Recurrent_Neural_Networks

December 6, 2016

1 Configuration

```
In [19]: #reveal configuration
         from notebook.services.config import ConfigManager
         cm = ConfigManager()
         cm.update('livereveal', {
                 'theme': 'white',
                 'transition': 'none',
                 'controls': 'false',
                 'progress': 'true',
         })
```

```
Out[19]: {'controls': 'false',
          'progress': 'true',
          'theme': 'white',
          'transition': 'none'}
```

```
In [20]: %%javascript
         require(['base/js/utils'],
         function(utils) {
            utils.load_extensions('calico-spell-check', 'calico-document-tools', 'c
         });
```

```
<IPython.core.display.Javascript object>
```

```
In [21]: %load_ext tikzmagic
```

```
The tikzmagic extension is already loaded. To reload it, use:
  %reload_ext tikzmagic
```

```
In [22]: %%html
         <style>
         .red { color: #E41A1C; }
         .orange { color: #FF7F00 }
         .yellow { color: #FFC020 }
         .green { color: #4DAF4A }
```

```css
.blue { color: #377EB8; }
.purple { color: #984EA3 }

h1 {
    color: #377EB8;
}

ctb_global_show div.ctb_hideshow.ctb_show {
    display: inline;
}

div.tabContent {
    padding: 0px;
    background: #ffffff;
    border: 0px;
}

.left {
    float: left;
    width: 50%;
    vertical-align: text-top;
}

.right {
    margin-left: 50%;
    vertical-align: text-top;
}

.small {
    zoom: 0.9;
    -ms-zoom: 0.9;
    -webkit-zoom: 0.9;
    -moz-transform:  scale(0.9,0.9);
    -moz-transform-origin: left center;
}

.verysmall {
    zoom: 0.75;
    -ms-zoom: 0.75;
    -webkit-zoom: 0.75;
    -moz-transform:  scale(0.75,0.75);
    -moz-transform-origin: left center;
}


.tiny {
    zoom: 0.6;
    -ms-zoom: 0.6;
```

```
    -webkit-zoom: 0.6;
    -moz-transform:  scale(0.6,0.6);
    -moz-transform-origin: left center;
}


.rendered_html blockquote {
    border-left-width: 0px;
    padding: 15px;
    margin: 0px;
    width: 100%;
}

.rendered_html th {
    padding: 0.5em;
    border: 0px;
}

.rendered_html td {
    padding: 0.25em;
    border: 0px;
}

#for reveal
.aside .controls, .reveal .controls {
    display: none !important;
    width: 0px !important;
    height: 0px !important;
}

.rise-enabled .reveal .slide-number {
    right: 25px;
    bottom: 25px;
    font-size: 200%;
    color: #377EB8;
}

.rise-enabled .reveal .progress span {
    background: #377EB8;
}

.present .top {
    position: fixed !important;
    top: 0 !important;
}

.present .rendered_html * + p, .present .rendered_html p, .present .render
    margin: 0.5em 0;
```

```
}

.present tr, .present td {
    border: 0px;
    padding: 0.35em;
}

.present th {
    border: 1px;
}

present .prompt {
    min-width: 0px !important;
    transition-duration: 0s !important;
}

.prompt {
    min-width: 0px !important;
    transition-duration: 0s !important;
}

.rise-enabled .cell li {
    line-height: 135%;
}

</style>

%load_ext tikzmagic
```

```
<IPython.core.display.HTML object>
```

Recurrent Neural Networks
University College London
Statistical Natural Language Processing Course, UCL
9th December 2016

2 Language Models (LMs)

A language model computes a **probability** for a **sequence of words**

$$p(w_1, \ldots, w_d)$$

Useful in a miriad of NLP tasks, such as machine translation:

$$p(\text{I}, \text{go}, \text{home}) > p(\text{I}, \text{go}, \text{house})$$

In *n-gram language models*, the probability $p(w_1, \ldots, w_d)$ of observing the sentence $(w_1, \ldots, w_d)$ is **approximated** as:

$$p(w_1, \ldots, w_T) = \prod_{i=1}^{m} p(w_i \mid w_1, \ldots, w_{i-1}) \approx \prod_{i=1}^{m} p(w_i \mid w_{i-(n-1)}, \ldots, w_{i-1})$$

Example with a **bigram** ($n = 2$) **language model**:

$$p(\text{I}, \text{go}, \text{home}) \approx p(\text{I} \mid \text{<s>})\ p(\text{go} \mid \text{I})\ p(\text{home} \mid \text{go})$$

2.0.1 A Simple Recurrent Neural Network (RNN) LM

Given a list of **word vectors** (e.g. one-hot, word2vec, GloVe):

$$x_1, \ldots, x_{t-1}, x_t, x_{t+1}, \ldots, x_d$$

At a single time step:

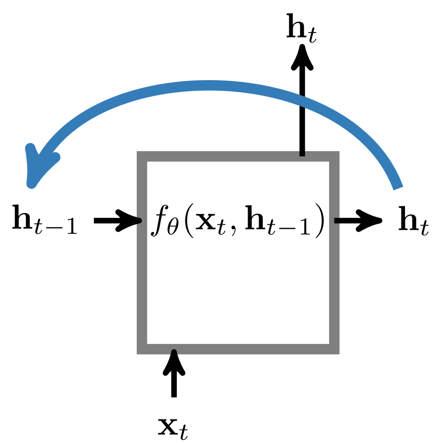
$$\begin{aligned}
\mathbf{h}_t &= \text{sigmoid}(\mathbf{W}^h \mathbf{h}_{t-1} + \mathbf{W}^x \mathbf{x}_t + \mathbf{b}) \\
\mathbf{y}_t &= \text{softmax}(\mathbf{W}^o \mathbf{h}_t)
\end{aligned}$$

where $\mathbf{y}_t \in \mathbb{R}^{|V|}$ is a **probability distribution** over words in the vocabulary V. The probability that the t-th word is w_j is given by:

$$p(w_j \mid x_t, \ldots, x_1) = \mathbf{y}_{t,j}$$

In [23]:
```
%%tikz -l arrows -s 1000,400 -sc 0.65 -f svg

\newcommand{\lstm}{
\definecolor{nice-red}{HTML}{E41A1C}
\definecolor{nice-orange}{HTML}{FF7F00}
\definecolor{nice-yellow}{HTML}{FFC020}
\definecolor{nice-green}{HTML}{4DAF4A}
\definecolor{nice-blue}{HTML}{377EB8}
\definecolor{nice-purple}{HTML}{984EA3}

%lstm first step

%lstm module box
\draw[line width=3pt, color=black!50] (0,0) rectangle (3,3);
}

\lstm
\node[] at (0.5,-1.25) {$\mathbf{x}_t$};
\node[] at (-1.5,2) {$\mathbf{h}_{t-1}$};
\node[] at (4.25,2) {$\mathbf{h}_t$};
\node[] at (2.5,5) {$\mathbf{h}_t$};

\draw[ultra thick, ->, >=stealth'] (0.5,-0.75) -- (0.5,0);
\draw[ultra thick, ->, >=stealth'] (-0.75,2) -- (0,2);
\draw[ultra thick, ->, >=stealth'] (3,2) -- (3.75,2);
```

```
\draw[ultra thick, ->, >=stealth'] (2.5,3) -- (2.5,4.75);

\path[line width=3pt, ->, >=stealth', color=nice-blue] (4, 2.5) edge[bend

\node[] at (1.5,2) {$f_\theta(\mathbf{x}_t, \mathbf{h}_{t-1})$};
```



2.1 Example

Consider the word sequence (I, go, home) $\rightarrow (x_1, x_2, x_3, x_4)$

Reminder: $\mathbf{h}_t = \text{sigmoid}(\mathbf{W}^h \mathbf{h}_{t-1} + \mathbf{W}^x \mathbf{x}_t + \mathbf{b})$

$$\begin{aligned} \mathbf{h}_1 &= \text{sigmoid}(\mathbf{W}^h \mathbf{h}_0 + \mathbf{W}^x \mathbf{x}_1) & \hat{\mathbf{y}}_1 &= \text{softmax}(\mathbf{W}^o \mathbf{h}_1) \\ \mathbf{h}_2 &= \text{sigmoid}(\mathbf{W}^h \mathbf{h}_1 + \mathbf{W}^x \mathbf{x}_2) & \hat{\mathbf{y}}_2 &= \text{softmax}(\mathbf{W}^o \mathbf{h}_2) \\ \mathbf{h}_3 &= \text{sigmoid}(\mathbf{W}^h \mathbf{h}_2 + \mathbf{W}^x \mathbf{x}_3) & \hat{\mathbf{y}}_3 &= \text{softmax}(\mathbf{W}^o \mathbf{h}_3) \end{aligned}$$

$$p(\text{I}, \text{go}, \text{home}) = \hat{\mathbf{y}}_{1,[\text{I}]}\ \hat{\mathbf{y}}_{2,[\text{go}]}\ \hat{\mathbf{y}}_{3,[\text{home}]}$$

- $\mathbf{h}_0 \in \mathbb{R}^{D_h}$ is an initialization vector
- $\mathbf{W}^h \in \mathbb{R}^{D_h \times D_h}$, $\mathbf{W}^x \in \mathbb{R}^{D_h \times D_x}$, $\mathbf{W}^o \in \mathbb{R}^{|V| \times D_h}$ are model parameters

Such parameters can be **learned from data**.

2.2 Objective Function for LMs

Recall that $\mathbf{y}_t \in \mathbb{R}^{|V|}$ is a probability distribution over the vocabulary V.

