Zajęcie 5. Funkcje aktywacji

Abstract

Celem jest nabycie podstawowej znajomości użycia funkcji aktywacji.

1. Ćwiczenie praktyczne: Funkcje aktywacji

1.1. Cel ćwiczenia

Celem ćwiczenia jest zrozumienie roli funkcji aktywacji w sieciach neuronowych oraz analiza ich własności matematycznych, w szczególności:

- nieliniowości,
- pochodnych,
- saturacji,
- stabilności numerycznej.

1.2. Wprowadzenie teoretyczne

W klasycznym neuronowym modelu perceptronu sygnał wejściowy $x \in \mathbb{R}^n$ jest przekształcany liniowo, a następnie przechodzi przez funkcję aktywacji:

$$z = w^\top x + b, \qquad a = \sigma(z),$$

gdzie:

$$\sigma : \mathbb{R} \to \mathbb{R}$$

jest funkcją aktywacji, która wprowadza nieliniowość konieczną do aproksymacji złożonych zależności.

1.3. Najważniejsze funkcje aktywacji

1.3.1. Funkcja Sigmoidalna (Logistic)

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

Pochodna:

$$\sigma'(z) = \sigma(z)(1 - \sigma(z)).$$

Własności:

- zakres: $(0, 1)$,
- saturacja dla $z \to \pm\infty$,
- skłonność do zanikającego gradientu.

1.3.2. Funkcja Tangens Hiperboliczny (tanh)

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

Pochodna:

$$\tanh'(z) = 1 - \tanh^2(z).$$

Zakres:

$$\tanh(z) \in (-1, 1).$$

1.3.3. ReLU (Rectified Linear Unit)

$$\mathrm{ReLU}(z) = \max(0, z)$$

Pochodna:

$$\mathrm{ReLU}'(z) = \begin{cases} 0, & z < 0, \\ 1, & z > 0. \end{cases}$$

Własności:

- brak saturacji dla $z > 0$,
- ryzyko "martwych neuronów" dla $z < 0$.

1.3.4. Leaky ReLU

$$\text{LeakyReLU}(z) = \begin{cases} \alpha z, & z < 0, \\ z, & z \geq 0, \end{cases} \qquad \alpha \in (0,1)$$

1.3.5. Softmax (dla warstwy wyjściowej klasyfikacji wieloklasowej)

Dla wektora logitów $z = (z_1, \ldots, z_K)$:

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}.$$

Wersja numerycznie stabilna:

$$\text{softmax}(z_i) = \frac{e^{z_i - \max(z)}}{\sum_{j=1}^{K} e^{z_j - \max(z)}}.$$

2. Najważniejsze funkcje aktywacji i ich własności

W sieciach neuronowych funkcje aktywacji odpowiadają za wprowadzanie nieliniowości, umożliwiając aproksymację złożonych funkcji. Poniżej przedstawiono najczęściej stosowane funkcje wraz z ich własnościami analitycznymi.

2.1. Funkcja sigmoidalna (logistyczna)

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

Pochodna:.

$$\sigma'(z) = \sigma(z)\,(1 - \sigma(z)).$$

Własności:.

- zakres: $(0, 1)$,
- saturacja dla $z \to \pm\infty$,
- pochodna praktycznie znika dla dużych wartości bezwzględnych $|z|$,
- funkcja monotoniczna,
- nieliniowa.

2.2. Tangens hiperboliczny

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

Pochodna:.

$$\tanh'(z) = 1 - \tanh^2(z).$$

Własności:.

- zakres: $(-1, 1)$,
- funkcja nieparzysta (symetryczna względem początku),
- pochodna największa w pobliżu $z = 0$,
- problem zanikającego gradientu pozostaje, lecz mniejszy niż dla sigmoidu.

2.3. ReLU (Rectified Linear Unit)

$$\mathrm{ReLU}(z) = \max(0, z)$$

Pochodna:.

$$\mathrm{ReLU}'(z) = \begin{cases} 0, & z < 0, \\ 1, & z > 0. \end{cases}$$

Własności:.

