

Layman to Layman

Kolmogorov-Arnold Networks

Open-space Machine Learning Paper Reading Club

16 Nov 2025



Thura Aung

thuraaung.ai.mdy@gmail.com

Software Engineering (International Program)

King Mongkut's Institute of Technology Ladkrabang

- Lab. Member, Language Understanding Lab, Myanmar.
- AI Engineer Intern (Part-time), AI Singapore/ NUS, Singapore.
- Machine Learning Research (Part-time), Looloo Tech, Thailand.

Intelligence

- Intelligence is a continuum not a dichotomy.
 - ဉာဏ်ရည် အဆင့်အတန်း ဟူသည် ပုံးနှံသည့်သဘော၊ ပကတိ အစွမ်း သဘော မဟုတ်
 - လူသား သည် ဘာသာစကား**ကြောင့်** ထိုးတက်
 - လူဘာသာစကားဟာ လူမဟုတ်တဲ့ တြော်းတိရှိစွာဖော်တွေရဲ့ ဘာသာစကားနဲ့ မတူပါ
 - compositional/productive (“အဓိပ္ပာယ်ရှိတဲ့ အသုံးအနှစ်းတွေက တြော်းအဓိပ္ပာယ်ရှိတဲ့ အသုံးအနှစ်းတွေကနေ တည်ဆောက်ထားတာ”) **ဖြစ်ပါတယ်**
 - open-ended → အချိန်**ကြောလတာနဲ့အမျှ** စကားလုံးအသစ်တွေနဲ့ စာ**ကြောင်း**အသစ်တွေ ထပ်မံဖန်တီးလာနိုင်ပါတယ်
 - abstract → စိတ္တဇာ (ဒြပ်ကင်းသဘော) နာမ်တွေကို ဖော်ပြန်

<https://cims.nyu.edu/ai/educational-programs/broad-interest-courses/big-ideas-artificial-intelligence/>

Can Consciousness be expressed as a function?

- Consciousness ကို information processing အနေနဲ့ သဘောထားတယ်။
- Sensory inputs နဲ့ internal states တွေကို awareness နဲ့ behavior အပြစ် map လုပ်တယ်။
- Functionalism, neuroscience, AI models တွေက ထောက်ခံတယ်။
- Computational theories (global workspace theory, Integrated Information Theory)
- Consciousness ဆိုတာ neurons (သို့) algorithms တွေနဲ့ အကောင်အထည်ဖော်ထားတဲ့ computable function တစ်ခုဖြစ်တယ်။



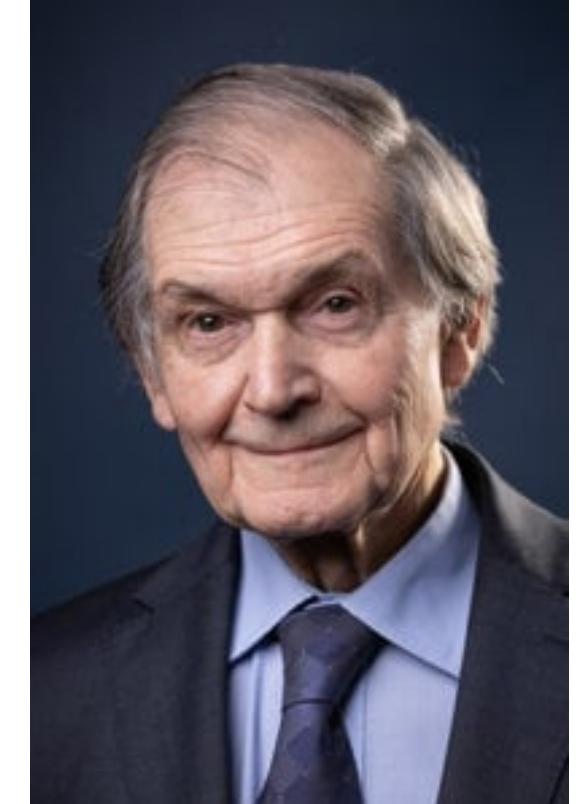
Bernard Baars



Giulio Tononi

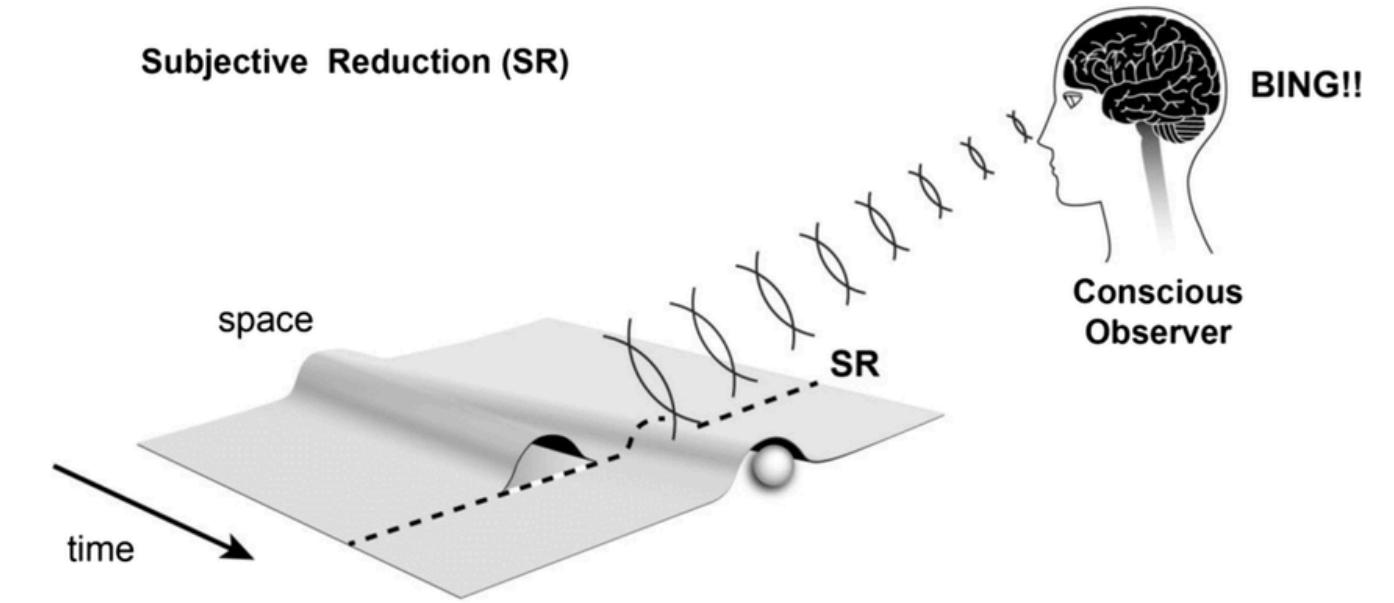
Consciousness CANNOT be expressed as a function

- Consciousness မှာ non-computable processes
တွေ ပါဝင်နေတယ်။
- Gödel argument အရ လူသားရဲ့ ထိုးထွင်းညက်က
algorithmic rules တွေကို ကျော်လွှန်တယ်။
- Consciousness ဘာ quantum gravitational
state reduction (Orch-OR) ကနေ ပေါ်စွာကဲ
တာ။ (Penrose's Claim)
- Non-algorithmic ဖြစ်လို့ ဘယ်လို့ function ဒါမှ
မဟုတ် Turing machine နဲ့မှ ဖမ်းယူလို့ မရနိုင်ဘူး။
- Neural computation ကို ကျော်လွှန်တဲ့ new
physics လိုအပ်တယ်။

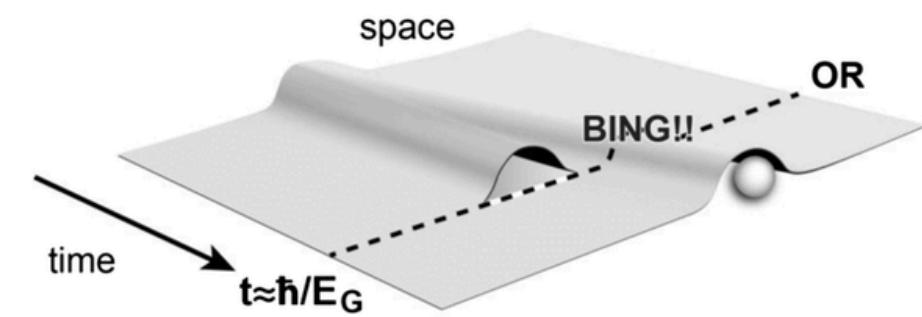


Roger Penrose

Subjective Reduction (SR)



Objective Reduction (OR)



Learning

- အတွေ့အကြုံများမှ ဉာဏ်ရည်ပေါ်ထွက်လာသည်

ကျို့စာတော်တို့ ဘယ်လို့ "သိ" ကြတဲလဲ

- အရာဝတ္ထုတွေနဲ့ အပြန်အလှန်ဆက်ဆံခြင်းအားဖြင့်: မြင်ခြင်း၊ ကြားခြင်း၊ ထံတွေ့ခြင်း၊ အရသာခံခြင်း စသည်ဖြင့်

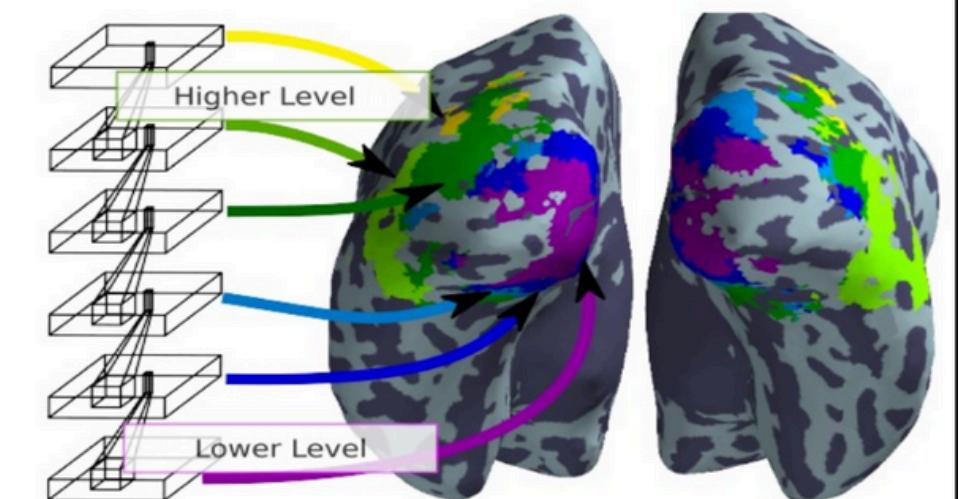
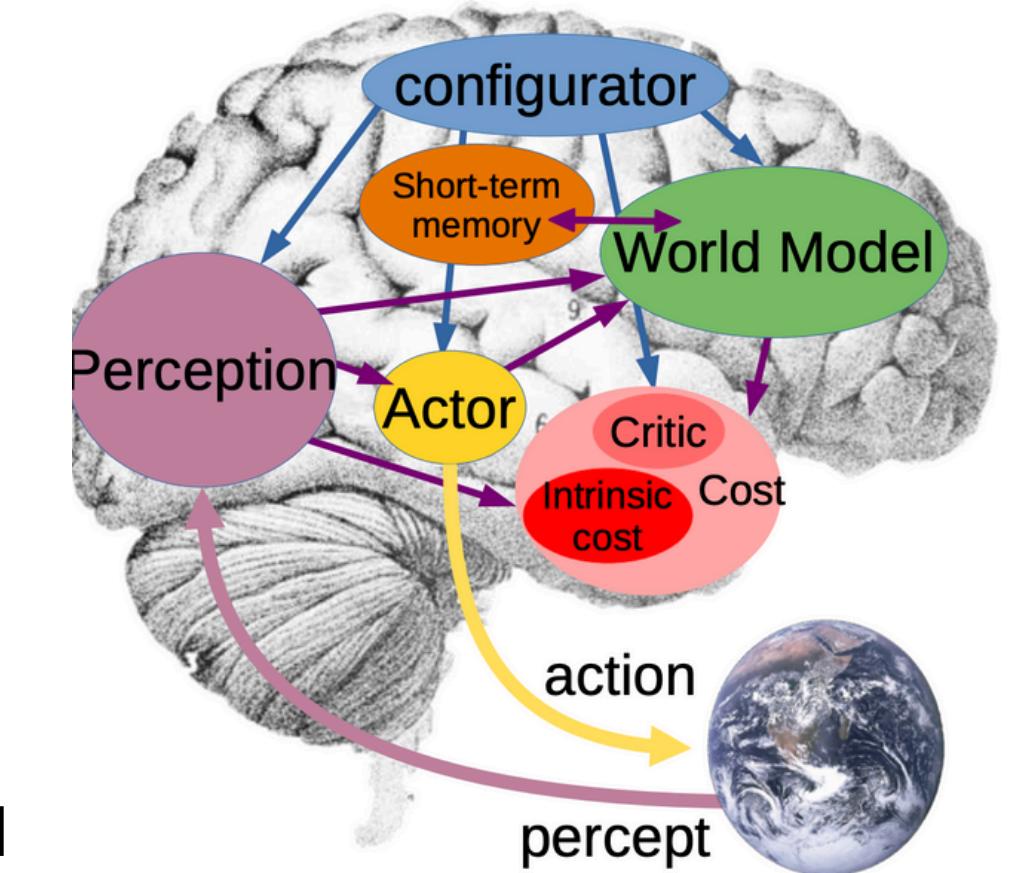
ကျို့စာတော်တို့ဟာ ကျွမ်းကျင်မူတွေကို ဘယ်လို့ "သင်ယူ" ကြတဲလဲ။

- လက်တွေ့လုပ်ဆောင်ကြည့်ခြင်းအားဖြင့်: လမ်းလျှောက်ခြင်း၊ ပြေးခြင်း၊ နောက်ပြန်ပစ်ခြင်း (back-flip)၊ ချက်ပြုတ်ခြင်း စသည်ဖြင့်

ဒဏ္ဍာရီထဲက ပျော်စေထိုးကို မြင်ဖူးလား?

