Algorytmy numeryczne znajdujące minimum lokalne zadanej funkcji celu

dr hab. inż. Piotr Gawron

Instytut Informatyki Teoretycznej i Stosowanej Polskiej Akademii Nauk

11 stycznia 2025

AKADEMIA SZTUKI KWANTOWEJ

Ministerstwo Nauki i Szkolnictwa Wyższego

NAUKA DLA SPOŁECZEŃSTWA

IITiS

Motywacja

- Bardzo wiele zadań w technice da się wyrazić w postaci problemu optymalizacyjnego.
- W uczeniu maszynowym często mówimy o funkcji straty (ang. Loss function)

$$f(\theta, \mathbf{X} \times \mathbf{Y}),$$

  zależnej od pewnego **zbioru parametrów** θ i od pewnego **zbioru danych** $\mathbf{X} \times \mathbf{Y}$.
- Takimi funkcjami opisane są np. sieci neuronowe.
- Uczenie sieci neuronowej polega na znalezieniu takich wartości parametrów θ, które dla zbioru uczącego $\mathbf{X} \times \mathbf{Y}$ „minimalizują”[1] funkcję straty.

[1] W sensie lokalnego minimum.

Motywacja

- Weźmy np. problem klasyfikacji binarnej
  - Dane uczące składają się z wektorów cech i etykiet[2]:

$$\mathbf{X} \times \mathbf{Y} = \{(X_1, y_1), (X_2, y_2), \ldots, (X_N, y_N)\}$$

  - Funkcja klasyfikująca przyporządkowuje wektorowi cech etykietę:

$$f_{\text{cls}} : \mathbf{X} \ni X \mapsto y \in \mathbf{Y}$$

- W procesie uczenia wykorzystamy funkcję straty:

$$l(y, \hat{y}) = (y - \hat{y})^2,$$

gdzie $\hat{y} = f_{\text{cls}}(X)$.

[2] Dla uproszczenia nie dzielimy zbioru danych na zbiory uczący i testujący.

Metoda spadku wzdłuż gradientu

Funkcja straty dla zbioru uczącego

- Funkcja straty $l(\theta, y, \hat{y}) = (y - \hat{y})^2$, jest zależna od parametrów θ, gdyż $\hat{y} = f_{\text{cls}}(\theta, X)$.
- Możemy uogólnić funkcję straty tak, by brała wartości z całego zbioru $\mathcal{X} \times \mathcal{Y}$ na raz:

$$L(\theta, \mathbf{X}, \mathbf{y}) = \sum_{i=1}^{N} \left(y_i - f_{\text{cls}}(\theta, X_i)\right)^2 .$$

- Ponieważ zbiór $\mathcal{X} \times \mathcal{Y}$ jest ustalony, to funkcję L można zapisać jako $L(\theta)$.

Optymalizacja funkcji straty

- Proces uczenia modelu (np. sieci neuronowej) polega na procesie minimalizacji funkcji straty $L(\theta)$.
- Minimalizację funkcji L można dokonać iteracyjnie metodą spadku wzdłuż gradientu $\nabla_\theta L(\theta)$.

Algorytm minimalizacji zgodnie z metodą spadku wzdłuż gradientu

Wybierz θ_{init} ▷ np. (pseudo-)losowo z ustalonego rozkładu
$\theta_0 \leftarrow \theta_{\text{init}}$
for $k \in \{1, 2, \ldots, K\}$ **do**
    $g_k \leftarrow \boldsymbol{\nabla}_{\theta_{k-1}} L(\theta_{k-1})$ ▷ Policz kierunek największego spadku
    $\theta_k \leftarrow \theta_{k-1} - \gamma_k g_k$ ▷ Zrób krok w dół o długości γ_k
end for

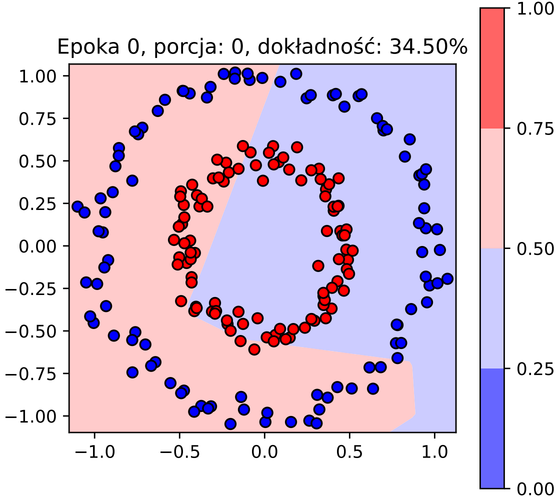
Przypomnienie

$$L(\theta) = L(\theta, \mathbf{X}, \mathbf{y}) = \sum_{i=1}^{N} (y_i - f_{\text{cls}}(\theta, X_i))^2$$

Przykład 1

Uczenie klasyfikatora

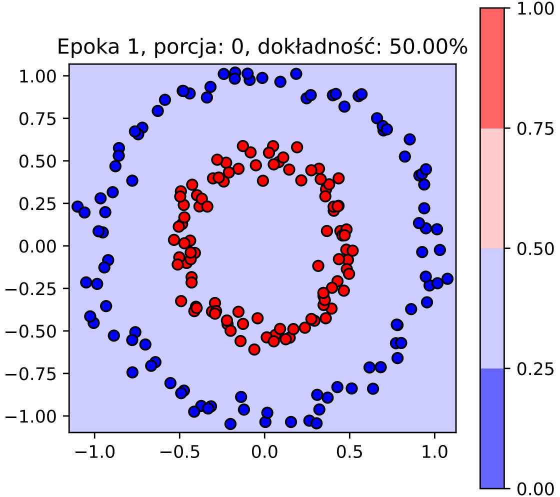
- Klasyfikacja binarna z wykorzystaniem perceptronu.
- Problem klasyfikacji zbioru nieseparowalnego liniowo.
- Sieć neuronowa dwuwarstwowa.
- Parametry θ to wagi sieci neuronowej.
- Wykorzystujemy do optymalizacji wszystkie dostępne punkty jednocześnie.



