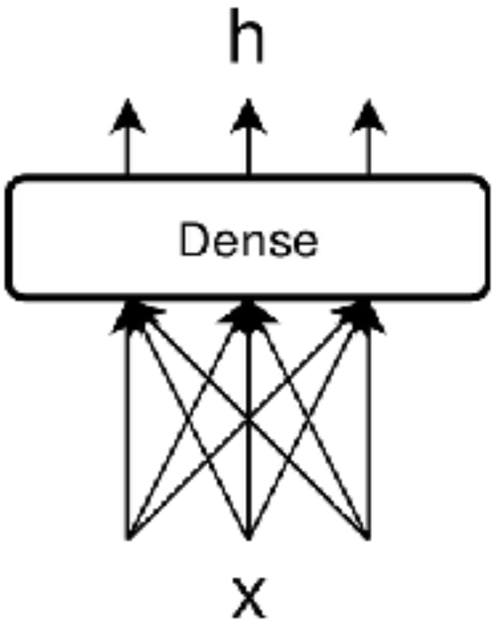
Recurrent Neural Networks

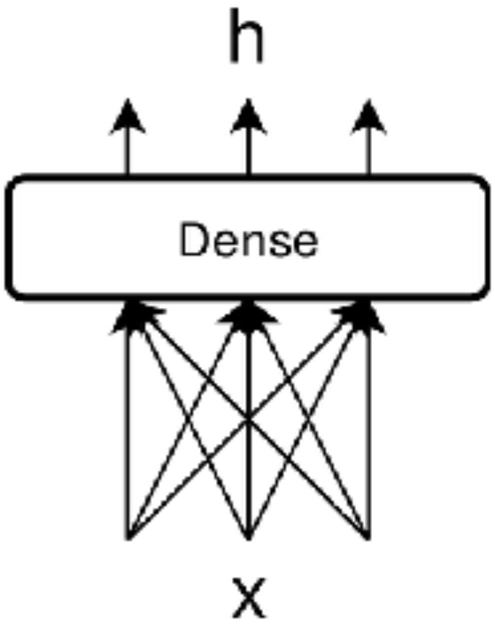
Mikhail Arkhipov
Laboratory of Neural Systems and Deep Learning
MIPT

