

# KIV/PRO

## ROZPOZNÁVÁNÍ POZNÁVACÍCH ZNAČEK S POUŽITÍM ZPĚTNÉ PROPAGACE, NEURÁLNÍCH SÍTÍ A GENETICKÝCH ALGORITMŮ

# Obsah

<b>1</b>	<b>Abstrakt</b>	<b>2</b>
<b>2</b>	<b>Úvod</b>	<b>2</b>
2.1	Neuronová síť . . . . .	3
<b>3</b>	<b>Předzpracování snímku</b>	<b>3</b>
<b>4</b>	<b>Neuronová síť</b>	<b>5</b>
4.1	Inicializace sítě . . . . .	5
4.2	Activační funkce . . . . .	5
<b>5</b>	<b>Genetický algoritmus</b>	<b>5</b>
5.1	Reprezentace, populace a inicializace . . . . .	5
5.2	Fitness funkce . . . . .	6
5.3	Nová generace, křížení a mutace . . . . .	6
<b>6</b>	<b>Výsledky</b>	<b>7</b>

# 1 Abstrakt

Účelem této práce je zhodnocení efektivity a hledání optimálních parametrů neuronové sítě s využitím genetických algoritmů. Pro zpracování obrázků byly použity metody pro konverzi do šedotónu, top-hat a binární morfologické transformace, dále metoda prahování Otsu a binární projekce.

Výsledky ukazují že neurální síť optimalizovaná genetickými algoritmy se zpětnou propagací vyžaduje průměrně 2230 cyklů pro natrénování na daných mezích. Tento způsob je tedy o 36,83% rychlejší než generická síť bez použité optimalizace genetickým algoritmem.

# 2 Úvod

Automatický rozpoznávací systém je určen jako náhrada za personální vyhodnocování v místech, kde je třeba znalostí, rozhodování a zkušeností. V našem případě se jedná o systém optického rozpoznávání znaků, který bez pomoci člověka dokáže rozpoznat a interpretovat znaky získané z fotografií poznávacích značek automobilů. Pro realizaci takového systému se nabízejí metody porovnávající vzory (SVM), neuronové sítě a podobně.

K nejpopulárnějším patří zmíněné neuronové sítě, které se pokouší simulovat biologické vlastnosti neuronů mozku. Jednotlivé neurony jsou navzájem propojeny s různými váhami tyto synapse v jistém smyslu uchovávají informaci, která se dá upravit trénováním. Jedním ze způsobů trénování neuronové sítě je zpětná propagace, kde je síti představována trénovací množina a vypočtené chyby při rozpoznávání se zpětně přenášejí synapsemi a upravují jejich váhy tak, aby příště byla chyba menší. Opakováním tohoto procesu se síť postupně učí rozpoznávat.

## 2.1 Neuronová síť

Rychlost trénování sítě je ovlivněna rychlostí učení a jeho momentem. Tyto dva atributy jsou úzce vázány na konkrétní síť a její použití. Optimální parametry nelze určit obecně. Atributy určují jak moc je síť náchylná k uvíznutí v lokálním optimu (nedosažení globálního optima). Rychlost učení při zpětné propagaci silně ovlivňuje také počet skrytých (vnitřních) neuronů sítě. Větší počet těchto neuronů umožňuje síti naučit se více informací za cenu delšího trénování. Ovšem příliš velký počet neuronů může vést k tzv. přeučení. Síť by v takovém případě nedokázala dobře zobecňovat a rozpoznávala by dobře pouze trénovací množinu a nic jiného.

Aby jsme získali optimální síť potřebujeme najít způsob jak určit tyto parametry. Pro určení optimálních hodnot lze použít metodu pokus-omyl, kdy náhodně nastavujeme parametry sítě a zkoušíme jejich úspěšnost. Tato metoda je ale časově náročná a vyžaduje interakci člověka. Pro tento účel využijeme genetického algoritmu, kde se využívá přirozeného výběru. Vytvořením vektorů s náhodnými parametry sítě, které budou mutovat a křížit se navzájem. V každé generaci proběhne křížení a odstraní se nejslabší prvky generace. Postupným vyřazováním slabých kombinací získáme set optimálních nastavení.

## 3 Předzpracování snímku

V tomto kroku potřebujeme získat informace o vlastní poznávací značce z daného snímku.

Postup předzpracování:

- Změna rozměrů snímku na unifikovanou výšku 768px se zachováním poměru stran.
- Převod snímku do šedotónu s ohledem na vnímání člověka. Jas pixelu získáme jako:

$$I_Y = 0.30f_R + 0.59f_G + 0.11f_B$$

kde  $f$  jsou jednotlivé barevné složky.

