



GHOST

Group of Horribly Optimistic Statisticians



ML Next Step

#2 Sieci Neuronowe





GHOST

Group of Horribly Optimistic Statisticians





GHOST

Group of Horribly Optimistic Statisticians



Agenda

1. Czym są sieci
2. Podstawy i terminologia
3. PyTorch
4. Projekt

How Neural Network works?
Neurons:





GHOST

Group of Horribly Optimistic Statisticians

Podstawy



GHOST

Group of Horribly Optimistic Statisticians



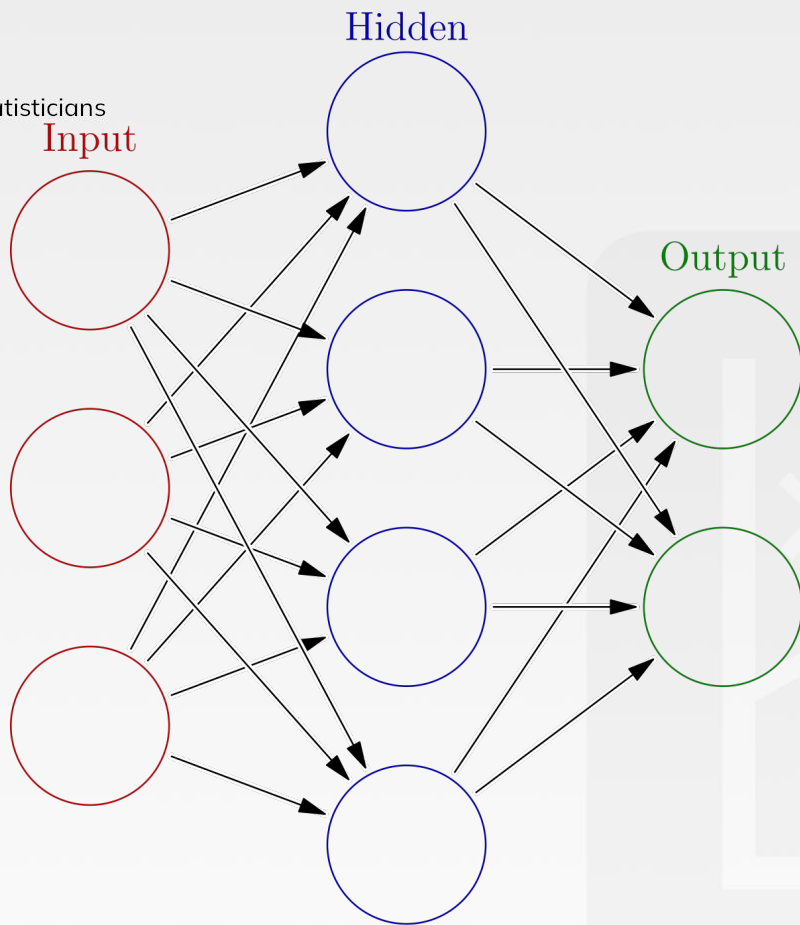
Czym są sieci neuronowe?

Sieci neuronowe to modele uczenia maszynowego inspirowane strukturą ludzkiego mózgu. Składają się z wielu prostych jednostek – sztucznych neuronów – które razem potrafią uczyć się złożonych zależności w danych. Działają poprzez przekształcanie wejścia (np. obrazu) w wyjście (np. etykietę klasy) za pomocą warstw matematycznych operacji.



GHOST

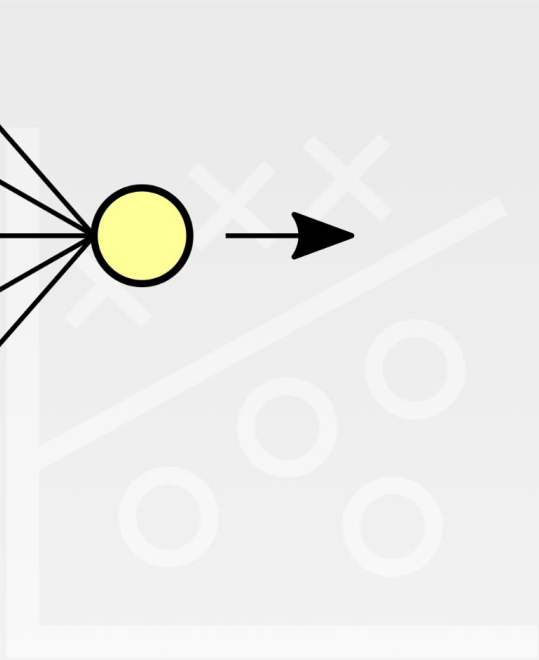
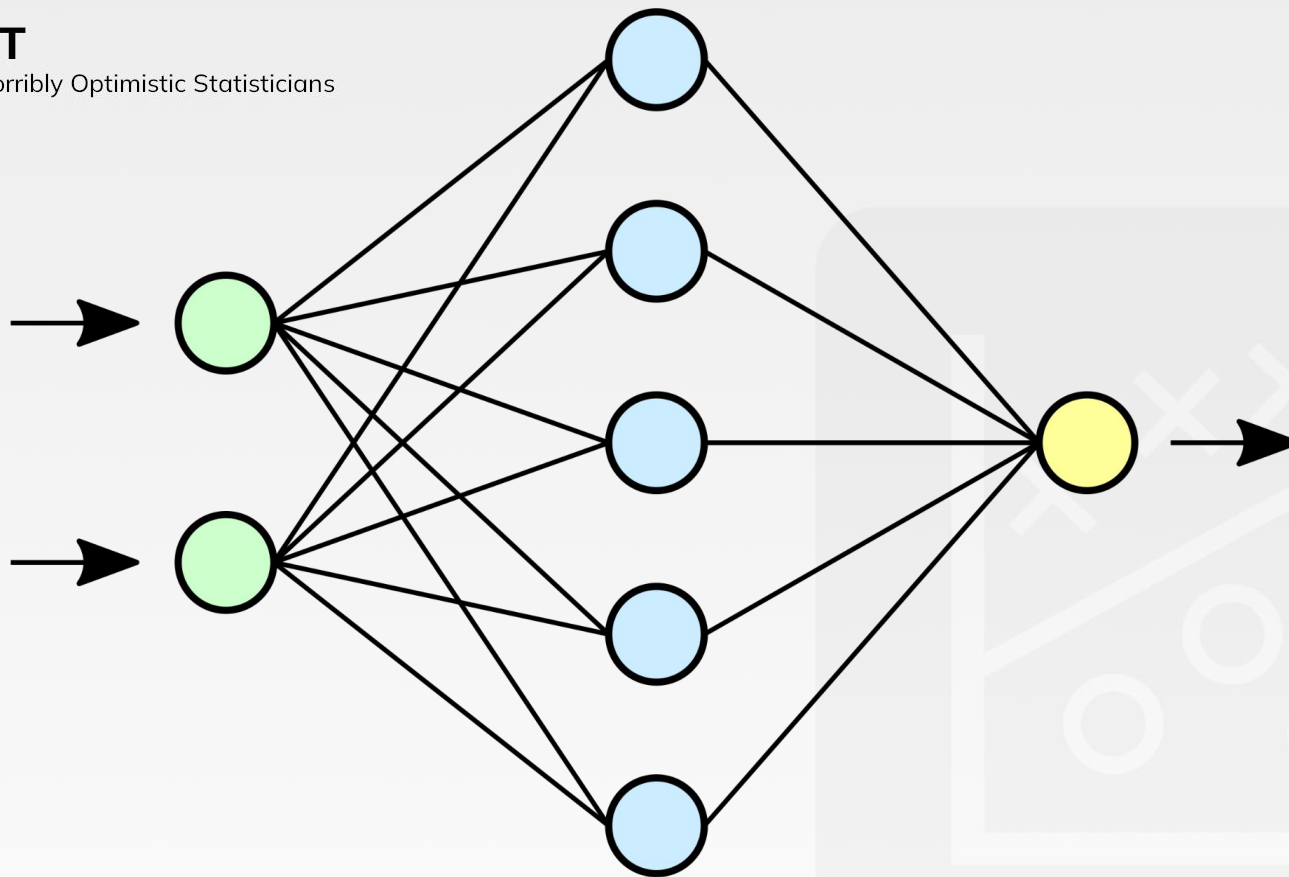
Group of Horribly Optimistic Statisticians





