HIGH PERFORMANCE RESEARCH COMPUTING

Introduction to Deep Learning

Zhenhua He

High Performance
Research Computing
DIVISION OF RESEARCH

Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL)

AI is intelligence demonstrated by machines, as opposed to natural intelligence displayed by animals including humans.

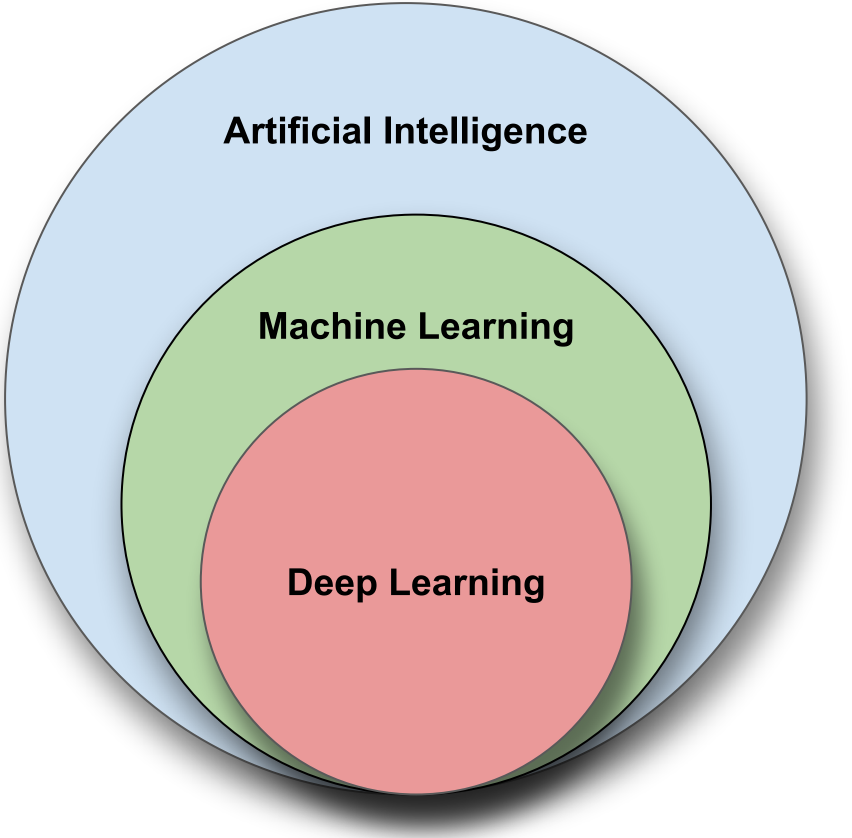
ML is the study of computer algorithms that can improve automatically through experience and using data. It is seen as a part of artificial intelligence.

DL is part of a broader family of machine learning methods based on artificial neural networks.

(Source: Wikipedia)

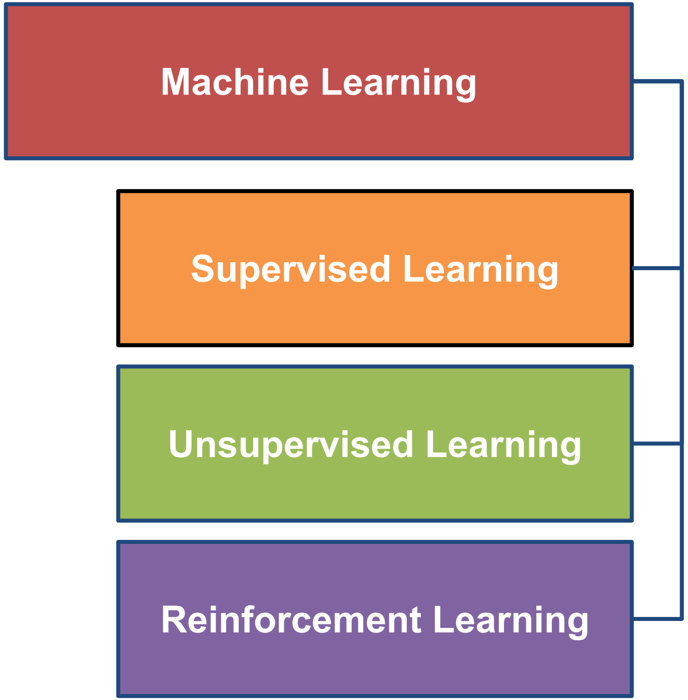
Relationship of AI, ML and DL

- **Artificial Intelligence (AI)** is anything about man-made intelligence exhibited by machines.
- **Machine Learning (ML)** is an approach to achieve **AI**.
- **Deep Learning (DL)** is one technique to implement **ML**.



Types of ML Algorithms

- **Supervised Learning**
  - trained with labeled data; including regression and classification problems
- **Unsupervised Learning**
  - trained with unlabeled data; clustering and association rule learning problems.
- **Reinforcement Learning**
  - no training data; stochastic Markov decision process; robotics and self-driving cars.



What is Deep Learning?

Deep learning is a class of machine learning algorithms that:

- use a cascade of multiple layers of nonlinear processing units for feature extraction and transformation. Each successive layer uses the output from the previous layer as input.
- learn in supervised (e.g., classification) and/or unsupervised (e.g., pattern analysis) manners.
- learn multiple levels of representations that correspond to different levels of abstraction; the levels form a hierarchy of concepts.

