

Aplikace vícevrstvých sítí

- ▶ Diagnóza srdečního infarktu
- ▶ Predikce časových řad
- ▶ NETtalk
- ▶ ALVINN
- ▶ Rozpoznávání směrovacích čísel
- ▶ Kompresa dat

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

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

$$\sigma(\xi) = \frac{1}{1 + e^{-\xi}}$$

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)

Predikce časových řad

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

(hrubá) matematická formulace:

- ▶ Mějme numerickou řadu a_1, a_2, a_3, \dots
(např. hodnota koruny vůči euru po jednotlivých dnech)
- ▶ Chceme odhadnout $a_{\ell+1}$ na základě $a_\ell, a_{\ell-1}, \dots, a_{\ell-k}$

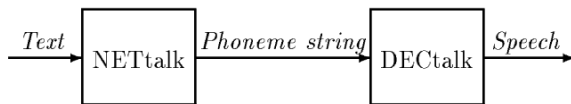
Uvažme síť s k vstupními a jedním výstupním neuronem. Tuto síť natrénujeme na části řady tak, že jí budeme předkládat $a_\ell, a_{\ell-1}, \dots, a_{\ell-k}$ na vstupy a $a_{\ell+1}$ očekávaný výstup.

(V případě měnových kurzů použijeme historii vývoje za poslední roky.)

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

- ▶ další vstupy modelující prostředí v němž se řada vyvíjí
(např. hospodářské parametry státu)
- ▶ výstup se často vrací zpět jako další vstup, čímž lze dosáhnout predikce na více kroků

- ▶ Neuronová síť, která převádí anglicky psaný text na řetěz fonémů (tj. provádí fonetický přepis)
Foném je nejmenší součást zvukové stránky řeči, která má rozlišovací funkci v systému konkrétního jazyka.
- ▶ Výstup potom může být vstupem hlasového syntezátoru.



- ▶ vstup je reprezentován pomocí 29 symbolů (anglická abeceda + čárka, tečka, mezera)
- ▶ mezivýstup (výstup sítě) je posloupnost 26-dimenzionálních artikulačních vektorů

vektor popisuje způsob, jakým je zvuk generován, např. pozice v ústech, přízvuk, intonace, mlaskavost, ...

- ▶ artikulační vektor je poté přeložen na foném

Zkoumali více způsobů překladu:

- ▶ foném s Euklidovskými nejbližším artikulačním vektorem
- ▶ foném jehož artikulační vektor svírá nejmenší úhel s výstupním vektorem

Organizační dynamika:

- ▶ Síť je vícevrstvá.
- ▶ 203 vstupních neuronů
 - ▶ systému se na vstup dává „okno“ obsahující 7 symbolů, čte se prostřední symbol, ostatní dávají kontext
 - ▶ každý znak je kódován unárně (tj. na každý symbol připadá 29 vstupů ($7 \cdot 29 = 203$))
- ▶ 1 až 2 skryté vrstvy čítající až 0 až 120 neuronů

Standardně se tento příklad prezentuje s jednou skrytou vrstvou obsahující 80 neuronů
- ▶ výstupní vrstva obsahuje 26 výstupních neuronů, které představují artikulační vektor

(výstup se nakonec přeloží na foném)

Aktivní dynamika:

aktivační funkce: standardní logistická sigmoida $\sigma(\xi) = \frac{1}{1+e^{-\xi}}$.