Przykład 1

Uczenie klasyfikatora

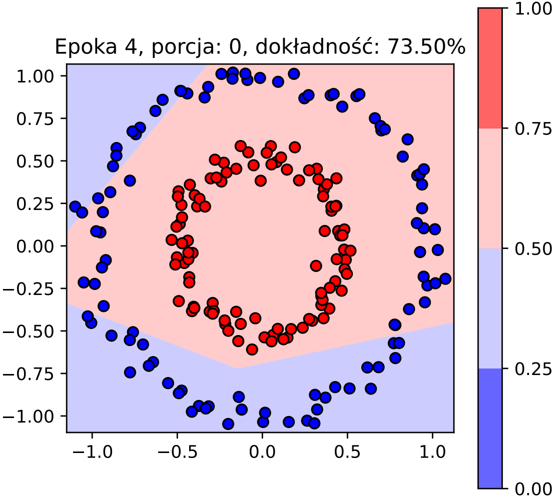
- Klasyfikacja binarna z wykorzystaniem perceptronu.
- Problem klasyfikacji zbioru nieseparowalnego liniowo.
- Sieć neuronowa dwuwarstwowa.
- Parametry θ to wagi sieci neuronowej.
- Wykorzystujemy do optymalizacji wszystkie dostępne punkty jednocześnie.



Przykład 1

Uczenie klasyfikatora

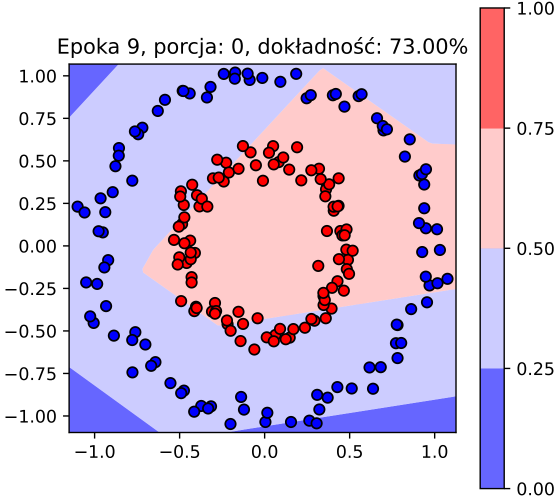
- Klasyfikacja binarna z wykorzystaniem perceptronu.
- Problem klasyfikacji zbioru nieseparowalnego liniowo.
- Sieć neuronowa dwuwarstwowa.
- Parametry θ to wagi sieci neuronowej.
- Wykorzystujemy do optymalizacji wszystkie dostępne punkty jednocześnie.



Przykład 1

Uczenie klasyfikatora

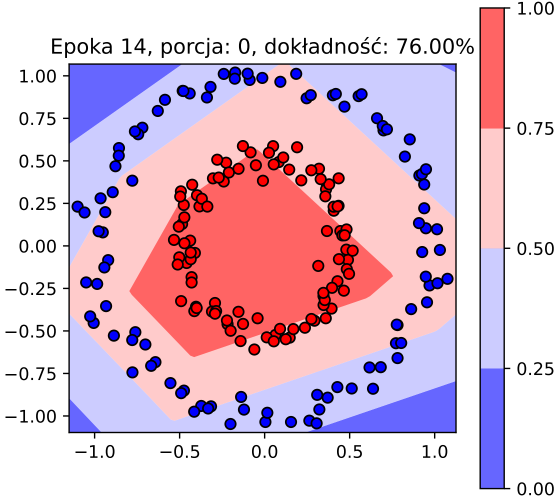
- Klasyfikacja binarna z wykorzystaniem perceptronu.
- Problem klasyfikacji zbioru nieseparowalnego liniowo.
- Sieć neuronowa dwuwarstwowa.
- Parametry θ to wagi sieci neuronowej.
- Wykorzystujemy do optymalizacji wszystkie dostępne punkty jednocześnie.



Przykład 1

Uczenie klasyfikatora

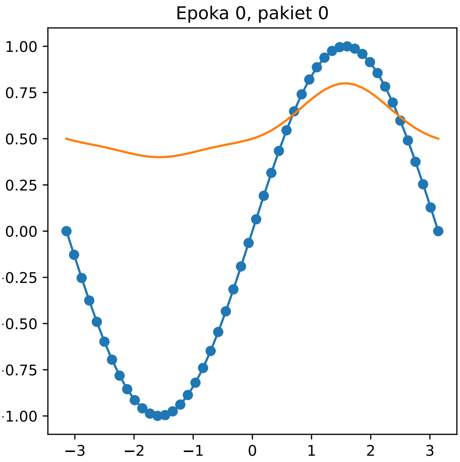
- Klasyfikacja binarna z wykorzystaniem perceptronu.
- Problem klasyfikacji zbioru nieseparowalnego liniowo.
- Sieć neuronowa dwuwarstwowa.
- Parametry θ to wagi sieci neuronowej.
- Wykorzystujemy do optymalizacji wszystkie dostępne punkty jednocześnie.



Przykład 2

Przybliżenie funkcji $\sin$ wielomianem trzeciego stopnia

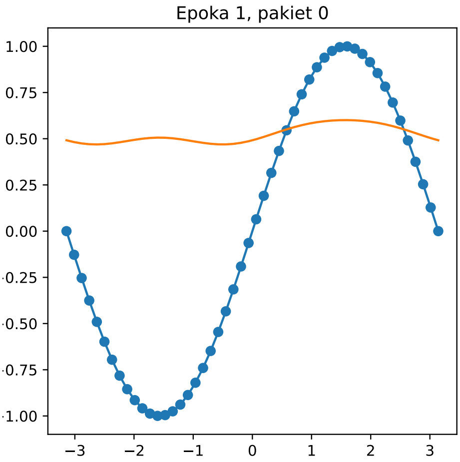
- Funkcja straty: $L(\theta) = \sum_i (\sin(X_i) - w(X_i))^2$.
- Wielomian: $w(x) = \theta_1 + \theta_2 x + \theta_3 x^2 + \theta_4 x^3$.
- Wykorzystujemy do optymalizacji wszystkie dostępne punkty jednocześnie.



Przykład 2

Przybliżenie funkcji $\sin$ wielomianem trzeciego stopnia

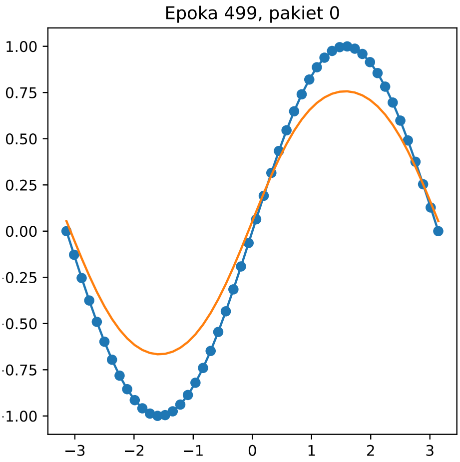
- Funkcja straty: $L(\theta) = \sum_i (\sin(X_i) - w(X_i))^2$.
- Wielomian: $w(x) = \theta_1 + \theta_2 x + \theta_3 x^2 + \theta_4 x^3$.
- Wykorzystujemy do optymalizacji wszystkie dostępne punkty jednocześnie.



