Primer on Analysis of Experimental Data and Design of Experiments

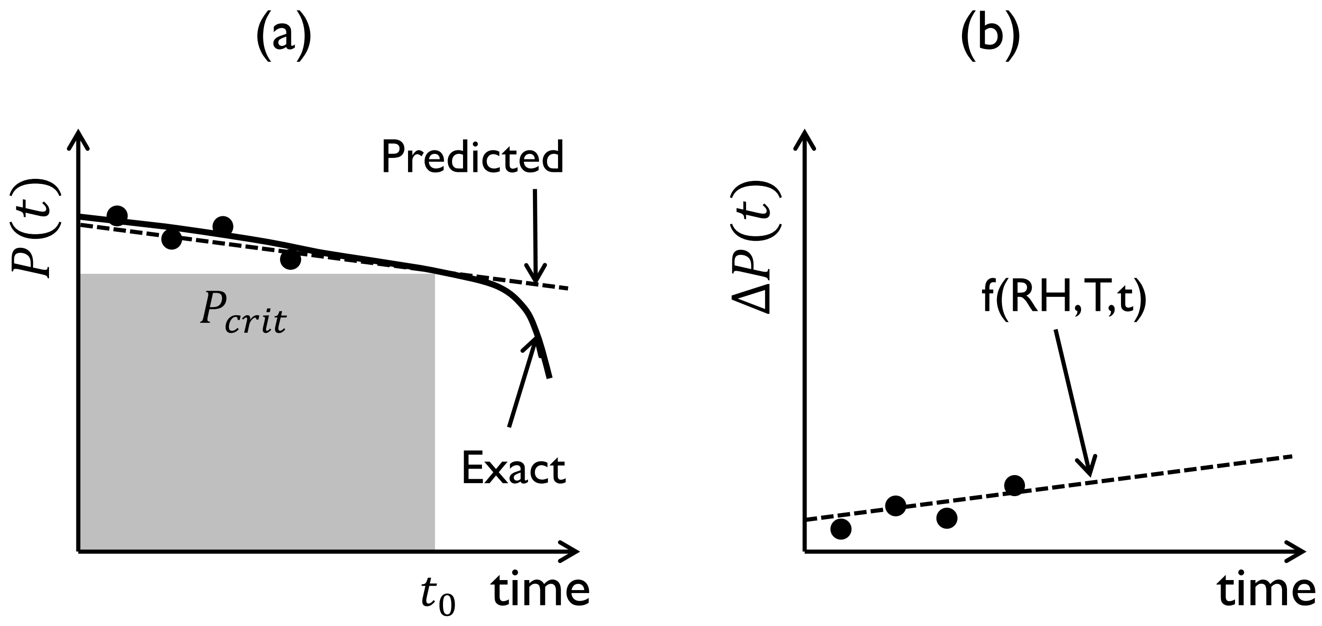
Lecture 13. Deep Learning, Karnaugh Mapping, and Unsupervised Classification

Muhammad A. Alam
alam@purdue.edu

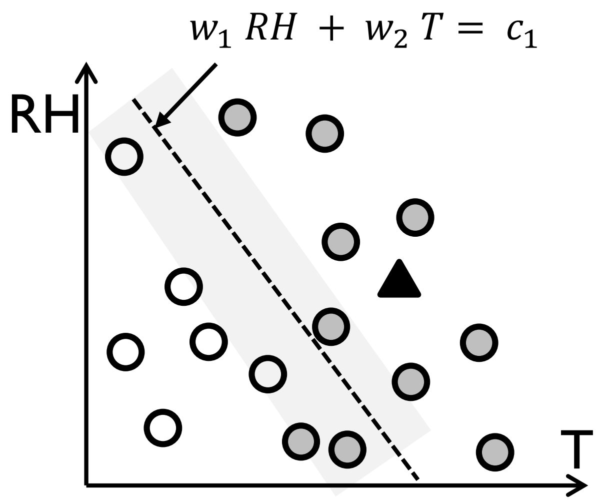
Outline

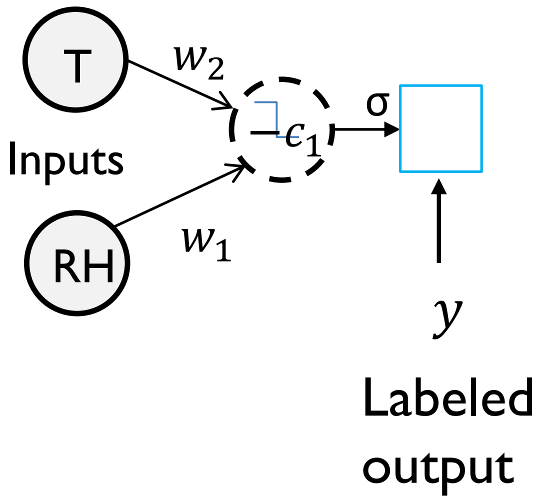
1. Introduction
2. A two input, single and multiple perceptron problem
3. Backpropagation and coefficient fitting
4. Machine learning and Karnaugh mapping
5. Other forms of Machine Learning (Unsupervised, optical, quantum)
6. Conclusions

Reliability of Solar Farms …



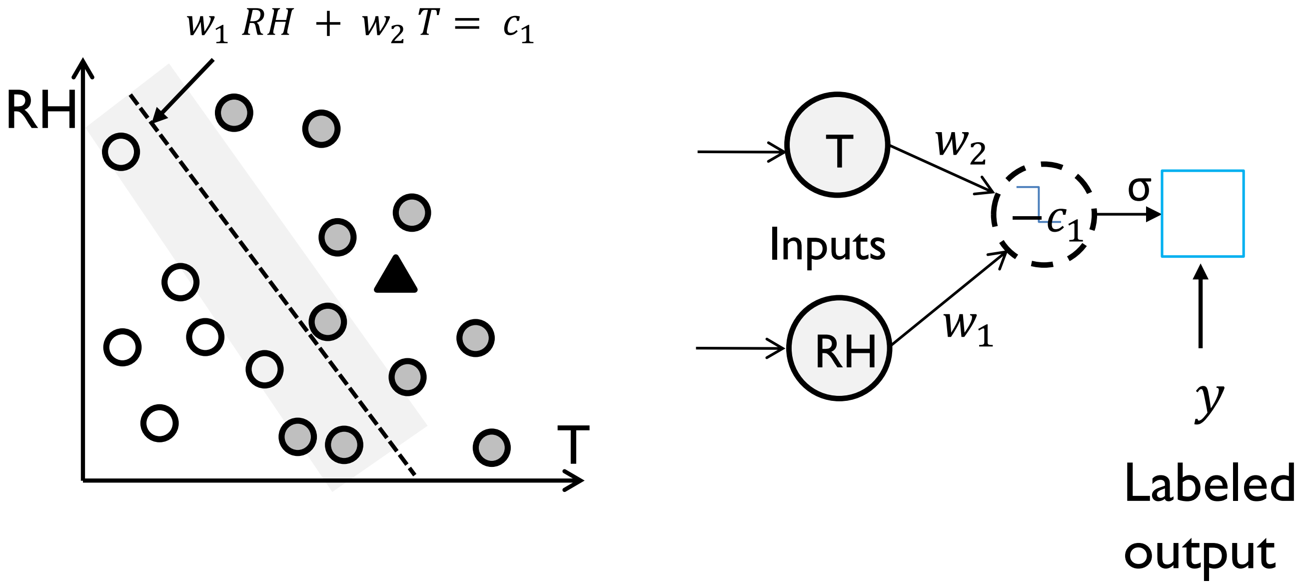
…. represented by two input ANN





$$\sigma(w_1, w_2, c) = \frac{1}{1 + \exp(-(w_1 T + w_2\,RH - c_1)/\sigma)}$$

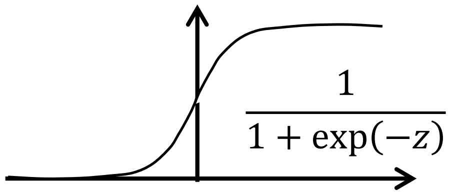
Training by backpropagation



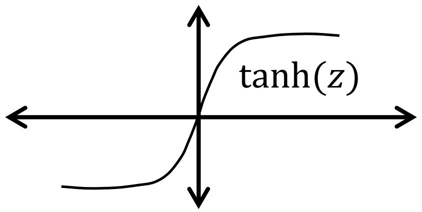
Algorithms by computer scientists
We only have straight lines, hence many layers

