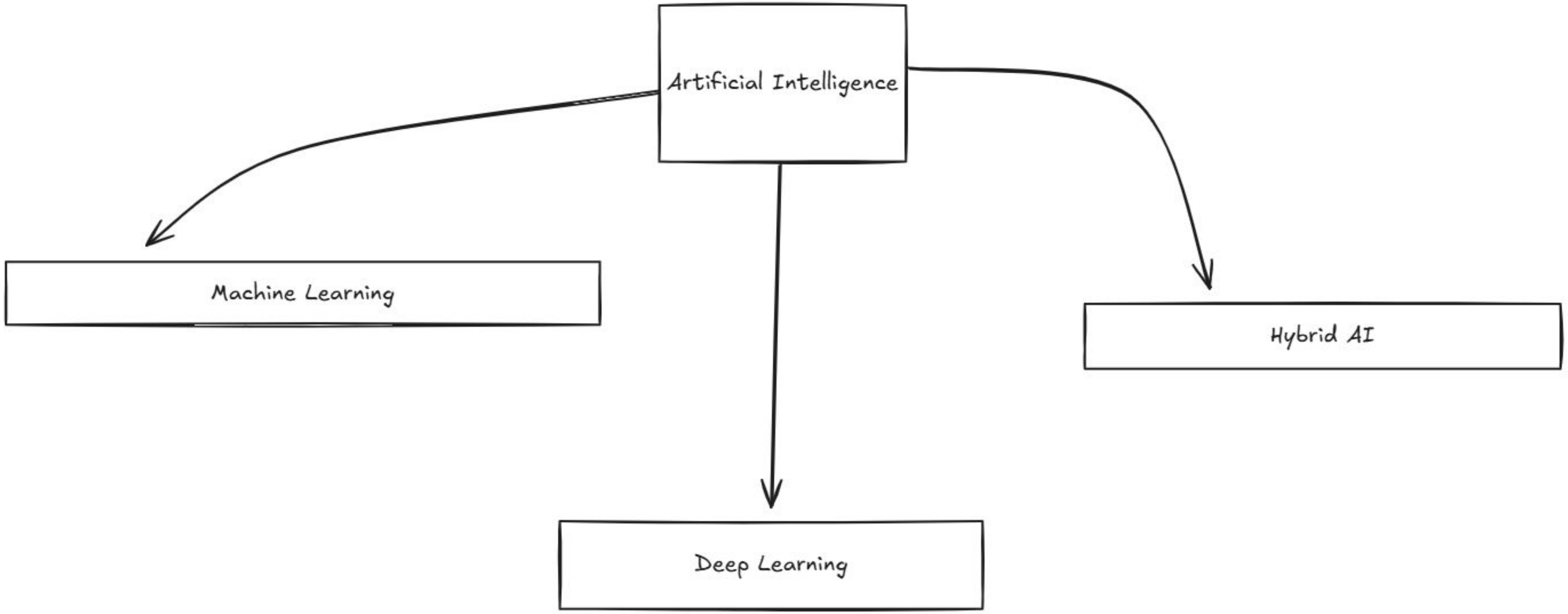
AI in Ramadan

Session 2 : Diving into abyss

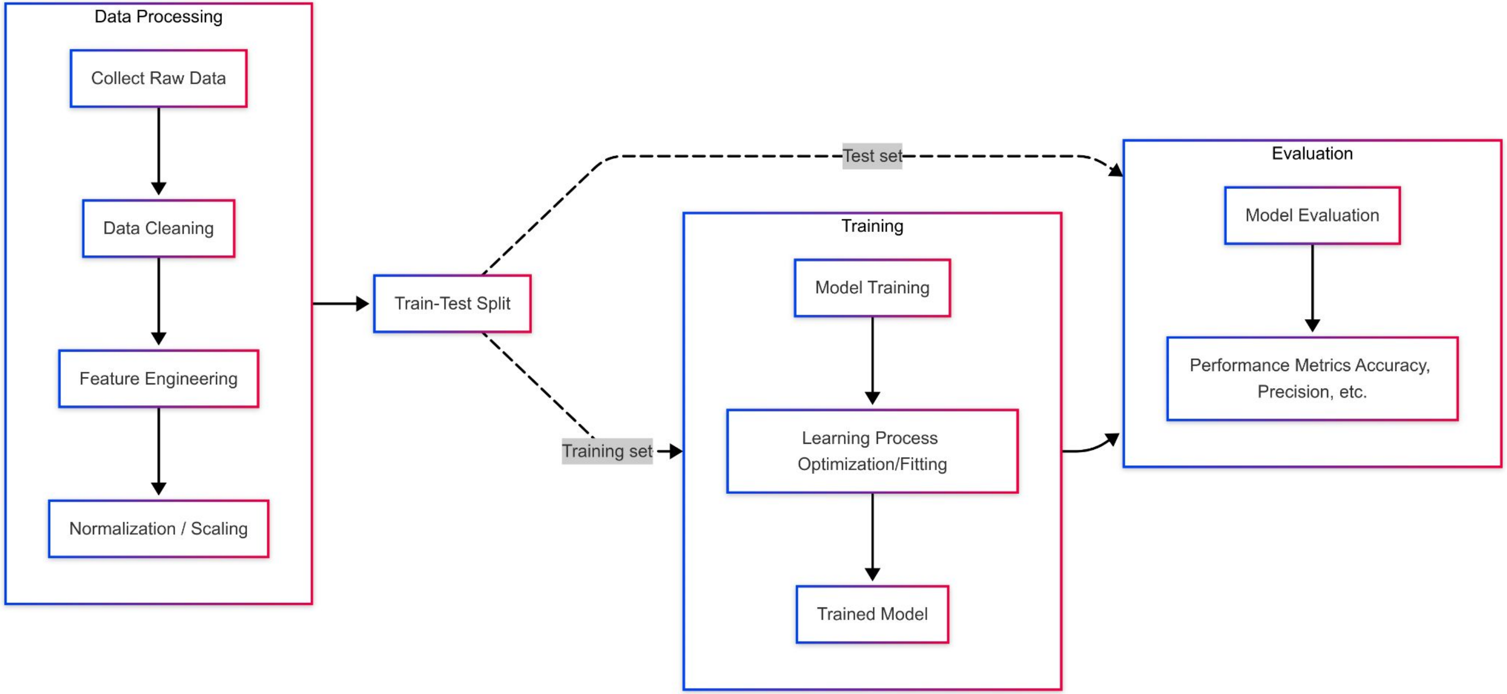
By Loukmane MAADA

Recap

AI approaches



AI workflow


Data Processing
Collect Raw Data
Data Cleaning
Feature Engineering
Normalization / Scaling
Train-Test Split
Test set
Training set
Training
Model Training
Learning Process Optimization/Fitting
Trained Model
Evaluation
Model Evaluation
Performance Metrics Accuracy, Precision, etc.

Introduction

Introduction

