Deep Learning

Lecture 4

Recap

- Weight initialization
    - Zero
    - Random
    - Xavier
- Batch Normalization
    - Why?
    - Layer, Instance, Group norms
- Convolutions
    - Forward
    - Backward
    - Parameters

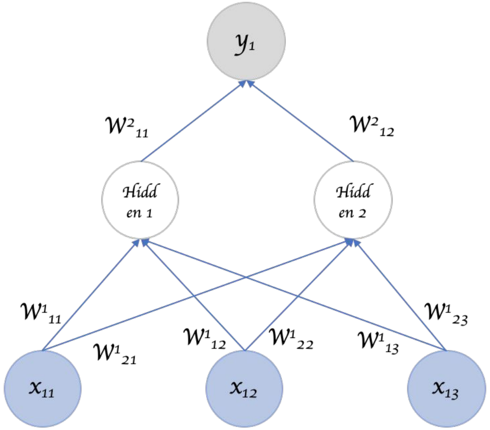
Motivation

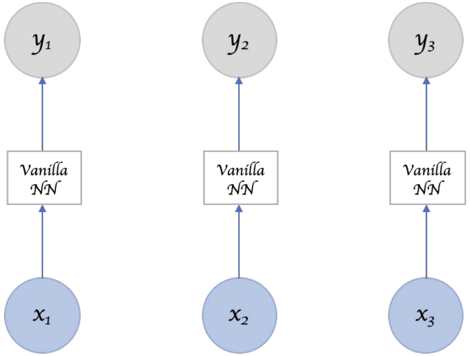
Input: a sequence of arbitrary length

Output: a sequence of arbitrary length

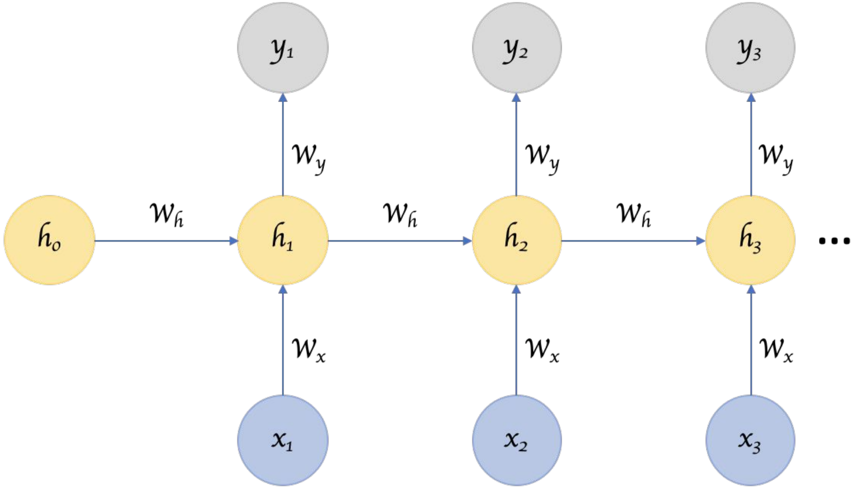
Why not to use feedforward nets?

Or may be CNN?





RNN – Recurrent Neural Network



$$h_t = g(W_x x_t + W_h h_{t-1} + b_h)$$

$$z_t = W_x x_t + W_h h_{t-1} + b_h$$

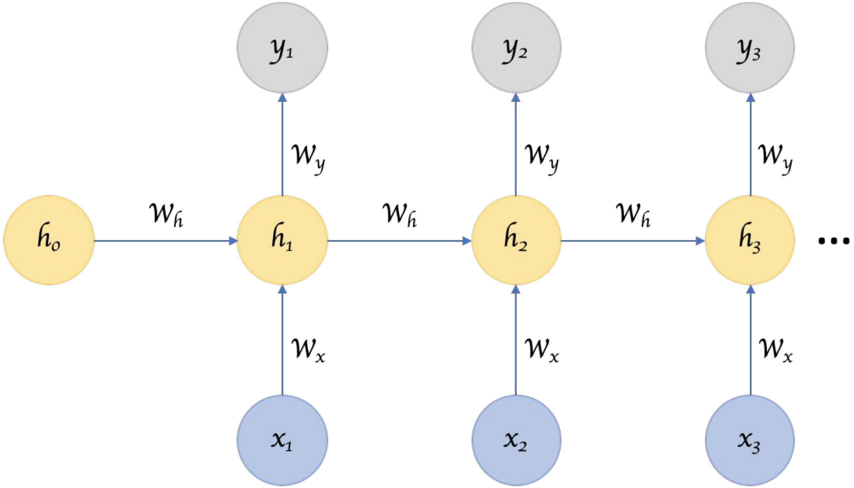
$$h_t = g(z_t)$$

$$y_t = g(W_y h_t + b_y)$$

$$\nabla_{W_h} L - ?$$

RNN backpropagation

$$L = \sum_{t=1}^{T} L(x_t, y_t) \qquad z_t = W_x x_t + W_h h_{t-1} + b_h$$



$\nabla_{W_h} L - ?$ Let $\nabla_{h_t} L$ is given.

$$dL = \nabla_{h_t} L^T dh_t = \nabla_{h_t} L^T \frac{\partial g}{\partial z_t} dW_h h_{t-1} =$$

$$= trace(h_{t-1} \nabla_{h_t} L^T \frac{\partial g}{\partial z_t} dW_h)$$

$$\nabla_{W_h} L^T = h_{t-1} \nabla_{h_t} L^T \frac{\partial g}{\partial z_t}$$

$$\nabla_{W_h} L = \sum_{t=1}^{T} (\frac{\partial g}{\partial z_t})^T \nabla_{h_t} L h_{t-1}^T$$

$$= \sum_{t=1}^{T} (\frac{\partial g}{\partial z_t})^T (\frac{\partial h_{t+1}}{\partial h_t})^T (\frac{\partial h_{t+2}}{\partial h_{t+1}})^T \ldots (\frac{\partial h_T}{\partial h_{T-1}})^T (\frac{\partial y_T}{\partial h_T})^T * \nabla_{y_T} L (h_{t-1})^T$$

RNN backpropagation

$$= \sum_{t=1}^{T} (\frac{\partial g}{\partial z_t})^T (\frac{\partial h_{t+1}}{\partial h_t})^T (\frac{\partial h_{t+2}}{\partial h_{t+1}})^T \dots (\frac{\partial h_T}{\partial h_{T-1}})^T (\frac{\partial y_T}{\partial h_T})^T * \nabla_{y_T} L(h_{t-1})^T$$

$$= \sum_{t=1}^{T} (\frac{\partial g}{\partial z_t})^T \prod_{k=t}^{T-1} (\frac{\partial h_{k+1}}{\partial h_k})^T (\frac{\partial y_T}{\partial h_T})^T * \nabla_{y_T} L(h_{t-1})^T$$

$$\|\nabla_{W_h} L\| \leq \sum_{t=1}^{T} \|\frac{\partial g}{\partial z_t}\| \|\frac{\partial y_T}{\partial h_T}\| \|\nabla_{y_T} L(h_{t-1})^T\| \prod_{k=t}^{T-1} \|\frac{\partial h_{k+1}}{\partial h_k}\|$$

What will happen if $\|\frac{\partial h_{k+1}}{\partial h_k}\| < 1$?