Aside: Transition Functions

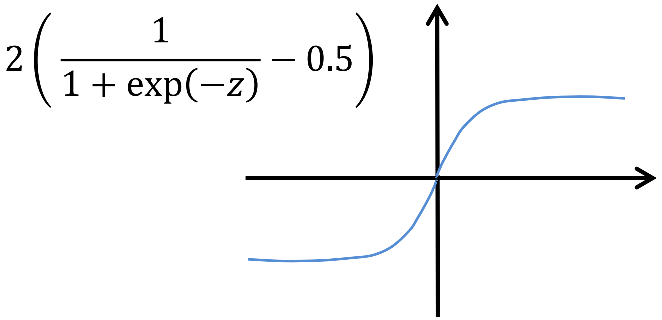
sigmoid

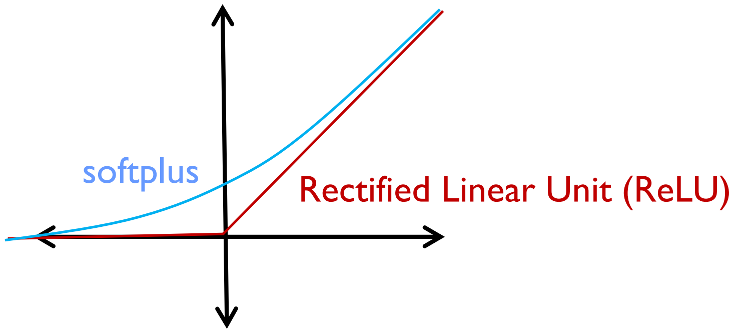


tanh



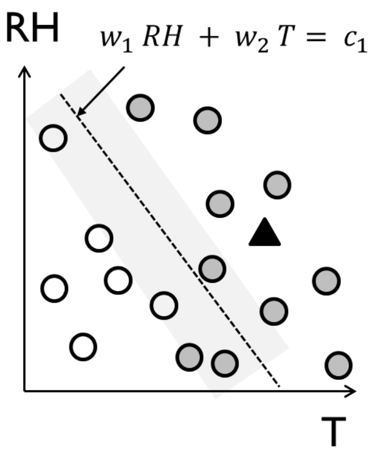
Shifted sigmoid



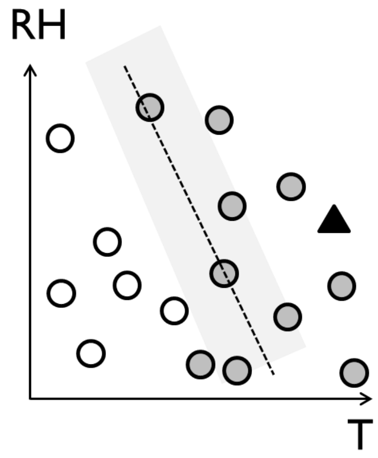


Sigmoid/tanh emphasizes points close to transition

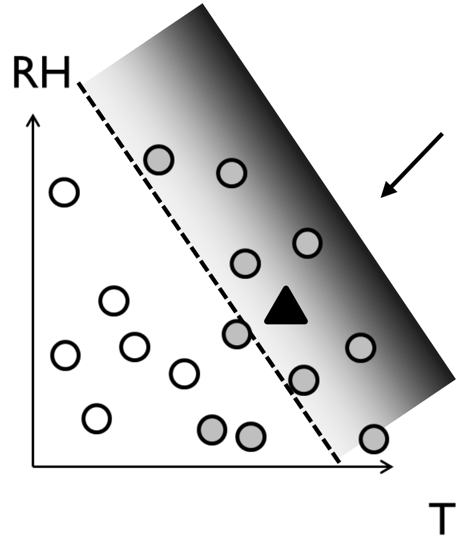
Aside: Different transition functions



Sigmoidal

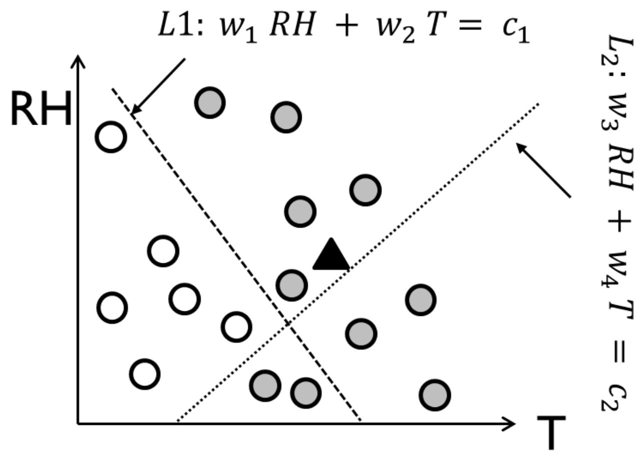


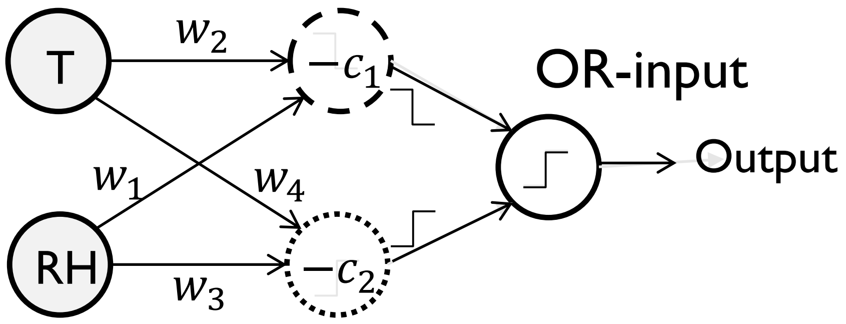
Support Vector Machine



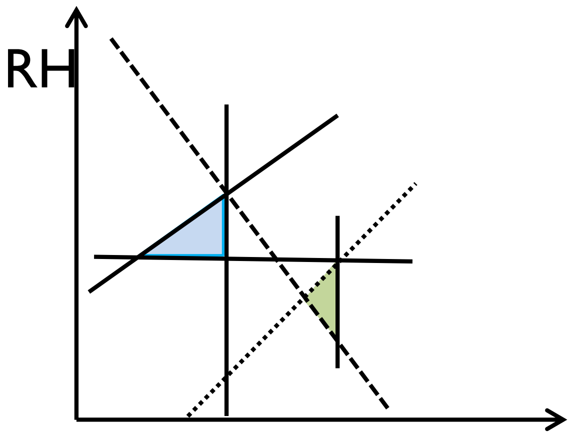
LiRu

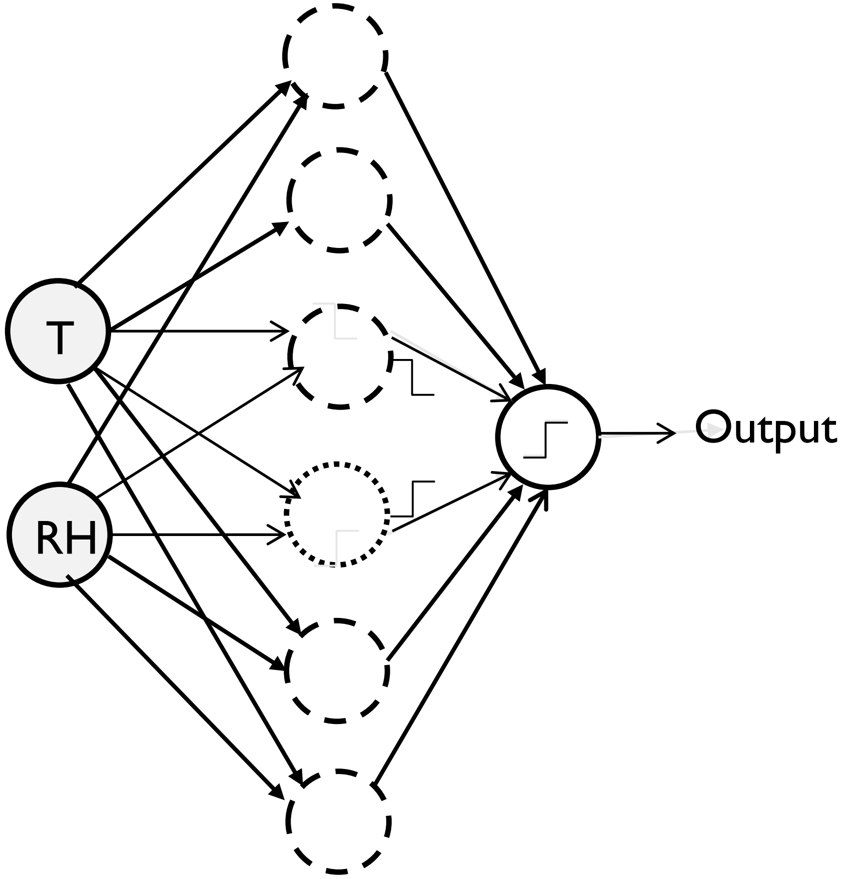
Region defined by two lines



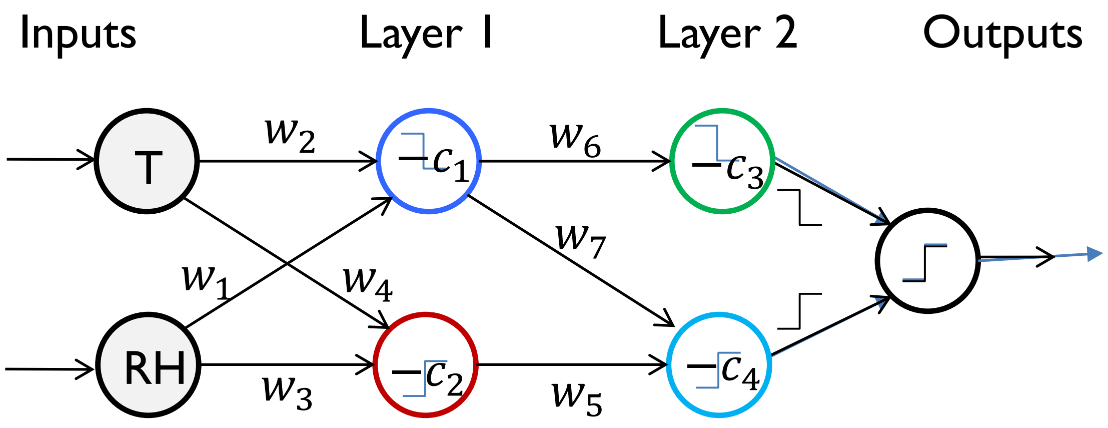


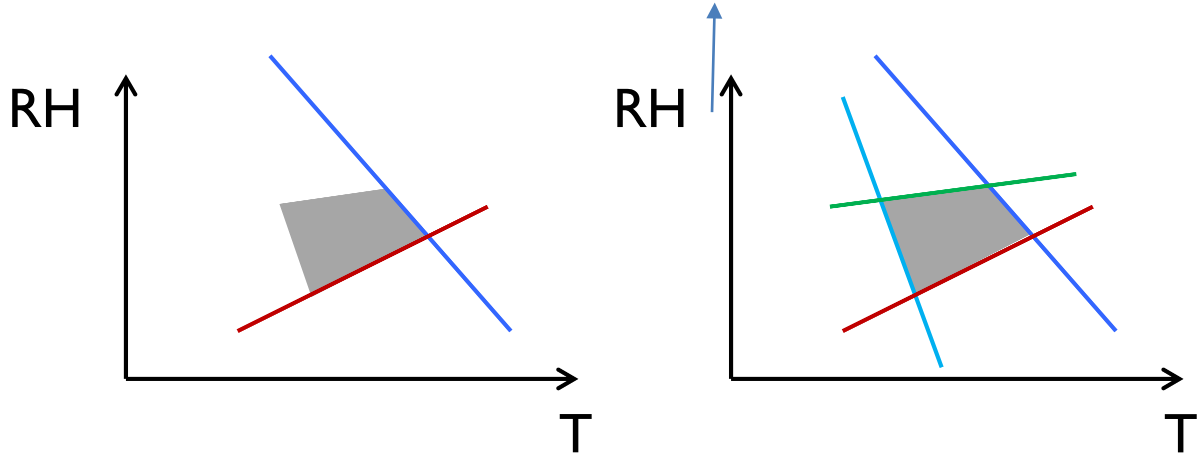
Region defined by multiple lines





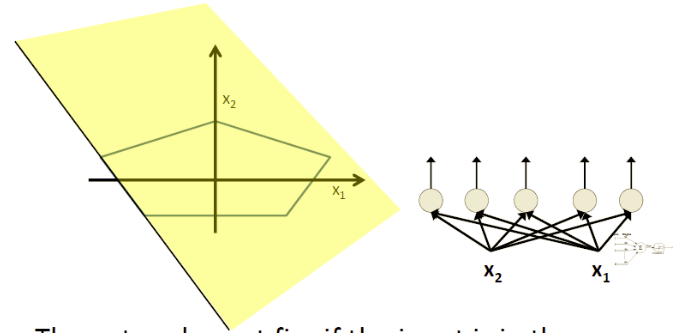
Deep network





2D Image Recognition by Deep Network

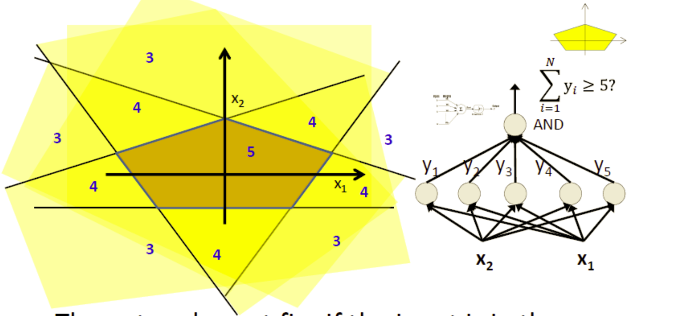
Booleans over the reals



- The network must fire if the input is in the coloured area

74

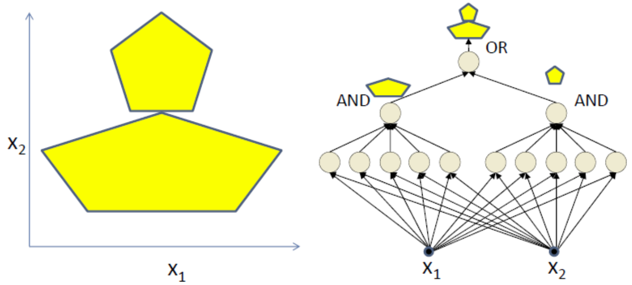
Booleans over the reals