Przykład 2

Przybliżenie funkcji $\sin$ wielomianem trzeciego stopnia

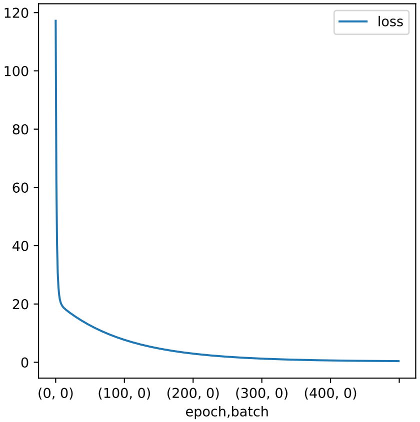
- Funkcja straty: $L(\theta) = \sum_i (\sin(X_i) - w(X_i))^2$.
- Wielomian: $w(x) = \theta_1 + \theta_2 x + \theta_3 x^2 + \theta_4 x^3$.
- Wykorzystujemy do optymalizacji wszystkie dostępne punkty jednocześnie.



Przykład 2

Przybliżenie funkcji $\sin$ wielomianem trzeciego stopnia

- Funkcja straty: $L(\theta) = \sum_i (\sin(X_i) - w(X_i))^2$.
- Wielomian: $w(x) = \theta_1 + \theta_2 x + \theta_3 x^2 + \theta_4 x^3$.
- Wykorzystujemy do optymalizacji wszystkie dostępne punkty jednocześnie.



Metoda stochastycznego spadku wzdłuż gradientu

Iteracja, porcja, epoka

Uczenie modeli jest zazwyczaj realizowane iteracyjnie.

- **Iteracją** nazywamy pojedynczy krok algorytmu optymalizacyjnego.
- **Porcją** (ang. batch) nazywamy podzbiór danych przetwarzany w jednej iteracji.
- **Epoka** to zbiór iteracji, podczas wykonywania którego wykorzystano wszystkie dane uczące dokładnie raz.

Metoda stochastycznego spadku wzdłuż gradientu

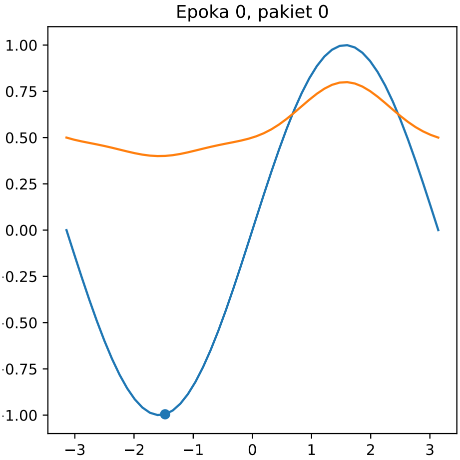
- Tym razem zbiór danych tasujemy tzn. ustawiamy w przypadkowej kolejności,
- i będziemy liczyć gradient $l(\theta, y, \hat{y}) = (y - \hat{y})^2$ z $\hat{y} = f_{\text{cls}}(\theta, X)$ dla każdego punktu osobno.

$(X_i, y_i) \leftarrow (X_{\sigma(i)}, y_{\sigma(i)})$ dla każdego i ▷ Tasowanie permutacją σ
Wybierz θ_{init} ▷ np. (pseudo-)losowo z ustalonego rozkładu
$\theta_0 \leftarrow \theta_{\text{init}}$
for $k \in \{1, 2, \ldots, K\}$ **do** ▷ Dla każdej epoki
    for $i \in \{1, 2, \ldots, N\}$ **do** ▷ Dla każdego punktu
        $g_{(k,i)} \leftarrow \boldsymbol{\nabla}_{\theta_{(k,i)-1}} l(\theta_{(k,i)-1})$ ▷ Policz kierunek największego spadku
        $\theta_{(k,i)} \leftarrow \theta_{(k,i)-1} - \gamma_k g_{(k,i)}$ ▷ Zrób krok w dół o długości γ_k
    end for
end for

Przykład 2
Optymalizacja stochastyczna

Przybliżenie funkcji sin wielomianem trzeciego stopnia

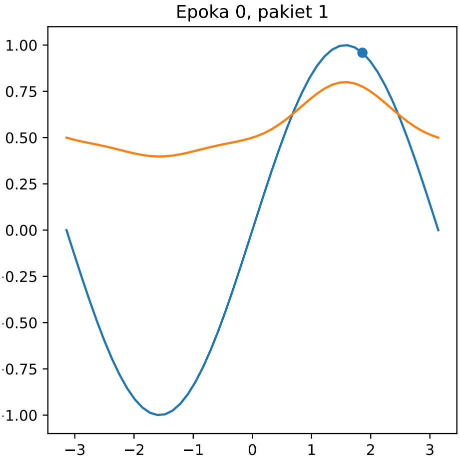
- Funkcja straty:
  $l(\theta) = (\sin(X_i) - w(X_i))^2.$
- Wielomian:
  $w(x) = \theta_1 + \theta_2 x + \theta_3 x^2 + \theta_4 x^3.$
- Wykorzystujemy do optymalizacji tylko jeden punkt.



Przykład 2
Optymalizacja stochastyczna

Przybliżenie funkcji $\sin$ wielomianem trzeciego stopnia

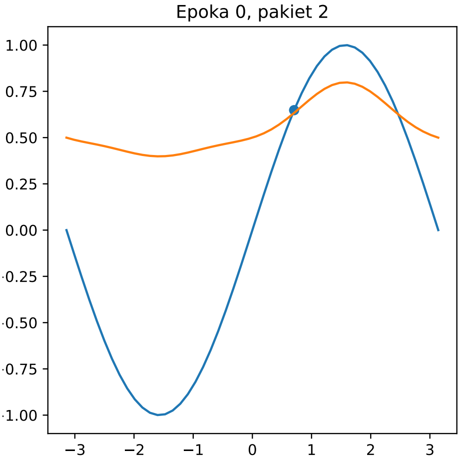
- Funkcja straty: $l(\theta) = (\sin(X_i) - w(X_i))^2$.
- Wielomian: $w(x) = \theta_1 + \theta_2 x + \theta_3 x^2 + \theta_4 x^3$.
- Wykorzystujemy do optymalizacji tylko jeden punkt.



