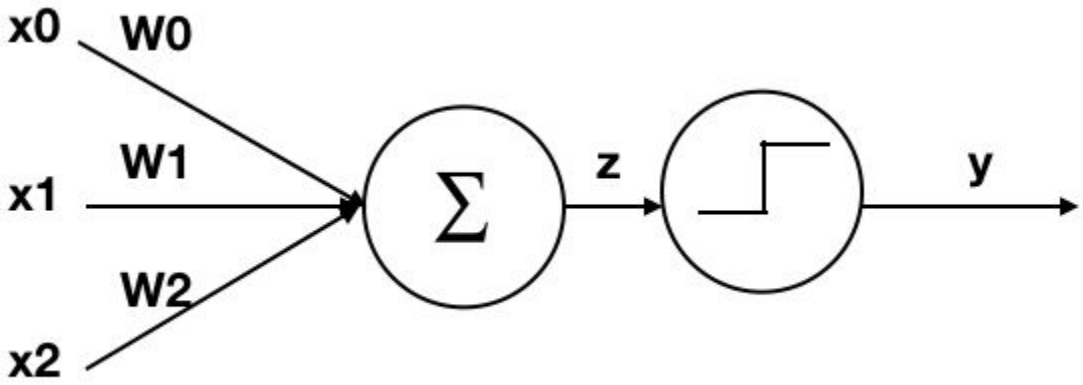
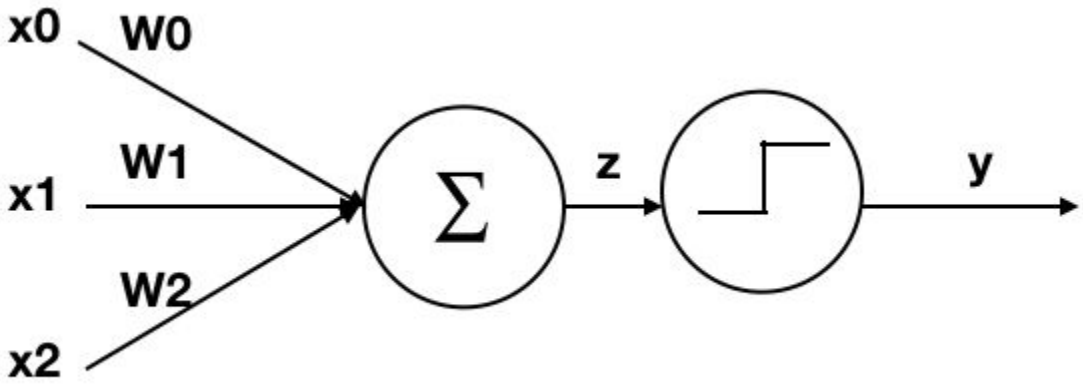
RNN

