# Deep Learning - Introduction

# Syllabus

| Syllabus: | Teaching Hours |
|---|---|
| **UNIT I Review of Artificial Neural Networks:** Perceptron Learning, Feed Forward Neural Networks, Backpropagation, Unstable Gradient Problem, Limitations of Feed Forward Neural Networks for Computer Vision Problems | 6 |
| **UNIT 2 Convolutional Neural Networks:** Convolution & Pooling, Dropout, Batch Normalization, State-of-the-art CNNs | 9 |
| **UNIT 3 Transfer Learning & Domain Adaptation:** Transfer Learning Scenarios, Applications of Transfer Learning, Transfer Learning Methods, Fine Tuning and Data Augmentation, Supervised, Semi Supervised and Unsupervised Deep Learning | 5 |
| **UNIT 4 Convolutional Neural Networks for Computer Vision:** Image Classification, Image Classification with Localization, Semantic Segmentation, Object Detection | 9 |
| **UNIT 5 Sequence Models:** Recurrent Neural Networks (RNN), Language Modeling, Long-Short Term Memory Network, Gated Recurrent Unit, Bi-directional RNN, Deep RNN, Applications of Sequence Models | 9 |
| **UNIT 6 Miscellaneous:** Auto encoders and Stacked Auto encoders, Generative Adversarial Networks, Deep Reinforcement Learning | 10 |

# Referencess


1. Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press

2. Charu C. Aggarwal, Neural Networks and Deep Learning – A Textbook, Springer

3. Adam Gibson, Josh Patterson, Deep Learning, O'Reilly Media, Inc.

4. Duda, R.O., Hart, P.E., and Stork, D.G., Pattern Classification, Wiley.

5. Theodoridis, S. and Koutroumbas, K., Pattern Recognition. Academic Press

6. Russell, S. and Norvig, N. Artificial Intelligence: A Modern Approach. Prentice Hall Series in Artificial Intelligence

# References

7. Bishop, C. M. Neural Networks for Pattern Recognition. Oxford University Press.

8. Hastie, T., Tibshirani, R. and Friedman, J. The Elements of Statistical Learning, Springer

9. Koller, D. and Friedman, N. Probabilistic Graphical Models. MIT Press

10. Richard Szeliski, Computer Vision: Algorithms and Applications, Springer

***11. Research Papers and Web Links***

# Blog and Course Site

Course Site:
https://sites.google.com/a/nirmauni.ac.in/it7f4---deep-learning/

# Teaching & Evaluation Scheme

## Teaching Scheme:

| Theory | Tutorial | Practical | Credits |
|---|---|---|---|
| 3 | 0 | 2 | 4 |

## Evaluation Scheme:

| | LPW | SEE | CE |
|---|---|---|---|
| Exam Duration | Continuous Evaluation + 2 Hrs. End Semester Exam | 3.0 Hrs. | Continuous Evaluation |
| Component Weightage | 0.2 | 0.4 | 0.4 |