- အရေးအသား၊ အပြောအဆိုက တဆင့် သိရတာ (အမှန် အမှား အကုန်ပါ)

ဘာသာစကားတွေလည်း မတဲ့



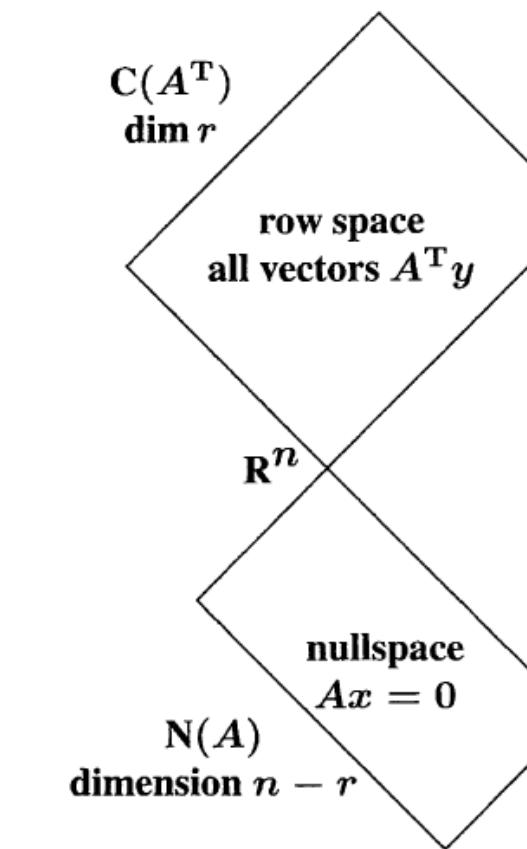
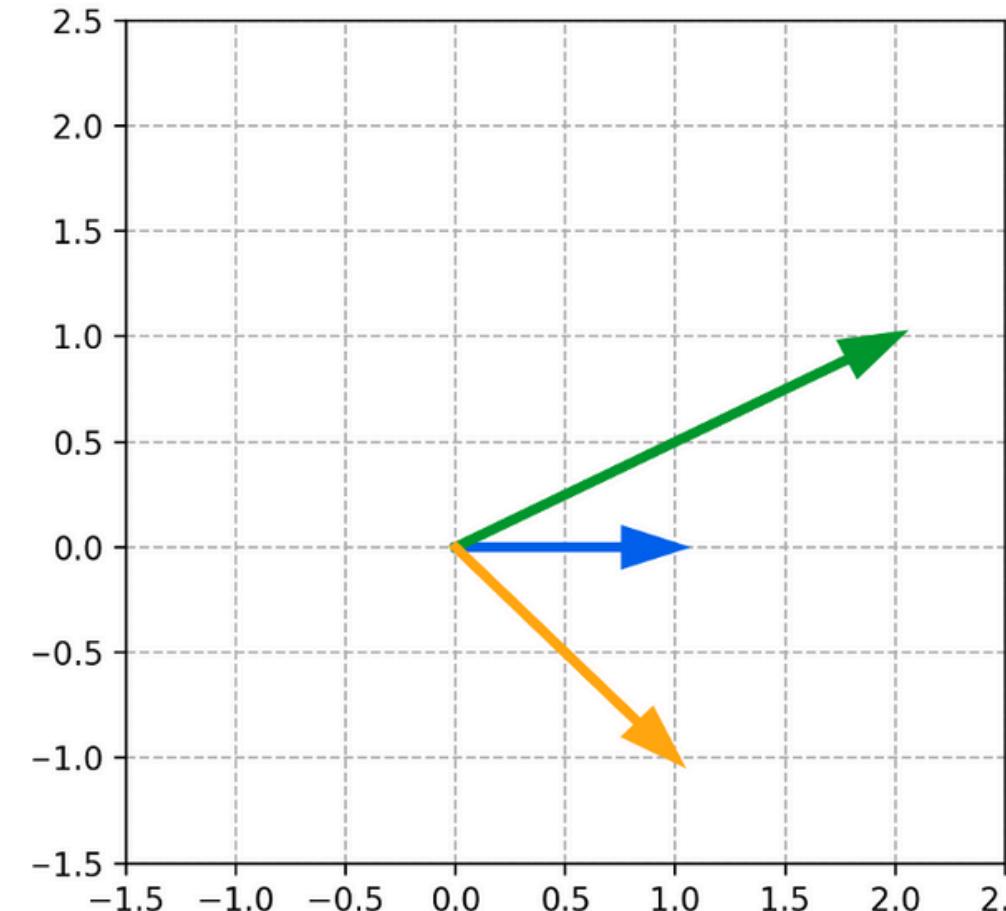
[Eickenberg 2016]

Data as Vector

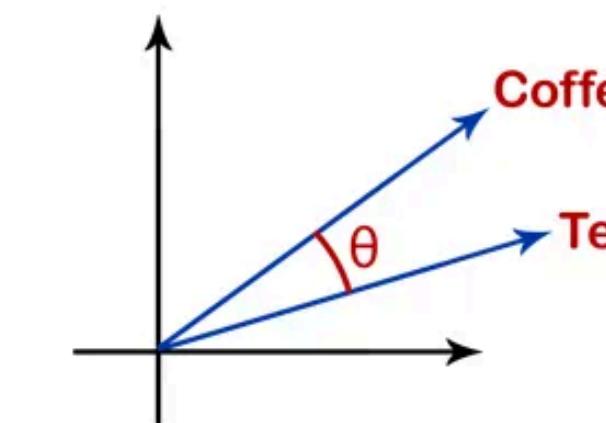
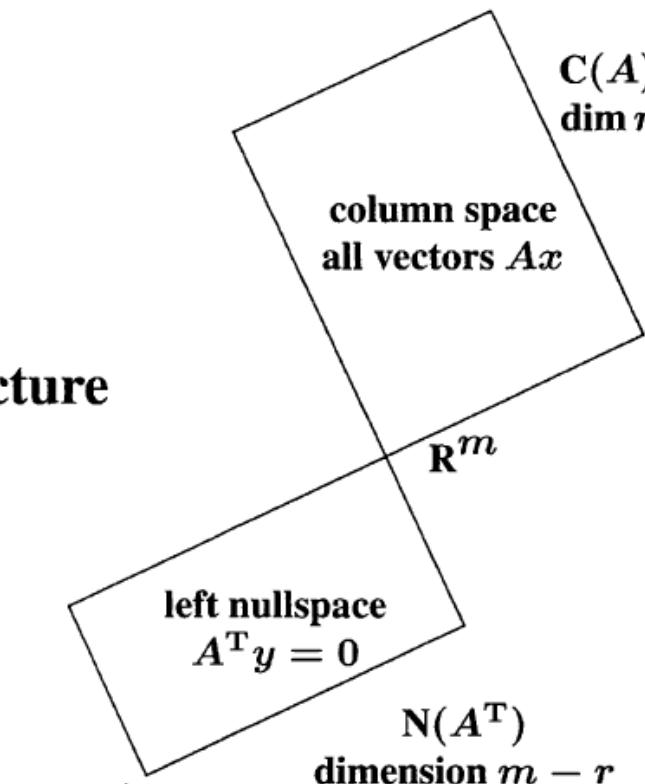
$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & -1 \end{bmatrix}$$

Row-Vector Space

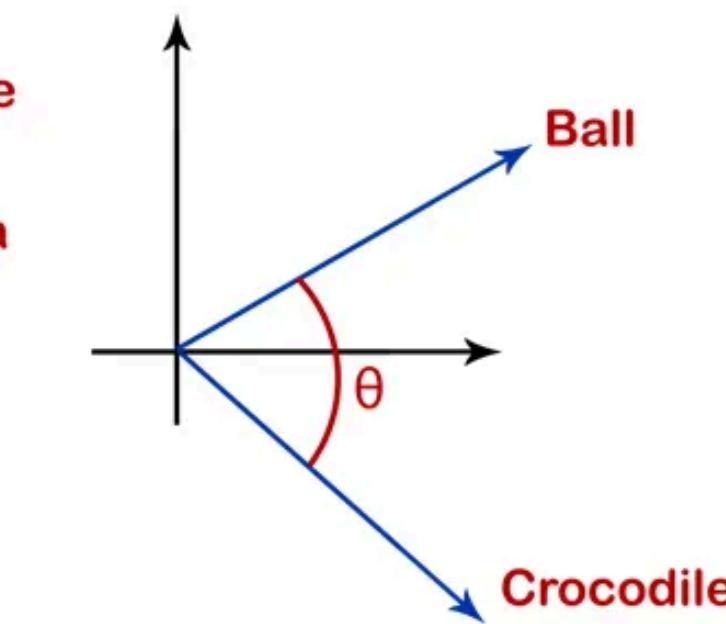
Dimension = 2



The big picture



$$\text{sim}(A, B) = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|}$$

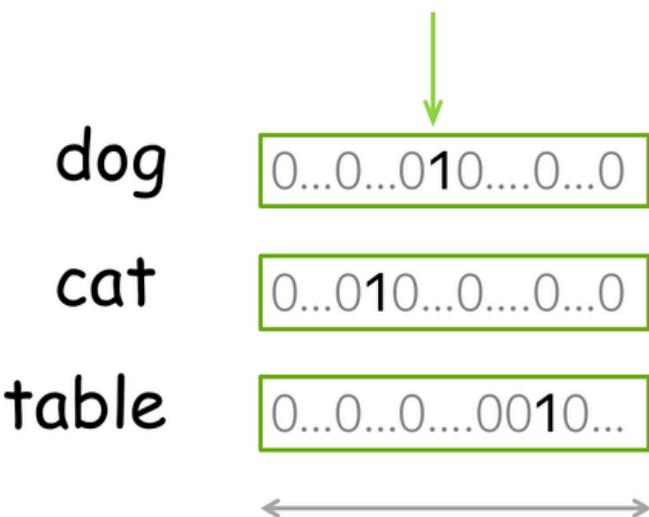


Cosine Similarity

Bag of Word

- ဝါကျရဲ့အဓိပ္ပာယ်ကို စကားလုံးတွေ တစ်လုံးချင်းစီ ဘယ်နှစ်ခါပါလဲဆိုတာကို ကြည့်ပြီး ဆုံးဖြတ်ခြင်း
- စကားလုံးများကို discrete symbols များအဖြစ်ဖော်ပြခြင်း
- ဆန့်ကျင်ဘက် စကားလုံးတွေ စကားလုံး စီစဉ်ပုံတွေ ဆင်တူတဲ့ စကားလုံးတွေကို ထည့်သွင်းစဉ်းစားမထားပါ

One is 1, the rest are 0



Embedding dimension =
vocabulary size

- စကားလုံးများကို one-hot vectors များဖြင့် ဖော်ပြနိုင်ပါတယ်။
- Vector dimension = ဝေါဟာရစကားလုံးအရေအတွက်

Distributional hypothesis

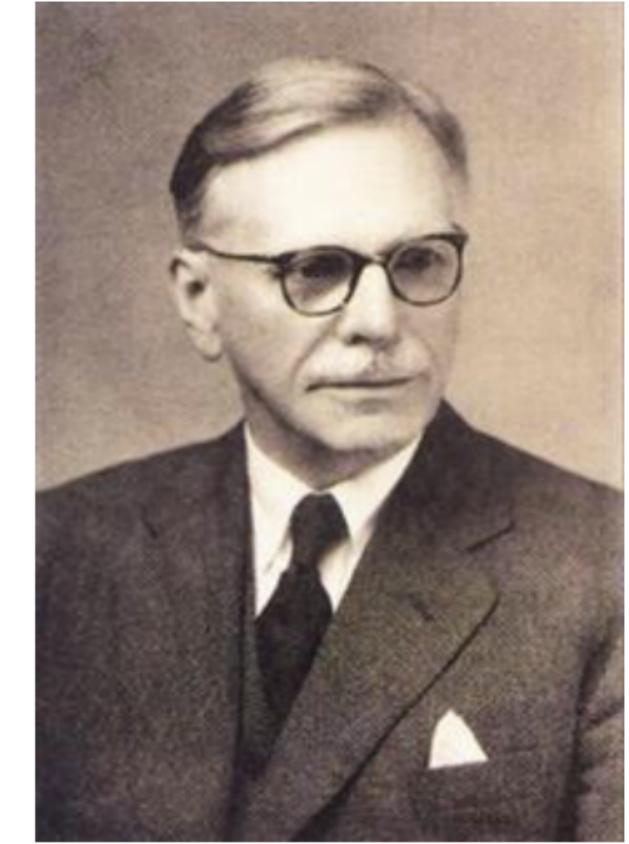
- စကားလုံးတွေဟာ ဆင်တူတဲ့ context တွေမှာ တွေ့ရတတ်ပြီး
- အခါတွေဟာ ဆင်တူတဲ့ အဓိပ္ပာယ်တွေ ရှိတတ်ကြပါတယ်
- You shall know a word by the company it keeps*
- အဓိပ္ပာယ်ဟာ စကားလုံးကိုယ်တိုင်မှာ မရှိဘဲ၊ စကားလုံးပေါ်လာတဲ့ context မှာပဲ ရှိပါတယ်။
- context ကို ကျယ်ကျယ်ပြန့်ပြန့် အနက်ဖွံ့ဖြိုးသင့်တယ်။
- Textual context
- Multimodal context



