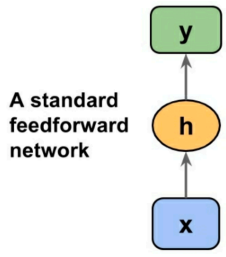
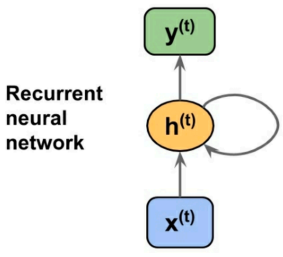
Introduction of LSTM

Neural network & RNN

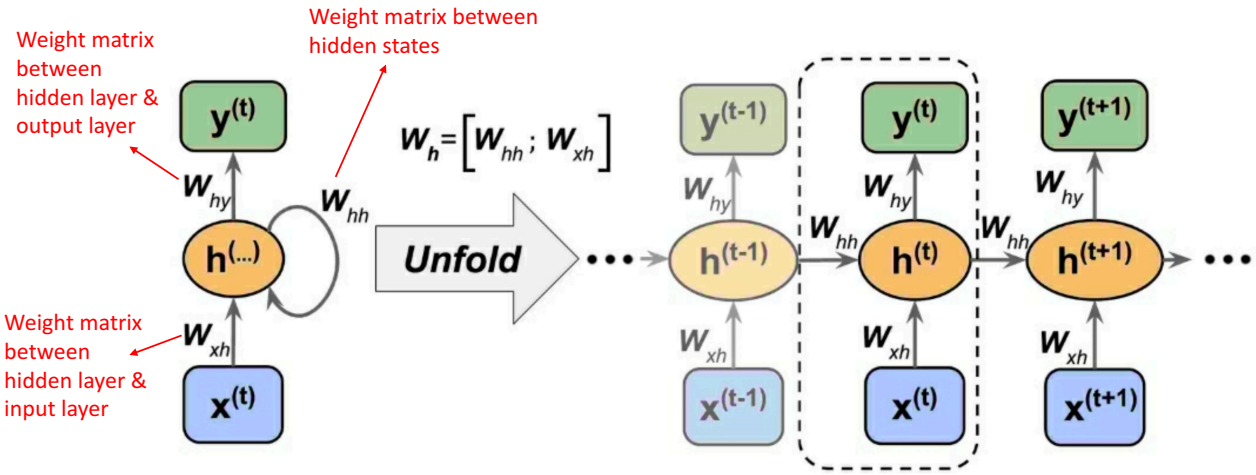
- IID data
- Neural network



- Time dependent data
- RNN



Unfold of RNN



Activation of hidden units at time t: $\boldsymbol{h}^{(t)} = \phi_h\left(\boldsymbol{z}_h^{(t)}\right) = \phi_h\left(\boldsymbol{W}_{xh}\boldsymbol{x}^{(t)} + \boldsymbol{W}_{hh}\boldsymbol{h}^{(t-1)} + \boldsymbol{b}_h\right)$

Activation of output units at time t: $\boldsymbol{y}^{(t)} = \phi_y\left(\boldsymbol{W}_{hy}\boldsymbol{h}^{(t)} + \boldsymbol{b}_y\right)$

Training RNNs

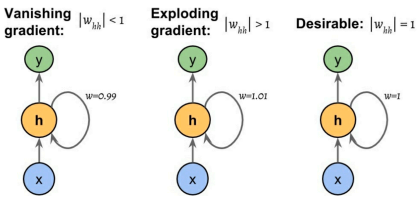
Overall loss function: $L = \sum_{t=1}^{T} L^{(t)}$ Sum of all loss functions at times t=1 to t=T

Gradient of L w.r.t $\boldsymbol{W_{hh}}$: $\frac{\partial L^{(t)}}{\partial \boldsymbol{W}_{hh}} = \frac{\partial L^{(t)}}{\partial \boldsymbol{y}^{(t)}} \times \frac{\partial \boldsymbol{y}^{(t)}}{\partial \boldsymbol{h}^{(t)}} \times \left(\sum_{k=1}^{t} \frac{\partial \boldsymbol{h}^{(t)}}{\partial \boldsymbol{h}^{(k)}} \times \frac{\partial \boldsymbol{h}^{(k)}}{\partial \boldsymbol{W}_{hh}} \right)$

Recall: activation of hidden states $\boldsymbol{h}^{(t)} = \phi_h\left(\boldsymbol{z}_h^{(t)}\right) = \phi_h\left(\boldsymbol{W}_{xh}\boldsymbol{x}^{(t)} + \boldsymbol{W}_{hh}\boldsymbol{h}^{(t-1)} + \boldsymbol{b}_h\right)$