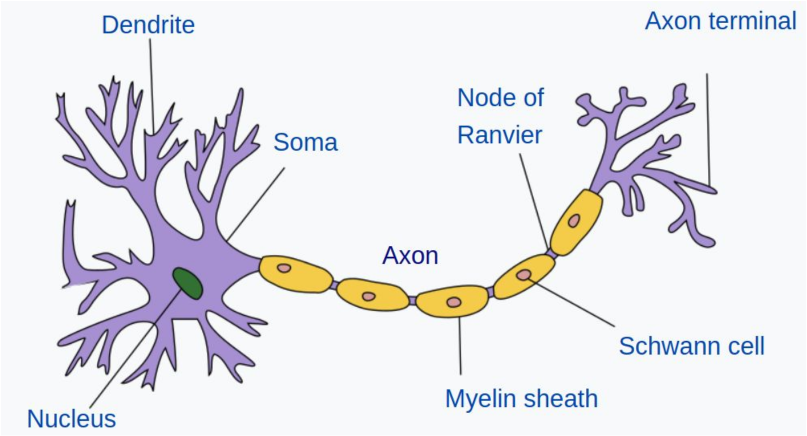
(Source: Wikipedia)

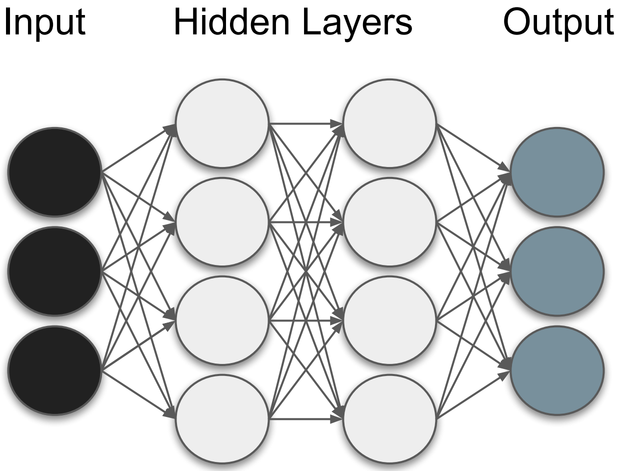
Why Deep Learning?

- Limitations of traditional machine learning algorithms
  - not good at handling high dimensional data.
  - difficult to do feature extraction and object recognition.

- Advantages of deep learning
  - DL is computationally expensive, but it is capable of handling high dimensional data.
  - feature extraction is done automatically.

Artificial Neural Network





(Image Credit: Wikipedia)

Inputs and Outputs

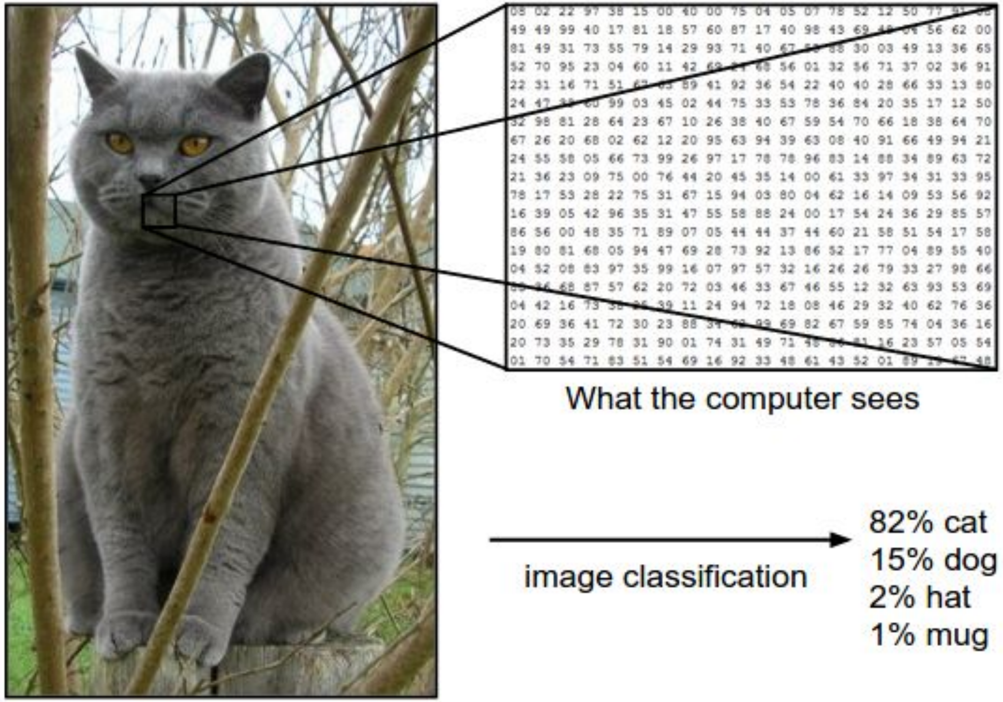
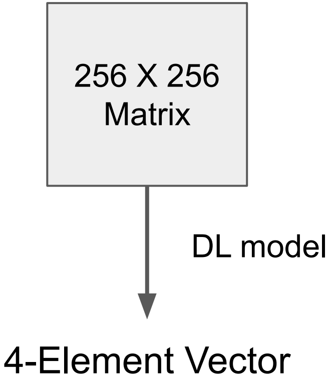


