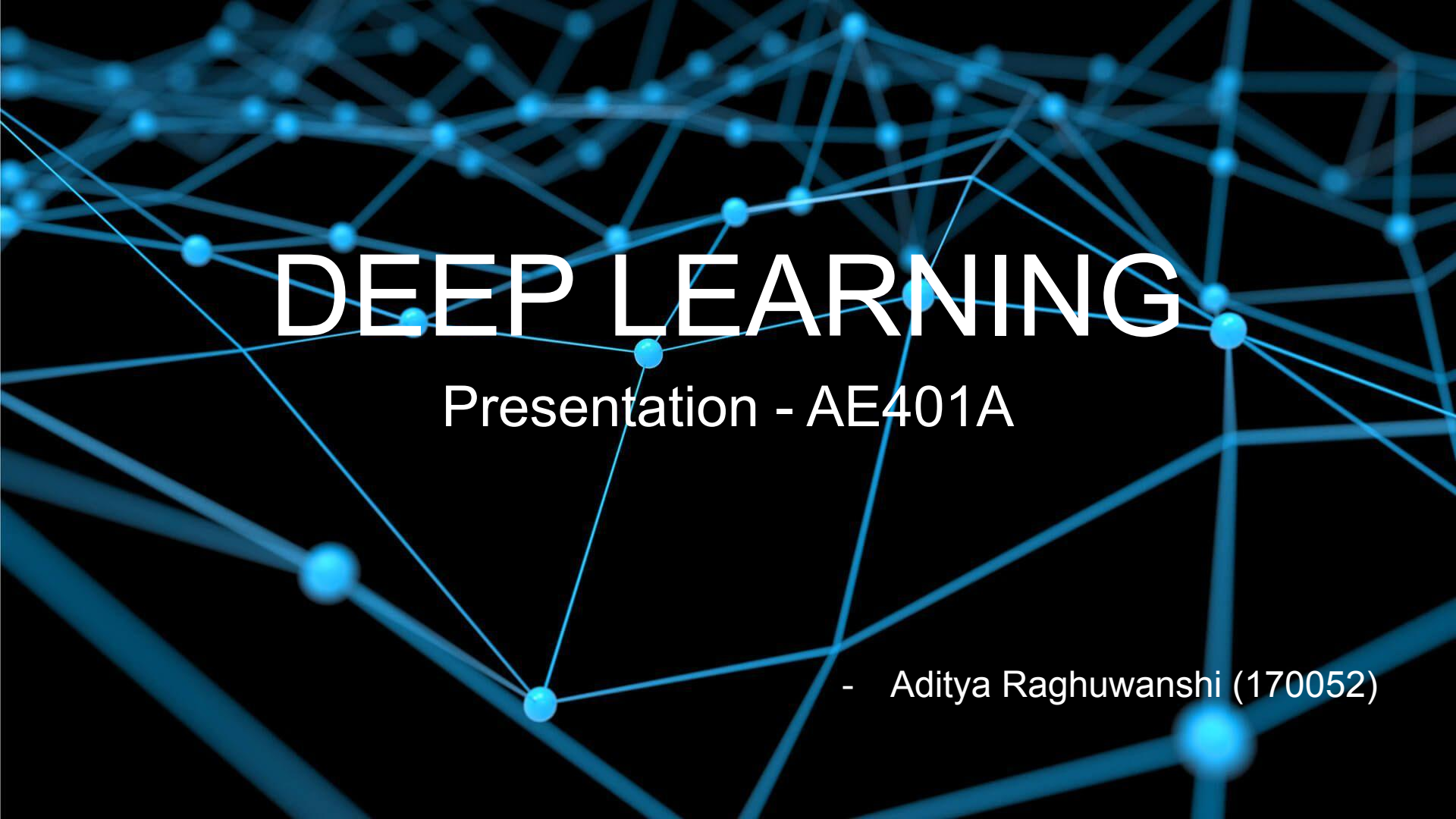
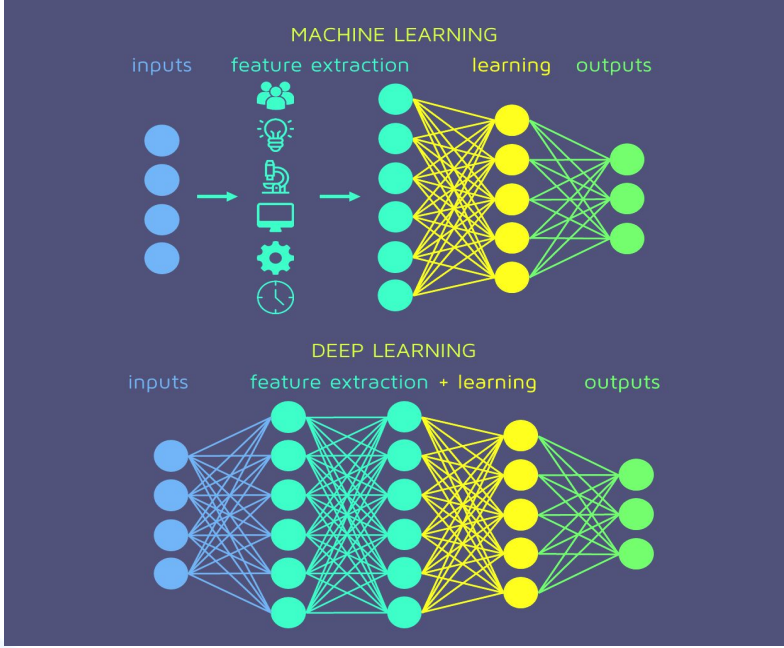
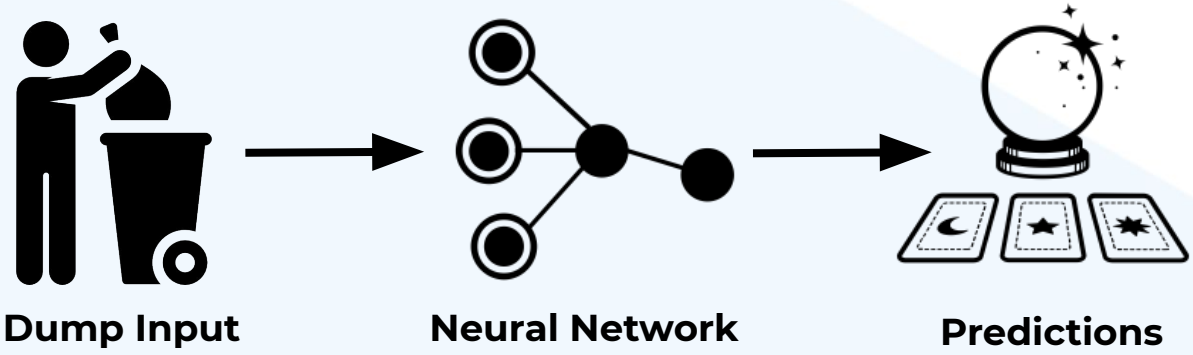