- The network must fire if the input is in the coloured area

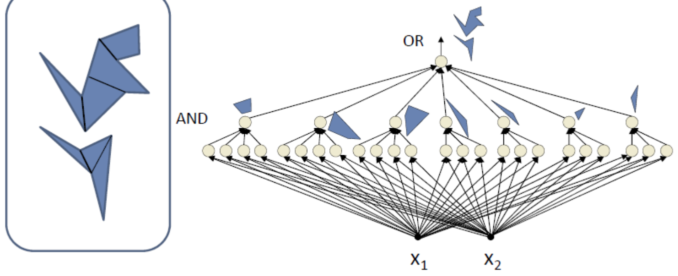
75

More complex decision boundaries



- Network to fire if the input is in the yellow area
  - "OR" two polygons

Complex decision boundaries



- Can compose *arbitrarily* complex decision boundaries

78

Outline

1. Introduction
2. A two input, single and multiple perceptron problem
3. Backpropagation and coefficient fitting
4. Machine learning and Karnaugh mapping
5. Other forms of Machine Learning (Unsupervised, optical, quantum)
6. Conclusions

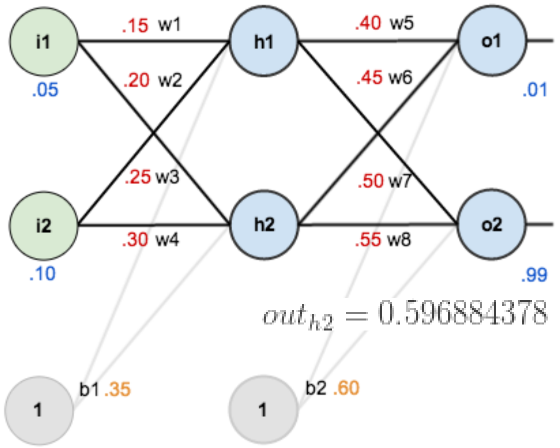
Backpropagation algorithm

Input pair: 0.05, 0.10   Output pair 0.01, 0.99

$$net_{h1} = 0.15 * 0.05 + 0.2 * 0.1 + 0.35 * 1 = 0.3775$$

$$out_{h1} = \frac{1}{1+e^{-net_{h1}}} = \frac{1}{1+e^{-0.3775}} = 0.593269992$$

$$net_{o1} = 0.4 * 0.593269992 + 0.45 * 0.596884378 + 0.6 * 1 = 1.105905967$$



$$out_{o1} = \frac{1}{1+e^{-net_{o1}}} = \frac{1}{1+e^{-1.105905967}} = 0.75136507$$

$$E_{o1} = \frac{1}{2}(target_{o1} - out_{o1})^2 = \frac{1}{2}(0.01 - 0.75136507)^2 = 0.274811083$$

