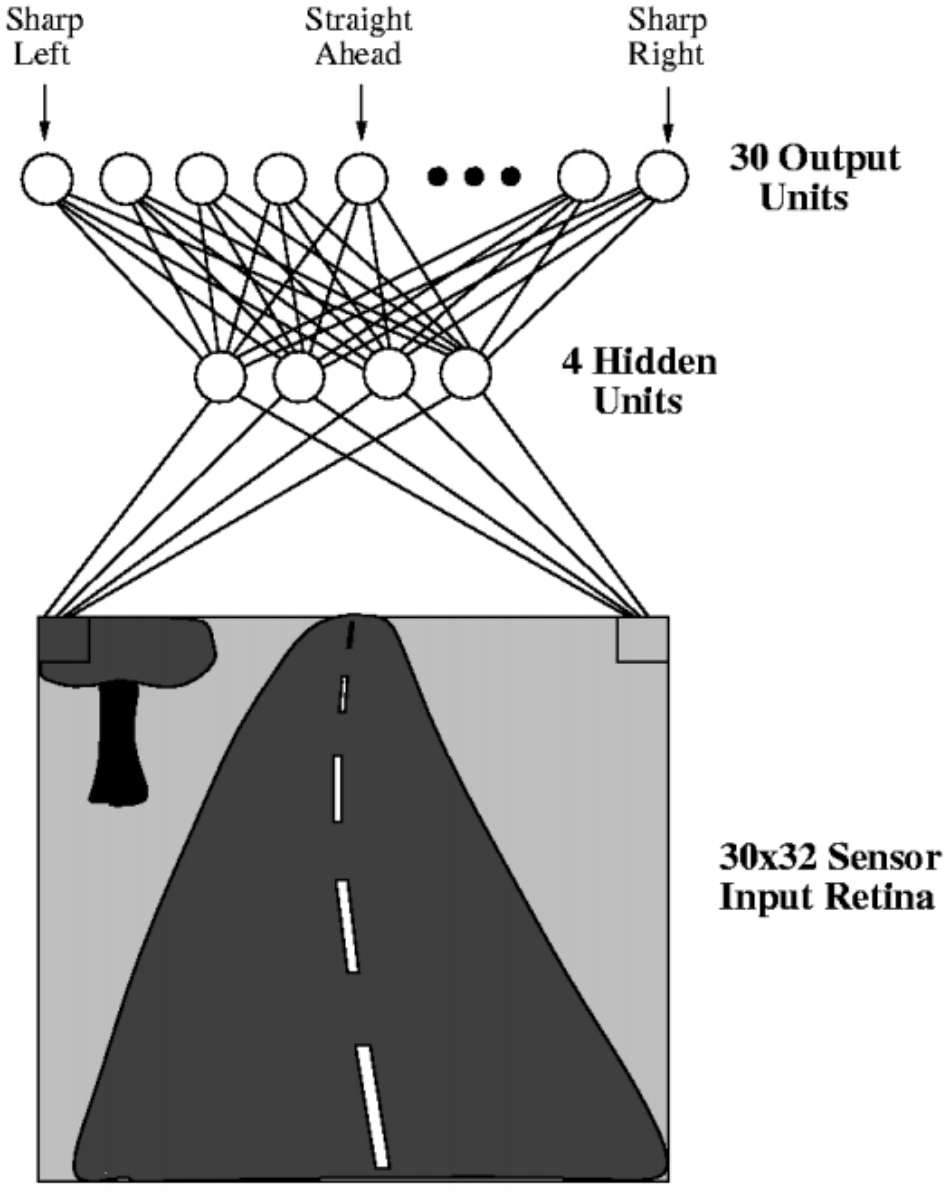
Sítě pro predikci časových řad se obvykle rozšiřují o další vlastnosti:

- další vstupy modelující prostředí v němž se řada vyvíjí (např. hospodářské parametry státu), chyby předchozích odhadů (ARMA(p,q) a další metody)
- výstup se často vrací zpět jako další vstup, čímž lze dosáhnout predikce na více kroků

ARMA(p,q) ilustrace



ALVINN


Sharp Left
Straight Ahead
Sharp Right
30 Output Units
4 Hidden Units
30x32 Sensor Input Retina

ALVINN

Organizační dynamika:

- vícevrstvá síť, 960 – 4 – 30 (někdy 960 – 5 – 30)
- komponenty vstupu odpovídají bodům obrazu z kamery

Aktivní dynamika:

- aktivační funkce: skryté neurony mají sigmoidální funkce, výstupní mají lineární
- Směr jízdy odpovídá těžišti všech výstupních neuronů

  tj. výstupní neurony lze uvažovat jako hmotné body umístěné na přímce se stejným rozestupem, hmotnost neuronu se rovná jeho hodnotě

ALVINN

Adaptivní dynamika: Trénováno za jízdy.

- Aktuální výhled na silnici snímán kamerou, zhruba 25 obrazů za vteřinu
- Tréninkové vzory tvaru $(\vec{x}_k, \vec{d}_k)$ kde
  - $\vec{x}_k$ = obraz silnice
  - $\vec{d}_k$ = příslušné natočení volantu řidiče
- natočení volantu distribuováno pomocí Gaussova rozložení na výstupy:

$$d_{ki} = e^{-D_i^2/10}$$

kde D_i je vzdálenost i-tého výstupu od toho, který odpovídá natočení volantu

(Toto je lepší než binární výstup, protože reakce na podobné silnice jsou velmi blízké.)

Výběr vzorů

Naivní přístup: brát vstupy z kamery a podle nich adaptovat.

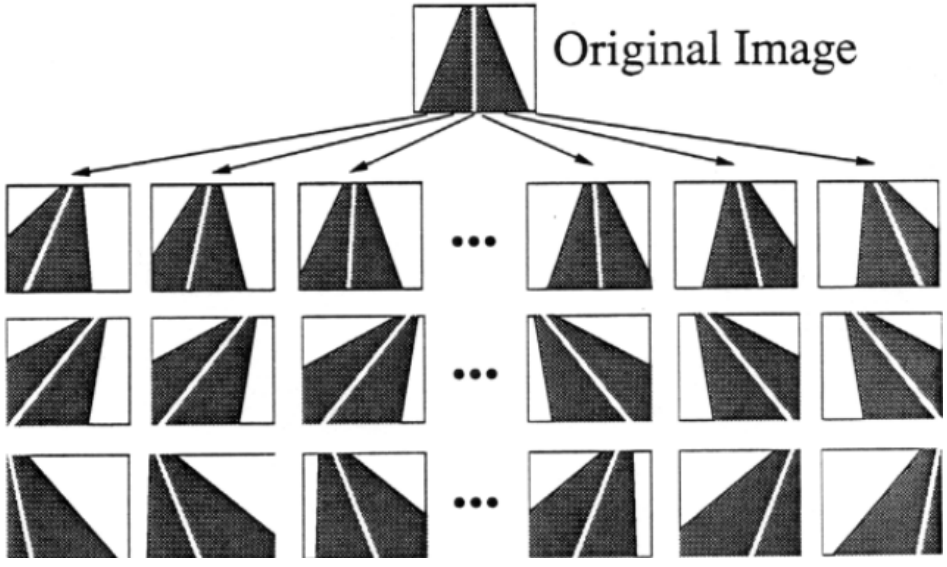
To vede k následujícím problémům:

- Jestliže řidič jede dobře, síť se nenaučí řešit odchylky od trasy. Možná drsná řešení jsou
  - vypnout přechodně učení a sjet z trasy, poté zapnout učení a nechat síť sledovat, jak se s tím řidič vyrovná
  - nechat řidiče jezdit divoce (poněkud nebezpečné, drahé, nespolehlivé)
- aktuální výhledy z okna jsou poněkud repetitivní, síť se může přetrénovat na málo vzorech

Výběr vzorů

Problém s příliš „správnou“ jízdou řidiče se řeší takto:

- každý výhled na silnici je posouváním rozmnožen na 15 podobných kopií



- požadovaný výstup se vygeneruje pro každou kopii

Repetitivnost aktuálních výhledů z okna se řeší takto:

- systém má buffer 200 obrázků (včetně 15 kopií aktuálního), v každém kole tréninku se trénuje na těchto vzorech
- po tréninku se sejme nový obraz, udělá se 15 kopií a těmi se nahradí 15 obrazů z bufferu (10 s nejmenší chybou, 5 náhodně)

