

WSI Zadanie 5 – Klasyfikacja odręcznie narysowanych cyfr

Rafał Mironko, Adrian Garbowski

Założenia:

W ramach tego projektu zdecydowaliśmy się na nieco inne podejście niż standardowe testowanie wpływu hiperparametrów na działanie funkcji. Zamiast tego ograniczyliśmy się tylko do krótkiego sprawdzenia najważniejszych parametrów, skupiając się na zbiorze MNIST z biblioteki TensorFlow (obrazek 28x28 px zamiast 8x8) i na stworzeniu programu, który poradzi sobie z rozpoznawaniem cyfr rysowanych bezpośrednio przez użytkownika.

Ponieważ chcieliśmy skupić się na istocie zadania, nie testowaliśmy różnych funkcji aktywacji i straty. Po przeprowadzeniu odpowiedniego researchu zdecydowaliśmy, że optymalne rezultaty osiągniemy przy użyciu funkcji aktywacji ReLU oraz Cross Entropy Loss, a w ostatniej warstwie zastosujemy funkcję Softmax.

Testowanie parametrów

Przeanalizowaliśmy wpływ kilku kluczowych parametrów na działanie sieci neuronowej, koncentrując się na liczbie warstw, wartości learning rate oraz rozmiarze batchy.

Liczba warstw:

Eksperymenty wykazały, że dodawanie kolejnych warstw niekoniecznie poprawia skuteczność modelu – wręcz przeciwnie, zauważyliśmy minimalny spadek trafności przy większej głębokości sieci. Ostatecznie zdecydowaliśmy się na architekturę z warstwami [784, 128, 64, 10], która okazała się wystarczająco efektywna.

Learning rate:

Kolejnym krokiem było dobranie optymalnej wartości learning rate, czyli „kroku” podczas poruszania się w dół gradientu. Co ciekawe, sugerowana w literaturze wartość 0.01 nie sprawdziła się tak dobrze, jak się spodziewaliśmy. Po serii testów okazało się, że learning rate = 0.1 daje lepsze rezultaty, więc przyjęliśmy tę wartość.

Rozmiar batchy:

Na koniec sprawdziliśmy, jaki wpływ ma wielkość batchy. Okazało się, że zmiana rozmiaru batcha nie ma znaczącego wpływu na działanie modelu na zbiorze MNIST – mniejsze batche jedynie wydłużają czas treningu. Dlatego pozostaliśmy przy wartości 128, która zapewnia dobra równowagę między szybkością a efektywnością.

Finalna konstrukcja modelu

Podsumowując, nasz ostateczny model charakteryzuje się następującymi parametrami:

- **Funkcja aktywacji:** ReLU dla większości warstw, Softmax w ostatniej warstwie
- **Funkcja straty:** Cross Entropy Loss
- **Learning rate:** 0.1
- **Architektura warstw:** [784, 128, 64, 10]
- **Batch size:** 128
- **Liczba epok:** 10

Podsumowanie

Zebraliśmy rekomendowane w literaturze wartości parametrów, przetestowaliśmy kilka z nich, a następnie postawiliśmy sobie za cel rozpoznawanie cyfr rysowanych bezpośrednio przez użytkownika. Uważamy, że taki cel był ciekawszy niż dotychczasowe przedstawianie statystyk na wykresach i sprawił, że było to interesujące wyzwanie. Mamy nadzieję, że Prowadzący doceni nasze podejście oraz zrozumie, że brak dokładnych statystyk wynikał ze skoncentrowania się na funkcjonalności programu, a nie tylko na strojeniu parametrów.