LAB5

# Wstęp

Implementacja algorytmu znajduje się w pliku MLP, generowanie wykresów w pliku graphs.

Użyte biblioteki/funkcje:

* sklearn
* numpy
* ProcessPoolExecutor z concurrent.futures
* Pyplot

W implementacji grafów została użyte ProcessPoolExecutror w celu znaczącego przyśpieszenia wykonywania się wykresów

Obraz zawierający tekst, zrzut ekranu, numer, Równolegle

Opis wygenerowany automatycznie

Część przedstawionych wykresów została wykonana na stacjonarnym komputerze z procesorem z 16 dostępnymi wątkami i wykonywanie ich z podanymi parametrami może skutkować bardzo długim czasem wykonania na jednostce obsługującej mniej wątków lub posiadającej znacząco słabsze rdzenie.

# Opis algorytmu

Zadanie polegało na zaimplementowaniu perceptronu wielowarstwowego oraz wybranego algorytmu optymalizacji gradientowej z algorytmem propagacji wstecznej.

Perceptron wielowarstwowy został wytrenowany do klasyfikacji zbioru danych MNIST (http://yann.lecun.com/exdb/mnist/). Zbiór MNIST został załadowany poprzez (sklearn.datasets.load digits()).

# Przygotowania

Domyślne parametry dla dalszych eksperymentów, zostały określone metodą prób i błędów i wynoszą:

liczba treningów na tym samym zbiorze danych - epochs: 100

wielkość skoku w algorytmie gradientu prostego - learning rate: 0,01

liczba przeanalizowanych przykładów przed wykonaniem propagacji wstecznej - batch\_size: 8

liczba neuronów w kolejnych warstwach sieci – layers: 64, 128, 64, 10

Warstwy w dalszej części będą odnosić się do wartości między 64 i 10, takie liczby neuronów na początkowej i ostatniej warstwie są wymagane do poprawnego działania programu.

# Wpływ learning rate na accuracy:

**Pierwszy test:**

Parametry: learning rate z zakresu 0.001 – 0.068 – 100 punktów na wykresie

Obraz zawierający tekst, zrzut ekranu, linia, Wykres

Opis wygenerowany automatycznie

Obserwacje:

Bardzo małe wartości learning rate osiągają mniejsze wartości accuracy

**Drugi test:**

Te same parametry wykres jest medianą 5 wykonań

Obraz zawierający tekst, zrzut ekranu, linia, Wykres

Opis wygenerowany automatycznie

**Trzeci test:**

Początkowa wartość learning rate została ustawiona na 0,003, aby zapewnić dokładniejszy wykres

Obraz zawierający tekst, zrzut ekranu, linia, diagram

Opis wygenerowany automatycznie

Obserwacje:

Zauważalny mały trend spadkowy zaczynający się około 0.05

# Wpływ epochs na accuracy:

Ustawiony został zakres epochs między 1 i 80, badane jest 40 punktów, zakres dobrany dla optymalnej czytelności wykresu, wartości poniżej są medianą 10 wywołań

Obraz zawierający tekst, Wykres, diagram, zrzut ekranu

Opis wygenerowany automatycznie

Obserwacje:

Małe wartości osiągają mniejsze wartości accuracy

# Wpływ batch size na accuracy:

Ustawiony został zakres epochs między 1 i 64, badane jest 16 punktów, wartości poniżej są medianą 10 wywołań

Obraz zawierający tekst, diagram, linia, Wykres

Opis wygenerowany automatycznie

Obserwacje:

Wykonywanie treningu dla bardzo małych wartości zwiększa czas wykonania, natomiast generuje minimalnie lepsze wyniki

# Wpływ ilości neuronów i ich ustawienie na accuracy :

Wszystkie testy dla warstw są medianą 5 wykonań

### Pojedyncza warstwa:

**Pierwszy test:**

Warstwa o wielkości od 1 do 62

Obraz zawierający tekst, zrzut ekranu, linia, diagram

Opis wygenerowany automatycznie

Obserwacje:

Bardzo mała liczba neuronów, uzyskuje bardzo małe wartości accuracy

**Drugi test:**

Wartości między 7 i 40

Obraz zawierający tekst, diagram, Wykres, linia

Opis wygenerowany automatycznie

Obserwacje:

Wzrost accuarcy wraz z liczbą neuronów

**Trzeci test:**

Warstwa o wielkości od 25 do 124

Obraz zawierający tekst, Czcionka, zrzut ekranu, linia

Opis wygenerowany automatycznie

Obserwacje:  
Mały wzrost dokładności wraz ze zwiększającą się liczbą neuronów

### Podwójna warstwa:

**Pierwszy test:**

1/8 kombinacji 2 warstw o zakresach od 1 do 64

Obraz zawierający linia, tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, linia, Wykres

Opis wygenerowany automatycznie

Obserwacje:

Pojedynczy neuron na ostatniej warstwie, obniża znacząco accuracy. Mała liczba neuronów obniża accuracy.