Vanishing gradients

$$\|\frac{\partial h_{k+1}}{\partial h_k}\| < 1$$

Possible solutions:

- Orthogonal matrices: if $W^TW = WW^T = I \Rightarrow \|\nabla_x L\|_2 = \|W^T\nabla_z L\|_2 = \|\nabla_z L\|_2$
  - Right initialisation
  - Regularization $+\lambda\|W^TW - I\|_F^2$
  - Riemannian optimization
  - Note: we need square matrices!
- Batch Normalization (outputs have the same scale)
- Different architectures

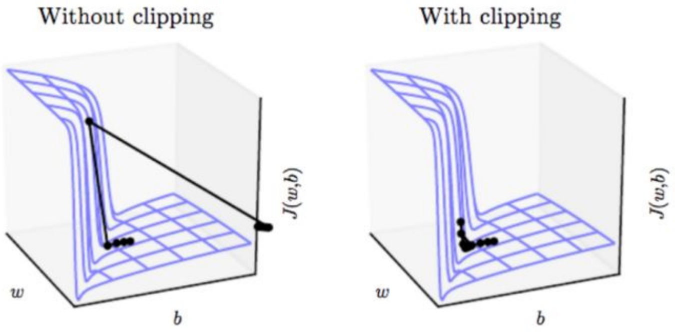
Exploding gradients

$$\prod_{k=t}^{T-1} \|\frac{\partial h_{k+1}}{\partial h_k}\| \rightarrow \infty$$

Possible solution:

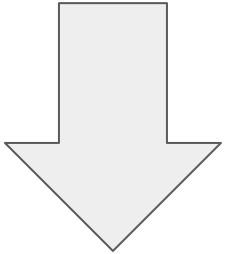
- Gradient clipping

$$g \geq c \Rightarrow \hat{g} = \frac{cg}{\|g\|}$$



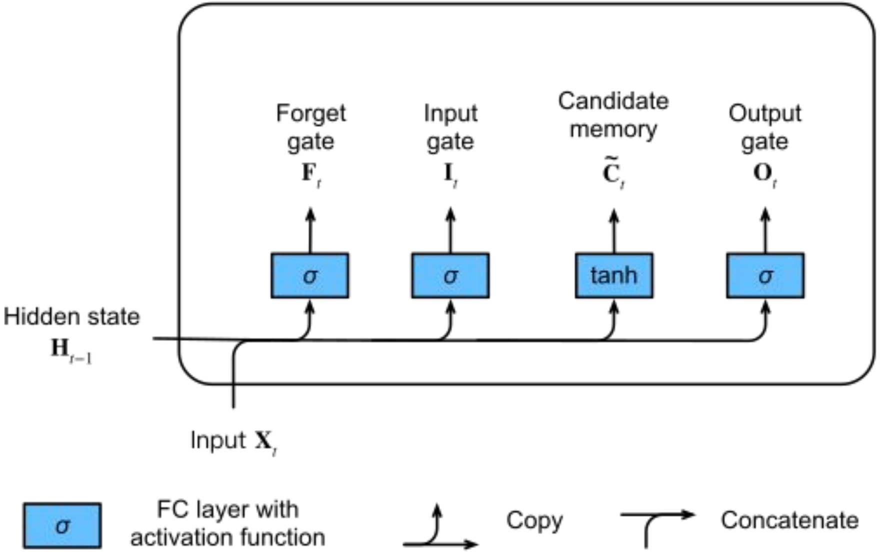
Intuition

15*adhhdsghsjkjljsjbdbjkhjdkljkljdhfkhonxcbzmclcfhjkfhjksahjkhjhxhjc#



ccccccccccccccc

LSTM architecture (Long Short Term Memory)



Forget gate $\mathbf{F}_t$
Input gate $\mathbf{I}_t$
Candidate memory $\tilde{\mathbf{C}}_t$
Output gate $\mathbf{O}_t$
σ
σ
tanh
σ
Hidden state $\mathbf{H}_{t-1}$
Input $\mathbf{X}_t$
σ
FC layer with activation function
Copy
Concatenate

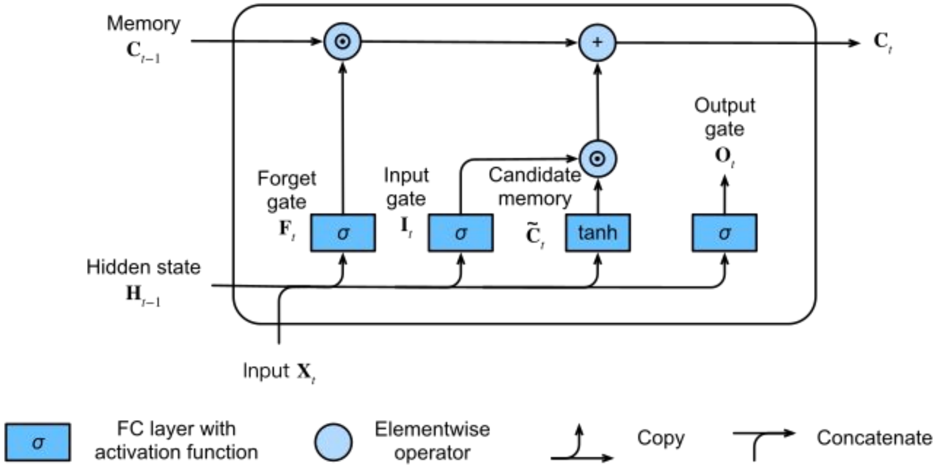
$$\widetilde{c}_t = g(W_x x_t + W_h h_{t-1} + b)$$

$$i_t = \sigma(W_x^i x_t + W_h^i h_{t-1} + b_i) \in (0,1)$$

$$o_t = \sigma(W_x^o x_t + W_h^o h_{t-1} + b^o) \in (0,1)$$

$$f_t = \sigma(W_x^f x_t + W_h^f h_{t-1} + b^f) \in (0,1)$$

LSTM architecture (Long Short Term Memory)



