

# **Brain optimizer**

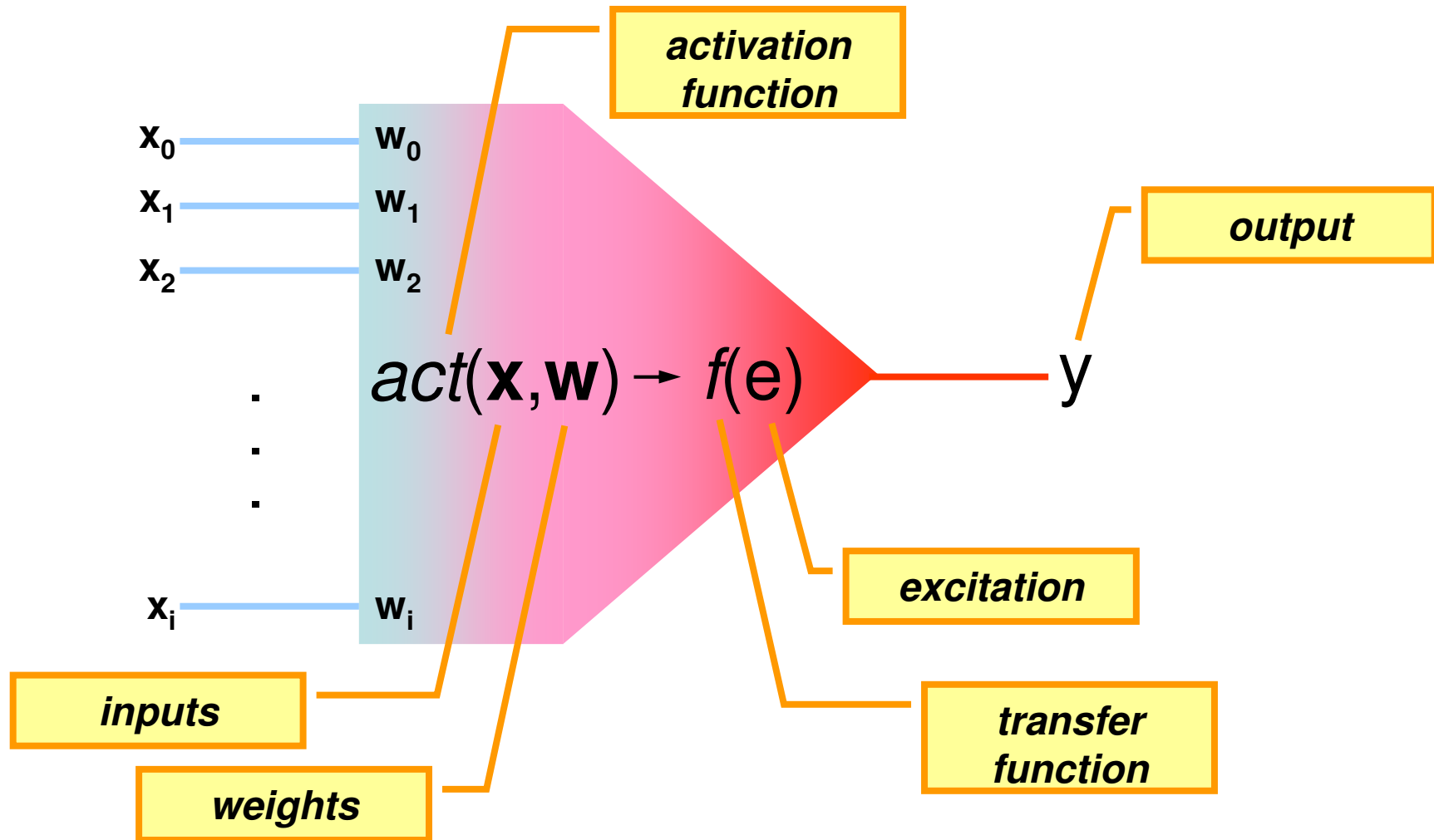
Symplifikator sieci neuronowej

Piotr Sobkowski

# Optymalizacja sieci neuronowych

- Wyszukiwanie miejsc optymalizacji
- Redukcja sieci
- Minimalizacja błędów wyników zredukowanej sieci
- Mniejsza złożoność obliczeniowa
- Zbliżone wyniki przy ustalonym progu tolerancji błędów