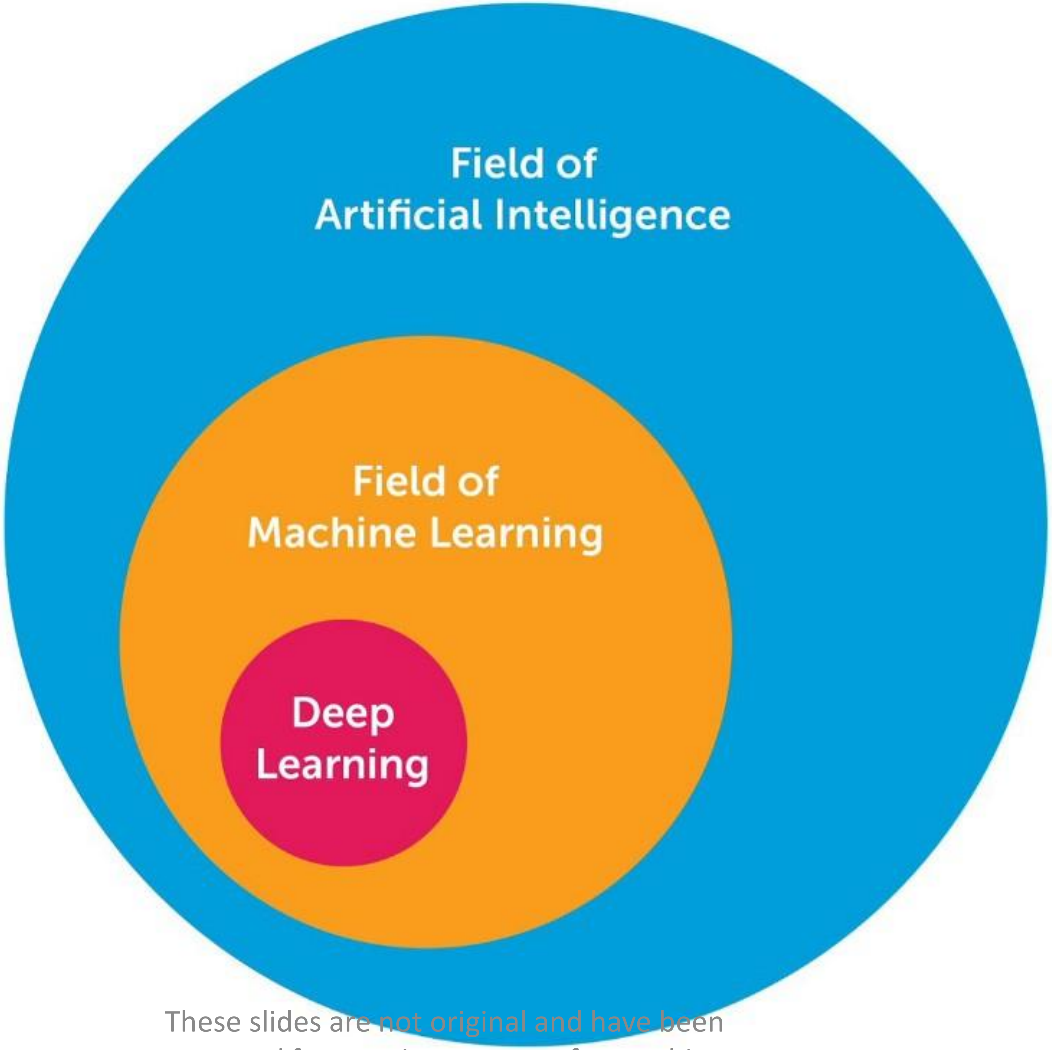
# Teaching & Evaluation Scheme

## Breakup of CE

| | Unit 1 | Unit 2 | Unit 3 |
|---|---|---|---|
| Exam | Class Test | Sessional Exam | Assignments |
| Inter Component Weightage | 0.35 | 0.35 | 0.3 |
| Numbers | 1 | 1 | 1 |
| Marks of Each | 35 | 50 | 30 |

