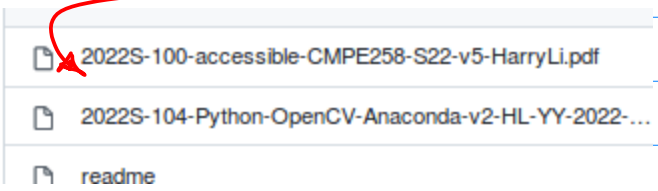
CMPE258 Spring2022

1/

Feb1st. Organizational meeting.

1. Today's Topics "Greensheet"



Naming Convention Yr+Semester+ID +Name+Date

Content Information: E-mail: hua.li@sjsu.edu

Text message to (650)400-1116.

Office Hours: M.W. 4:30-5:30pm.

Zoom (link to be shared in the email)
Join from PC, Mac, Linux, iOS or Android: https://sjsu.zoom.us/j/85616325978?pwd=MzlRbDJXVHBDQ2g1U0RPM2tYc045Zz09

Password: 451032

On-Line materials on github
https://github.com/hualili/opencv/tree/master/deep-learning-2020S
https://github.com/hualili/opencv/tree/master/deep-learning-2022s

Also, CANVAS → mostly for Assignments and projects.

All Assignment/Projects are posted on Both github & SJSU CANVAS.

Lecture Material consists of PPT. posted on github, and Lecture Notes (White-Board Written Notes)

Core Emphases of the Class: Deep Convolutional Neural Networks, And their Application in Image Analysis, Video Analysis.

1. Text Book:
https://github.com/hualili/opencv/blob/master/IP120-AI-DL/2018F/2018F-6-DeepLearningCh02.pdf

2. Computer Vision Book By Horn as a reference for Convolution & Image Segmentation, Contours Analysis (Binary Image).

3. OpenCV Reference Book (2nd Edition) together with On-Line Document (OpenCV).

Note: OpenGL (GL: Graphics Library) is just for Reference purpose, no need for this class. (But maybe helpful for the future research).

Unity is game Development platform, interactive 3D Graphics Design platform.

Programming Languages:
1' Python. 3.6 or 3.7
2. C/C++

Feb.8th.
Homework (Due A week from today)
No Submission. Submit A screen

Capture that shows OpenCV installed successfully, with Jpg or png file with Naming of the file as follows:

FirstName_LastName_SID_OpenCV.jpg

This Homework will be posted on CANVAS, Submission is on CANVAS

Homework, Installation of Tensor Flow, Due 2weeks Feb 15th.

Submission: Screen Capture that shows the installation is successful. Submission on CANVAS.

Submit jpg, png file with the Naming convention as follows:

FirstName_LastName_SID_TF.jpg

Note: Optional, for Edge AI Computing, Consider using NVDA Jetson NANO (4GB) version.

5% Bonus

Grading:

Homework, Projects : 30%
5% 25%

Project 1. Computer Vision for Preprocessing. plus Deep Convolutional Neural Nets To give Real Time Detection Result of Last 4 Digits of a Student ID.

10%.

Project 2. "Semester Long" Project, with technical requirements (List)

Team project. 4 person Team.

Each person has clear Definition of the tasks (Programming/Coding) And Balanced Contribution.

Final PPT, Demo Presentation

15%.

Midterm Exam: 30%.

Need to use your Laptop Computer, to Run/Execute code, modify the code.

Final 40%

Introduction

Topics { Neural Networks formulation (Basic Building Blocks)
Digital Images/Videos.

Example: A Single Neuron Formulation (Some kind Brain Cell)

Step 1. Summation function.

Fig 1.

"Σ" Summation function.

Note: $\sum_{K=1}^{N} x_K = x_1 + x_2 + \cdots + x_N$

Step 2. Inputs

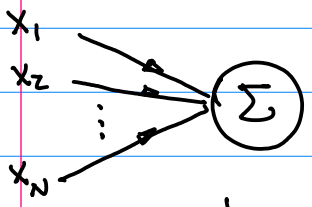
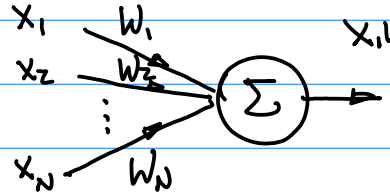


Fig 2.

