PARALLEL NEURAL NETWORK TRAINING FOR SPEECH RECOGNITION

Stanislav Kontár, Master Degree Programme (5)
Dept. of Computer Graphics and Multimedia, FIT, BUT
E-mail: xkonta00@stud.fit.vutbr.cz

Supervised by: Ing. Petr Schwarz

ABSTRACT

In speech recognition forward multi-layer neural networks are used as clasificators for phoneme recognisers, for speech parametrization, in language models and for language or speaker recognition.

This paper discusses possibilities of training forward multi-layer neural networks using parallel algoritms. Need for parallel training of neural networks for speech recognition is caused by huge quantity of training data used in these tasks.

1 ÚVOD

Umělé neuronové sítě jsou matematické modely umělé inteligence, které vycházejí ze stavby biologického mozku. Skládají se z jednoduchých jednotek nazývaných neurony, které mohou být spojené do složitých struktur.

Dopředné vícevrstvé neuronové sítě mají široké možnosti použití pro klasifikaci, rozpoznávání vzorů a obrazů, predikci, redukci dimenzí nebo pro řízení.

Nejčastěji jsou využívány pro rozpoznávání a klasifikaci, protože navrhnout klasický algoritmus pro stejnou úlohu bývá příliš složité. Obvyklým důvodem také je, že není známa korespondence mezi vstupními a výstupními veličinami.

2 NEURONOVÉ SÍTĚ V ROZPOZNÁVÁNÍ ŘEČI

V rozpoznávání řeči jsou používány nejčastěji dva druhy klasifikátorů, založených na skrytých Markovových modelech (HMM) nebo na neuronových sítích. Pro každý vstupní vektor požadujeme jako odpověď zařazení tohoto vektoru do správné třídy.

Pro trénování klasifikátorů je k dispozici obrovské množství vstupních dat. Pro některé úlohy je možné získat až 2000 hodin řeči (720 milionů vektorů).

Trénování klasifikátorů využívajících skryté Markovovy modely je možné snadno paralelizovat. Sběr statistik pro odhad nových modelů je možné rozdělit a provádět odděleně na více procesorech.

Neuronové sítě mají lepší úspěšnost i s menším množstvím parametrů, umožňují pracovat s větší dimenzionalitou vektorů a tyto vektory nemusí být dekorelovány. U neuronových sítí je však paralelizace složitější: Váhy neuronové sítě se průběžně mění, což způsobuje, že není možné neuronovou síť učit nezávisle v paralelních větvích.

Běžná neuronová síť trénovaná pro potřeby rozpoznávání řeči má 253 vstupů, 1500 skrytých a 129 výstupních neuronů. Její trénování na 40 hodinách řeči (14,5 milionu vektorů) trvá na procesoru Pentium 4 2.8GHz 84 hodin. Pokud by se použilo 2000 hodin řeči, trénování by trvalo cca 5 měsíců. Z toho vyplývá potřeba trénování urychlit – nejlépe rozdělit zátěž mezi více procesorů.

3 PARALELIZACE ALGORITMU BACK-PROPAGATION

Algoritmus backpropagation pro učení neuronové sítě není z principu příliš dobře paralelizovatelný, protože při výpočtech je potřeba znát hodnoty z téměř celé sítě (forward propagation) a ihned po propagaci jednoho vektoru je potřeba provést zpětné šíření chyby (back propagation).

Proto se v existujících paralelně trénovaných neuronových sítích používají dva přístupy, které obcházejí principiální potřeby backpropagation algoritmu.

Pro potřebu rozpoznávání řeči je vhodnější přístup tzv. „rozdělení dat“. Tento přístup dovoluje trénovat neuronovou síť na každém počítači pomocí části trénovacích dat. Obchází se potřeba měnit váhy po každém trénovacím vektoru. Experimentálně bylo zjištěno, že pokud se provede změna vah až po daném množství trénovacích vektorů (nazývaném bunch-size), je vliv na přesnost neuronové sítě zanedbatelný (přesnost klesne o méně než 0,5%). Velikost bunch-size je možné nastavit pro každou úlohu jinak, ve zpracování řeči se za bezpečnou hodnotu považuje 1000 vektorů.