$$\widetilde{c}_t = g(W_x x_t + W_h h_{t-1} + b)$$

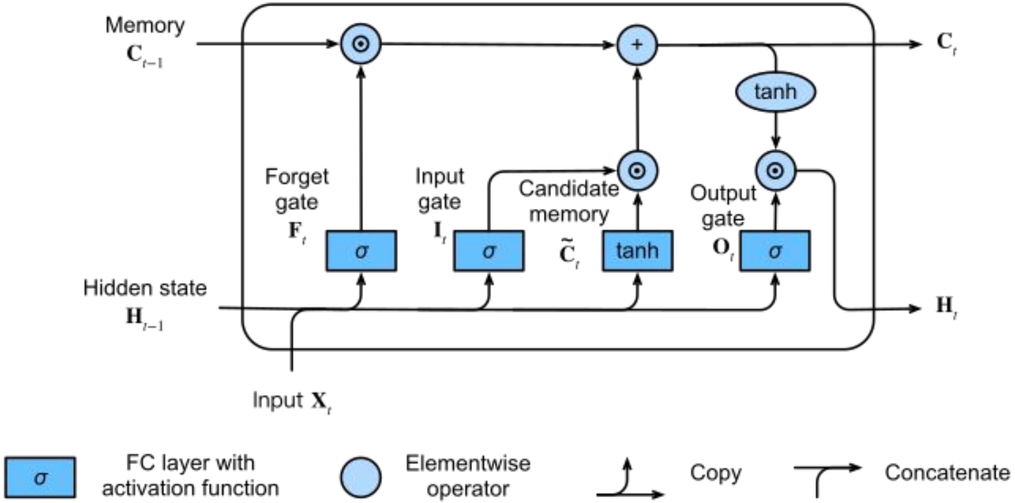
$$i_t = \sigma(W_x^i x_t + W_h^i h_{t-1} + b_i) \in (0,1)$$

$$o_t = \sigma(W_x^o x_t + W_h^o h_{t-1} + b^o) \in (0,1)$$

$$f_t = \sigma(W_x^f x_t + W_h^f h_{t-1} + b^f) \in (0,1)$$

$$c_t = c_{t-1} + i_t \odot \widetilde{c}_t$$

LSTM architecture (Long Short Term Memory)



$$\widetilde{c}_t = g(W_x x_t + W_h h_{t-1} + b)$$

$$i_t = \sigma(W_x^i x_t + W_h^i h_{t-1} + b_i) \in (0,1)$$

$$o_t = \sigma(W_x^o x_t + W_h^o h_{t-1} + b^o) \in (0,1)$$

$$f_t = \sigma(W_x^f x_t + W_h^f h_{t-1} + b^f) \in (0,1)$$

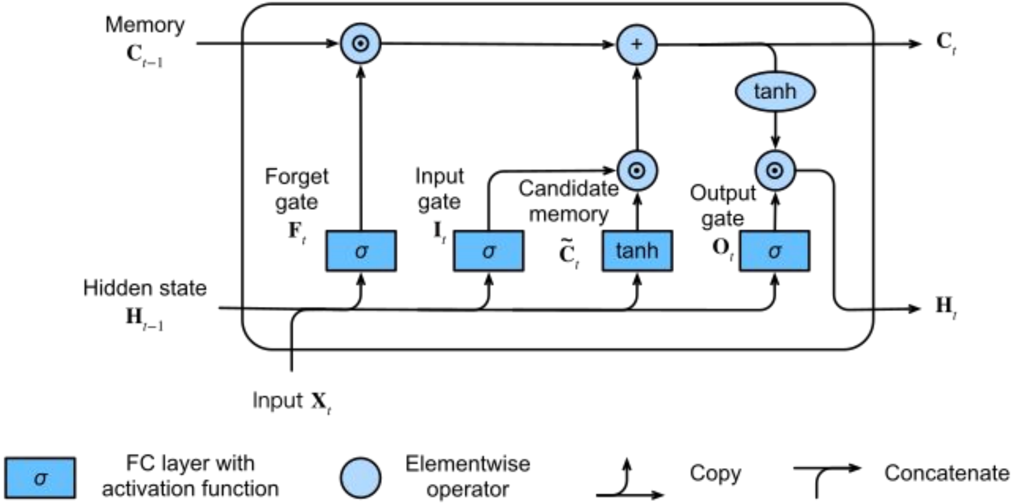
$$c_t = c_{t-1} + i_t \odot \widetilde{c}_t$$

$$h_t = o_t \odot tanh(c_t)$$

$$f_t = \sigma(W_x^f x_t + W_h^f h_{t-1} + b^f) \in (0,1)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \widetilde{c}_t$$

LSTM architecture (Long Short Term Memory)



Memory $\mathbf{C}_{t-1}$
$\mathbf{C}_t$
tanh
Forget gate $\mathbf{F}_t$
Input gate $\mathbf{I}_t$
Candidate memory $\tilde{\mathbf{C}}_t$
Output gate $\mathbf{O}_t$
σ
σ
tanh
σ
Hidden state $\mathbf{H}_{t-1}$
$\mathbf{H}_t$
Input $\mathbf{X}_t$
σ
FC layer with activation function
Elementwise operator
Copy
Concatenate

$$\widetilde{c}_t = g(W_x x_t + W_h h_{t-1} + b)$$

$$i_t = \sigma(W_x^i x_t + W_h^i h_{t-1} + b_i) \in (0,1)$$

$$o_t = \sigma(W_x^o x_t + W_h^o h_{t-1} + b^o) \in (0,1)$$

$$f_t = \sigma(W_x^f x_t + W_h^f h_{t-1} + b^f) \in (0,1)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \widetilde{c}_t$$

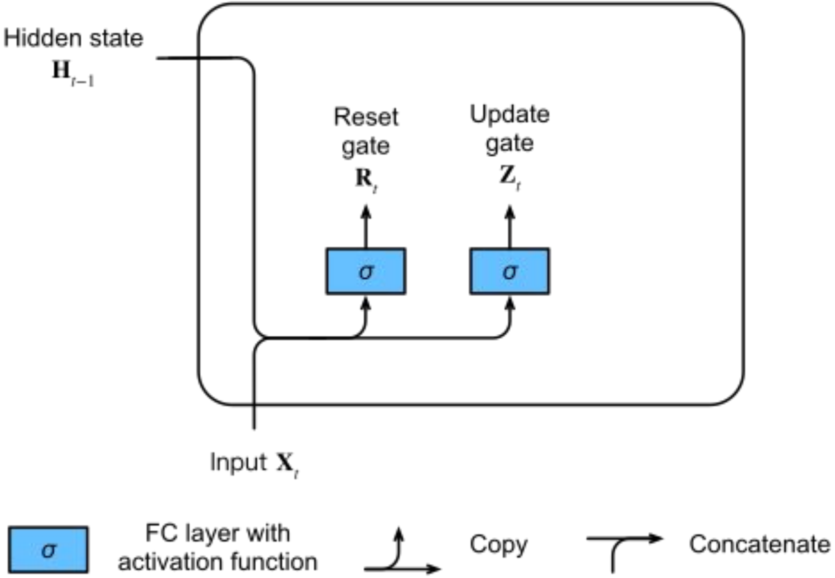
$$h_t = o_t \odot tanh(c_t)$$

$$f_t = \sigma(W_x^f x_t + W_h^f h_{t-1} + b^f) \in (0,1)$$

Why we need tanh? Any ideas?

