Theoretical and Societal Topics in AI and Deep Learning for Physicists

P.Baldi

Department of Computer Science
AI in Science Institute
Center for Machine Learning and Intelligent Systems
University of California, Irvine

Outline

1. The Overparameterization “Problem”
2. Applications of Transformers in Physics
3. AI Concerns and Safety

Outline

1. The Overparameterization “Problem”
2. Applications of Transformers in Physics
3. AI Concerns and Safety

The Overparametrization "Problem"

- The problem comes from the following dogma:

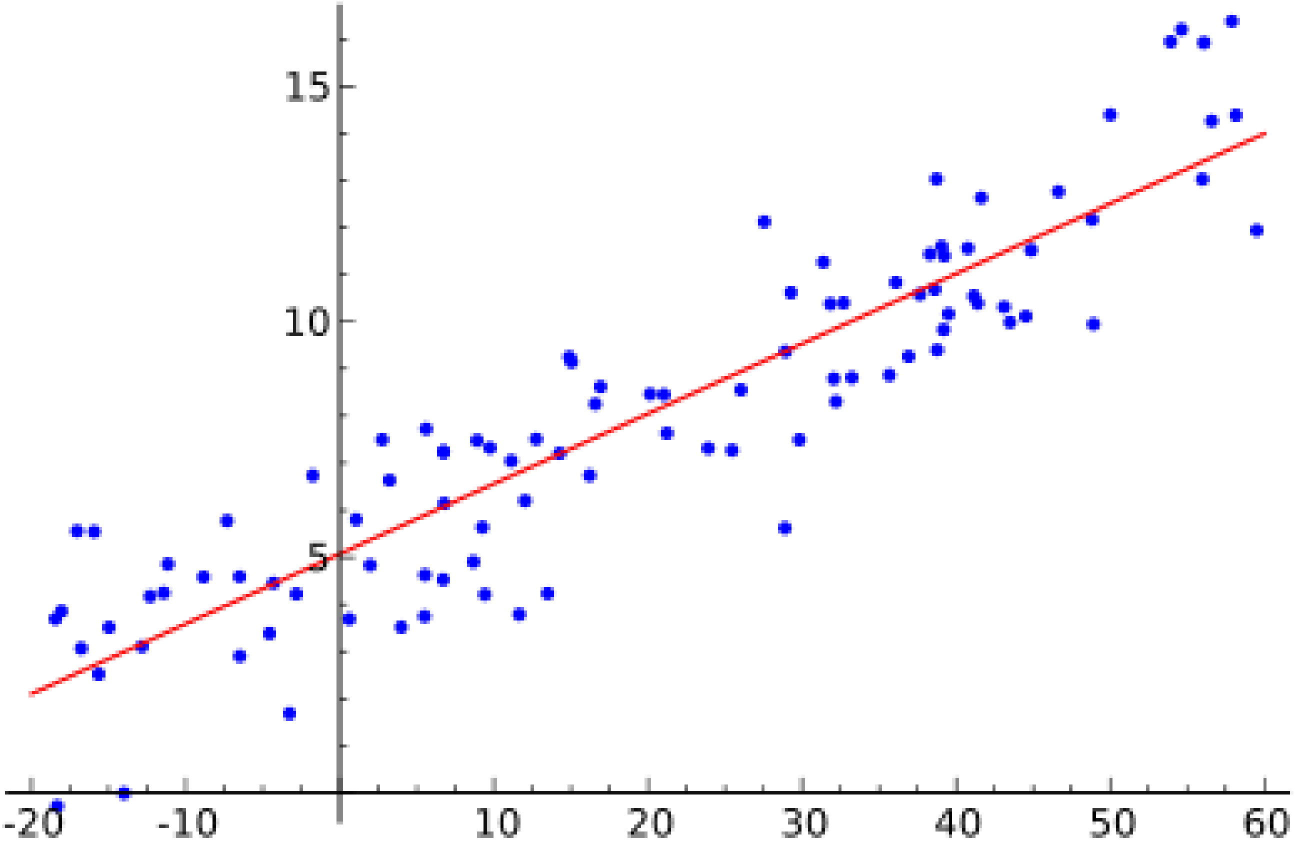
To fit a model with W parameters one ought to have a number of data points D satisfying $D \geq W$.

- Many papers have been written stating that one ought to have, for instance, $D \geq 10W$.

- However, it is common in deep learning applications to see good results in the regime where $D < W$, even $D \ll W$.

Dogma's Origin: Shallow Learning

Carl Friedrich Gauss (1777 – 1855)
Adrien-Marie Legendre (1752-1833)



- Shallow learning already contains most of the ideas behind machine learning/DL.
- However, it is misleading in 3 fundamental aspects: (1) existence of closed form solution; (2) easy to interpret and explain; (3) model with n parameters needs n data points.

The Overparametrization “Conjecture”

Dogma. To fit a model with W parameters one ought to have a number of data points D satisfying $D \geq W$.

Dogma is wrong. It is common in deep learning applications to see good results in the regime where $D < W$, even $D \ll W$.