A subset of AI that uses neural networks with multiple layers (hence "deep") to automatically learn representations from data. Deep learning (DL) eliminates much of the manual feature engineering by learning layered feature hierarchies directly from raw data.

Why DL ?

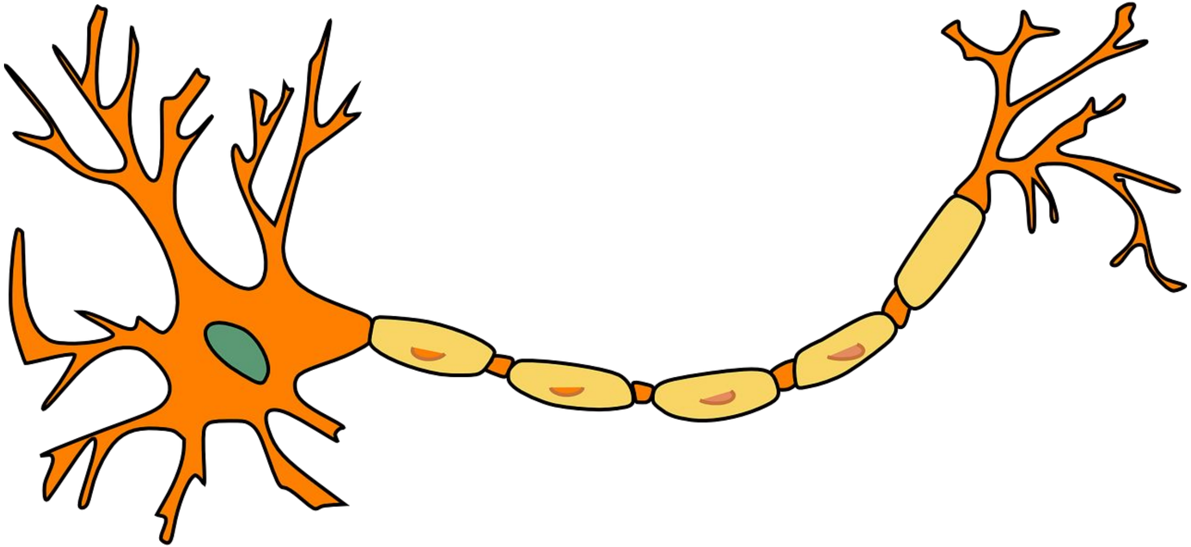
Deep learning approaches have achieved state-of-the-art results in many domains. They can automatically extract intricate patterns and relationships in data through deep neural network architectures.