Przykład 2
Optymalizacja stochastyczna

Przybliżenie funkcji $\sin$ wielomianem trzeciego stopnia

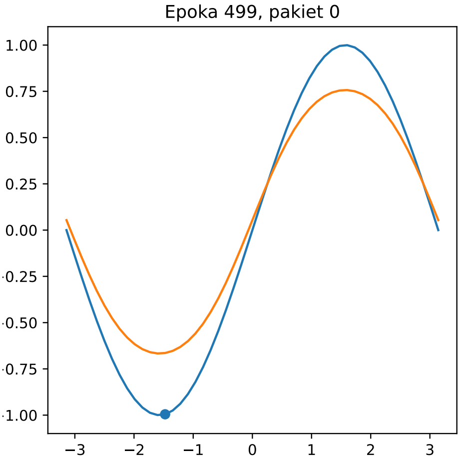
- Funkcja straty: $l(\theta) = (\sin(X_i) - w(X_i))^2$.
- Wielomian: $w(x) = \theta_1 + \theta_2 x + \theta_3 x^2 + \theta_4 x^3$.
- Wykorzystujemy do optymalizacji tylko jeden punkt.



Przykład 2
Optymalizacja stochastyczna

Przybliżenie funkcji $\sin$ wielomianem trzeciego stopnia

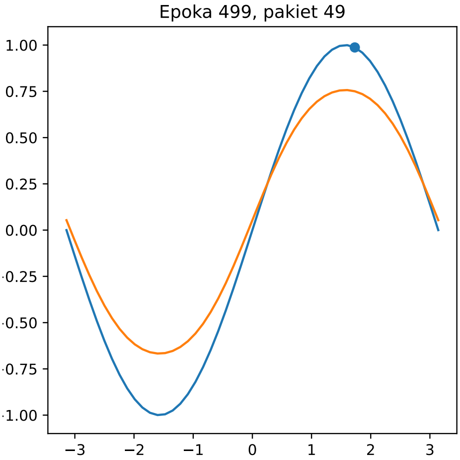
- Funkcja straty: $l(\theta) = (\sin(X_i) - w(X_i))^2$.
- Wielomian: $w(x) = \theta_1 + \theta_2 x + \theta_3 x^2 + \theta_4 x^3$.
- Wykorzystujemy do optymalizacji tylko jeden punkt.



Przykład 2
Optymalizacja stochastyczna

Przybliżenie funkcji sin wielomianem trzeciego stopnia

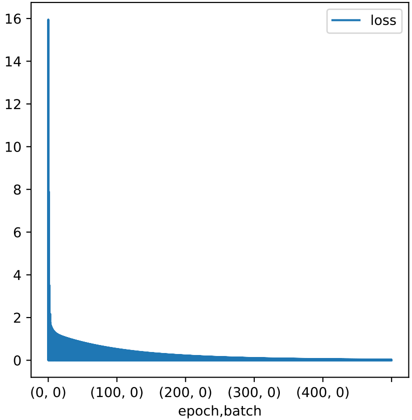
- Funkcja straty: $l(\theta) = (\sin(X_i) - w(X_i))^2$.
- Wielomian: $w(x) = \theta_1 + \theta_2 x + \theta_3 x^2 + \theta_4 x^3$.
- Wykorzystujemy do optymalizacji tylko jeden punkt.



Przykład 2
Optymalizacja stochastyczna

Przybliżenie funkcji $\sin$ wielomianem trzeciego stopnia

- Funkcja straty: $l(\theta) = (\sin(X_i) - w(X_i))^2$.
- Wielomian: $w(x) = \theta_1 + \theta_2 x + \theta_3 x^2 + \theta_4 x^3$.
- Wykorzystujemy do optymalizacji tylko jeden punkt.



Praktyka — uczenie porcjami

Funkcja straty dla porcji próbek

- Tym razem tasujemy i dzielimy zbiór danych na porcje (ang. batches):
- $(X, y)_b = \{(X_1, y_1)_b, \ldots, (X_{B_b}, y_{B_b})_b\}$ oraz $\bigcup_{b=1}^{B} (X, y)_b = \mathcal{X} \times \mathcal{Y}$.
- Zatem naszą funkcją straty będzie:

$$L(\theta) = L(\theta, \mathbf{X}_b, \mathbf{y}_b) = \sum_i \left(y_{b,i} - f_{\text{cls}}(\theta, X_{b,i})\right)^2 ,$$

ale będzie ona liczona dla każdej porcji.

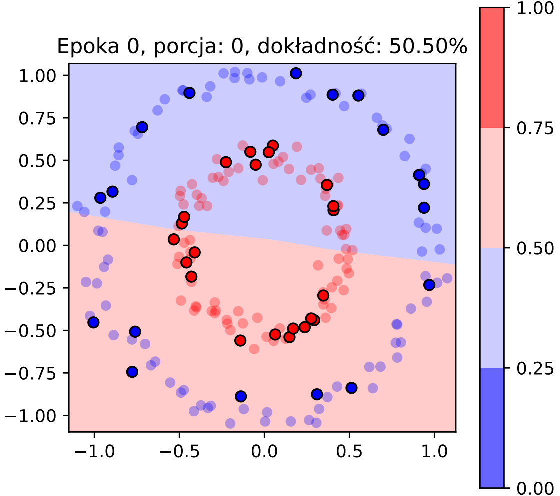
Algorytm minimalizacji zgodnie z metodą spadku wzdłuż gradientu wykorzystujący porcje

Wybierz θ_{init} ▷ np. (pseudo-)losowo z ustalonego rozkładu
$\theta_0 \leftarrow \theta_{\text{init}}$
for $k \in \{1, 2, \ldots, K\}$ **do** ▷ Dla każdej epoki
  for $b \in \{1, 2, \ldots, B\}$ **do** ▷ Dla każdej porcji
    $g_{(k,b)} = \boldsymbol{\nabla}_{\theta_{(k,b)-1}} L(\theta_{(k,b)-1})$ ▷ Kierunek największego spadku
    $\theta_{(k,b)} \leftarrow \theta_{(k,b)-1} - \gamma_k g_{(k,b)}$ ▷ Zrób krok w dół o długości γ_k
  end for
end for