- Odstranění reflexí pomocí top-hat[1] transformace.
- Prahováním metodou Otsu[3] získáme kandidáty pixelů znaků na poznávací značce (binární obraz viz levá část obr. 1).

- Provedeme operaci zavírání (eroze + dilatace maskou  $2 \times 2$ ) pro odstranění malých defektů ve znacích (osamocené pixely, "praskliny", jemné výčnělky, ...). Dále provedeme erozi (maskou  $3 \times 3$ ) kvůli ztenčení znaků po operaci zavírání.
- Vertikální projekci určíme nejširší pás ve snímku viz obr.1 jako oblast příslušící znakům a ořízneme ze snímku přebytečné části.



Obrázek 1: Vertikální projekce.

- Pro rozlišení jednotlivých znaků využijeme metody connected-component labeling z teorie grafů. Metoda nám poskytne všechny souvislé komponenty. Jednotlivé komponenty jsou dále separovány do segmentů ve snímku pomocí dat z jejich horizontálních projekcí (podobně jako u vertikální projekce). Správnost segmentace lze ověřit porovnáním poměrů výšky a šířky jednotlivých segmentů a jejich minimálních velikostí.
- Kvůli zpracování jednotlivých znaků neuronovou sítí je nutné změnit rozlišení (rozměry) každého segmentu znaku, tak aby odpovídalo vstupu naší neuronové sítě. Rozměry  $12 \times 9$  pixelů odpovídá 108 neuronům ve vstupní vrstvě sítě. Každý pixel náleží jednomu neuronu.
- Aby hodnoty vstupu do sítě měli jednotný tvar, provedeme normalizaci vstupu převedením úrovně šedi na hodnotu intervalu od 0 do 1.

## 4 Neuronová síť

### 4.1 Inicializace sítě

V našem případě byla použita třívrstvá síť skládající se ze vstupní, skryté a výstupní vrstvy. Vstupní vrstva obsahuje 109 neuronů (108 pixelů + 1 ladící "bias neuron"). Skrytá vrstva bude obsahovat počet neuronů získaný jako výsledek genetického algoritmu. Výstupní vrstva má 36 neuronů pro jednotlivé znaky (A-Z + 0-9). Inicializaci vah vstupní vrstvy provedeme podle návrhu Nguyen-Widrow[4]. Váhy mezi skrytou a výstupní vrstvou vygenerujeme náhodné v rozmezí hodnot od  $-0.5$  do  $0.5$ .

### 4.2 Aktivační funkce

Jako aktivační funkce neuronu použijeme standardní funkci binárního sigmoidu[2] (sigmoid s minimem 0 a maximem 1). Výstup z neuronu je tedy hodnotou této funkce v závislosti na jeho vstupu.

## 5 Genetický algoritmus

Genetický algoritmus použijeme pro získání parametrů rychlosti, momentu učení a počtu neuronů ve skryté vrstvě sítě. Trénování proběhne 1600 krát pro každý prvek dané generace parametrů.

### 5.1 Reprezentace, populace a inicializace

Složky prvku (vektoru) jsou rychlost učení, moment a počet skrytých neuronů. Rychlost a moment budou reprezentovány jako reálné hodnoty od 0 do 1 a počet neuronů jako celé číslo. Jednotlivé složky převedeme do binární podoby. Počet neuronů bude 8-bitové číslo. Optimální rychlost učení pro nejlepší zobecňování se podle Wison a Martinez[6] nachází pod hranicí 0.005. Budeme tedy rychlost a moment jako 9-bitová čísla. Velikost 9-bitů nám poskytují 512 hodnot a naše výsledné hodnoty můžou tedy nabývat velikostí od  $\frac{1}{512}$  až  $\frac{512}{512}$ . Serializací složek získáme 26-bitovou reprezentaci jednoho prvku.

Populace jedné generace bude limitována na 50 prvků. V každé generaci vytvoříme nové potomky křížením a odstraníme přebytečný počet nejslabších prvků (podle fitness funkce viz 5.2) tak, aby populace čítala vždy 50

prvků. Inicializaci první generace provedeme pomocí generování náhodného 26-bitového čísla pro každý prvek.

