

August 23 (Tue)

First Day of the Class

1. Organizational meeting

"Green Sheet"

Repo: [github/Pradil10/OpenCV-deep-learning-2022](https://github.com/Pradil10/OpenCV-deep-learning-2022) | the github.

Email: hua.li@sjtu.edu

(650) 400-1116 Cellphone for
Text message Only.Office Hours: M.W. On Zoom.
(see syllabus for the
Zoom link).

2. Software Tools:

Anaconda — Install it by the end
of this week;

TensorFlow, TF 2.0

OpenCV.

3. Prerequisites: CMP255 & CMP257

Homework: To upload a copy of
your un-official transcript to
show the required courses satisfied.

On CANVAS.

4. Textbook: Deep Learning with Python.

Keras (API) for TF.

Robot Vision Book By Horn (theory theoretical
book, good reference for OpenCV Algorithms.

Good Theoretical Foundations)

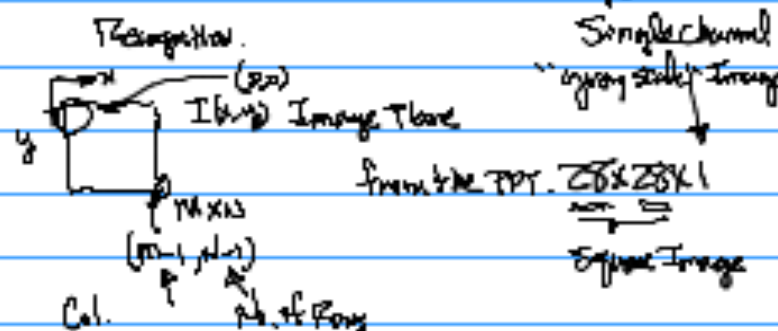
5. Projects { Mandatory Assigned Project
Team Project
(Mandatory)4-person Team. Presentation by the
End of the Semester.

August 23 (Wed)

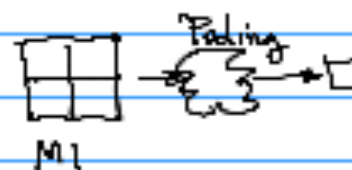
Note: 1. The lecture Note will be posted on
to the github.2. Zoom Recording will be posted on the
the github.Homework: By A week from today. 1. Anaconda
Installation; 2. OpenCV Installation. Submission
On CANVAS. Jpg/Png Image \rightarrow pdf \rightarrow zip
2pts.

Example: (github: 2022-fur-113)

MNIST Architecture for Handwritten Digits

(gray scale image \rightarrow 1 channel \rightarrow 8 bit \rightarrow [0, 255])

First Layer of the MNIST Architecture

n1 channel/plane of the 1st convolutional layer
C1Next, pooling \rightarrow Reduction of Resolution \rightarrow 28x28 becomes

From the Architecture diagram:

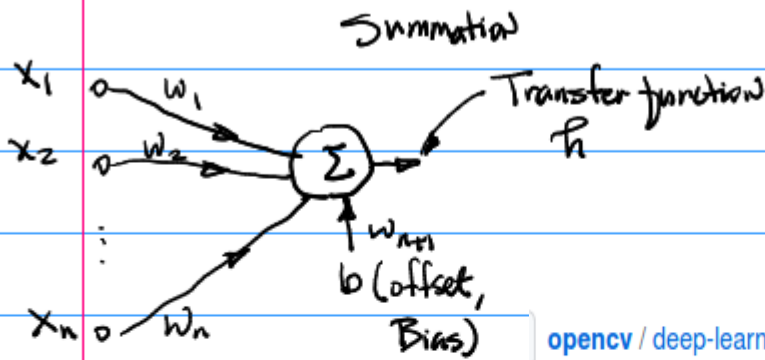
C1 M1 \rightarrow C2 M2 \rightarrow Flatten \rightarrow FFNN (FF)To generalize the quick inspection of the
the CNs, we have to investigate the behavior
of Each Single Neuron as the Basic Building
Block.