Przykład 1
Optymalizacja porcjami

Uczenie klasyfikatora

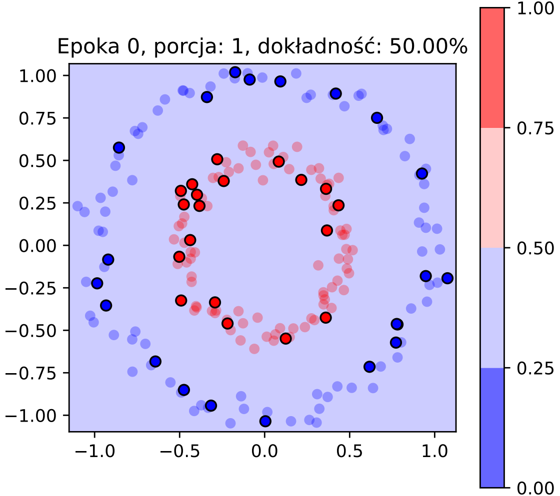
- Klasyfikacja binarna z wykorzystaniem perceptronu.
- Problem klasyfikacji zbioru nieseparowalnego liniowo.
- Sieć neuronowa dwuwarstwowa.
- Parametry θ to wagi sieci neuronowej.
- Wykorzystujemy do optymalizacji porcje 40 punktów z 200.



Przykład 1
Optymalizacja porcjami

Uczenie klasyfikatora

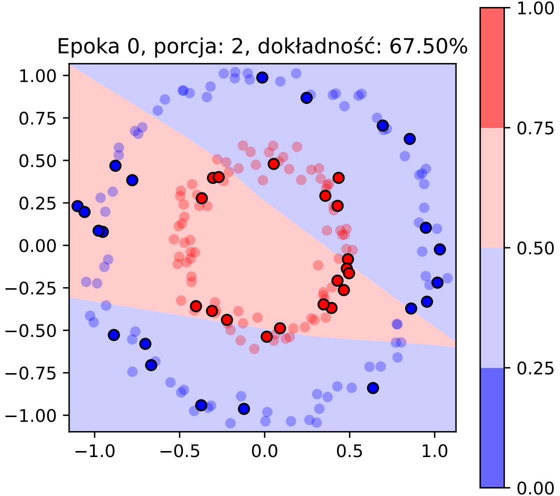
- Klasyfikacja binarna z wykorzystaniem perceptronu.
- Problem klasyfikacji zbioru nieseparowalnego liniowo.
- Sieć neuronowa dwuwarstwowa.
- Parametry θ to wagi sieci neuronowej.
- Wykorzystujemy do optymalizacji porcje 40 punktów z 200.



Przykład 1
Optymalizacja porcjami

Uczenie klasyfikatora

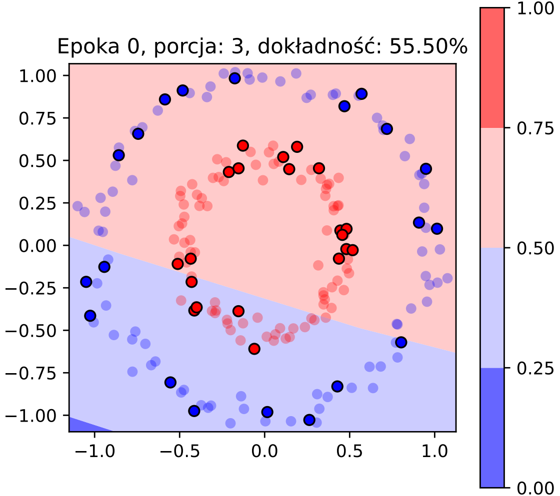
- Klasyfikacja binarna z wykorzystaniem perceptronu.
- Problem klasyfikacji zbioru nieseparowalnego liniowo.
- Sieć neuronowa dwuwarstwowa.
- Parametry θ to wagi sieci neuronowej.
- Wykorzystujemy do optymalizacji porcje 40 punktów z 200.



Przykład 1
Optymalizacja porcjami

Uczenie klasyfikatora

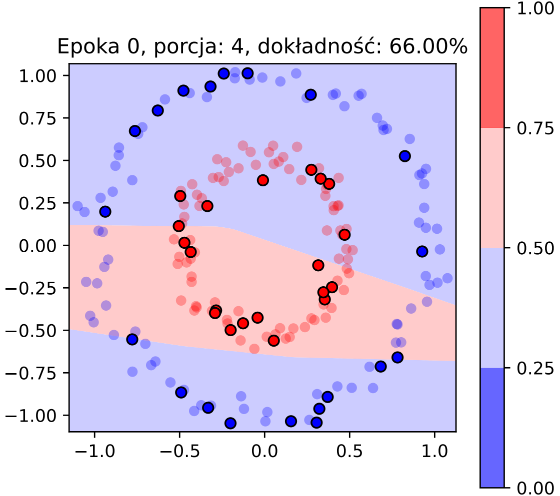
- Klasyfikacja binarna z wykorzystaniem perceptronu.
- Problem klasyfikacji zbioru nieseparowalnego liniowo.
- Sieć neuronowa dwuwarstwowa.
- Parametry θ to wagi sieci neuronowej.
- Wykorzystujemy do optymalizacji porcje 40 punktów z 200.



Przykład 1
Optymalizacja porcjami

Uczenie klasyfikatora

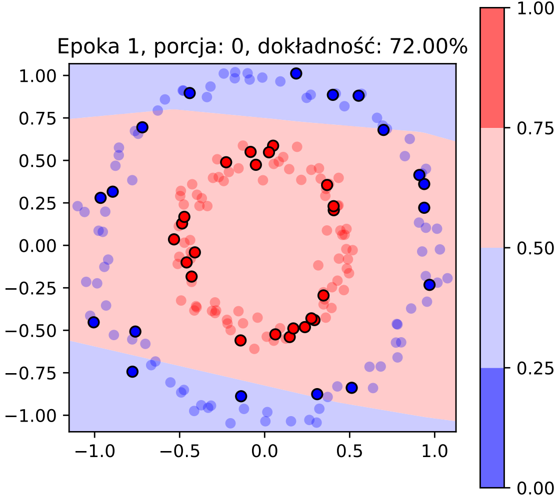
- Klasyfikacja binarna z wykorzystaniem perceptronu.
- Problem klasyfikacji zbioru nieseparowalnego liniowo.
- Sieć neuronowa dwuwarstwowa.
- Parametry θ to wagi sieci neuronowej.
- Wykorzystujemy do optymalizacji porcje 40 punktów z 200.