We can train a RNN by minimizing the **cross-entropy loss**, predicting **words** instead of classes:

$$J^t = -\sum_{i=1}^{|V|} \mathbf{y}_{t,i} \log \hat{\mathbf{y}}_{t,i}, \quad \mathbf{y}_{t,i} = \begin{cases} 1 \text{ if the } t\text{-th word is } w_i, \\ 0 \text{ otherwise.} \end{cases}$$

Evaluation - negative of average log-probability over corpus:

$$J = -\frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{|V|}\mathbf{y}_{t,j}\log\hat{\mathbf{y}}_{t,j}$$

Or also **PERPLEXITY**:

$$PP(w_1,\ldots,w_T) = \sqrt[T]{\prod_{i=1}^{T}\frac{1}{p(w_i|w_1,\ldots,w_{i-1})}}$$

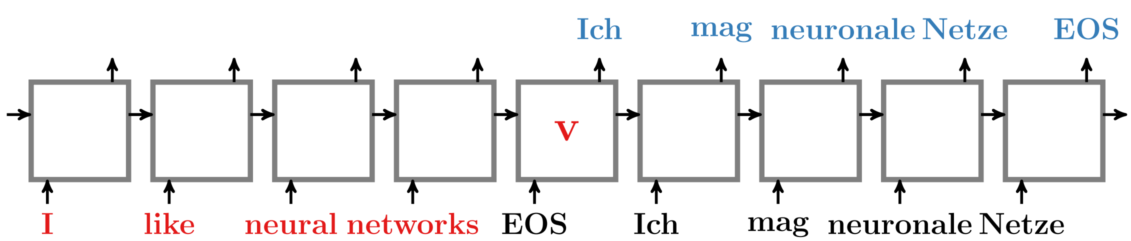
2.3 Sequence-to-Sequence Models

A RNN can also **generate** sequences - Seq2Seq models are composed by: - An **encoded** that processes the input and generates a vector $\mathbf{v} \in \mathbb{R}^d$ - A **decoder** that generates the output starting from $\mathbf{v}$

Widely popular in e.g. *Neural Machine Translation* and *Learning to Execute*

In [24]:
```
%%tikz -l arrows -s 1000,400 -sc 0.65 -f svg

\definecolor{nice-red}{HTML}{E41A1C}
\definecolor{nice-orange}{HTML}{FF7F00}
\definecolor{nice-yellow}{HTML}{FFC020}
\definecolor{nice-green}{HTML}{4DAF4A}
\definecolor{nice-blue}{HTML}{377EB8}
\definecolor{nice-purple}{HTML}{984EA3}

\newcommand{\lstm}{

%lstm first step

%lstm module box
\draw[line width=3pt, color=black!50] (0,0) rectangle (3,3);

\draw[ultra thick, ->, >=stealth'] (0.5,-0.75) -- (0.5,0);
\draw[ultra thick, ->, >=stealth'] (-0.75,2) -- (0,2);
\draw[ultra thick, ->, >=stealth'] (3,2) -- (3.75,2);
\draw[ultra thick, ->, >=stealth'] (2.5,3) -- (2.5,3.75);
}

%\lstm

%\node[] at (0.5,-1.25) {$\mathbf{x}_t$};
%\node[] at (-1.5,2) {$\mathbf{h}_{t-1}$};
%\node[] at (4.25,2) {$\mathbf{h}_t$};
%\node[] at (2.5,5) {$\mathbf{h}_t$};
%\path[line width=3pt, ->, >=stealth', color=nice-blue] (4, 2.5) edge[benc
%\node[] at (1.5,2) {$f_\theta(\mathbf{x}_t, \mathbf{h}_{t-1})$};
```

```
\foreach \x/\w in {0/I, 1/like, 2/neural, 3/networks} {
    \begin{scope}[shift={(\x*3.75,0)}]
        \lstm
        \node[font=\LARGE, text height=1.5ex, color=nice-red] at (0.5,-1.5
    \end{scope}
}

\foreach \x/\w/\t in {4/EOS/Ich, 5/Ich/mag, 6/mag/neuronale, 7/neuronale/N
    \begin{scope}[shift={(\x*3.75,0)}]
        \lstm
        \node[font=\LARGE, text height=1.5ex] at (0.5,-1.5) {\bf\w};
        \node[font=\LARGE, text height=1.5ex, color=nice-blue] at (2.5,4.5
    \end{scope}
}

\node[font=\Huge, color=nice-red] at (16.5,1.5) {$\mathbf{v}$};
```



2.4 Problem - Training RNNs is Hard

- Vanashing and Exploding gradients

Why? - Multiply the same matrix at each time step during forward propagation. The gradient might either tend to 0 (**vanish**) or be too large (**explode**).

Several solution in the literature:

- Bound the gradient to a threshold (*Gradient Clipping*) [Pascanu et al. 2013]
- Use $\mathrm{ReLU}(x) = \max(0, x)$ as a non-linearity instead of $\mathrm{sigmoid}(x)$ [Glorot et al. 2011].
- Clever initialization of parameters ($\mathbf{W}^h = \mathbf{I}$) [Socher et al. 2013, Le et al. 2015].
- Use different recurrent network architectures that favour backpropagation (such as Long Short-Term Memory networks [Hochreiter et al. 1997]).