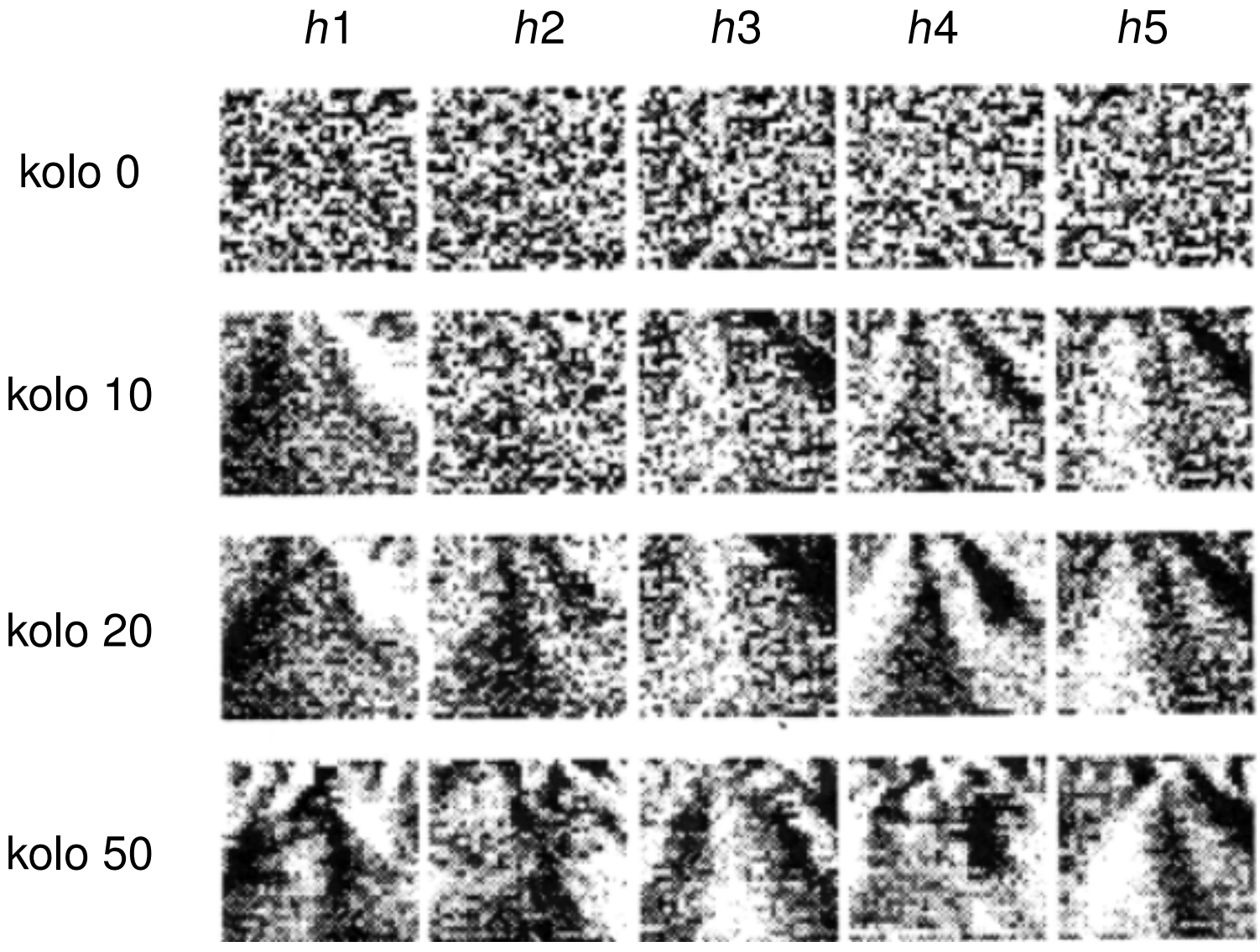
ALVINN - učení

- standardní zpětná propagace
- konstantní rychlost učení pro každý neuron zvlášť, úměrná počtu vstupů
- pomalu rostoucí moment

Výsledek:

- Trénink trval 5 minut, řidič jel rychlostí 4 míle za hodinu
- ALVINN byl schopen jet i po částech silnice, které nikdy „neviděl“ a za rozličného počasí
- v době vzniku byl schopen jet maximální rychlostí, kterou zvládal hydraulický ovladač

ALVINN - vývoj vah



Zde $h1, \ldots, h5$ jsou skryté neurony.

ALVINN - komentáře

Srovnání ALVINNa s „explicitním programováním“.
Pro efektivní řízení je potřeba:

- najít vlastnosti obrázků důležité pro řízení
  (ALVINN najde sám)
- tyto vlastnosti detekovat v aktuálních obrazech
  (ALVINN si vytvoří vlastní detektory)
- implementovat řízení v reakci na vlastnosti obrázků
  (ALVINN se to naučí sám od řidiče (rychle))

Nevýhody ALVINNa (později řešené celou škálou rozšíření)

- uměl jezdit jen po jednom typu silnice (různý povrch, počet pruhů, atd.)

  Později řešeno pomocí slučování více ALVINNů spojených do jedné sítě, každý natrénován na jiný typ silnice (MANIAC)
- nebyl nijak napojen na „vyšší“ řízení, například sledování cesty po mapě apod.

  Řešeno např. včleněním ALVINNa do většího učícího systému.

Rozpoznávání směrovacích čísel

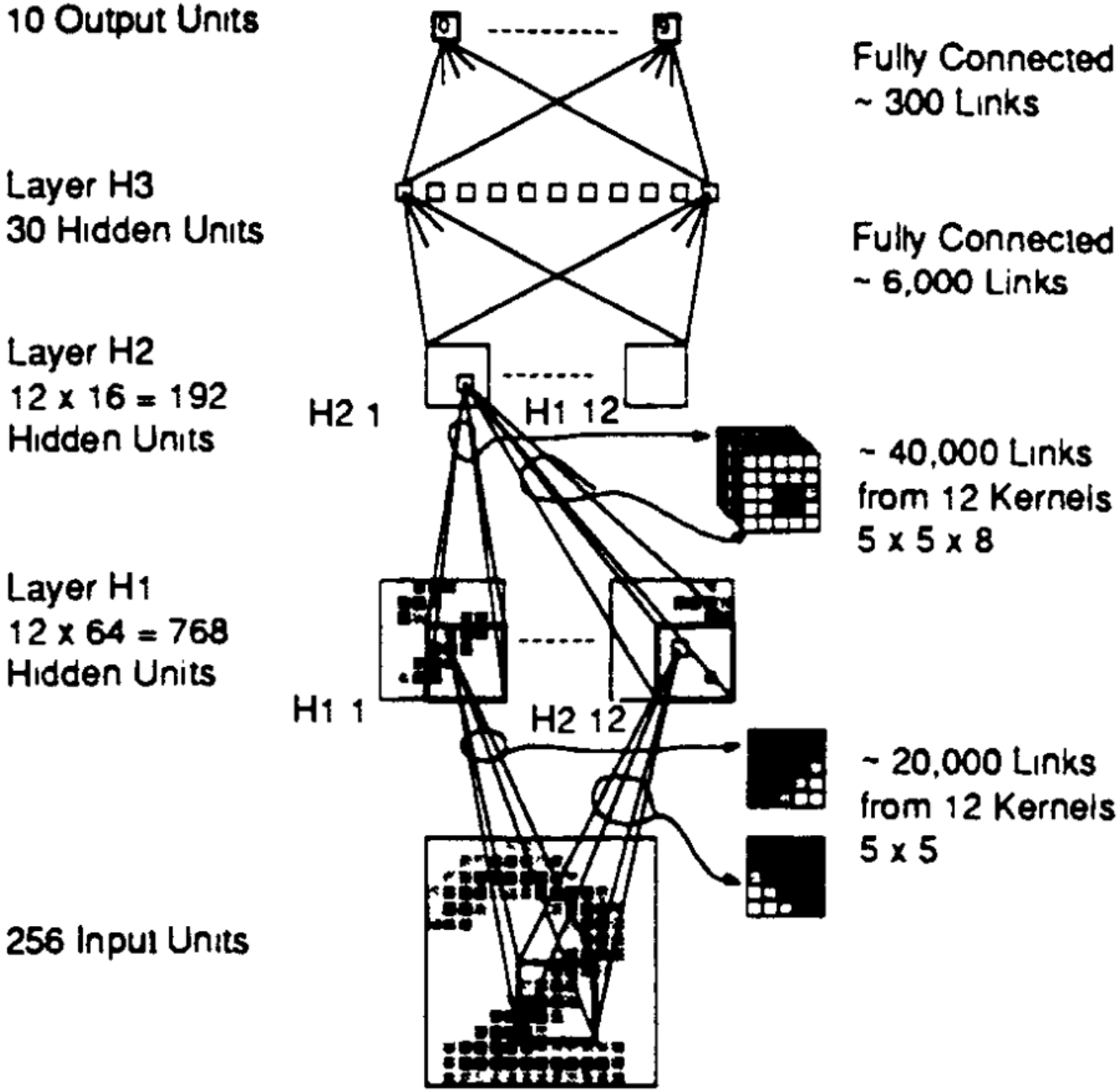
Cílem je rozpoznat rukou psané číslice

- vstupy: obrázky číslic 16×16, stupně šedi normalizovány do $[-1, 1]$

Fig. 4. Size-normalized examples from the MNIST database.

