Machine Learning Techniques for Radar Automatic Target Recognition (ATR)

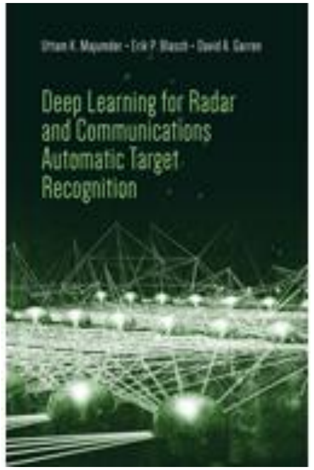
Chapter 3: Machine Learning Algorithms Review

Uttam K. Majumder

Machine Learning Techniques for Radar Automatic Target Recognition (ATR)

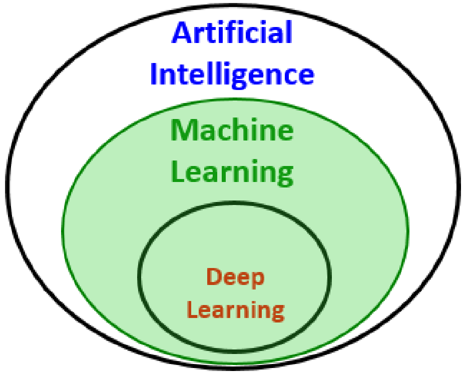
Lecture Outline

1. **Radio Frequency ATR: Past, Present, and Future:**
   20 min
2. **Mathematics for Machine Learning / Deep Learning:**
   20 min
3. **Review of ML Algorithms: 25 min**
4. **Deep Learning Algorithms: 30 min**
5. **RF Data for ML Research: 15 min**
6. **DL for Single Target Classification: 25 min**
7. **DL for Many Targets Classification: 25 min**
8. **RF Signals Classification: 20 min**
9. **RF ATR Performance Evaluation: 25 min**
10. **Emerging ML Algorithms for RF ATR: 35 min**

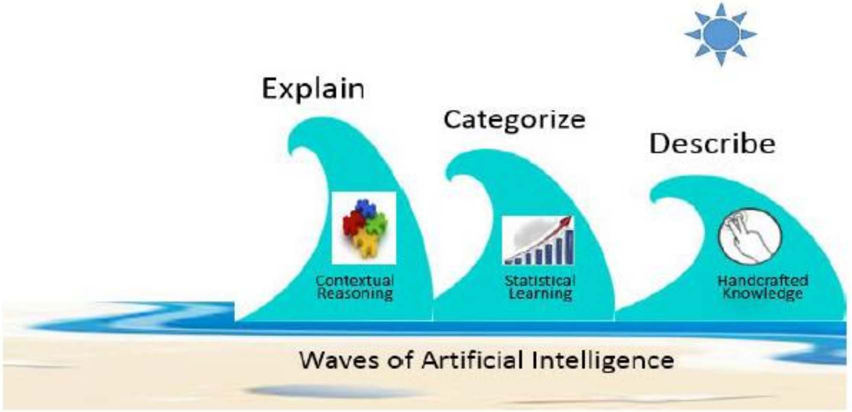


1.4 AI / ML / DL

Prof. Geoffrey Hinton
Prof. Yann LeCun
Prof. Yoshua Bengio
~ Godfathers of AI / Deep Learning ~



- **AI – Machines** to think/behave/react - ANN
- **ML – Data** for (Machines) to learn - RL, BN, ILP
- **DL – Brain-Inspired** NN for robust methods – CNN, RNN
  - (mostly supervised from labeled data)