GHOST

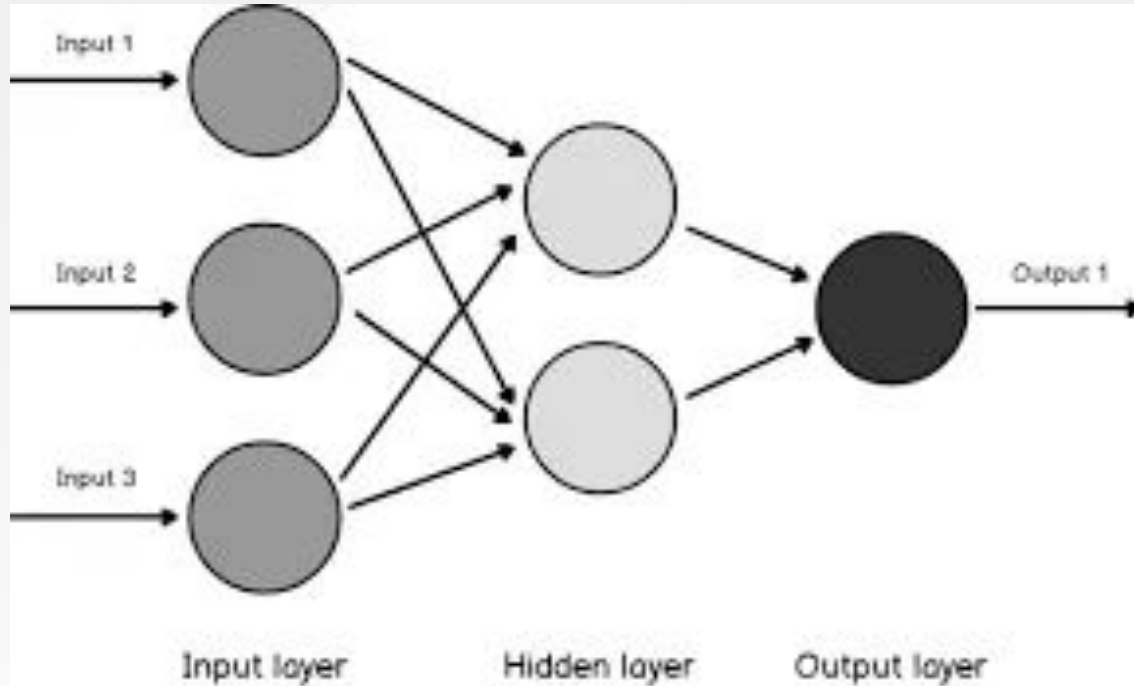
Group of Horribly Optimistic Statisticians





GHOST

Group of Horribly Optimistic Statisticians



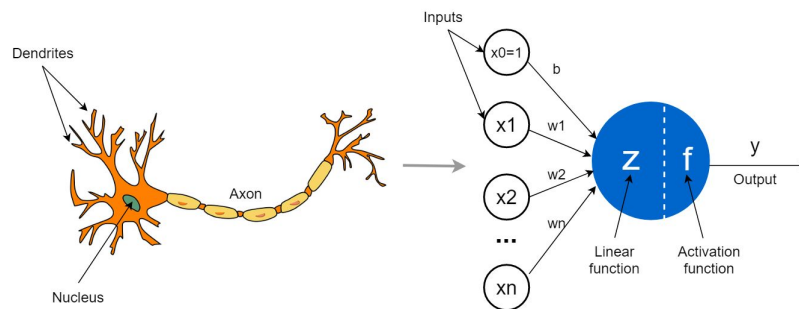
**GHOST**

Group of Horribly Optimistic Statisticians



Neuron

Neuron to najmniejsza jednostka w sieci neuronowej, odpowiedzialna za wykonywanie prostych operacji matematycznych. Przyjmuje dane wejściowe (liczby), które przelicza za pomocą przypisanych wag i dodanego biasu. W praktyce pojedynczy neuron jest bardzo prosty, ale wiele neuronów połączonych razem tworzy potężny model.