Perceptron

Perceptron

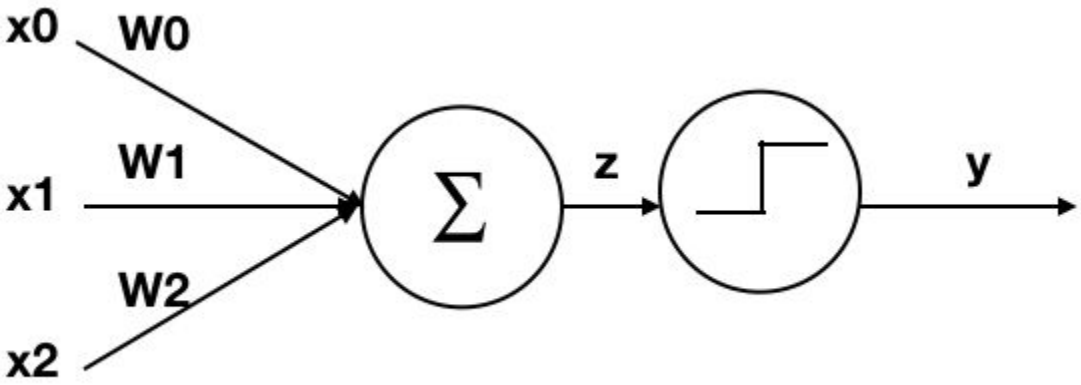


Perceptron



$$z = \sum_{i=1}^{n} W_i x_i + b = W^T x + b$$

Perceptron



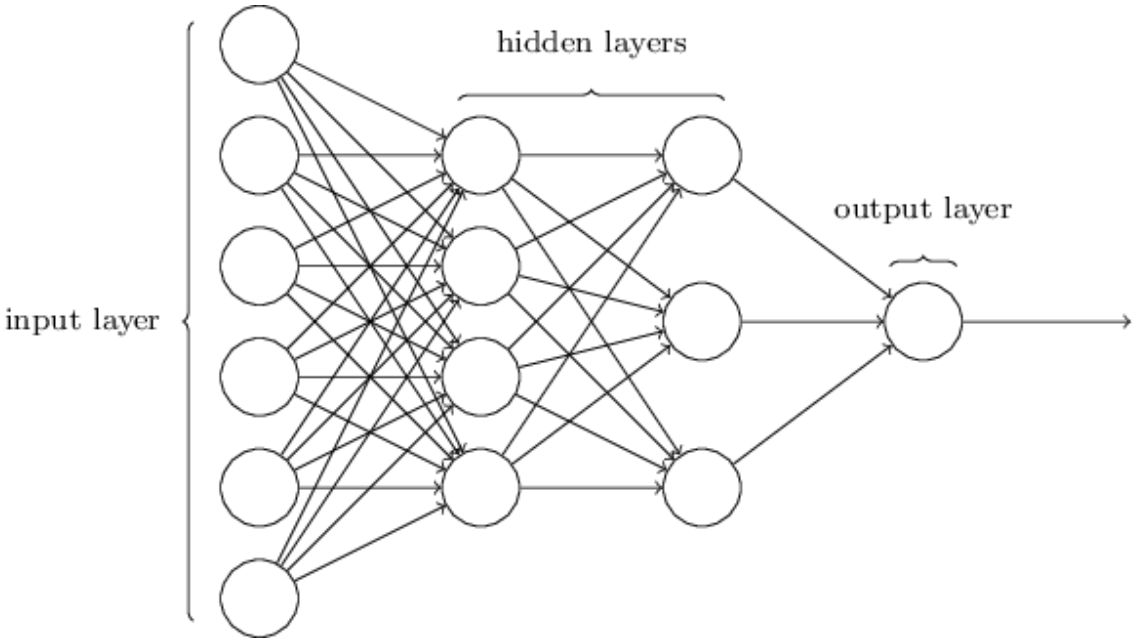
$$f(x) = \begin{cases} 1 \; if \; W^T x + b > 0 \\ 0 \; otherwise \end{cases}$$

FFNN

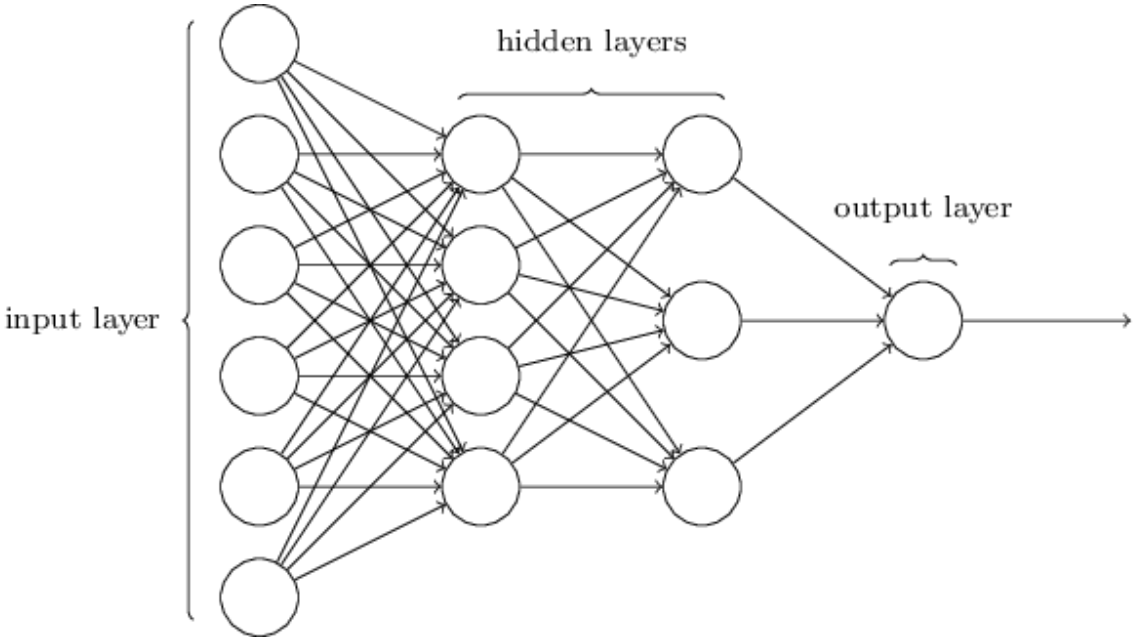
Feedforward neural networks (FFNNs)

- Supervised learning
- Labeled training set $D = \{(x_i, y_i) | i = 1, \ldots, n\}$
- Input x_i and output y_i
- Minimize training error by (stochastic) gradient descent

Feedforward neural networks (FFNNs)



Feedforward neural networks (FFNNs)



$$y_k(\mathbf{x}, \mathbf{w}) = \sigma\left(\sum_{j=0}^{M} w_{kj}^{(2)} \underbrace{f\left(\sum_{i=0}^{D} w_{ji}^{(1)} x_i\right)}_{z_j}\right)$$

RNN

Architecture

FFNN like CNN needs a determined input size to process.

RNNs are designed to take a series of input with no predetermined limit on size.

Single input item from the series is related to others and likely has an influence on its neighbors.

We need something that captures this relationship across inputs meaningfully.

Architecture

FFNN like CNN needs a determined input size to process.