Adaptivní dynamika:

- ▶ dva druhy vzorů: Miriam Webster's Pocket Dictionary a dětská neformální řeč
- ▶ gradientní sestup (zpětná propagace)
- ▶ iniciačně váhy nastaveny náhodně uniformně z intervalu $[-0.3, 0.3]$

Vzory: Miriam Webster's Pocket Dictionary

- ▶ celý slovník měl 20012 slov, trénováno na podmnožině o 1000 nejfrekventovanějších slovech
- ▶ slova vybírána náhodně
- ▶ adaptace vždy po jednom slově (tj. rámečkem se projelo slovo, podle kumulovaného gradientu se upravily váhy)

Výsledky pro Miriam Webster's:

- ▶ na tréninkových vzorech dosáhli přesnosti 98% (při 120 skrytých neuronech)
- ▶ při aplikaci takto naučené sítě na celý slovník dostali přesnost 77%
 - ▶ při průběžném učení na celém slovníku dosáhli po jednom průběhu 85%
 - ▶ po pěti průchodech 90%
- ▶ zvyšování počtu neuronů, velikosti rámečku, počtu vrstev vždy vedlo ke zlepšení (byli limitováni výpočetním výkonem cca 2 znaky za vteřinu (rok 1987))

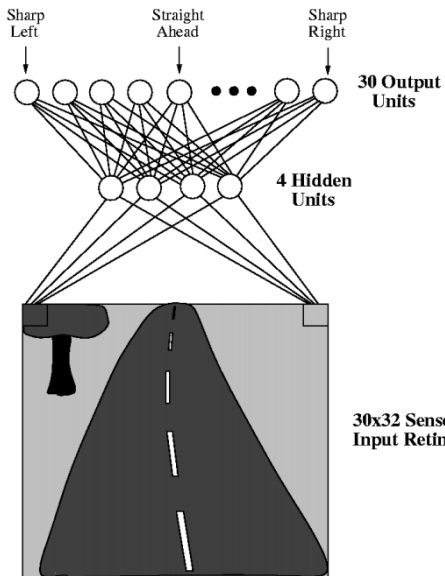
Vzory: Dětská neformální řeč:

- ▶ trénink na 1024 slovech, testováno na dalších 439
- ▶ rámečkem se projížděl celý text, hranice slov byly speciálně vyznačeny
- ▶ problematické, jedno slovo bylo vyslovováno mnoha způsoby

Výsledky pro neformální řeč:

- ▶ na tréninkových vzorech dosáhli 95%, při testu pak 78% (tj. docela dobrá generalizace)
- ▶ testovali odolnost proti narušení: přičtení náhodně vygenerovaných hodnot z $[-0.5, 0.5]$ k váhám se projevilo jen minimálně
- ▶ naučená síť měla většinu vah v průměru 0.8 (téměř všechny byly menší než 2)

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

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.)

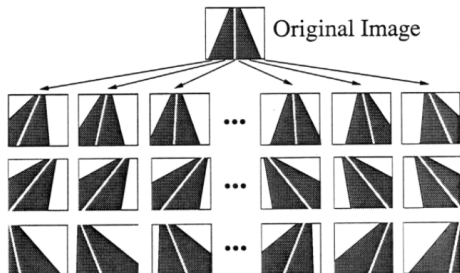
Naivní přístup: brát vstupy z kamery a podle nich adaptovat.

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

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:

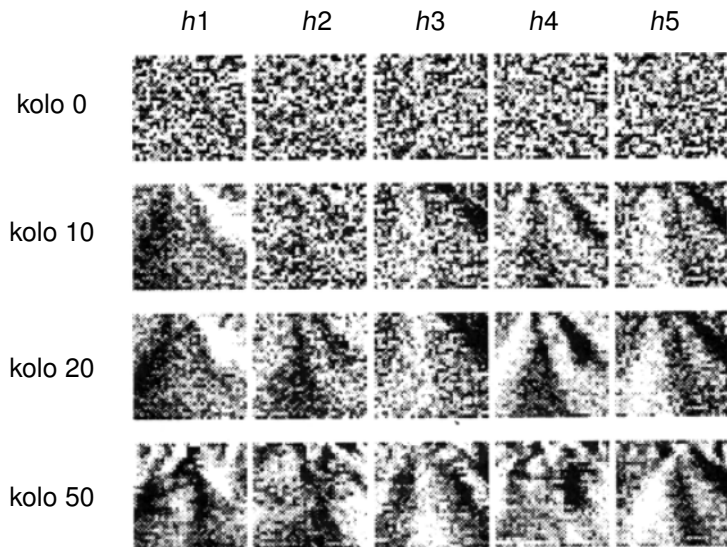
- ▶ systém má buffer 200 obrázků (včetně 15 kopií aktuálního), v každém kole tréninku se trénuje na těchto vzorech
- ▶ po tréninku se sejme nový obraz, udělá se 15 kopií a těmi se nahradí 15 obrázků z bufferu (10 s nejmenší chybou, 5 náhodně)