$\frac{\partial c_t}{\partial c_{t-1}} = I$ leads to not vanishing gradients without forget gate. In other case, $b_f >> 0$

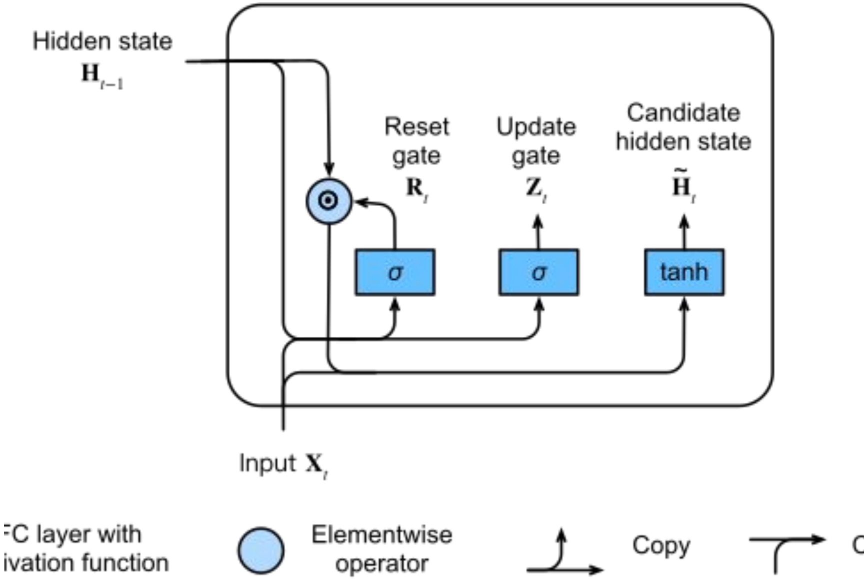
GRU architecture


Hidden state $\mathbf{H}_{t-1}$
Reset gate $\mathbf{R}_t$
Update gate $\mathbf{Z}_t$
σ
σ
Input $\mathbf{X}_t$
σ
FC layer with activation function
Copy
Concatenate

$$r_t = \sigma(W_x^r x_t + W_h^r h_{t-1} + b^r) - \text{reset}$$

$$z_t = \sigma(W_x^u x_t + W_h^u h_{t-1} + b^u) - \text{update}$$

GRU architecture

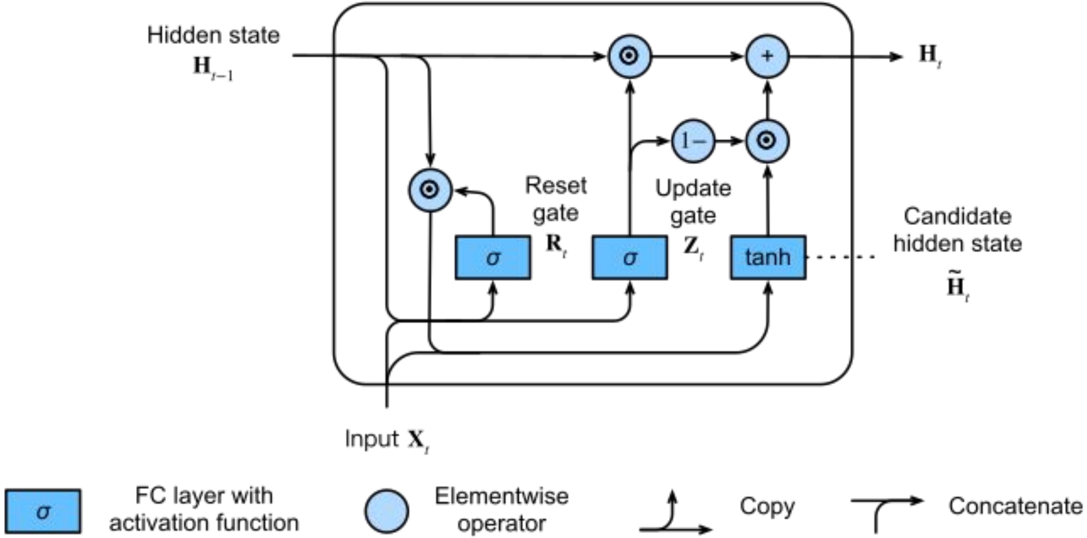


$$r_t = \sigma(W_x^r x_t + W_h^r h_{t-1} + b^r) - \text{reset}$$

$$z_t = \sigma(W_x^u x_t + W_h^u h_{t-1} + b^u) - \text{update}$$

$$\widetilde{h}_t = g(W_x x_t + W_h(h_{t-1} \odot r_t) + b)$$

GRU architecture



$$r_t = \sigma(W_x^r x_t + W_h^r h_{t-1} + b^r)$$

$$z_t = \sigma(W_x^u x_t + W_h^u h_{t-1} + b^u)$$

$$\widetilde{h}_t = g(W_x x_t + W_h(h_{t-1} \odot r_t) + b)$$

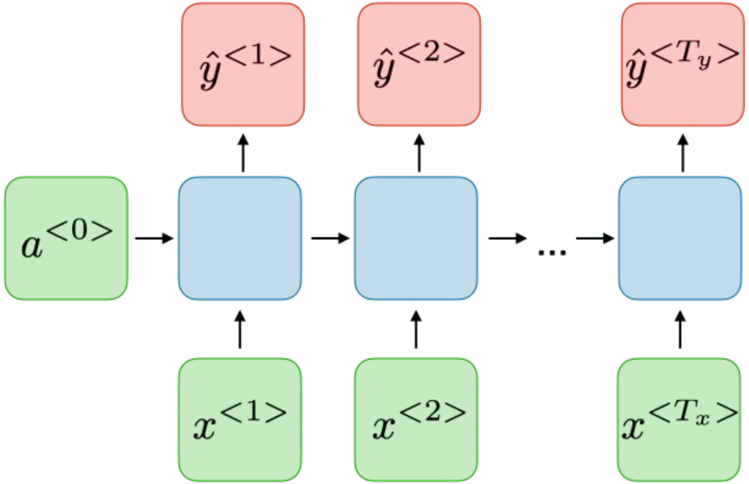
$$h_t = (1 - z_t) \odot g_t + u_t \odot h_{t-1}$$

Applications

Classification of sequence elements

Synced sequence input and output:

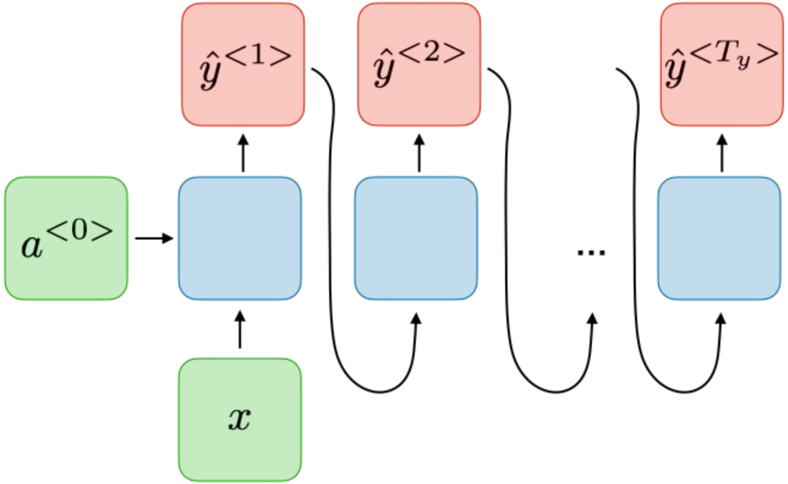
- POS tagging
- Video frames classification