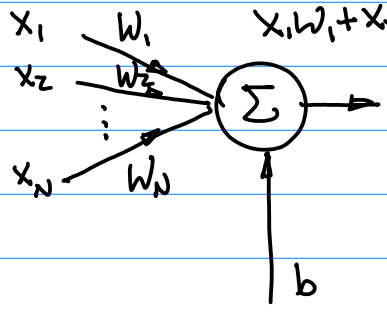
Step 3. Weights (Knowledge)



$x_1 w_1 + x_2 w_2 + \cdots + x_N w_N = \sum_{K=1}^{N} x_K w_N$

Fig 3.

$$\sum_{K=1}^{N} x_K w_N = x_1 w_1 + x_2 w_2 + \cdots + x_N w_N \quad \cdots (1)$$



$x_1 w_1 + x_2 w_2 + \cdots + x_N w_N + b$

Fig 4

$$\sum_{K=1}^{N} x_K w_N = x_1 w_1 + x_2 w_2 + \cdots + x_N w_N + b \quad \cdots (1b)$$

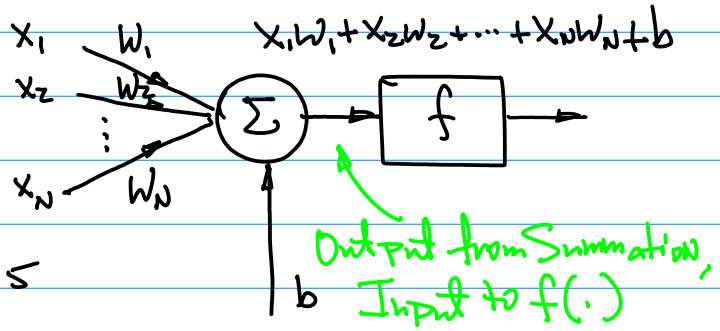


Fig 5

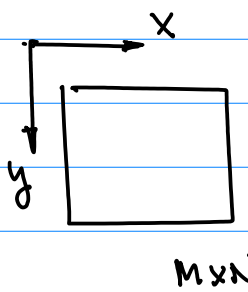
Note: Activation function f, denoted as f(·) (A function of Independent Variable "·", or A function of Input "·")

$$f(\cdot) = f(x_1 w_1 + x_2 w_2 + \cdots + x_N w_N + b)$$

$$= f\left(\sum_{K=1}^{N} w_K x_K + b\right) \quad \cdots (2)$$

Summary: The output of a Single is given by Eqn (2). Where Activation function f(·) can take different forms, it affects the Learning, Learning Speed.

Example: Digital Image, I(x,y)



$I(x,y)$ ← (x,y) Location of A picture element, "pixel"

Intensity, And/OR Color of An Image

In Case of a Single pixel, (x,y) is the Location of this pixel, I is Color/Intensity of the pixel

Note: For An Image $I(x,y)$ its features include { Resolution $M \times N$ (Nb. of pixels per Row, Nb. of Rows per frame); Pixel Depth: bpp (Bit per pixel)

$I(x,y)_{M \times N}$ OR $I(x,y)$

1024 x 768 = pixels/Row × Rows

For A color Image, A pixel depth very often is equal to 24 (bpp)

r, g, b Primitive color of red (r), green (g), blue (b) has 8 bits quantization level, e.g.

r: [0, 255], g: [0, 255], b: [0, 255].

Feb 8th (Tue)

Today's Topics: 1° Introduction, Basic Building Blocks, Math Formulation. 2° Simple Python Code for OpenCV.

