Show, Attend and Tell: Neural Image Caption Generation with Visual Attention

Encoder-Decoder FrameWork,

RNN,

LSTM,

Soft Attention Model

Travel Time Prediction Review

- Native
- Parametric models:Kalman filter/ARIMA
- Nearest neighborbased approach:k-NN
- Neural network methods:LSTM

Discovering the Behavior Switching

Deep learning is an important concept in the learning theory at present.

Machine Translation

深度学习是目前学习理论中的一个重要概念。

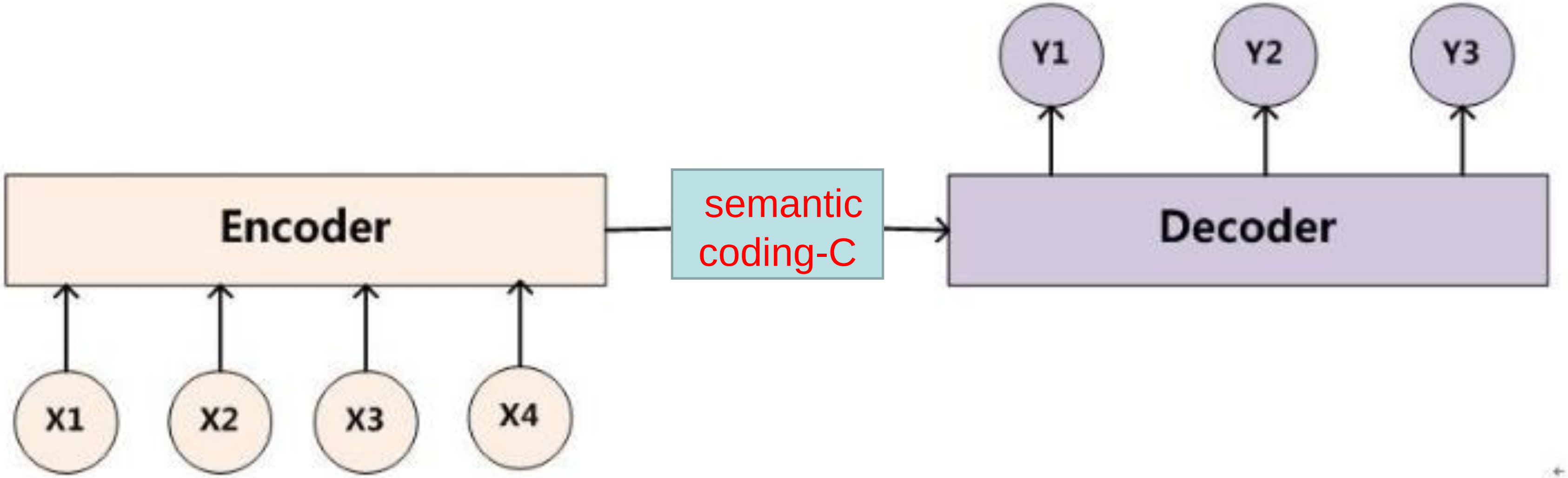
Speech Recognition

speech recognition

Image Captioning

A little boy is looking at you.

Encoder-Decoder FrameWork



$X = (x_1, x_2 \ldots x_m)$

$Y = (y_1, y_2 \ldots y_n)$

$$C = \mathcal{F}(x_1, x_2 \ldots x_m)$$

$$y_i = \mathcal{G}(C, y_1, y_2 \ldots y_{i-1})$$

Variant Combinations

- Encoder:CNN/RNN/BiRNN/GRU/LSTM/...

- Decoder:CNN/RNN/BiRNN/GRU/LSTM/...

- Different applications may have variant combinations:

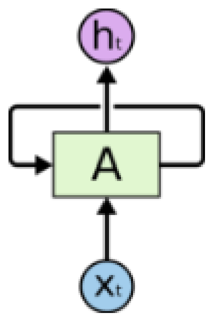
  LSTM+LSTM(Machine Translation)

  CNN+LSTM(Image Captioning)

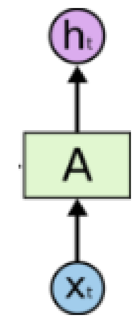
Human's Thought

- As you read this ppt, you understand each page based on your understanding of previous pages.
- You don't throw everything away and start thinking from scratch again.
- Your thoughts have persistence.

Recurrent VS. Traditional neural networks



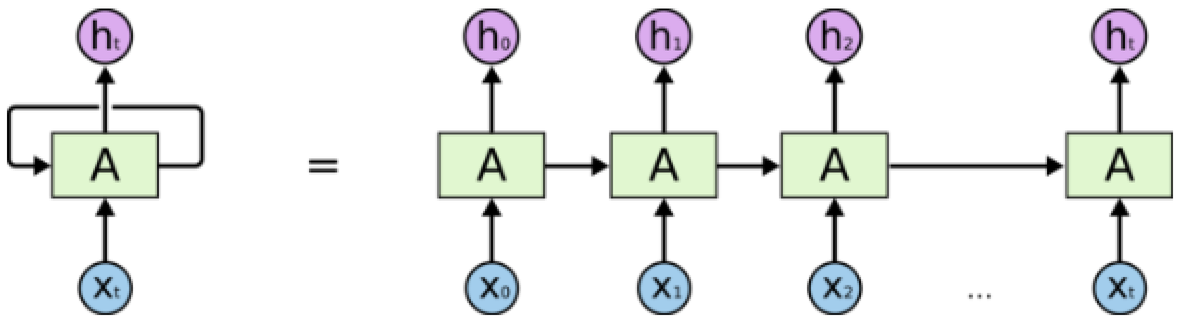
Recurrent
Neural Networks



Traditional
Neural Networks

- Traditional neural networks are transient when processing continuous data.
- RNN are networks with loops in them, allowing information to persist.