Can we save the dogma? Does a possibly very small constant c exist such that:

To fit a model with W parameters one ought to have a number of data points D satisfying $D \geq cW$?

Counter-Example

No such constant c exists.

Counter-example: Neural network of depth n, with one unit per layer. All units are linear and without bias. As a result, W=n.

Output = w_1 w_2 ….w_n (Input)= P(Input)

Such a network can be trained with D= 1, while W can be arbitrarily large.

Many possible generalizations: Output = W1….W_n (Input).

Rescue Concepts

Free Parameters?

No. Counter-example has W-1=n-1 free parameters and D=1

Effective Parameters?

May be. But what is an effective parameter? How does one count the number of effective parameters?

The Overparametrization "Problem"

- The problem comes from the following dogma:

- To fit a model with W parameters one ought to have a number of data points D satisfying $D \geq W$.

- Many papers have been written stating that one ought to have, for instance, $D \geq 10W$.

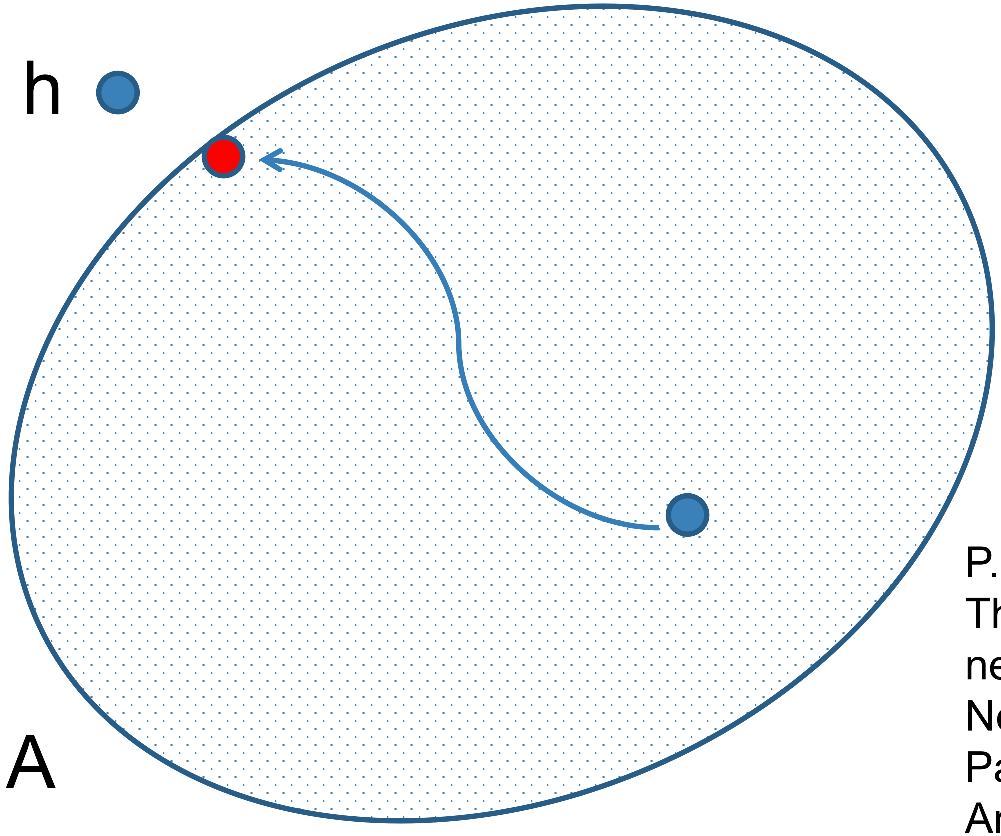
- However, it is common in deep learning applications to see good results in the regime where $D < W$, even $D \ll W$.

Plausible Conjecture?

Cardinal Capacity

h = target function (typically known from examples)

A = class of hypothesis or approximating functions (typically associated with a NN architecture)



$$C(A) = \log_2 |A|$$

- Average number of bits required to specify a function in A.
- In a neural architecture, number of bits that must be transferred from the data to the synapses during learning