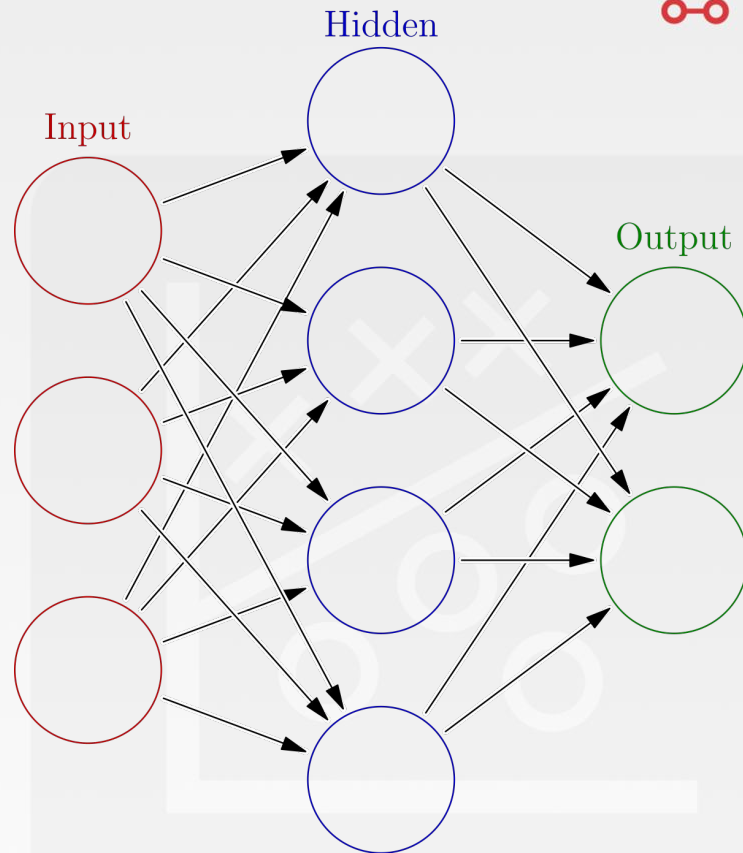


**GHOST**

Group of Horribly Optimistic Statisticians

Warstwa (layer)

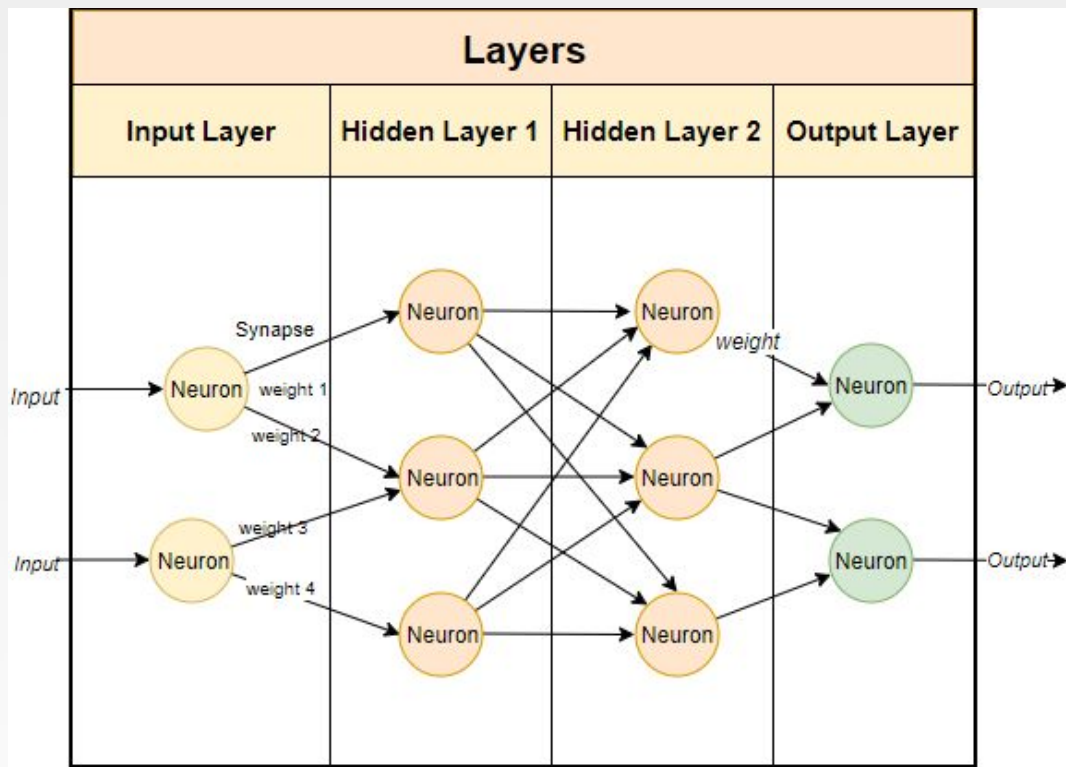
Warstwa w sieci neuronowej to zbiór wielu neuronów działających równoległe. Każdy neuron w warstwie wykonuje obliczenia niezależnie, ale wszystkie razem przekształcają dane w nową formę. Pierwsza warstwa w sieci przyjmuje dane wejściowe (np. piksele obrazka), warstwy ukryte odpowiadają za "przetwarzanie" i szukanie wzorców, a warstwa wyjściowa daje końcowy wynik. Liczba warstw i neuronów wpływa na złożoność i możliwości sieci.





GHOST

Group of Horribly Optimistic Statisticians



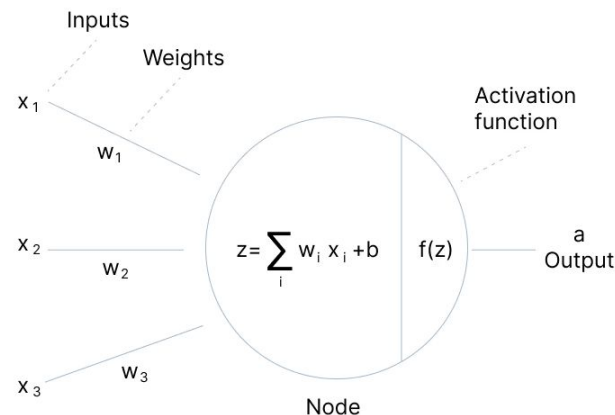
**GHOST**

Group of Horribly Optimistic Statisticians



Funkcja aktywacji

Funkcja aktywacji to specjalna matematyczna funkcja stosowana po każdej warstwie w sieci. Jej zadaniem jest wprowadzenie nieliniowości, czyli umożliwienie sieci wykrywania skomplikowanych wzorców i zależności w danych. Bez funkcji aktywacji sieć zachowywałaby się jak zwykła funkcja liniowa, a to oznaczałoby bardzo ograniczone możliwości uczenia. Istnieje wiele różnych funkcji aktywacji, a wybór odpowiedniej ma duży wpływ na skuteczność modelu.