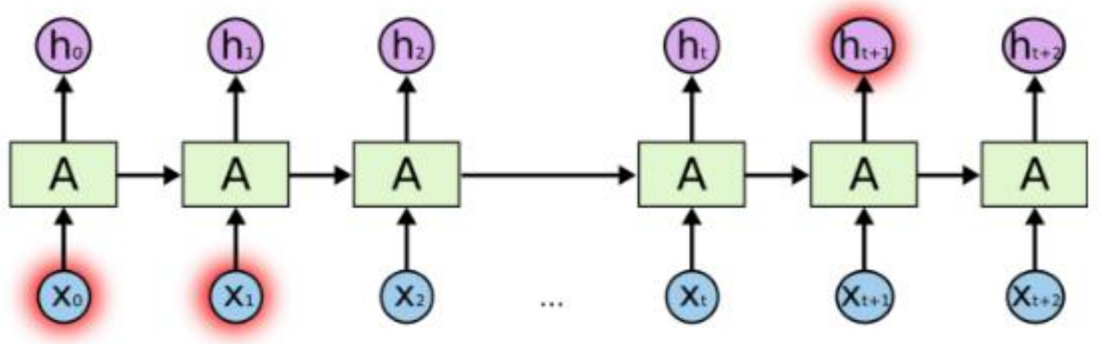
Unfold RNNs



An unrolled recurrent neural network.

- This chain-like nature reveals that recurrent neural networks are intimately related to sequences and lists.
- They're the natural architecture of neural network to use for such data.

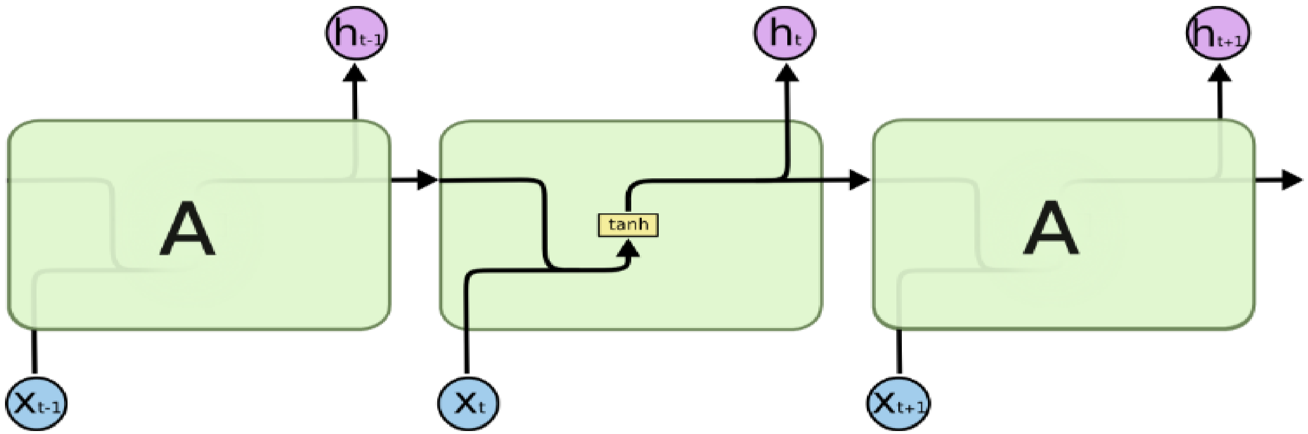
The Problem of Long-Term Dependencies



- It's entirely possible for the gap between the relevant information and the point where it is needed to become very large.

- As that gap grows, RNNs become unable to learn to connect the information(gradient exploding/vanishing).

