

# SIECI NEURONOWE

## Raport z laboratorium 7

Student: Van Hien Le - 257795

### Narzędzia i technologie

- Python, Pandas, Numpy, Scikit-learn, PyTorch, Keras
- Jupyter Notebook

### Dataset

Zbiór danych IMDB, dostępny za pośrednictwem biblioteki Keras, składa się z dużego zbioru recenzji filmów. Recenzje te są oznaczone jako pozytywne lub negatywne.

Dostęp do zbioru danych przez:

```
from keras.datasets import imdb  
  
(x_train, y_train), (x_test, y_test) = imdb.load_data()
```

### Architektura sieci

W tym ćwiczeniu skupiłem się na budowaniu sieci neuronowych przy użyciu warstw rekurencyjnych. Poniżej znajdują się najważniejsze elementy architektury:

- **Warstwa Embedding:** Ta warstwa jest skonfigurowana do obsługi maksymalnej liczby obiektów (`max_features`) i przekształca je w 128-wymiarowe gęste wektory. Wybór 128 wymiarów wektorów embedding zapewnia równowagę między złożonością modelu a możliwością uchwycenia znaczących relacji między słowami.
- **Warstwa rekurencyjna:** Rdzeniem modelu jest warstwa rekurencyjna, która jest elastyczna i pozwala na dostosowanie się do różnych typów warstw rekurencyjnych (takich jak LSTM lub SimpleRNN). Liczba jednostek w warstwie rekurencyjnej jest domyślnie ustawiona na 5, ale można to dostosować. Jednostki te reprezentują wymiarowość przestrzeni wyjściowej warstwy i mogą znacząco wpłynąć na zdolność modelu do wychwytywania zależności w danych.
- **Warstwa wyjściowa:** Po warstwie rekurencyjnej się następuje warstwa wyjściowa z pojedynczą jednostką i sigmoidalną funkcją aktywacji.

### Hiperparametry

Eksperymenty przeprowadzono z różnymi ustawieniami:

- **Rodzaj warstwy rekurencyjnej:** SimpleRNN i LSTM
- **Wymiar warstwy rekurencyjnej:** 2, 5 i 11
- **Długości przycinania sekwencji:** 250, 20 i 10

Domyślne hiperparametry:

- **Szybkość uczenia:** Początkowa 0.0001, kroki zaniku to 782, szybkość zaniku wynosi 1
- **Liczba epok:** 30

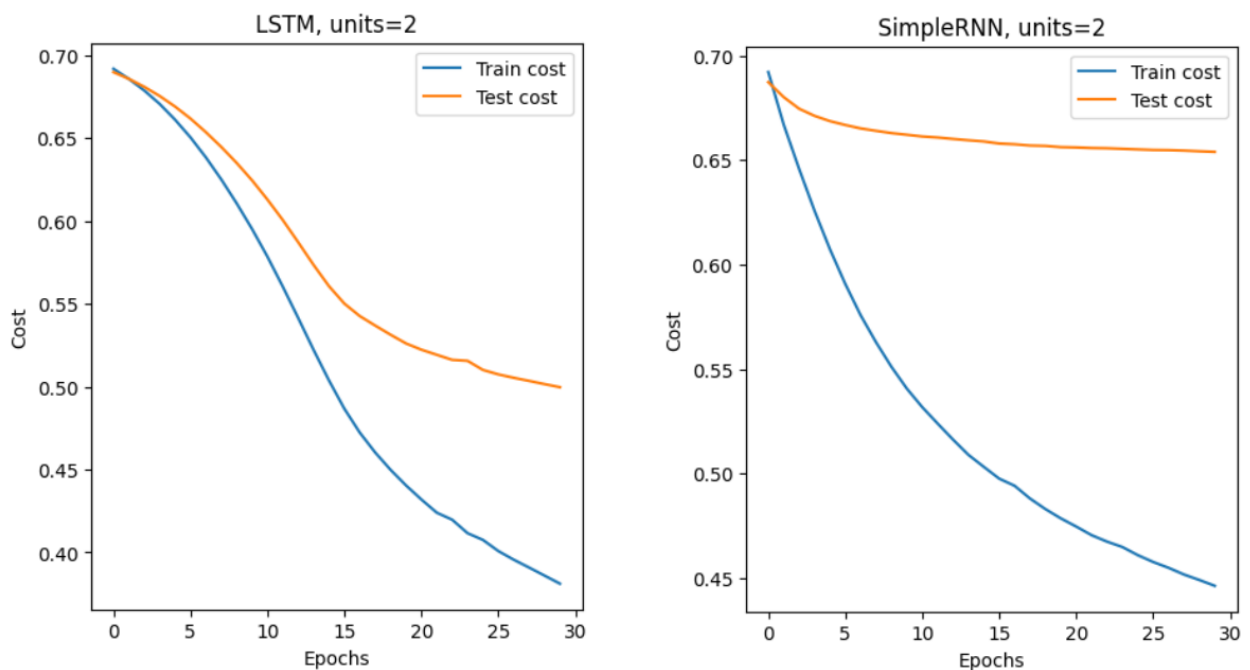
- **Optimizer:** Adam z LearningRateScheduler
- **Funkcja kosztu:** Cross-Entropy loss
- **Rozmiar batcha:** 64