V7 Labs



GHOST

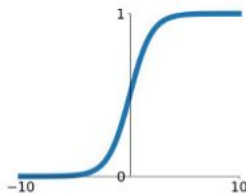
Group of Horribly Optimistic Statisticians



Activation Functions

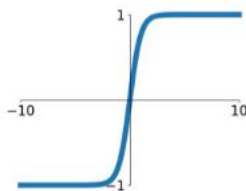
Sigmoid

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



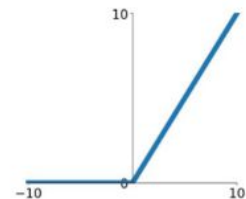
tanh

$$\tanh(x)$$



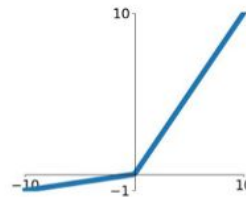
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

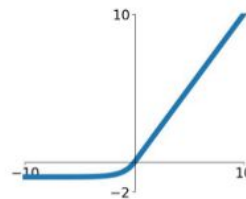


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



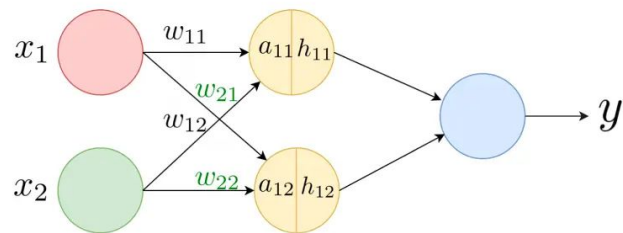
**GHOST**

Group of Horribly Optimistic Statisticians



Wagi (weights)

Wagi to najważniejsze parametry uczone przez sieć neuronową podczas treningu. Każde połączenie między neuronami posiada swoją wagę, która określa, jak silny jest przepływ informacji w tym miejscu. Na początku wagi są losowe, ale z czasem sieć dostosowuje je, aby lepiej radzić sobie z przewidywaniem wyników. Właśnie wartości wag przechowują "wiedzę" modelu o danych.



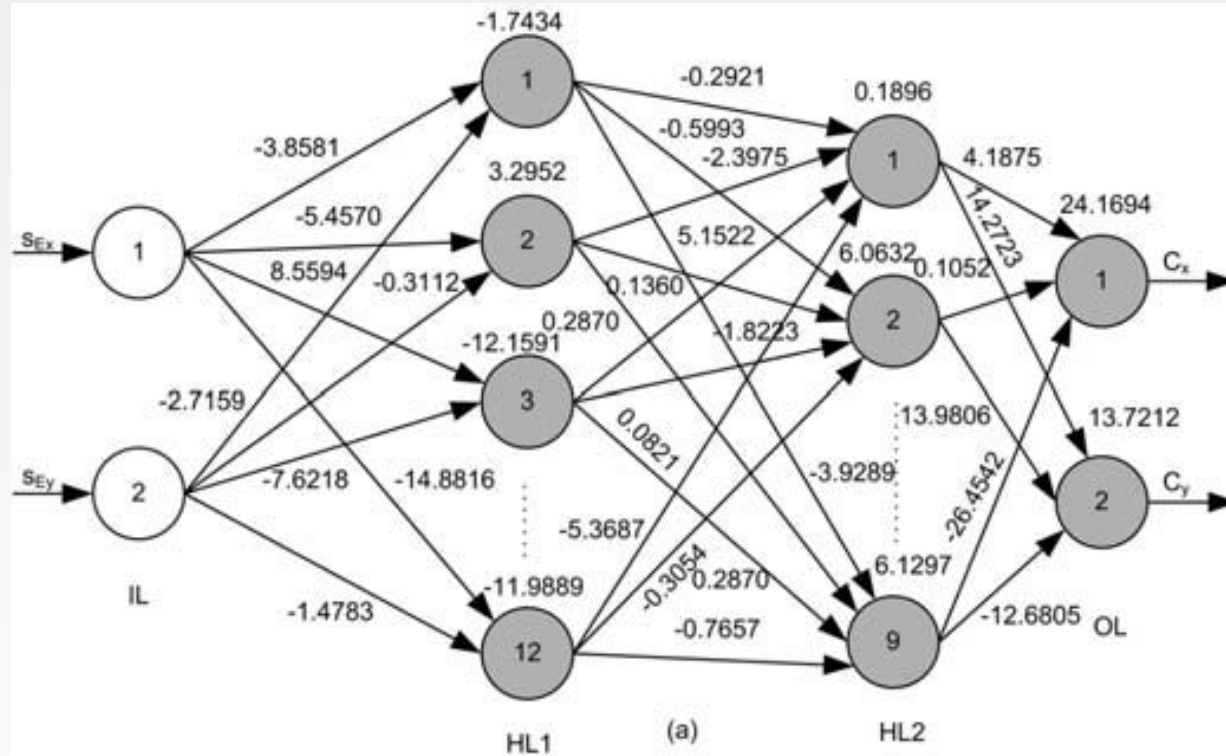
$$a_{11} = w_{11}x_1 + w_{12}x_2$$

$$a_{12} = w_{21}x_1 + w_{22}x_2$$



GHOST

Group of Horribly Optimistic Statisticians



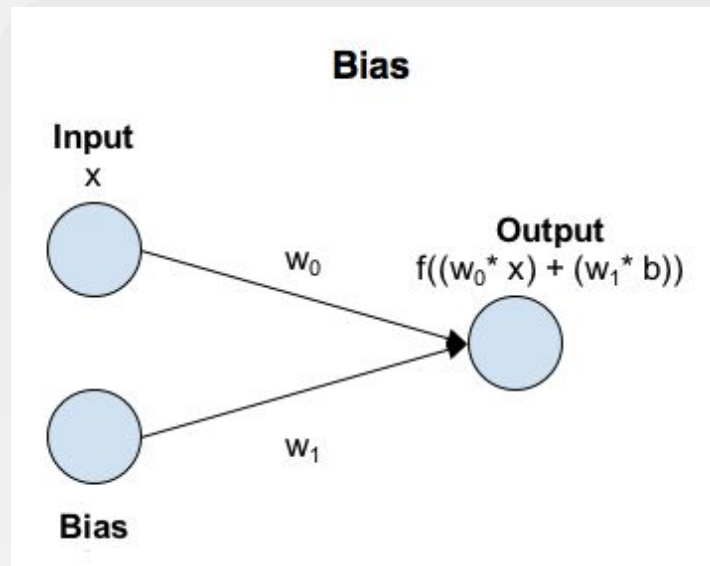
**GHOST**

Group of Horribly Optimistic Statisticians



Bias (przesunięcie)

Bias, to dodatkowy parametr w każdym neuronie. Dzięki niemu nawet jeśli wszystkie dane wejściowe mają wartość 0, neuron może zwrócić jakąś wartość wyjściową. Bias pozwala modelowi lepiej dopasować się do danych i zwiększa jego elastyczność. Umożliwia np. przesunięcie funkcji aktywacji w poziomie, co pomaga znaleźć lepsze rozwiązanie.