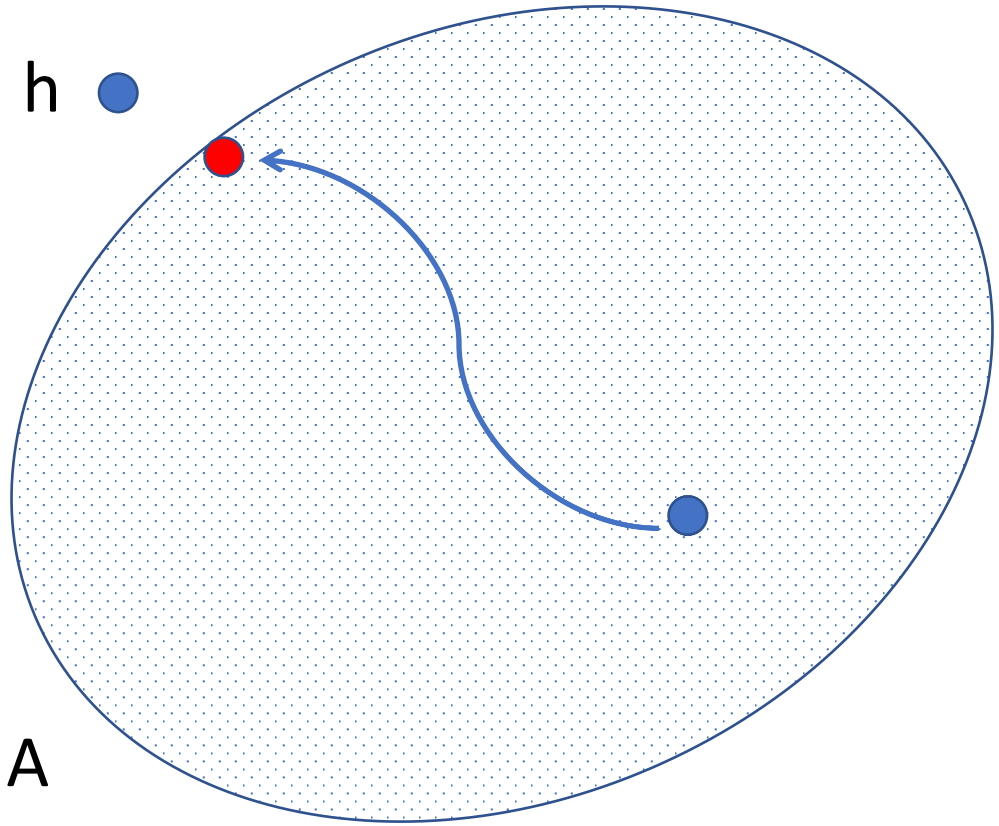
P. Baldi and R. Vershynin. The capacity of feedforward neural networks. Neural Networks, 116, August 2019, Pages 288-311, (2019). Also Arxiv 1901.00434.

Outline

1. The Overparameterization “Problem”
2. Applications of Transformers in Physics
3. AI Concerns and Safety

Neural Network Capacity

- h = target function (typically known from examples)
- A = class of hypothesis or approximating functions (typically associated with a NN architecture)



$$\mathbf{C(A) = \log_2 |A|}$$

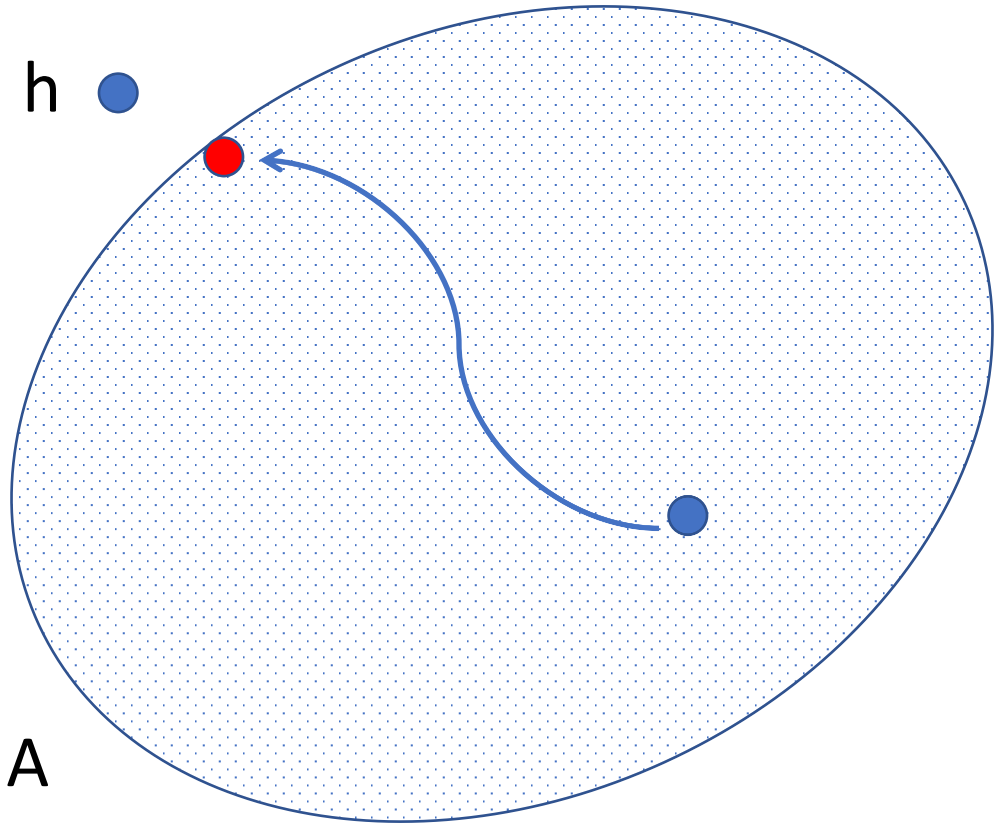
Number of bits required to specify a function in A

Neural Network Capacity

- Can we compute C(A) for specific, interesting, neural networks?