- výstup: jedna z deseti hodnot

Rozpoznávání číslic



Rozpoznávání číslic

Organizační dynamika: acyklická síť (4 vrstvy), něco jako vícevrstvá, ale nemusí vést spoj mezi všemi neurony sousedících vrstev

- 1. vrstva:
  - 12 skupin po 8×8 neuronech (skupina 8×8 tvoří „obrázek“ pro vyší vrstvu)
  - každý neuron z jedné skupiny má vstupy z okna 5×5 ze vstupu (okna se překrývají, jsou položena ob dva pixely)
  - všechny neurony z jedné skupiny *sdílí stejné váhy* (mají za úkol detekovat stejnou vlastnost v různých částech obrázku)
- 2. vrstva:
  - 12 skupin po 4×4 neuronech
  - každý neuron z jedné skupiny má vstupy z oken 5×5 z osmi skupin z nižší vrstvy (všech 8 oken pro jeden neuron má stejnou polohu)
  - všechny neurony z jedné skupiny sdílí stejné váhy
- 3. vrstva: 30 neuronů, kompletně spojena s předchozí vrstvou

  4. vrstva: 10 výstupních n., kompletně spojena s předchozí

Rozpoznávání číslic

Aktivní dynamika: aktivační funkce: hyperbolický tangens

Dvojí interpretace výstupu:

- výstupní neuron s největší hodnotou identifikuje číslici
- totéž, ale vstupy se dvěma a více velkými výstupními hodnotami (tj. nejednoznačné) byly odmítnuty

Adaptivní dynamika:
Vstupy:

- trénink na 7291 vzorech, testováno na 2007 vzorech
- mnoho příkladů bylo hodně pokřivených

Trénink:

- modifikovaná zpětná propagace (v podstatě sdružené gradienty), online
- váhy iniciálně náhodně z $[-2.4, 2.4]$, poděleny počtem vstupů daného neuronu

Rozpoznávání číslic - výsledky

- výsledky bez odmítání nejednoznačných: 0.14% špatně klasifikovaných na tréninkové množině, 5% na testovací
- výsledky s odmítáním nejednoznačných: 1% špatně na testovacích za cenu 12% odmínutých

- obyčejná dvouvrstvá síť se 40 skrytými neurony se dostala na 1% špatně klasifikovaných za cenu 19.4% odmítnutých

Databáze MNIST

- Databáze (označkovaných) obrázků rukou psaných číslic: 60 000 tréninkových, 10 000 testovacích.
- Dimenze obrázků je 28 x 28 pixelů, jsou vycentrované do "těžiště" jednotlivých pixelů a normalizované na fixní velikost
- Více na `http://yann.lecun.com/exdb/mnist/`

Databáze se používá jako standardní benchmark v mnoha publikacích o rozpoznávání vzorů (nejen) pomocí neuronových sítí.

Lze porovnávat přesnost klasifikace různých metod.

Například jeden z nejlepších výsledků je:

6-layer NN 784-2500-2000-1500-1000-500-10 (on GPU)

Abstrakt: Good old on-line back-propagation for plain multi-layer perceptrons yields a very low 0.35 error rate on the famous MNIST handwritten digits benchmark. All we need to achieve this best result so far are many hidden layers, many neurons per layer, numerous deformed training images, and graphics cards to greatly speed up learning.

Komprese dat

Komprese obrazových dat pro přenos signálu.

- Použita dvouvrstvá síť $n - \frac{n}{4} - n$ (tj. počet skrytých neuronů je 4x menší než počet vstupních a výstupních).
- Síti byly předkládány obrazy, stejný obraz na vstup i výstup.
- Naučená síť potom byla použita takto:
  - Vysílající vypočte pro daný vstup hodnoty skrytých neuronů
  - Hodnoty skrytých neuronů jsou odeslány příjemci
  - Příjemce vypočte hodnoty výstupních neuronů po dosazení hodnot skrytých neuronů

Metoda funguje pokud jsou vysílané obrazy podobné tréninkovým vzorům.

Dá se ukázat, že tato metoda realizuje PCA na obrazových datech - tedy nejlepší možnou redukci dimenze dat (probereme později)

Komprese dat - konkrétní implementace (historická)

Organizační dynamika: Vícevrstvá síť 64 – 16 – 64

Aktivní dynamika: aktivační funkce: sigmoidální, bipolární (tedy nějaký hyperbolický tangens s extrémy -1 a 1)

Adaptivní dynamika:
Vstupy:

- obrázky 256×256, 8 bitů na pixel
- vzory: vstup i výstup byl rámeček 8×8, který se náhodně volil z obrázku
- vstupy normalizovány do intervalu $[-1, 1]$

Učení:

- zpětná propagace
- rychlost učení: 0.01 pro vnitřní, 0.1 pro výstupní
- trénováno v 50 000 - 100 000 iteracích

Komprese dat - výsledky

(A)

(B)

(C)

(D)

Tréninkový obraz

obraz 256×256 se projede rámečkem 8×8 (jednotlivá „přiložení“ rámečku se nepřekrývají)

- **(A)** originál
- **(B)** komprese
- **(C)** komprese + zaokrouhlení hodnot vnitřních neuronů na 6 bitů (přenos 1.5 bitu na pixel)
- **(D)** komprese + zaokrouhlení hodnot vnitřních neuronů na 4 bity (přenos 1 bit na pixel)

Komprese dat - výsledky

(A)

(B)

(C)

(D)

Nový obraz

- **(A)** originál
- **(B)** komprese
- **(C)** komprese + zaokrouhlení hodnot vnitřních neuronů na 6 bitů (přenos 1.5 bitu na pixel)
- **(D)** komprese + zaokrouhlení hodnot vnitřních neuronů na 4 bity (přenos 1 bit na pixel)

Asociativní sítě

Cílem je uchovat množinu vzorů $\{(\vec{x}_k, \vec{d}_k) \mid k = 1, \ldots, p\}$ tak, aby platilo následující:

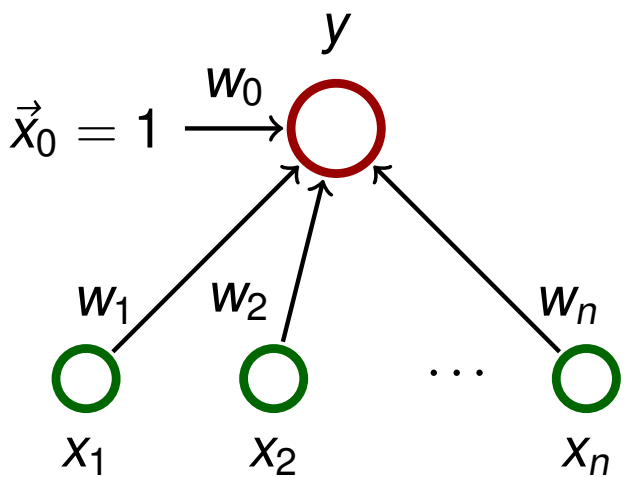
Po předložení nového vstupu $\vec{x}$, který je „blízko“ některému $\vec{x}_k$ bude výstup sítě roven nebo alespoň blízko $\vec{d}_k$
(tato schopnost se nazývá *asociace*).

Zejména by síť měla mít schopnost *reprodukce*: Pro vstup $\vec{x}_k$ by měla dát výstup $\vec{d}_k$.