Architecture

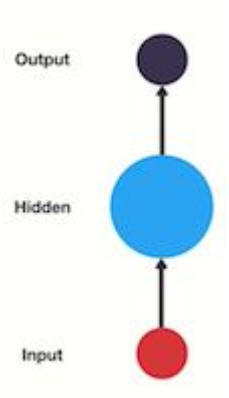
FFNN like CNN needs a determined input size to process.

Architecture

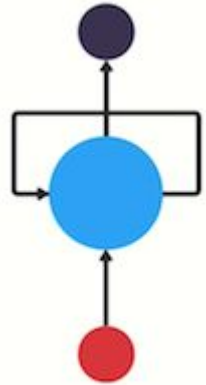
RNNs are designed to take a series of input with no predetermined limit on size.

Architecture

RNNs are designed to take a series of input with no predetermined limit on size.



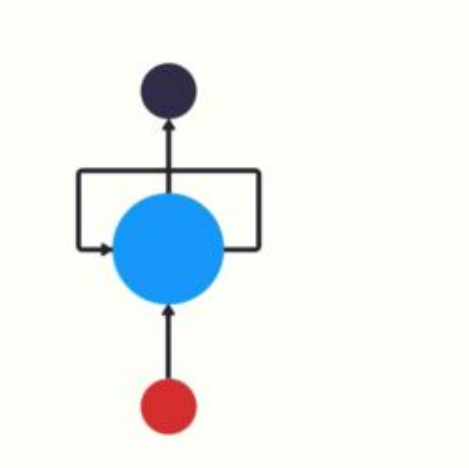
FFNN



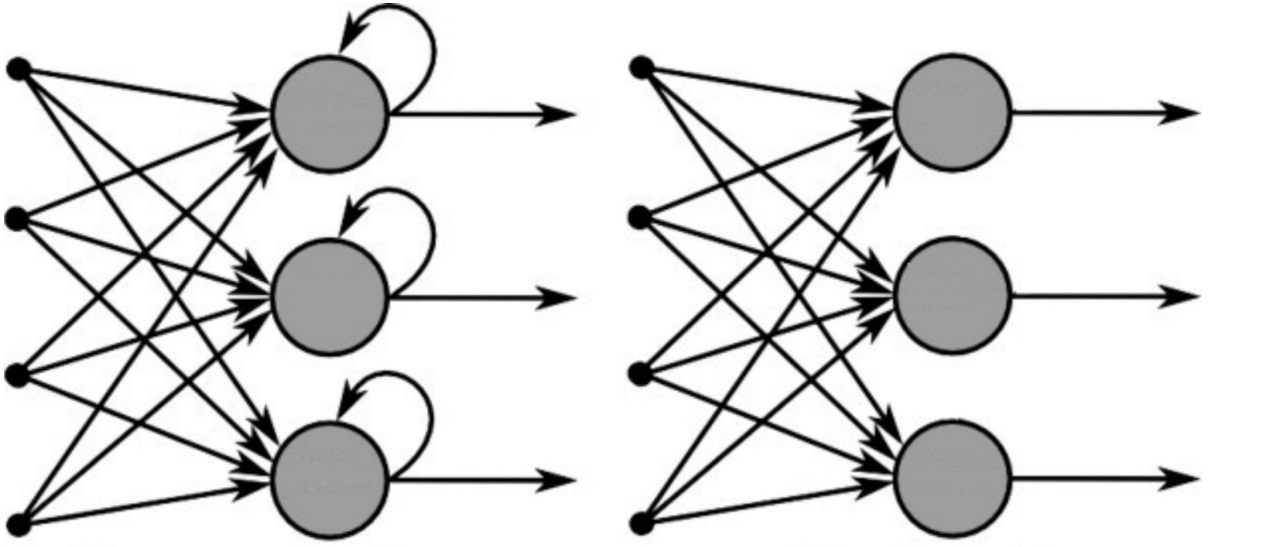
RNN

Architecture

RNNs are designed to take a series of input with no predetermined limit on size.



Architecture

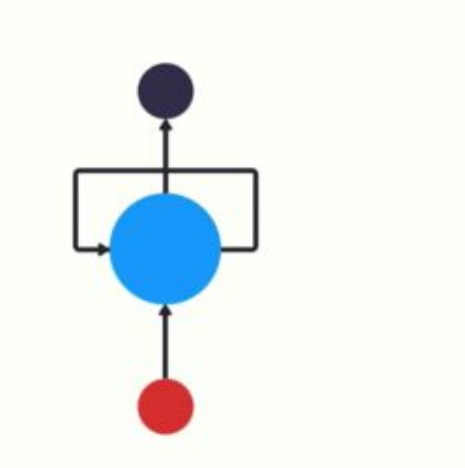


Recurrent Neural Network

Feed-Forward Neural Network

RNN

- RNNs use their internal state (memory) to process sequences of inputs
- They remember things learnt from prior inputs while generating outputs.
- “hidden” state vector representing the context based on prior input(s)/output(s).
- The same input could produce a different output depending on previous inputs in the series.



