



- neuroni su u 2., 3. i 4. sloju.
- svaki brid na gornjoj slici je ujedno i težina (faktor) kojom se signal koji putuje tim bridom množi
- svaki neuron ima također jednu slobodnu težinu koju ćemo označiti s  $W_{Si}$ , 'i' je broj neurona
- svaki neuron računa težinsku sumu ulaza; npr. neuron 9 računa
 
$$net_9 = W_A \cdot y_6 + W_B \cdot y_7 + W_C \cdot y_8 + W_{S9}$$
 neuron 2 računa
 
$$net_2 = W_D \cdot X + W_{S2}$$
- izlaz neurona slojeva 2. i 3. dobije se propuštanjem  $net$ -a kroz nelinearnost oblike  $\frac{1}{1+e^{-x}}$ ; npr. izlaz  $y_6$  računa se kao:
 
$$y_6 = \frac{1}{1 + e^{-net_6}}$$
- izlaz izlaznog neurona je direktno  $net$ ;  $y = net_9$

## ZADATAK

Zadane su funkcije:

$$a) f(x) = \sin(x), \quad x \in [0, 2\pi]$$

$$b) f(x) = \sin(x) + \sin\left(4x + \frac{\pi}{7}\right) \cdot \frac{1}{2} \quad x \in [0, 2\pi]$$

## Priprema podataka

Za obje funkcije unaprijed pripremite 40 parova  $(x, f(x))$  pri čemu domenu od  $x$  uniformno uzorkujte

Zadatok je pronaći takve težinske faktore uz koje će mreža raditi minimalnu pogrešku nad uzorcima koje ste pripremili. Funkcija koju optimirate je dakle  $E(\vec{w})$  koja se računa kao srednja vrijednost apsolutnih vrijednosti razlika zadanog i dobivenog izlaza za svaki pripremljeni uzorak.

Prijedlog: čuvajte sve težine kao jedno polje double-ova; svaki neuron tada može pamtiati samo indekse njegovih težina čime je trivijalno mreži zamijeniti težine. Isti trik možete primijeniti i za sve ulaze tj. izlaze (možete da je izlaz jednog neurona uvijek ulaz nekog (ili više) drugih).