

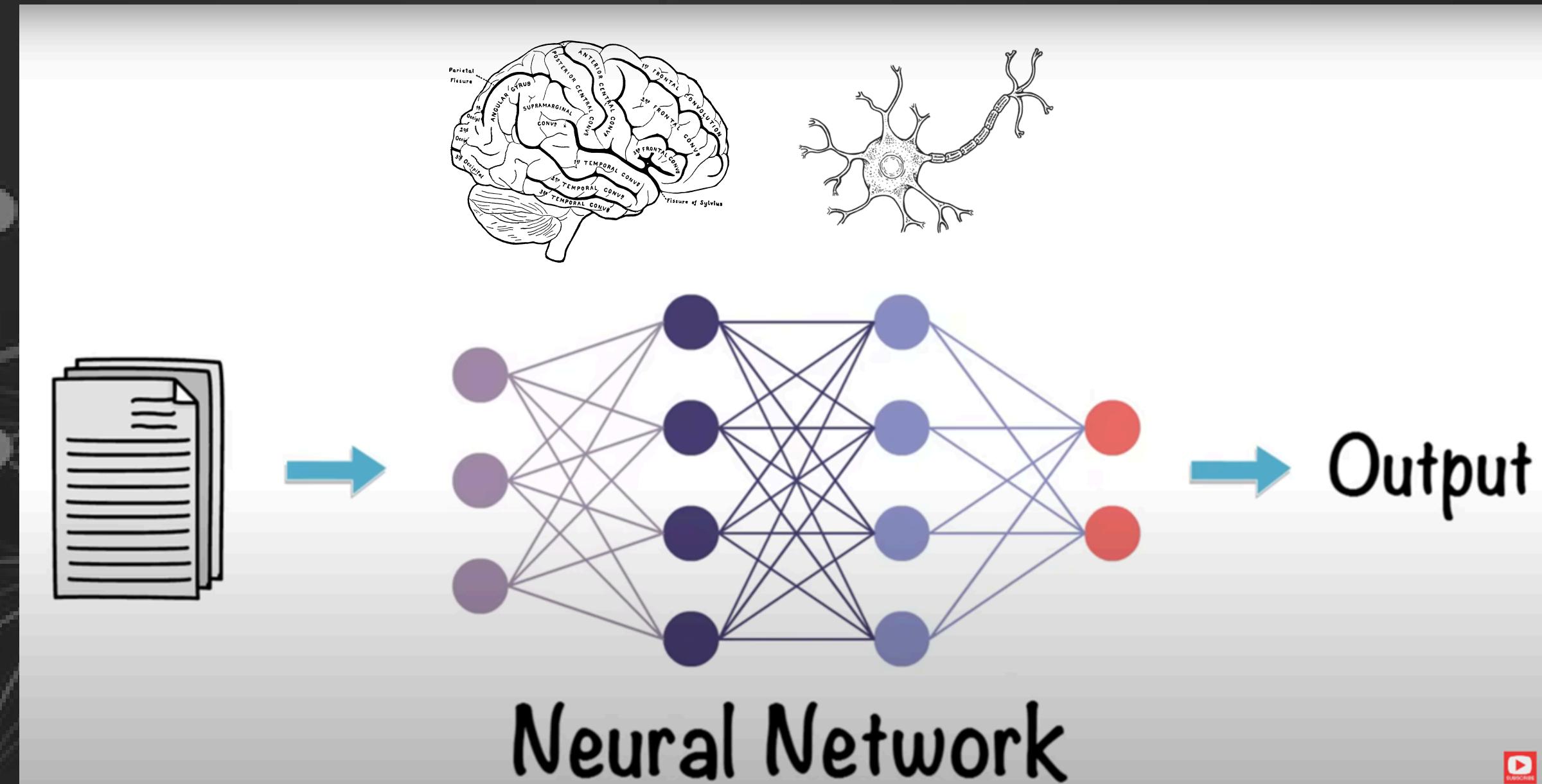
Charakterystyka wybranych sieci neuronowych