Lineární asociativní síť (alias ADALINE s Hebbovým učením)

(Pro jednoduchost a srovnání s ADALINE uvážíme pouze jeden výstup)

Organizační dynamika LAS:



$\vec{w} = (w_0, w_1, \ldots, w_n)$ a $\vec{x} = (x_0, x_1, \ldots, x_n)$ kde $x_0 = 1$.

Aktivní dynamika:

funkce sítě: $y[\vec{w}](\vec{x}) = \vec{w} \cdot \vec{x} = \sum_{i=0}^{n} w_i x_i$

Hebbovo učení

Adaptivní dynamika:

Dána množina **tréninkových vzorů**

$$\mathcal{T} = \left\{\left(\vec{x}_1, d_1\right), \left(\vec{x}_2, d_2\right), \ldots, \left(\vec{x}_p, d_p\right)\right\}$$

Zde $\vec{x}_k = (x_{k0}, x_{k1} \ldots, x_{kn})^\top \in \mathbb{R}^{n+1}$, $x_{k0} = 1$, je vstup k-tého vzoru a $d_k \in \mathbb{R}$ je očekávaný výstup.

Intuice: chceme, aby síť počítala afinní aproximaci funkce, jejíž (některé) hodnoty nám předepisuje tréninková množina.

Hebbovo učení

Hebbův zákon: *When an axon of a cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased.*

Zákon formuloval neuropsycholog Donald Hebb v knize „the organization of Behavior" z roku 1949.

Jinými slovy: *Cells that fire together, wire together.*

Formulace používaná v umělých NS:
Změna váhy spoje mezi dvěma neurony je úměrná jejich souhlasné aktivitě.

Hebb se snažil vysvětlit podmíněné reflexy: Současná aktivita/pasivita presynaptického neuronu (příčina) a postsynaptického neuronu (reakce) posiluje/zeslabuje synaptickou vazbu.

Hebbovo učení LAS

Algoritmus počítá posloupnost vektorů vah $\vec{w}^{(0)}, \vec{w}^{(1)}, \ldots, \vec{w}^{(p)}$:

- $\vec{w}_i^{(0)} = 0$ pro $0 \leq i \leq n$,
- v kroku k (zde $k = 1, 2, \ldots$) je síti předložen vzor $(\vec{x}_k, d_k)$ a váhy se adaptují podle Hebbova zákona:

$$\vec{w}^{(k)} = \vec{w}^{(k-1)} + \vec{x}_k d_k$$

Výsledný vektor:

$$\vec{w} = \vec{w}^{(p)} = \sum_{k=1}^{p} \vec{x}_k d_k = X^\top \vec{d}$$

kde X je matice, která má v *i-tém řádku* vektor $\vec{x}_i^\top$ a $\vec{d} = (d_1, \ldots, d_p)^\top$

LAS a ortonormální vstupy

Pokud jsou $\vec{x}_1, \ldots, \vec{x}_p$ ortonormální, tedy

$$\vec{x}_i^\top \vec{x}_j = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$$

pak LAS má schopnost **reprodukce**:

$$\vec{w}^\top \vec{x}_i = \sum_{k=1}^{p} (\vec{x}_k d_k)^\top \vec{x}_i = \sum_{k=1}^{p} d_k (\vec{x}_k^\top \vec{x}_i) = d_i$$

LAS a ortonormální vstupy

... a **asociace**:

Uvažme vstup: $\vec{x}_r + \vec{u}$ kde norma $\left\|\vec{u}\right\|$ je malá.

Chyba sítě pro r-tý vzor perturbovaný vektorem $\vec{u}$:

$$E_r := |\vec{w}^\top(\vec{x}_r + \vec{u}) - d_r| = |\vec{w}^\top \vec{x}_r + \vec{w}^\top \vec{u} - d_r| = |\vec{w}^\top \vec{u}|$$

Pokud $\vec{d}_r \in \{-1, 1\}$, pak

$$E_r = |\vec{w}^\top \vec{u}| = \left|\left(\sum_{k=1}^{p} \vec{x}_k d_k\right)^\top \vec{u}\right| = \left|\sum_{k=1}^{p} d_k(\vec{x}_k^\top \vec{u})\right|$$

$$\leq \sum_{k=1}^{p} |d_k(\vec{x}_k^\top \vec{u})| \leq \sum_{k=1}^{p} |d_k| \left\|\vec{x}_k\right\| \left\|\vec{u}\right\| \leq n \left\|\vec{u}\right\|$$

(Zde první nerovnost plyne z trojúhelníkové nerovnosti, druhá z Cauchyho-Schwarzovy nerovnosti a poslední z $p \leq n$, protože mohutnost množiny ortonormálních vektorů v $\mathbb{R}^n$ nemůže být větší než n.)

Tedy pro vstupy blízké vzorům odpovídá přibližně požadovaným výstupem

Hopfieldova síť

- Definice
- Energetická funkce
- Reprodukce
- Asociace

Hopfieldova síť

Autoasociativní síť.

Organizační dynamika:

- úplná topologie, tj. každý neuron je spojen s každým
- všechny neurony jsou současně vstupní i výstupní
- označme $\xi_1, \ldots, \xi_n$ vnitřní potenciály a $y_1, \ldots, y_n$ výstupy (stavy) jednotlivých neuronů
- označme w_{ji} *celočíselnou* váhu spoje od neuronu $i \in \{1, \ldots, n\}$ k neuronu $j \in \{1, \ldots, n\}$
- **Zatím:** žádný neuron nemá bias a přepokládáme $w_{jj} = 0$ pro každé $j = 1, \ldots, n$

Hopfieldova síť

Adaptivní dynamika: Dána tréninková množina

$$\mathcal{T} = \{\vec{x}_k \mid \vec{x}_k = (x_{k1}, \ldots, x_{kn}) \in \{-1, 1\}^n, k = 1, \ldots, p\}$$

Adaptace probíhá podle Hebbova zákona (podobně jako u LAS). Výsledná konfigurace je

$$w_{ji} = \sum_{k=1}^{p} x_{kj} x_{ki} \qquad 1 \leq j \neq i \leq n$$

Všimněte si, že $w_{ji} = w_{ij}$, tedy matice vah je symetrická.

Adaptaci lze vidět jako hlasování vzorů o vazbách neuronů: $w_{ji} = w_{ij}$ se rovná rozdílu mezi počtem souhlasných stavů $x_{kj} = x_{ki}$ neuronů i a j a počtem rozdílných stavů $x_{kj} \neq x_{ki}$.

Hopfieldova síť

Aktivní dynamika: Iniciálně jsou neurony nastaveny na vstup $\vec{x} = (x_1, \ldots, x_n)$ sítě, tedy $y_j^{(0)} = x_j$ pro každé $j = 1, \ldots, n$.

Cyklicky aktualizujeme stavy neuronů, tedy v kroku $t + 1$ aktualizujeme neuron j, t. ž. $j = (t \mod p) + 1$, takto:

nejprve vypočteme vnitřní potenciál

$$\xi_j^{(t)} = \sum_{i=1}^{n} w_{ji} y_i^{(t)}$$

a poté

$$y_j^{(t+1)} = \begin{cases} 1 & \xi_j^{(t)} > 0 \\ y_j^{(t)} & \xi_j^{(t)} = 0 \\ -1 & \xi_j^{(t)} < 0 \end{cases}$$

Hopfieldova síť - aktivní dynamika

Výpočet končí v kroku t^* pokud se síť nachází (poprvé) ve *stabilním* stavu, tj.

$$y_j^{(t^*+n)} = y_j^{(t^*)} \qquad (j = 1, \ldots, n)$$

Věta

Za předpokladu symetrie vah, výpočet Hopfieldovy sítě skončí pro každý vstup.