- ▶ 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 h_1, \dots, h_5 jsou skryté neurony.

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 $[-1, 1]$

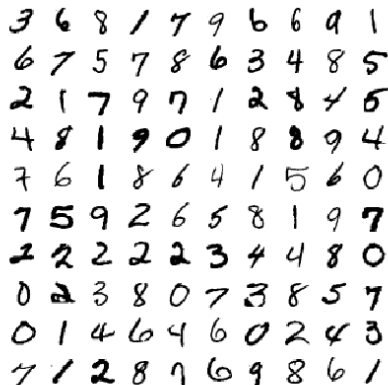


Fig. 4. Size-normalized examples from the MNIST database.

- ▶ výstup: jedna z deseti hodnot

Rozpoznávání číslic

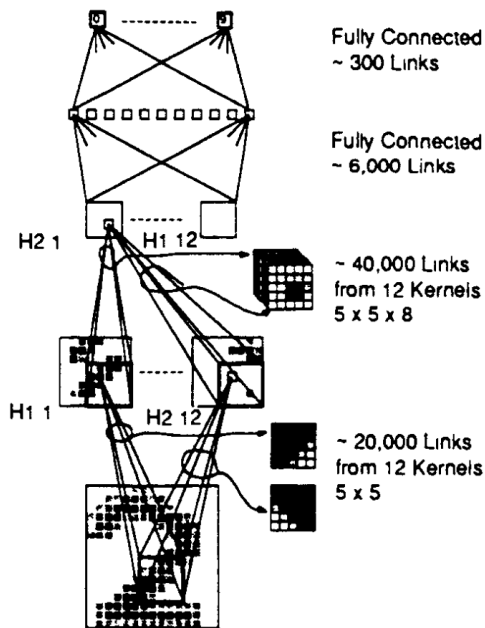
10 Output Units

Layer H3
30 Hidden Units

Layer H2
 $12 \times 16 = 192$
Hidden Units

Layer H1
 $12 \times 64 = 768$
Hidden Units

256 Input Units



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 vyšší 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

Aktivní dynamika: aktivační funkce: hyperbolický tangens

Dvojitá 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

- ▶ 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% odmínutých

Kompresa obrazových dat pro prenos signalu.

- ▶ Pouzita dvouvrstva sit' $n - \frac{n}{4} - n$ (tj. pocet skrytych neuronu je 4x mensi nez pocet vstupnich a vystupnich).
- ▶ Siti byly predkladany obrazy, stejny obraz na vstup i vystup.
- ▶ Naučena sit' potom byla pouzita takto:
 - ▶ Vysilajici vypočte pro daný vstup hodnoty skrytych neuronu
 - ▶ Hodnoty skrytych neuronu jsou odeslany prijemci
 - ▶ Prijemce vypočte hodnoty vystupnich neuronu po dosazeni hodnot skrytych neuronu

Metoda funguje pokud jsou vysilane obrazy podobne treninkovym vzorum.

Da se ukazat, ze tato metoda realizuje PCA na obrazovych datech - tedy nejlepsi moznou redukci dimenze dat (probereme pozdeji)

Kompresa 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

Kompresa 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 bitu na pixel)

Kompresa dat - výsledky



(A)



(B)

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 bitu na pixel)



(C)



(D)