DEEP LEARNING
Presentation - AE401A
- Aditya Raghuwanshi (170052)

Introduction

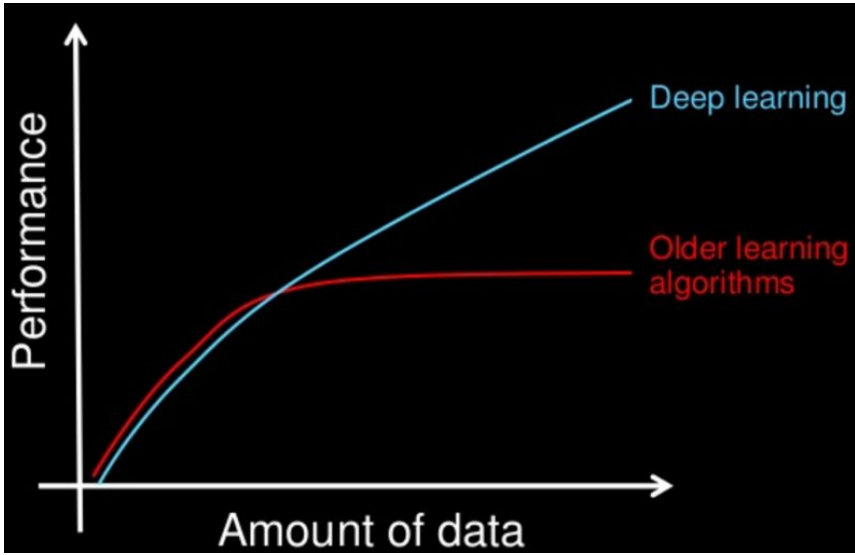
- Deep Learning uses a “Neural Networks” architecture to build models.
- NN architecture is designed to work the same way a **human brain** does.
- Deep Learning, unlike other machine learning techniques, performs **feature engineering**.
- Deep Learning suits our ideal machine learning model imagination, “**dump the data into the model and get prediction**”.





