Session #04&05

CMSC 409: Artificial Intelligence

http://

Virginia Commonwealth University,
Fall 2023,
Dr. Milos Manic
(mmanic@vcu. edu)

1

CMSC 409: Artificial Intelligence

Session # 04&05

Topics for today

- Announcements
- Previous session review
- Learning of Agents…
  - *functionalities of learning units*
  - *AND, OR, NOT, other examples*
- Normalization

CMSC 409: *Artificial Intelligence*

Session # 04&05

Announcements

- Canvas
  - *New slides posted*
- Office hours zoom
  - *Zoom disconnects me after 45 mins of inactivity. Feel free to chat me via zoom if that happens and I will reconnect (zoom chat welcome outside of office hours as well)!*
- "zeroth" assignment (form a team)
  - *Deadline today (Aug.31) – all teams formed!*
- Project #1
  - *Deadline Sep. 14 (noon)*
- Paper (optional)
  - *First draft - due Sep. 12 (noon)*
  - *Think about the topic of your paper and confirm on 1st draft deliverables (class paper instructions)*
- Subject line and signature
  - *Please use [CMSC 409] Last_Name Question*

Vertically Integrated Projects

Multiyear • Multidisciplinary • Team-based

Search

Teams | Students | Faculty | Advising | Industry P

Home / AI/ML in Cybersecurity of Critical Infrastructures

AI/ML in Cybersecurity of Critical Infrastructures

The performance of the supervised ML algorithms has been remarkable and useful for several areas. However, the biggest limitation of the supervised approaches is the availability of labeled data, that's why unsupervised and self-supervised algorithms are gaining more popularity. In case of unsupervised ML, the algorithm learns the feature representation from the provided unlabeled data. This approach can be useful in cybersecurity, because unsupervised algorithms will focus on learning the normal communications of the network, and the small amount of anomalous data will be used for the algorithm testing and validating.

Goals:

- Develop an understanding of the differences between supervised and unsupervised ML
- Develop an understanding of the internet protocols
- Extracting cybersecurity features from network monitoring
- Develop an unsupervised ML for detecting network anomalies
- Create a visualization for displaying results.

Key Elements: Develop and validate an unsupervised machine learning algorithm for identification of abnormal network traffic. Extracting relevant features from network packets. Create a user interface for displaying results.

Skills: Machine Learning, Unsupervised learning, CyberSecurity, Tensorflow, Keras, python, Programming IDE

Projects:

- Open source and custom network security monitoring
- Feature extractor of cybersecurity features from network monitoring
- Developing a database for managing the data pipelines
- Developing of a machine learning anomaly detection system
- Create a dashboard for result visualization

You can earn 1-2 credits with VIP

https://vip.vcu.edu/cybersecurity

• From the AI world….

SCIENCE \ TECH

Elon Musk promises demo of a workin Neuralink device on Friday

The company is developing brain-machine interface technology

By Jon Porter and James Vincent | Aug 26, 2020, 5:06am EDT

(Aug. 2020):

SHARE

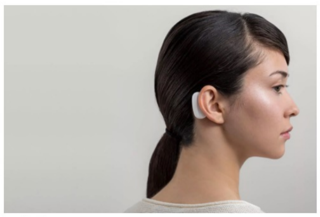


Image: Neuralink

Elon Musk has said that his secretive neurotech firm Neuralink will demonstrate a working "device," presumably a brain-machine interface, at 6PM ET on Friday. Musk has spoken repeatedly about his belief that BMI devices are needed to help humans keep up with AI by supplementing our brainpower, but right now, his goal is much simpler: to create an implantable device that lets people control phones or computers with their mind.

Musk initially announced the August 28th "progress update" back in July, and has now offered more details on what will be shown. He says the update will include the unveiling of a second-generation robot designed to attach the company's technology to the brain, and a demo of neurons "firing in real-time," though it's not clear exactly what is meant by this.

Even compared to Musk's other ventures like Tesla and SpaceX, Neuralink is ambitious. The company wants to connect to the brain using flexible electrodes thinner than a human hair that it calls "threads." Current BMI devices use stiff electrodes for this job, which can cause damage. But inserting flexible electrodes is a much more delicate and challenging task, ...uilding a "sewing machine" like robot to do the job.