RNN

RNNs can use their internal state (memory) to process sequences of inputs

- Unsegmented, connected handwriting recognition

RNN

RNNs can use their internal state (memory) to process sequences of inputs

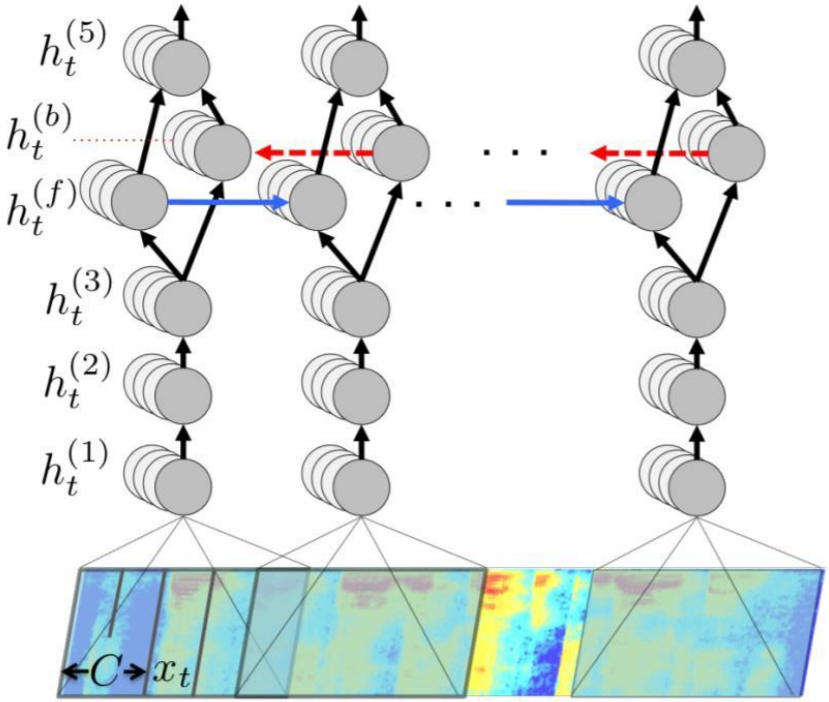
- Unsegmented, connected handwriting recognition
- Speech recognition

RNN

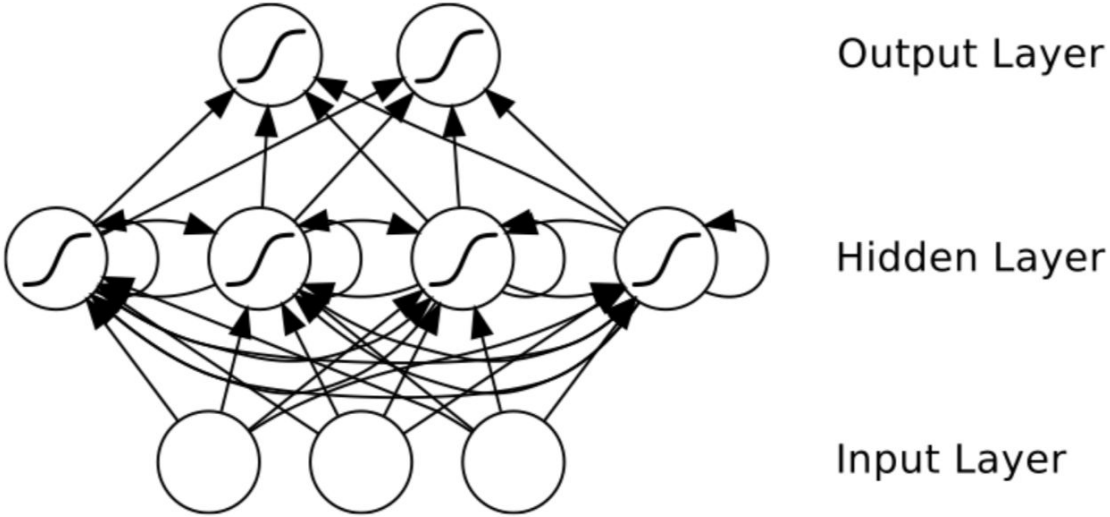
RNNs can use their internal state (memory) to process sequences of inputs

- Unsegmented, connected handwriting recognition
- Speech recognition
- Time Series

