Závěrečná zpráva k projektu z Neuronových sítí

Marie Drábková, Jiří Novotný, Jakub Peschel

17. prosince 2015

Úvod

Našim cílem je vytvořit neuronovou síť, která rozpoznává, kdo vyhrál v piškvorkách 3×3. Vstupem sítě je finální konfigurace hrací plochy (dále endgame). Výstupem je klasifikace, zda vyhrál „křížek“, „kolečko“ nebo hra skončila remízou.

Cíl jsme si rozdělili do tří fází. V poslední fázi by síť měla ideálně klasifikovat bitmapový obrázek hrací plochy.

Data

Téma práce je inspirováno databází UCI Machine Learning [1], ale my jsme se rozhodli klasifikovat i případy výhry „kolečka“ a remízy, které databáze nezahrnovala. Proto jsme si vytvořili vlastní generátor endgames v textovém formátu. Dále jsme vytvořili konvertor textových dat do bitmapových obrázků.

Existuje celkem 958 možných endgames. Z nich v 626 případech vyhrál X, v 316 vyhrál O a v 16 hra skončila remízou. Základní data představuje rozdělení do tréninkových, validačních a testovacích množin v poměru 60:20:20 tak, že v každé množině je stejné procentuální zastoupení vítězství X, O a remíz. Dále jsme data různým způsobem modifikovali, což bude popsáno níže.

Implementace

Rozhodli jsme se pro řešení problému pomocí vícevrstvé sítě a backpropagation. Tuto síť jsme naprogramovali v jazyce C++ za použití návodu z online učebnice „Neural networks and deep learning“ [2]. Pro zachování konzistence jsme používali stejnou terminologii (epocha, batch size, eta,...). Jako aktivační funkci jsme použili sigmoidu optimalizovanou pro rychlost $\frac{x}{1+|x|}$ a její derivaci $\frac{1}{(1+|x|)^2}$. Jako chybovou funkci jsme zvolili kvadratickou. Ze začátku jsme váhy inicializovali pomocí normálního rozdělení $N(0,1)$. Abychom zrychlili učení, přešli jsme na normální rozložení $N\left(0, \frac{1}{\sqrt{n}}\right)$, kde n je počet vstupních neuronů dané vrstvy.

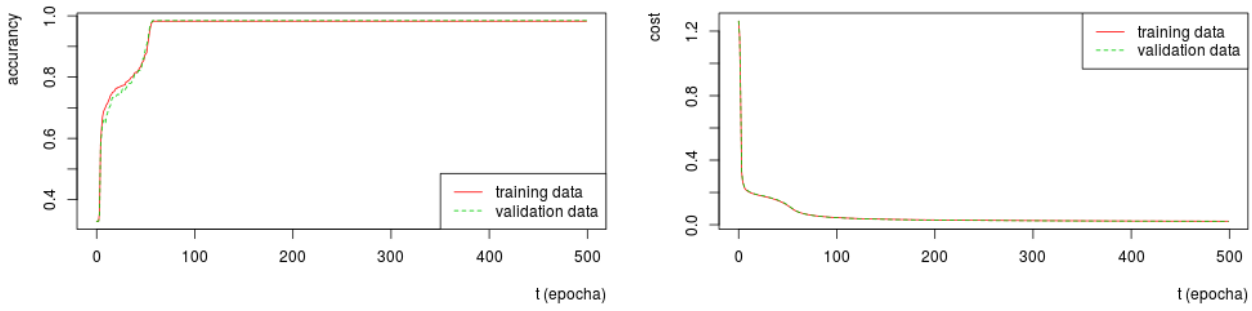
Učení

Textový vstup

Nejdřív jsme síť naučili na jednoduchém textovém vstupu, který představuje dvourozměrné pole 3×3. Jeho prvky jsou hodnoty z množiny $\{-1, 0, 1\}$, které reprezentují postupně „kolečko“, prázdné pole a „křížek“.

Jako první jsme vyzkoušeli velikost sítě (9,9,3), rychlost učení 0,05, 500 epoch. Síť se naučila rozpoznávat hry s přesností 98,4 % za cca 60 epoch. Dále byla přesnost stabilní. Ačkoliv cost function nadále klesala, síť se nepřeučovala, viz obrázek 1. Matice zmatenosti – tabulka 1 (R znamená remíza) ukazuje, že síť dobře rozpoznává vítězství X a O, ale má problém poznat remízu, protože v tréninkových datech se konfigurace znamenající remízu vyskytují příliš málo a síť se je nezvládne naučit.