$C(A) = \log_2 |A|$

For neural networks: the number of bits that must be communicated from the training data to the synapses



Layered Fully Connected Feedforward Neural Networks

- Layered, feedforward, fully-connected network with layers of size $n_1, n_2, \ldots, n_L$:

$$C(n_1, n_2, \ldots, n_L) \approx \sum_{k=1}^{L-1} \min(n_1, n_2, \ldots, nk)\, n_k n_{k+1}$$

- Extensions to polynomial threshold gates, partial connectivity, weight sharing (CNNs)

P. Baldi and R. Vershynin. The capacity of feedforward neural networks. Neural Networks, 116, August 2019, Pages 288-311, (2019). Available online 22 April 2019. https://doi.org/10.1016/j.neunet.2019.04.009. Also: Arxiv 1901.00434

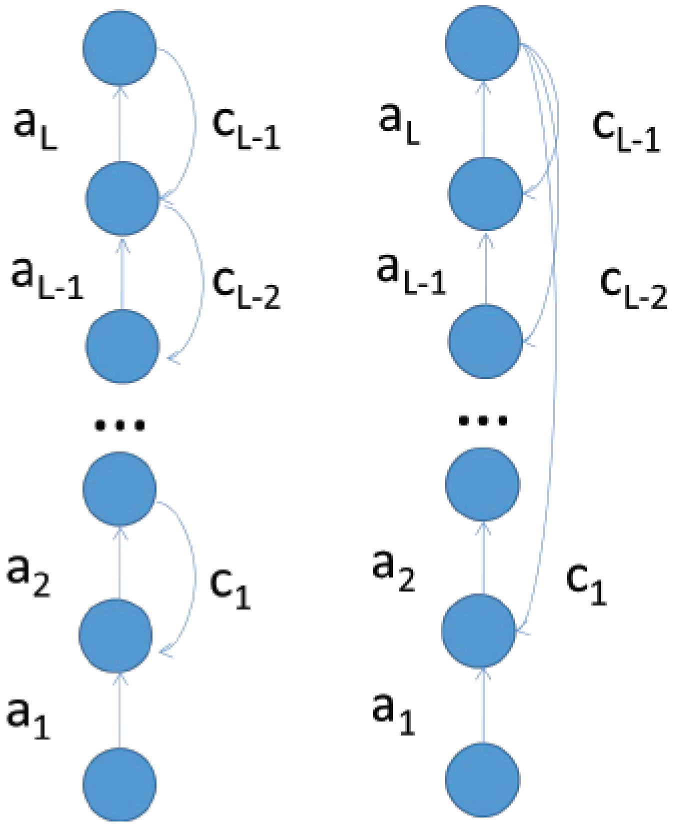
Layered Fully Connected Feedforward Neural Networks

- Layered, feedforward, fully-connected network with layers of size $n_1, n_2, \ldots, n_L$:

$$C(n_1, n_2, \ldots, n_L) \approx \sum_{k=1}^{L-1} \min(n_1, n_2, \ldots, nk)\, n_k n_{k+1}$$

- Extensions to polynomial threshold gates, partial connectivity, weight sharing (CNNs)

P. Baldi and R. Vershynin. The capacity of feedforward neural networks. Neural Networks, 116, August 2019, Pages 288-311, (2019). Available online 22 April 2019. https://doi.org/10.1016/j.neunet.2019.04.009. Also: Arxiv 1901.00434



$$O = PI = a_1 a_2 \dots a_L I \qquad \alpha = E(TI), \text{ and } \beta = E(I^2)$$

$$\begin{cases} \frac{da_1}{dt} = c_1(\alpha - \beta P) \\ \frac{da_2}{dt} = c_2 a_1(\alpha - \beta P) \\ \dots \\ \frac{da_{L-1}}{dt} = c_{L-1} a_1 a_2 \dots a_{L-2}(\alpha - \beta P) \\ \frac{da_L}{dt} = a_1 \dots a_{L-1}(\alpha - \beta P) \end{cases}$$

$$c_i \frac{da_{i+1}}{dt} = c_{i+1} a_i \frac{da_i}{dt} \qquad a_{i+1} = \frac{c_{i+1}}{2c_i} a_i^2 + C$$