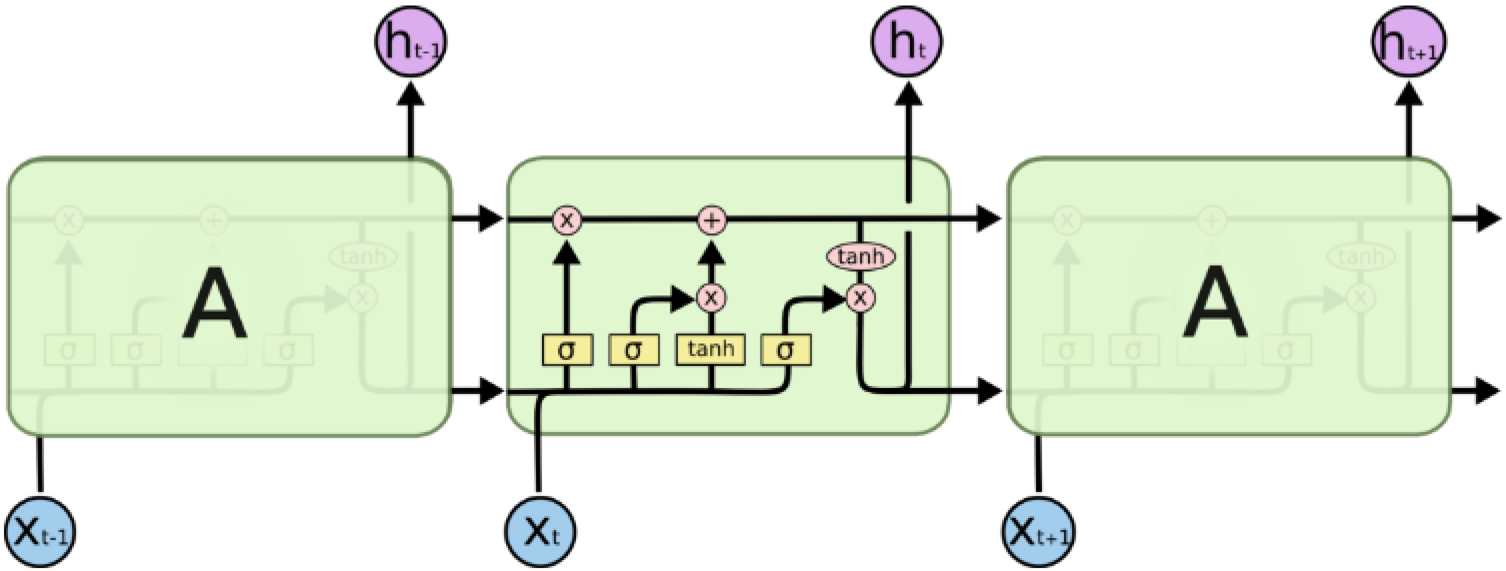
Standard RNNs



The repeating module in a standard RNN contains a single layer.

- All recurrent neural networks have the form of a chain of repeating modules of neural network.
- In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer.

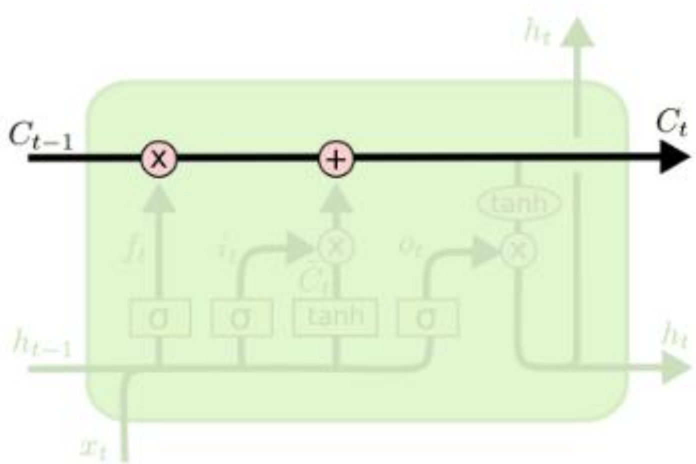
The repeating module of LSTMs

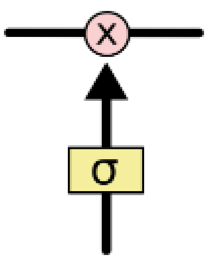


The repeating module in an LSTM contains four interacting layers.

- Instead of having a single neural network layer, there are four, interacting in a very special way.

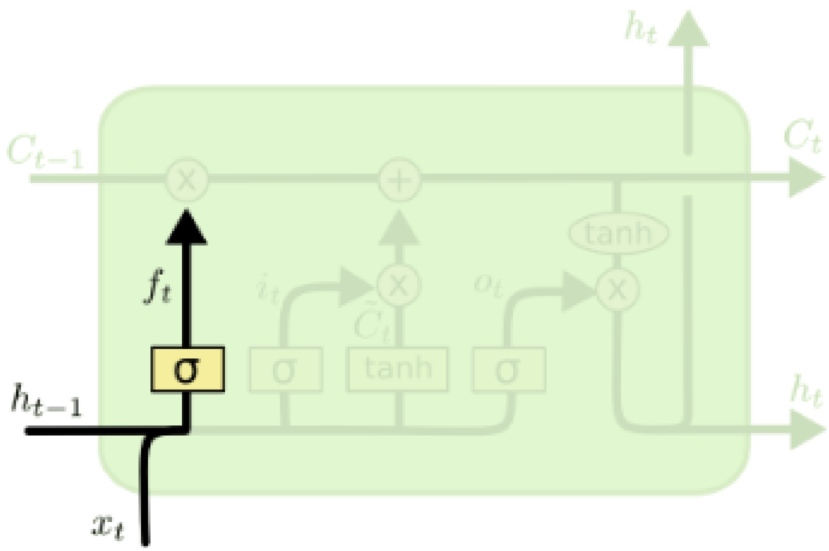
Cell State and Gates





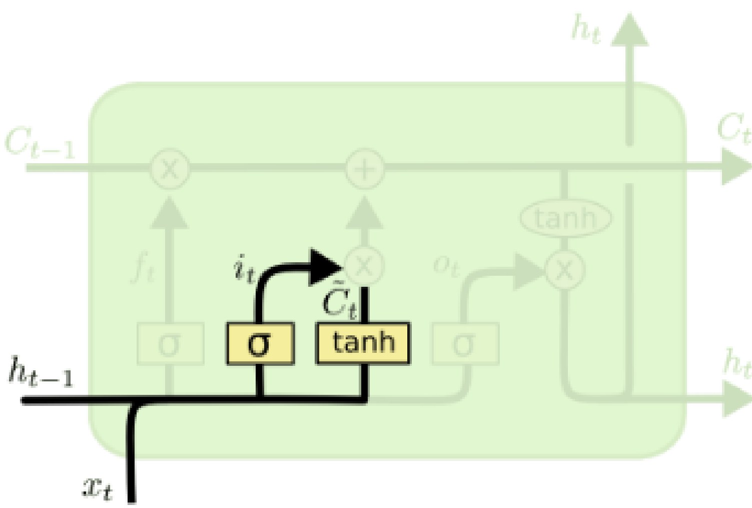
- The LSTM does have the ability to remove or add information to the cell state, carefully regulated by structures called gates.