$$E_{o2} = 0.023560026$$

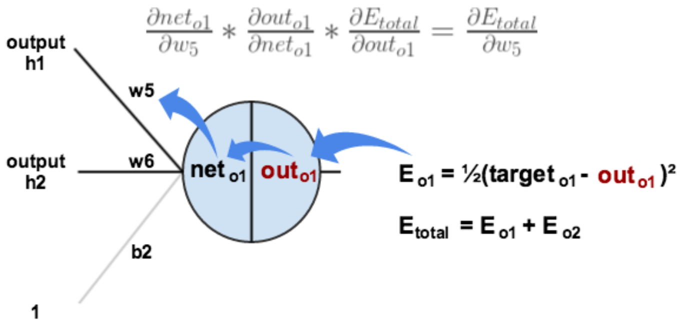
$$out_{o2} = 0.772928465$$

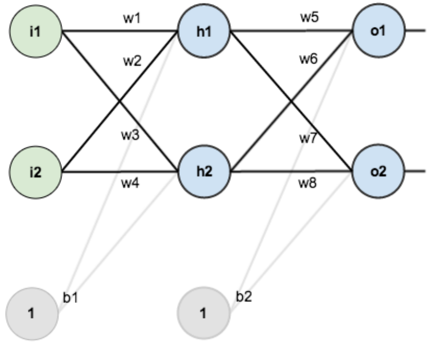
$$out_{h2} = 0.596884378$$

$$E_{total} = E_{o1} + E_{o2} = 0.274811083 + 0.023560026 = 0.298371109$$

https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/

Backpropagation algorithm





$$E_{total} = \frac{1}{2}(target_{o1} - out_{o1})^2 + \frac{1}{2}(target_{o2} - out_{o2})^2$$

$$\frac{\partial E_{total}}{\partial out_{o1}} = 2 * \frac{1}{2}(target_{o1} - out_{o1})^{2-1} * -1 + 0$$

$$\frac{\partial E_{total}}{\partial out_{o1}} = -(target_{o1} - out_{o1}) = -(0.01 - 0.75136507) = 0.74136507$$

$$\frac{\partial out_{o1}}{\partial net_{o1}} = out_{o1}(1 - out_{o1}) = 0.75136507(1 - 0.75136507) = 0.186815602$$

$$net_{o1} = w_5 * out_{h1} + w_6 * out_{h2} + b_2 * 1$$

$$\frac{\partial net_{o1}}{\partial w_5} = 1 * out_{h1} * w_5^{(1-1)} + 0 + 0 = out_{h1} = 0.593269992$$

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o1}} * \frac{\partial out_{o1}}{\partial net_{o1}} * \frac{\partial net_{o1}}{\partial w_5}$$

$$\frac{\partial E_{total}}{\partial w_5} = 0.74136507 * 0.186815602 * 0.593269992 = 0.082167041$$

$$w_6^+ = 0.408666186$$

$$w_7^+ = 0.511301270$$

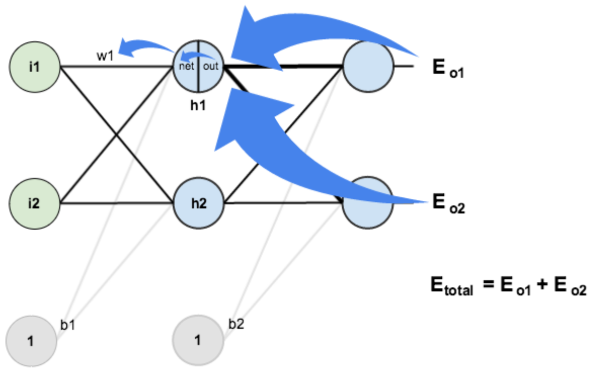
$$w_8^+ = 0.561370121$$

$$w_5^+ = w_5 - \eta * \frac{\partial E_{total}}{\partial w_5} = 0.4 - 0.5 * 0.082167041 = 0.35891648$$

… continued: Backpropagation algorithm

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h1}} * \frac{\partial out_{h1}}{\partial net_{h1}} * \frac{\partial net_{h1}}{\partial w_1}$$

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}}$$



$$out_{h1} = \frac{1}{1+e^{-net_{h1}}}$$

$$\frac{\partial out_{h1}}{\partial net_{h1}} = out_{h1}(1 - out_{h1}) = 0.59326999(1 - 0.59326999) = 0.241300709$$

$$net_{h1} = w_1 * i_1 + w_3 * i_2 + b_1 * 1 \qquad \frac{\partial net_{h1}}{\partial w_1} = i_1 = 0.05$$

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h1}} * \frac{\partial out_{h1}}{\partial net_{h1}} * \frac{\partial net_{h1}}{\partial w_1}$$

$$\frac{\partial E_{total}}{\partial w_1} = \left(\sum_o \frac{\partial E_{total}}{\partial out_o} * \frac{\partial out_o}{\partial net_o} * \frac{\partial net_o}{\partial out_{h1}}\right) * \frac{\partial out_{h1}}{\partial net_{h1}} * \frac{\partial net_{h1}}{\partial w_1}$$

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}} \qquad \frac{\partial E_{o1}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial net_{o1}} * \frac{\partial net_{o1}}{\partial out_{h1}}$$

$$w_1^+ = w_1 - \eta * \frac{\partial E_{total}}{\partial w_1} = 0.15 - 0.5 * 0.000438568 = 0.149780716$$

$$\frac{\partial E_{o1}}{\partial net_{o1}} = \frac{\partial E_{o1}}{\partial out_{o1}} * \frac{\partial out_{o1}}{\partial net_{o1}} = 0.74136507 * 0.186815602 = 0.138498562$$

$$w_2^+ = 0.19956143$$

$$net_{o1} = w_5 * out_{h1} + w_6 * out_{h2} + b_2 * 1 \qquad \frac{\partial net_{o1}}{\partial out_{h1}} = w_5 = 0.40$$

$$w_3^+ = 0.24975114$$

$$\frac{\partial E_{o1}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial net_{o1}} * \frac{\partial net_{o1}}{\partial out_{h1}} = 0.138498562 * 0.40 = 0.055399425$$

$$w_4^+ = 0.29950229$$

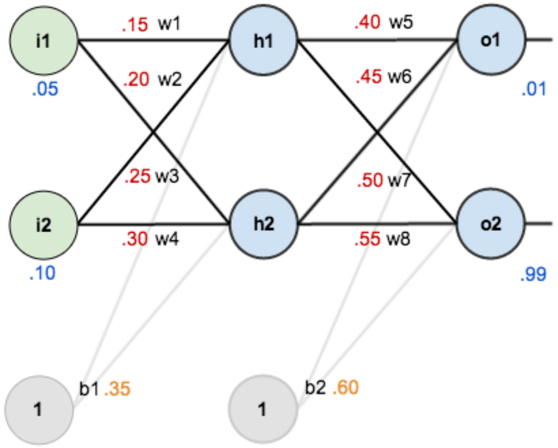
$$\frac{\partial E_{o2}}{\partial out_{h1}} = -0.019049119$$

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}} = 0.055399425 + -0.019049119 = 0.036350306$$

Old error: 0.298371109
New error: 0.291027924

… continued: Updated Coefficients & Error

	Epoch 1	Epoch 2	Epoch
w_1	0.1500	0.1498	
w_2	0.2000	0.1996	
w_3	0.2500	0.2498	
w_4	0.3000	0.2995	
w_5	0.4000	0.3589	
w_6	0.4500	0.4087	
w_7	0.5000	0.5113	
w_8	0.5500	0.5614	
I1,i2	O1,o2	O1,o2	O1,o2
0.05 0.10	0.7514, 0.77293	0.2910	0.0156 0.9846
Err.	0.2983	0.2910	3.5e-5



After 10k iteration, gets within 1% of the final result (0.01, 0.99)

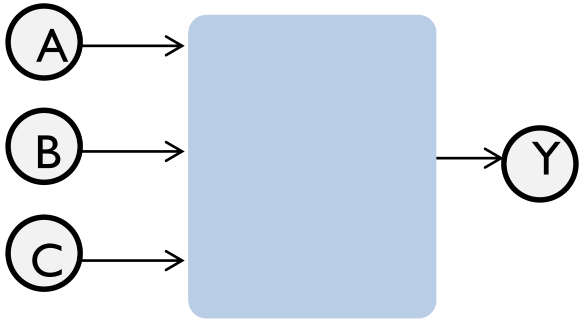
Outline

1. Introduction
2. A two input, single and multiple perceptron problem
3. Backpropagation and coefficient fitting
4. Machine learning and Karnaugh mapping
5. Other forms of Machine Learning (Unsupervised, optical, quantum)
6. Conclusions