W analizie danych

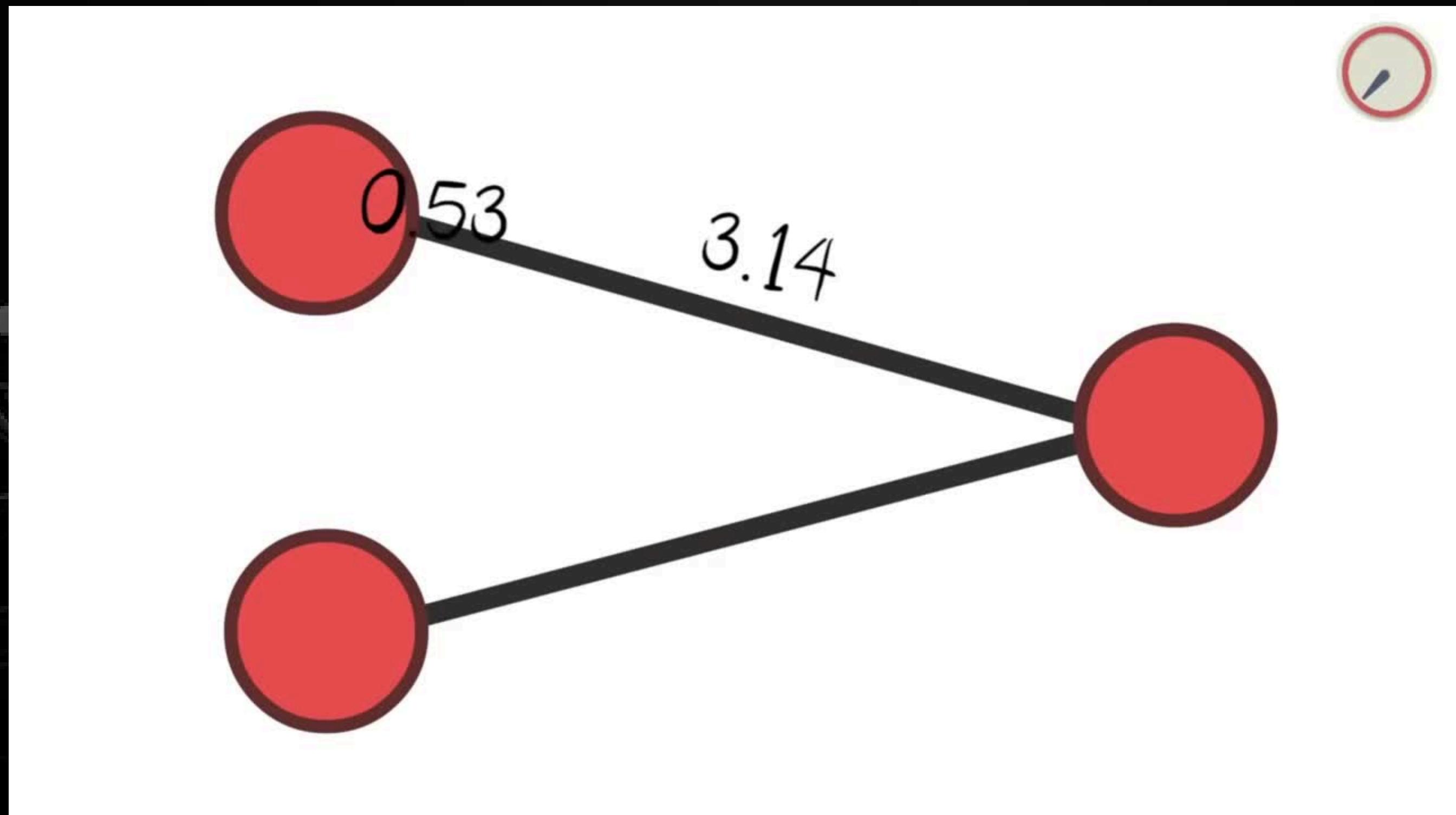
Schemat sieci neuronowej

Przykładowe dane:

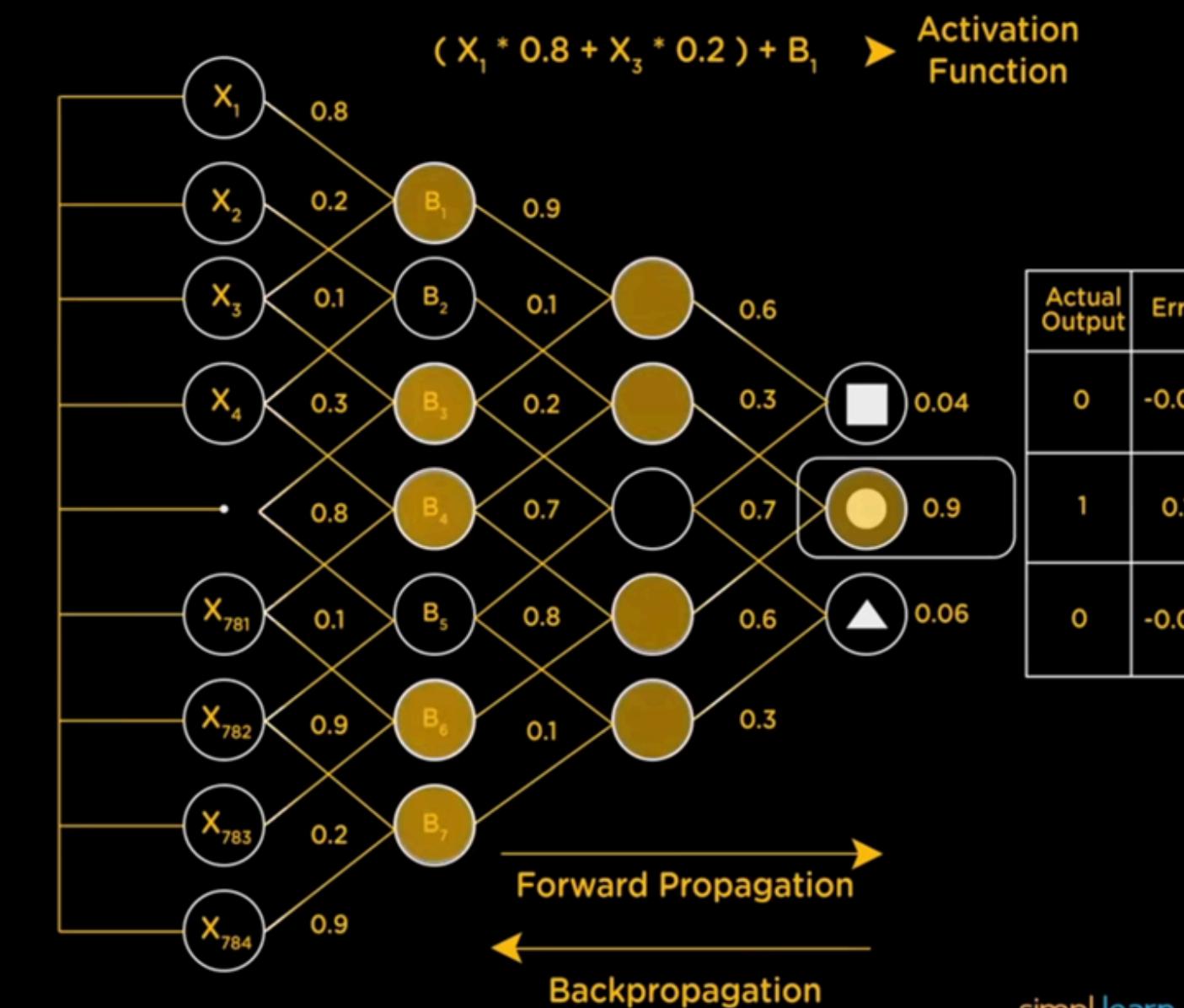
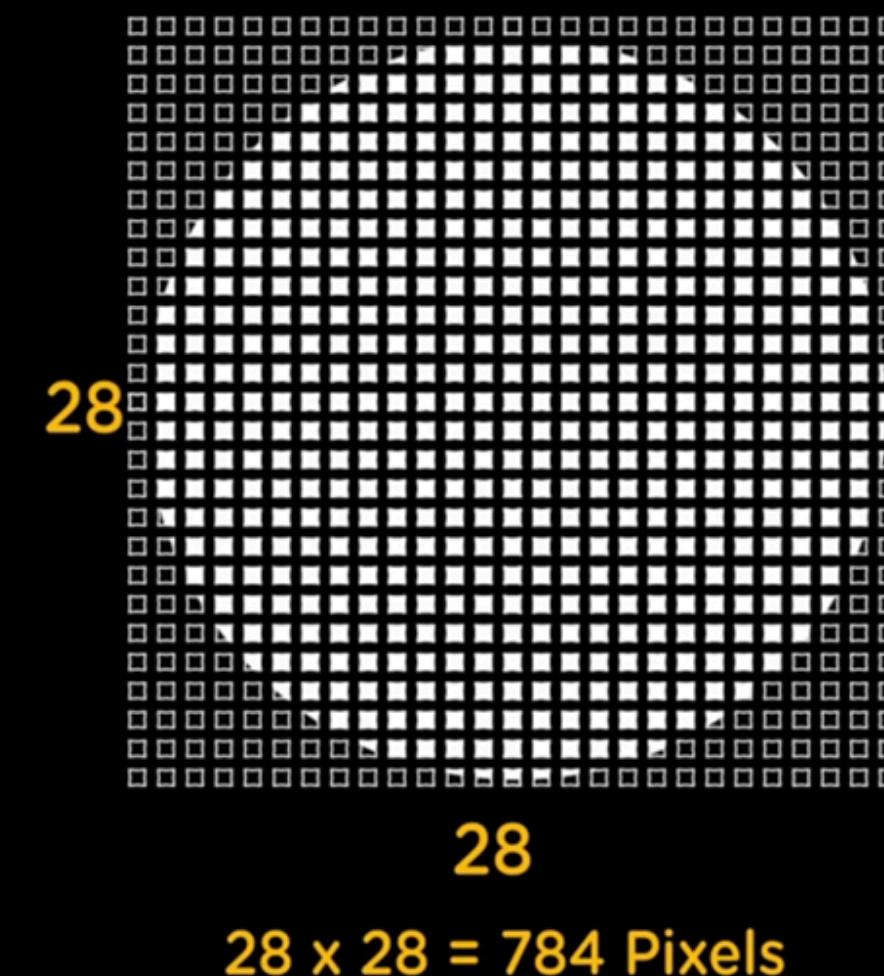
- Dane tabularyczne
(xlsx, csv,
pd.DataFrame)
- Dane tekstowe
(prompty)
- Grafika (jpg,png)