CMPE258

Aug. 25, 22

August 30, Tue

2/



Ref. 1

<https://github.com/hualili/opencv/blob/master/deep-learning-2022s/2022F-103b-NN-Intro-Python-v5-2022-8-25.pdf>

[opencv / deep-learning-2022s / 2022S-103a-notation-neuro-loss-function-2022-2-8.pdf](https://github.com/hualili/opencv/blob/master/deep-learning-2022s/2022S-103a-notation-neuro-loss-function-2022-2-8.pdf)

Input/Excitation in Vector Form: $\mathbf{x} = (x_1, x_2, \dots, x_n) \dots (1)$

Weights, links each excitation to the Neuron

Ref. 2. Code

$$\mathbf{W} = (w_1, w_2, \dots, w_n) \dots (2)$$

<https://github.com/hualili/opencv/blob/master/deep-learning-2022s/2022S-110b-%232019S-31-6mnist-numerals-ch02.py>

$$x_1 w_1 + x_2 w_2 + \dots + x_i w_i + \dots + x_n w_n + b w_{n+1} = h \quad \text{Example:}$$

$$\sum_{i=1}^n x_i w_i + b w_{n+1} = h(x_i w_i) \text{ or simply } h(\mathbf{x} \mathbf{W}) \dots (3)$$

$$h(\mathbf{x} \mathbf{W}; b), h(\mathbf{x} \mathbf{W})$$

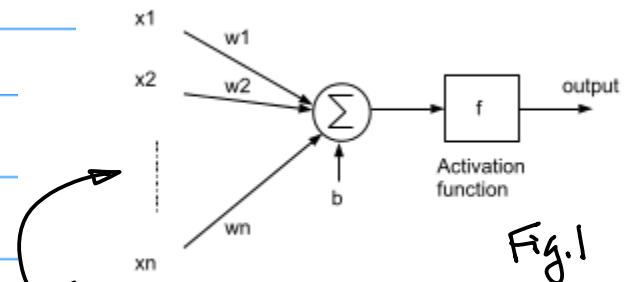
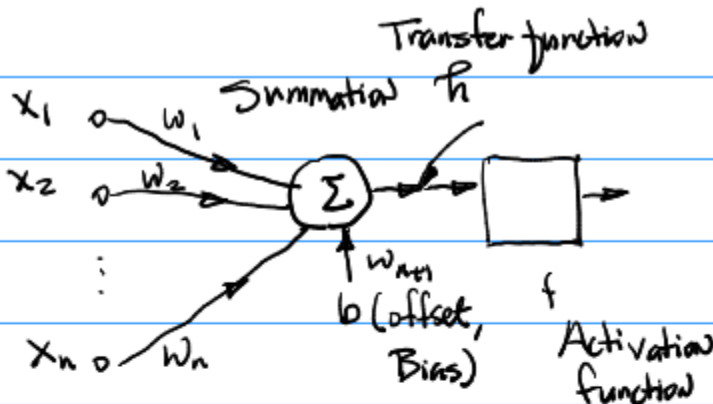


Fig. 1

(x_1, x_2, \dots, x_n) Feature Vector with Dimension N .

$$f(h(\mathbf{x} \mathbf{W})) = f\left(\sum_{i=1}^n x_i w_i + b w_{n+1}\right) \dots (4)$$

$$h(\mathbf{x} \mathbf{W}) = \sum_{i=1}^n x_i w_i + b w_{n+1}$$

$$h = \sum_{i=1}^N w_i x_i = \mathbf{W} \cdot \mathbf{X} + b \quad (11)$$

Transfer function $h(\cdot)$.

$$w_{n+1} b = b'$$

Examples of Different Activation functions include RELU. A piecewise Linear.

Note: Be Able to Build A Single Neuron per a technical Specification, Such as uo11, Activation $f(\cdot)$, Draw a Block

$$y = f\left(\sum_{i=1}^N w_i x_i = \mathbf{W} \cdot \mathbf{X} + b\right). \quad (17)$$

Activation function. Its output is the Response of the Neuron.

CMP258

Aug. 30.

Consider the output of the Neuron y from Eqn (17).

Output of A Single Neuron.

For Multiple Neuron Output, see Fig. 2

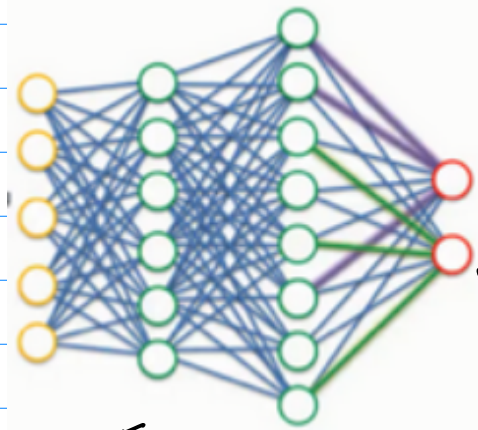


Fig. 2

$y_{di} \dots (1)$

SubScript: $i=1,2$
No. of Output at the Output Layer.