Digital Synthesis has similar form

Online calculator:
http://www.32x8.com/

In			Out
A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1

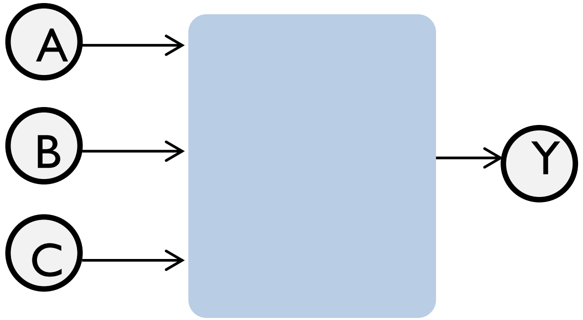


		BC			
		00	01	11	10
A	0	0	0	1	0
	1	0	1	1	1

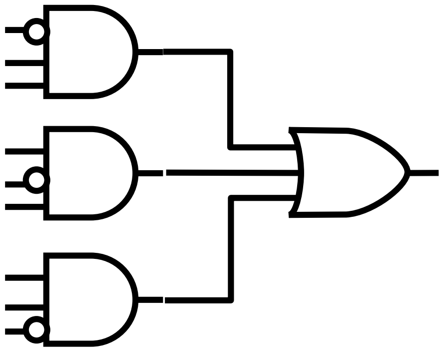
$Y = \overline{A}\,.\,B.C\; + \;A.\overline{B}.C\; + \;A.B.\overline{C}$

$S_1 = AC + BC + AB$

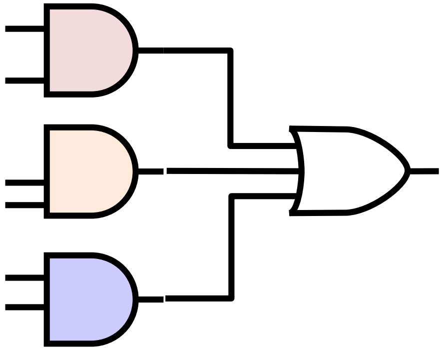
Neural Network and Digital logic Synthesis



$$Y = \overline{A}\,.\,B.C \;+\; A.\overline{B}.C \;+\; A.B.\overline{C}$$



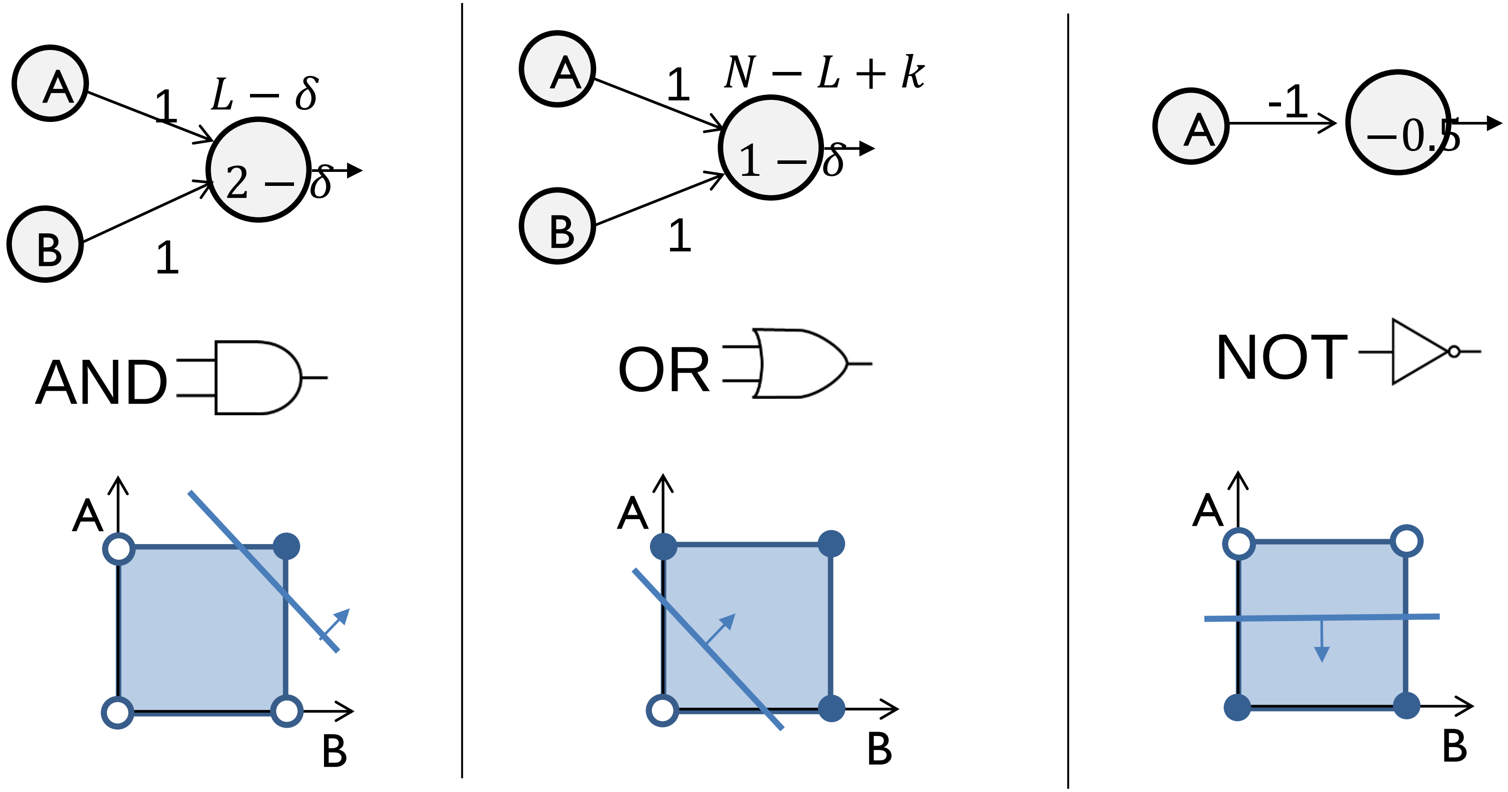
$$Y = AC \;+ BC \;+\; AB$$



Any logic circuit can be synthesized by AND, OR, and NOT gates …

A perceptron implements AND, ON, and NOT gates

L-N + k   here L= (positive input), N= total number =2, k=1 is the threshold for binary logic



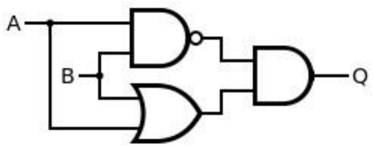
A	–
1	
0	

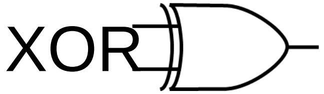
XOR cannot be represented by one layer (need for depth, MLP is universal Boolean function)

A	B	Z
0	0	0
1	0	1
0	1	1
1	1	0

	0	1
0	0	1
1	1	0

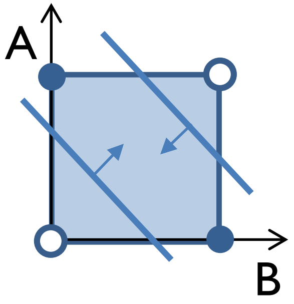
NAND





$$Z = \overline{A}.B + A.\overline{B}$$
$$= (A+B).(\overline{A} + \overline{B})$$
$$= (A+B).(\overline{A.B})$$