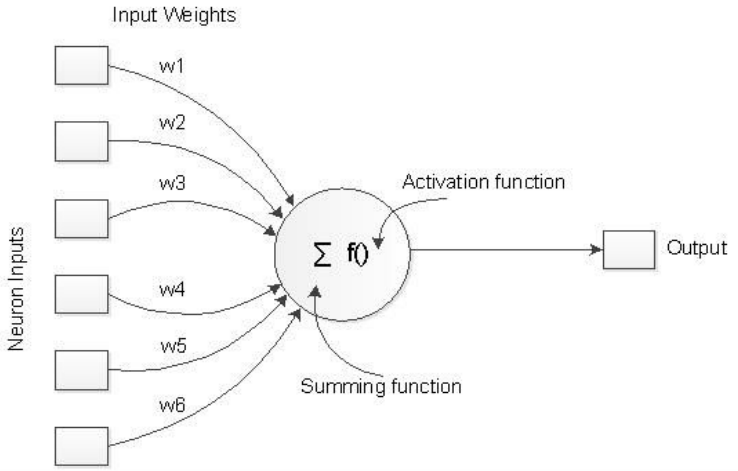
Motivation

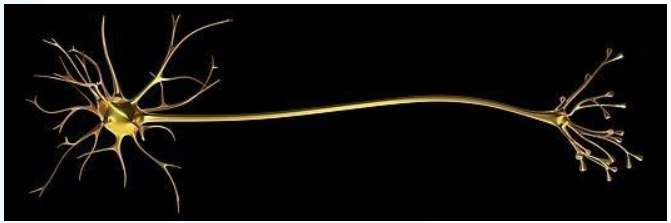
- DL outperform other techniques if the data size is large compared to other algos.
- Deep Learning is fueled by “**Big Data Era**” and advancements in computational power.
- It is used by world’s largest companies.



A Single Neuron (Node)

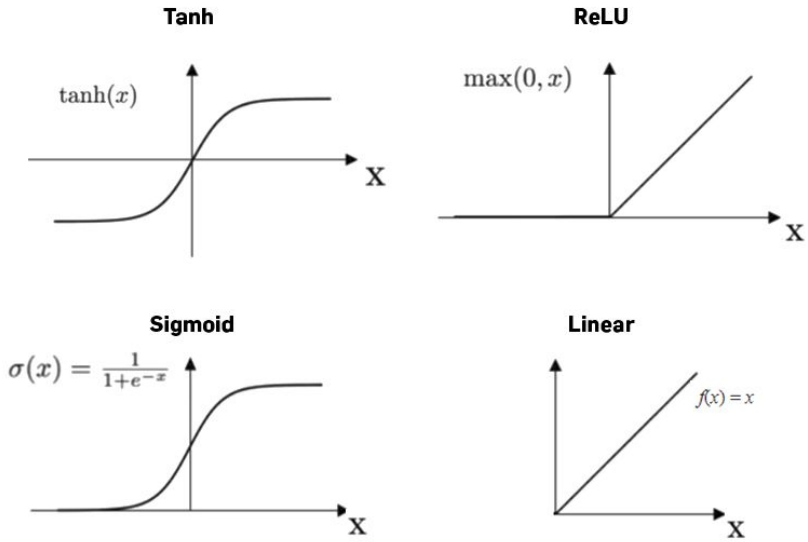
- It is the fundamental unit of neural network
- It performs a simple **linear regression**, more specifically does 2 operations:
  1. Performs weighted sum of input and a bias term
  2. Maps calculated sum to an activation function
- Each neuron gets **activated** (or fire) by some feature(s) in the input data.
- Takes output of previous layers as input, its output is used as input for next layer.
- All weights and bias terms are the **parameters** to be learned by the model.





Activation Functions

- Theoretically, output of a neuron can varies from -∞ to +∞.
- Act. Fun. maps input to a **desired range** that avoids **overflow** and also outputs values which are more useful.
- **Nonlinear function** are, usually, used to learn complex models
  - It can also learn probabilities of a class
  - They allow **back propagation**
- Linear act. fun. are not useful
- Most widely used activation functions are:
  - Sigmoid
  - Tanh
  - ReLu (Rectified Linear Unit)
  - Leaky ReLu



Building A Neural Networks

- Image shows a simple neural network built with:
  - Input Layer, having 3 nodes
  - A Hidden Layer, having 2 nodes
  - Output Layer with single node
- For layer 1, inputs = [x_1, x_2 ,x_3]