# Introduction

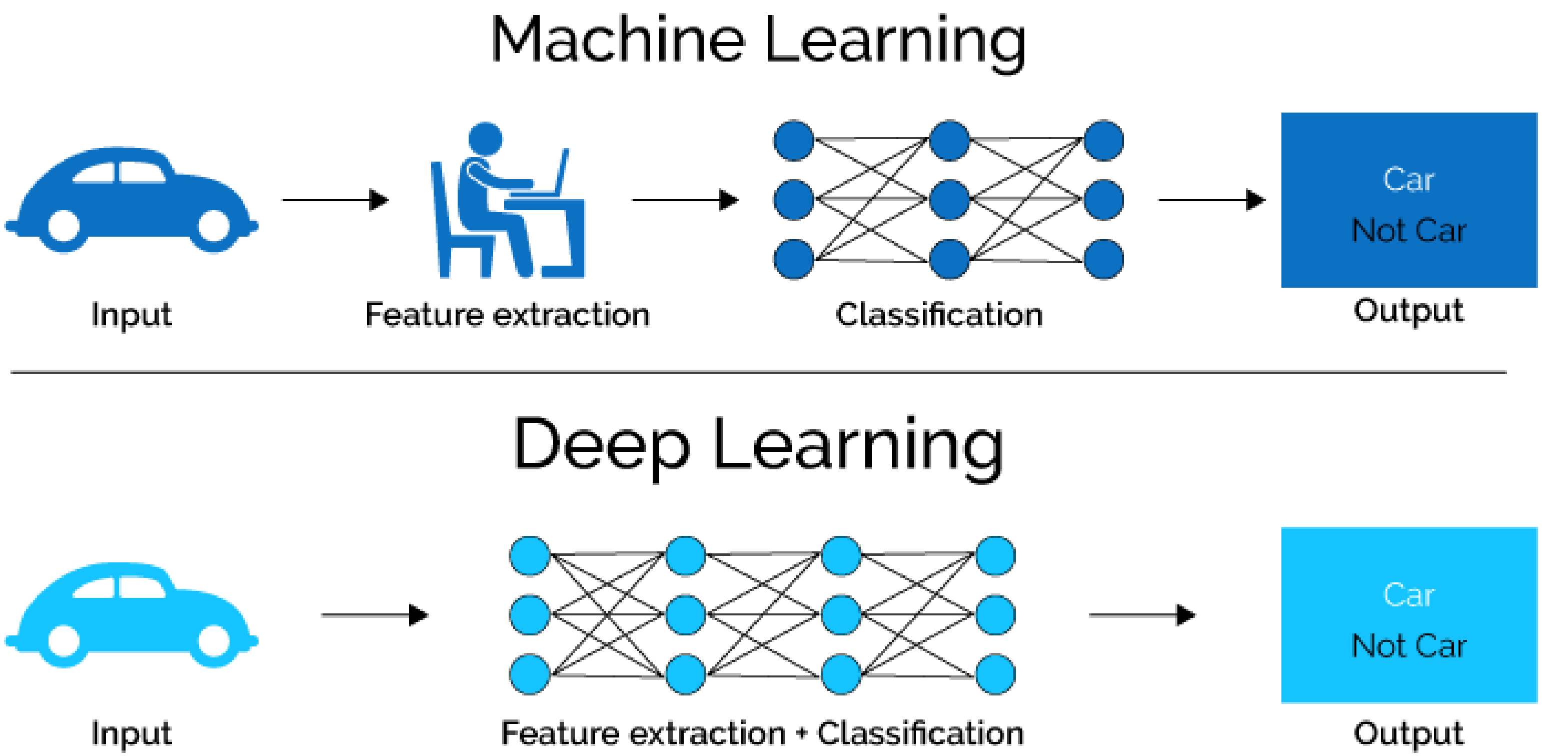
- AI, ML and DL




Source: [1]