Dodatkowo, rozmiar zbioru danych treningowych i testowych wynosi 5000.

## Wyniki eksperymentów

### Eksperyment 1: Rodzaj warstwy rekurencyjnej

Warstwa rekurencyjna	Metryk Dokładność
SimpleRNN	0.61
LSTM	0.81

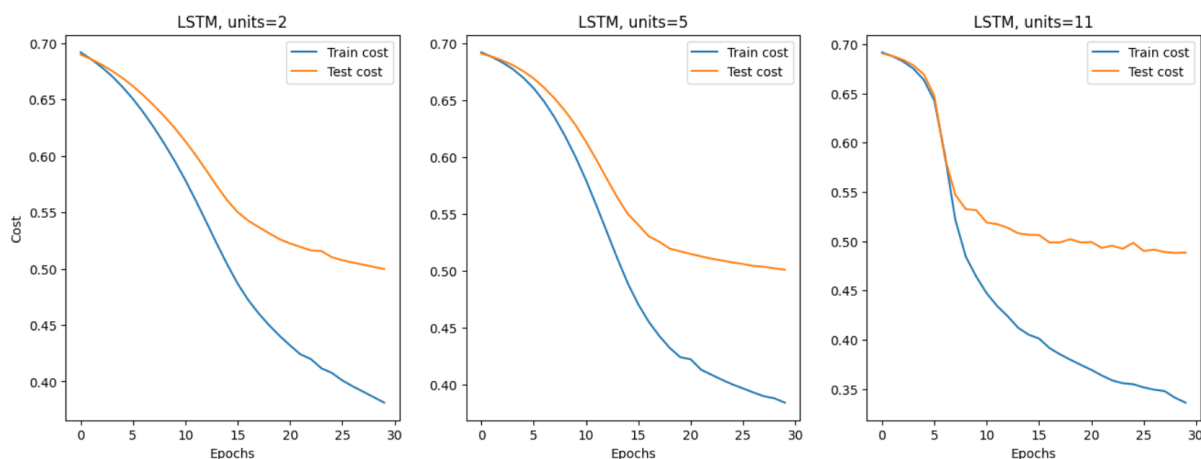


Analiza: Koszty trenowania i testowania w przypadku obu modeli maleją z czasem, co wskazuje, że oba modele się uczą. Jednak koszty testowania modelu z warstwą LSTM są niższe niż w przypadku SimpleRNN, również model z warstwą LSTM ma wyższą metrykę Dokładność, która wynosi 0,81 w porównaniu do 0,61 modelu z warstwą SimpleRNN. Ogólnie rzecz biorąc, pierwszy model lepiej radzi sobie z uogólnianiem na niewidoczne dane.

### Eksperyment 2: Wymiar warstwy rekurencyjnej

- W eksperymencie wybrano LSTM jako warstwę rekurencyjną.