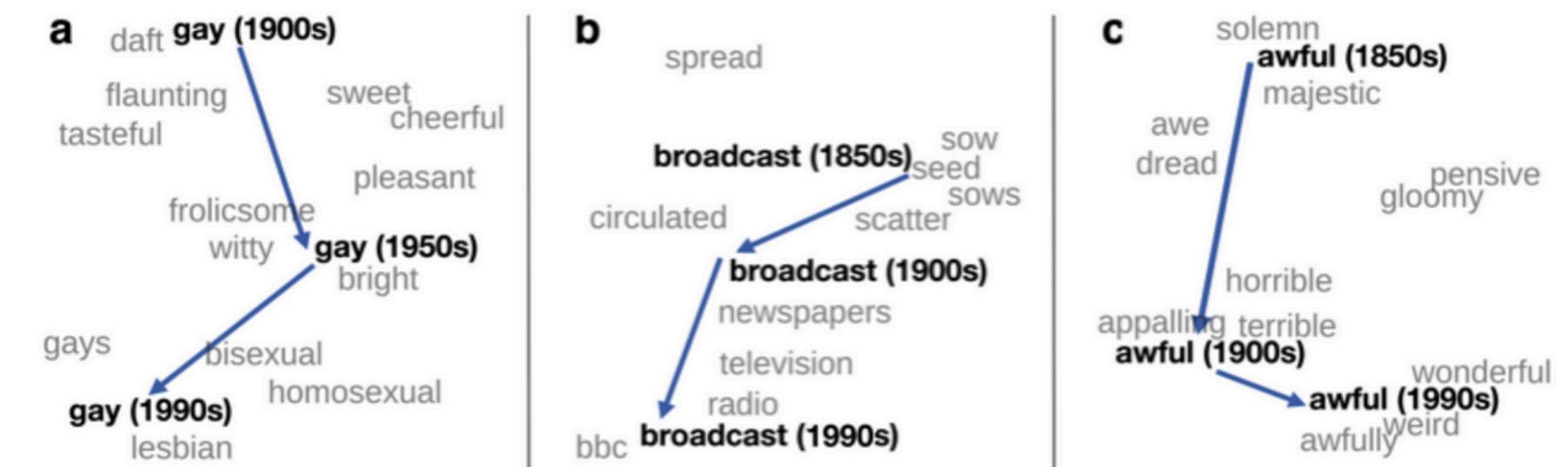
မြေစိမ်း



တောက်



J.R.Firth, 1957



Hamilton WL, Leskovec J, Jurafsky D. 2016. Diachronic word embeddings reveal statistical laws of semantic change, In Proc. of ACL

Word Vectors

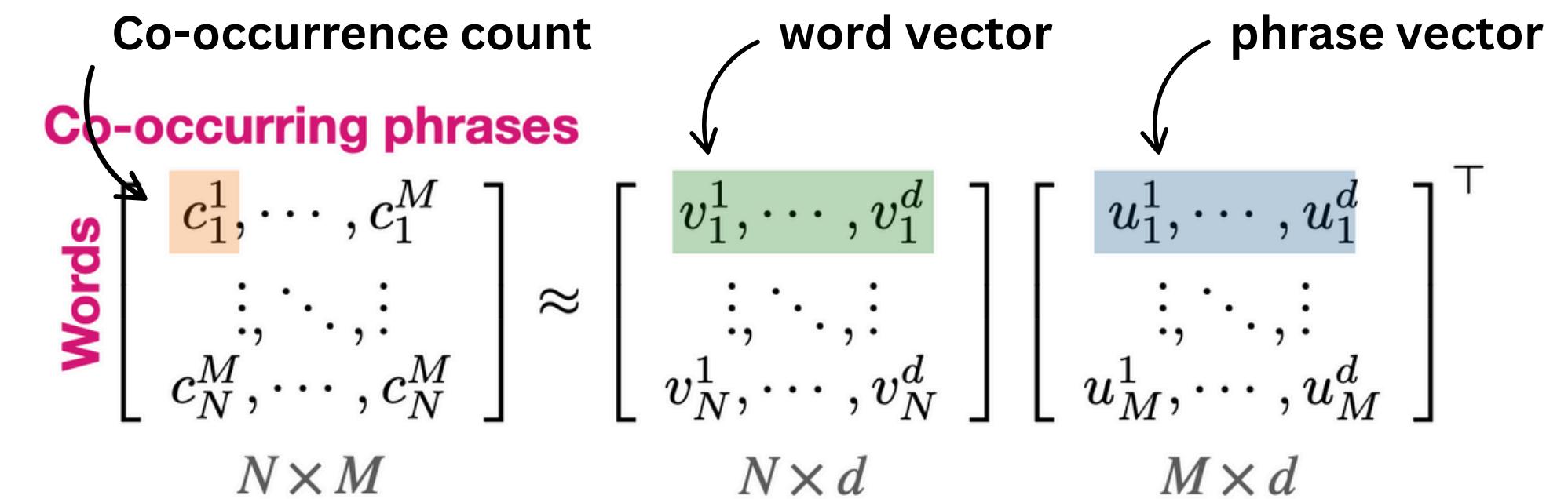
- Word vector ဆိုတာ word co-occurrence matrix ထဲက row တစ်ခုဖြစ်ပါတယ်
 - ဆင်တူတဲ့ word vectors တွေရှိတဲ့ စကားလုံးတွေဟာ အဓိပ္ပာယ်ဆင်တူကြပါတယ်
 - Word vectors တွေဟာ sparse နဲ့ high-dimensional ဖြစ်ကြပါတယ်
 - စကားလုံးအများစုနဲ့ စကားစုတွေဟာ အတူတူမရှိကြတဲ့ အတွက် zeros တွေ အများကြီးပါဝင်လိုပါ
 - Solution: **Word Vector Compression** → ဆင်တူတဲ့ စကားလုံးတွေကို တစ်ခုနဲ့တစ်ခု နှီးကပ်စေ
 - **Co-occurrence count** ကို word နဲ့ phrase vectors က ခန့်မှန်းမယ် [Mikolov et al., 2013]

Co-occurring phrases

	The	a	-	...	0
The	0	1	32	...	0
a	3	0	0
I	0	9
:	:	:	:	:	:
Gobbledygook	9	11	0	...	1

Words

Co-occurrence counts



Word Vectors (Cont'd)

- Word vectors တွေဟာ compositional ဖြစ်ပါတယ်
- Compositionality ကို vector arithmetics အပြစ် သတ်မှတ်နိုင်ပါတယ်
- Word vector arithmetics (e.g. Russia - Beijing + China = Moscow)



Hinton

Querying analogy for: လက်ရှည် - လက်တို့ + ဘေးသီရှည်
Analogies result:
Word: ဝတ်, Score: 0.5567076206207275
Word: ဖုန်းသီတေသန, Score: 0.5551380515098572
Word: အကျိုး, Score: 0.5383158326148987
Word: ထိုင်မသိမ်း, Score: 0.5366981625556946
Word: ထဘိ, Score: 0.5333731174468994
Word: အဝါး, Score: 0.5322700142860413
Word: ဉာဏ်ပိုင်တဲ့, Score: 0.5307892560958862
Word: ဝတ်ဆင်, Score: 0.5197224020957947
Word: အကျိုး, Score: 0.519253671169281
Word: ရော်ရောင်, Score: 0.5191075205802917

Vector for 'သီ':
[0.03071883 -0.25971985 0.40830743 0.11242703 -0.12839618 -0.00550127
0.05469747 -0.10745972 -0.15699431 0.3023142 -0.29259813 0.16690251
0.07785437 -0.58479047 -0.7412099 -0.04547881 -0.03287222 -0.11044414
-0.08638554 -0.02459145 -0.08973993 -0.02339625 -0.28159484 0.24005242
0.0358529 -0.6657557 0.27617294 -0.59123087 0.1455806 0.18249877
0.12091456 -0.4804478 -0.00294562 -0.56233424 0.278252 0.11192498
0.14295132 0.05967402 -0.12034112 0.05577437 -0.293318 -0.31646758
0.21822222 -0.40633217 0.17665516 0.2565288 -0.14461318 -0.08262646
0.4971917 0.3262872 0.09940107 0.18980542 -0.04641127 0.9501322
0.08271138 0.02752739 -0.17911749 -0.22868471 0.34966385 0.13285655
-0.06082508 -0.1977889 0.9427577 0.50436085 0.2823574 -0.19498166
0.1311327 -0.09055237 0.31270525 0.13989182 -0.84258276 0.30009115
-0.3908675 0.14674334 -0.4403673 -0.3262866 -0.26670656 0.24394107
0.18048933 -0.23953764 0.02176909 -0.29563728 0.10838298 0.12806904
0.09947888 0.0433741 -0.11379734 0.20521459 0.01840506 0.24994595
0.24415538 -0.19377986 -0.1143683 -0.30197594 0.12075205 -0.24896991
-0.11793793 0.49135238 0.0792672 0.16126719]

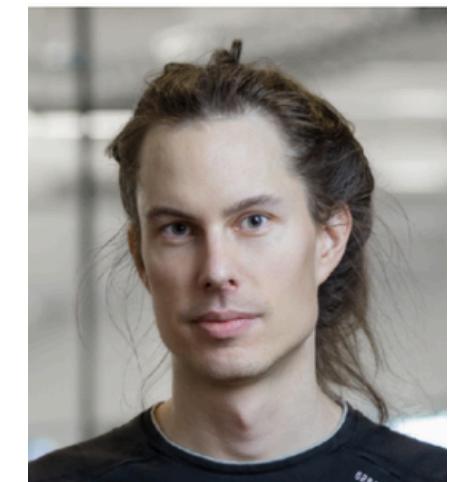
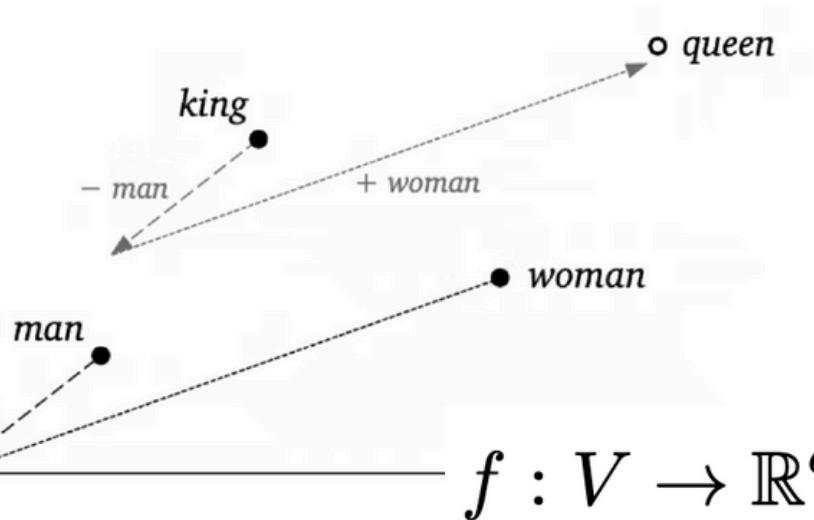
Nearest neighbors for 'အင်တာနှင်း':
Score: 0.697070837020874, Word: မြန်မာနှင်း
Score: 0.6930655837059021, Word: တာနှင်း
Score: 0.6491116881370544, Word: ဘေးနှင်း
Score: 0.6279654502868652, Word: အင်တာနာ
Score: 0.6255528926849365, Word: တာဝါတိုင်
Score: 0.609035849571228, Word: ကွန်ရက်လိုင်း
Score: 0.598670482635498, Word: ဖွံ့ဖြိုးတွေ
Score: 0.5967662334442139, Word: မှုဘိုင်းဖုန်း
Score: 0.5892794132232666, Word: ဒေဝါ
Score: 0.5861119627952576, Word: လူမှုကွန်ရက်

$$v_{\text{cat}} = \begin{pmatrix} -0.224 \\ 0.130 \\ -0.290 \\ 0.276 \end{pmatrix} \quad v_{\text{dog}} = \begin{pmatrix} -0.124 \\ 0.430 \\ -0.200 \\ 0.329 \end{pmatrix}$$

$$v_{\text{the}} = \begin{pmatrix} 0.234 \\ 0.266 \\ 0.239 \\ -0.199 \end{pmatrix} \quad v_{\text{language}} = \begin{pmatrix} 0.290 \\ -0.441 \\ 0.762 \\ 0.982 \end{pmatrix}$$



[Hinton, Geoffrey E., "Learning Distributed Representations of Concepts."](#) [Proceedings of the Eighth Annual Conference of the Cognitive Science Society, Erlbaum, 1986, pp. 1-12.](#)