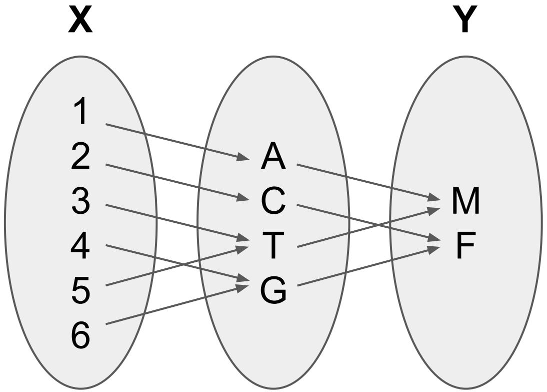
Image from the Stanford CS231 Course



256 X 256 Matrix

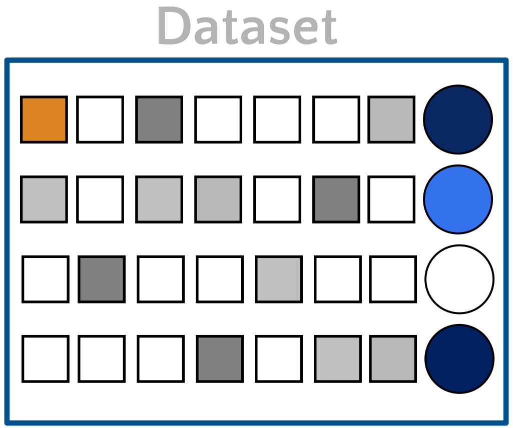
DL model

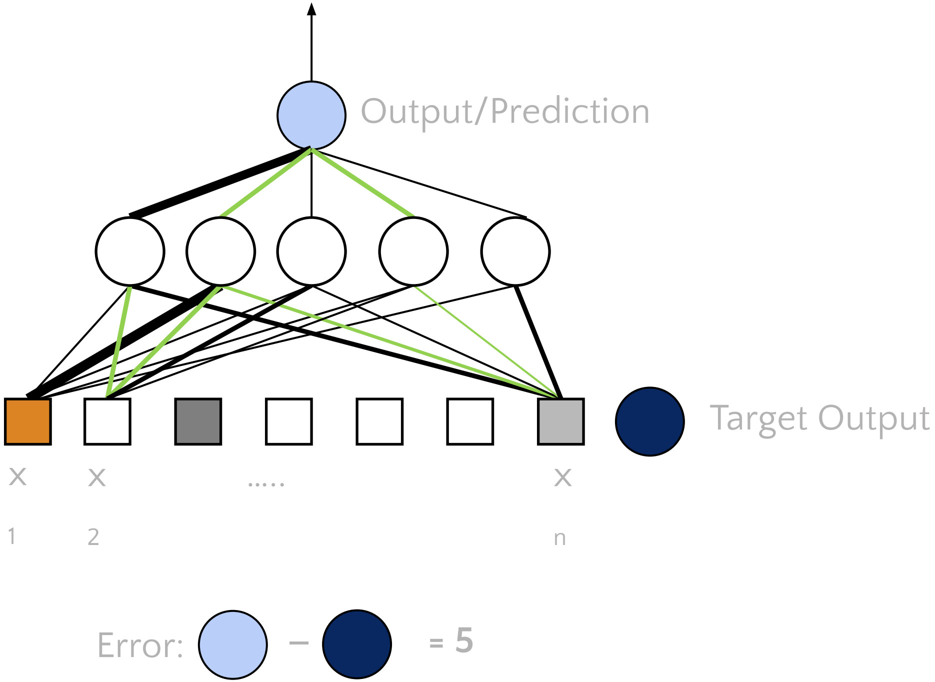
4-Element Vector



With deep learning, we are searching for a **surjective** (or **onto**) function **f** from a set **X** to a set **Y**.

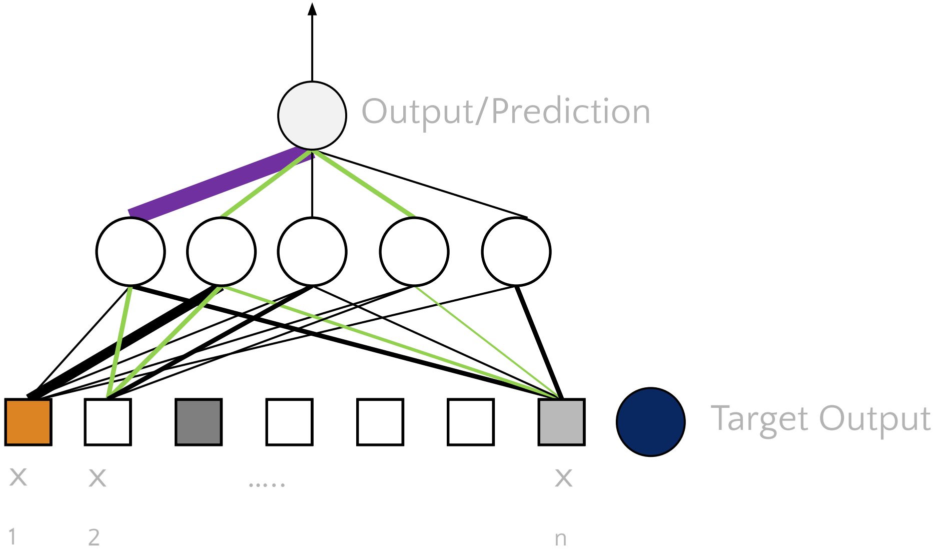
Learning Principle





(Image Credit: NVIDIA Deep Learning Institute)