Dense Layers



$$h = f(Wx + b)$$

Dense Layers

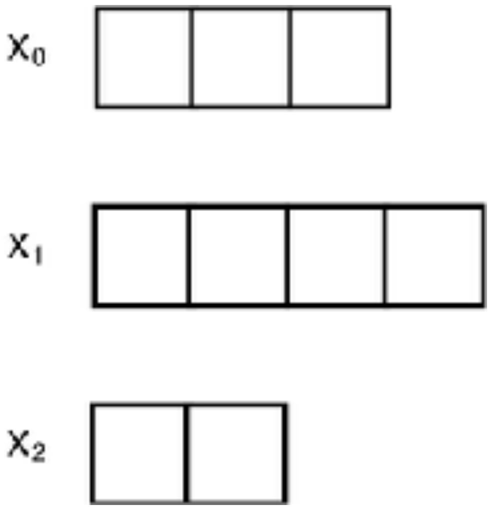


H x 1 H x N H x 1

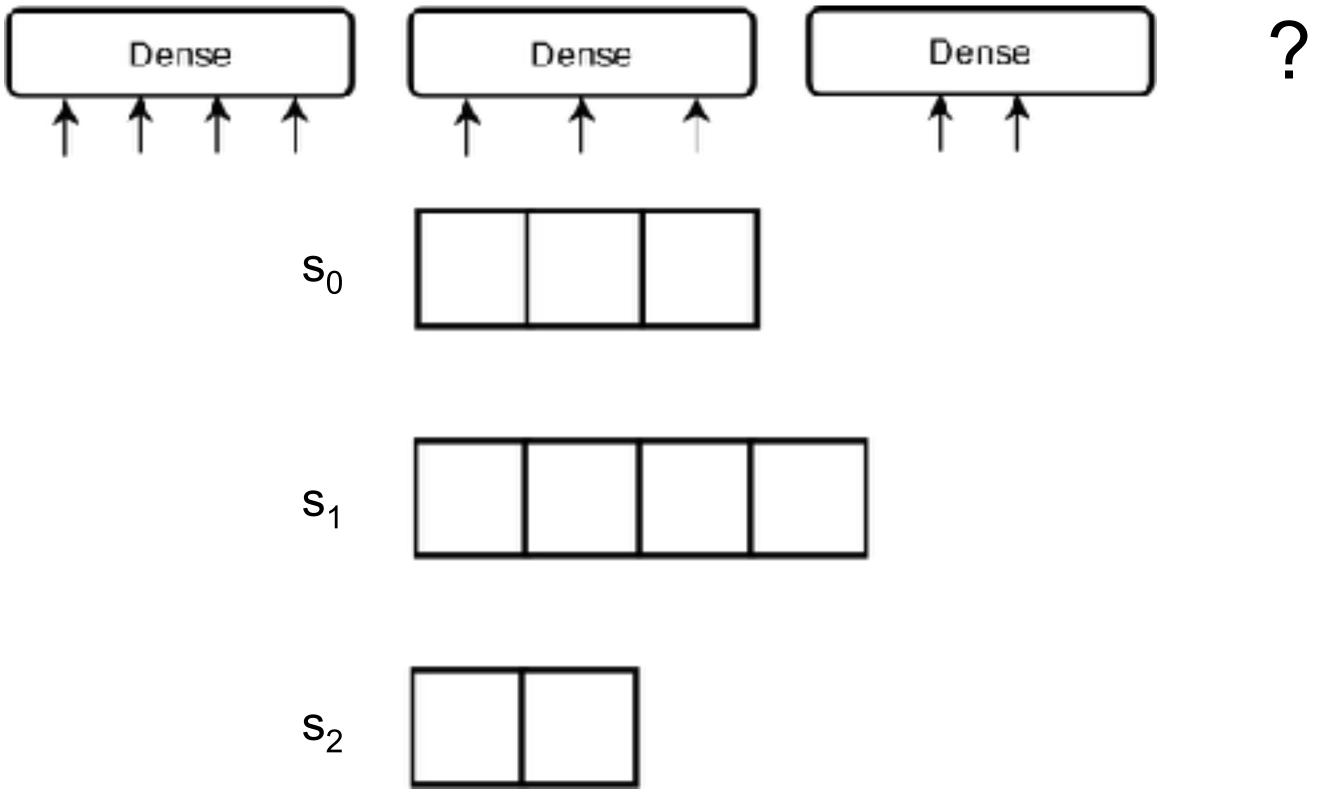
$$h = f(Wx + b)$$

N x 1

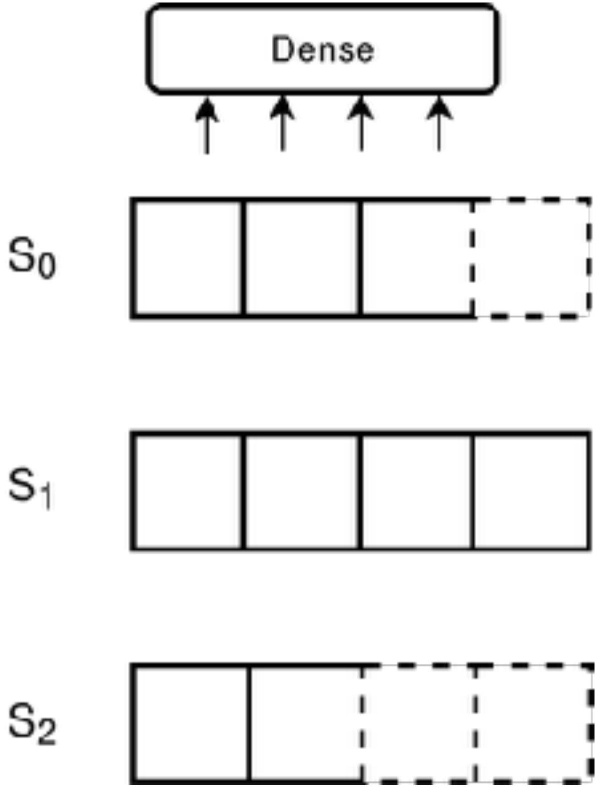
Variable sequence length



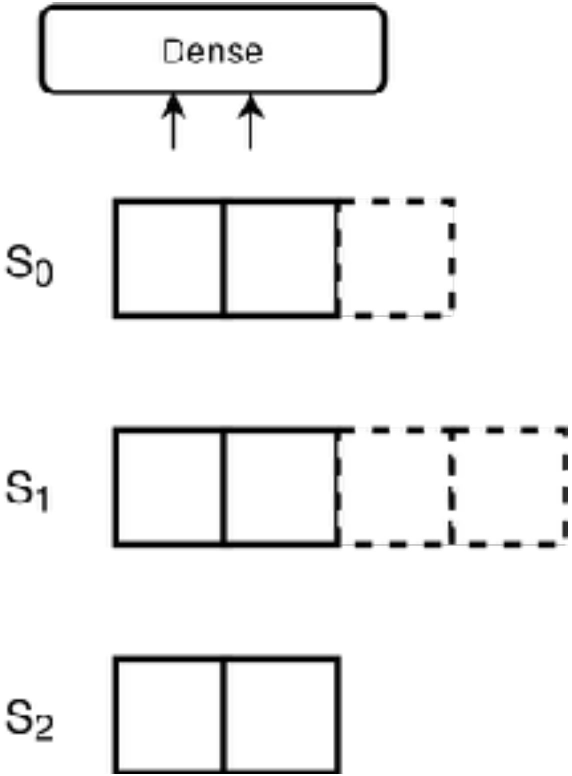
Variable sequence length



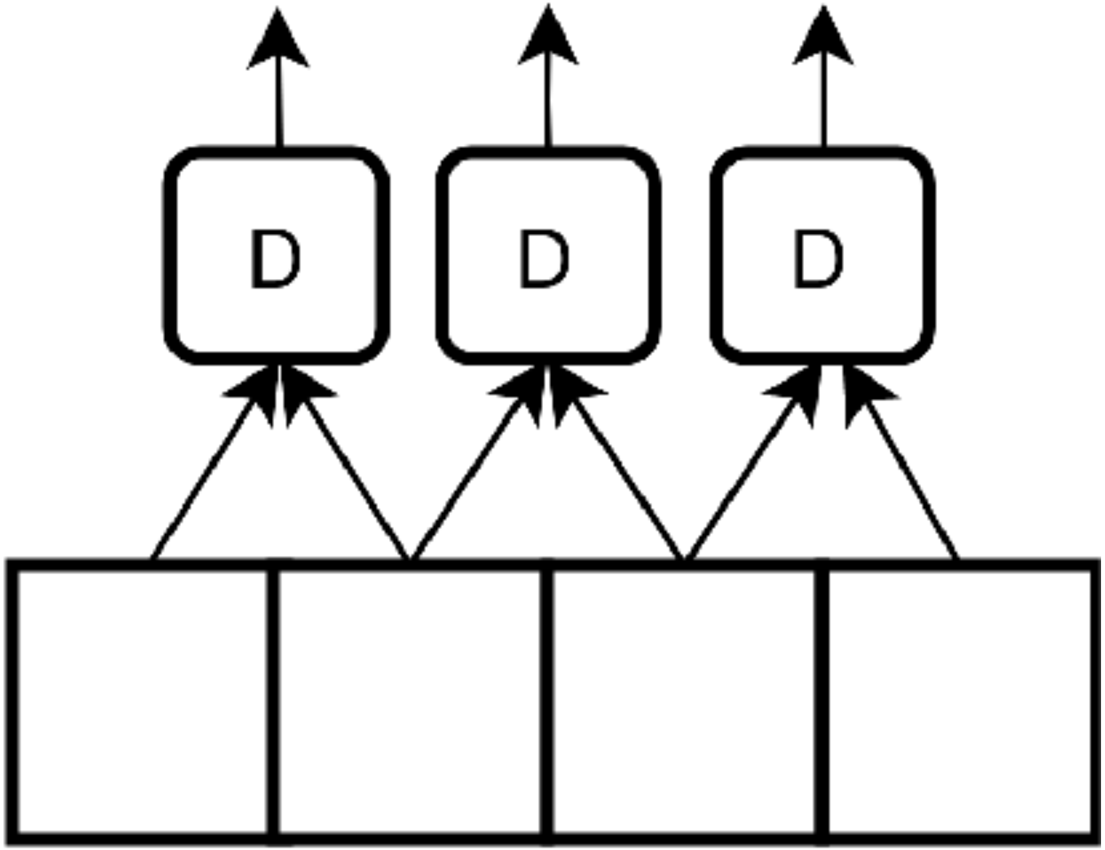
Variable sequence length

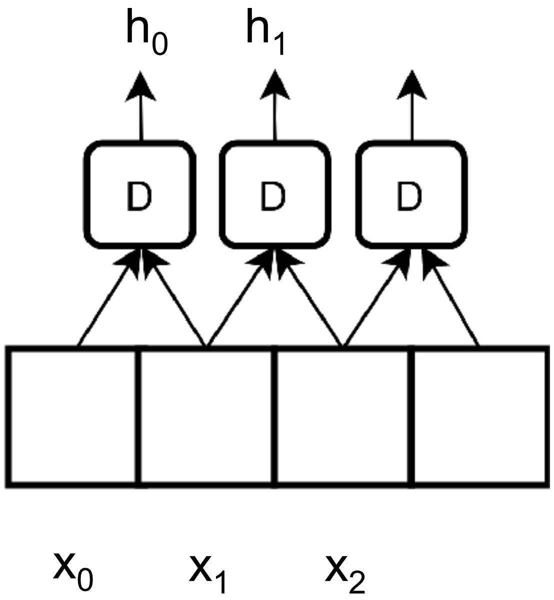


Variable sequence length

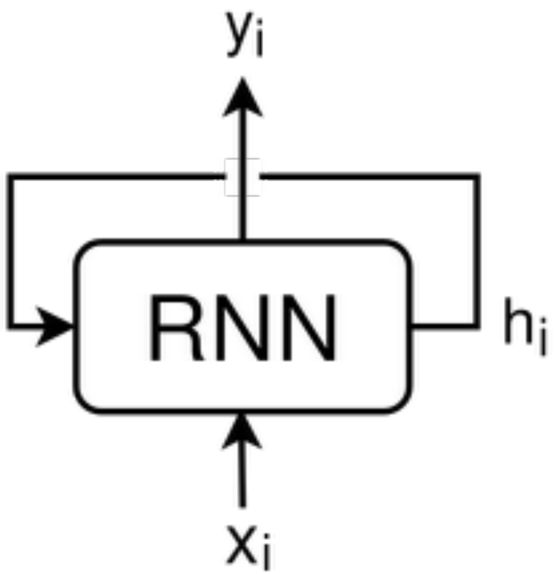


Convolutional Neural Networks



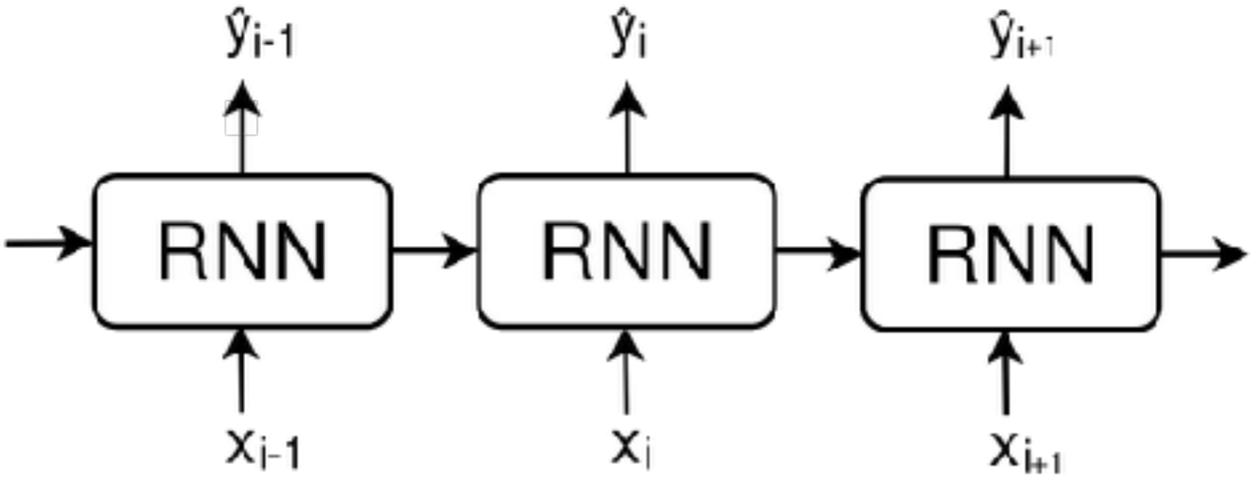

h_0
h_1
D
D
D
x_0
x_1
x_2

Recurrent neural network



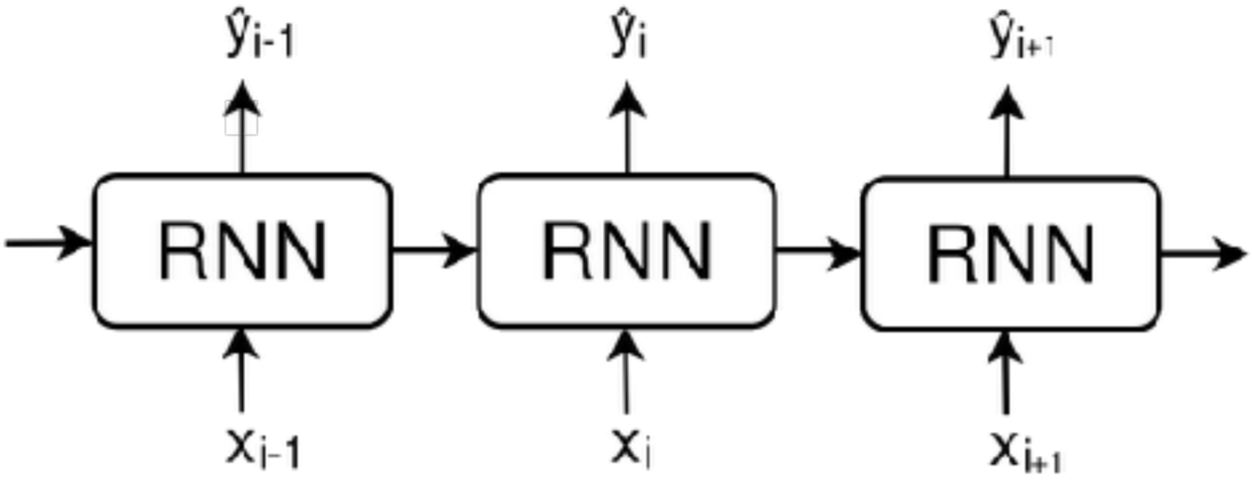
$$h_i = f_h(Wx_i + Vh_{i-1} + b_h) \qquad \hat{y}_i = f_y(Uh_i + b_y)$$