Throw:Forget Gate Layer



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

Store:Input Gate Layer and tanh Layer



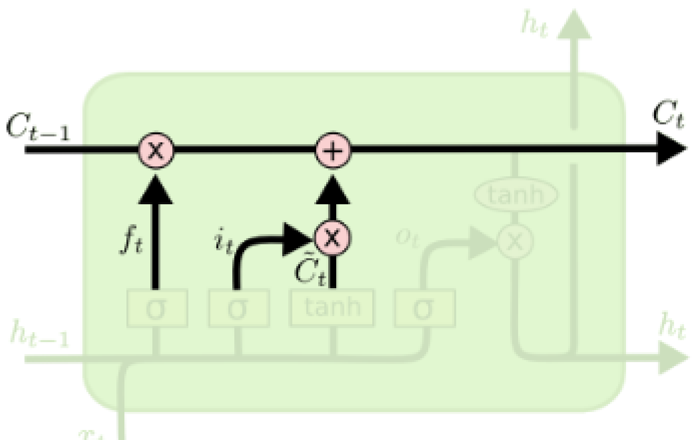
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$

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

- First, a sigmoid layer called the "input gate layer" decides which values we'll update.

- Next, a tanh layer creates a vector of new candidate values, C_t, that could be added to the state

Update Your Memory

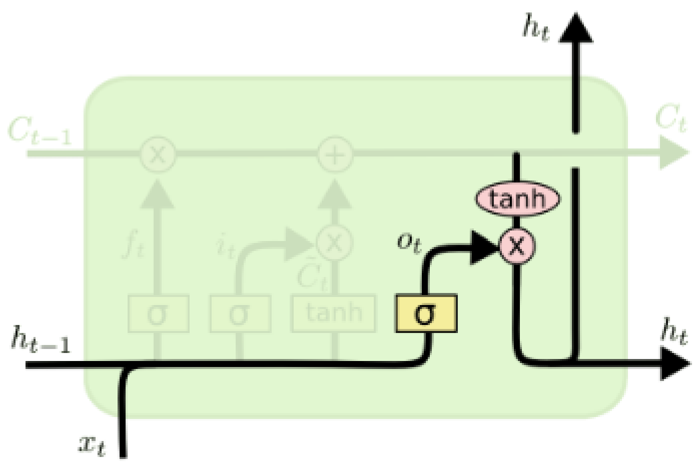


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

- We multiply the old state by f_t, forgetting the things we decided to forget earlier.

- Then we add $i_t * C_t$. This is the new candidate values, scaled by how much we decided to update each state value.

Output:Based On Cell State



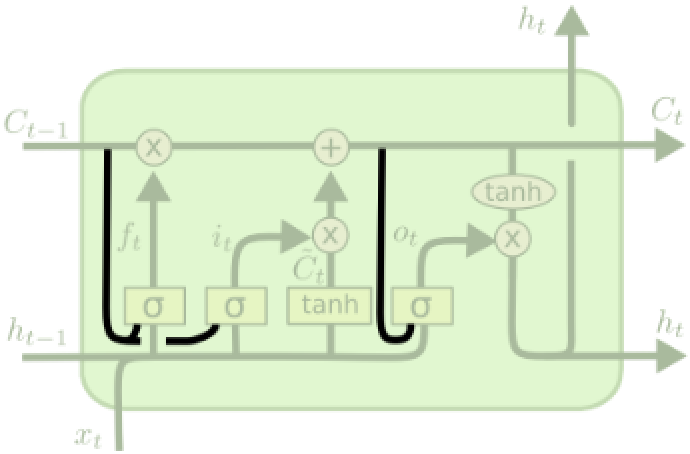
$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$

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

- We multiply the old state by f_t, forgetting the things we decided to forget earlier.

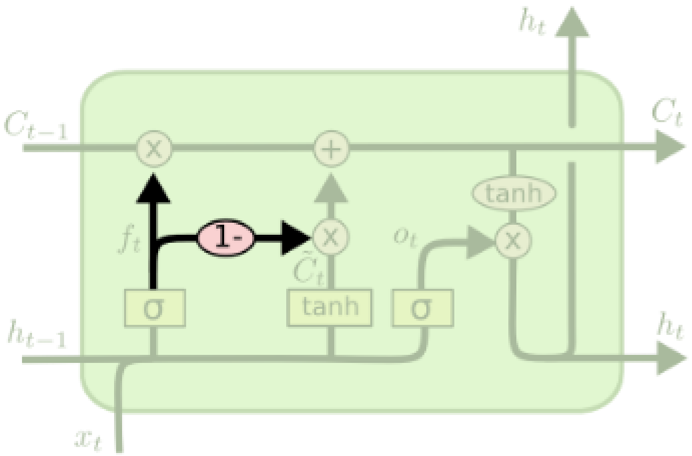
- Then we add $i_t * C_t$. This is the new candidate values, scaled by how much we decided to update each state value.