  $a_1 = w_{11}x_1+w_{12}x_2+w_{13}x_3$

  $a_2 = w_{21}x_1+w_{22}x_2+w_{23}x_3$

  $h_1=f(a_1)$ and $h_2=f(a_2)$

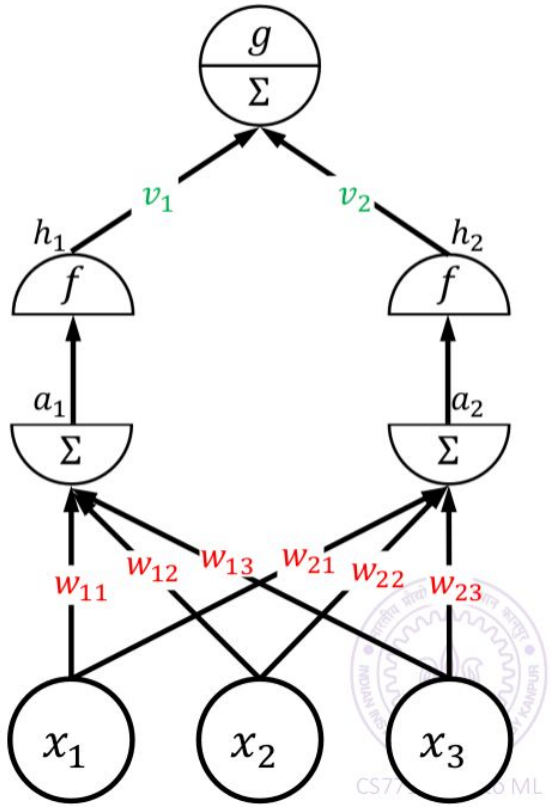
  Outputs = [h_1, h_2]

  For output layer, inputs = [h_1, h_2]

  $b = v_1h_1+v_2h_2$

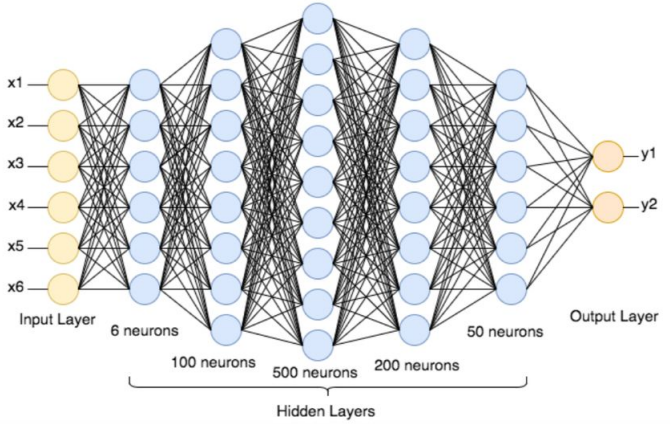
  Output = g(b)

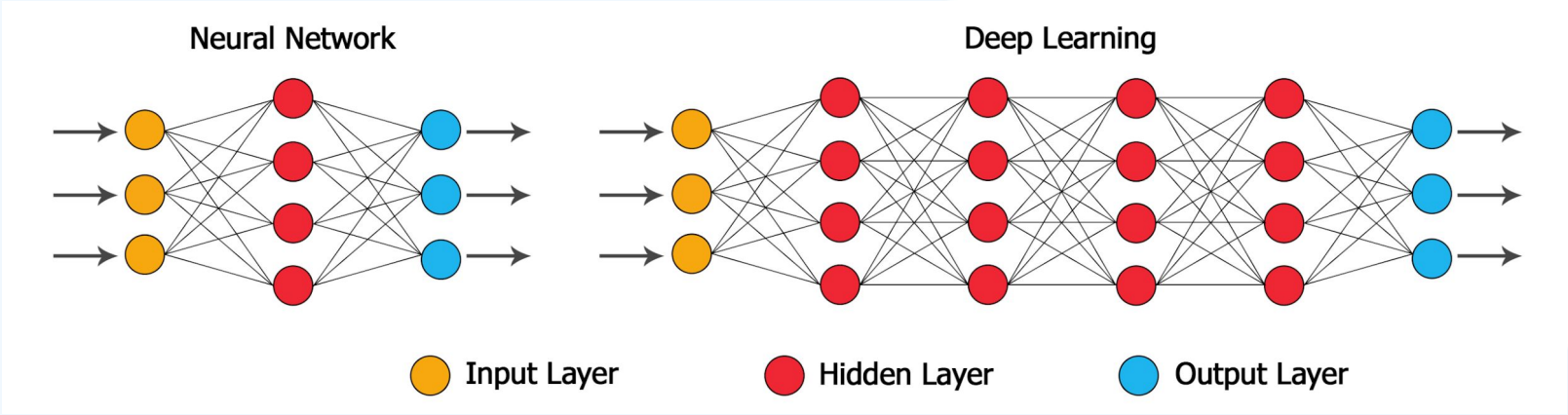
  “f” and “g” are activation functions
- This is known as **Forward Propagation**