Recurrent neural network



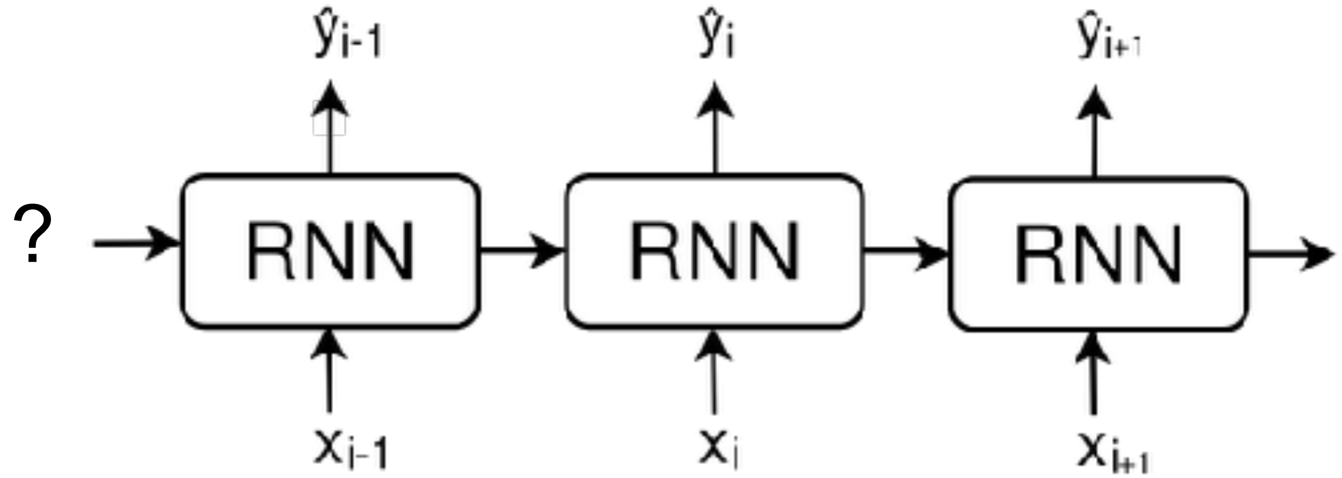
$$h_i = f_h(Wx_i + Vh_{i-1} + b_h) \qquad \hat{y}_i = f_y(Uh_i + b_y)$$

Recurrent neural network



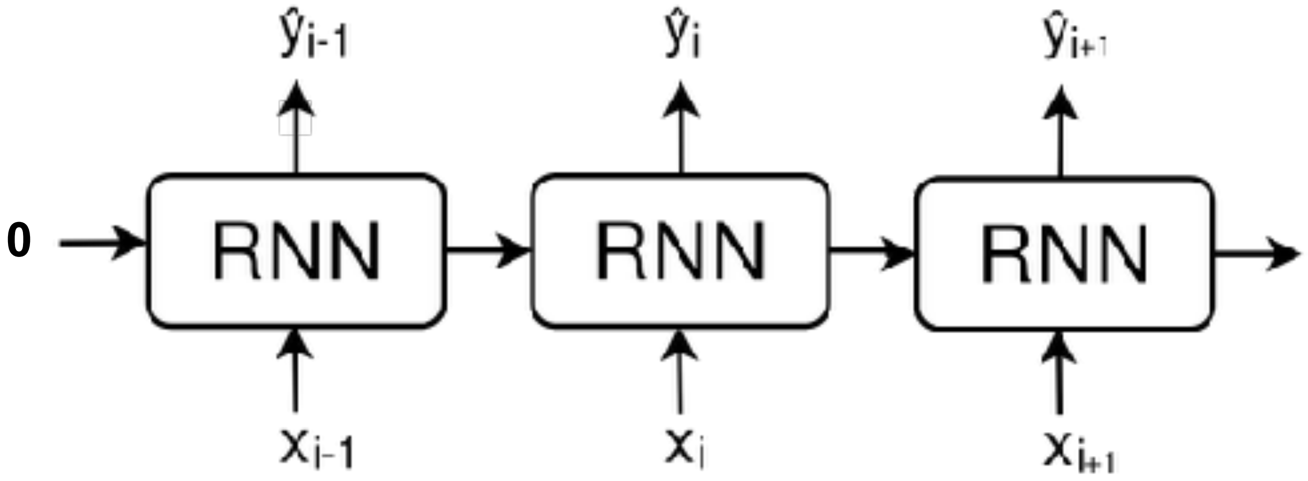
$$h_i = f_h(\overset{H \times N}{W} \underset{N \times 1}{x_i} + \overset{H \times H}{V} \underset{H \times 1}{h_{i-1}} + \underset{H \times 1}{b_h}) \qquad \hat{y}_i = f_y(U h_i + b_y)$$

Recurrent neural network initial states



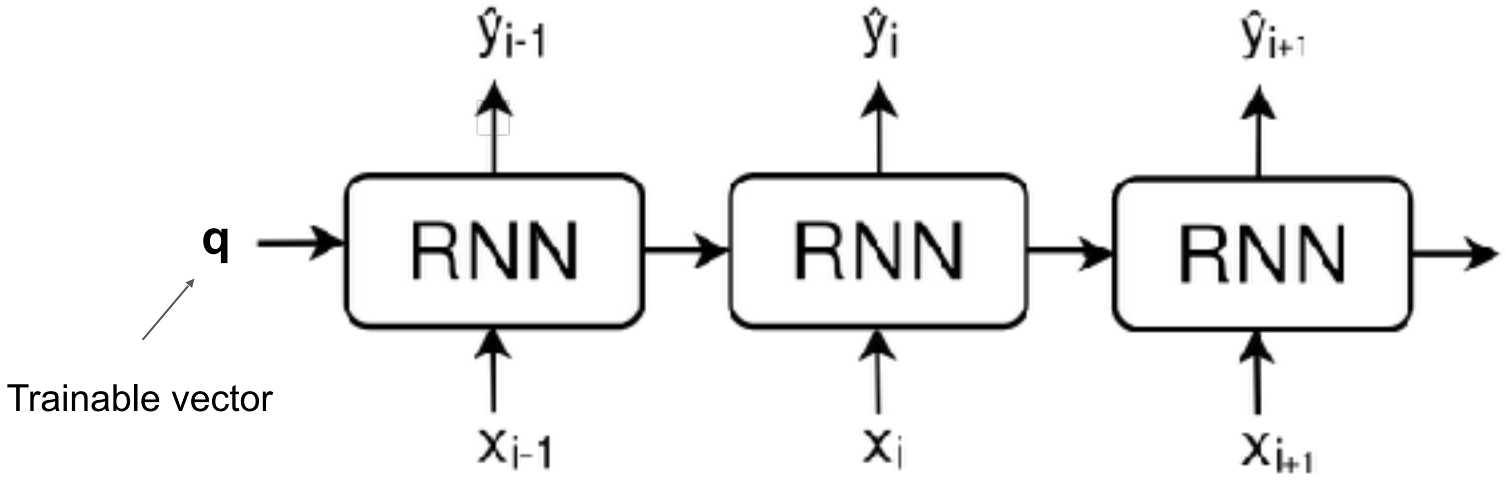
$$h_i = f_h(Wx_i + Vh_{i-1} + b_h) \qquad \hat{y}_i = f_y(Uh_i + b_y)$$

Recurrent neural network initial states



$$h_i = f_h(Wx_i + Vh_{i-1} + b_h) \qquad \hat{y}_i = f_y(Uh_i + b_y)$$

Recurrent neural network initial states

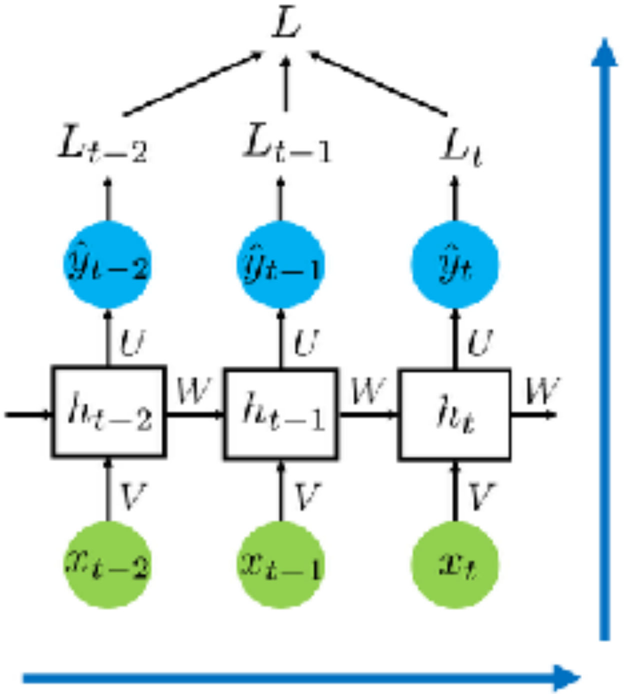


$$h_i = f_h(Wx_i + Vh_{i-1} + b_h) \qquad \hat{y}_i = f_y(Uh_i + b_y)$$

Backpropagation through time

Forward pass:

$h_t,\ \hat{y}_t,\ L_t,\ L$



Ekaterina Lobacheva. DeepBayes RNN presentation

Backpropagation through time

Forward pass:

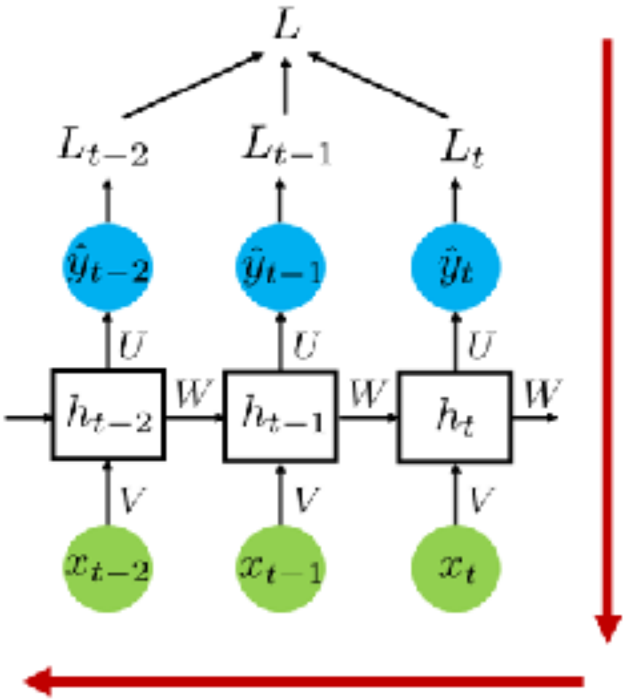
$$h_t,\ \hat{y}_t,\ L_t,\ L$$

Backward pass:

$$\frac{\partial L}{\partial U}, \frac{\partial L}{\partial V}, \frac{\partial L}{\partial W},$$

$$\frac{\partial L}{\partial b_x}, \frac{\partial L}{\partial b_h}$$

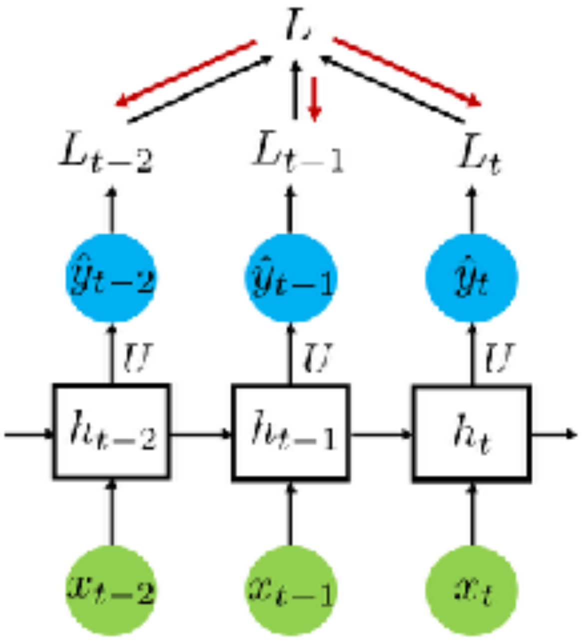
We backpropagate through layers and time