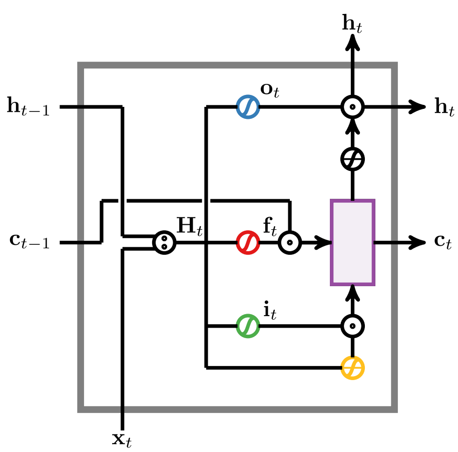
2.4.1 Long Short-Term Memory (LSTM)

[Hochreiter and Schmidhuber, 1997]

```
In [25]: %%tikz -l arrows -s 1000,400 -sc 0.65 -f svg

         \newcommand{\lstm}{
         \definecolor{nice-red}{HTML}{E41A1C}
         \definecolor{nice-orange}{HTML}{FF7F00}
         \definecolor{nice-yellow}{HTML}{FFC020}
         \definecolor{nice-green}{HTML}{4DAF4A}
         \definecolor{nice-blue}{HTML}{377EB8}
         \definecolor{nice-purple}{HTML}{984EA3}

         %lstm first step

         %lstm module box
         \draw[line width=3pt, color=black!50] (-6,-3) rectangle (1.5,5.25);
         \draw[ultra thick] (0,0) rectangle (1,2);

         %memory ct
         \draw[ultra thick, color=nice-purple, fill=nice-purple!10] (0,0) rectangle

         %non-linearities
         \foreach \w/\h/\color in {-2/4.25/nice-blue,-2/1/nice-red,-2/-1/nice-green
             \begin{scope}[shift={(\w,\h)},scale=0.5]
                 \draw[ultra thick, yshift=-0.5cm, color=\color] plot [domain=-0.3:
                 \draw[ultra thick, color=\color] (0,0) circle (0.5cm);
             \end{scope}
         }

         %tanh
         \draw[thick, color=black] (0.25,3) -- (0.75,3);
         \draw[thick, color=nice-yellow] (0.25,-2) -- (0.75,-2);


         %component-wise multiplications
         \foreach \w/\h in {-1/1,0.5/-1,0.5/4.25} {
             \begin{scope}[shift={(\w,\h)},scale=0.5]
                 \draw[ultra thick, color=black] (0,0) circle (0.05cm);
                 \draw[ultra thick, color=black] (0,0) circle (0.5cm);
             \end{scope}
         }

         %vector concat
         \begin{scope}[shift={(-4,1)},scale=0.5]
             \draw[ultra thick,yshift=0.2cm] (0,0) circle (0.05cm);
             \draw[ultra thick,yshift=-0.2cm] (0,0) circle (0.05cm);
             \draw[ultra thick] (0,0) circle (0.5cm);
```

```
\end{scope}


\foreach \fx/\fy/\tx/\ty in {
    -5/-3.5/-5/0.85, %xt
    -5/0.85/-4.2/0.85,
    -6.5/4.25/-5/4.25, %ht1
    -5/4.25/-5/1.15,
    -5/1.15/-4.2/1.15,
    -3.75/1/-3/1, %H
    -3/4.25/-3/-2,
    -3/-2/0.25/-2, %i
    0.5/-1.75/0.5/-1.25,
    -3/-1/-2.25/-1, %it
    -1.75/-1/0.25/-1,
    -3/1/-2.25/1, %ft
    -1.75/1/-1.25/1,
    -0.75/1/0/1,
    -3/4.25/-2.25/4.25, %ot
    -1.75/4.25/0.25/4.25,
    0.5/2/0.5/2.75, %ct
    -5.5/2/-5.1/2, %ct1
    -5.5/2/-5.5/1,
    -6.5/1/-5.5/1,
    -4.9/2/-3.1/2,
    -2.9/2/-1/2,
    -1/2/-1/1.25
} {
    \draw[ultra thick] (\fx,\fy) -- (\tx,\ty);
}

\foreach \fx/\fy/\tx/\ty in {
    0.5/-0.75/0.5/0, %it
    -0.75/1/0/1, %ft
    1/1/2.25/1,
    0.5/3.25/0.5/4,
    0.75/4.25/2.25/4.25, %ht
    0.5/4.5/0.5/6
} {
    \draw[->, >=stealth', ultra thick] (\fx,\fy) -- (\tx,\ty);
}
}

%\begin{scope}[scale=0.8]
%\foreach \d in {0,1} {
%\foreach \t in {0,1,2,3,4} {
%\begin{scope}[shift={(\t*8.5+\d*5.5,\d*9.5)}]
%    \lstm
```

```
%\end{scope}
%}
%}
%\end{scope}


\lstm

%annotations
\node[] at (-5,-3.75) {$\mathbf{x}_t$};
\node[anchor=east] at (-6.5,4.25) {$\mathbf{h}_{t-1}$};
\node[anchor=east] at (-6.5,1) {$\mathbf{c}_{t-1}$};
\node[] at (0.5,6.25) {$\mathbf{h}_t$};
\node[anchor=west] at (2.25,4.25) {$\mathbf{h}_t$};
\node[anchor=west] at (2.25,1) {$\mathbf{c}_t$};
\node[xshift=0.4cm,yshift=0.25cm] at (-4,1) {$\mathbf{H}_t$};
\node[xshift=0.35cm,yshift=0.25cm] at (-2,-1) {$\mathbf{i}_t$};
\node[xshift=0.35cm,yshift=0.25cm] at (-2,1) {$\mathbf{f}_t$};
\node[xshift=0.35cm,yshift=0.25cm] at (-2,4.25) {$\mathbf{o}_t$};

%dummy node for left alignment
\node[] at (17,0) {};
```



