Jak Sieci Neuronowe Naprawdę Się Uczą

Praktyczny przewodnik przez forward pass, backpropagation i gradient descent – z pełnymi obliczeniami krok po kroku

Jak To Rozumieć Intuicyjnie?

Uczenie sieci neuronowej to sześć prostych kroków, które powtarzają się w kółko. Każda iteracja sprawia, że sieć staje się trochę mądrzejsza.

1

Forward Pass

Neuron próbuje wyprodukować wynik na podstawie obecnych wag

2

Loss

Mierzymy, jak bardzo neuron się pomylił względem prawdziwej wartości

3

Backpropagation

Mówimy neuronowi, gdzie dokładnie zrobił błąd

4

Gradient

Wskazujemy, która waga jest odpowiedzialna za błąd

5

Update

Poprawiamy każdą wagę o małą wartość w dobrym kierunku

6

Iteracja

Neuron próbuje znowu, tym razem z lepszymi wagami

To jest cały proces uczenia sieci neuronowej – prosty, ale potężny.

Nasz Przykład: Prosty Neuron

Dane wejściowe

Neuron dostaje dwa sygnały:

$$x_1 = 1, \quad x_2 = 2$$

Parametry neuronu

$$w_1 = 1, \quad w_2 = 1$$

$$b = 1$$

Cel uczenia

Chcemy, aby neuron nauczył się produkować wyjście:

$$y_{true} = 5$$

Tempo uczenia

$$\eta = 0.1$$

Kontroluje, jak szybko neuron się uczy

Krok 1: Forward Pass – Neuron Liczy

Najpierw neuron oblicza sumę ważoną wszystkich wejść. To podstawowa operacja każdego neuronu.

01

Suma ważona

Mnożymy każde wejście przez odpowiednią wagę i dodajemy bias:

$$z = w_1 x_1 + w_2 x_2 + b$$

02

Podstawiamy wartości

$$z = 1 \cdot 1 + 1 \cdot 2 + 1 = 4$$

03

Funkcja aktywacji ReLU

ReLU albo przepuszcza sygnał, albo go ucina:

$$ReLU(z) = \begin{cases} z & \text{jeśli } z > 0 \\ 0 & \text{jeśli } z \leq 0 \end{cases}$$

04

Wynik neuronu

Ponieważ $z = 4 > 0$, neuron jest aktywny:

$$y = ReLU(4) = 4$$

Krok 2: Loss – Jak Bardzo Się Pomyliliśmy?

Funkcja straty MSE

Mean Squared Error mierzy kwadrat różnicy między przewidywaniem a prawdą:

$$L = \frac{1}{2}(y - y_{true})^2$$

Obliczamy błąd

$$L = \frac{1}{2}(4 - 5)^2 = \frac{1}{2} \cdot 1 = 0.5$$

Neuron pomylił się o 1, więc strata wynosi 0.5. Im większa strata, tym gorsze przewidywanie.

Krok 3: Backpropagation – Skąd Błąd?

Teraz rozpoczynamy backpropagation – idziemy wstecz przez sieć, aby zrozumieć źródło błędu.

Krok 3a: Błąd wyjścia

Jak bardzo wyjście różni się od celu?

$$\delta_y = y - y_{true}$$

$$\delta_y = 4 - 5 = -1$$

Wynik jest o 1 za mały

Krok 3b: Pochodna ReLU

Czy neuron był aktywny?

$$ReLU'(z) = \begin{cases} 1 & \text{jeśli } z > 0 \\ 0 & \text{jeśli } z \leq 0 \end{cases}$$

$$\delta_z = \delta_y \cdot ReLU'(z) = -1 \cdot 1 = -1$$

Gradient przepływa dalej

Krok 4: Gradienty – Kto Jest Winny?

Każda waga pyta: "Jak bardzo JA jestem odpowiedzialna za błąd?" Gradienty dają odpowiedź.

1

Gradient wagi w_1

Im większe wejście, tym większy wpływ wagi na błąd:

$$\frac{\partial L}{\partial w_1} = \delta_z \cdot x_1 = -1 \cdot 1 = -1$$

2

Gradient wagi w_2

Wejście było 2 razy większe, więc wpływ też większy:

$$\frac{\partial L}{\partial w_2} = \delta_z \cdot x_2 = -1 \cdot 2 = -2$$

3

Gradient biasu

Bias wpływa równo na wszystkie obliczenia:

$$\frac{\partial L}{\partial b} = \delta_z = -1$$

Ujemne gradienty oznaczają, że wagi są za małe i powinny być zwiększone, aby zmniejszyć błąd.