Przykład 1
Optymalizacja porcjami

Uczenie klasyfikatora

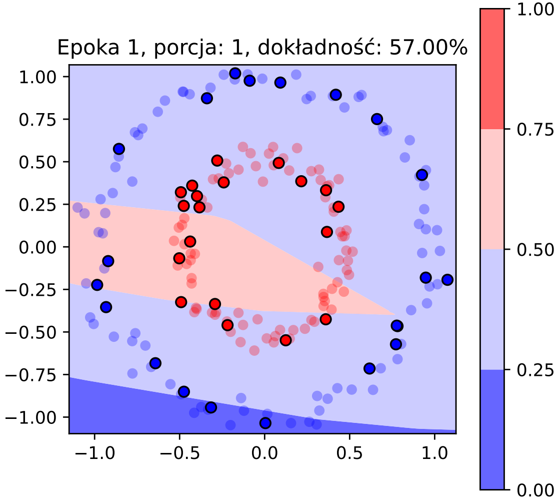
- Klasyfikacja binarna z wykorzystaniem perceptronu.
- Problem klasyfikacji zbioru nieseparowalnego liniowo.
- Sieć neuronowa dwuwarstwowa.
- Parametry θ to wagi sieci neuronowej.
- Wykorzystujemy do optymalizacji porcje 40 punktów z 200.



Przykład 1
Optymalizacja porcjami

Uczenie klasyfikatora

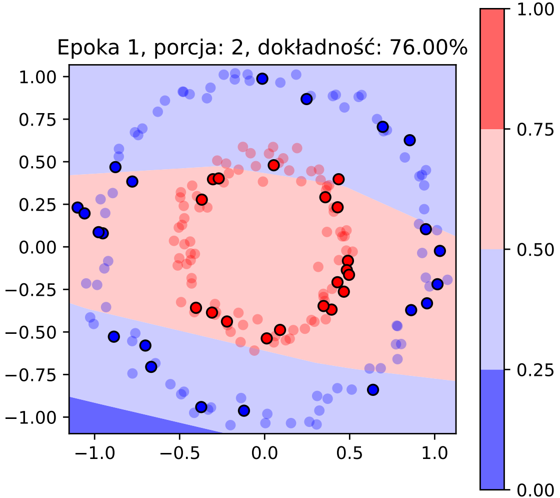
- Klasyfikacja binarna z wykorzystaniem perceptronu.
- Problem klasyfikacji zbioru nieseparowalnego liniowo.
- Sieć neuronowa dwuwarstwowa.
- Parametry θ to wagi sieci neuronowej.
- Wykorzystujemy do optymalizacji porcje 40 punktów z 200.



Przykład 1
Optymalizacja porcjami

Uczenie klasyfikatora

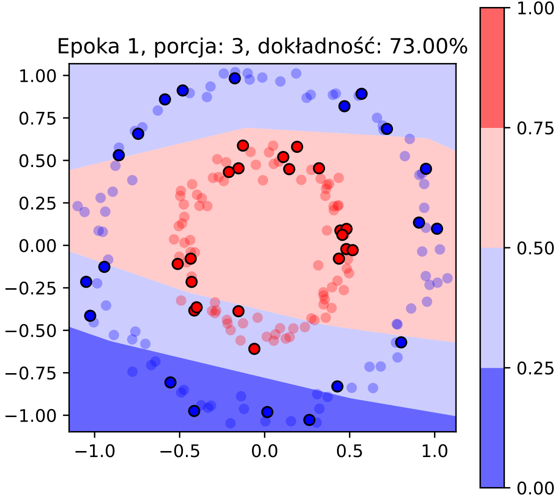
- Klasyfikacja binarna z wykorzystaniem perceptronu.
- Problem klasyfikacji zbioru nieseparowalnego liniowo.
- Sieć neuronowa dwuwarstwowa.
- Parametry θ to wagi sieci neuronowej.
- Wykorzystujemy do optymalizacji porcje 40 punktów z 200.



Przykład 1
Optymalizacja porcjami

Uczenie klasyfikatora

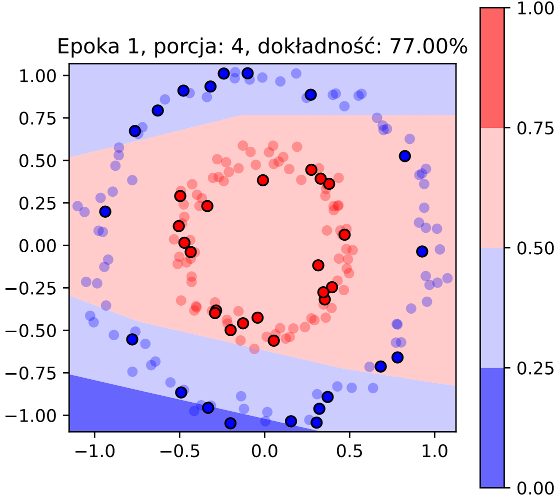
- Klasyfikacja binarna z wykorzystaniem perceptronu.
- Problem klasyfikacji zbioru nieseparowalnego liniowo.
- Sieć neuronowa dwuwarstwowa.
- Parametry θ to wagi sieci neuronowej.
- Wykorzystujemy do optymalizacji porcje 40 punktów z 200.



Przykład 1
Optymalizacja porcjami

Uczenie klasyfikatora

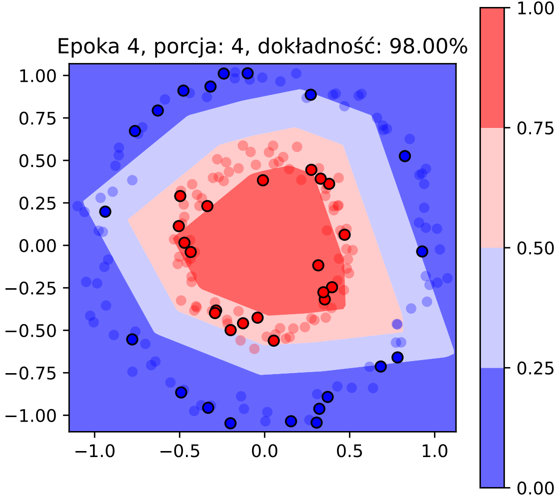
- Klasyfikacja binarna z wykorzystaniem perceptronu.
- Problem klasyfikacji zbioru nieseparowalnego liniowo.
- Sieć neuronowa dwuwarstwowa.
- Parametry θ to wagi sieci neuronowej.
- Wykorzystujemy do optymalizacji porcje 40 punktów z 200.