- brak saturacji dla $z > 0$,
- nieliniowa, ale częściowo liniowa,
- tania obliczeniowo,
- możliwość wystąpienia "martwych neuronów" przy $z < 0$.

2.4. Leaky ReLU

$$\mathrm{LeakyReLU}(z) = \begin{cases} \alpha z, & z < 0, \\ z, & z \geq 0, \end{cases} \quad \alpha \in (0, 1)$$

Pochodna:.

$$\text{LeakyReLU}'(z) = \begin{cases} \alpha, & z < 0, \\ 1, & z \geq 0. \end{cases}$$

Własności:.

- eliminuje problem martwych neuronów dzięki $\alpha > 0$,
- brak saturacji dla $z > 0$,
- fragmentarycznie liniowa.

2.5. Parametryczna ReLU (PReLU)

$$\text{PReLU}(z) = \begin{cases} az, & z < 0, \\ z, & z \geq 0, \end{cases} \quad a \text{ jest uczone podczas treningu.}$$

Pochodna:.

$$\text{PReLU}'(z) = \begin{cases} a, & z < 0, \\ 1, & z \geq 0. \end{cases}$$

Własności:.

- współczynnik nachylenia ujemnej części uczony jest wraz z wagami,
- zwiększona elastyczność modelu.

2.6. Funkcja ELU (Exponential Linear Unit)

$$\text{ELU}(z) = \begin{cases} z, & z \geq 0, \\ \alpha(e^z - 1), & z < 0, \end{cases}$$

Pochodna:.

$$\text{ELU}'(z) = \begin{cases} 1, & z \geq 0, \\ \alpha e^z, & z < 0. \end{cases}$$

Własności:.

- gładka (różniczkowalna w całej dziedzinie),
- dla $z < 0$ wartości są ujemne, co poprawia średni sygnał aktywacji,
- wolniejsza obliczeniowo niż ReLU.

2.7. Softmax (dla klasyfikacji wieloklasowej)

Dla wektora logitów $z = (z_1, \ldots, z_K)$:

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

Pochodna (macierz Jacobiego):.

$$\frac{\partial \text{softmax}(z_i)}{\partial z_j} = \text{softmax}(z_i) \left(\delta_{ij} - \text{softmax}(z_j)\right),$$

gdzie δ_{ij} to delta Kroneckera.

Własności:.

- przekształca wektor logitów w rozkład prawdopodobieństwa,
- silna nieliniowość,
- maksymalnie czuła na różnice między logitami,
- wymaga ostrożności numerycznej (stabilizowana przez odejmowanie $\max(z)$).

2.8. Funkcja liniowa (dla regresji)

$$\sigma(z) = z$$

Pochodna:.

$$\sigma'(z) = 1$$

Własności:.

- brak nieliniowości – stosowana tylko w warstwach wyjściowych do regresji,
- niezdolna do modelowania nieliniowych zależności.

3. Funkcje aktywacji – definicje, własności i porównanie

3.1. Wprowadzenie

Funkcje aktywacji są kluczowym elementem sieci neuronowych, umożliwiającym modelowanie nieliniowych relacji.

3.2. Klasyczne funkcje aktywacji

3.2.1. Funkcja sigmoidalna

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

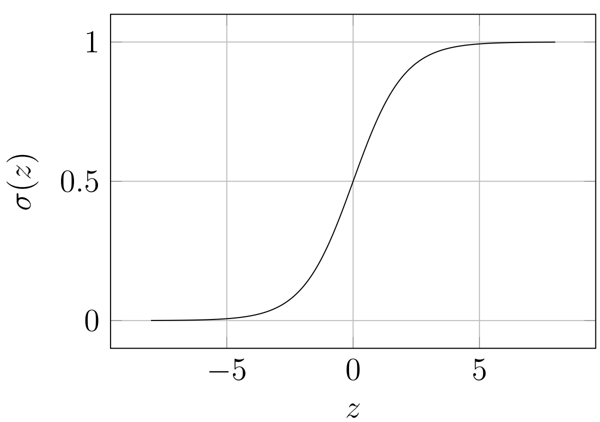
Pochodna:

$$\sigma'(z) = \sigma(z)(1 - \sigma(z)).$$

Własności:

- zakres $(0, 1)$,
- saturacja dla $|z| \to \infty$,
- powoduje zanikający gradient,
- monotoniczna i gładka.