$\frac{\partial \boldsymbol{h}^{(t)}}{\partial \boldsymbol{h}^{(k)}} = \prod_{i=k+1}^{t} \frac{\partial \boldsymbol{h}^{(i)}}{\partial \boldsymbol{h}^{(i-1)}}$ Recurrent: multiplication of adjacent steps

Vanishing or exploding problems:



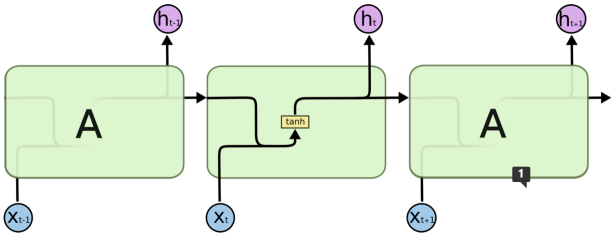
Solutions:

- Truncated backpropagation through time (TBPTT)
- Long short-term memory (LSTM)

Structure of LSTM

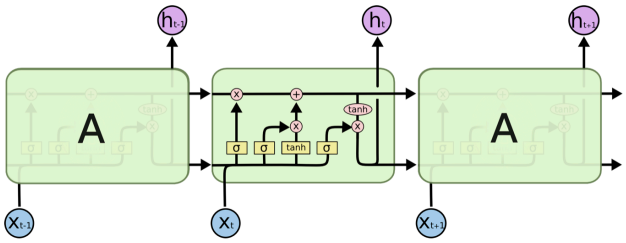
RNN

$$\boldsymbol{h}^{(t)} = \phi_h\left(\boldsymbol{z}_h^{(t)}\right) = \phi_h\left(\boldsymbol{W}_{xh}\boldsymbol{x}^{(t)} + \boldsymbol{W}_{hh}\boldsymbol{h}^{(t-1)} + \boldsymbol{b}_h\right)$$
$$\boldsymbol{y}^{(t)} = \phi_y\left(\boldsymbol{W}_{hy}\boldsymbol{h}^{(t)} + \boldsymbol{b}_y\right)$$

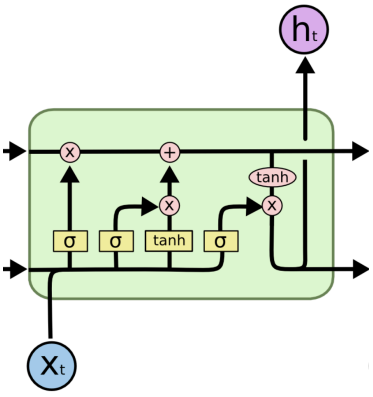


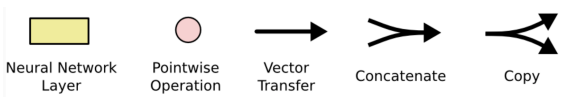
LSTM

- Building block: memory cell
- Recurrent edge with w=1
- Value associated with recurrent edge: cell state

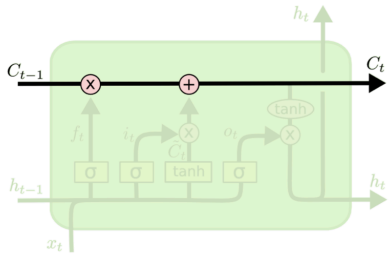


Elements of LSTM

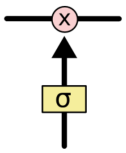




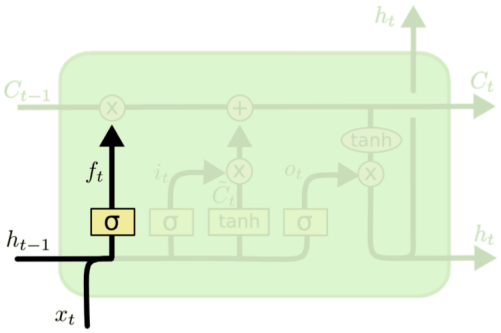
Core: cell state
Can add or remove information regulated by gates



