Fully recurrent network

Output activation function & mainly depends on the task & not on the architecture of the network

Activation function or non-linearity

Forward pass of FRNN :

Recurrent weight matrix. Holds the weights of the loop connections & determines time dependent behaviour of the system.

$$\vec{s}(t) = \vec{W}^T \vec{x}(t) + \vec{R}^T \vec{a}(t-1)$$

$$\vec{a}(t) = f(\vec{s}(t))$$

$$\hat{\vec{y}}(t) = \varphi(\vec{V}^T \vec{a}(t))$$

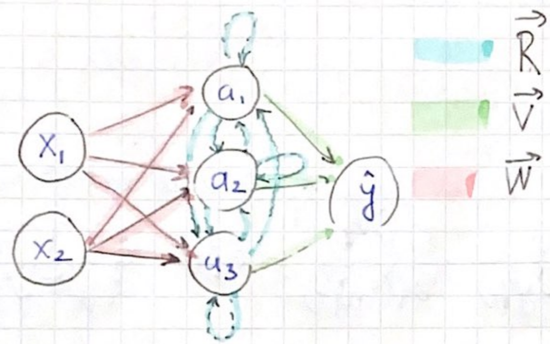
Input weight matrix. It's trans-on between input space $\mathbb{R}^D$ & hidden space $\mathbb{R}^I$, i.e. maps input features to hidden repr-ons

where $\vec{W} \in \mathbb{R}^{D \times I}$, $\vec{R} \in \mathbb{R}^{I \times I}$, $\vec{V} \in \mathbb{R}^{I \times K}$

Output weight matrix. Maps the network's hidden repr-s to the output space $\mathbb{R}^K$

In contrast to the Elman network, we'll now treat the recurrent weights $\vec{R}$ as trainable $\Rightarrow$ weight updates on $\vec{R}$

Also, in contrast to the Elman networks, a hidden unit may now not only depend on a former version of itself, but also of all neighboring neurons.



Before we look at the derivatives with respect to $r_{ij} = \vec{R}$, $v_{ij} = \vec{V}$, $w_{id} = \vec{W}$ we introduce an ancillary concept: derivatives of the loss with respect to pre-activations, which are called deltas.

Deltas are helpful in backpropagation because they enable a recursive calculation of the gradients through time.

We define :

$$\vec{\delta}(t)^T = \frac{\partial L}{\partial \vec{s}(t)} = \frac{\partial L}{\partial \vec{a}(t)} \frac{\partial \vec{a}(t)}{\partial \vec{s}(t)} =$$

$$= \left(\frac{\partial L(\vec{y}(t), \hat{\vec{y}}(t))}{\partial \vec{a}(t)} + \frac{\partial L}{\partial \vec{s}(t+1)} \cdot \frac{\partial \vec{s}(t+1)}{\partial \vec{a}(t)} \right) \frac{\partial \vec{a}(t)}{\partial \vec{s}(t)} =$$

$$= \left(\frac{\partial L}{\partial \hat{\vec{y}}(t)} \cdot \frac{\partial \hat{\vec{y}}(t)}{\partial \vec{a}(t)} + \frac{\partial L}{\partial \vec{s}(t+1)} \cdot \frac{\partial \vec{s}(t+1)}{\partial \vec{a}(t)} \right) \cdot \frac{\partial \vec{a}(t)}{\partial \vec{s}(t)} =$$

$$= \left(\vec{e}(t)^T \operatorname{diag}(\varphi'(\vec{V}^T \vec{a}(t))) \vec{V}^T + \vec{\delta}(t+1)^T \vec{R}^T \right) \operatorname{diag}(f'(\vec{s}(t)))$$

derivative of the loss w.r.t. model output

derivative of the model output w.r.t. output weights

Now, let's express derivatives w.r.t. $\vec{R}$ and $\vec{W}$:

(I) $$\frac{\partial L}{\partial \vec{R}} = \sum_{t=1}^{T} \frac{\partial L}{\partial \vec{s}(t)} \cdot \frac{\partial \vec{s}}{\partial \vec{R}} = \sum_{t=1}^{T} \vec{\delta}(t) \cdot \frac{\partial\left(\vec{W}(t)^T \vec{x}(t) + \vec{R}(t)^T \vec{a}(t-1)\right)}{\partial \vec{R}} =$$

$$= \sum_{t=1}^{T} \vec{\delta}(t) \cdot \vec{a}(t-1)$$

(II) $$\frac{\partial L}{\partial \vec{W}} = \sum_{t=1}^{T} \frac{\partial L}{\partial \vec{s}(t)} \cdot \frac{\partial \vec{s}}{\partial \vec{W}} = \sum_{t=1}^{T} \vec{\delta}(t) \cdot \frac{\partial\left(\vec{W}(t)^T \vec{x}(t) + \vec{R}(t)^T \vec{a}(t-1)\right)}{\partial \vec{W}} =$$

$$= \sum_{t=1}^{T} \vec{\delta}(t) \cdot \vec{x}(t)$$

In order to derive wrt $\vec{V}$, we have to include additional definition:

$$\boxed{\vec{\psi}(t)^T} = \frac{\partial L}{\partial \vec{V}^T \vec{a}(t)} = \frac{\partial L}{\partial \hat{\vec{y}}(t)} \cdot \frac{\partial \hat{\vec{y}}(t)}{\partial \vec{V}^T \vec{a}(t)} = \vec{e}(t)^T \cdot \frac{\partial\left(\varphi(\vec{V}^T \vec{a}(t))\right)}{\partial(\vec{V}^T \vec{a}(t))} =$$

$$= \boxed{\vec{e}(t)^T \circ \varphi'(\vec{a}(t)^T \vec{V})}$$

Hence $\vec{a}(t)$ can be multiplied with $\psi(t)$ and we can get $\frac{\partial L}{\partial \vec{V}}$

(III) $$\frac{\partial L}{\partial \vec{V}} = \boxed{\vec{\psi}(t)}\, a(t)^T = \boxed{\left(\vec{e}(t)^T \circ \varphi'(\vec{a}(t)^T \vec{V}\right)}\, a(t)^T$$