Wykres (TikZ):.



—

3.2.2. Tangens hiperboliczny

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

Pochodna:

$$\tanh'(z) = 1 - \tanh^2(z).$$

Własności:

- zakres $(-1, 1)$,
- funkcja nieparzysta,
- gładka i monotoniczna,
- wciąż podatna na zanik gradientu.

—

3.2.3. ReLU

$$\text{ReLU}(z) = \max(0, z)$$

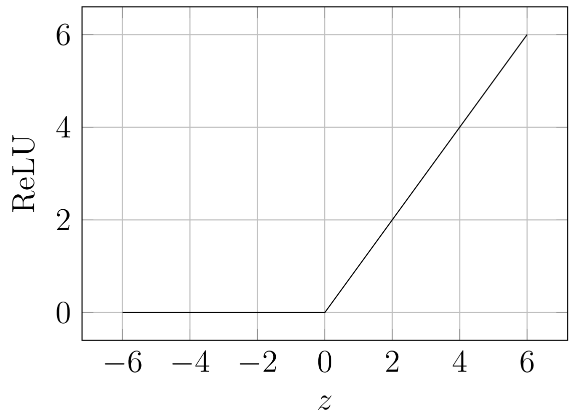
Pochodna:

$$\text{ReLU}'(z) = \begin{cases} 0, & z < 0, \\ 1, & z > 0. \end{cases}$$

Własności:

- brak saturacji dla $z > 0$,
- wydajna obliczeniowo,
- problem martwych neuronów.

Wykres:.



—

3.3. Nowoczesne funkcje aktywacji

3.3.1. Swish

$$\text{Swish}(z) = z \cdot \sigma(z)$$

Pochodna:

$$\text{Swish}'(z) = \sigma(z) + z\sigma(z)(1 - \sigma(z)).$$

Własności:

- gładka i nieliniowa,
- zachowuje niewielkie wartości ujemne,
- lepsza stabilność niż ReLU.

—

3.3.2. SiLU (Sigmoid Linear Unit)

$$\text{SiLU}(z) = z\sigma(z)$$

Pochodna:

$$\text{SiLU}'(z) = \sigma(z) + z\sigma(z)(1 - \sigma(z)).$$

Własności:

- identyczna z Swish-1,
- stosowana w YOLOv5,
- gładka i stabilna.

—

3.3.3. GELU

$$\mathrm{GELU}(z) = z\Phi(z)$$

Aproksymacja:

$$\mathrm{GELU}(z) \approx 0.5z\left(1 + \tanh\left(\sqrt{\frac{2}{\pi}}(z + 0.044715z^3)\right)\right)$$

Własności:

- używana w Transformerach,
- probabilistyczna interpretacja aktywacji,
- bardzo gładka.

—

3.3.4. Mish

$$\mathrm{Mish}(z) = z \cdot \tanh(\mathrm{softplus}(z))$$

Pochodna:

$$\mathrm{Mish}'(z) = \tanh(\mathrm{softplus}(z)) + z\,\mathrm{sech}^2(\mathrm{softplus}(z))\,\sigma(z)$$

Własności:

- jedna z najgładszych funkcji aktywacji,
- lepsze działanie niż ReLU w detekcji i klasyfikacji obrazów.

—

3.4. Tabela porównawcza

Funkcja	Gładkość	Ujemne wartości	Saturacja	Zastosowanie
Sigmoid	gładka	nie	tak	klasyfikacja
Tanh	gładka	tak	tak	RNN
ReLU	częściowo	nie	tylko <0	CNN
Swish	gładka	tak	nie	sieci głębokie
GELU	bardzo gładka	tak	nie	NLP, Transformers
Mish	ekstremalnie gładka	tak	nie	SOTA CNN
SiLU	gładka	tak	nie	YOLO