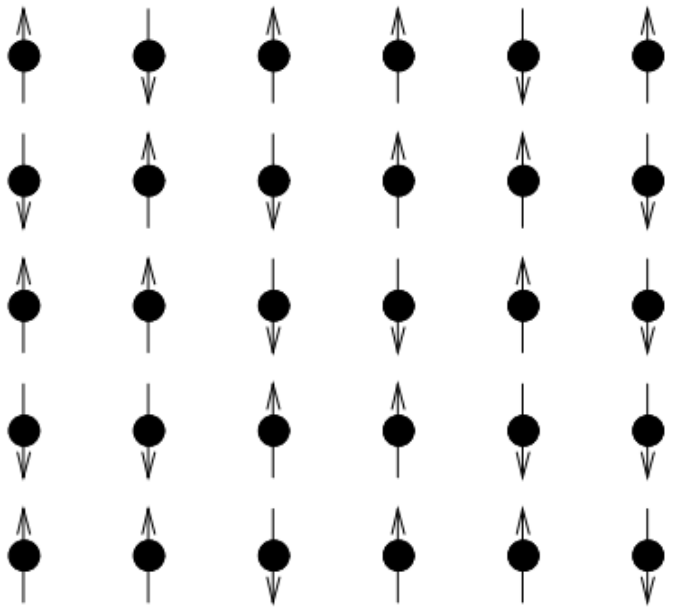
Z toho plyne, že Hopfieldova síť počítá funkci z $\{-1, 1\}^n$ do $\{-1, 1\}^n$ (která závisí na hodnotách vah neuronů).

Označme $\vec{y}(W, \vec{x}) = \left(y_1^{(t^*)}, \ldots, y_n^{(t^*)}\right)$ hodnotu funkce sítě pro vstup $\vec{x}$ a matici vah W. Dále označme $y_j(W, \vec{x}) = y_j^{(t^*)}$ složku hodnoty funkce sítě, která odpovídá neuronu j.

Pokud bude W jasné z kontextu, budu psát jen $y(\vec{x})$ a $y_j(\vec{x})$

Fyzikální analogie (Isingův model)

Jednoduché modely magnetických materiálů připomínají Hopfieldovu síť.



- atomické magnety poskládané do mřížky
- každý magnet může mít pouze jednu ze dvou orientací (v Hopfieldově síti $+1$ a -1)
- orientaci každého magnetu ovlivňuje jednak vnější magnetické pole (vstup sítě), jednak magnetické pole ostatních magnetů (závisí na jejich orientaci)
- synaptické váhy modelují vzájemnou interakci magnetů

Energetická funkce

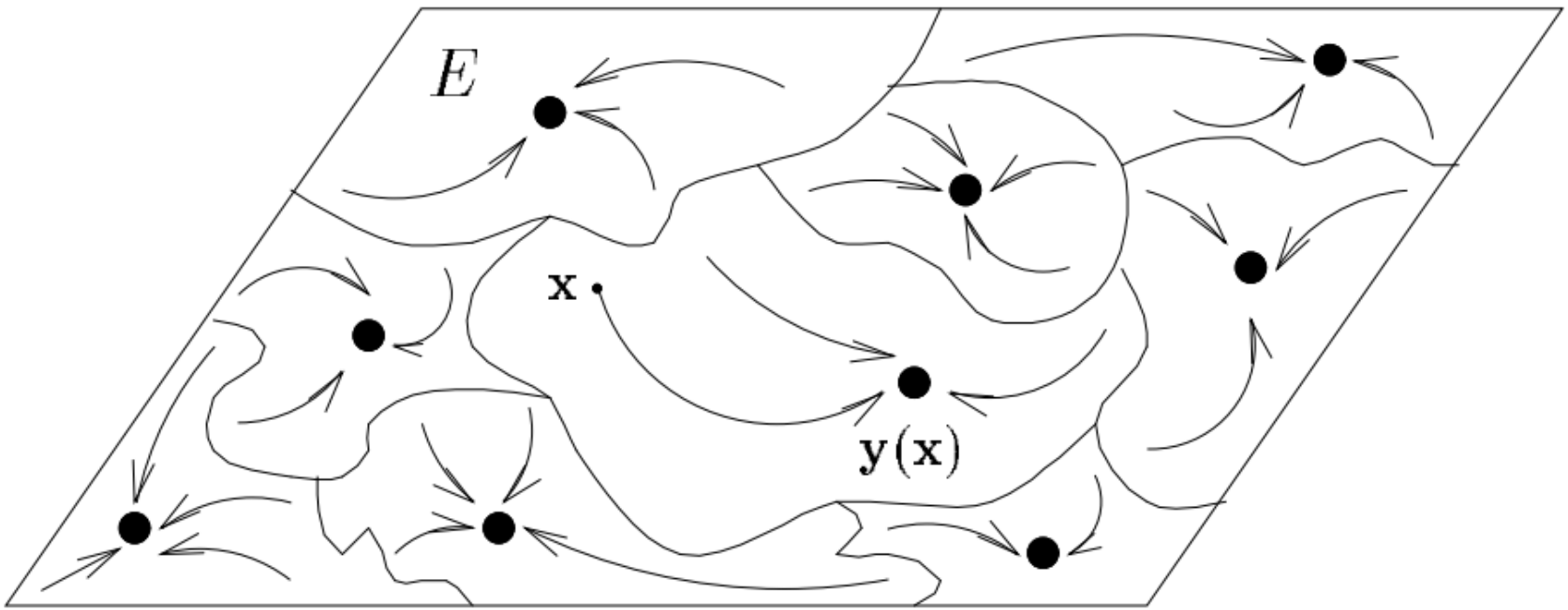
Energetická funkce E přiřazuje každému stavu sítě $\vec{y} \in \{-1, 1\}^n$ potenciální energii danou

$$E(\vec{y}) = -\frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{n} w_{ji} y_j y_i$$

- stavy s nízkou energií jsou stabilní (málo neuronů „chce“ změnit svůj stav), stavy s vysokou energií jsou nestabilní
- tj. velké (kladné) $w_{ji} y_j y_i$ je stabilní a malé (záporné) $w_{ji} y_j y_i$ nestabilní

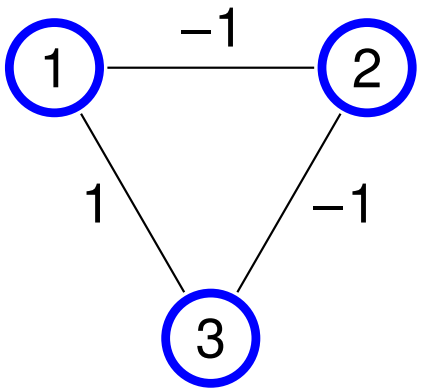
V průběhu výpočtu se energie nezvyšuje: $E(\vec{y}^{(t)}) \geq E(\vec{y}^{(t+1)})$, stav $\vec{y}^{(t^*)}$ odpovídá lokálnímu minimu funkce E.

Energetická plocha



Obr. 3.4: Energetická plocha.

Hopfield - příklad



y_1	y_2	y_3	E
1	1	1	1
1	1	−1	1
1	−1	1	−3
1	−1	−1	1
−1	1	1	1
−1	1	−1	−3
−1	−1	1	1
−1	−1	−1	1

- Hopfieldova síť se třemi neurony
- naučili jsme ji jeden vzor $(1, -1, 1)$ pomocí Hebbova učení (síť se automaticky „naučila“ i vzor $(-1, 1, -1)$)

Energetická funkce - konvergence výpočtu sítě

Pomocí pojmu energie lze snadno dokázat, že výpočet Hopfieldovy sítě vždy zastaví:

- v průběhu výpočtu se energie nezvyšuje: $E(\vec{y}^{(t)}) \geq E(\vec{y}^{(t+1)})$
- pokud dojde v kroku $t+1$ ke změně stavu, pak $E(\vec{y}^{(t)}) > E(\vec{y}^{(t+1)})$
- existuje pouze konečně mnoho stavů sítě: výpočet dosáhne lokálního minima funkce E, ze kterého už se nedostane

Hopfieldova síť - odučování

Při učení podle Hebbova zákona mohou vznikat lokální minima funkce E, tzv. nepravé vzory (*fantomy*), které neodpovídají tréninkovým vzorům.

Fantomy je možné odučovat např. pomocí následujícího pravidla: Mějme fantom $(x_1, \ldots, x_n) \in \{-1, 1\}^n$ a váhy w_{ji}, pak nové váhy w'_{ji} spočítáme pomocí

$$w'_{ji} = w_{ji} - x_i x_j$$

(tj. podobně jako při adaptaci podle Hebbova zákona, ale s opačným znaménkem)

Reprodukce - statistická analýza

Kapacita Hopfieldovy paměti je dána poměrem p/n.

Zde n je počet neuronů a p je počet vzorů.

Předpokládejme, že tréninkové vzory jsou voleny náhodně takto: při volbě $\vec{x}_k$ volím postupně (nezávisle) jednotlivé složky (1 s pravd. $1/2$ a -1 s pravd. $1/2$).