$$a_i = k_0 + k_1 a_1^2 + \dots + k_{i-1} a_1^{2^{i-1}}$$

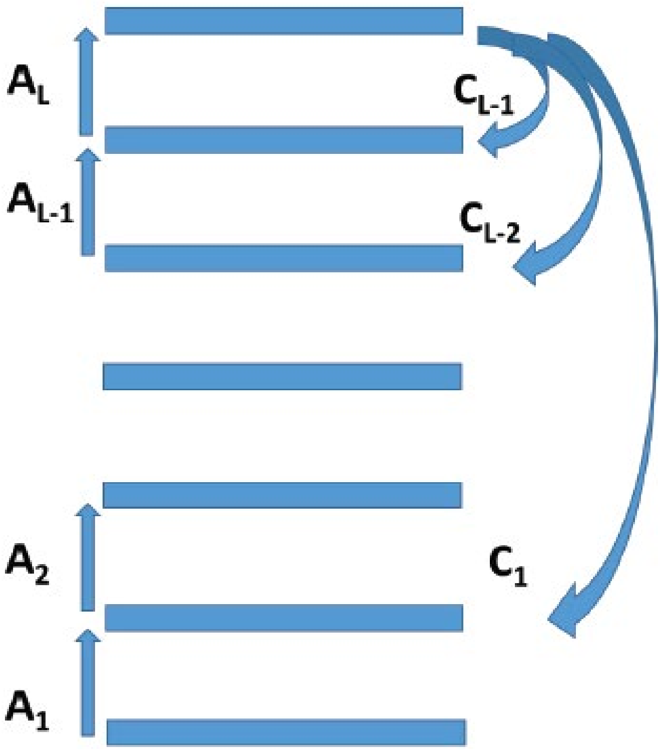
$$k_{i-1} = \frac{c_i}{2c_{i-1}} \left(\frac{c_{i-1}}{2c_{i-2}}\right)^2 \left(\frac{c_{i-2}}{2c_{i-3}}\right)^4 \dots \left(\frac{c_3}{2c_2}\right)^{2^{i-1}}$$

$$da_1/dt = Q(a_1)$$

Q is a polynomial of degree $2^L - 1$ with negative leading coefficient

Theorem: **From any starting condition, the system converges to a fixed point on the manifold α-βP=0**

General Linear Case



$$\frac{dA_i}{dt} = C_i(\Sigma_{TI} - P\Sigma_{II})A_1^t \dots A_{i-1}^t$$

SRBP $\leftrightarrow$
RBP

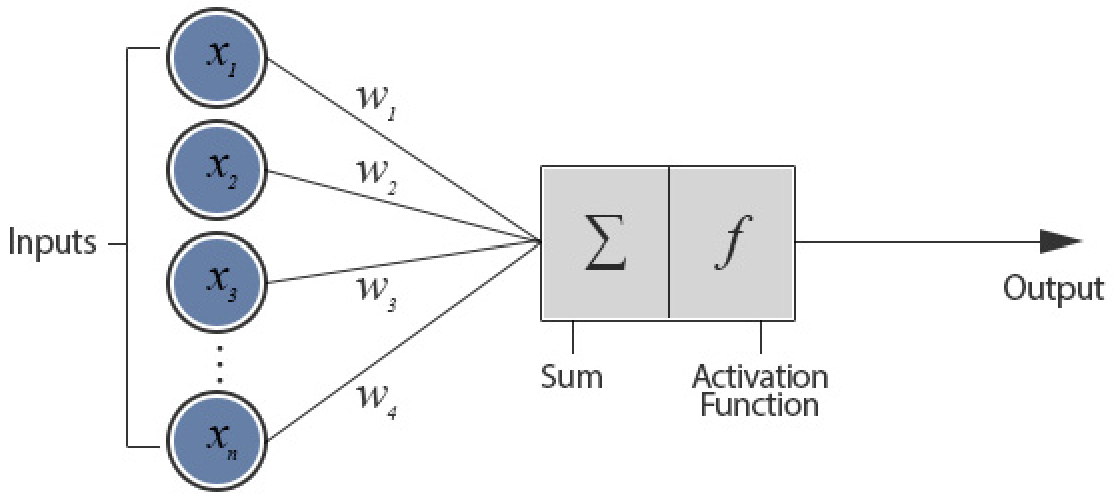
$$\frac{dC_i}{dt} = A_{i-1} \dots A_1(\Sigma_{TI} - P\Sigma_{II})^t$$

ASRBP

$$\frac{dA_i}{dt} = A_{i+1}^t \dots A_L^t(\Sigma_{TI} - P\Sigma_{II})A_1^t \dots A_{i-1}^t$$

BP

The Standard Model



- SM universal approximation properties
- SM extensions (softmax, polynomial activations, product of outputs, ….)