Peephole Connections



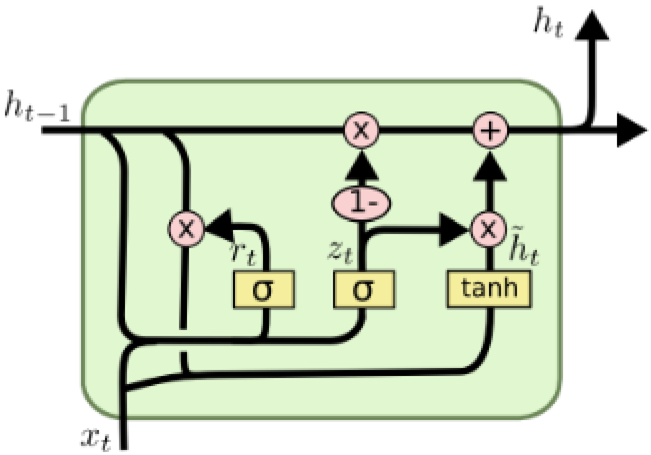
$$f_t = \sigma\left(W_f \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \ + \ b_f\right)$$
$$i_t = \sigma\left(W_i \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \ + \ b_i\right)$$
$$o_t = \sigma\left(W_o \cdot [\boldsymbol{C_t}, h_{t-1}, x_t] \ + \ b_o\right)$$

Coupled Forget and Input Gates



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

Gated Recurrent Unit



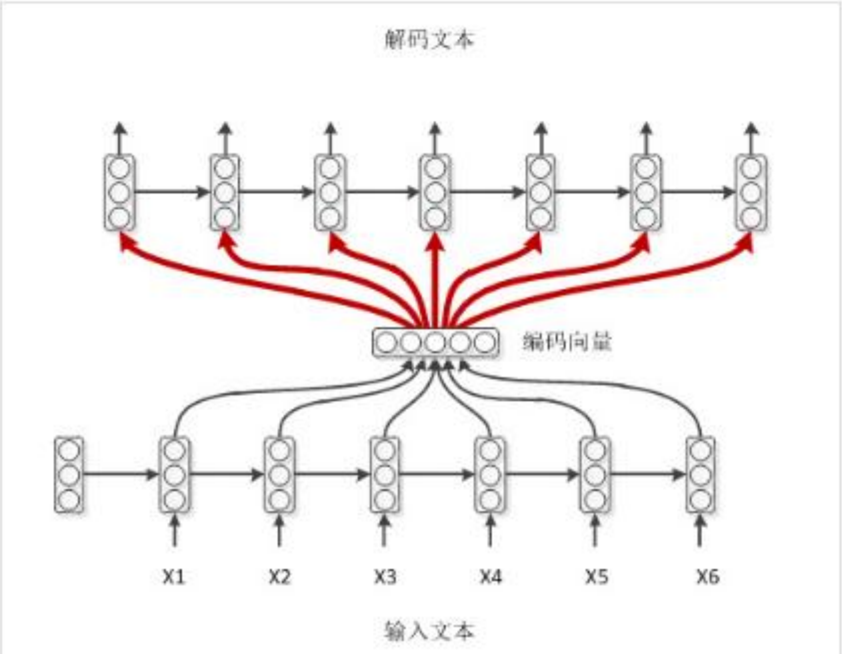
$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

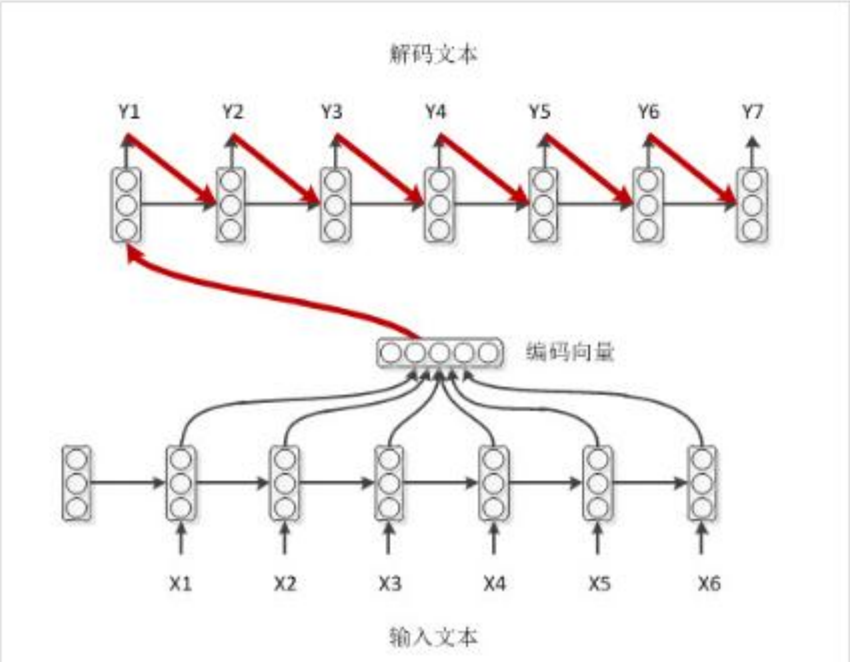
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