Example: Notations & Formulation

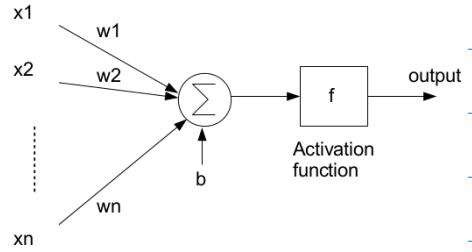


Fig.1

1. Notation for Input $X_i, i=1,2,\cdots,N$

$$\{X_i | i=1,2,\cdots,N\} \quad \cdots (1)$$

Vector form,

$$(X_1, X_2, \cdots, X_N) \quad \cdots (1b)$$

$\rightarrow X$

Introduce Superscript j for Experiment j

Input x_i^j, $i=1,2,\cdots,N$; $j=1,2,\cdots,P$

Hence Eqn (1) Becomes

$$\{x_i^j | i=1,2,\cdots,N; j=1,2,\cdots,P\}$$

$(x_1^j, x_2^j, \cdots, x_N^j)$ for Experiment j

2. Notation for Weight

w_i, for $i=1,2,\cdots,N$

Hence,

$$(w_1, w_2, \cdots, w_N) \quad \cdots (2)$$

$\rightarrow W$

3. Inputs & weights

x_i w_i

$w_i x_i$ for $i=1,2,\cdots,N$

$(x_1, x_2, \cdots, x_N) \cdot (w_1, w_2, \cdots, w_N)$

$= w_1x_1 + w_2x_2 + \cdots + w_Nx_N = \sum_{i=1}^{N} w_i x_i$

$(w_1, w_2, \cdots, w_N) \cdot (x_1, x_2, \cdots, x_N)$

$= w_1x_1 + w_2x_2 + \cdots + w_Nx_N = \sum_{i=1}^{N} w_i x_i$

4. Transfer Function, ~~Denoted as~~ h or $h(\cdot)$

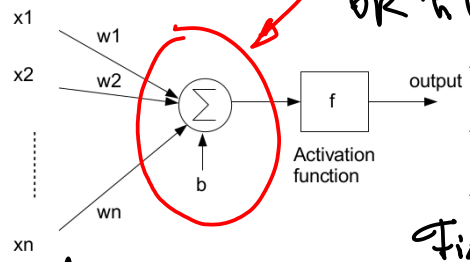


Fig. 2

$h = \sum_{i=1}^{N} w_i x_i + b = W \cdot X + b$ ... (3)

Offset ("Bias")

h, or $h(\cdot)$, or Eqn (3), or $h(w_i; b)$, or $h(w_i)$

5. Activation Function. f

Acts like a switch, ON/OFF & Attenuate the Output

f, $f\left(\sum_{i=1}^{N} w_i x_i + b\right)$, OR $f(h(w_i, b))$, OR $f(h(\cdot))$

The output of A Neuron is denoted as y, and $y = f\left(\sum_{i=1}^{N} w_i x_i + b\right)$

$= f(h(w_i; b))$ ... (4)

5. Outputs for A Neural Network

y_k, for $k=1,2,\cdots,Q$

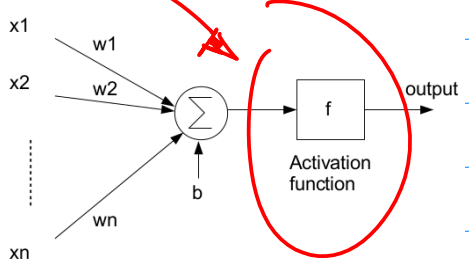
if $Q=1$, y_1, also better to just y_1 for a single Neuron as y

Question: what output? whose output?

Output from the Neuron (or Network) $\tilde{y}$ (Tilde)

Ground Truth is denoted as y

6. Loss function, Objective Function, or difference



"Supervised Learning"

For Experiment j (multiple experiments)

$$\tilde{y}^j - y^j \quad , \quad j=1,2,\cdots,P \quad \cdots (5)$$

$\tilde{y}^j$: Output from the NN (Neuron or Network)

y^j: Ground Truth for the experiment

for $j=1$, $\tilde{y}^1 - y^1$

$j=2$, $\tilde{y}^2 - y^2$

$\vdots$

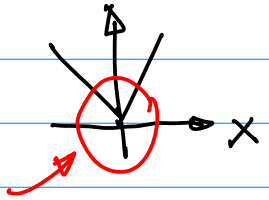
$j=P$, $\tilde{y}^P - y^P$

Therefore, put all these differences (Loss) together

$$(\tilde{y}^1 - y^1) + (\tilde{y}^2 - y^2) + \cdots + (\tilde{y}^P - y^P)$$

$$= \sum_{k=1}^{P} (\tilde{y}^k - y^k) \quad \cdots (6)$$

Note: $y = |x|$



To deal with the issue of Absolute value of a function, Let's square it.