Introducing Neuralink:
https://www.youtube.com/watch?v=kPGa_FuGPIc

WIRED BACKCHANNEL BUSINESS CULTURE GEAR IDEAS SCIENCE SECURITY SIGN IN | SUBSCRIBE

WILL KNIGHT BUSINESS 08.25.2020 08:00 AM

A Dogfight Renews Concerns About AI's Lethal Potential

(Aug. 2020):

Alphabet's DeepMind pioneered reinforcement learning. A California company used it to create an algorithm that defeated an F-16 pilot in a simulation.

The Pentagon's AlphaDogfight contest pitted artificial intelligence programs for directing an F-16 against an Air Force pilot.
ICON SPORTSWIRE

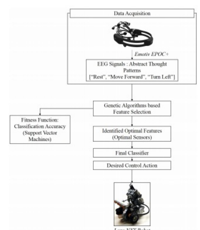
AlphaDogfight Trials (Apr. 2020):
https://www.youtube.com/watch?v=-cqvNWZBku4

Driving robot via thought:

K. Amarasinghe, P. Sivils, M. Manic, "EEG Feature Selection for Thought Driven Robots using Evolutionary Algorithms," in Proc. of IEEE 9th International Conference on Human System Interaction, IEEE HSI 2016, Portsmouth, England, July 6-8, 2016. PDF, DOI: 10.1109/HSI.2016.7529657

K. Amarasinghe, D. Wijayasekara, M. Manic, "EEG based brain activity monitoring using Artificial Neural Networks," in Proc. of IEEE 7th International Conference on Human System Interaction, IEEE HSI 2014, Lisbon, Portugal, June 16-18, 2014. Abstract , PDF, DOI: 10.1109/HSI.2014.6860449

D. Wijayasekara, M. Manic, "Human Machine Interaction via Brain Activity Monitoring," in Proc. of IEEE 6th International Conference on Human System Interaction, IEEE HSI 2013, Gdansk, Poland, June 6-8, 2013. Abstract, PDF, DOI: 10.1109/HSI.2013.6577809



Epilepsy:

H. Carey, M. Manic, "Epileptic Spike Detection with EEG using Artificial Neural Networks," in Proc. of IEEE 9th International Conference on Human System Interaction, IEEE HSI 2016, Portsmouth, England, July 6-8, 2016.PDF, DOI: 10.1109/HSI.2016.7529614

Learning of Agents…

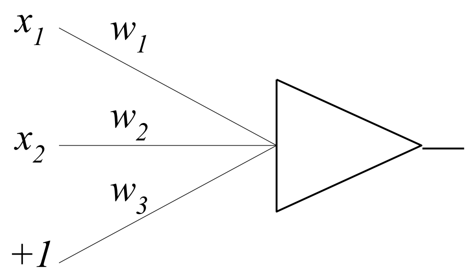
Counterparts to biological neurons, examples

© M. Manic, CMSC 409: Artificial Intelligence, F23

McCulloch-Pitts neurons

$$net = \sum_{i=1}^{n} w_i x_i \qquad out = \begin{cases} 1 & if\ net \geq 0 \\ 0 & if\ net < 0 \end{cases}$$

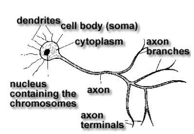
(unipolar hard activation)

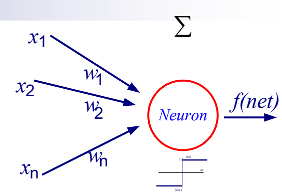


Unipolar neurons –> possible outputs are: ?

© M. Manic, CMSC 409: Artificial Intelligence, F23

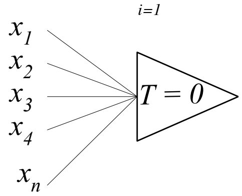
Artificial neuron

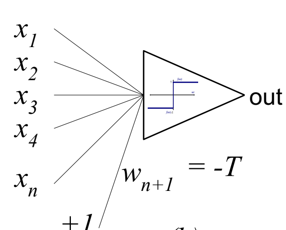




$$net = \sum_{i=1}^{n} w_i x_i + w_{n+1}$$

$$net = \sum_{i=1}^{n} w_i x_i$$



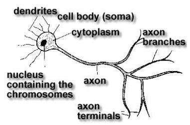


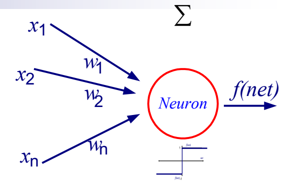
(a) (b)

Extra connection is setting the threshold.