# Introduction

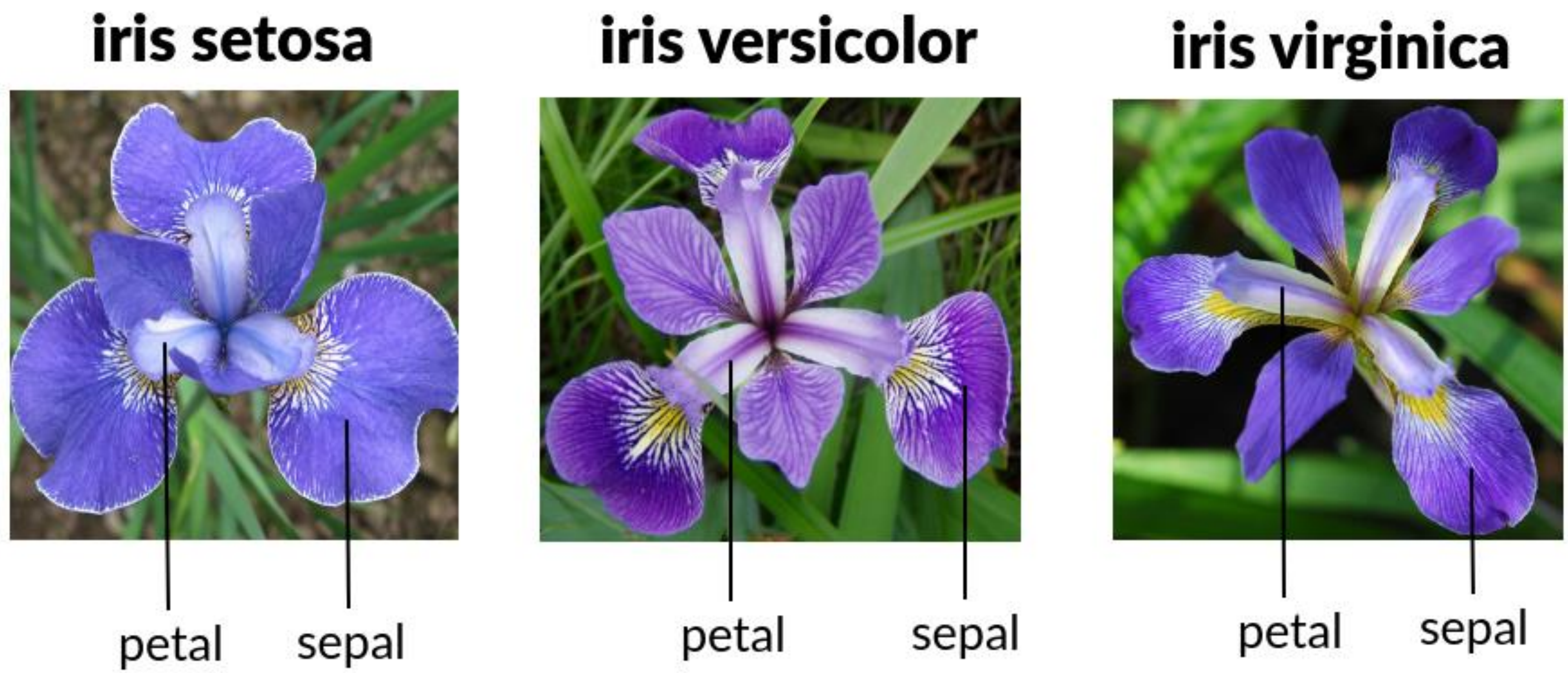
➢ Machine Learning vs Deep Learning




Source: [2]

# Introduction

➢ Machine Learning vs Deep Learning




Source of the Image:
https://editor.analyticsvidhya.com/uploads/51518iris%20img1.png

# Major Architectures of Deep Networks

- Four Major Architectures:
  - Unsupervised Pretrained Networks (UPNs)
  - Convolutional Neural Networks (CNNs)
  - Recurrent Neural Networks
  - Recursive Neural Networks

# Major Architectures of Deep Networks

- Four Major Architectures:
  - Unsupervised Pretrained Networks (UPNs)
    - Autoencoders
    - Deep Belief Networks (DBNs)
    - Generative Adversarial Networks (GANs)

    - Use Cases:
      - Feature Extraction
      - Initialization
      - Synthesizing

Source: [3]

# Major Architectures of Deep Networks

- Four Major Architectures:
  - Convolutional Neural Networks (CNNs)
    - Lenet-5
    - AlexNet
    - VGGNet
    - GoogleNet (Inception)
    - ResNet
    - ResNext
    - DenseNet
    - RCNN (Region Based CNN)
    - YOLO (You Only Look Once)
    - SqueezeNet
    - SegNet

# Major Architectures of Deep Networks

- Four Major Architectures:
  - Convolutional Neural Networks (CNNs)
  - Use Cases:
    - Computer Vision
    - Natural Language Processing

# Major Architectures of Deep Networks

- Four Major Architectures:
  - Recurrent Neural Networks
    - Hopfield Network
    - Long Short-Term Memory (LSTM)
    - Gated Recurrent Unit (GRU)
  - Use Cases:
    - Sentiment Classification
    - Image Captioning
    - Language Translation
    - Video Captioning

