

Sztuczne sieci neuronowe

Wstęp

Celem programu jest implementacja perceptronu wielowarstwowego z wykorzystaniem optymalizacji gradientowej z algorytmem propagacji wstecznej, przy pomocy którego dokonywana jest klasyfikacja zbioru danych MNIST (<http://yann.lecun.com/exdb/mnist/>). Zbiór MNIST wczytywany jest z API Keras będącego częścią modułu **Tensorflow**.

Sposób działania

Na początku programu są pobierane i wstępnie przetwarzane dane ze zbioru MNIST. Następnie tworzona jest sieć neuronowa na podstawie podanych parametrów. Należą do nich:

- **shape** - tablica liczb całkowitych o długości > 1 , oznaczająca kolejne rozmiary warstw ukrytych sieci
- **x_data, y_data** - dane uczące pobierane z API
- **activation** - sigmoidalna funkcja aktywacji neuronu, tutaj jest to funkcja $\frac{1}{1+\exp(-x)}$
- **activation_prime** - pochodna funkcji activation, tu wynosząca $\frac{\exp(x)}{(\exp(x)+1)^2}$
- **loss** - funkcja straty, tu mająca wzór $\frac{1}{N} \sum_{i=1}^N ||y_i - \hat{f}(x_i; \theta)||^2$
- **loss_prime** - pochodna funkcji straty, tu mająca wzór $\frac{1}{N} \sum_{i=1}^N 2 \cdot (y_i - \hat{f}(x_i; \theta))$
- **epochs** - liczba epok, domyślnie równa 30
- **learning_rate** - parametr $\in (0, 1)$ używany przy propagacji wstecznej z użyciem gradientu prostego, domyślnie równy 0,1
- **test_size** - rozmiar zbioru uczącego, domyślnie równy 1000

W tej implementacji sieci neuronowej każda warstwa reprezentowana jest przez dwie klasy: **CLayer** i **ActiveLayer**, gdzie **CLayer** reprezentuje statyczną część warstwy, zaś **ActiveLayer** część “myślącą”. Obie klasy dziedziczą po tej samej klasie abstrakcyjnej **Layer**. Po utworzeniu sieci i dodaniu do niej warstw uruchamiana jest pętla trenująca tyle razy ile wynosi wartość epochs. W jej wnętrzu wykonują się kolejno propagacja postępową, ocena funkcji straty, oraz propagacja wsteczna. Po wyjściu z pętli sieć jest wytrenowana.

Uruchamianie

Program może zostać uruchomiony na 2 sposoby 1. Bez żadnych argumentów wejściowych. Stworzona zostanie seria sieci dla parametrów $\text{test_size} \in \{1000, 2000, 5000, 10000, 30000\}$ oraz $\text{epochs} \in [200, 100, 50, 25]$, a następnie wykres trafności dla zbioru testowego mnist zawierającego 1000 osobników 2. Z dowolną liczbą l. całkowitych (które oznaczać będą wymiary dla kolejnych warstw ukrytych), oraz opcjonalnymi argumentami $-\text{test_size} -\text{epochs}$. Uruchomiona zostanie jedna sieć neuronowa dla podanych argumentów. Następnie wyświetlone zostaną wyniki oceny trafności w kolejnych epokach dla zbioru mnist oraz dla ręcznie narysowanych obrazów.

Analiza wyników

W celu przetestowania utworzonego algorytmu sporządzone zostały wykresy zależności jakości klasyfikacji w zależności od rozmiaru zbioru testowego oraz liczby epok. Pierwszy z nich został wykonany dla rozmiarów warstw ukrytych równych odpowiednio: 200, 100, 50, 25:

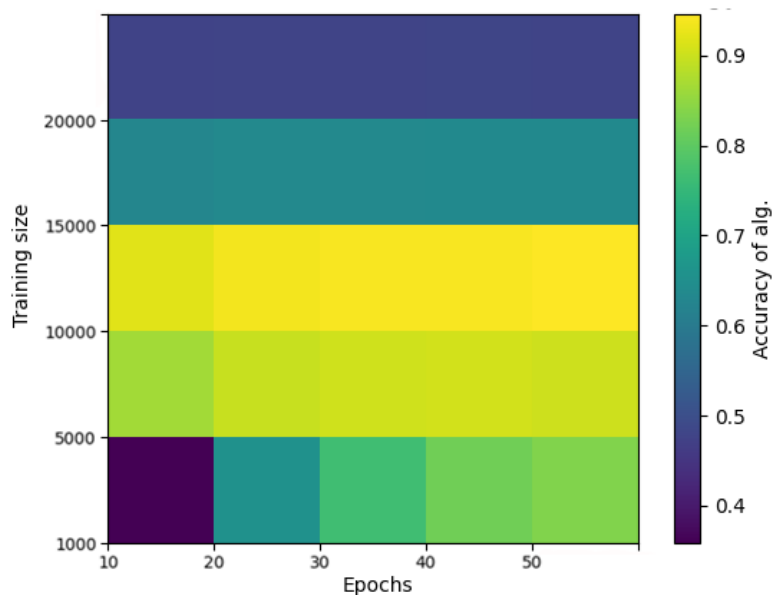


Figure 1: plot

Jak widać zarysowawały się dwie zależności

1. Liczba epok była mocno skorelowana z trafnością klasyfikacji. Dla wybranych parametrów nie zaszło zjawisko przeuczenia
2. Rozmiar zbioru testowego w pewnych sytuacjach może być zbyt duży, czego skutkiem jest drastyczne obniżenie jakości algorytmu
3. Nawet dla najgorszego zestawu parametrów jakość algorytmu jest lepsza od EX , gdzie X to zm. losowa o rozkładzie dyskretnym opisująca prawdopodobieństwo poprawnych przypisań danej liczby cyfr w następującym zdarzeniu: każdej cyfrze przypisujemy jedną cyfrę bez powtórzeń.
 $EX = \sum_{x_i \in S_x} \cdot P(X = x_i) = \frac{1}{10} + \frac{1}{9} + \frac{9}{160} + \frac{3}{175} + \frac{7}{2000} + \frac{63}{125000} + \frac{105}{2000000} + 4(\frac{1}{10})^6 + \frac{47}{2}(\frac{1}{10})^8 + (\frac{1}{10})^{10} \approx 0,288 < 0,35$ co dowodzi skuteczności algorytmu