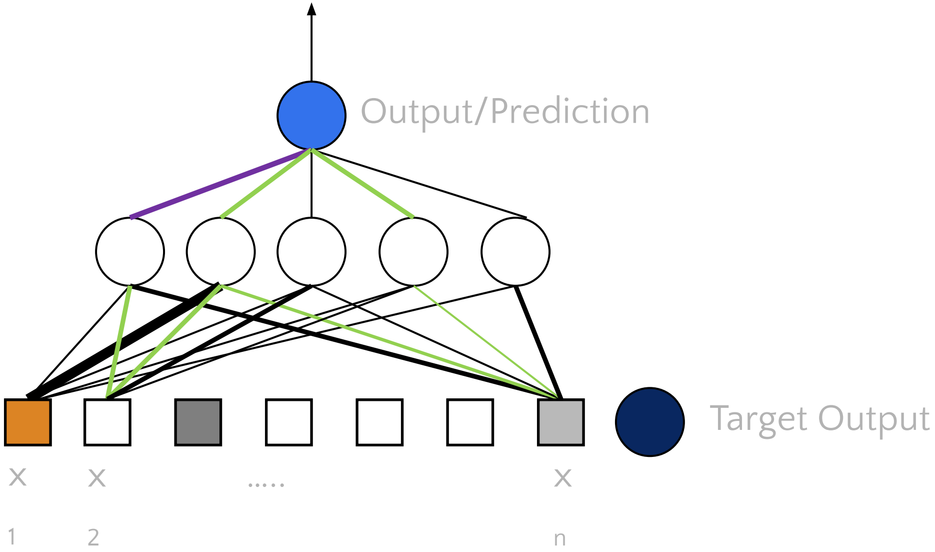
Learning Principle



Error: ○ – ● = 15

(Image Credit: NVIDIA Deep Learning Institute)

Learning Principle



Error: ● – ● = 2.5

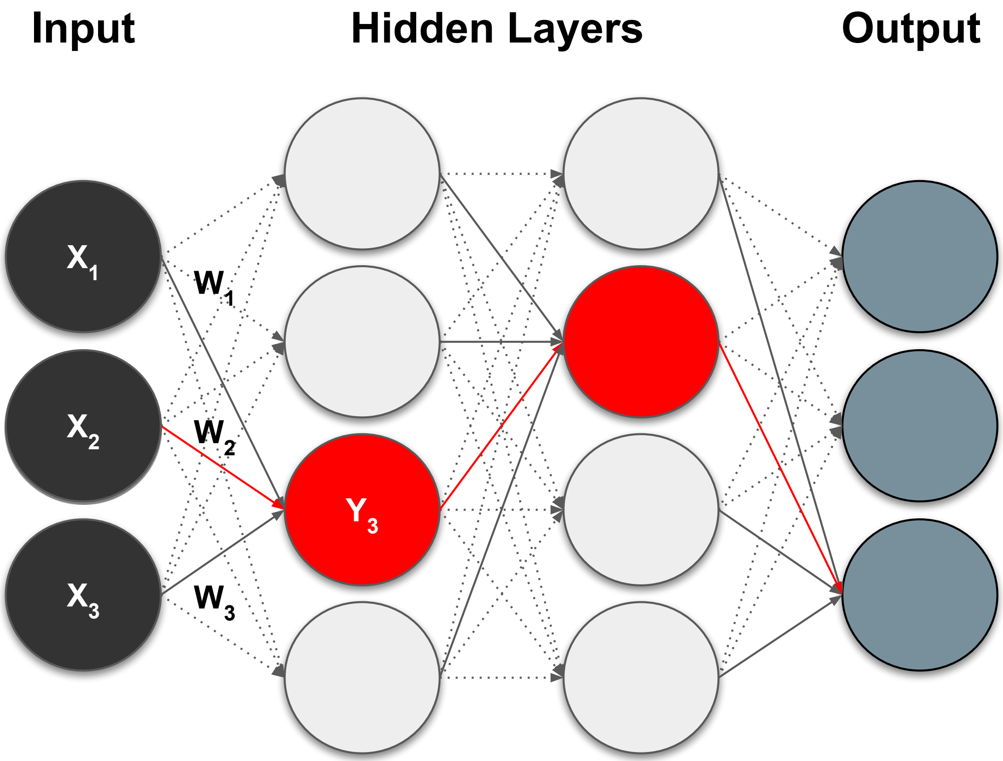
(Image Credit: NVIDIA Deep Learning Institute)

Supervised Deep Learning with Neural Networks

From one layer to the next

$$Y_j = f\left(\sum_i W_i X_i + b_i\right)$$

f is the activation function, W_i is the weight, and b_i is the bias.



Training - Minimizing the Loss

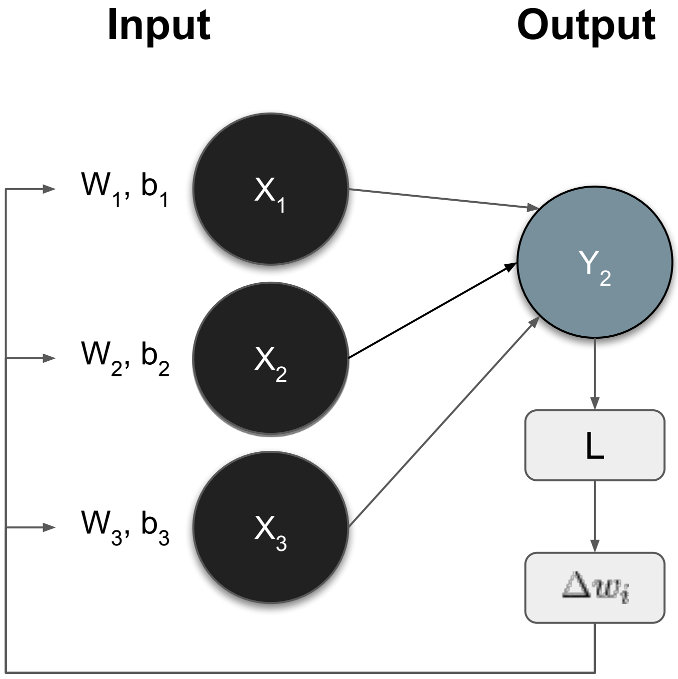
The loss function with regard to weights and biases can be defined as

$$L(\mathbf{w},\mathbf{b}) = \frac{1}{2}\sum_i (\mathbf{Y}(\mathbf{X},\mathbf{w},\mathbf{b}) - \mathbf{Y}'(\mathbf{X},\mathbf{w},\mathbf{b}))^2$$