# Major Architectures of Deep Networks

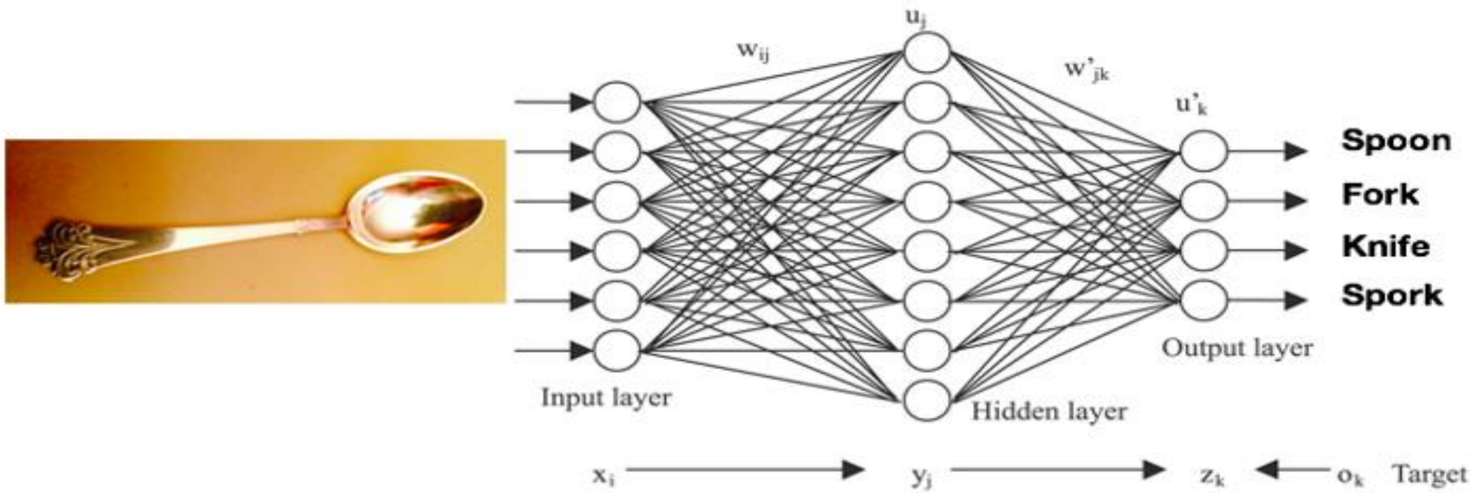
- Four Major Architectures:
  - Recursive Neural Networks
    - Recursive Autoencoder
    - Recursive Neural Tensor Network
  - Use Cases:
    - Image scene decomposition
    - NLP
    - Audio-to-text transcription

# Computer Vision & Vanilla Neural Networks

- Feature Engineering
- Loss of Structural Information
- Difference in Indented Part, Orientation, Backdrop, Size, Location
- Noise
- Scalability