# Neuron - schema



# Framsticks, neurony

| symbol  | nazwa               | wejścia |
|---------|---------------------|---------|
| [N]     | Sigmoid             | any     |
| [Nu]    | Sigmoid             | any     |
| [D]     | Differentiate       | any     |
| [Ch]    | Channelize          | any     |
| [ChMux] | Channel multiplexer | 2       |
| [ChSel] | Channel selector    | 1       |
| [Sin]   | Sin generator       | 1       |
| [Delay] | Delay               | 1       |
| [Nn]    | Noisy neuron        | any     |
| [Thr]   | Threshold           | 1       |

# Neuron sigmoidalny [N]

$$input = act(\mathbf{x}, \mathbf{w}) = \sum_i w_i x_i$$

**activation function**

$$velocity_t = force \cdot (input - state_{t-1}) + interia \cdot velocity_t$$

$$state_t = state_{t-1} + velocity_t$$

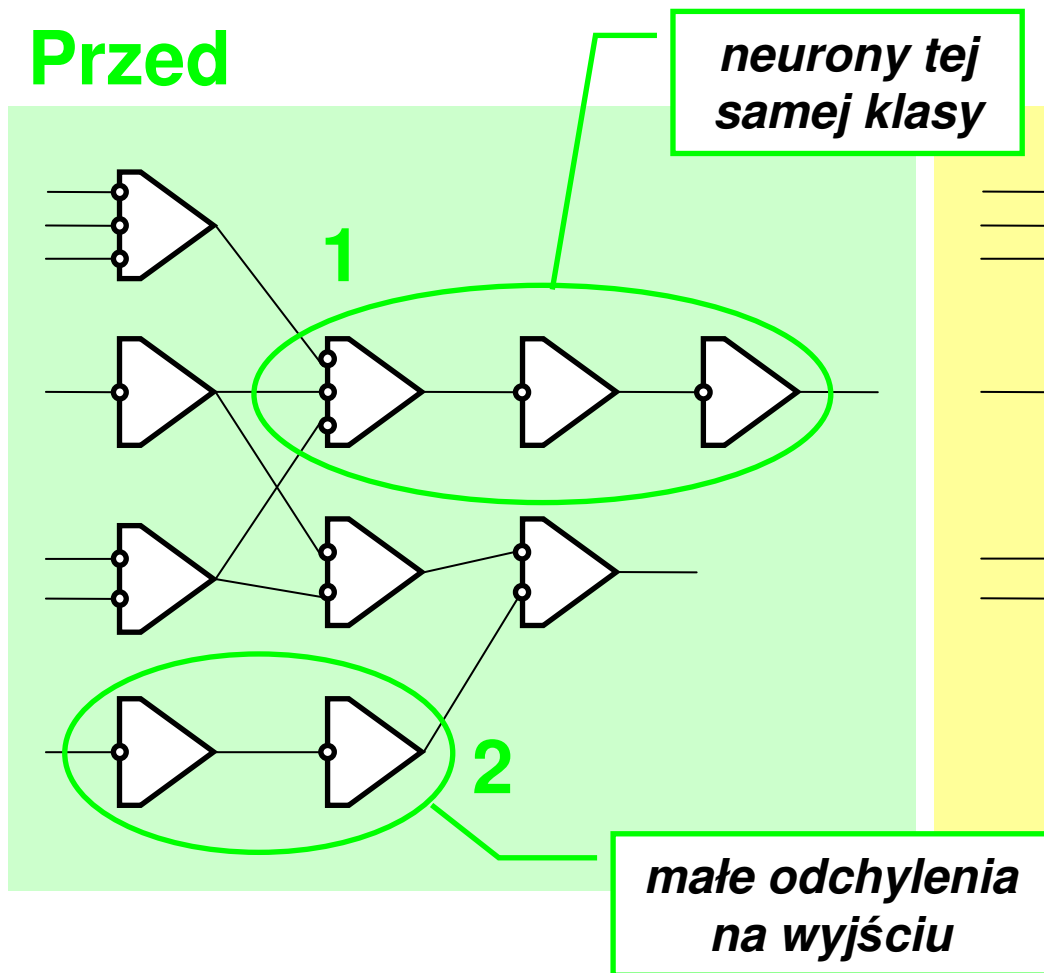
$$output = f(velocity_t) = \frac{2}{1 + e^{-sigmo \cdot velocity_t}} - 1$$

