

SIECI NEURONOWE

Sprawozdanie - Ćwiczenie 3

Aleksander Poławski
Grupa - Poniedziałek 18:15
Prowadzący - mgr inż. Jan Jakubik

29 listopad, 2020

1 Cel ćwiczenia

Celem ćwiczenia trzeciego laboratoriów kursu Sieci Neuronowe było zapoznanie się z technikami poprawiającymi szybkość uczenia sieci poprzez implementację metod optymalizacji współczynnika uczenia oraz metod inicjalizacji wag.

2 Plan ćwiczenia oraz badań

- implementacja metod optymalizacji współczynnika uczenia (MOMENTUM PROSTE/MOMENTUM NESTEROVA/ADAGRAD/ADADELTA/ADAM)
- implementacja metod inicjalizacji wag (HE/XAVIER)
- przeprowadzenie eksperymentów badających skuteczność implementowanych metod

3 Opis zaimplementowanego programu

Do wykonania zadania rozwinięto program implementowany do zadania drugiego laboratorium.

Program zaimplementowano w środowisku PyCharm w języku Python, korzystając między innymi z bibliotek Numpy do przetwarzania obliczeń macierzowych.

W jego skład wchodzi następujące elementy:

- klasa Loader - umożliwia wczytywanie, przechowywanie i konwersję zbiorów uczących i testowych. Zawiera proste funkcje pomagające stwierdzić poprawność wczytania zbiorów.
- klasa MLP - zawiera całą logikę tworzenia, ustawień, uczenia i testowania sieci MLP
- klasa MLPLayer - zawiera całą logikę tworzenia, ustawień i działania poszczególnych warstw sieci
- plik main - manager programu - organizujący kolejność wykonywania zadań programu, zawierający predefiniowane testy potrzebne do wykonania badań przewidzianych w ćwiczeniu

Program w sposób prosty i intuicyjny umożliwia wykonanie wszystkich zaplanowanych w ćwiczeniu badań. Rozszerzono go o metody inicjalizacji wag (HE/XAVIER) oraz metody optymalizacji współczynnika uczenia (MOMENTUM PROSTE/MOMENTUM NESTEROVA/ADAGRAD/ADADELTA/ADAM).

4 Badania

W poniższej sekcji zamieszczono i opisano wyniki badań.

4.1 Wpływ optymalizatora współczynnika uczenia na skuteczność sieci MLP

a) Założenia

- liczba warstw ukrytych: [3]
- liczba neuronów w warstwach ukrytych: [10]
- funkcja aktywacji w warstwach ukrytych: [ReLU]
- inicjalizacja wag i biasów: [rozkład normalny]
- odchylenie standardowe w inicjalizacji wag: [0.1]
- wielkość paczki (batch): [25]
- wyjściowy współczynnik α : [0.001]
- **zmienny optymalizator współczynnika:** [MOMENTUM], [NESTEROV], [ADAGRAD], [ADADELTA], [ADAM], [brak]

b) Przebieg dla każdego ustawienia

- wykonanych zostało 10 procedur po 5 iteracji uczenia
- dokonano klasyfikacji obiektów zbioru testowego dla każdej iteracji
- obliczono średnią trafność klasyfikacji dla każdej iteracji ze wszystkich procedur uczenia

c) Otrzymane wyniki

optymalizator	średnia trafność klasyfikacji po iteracji				
	I	II	III	IV	V
brak	09.79%	10.01%	09.11%	10.11%	10.07%
MOMENTUM	15.32%	24.25%	65.63%	81.05%	83.44%
NESTEROV	11.35%	13.03%	23.34%	69.95%	80.10%
ADAGRAD	36.81%	51.87%	62.86%	66.99%	68.85%
ADADELTA	75.28%	79.23%	80.08%	81.42%	83.12%
ADAM	76.01%	79.23%	82.85%	83.11%	84.97%

Tabela 1: Wpływ optymalizatora współczynnika uczenia na skuteczność sieci MLP

d) Komentarz

- Wszystkie implementowane metody znacznie poprawiają wydajność i czas uczenia sieci dla większej (niż jedna) liczby warstw ukrytych.
- Brak użycia optymalizatorów praktycznie uniemożliwia dobry dobór współczynnika uczenia, parametr ten jest bardzo wrażliwy. W wynikach tego badania zauważyć można, że skorzystanie ze współczynnika równego 0.001 przy trzech warstwach sieci skutkuje brakiem uczenia.
- ADAGRAD zapewnia szybsze uczenie w pierwszych iteracjach (niż momentum proste lub nesterova), lecz wydajność szybko wygasa z kolejnymi iteracjami. Potwierdza to założenia teoretyczne (suma kwadratów gradientów w mianowniku modyfikatora wag ciągle rośnie, aż staje się tak duża, że modyfikacja zaczyna wynosić zero).
- Najlepsze wyniki wykazały optymalizatory ADADELTA oraz ADAM.