# Computer Vision & Vanilla Neural Networks

- Loss of Structural Information



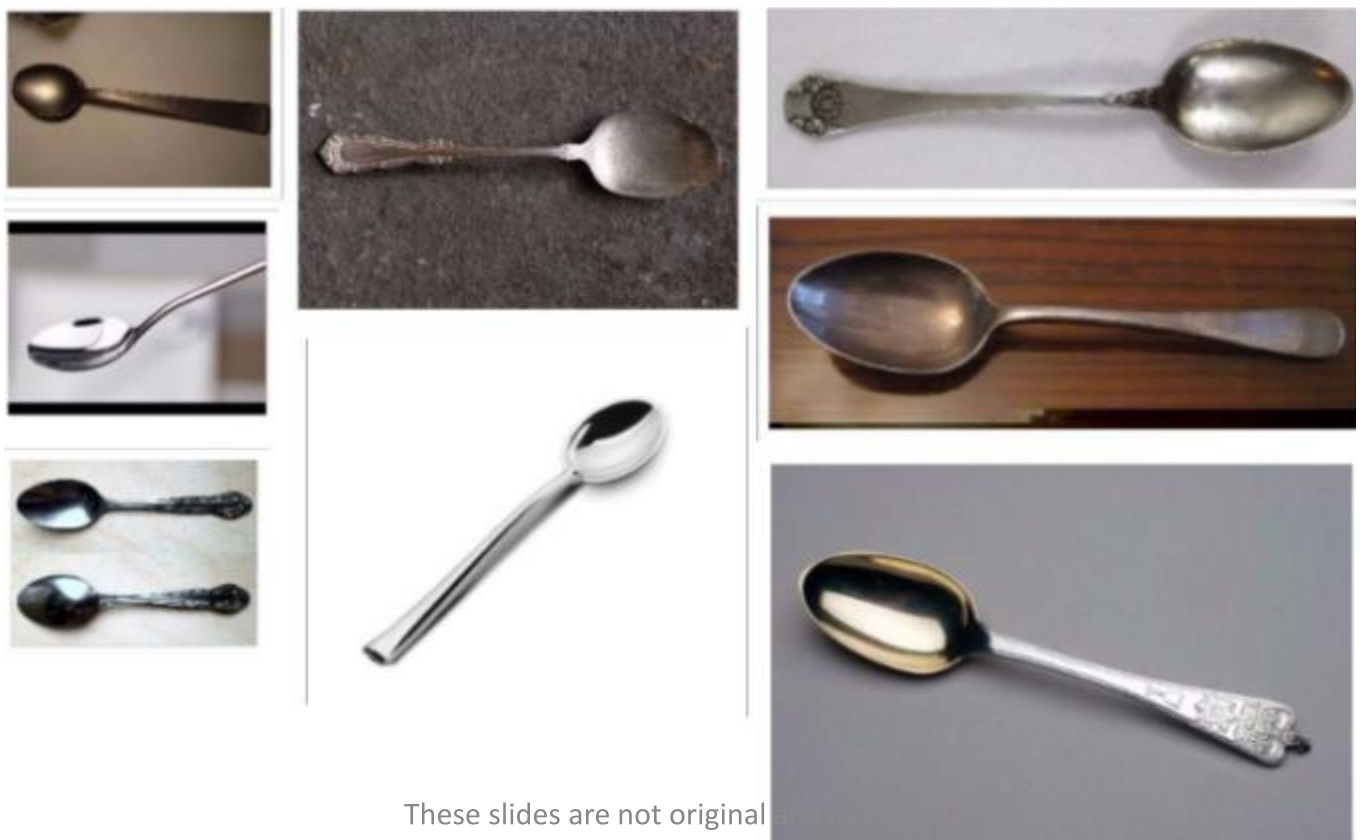


$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Pixel Vector

# Computer Vision & Vanilla Neural Networks

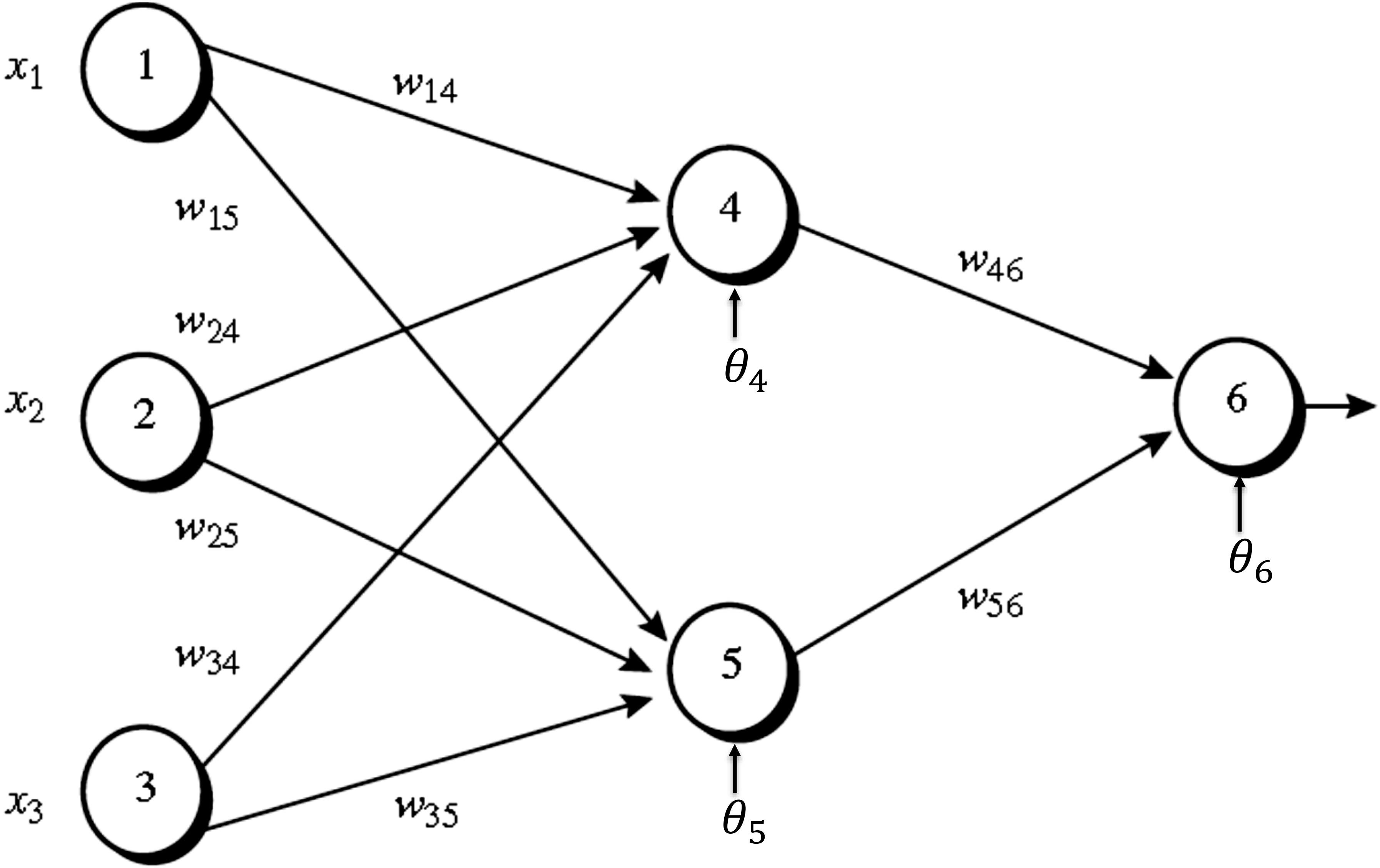
- Difference in Indented Part, Orientation, Backdrop, Size, Location
- Noise