OR NAND



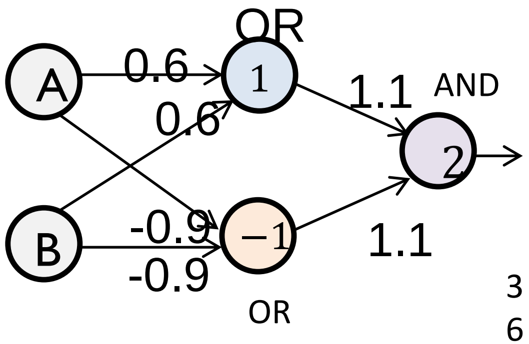
XOR cannot be represented by one layer (need for depth, MLP is universal Boolean function)

$$Z = \bar{A}.B + A.\bar{B}$$
$$= (A + B).(\bar{A} + \bar{B})$$

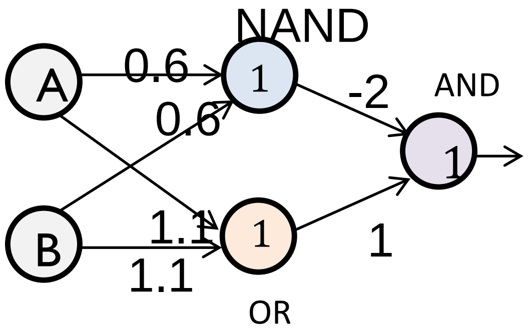
OR AND OR

$$Z = \bar{A}.B + A.\bar{B}$$
$$= (A + B).(\overline{A.B})$$

OR AND NAND



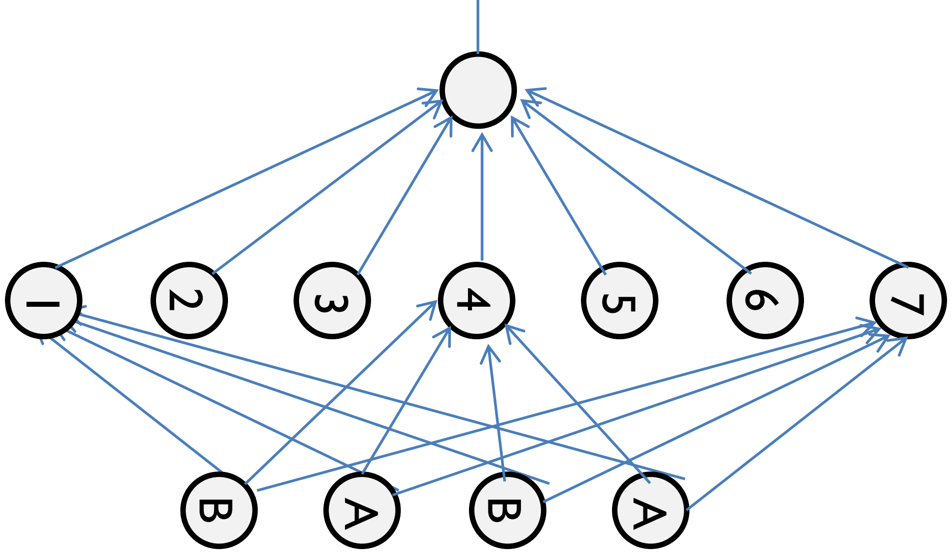
3 perceptrons, two layers
6 weights and 3 threshold
9 parameters.



Another example ….single layer in disjunctive normal form can express any truth table

	AB				
CD		00	01	11	10
	00	1			1
	01	1	1		
	11	1			
	10	1			1

$$Z = \overline{A}.\ \overline{B}.\overline{C}.\ \overline{D} + \overline{A}.\ \overline{B.C}.D + \overline{A}.\overline{(B)}C.D + \overline{A}.\ \overline{B}.C.\ \overline{D}$$
$$+\overline{A}.B.\overline{C}.D + A.\overline{B}.\overline{C}.D + \overline{A}.\ \overline{B.C}.\ \overline{D}$$



2

-1 -1 1

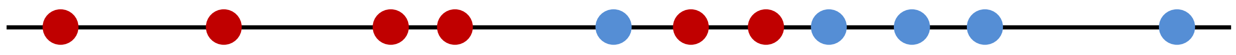
2^{N-1} perceptron width in a single layer. Exponential in N

Requires (O $N * 2^{N-1}$ weights superexponentnial in N

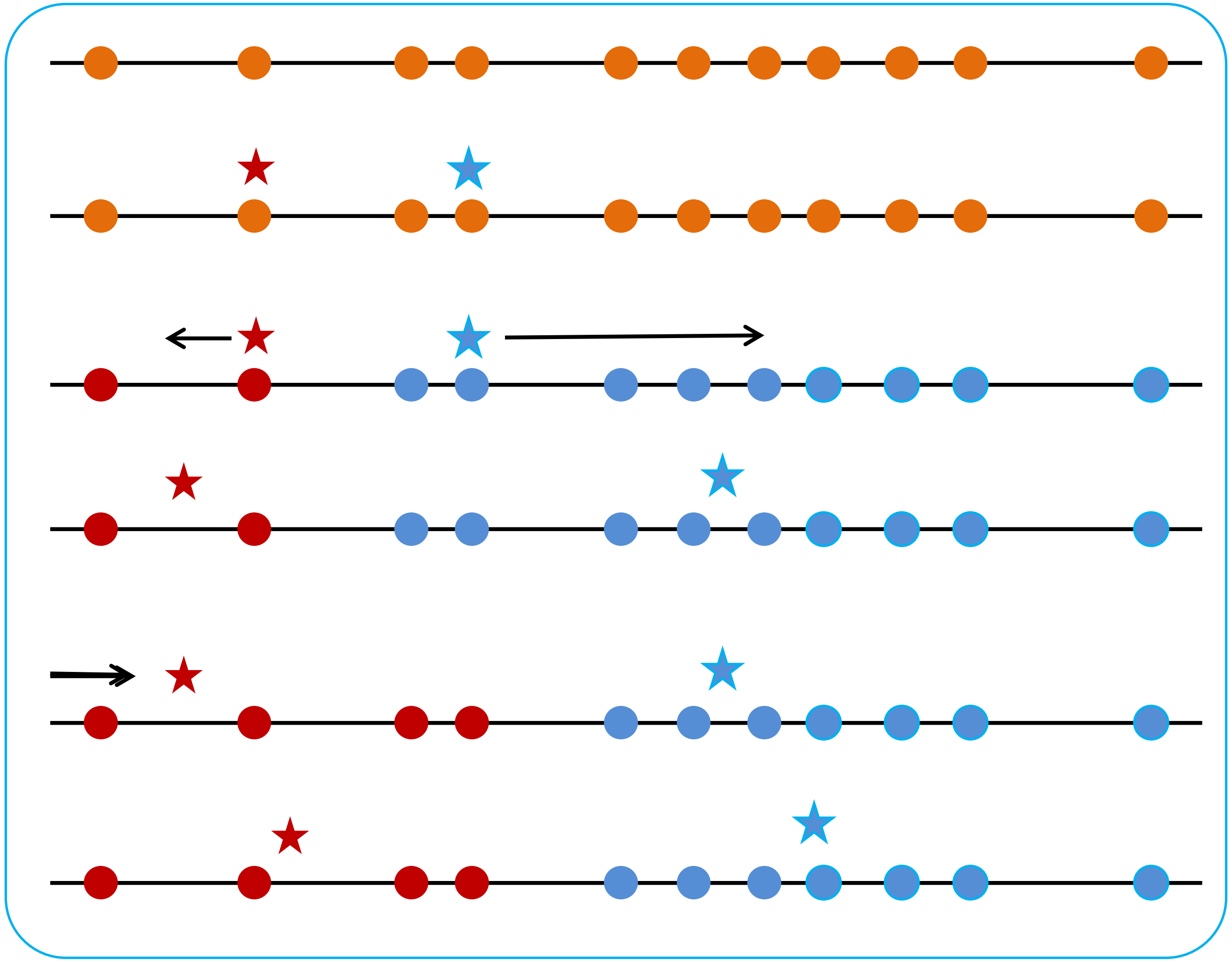
Outline

1. Introduction
2. A two input, single and multiple perceptron problem
3. Backpropagation and coefficient fitting
4. Machine learning and Karnaugh mapping
5. Other forms of Machine Learning (Unsupervised, optical, quantum)
6. Conclusions