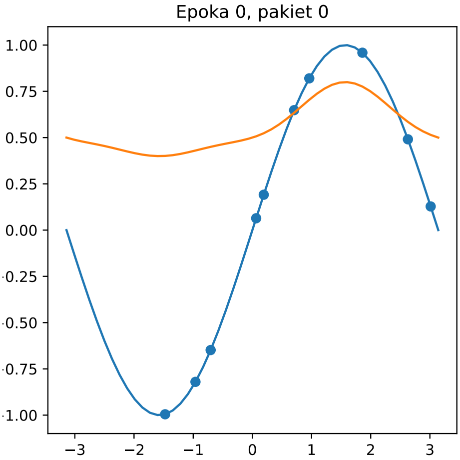
Przykład 1
Optymalizacja porcjami

Uczenie klasyfikatora

- Klasyfikacja binarna z wykorzystaniem perceptronu.
- Problem klasyfikacji zbioru nieseparowalnego liniowo.
- Sieć neuronowa dwuwarstwowa.
- Parametry θ to wagi sieci neuronowej.
- Wykorzystujemy do optymalizacji porcje 40 punktów z 200.



Przykład 2
Optymalizacja porcjami



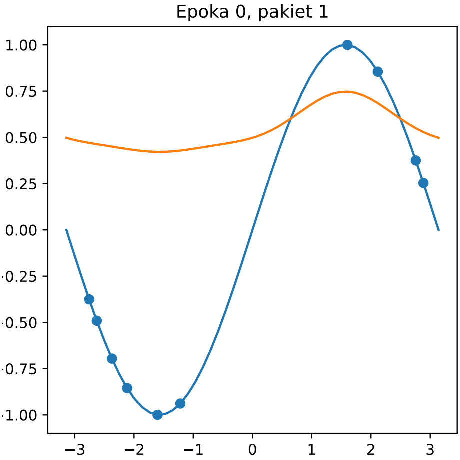
Przybliżenie funkcji $\sin$ wielomianem trzeciego stopnia

- Funkcja straty: $l(\theta) = (\sin(X_i) - w(X_i))^2$.
- Wielomian: $w(x) = \theta_1 + \theta_2 x + \theta_3 x^2 + \theta_4 x^3$.
- Wykorzystujemy do optymalizacji porcję 10 punktów z 50.

Przykład 2
Optymalizacja porcjami

Przybliżenie funkcji $\sin$ wielomianem trzeciego stopnia

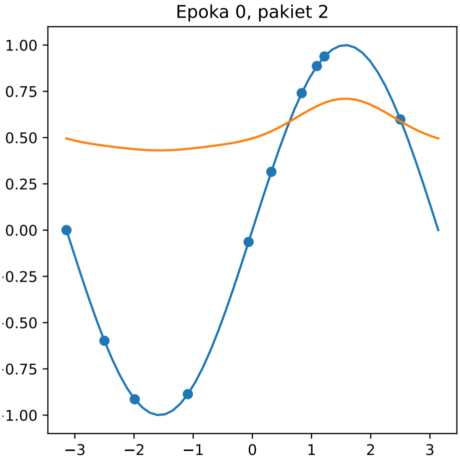
- Funkcja straty:
  $l(\theta) = (\sin(X_i) - w(X_i))^2.$
- Wielomian:
  $w(x) = \theta_1 + \theta_2 x + \theta_3 x^2 + \theta_4 x^3.$
- Wykorzystujemy do optymalizacji porcję 10 punktów z 50.



Przykład 2
Optymalizacja porcjami

Przybliżenie funkcji $\sin$ wielomianem trzeciego stopnia

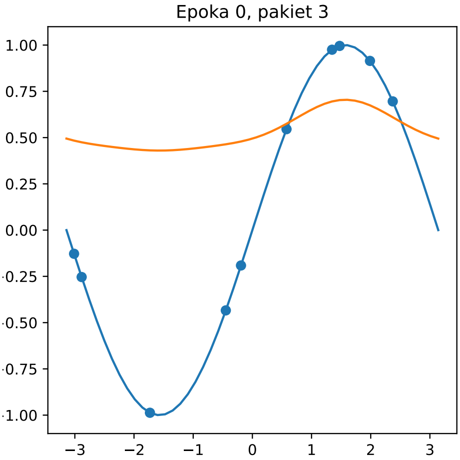
- Funkcja straty: $l(\theta) = (\sin(X_i) - w(X_i))^2$.
- Wielomian: $w(x) = \theta_1 + \theta_2 x + \theta_3 x^2 + \theta_4 x^3$.
- Wykorzystujemy do optymalizacji porcję 10 punktów z 50.



Przykład 2
Optymalizacja porcjami

Przybliżenie funkcji $\sin$ wielomianem trzeciego stopnia

- Funkcja straty: $l(\theta) = (\sin(X_i) - w(X_i))^2$.
- Wielomian: $w(x) = \theta_1 + \theta_2 x + \theta_3 x^2 + \theta_4 x^3$.
- Wykorzystujemy do optymalizacji porcję 10 punktów z 50.

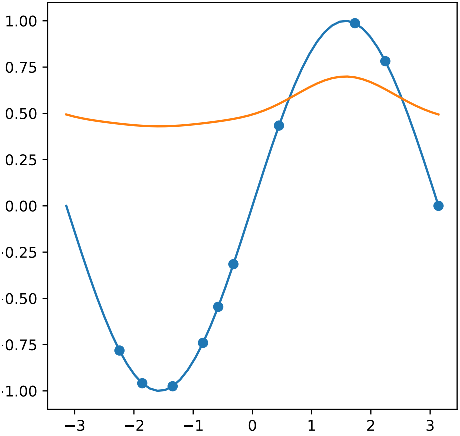


Przykład 2
Optymalizacja porcjami

Przybliżenie funkcji $\sin$ wielomianem trzeciego stopnia

- Funkcja straty: $l(\theta) = (\sin(X_i) - w(X_i))^2$.
- Wielomian: $w(x) = \theta_1 + \theta_2 x + \theta_3 x^2 + \theta_4 x^3$.
- Wykorzystujemy do optymalizacji porcję 10 punktów z 50.