Wizualizacja działania wag i ładunków w sieci

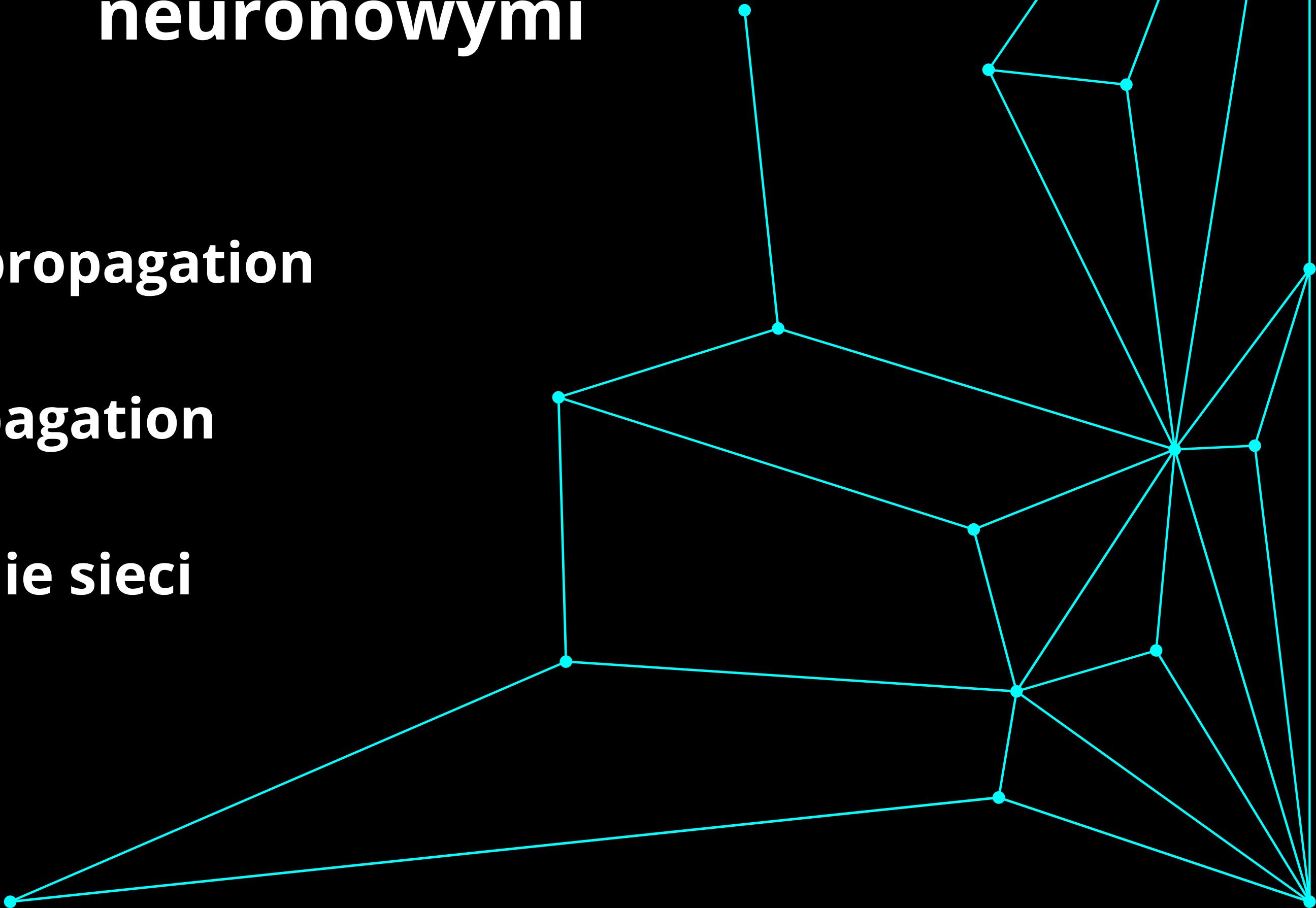


Przykład działania sieci neuronowej na prostym zadaniu klasyfikacji



Podstawowe pojęcia związane z sieciami neuronowymi

- Forward propagation
- Back propagation
- Trenowanie sieci



Forward propagation

Forward propagation (propagacja w przód) w sieciach neuronowych to proces, w którym dane wejściowe przechodzą przez kolejne warstwy sieci, a na końcu sieć generuje wynik.

Back propagation

Backward propagation (propagacja wsteczna) to proces, który ma na celu poprawienie wag w sieci neuronowej, by wynik sieci był jak najbardziej zbliżony do oczekiwanej. Działa to jak proces uczenia się na podstawie błędów. Jeśli wynik sieci jest błędny, propagacja wsteczna pomaga sieci „nauczyć się” lepszych wag.

Trenowanie sieci

Trenowanie sieci neuronowej to proces dostosowywania wag modelu,

- 1) Przekazanie wejścia przez sieć (Propagacja do przodu)**
- 2) Obliczenie błędu (Funkcja kosztu)**
- 3) Propagacja wsteczna (Backpropagation)**
- 4) Aktualizacja wag**

Rekurencyjne sieci neuronowe

Rekurencyjna sieć neuronowa (RNN) to typ sieci neuronowej , w której dane wyjściowe z poprzedniego kroku są podawane jako dane wejściowe do bieżącego kroku.

Przykłady zastosowania:

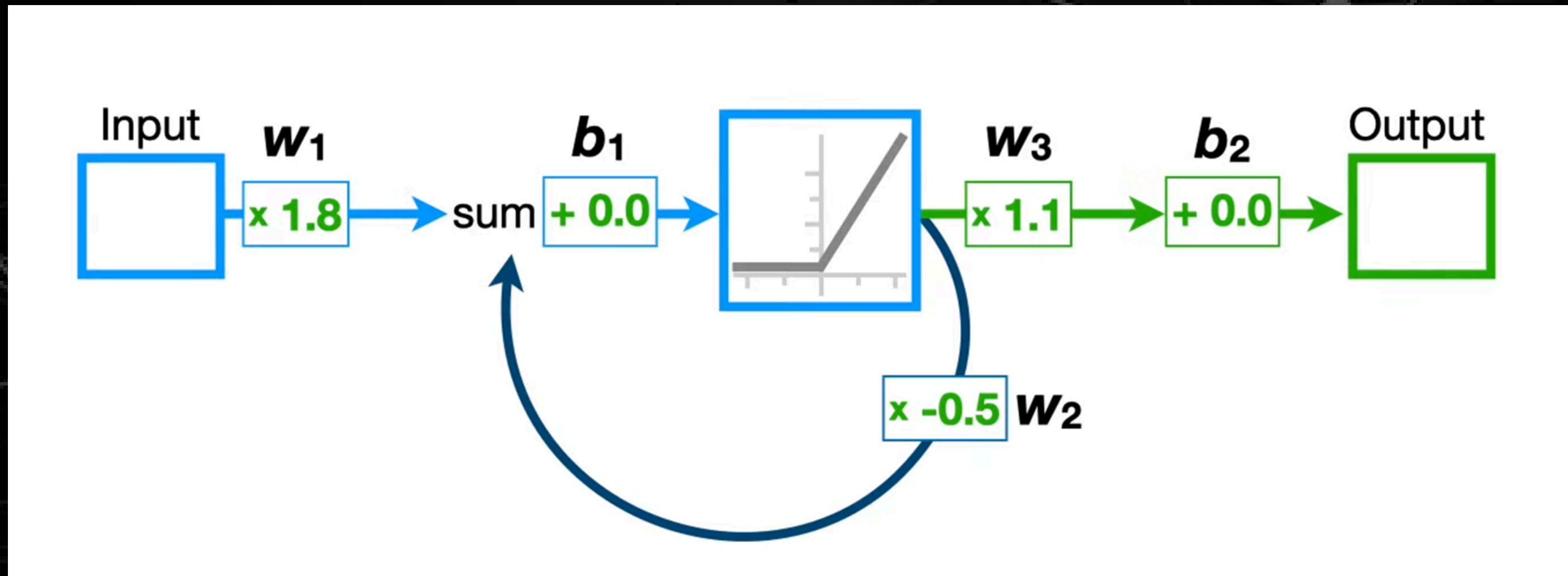
**Przetwarzanie
szeregów
czasowych i
prognozowanie(np.
przewidywanie cen
towarów)**