Building A “**Deep**” Neural Networks

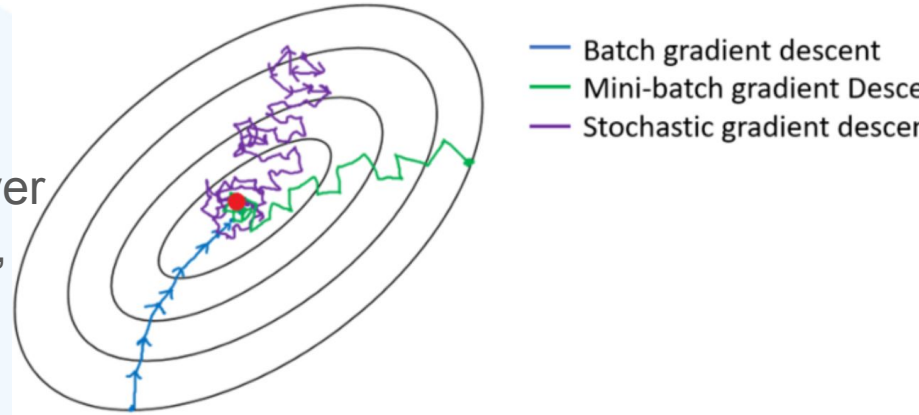
- Many neurons are stacked to form a layer and many layers are stacked to form NN.
- Number of neurons in each layer and number of layers (depth) can vary and are **hyperparameter**.
- “Deep” in deep learning actually refers to large “depth” of neural architecture.





Training of a Deep NN

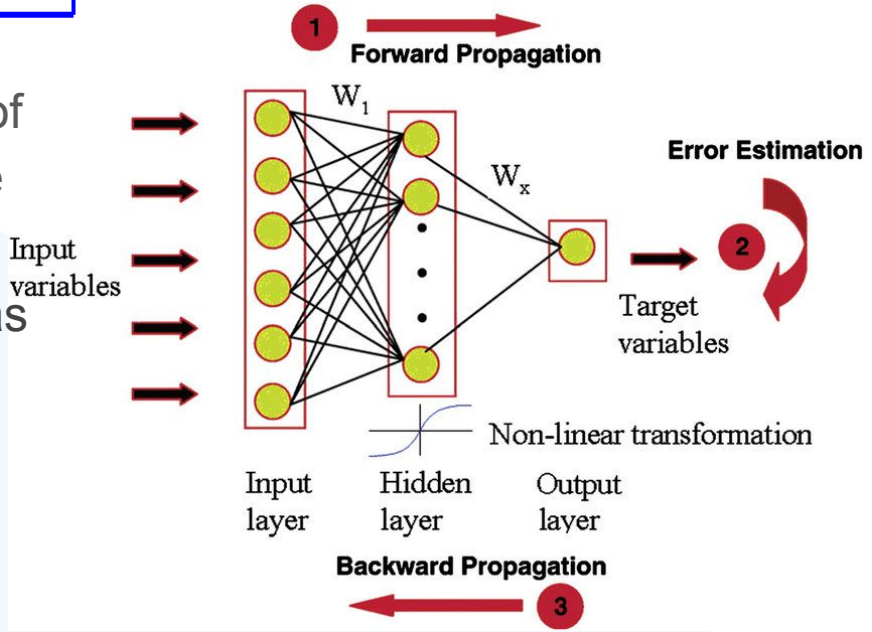
- **Stochastic Gradient Descent** is an optimizing algorithm that adjusts the weights to minimize the loss. It is iterative algorithms which performs steps.
  - Make predictions on some data (batch)
  - Compute Loss on the predictions
  - Update parameters to minimise the loss
- **Iterate** on these step until loss is below certain threshold value.
- Predictions are made using **Forward Propagation**, where we sweeps all layer starting from input layer to output layer, calculating outputs of all neurons.



Back Propagation in NN

- Loss are computed using a function called **Loss Function**.
- In Back Prop, model sweeps layers in reverse order and find partial derivative of loss function wrt. parameters and update them to minimise the loss function.
- This is step is **heart of Deep Learning** as "**learning**" happens here.
- **Learning Rate** and **Batch Size** are hyperparameters needed to be tuned using validation data.

symbol	name	equation
$\mathcal{L}_1$	L_1 loss	$\|\mathbf{y}-\mathbf{o}\|_1$
$\mathcal{L}_2$	L_2 loss	$\|\mathbf{y}-\mathbf{o}\|_2^2$
log	log (cross entropy) loss	$-\sum_j \mathbf{y}^{(j)} \log \sigma(\mathbf{o})^{(j)}$
$\log^2$	squared log loss	$-\sum_j [\mathbf{y}^{(j)} \log \sigma(\mathbf{o})^{(j)}]^2$