$y_i, i=1,2,\dots,M.$

In practical Application,

$y_{di}^j \dots (2)$

SuperScript
 $j=1,2,\dots,P$ No. of Experiments
Performed, Training Performed.

Look at the Concept & Definition of Loss function.

Mathematically To Compare a Neural Network Output (Single Neuron Output)

function f . function g
Comparison of the Similarity or
difference between f and g .

$$\left\{ \begin{array}{l} f - g \\ f/g \end{array} \right.$$

Difference Between Two Functions.

Take this Approach to define
Loss function,

$$y - \hat{y} \dots (3)$$

Ground Truth.

Output (prediction) from
the Neuron

Outputs

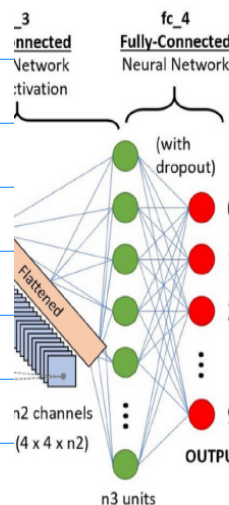


Fig. 3.

\hat{y}_{di}
 $i=0,1,2,\dots,9$

August 30.

$$y_i - \hat{y}_i \quad \dots (4-a)$$

To measure All the output for Each

Training/Experiment

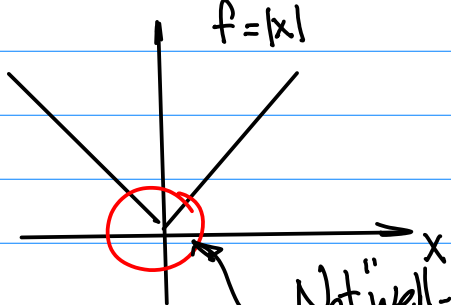
$$\sum_{i=0}^q (y_i - \hat{y}_i) \quad \dots (4-b)$$

Expand this to Experiment/Training up to "P" Times

$$\sum_{j=1}^P \left[\sum_{i=0}^q (y_i^j - \hat{y}_i^j) \right] \quad \dots (4-c)$$

Note: Eqn (4-c) may lead to positive & Negative Terms Cancellation.

Fix: Absolute Value? \rightarrow Squared Instead,



"Well Behaved" System (Function) \rightarrow derivative/partial Derivative up to order "K".

$$L = \sum_{j=1}^P \left[\sum_{i=0}^q (y_i^j - \hat{y}_i^j)^2 \right] \quad \dots (4-d)$$

$\rightarrow J, \text{ or } \Phi$

$$L_{total} = \frac{1}{2} \sum_{j=1}^P (\tilde{y}^j - y^j)^2 \quad (23)$$

Ground Truth

For a Single Neuron @ the Output Layer

Training Based Steepest Gradient Descent SGD.

Example: Given A

function $f(x) = x^2$, Find its Derivative

$$\frac{df}{dx} = 2 \cdot x$$

To get rid of Coefficient from the derivative,

Let's define $f(x) \triangleq \frac{1}{2} x^2$

$$\frac{d}{dx} f(x) = \frac{1}{2} \cdot 2 \cdot x = x$$

$$\frac{\partial L}{\partial w_{i,k}} = \frac{\partial}{\partial w_{i,k}} \frac{1}{2} \sum_{j=1}^P \sum_{i=1}^M (\tilde{y}_i^j - y_i^j)^2 \quad (24)$$

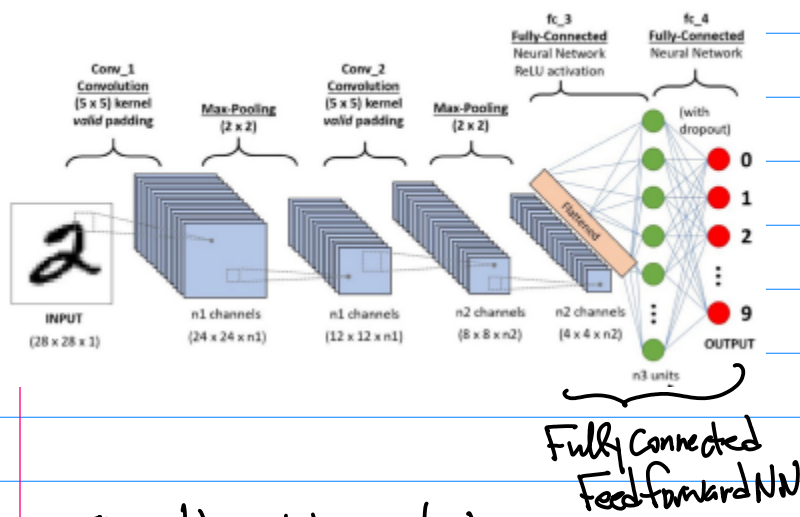
Sept 1st (Th).