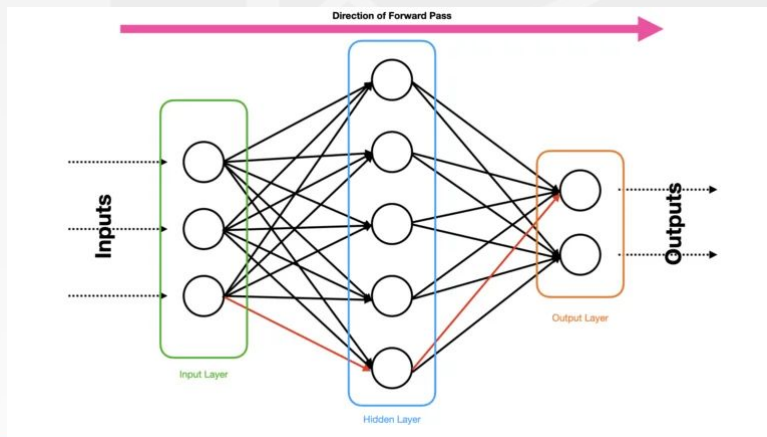
**GHOST**

Group of Horribly Optimistic Statisticians



Forward pass

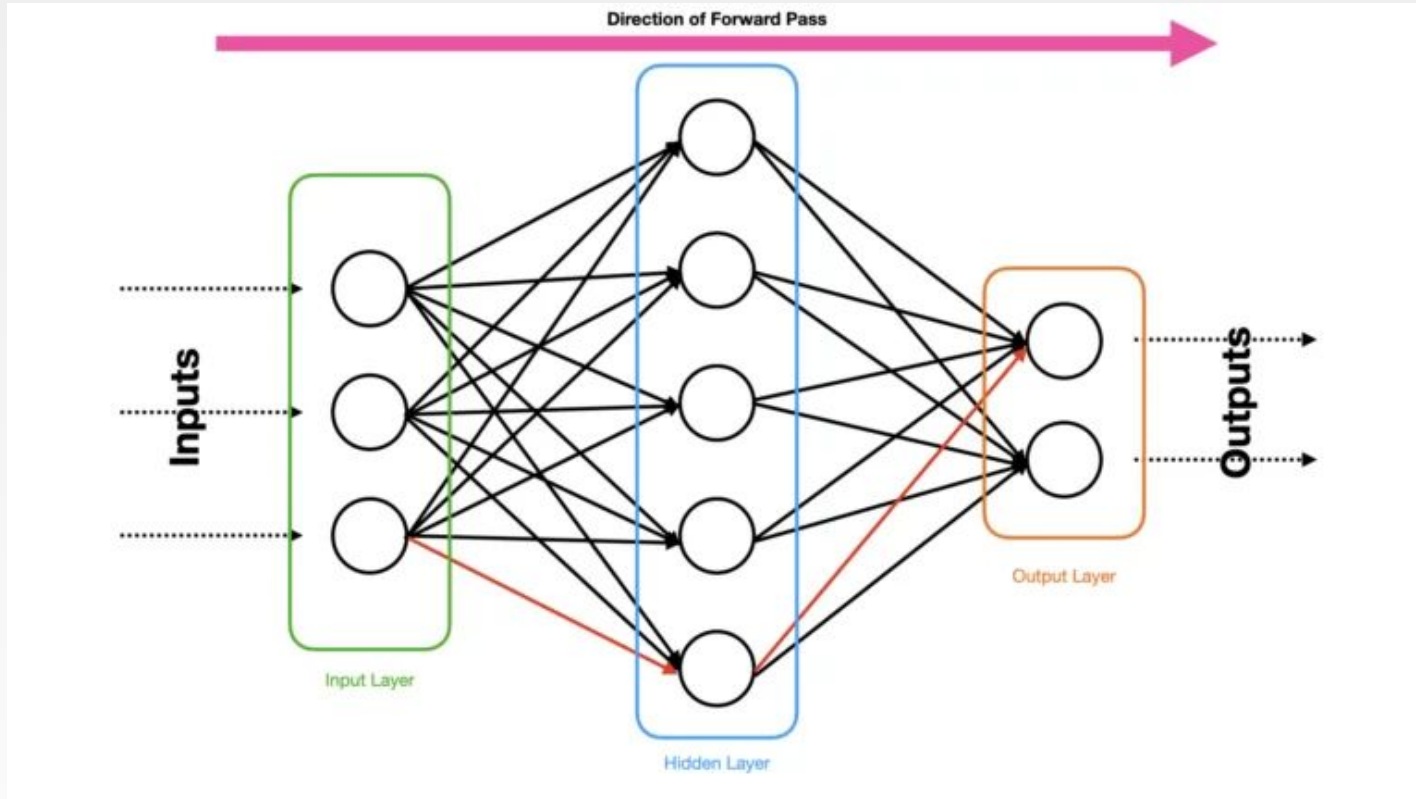
Forward pass to etap, w którym dane przepływają przez sieć neuronową – od wejścia aż do warstwy wyjściowej. W tym procesie kolejne warstwy wykonują obliczenia, przekształcają dane i przekazują je dalej. Na końcu sieć zwraca wynik, który jest prognozą dla danego przykładu. Forward pass jest wykonywany zarówno w trakcie treningu, jak i podczas używania gotowego modelu.





GHOST

Group of Horribly Optimistic Statisticians



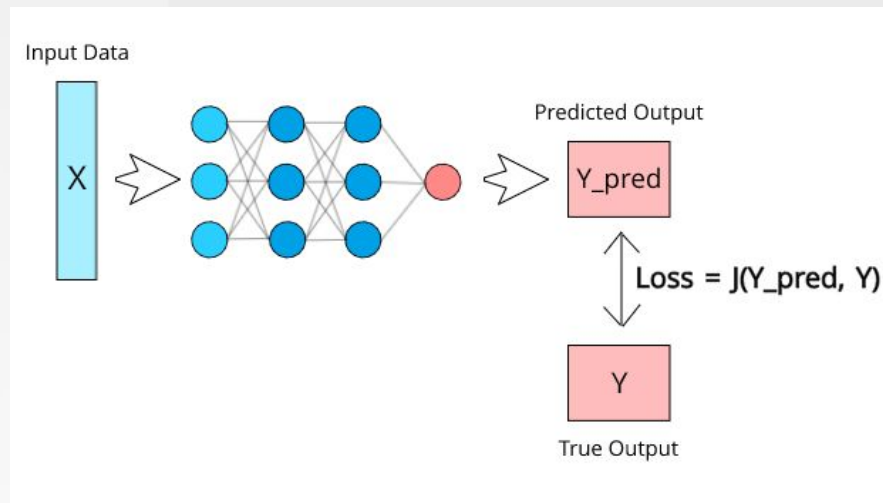
**GHOST**

Group of Horribly Optimistic Statisticians



Loss function (funkcja straty)

Funkcja straty to sposób na zmierzenie, jak bardzo model się myli. Porównuje przewidywany wynik sieci z rzeczywistym wynikiem i zwraca liczbę, która opisuje ten błąd. Im większa strata, tym gorzej działa model – dlatego celem treningu jest jej minimalizacja. W zależności od problemu stosuje się różne funkcje straty – inne dla klasyfikacji, inne dla regresji.