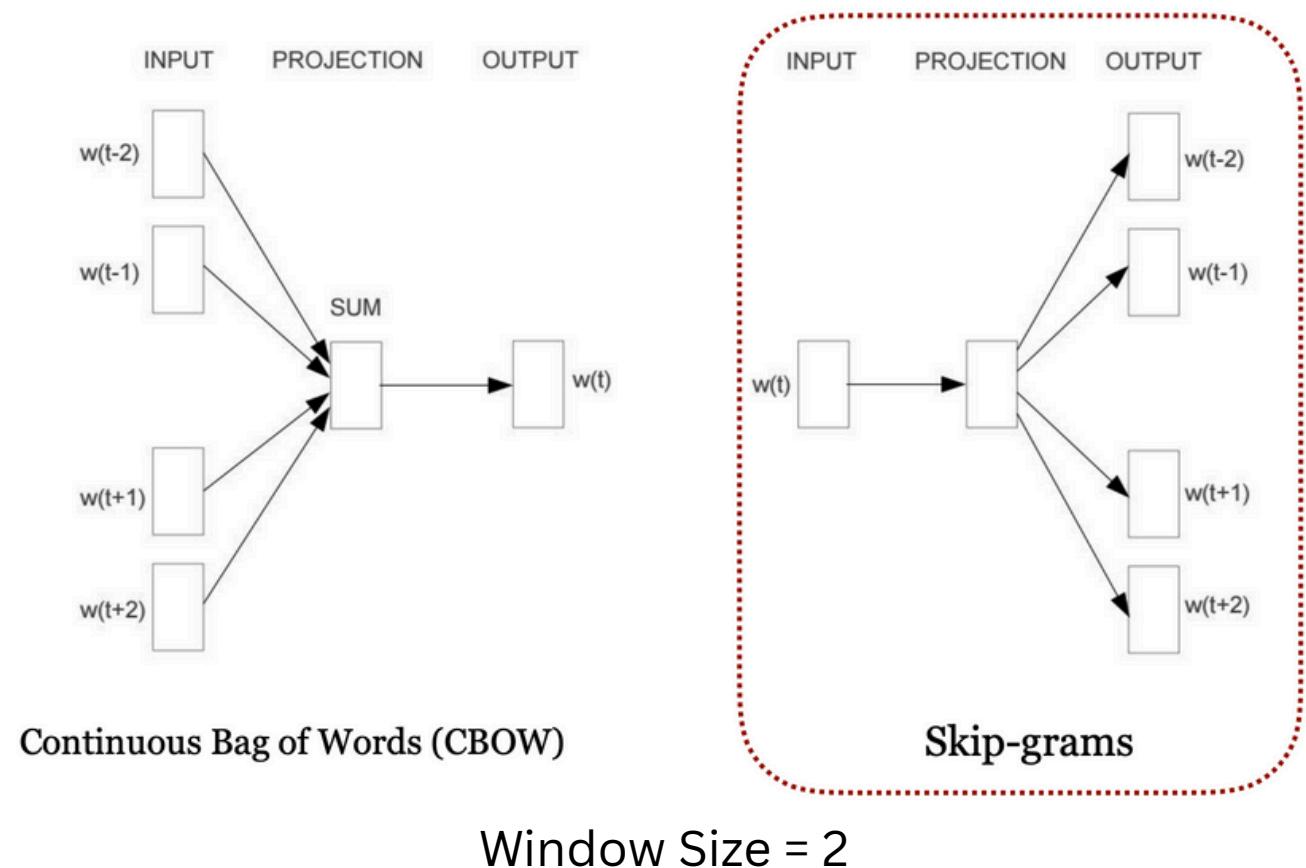


Mikolov

[Mikolov, T., Chen, K., Corrado, G., & Dean, J. \(2013\). Efficient Estimation of Word Representations in Vector Space.](#) In [1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings.](#)

Word Vectors (Cont'd)

- Unsupervised learning ကို နှစ်ခုစလုံးက အသုံးပြုထားတဲ့အတွက် human label တွေ မလိုအပ်ပါဘူး။
- CBOW: context စကားလုံးများပေါ်မှုတည်ပြီး target စကားလုံးကို ခန့်မှန်းပါတယ်။
- CBOW က embeddings တွေကို သင်ယူပြီး ဆင်တူတဲ့ context တွေကနေ စကားလုံးတစ်ခုတည်းကို ခန့်မှန်းနိုင်ပါတယ်။
- Skipgram- target စကားလုံးတစ်ခုပေးထားရင် context စကားလုံးတွေကို ခန့်မှန်းပါတယ်။ (CBOW ခဲ့ ပြောင်းပြန်)
- Skip-gram က target စကားလုံးတစ်ခုစိုးအပေါ်အရုံးစိုးကို အတွက် ရှားပါးတဲ့စကားလုံးတွေအတွက် ပိုကောင်းပါတယ်။
- Word2Vec, GloVe, fastText



context ဟာ စကားလုံး တစ်ခုချင်းစိရဲ့ အဓိပ္ပာယ်ကို
ဖြစ်ပေါ်စေပါတယ်

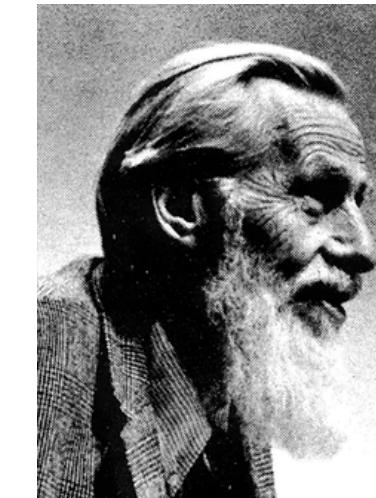
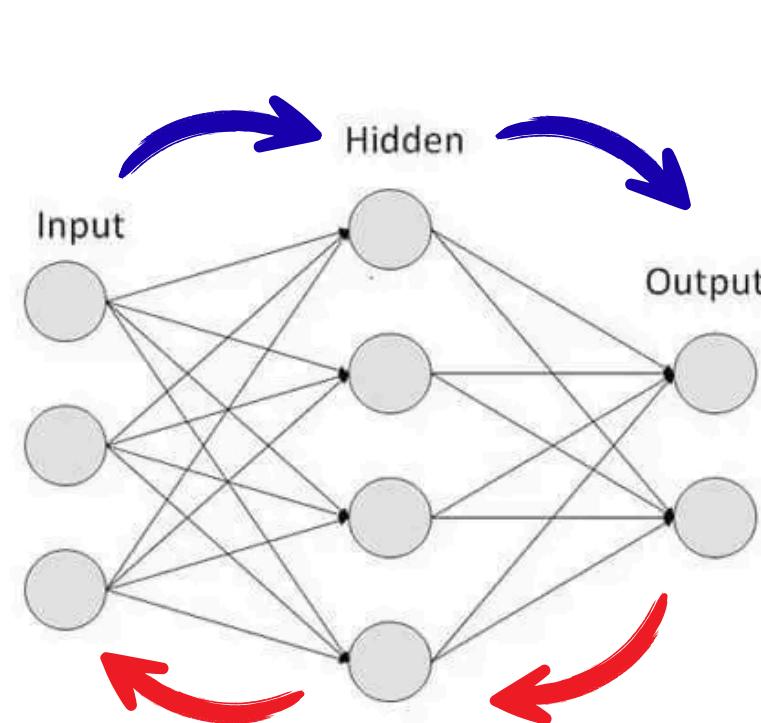
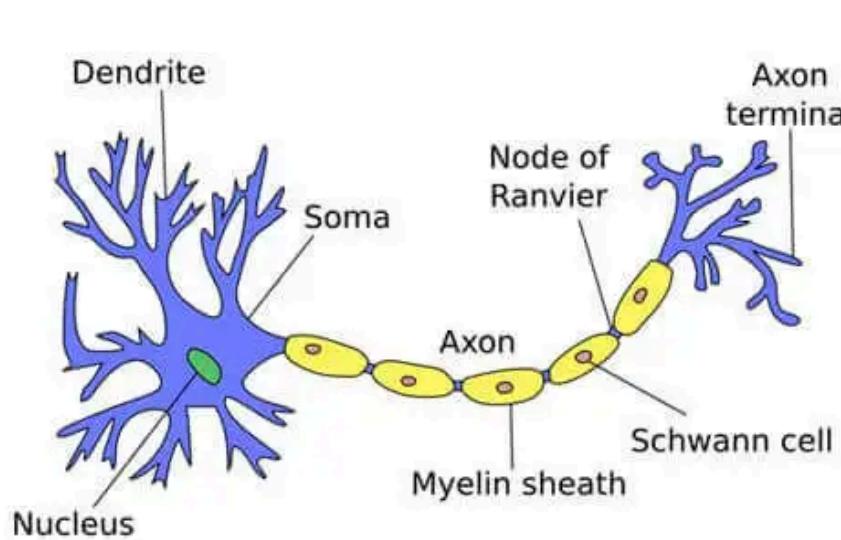
↑ ↓

စကားလုံး တစ်ခုချင်းစိနဲ့ ငြင်းတို့ရဲ့ အထားအသို့ က
စသားရဲ့ အဓိပ္ပာယ် (context) ကို ဖြစ်ပေါ်စေပါတယ်

စကားလုံးတစ်ခုဟာ သူရဲ့ ဘေးပတ်ဝန်းကျင်က စကားလုံးတွေပေါ်မှာ အဓိပ္ပာယ်သက်ရောက်ပါတယ်

Neural Networks

- ဦးနှောက်ကို အခြေခံပြီး၊ အချင်းချင်းချိတ်ဆက်ထားတဲ့ nodes (neurons) တွေရဲ့ layer တွေနဲ့ ဖွဲ့စည်းထားပါတယ်
- Input Layer – Train data တွေကို လက်ခံရယူပါတယ် (ဥပမာ- စာသား၊ ပုံရဲ့ pixels တွေ)
- Hidden Layers – data ကို weights နဲ့ activation functions တွေအသုံးပြုပြီး လုပ်ဆောင်ပါတယ်
- Output Layer – ခန့်မှန်းချက်တွေကို ထုတ်လုပ်ပေးပါတယ် (ဥပမာ- class labels, ကိန်းဂဏ်န်းတွေ)



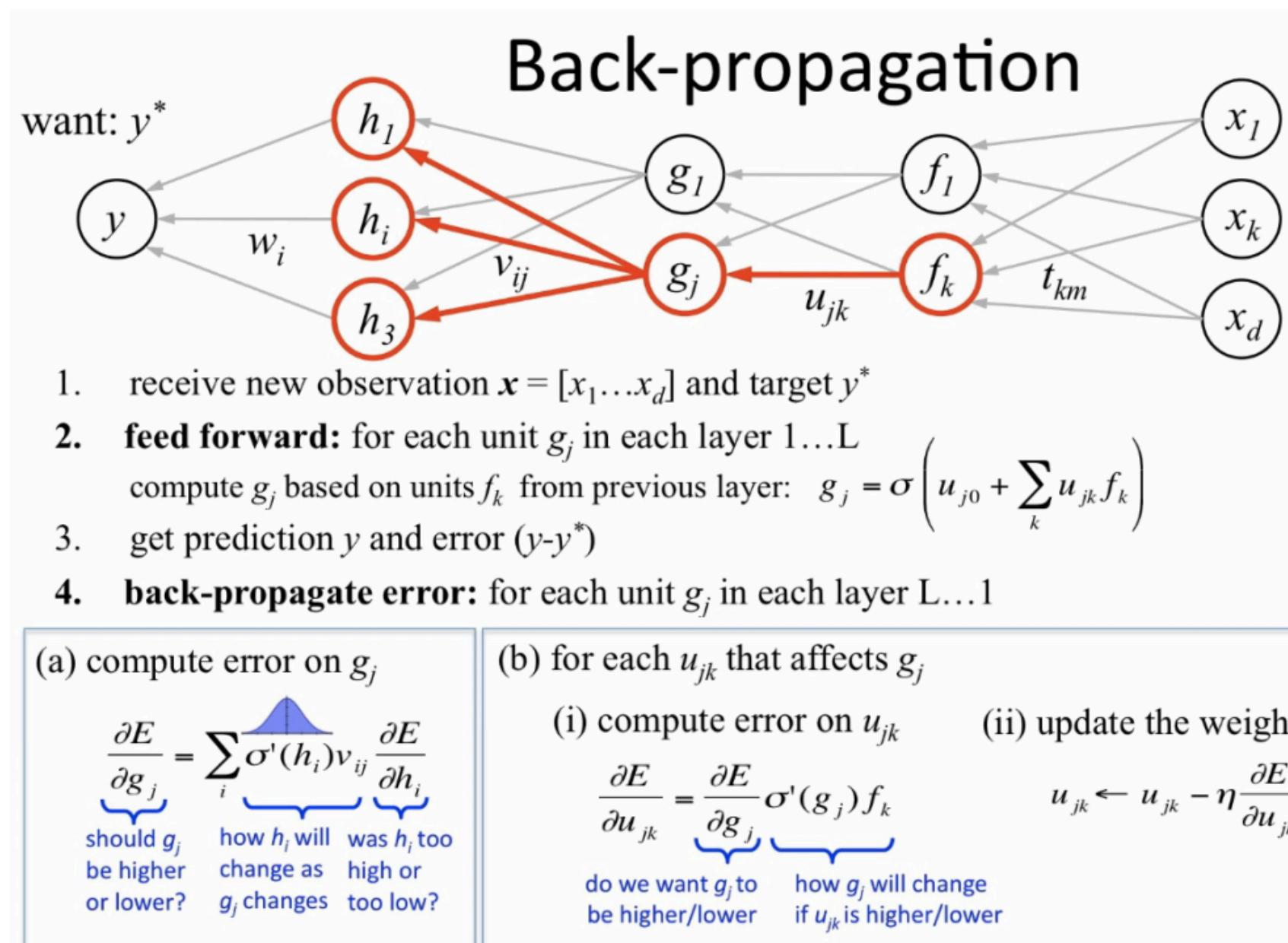
Warren McCulloch



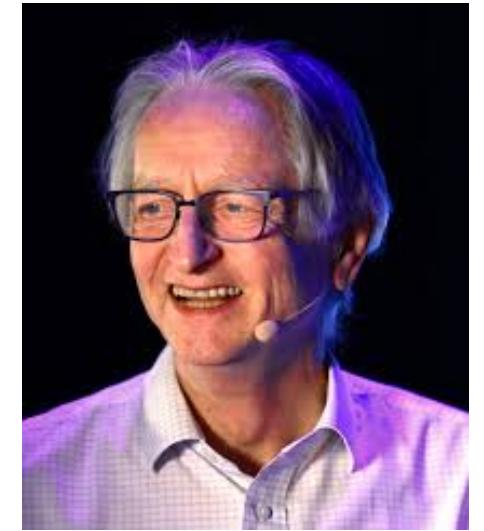
Walter Pitts

Neural Networks (Cont'd)

- Multilayer perceptron (MLP) ဆိုတာက အလွှာပေါင်းများစွာရှိတဲ့ neurons တွေနဲ့ဖွံ့စည်းထားတဲ့ neural network ဖြစ်။
- သူက feed-forward ပုံစံမျိုးနဲ့စီစဉ်ထားပြီး အလွှာတွေကြားမှာ nonlinear activation functions တွေ ပါဝင်ပါတယ်။



**Paul
Werbos**



**Geoffery
Hinton**

Werbos, Paul. (1990).
Backpropagation through time: what it does and how to do it. Proceedings of the IEEE.
78. 1550 - 1560.
10.1109/5.58337.