En-Dn:学霸型



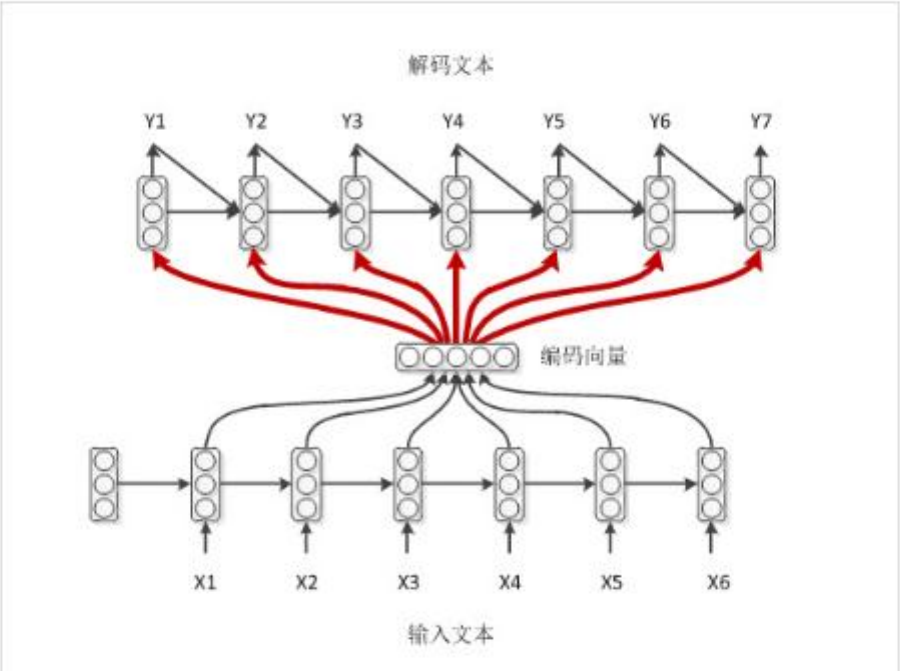
- 把编码端得到的编码向量做为解码模型每个时刻的输入特征

En-Dn:学弱型



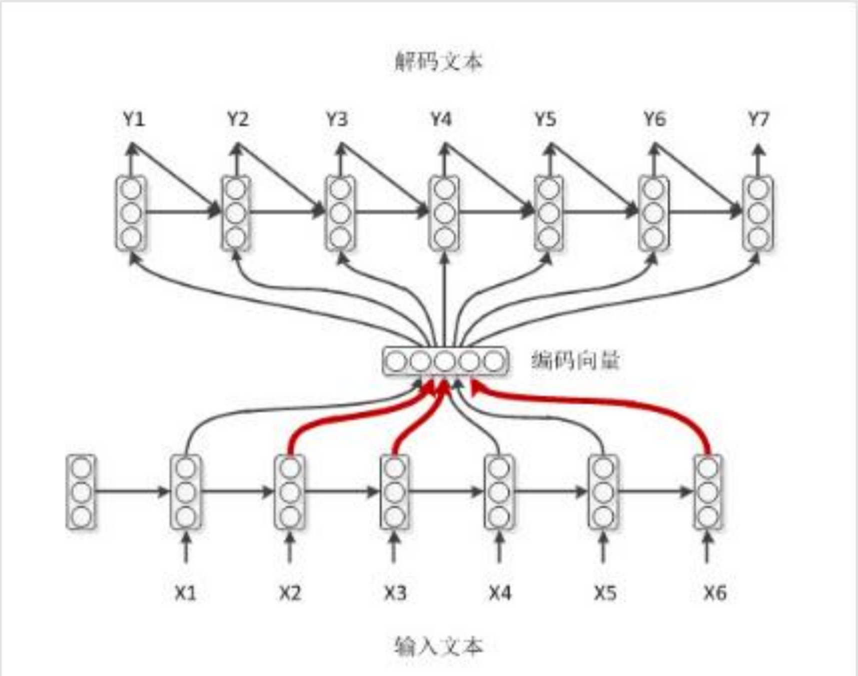
- 带输出回馈的解码方式，将当前时刻之前的输出也作为解码器的输入

En-Dn:学弱型



- 编码向量参与到解码的各个时刻之中

En-Dn:学渣型



- 知道对于当前时刻而言，各个输入的权重/影响力
- 注意力模型

Machine Translation Example

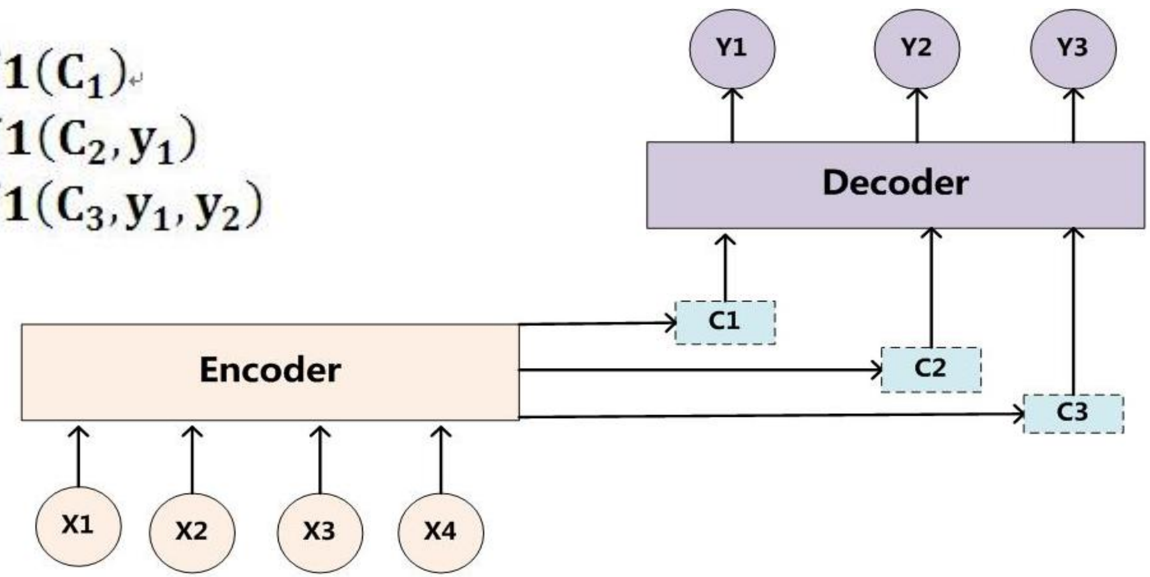
- 举例：Students love science.——学生热爱科学.
- y1=students,y2=love,y3=science.

$$y_1 = f(C)$$
$$y_2 = f(C, y_1)$$
$$y_3 = f(C, y_1, y_2)$$

- 再举例：Science is an art, and students love it.
- it?science:art.