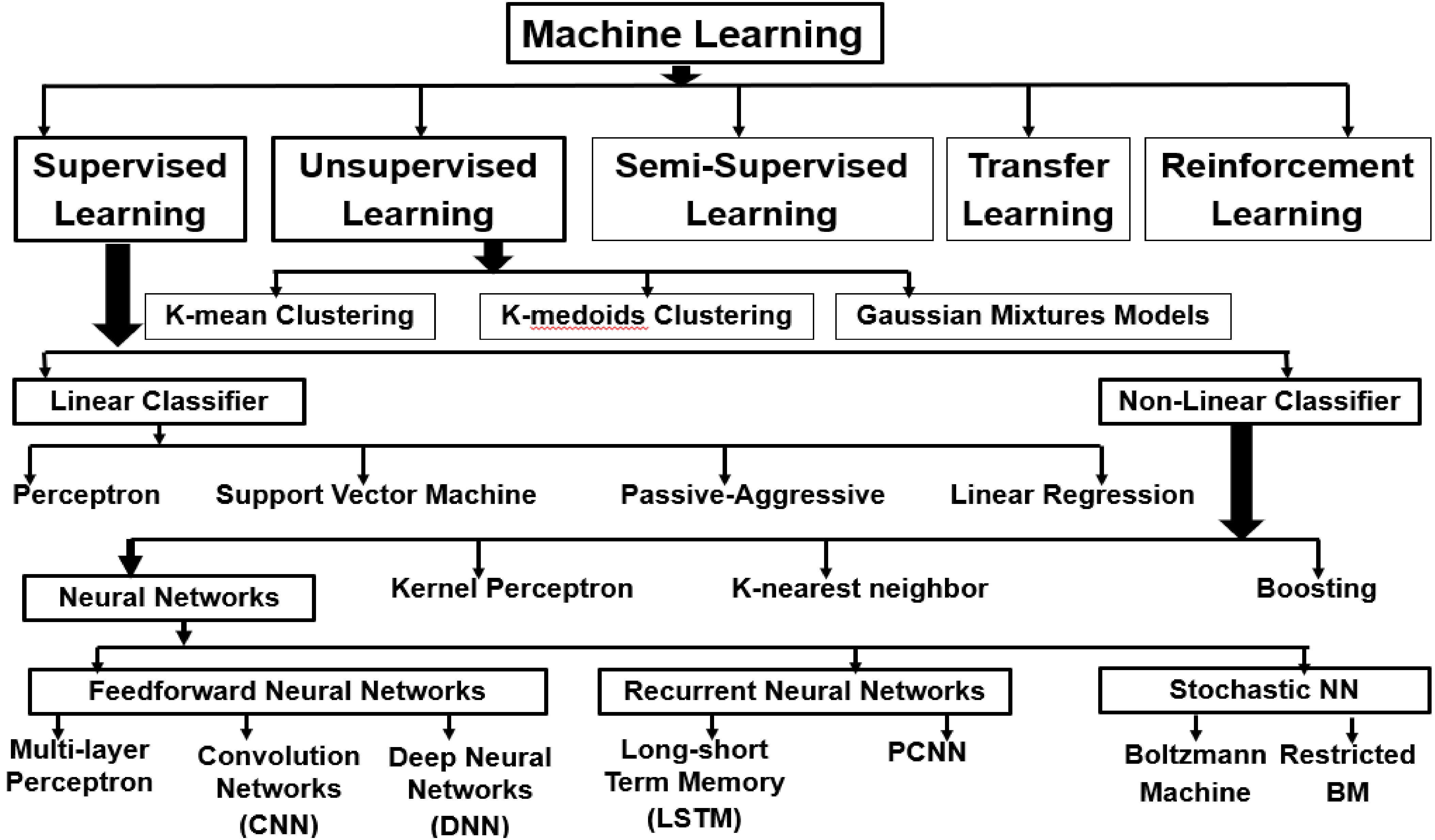
AI: Artificial Intelligence
ANN: Artificial Neural Networks
RL: Reinforcement Learning
BN: Bayesian Networks
ILP: Inductive Logic Programming
CNN: Convolutional Neural Networks
RNN: Recurrent Neural Networks

Reference: Andrew Fogg, A histroy of Deep Learning, (import.io)

ML Algorithms: Types

- **ML Algorithms in Broad Categories**
  - Supervised Learning
  - Unsupervised Learning
  - Semi-supervised Learning
  - Transfer Learning
  - Reinforcement Learning
  - Generative Adversarial Network/Learning

ML Algorithms Types in Detail



Supervised Learning (SL)

- **Many Practical ML uses Supervised Learning**
- **Input/Training Data are given with a "Level" or "Tag"**
- **Supervised Learning seeks to develop a very robust mapping function** $Y = f(X)$

  where X is the input data and Y is the output data
- **The goal is that when we have new input data X (e.g. images), can predict Y (e.g., class labels) for that data**

f

- SL Generally requires large amounts of training data
- Two main types of supervised learning:
  - Classification – the output value is a class label
  - Regression – the output value is a real value

Supervised Learning

- **In general, Supervised Learning is divided into two areas:**

1. **Linear Classifier**
   - Perceptron
   - Support Vector Machine
   - Passive Aggressive
   - Linear Regression
2. **Non-Linear Classifier**
   - Neural Networks
   - Kernel Perceptron
   - Boosting
   - K-nearest Neighbor

Supervised Learning

➢ Linear Classifier: Perceptron

- Binary Classifier, Invented 1957, at Cornell Aeronautical lab by Frank Rosenblatt

- Also, known as single-layer perceptron (there is multi-layer perceptron). Perceptron is the simplest form of feedforward neural network
- The perceptron algorithm attempts to divide the input space into two halves with a single line
- Perceptron does not consider the entire dataset at the same time, rather, it looks at one example at a time, processes it, then moves onto the next example ("online")