Ekaterina Lobacheva. DeepBayes RNN presentation

Backpropagation through time

$$\frac{\partial L}{\partial U} = \sum_{i=0}^{T} \frac{\partial L_i}{\partial U}$$



Ekaterina Lobacheva. [DeepBayes RNN presentation]

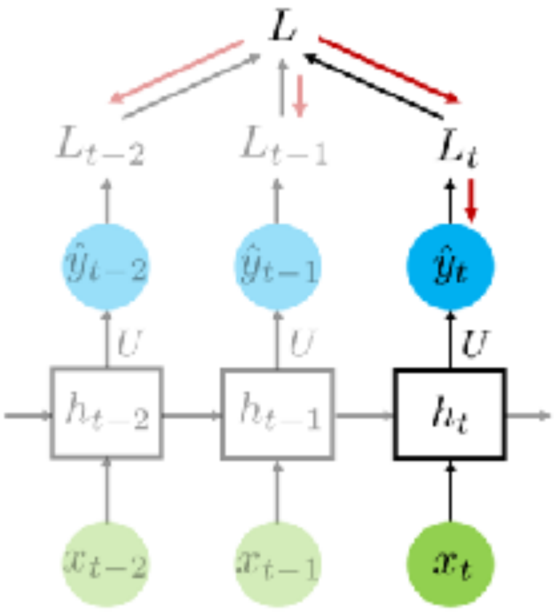
Backpropagation through time

$$\frac{\partial L}{\partial U} = \sum_{i=0}^{T} \frac{\partial L_i}{\partial U}$$

$$\frac{\partial L_t}{\partial U} = \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial U}$$

$$\hat{y}_t = f_y(\boxed{U} h_t + b_y)$$

this is the only dependence

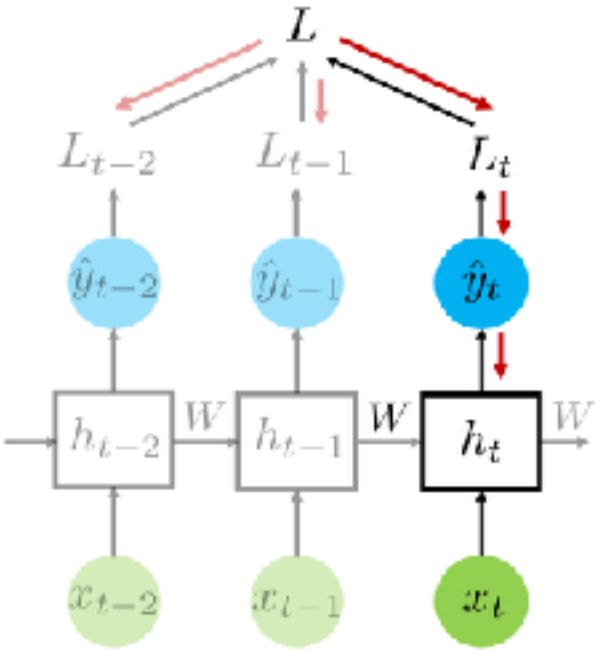


Ekaterina Lobacheva. [DeepBayes RNN presentation]

Backpropagation through time

$$\frac{\partial L}{\partial W} = \sum_{i=0}^{T} \frac{\partial L_i}{\partial W}$$

$$\frac{\partial L_t}{\partial W} = \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_t} \frac{\partial h_t}{\partial W}$$



Ekaterina Lobacheva. DeepBayes RNN presentation

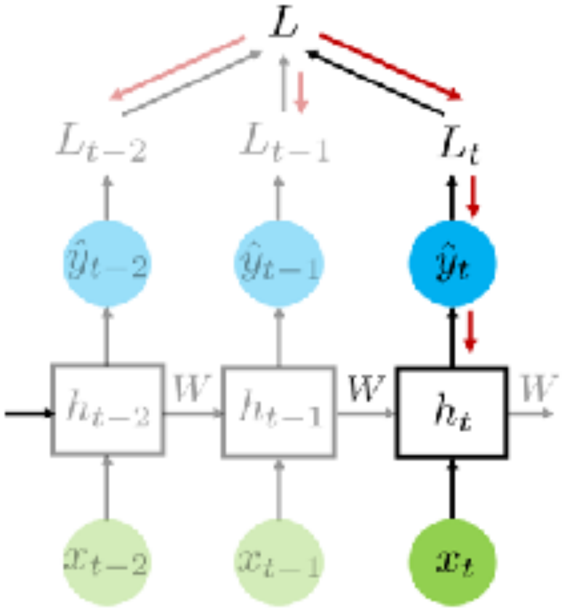
Backpropagation through time

$$\frac{\partial L}{\partial W} = \sum_{i=0}^{T} \frac{\partial L_i}{\partial W}$$

$$\frac{\partial L_t}{\partial W} = \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_t} \frac{\partial h_t}{\partial W}$$

$$h_t = f_h(Vx_t + \boxed{W} h_{t-1} + b_h)$$

This is NOT the only dependence!



Ekaterina Lobacheva. DeepBayes RNN presentation

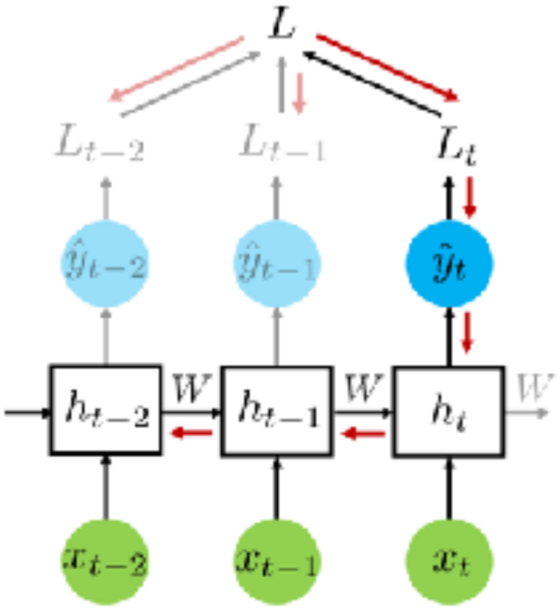
Backpropagation through time

$$\frac{\partial L}{\partial W} = \sum_{i=0}^{T} \frac{\partial L_i}{\partial W}$$

$$\frac{\partial L_t}{\partial W} = \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_t} \frac{\partial h_t}{\partial W}$$

$$h_t = f_h(V x_t + \boxed{W} h_{t-1} + b_h)$$

This is NOT the only dependence!



$$\frac{\partial L_t}{\partial W} = \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_t} \boxed{\left(\frac{\partial h_t}{\partial W} + \frac{\partial h_t}{\partial h_{t-1}} \frac{\partial h_{t-1}}{\partial W} + \dots \right)}$$

Ekaterina Lobacheva. DeepBayes RNN presentation

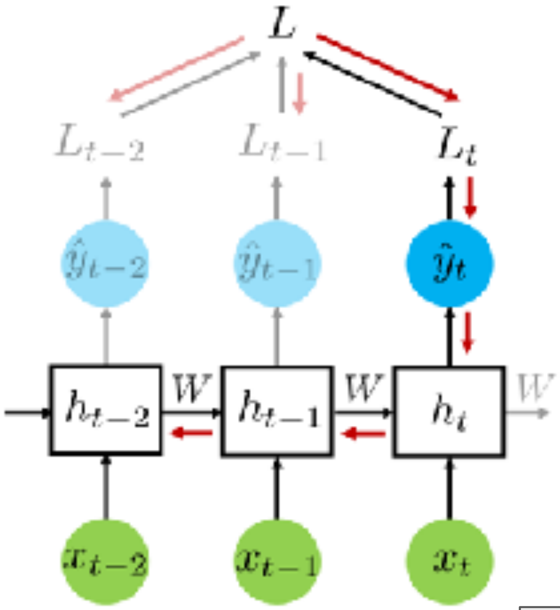
Backpropagation through time

$$\frac{\partial L}{\partial W} = \sum_{i=0}^{T} \frac{\partial L_i}{\partial W}$$

$$\frac{\partial L_t}{\partial W} = \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_t} \frac{\partial h_t}{\partial W}$$

$$h_t = f_h(V x_t + \boxed{W} h_{t-1} + b_h)$$

This is NOT the only dependence!



$$\frac{\partial L_t}{\partial W} = \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_t} \boxed{\left(\frac{\partial h_t}{\partial W} + \frac{\partial h_t}{\partial h_{t-1}} \frac{\partial h_{t-1}}{\partial W} + \dots \right)}$$

$$\boxed{f(x, y(x)) = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} \frac{\partial y}{\partial x}}$$

Ekaterina Lobacheva. [DeepBayes RNN presentation]