Uvažme konfiguraci W, kterou obdržíme Hebbovským učením na zvolených vzorech.

Označme

$$\beta = \mathbf{P}\left[\vec{x}_k = \vec{y}(W, \vec{x}_k) \text{ pro } k = 1, \ldots, p\right]$$

Pak pro $n \to \infty$ a $p \leq n/(4 \log n)$ dostaneme $\beta \to 1$.

Tj. maximální počet vzorů, které lze věrně uložit do Hopfieldovy paměti je úměrný $n/(4 \log n)$.

Hopfieldova síť - asociace

Problém:

- příliš mnoho vzorů implikuje existenci lokálních minim funkce E, která neodpovídají vzorům (tzv. *fantomy*)
- lokální minima pro vzory mohou dokonce zanikat

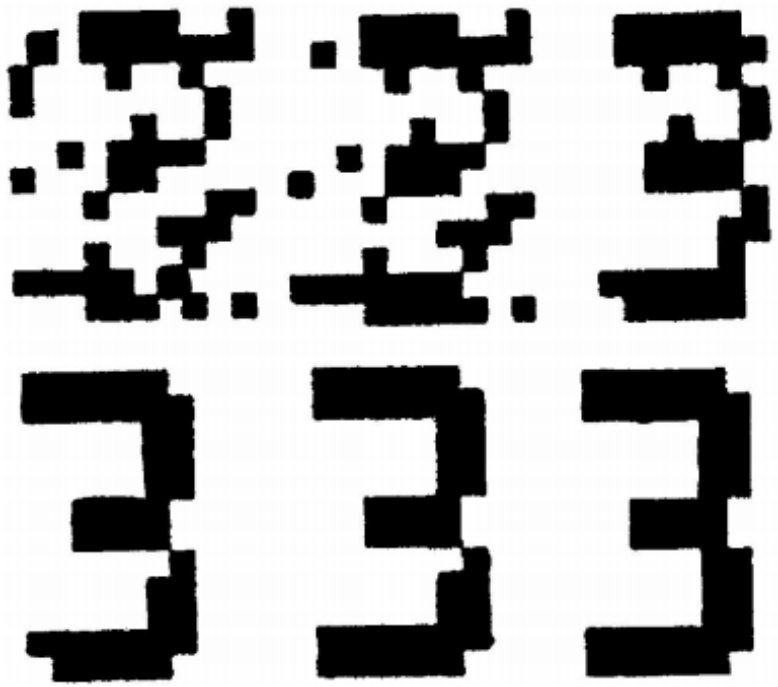
Podrobná analýza ukazuje následující

- Pro $p \leq 0.138n$ tréninkové vzory *zhruba* odpovídají lokálním minimům funkce E
- Pro $p > 0.138n$ lokální minima podobající se vzorům zanikají (ostrá nespojitost v 0.138)
- Pro $p < 0.05n$ energie stavů podobajících se tréninkovým vzorům odpovídají globálním minimům E a fantomy mají ostře větší energii

Tj. pro dobré zapamatování 10 vzorů je potřeba 200 neuronů a tedy 40000 spojů ohodnocených celočíselnými váhami

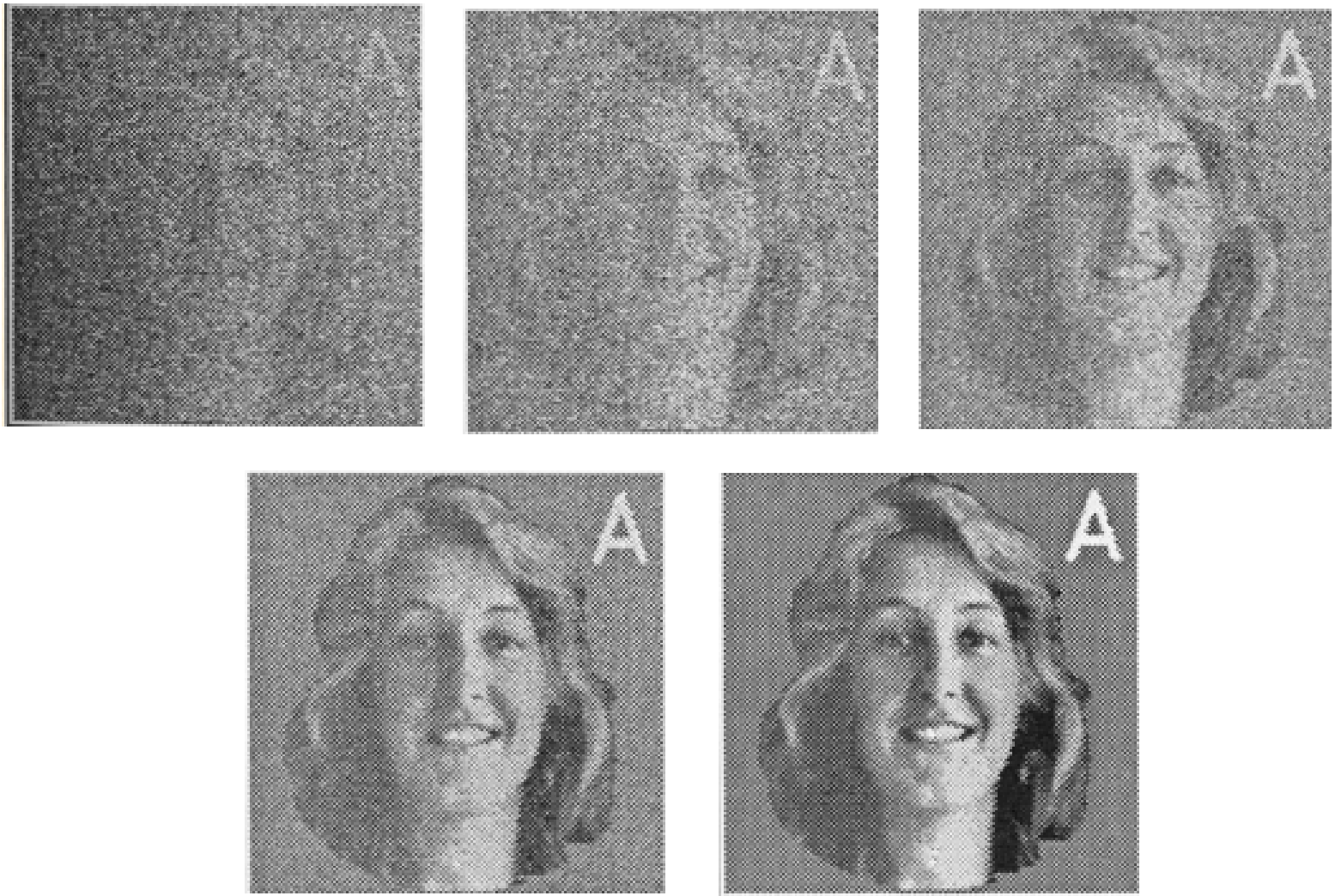
Pozn. Nevýhodou Hopfieldovy sítě je deterministický výpočet, který může skončit v mělkém lokálním minimu E bez možnosti uniknout. Tento problém částečně vyřeší stochastická verze aktivní dynamiky.