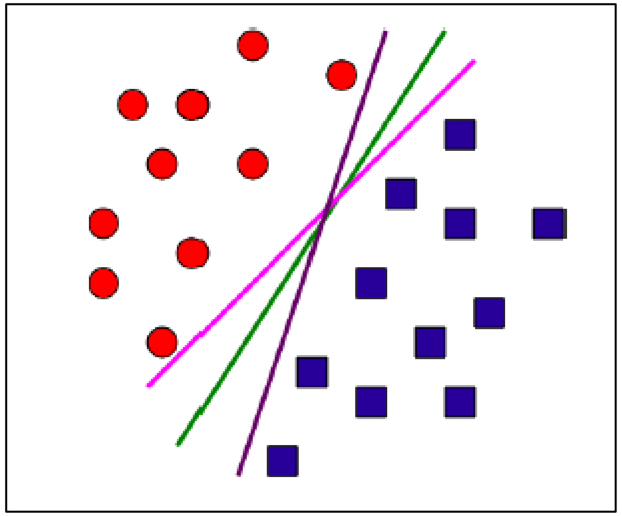


Fig. 1: Perceptron Algorithm [1]

Supervised Learning

➢ Linear Classifier: Support Vector Machine (SVM)

- Invented 1963, by Vladimir Vapnik and Alexey Chervonenkis

- Can be used for Image Classification and Character Recognition
- Global Optimization for the loss function
- **On-line version: *<u>Pegasos</u>*** algorithm
- Support Vector Machine works similarly to Perceptron in that it attempts to divide the input space with a linear separator.
- The main difference is that the SVM works to maximize the margin between the two classes in the input space and therefore may still update even when it guesses correctly.

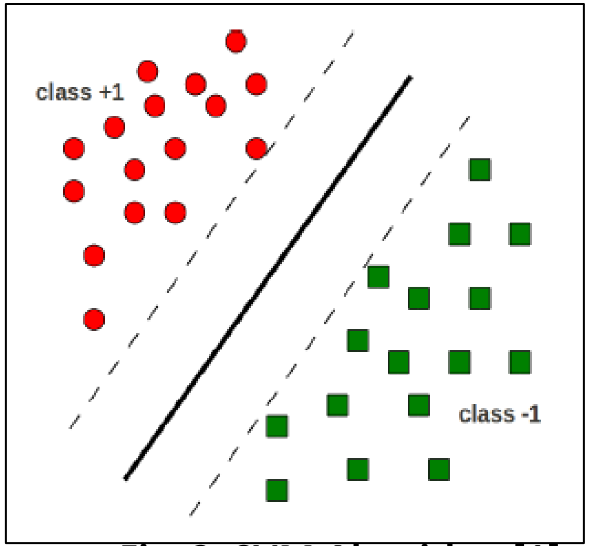


Fig. 2: SVM Algorithm [1]

Supervised Learning

- **Linear Classifier: Passive-Aggressive**
    - Invented 1963, by Vladimir Vapnik and Alexey Chervonenkis
  - This is an extension of the SVM algorithm
  - Uses point-by-point optimization for the loss function

Supervised Learning

➢Linear Classifier: Linear Regression

- Ordinary Least Square is most popular algorithm
- Works to minimize the sum of the squared residuals.

Given a regression line through the data, calculate the distance from each data point to the regression line, square that distance, and sum the distances for each point (i.e. the sum of the squared residuals).

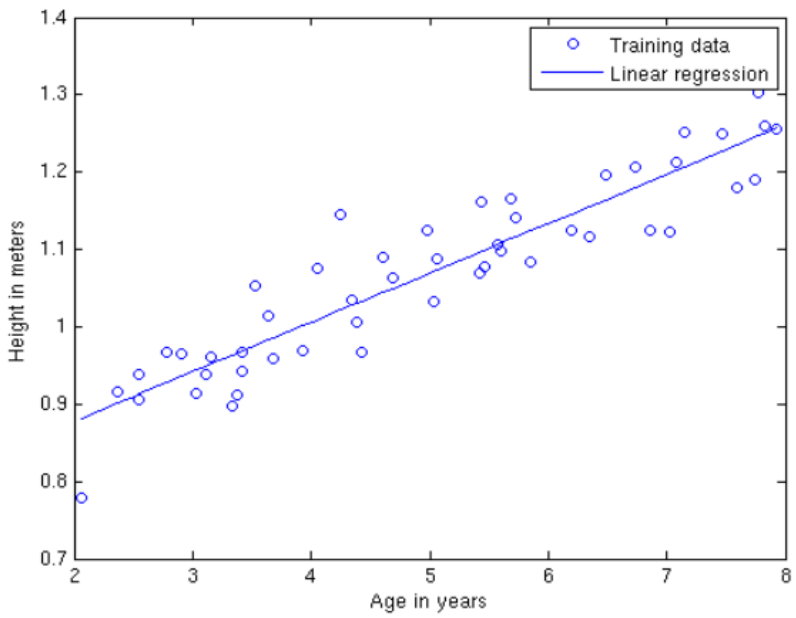


Fig. 3: Linear Regression [17]