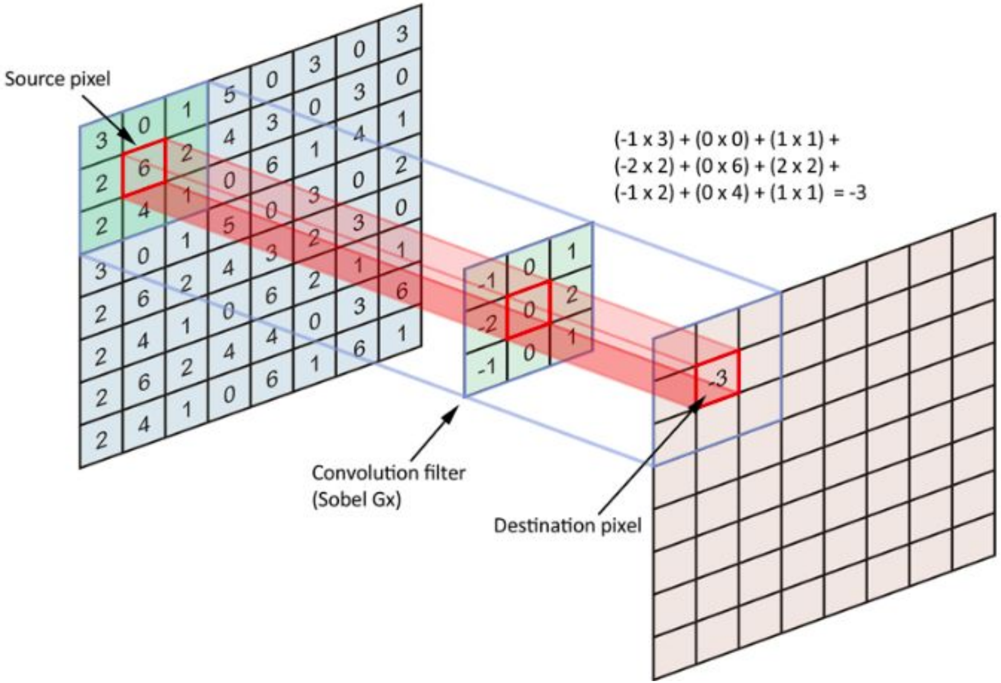
The weight update is computed by moving a step to the opposite direction of the cost gradient.

$$\Delta w_i = -\alpha \frac{\partial L}{\partial w_i}$$

Iterate until L stops decreasing.



Convolution in 2D



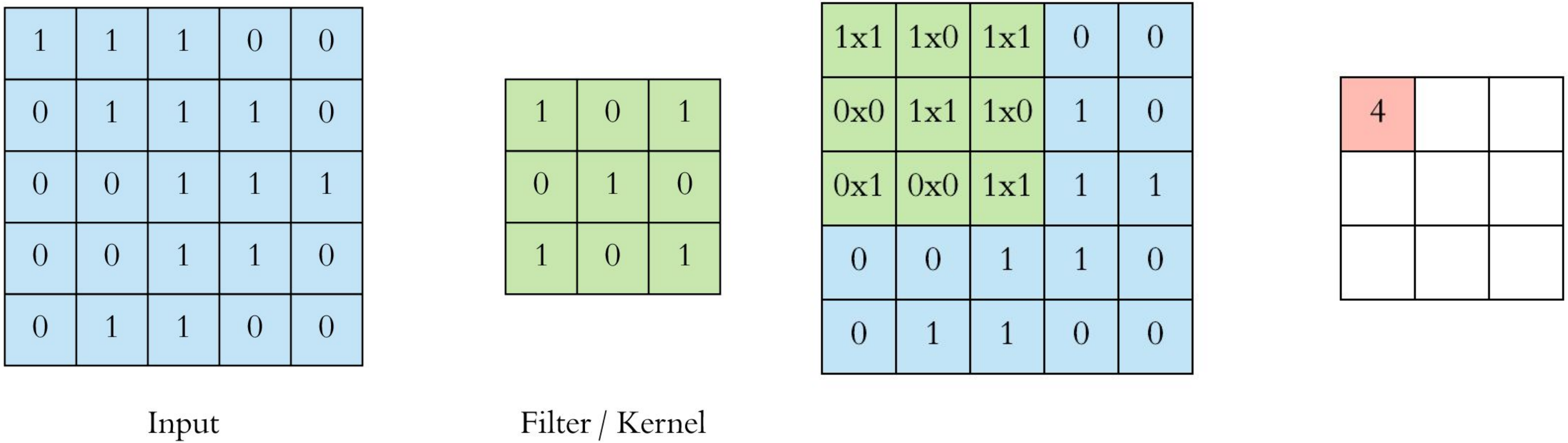
3	0	1	5	0	3	0	3
2	6	2	4	3	0	3	0
2	4	1	0	6	1	4	1
3	0	1	5	0	3	0	2
2	6	2	4	3	2	3	0
2	4	1	0	6	2	1	1
2	6	2	4	4	0	3	6
2	4	1	0	6	1	6	1

-1	0	1
-2	0	2
-1	0	1

(-1 x 3) + (0 x 0) + (1 x 1) +
(-2 x 2) + (0 x 6) + (2 x 2) +
(-1 x 2) + (0 x 4) + (1 x 1) = -3

(Image Credit: Applied Deep Learning | Arden Dertat)