[1] `https://archive.ics.uci.edu/ml/datasets/Tic-Tac-Toe+Endgame`
[2] `http://neuralnetworksanddeeplearning.com/`



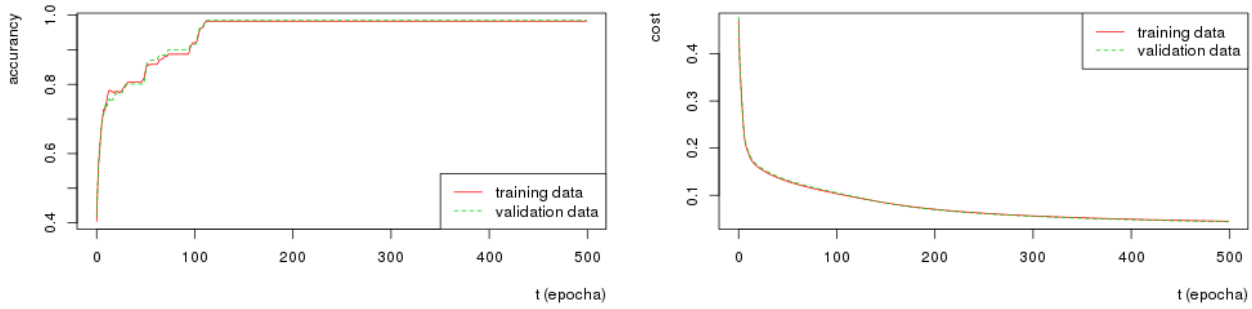
(a) Accurancy (b) Cost

Obrázek 1: Textový vstup, síť (9,9,3)

		Actual		
		X	O	R
Predicted	X	125	0	3
	O	0	63	0
	R	0	0	0

Tabulka 1: Matice zmatenosti pro síť (9,9,3) a textový vstup

Poté jsme postupně zkoušeli zmenšovat velikost sítě na (9,3,3) až (9,3). I síť bez vnitřní vrstvy byla schopna dávat stejné výsledky jako výše popsaná síť (9,9,3). Pouze průběh učení byl jiný, viz obrázek 2. Matice zmatenosti vypadá stejně, viz tabulka 1.



(a) Accurancy (b) Cost

Obrázek 2: Textový vstup, síť (9,3)

Po podrobnějším zkoumání vah jsme dospěli k závěru, že síť považuje za výherce X právě tehdy, když suma na hrací ploše je 1 a pokud je suma 0, považuje za výherce O. Jenomže pokud hrací plocha představuje remízu, suma je taky jedna a síť tuto situaci klasifikuje jako výhru X.

Pokus o řešení špatně interpretovaných remíz

Nejdříve jsme chtěli síti umožnit větší šanci pro naučení remízových stavů, a proto jsme zduplikovali remízové endgames v tréninkové množině (dále duplikovaná data). Tato data jsme použili pro trénink sítě velikosti (9,3), ale nic nového se nenaučila.

Větší síť (9,9,3) zvýšila svou přesnost na testovací množině z 98,4 % na 99,5 %. Z matice zmatenosti 2 je vidět, že některé případy remízy dokážeme rozpoznat. Pro ilustraci uvádíme tabulky zmatenosti trénovací (tabulka 3

i validační (tabulka 4) množiny při celkové přesnosti na testovacích datech 99,5 %.

Pak jsme zmenšili rychlost učení z 0,05 na 0,01 a zvětšili počet epoch z 500 na 1000. Touto úpravou jsme dosáhli přesnosti 100 % na testovacích datech, ale tento výsledek není stabilní. Při replikaci pokusu jsme dosahovali různých výsledků mezi 97 % a 100 %.

		Actual		
		X	O	R
Predicted	X	125	0	1
	O	0	63	0
	R	0	0	2

Tabulka 2: Matice zmatenosti pro síť (9,9,3) a textový vstup s opakováním na tréninkové množině

		Actual		
		X	O	R
Predicted	X	374	0	0
	O	0	190	0
	R	2	0	270

Tabulka 3: Matice zmatenosti trénovací množiny pro síť (9,9,3) a textový vstup s opakováním na tréninkové množině

		Actual		
		X	O	R
Predicted	X	125	0	2
	O	0	63	0
	R	0	0	1

Tabulka 4: Matice zmatenosti validační množiny pro síť (9,9,3) a textový vstup s opakováním na tréninkové množině