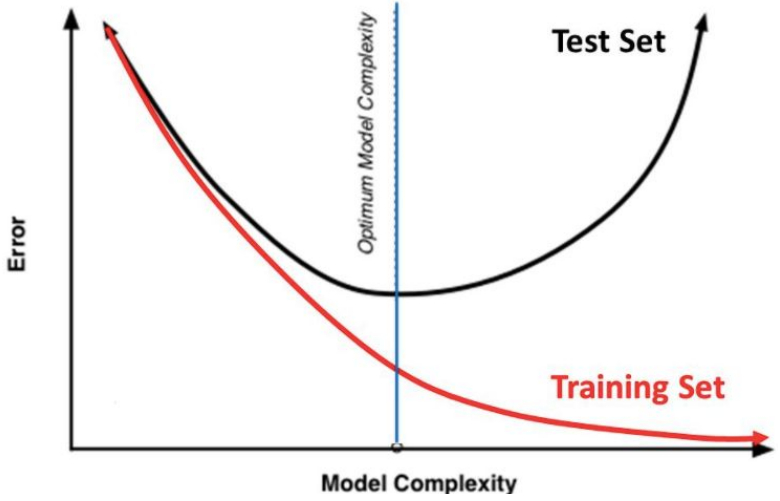


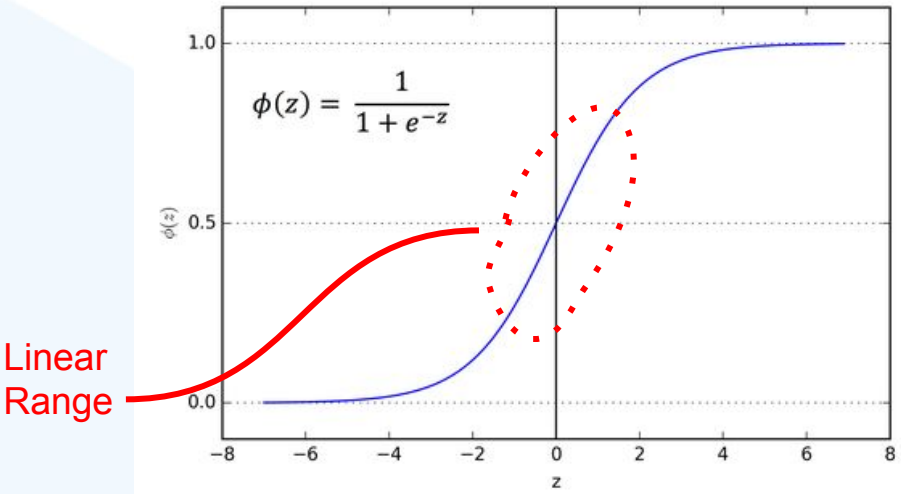
Overfitting

- Deep Neural Networks performs “too well” on not so complex data which drops performance on test data.
- Solution: **Regularization**
  - Weight Decay Method
  - Dropout Method
  - Early Stopping
- **Weight Decay Method**

Total Loss: $L = (\sum_i L_i^2 + \lambda . \sum_i ||w||_F^2) / m$
It brings activation closer to the origin.

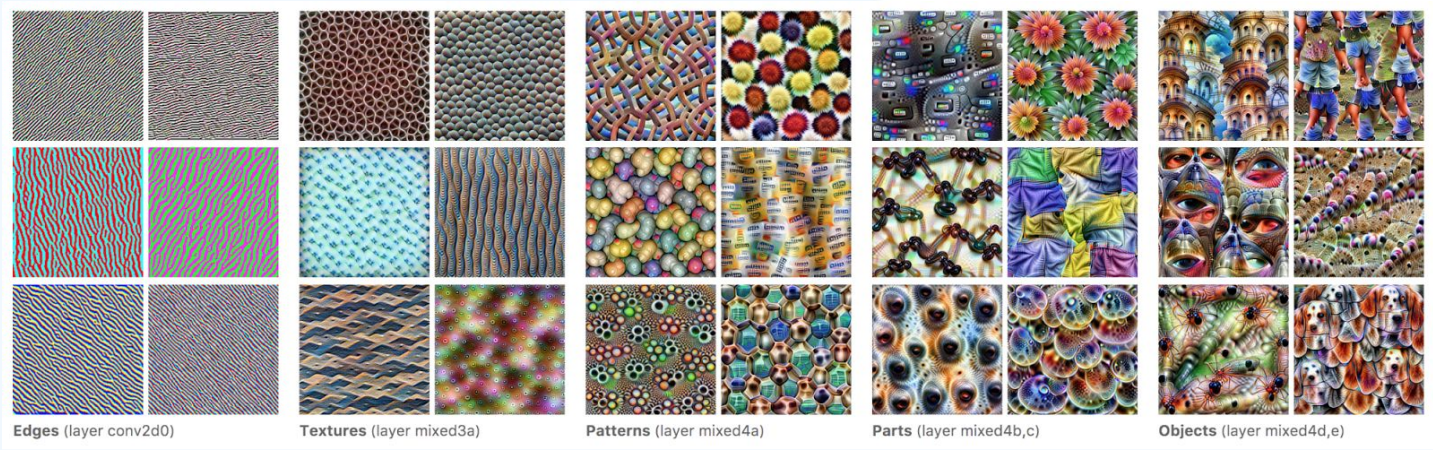
So, as area near origin lies in linear range and model **CANNOT** learn complex nonlinear models.