Rumelhart, D., Hinton, G. & Williams, R. Learning representations by back-propagating errors. Nature 323, 533–536 (1986).
<https://doi.org/10.1038/323533a0>

Neural Networks (Cont'd)

$$z_1 = (r_1 + b_1) = (\sum_{i=1}^3 a_i w_i + b_1)$$

$\mathbf{X} = (10, 3)$

	f1	f2	f3
Item 1	a ₁	a ₂	a ₃
Item 2			
Item 3			
Item 4			
Item 5			
Item 6			
Item 7			
Item 8			
Item 9			
Item 10			

$$\mathbf{O} = \mathbf{XW}^T + \mathbf{b}$$

$W^T = (3, 5)$

	n1	n2	n3	n4	n5
w ₁					
w ₂					
w ₃					

$$\mathbf{b} = (1, 5)$$

$$\mathbf{XW}^T = (10, 5)$$

b				
1				

+

r ₁				

$O = (10, 5)$

	f1	f2	f3	f4	f5
Item 1	z ₁				
Item 2					
Item 3					
Item 4					
Item 5					
Item 6					
Item 7					
Item 8					
Item 9					
Item 10					

Neural Networks (Cont'd)

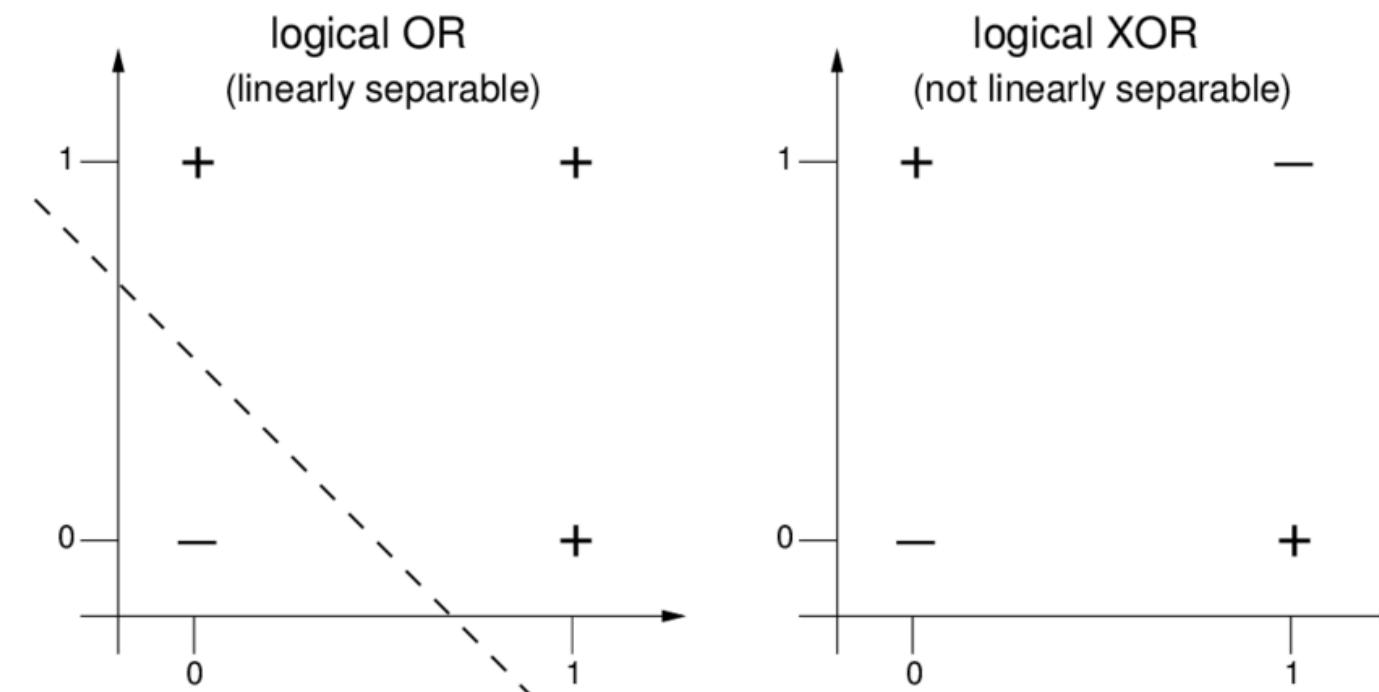
- Linear layer တစ်ခုစီ ပြီးတိုင်း၊ ကျွန်တော်တိုက nonlinear activation function တစ်ခုကို ပုံမှန်အားဖြင့် အသုံးပြုကြတယ်။
- activation functions တွေကို အသုံးမပြုဘူးဆိုရင်၊ output ဟာ inputs တွေရဲ့ linear combination တစ်ခုပဲဖြစ်နေမှာပါ။
- ဆိုလိုတာကတွေ၊ MLP ဟာ input နဲ့ output ကြားက non-linear mapping တွေကို လွှဲလာနိုင်တွေမှာ မဟုတ်ပါဘူး။
- အဲဒီ non-linear mapping တွေကပဲ လက်တွေကမ္မာက data အများစုကို ကိုယ်စားပြုတာ ဖြစ်ပါတယ်။

$$O_1 = xW_1^T + b_1$$

$$O_2 = (O_1)W_2^T + b_2$$

$$O_2 = (xW_1^T + b_1)W_2^T + b_2$$

$$O_2 = xW_1^T W_2^T + b_1 W_2^T + b_2$$

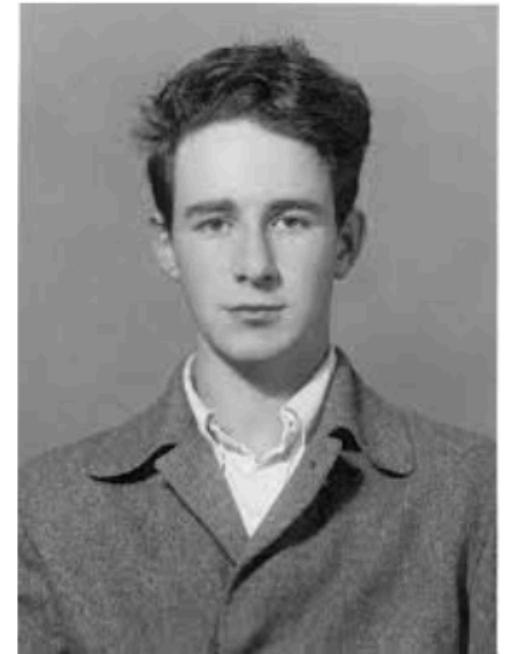
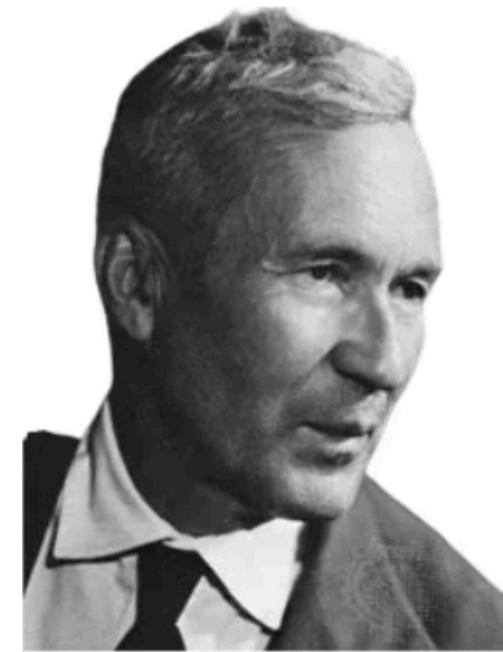


Input 1	Input 2	Output
1	1	1
1	0	1
0	1	1
0	0	0

Input 1	Input 2	Output
1	1	0
1	0	1
0	1	1
0	0	0

Kolmogorov-Arnold Representation Theorem

The works of Vladimir Arnold and Andrey Kolmogorov established that if f is a multivariate continuous function, then f **can be expressed** as a finite **composition** of continuous functions of a single variable and the **binary operation** of **addition**.



$$f(\mathbf{x}) = f(x_1, \dots, x_n) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^n \phi_{q,p}(x_p) \right),$$

Andrey Kolmogorov

Vladimir Arnold

where $\phi_{q,p}: [0, 1] \rightarrow \mathbb{R}$ and $\Phi_q: \mathbb{R} \rightarrow \mathbb{R}$.

Simplifying the function

$$f(\mathbf{x}) = f(x_1, \dots, x_n) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^n \phi_{q,p}(x_p) \right),$$

$$f(x_1, x_2) = \Phi_1(t_1) + \Phi_2(t_2) + \Phi_3(t_3) + \Phi_4(t_4) + \Phi_5(t_5)$$

$$f(x_1, x_2, x_3) = \Phi_1(t_1) + \Phi_2(t_2) + \Phi_3(t_3) + \Phi_4(t_4) + \Phi_5(t_5) + \Phi_6(t_6) + \Phi_7(t_7)$$

$$f(x_1, x_2, \dots, x_n) = \Phi_1(t_1) + \Phi_2(t_2) + \dots + \Phi_{2n+1}(t_{2n+1})$$

n variables $2n + 1$ terms

$$xy = -\frac{x^2}{2} - \frac{y^2}{2} + \frac{(x+y)^2}{2}$$
$$f(x) + g(y) + h(z)$$

function of x function of y function of z

Simplifying the function (Cont'd)

Which one is simpler? - Separable and Non-Separable

- $x + y$ **vs** $x.y$
- $x^2 + \sin(y)$ **vs** xy
- $x^2 + \sin(y)$ **vs** $x + \sin(xy)$
- $x + y$ **vs** $\cos(x + y)$
- $2x^2 + y^2 + \ln(z)$ **vs** $(x^2).y + z$

Sum and Difference Rule

$$\frac{d}{dx}[f(x) \pm g(x)] = \frac{d}{dx}[f(x)] \pm \frac{d}{dx}[g(x)]$$

Product Rule

$$\frac{d}{dx}[f(x)g(x)] = f(x)g'(x) + f'(x)g(x)$$

Warmup: Summing is simpler than Multiplying in mathematics.

Kolmogorov-Arnold Network

KAN: If f is a multivariate continuous function, then f can be **approximated** as a finite **composition** of continuous functions of a single variable and the **binary operation** of addition.