- Both inputs and outputs are numeric

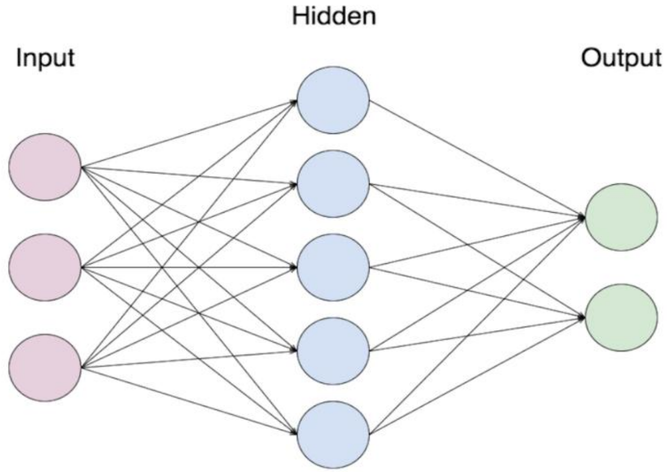
Supervised Learning

- **Non-Linear Classifier: Neural Networks**
    - Invented 1943, by Warren McCulloch and Walter Pitts
  - **Feedforward Neural Networks (FNN)**
    - Multi-Layer Perceptron
    - Deep Neural Networks
    - Convolution Neural Networks
  - **Recurrent Neural Network (RNN)**
    - LSTM
    - Boltzman Machine
    - Reservoir Computing
    - Liquid State Machine

Supervised Learning

➢Non-Linear Classifier: Neural Networks

- Neural Networks are designed to recognize numerical patterns in the input data, and ultimately learn the mapping function between the input and output data.
- A NN is a corrective feedback loop, rewarding weights that support correct guesses and punishing weights that lead to error
- Each hidden layer attempts to learn a distinctive set of features based on the previous layer's output. In general, the deeper the network, the more complex features can be learned (feature hierarchy).



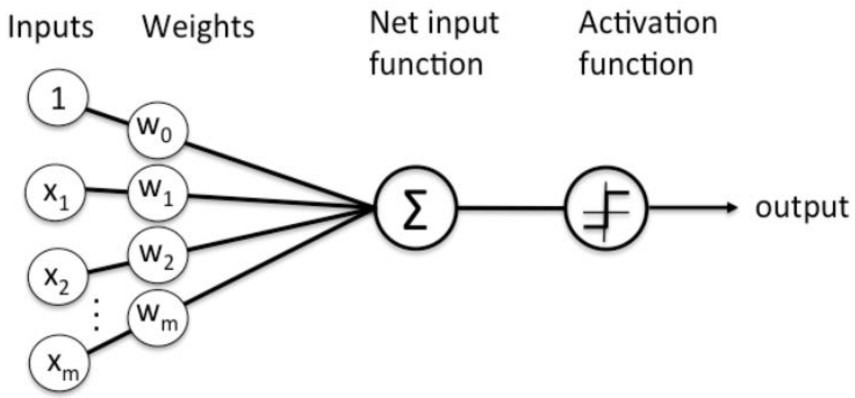


Fig. 4: NN Concept

Supervised Learning

- **Non-Linear Classifier:** Multi-layer Perceptron (MLP)
  - MLP is the simplest form of feedforward NN based upon Linear Perceptron.
  - Generally, MLP consists of three or more layers of non-linearly activating nodes
  - The network learns from backpropagation process

Supervised Learning

➢Non-Linear Classifier: Feed Forward Neural Networks

- **Deep Neural Networks (DNN)**
  - A network is considered “deep” if it has several hidden layers
- **Important DNNs**
  - ResNet
  - Wide ResNet
  - VGG (Visual Geometry Group)
  - AlexNet
  - GoogleNet
  - Generative Adversarial Networks

Many of these above DNNs use convolution filters for feature extraction; hence these could be referred to CNN as well

Supervised Learning

➢Non-Linear Classifier: Convolutional NN (CNN)

- Use a combination of filters that each search the input space for very specific features
- Each filter creates a feature map that gets fed into the next layer to be filtered again
- CNN’s create very highly dimensional representations of inputs depending on how many filters are used and as a result must be resized with pooling/down-sampling layers.

