

Architektury sieci neuronowych i ich uczenie w tensorflow

SZTUCZNA INTELIGENCJA I AUTOMATYZACJA PROCESÓW BIZNESOWYCH W UJĘCIU TECHNICZNYM

31.01.2026

© MGR INŻ. NATALIA POTRYKUS

Program

3 spotkania:

- 17.01.2026 r. 13:00-16:15
Encoder-decoder, segmentacja obrazu w praktyce
- 31.01.2026 r. 9:00-12:15
Dostrajanie hiperparametrów, praca z danymi sekwencyjnymi, generowanie tekstu
- 31.01.2026 r. 13:00-16:15
Transformery: pierwsze starcie, konsultacje, Q&A

MLP

Wejście: wektor liczb

Wyjście: wektor liczb

CNN

Wejście: n-wymiarowe tensory

Wyjście: wektor liczb / n-wymiarowe tensory

Co jeszcze może być naszymi danymi wejściowymi?

- liczby (tablice, wektory cech)
- obrazy (zdjęcia, grafika, ale i spektrogramy)
- tekst (sekwencja słów / liter)
- video (sekwencja obrazów)
- fala audio (sekwencja liczb)
- grafy
- mix
- ...

Dane sekwencyjne

Dane, które zawierają więcej niż jeden element, a także istotna jest ich kolejność.

Przykład - zdanie:

The queen is eating the duck. -> ["the", "queen", "is", "eating", "the", "duck"]

The duck is eating the queen. -> ["the", "duck", "is", "eating", "the", "queen"]

Tylko Ania dostała wyróżnienie. <-> Ania dostała tylko wyróżnienie.

Słowa zachowane, ale sens jakby inny...

Dane sekwencyjne

Ponadto, sekwencje mogą mieć różne długości.

Przykład - maszynowe tłumaczenie z użyciem sieci neuronowej:

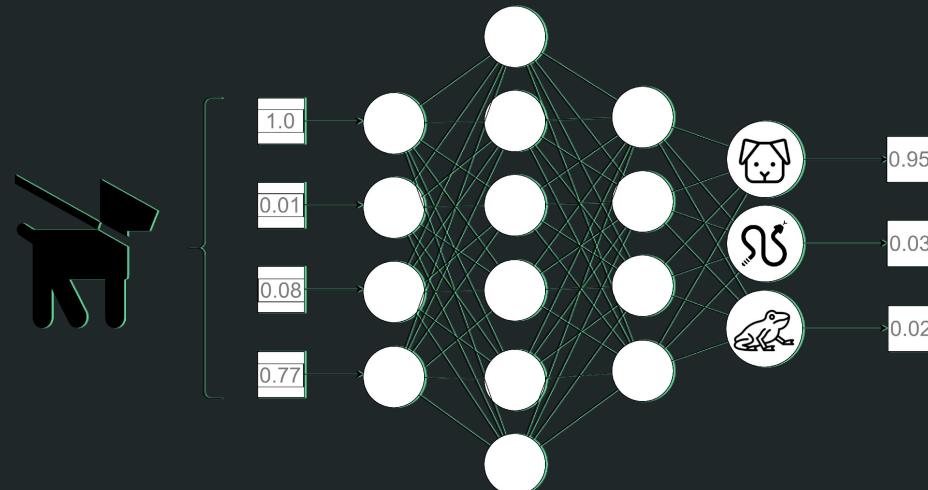
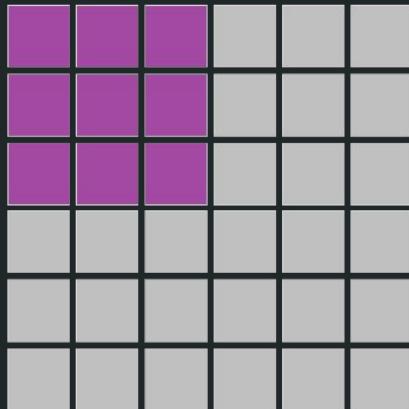
I bake the best brownie. (5 słów) -> *Piekę najlepsze brownie.* (3 słowa)

My umbrella is useless, but pretty colorful. (7 słów) -> ...

1. Sekwencje, które wprowadzamy jako wejście do modelu mogą mieć różne długości
2. Jeżeli chcemy otrzymać sekwencję również na wyjście, jej długość też może się różnić od wejściowej (i być od niej w miarę niezależna - nie jesteśmy w stanie stwierdzić, że dla sekwencji złożonej z x elementów, wyjście zawsze będzie miało y słów)

Czy CNN / MLP potrafią to przetworzyć?

Nie bardzo...