© M. Manic, CMSC 409: Artificial Intelligence, F23

Artificial neuron





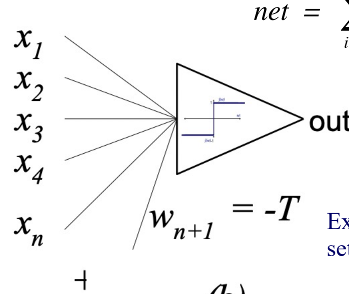
X1 * Wx1 + X2 * Wx2+…+Xn*Wxn+Xn+1*Wxn+1>=0

X1 * Wx1 + X2 * Wx2+…+ Xn*Wxn >= - Wxn+1

X1 * Wx1 + X2 * Wx2+…+ Xn*Wxn >= T

$$net = \sum_{i=1}^{n} w_i x_i + w_{n+1}$$

Outcome of inequality above can be either true or false.
When true, we say neuron "fires".



Extra connection is setting the threshold.

(b)

© M. Manic, CMSC 409: Artificial Intelligence, F23

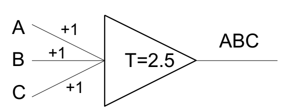
McCulloch-Pitts neurons

$$net = \sum_{i=1}^{n} w_i x_i + w_{n+1}$$

$$out = \begin{cases} 1 & if\ net \geq 0 \\ 0 & if\ net < 0 \end{cases}$$

Unipolar neurons –> possible outputs are: ?

AND



000
001
010
011->
…
111

Basic logical operations

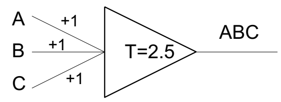
© M. Manic, CMSC 409: Artificial Intelligence, F23

McCulloch-Pitts neurons

$$net = \sum_{i=1}^{n} w_i x_i + w_{n+1} \qquad out = \begin{cases} 1 & if\ net \geq 0 \\ 0 & if\ net < 0 \end{cases}$$

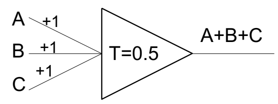
Unipolar neurons –> possible outputs are: ?

AND



(2, 3]

OR



Basic logical operations

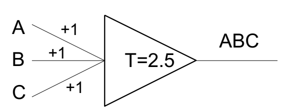
11

McCulloch-Pitts neurons

$$net = \sum_{i=1}^{n} w_i x_i + w_{n+1} \qquad out = \begin{cases} 1 & if\ net \geq 0 \\ 0 & if\ net < 0 \end{cases}$$

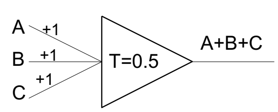
Unipolar neurons –> possible outputs are: ?

AND



(2, 3]

OR



(0, 1]

Basic logical operations

12

McCulloch-Pitts neurons

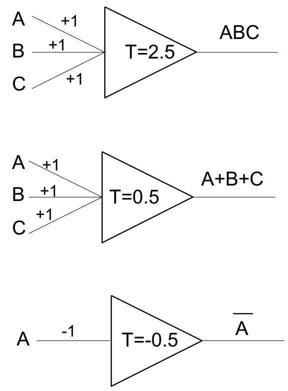
$$net = \sum_{i=1}^{n} w_i x_i + w_{n+1} \qquad out = \begin{cases} 1 & if\ net \geq 0 \\ 0 & if\ net < 0 \end{cases}$$

Unipolar neurons –> possible outputs are: ?

AND

OR

NOT



(2, 3]

(0, 1]

Basic logical operations

McCulloch-Pitts neurons

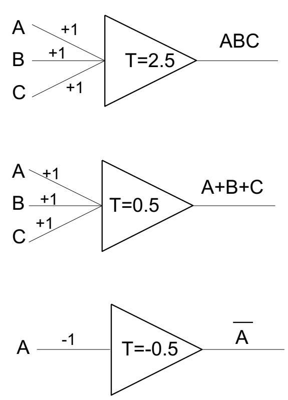
$$net = \sum_{i=1}^{n} w_i x_i + w_{n+1} \qquad out = \begin{cases} 1 & if\ net \geq 0 \\ 0 & if\ net < 0 \end{cases}$$

Unipolar neurons –> possible outputs are: ?