Pokud je určena hodnota bunch-size, každý procesor bude počítat matici změn pro část trénovací množiny o velikosti $\frac{bunchsize}{N}$, kde N je počet procesorů. Jeden procesor poté matice změn sečte, provede změnu vah a nové váhy rozešle ostatním.

Výhodou tohoto přístupu je, že se jedná v podstatě o nezměněný algoritmus backpropagation, tudíž dovoluje snadnou kontrolu správnosti.

4 IMPLEMENTACE

Úkolem bylo implementovat paralelní neuronovou síť pracující na výpočetním systému BLADE. Jedná se o svazek procesorů spojených gigabitovým ethernetem. Další podmínkou bylo, aby neuronová síť byla schopná pracovat s formáty dat používanými ve zpracování řeči a aby bylo možné před síť zařadit libovolnou transformaci (např. normalizaci).

První verze paralelní neuronové sítě SNet v1.0 byla napsána v jazyce C a postavena na STK toolkitu vyvinutém ve skupině zpracování řeči na FIT VUT v Brně. Tato knihovna se stará o kompatibilitu s ostatním software používaným pro zpracování řeči.

Pro co nejrychlejší výpočty maticových operací je použitá knihovna BLAS (Basic Linear Algebra Subprograms) a implementace rychlých exponenciál, která je založená na

manipulaci s exponentem a sestává pouze z jednoho násobení a dvou sčítání.

SNet v1.0 využívá architektury server-client a spojení pomocí TCP/IP protokolu. Komunikace probíhá synchronně, každý procesor spočítá svou část, odešle matice změn, počká na nové váhy a poté pokračuje ve výpočtu. Server je vícevláknový a kromě synchronizace vah plní také funkci klienta – tj. také se podílí na zpracování části úlohy.

5 VÝSLEDKY

SNet v1.0 byl testován na výpočetním systému BLADE na 1, 3, 5 a 10 procesorech. Výpočet na jednom procesoru je brán jako referenční.

Počet procesorů	Relativní urychlení	Přesnost
1	1x	58.03%
3	2,1x	57.90%
5	2,5x	57.72%
10	1,4x	57.51%

Nárust výkonu je poměrně brzy omezen. Úzkým hrdlem je synchronizace vah. Pokud se některý procesor zpozdí, všechny ostatní na něj musí počkat. Čím větší je počet procesorů, tím častěji k tomuto jevu dochází. Již u pěti procesorů zabírá čekání 50% času.

Zanedbatelná ztráta přesnosti je způsobena rozdělením dat. Data se dělí po bunchsize vektorech a přebytek se na trénování nepoužije. Rozdělováním těchto zbytků vznikne více.

Pro synchronní variantu neuronové sítě SNet v1.0 se jeví jako nejvýhodnější poměr používat 3 procesory a urychlit tak výpočet 2x.

6 DALŠÍ PRÁCE

Pro další zrychlení je potřeba implementovat asynchronní aktualizaci vah a tak eliminovat nějvětší ztráty – způsobené čekáním na nové váhy. Neuronová síť bude chvíli počítat se starými váhami a nebude čekat na nové. Zrychlení pak bude přímo úměrné počtu procesorů, ale dojde ke ztrátě přesnosti. Zrychlení by však tuto ztrátu mělo vynahradit.

Probíhá testování a ladění pracovní verze asynchronní varianty SNet v2.0.

REFERENCE

[1] CSc. Zbořil František, Doc. Ing.: Materiály k přednáškám kurzu neuronové sítě. *https://www.fit.vutbr.cz/study/courses/NEU/private/*, 2005.

[2] Matthews James: Bp example: Xor net. *http://www.generation5.org/content/2001/xornet.asp*, 2001.

[3] Matthews James: Back-propagation for the uninitiated. *http://www.generation5.org/content/2002/bp.asp*, 2002.

[4] Basic Linear Algebra Subprograms Technical (BLAST) Forum Standard, 2001.