$$\mathbf{H}_t = \begin{bmatrix} \mathbf{x}_t \\ \mathbf{h}_{t-1} \end{bmatrix} \tag{1}$$

$$\mathbf{i}_t = \sigma(\mathbf{W}^i\mathbf{H} + \mathbf{b}^i) \tag{2}$$

$$\mathbf{f}_t = \sigma(\mathbf{W}^f\mathbf{H} + \mathbf{b}^f) \tag{3}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}^o\mathbf{H} + \mathbf{b}^o) \tag{4}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tanh(\mathbf{W}^c\mathbf{H} + \mathbf{b}^c) \tag{5}$$

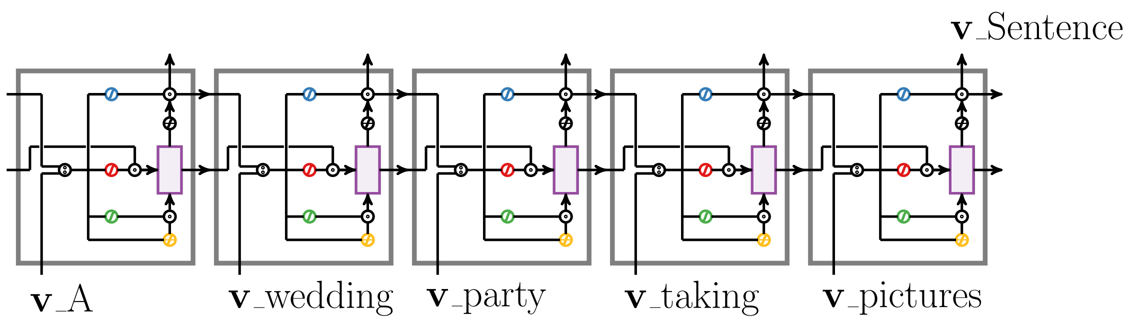
$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \tag{6}$$

Sentence Encoding

```
In [26]: %%tikz -l arrows -s 1000,400 -sc 0.65 -f svg

         \newcommand{\lstm}{
         \definecolor{nice-red}{HTML}{E41A1C}
         \definecolor{nice-orange}{HTML}{FF7F00}
         \definecolor{nice-yellow}{HTML}{FFC020}
         \definecolor{nice-green}{HTML}{4DAF4A}
         \definecolor{nice-blue}{HTML}{377EB8}
         \definecolor{nice-purple}{HTML}{984EA3}

         %lstm first step

         %lstm module box
         \draw[line width=3pt, color=black!50] (-6,-3) rectangle (1.5,5.25);
         \draw[ultra thick] (0,0) rectangle (1,2);

         %memory ct
         \draw[ultra thick, color=nice-purple, fill=nice-purple!10] (0,0) rectangle

         %non-linearities
         \foreach \w/\h/\color in {-2/4.25/nice-blue,-2/1/nice-red,-2/-1/nice-green
             \begin{scope}[shift={(\w,\h)},scale=0.5]
                 \draw[ultra thick, yshift=-0.5cm, color=\color] plot [domain=-0.3:
                 \draw[ultra thick, color=\color] (0,0) circle (0.5cm);
             \end{scope}
         }

         %tanh
         \draw[thick, color=black] (0.25,3) -- (0.75,3);
         \draw[thick, color=nice-yellow] (0.25,-2) -- (0.75,-2);


         %component-wise multiplications
         \foreach \w/\h in {-1/1,0.5/-1,0.5/4.25} {
             \begin{scope}[shift={(\w,\h)},scale=0.5]
                 \draw[ultra thick, color=black] (0,0) circle (0.05cm);
                 \draw[ultra thick, color=black] (0,0) circle (0.5cm);
             \end{scope}
         }

         %vector concat
         \begin{scope}[shift={(-4,1)},scale=0.5]
             \draw[ultra thick,yshift=0.2cm] (0,0) circle (0.05cm);
             \draw[ultra thick,yshift=-0.2cm] (0,0) circle (0.05cm);
             \draw[ultra thick] (0,0) circle (0.5cm);
         \end{scope}
```

```
\foreach \fx/\fy/\tx/\ty in {
    -5/-3.5/-5/0.85, %xt
    -5/0.85/-4.2/0.85,
    -6.5/4.25/-5/4.25, %ht1
    -5/4.25/-5/1.15,
    -5/1.15/-4.2/1.15,
    -3.75/1/-3/1, %H
    -3/4.25/-3/-2,
    -3/-2/0.25/-2, %i
    0.5/-1.75/0.5/-1.25,
    -3/-1/-2.25/-1, %it
    -1.75/-1/0.25/-1,
    -3/1/-2.25/1, %ft
    -1.75/1/-1.25/1,
    -0.75/1/0/1,
    -3/4.25/-2.25/4.25, %ot
    -1.75/4.25/0.25/4.25,
    0.5/2/0.5/2.75, %ct
    -5.5/2/-5.1/2, %ct1
    -5.5/2/-5.5/1,
    -6.5/1/-5.5/1,
    -4.9/2/-3.1/2,
    -2.9/2/-1/2,
    -1/2/-1/1.25
} {
    \draw[ultra thick] (\fx,\fy) -- (\tx,\ty);
}

\foreach \fx/\fy/\tx/\ty in {
    0.5/-0.75/0.5/0, %it
    -0.75/1/0/1, %ft
    1/1/2.25/1,
    0.5/3.25/0.5/4,
    0.75/4.25/2.25/4.25, %ht
    0.5/4.5/0.5/6
} {
    \draw[->, >=stealth', ultra thick] (\fx,\fy) -- (\tx,\ty);
}
}

\begin{scope}[scale=0.8]
\foreach \d in {0} {
\foreach \t/\word in {0/A,1/wedding,2/party,3/taking,4/pictures} {
    \node[font=\Huge, anchor=west] at (\t*8.5-5.75,-4.5) {$\mathbf{v}$\_\w
    \begin{scope}[shift={(\t*8.5+\d*5.5,\d*9.5)}]
        \lstm
```

```
    \end{scope}
}
}
\end{scope}

\node[font=\Huge, anchor=west] at (27,5.75) {$\mathbf{v}$\_Sentence};


%dummy node for left alignment
\node[] at (17,0) {};
```

