

Demonstrace učení BP - regularizace

Předmět Soft Computing, zimní semestr 2019/20, FIT VUT Brno

Jakub Zárybnický, xzaryb00

9. 12. 2019

1 Úvod do problematiky

Backpropagation je způsob, jak zabránit přetrénování neuronové sítě nad trénovací množinou dat tak, ať je schopná generalizovat i na data jiná, např. testovací množinu dat.

Technik pro zabránění přetrénování je celá řada, mimo regularizaci existují např.:

- dropout - technika, kdy jsou v jednotlivých vrstvách během učení náhodně vypořádány neurony
- šum - zavedení šumu do vstupních dat
- včasné zastavení - monitorování přesnosti sítě na testovacích datech a zastavení v případě, že se začne snižovat (příznak přetrénování)

Regularizace přetrénování zabraňuje jiným způsobem, a to tak, že omezuje velikosti jednotlivých koeficientů vah. Existují dva typy:

- L1 - Lasso Regression, která omezuje součet čtverců váhových koeficientů a
- L2 - Ridge Regression, která omezuje součet absolutní hodnoty váhových koeficientů

Oba dva způsoby přidávají další krok do zpětné propagace, kdy jsou na konci kroku učení přičteny nejen změny vah opravující chybu, ale i další členy, které zmenšují váhové koeficienty.

2 Implementace

Pro implementaci jsem neuronové sítě, učení zpětnou propagací a regularizaci jsem zvolil jazyk Java a nejbližší ekvivalent knihovny Numpy, co jsem v Javě našel, kterým byla knihovna ND4J, která je součástí projektu DeepLearning4J. ND4J obsahuje funkce pro práci s n-rozměrnými maticemi, což bohatě stačí pro implementaci neuronové sítě, a mnohonásobně zjednodušuje práci oproti ručnímu psaní maticových operací.

Pro překlad a spuštění programu stačí v adresáři se zdrojovými soubory spustit příkazem:

```
make
```

který si následně stáhne všechny potřebné závislosti, přeloží program a spustí ho. Předdefinovanými parametry je sada dat Iris, vnitřní vrstva o šířce 10, rychlost učení 0.01 a parametr regularizace 5.

Na výstupu běhu ukázkového programu je průběh 1000 epoch paralelního učení tří sítí, které vycházejí ze stejných parametrů i náhodných úvodních vah.

Na výstupu se každých deset epoch učení objeví počet chyb, které síť udělá nad testovacími daty, a cenová funkce z poslední dávky učení.

3 Výsledky

V následujícím grafu jsou zakreslené výsledky získané během učení náhodně inicializované dvouvrstvé plně propojené neuronové sítě (šířky vrstev 4, 6, 3 v prvním obrázku, 4, 10, 3 v druhém, aktivací funkce sigmoida, one-hot kódování ve výstupní vrstvě) třemi způsoby - bez regularizace, s L1 regularizací a s L2 regularizací.

Zakreslená je kvadratická cenová funkce (*mean square error (or loss) function*) proti epochám učení (jeden bod je každých 10 epoch).

Výsledné grafy se při mém testování často velmi lišily běh od běhu, což je způsobené inicializací vah v síti náhodnými hodnotami.