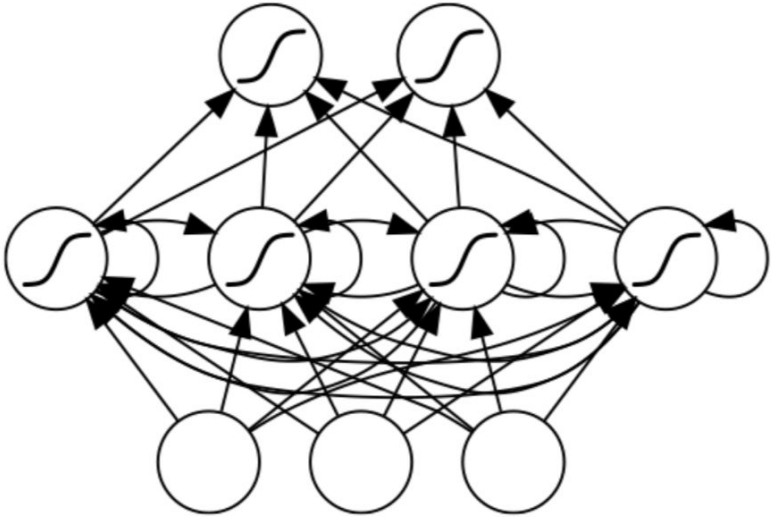
Deep Speech



RNN



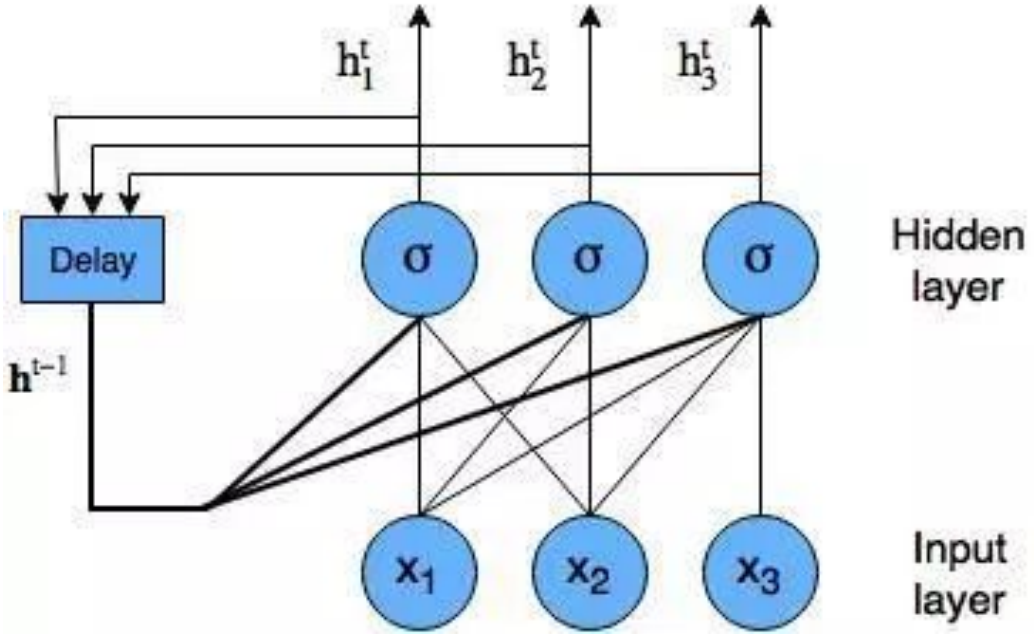
RNN



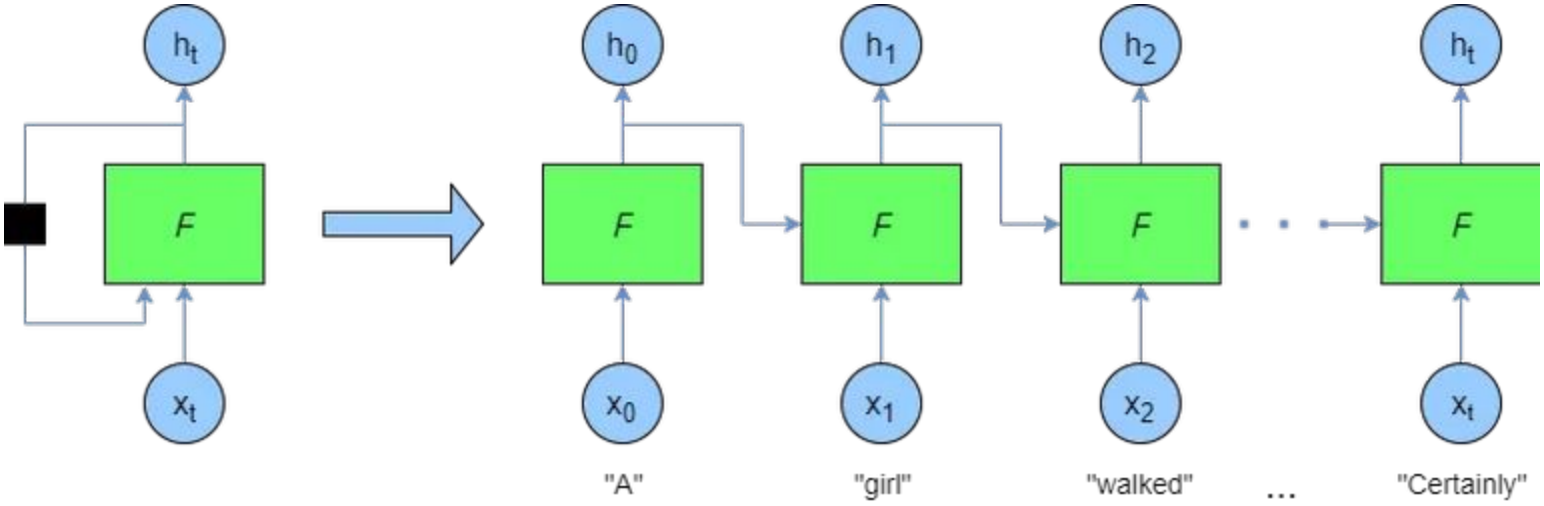
$$\mathbf{h}_t^{(1)} = f_1\left(W^{(1)}\mathbf{x}_t + W^{(\rightarrow)}\mathbf{h}_{t-1}^{(1)}\right)$$