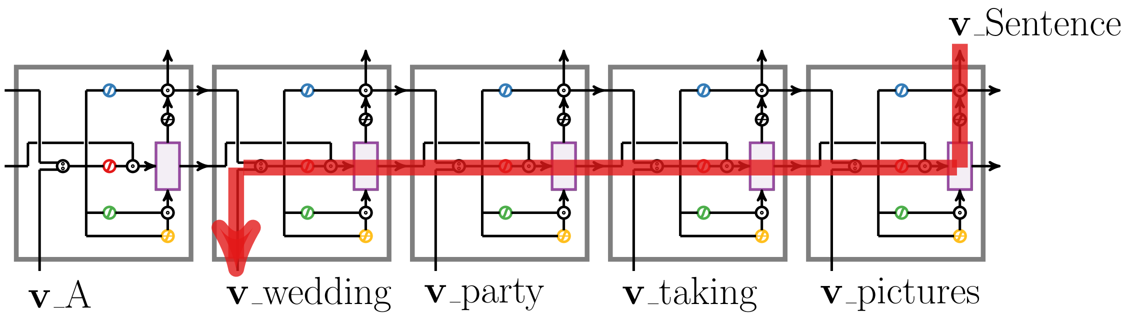


Gating

In [27]:
```
%%tikz -l arrows -s 1000,400 -sc 0.65 -f svg

\definecolor{nice-red}{HTML}{E41A1C}
\definecolor{nice-orange}{HTML}{FF7F00}
\definecolor{nice-yellow}{HTML}{FFC020}
\definecolor{nice-green}{HTML}{4DAF4A}
\definecolor{nice-blue}{HTML}{377EB8}
\definecolor{nice-purple}{HTML}{984EA3}

\newcommand{\lstm}{
%lstm first step

%lstm module box
\draw[line width=3pt, color=black!50] (-6,-3) rectangle (1.5,5.25);
\draw[ultra thick] (0,0) rectangle (1,2);

%memory ct
\draw[ultra thick, color=nice-purple, fill=nice-purple!10] (0,0) rectangle
```

```
%non-linearities
\foreach \w/\h/\color in {-2/4.25/nice-blue,-2/1/nice-red,-2/-1/nice-green
    \begin{scope}[shift={(\w,\h)},scale=0.5]
        \draw[ultra thick, yshift=-0.5cm, color=\color] plot [domain=-0.3:
        \draw[ultra thick, color=\color] (0,0) circle (0.5cm);
    \end{scope}
}

%tanh
\draw[thick, color=black] (0.25,3) -- (0.75,3);
\draw[thick, color=nice-yellow] (0.25,-2) -- (0.75,-2);


%component-wise multiplications
\foreach \w/\h in {-1/1,0.5/-1,0.5/4.25} {
    \begin{scope}[shift={(\w,\h)},scale=0.5]
        \draw[ultra thick, color=black] (0,0) circle (0.05cm);
        \draw[ultra thick, color=black] (0,0) circle (0.5cm);
    \end{scope}
}

%vector concat
\begin{scope}[shift={(-4,1)},scale=0.5]
    \draw[ultra thick,yshift=0.2cm] (0,0) circle (0.05cm);
    \draw[ultra thick,yshift=-0.2cm] (0,0) circle (0.05cm);
    \draw[ultra thick] (0,0) circle (0.5cm);
\end{scope}


\foreach \fx/\fy/\tx/\ty in {
    -5/-3.5/-5/0.85, %xt
    -5/0.85/-4.2/0.85,
    -6.5/4.25/-5/4.25, %ht1
    -5/4.25/-5/1.15,