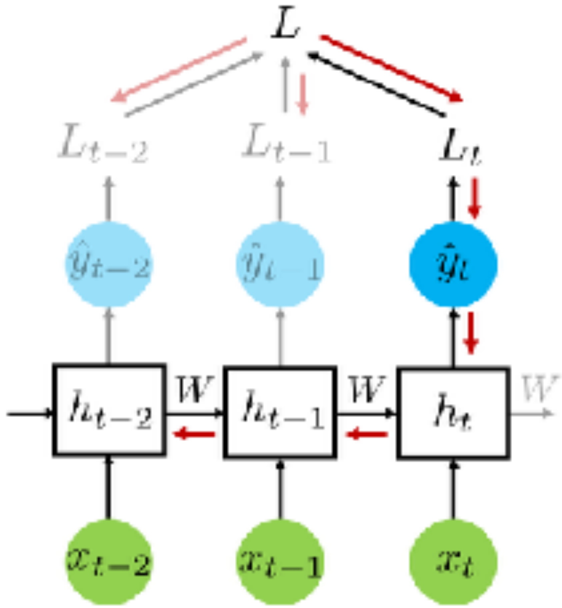
Backpropagation through time

$$\frac{\partial L}{\partial W} = \sum_{i=0}^{T} \frac{\partial L_i}{\partial W}$$

$$\frac{\partial L_t}{\partial W} = \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_t} \frac{\partial h_t}{\partial W}$$

$$h_t = f_h(V x_t + \boxed{W} h_{t-1} + b_h)$$

This is NOT the only dependence!



$$\frac{\partial L_t}{\partial W} = \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_t} \boxed{\sum_{k=0}^{t} \left(\prod_{i=k+1}^{t} \frac{\partial h_i}{\partial h_{i-1}} \right) \frac{\partial h_k}{\partial W}}$$

Ekaterina Lobacheva. DeepBayes RNN presentation

Vanishing and exploding grads

$$\frac{\partial L_t}{\partial W} \propto \sum_{k=0}^{t} \left(\prod_{i=k+1}^{t} \frac{\partial h_i}{\partial h_{i-1}} \right) \frac{\partial h_k}{\partial W}$$

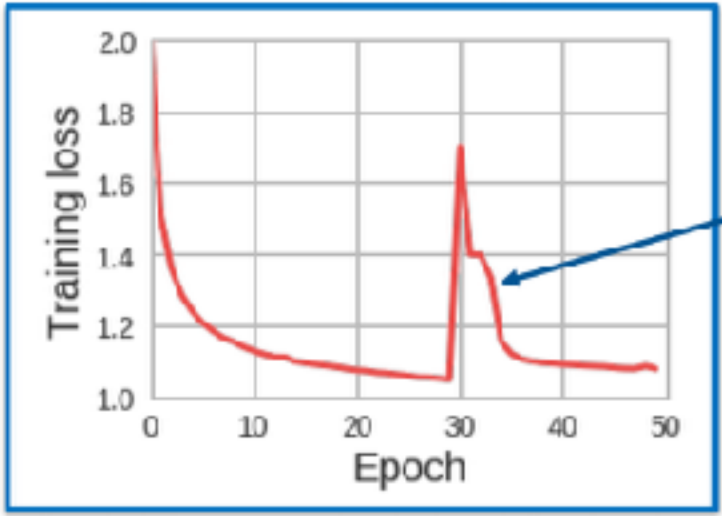
$$\left\| \frac{\partial h_i}{\partial h_{i-1}} \right\|_2 < 1 \Rightarrow \text{Vanishing gradients}$$

$$\left\| \frac{\partial h_i}{\partial h_{i-1}} \right\|_2 > 1 \Rightarrow \text{Exploding gradients}$$

Ekaterina Lobacheva. DeepBayes RNN presentation

Exploding gradients: detection

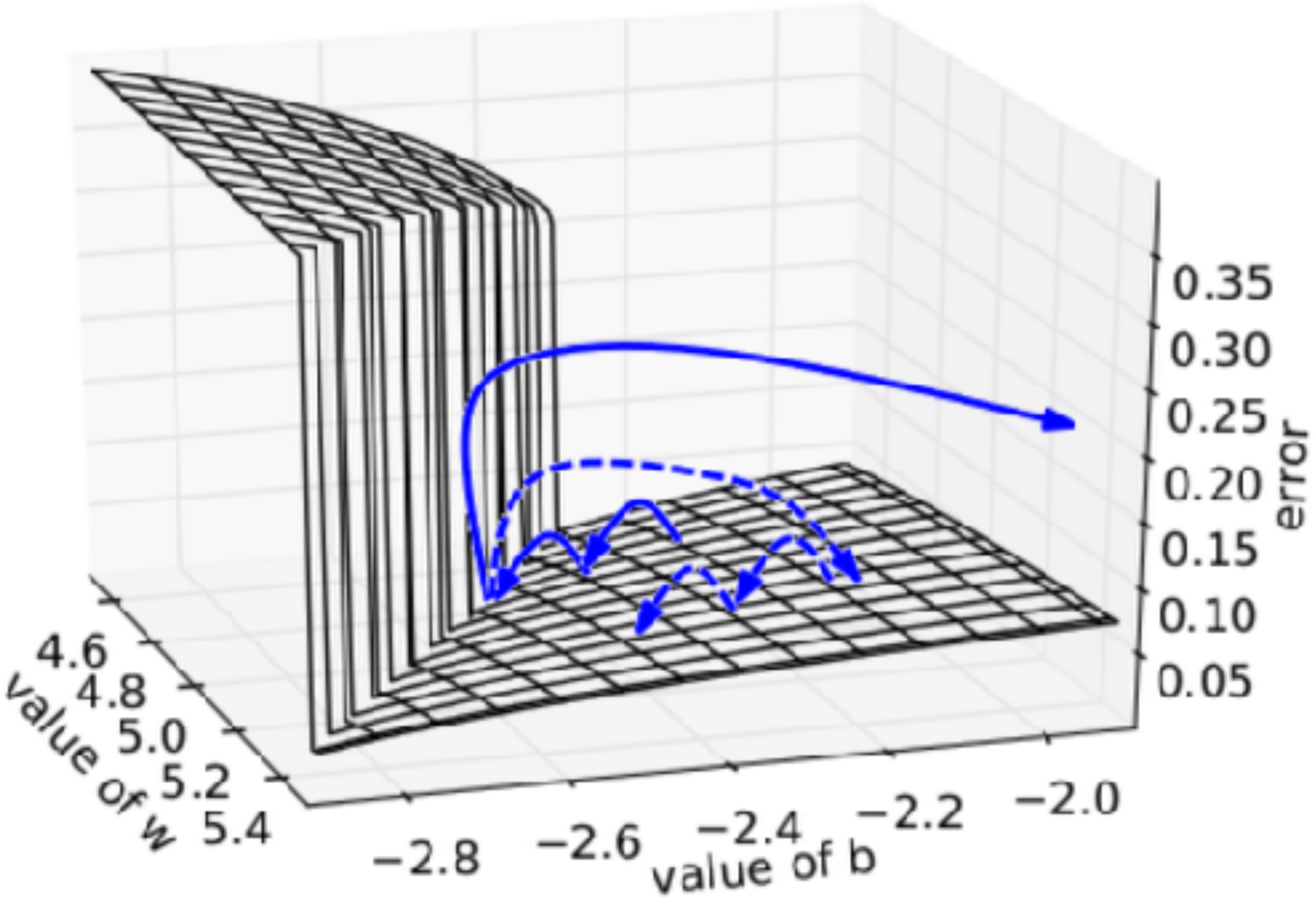
Unstable learning curve



This is it!

If the gradients contain NaNs you end up with NaNs in the weights

Ekaterina Lobacheva. [DeepBayes RNN presentation]



Ekaterina Lobacheva. [DeepBayes RNN presentation]

Gradient clipping

Gradient $g = \frac{\partial L}{\partial \theta}$, θ - all the network parameters

If $||g||>$ threshold:

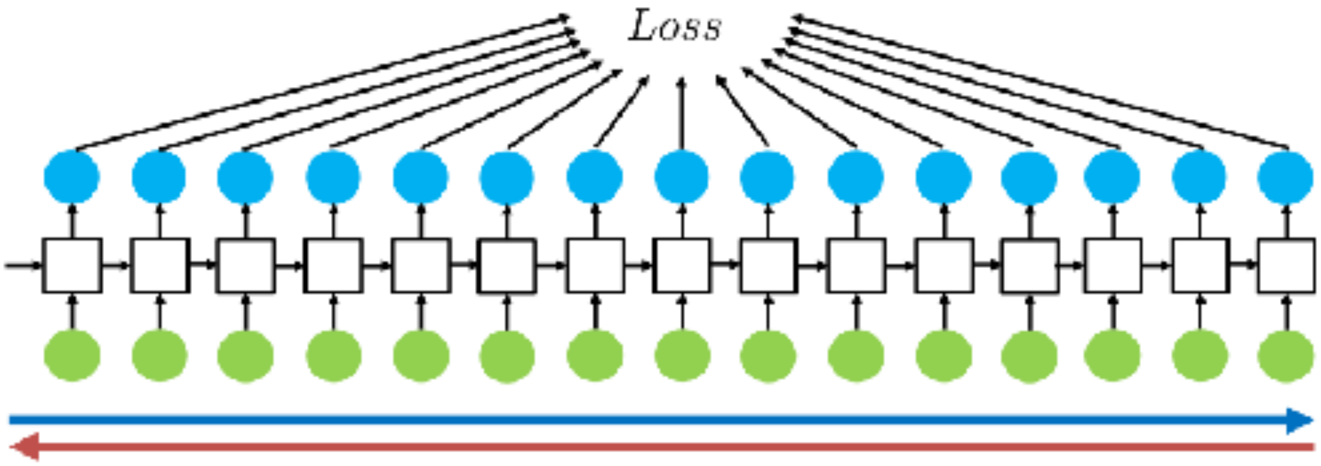
$$g \leftarrow \frac{threshold}{||g||} g$$

Simple but still very effective!