$$\mathbf{h}_t^{(2)} = f_2\left(W^{(2)}\mathbf{h}_t^{(1)}\right)$$

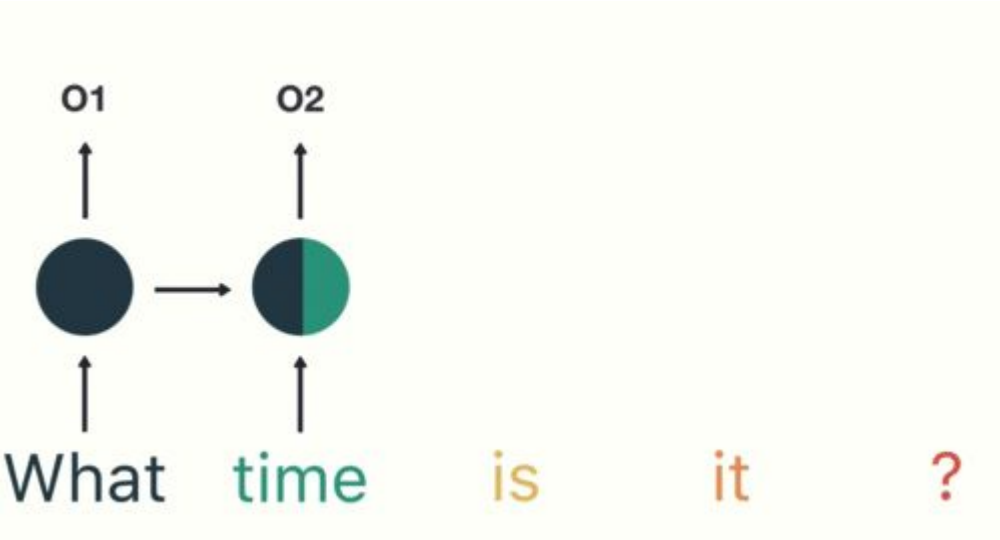
RNN



RNN

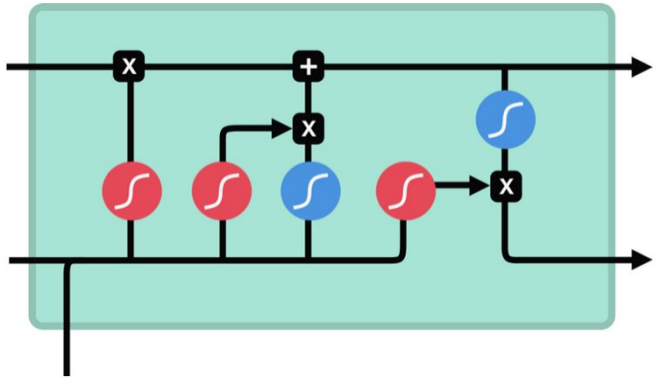


RNN

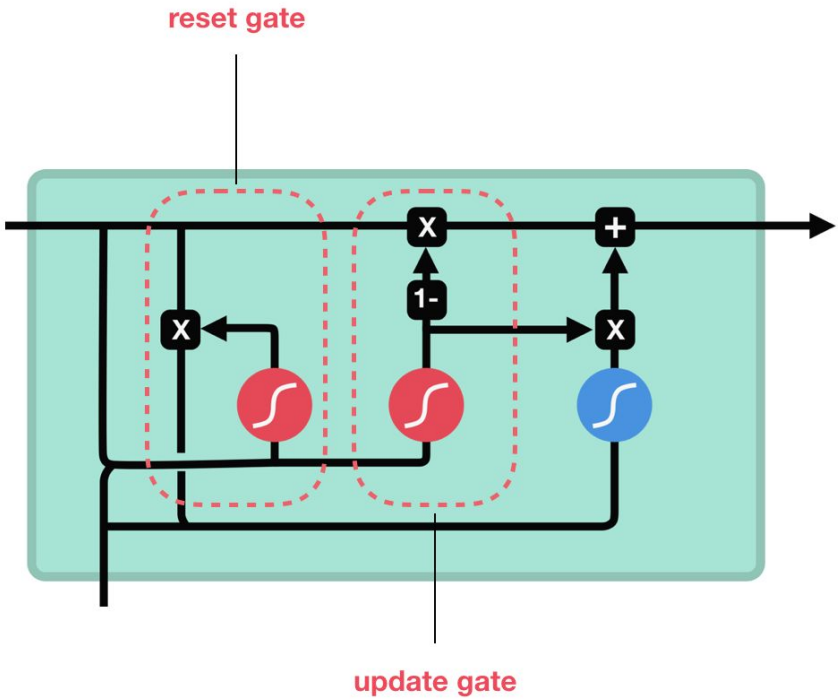


RNN - LSTM

- The Forget gate decides what is relevant to keep from prior steps.
- The input gate decides what information is relevant to add from the current step.
- The output gate determines what the next hidden state should be.