4.2 Wpływ metody inicjalizacji wag na skuteczność sieci MLP

a) Założenia

- liczba warstw ukrytych: [3]
- liczba neuronów w warstwach ukrytych: [10]
- **zmienna funkcja aktywacji w warstwach ukrytych: [ReLU], [TanH]**
- inicjalizacja wag i biasów: [rozkład normalny]
- wielkość paczki (batch): [25]
- współczynnik uczenia α : [0.001]
- odchylenie standardowe w inicjalizacji wag (jeśli brak optymalizatora): [0.1]
- **zmienna metoda optymalizacji wag: [HE], [XAVIER], [brak]**

b) Przebieg dla każdego ustawienia

- wykonanych zostało 10 procedur po 5 iteracji uczenia
- dokonano klasyfikacji obiektów zbioru testowego dla każdej iteracji
- obliczono średnią trafność klasyfikacji dla każdej iteracji ze wszystkich procedur uczenia

c) Otrzymane wyniki

inic. wag /aktywacja	średnia trafność klasyfikacji po iteracji				
	I	II	III	IV	V
brak/ReLU	09.79%	10.01%	09.11%	10.11%	10.07%
brak/TanH	10.58%	10.92%	11.03%	11.87%	16.33%
HE/ReLU	32.12%	41.71%	49.83%	60.61%	67.45%
HE/TanH	46.73%	58.69%	67.17%	73.83%	77.96%
XAVIER/TanH	32.35%	48.17%	55.93%	60.87%	66.77%
XAVIER/ReLU	16.11%	22.30%	22.88%	22.68%	23.55%

Tabela 2: Wpływ metody inicjalizacji wag na skuteczność sieci MLP

d) Komentarz

- Użycie automatycznych inicjalizatorów wag umożliwia uczenie nawet dla nie do końca korzystnych pozostałych ustawień (np. współczynnika alfa i braku użycia optymalizatorów współczynnika uczenia).
- Zgodnie z teorią metoda XAVIER wykazuje gorsze działanie w połączeniu z funkcją aktywacji ReLU.
- Metoda HE jest bardziej uniwersalna i poprawia wydajność w połączeniu z obiema funkcjami aktywacji.

4.3 Najlepsze ustawienie i porównanie z wynikami ćwiczenia 2

a) Założenia

- liczba warstw ukrytych: [3], [1]
- liczba neuronów w warstwach ukrytych: [10], [20]
- funkcja aktywacji w warstwach ukrytych: [ReLU]
- inicjalizacja wag i biasów: [rozkład normalny]
- wielkość paczki (batch): [25]
- **metoda optymalizacji wag: [HE]**
- **metoda optymalizacji współczynnika uczenia: [ADADELTA]**

b) Otrzymane wyniki

warstwy ukryte	średnia trafność klasyfikacji po iteracji				
	I	II	III	IV	V
3	82.01%	86.59%	88.18%	88.53%	88.59%
1	89.39%	91.01%	91.82%	92.27%	92.86%

Tabela 3: Najlepsze ustawienie i porównanie z wynikami ćwiczenia 2

d) Komentarz

- Jednoczesne użycie optymalizatora współczynnika uczenia i inicjalizacji wag poskutkowało najlepszymi rezultatami uczenia sieci MLP ze wszystkich testowanych konfiguracji aktualnego ćwiczenia oraz ćwiczenia poprzedniego.
- Udało się pokonać wszelkie inne ustawienia z zadania drugiego, gdzie najlepszy uzyskany wynik dla jednej warstwy wynosił 91.35% skuteczności, a dla trzech warstw 82.02%, a do tego udało się tego dokonać w znacznie mniejszej liczbie iteracji.
- **Najlepszy otrzymany wynik udało się osiągnąć z optymalizatorami ADADELTA i HE przy użyciu jednej warstwy ukrytej po 20 iteracjach i wyniósł on: 94.32%.**
- Warto jednak zauważyć, że sieć nie przestała się uczyć i mogła potencjalnie zwiększać dalej swoją skuteczność po kolejnych iteracjach uczenia.

5 Podsumowanie

Pomyślnie udało się zrealizować następujące wytyczne zadania:

- zaimplementowano pięć różnych metod optymalizacji współczynnika uczenia
- zaimplementowano dwa rodzaje metod inicjalizacji wag
- wykonano badania dyktowane w treści zadania oraz opracowano ciekawe wyniki dokumentujące zdobytą w zadaniu wiedzę

Skuteczność i wydajność sieci MLP zależy od wartości wielu hiper-parametrów i parametrów. Z tego powodu konfiguracja sieci jest czasochłonna i trudna. Z pomocą przychodzi możliwość automatyzacji doboru niektórych z ustawień w postaci implementacji różnych optymalizatorów i metod. Ich użycie skraca i upraszcza znacznie proces strojenia sieci.

Dodatkowo, bardziej zaawansowane metody pozwalają na automatyczną korekcję i adaptację parametrów w trakcie uczenia uwzględniając aktualne warunki procesu, co pozwala na uzyskiwanie jeszcze lepszych osiągnięć sieci.

W wyniku procesu wykonywania zadania rozwinęto swoją wiedzę na temat elementarnych pojęć dotyczących sieci neuronowych oraz nauczono się kolejnych mechanizmów działania sieci MLP.