

## Zadanie 4

Podobnie jak w wariacie zadania 3, do zrealizowania tego zadania będzie potrzebne wykorzystanie bibliotek do budowania sieci neuronowych: torch, tensorflow w tym keras, etc.

W zadaniu należy skorzystać ze zbioru recenzji IMDB:

<https://keras.io/api/datasets/imdb/>

podjąć próbę klasyfikacji tych recenzji z wykorzystaniem sieci rekurencyjnych - prostej RNN i LSTM. Sieci RNN i LSTM powinny być wytłumaczone na wykładzie, niemniej dla przypomnienia:

- sieć rekurencyjna przetwarza sekwencję wektorów
- stosuje takie samo przekształcenie dla każdego kroku w sekwencji
- w kroku  $n$ , wejściem jest krok  $n$  wejściowej sekwencji, oraz wyjście z kroku  $n-1$
- typy „komórek” (RNN/GRU/LSTM) różnią się dodatkowym przekształceniem nakładanym na wyjście z poprzedniego kroku

Należy zapoznać się z dokumentacją wybranej biblioteki (sugerowany po raz kolejny moduł keras, inne opcje jak najbardziej dopuszczalne), zaimplementować na zajęciach sieć RNN uczącą się klasyfikacji danych ze zbioru IMBD i opracować wyniki badań.

Oprócz samych warstw rekurencyjnych, dwie istotne nowości:

- warstwa Embedding, której celem jest skonwertowanie wartości ze skończonego zbioru na wektory. Dla problemów językowych, zbiorem może być np. zbiór wszystkich słów języka. Warstwa taka to po prostu macierz (liczba słów) na (wymiar wektora), z której tak samo jak z każdej innej macierzy w obrębie struktury sieci neuronowej jesteśmy w stanie wyliczyć gradient. A to znaczy, że możemy ją optymalizować, co kończy się znajdowaniem dla każdego słowa wektora, który w jakiś sposób oddaje „znaczenie” (a w zasadzie: powiązania między podobnymi znaczeniami, zakodowane w podobieństwie/odległości wektorów).
- Maskowanie: ponieważ typowo chcemy uczyć sieć korzystając z paczkowania wejść, musimy poradzić sobie z problemem sekwencji o potencjalnie różnej długości - ograniczenie typu „klasyfikator dla zdań które mają dokładnie 10 słów” raczej nie ma sensu praktycznego. Zwykle, w takiej sytuacji paduje się input zerami. Jednak zera te nie powinny wpływać na końcowy wynik - stąd potrzeba zadania maski na wejście. Jednym z badań do wykonania w ćwiczeniu będzie porównanie, na ile wpływa to na wynik.

### **Zadanie jest punktowane następująco:**

10 pkt - implementacja architektury na zajęciach

10 pkt - dostrojenie hiperparametrów (w szczególności proszę zwrócić uwagę na rozmiar warstwy embedding).

5 pkt - porównanie komórek RNN a LSTM

5 pkt - zbadanie wpływu stosowania maskowania/paddingu