AND

OR

NOT



(2, 3]

(0, 1]

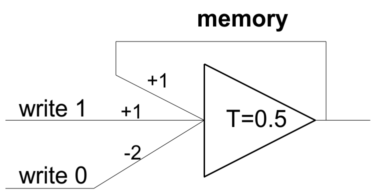
(-1, 0]

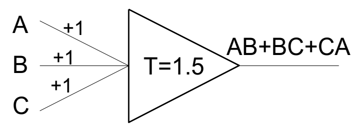
Basic logical operations

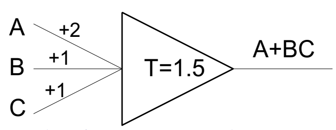
McCulloch-Pitts neurons

$$net = \sum_{i=1}^{n} w_i x_i + w_{n+1}$$

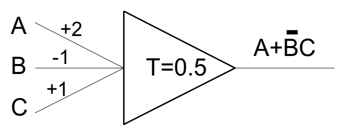
$$out = \begin{cases} 1 & if\ net \geq 0 \\ 0 & if\ net < 0 \end{cases}$$







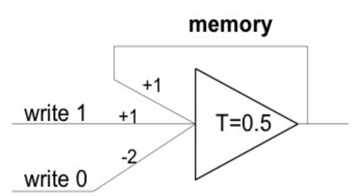
Boolean function: A or B and C



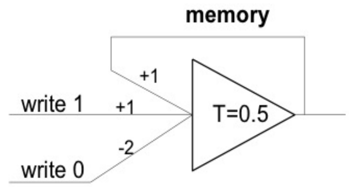
Boolean function: A or not B and C)

Examples of logical operations using McCulloch-Pitts neurons. Each one performs a different Boolean function.

Example…



Example…



Neuron definition:

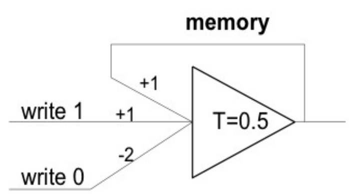
$$net = \sum_{i=1}^{n} w_i x_i + w_{n+1}; \qquad o = \begin{cases} 1, & if\, net \geq 0 \\ 0, & if\, net < 0 \end{cases}$$

Always start with definition of net and output!
The rest of the solution may not hold for different definition
(you may lose points definition missing)

© M. Manic, CMSC 409: Artificial Intelligence, F23

17

Example…



Neuron definition:

$$net = \sum_{i=1}^{n} w_i x_i + w_{n+1}; \qquad o = \begin{cases} 1, & if\, net \geq 0 \\ 0, & if\, net < 0 \end{cases}$$

Always start with definition of net and output!

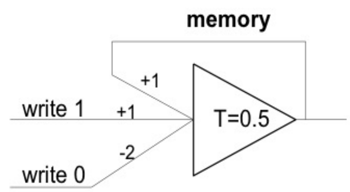
W1 (ti)	W0 (ti)	net (ti)	M (ti-1)	Output (ti)	Push/was
				1	1/x
1	0	+2	1	1	1/1
0	1	-1	1	0	0/1
0	1	-2	0	0	0/0
1	0	+1	0	1	1/0
1	1	not allowed at the same time			
0	0	relies on memory, i.e. output remains			

Threshold?

© M. Manic, CMSC 409: Artificial Intelligence, F23

18

Example…



Neuron definition:

$$net = \sum_{i=1}^{n} w_i x_i + w_{n+1}; \qquad o = \begin{cases} 1, & if\ net \geq 0 \\ 0, & if\ net < 0 \end{cases}$$

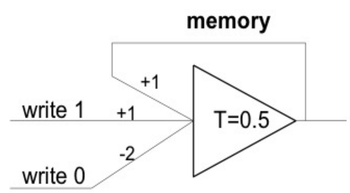
Always start with definition of net and output!

W1 (ti)	W0 (ti)	net (ti)	M (ti-1)	Output (ti)	Push/was
				1	1/x
1	0	+2	1	1	1/1
0	1	-1	1	0	0/1
0	1	-2	0	0	0/0
1	0	+1	0	1	1/0
1	1	not allowed at the same time			
0	0	relies on memory, i.e. output remains			

Threshold: According to possible net values, threshold can be anywhere between (-1,+1].