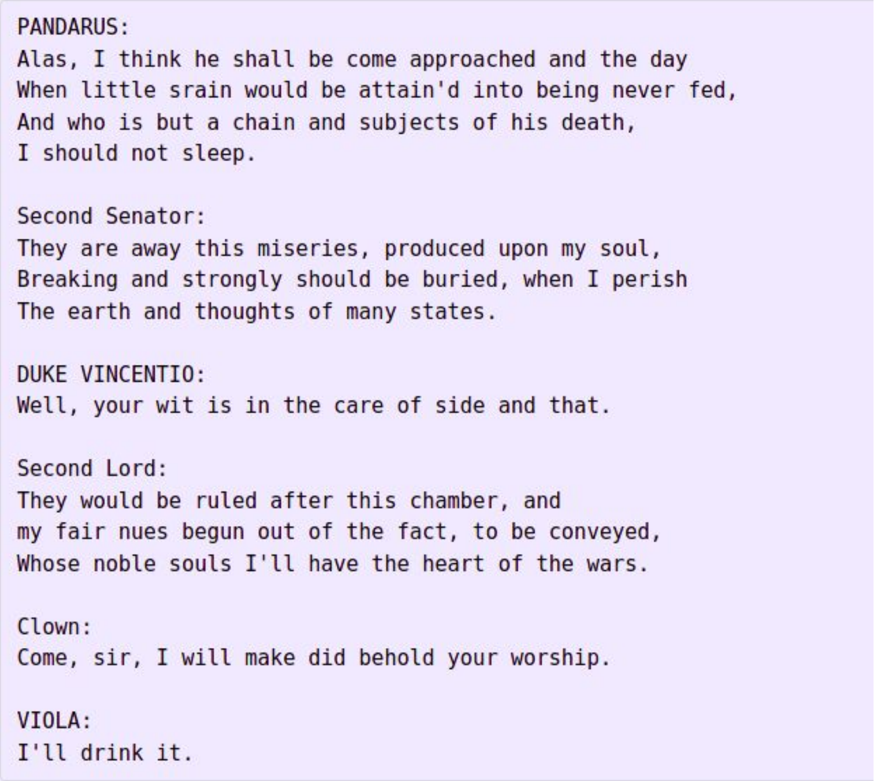
Generation

Text or music generation:

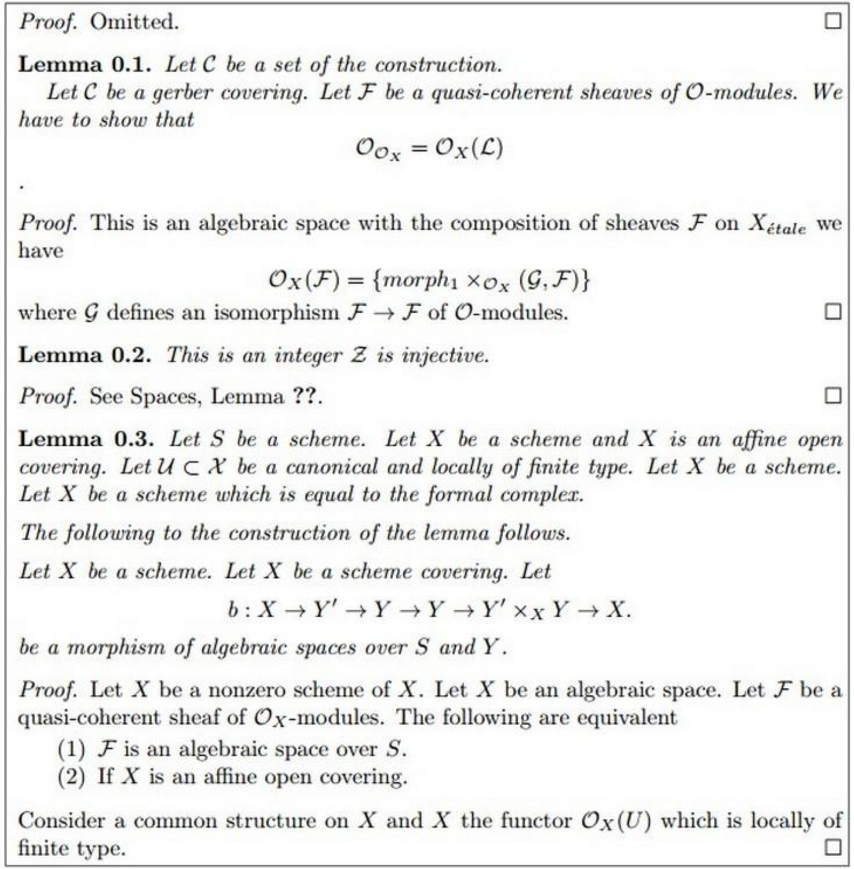
Current word\symbol -> new one



Examples



```
PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.

Second Lord:
They would be ruled after this chamber, and
my fair nues begun out of the fact, to be conveyed,
Whose noble souls I'll have the heart of the wars.

Clown:
Come, sir, I will make did behold your worship.

VIOLA:
I'll drink it.
```



Proof. Omitted. □

Lemma 0.1. *Let $\mathcal{C}$ be a set of the construction.*

Let $\mathcal{C}$ be a gerber covering. Let $\mathcal{F}$ be a quasi-coherent sheaves of $\mathcal{O}$-modules. We have to show that

$$\mathcal{O}_{\mathcal{O}_X} = \mathcal{O}_X(\mathcal{L})$$

.

Proof. This is an algebraic space with the composition of sheaves $\mathcal{F}$ on $X_{étale}$ we have

$$\mathcal{O}_X(\mathcal{F}) = \{morph_1 \times_{\mathcal{O}_X} (\mathcal{G}, \mathcal{F})\}$$

where $\mathcal{G}$ defines an isomorphism $\mathcal{F} \to \mathcal{F}$ of $\mathcal{O}$-modules. □

Lemma 0.2. *This is an integer $\mathcal{Z}$ is injective.*

Proof. See Spaces, Lemma **??**. □

Lemma 0.3. *Let S be a scheme. Let X be a scheme and X is an affine open covering. Let $\mathcal{U} \subset \mathcal{X}$ be a canonical and locally of finite type. Let X be a scheme. Let X be a scheme which is equal to the formal complex.*

The following to the construction of the lemma follows.

Let X be a scheme. Let X be a scheme covering. Let

$$b : X \to Y' \to Y \to Y \to Y' \times_X Y \to X.$$

be a morphism of algebraic spaces over S and Y.