Krok 5: Update – Neuron Się Uczy!

Zasada aktualizacji

Idziemy w dół gradientu, aby zmniejszyć stratę:

$$w \leftarrow w - \eta \cdot \frac{\partial L}{\partial w}$$

Jeśli gradient jest:

- **Dodatni** → waga maleje
- **Ujemny** → waga rośnie

Dlaczego? Bo chcemy iść w kierunku mniejszego błędu.

Nowe wartości wag

$$w_1^{new} = 1 - 0.1 \cdot (-1) = 1.1$$

$$w_2^{new} = 1 - 0.1 \cdot (-2) = 1.2$$

$$b^{new} = 1 - 0.1 \cdot (-1) = 1.1$$

Wszystkie parametry zostały zwiększone, bo gradienty były ujemne. Neuron skorygował swoje wagi w dobrym kierunku!

Krok 6: Druga Iteracja – Czy Sieć Jest Lepsza?

Sprawdzamy, czy nowe wagi dają lepsze wyniki. To moment prawdy!

1 — **Nowy forward pass**

$$z_{new} = 1.1 \cdot 1 + 1.2 \cdot 2 + 1.1 = 4.6$$

$$y_{new} = ReLU(4.6) = 4.6$$

2 — **Nowa strata**

$$L_{new} = \frac{1}{2}(4.6 - 5)^2 = 0.08$$

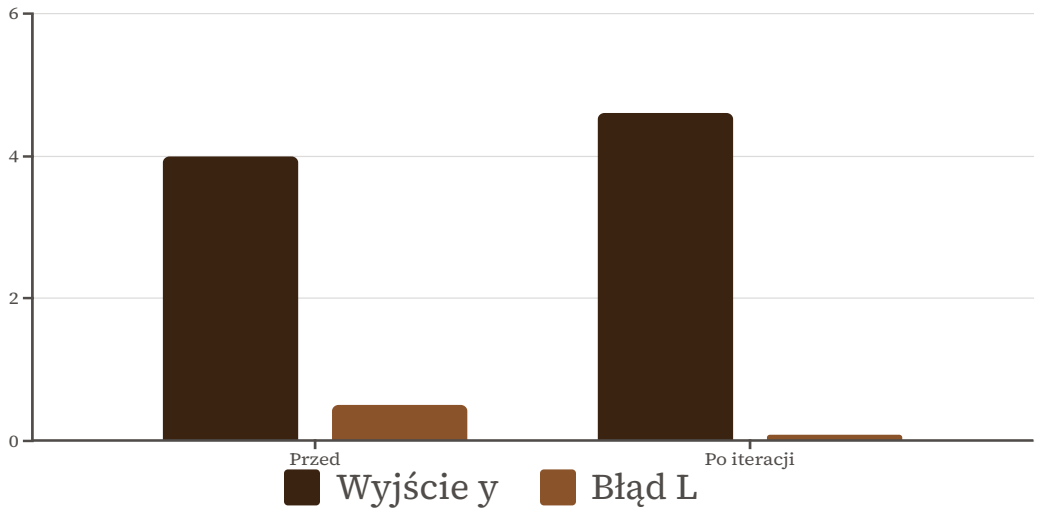
3 — **Porównanie**

Stara strata: **0.50**

Nowa strata: **0.08**

Redukcja: 84%!

Sukces! Sieć Się Nauczyła 🎉



Co się stało?

- **Wyjście przesunęło się** z 4.0 → 4.6 (cel: 5.0)
- **Strata drastycznie zmalała** z 0.50 → 0.08
- Neuron stał się mądrzejszy po **jednej iteracji**

To jest cała magia deep learningu: małe kroki, powtarzane tysiące razy, prowadzą do niesamowitych rezultatów.

Gdybyśmy kontynuowali ten proces przez kolejne iteracje, neuron w końcu nauczyłby się produkować wyjście bardzo bliskie 5.0. Tak właśnie działają sieci neuronowe!