Gate: a way to optionally let information through



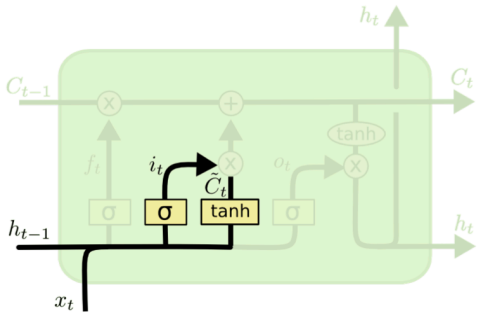
Step-by-step LSTM Walk Through



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Decide what information to throw away from the cell state
Forget gate layer: a sigmoid

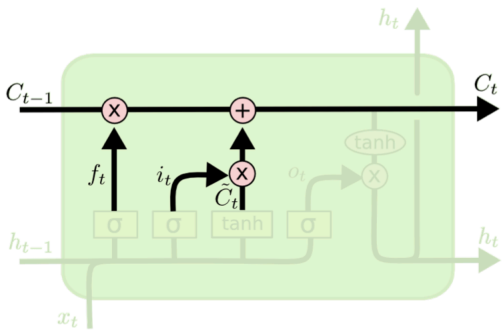
Step-by-step LSTM Walk Through



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

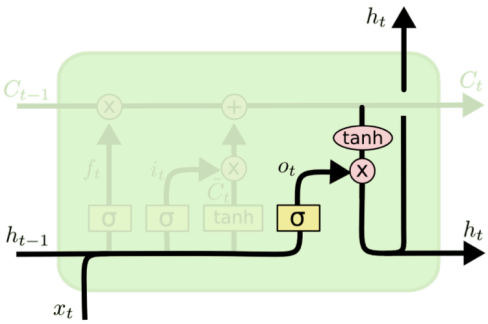
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Step-by-step LSTM Walk Through



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Step-by-step LSTM Walk Through



$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

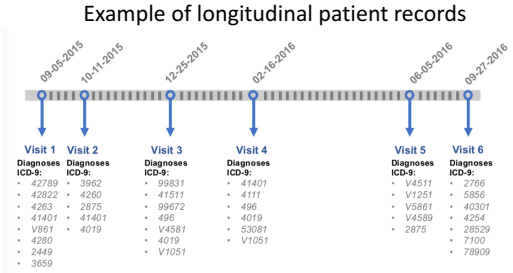
Challenges of using LSTM in medical data

- Time intervals of each visit are not evenly distributed
  - T-LSTM
- Poor interpretability
  - Attention mechanism

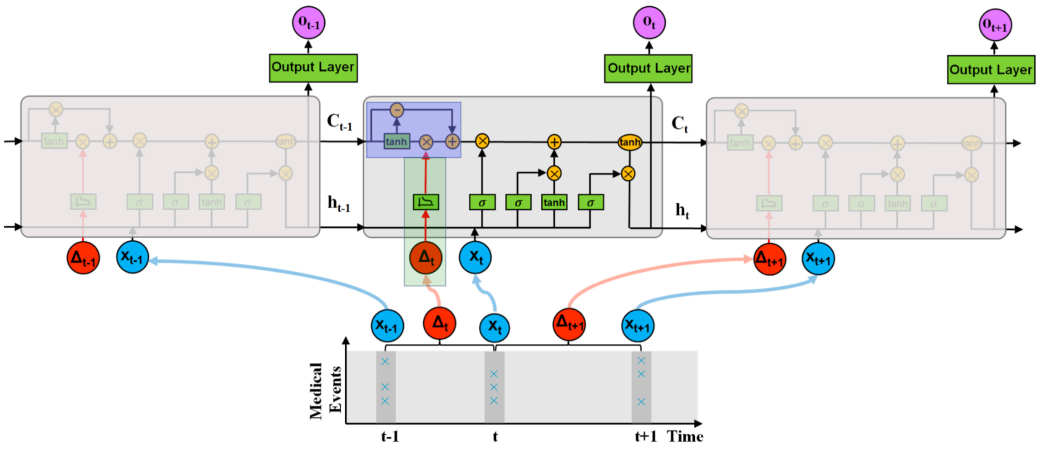
Time-aware LSTM

- Time irregularity is important in clinical data
  - ➢Variation in time gap could be indicative of certain impending disease condition
  - ➢Dependency on the previous memory should not play an active role in prediction of current status if the time interval is too large