**Systemy Rekomendacyjne
(np. Netflix czy Spotify,
wykorzystują RNN do
personalizacji rekomendacji
na podstawie
wcześniejszego zachowania
użytkownika.**

**Systemy Rekomendacyjne
(np. Netflix czy Spotify,
wykorzystują RNN do
personalizacji
rekomendacji na
podstawie wcześniejszego
zachowania użytkownika.**

Bardziej szczegółowe omówienie architektury.

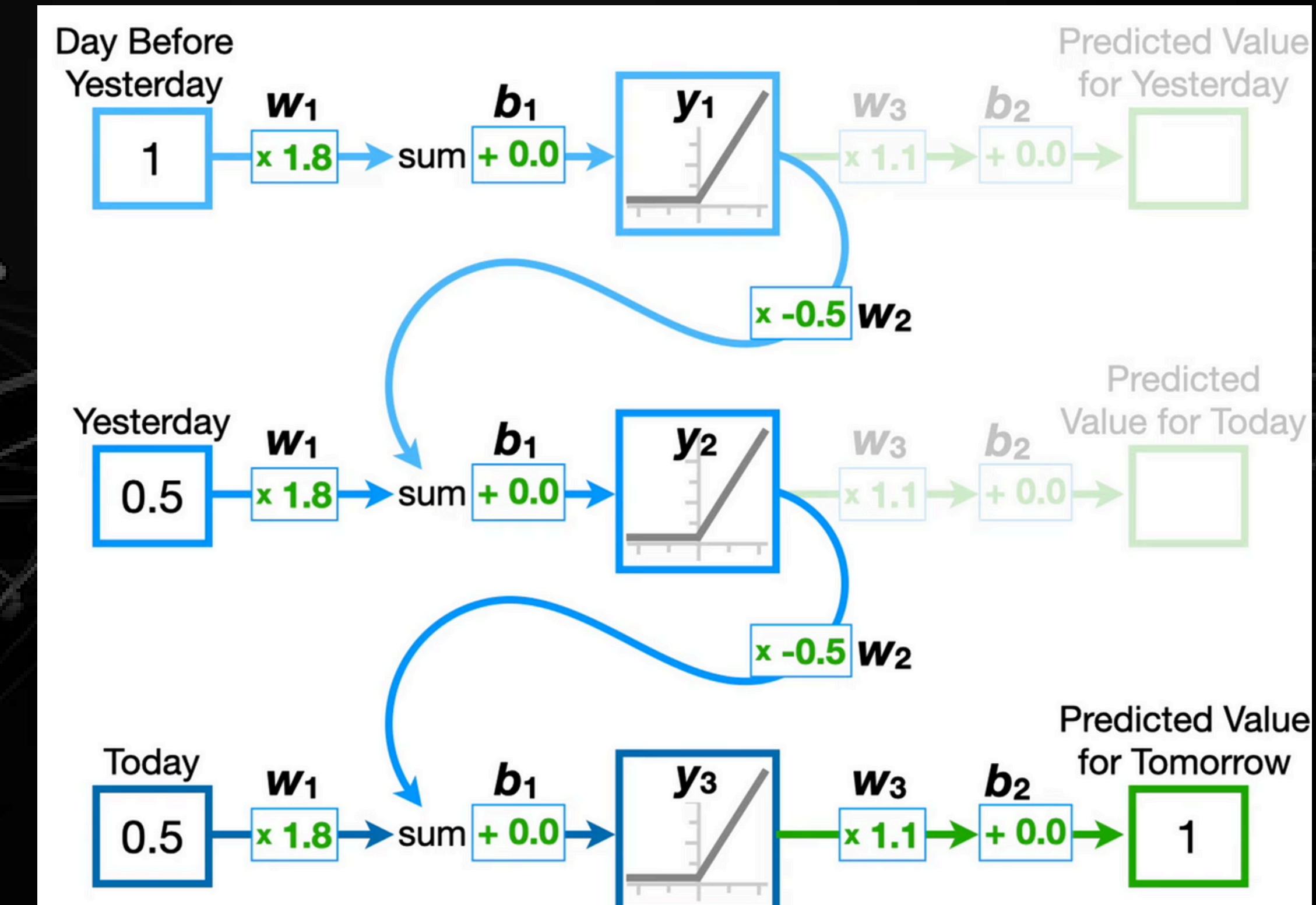
Tak samo jak inne sieci neuronowe, rekurencyjne również mają wagi, biasy, warstwy i funkcje aktywacji.



Ważnym elementem w tego rodzaju architekturze jest pętla na środku która będzie odwoływać się do poprzednich wartości



Przykład działania
tego typu sieci
na predykcji wartości
dla dnia jutrajszego.