- 科学是一门艺术，学生热爱科学。
- science(0.3),is(0.05),an(0.05),art(0.1),and(0.05),students(0.1), love(0.05),it(0.3)

En-Dn with Attention Model

$$y_1 = f1(C_1)$$
$$y_2 = f1(C_2, y_1)$$
$$y_3 = f1(C_3, y_1, y_2)$$



$C_{\text{学生}}$=g(0.6*f_2(“student”),0.2*f_2(“love”),0.2*f_2(“science”))
$C_{\text{热爱}}$=g(0.2*f_2(“student”),0.7*f_2(“love”),0.1*f_2(“science”))
$C_{\text{科学}}$=g(0.3*f_2(“student”),0.2*f_2(“love”),0.5*f_2(“science”))

How to calculate Attention?

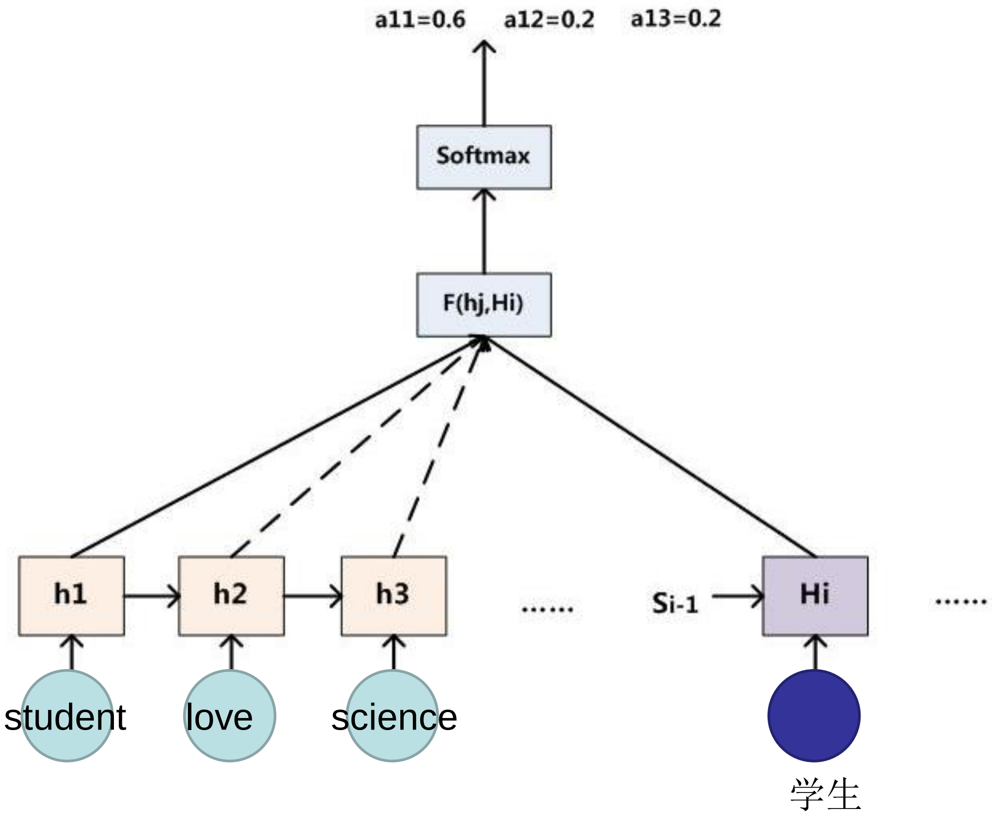
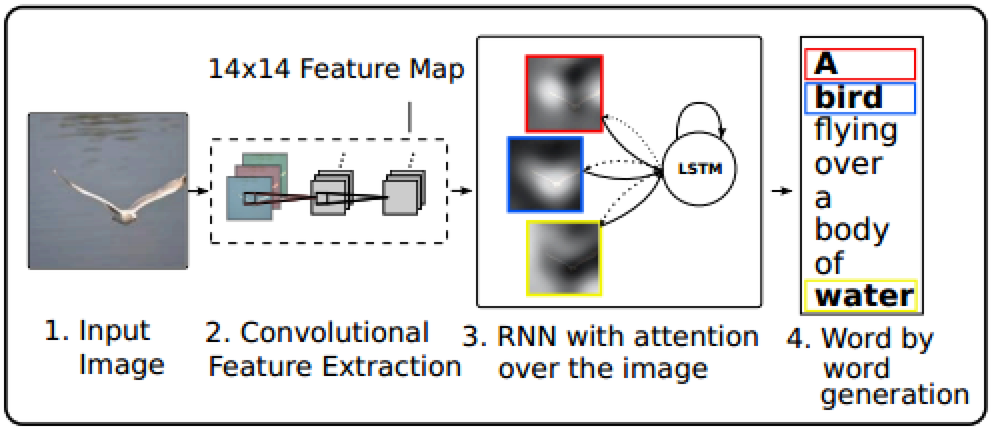