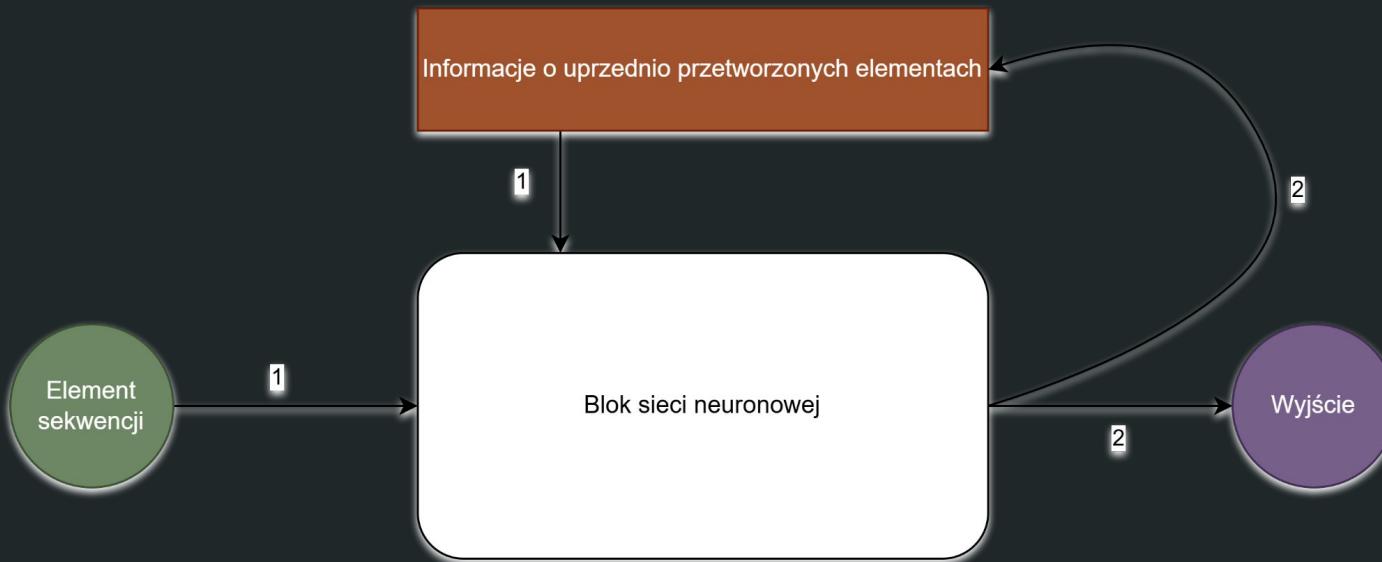
Jak sobie z tym poradzić?

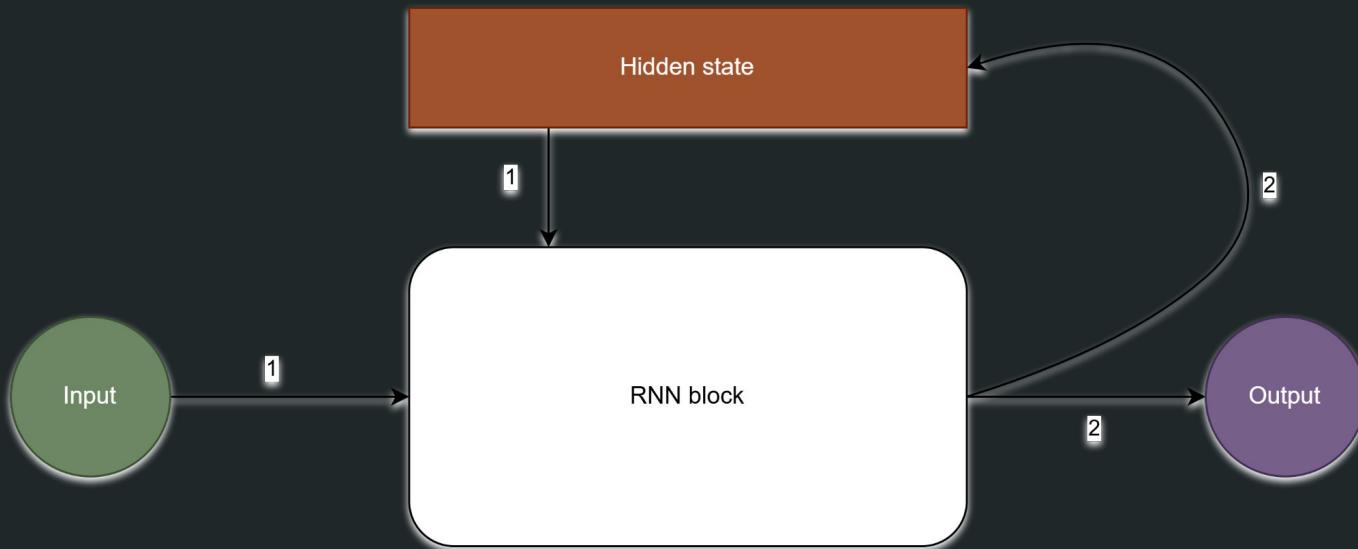
Potrzeba jakichś innych sieci...

Takich, które będą w stanie przetwarzać sekwencję element po elemencie.