Ref:

2022S-103a-notation-neuro-loss-function-2022-2-8.pdf

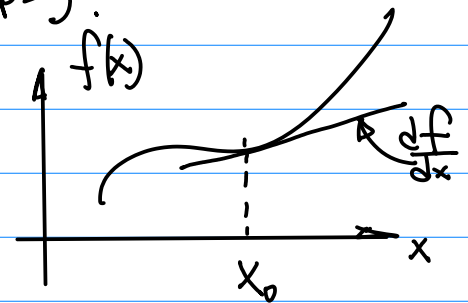
Example: Background on "Learning" of ANN.

2022F-103b-NN-Intro-Python-v5-2022-8-25.pdf



Or Decrease?

Derivative of A given function is a good indicator to give us the description of the function behavior Next Step Ahead (Very small tiny Step Δ).



To Train ANN, we take eqn (23), e.g. Loss function (error function),

$$L(\cdot) \rightarrow L(W_i)$$

Independent Variables

Minimize the Loss $L(\cdot)$, which is the process of Training, which leads to Learning for the NN.

Since the Loss function in Eqn (23) is formulated with a ground Truth y , this defines a Supervised Learning.

Math. Formula: Prediction of Function's Behavior, we like to know given the current function value (Loss function) how this function is going to change at next moment, increase? stay the same?

$$\frac{df}{dx} = \lim_{\Delta x \rightarrow 0} \frac{f(x+\Delta x) - f(x)}{\Delta x} \quad \dots (1)$$

Intuition

$f(x+\Delta x) - f(x) \rightarrow > 0$ (derivative)
 $\Delta x \approx 1 \text{ unit}$
if the derivative is greater than 0

then $f(x+\Delta x) > f(x)$, the next Step function $f(x)$ is increased;

if the derivative $\frac{df}{dx} < 0$, then

$$f(x+\Delta x) < f(x)$$

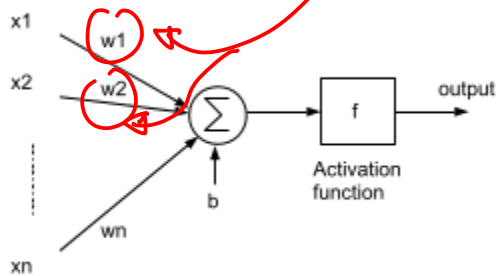
if the derivative $\frac{df}{dx} = 0$, then

$$f(x+\Delta x) = f(x)$$

Consider Two Dimensional Case as an Example for n -dimensional Case.

CMP258
Sept. 1st.

$f(x_1, x_2)$ { $\frac{\partial f(x_1, x_2)}{\partial x_1}$ $\frac{\partial f(x_1, x_2)}{\partial x_2}$ } → Single Neuron
In Case of Training They are "Weights"
 w_1, w_2 .



$\frac{\partial f}{\partial x_1} \rightarrow \frac{\partial f}{\partial w_1}$ In the Context of the Training.

Conclusion: Use Partial Derivate With respect to the weight w_i as an indicator to measure if the Loss function is got reduced or not.

$$\frac{\partial L}{\partial w_{i,k}} = \frac{\partial}{\partial w_{i,k}} \frac{1}{2} \sum_{j=1}^P \sum_{i=1}^M (\tilde{y}_i^j - y_i^j)^2$$

Example: Consider A technique which allows training to be more effect, e.g., ^{the} to minimize training and prediction error (Loss) function

Steps for Development of this technique :

6
Multidimensional Derivative
↓
gradient
↓
the Steepest gradient
↓
the Steepest Descent gradient
↓
The Core technique to train NN. SGD

Ref: from the github

2022S-105c-#20-2021S-4gradient-descent-v2-final-2021-2-8.pdf

Homework (Opt), Due A week from Today.
Sept. 8th, Thursday. ON CANVAS.

1^o Installation of TensorFlow. Version 2.0 or higher.

2^o Screen Capture to show the installation is successful

(24)

Note: All Different Development Tools/Environments including google co-lab, jupyter Notebook etc. Are OK, However for the Deployment purpose, Projects Homework Submission must be in Python Stand Alone form.