Example of longitudinal patient records



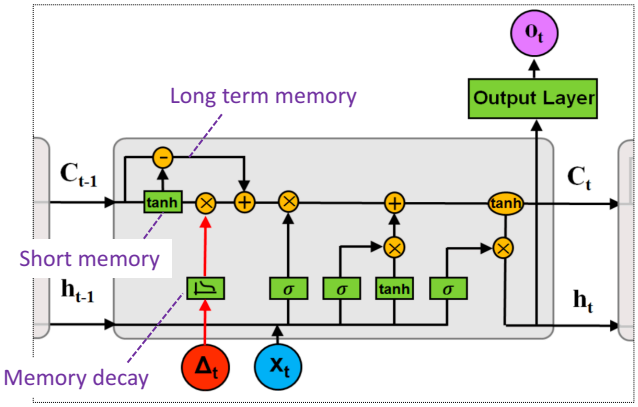
Unit of T-LSTM



Difference compared with standard LSTM:

memory decomposition + decay of short memory

Algorithm of T-LSTM



Short time memory learnt from C_{t-1}

$$C^S_{t-1} = \tanh(W_d C_{t-1} + b_d)$$

Short time memory*decay function

$$\hat{C}^S_{t-1} = C^S_{t-1} * g(\Delta_t)$$

Calculate long-term memory (no decay)

$$C^T_{t-1} = C_{t-1} - C^S_{t-1}$$

Reassessment of C_{t-1}

$$C^*_{t-1} = C^T_{t-1} + \hat{C}^S_{t-1}$$

Calculate Ct

$$C_t = f_t * C^*_{t-1} + i_t * \tilde{C}$$

Trade accuracy for interpretability in traditional models

- Identifying a set of rules
  - *e.g.* via decision trees
- Case-based reasoning by finding similar patients
  - *e.g.* via k-nearest neighbors & distance metric learning
- Identifying a list of risk factors
  - *e.g.* via LASSO coefficients

Attention mechanism

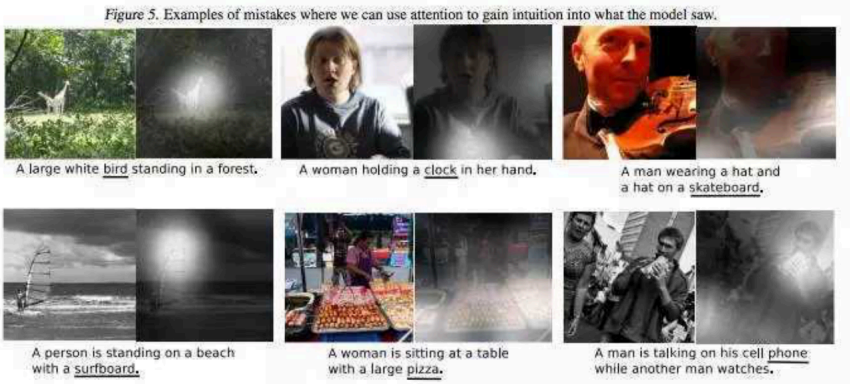
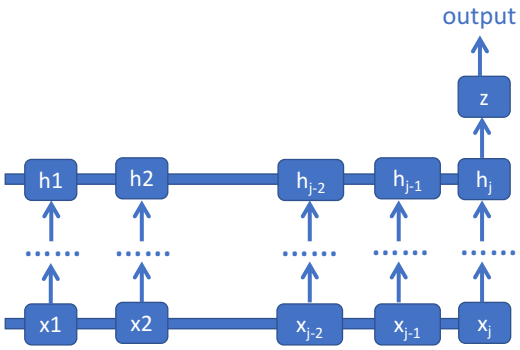


Figure 5. Examples of mistakes where we can use attention to gain intuition into what the model saw.

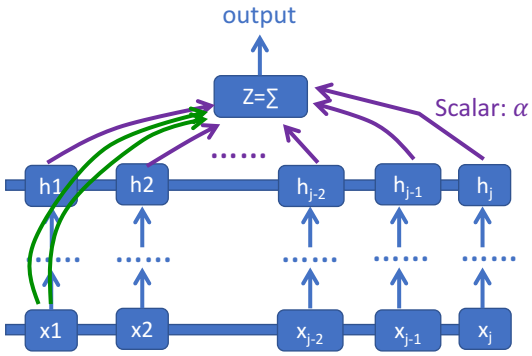
- Mimic working mechanism of human brain
- Used in figure caption, image classification, language translation
- Helps to assign different weights to input

Introducing attention mechanism in LSTM

Standard RNN

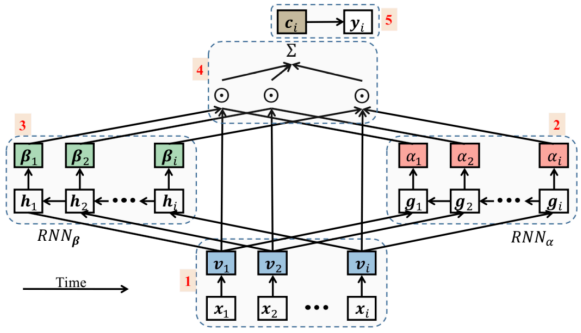


With attention



Scalar: α, importance of each visit
Vector: $\boldsymbol{\beta}$, importance of variables in each visit

