Dokumentacja wstępna

Podstawy Sztucznej Inteligencji

Autorzy
Antoni Charchuła
Michał Witkiewicz
Jakub Tokarzewski

Spis Treści

1. Wstęp....3
1.1. Technologia....3
1.2. Repozytorium....3
2. Etapy projektu....3
3. Realizacja zadania....4
3.1. Inicjalizacja sieci....4
3.2. Funkcja aktywacji....4
3.3. Propagacja wsteczna....4
3.4. Interpretacja wyjścia....4
3.5. Normalizacja wejść....5
3.6. Testowanie....5

1. Wstęp

Naszym poleceniem projektowym jest „Stworzyć, wytrenować i przeprowadzić walidację sieci neuronowej, która dokona predykcji, czy dane wino jest dobrej jakości: https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009 ”.

Jest to przykład problemu klasyfikacji za pomocą perceptronu wielowarstwowego, czyli grupowania win na podstawie jakości ich wybranych cech. Każde wino przedstawione jest za pomocą 12 cech, z których każde jest ocenione w innej skali. Finalna jakość wina określona jest przez ocenę w skali od 0 do 10.

1.1. Technologia

Aplikacja zostanie zrealizowana w języku Python z użyciem biblioteki *numpy,* wspomagającej prowadzenie obliczeń na macierzach.

1.2. Repozytorium

Wszystkie pliki źródłowe można na bieżąco obserwować w naszym repozytorium pod adresem https://github.com/tokerson/PSZT-Wine_Classification.

2. Etapy projektu

Projekt podzieliliśmy na kilka mniejszych etapów, które będą systematycznie realizowane.

1. Stworzenie modelu perceptronu dwuwarstwowego.
2. Implementacja obliczania wyjścia sieci neuronowej na podstawie wejść.
3. Wczytywanie zestawu danych z pliku.
4. Implementacja obliczania wartości funkcji kosztu.
5. Implementacja propagacji wstecznej za pomocą metody gradientu prostego.
6. Trenowanie sieci neuronowej korzystając z k-krotnej walidacji krzyżowej.
7. Badanie sieci neuronowej pod względem przeuczenia oraz zbyt małego dopasowania.
8. Poprawienie sieci na podstawie wniosków wyciągniętych z testów z pkt. 7 - 8.

Do dnia 21.12.2018r. zrealizowane zostały punkty 1 - 5.

3. Realizacja zadania

3.1. Inicjalizacja sieci

Po analizie budowy i działania perceptronu zdecydowaliśmy się, że warstwa ukryta naszej sieci będzie początkowo miała wielkość $\sqrt{|X|}$ (gdzie X to zbiór wejściowy). Liczba neuronów w warstwie ukrytej może ulec zmianie w związku z testami przeprowadzanymi w pkt. 7 planu realizacji projektu.

Początkowe wagi neuronów warstwy ukrytej to liczby z rozkładu

$$U \sim \left(\frac{-1}{\sqrt{(dim(X))}}, \frac{1}{\sqrt{(dim(X))}}\right)$$

gdzie dim(X) to wymiar wejścia sieci.

Początkowe bias-y zerowe.

Początkowe wagi krawędzi prowadzących z warstwy ukrytej do neuronu wyjściowego z rozkładu $U \sim \left(\frac{-1}{\sqrt{(dim(H))}}, \frac{1}{\sqrt{(dim(H))}}\right)$, gdzie dim(H) to liczba neuronów w warstwie ukrytej.

3.2. Funkcja aktywacji

Funkcja aktywacji użyta w naszej sieci to *sigmoid* wyrażona wzorem:

$$S(x) = \frac{1}{1+e^{-x}}$$

Ta sama funkcja aktywacji stosowana w warstwie ukrytej oraz na wyjściu sieci. Dzięki temu wynik będzie znormalizowany do przedziału 0 – 1.

3.3. Propagacja wsteczna

Sieć przelicza w jednym wywołaniu cały zbiór wejściowy i aktualizuje wagi oraz bias-y stosując metodę gradientu prostego. Funkcja kosztu, którą będziemy chcieli minimalizować i na podstawie której wyliczane są nowe wagi to

$J(\theta) = \frac{1}{2N}\sum_{t=1}^{N}\left\|y_t - \bar{f}(x_t;\theta)\right\|^2$, a jej gradient w każdym punkcie dziedziny ma postać

$$\nabla J(\theta) = \frac{1}{N}\sum_{t=1}^{N}\frac{dq(\bar{f}(x_i;\theta))}{d\theta^T}$$

Aktualizacja wag i bias-ów odbywa się korzystając z zależności $\theta_{t+1} = \theta_t - \beta_t \nabla J(\theta_t)$.
gdzie β to współczynnik tempa uczenia, który trzeba będzie dobrać eksperymentalnie na etapie testowania sieci neuronowej. Zbyt duży współczynnik będzie prowadził do zbyt dużych skoków, a zbyt mały do zbyt wolnego tempa nauki sieci. Uczenie będzie następowało w kilku epokach, korzystając z walidacji krzyżowej. Ilość epok również musi być dobrana w sposób eksperymentalny. Uczenie będzie trwało, do momentu uzyskania satysfakcjonującej wartości funkcji kosztu.

3.4. Interpretacja wyników

W warstwie wyjściowej naszego perceptronu mamy jeden neuron mogący przyjąć wartość od 0 do 1. Oceny wina są w skali 0 – 10, więc po przeskalowaniu ocen do zakresu 0 – 1, będziemy mogli stwierdzić jakiej dane wino jest jakości porównując je z wyjściem.

3.5. Normalizacja wejść

Dane wejściowe są podane w postaci 12 parametrów, które są ocenami danej cechy wina. Każda z tych ocen jest z innego przedziału, więc różnie będą wpływały na wynik obliczany przez sieć. Aby temu zapobiec wejścia należy przeskalować do przedziału 0 – 1.

3.6. Testowanie

Kluczowym elementem projektu jest testowanie zaimplementowanej sieci neuronowej i najważniejszych dla jej działania wartości (liczba neuronów ukrytych, β – współczynnik tempa uczenia, liczba epok, k-krotność walidacji krzyżowej). Na tym etapie nastąpi dobranie odpowiednich współczynników na podstawie badania dokładności klasyfikowania win przez sieć neuronową oraz badanie wyników pod względem przetrenowania. Poszczególne etapy dobierania współczynników zostaną przedstawione na odpowiednich wykresach, obrazujących wpływ czynnika na zachowanie się sieci.

Możliwe, że podczas testowania trzeba będzie zmienić metodę propagacji wstecznej, bądź inny z użytych algorytmów. W takim wypadku dokonamy zmiany i wnioski z osiągniętego rezultatu zapiszemy w dokumentacji końcowej.