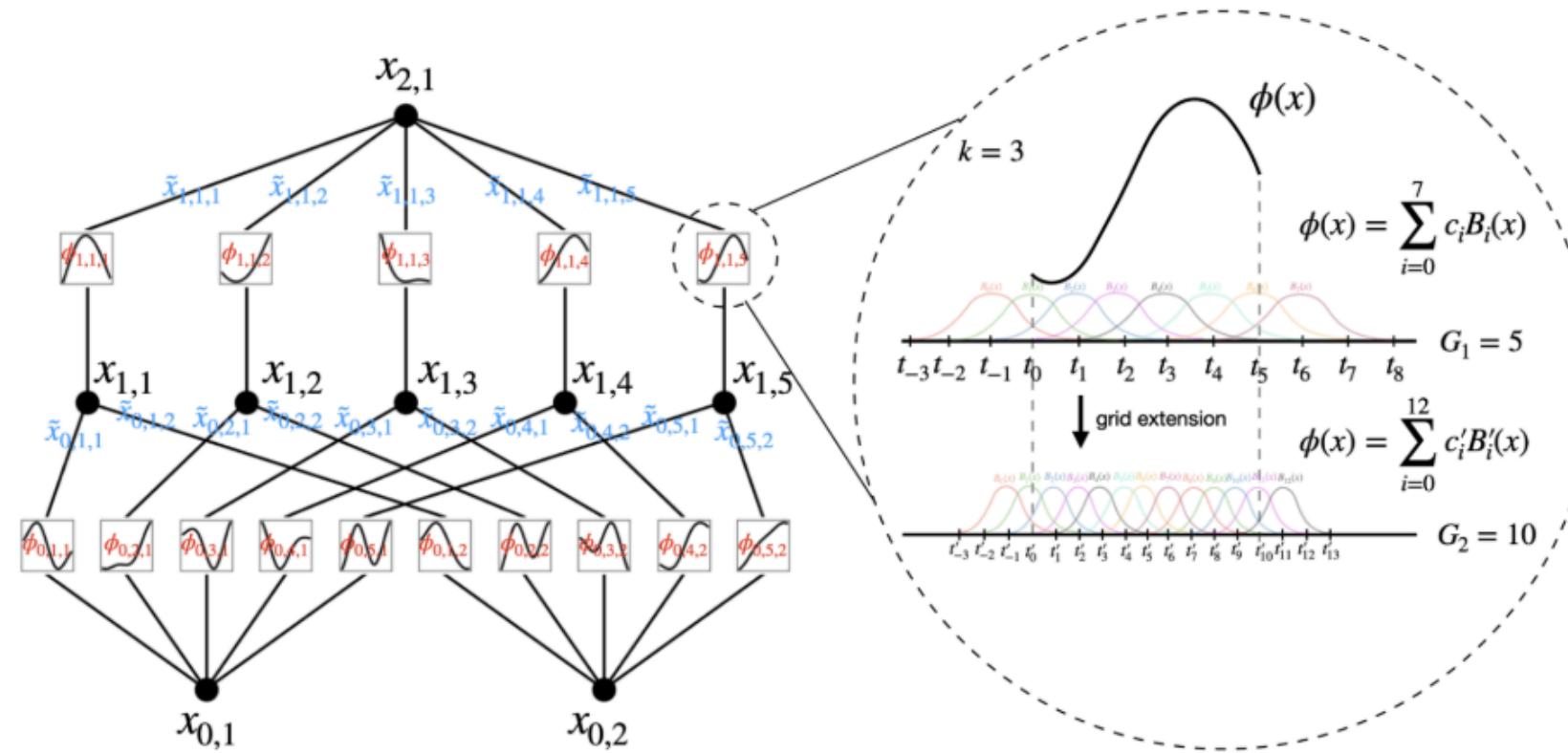


Figure 2.2: Left: Notations of activations that flow through the network. Right: an activation function is parameterized as a B-spline, which allows switching between coarse-grained and fine-grained grids.



Ziming Liu



Max Tegmark

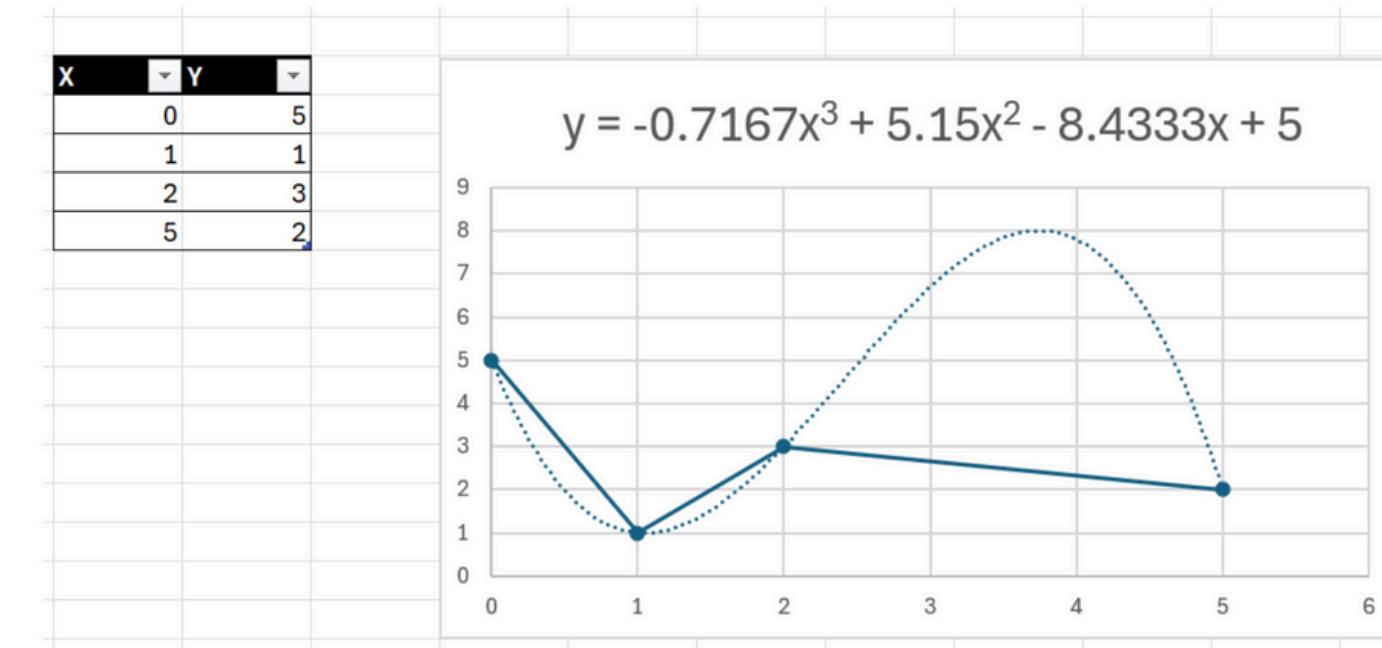
Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljacic, M., Hou, T., & Tegmark, M. (2025). KAN: Kolmogorov–Arnold Networks. In Y. Yue, A. Garg, N. Peng, F. Sha, & R. Yu (Eds.), *International Conference on Representation Learning* (pp. 70367–70413).

Data Fitting

- smooth curve ရဲ့ equation ကို ဘယ်လိုရာမလဲ?
- နည်းလမ်းတစ်ခုကတော့ polynomial curve ရဲ့ generic equation ကို ရေးပြီး၊ အဲဒီ curve ကို series of points တွေ အားလုံးကနေ ဖြတ်သွားအောင် လုပ်လိုက်တာပါ။ အဲလိုလုပ်လိုက်ရင် equation ရဲ့ coefficients တွေကို ရလာမှာပါ။
- ကျွန်တော်တို့မှာ points ငါ ခုရှိတော့ equations ငါ ခုပါတဲ့ system of equations တစ်ခုကို ဖန်တီးနိုင်ပါတယ်။
- ဆိုလိုတာက variables ငါ ခုအတွက် ဖြေရှင်းနိုင်ပြီး၊ degree 3 ရှိတဲ့ polynomial ကို ရှုံးမှာ ဖြစ်ပါတယ်။

$$y = ax^3 + bx^2 + cx + d$$

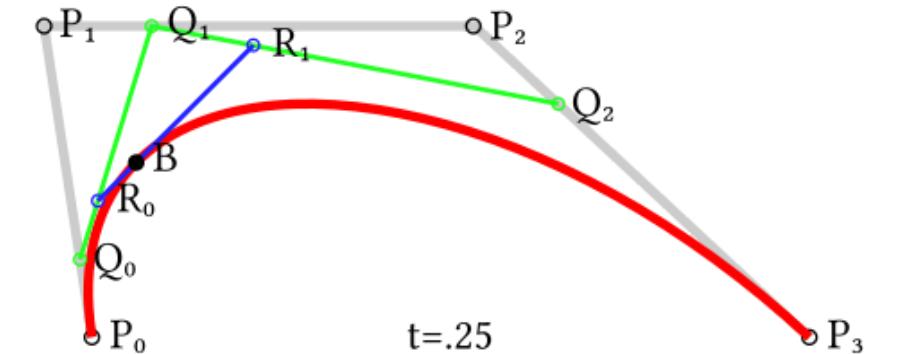
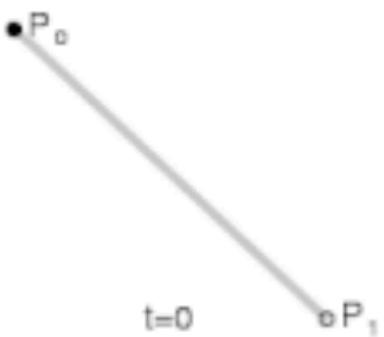
$$\begin{aligned} 5 &= a(0)^3 + b(0)^2 + c(0) + d \\ 1 &= a(1)^3 + b(1)^2 + c(1) + d \\ 3 &= a(2)^3 + b(2)^2 + c(2) + d \\ 2 &= a(5)^3 + b(5)^2 + c(5) + d \end{aligned}$$



Bézier curves

- Points የጤና ስምምነት አይነት በለንድ?
- Solution: Beizer Curve

$$\mathbf{B}(t) = \mathbf{P}_0 + t(\mathbf{P}_1 - \mathbf{P}_0) = (1 - t)\mathbf{P}_0 + t\mathbf{P}_1$$

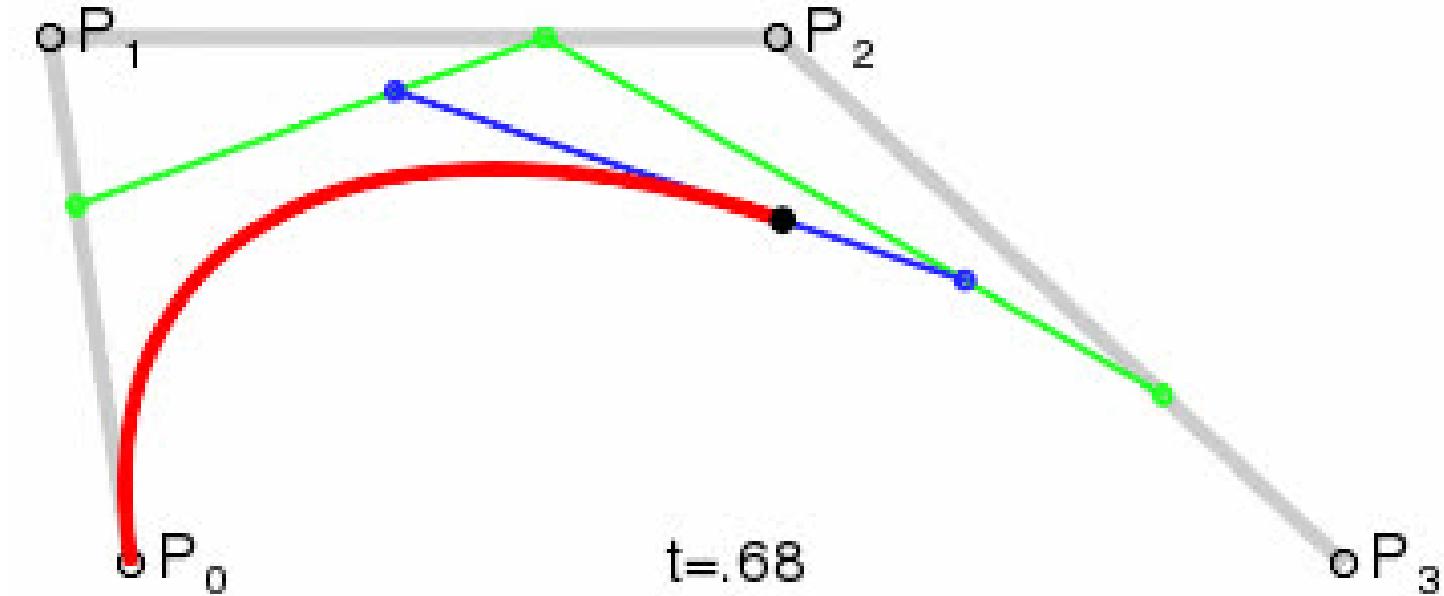


Pierre Bézier

$$\mathbf{Q}_0(t) = (1 - t)\mathbf{P}_0 + t\mathbf{P}_1$$

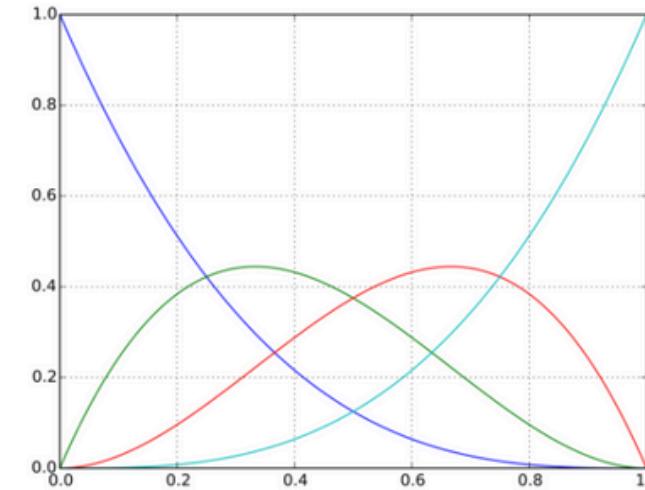
$$\mathbf{Q}_1(t) = (1 - t)\mathbf{P}_1 + t\mathbf{P}_2$$

$$\begin{aligned}\mathbf{B}(t) &= (1 - t)\mathbf{Q}_0 + t\mathbf{Q}_1 \\ &= (1 - t)[(1 - t)\mathbf{P}_0 + t\mathbf{P}_1] + t[(1 - t)\mathbf{P}_1 + t\mathbf{P}_2] \\ &= (1 - t)^2\mathbf{P}_0 + 2(1 - t)t\mathbf{P}_1 + t^2\mathbf{P}_2\end{aligned}$$