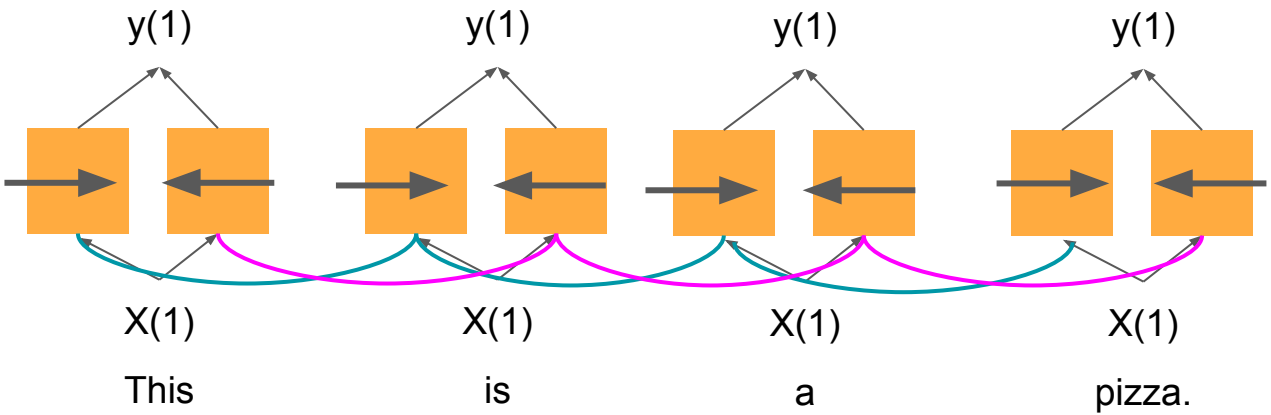


RNN - Gated Recurrent Unit

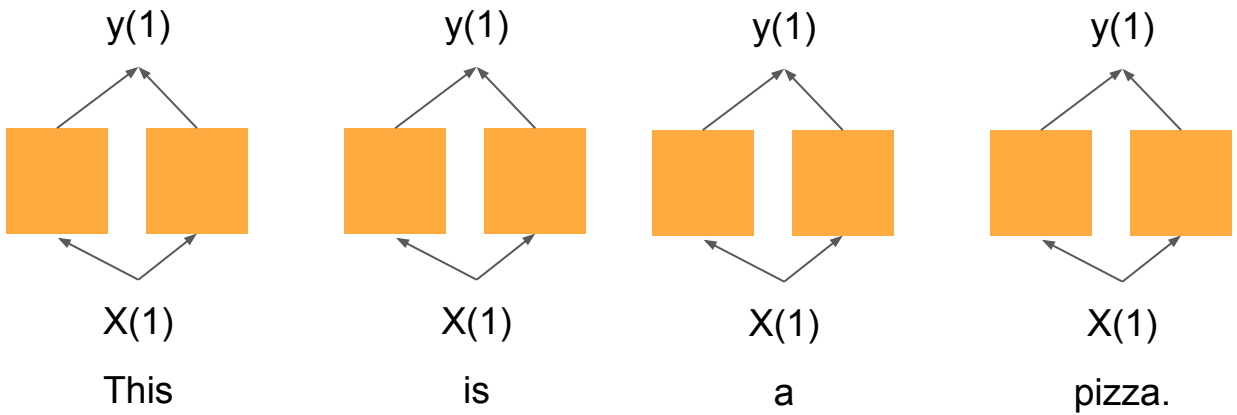


Attention

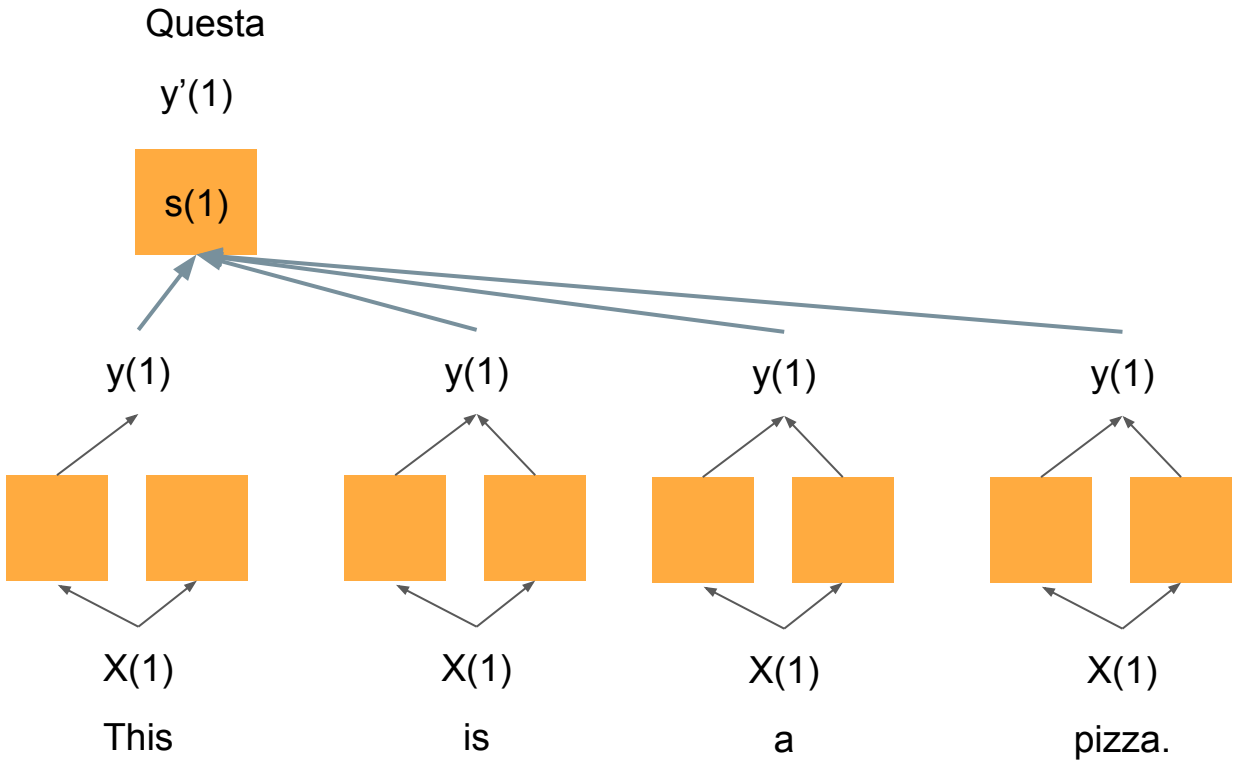