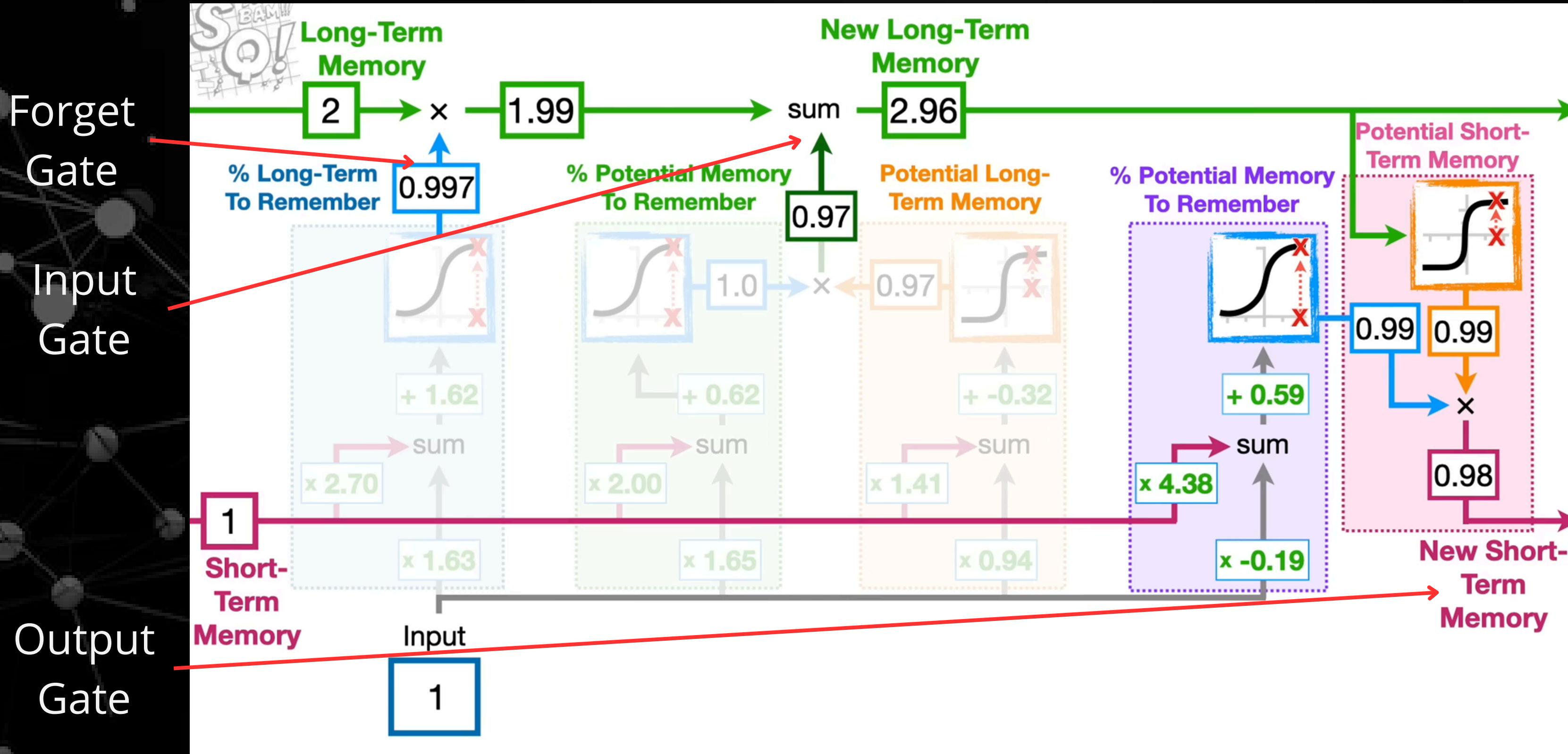


Sieci LSTM (Long Short-Term Memory)

- LSTM jest udoskonaloną wersją rekurencyjnej sieci neuronowej. Tradycyjny RNN ma jeden ukryty stan, który jest przekazywany w czasie, co może utrudniać sieci uczenie się długoterminowych zależności.
- Model LSTM, wprowadza komórkę pamięci, która jest pojemnikiem, który może przechowywać informacje przez dłuższy czas.

W porównaniu do zwykłych sieci rekurencyjnych architektura LSTM jest o wiele bardziej skomplikowana.

Zielona linia reprezentuje pamięć długoterminową, a fioletowa krótkoterminową.



Architektura pokazana na przykładzie przewidywania wartości dla kolejnych dni.