[Pascanu et al., 2012]

Ekaterina Lobacheva. DeepBayes RNN presentation

Truncated BPTT

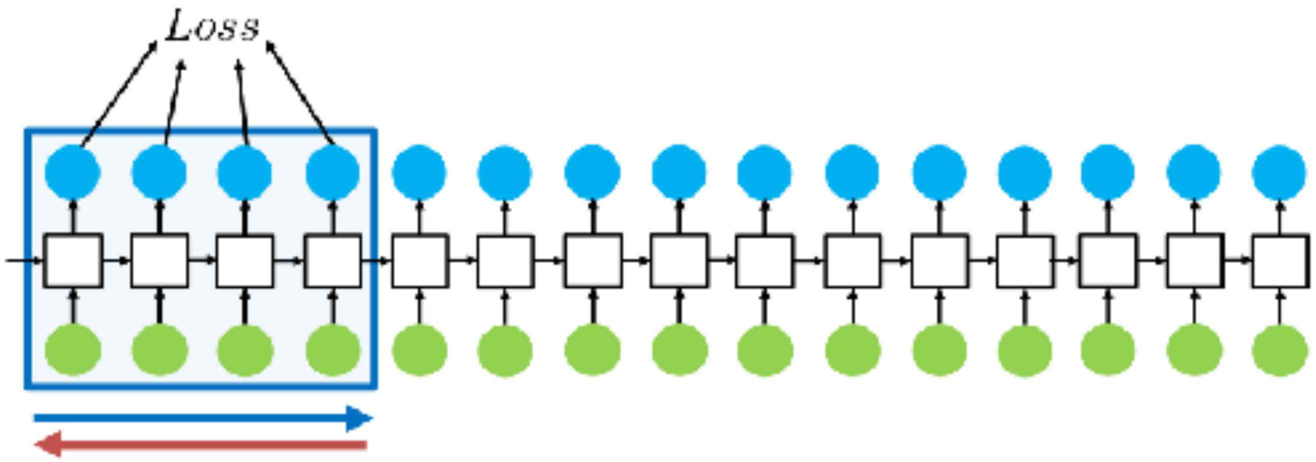


Forward pass through the entire sequence to compute the loss

Backward pass through the entire sequence to compute the gradient

Ekaterina Lobacheva. DeepBayes RNN presentation

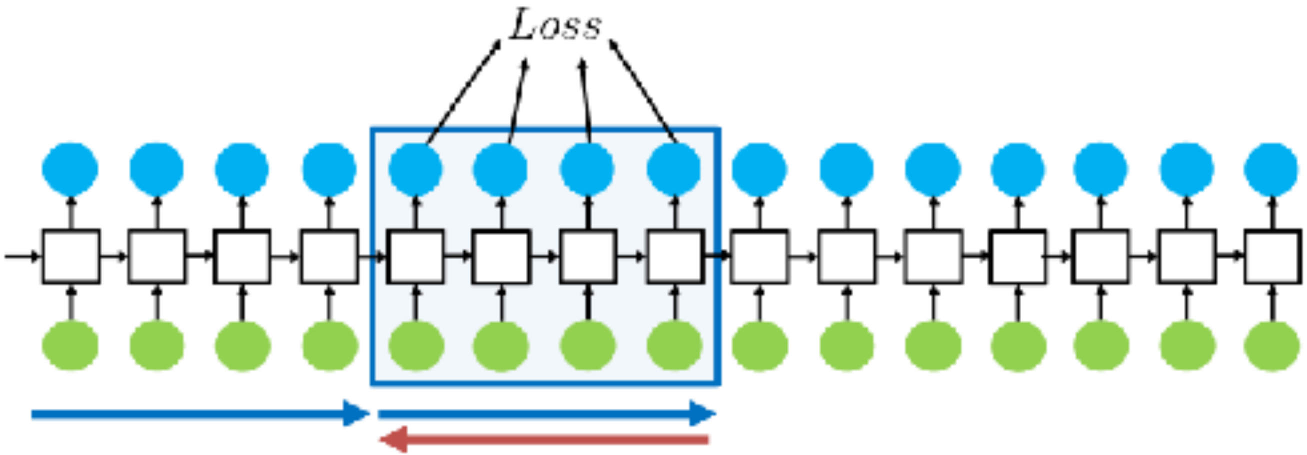
Truncated BPTT



Carry hidden states forward in time forever, but only backpropagate for some smaller number of steps.

Ekaterina Lobacheva. [DeepBayes RNN presentation]()

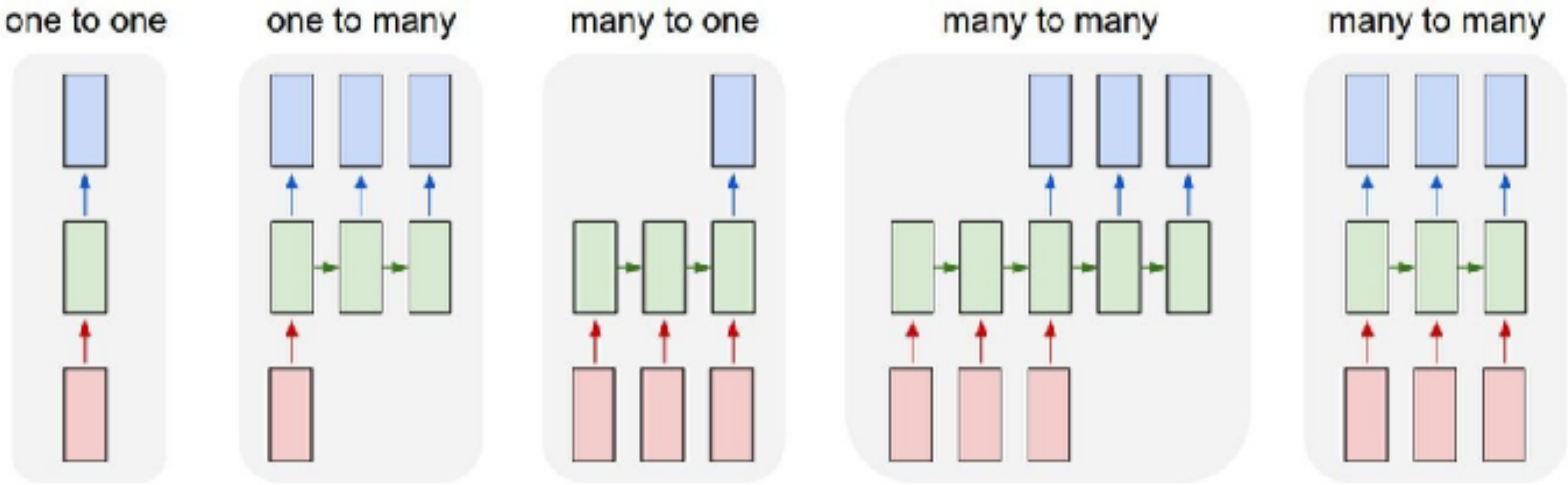
Truncated BPTT



Truncated BPTT is much faster but it doesn't come without a price! Dependencies longer than the chunk size don't affect the training but at least they still work at forward pass.

Ekaterina Lobacheva. [DeepBayes RNN presentation]