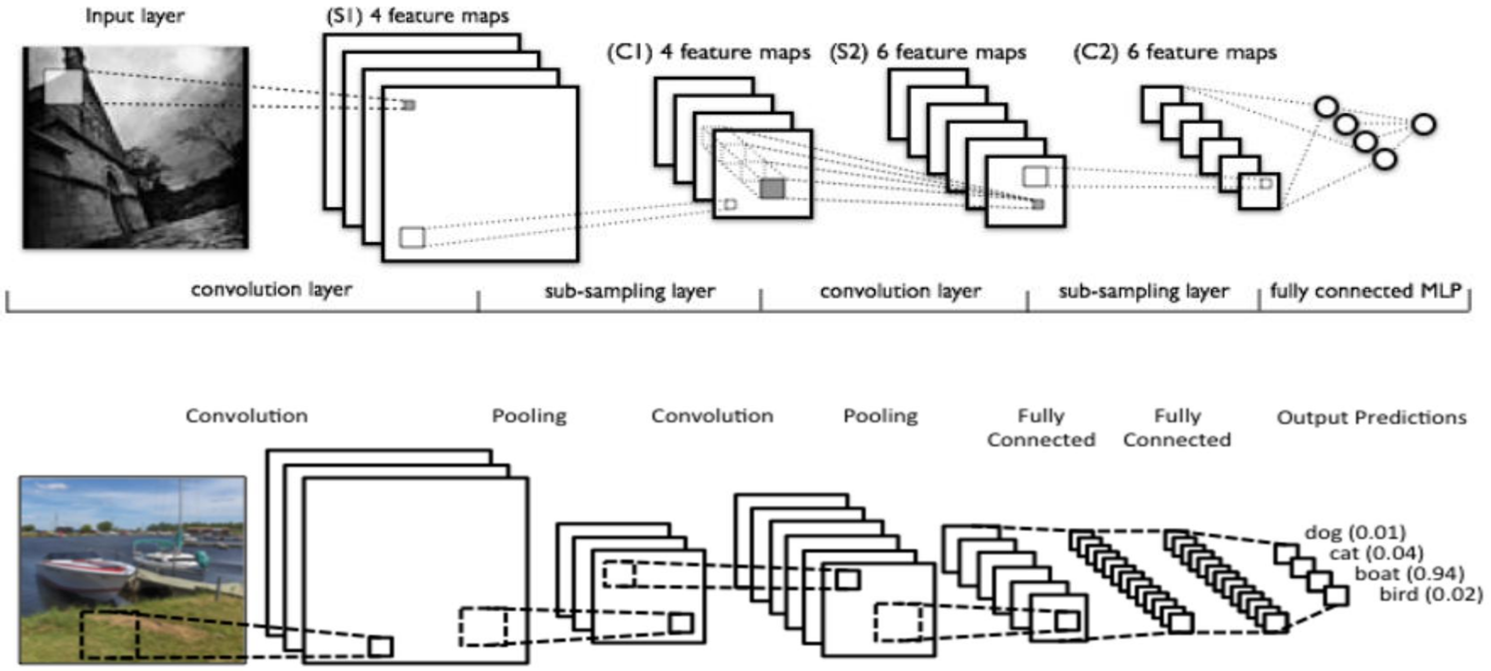


Fig. 5: CNN concept

Supervised Learning

- **Recurrent Neural Networks (RNN):** RNN is used for learning sequences. Applications such as speech recognition, handwriting recognition has been implemented using RNN.

- RNNs have feedback loops in learning process

- **Some Important RNNs:**
  - Long Short-term Memory (LSTM)
  - Boltzman Machine
  - Reservoir Computing
  - Liquid State Machine

Supervised Learning

➢Non-Linear Classifier: Kernel Perceptron

- Works similarly to perceptron in that it attempts to divide the input space
- The key idea is that the algorithm uses a Kernel function to transform the input space. The transformed space does not have to have the same dimensionality as the input space, and in many cases it has a higher dimensionality. The Kernel Perceptron then attempts to construct a n-1 dimensional hyperplane in the n-dimensional transformed space which correlates to a non-linear separation in the original input space.
- The effective algorithm and update rules are the same as the classic perceptron algorithm except for the transform itself.