Unsupervised k-mean clustering



Supervised



Unsupervised

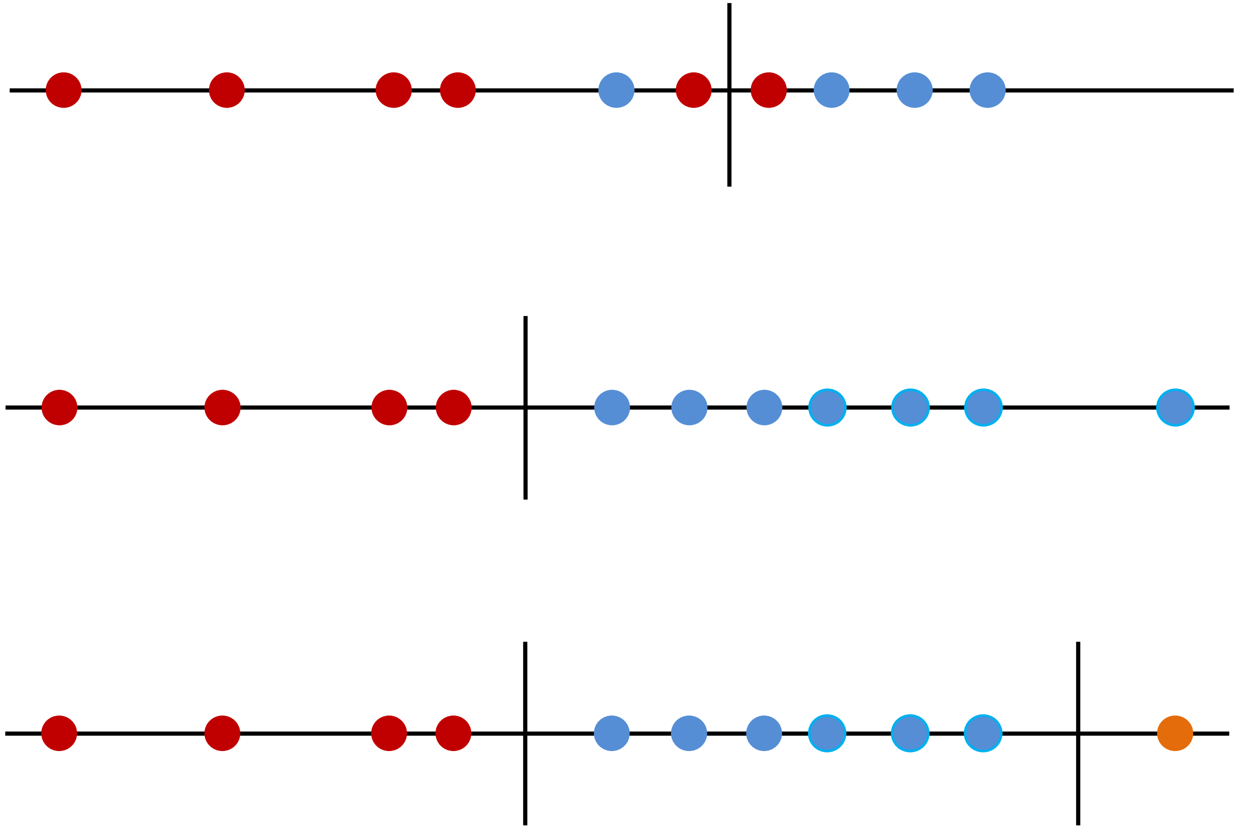
Two clusters
Pick centroids manually

Calculate ALL centroid-to-point distances. Shorter distance wins the cluster

Move centroid to new average

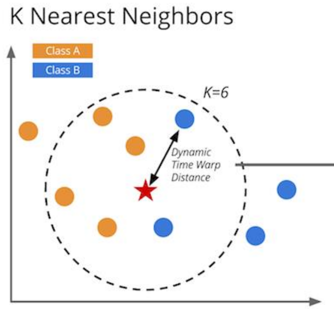
Repeat until points do not change cluster

Supervised vs. unsupervised clustering



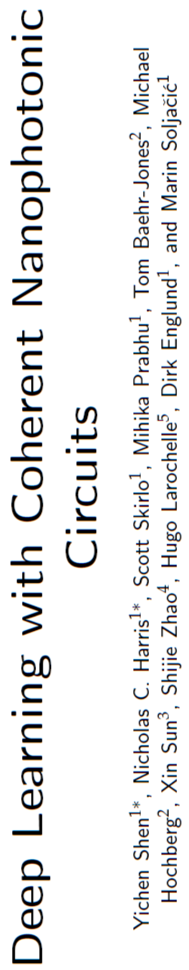
2-mean cluster

3-mean cluster

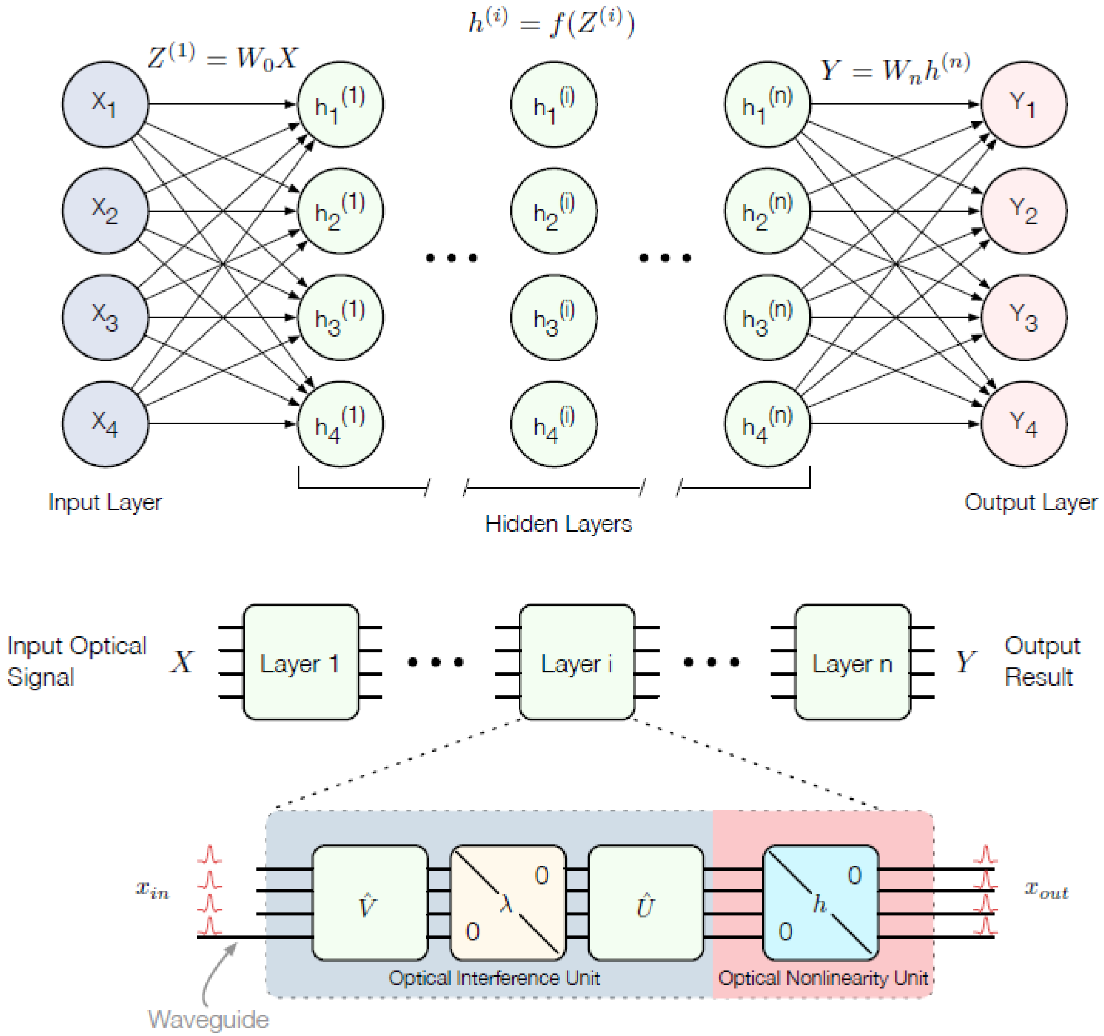


If you like this book, you may like this book
(because the folks that belong to your cluster do)

Deep Learning with Coherent Nanophotonic Circuits


Deep Learning with Coherent Nanophotonic Circuits

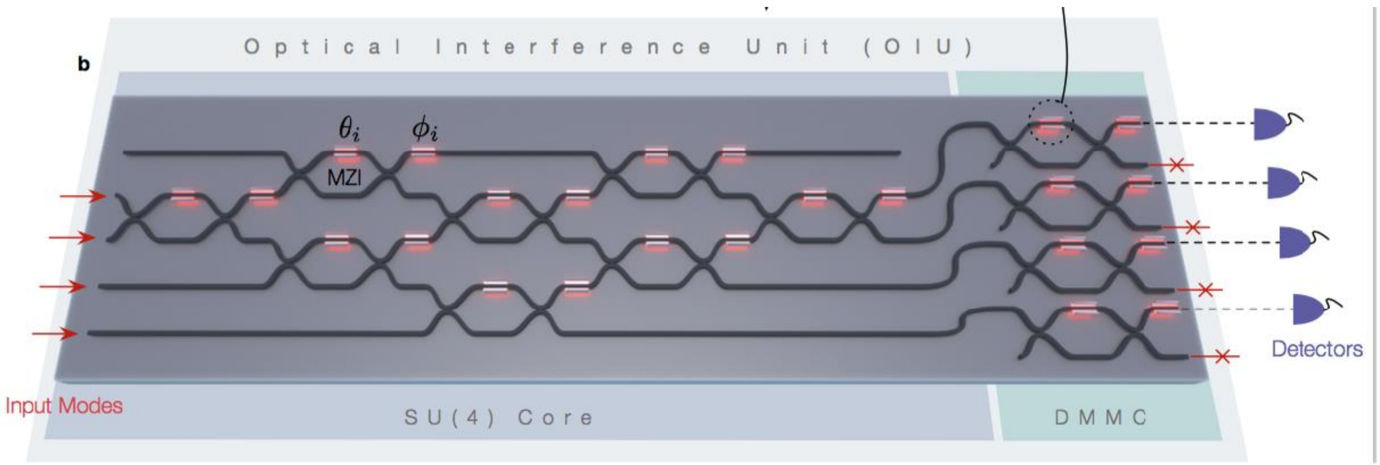
Yichen Shen[1*], Nicholas C. Harris[1*], Scott Skirlo[1], Mihika Prabhu[1], Tom Baehr-Jones[2], Michael Hochberg[2], Xin Sun[3], Shijie Zhao[4], Hugo Larochelle[5], Dirk Englund[1], and Marin Soljačić[1]