## 5.2 Fitness funkce

Fitness funkce slouží k určení, které prvky dané generace budeme považovat za silné. Pro výpočet funkce využijeme statistického rozptylu (střední kvadratické odchylky) MSE. Každý prvek generace se použil pro 1600 trénovacích cyklů sítě proto pro výpočet hodnoty fitness funkce použijeme průměrnou MSE dle vztahů:

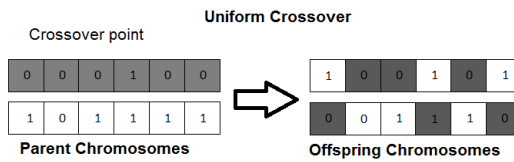
$$fitness_i = max(MSE) - \frac{\sum_{j=1}^{1600} MSE_j}{1600}$$

$$fitness_{in} = \frac{fitness_i}{max(fitness_i)}$$

$fitness_{in}$  představuje normalizovanou hodnotu z intervalu od 0 do 1.

## 5.3 Nová generace, křížení a mutace

Pro každý prvek generace máme nyní spočtenou jeho fitness hodnotu. Náhodně vybereme 10 kandidátů na křížení ze všech prvků vygenerování náhodného čísla od 0 do 1. Prvek s nejbližší hodnotou fitness funkce použijeme pro křížení. Postup opakujeme 10 krát a po dvojicích křížíme. Tímto způsobem docílíme upřednostnění extrémů v rámci shluků prvků s podobnou hodnotou fitness funkce (pokud se takové shluky vyskytnou).



Obrázek 2: Uniformní křížení.

Ke křížení použijeme uniformní metodu, kde každá bitová pozice potomka je vybrána z jednoho nebo druhého rodiče (s 50% pravděpodobností) viz obr. 2.

Mutaci zajistíme jako náhodné překlopení bitu s pravděpodobností 10% pro každý bit potomka.

Prvky generace seřadíme podle jejich fitness funkce připojíme nové prvky vzniklé křížením a odstraníme prvky s nejnižší hodnotou fitness funkce tak, aby populace měla opět 50 prvků.

Genetický algoritmus ukončíme po dosažení 150 generace prvků. Pro naši neuronovou síť použijeme parametry prvku s nejvyšší hodnotou fitness funkce z poslední generace.

## 6 Výsledky

Běh procesu optimalizace genetickým algoritmem trval 26 hodin a vybraný prvek obsahoval parametry dle tab. 1.

Parametr	Bitová reprezentace	Hodnota
Počet skrytých neuronů	11111100	252
Rychlost učení	000011001	0,050781
Moment učení	111111000	0,986328

Tabulka 1: Parametry neuronové sítě optimalizované genetickým algoritmem.

Parametr	Hodnota	cm
Počet skrytých neuronů	72	Trénování výsledné sítě probíhalo dokud hodnota MSE neklesla pod hranici 0.00001. Pro natrénování sítě s parametry, které byly optimalizovány genetickým algoritmem, bylo potřeba průměrně 2230
Rychlost učení	0,2	trénovacích cyklů oproti 3725 cyklům pro obecnou síť [5] (viz tab. 2).
Moment učení	0,9	

Tabulka 2: Parametry neoptimalizované neuronové sítě.

Data	Optimalizovaná	Neoptimalizovaná
Rozpoznání jednoho znaku	97,18%	96,94%
Čas trénovacího cyklu	2,50ms	1,38ms
Celkové rozpoznání značky	85,97%	84,62%

Tabulka 3: Porovnání optimalizované a neoptimalizované sítě.

Z výsledků vyplývá že doba trénovacího cyklu je pro naši síť delší což je způsobeno použitím většího počtu neuronů ve skryté vrstvě sítě. Ovšem díky snížení počtu potřebných trénovacích cyklů došlo k celkovému zrychlení trénovacího procesu sítě. Tedy i snížení výpočetních nároků bez ztráty na rozpoznávací schopnosti sítě.



## Reference

- [1] Woods RE Gonzalez RC. *Digital Image Processing*. Pearson Education New Jersey, 2008.
- [2] Olgac AV Karlik B. Performance analysis of various activation functions in generalized mlp architectures of neural networks. *International Journal of Artificial Intelligence and Expert Systems*, 2011.
- [3] Otsu N. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 1979.
- [4] Widrow B Nguyen D. Improving the learning speed of 2-layer neural network by choosing initial values of the adaptive weights. In *IEEE International Joint Conference on Neural Networks*, 1992.
- [5] Karsoliya S. Approximating number of hidden layer neurons in multiple hidden layer bpnn architecture. *International Journal of Engineering Trends and Technology*, 2012.
- [6] Martinez TR Wilson DR. The need for small learning rates on large problems. In *Proceedings. IJCNN'01. International Joint Conference*, 2001.