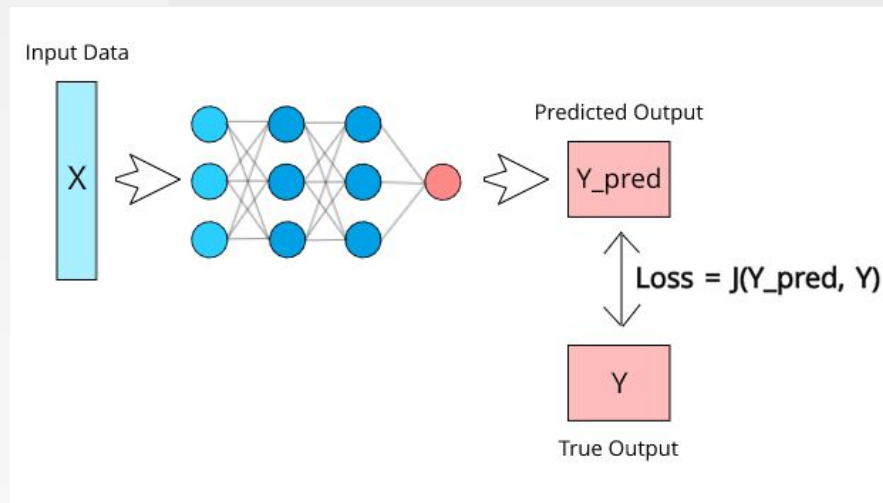
**GHOST**

Group of Horribly Optimistic Statisticians



Loss function (funkcja straty)

Funkcja straty to sposób na zmierzenie, jak bardzo model się myli. Porównuje przewidywany wynik sieci z rzeczywistym wynikiem i zwraca liczbę, która opisuje ten błąd. Im większa strata, tym gorzej działa model – dlatego celem treningu jest jej minimalizacja. W zależności od problemu stosuje się różne funkcje straty – inne dla klasyfikacji, inne dla regresji.



**GHOST**

Group of Horribly Optimistic Statisticians



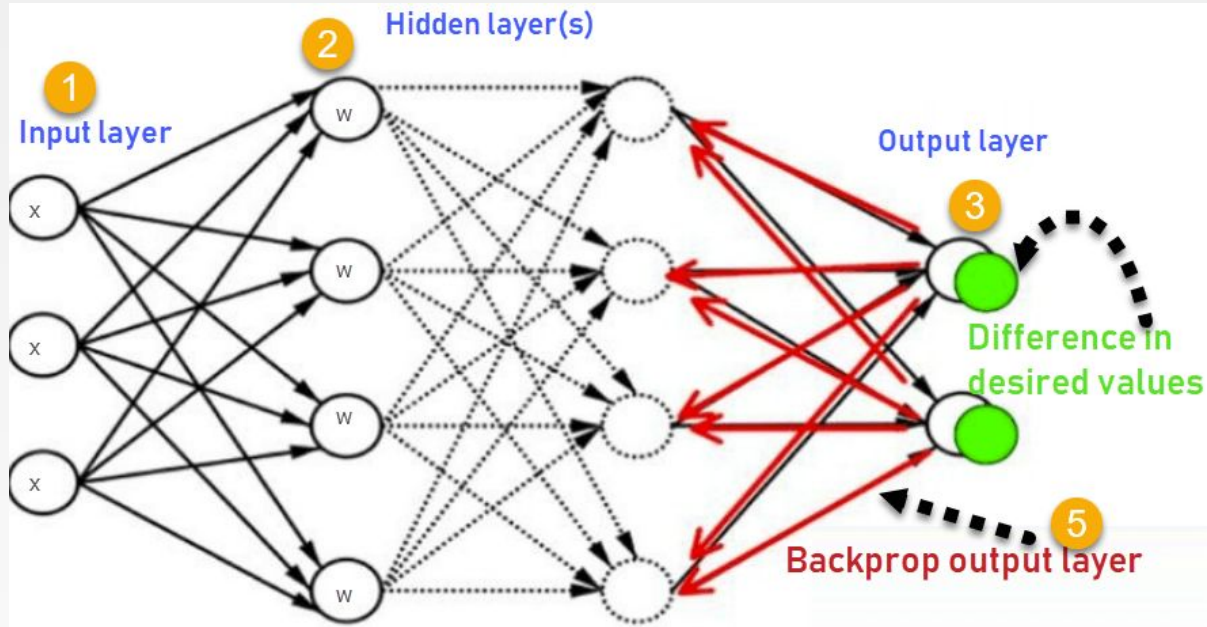
Backpropagation

Backpropagation to kluczowy algorytm używany do uczenia sieci neuronowych. Polega na tym, że po obliczeniu błędu (czyli straty) w wyniku forward pass, ten błąd jest "cofany" wstecz przez wszystkie warstwy sieci. Dzięki regule łańcuchowej rachunku różniczkowego możemy obliczyć, jak zmiana każdej wagi wpłynęłaby na końcowy błąd. Dla każdej wagi wyznaczany jest gradient funkcji straty względem tej wagi. Te gradienty mówią modelowi, w którą stronę należy zmienić wagi, by zmniejszyć błąd. W połączeniu z optymalizatorem (np. Adam), backpropagation pozwala modelowi uczyć się coraz lepiej z każdą iteracją.