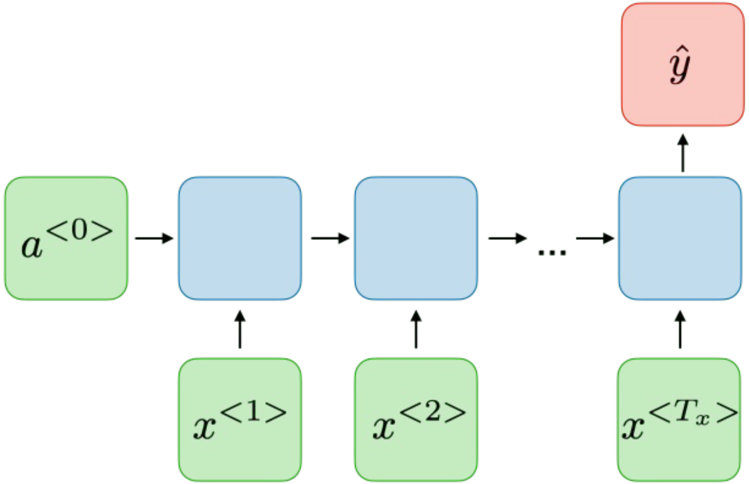
Proof. Let X be a nonzero scheme of X. Let X be an algebraic space. Let $\mathcal{F}$ be a quasi-coherent sheaf of $\mathcal{O}_X$-modules. The following are equivalent

(1) $\mathcal{F}$ is an algebraic space over S.
(2) If X is an affine open covering.

Consider a common structure on X and X the functor $\mathcal{O}_X(U)$ which is locally of finite type. □

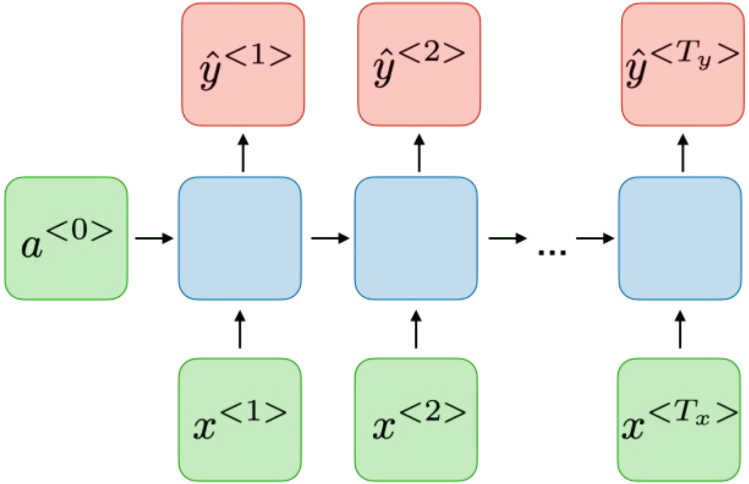
Sequence classification

- Sentiment analysis
- Time series classification



Sequence to sequence tasks

- Handwriting to text/text to handwriting
- Speech to text/text to speech
- Note: input and output can have different lengths



