SIECI NEURONOWE – ćwiczenie 2

W drugim ćwiczeniu zajmiemy się uczeniem maszynowym. Wykorzystamy regresję logistyczną – metodę statystyczną która można uznać za formę najprostszej (jednowarstwowej) sieci neuronowej. Model będziemy wykorzystywać do klasyfikacji, czyli chcemy, aby aproksymował prawdopodobieństwo przynależności próbki opisanej wektorem x do właściwej klasy. Do tego celu możemy wykorzystać zbiór przykładów uczących w postaci par wejście-klasa.

Możemy zdefiniować wyjście jako:

$$p(x) = \sigma(Wx + b)$$

Gdzie x to wektor wejściowy, W i b to parametry modelu natomiast funkcja sigma to:

$$\sigma(\mathrm{n}) = \frac{1}{1 + e^{-n}}$$

Jest to wygodna funkcja dająca wyniki przedziale [0,1] dla każdego rzeczywistego n, przy okazji rosnąca i różniczkowalna.

Aby wyuczyć model w jakikolwiek sposób, potrzebujemy zdefiniowanego zadania optymalizacji, a do tego potrzebujemy funkcji kosztu którą będziemy minimalizować. Korzystamy z entropii krzyżowej, dla przykładu (x,y) w klasyfikacji binarnej:

$$L = -y \ln p(x) - (1 - y) \ln(1 - p(x))$$

(y będzie zawsze zerem lub jedynką, czyli tylko jeden ze składników będzie niezerowy. Wartość sumujemy po wszystkich przykładach w zbiorze danych, suma będzie pominięta w wzorach.)

Zaletą takiego kosztu jest fakt, że jej pochodna po wagach modelu jest bardzo prosta:

$$\frac{\partial L}{\partial w_i} = -(y - p(x))x_i$$

Optymalizacja modelu będzie polegała na wykonywaniu niewielkich kroków w kierunku wyznaczonym przez gradient – pochodne cząstkowe po wszystkich wagach – w pętli uczenia, aż model osiągnie zbieżność. Innymi słowy, aktualizujemy wagi według:

$$w_i' = w_i - \alpha \frac{\partial L}{\partial w_i}$$

Gdzie alfa to pewien współczynnik uczenia, hiperparametr który musimy ustawić z góry.

Zadaniem na zajęcia 2 jest implementacja działającego na zbiorze *heart disease* modelu klasyfikacji, przy czym:

- Zbieżność modelu można zdefiniować przez wystarczająco małą zmianę funkcji kosztu w danej iteracji i pewną maksymalną liczbę iteracji
- Model może uczyć się wyliczając sumaryczną pochodną z funkcji kosztu po całym zbiorze, po jednym przykładzie lub po paczce przykładów w iteracji. Dla sieci neuronowej w następnym zadaniu będzie już wymagany tryb paczkowania, więc warto przećwiczyć operacje na całych macierzach a nie tylko pojedynczych wektorach danych
- Uczenie się modelu powinno być weryfikowalne metryką (np. accuracy, fscore, precision – można korzystać z bibliotek)
- Weryfikacja powinna uwzględnić podział na dane uczące i testowe

Ćwiczenie oceniane jest w skali 0-10 pkt.