Examples of DL in real life

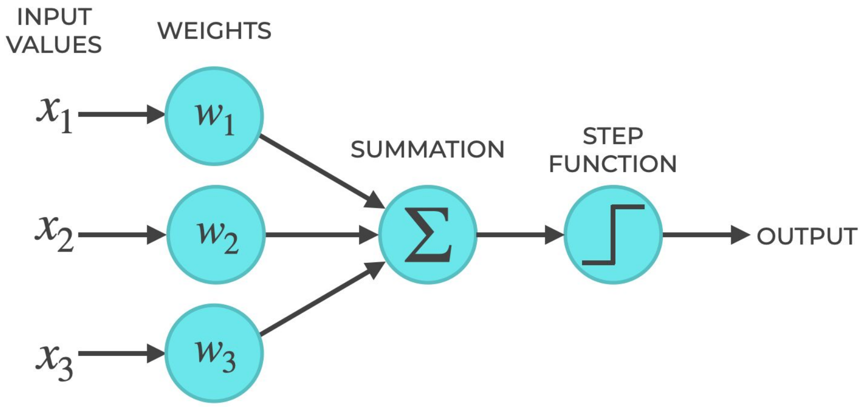
- Computer Vision: Image classification AlexNet (e.g. identifying objects in photos) and facial recognition.
- Natural Language Processing: Machine translation (e.g. Google Translate), sentiment analysis of text, chatbots and virtual assistants.
- Robotics and Games: Autonomous driving systems, drones, and AI agents mastering complex games (e.g. AlphaGo, OpenAI 5).

The perceptron

Neuron



Perceptron



Activation function

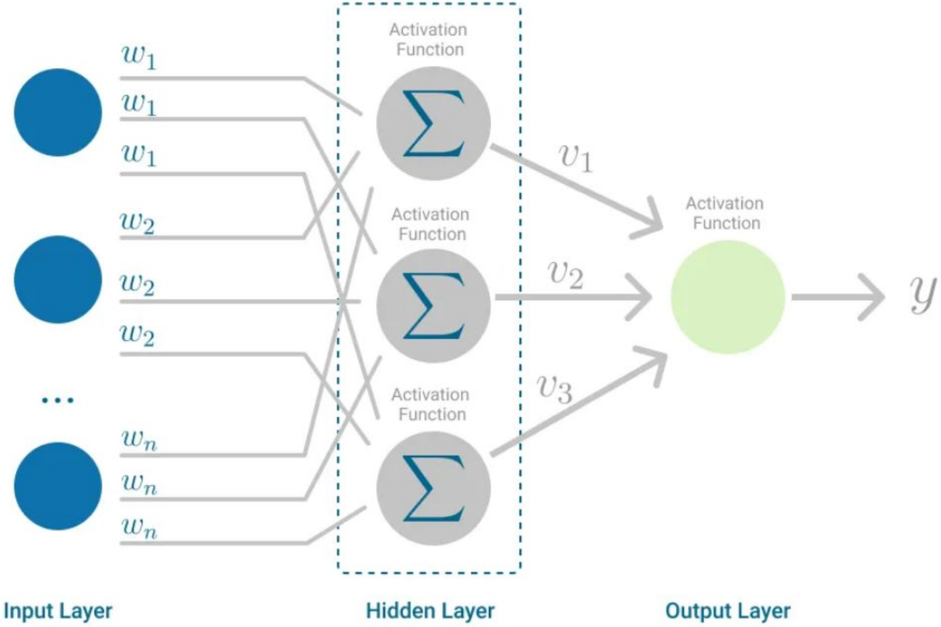
$x = {}^{\mathbf{T}}[x_1, \ldots, x_n]$

$$f(\mathbf{x}) = h(\mathbf{w} \cdot \mathbf{x})$$

$w = [w_1, \ldots . w_n]$

Output

Multilayer Perceptron



Training

Loss function

A loss (or cost) function measures the error between the network's predictions and the true targets. It's a single number that the training process tries to minimize.