force float 0...999  
interia float -99999...99999  
sigmo float -1...1

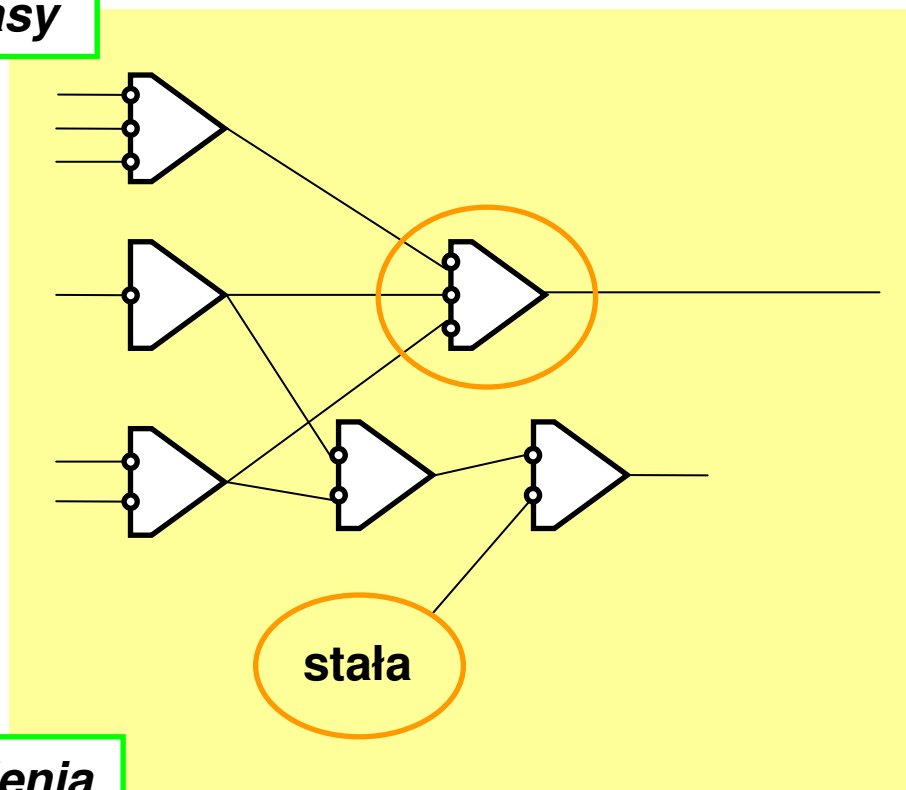
**transfer function**

# Miejsca optymalizacji

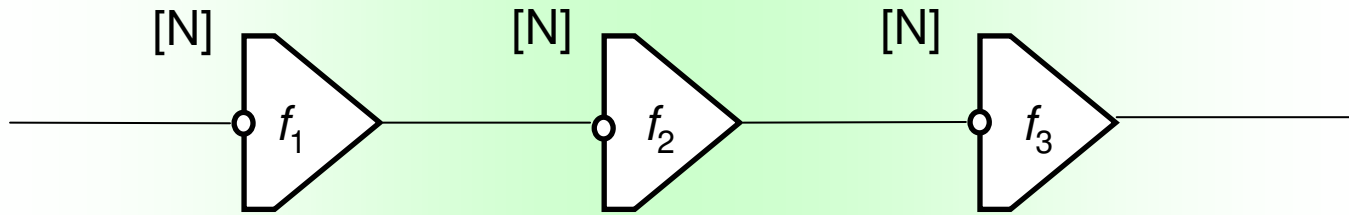
Przed



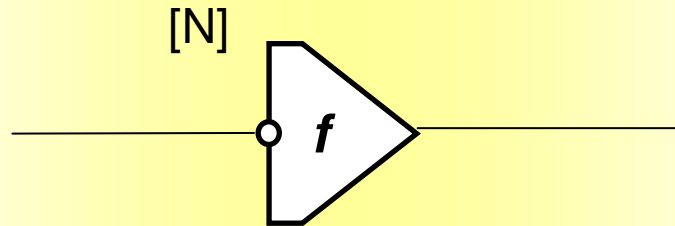