Wymiar warstwy rekurencyjnej LSTM	Metryk Dokładność
2	0.81
5	0.81
11	0.82

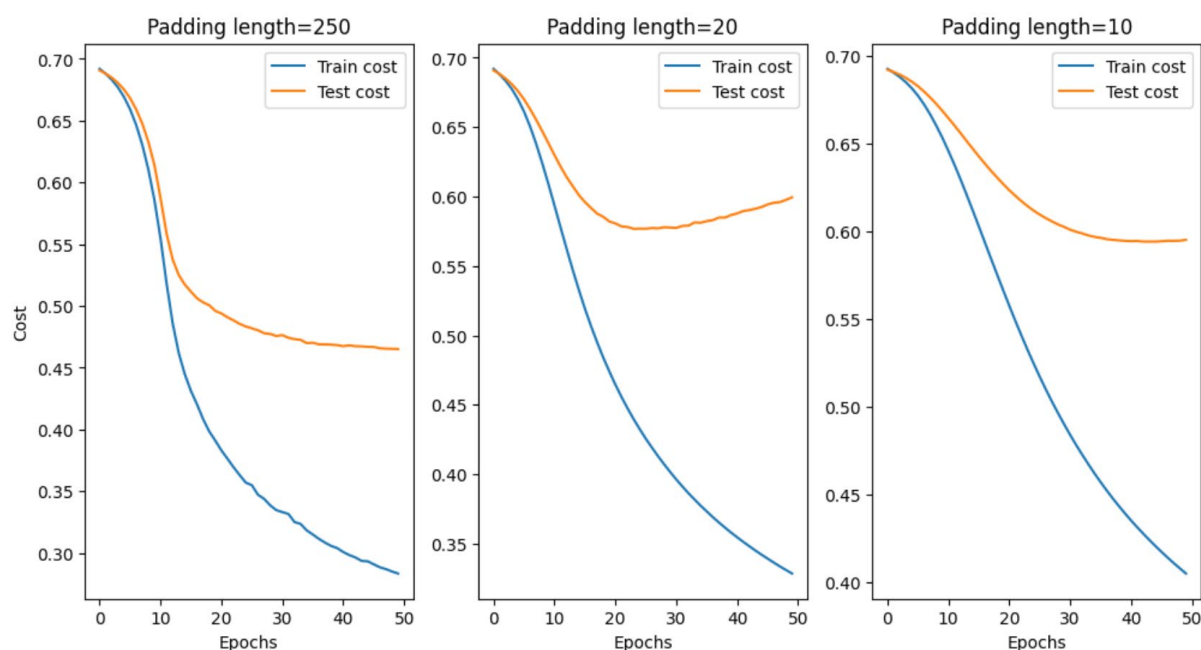


- Analiza: Jak widzimy, koszty uczenia i testowania wszystkich trzech modeli są stosunkowo niskie, co sugeruje, że modele są w stanie dobrze uczyć się danych uczących i dobrze generalizować na dane niewidoczne. Koszt modelu z 11 jednostkami jest nieco niższy niż modeli z 2 i 5 jednostkami. Poza tym dokładność modelu z 11 jednostkami jest również nieco wyższa niż w przypadku pozostałych modeli. Wydaje się, że użycie większej liczby jednostek może prowadzić do lepszej generalizacji i wydajności, ale w tym przypadku różnica nie jest wyraźnie zauważalna.

### Eksperyment 3: Długości przycinania sekwencji

- W eksperymencie wybrano LSTM jako warstwę rekurencyjną. Wymiar warstwy LSTM to 5.

Długość przycinania	Metryk Dokładność
250	0.81
20	0.70
10	0.68



- Analiza: Model o pełnej długości sekwencji 250 jest najlepszym z trzech modeli, ponieważ osiąga najwyższą metrykę Dokładność. Patrząc na krzywe uczenia się tego model, widzimy, że może dobrze się uczyć i generalizować. Kiedy zmniejszymy długość sekwencji do 20 lub 10, modele nadal uczą się na danych treningowych, ale wydaje się, że nie są w stanie uogólniać, ponieważ koszty testowania są

nadal wysokie, a metryki Dokładność wynoszą zaledwie około 0,70. Jest to zrozumiałe, ponieważ jeśli zachowamy oryginalne długości sekwencji, model może uchwycić więcej kontekstu w sekwencjach, ale może to prowadzić do wydłużenia czasu uczenia się.