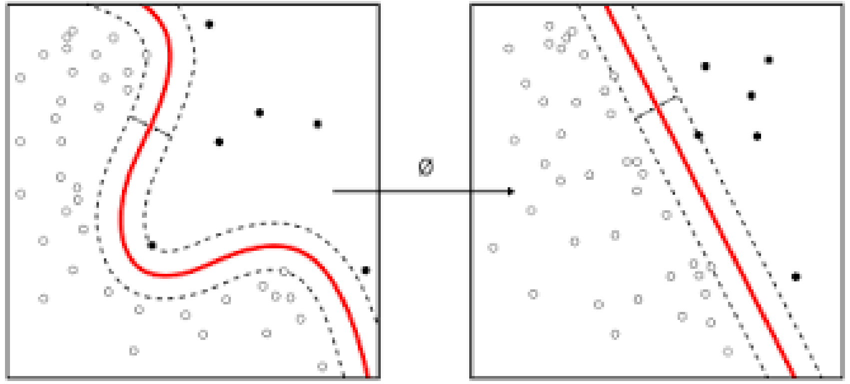


Fig. 6: Kernel Perceptron Concept [24]

Supervised Learning

➢Non-Linear Classifier: K-nearest Neighbor (KNN)

- Works on entire data set at once; KNN does NOT learn any model, rather the model is the training set itself
- For each new instance, the algorithm searches through the entire training set, calculating the difference between the new instance and each training model.
- For classification, the output is the class with the K-most similar neighbors.
- For regression, the output value is based on the mean or median of the K-most similar instances.

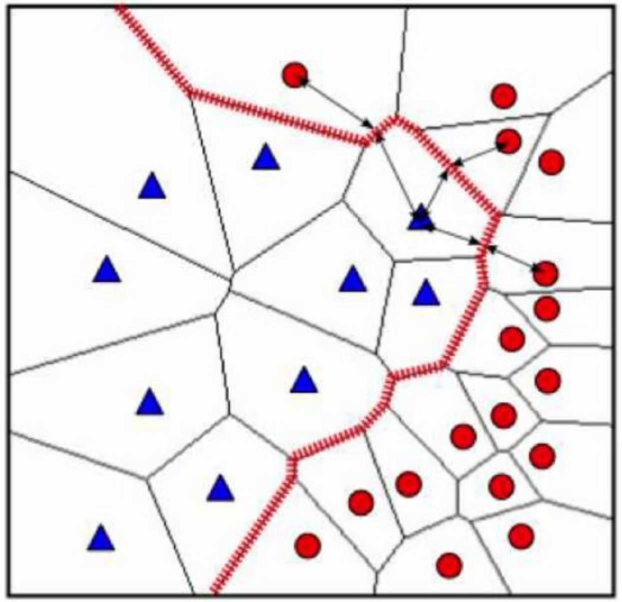


Fig. 7: K-nearest neighbor

Supervised Learning

➢Non-Linear Classifier: Boosting

- Use several different classifiers that are each good at identifying based on certain features
- Have each classifier vote on what the final result is
- H(x) = (w1*h1(x) + w2*h2(x) + … + wn*hn(x))
    - Where H(x) is the overall classifier, the h functions are the individual classifiers, and the w's are the weights of each individual classifier
- "Wisdom of a weighted crowd of experts" – Prof. Patrick Winston, MIT
- The weights for each of the classifiers are updated based on the error they contribute
- The update rule for the weights is remarkably simple as it is a scaling

Unsupervised Learning

- **In Unsupervised Learning, the algorithms discover “hidden pattern” from the input data.**
  - During training, we only have the input data and no corresponding output variables (labels), as opposed to supervised learning where we have both
  - The goal of this type of learning is to model the underlying structure or distribution of the data
  - “Algorithms are left to their own devices to discover interesting structure in the data”
  - Clustering – discover inherent groupings in the data
  - Association – discover rules that describe large portions of the data
- **Unsupervised Learning Algorithms:**
  - Clustering
  - K-medoids
  - Gaussian Mixtures Models
  - Autoencoder

Unsupervised Learning

➢Clustering

- Idea: A data set with N objects can be grouped into any number of clusters between 1 and N. The goal of the algorithm is to identify regions in which the data points are concentrated, and group the points accordingly.
- No guarantee that a globally optimal solution will be reached, as it depends on the initial seeding of the cluster centers (centroids)
  - K-means
    - Aims to partition N objects into K clusters, in which each observation belongs to the cluster with the nearest mean.
    - Goal is to minimize the average squared Euclidean distance of objects from their centroids
    - The measure of how well the centroids represent the members of their clusters is the residual sum of squares (RSS), which is the sum of the squared distances from each observation to its centroid.
  - K-medoids
    - This algorithm is very similar to K-means.
    - Rather than calculating centroids as the cluster centers, it uses medoids which are the most centrally located objects in the cluster, not just points in space.
    - Less sensitive to outliers