Mean Squared Error (MSE)

$$\mathrm{MSE} = \overset{\text{Mean}}{\boxed{\frac{1}{n}\sum_{i=1}^{n}}}\Big(\overset{\text{Error}}{\boxed{Y_i - \hat{Y}_i}}\Big)^{\overset{\text{Squared}}{\boxed{2}}}$$

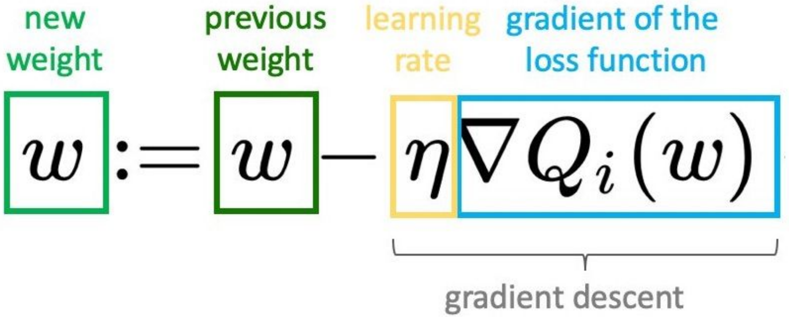
Cross-entropy loss

True probability distribution
(one-shot)

$$H(p, q) = -\sum_{x \epsilon classes} p(x) log\ q(x)$$

Your model's predicted
probability distribution

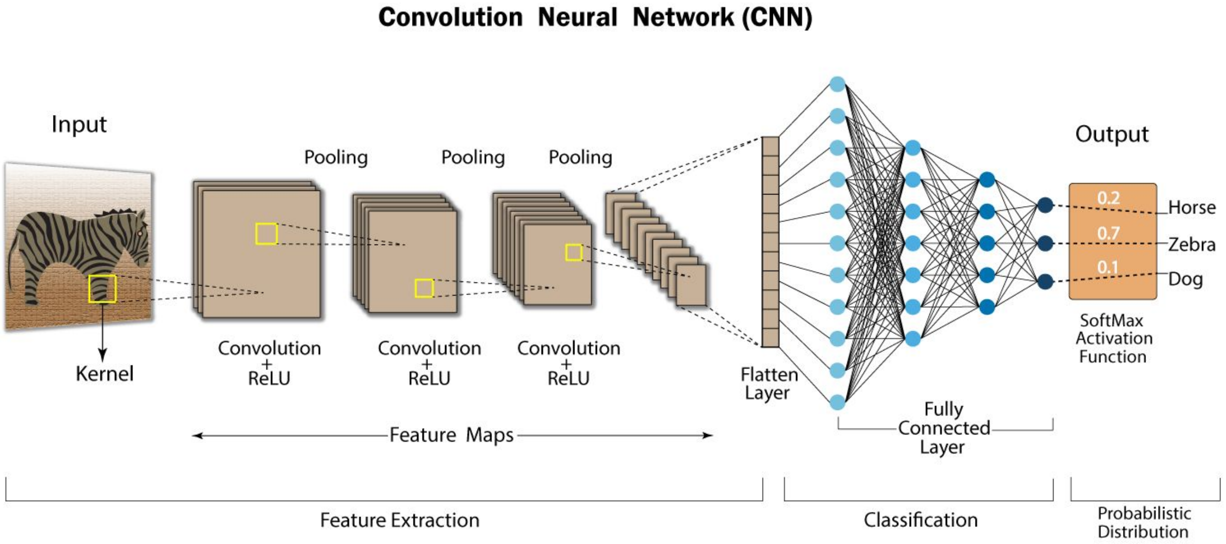
Gradient descent (backpropagation)



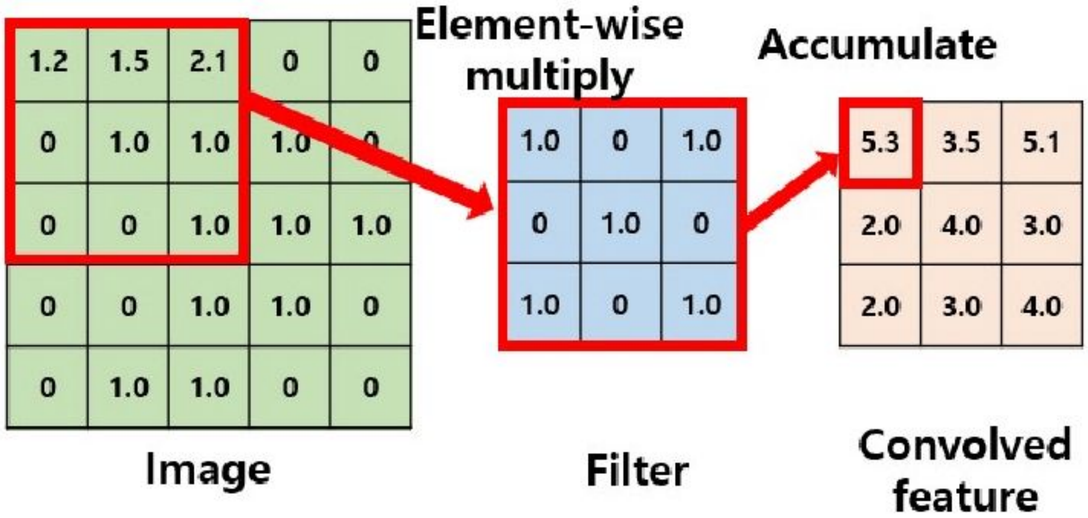
$$\overset{\text{new weight}}{\boxed{w}} := \overset{\text{previous weight}}{\boxed{w}} - \underbrace{\overset{\text{learning rate}}{\boxed{\eta}}\overset{\text{gradient of the loss function}}{\boxed{\nabla Q_i(w)}}}_{\text{gradient descent}}$$