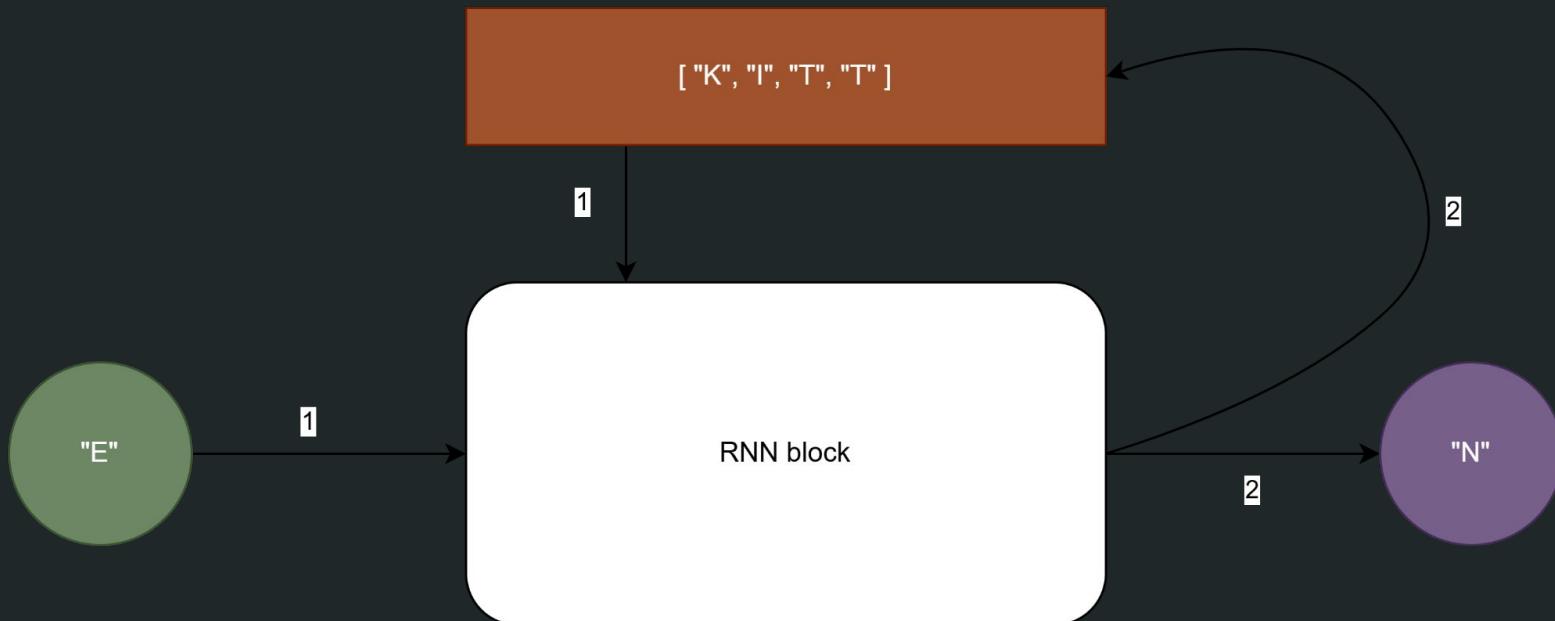
Takich, które na każdym “kroku” przetwarzania będą brały pod uwagę zarówno najnowsze dane wejściowe (bieżący element sekwencji), jak i te, które przetworzone zostały już poprzednio.

Recurrent Neural Networks (RNNs)









Sieci rekurencyjne

- RNN
- LSTM
- GRU
-

“Vanilla” RNN

- Każdy kolejny element sekwencji jest tak samo ważny
- W zdaniach poszczególne słowa mają różną “wagę” (The Lord of the Rings)
- Z każdym kolejnym elementem inputu, poprzednie coraz bardziej “znikają z pamięci”

Rozwiążanie: LSTM (Long-Short-Term Memory)

- Od teraz blok sieci odpowiadający za “pamiętanie” informacji z przeszłości będzie trochę bardziej skomplikowany (ale nie przejmujemy się tym, Keras już to za nas zaimplementował!)
- W każdym bloku będą teraz 3 komponenty:
 - Input gate: decyduje, które nowe informacje z aktualnego wejścia i poprzedniego stanu powinny zostać zapisane
 - Forget gate: decyduje, które informacje ze starego stanu komórki można usunąć
 - Output gate: decyduje, które informacje ze stanu komórki będą wyjściem sieci

Rozwiążanie (nowsze): GRU

- Od teraz blok sieci odpowiadający za “pamiętanie” informacji z przeszłości będzie nieco mniej skomplikowany niż w LSTM (Keras także już to za nas zaimplementował)
- W każdym bloku będą teraz 2 komponenty:
 - Reset gate: kontroluje, które części poprzedniego stanu ukrytego (stanu w pamięci) są ignorowane przy tworzeniu nowego kandydata na stan ukryty
 - Update gate: decyduje, które informacje z poprzedniego stanu zachować, a które zastąpić nowymi danymi

Trochę kodu

Next word(s) generation

Polecajki

- [3Blue1Brown](#) - youtube
- [Andrej Karpathy](#) - youtube
- [Hugging Face](#) - społeczność, repozytorium modeli
- [Weights&Biases](#) - platforma do logowania treningów
- (next level): [Attention is all you need](#) - papier o transformerach
- (next level): [An image is worth more than 16x16 patches](#) - papier o Visual Transformers