    -5/1.15/-4.2/1.15,
    -3.75/1/-3/1, %H
    -3/4.25/-3/-2,
    -3/-2/0.25/-2, %i
    0.5/-1.75/0.5/-1.25,
    -3/-1/-2.25/-1, %it
    -1.75/-1/0.25/-1,
    -3/1/-2.25/1, %ft
    -1.75/1/-1.25/1,
    -0.75/1/0/1,
    -3/4.25/-2.25/4.25, %ot
    -1.75/4.25/0.25/4.25,
    0.5/2/0.5/2.75, %ct
    -5.5/2/-5.1/2, %ct1
```

```
    -5.5/2/-5.5/1,
    -6.5/1/-5.5/1,
    -4.9/2/-3.1/2,
    -2.9/2/-1/2,
    -1/2/-1/1.25
} {
    \draw[ultra thick] (\fx,\fy) -- (\tx,\ty);
}

\foreach \fx/\fy/\tx/\ty in {
    0.5/-0.75/0.5/0, %it
    -0.75/1/0/1, %ft
    1/1/2.25/1,
    0.5/3.25/0.5/4,
    0.75/4.25/2.25/4.25, %ht
    0.5/4.5/0.5/6
} {
    \draw[->, >=stealth', ultra thick] (\fx,\fy) -- (\tx,\ty);
}
}

\begin{scope}[scale=0.8]
\foreach \d in {0} {
\foreach \t/\word in {0/A,1/wedding,2/party,3/taking,4/pictures} {
    \node[font=\Huge, anchor=west] at (\t*8.5-5.75,-4.5) {$\mathbf{v}$\_\w
    \begin{scope}[shift={(\t*8.5+\d*5.5,\d*9.5)}]
        \lstm
    \end{scope}
}
}
\end{scope}

\node[font=\Huge, anchor=west] at (27,5.75) {$\mathbf{v}$\_Sentence};


\draw[line width=10pt, color=nice-red, opacity=0.8] (27.6,5) -- (27.6,0.75
\draw[line width=10pt, color=nice-red, opacity=0.8] (27.5,0.75) -- (3,0.75
\draw[->, >=stealth', line width=10pt, color=nice-red, opacity=0.8] (2.75,


%dummy node for left alignment
\node[] at (17,0) {};
```




```
In [28]: %%html

         <center>
         <video controls autoplay loop>
         <source src="images/vanishing.mp4" type="video/mp4">
         </video>
         </center>
```

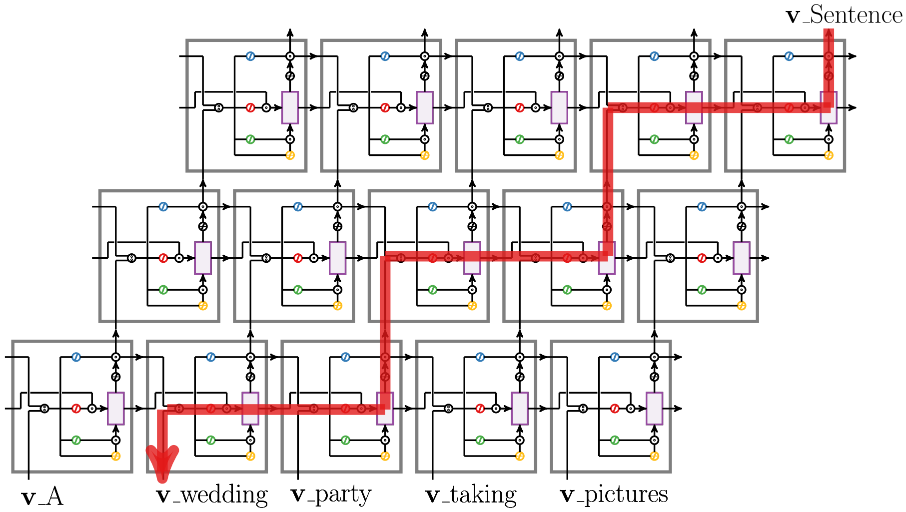
```
<IPython.core.display.HTML object>
```