**Drugi test:**

Wykres wszystkich kombinacji 2 warstw o wielkości z zakresu 1-16

Obraz zawierający tekst, linia, Wykres, zrzut ekranu

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, linia, Wykres

Opis wygenerowany automatycznie

Obserwacje:

Bardzo mała dokładność w przypadku małej ilości neuronów w drugiej warstwie.

Przesortowany wykres względem a\*b, a i b liczba neuronów na warstwach  
Obraz zawierający tekst, linia, diagram, zrzut ekranu

Opis wygenerowany automatycznie

Obserwacje:

Małe wartości oraz warstwy z jednym neuronem utrudniają czytelność

**Trzeci test:**

Przesortowane wartości z poprzedniego testu, oraz ustawiono minimalną liczbę neuronów na warstwie na 4

Obraz zawierający zrzut ekranu, linia, Wykres, tekst

Opis wygenerowany automatycznie

Obserwacje:

Mała liczba neuronów na dowolnej warstwie obniża accuracy. Wraz z wzrostem liczby neuronów na obydwu warstwach wzrasta accuracy.

**Czwarty test:**

Wpływ batch size na poprzedni test, ustawiono wartość 2  
Obraz zawierający tekst, zrzut ekranu, linia, Wykres

Opis wygenerowany automatycznie

Obserwacje:

Wykres jest bardziej „spłaszczony”, co onzacza mniejsze spadki w dokładności. Czas obliczeniowy znacząco wzrósł, ponieważ propagacja wsteczna jest wykonywana znacznie częściej.

**Czwarty test:**

Wpływ batch size=1, na ten sam zakres nieposortowany, wartości z raz wykonanego testu

Obraz zawierający linia, zrzut ekranu, Czcionka, Wykres

Opis wygenerowany automatycznie

Obserwacje:

Wykres stał się bardziej „chaotyczny”

### Trzywarstwowe sieci:

**Pierwszy test:**

Wszystkie konfiguracje 3 warstwowych sieci z zakresem neuronów między 10-16

Obraz zawierający linia, Wykres, zrzut ekranu

Opis wygenerowany automatycznie

Obserwacje:

Mało czytelny wykres. Wszystkie wartości osiągają podobne wyniki

**Drugi test:**

Porównanie wybranych wartości layers, które mają równą wartość mnożenia liczy neuronów przez warstwy

Obraz zawierający diagram, Wykres, linia, tekst

Opis wygenerowany automatycznie

Obraz zawierający diagram, linia, tekst, Wykres

Opis wygenerowany automatycznie

Obserwacje:

Warstwy z małą liczbą neuronów, obniżają accuracy

**Trzeci test:**

Porównanie wartości accuracy dla sieci z dużą liczbą neuronów

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, diagram, Wykres

Opis wygenerowany automatycznie

# Wyniki i wnioski

Z analizy wpływu parametrów na accuracy wynika, że bardzo małe wartości learning rate, małe wartości epochs, oraz mała liczba neuronów ma negatywny wpływ na accuracy. Dodatkowo za duże wartości learning rate >0.07 powodują bardzo słabe wyniki accuracy.

Duże wartości dla batch size powodują mała utratę accuracy, natomiast bardzo małe wartości powodują dłuższe wykonywanie algorytmu jednak z minimalnym pozytywnym wpływem na accuracy, co dowodzi poprawności działania algorytmu optymalizującego.

Wartości epochs dla dobranych wartości negatywnie wpływają tylko dla małych wartości, po osiągnięciu wartości koło 20 wykres zaczyna się wypłaszczać.

Największy wpływ oraz ciekawe wartości generuje dobieranie wartości layers. Dla sieci z jedną wewnętrzną warstwą im więcej neuronów tym wyższa dokładność, która wypłaszcza się po osiągnięciu 70 neuronów.

Dla sieci dwu warstwowej można zauważyć negatywny wpływ warstw o małej liczbie neuronów w szczególności dla jednego neurona. Można zauważyć, że taka warstwa pomimo dużej liczby neuronów na poprzedniej warstwie, bardzo negatywnie wpływa na dokładność. Również inne wartości <8 negatywnie wpływają na dokładność, jednak na nią częściowo negatywnie wpływa również za duża wartość batch size jak ukazuje wykres w czwartym teście dla sieci dwuwarstwowych.

Sieci powyżej dwóch warstw nie poprawiają znacząco accuracy, jednak znacząco zwiększają czas wykonywania się programu. Negatywny wpływ na dokładność mają również warstwy o za małej liczbie neuronów, szczególnie <8.