Po



# Redukcja neuronów

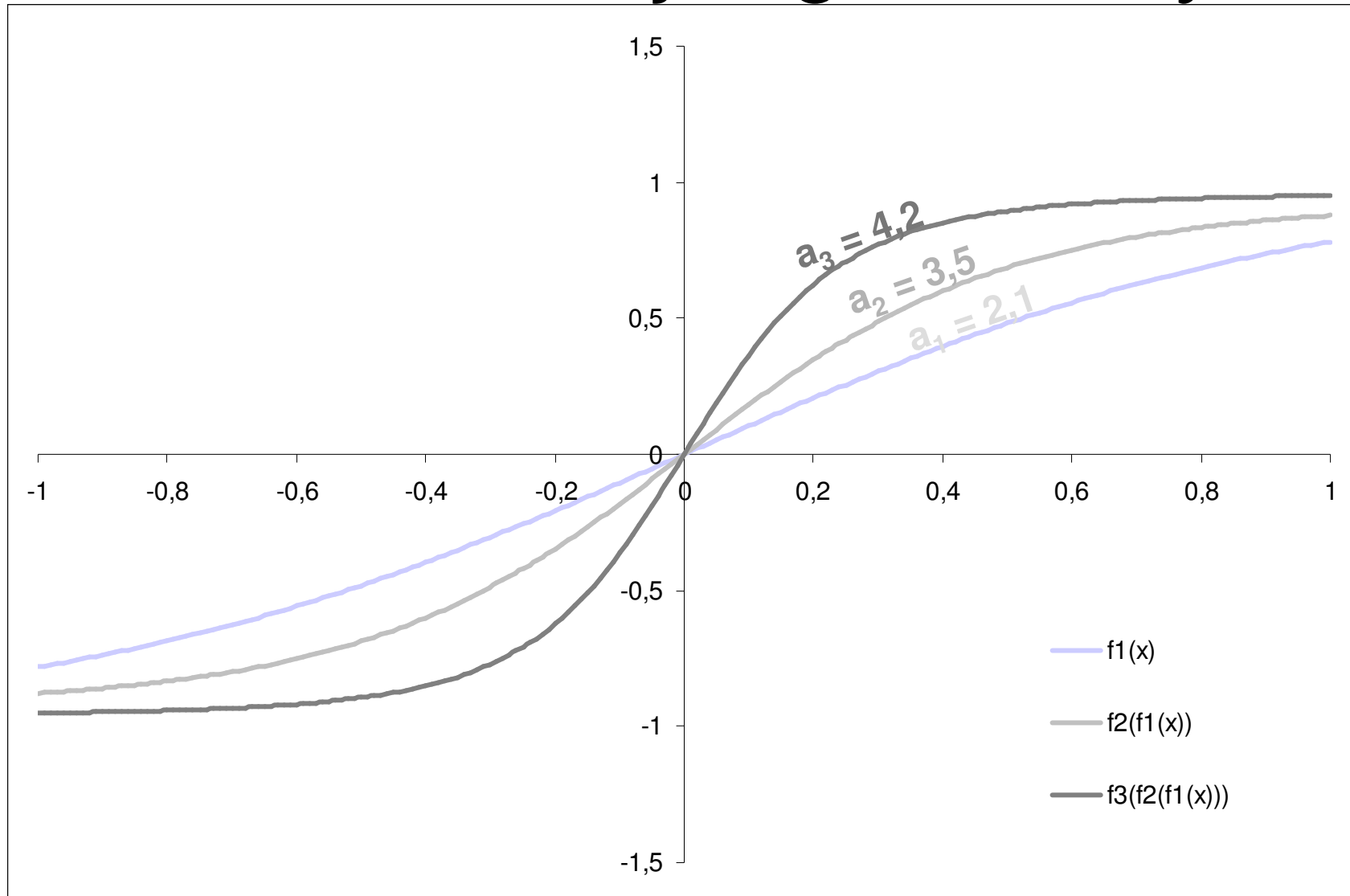


$$f_1(x) = \frac{2}{1+e^{-a_1 \cdot x}} - 1 \quad f_2(f_1(x)) = \frac{2}{1+e^{-a_2 \cdot f_1(x)}} - 1 \quad f_3(f_2(f_1(x))) = \frac{2}{1+e^{-a_3 \cdot f_2(f_1(x))}} - 1$$