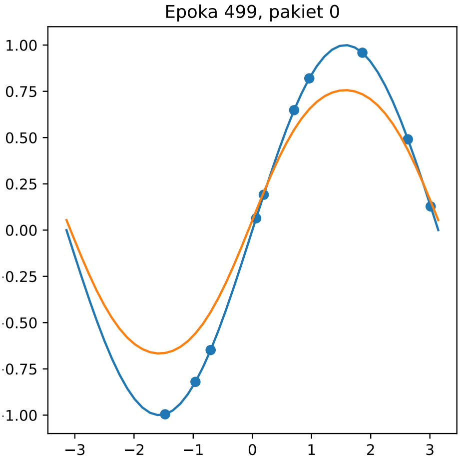
Epoka 0, pakiet 4



Przykład 2
Optymalizacja porcjami

Przybliżenie funkcji $\sin$ wielomianem trzeciego stopnia

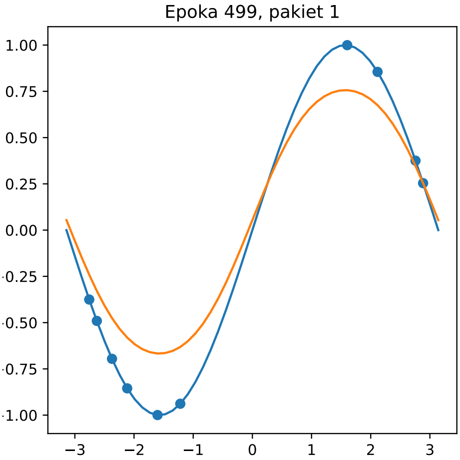
- Funkcja straty: $l(\theta) = (\sin(X_i) - w(X_i))^2$.
- Wielomian: $w(x) = \theta_1 + \theta_2 x + \theta_3 x^2 + \theta_4 x^3$.
- Wykorzystujemy do optymalizacji porcję 10 punktów z 50.



Przykład 2
Optymalizacja porcjami

Przybliżenie funkcji $\sin$ wielomianem trzeciego stopnia

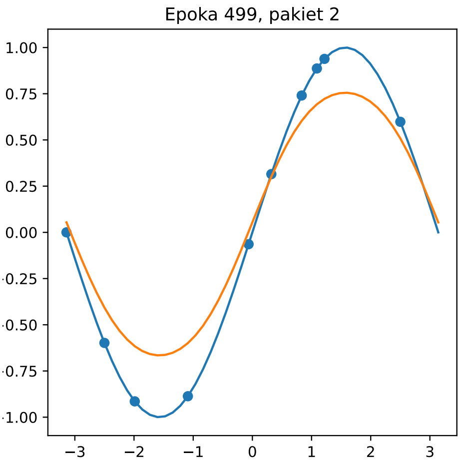
- Funkcja straty: $l(\theta) = (\sin(X_i) - w(X_i))^2.$
- Wielomian: $w(x) = \theta_1 + \theta_2 x + \theta_3 x^2 + \theta_4 x^3.$
- Wykorzystujemy do optymalizacji porcję 10 punktów z 50.



Przykład 2
Optymalizacja porcjami

Przybliżenie funkcji $\sin$ wielomianem trzeciego stopnia

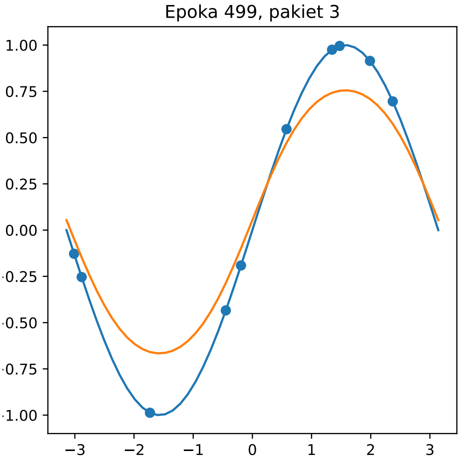
- Funkcja straty: $l(\theta) = (\sin(X_i) - w(X_i))^2.$
- Wielomian: $w(x) = \theta_1 + \theta_2 x + \theta_3 x^2 + \theta_4 x^3.$
- Wykorzystujemy do optymalizacji porcję 10 punktów z 50.



Przykład 2
Optymalizacja porcjami

Przybliżenie funkcji $\sin$ wielomianem trzeciego stopnia

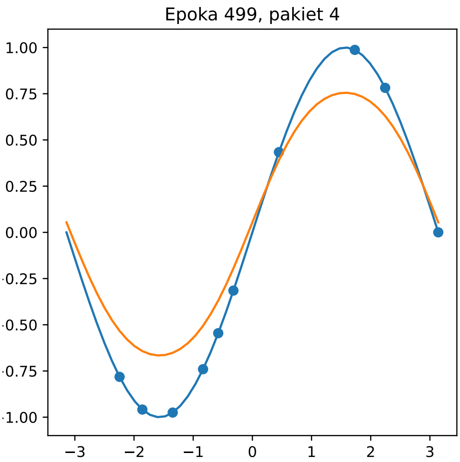
- Funkcja straty: $l(\theta) = (\sin(X_i) - w(X_i))^2$.
- Wielomian: $w(x) = \theta_1 + \theta_2 x + \theta_3 x^2 + \theta_4 x^3$.
- Wykorzystujemy do optymalizacji porcję 10 punktów z 50.



Przykład 2
Optymalizacja porcjami

Przybliżenie funkcji $\sin$ wielomianem trzeciego stopnia

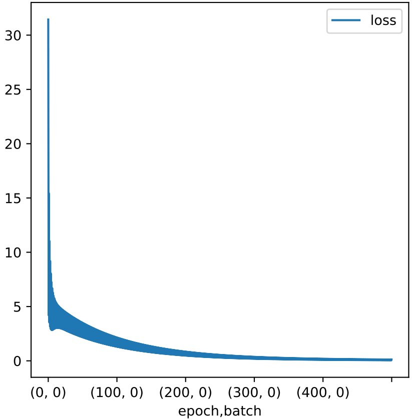
- Funkcja straty: $l(\theta) = (\sin(X_i) - w(X_i))^2$.
- Wielomian: $w(x) = \theta_1 + \theta_2 x + \theta_3 x^2 + \theta_4 x^3$.
- Wykorzystujemy do optymalizacji porcję 10 punktów z 50.



Przykład 2
Optymalizacja porcjami

Przybliżenie funkcji $\sin$ wielomianem trzeciego stopnia

- Funkcja straty: $l(\theta) = (\sin(X_i) - w(X_i))^2$.
- Wielomian: $w(x) = \theta_1 + \theta_2 x + \theta_3 x^2 + \theta_4 x^3$.
- Wykorzystujemy do optymalizacji porcję 10 punktów z 50.



Podsumowanie

- Optymalizacja (uczenie) porcjami jest często wykorzystywane w praktyce.
- Pozwala na umieszczenie w pamięci GPGPU modelu i części danych jednocześnie.
- Wykorzystuje architekturę wektorową procesorów GPGPU.
- Pozwala również wyskoczyć z lokalnego minimum podczas optymalizacji.