RETAIN: reverse time attention model



Embedding:

$$\mathbf{v}_i = \mathbf{W}_{emb}\mathbf{x}_i, \qquad \text{(Step 1)}$$

Generate α & β using two RNNs:

$$\begin{aligned} \mathbf{g}_i, \mathbf{g}_{i-1}, \ldots, \mathbf{g}_1 &= \mathrm{RNN}_\alpha(\mathbf{v}_i, \mathbf{v}_{i-1}, \ldots, \mathbf{v}_1), \\ e_j &= \mathbf{w}_\alpha^\top \mathbf{g}_j + b_\alpha, \quad \text{for} \quad j = 1, \ldots, i \\ \alpha_1, \alpha_2, \ldots, \alpha_i &= \mathrm{Softmax}(e_1, e_2, \ldots, e_i) \qquad \text{(Step 2)} \\ \mathbf{h}_i, \mathbf{h}_{i-1}, \ldots, \mathbf{h}_1 &= \mathrm{RNN}_{\boldsymbol{\beta}}(\mathbf{v}_i, \mathbf{v}_{i-1}, \ldots, \mathbf{v}_1) \\ \boldsymbol{\beta}_j &= \tanh\left(\mathbf{W}_{\boldsymbol{\beta}}\mathbf{h}_j + \mathbf{b}_{\boldsymbol{\beta}}\right) \quad \text{for} \quad j = 1, \ldots, i, \qquad \text{(Step 3)} \end{aligned}$$

Generate context vector using attention and representation vectors:

$$\mathbf{c}_i = \sum_{j=1}^{i} \alpha_j \boldsymbol{\beta}_j \odot \mathbf{v}_j, \qquad \text{(Step 4)}$$

Making prediction:

$$\hat{\mathbf{y}}_i = \mathrm{Softmax}(\mathbf{W}\mathbf{c}_i + \mathbf{b}), \qquad \text{(Step 5)}$$

Interpreting RETAIN:

$$p(\mathbf{y}_i|\mathbf{x}_1,\ldots,\mathbf{x}_i) = p(\mathbf{y}_i|\mathbf{c}_i) = \text{Softmax}\left(\mathbf{W}\mathbf{c}_i + \mathbf{b}\right)$$

$$p(\mathbf{y}_i|\mathbf{x}_1,\ldots,\mathbf{x}_i) = p(\mathbf{y}_i|\mathbf{c}_i) = \text{Softmax}\left(\mathbf{W}\left(\sum_{j=1}^{i} \alpha_j \boldsymbol{\beta}_j \odot \mathbf{v}_j\right) + \mathbf{b}\right)$$

$$p(\mathbf{y}_i|\mathbf{x}_1,\ldots,\mathbf{x}_i) = \text{Softmax}\left(\mathbf{W}\left(\sum_{j=1}^{i} \alpha_j \boldsymbol{\beta}_j \odot \sum_{k=1}^{r} x_{j,k}\mathbf{W}_{emb}[:,k]\right) + \mathbf{b}\right)$$

$$= \text{Softmax}\left(\sum_{j=1}^{i}\sum_{k=1}^{r} x_{j,k}\,\alpha_j \mathbf{W}\left(\boldsymbol{\beta}_j \odot \mathbf{W}_{emb}[:,k]\right) + \mathbf{b}\right)$$

$$\omega(\mathbf{y}_i, x_{j,k}) = \underbrace{\alpha_j \mathbf{W}(\boldsymbol{\beta}_j \odot \mathbf{W}_{emb}[:,k])}_{\text{Contribution coefficient}} \underbrace{x_{j,k}}_{\text{Input value}},$$

references

- Background LSTM: http://colah.github.io/posts/2015-08-Understanding-LSTMs/
- T-LSTM:
  - Original paper: http://biometrics.cse.msu.edu/Publications/MachineLearning/Baytasetal_PatientSubtypingViaTimeAwareLSTMNetworks.pdf
  - Github: https://github.com/illidanlab/T-LSTM
- RETAIN:
  - Original paper: https://arxiv.org/abs/1608.05745
  - Github: https://github.com/mp2893/retain