19

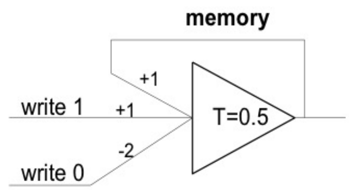
Example…(cont).



Or, if you do consider cases (0,0) and (1,1):

20

Example…(cont).

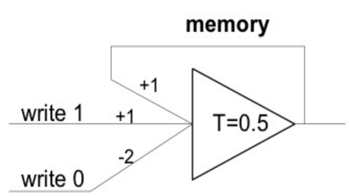


Or, if you do consider cases (0,0) and (1,1):

W1(t)	W0(t)	mem(t-1)	net(t)	mem(t) {output}
0	0	0	0	0
0	0	1	+1	1
0	1	0	-2	0
0	1	1	-1	0
1	0	0	+1	1
1	0	1	+2	1
1	1	x	x	x

Threshold?

Example…(cont).

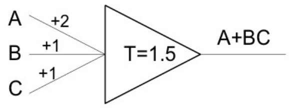


Or, if you do consider cases (0,0) and (1,1):

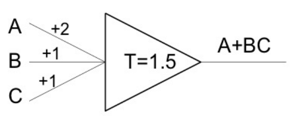
W1(t)	W0(t)	mem(t-1)	net(t)	mem(t) {output}
0	0	0	0	0
0	0	1	+1	1
0	1	0	-2	0
0	1	1	-1	0
1	0	0	+1	1
1	0	1	+2	1
1	1	x	x	x

Threshold: According to possible net values, threshold can be anywhere between (0,+1].

Example…



Example…

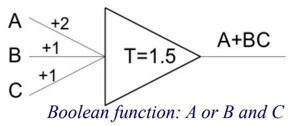


Provide first neuron definition:

$$net = \sum_{i=1}^{n} w_i x_i + w_{n+1}; \qquad o = \begin{cases} 1, & if\, net \geq 0 \\ 0, & if\, net < 0 \end{cases}$$

Always start with definition of net and output!

Example…



Boolean function: A or B and C

Provide first neuron definition:

$$net = \sum_{i=1}^{n} w_i x_i + w_{n+1}; \qquad o = \begin{cases} 1, & if\, net \ge 0 \\ 0, & if\, net < 0 \end{cases}$$

Always start with definition of net and output!

Next provide the truth table with inequalities:

A	B	C	A+BC	inequalities	out
0	**0**	**0**	0	0 < T; T=1.5	0 < 1.5
0	**0**	**1**	0	*wc* < T	+1 < 1.5
0	**1**	**0**	0	*wb* < T	+1 < 1.5
0	**1**	**1**	1	*wb*+*wc* ≥ T	+2 ≥ 1.5
1	**0**	**0**	1	*wa* ≥ T	+2 ≥ 1.5
1	**0**	**1**	1	*wa*+ *wc* ≥ T	+3 ≥ 1.5
1	**1**	**0**	1	*wa*+ *wb* ≥ T	+3 ≥ 1.5
1	**1**	**1**	1	*wa*+ *wb*+*wc* ≥ T	+4 ≥ 1.5

$Aw_A+Bw_B+Cw_C>=T$; 2A + 1B + 1C >= T

Project 1:

Normalization

- refresher
- when/why needed

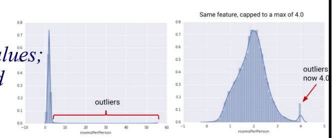
Normalization techniques at a glance (Google ML)

- Original data: $x = (x_1, x_2, \dots, x_n)$
- Normalized data: $z = (z_1, z_2, \dots, z_n)$

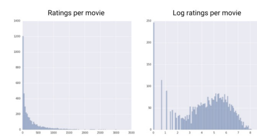
- Scaling to a range (min-max scaling)
  - *To values between 0 & 1; need to know upper/lower bounds; data somewhat uniformly distributed (e.g. age, not income)*

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}$$

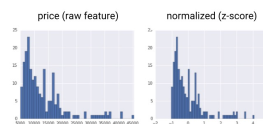
- (Feature) clipping
  - *In case of extreme outlies; caps values outside certain values; clip all temp values above 40 to exactly 40; can be applied before/after other normalizations*