# Computer Vision & Vanilla Neural Networks

- Scalability

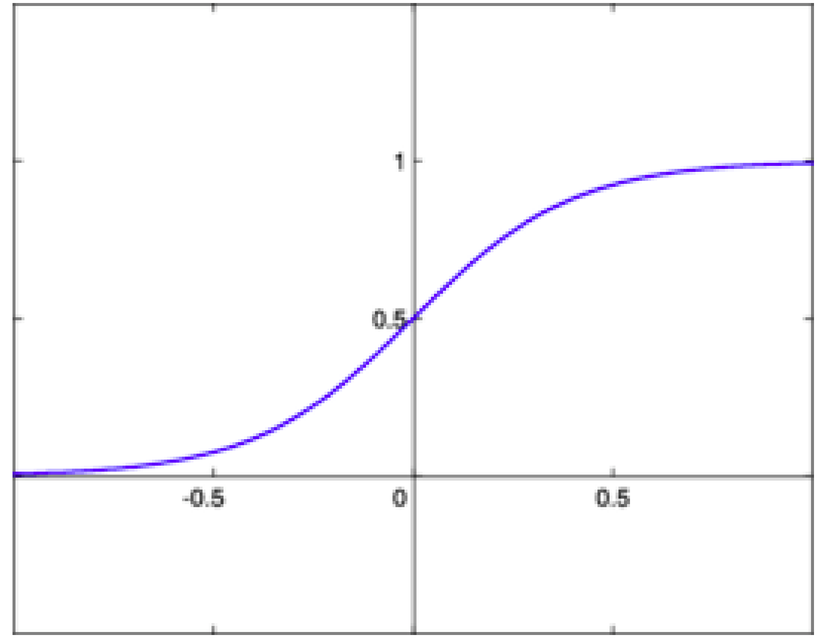
# Backpropagation



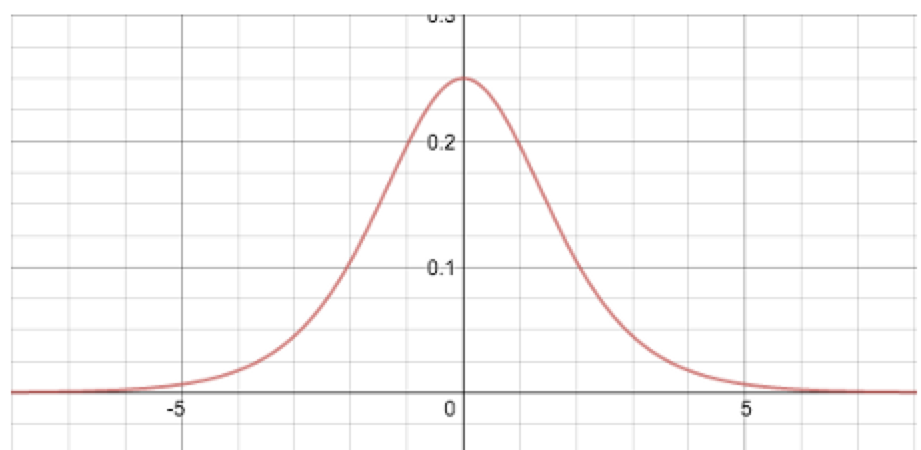

An example of a multilayer feed-forward neural network.