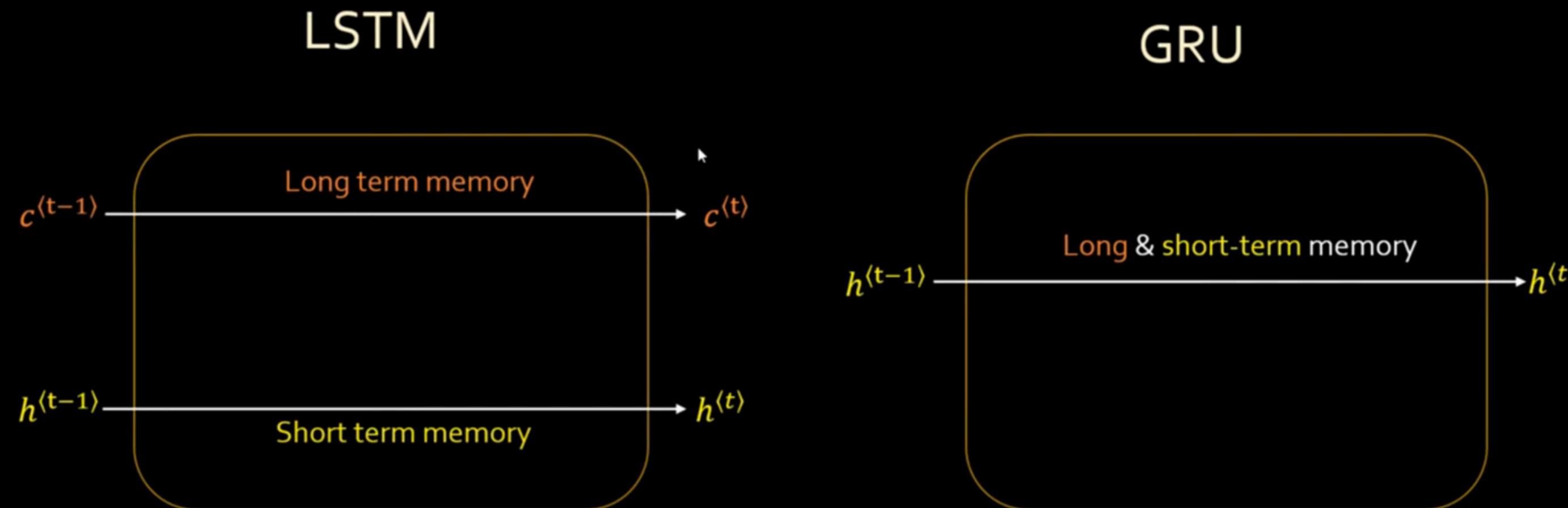


Predykcja na kolejny dzień

Sieci GRU (Gated Recurrent Unit)

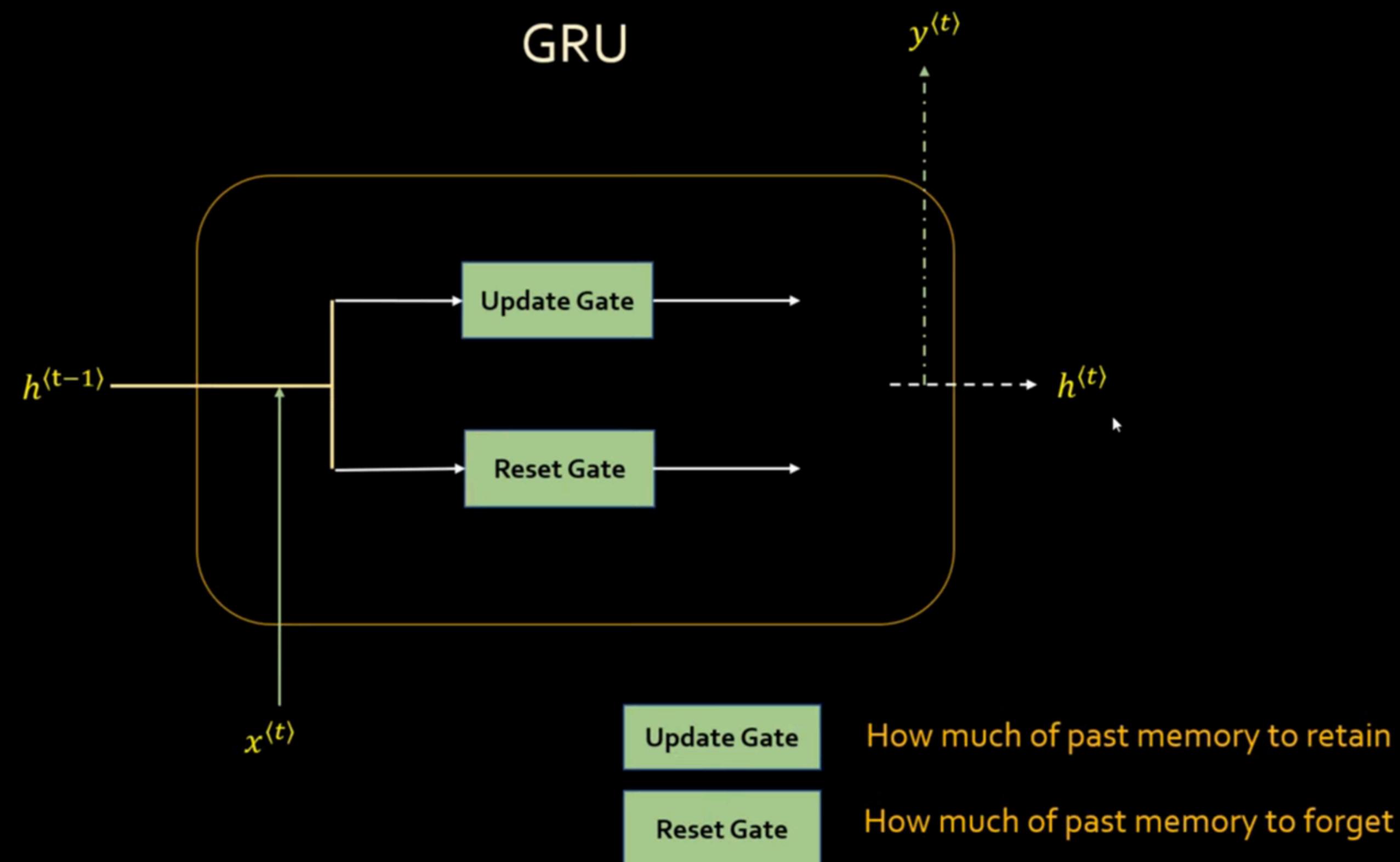
GRU są podobne do LSTM i dobrze radzą sobie z analizą sekwencji danych. Podobnie jak LSTM, GRU może przetwarzać dane sekwencyjne, takie jak tekst, mowa i dane szeregów czasowych. Tak samo jak w przypadku LSTM występuje tutaj mechanizm bramek i zapisywania informacji w pamięci.