Bézier curves: going deeper

points $n + 1$ ခု ရှိတယ်ဆိုရင်၊ အဲဒီ points တွေကို အသုံးပြုပြီး n degree Bézier curve ကို အောက်ပါ formula နဲ့ရာဖွေနိုင်ပါတယ်



Blue: $b_{0,3}(t)$

Green: $b_{1,3}(t)$

Red: $b_{2,3}(t)$

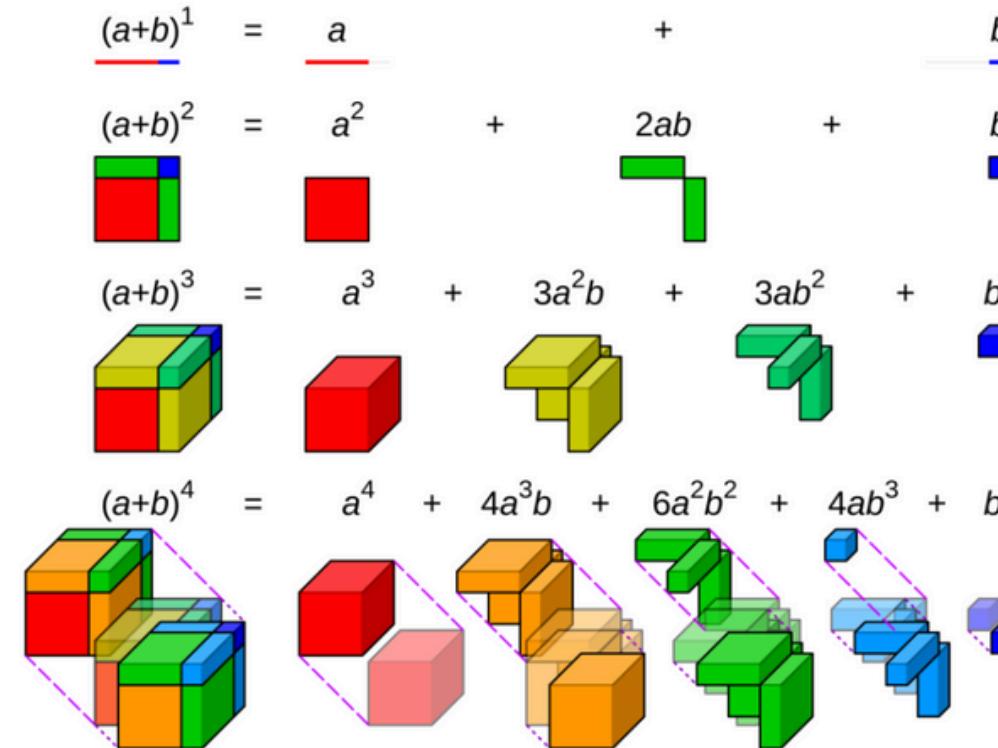
Cyan: $b_{3,3}(t)$

Binomial coefficients

$$\binom{n}{i} = \frac{n!}{i! (n - i)!}$$

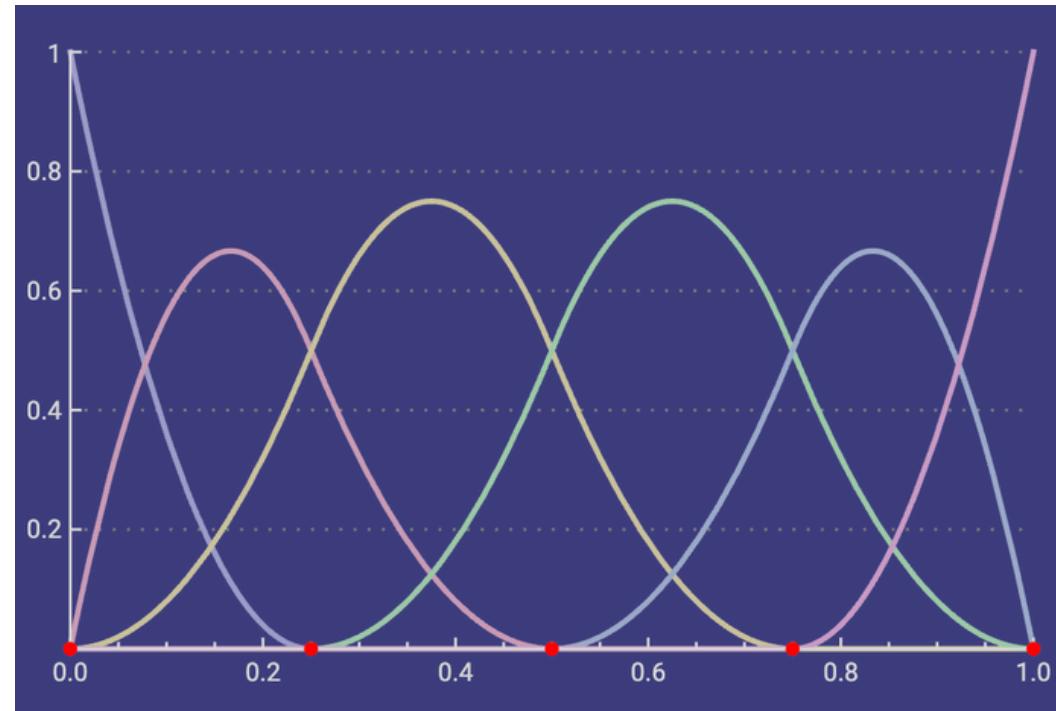
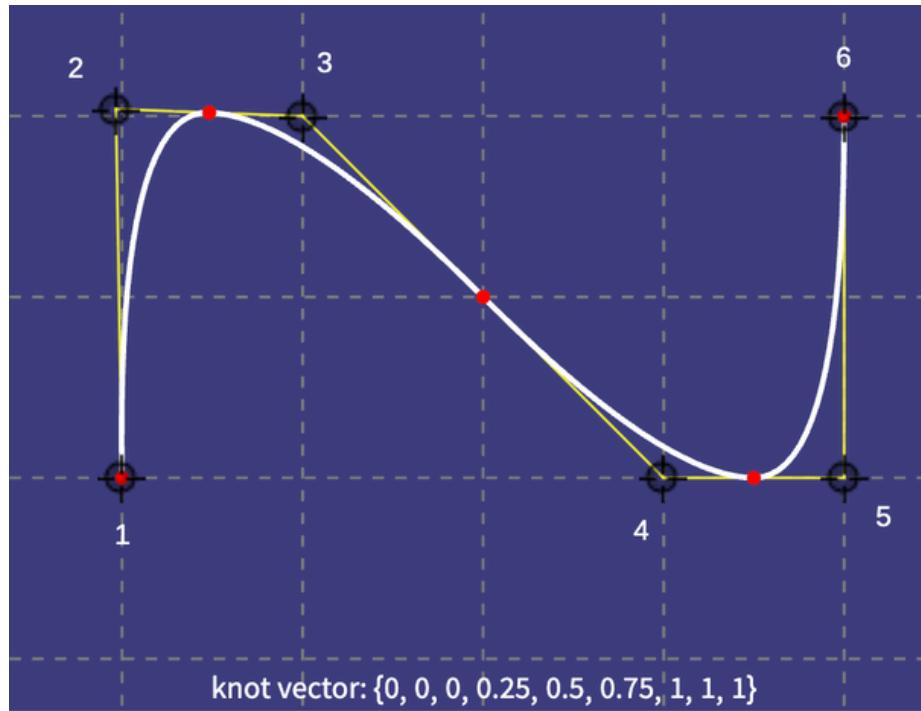
$$B(t) = \sum_{i=0}^n \binom{n}{i} (1-t)^{n-i} t^i \mathbf{P}_i = \sum_{i=0}^n b_{i,n}(t) \mathbf{P}_i$$

Bernstein basis polynomials



From Bézier curves to B-Splines

- Points (ဥပမာ n ခု) အများကြီး ရှိတယ်ဆိုရင်၊ အဲဒါကို ကောင်းကောင်း approximate လုပ်နိုင်ဖို့ degree $n-1$ ရှိတဲ့ Bézier curve တစ်ခု လိုအပ်ပါတယ်။
- ဒါပေမဲ့ အဲဒီလို curve ကို computationally တွက်ချက်ဖို့က အတော်လေး ရှုပ်ထွေးနိုင်ပါတယ်။
- points အားလုံးကို interpolate လုပ်တဲ့ Bézier curve ကြီးတစ်ခုတည်းကို သုံးမယ့်အစား၊ ဒီ points အားလုံးကြားမှာ Bézier curves အများကြီးကို ဆက်ပေးလိုက်ရင် မကောင်းဘူးလား?
- B-Spline Local Control: Control point တစ်ခုကို ရွှေ့လိုက်တာက အဲဒီ control point ရဲ့ proximity (အနီးအနား) မှာရှိတဲ့ curve ကို locally ပြောင်းလဲစေပြီး၊ သေးနားကပ်လျက်ရှိတဲ့ Bezier curves တွေကိုတော့ ပြောင်းလဲခြင်းမရှိဘဲ ကျွန်ုပ်ရှိစေပါတယ်။

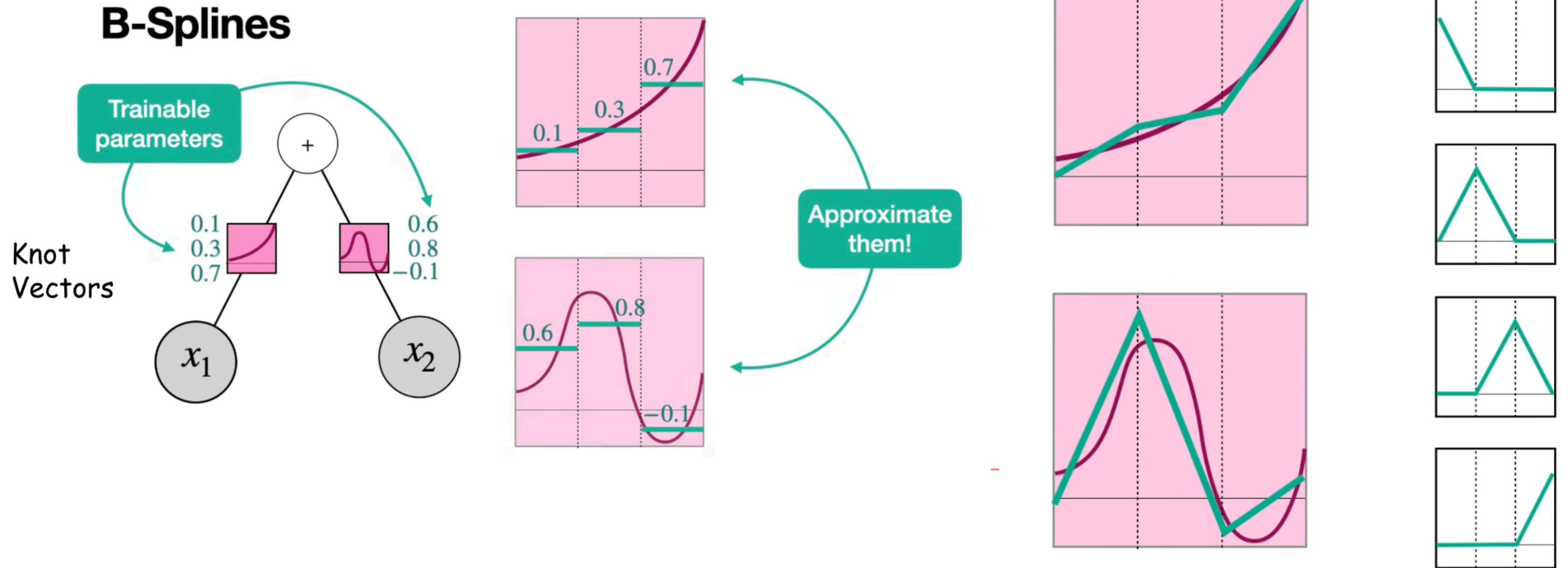


<https://demonstrations.wolfram.com/BSplineCurveWithKnots/>

Umar Jamil – <https://github.com/hkproj/kan-notes>

Nikolaï Ivanovich
Lobachevskiĭ

Approximation with B-Splines

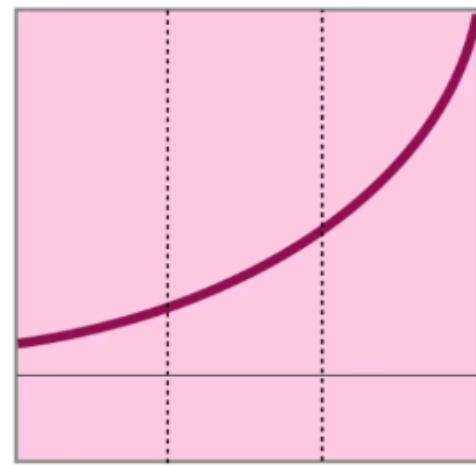


<https://demonstrations.wolfram.com/BSplineCurveWithKnots/>

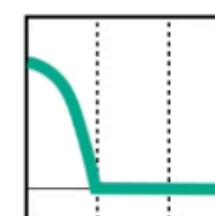
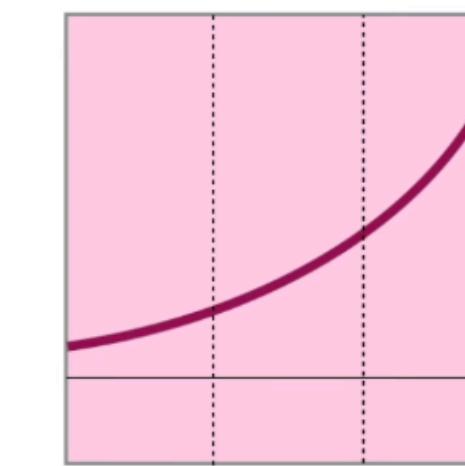
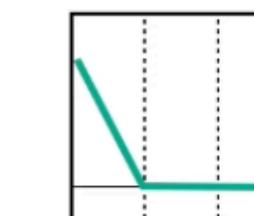
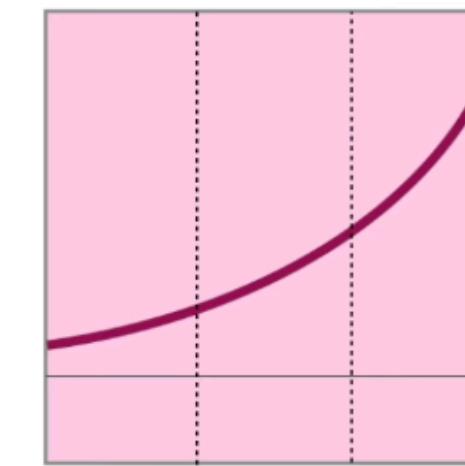
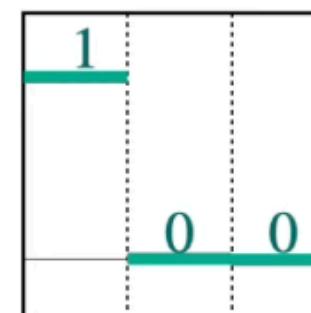
Dr. Luis Serrano's YouTube channel