Advanced Models

CNN


Convolution Neural Network (CNN)
Input
Kernel
Pooling
Pooling
Pooling
Convolution + ReLU
Convolution + ReLU
Convolution + ReLU
Feature Maps
Flatten Layer
Fully Connected Layer
Output
0.2
0.7
0.1
Horse
Zebra
Dog
SoftMax Activation Function
Feature Extraction
Classification
Probabilistic Distribution

Convolution



Element-wise multiply

Accumulate

Image

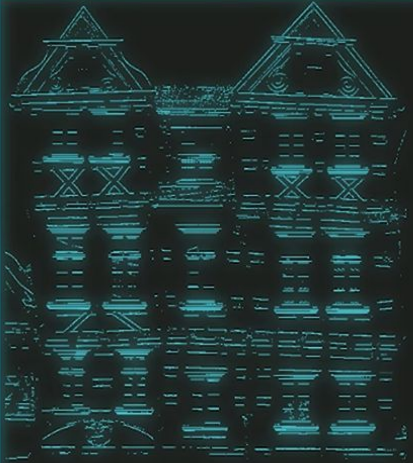
1.2	1.5	2.1	0	0
0	1.0	1.0	1.0	0
0	0	1.0	1.0	1.0
0	0	1.0	1.0	0
0	1.0	1.0	0	0

Filter

1.0	0	1.0
0	1.0	0
1.0	0	1.0

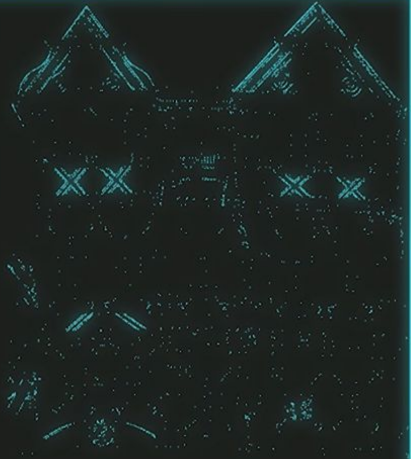
Convolved feature

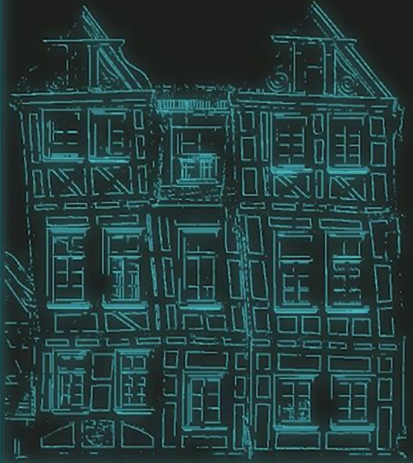
5.3	3.5	5.1
2.0	4.0	3.0
2.0	3.0	4.0



2	2	4	2	2
1	1	2	1	1
0	0	0	0	0
−1	−1	−2	−1	−1
−2	−2	−4	−2	−2

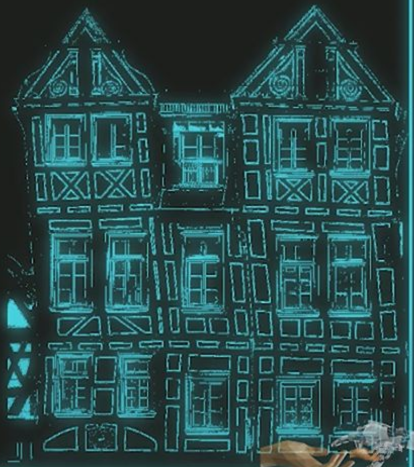
1	1	0	−1	−1
1	1	0	−1	−1
0	0	0	0	0
−1	−1	0	1	1
−1	−1	0	1	1