… continued: Nanophotonic Circuits

Deep Learning with Coherent Nanophotonic Circuits

Yichen Shen[1*], Nicholas C. Harris[1*], Scott Skirlo[1], Mihika Prabhu[1], Tom Baehr-Jones[2], Michael Hochberg[2], Xin Sun[3], Shijie Zhao[4], Hugo Larochelle[5], Dirk Englund[1], and Marin Soljačić[1]



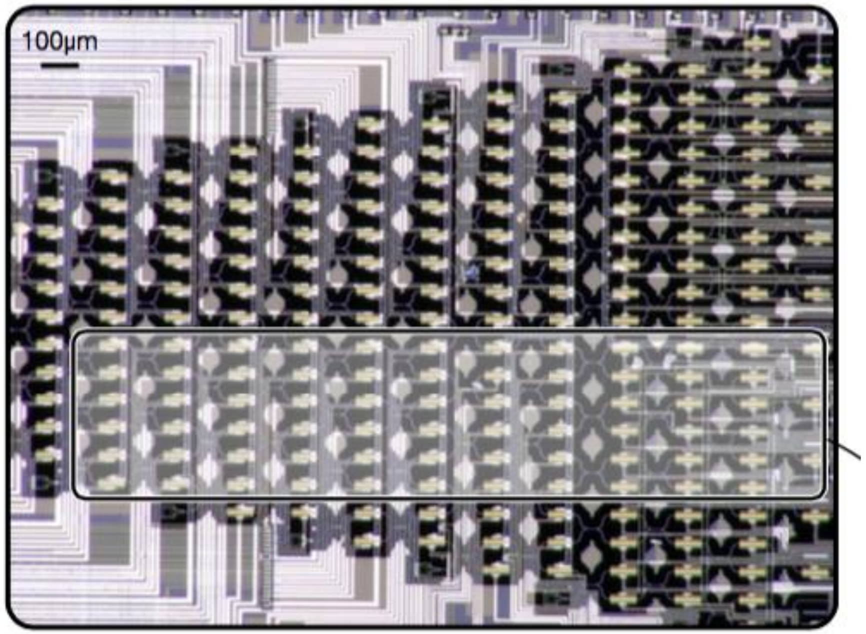


FIG. 2. **Illustration of Optical Interference Unit** a. Optical micrograph of an experimentally fabricated 22-mode on-chip optical interference unit; the physical region where the optical neural network program exists is highlighted in grey. The system acts as an optical field-programmable gate array–a test bed for optical experiments. b. Schematic illustration of the optical neural network program demonstrated here which realizes both matrix multiplication and amplification fully optically. c. Schematic illustration of a single phase shifter in the Mach-Zehnder Interferometer (MZI) and the transmission curve for tuning the internal phase shifter of the MZI

Conclusions

1. Multi-input, multiple layer network can be used to represent complex functional forms.

2. Since the approach does not rely on physics, it can handle complex interpolation problem. The out-of-domain predictions are difficult and error prone.

3. The mapping onto the digital logic synthesis (i.e. Karnaugh mapping) answers some key questions regarding the depth and width of the network. It also suggests how the neural network synthesizes logic step-by-step.

4. There are variety of machine learning tools: Supervised vs. unsupervised, random forest methods, optical methodologies. All these address specific issues, such as speed of classification, energy cost of training, etc.

References

The example involving passing probability vs. hours studied is taken from
Real Statistics with Excel: Logistics Regression http://www.real-statistics.com/logistic-regression

Logistic Regression by Excel in Youtube:
https://www.youtube.com/watch?v=EKRjDurXau0

Has step by step:
https://www.youtube.com/watch?v=jQI4pkKP9k4

The logistic calculator is here:
http://astatsa.com/Logit_Probit/

The corresponding Wikipedia page gives detailed information
https://en.wikipedia.org/wiki/Logistic_regression

Random Forest Model:
https://www.youtube.com/watch?v=gmmV4drPTS4

IRandom Forest Tutorial:
https://medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d

Support Vector Machine
p. 67 Machine Learning for Absolute Beginners by Oliver Theobald

An excellent presentation by Xavier Amatraian on NETFlix Recommendation Systems presented at 2012 ACM Meeting https://www.slideshare.net/xamat/netflix-recommendations-beyond-the-5-stars

Also see
https://www.slideshare.net/xamat/kdd-2014-tutorial-the-recommender-problem-revisited

An excellent tutorial in Kaggle regarding the need for multiple hidden layers (staircase problem)
http://blog.kaggle.com/2017/11/27/introduction-to-neural-networks/
http://blog.kaggle.com/2017/12/06/introduction-to-neural-networks-2/

TesnsorFlow: REF:
https://www.coursera.org/lecture/deep-learning-business/6-1-introduction-to-tensorflow-playground-ArfBs
https://cloud.google.com/blog/products/gcp/understanding-neural-networks-with-tensorflow-playground
https://developers.google.com/machine-learning/crash-course/introduction-to-neural-networks/playground-exercises

Machine Learning References

Intuitive explanation:
https://www.youtube.com/watch?v=nz-FrbAa8dY
https://www.youtube.com/watch?v=eX2sY2La4Ew

Excel: https://www.youtube.com/watch?v=jQI4pkKP9k4
https://www.youtube.com/watch?v=gNhogKJ_q7U

Simple Visual Explanation
https://www.youtube.com/watch?v=yIYKR4sgzI8

Derivation of the parameters
https://www.youtube.com/watch?v=YMJtsYIp4kg

https://www.youtube.com/watch?v=YMJtsYIp4kg

Step by step:
https://www.youtube.com/watch?v=HQ7P-Ft7Cuc

Artificial Neural Network and Digital Logic Synthesis

http://toritris.weebly.com/
http://toritris.weebly.com/perceptron-5-xor-how--why-neurons-work-together.html
https://www.youtube.com/watch?v=RALqlk7T4xc
http://www-inst.eecs.berkeley.edu/~ee40/fa03/lecture/lecture29.pdf
https://www.youtube.com/watch?v=FOf00W8WSBg
https://www.youtube.com/watch?v=UdpV-ksadkQ (must make the group in powers of 2)

How many hidden layers:
https://cse.buffalo.edu/~hungngo/classes/2010/711/lectures/0081.pdf

What size net gives valid generalization?
E. B. Baum and David Haussler

Machine Learning References

[1] Wide Residual Networks
Sergey Zagoruyko, Nikos Komodakis
(Submitted on 23 May 2016 (v1), last revised 14 Jun 2017 (this version, v4))
URL: https://arxiv.org/abs/1605.07146

[2] M. Bianchini and F. Scarselli, "On the Complexity of Neural Network Classifiers: A Comparison Between Shallow and Deep Architectures," in IEEE Transactions on Neural Networks and Learning Systems, vol. 25, no. 8, pp. 1553-1565, Aug. 2014.

[3] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016.

[4] https://www.quora.com/Why-are-neural-networks-becoming-deeper-more-layers-but-not-wider-more-nodes-per-layer#

- Discrete Weights:

Baldassi, C., Borgs, C., Chayes, J.T., Ingrosso, A., Lucibello, C., Saglietti, L. and Zecchina, R., 2016. Unreasonable effectiveness of learning neural networks: From accessible states and robust ensembles to basic algorithmic schemes. Proceedings of the National Academy of Sciences, 113(48), pp.E7655-E7662.

http://www.pnas.org/content/113/48/E7655.short

Review Questions

1. Any function can be represented by a single layer neurons. If so, why does one use multiple layer “deep” network?
2. Explain why we use tanh, sigmoid, or LiRu other saturated function in machine learning.
3. What are the support vectors in a support vector machine?
4. What are ID3, decision tree, and random forest model? Explain the applicability of the model.
5. What is the difference between supervised vs. non-supervised learning? How does the random clustering model work?