Porównanie GRU i LSTM



W przypadku GRU mamy tylko jeden ukryty stan który odpowiada za aktualizację dugo i krótko terminowej pamięci

Schemat sieci GRU

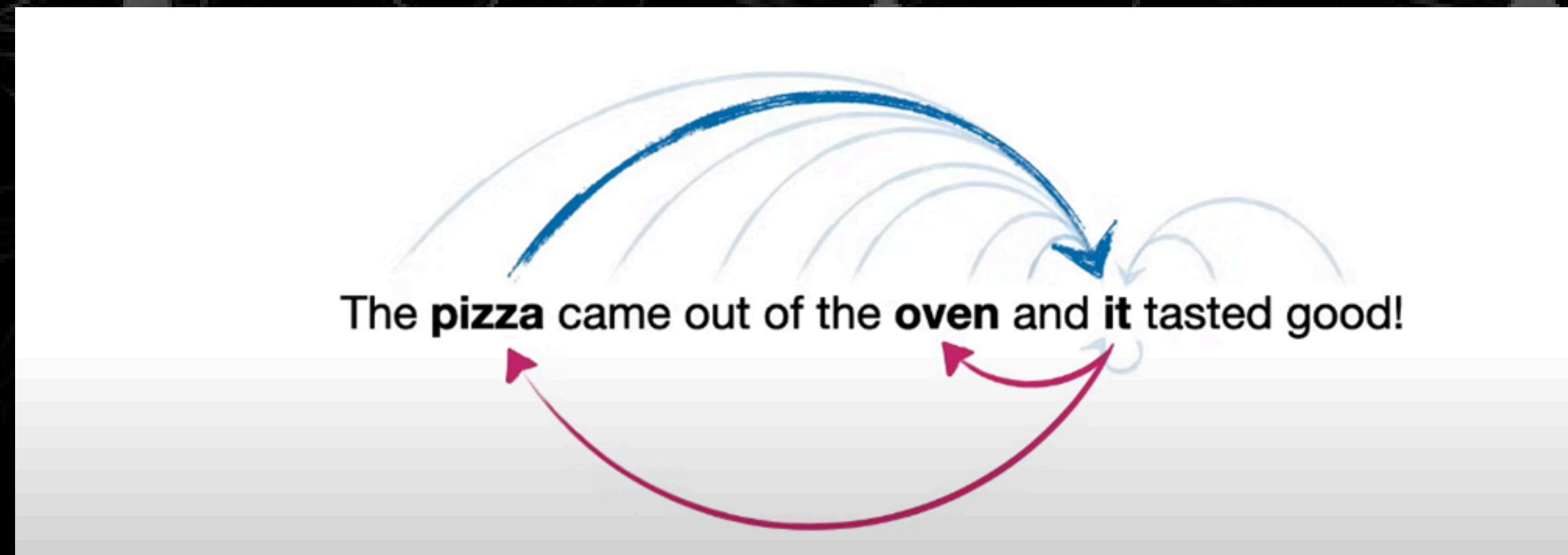


Sieci Transformer

- Transformery są nowoczesnym podejściem które zostało przedstawione w artykule „Attention is All You Need” w 2017 roku.
- Najpopularniejszym produktem wykorzystującym tę architekturę jest CHAT GPT (Generative Pre-trained Transformer).
- Główną innowacją w architekturze Transformer jest mechanizm uwagi, oraz BRAK struktury rekurencyjnej w tradycyjnym sensie.
- Pomimo iż GPT nie posiada formalnej struktury rekurencyjnej, mechanizm uwagi pozwala uwzględniać kontekst historyczny.

Self-attention

Mechanizm uwagi (attention) w transformerze pozwala modelowi skupić się na najistotniejszych częściach sekwencji wejściowej, niezależnie od ich pozycji. Proces ten działa dzięki tzw. self-attention. Każdy element sekwencji porównuje się z każdym innym, aby określić "wagę uwagi", czyli poziom ważności poszczególnych elementów względem siebie.



Transformery dla szeregów czasowych

Powstały pewne adaptacje transformerów specjalnie dla szeregów czasowych.

Temporal Fusion Transformer (TFT) (2019): Model, który uwzględnia zarówno zmienne czasowe, jak i statyczne. W Temporal Fusion Transformer mechanizm uwagi jest bardziej zaawansowany, ponieważ uwzględnia różne rodzaje danych czasowych (zmienne czasowe i statyczne) TFT wykorzystuje uwagę skupioną na czasie (time-based attention), która dynamicznie waży wpływ poszczególnych cech w różnych momentach.

Informer(2020): Zoptymalizowana wersja transformera zaprojektowana specjalnie do prognozowania długoterminowego szeregów czasowych. W Informerze zastosowano usprawniony mechanizm uwagi zwany sparse attention (uwaga rozrzedzona), który optymalizuje tradycyjny mechanizm self-attention, aby zmniejszyć jego złożoność obliczeniową. Informer wybiera tylko najbardziej znaczące wartości z długiej sekwencji, ignorując mniej istotne punkty czasowe, co pozwala modelowi szybciej przetwarzać duże ilości danych.



Dziękujemy za uwagę !