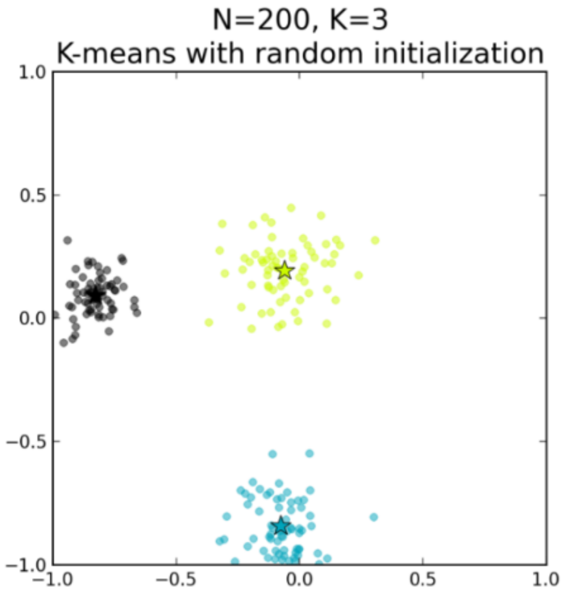


Fig. 8: Clustering

Unsupervised Learning

➢Gaussian Mixture Models (GMM)

- GMM is used for data clustering
- GMM Parameters are estimated from training data by using expectation-maximizatiom (EM) or Maximum-A-Posteriori (MAP)

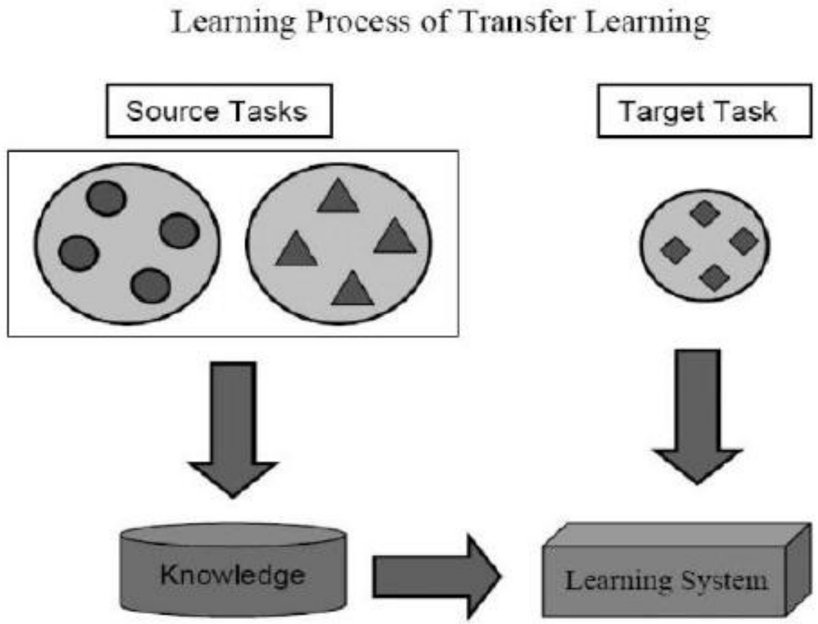
  algorithms

Semi-Supervised Learning

- In Semi-supervised Learning, some input data are leveled. This information combined with unsupervised learning such as “clustering” can be used for classifying data
  - Some of the input data are labeled and some are not
  - Use a mixture of supervised and unsupervised learning techniques
  - Note: Most of the data in the world is unlabeled, there is only a small fraction that is. Unsupervised and semi-supervised learning techniques can be applied to much larger, unlabeled datasets, making them very appealing to some researchers

Transfer Learning

- DNN Requires a large training dataset to produce accurate learning representation from test data
  - When a new dataset are comparatively small, we may use already learned data that are similar to a new dataset
  - Transfer Learning offers a way to leverage existing dataset to perform well on new dataset
  - The idea is to use information that you already know and that you have already worked hard to learn and apply that knowledge to a new, similar dataset



Transfer Learning

- **One-Shot learning:**
  - One of most powerful ideas in Transfer Learning
  - Rather than having thousands of training models per class, just have one, and take advantage of knowledge from previously learned categories and the information from the training model to learn the new model
  - Make use of the knowledge that has been gained so far rather than starting from scratch each time we want to learn a new category/class
- **Low-shot Learning:**

Reinforcement Learning

- Learning by interacting with the environment
- The learning agent learns from the consequences of its actions, rather than from being explicitly taught, and it selects it actions based on its past experiences (exploitation) and also by new choices (exploration).
- The agent receives a numerical reward for each of its actions which encodes a level of success. The agent then learns to perform actions that maximize the reward.