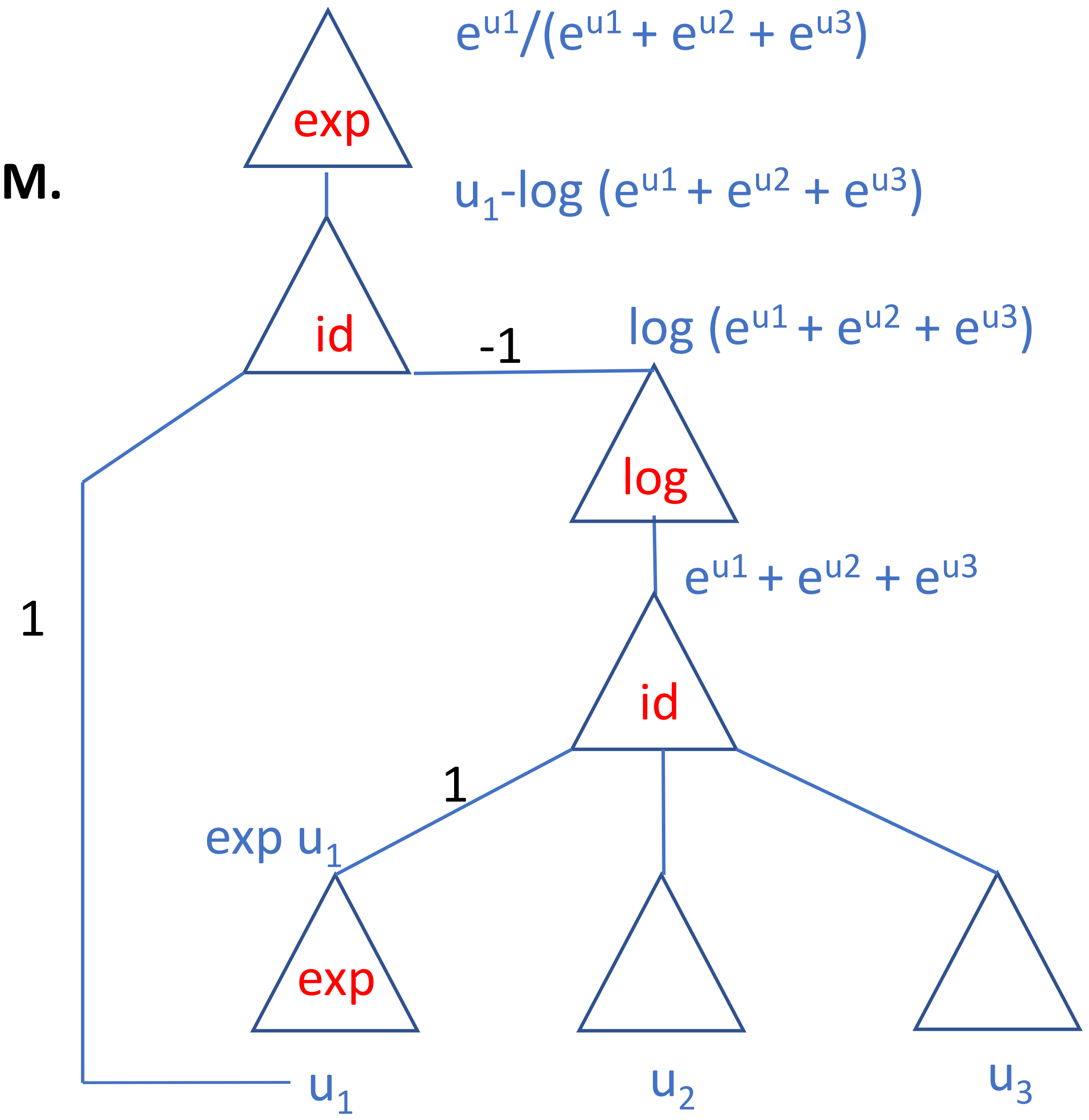
O=f($\sum w_i x_i$)

Basic elementary operations:

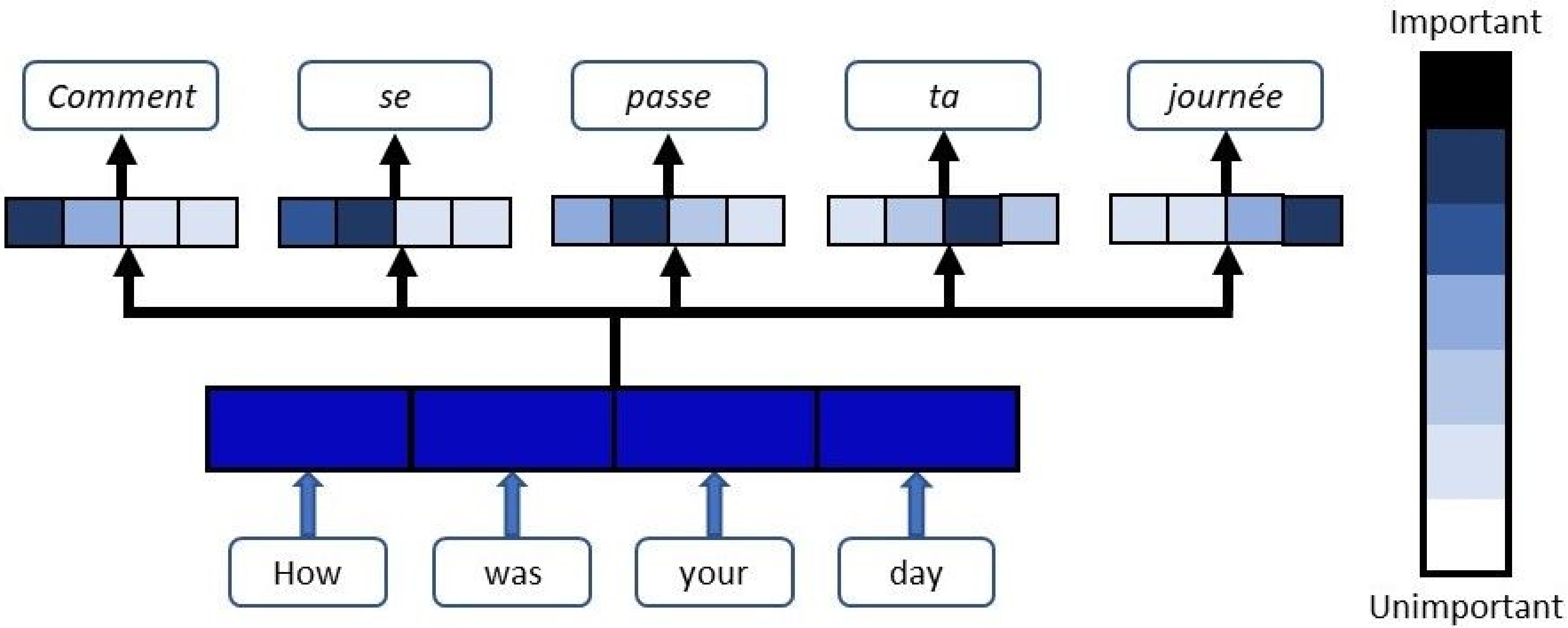
1) Activation S= Dot product x.w

2) Output O=f(S) (f linear or non-linear activation function)

SoftMax is an extension of the SM.



Attention in DL and NLP applications



Sequence to sequence models

Attention Mechanisms in DL and NLP

Various formulations:

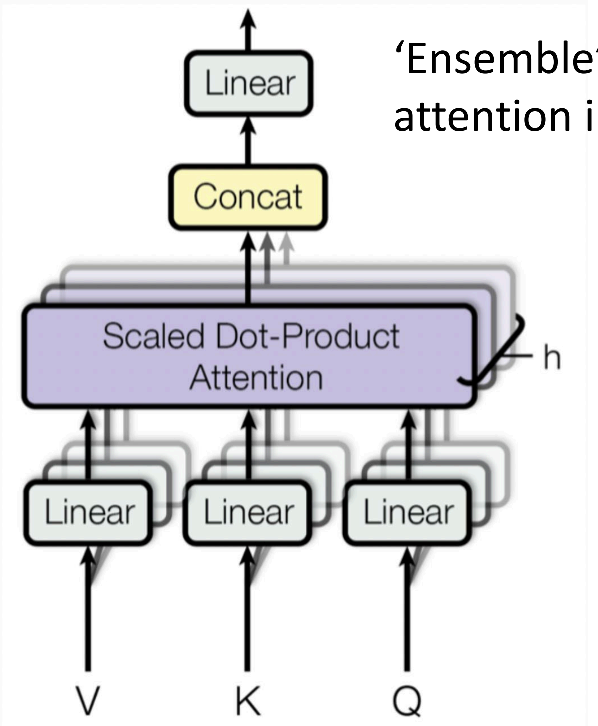
- Content-base attention Graves et al., 2014
- Dot-Product attention Luong et al., 2015
- Additive attention Bahdanau et al., 2015
- Vaswani et al. 2017
- ………..
- **Transformer Architectures**
- Standard modules in DL packages (TensorFlow, PyTorch)
- Google’s BERT, OpenAI’s GPT , XLNet ….

Transformer Model & (self)-attention

The Transformer Model is **entirely** built on the self-attention mechanisms, **without** using sequence-aligned recurrent architectures.

Every input element has three learnable vectors: Query (Q), Key (K), and Value (V)

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\mathsf{T}}}{\sqrt{n}}\right)\mathbf{V}$$



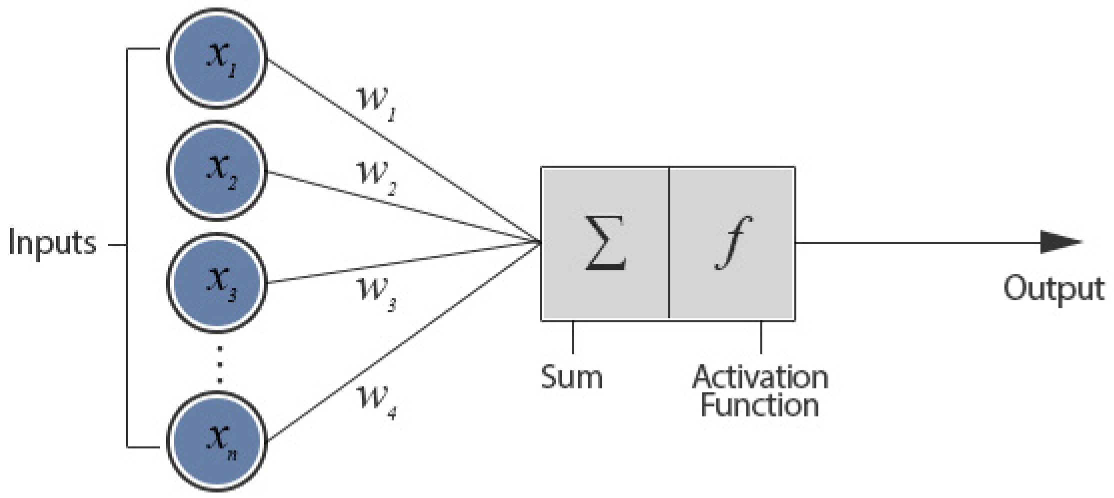
'Ensemble' multiple attention in parallel

Rather than only computing the attention once, the multi-head mechanism runs through the scaled dot-product attention multiple times in parallel.