Hopfieldova síť - příklad kódování

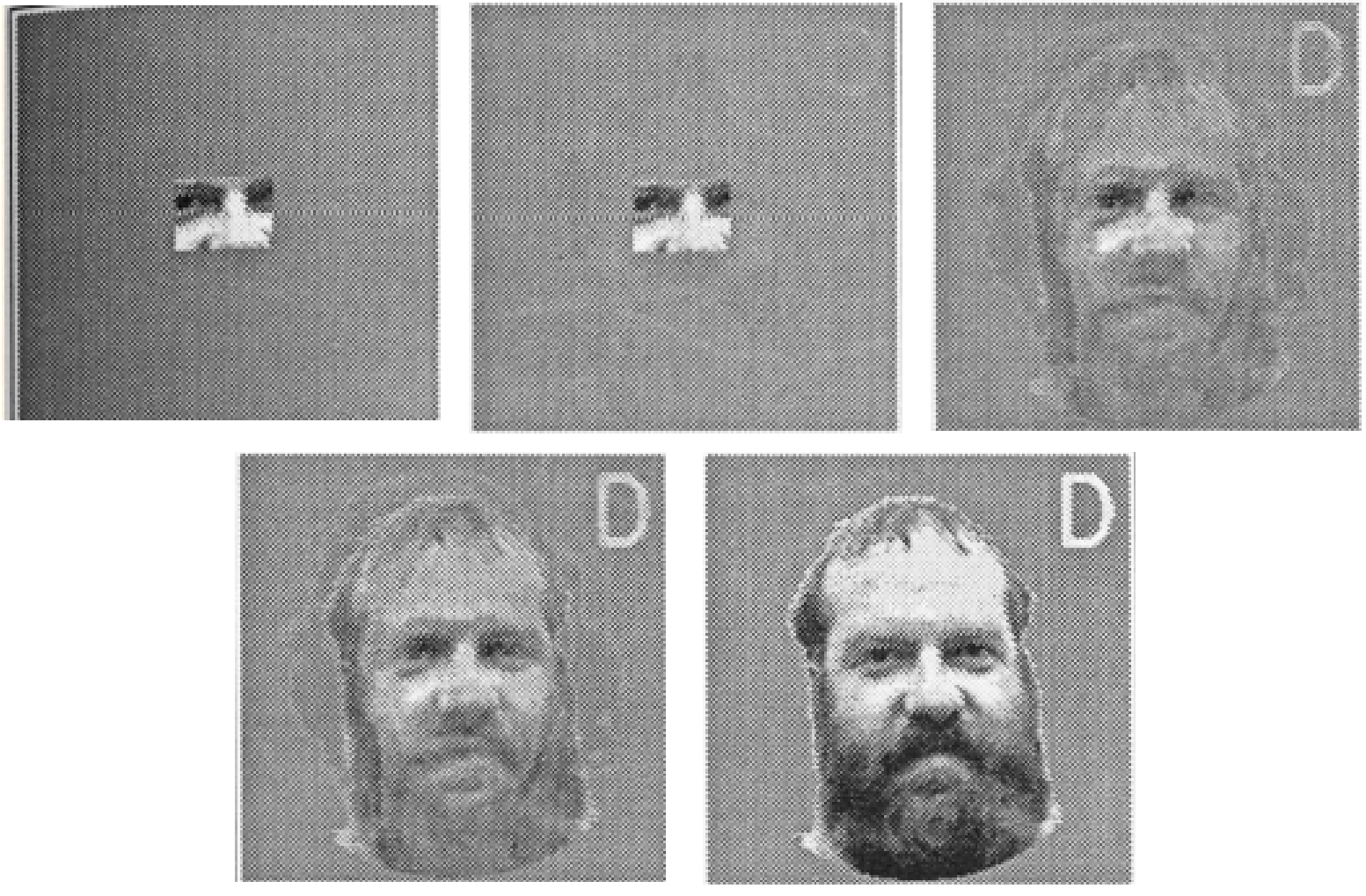


- číslice 12×10 bodů
  (120 neuronů, -1 je bílá a 1 je černá)
- naučeno 8 číslic
- vstup vygenerován ze vzoru 25% šumem
- obrázek ukazuje postup výpočtu Hopfieldovy sítě

Hopfieldova síť - příklad obnovení vzoru



Hopfieldova síť - příklad rekonstrukce vzoru



Hopfieldova síť (pro optimalizační úlohy)

Autoasociativní síť.

Organizační dynamika:

- úplná topologie, tj. každý neuron je spojen s každým
- všechny neurony jsou současně vstupní i výstupní
- označme $\xi_1, \ldots, \xi_n$ vnitřní potenciály a $y_1, \ldots, y_n$ výstupy (stavy) jednotlivých neuronů
- označme w_{ji} *celočíselnou* váhu spoje od neuronu $i \in \{1, \ldots, n\}$ k neuronu $j \in \{1, \ldots, n\}$.
- přepokládáme $w_{jj} = 0$ pro každé $j = 1, \ldots, n$.
- **Nyní:**
  - každý neuron má bias θ_i
  - stavy neuronů jsou z $\{0, 1\}$

Hopfieldova síť (s biasy)

Aktivní dynamika: Iniciálně jsou neurony nastaveny na vstup $\vec{x} = (x_1, \ldots, x_n) \in \{0,1\}^n$ sítě, tedy $y_j^{(0)} = x_j$ pro $j = 1, \ldots, n$.

V kroku $t+1$ aktualizujeme neuron j, t. ž. $j = (t \mod p) + 1$, takto:

nejprve vypočteme vnitřní potenciál

$$\xi_j^{(t)} = \sum_{i=1}^{n} w_{ji} y_i^{(t)} - \theta_j$$

a poté

$$y_j^{(t+1)} = \begin{cases} 1 & \xi_j^{(t)} > 0 \\ y_j^{(t)} & \xi_j^{(t)} = 0 \\ 0 & \xi_j^{(t)} < 0 \end{cases}$$

Hopfieldova síť - aktivní dynamika

Výpočet končí v kroku t^* pokud se síť nachází (poprvé) ve *stabilním* stavu, tj.

$$y_j^{(t^*+n)} = y_j^{(t^*)} \qquad (j = 1, \ldots, n)$$

Energetická funkce E přiřazuje každému stavu sítě $\vec{y} \in \{0,1\}^n$ potenciální energii danou

$$E(\vec{y}) = -\frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{n} w_{ji} y_j y_i + \sum_{i=1}^{n} \theta_i y_i$$

V průběhu výpočtu se energie nezvyšuje: $E(\vec{y}^{(t)}) \geq E(\vec{y}^{(t+1)})$, stav $\vec{y}^{(t^*)}$ odpovídá lokálnímu minimu funkce E.

Věta

Za předpokladu symetrie vah, výpočet Hopfieldovy sítě skončí pro každý vstup.

(Důkaz stejný jako předtím)

Hopfieldova síť a optimalizační úlohy

Optimalizační úloha je zadána množinou přípustných řešení a účelovou funkcí. Cílem je nalézt přípustné řešení, které minimalizuje účelovou funkci U.

Pro mnoho optimalizačních úloh lze nalézt Hopfieldovu síť takovou, že

- minima $E \approx$ přípustná řešení vzhledem k U
- globální minima $E \approx$ řešení minimalizující U

Cílem je nalézt globální minimum funkce E (a tedy i U).

Příklad: multiflop

Cílem je nalézt vektor z $\{0,1\}^n$, který má všechny složky nulové kromě právě jedné.

Definujeme účelovou funkci $U : \{0,1\}^n \to \mathbb{R}$ takto:

$$U(\vec{u}) = \left(\left(\sum_{i=1}^{n} u_i\right) - 1\right)^2$$

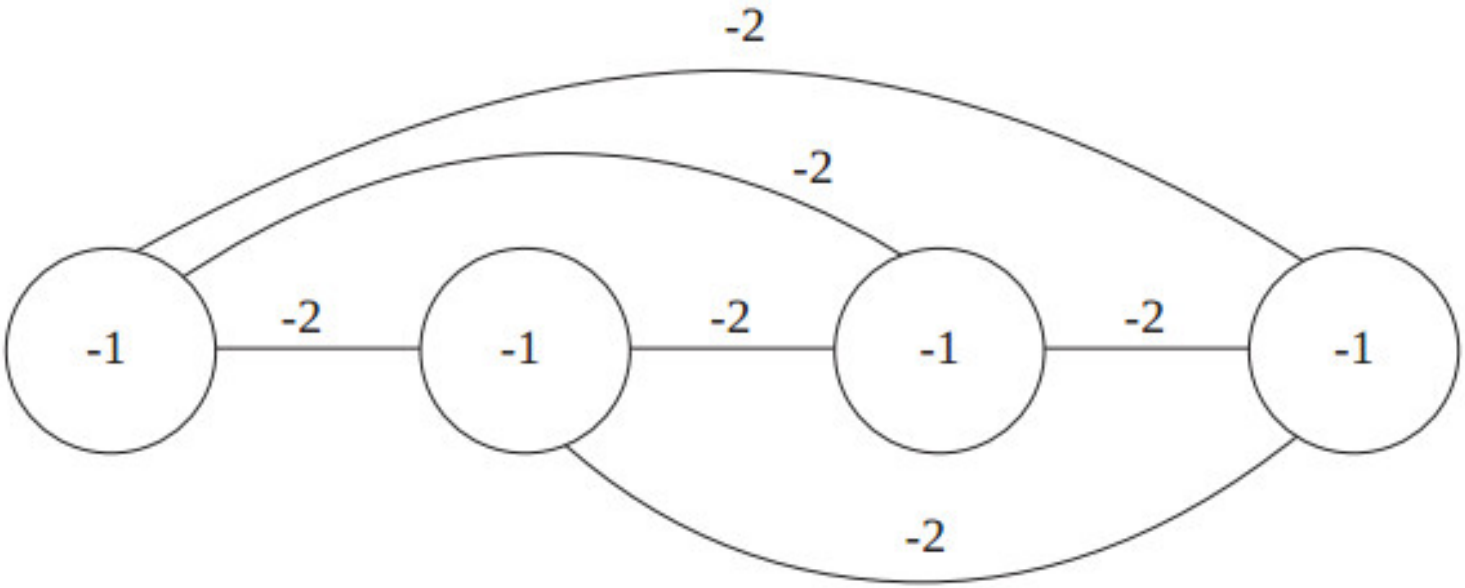
Požadované vektory jsou právě minima této funkce.