Generative Adversarial Networks/Learning

- Known as GAN/AML:
  - Youtube Video by Ian Goodfellow: (https://www.youtube.com/watch?v=HN9NRhm9waY)
  - Malicious inputs designed to fool ML models
  - Injecting Adversarial examples into training set increases robustness of neural networks to adversarial examples
  - Discriminator and Generator
  - DC-GAN paper by Alec Radford, et.al, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks" CVPR 2015
  - More:
    - Adversarial Machine Learning, Invited Talk by Ling Huang, et.al. at AISec 2010
    - Adversarial Machine Learning at Scale, by A. Kurakin, Ian Goodfellow, Samy Bengio at ICLR 2017

Summary of ML Algorithms Overview

- **We Provided an Overview of Various ML Algorithms**
    - Supervised Learning
    - Unsupervised Learning
    - Semi-supervised Learning
    - Transfer Learning
    - Reinforcement Learning
    - Generative Adversarial Network/Learning
- **This overview will serve as a foundation to understand "Big Picture" of ML**
    - ML applied to video imagery (Facebook, Google etc.)
    - ML applied to RF imagery
    - ML applied to many expert systems

References

1. Rai, Piyush. *Hyperplane based Classification: Perceptron and (Intro to) Support Vector Machines.* University of Utah. CS5350 Lecture Notes. 08 Sept 2011.
2. Crammer, Dekel, Keshet, Shalev-Shwartz, Singer. *Online Passive-Aggressive Algorithms.* The Hebrew University School of Computer Science and Engineering. 06 June 2006.
3. Brownlee, Jason. *Linear Regression for Machine Learning*. Machine Learning Algorithms. Web. 25 March 2016.
4. *Introduction to Deep Neural Networks*. DeepLearning4J. Web. 2017. https://deeplearning4j.org/neuralnet-overview.html#forward
5. *Convolutional Networks*. DeepLearning4J. Web. 2017. https://deeplearning4j.org/convolutionalnets.html
6. *A Beginner's Tutorial for Restricted Boltzmann Machines*. DeepLearning4J. Web. 2017. https://deeplearning4j.org/restrictedboltzmannmachine.html
7. *A Beginner's Guide to Recurrent Networks and LSTMs*. DeepLearning4J. Web. 2017. https://deeplearning4j.org/lstm.html
8. Johnson, Mark. *A brief introduction to kernel classifiers.* Brown University. CS195 Lecture Notes. Oct 2009.
9. Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.
10. The Data Science Lab. 2014. *Selection of K in K-means Clustering, Reloaded*. https://datasciencelab.wordpress.com/tag/k-means/
11. Winston, Patrick. *Lecture 17: Learning: Boosting.* Massachusetts Institute of Technology. Course Number 6.034. Fall 2010
12. Leif E. Peterson (2009) K-nearest neighbor. Scholarpedia, 4(2):1883.
13. Ghahramani, Zoubin. *Unsupervised Learning*. University College London, UK. 16 Sept 2004.
14. Brownlee, Jason. *Supervised and Unsupervised Machine Learning Algorithms.* Machine Learning Algorithms. Web. 16 March 2016.
15. Torrey, Lisa and Shavlik, Jude. Transfer Learning. University of Wisconsin.
16. Koch, Zemel, Salakhutdinov. *Siamese Neural Networks for One-Shot Image Recognition*. University of Toronto. Toronto, Ontario, Canada.
17. http://openclassroom.stanford.edu/MainFolder/DocumentPage.php?course=MachineLearning&doc=exercises/ex2/ex2.html

References

18. Chen, S., Wang, H., Xu, F., & Ya-Qiu, J. (2016). Target classification using the deep convolutional networks for SAR images. *IEEE Transactions on Geoscience and Remote Sensing, 54*(8), 4806-4817.

19. Karpathy, Andrej. "CS231n Convolutional Neural Networks for Visual Recognition." *CS231n Convolutional Neural Networks for Visual Recognition*. Stanford University, n.d. Web. 04 Jun. 2016.

20. "Caffe.," Berkeley Vision and Learning Center. Web. 04 Jun. 2016. <http://caffe.berkeleyvision.org/>.

21. Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi. "You Only Look Once: Unified, Real-Time Object Detection." (2016): n. pag. Cornell University Library. 9 May 2016. Web. 22 July 2016.

22. Song, H.; Ji, K.; Zhang, Y.; Xing, X.; Zou, H. Sparse Representation-Based SAR Image Target Classification on the 10-Class MSTAR Data Set. Appl. Sci. 2016, 6(1), 26

23. Anderson, Peter. "A Practical Introduction to Deep Learning with Caffe."*Australian Centre for Robotic Vision*. The Australian National University, Dec. 2015. Web.

24. http://www.wikiwand.com/en/Perceptron

25. Christopher M. Bishop, “Neural Networks for Pattern Recognition”, Oxford University Press, 2003

Thank you!