

[WSI] Kajetan Rożej/Mikołaj Szawerda - Zadanie 5. (Sieci Neuronowe)

1. Implementowane algorytmy

Głównym celem zadania ćwiczenia 5. była implementacja perceptronu wielowarstwowego (ang. MLP). Jest to najpopularniejszy typ sztucznych sieci neuronowych. Sieć tego typu składa się zwykle z jednej warstwy wejściowej, kilku warstw ukrytych oraz jednej warstwy wyjściowej. Neurony warstw ukrytych posiadają wiele wejść i jedno wyjście, a jego wartość obliczana jest w następujący sposób:

1. $s = \sum_{i=1}^n x_i w_i + b_0$ to można poprawić, bo nie widać, że wagi się różnią dla każdego neuronu
2. $s = f(s)$, gdzie s to wybrana funkcja aktywacji

W naszej sieci jako funkcję aktywacji zdecydowaliśmy się przyjąć gaussian ($f(x) = \exp(-x^2)$)

Trenowanie sieci polega na poszukiwaniu takiego zestawu wag i biasów, które pozwolą na jak najlepszą estymację zadanej funkcji. W tym celu wykorzystywać można metodę propagacji wstecznej, która to rozkłada uzyskany błąd na poszczególne neurony, wskazując kierunek, w którym w danej iteracji należy zmodyfikować wagi w celu zmniejszenia popełnionego błędu. Tempo modyfikacji wag określone jest natomiast za pomocą współczynnika uczenia. Wykorzystanie tej metody w połączeniu z algorytmem gradientowym jest jednym z najpopularniejszych sposobów na znajdowanie wag sieci.

W związku z wymaganiami zaimplementowaliśmy również poszukiwanie wag przy pomocy algorytmu ewolucyjnego, który do ich poszukiwania korzysta jedynie z losowych mutacji, niejako sam starając się ustalić optymalny kierunek poszukiwań. Po konsultacji z prowadzącym, aby upodobnić sposób ten do metody gradientowej i skrócić czas wykonania zdecydowaliśmy się na wykorzystanie populacji liczącej jednego osobnika.

2. Zaplanowane eksperymenty

Do eksperymentów użyliśmy perceptronu z dwiema warstwami ukrytymi. Głównym zadaniem było zbadanie wpływu ilości neuronów w poszczególnych warstwach i użytego algorytmu (gradientowego lub ewolucyjnego) na przebieg procesu uczenia i jakość estymacji. Po wstępnym rozpoznaniu zdecydowaliśmy się na następujące ilości neuronów:

- 2, 2
- 5, 4
- 10, 10

plapala

Aby uzyskać pełniejszy obraz jakości poszczególnych rozwiązań, zmierzaliśmy również czas potrzebny na wytrenowanie poszczególnych wariantów sieci.

Do trenowania użyliśmy zbioru ... Natomiast zbiór testowy stanowiło ...

W przypadku algorytmu ewolucyjnego ewaluacja rozwiązania przeprowadzana była dla całego batcha (XXX wartości), natomiast na potrzeby algorytmu gradientowego epoch został podzielony na N batchy po M

wartości, po których następowała aktualizacja wag. Liczbę iteracji ustalono na poziomie 3000 dla algorytmu ewolucyjnego i YYY dla algorytmu gradientowego. Aby uzyskać miarodajne wyniki związane z wykorzystaniem losowości do inicjalizacji wektora wag (metoda gradientowa), jak i ogólnym działaniem algorytmu (metoda ewolucyjna), każdy eksperyment został uruchomiony kilka razy a do analizy użyto najlepszą z uzyskanych prób.

3. Uzyskane rezultaty

Wykres wartości MSE w poszczególnych przypadkach: ...

Wykres porównujący jakość estymacji w porównaniu z funkcją wejściową: (zastanawiam się, czy wszystko na jednym wykresie, żeby czytelnie było) ...

Wyres porównujący czas wykonania poszczególnych warintów:

4. Wnioski i przemyślenia

Tu będzie trzeba coś mądrego napisać. Spodziewane przeze mnie wnioski, które nie muszą się pokryć i lepiej się nimi nie sugerować

- większą liczbą neuronów to lepsza aproksymacja
- może coś nt. tego czy ma znaczenie która warstwa jest bardziej znacząca (jeśli to wyjdzie)
- algorytm ewolucyjny działa dłużej niż gradientowy
- algorytm gradientowy daje lepsze wyniki niż ewolucyjny
- gradientowy stale i sukcesywnie zmniejsza MSE, a ewolucyjny wykonuje 'skoki' i czasami utyka.