Sequence to sequence tasks

- Machine translation

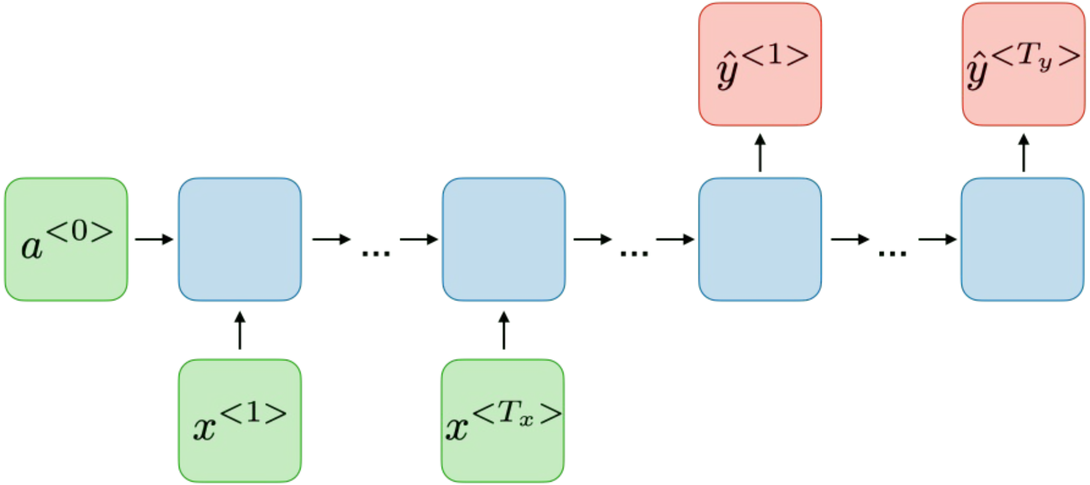
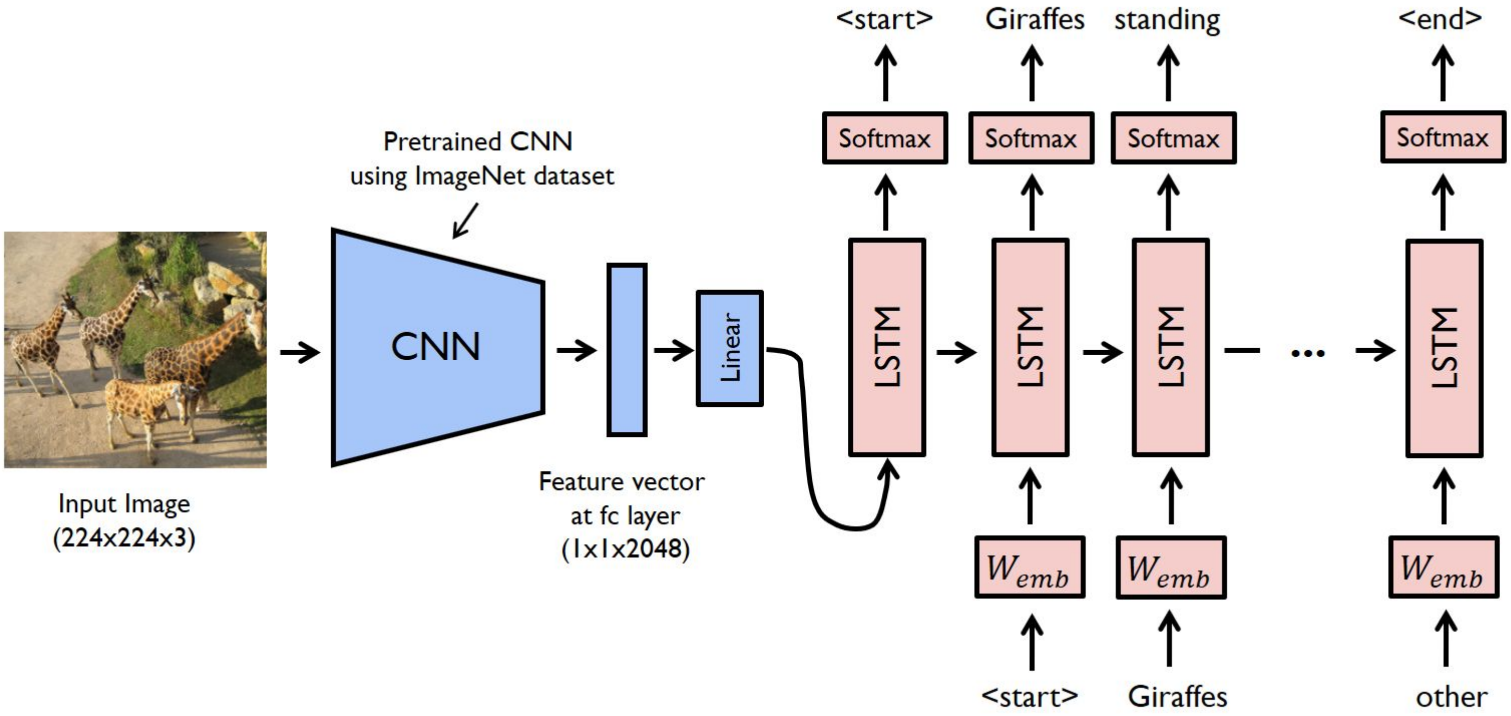
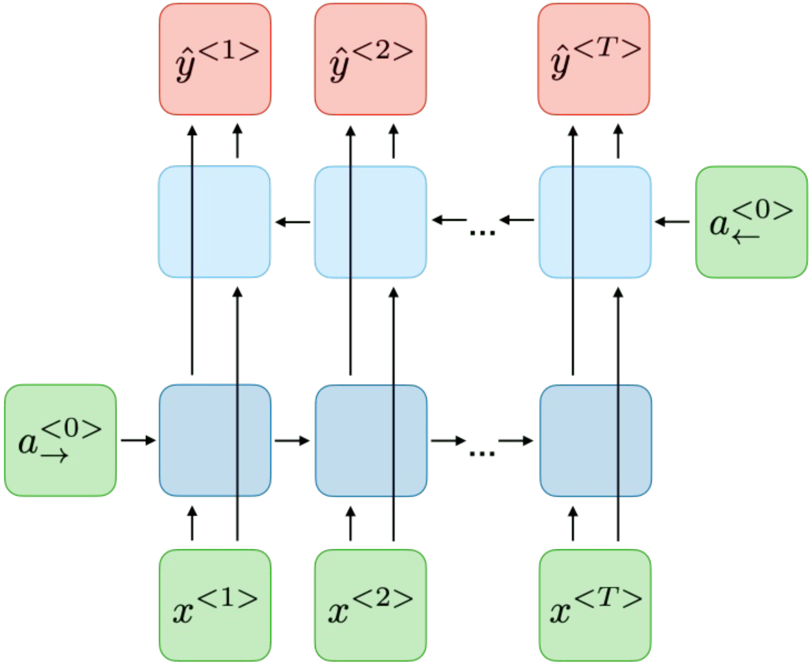
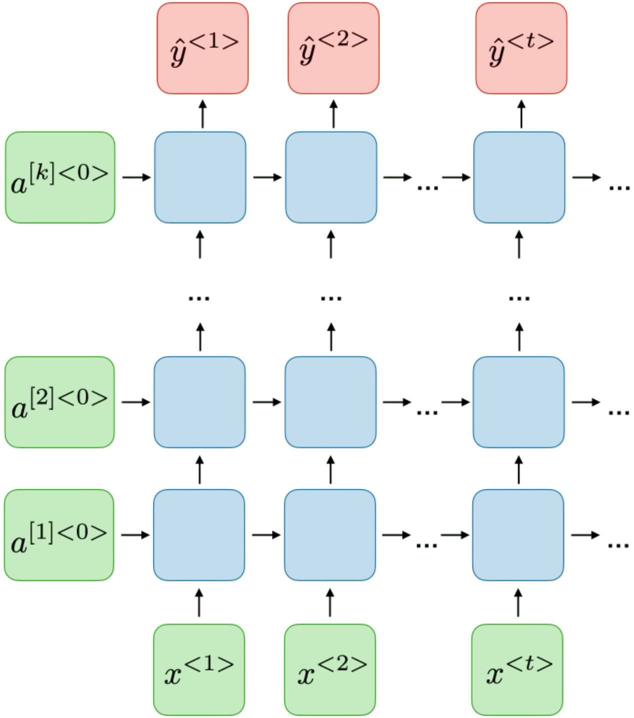


