ANN: Artificial Neural Network

Electronic Brain – Adjustable weights  
Perceptron: Learnable Weight and Threshold  
‘XOR’ Problem   
Multi-Layered Perceptron – Backpropagation  
SVM –Support Vector machines, easier to train  
Deep Neural Network –Pretraining

Advances in computation architecture and improved algorithms enable the adoption of DL

Training of deep networks was made computationally feasible by:  
 - Faster CPUs  
- The move to parallel CPU architectures  
- Advent of GPU computing

Improvements in algorithms

Neural networks are often represented as a matrix of weight vectors  
- GPUs are optimized for very fast matrix multiplication  
- 2008: NVIDIA’s CUDA library for GPU computing is released

First major focus of deep learning groups was computer vision  
 - minimized error of classification by 37%

Up until 2012, imagenet competitions were always won by traditional ML Algorithms

Ultimate goal of DL: No need for human curation

Deep learning algorithms attempt to learn good features or representations by using a hierarchy of multiple layers

**How a Neural Network Works:**

A neuron takes weighted sum of inputs a and feeds the result through an activation function O

Output of the activation function produces decision boundary which can be used in classification

A neural network =running several logistic regressions at the same time

Adding layers to artificial neural network allows handling more complex spaces, but it comes at a cost of requiring more parameters

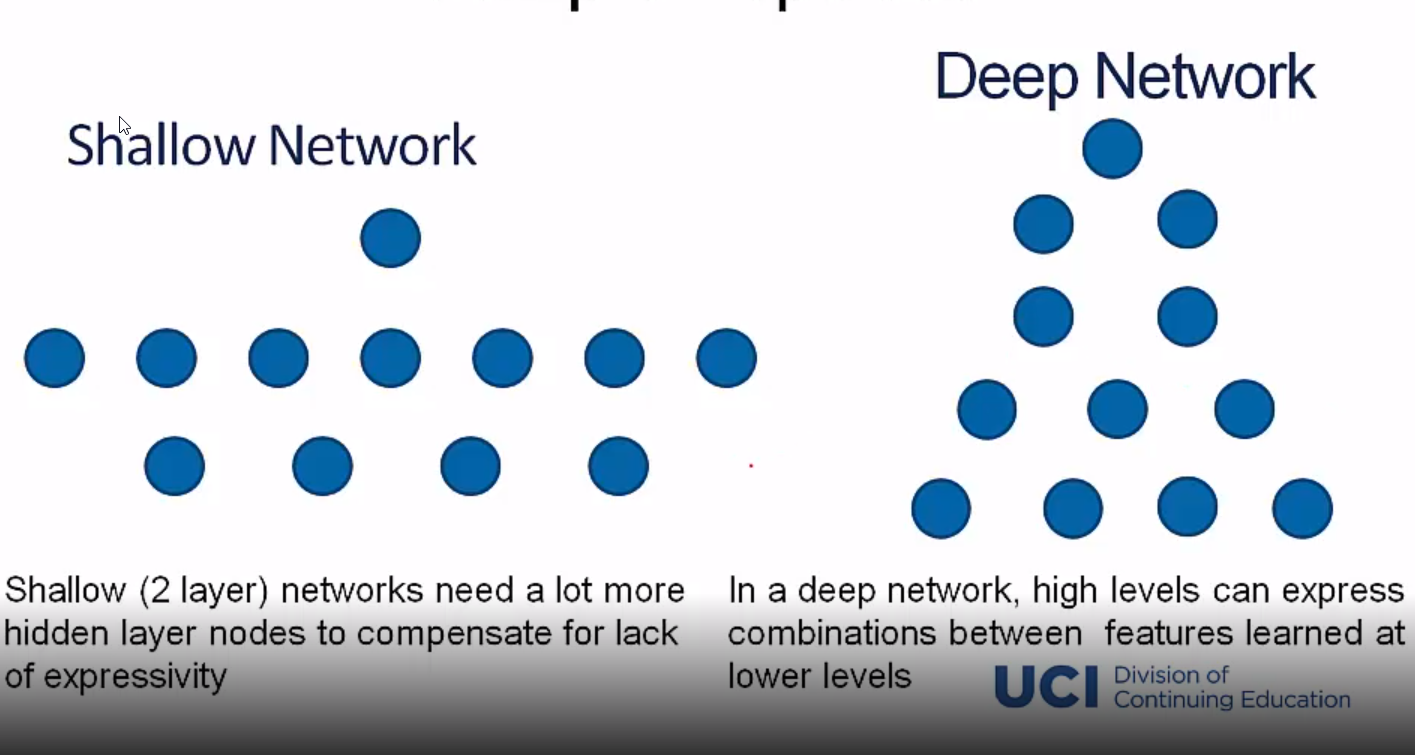
**Many Deep Networks use rectified linear unit (ReLU)**Takes a real-valued number and thresholds it at zero

Neural Networks with more neurons can express more complex functions

However we have to balance between generalization and overfitting

‘Attempts to memorize the values to become a lookup table’

**Shallow networks can also model complex spaces**



**Training DL Network**

Sample labeled data( Batch) -> Forward it through the network, get predictions -> Back Propagate the errors -> Update the network weights

Optimize (min / max) objective/cost function J(theta)

Generate error signal that measures difference between predict6ions and target values

Weights are updated using the partial derivative of the activation function w.r.t the error

**Feed forward/ Backpropagation Neural Network**

Feed forward algorithm: Activate the neurons from the bottom to the top

Backpropagation: Randomly initialize the aprameters  
Calculat e total error at the top  
Then calculate contributions to error, at each step going backwards

**Regularization**

Dropout: Randomly drop units during training  
- Each unit retained with fixed probability p, independent of other units  
- Hyper parameter p to be chosen (tuned)

L2 = Weight Delay  
- Regularization term that penalizes big weights, added to the objective  
- Weight decay value determines how dominant regularization is during gradient computation  
- Big weight decay coefficient -> big penalty for big weights

Early-Stopping  
- Use Validation error to decide when to stop training  
- stop when monitored quantity has not improved after ‘n’ subsequent epochs  
- n is called patience

**Types of Deep Neural Networks**

**Supervised:** Learning with a labeled training set  
 Example: Sentiment text classification based on manually annotated sample

**Unsupervised:**  Discover patterns in unlabeled data  
 Example: Cluster similar news articles based on content

**Reinforcement Learning:** Learning to act based on feedback / rewards  
 Example: Learn to exit labyrinth with fewest moves possible

**The Neural Network Zoo –** asimovinstitute.org/neural-network-zoo

**Types of deep nets and their applications**

**Unsupervised Learning:**  
 - Restricted Boltzmann Machine  
 - Autoencoder