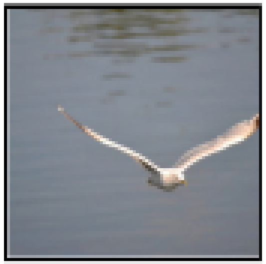
Image Caption

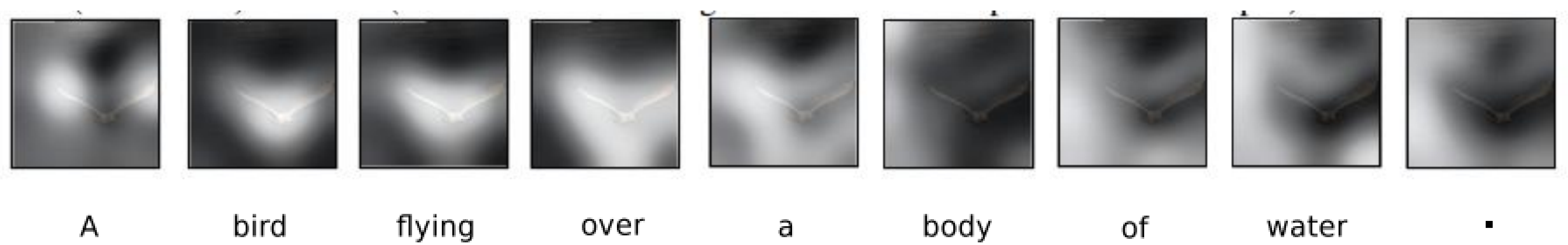


- ➢ Automatically generating captions of an image

$$y = \{\mathbf{y}_1, \ldots, \mathbf{y}_C\}, \; \mathbf{y}_i \in \mathbb{R}^K$$

Example





Encoder:CNN

- We use a convolutional neural network in order to extract a set of feature vectors which we refer to as annotation vectors.

- The extractor produces L vectors, each of which is a D-dimensional representation corresponding to a part of the image.

$$a = \{\mathbf{a}_1, \ldots, \mathbf{a}_L\}, \ \mathbf{a}_i \in \mathbb{R}^D$$

Decoder:LSTM

$$\begin{pmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} T_{D+m+n,n} \begin{pmatrix} \mathbf{E}\mathbf{y}_{t-1} \\ \mathbf{h}_{t-1} \\ \hat{\mathbf{z}}_t \end{pmatrix} \tag{1}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t \tag{2}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t). \tag{3}$$

- the context vector z_t is a dynamic representation of the relevant part of the image input at time t

- the relative importance to give to location i in blending the a_i's together

Soft Attention Model

- The weight α_i of each annotation vector a_i is computed by an attention model f_{att}

- f_{att} :we use a multilayer perceptron conditioned on the previous hidden state h_{t-1}.

$$e_{ti} = f_{\text{att}}(\mathbf{a}_i, \mathbf{h}_{t-1})$$

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^{L} \exp(e_{tk})}.$$

$$\hat{\mathbf{z}}_t = \phi\left(\{\mathbf{a}_i\}, \{\alpha_i\}\right)$$

Initialization and Output

- The initial memory state and hidden state of the LSTM are predicted by an average of the annotation vectors:

$$\mathbf{c}_0 = f_{\text{init,c}}\left(\frac{1}{L}\sum_i^L \mathbf{a}_i\right)$$

$$\mathbf{h}_0 = f_{\text{init,h}}\left(\frac{1}{L}\sum_i^L \mathbf{a}_i\right)$$

- Compute the output word probability given the LSTM state, the context vector and the previous word:

$$p(\mathbf{y}_t|\mathbf{a}, \mathbf{y}_1^{t-1}) \propto \exp(\mathbf{L}_o(\mathbf{E}\mathbf{y}_{t-1} + \mathbf{L}_h\mathbf{h}_t + \mathbf{L}_z\hat{\mathbf{z}}_t))$$

想法与计划

- 数据预处理：1）过滤脏数据；2）平滑或离散化数据；3）道路权重信息

- 需要纳入考虑的：1）将道路间的相互影响纳入考虑；2）合适的Encoder选取；3）拥堵的定义；4）工作日与非工作日、高峰期与非高峰期、长期历史数据与最近短期数据的权衡

- 难点：1）神经网络的设计（特别是加入道路权重等因素）；2）动态的道路权重；3）降低神经网路训练的难度