GHOST

Group of Horribly Optimistic Statisticians



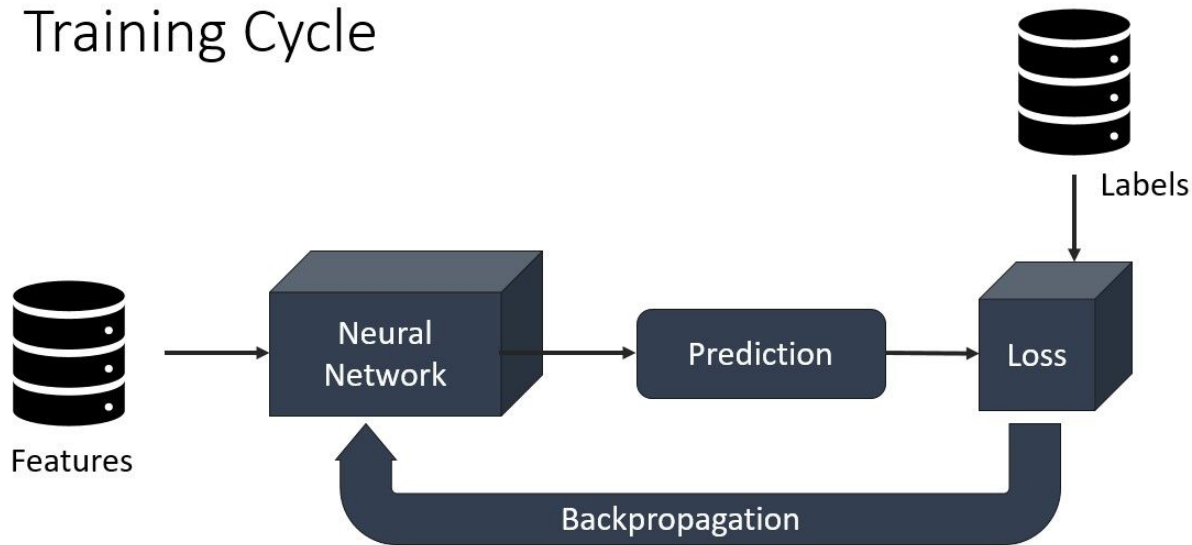


GHOST

Group of Horribly Optimistic Statisticians



Training Cycle



**GHOST**

Group of Horribly Optimistic Statisticians



Gradient

Gradient to pochodna funkcji straty względem danej wagi lub biasu w modelu. Informuje nas, jak mocno i w jakim kierunku trzeba zmienić daną wagę, aby zmniejszyć wartość błędu. Jeżeli gradient jest duży, to znaczy, że zmiana tej wagi ma duży wpływ na wynik – i trzeba ją odpowiednio dostosować. Gradienty są podstawą metody optymalizacji — bez nich model nie wiedziałby, jak poprawiać swoje parametry. W praktyce gradienty są automatycznie wyliczane przez framework (np. PyTorch) za pomocą wbudowanego mechanizmu autograd.

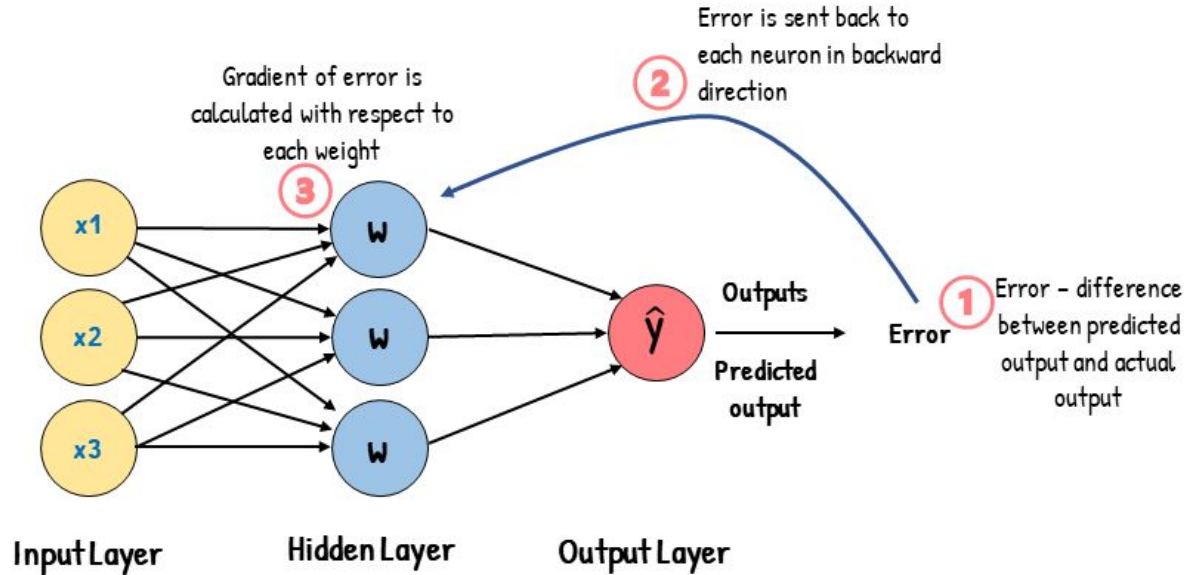


GHOST

Group of Horribly Optimistic Statisticians



Backpropagation





GHOST

Group of Horribly Optimistic Statisticians



Optymalizator

Optymalizator to algorytm, który na podstawie gradientów aktualizuje wagi w sieci neuronowej. Jego zadaniem jest znaleźć najlepszy zestaw wag, który minimalizuje funkcję straty. Popularne optymalizatory to SGD (prosty) i Adam (bardziej zaawansowany i szybciej działający). To właśnie dzięki optymalizatorowi model faktycznie "uczy się" na danych.



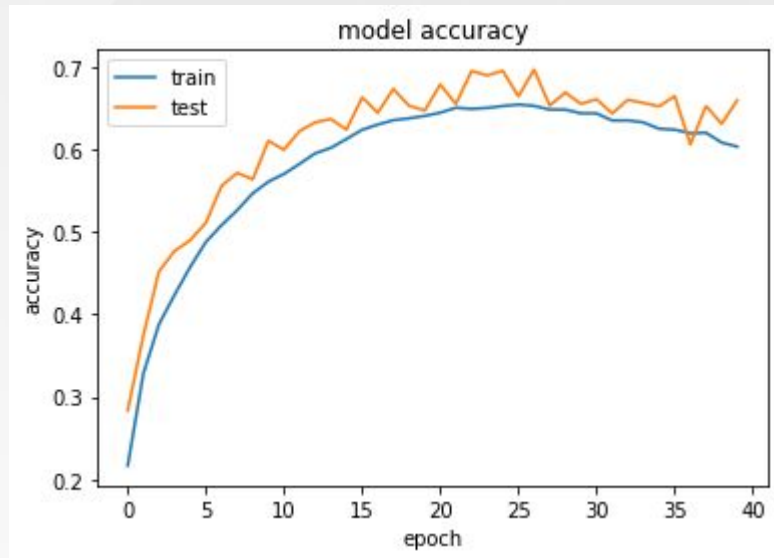
**GHOST**

Group of Horribly Optimistic Statisticians



Epoch (epoka)

Epoka to jedno pełne przejście przez cały zbiór treningowy. W czasie jednej epoki sieć widzi wszystkie dane raz, dzielone na mniejsze porcje (batch'e). Po każdej epoce zazwyczaj oceniamy, czy model poprawił się – obserwując np. spadek straty lub wzrost dokładności. Sieć zazwyczaj trenuje przez wiele epok, aby miała czas "nauczyć się" wystarczająco dobrze wzorców w danych.