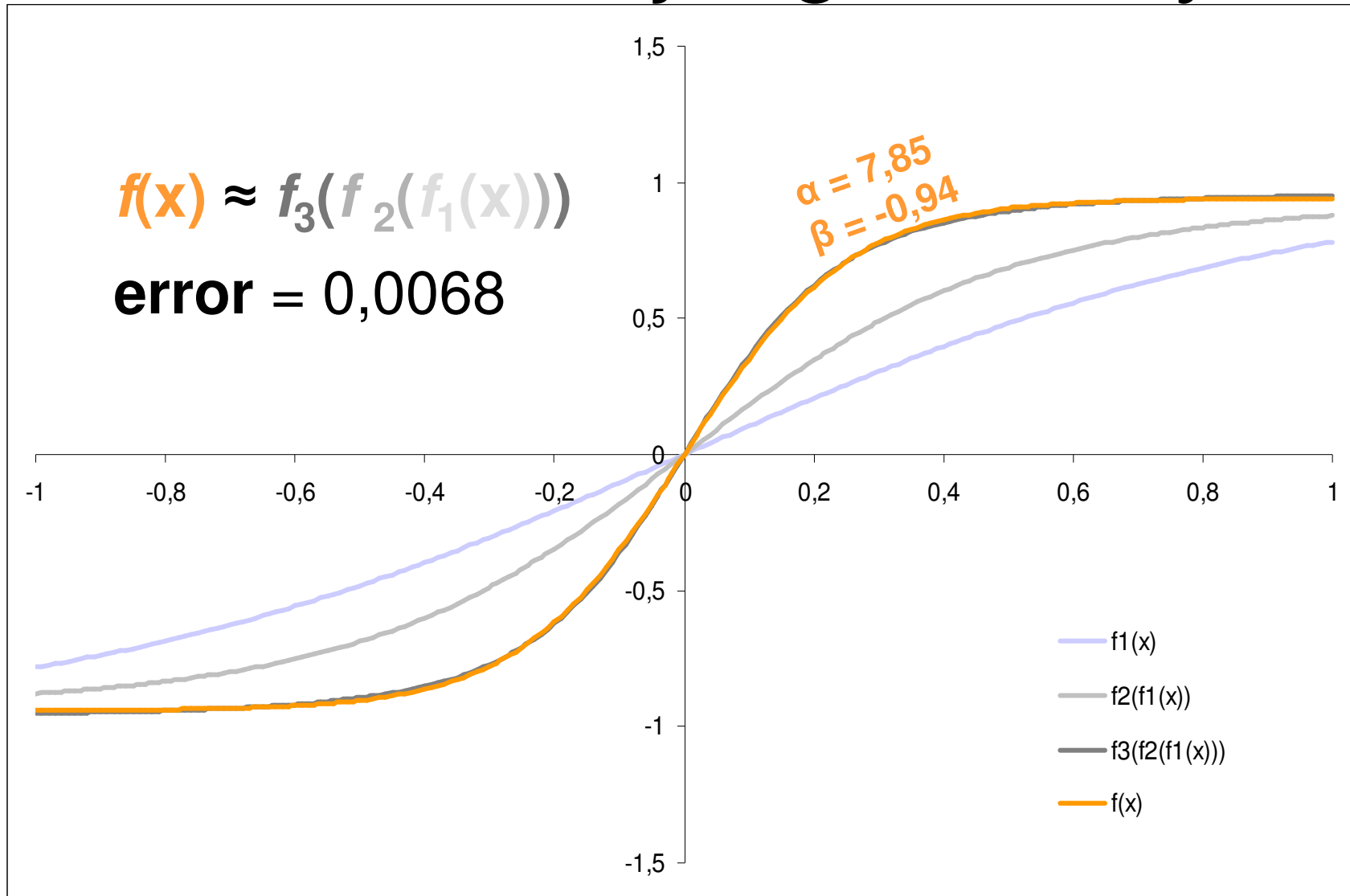
$$f_3(f_2(f_1(x))) \approx f(x) = \beta \left( \frac{2}{1+e^{-\alpha x}} - 1 \right)$$

# Złożenie funkcji sigmoidalnych





# Złożenie funkcji sigmoidalnych



# Realizacja

- Środowisko Framsticks, skrypty
- Analiza sieci neuronowej kreatury
- Budowa uproszczonego modelu
- Testowanie
- Zastąpienie sieci modelem w przypadku gdy model spełnia oczekiwania