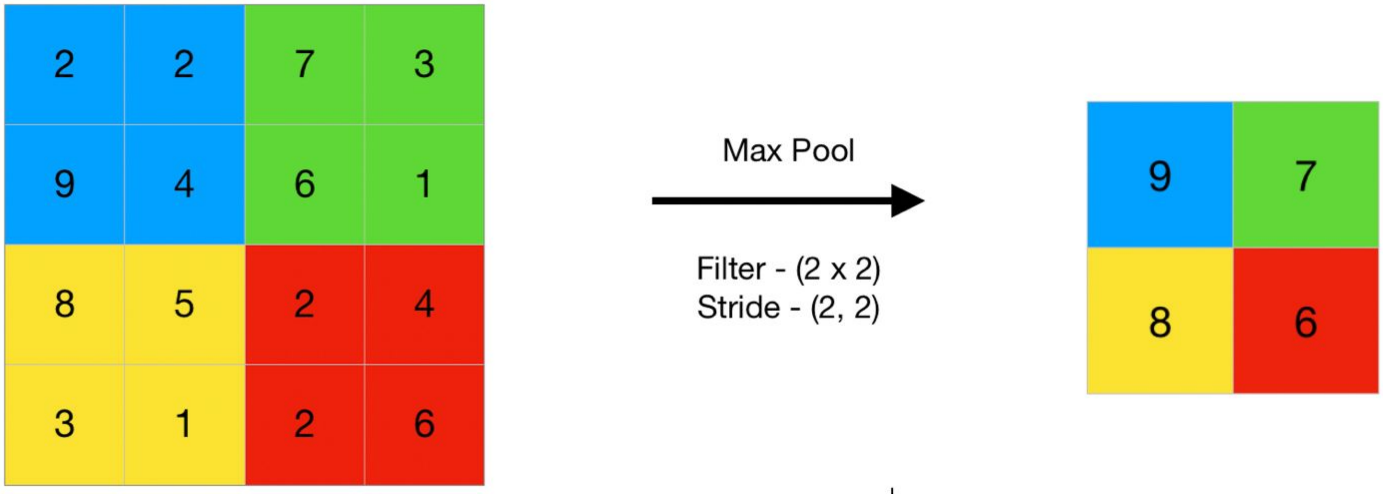


0	1	2	3	4
−1	0	1	2	3
−2	−1	0	1	2
−3	−2	−1	0	1
−4	−3	−2	−1	0

0	0	−1	0	0
0	0	−1	0	0
−1	−1	5	−1	−1
0	0	−1	0	0
0	0	−1	0	0



pooling



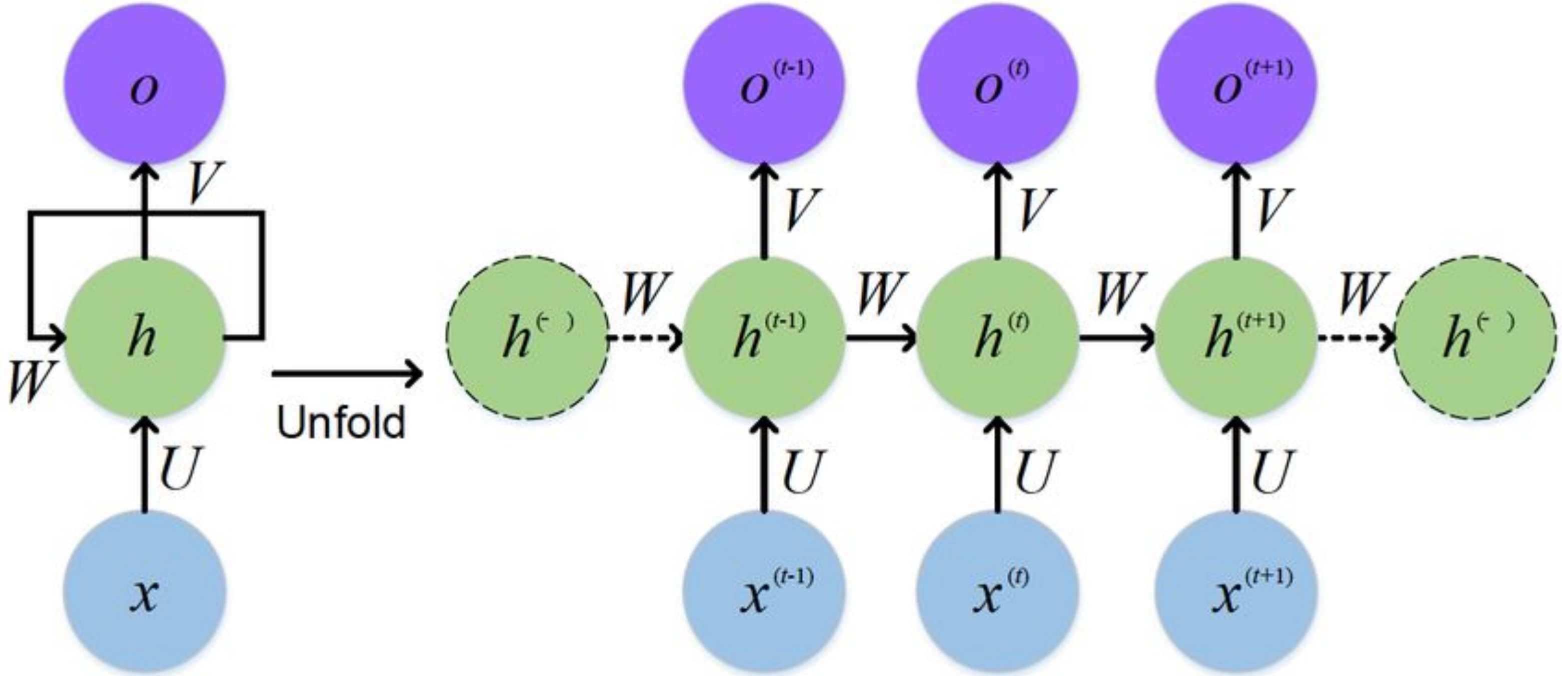
2	2	7	3
9	4	6	1
8	5	2	4
3	1	2	6

Max Pool