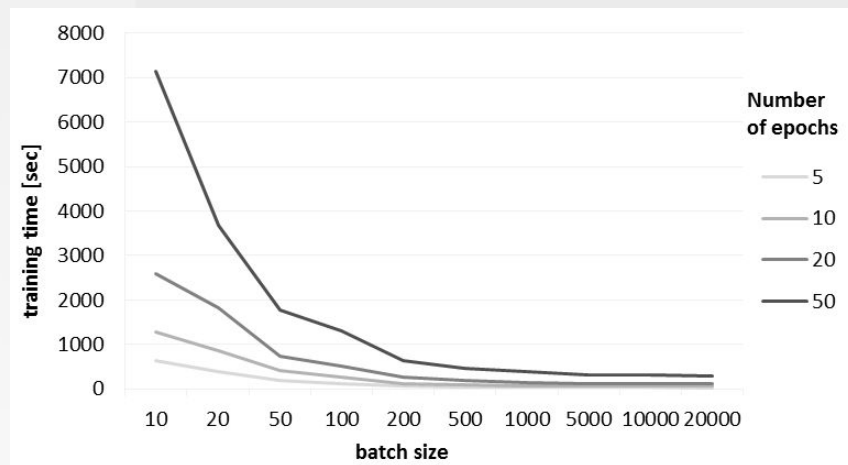
GHOST

Group of Horribly Optimistic Statisticians



Batch

Batch to niewielka porcja danych treningowych przetwarzana naraz. Zamiast przeliczać całą sieć dla tysięcy przykładów na raz, dzielimy dane na mniejsze zestawy (np. po 32 lub 64 elementy), aby model mógł trenować szybciej i stabilniej. Dzięki używaniu batchy sieć uczy się bardziej uogólnionych cech, a cały proces jest bardziej wydajny pamięciowo.





GHOST

Group of Horribly Optimistic Statisticians

PyTorch



GHOST

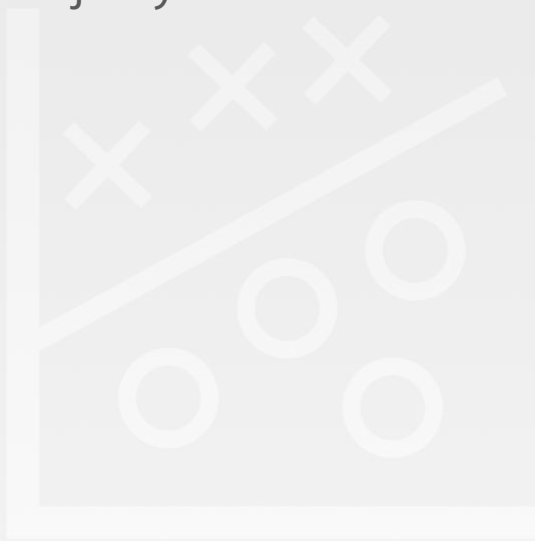
Group of Horribly Optimistic Statisticians



Co jest potrzebne do sieci w PyTorchu?

Aby zbudować sieć neuronową w PyTorchu, potrzebujemy kilku elementów:

- dane (wejścia i etykiety),
- model (czyli architektura sieci),
- funkcja straty,
- optymalizator,
- pętla treningowa.





GHOST

Group of Horribly Optimistic Statisticians

Projekt



GHOST

Group of Horribly Optimistic Statisticians



Przewidywanie ceny kawy w kawiarni (Regresja)

Sieć przewiduje cenę kawy na podstawie jej parametrów.

Dane (samodzielnie przygotowane lub sztuczne):

- wielkość kubka (mała/średnia/duża)
- ilość dodatków (0-3)
- czy na wynos (0/1)

Wynik: Cena kawy w złotych (liczba float)





GHOST

Group of Horribly Optimistic Statisticians



Rozpoznawanie dnia tygodnia po godzinach aktywności

Sieć próbuje zgadnąć, jaki to dzień tygodnia, patrząc na to, ile ktoś aktywnie działał o różnych porach.

Dane (samodzielnie przygotowane lub sztuczne):

- Aktywność rano, popołudniu, wieczorem
- Ilość godzin spania
- Ilość godzin przy nauce/pracy

Wynik: Dzień tygodnia (0-6)





GHOST

Group of Horribly Optimistic Statisticians



Klasyfikacja obrazków: Kółko vs Kwadrat vs Trójkąt

Sieć przewiduje figurę.

Dane (samodzielnie przygotowane lub sztuczne):

- Obrazki ze zdjęciami

Wynik: Etykieta - Kółko/Kwadrat/Trójkąt





GHOST

Group of Horribly Optimistic Statisticians



Przewidywanie wyniku w grze Kamień, Papier, Nożyce

Sieć uczy się przewidywać co najczęściej wybiera przeciwnik.

Dane (samodzielnie przygotowane lub sztuczne):

- Ostatnie 3 ruchy gracza
- Output: Co prawdopodobnie zagra teraz

Wynik: Etykieta - kamień / papier / nożyce





GHOST

Group of Horribly Optimistic Statisticians



Przewidywanie oceny filmu po danych o użytkowniku

Sieć uczy się przewidywać czy użytkownik polubi film.

Dane (samodzielnie przygotowane lub sztuczne):

- Wiek użytkownika
- Gatunek filmu
- Czas trwania filmu
- Popularność filmu

Wynik: Etykieta - polubi lub nie polubi





GHOST

Group of Horribly Optimistic Statisticians



Rozpoznawanie cyfr (MNIST)

Zbuduj sieć neuronową, która rozpoznaje odręcznie napisane cyfry od 0 do 9 ze zbioru MNIST. To jeden z najbardziej znanych i najprostszych zestawów danych w uczeniu maszynowym.

Dane: Zbiór MNIST — 60 000 obrazków 28x28 px do treningu, 10 000 do testu.
Wczytywanie gotowe w `torchvision.datasets.MNIST`.

Wynik: Cyfra



GHOST

Group of Horribly Optimistic Statisticians



Rozpoznawanie cyfr (MNIST)

Co trzeba zrobić:

- Załaduj dane za pomocą `torchvision.datasets.MNIST`
- Przekształć obrazki do tensorów (flatten lub użyj CNN)
- Zbuduj model sieci neuronowej (np. Linear \rightarrow ReLU \rightarrow Linear \rightarrow Softmax)
- Użyj funkcji straty `CrossEntropyLoss`
- Oblicz dokładność (accuracy) na zbiorze testowym





GHOST

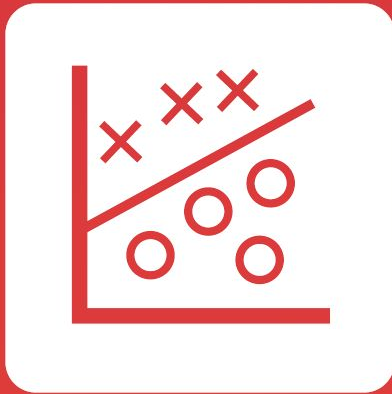
Group of Horribly Optimistic Statisticians

Dziękuję za uwagę

People with no idea
about AI, telling me my
AI will destroy the world

Me wondering why my
neural network is
classifying a cat as a dog..





GHOST

Group of Horribly Optimistic Statisticians