Convolution Kernel



1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Input

1	0	1
0	1	0
1	0	1

Filter / Kernel

1x1	1x0	1x1	0	0
0x0	1x1	1x0	1	0
0x1	0x0	1x1	1	1
0	0	1	1	0
0	1	1	0	0

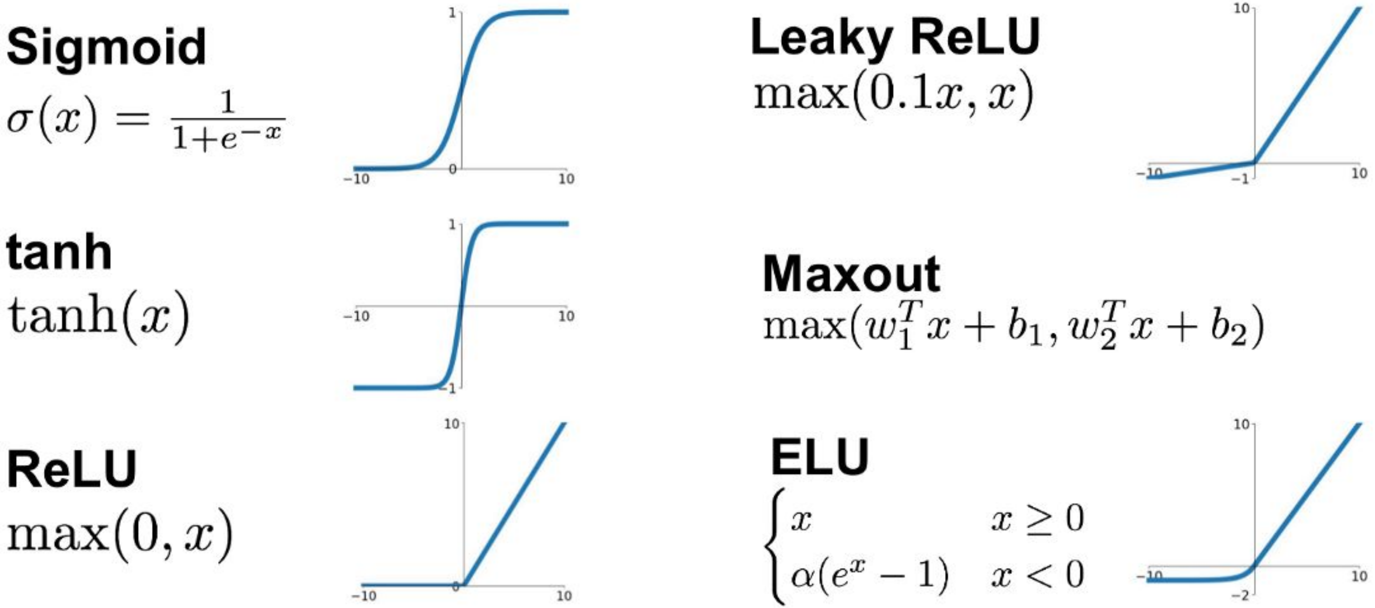
4		

(Image Credit: Applied Deep Learning | Arden Dertat)

Convolution on Image

Image Credit: Deep Learning Methods for Vision | CVPR 2012 Tutorial

Activation Functions



Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$

tanh

$\tanh(x)$

ReLU

$\max(0, x)$

Leaky ReLU

$\max(0.1x, x)$

Maxout

$\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

Image Credit: towardsdatascience.com

Introducing Non-Linearity (ReLU)