Stacking

```
In [29]: %%tikz -l arrows -s 1100,500 -sc 0.65 -f svg

         \definecolor{nice-red}{HTML}{E41A1C}
         \definecolor{nice-orange}{HTML}{FF7F00}
         \definecolor{nice-yellow}{HTML}{FFC020}
         \definecolor{nice-green}{HTML}{4DAF4A}
         \definecolor{nice-blue}{HTML}{377EB8}
         \definecolor{nice-purple}{HTML}{984EA3}

         \newcommand{\lstm}{
         %lstm first step

         %lstm module box
         \draw[line width=3pt, color=black!50] (-6,-3) rectangle (1.5,5.25);
         \draw[ultra thick] (0,0) rectangle (1,2);

         %memory ct
         \draw[ultra thick, color=nice-purple, fill=nice-purple!10] (0,0) rectangle
```

```
%non-linearities
\foreach \w/\h/\color in {-2/4.25/nice-blue,-2/1/nice-red,-2/-1/nice-green
    \begin{scope}[shift={(\w,\h)},scale=0.5]
        \draw[ultra thick, yshift=-0.5cm, color=\color] plot [domain=-0.3:
        \draw[ultra thick, color=\color] (0,0) circle (0.5cm);
    \end{scope}
}

%tanh
\draw[thick, color=black] (0.25,3) -- (0.75,3);
\draw[thick, color=nice-yellow] (0.25,-2) -- (0.75,-2);


%component-wise multiplications
\foreach \w/\h in {-1/1,0.5/-1,0.5/4.25} {
    \begin{scope}[shift={(\w,\h)},scale=0.5]
        \draw[ultra thick, color=black] (0,0) circle (0.05cm);
        \draw[ultra thick, color=black] (0,0) circle (0.5cm);
    \end{scope}
}

%vector concat
\begin{scope}[shift={(-4,1)},scale=0.5]
    \draw[ultra thick,yshift=0.2cm] (0,0) circle (0.05cm);
    \draw[ultra thick,yshift=-0.2cm] (0,0) circle (0.05cm);
    \draw[ultra thick] (0,0) circle (0.5cm);
\end{scope}


\foreach \fx/\fy/\tx/\ty in {
    -5/-3.5/-5/0.85, %xt
    -5/0.85/-4.2/0.85,
    -6.5/4.25/-5/4.25, %ht1
    -5/4.25/-5/1.15,
    -5/1.15/-4.2/1.15,
    -3.75/1/-3/1, %H
    -3/4.25/-3/-2,
    -3/-2/0.25/-2, %i
    0.5/-1.75/0.5/-1.25,
    -3/-1/-2.25/-1, %it
    -1.75/-1/0.25/-1,
    -3/1/-2.25/1, %ft
    -1.75/1/-1.25/1,
    -0.75/1/0/1,
    -3/4.25/-2.25/4.25, %ot
    -1.75/4.25/0.25/4.25,
    0.5/2/0.5/2.75, %ct
    -5.5/2/-5.1/2, %ct1
```

```
    -5.5/2/-5.5/1,
    -6.5/1/-5.5/1,
    -4.9/2/-3.1/2,
    -2.9/2/-1/2,
    -1/2/-1/1.25
} {
    \draw[ultra thick] (\fx,\fy) -- (\tx,\ty);
}

\foreach \fx/\fy/\tx/\ty in {
    0.5/-0.75/0.5/0, %it
    -0.75/1/0/1, %ft
    1/1/2.25/1,
    0.5/3.25/0.5/4,
    0.75/4.25/2.25/4.25, %ht
    0.5/4.5/0.5/6
} {
    \draw[->, >=stealth', ultra thick] (\fx,\fy) -- (\tx,\ty);
}
}

\begin{scope}[scale=0.8]
\foreach \d in {0,1,2} {
\foreach \t/\word in {0/A,1/wedding,2/party,3/taking,4/pictures} {
    \node[font=\Huge, anchor=west] at (\t*8.5-5.75,-4.5) {$\mathbf{v}$\_\w
    \begin{scope}[shift={(\t*8.5+\d*5.5,\d*9.5)}]
        \lstm
    \end{scope}
}
}
\end{scope}

\node[font=\Huge, anchor=west] at (34,20.75) {$\mathbf{v}$\_Sentence};

\draw[line width=10pt, color=nice-red, opacity=0.8] (36.4,16) -- (36.4,20)
\draw[line width=10pt, color=nice-red, opacity=0.8] (25.25,16) -- (36.5,16
\draw[line width=10pt, color=nice-red, opacity=0.8] (25.25,8.5) -- (25.25,
\draw[line width=10pt, color=nice-red, opacity=0.8] (14,8.5) -- (25.25,8.5
\draw[line width=10pt, color=nice-red, opacity=0.8] (14,8.5) -- (14,0.75);
\draw[line width=10pt, color=nice-red, opacity=0.8] (14,0.75) -- (3,0.75);
\draw[->, >=stealth', line width=10pt, color=nice-red, opacity=0.8] (2.75,


%dummy node for left alignment
\node[] at (17,0) {};
```



2.5 Applications

- Language Modeling
- Machine Translation
- Question Answering
- Dialog Modeling
- Language Generation
- Sentence Summarization
- Paraphrasing
- Sentiment Analysis
- Recognizing Textual Entailment
- ...

2.6 Learning to Execute

[Zaremba and Sutskever, 2014]

```
In [ ]:
```