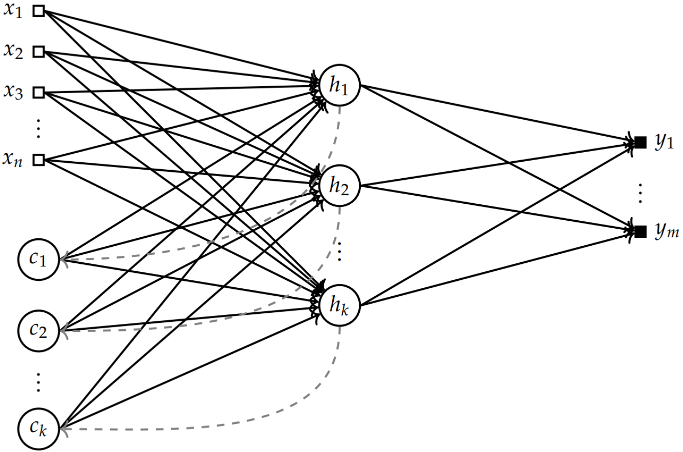
Neural Networks

7. Recurrent Networks

Center for Cognitive Science
Department of Applied Informatics
Faculty of Mathematics, Physics and Informatics
Comenius University in Bratislava

Tuesday 9th April, 2024

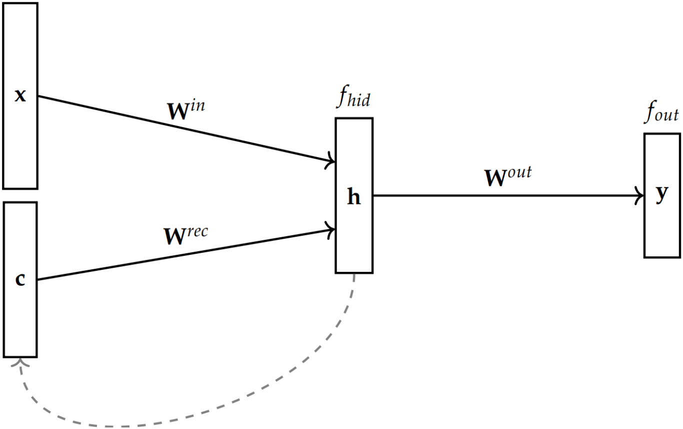
Simple Recurrent Network (SRN) – architecture



x - input ***c*** - context ***h*** - hidden layer ***y*** - output

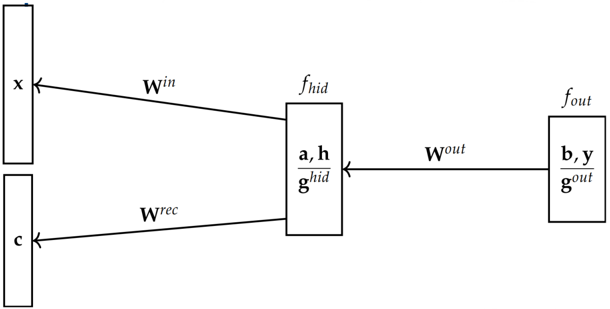
* bias terms applied on ***x***, ***c*** and ***h***

Simple Recurrent Network (SRN) – forward pass



$$\boldsymbol{c}(t) = \boldsymbol{h}(t-1) \quad \boldsymbol{h} = f_{hid}(\boldsymbol{W}^{in}\boldsymbol{x}' + \boldsymbol{W}^{rec}\boldsymbol{c}') \quad \boldsymbol{y} = f_{out}(\boldsymbol{W}^{out}\boldsymbol{h}')$$

Simple Backpropagation



$$\boldsymbol{g}^{out} = f'_{out}(\boldsymbol{b}) \odot (\boldsymbol{d} - \boldsymbol{y})$$
$$\boldsymbol{g}^{hid} = f'_{hid}(\boldsymbol{a}) \odot \boldsymbol{W}^{out^T} \boldsymbol{g}^{out}$$

$$\Delta \boldsymbol{W}^{in} = \boldsymbol{g}^{hid} \boldsymbol{x}'^T$$
$$\Delta \boldsymbol{W}^{rec} = \boldsymbol{g}^{hid} \boldsymbol{c}'^T$$
$$\Delta \boldsymbol{W}^{out} = \boldsymbol{g}^{out} \boldsymbol{h}'^T$$

$\boldsymbol{u} \odot \boldsymbol{v}$ – element-wise multiplication of vectors $\boldsymbol{u}$ and $\boldsymbol{v}$

Simple Recurrent Network – task

- **one-step prediction:**
  - given a sequence of inputs $\boldsymbol{x}(1), ..., \boldsymbol{x}(T)$
  - predict the next value $\boldsymbol{x}(T+1)$
  - context-based regression $\boldsymbol{x}(t) \rightarrow \boldsymbol{x}(t+1)$
- we need to reset the context to neutral activation before each run, e.g.
  - set $\boldsymbol{c}(1) = \boldsymbol{h}(0)$ as if $\boldsymbol{x}(0) = \boldsymbol{c}(0) = \mathbf{0}$

Task

- elman.py TODO:
    - initialize weights (can be sensitive)
    - initialize context by feeding zero input
    - forward pass & backward pass
    - weight adjustment