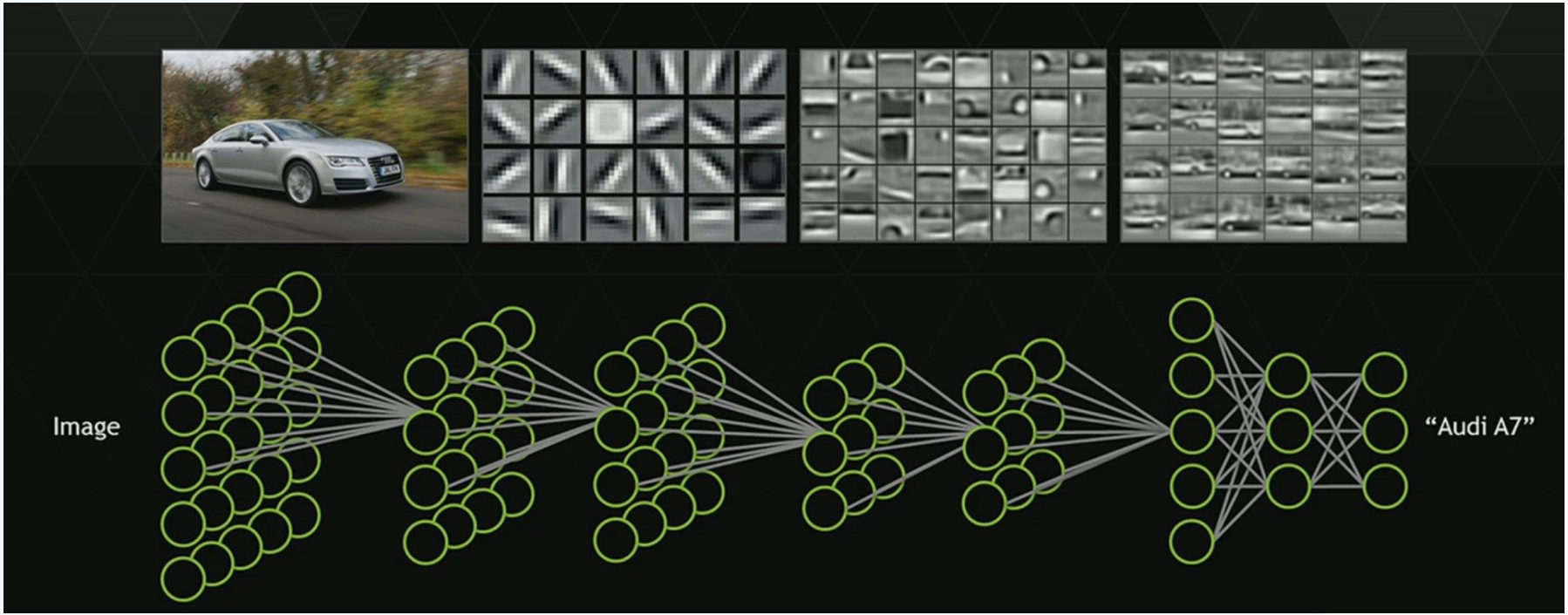


What actually is happening inside a NN?