Zobaczmy teraz jak wykres będzie wyglądał dla jednej, znacznie większej warstwy ukrytej: 1000:

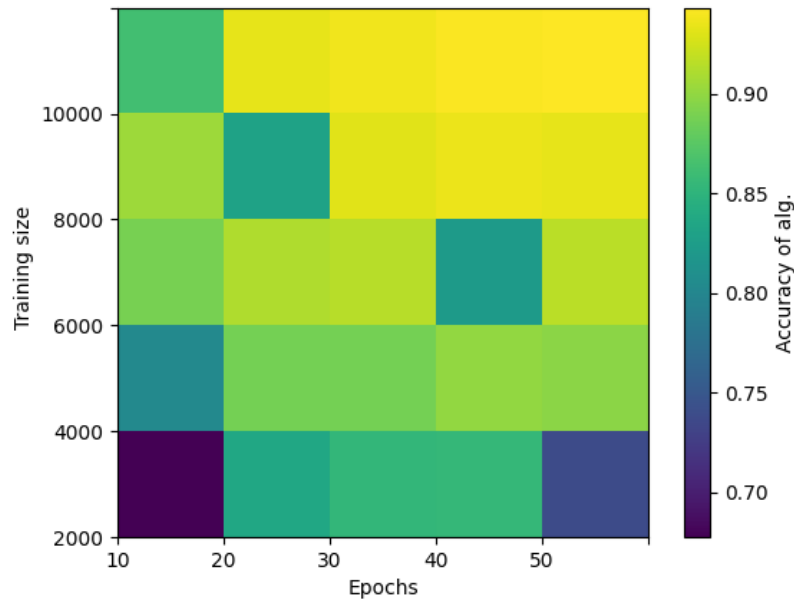


Figure 2: plot

Zwiększenie rozmiaru warstw ukrytych ma ogromne znaczenie dla skuteczności działania algorytmu. W tym wypadku nastąpiło znaczne polepszenie jakości wyników jednak dla mniejszych zaczęło odgrywać rolę zjawisko przeuczenia. W sytuacji, gdy nie są wykorzystywane techniki wczesnego zatrzymania uczenia, regularyzacji ani odrzucania, rozmiar warstwy ukrytej okazał się być zbyt duży

dla małego zbioru testowego, choć dalej był znacznie skuteczniejszy od sieci z mniejszymi warstwami ukrytymi. Jaki rozmiar warstwy jest zatem optymalny? Jedynym sposobem na określenie tego byłoby dalsze stosowanie metody prób i błędów, gdyż w sieciach neuronowych nie ma jednoznacznie określonego wzoru na wyliczanie tych parametrów. Nie jest również regułą to, że warstwa ukryta musi być mniejsza od warstwy wejściowej. Warto również nadmienić, że dwa artefakty powstałe dla parametrów (20, 8000) oraz (40, 6000) są możliwą konsekwencją zastosowania perceptronu dwuwarstwowego zamiast wielowarstwowego.

Dalszym etapem rozważań mogłoby być pytanie o wydajność. Sieci neuronowe ze względu na konieczność wykonywania wielu iteracji pewnych problemów potrafią być wyjątkowo zasobożerne. Generowanie pierwszego z wykresów trwało 8h 40min. Drugi powstawał już przez 12h, z kolei próba generowania wykresu dla warstwy ukrytej o rozmiarze 3000 musiało być przedwcześnie zakończone po 33h. Dlatego czasem konieczne jest poświęcenie jakości sieci na rzecz skrócenia czasu jest formowania

Ostatnią funkcjonalnością sieci jest możliwość klasyfikowania ręcznie utworzonych obrazów o rozmiarze 28x28 p. Dla sieci neuronowej z 4 warstwami ukrytymi [200, 100, 50, 25] , rozmiarem zb. treningowego 30000 i liczbą epok 10, wynik klasyfikacji 10 cyfr z pliku samples prezentuje się następująco:

```
NETWORK CLASSIFIED 0 as 0 with probability of 0.9015740215283164
NETWORK CLASSIFIED 1 as 9 with probability of 0.7581752169782214
NETWORK CLASSIFIED 2 as 2 with probability of 0.9788834962324736
NETWORK CLASSIFIED 3 as 3 with probability of 0.5370101401752179
NETWORK CLASSIFIED 4 as 8 with probability of 0.23128159669599963
NETWORK CLASSIFIED 5 as 5 with probability of 0.8046032596204756
NETWORK CLASSIFIED 6 as 6 with probability of 0.40650633789413326
NETWORK CLASSIFIED 7 as 2 with probability of 0.778977682926675
NETWORK CLASSIFIED 8 as 8 with probability of 0.5401737087583314
NETWORK CLASSIFIED 9 as 3 with probability of 0.45197299505509914
```