- Log scaling $z_i = \log(x_i)$
  - *When handful values have many points, while most other just a few (e.g. movie ratings – most have few, but a few have lots of ratings)*



- Z-score (standardization) $z_i = (x_i - \mu) / \sigma$
  - *Number of standard deviations* (σ) *away from the mean* (μ)
  - *When you have a few outliers, but not so extreme that clipping is needed; to ensure μ=0 and* σ=1;



© M. Manic, CMSC 409: Artificial Intelligence, F23

Google ML https://developers.google.com/machine-learning/data-prep/transform/normalization

27

Normalization (min-max scaling)

- Normalizing data to [0-1] range
  - Normalization is used to scale data

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}$$

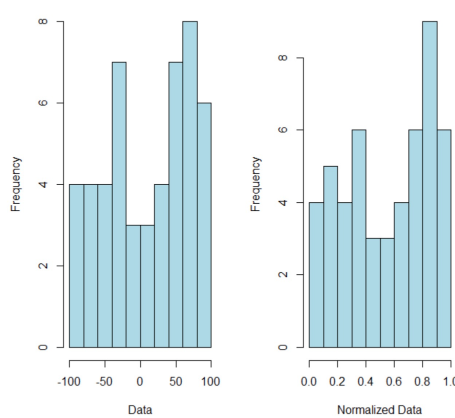
  - Original data:

$$x = (x_1, x_2, \dots, x_n)$$

  - Normalized data:

$$z = (z_1, z_2, \dots, z_n)$$

Normalization along one dimension



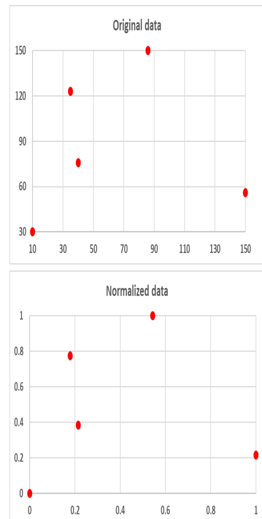
https://stats.stackexchange.com/questions/70801/how-to-normalize-data-to-0-1-range

© M. Manic, CMSC 409: Artificial Intelligence, F23

28

Normalization (min-max scaling)

Normalization along two dimensions



- Normalizing data to [0-1] range
  - Normalization is used to scale data

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}$$

  - Original data:

$$x = (x_1, x_2, \dots, x_n)$$

  - Normalized data:

$$z = (z_1, z_2, \dots, z_n)$$

Note: sensitive to outliers!

More at: Statology, Wiki statistics, Python sklearn, StackExchange, Google ML

© M. Manic, CMSC 409: Artificial Intelligence, F23

Session 04&05, Updated on 9/5/23 4:46:26 PM

29

Standardization

(if you have used this in place of normalization, no need to change)

- Standard score (z-score):

$$z_i = \frac{x_i - \mu}{\sigma}$$

- σ – standard deviation of the population, μ - mean of the population, z - distance between x_i and the population mean in units of the standard deviation (z is negative when the x is below the mean, positive when above)

- Normalized data:

$$z = (z_1, z_2, \dots, z_n)$$

Note:

- Standardization creates new data not bounded (unlike normalization); can be negative.
- Normalization usually means to scale a variable to have values between 0 and 1, while standardization transforms data to have a mean of zero and a standard deviation of 1.
- Normalization or standardization, it should be applied to a whole (complete) dataset.

© M. Manic, CMSC 409: Artificial Intelligence, F23

Session 04&05, Updated on 9/5/23 4:46:26 PM

30

Things to remember…

- **For now…**
  - *Working with binary neurons only (0, 1 for inputs & outputs)*
  - *Later we will relax to any real values*
- **Presented AND & OR neurons**
  - *The only difference is T (same # of inputs, same input weights)*
  - *If you change T, you can create other neuron functionality (like AB+BC+CA)*
- **Always…**
  - *Define net and output first.*
  - *Solution is not valid otherwise…*
- **Threshold vs. bias**
  - *Threshold is bias (with opposite sign), and vice versa*
  - *Careful to count T (bias) once only*
- **Truth tables**
  - *Always consider ALL possible input patterns*
  - *Neuron inequality must be true for all possible input pattern*