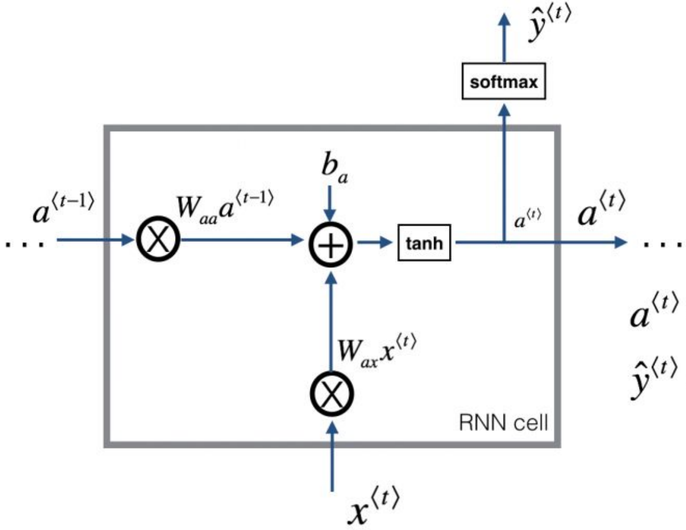
Filter - (2 x 2)
Stride - (2, 2)

9	7
8	6

RNN


o
V
W
h
W
U
x
Unfold
$h^{(\cdot)}$
W
$o^{(t-1)}$
V
$h^{(t-1)}$
U
$x^{(t-1)}$
W
$o^{(t)}$
V
$h^{(t)}$
U
$x^{(t)}$
W
$o^{(t+1)}$
V
$h^{(t+1)}$
U
$x^{(t+1)}$
W
$h^{(\cdot)}$