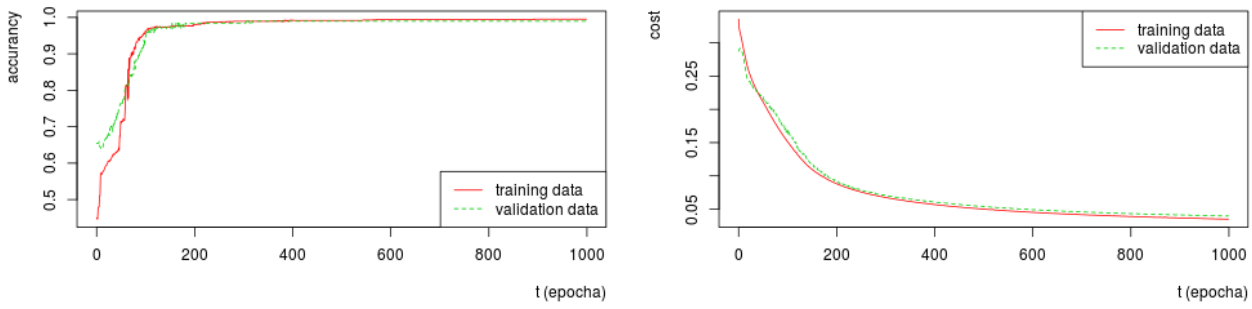
Grafický vstup 11×11

V druhé fázi jsme síti předložili černobílý bitmapový obrázek o velikosti 11×11 pixelů (viz obrázek 4a). Na základních datech je výsledek stejný jako v případě textového vstupu, síť nerozpoznává remízu.

Hráli jsme si s různým nastavením sítě, ale přesnost se nijak nezlepšovala. Nejlepšího výsledku jsme dosáhli použitím duplikovaných dat popsaných v předchozí sekci a dosáhli tak přesnosti 99,5 %, viz obrázek 3.

Mezi neúspěšné pokusy patří např.:

- Vyrovnali jsme množství jednotlivých kategorií. Tzn. vzali z trénovací množiny 300 výher X a zbytek znásobili, aby i počet výher O a remíz byl 300.
- Přidali jsme data tak, že jsme původní data rozšířili o konfigurace takové, kde jsme v původních datech zaměnili X a O. Prakticky to znamená přidání endgames, kdy začínal O.



(a) Accurancy

(b) Cost

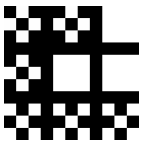
Obrázek 3: Grafický vstup 11×11, síť (121,9,3), duplikovaná data

Grafický vstup 14×14

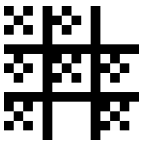
Dále jsme se rozhodli vyzkoušet síť na složitějším vstupu, kdy křížek nebo kolečko o velikosti 3×3 pixely vkládáme náhodně do políčka o velikosti 4×4 pixely, viz obrázek 4b. To nám umožňuje vytvořit až 4^9 obrázků pro každý endgame.

I síť (196,3) dává na základních datech stále stejný výsledek jako textová síť, tedy přesnost 98,4 % kvůli nerozpoznávání remíz.

Snažili jsme se problém rozpoznávání remíz vyřešit podobným způsobem jako v předchozích případech (duplikovaná data), ale nedokázali jsme najít vhodné parametry sítě. Síť se už nedokázala v rozumném čase data naučit. Takto jsme dosáhli nejlepší přesnosti cca 86 %. Věříme, že bychom dokázali přesnost zlepšit, ale výpočet byl příliš zdlouhavý.



(a) 11×11



(b) 14×14

Obrázek 4: Vzorový grafický vstup

Přínosy členů týmu

Marie Drábková vytvořila generátor endgames a zabývala se rozdělením dat do tréninkových, validačních a testovacích množin. Jiří Novotný se postaral o vlastní implementaci sítě. Jakub Peschel vytvořil konvertor textových dat do bitmapových obrázků. Učení sítě jsme prováděli společně ve skupině. Závěrečnou zprávu napsali Jiří Novotný a Marie Drábková.

Závěr

Snažili jsme se vytvořit klasifikátor výherce piškvorek. Vytvořili jsme obecnou implementaci vícevrstvé neuronové sítě, generátor endgames a konvertor do bitmapových obrázků.

Síť jsme nejprve učili na textových datech, kde jsme úpravou dat dosáhli nejlepší přesnosti 99,5 %, avšak síť má problémy s rozpoznáváním rozlišováním výhry X a remízy.

Dále jsme síť učili na grafickém vstupu 11×11 pixelů. Dosáhli jsme stejného výsledku 99,5 %.

Pro grafický vstup 14×14 pixelů se nám už nepodařilo aplikovat stejný princip duplikace dat. Nejlepší dosaženou přesností je tedy 98,4 %.

Výsledné neuronové sítě neměli problém s interpretací vstupu, jak textového, tak grafického. Největším problémem bylo odlišení remíz od výher X, ale i tak síť dosahuje dobrých výsledků, protože se plete pouze v jednotkách případů.

Použití vícevrstvé sítě pro řešení problému piškvorek zřejmě není nejvhodnější řešení, protože zanedbává prostorové rozložení dat. Domníváme se, že by bylo vhodnější použít konvoluční síť.