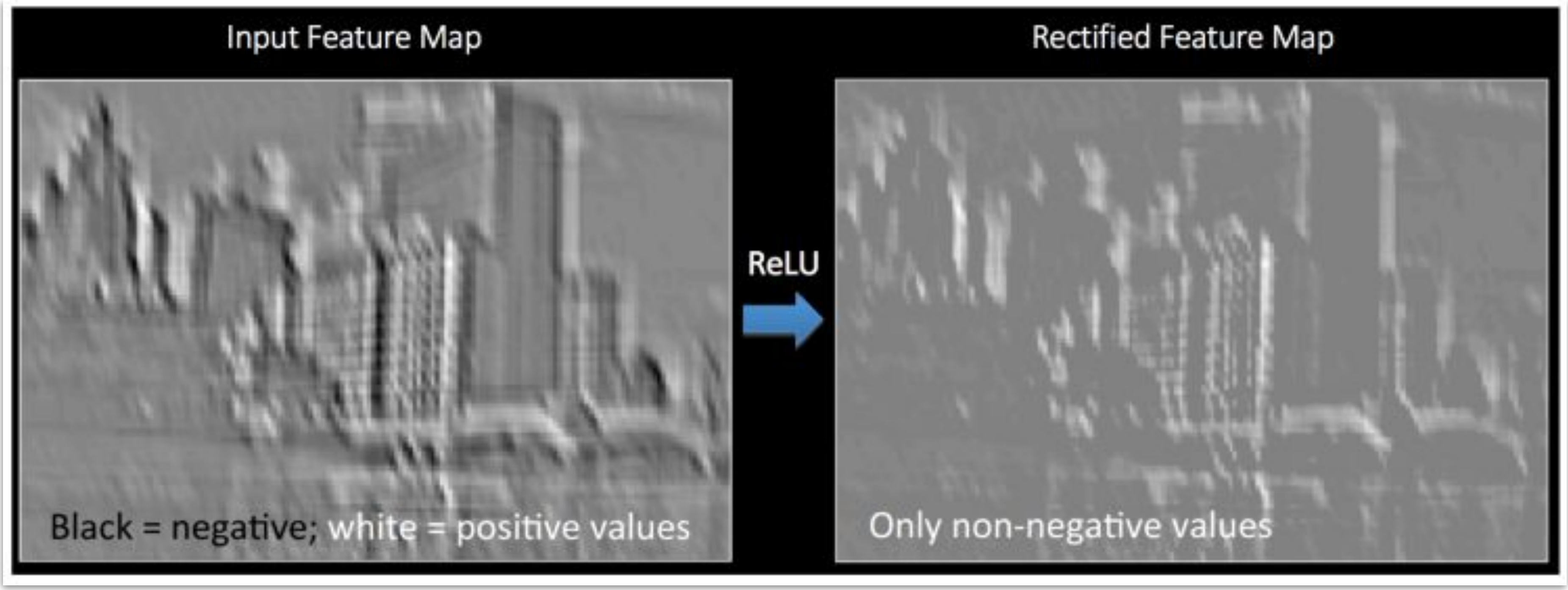
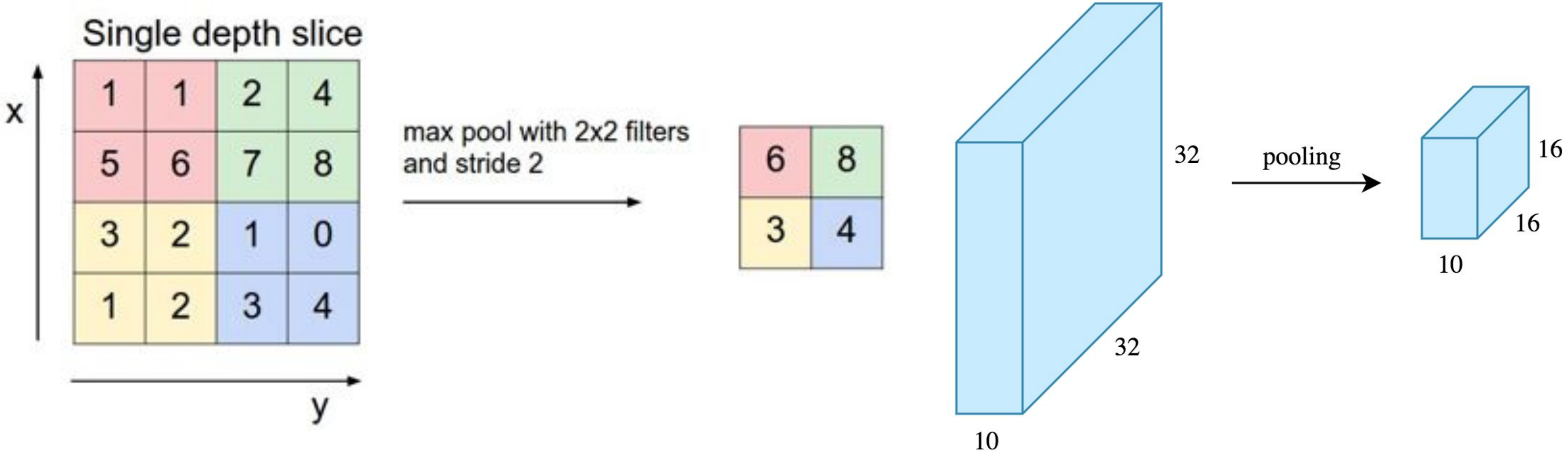


Image Credit: [Deep Learning Methods for Vision | CVPR 2012 Tutorial]

Max Pooling



1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

6	8
3	4

(Image Credit: Applied Deep Learning | Arden Dertat)