RNN heuristic

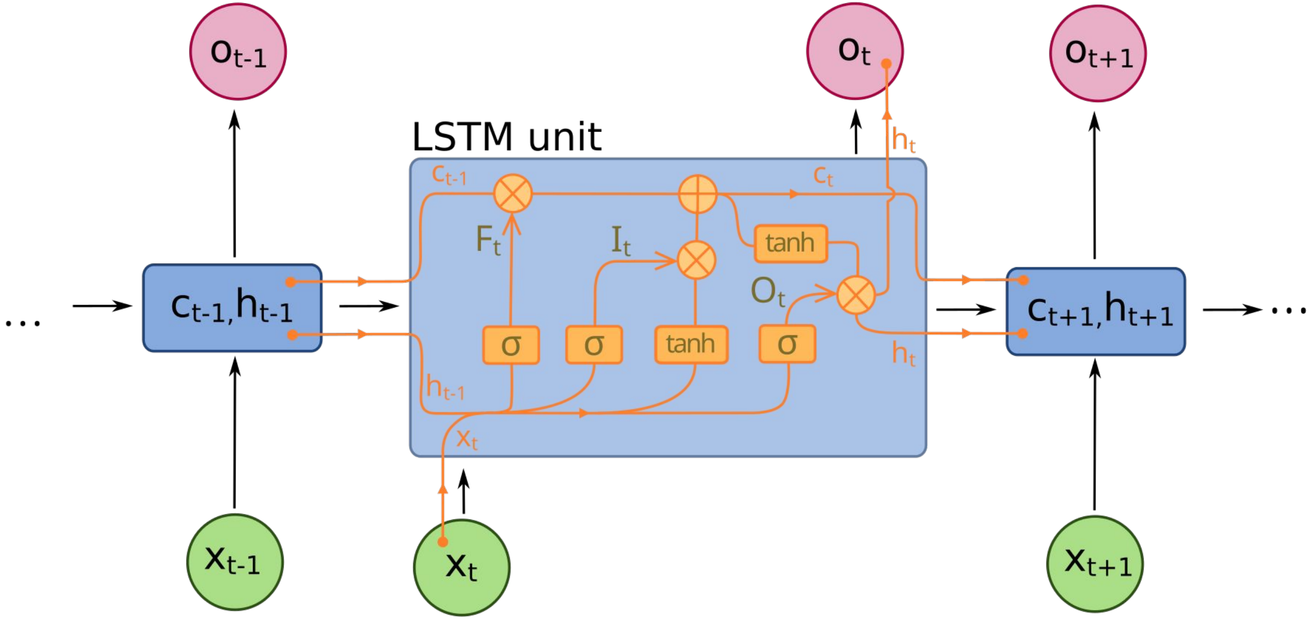


$\hat{y}^{\langle t \rangle}$
softmax
b_a
$a^{\langle t-1 \rangle}$
$W_{aa}a^{\langle t-1 \rangle}$
tanh
$a^{\langle t \rangle}$
$a^{\langle t \rangle}$
$W_{ax}x^{\langle t \rangle}$
RNN cell
$x^{\langle t \rangle}$

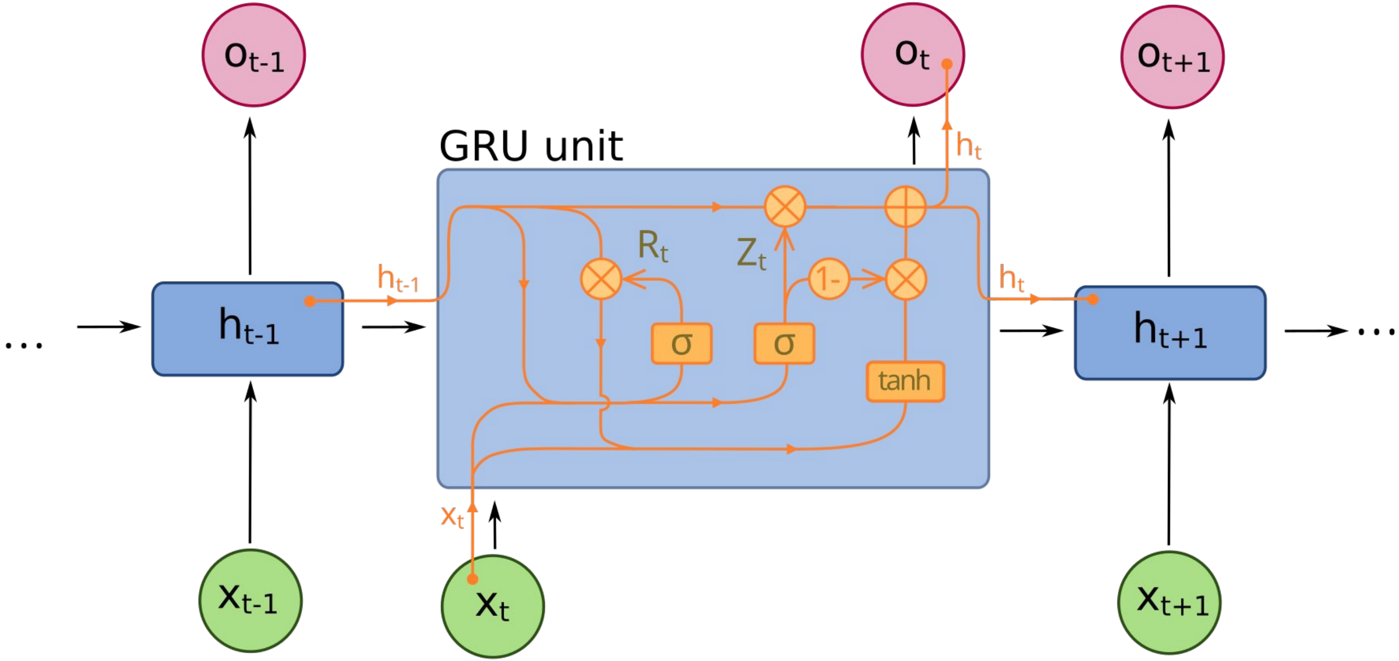
$$a^{\langle t \rangle} = \tanh(W_{ax}x^{\langle t \rangle} + W_{aa}a^{\langle t-1 \rangle} + b_a)$$

$$\hat{y}^{\langle t \rangle} = soft\max(W_{ya}a^{\langle t \rangle} + b_y)$$

LSTM Architecture



GRU Architecture



Questions?

See you

next time

つづく