## Úloha s lineárním neuronem

Uvažujme vstupní signál jako funkci času $u(t)$ a odezvu systému (reakci na tento signál) $y(t)$. Vstpní i výstupní signál máme v podobě vektoru $\mathbf{u}$ a vektoru $\mathbf{y}$, kde jednotlivé prvky odpovídají jednotlivým hodnotám změřeným s konstatním vzorkováním, viz soubor *uy.txt*[1].

Cílem je predikovat chování systému (tedy nějakým způsobem predikovat hodnoty funkce v určitém čase $y(t)$), v závislosti na zadaném vstupu $u(t)$ a stavu systému v minulosti, tedy $y(t_i < t)$. Využijeme proto diskrétních dat a provedeme tzv. linerání interpolaci (resp. sestrojíme lineární neuron) v podobě

$$Y_{\text{pred}} = \mathbf{W} \cdot \mathbf{X}$$

kde $\mathbf{Y}_{\text{pred}}$ znaží predikovanou hodnotu, $\mathbf{W}$ vektor vah, které budeme chtít určit a $\mathbf{X}$ vektor argumentů neuronu. Konstrukce lineárního neuronu spočívá právě ve vhodné volbě argument pro neuron $\mathbf{X}$.

V našem případě budeme volit

$$Y_{\text{pred}}(t_k) = \mathbf{W} \cdot \mathbf{X}(t_k, \mathbf{u}, \mathbf{v})$$

$$Y_{\text{pred}}(t_k) = \mathbf{W} \cdot \mathbf{X} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} \cdot \begin{bmatrix} y(t_{k-1}) \\ y(t_{k-2}) \\ u(t_{k-1}) \\ u(t_{k-2}) \end{bmatrix} \tag{1}$$

neboli hodnota v časovém kroku $t_k$ bude *předpovězěna* na základě vstupního signálu $u$ v předchozích dvou krocích (tj. $u(t_{k-1})$ a $u(t_{k-2})$) a skutečného stavu systému $y$ v předchozích dvou krocích (tj. $y(t_{k-1})$ a $y(t_{k-2})$) a kombinaci váhových parametrů $\mathbf{W}$, které musíme určit.

### Algoritmus programu

- K dispozici máme $N$ časových kroků $k$, jednotlivé hodnoty vektorů $u$ a $y$ odpovídají jednotlivým časovým krokům
- V každém časovém kroku $k$ vyčíslíme vektor $\mathbf{X}(t_k)$
- Provedeme predikci hodnot $Y_{\text{pred}}(t_k) = \mathbf{W} \cdot \mathbf{X}(t_k)$
- Určíme chybu mezi měřenými daty a vypočítanou (predikovanou) hodnotou $e(t_k) = y(t_k) - Y_{\text{pred}}(t_k)$
- Určíme, o kolik je potřeba změnit koeficienty $\mathbf{W}$ tj. $\Delta\mathbf{W}(t_k) = \mu e(t_k)\mathbf{X}(t_k)$ , přičemž $\mu = 0.1$ je tzv. *learning rate*
- Provedeme aktualizaci vah $\mathbf{W} = \mathbf{W} + \Delta\mathbf{W}(t_k)$

Celý proces opakujme 10-krát, tj. proveďme 10 *epoch* trénování. Poté bychom měli mít již dostatečně dobře přizpůsobené váhy a dostatečně dobrou predikci hodnot $Y_{\text{pred}}(t_k)$, resp. vektoru $\mathbf{Y}_{\text{pred}}$.

*pozn.* Vektor $\mathbf{W}$ inicializujte náhodnými čísly.

*pozn.* Jedná se o tzv. krokovou adaptaci, učení (přizpůsobování vah) se děje v každém časovém kroku.

[1]Data a námět jsou s laskavým poděkováním převzata z ukázek Doc. Bukovskeho.

### Zpracování výsledků

Výsledky zobrazte pomocí funkce *subplot* (*help subplot*) jako tři grafy pod sebou.

- V prvním grafu bude vynesen vstupní signál a měřená data
- V druhém grafu budou vynesena měřena a predikována data
- Ve třetím grafu bude vynesena chyba mezi perdikovanymi a měřenými daty

### pozn. Kde se bere vztah pro aktualizaci vah?

Po provedení predikce pomocí neuronu určujeme rozdíl mezi predikovanými a měřenými (trénovacími daty) jako $e(t_k) = y(t_k) - Y_{\mathrm{pred}}(t_k)$. Následně zavedeme tzv. *chybové kritérium* $Q(t_k)$, tedy nějakou funkci chyby v časovém kroku, například kvadrát chyby $Q(t_k) = e^2(t_k)$, který vystihuje *vzdálenost* mezi trénovacími daty a predikovanou hodnotou.

Protože k predikci jsme použili seznam vah $\mathbf{W}$ a protože predikce nedává hodnotu, která by přesně odpovídala měřeným datům, je potřeba vhodným způsobem upravit koeficienty $\mathbf{W}$ tak, abychom minimalizovali chybu $e$. Kde je minimalizace, tam je gradient. Zvolíme tedy změnu hodnot vah $\Delta\mathbf{W}$ tak, že půjde o nějaký přírůstek ve směru gradientu chyby $e$, neboli

$$\mathbf{W}(t_k) = \mathbf{W}(t_{k-1}) + \Delta\mathbf{W}(t_k); \quad \Delta\mathbf{W} = -\frac{\partial Q(t_k)}{\partial \mathbf{W}}\frac{\mu}{2}$$

kde $\mu$ je dodatečný, uměle vložený koeficient (tzv. *koeficient učení, learning rate*) přidaný za účelem stabilizace. Po troše počítání

$$\Delta\mathbf{W} = \frac{\partial Q(t_k)}{\partial \mathbf{W}}\frac{\mu}{2} = -\frac{\partial e^2(t_k)}{\partial \mathbf{W}}\frac{\mu}{2} = -\frac{\partial e(t_k)}{\partial \mathbf{W}}2e(t_k)\frac{\mu}{2} = -\frac{\partial e(t_k)}{\partial \mathbf{W}}2e(t_k)\frac{\mu}{2} = -\mu_n e(t_k)\frac{\partial(y(t_k) - Y_{\mathrm{pred}}(t_k))}{\partial \mathbf{W}} = \dots$$

kde jsme všechny konstanty zabalili do koeficientu učení $\mu$ a protože měřená data $y(t_k)$ jsou konstantní

$$\dots = \mu_n e(t_k)\frac{\partial Y_{\mathrm{pred}}(t_k)}{\partial \mathbf{W}}$$

odtud je již patrné, že

$$\Delta\mathbf{W}(t_k) = \mu_n e(t_k)\mathbf{X}(t_k)$$