Approximation with B-Splines (Cont'd)

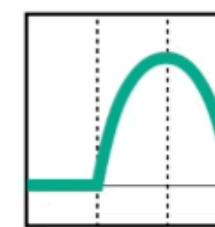
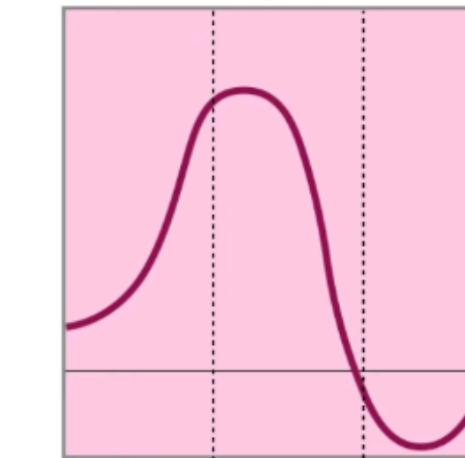
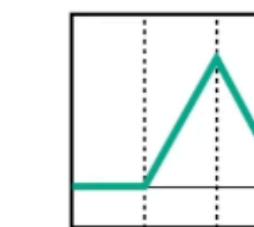
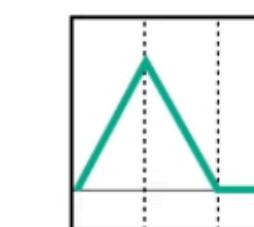
Constant



Linear



Quadratic



"B" stands for "basis"—B-splines form the building blocks to represent more complex curves.

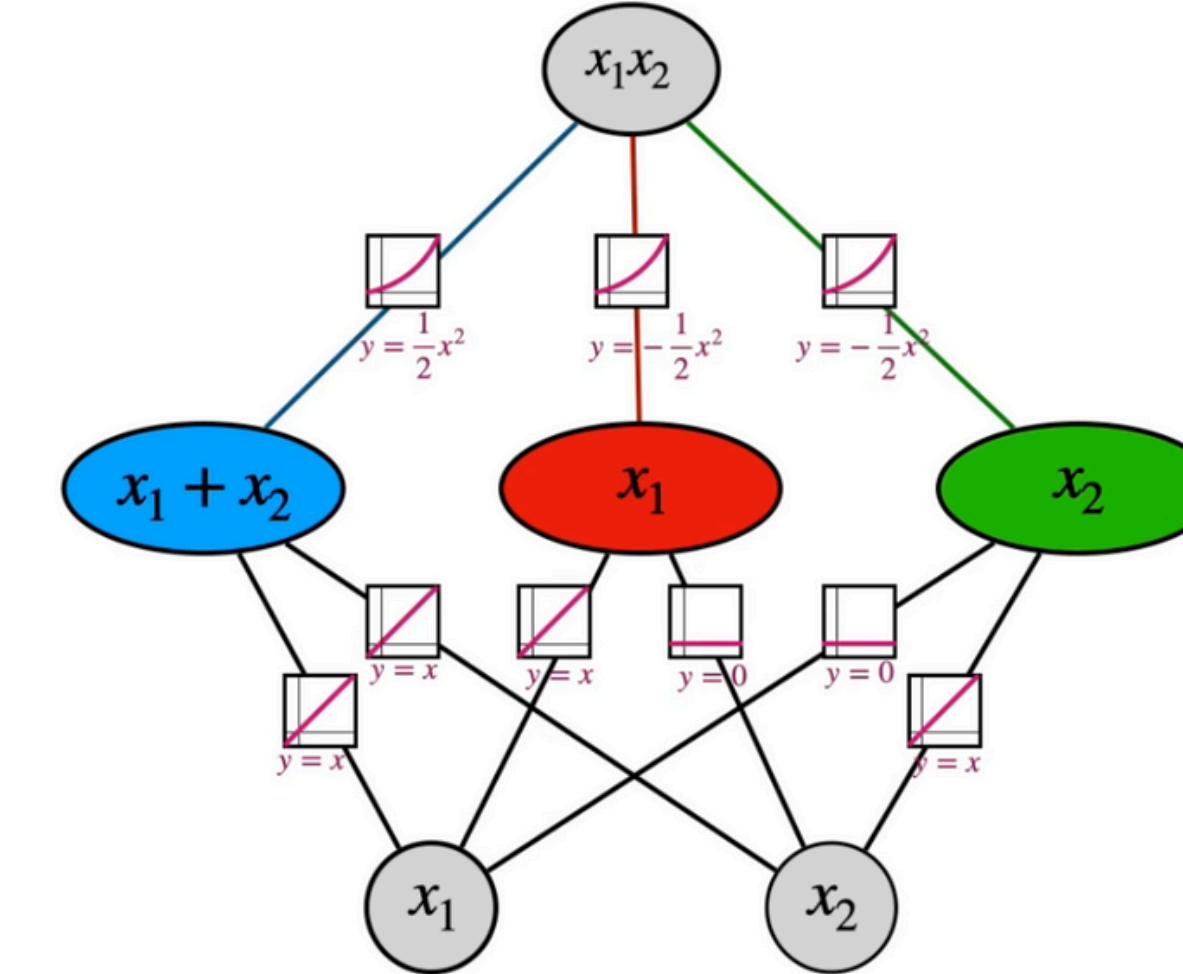
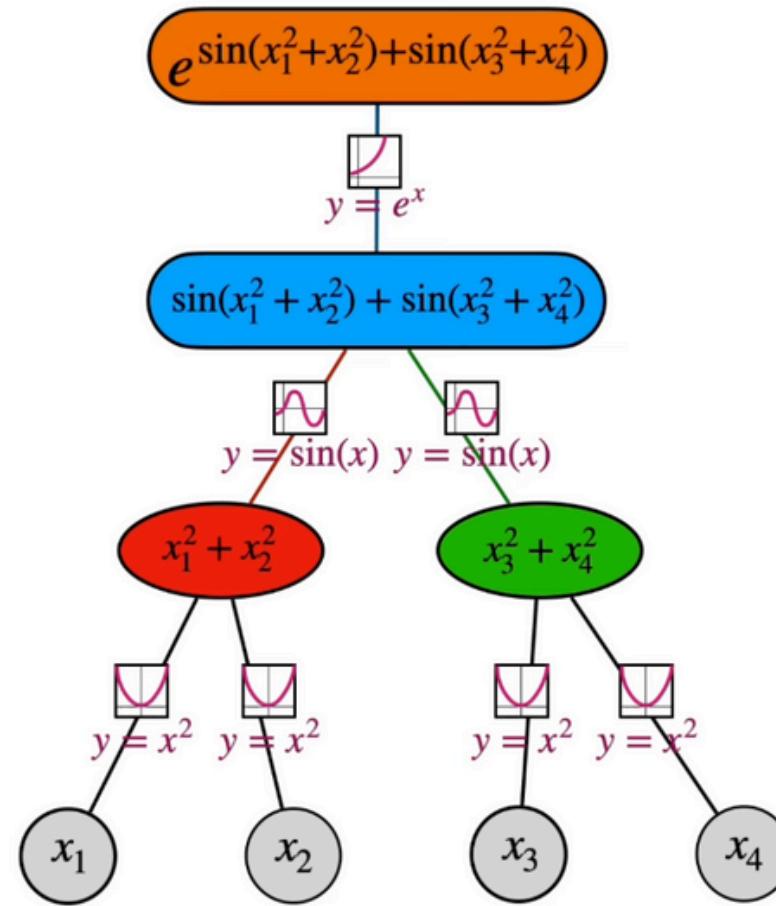
Dr. Luis Serrano's YouTube channel

<https://demonstrations.wolfram.com/BSplineCurveWithKnots/>

Approximation with B-Splines (Cont'd)

$$x_1 x_2 = \frac{1}{2}(x_1 + x_2)^2 - \frac{1}{2}x_1^2 - \frac{1}{2}x_2^2$$

$$e^{\sin(x_1^2 + x_2^2) + \sin(x_3^2 + x_4^2)}$$



Dr. Luis Serrano's YouTube channel

<https://demonstrations.wolfram.com/BSplineCurveWithKnots/>

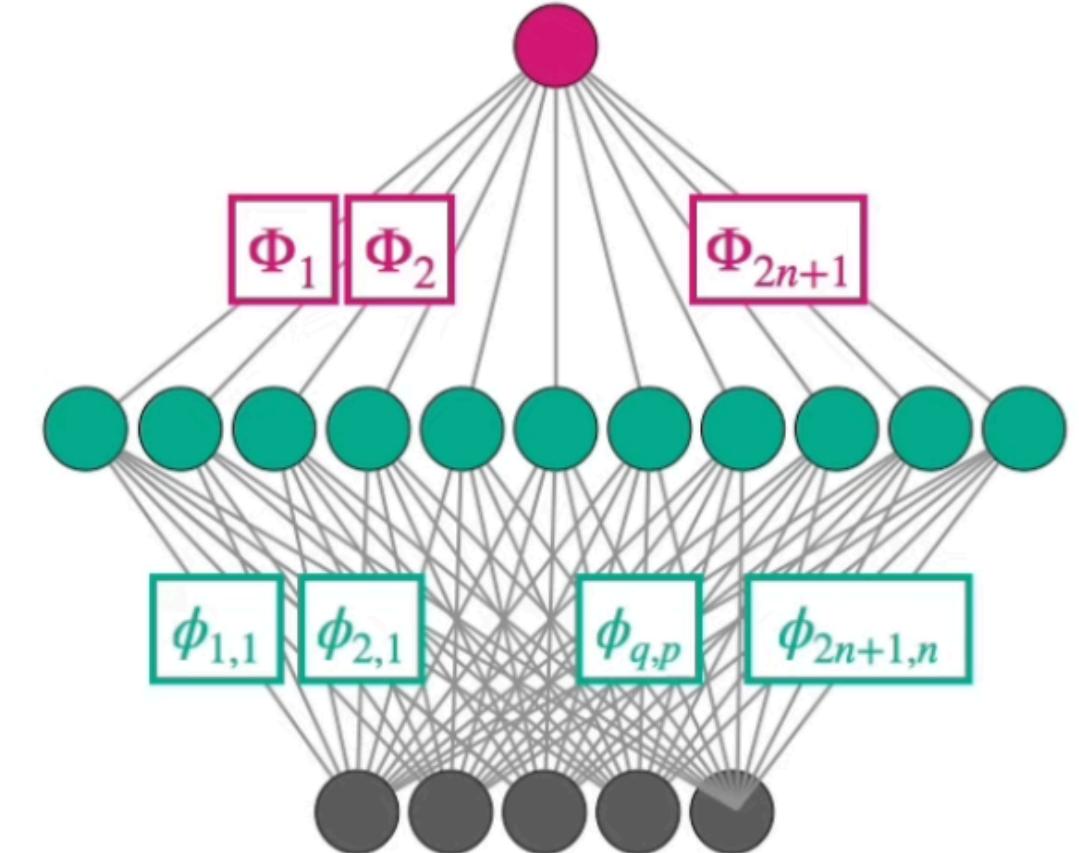
Kolmogorov-Arnold Network

$$f(x_1, x_2, \dots, x_n) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^n \phi_{q,p}(x_p) \right)$$

Diagram illustrating the structure of the Kolmogorov-Arnold Network:

- Layer 2:** Represented by a pink box labeled Φ_q .
- Layer 1:** Represented by a green box labeled $\phi_{q,p}$.
- Inputs:** Represented by a grey box.

The network consists of two layers. Layer 1 has n neurons, each receiving input from all n inputs. Layer 2 has $2n+1$ neurons, each receiving input from all neurons in Layer 1.



Dr. Luis Serrano's YouTube channel

<https://demonstrations.wolfram.com/BSplineCurveWithKnots/>

**Thank You
Q and A**