Hence Eqn (6) becomes

$$\sum_{k=1}^{P} (\tilde{y}^k - y^k)^2 \quad \cdots (6b)$$

7. Objective function

$$L \triangleq \sum_{k=1}^{P} (\tilde{y}^k - y^k)^2 \quad \cdots (7)$$

OR,

$$L(w_i) \triangleq \sum_{k=1}^{P} (\tilde{y}^k - y^k) \quad \cdots (7b)$$

$$= \sum_{k=1}^{P} (\tilde{y}^k(f(w_i)) - y^k) \quad \cdots (7c)$$

8. To generalize the result in Eqn (7c) for multiple Output, we have the following formulation.

From Eqn (7)

$$\sum_{k=1}^{P} (\tilde{y}^k_{k_2} - y^k_{k_2})^2$$

k_2: Output k_2 for multiple output y_j, $j=1,2,\cdots,k_2,k_2+1,\cdots,M$

We can count All outputs loss.

$$\sum_{k=1}^{P} (\tilde{y}^k_1 - y^k_1), \quad \sum_{k=1}^{P} (\tilde{y}^k_2 - y^k_2), \quad \cdots$$

$$\cdots \sum_{k=1}^{P} (\tilde{y}^k_M - y^k_M)$$

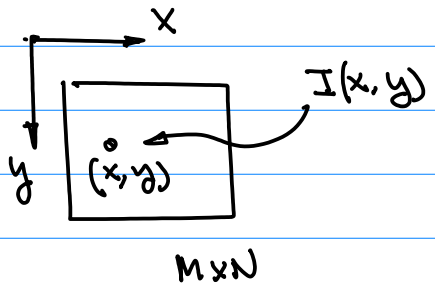
Hence

$$\sum_{k_2=1}^{M} \sum_{k_1=1}^{P} (\tilde{y}^{k_1}_{k_2} - y^{k_1}_{k_2})^2 \quad \cdots (8)$$

CmpE258 Feb8,22

$$L(W_i) \triangleq \frac{1}{2} \sum_{h_2=1}^{m} \sum_{h_1=1}^{p} \left(\tilde{y}^{h_2}_{h_2} - y^{h_2}_{h_2} \right)^2 \quad \cdots (8b)$$

Example: Digital Image I(x,y)

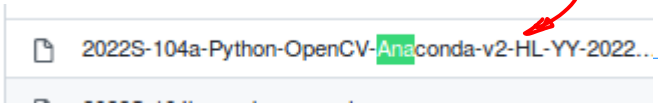


Installation of OpenCV → Python OR (C/C++, But Python is Better)

Different Packages for ML/DL, may Need different Version of Python, And different Packages

Anaconda is a well developed, Adopted tool for Package management.

1. Check github class Reference



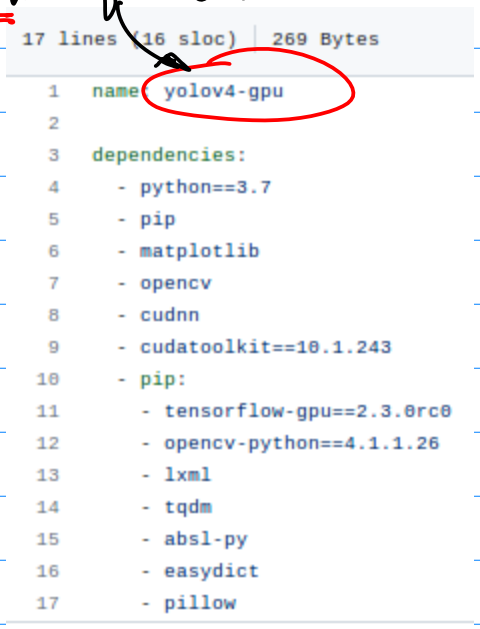
Note: For installation of Anaconda, Check github document, "2022S-104-~"

Once Anaconda is installed, then Let's take a look the configuration for

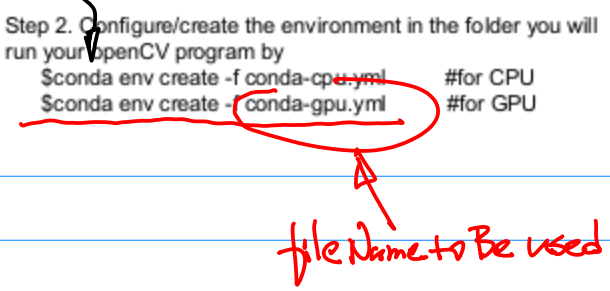
Tensorflow & OpenCV Environment

Step 1.