# Vanishing Gradient Problem

$$Sigmoid = S(\alpha) = \frac{1}{1+e^{-\alpha}}$$




$$\frac{1}{1+e^{-\alpha}}\left[1-\frac{1}{1+e^{-\alpha}}\right]$$




Simply: S(1-S)

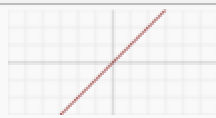
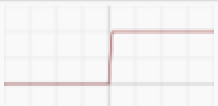
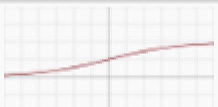
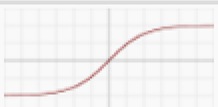
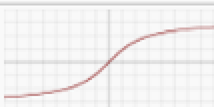
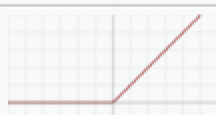
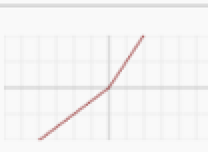
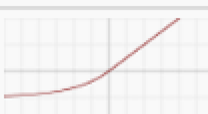
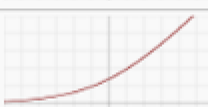
These slides are not original and have been prepared from various sources for teaching purpose

Note: Images are not original

# Vanishing Gradient Problem

- How does ReLU solve (delay) the problem?
- Dead Neuron in case of RELU and its implication
- Leaky/Parameterized ReLU

# Vanishing Gradient Problem

| Name | Plot | Equation | Derivative |
|---|---|---|---|
| Identity |  | $f(x) = x$ | $f'(x) = 1$ |
| Binary step |  | $f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$ | $f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$ |
| Logistic (a.k.a Soft step) |  | $f(x) = \frac{1}{1+e^{-x}}$ | $f'(x) = f(x)(1 - f(x))$ |
| TanH |  | $f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$ | $f'(x) = 1 - f(x)^2$ |
| ArcTan |  | $f(x) = \tan^{-1}(x)$ | $f'(x) = \frac{1}{x^2+1}$ |
| Rectified Linear Unit (ReLU) |  | $f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$ | $f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$ |
| Parameteric Rectified Linear Unit (PReLU)[2] |  | $f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$ | $f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$ |
| Exponential Linear Unit (ELU)[3] |  | $f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$ | $f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$ |
| SoftPlus |  | $f(x) = \log_e(1 + e^x)$ | $f'(x) = \frac{1}{1+e^{-x}}$ |

# Disclaimer

- These slides are not original and have been prepared from various sources for teaching purpose.

# References


1. https://towardsdatascience.com/the-10-deep-learning-methods-ai-practitioners-need-to-apply-885259f402c1
2. https://semiengineering.com/deep-learning-spreads/
3. https://www.safaribooksonline.com/library/view/deep-learning/9781491924570/ch04.html

# References


4. Data mining: concepts and techniques, J. Han, and M. Kamber. Morgan Kaufmann, (2006)

5. Elements of Artificial Neural Networks, Kishan Mehrotra, Chilukuri K. Mohan, Sanjay Ranka. MIT Press, (1997)

6. Matlab Neural Network Tollbox Documentation

7. LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998): 2278-2324.

8. Srivastava, Nitish, et al. "Dropout: A simple way to prevent neural networks from overfitting." The Journal of Machine Learning Research 15.1 (2014): 1929-1958.

# References

9. https://ayearofai.com/rohan-lenny-2-convolutional-neural-networks-5f4cd480a60b

10. http://cs231n.github.io/convolutional-networks/

11. https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

12. Xu, Bing, et al. "Empirical evaluation of rectified activations in convolutional network." arXiv preprint arXiv:1505.00853 (2015).

# Disclaimer

- These slides are not original and have been prepared from various sources for teaching purpose.