RECURRENT NEURAL NETWORK

SEQUENCE

x

y

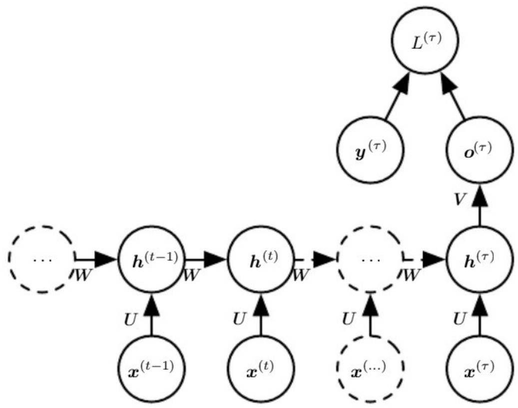
0

Bowling

https://www.crcv.ucf.edu/research/data-sets/ucf101/

RNN

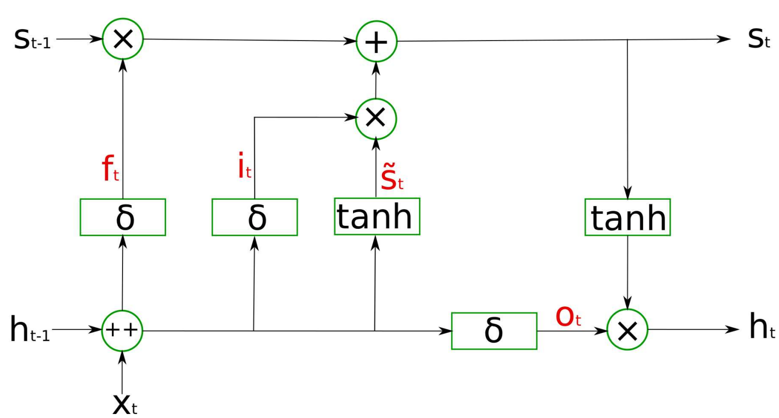
1. It is with recurrent connections specialized for processing sequential data



A1

LSTM

Long Short-Term Memory (LSTM) is a widely used type of RNN.



AARHUS UNIVERSITY
DEPARTMENT OF BIOSCIENCE
9 MAY 2019 | XIAODONG DUAN
POSTDOC

Slide 4

A1 Author; 13-05-2019

LSTM

$$\boldsymbol{f}_t = \delta(\boldsymbol{W}_f \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_f)$$

$$\boldsymbol{i}_t = \delta(\boldsymbol{W}_i \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_i)$$

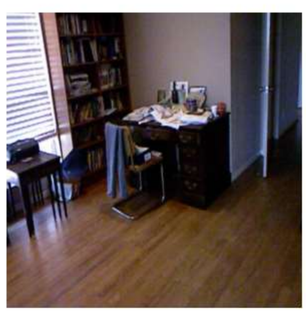
$$\boldsymbol{o}_t = \delta(\boldsymbol{W}_o \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_o)$$

$$\tilde{\boldsymbol{s}}_t = tanh(\boldsymbol{W}_s \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_s)$$

$$\boldsymbol{s}_t = \boldsymbol{f}_t \times \boldsymbol{s}_{t-1} + \boldsymbol{i}_t \times \tilde{\boldsymbol{s}}_t$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \times tanh(\boldsymbol{s}_t)$$

AN EXAMPLE



What is on the wall to the right side of the bookshelf?

https://www.mpi-inf.mpg.de/departments/computer-vision-and-multimodal-computing/research/vision-and-language/visual-turing-challenge/

AN EXAMPLE

What is the colour of the pillow cover?

https://www.mpi-inf.mpg.de/departments/computer-vision-and-multimodal-computing/research/vision-and-language/visual-turing-challenge/

CONVOLUTIONAL NEURAL NETWORK

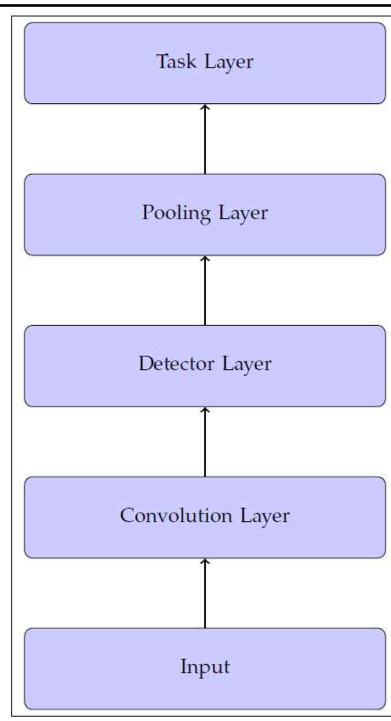
CNN

1. It is widely used in image processing.
2. The core is the two-dimensional convolution operation.

$$S(i,j) = \sum_{m=-M}^{M} \sum_{n=-N}^{N} I(i-m, j-n)K(m,n)$$

3. It usually consists of many layers
   - Residual Neural Network (ResNet) : 152 layers
   - He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

TYPICAL STRUCTURE



CONVOLUTION LAYER

Zero padding

Filters (Convolution Kernels)

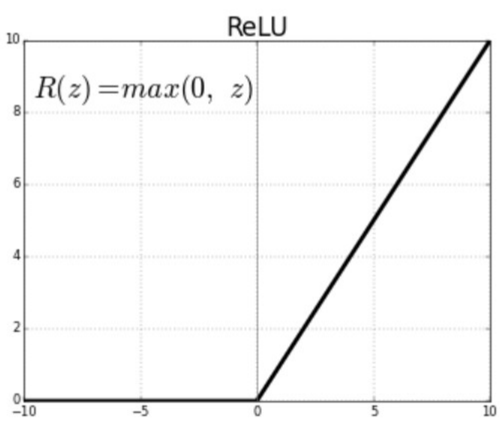
Identity $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

Edge detection $\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 0 & -1 \\ -1 & -1 & -1 \end{bmatrix}$

Strides

DETECTOR LAYER

Nonlinear activations
- Increase the non-linearity
- ReLU



POOLING

1. Reducing the spatial dimensions
2. Invariant to small translations of the input
3. Max pooling

$$\begin{bmatrix} 1 & 2 & 5 & 6 \\ 3 & 4 & 7 & 8 \\ 9 & 8 & 5 & 4 \\ 7 & 6 & 3 & 2 \end{bmatrix} \xrightarrow{\text{Max pool with } 2 \times 2 \text{ filter and stride 2}} \begin{bmatrix} 4 & 8 \\ 9 & 5 \end{bmatrix}$$

TASK LAYER

1. It depends on your application
2. Fully connected neural network

AARHUS UNIVERSITY
DEPARTMENT OF BIOSCIENCE
9 MAY 2019 | XIAODONG DUAN
POSTDOC

VGG

Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

CNN IN KERAS

1. Conv2D(): convolution + relu
2. MaxPooling2D()
3. Flatten()

AARHUS
UNIVERSITY