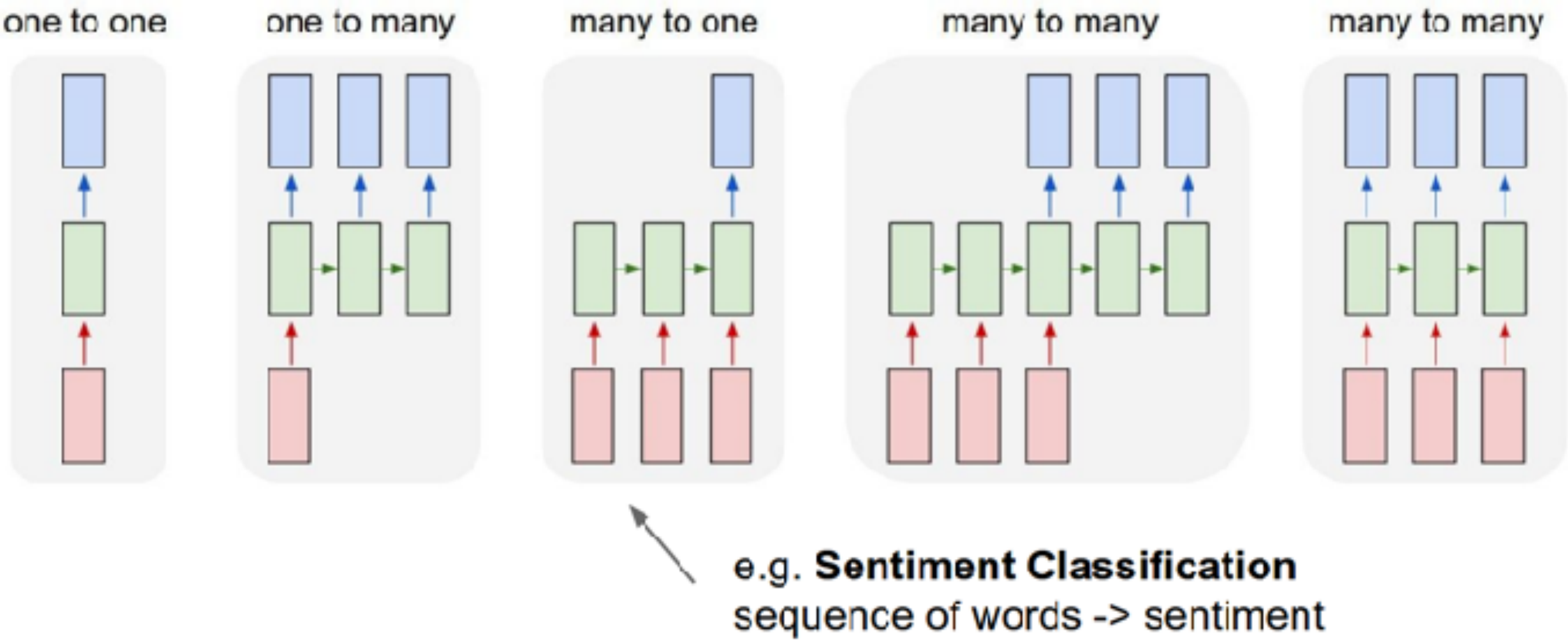
Types of tasks



e.g. **Image Captioning**
image -> sequence of words

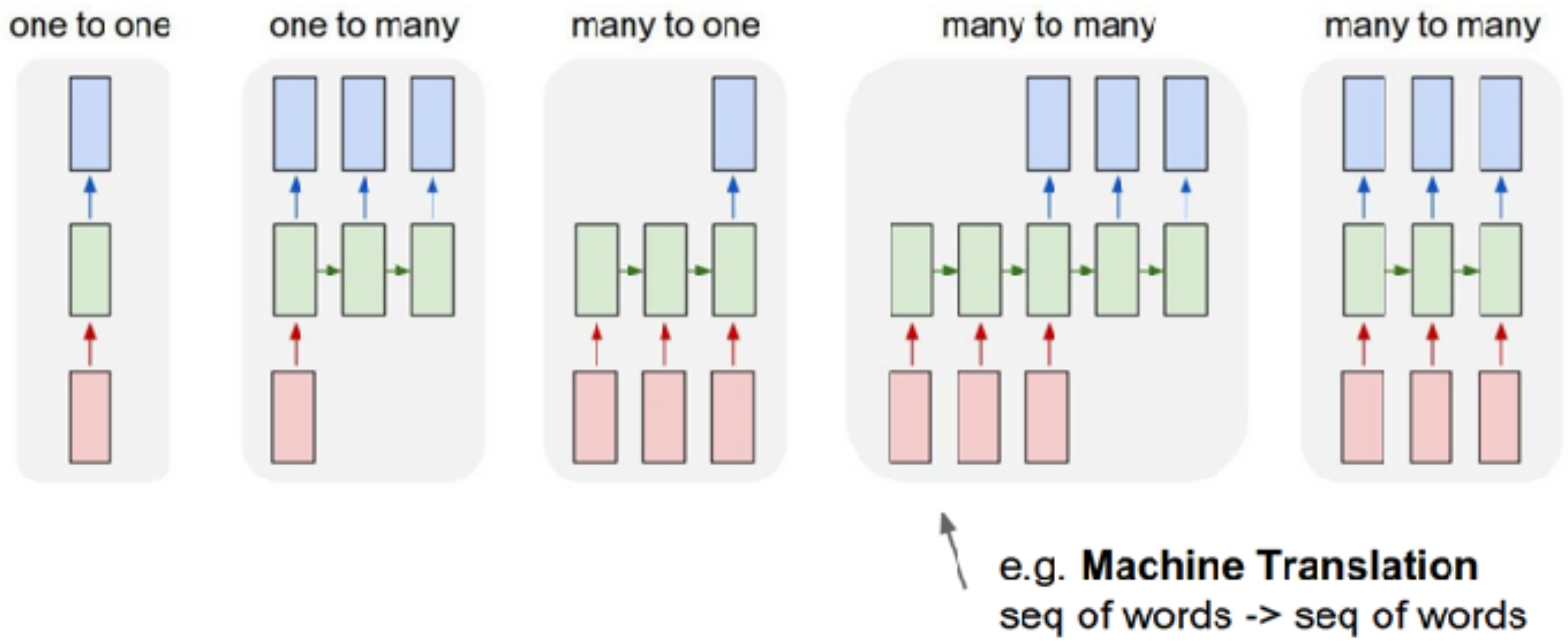
Stanford [CS234n]

Types of tasks



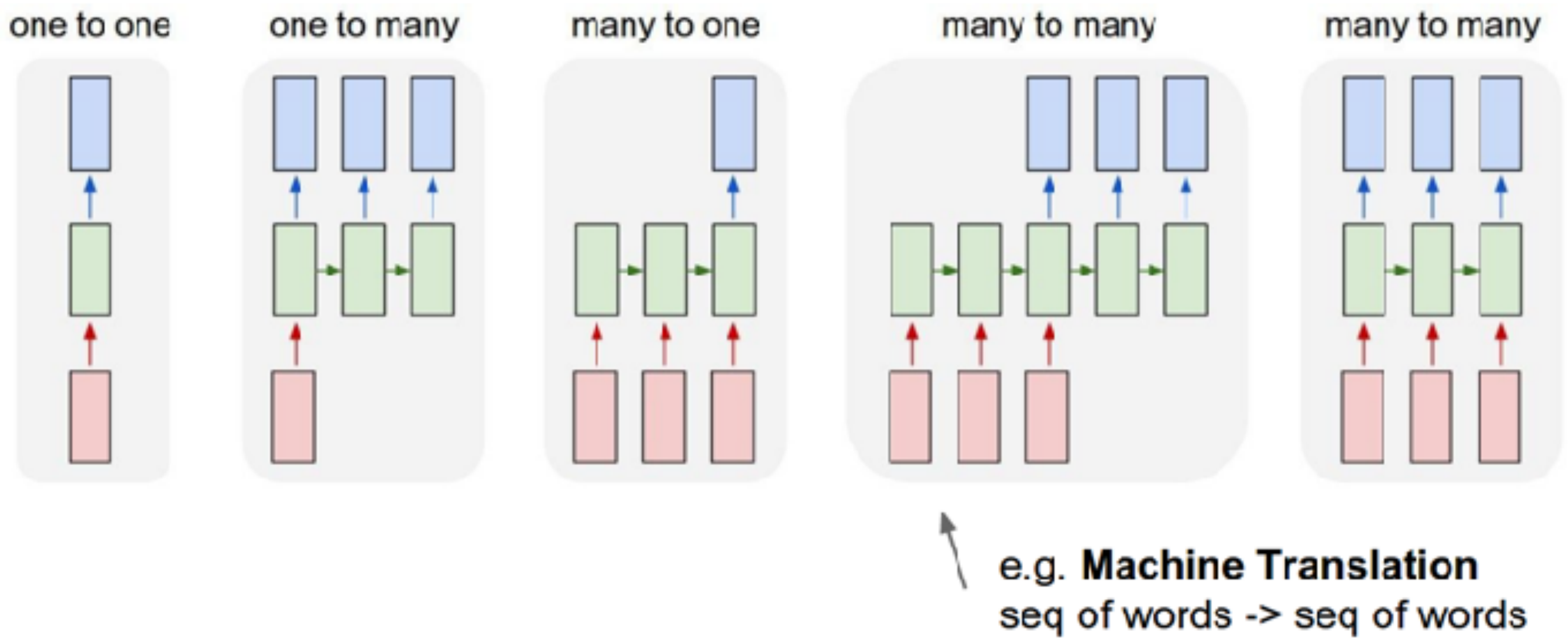
Stanford CS234n

Types of tasks



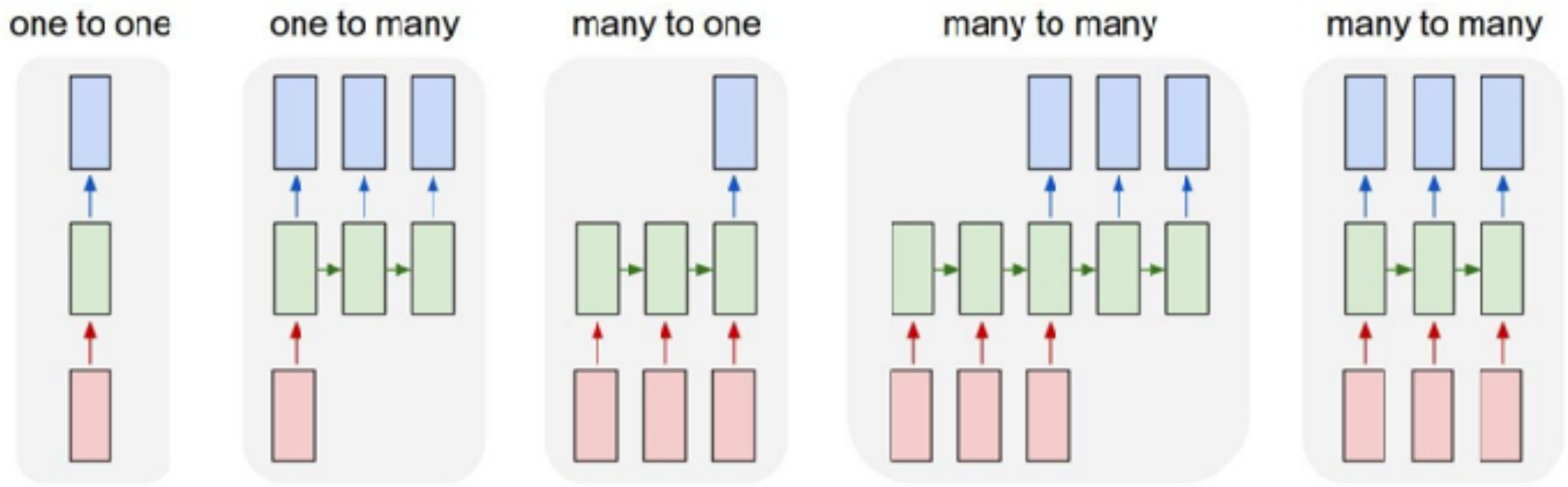
Stanford CS234n

Types of tasks



Stanford [CS234n]