Ale

$$U(\vec{u}) = -\frac{1}{2}\sum_{i\neq j}^{n}(-2)u_i u_j + \sum_{i=1}^{n}(-1)u_i + 1$$

a tedy $U(\vec{u}) - 1$ je energetickou funkcí sítě (viz. následující slajd).

Příklad: multiflop (síť)



$$E(\vec{u}) = -\frac{1}{2}\sum_{i\neq j}^{n}(-2)u_i u_j + \sum_{i=1}^{n}(-1)u_i$$

Příklad: n věží

Cílem je rozmístit n věží na šachovnici $n \times n$ tak, aby se vzájemně neohrožovaly.

Definujeme účelovou funkci $U_1 : \{0,1\}^{n \cdot n} \to \mathbb{R}$:

$$U_1(\vec{u}) = \sum_{j=1}^{n} \left(\left(\sum_{i=1}^{n} u_{ji} \right) - 1 \right)^2$$

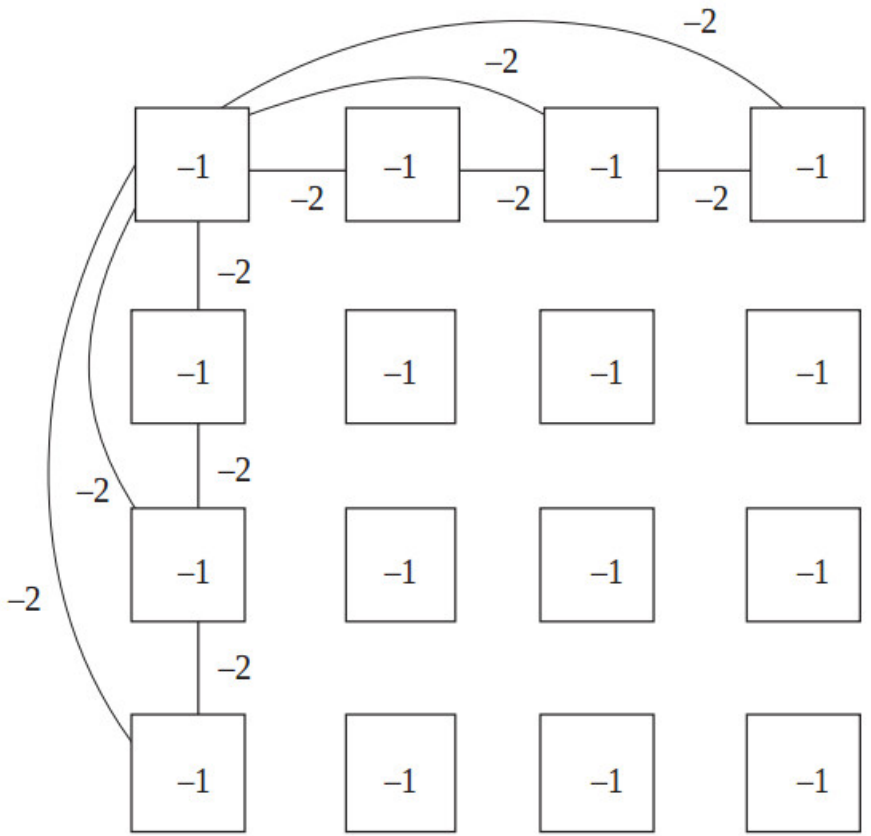
a $U_2 : \{0,1\}^{n \cdot n} \to \mathbb{R}$:

$$U_2(\vec{u}) = \sum_{i=1}^{n} \left(\left(\sum_{j=1}^{n} u_{ji} \right) - 1 \right)^2$$

Požadované vektory jsou právě minima funkce $U = U_1 + U_2$.

Minima U odpovídají stavům s minimální energií v následující síti.

Příklad: n věží (síť)



$$E(\vec{u}) = U(\vec{u}) - 2n$$

(Tento příklad se dá zobecnit na problém obchodního cestujícího)

Hopfieldova síť a lokální minima

Hledáme (globální) minima energie E ....

Problém: není jasné, v jakém stavu začít, abychom dosáhli globálního minima. Síť může skončit v mělkém minimu.

Řešení: V každém stavu umožníme s malou pravděpodobností přechod do stavů s vyšší energií. Tuto pravděpodobnost budeme postupně snižovat. Využijeme dynamiku Boltzmannova stroje ...

„Boltzmannovská“ aktivní dynamika

Aktivní dynamika: Stavy neuronů jsou iniciálně nastaveny na hodnoty z množiny $\{0,1\}$, tj. $y_j^{(0)} \in \{0,1\}$ pro $j \in \{1,\ldots,n\}$.

V kroku $t+1$ aktualizujeme náhodně vybraný neuron $j \in \{1,\ldots,n\}$ takto: nejprve vypočteme vnitřní potenciál

$$\xi_j^{(t)} = \sum_{i=1}^{n} w_{ji} y_i^{(t)} - \theta_j$$

a poté náhodně zvolíme hodnotu $y_j^{(t+1)} \in \{0,1\}$ tak, že

$$\mathbf{P}\left[y_j^{(t+1)} = 1\right] = \sigma\left(\xi_j^{(t)}\right)$$

kde

$$\sigma(\xi) = \frac{1}{1 + e^{-2\xi/T(t)}}$$

Parametr $T(t)$ se nazývá **teplota** v čase t.

Teplota a energie

- Velmi vysoká teplota $T(t)$ znamená, že $\mathbf{P}\left[y_j^{(t+1)} = 1\right] \approx \frac{1}{2}$ a síť se chová téměř (uniformně) náhodně.
- Velmi nízká teplota $T(t)$ znamená, že buď $\mathbf{P}\left[y_j^{(t+1)} = 1\right] \approx 1$ nebo $\mathbf{P}\left[y_j^{(t+1)} = 1\right] \approx 0$ v závislosti na tom, jestli $\xi_j^{(t)} > 0$ nebo $\xi_j^{(t)} < 0$. Potom se sít chová téměř deterministicky (tj. jako v původní aktivní dynamice).

Poznámky:

- Boltzmannovská aktivní dynamika funguje jako deterministická dynamika s náhodným šumem,
- energie $E(\vec{y}) = -\frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{n} w_{ji}y_jy_i + \sum_{i=1}^{n}\theta_i y_i$ se může (s pravděpodobností závislou na teplotě) zvýšit,
- pravděpodobnost přechodů do vyšší energetické hladiny se exponenciálně zmenšuje s velikostí energetického skoku.

Simulované žíhání

Následujícím postupem lze dosáhnout globálního minima funkce E:

- Na začátku výpočtu nastavíme vysokou teplotu $T(t)$
- Teplotu postupně snižujeme, např takto:
  - $T(t) = \eta^t \cdot T(0)$ kde $\eta < 1$ je blízko 1
  - nebo $T(t) = T(0)/\log(1+t)$

Lze dokázat, že při vhodném postupu chlazení dosáhneme globálního minima.

Pozn:

- Tento proces je analogií žíhání používané při výrobě tvrdých kovových materiálů s krystalickou strukturou: materiál se nejprve zahřeje, čímž se poruší vazby mezi atomy, v průběhu následného pomalého chlazení se materiál „usadí“ do stavu s minimální vnitřní energií a s pravidelnou vnitřní strukturou.
- Jedná se také o rozšíření fyzikální motivace Hopfieldovy sítě: orientace magnetů jsou ovlivněny nejen vnitřním a vnějším magnetickým polem, ale také termálními fluktuacemi.