Name of the environment



Step2. Create the environment



Step3 Activate the enviroment

Step4. Run Your Python Code (OpenCV or T.F.)

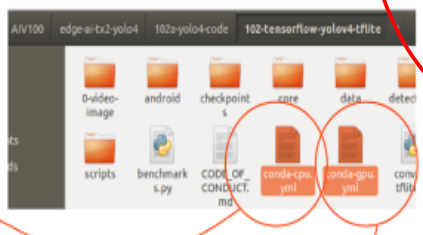
Running OpenCV with Conda Environment

Step 1. Create configuration file .yml for CPU and GPU version

conda-cpu.yml

```
name: yolov4-cpu

dependencies:
  - python==3.7
  - pip
  - matplotlib
  - opencv
  - pip:
    - opencv-python==4.1.1.26
    - lxml
    - tqdm
    - tensorflow==2.3.0rc0
    - absl-py
    - easydict
    - pillow
```



Step 2. Configure/create the environment in the folder you will run your openCV program by

```
$conda env create -f conda-cpu.yml      #for CPU
$conda env create -f conda-gpu.yml      #for GPU
```

Step 3. Activate the environment with the name that you have created by

```
$ conda activate yolov4-cpu  #for CPU
$ conda activate yolov4-gpu  #for GPU
```

Name of the environment in the .yml file

conda-gpu.yml

```
name: yolov4-gpu

dependencies:
  - python==3.7
  - pip
  - matplotlib
  - opencv
  - cudnn
  - cudatoolkit==10.1.243
  - pip:
    - tensorflow-gpu==2.3.0rc0
    - opencv-python==4.1.1.26
    - lxml
    - tqdm
    - absl-py
    - easydict
    - pillow
```

```
V100/edge-ai-tx2-yolo4/102a-yolo4-code/102-tensorflow-yolov4-tflite$ conda activate yolov4-gpu
(yolov4-gpu) harry@workstation:/media/harry/easystore/backup-2020-2-15/CT1/3proejcts/3-8-smart-
V100/edge-ai-tx2-yolo4/102a-yolo4-code/102-tensorflow-yolov4-tflite$
```

Step 4. Now run your Python OpenCV program

```
$python myOpenCV.py image.png
```

conda-cpu.yml

```
name: yolov4-cpu

dependencies:
  - python==3.7
  - pip
  - matplotlib
  - opencv
  - pip:
    - opencv-python==4.1.1.26
    - lxml
    - tqdm
    - tensorflow==2.3.0rc0
    - absl-py
    - easydict
    - pillow
```

Name of the environment to be created

Note: Install Anaconda → Create the environment for T.F. & OpenCV

Activate the environment → Execute OpenCV Sample Code (From the class github)

2022S-104d-#2-pdisplay-2019-1-30.py

a. Program Header — Name, Program Name, Version, Status (Debug, Release), Date, Note

b. cv2 OpenCV

Caption of the window

```
"""
pdisplay.py
Demo read and display image
"""
import sys
import cv2
import numpy as np

#main(sys.argv[1:])
window_name = 'Display Image'

imageName = sys.argv[1] #get file name from command line

src = cv2.imread(imageName, cv2.IMREAD_COLOR)

if src is None:
   print ('Error opening image!')
   print ('Usage: pdisplay.py image_name\n')
```

```
22 while True:
23     cv2.imshow(window_name, src)
24
25     c = cv2.waitKey(500)
26     if c == 27:  #ESC
27         break
28
29     ind += 1
```

③

Keyboard input to exit.

Note: These Python functions are required. (memorize them!)

① import cv2 ② cv2.imread( )

③ cv2.imshow( ).

Homework (Due A week from today) Feb 15

1° Installation of OpenCV.

2° Installation of Anaconda.

3° Use your Smartphone to take photo, and Save it for OpenCV Program

4° Write A Python Program (Ref. Code from the class github is ok) to Display:

a. Your Name + SID (4 Digits)

b. Your Smartphone picture.

5° Submission: One pdf file And Zipped. On CANVAS.