Pooling - Max-Pooling and Sum-Pooling

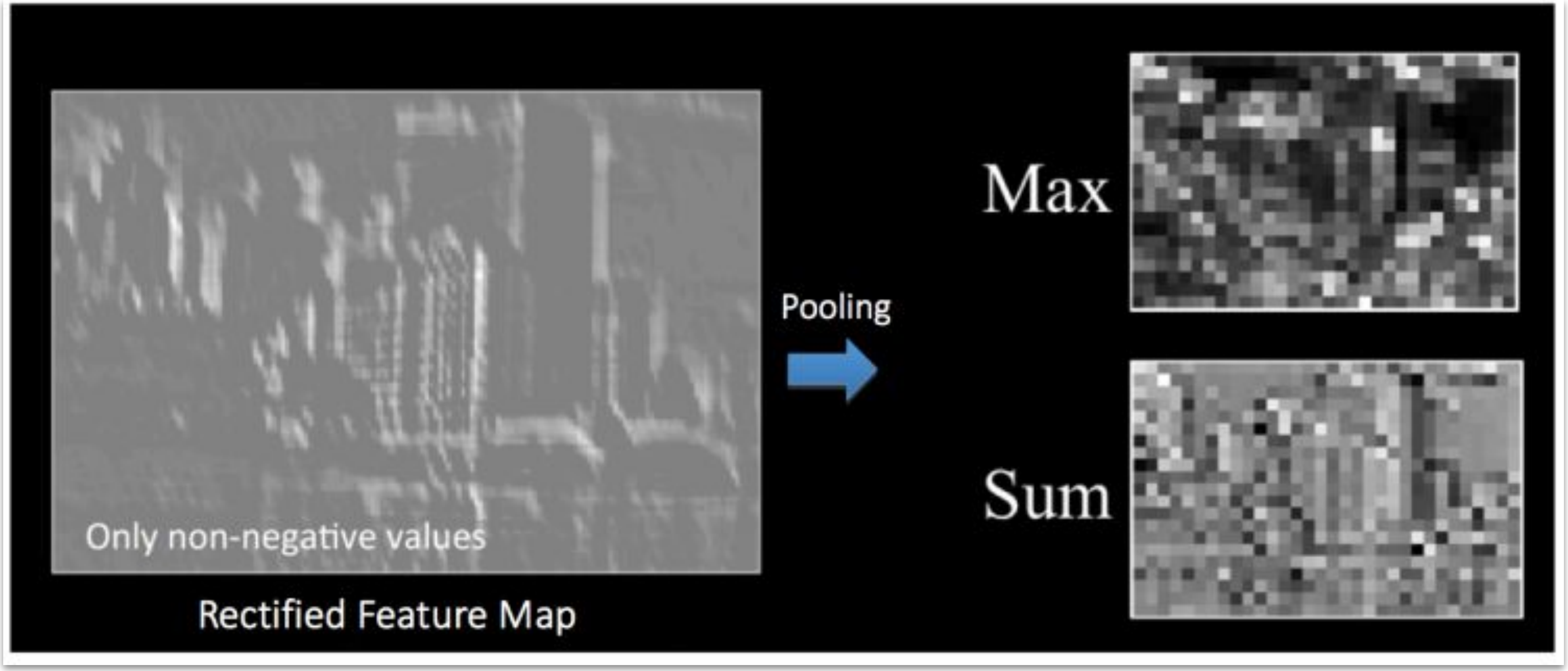
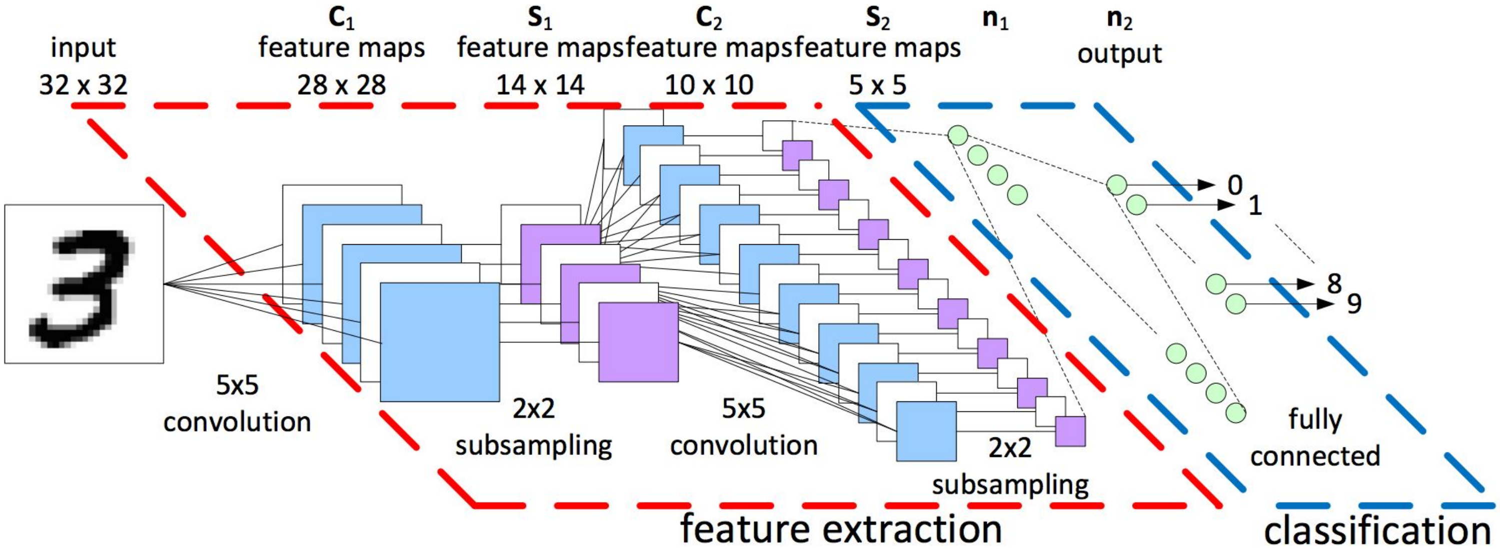


Image Credit: [Deep Learning Methods for Vision | CVPR 2012 Tutorial]()

Convolutional Neural Networks

A convolutional neural network (**CNN**, or **ConvNet**) is a class of deep, feed-forward artificial neural networks that explicitly assumes that the inputs are images, which allows us to encode certain properties into the architecture.



LeNet-5 Architecture (Image Credit: https://becominghuman.ai)

Demos: Introduction to PyTorch and TensorFlow

- Access FASTER (Please follow the instructions in the first session on Tuesday)
- Navigate to your personal scratch directory

```
$ cd $SCRATCH
```

- Files for this course are located at

```
/scratch/training/Intro-deep-learning
```

  Make a copy in your personal scratch directory

```
$ cp -r /scratch/training/Intro-deep-learning $SCRATCH
```

- Enter this directory (your local copy)

```
$ cd Intro-deep-learning
```

Reaching a Compute node

- Execute slurm command to get a compute node

  ```
  $ sbatch intel-jupyterlab-tunnel.slurm
  ```

- View the job output file

  ```
  $ cat intel-jupyterlab.job.*
  ```

- Copy, paste, and execute the ssh command that appears near the top of the output file. Example:

  ```
  $ ssh -4 -L <port>:localhost:<port> <nodename>
  ```

Open Jupyter Lab in Browser

- Towards the end of the job output file (viewed like this)

```
$ cat intel-jupyterlab.job.*
```

- login instructions will appear. Example:

```
To access the server, open this file in a browser:

    file:///home/<username>/.local/share/jupyter/runtime/jpserver-462321-open.html

Or copy and paste one of these URLs:

    http://localhost:<port>/lab?token=67b0e1263053b6bc449c59999984bbfc30a97fa61fcd9e18

 or http://127.0.0.1:<port>/lab?token=612b6b0iiic840c449c5a97fa61bbfc3fcd9e7b630530e18
```

- Copy and paste the link into your browser.
- In the JupyterLab file explorer, click on the folder ***Intro-deep-learning***.