Machine Translation



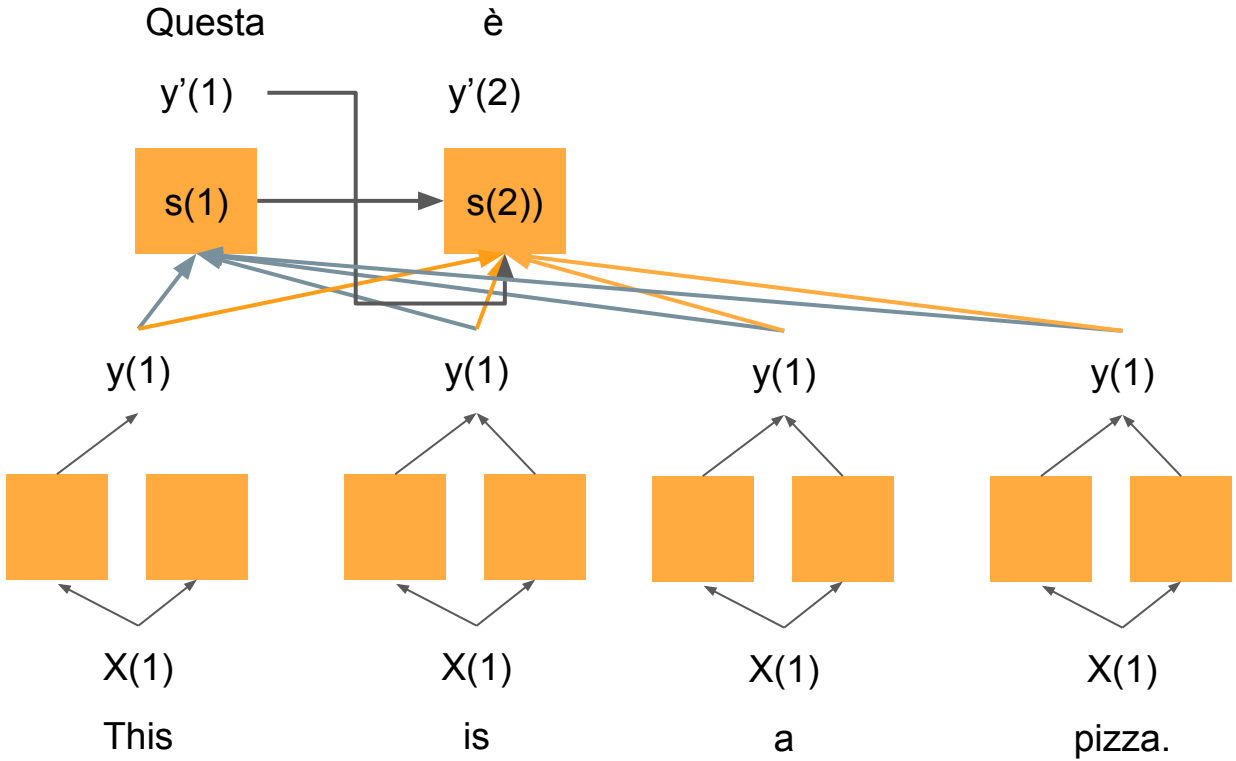
Machine Translation



Machine Translation



Machine Translation



Machine Translation