Image caption generation

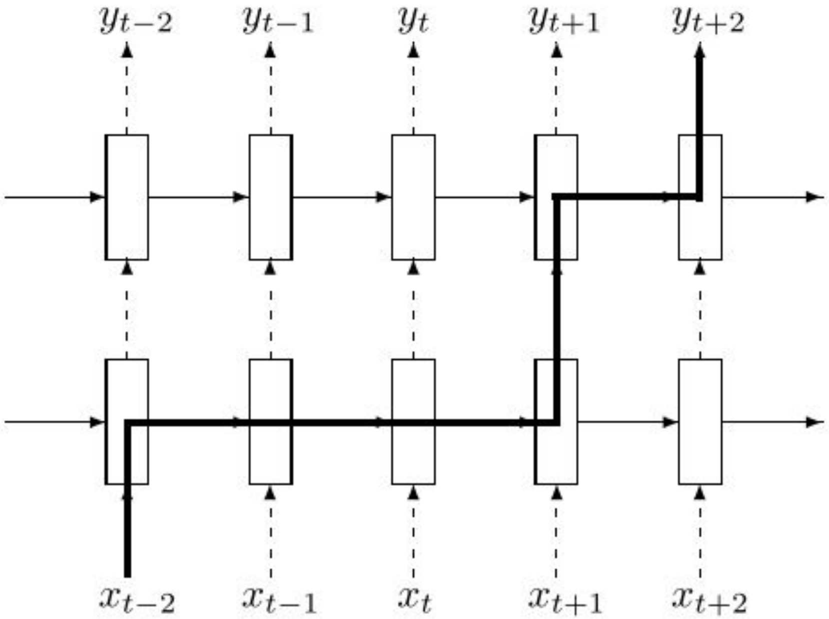


Bidirectional RNN and Deep RNN



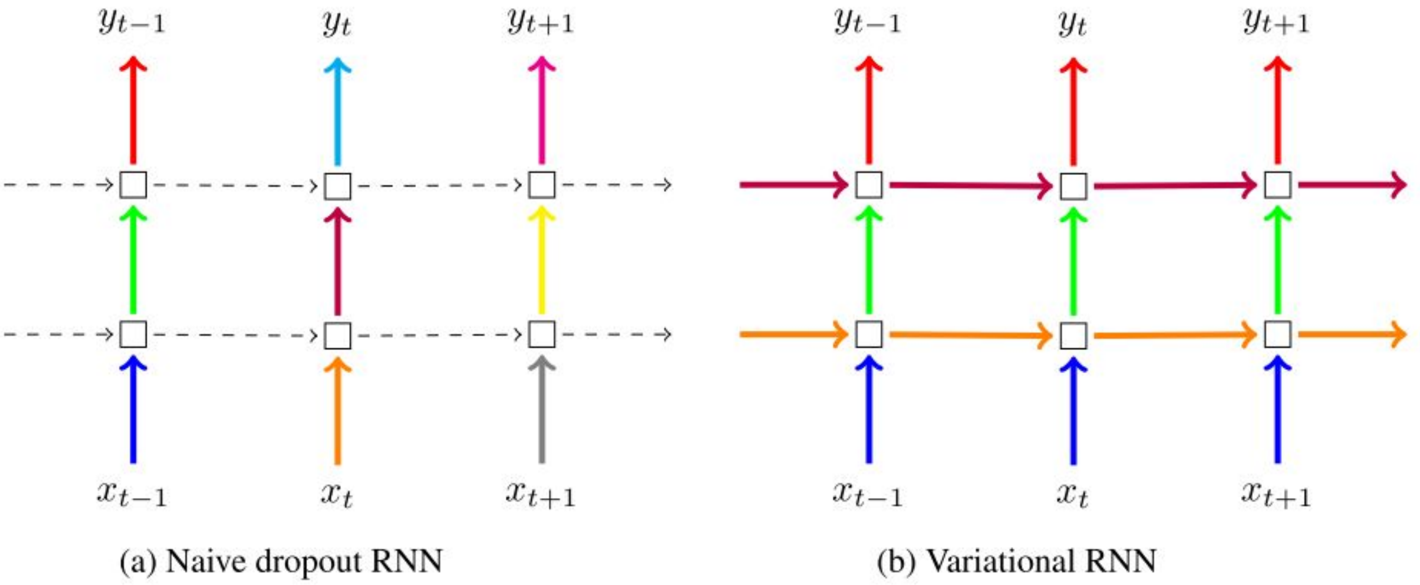


Naive Dropout in RNN



Dropout is only applied to the non-recurrent connections (ie only applied to the feedforward dashed lines). The thick line shows a typical path of information flow in the LSTM.

Variational Dropout

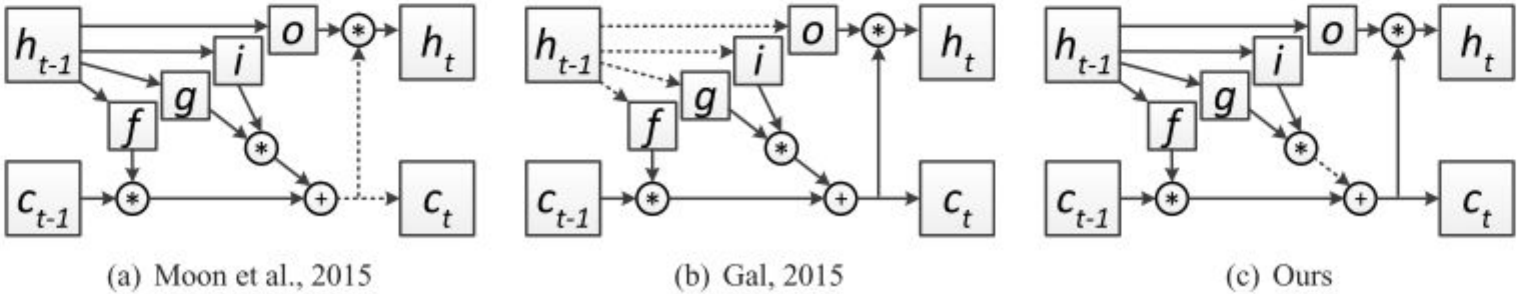


(a) Naive dropout RNN

(b) Variational RNN

Naive dropout (a) (Zaremba et al., 2014) uses different masks at different time steps, with no dropout on the recurrent layers. Variational Dropout (b) uses the same dropout mask at each time step, including the recurrent layers (colours representing dropout masks, solid lines representing dropout, dashed lines representing standard connections with no dropout).

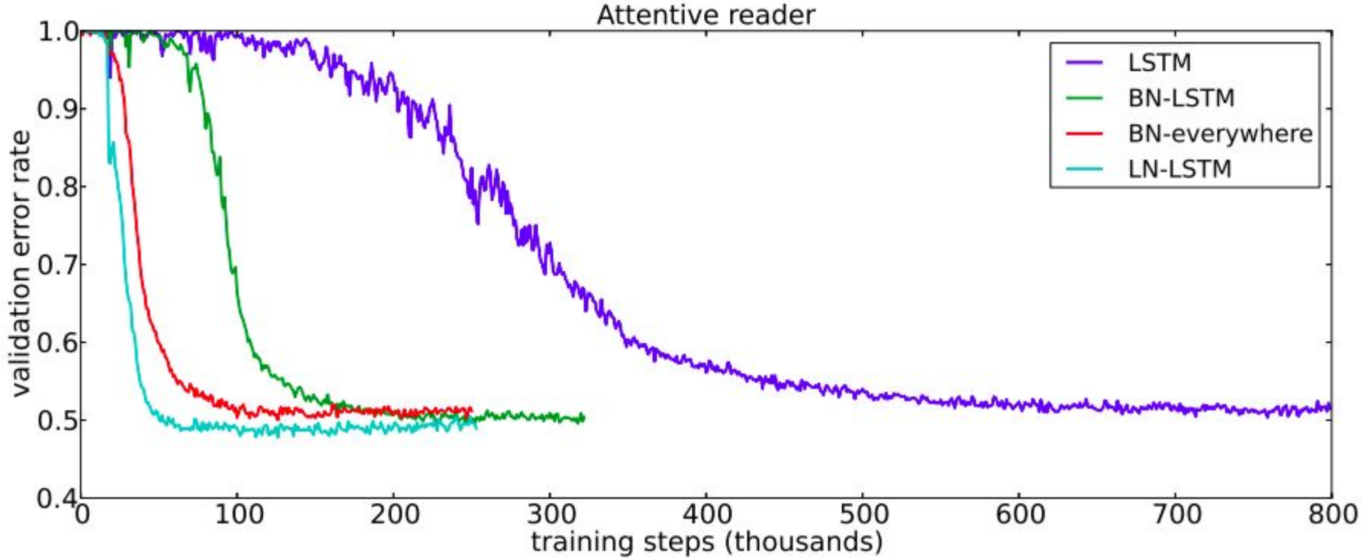
Dropout for LSTM and GRU



Semeniuta et al. “Illustration of the three types of dropout in recurrent connections of LSTM networks. Dashed arrows refer to dropped connections. Input connections are omitted for clarity.”

Layer normalization for RNN

$$\mathbf{h}^t = f\left[\frac{\mathbf{g}}{\sigma^t} \odot \left(\mathbf{a}^t - \mu^t\right) + \mathbf{b}\right] \qquad \mu^t = \frac{1}{H}\sum^{H} a_i^t \qquad \sigma^t = \sqrt{\frac{1}{H}\sum^{H} \left(a_i^t - \mu^t\right)^2}$$



Lei Ba et al., (2016)

Recap

- RNN
    - Backward
    - Vanishing gradients
    - Exploding gradients
- LSTM
- GRU
- Applications
- Dropout and BN