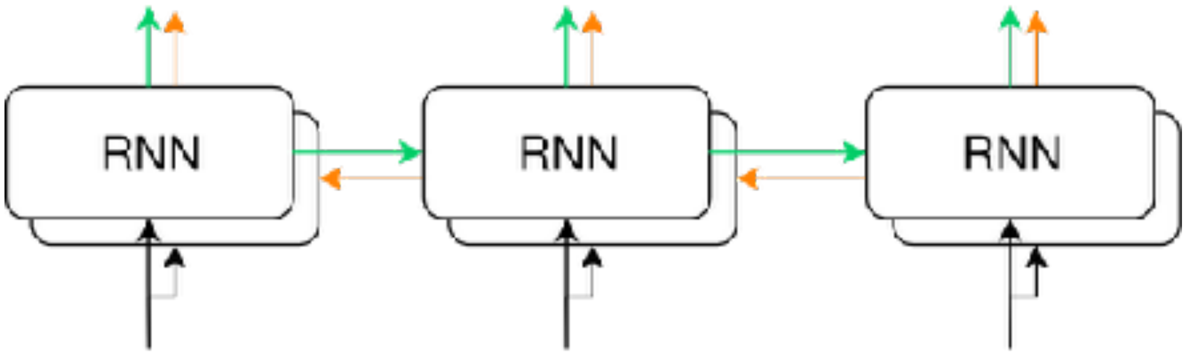
Types of tasks



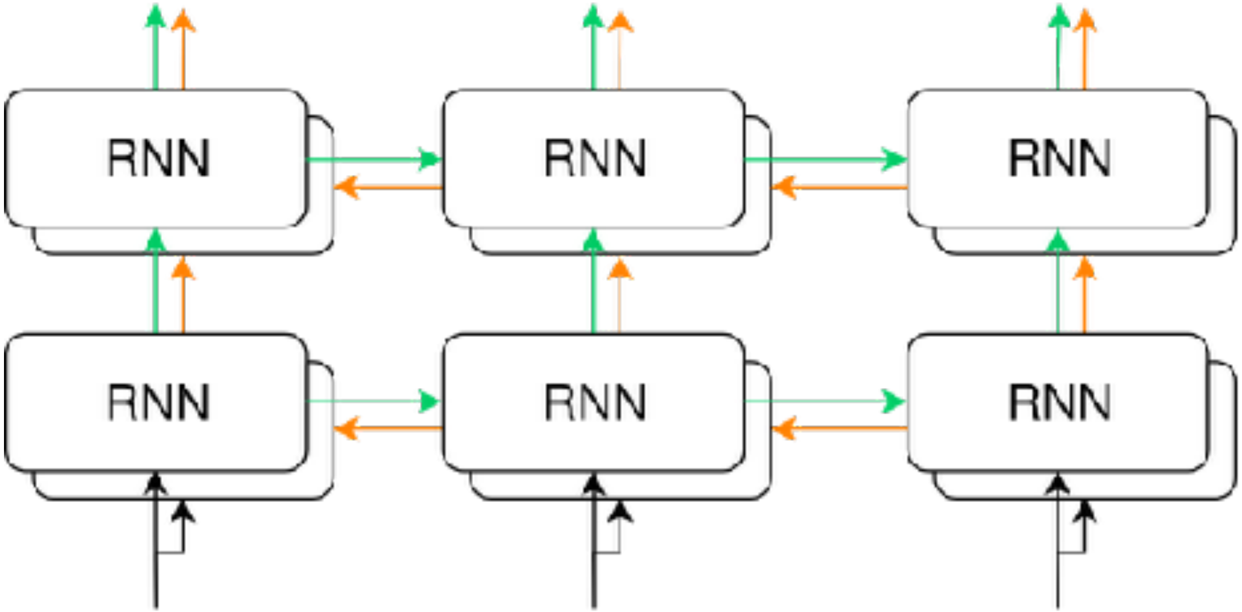
e.g. **POS tagging**,
sequence of tokens to sequence of tags

Stanford CS234n

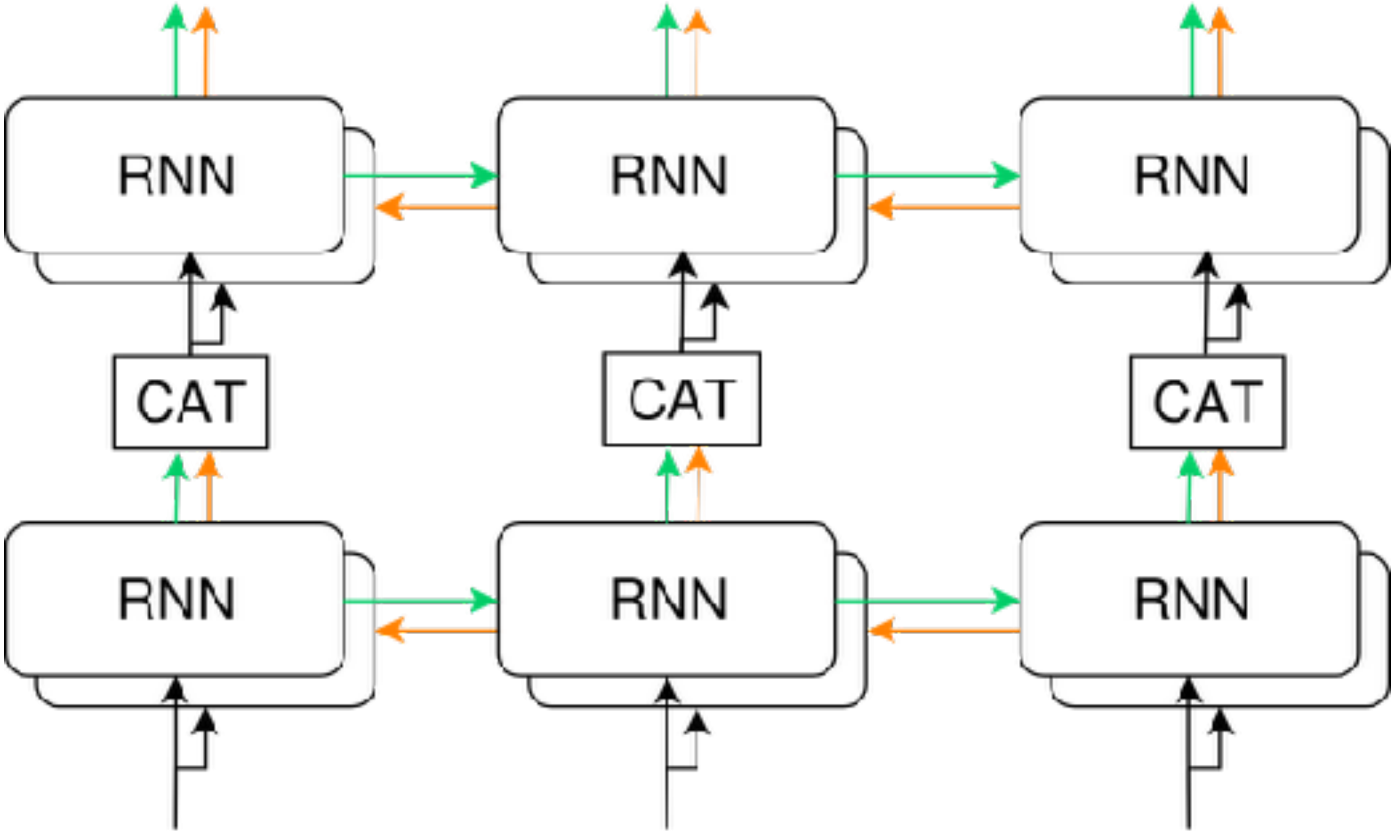
Bi-Directional RNN



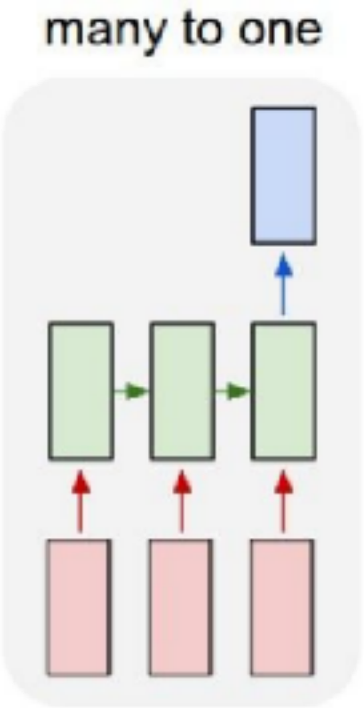
Stacked Bi-Directional RNN

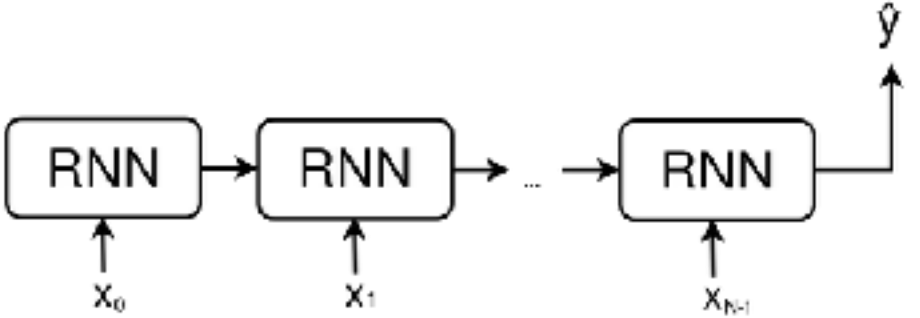


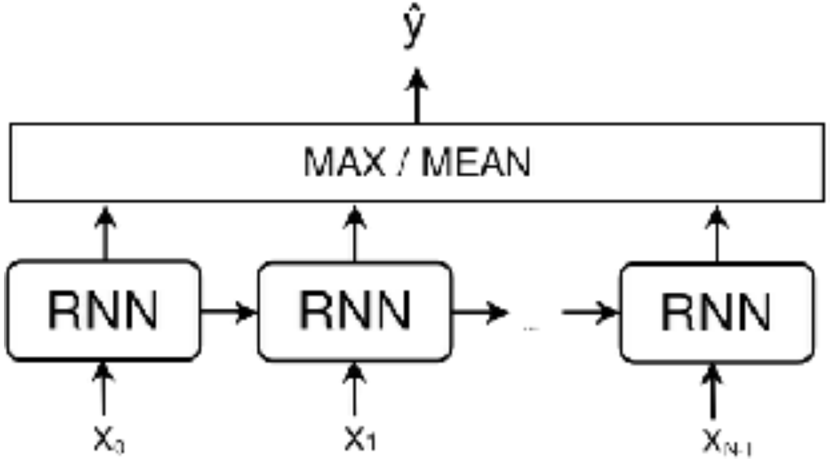
Stacked Bi-Directional RNN with Concatenation



Aggregation methods







Spasibo