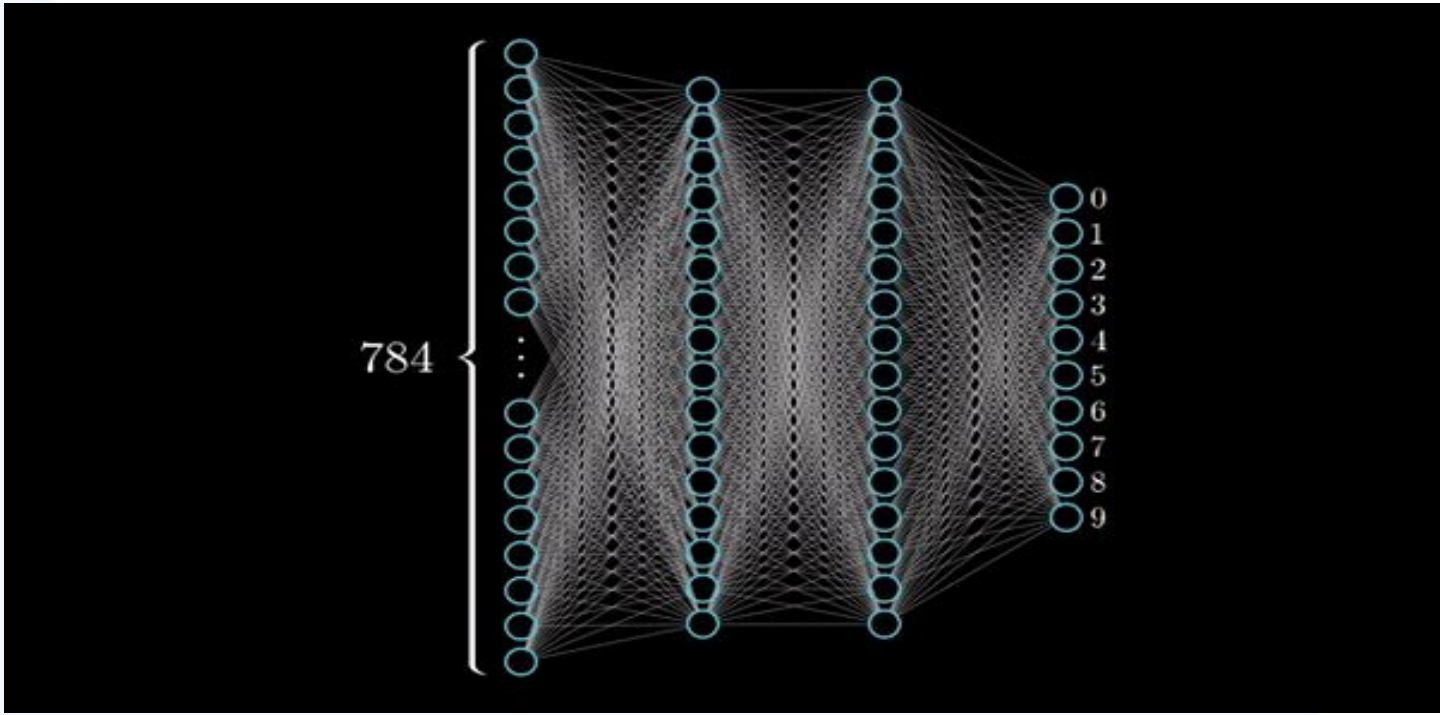
- During training, each neuron learns a specific feature.
- Neurons in shallow layer learns simple feature (lines or curves) whereas in deeper layer learns complex features (facial features like nose).
- Each neuron in a single layer learns features of same complexity (nearly).
- It is necessary to **randomly initialise** weights to a non-zero value, otherwise all neurons in one single layer will always same value (learn same feature).



Visualizing features learned at each layer



Visualizing a Working NN!

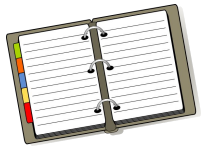


In each layer neurons are activated when feature they learned is detected in the image.

Disadvantages of Deep Learning

- Large dataset are required to train a “good” deep neural network.
- Using “Sigmoid” Act. Fun., training slows down at the end.
  - Solution: Use **ReLU** act. fun.
- A good NN model contain thousands or even millions of parameters.
- Have to write large complex code to get a good model.
  - Solution: Use frameworks like **TensorFlow**, **Keras**, etc.
- Training process requires high computational power and is time taking.
  - Solution: **Transfer Learning**
- Require to tune lots of hyperparameters
  - Learning Rate
  - Number of Layers
  - Number of neuron in each layer
  - λ (in regularization)
  - Number of iterations (epoch)
  - Batch Size

References



- Kaggle DL course: https://www.kaggle.com/learn/intro-to-deep-learning
- DL Explained (Melanie Swan): https://www.slideshare.net/lablogga/deep-learning-explained
- CS771A: Intro to Machine Learning Slides, IIT Kanpur by Prof Purushottam Kar
- Deep Learning Coursera, by Andrew NG: https://www.coursera.org/programs/indian-institute-of-technology-kanpur-on-coursera-4adct?collectionId=&currentTab=MY_COURSES&productId=W62RsyrdEeeFQQqyuQaohA&productType=s12n&showMiniModal=true
- 3Blue1Brown: https://www.youtube.com/watch?v=aircAruvnKk&t=108s
- https://medium.com/@xzz201920/activation-functions-linear-non-linear-in-deep-learning-relu-sigmoid-softmax-swish-leaky-relu-a6333be712ea
- https://medium.com/@jorgesleonel/hyperparameters-in-machine-deep-learning-ca69ad10b981
- https://towardsdatascience.com/why-deep-learning-is-needed-over-traditional-machine-learning-1b6a99177063#:~:text=When%20there%20is%20lack%20of,language%20processing%2C%20and%20speech%20recognition.
- https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
- https://www.cleanpng.com/png-rubbish-bins-waste-paper-baskets-hazardous-waste-w-4062584/
- Google Images

Questions???

Some food for thoughts:

- **Combination of traditional machine learning techniques with DL!**
- **To decrease training time, can we use some pre-computational methods?**
- **What NN architecture to use for text and image data?**