RoadMap

1. Introduction to Attention and the Standard Model
2. A Taxonomy of Attention Mechanisms (Quarks)
3. Transformers and Attention
4. Applications of Attention
5. Mathematical Theory of Attention
6. Large Language Models

The Standard Model



O=f($\sum w_i x_i$)

Basic elementary operations:

1) Activation S= Dot product x.w

2) Output O=f(S) (f linear or non-linear activation function)

3 variable types:

S, O, w

Classification of Attention Mechanisms (or Extensions of the SM)

- In the SM, there are 3 types of variables: S (activation), O (output), and w (synaptic weights).
- Attention signals can be classified according to their attending Origin, their attended Target, and the underlying Mechanism.
- With two mechanisms, addition and multiplication, this corresponds to 18 possibilities:

	S	O	W
S	+, x	+, x	+, x
O	+, x	+, x	+, x
W	+, x	+, x	+, x

- Multiplicity issues.
- Origin: only of type O → 6 possibilities.

Classification of Attention Mechanisms

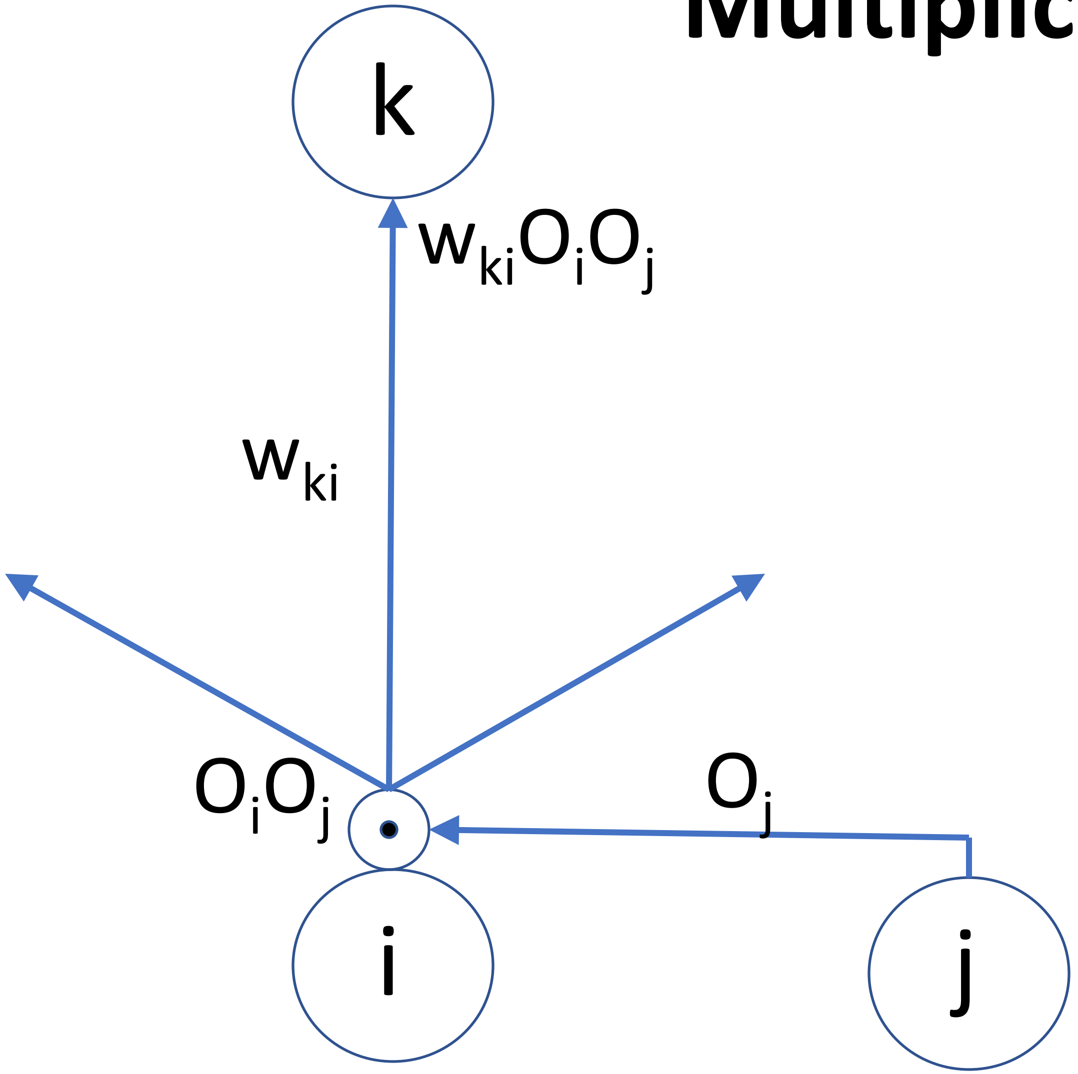
- **Origin is of type O**
- **Six possibilities:**

Target

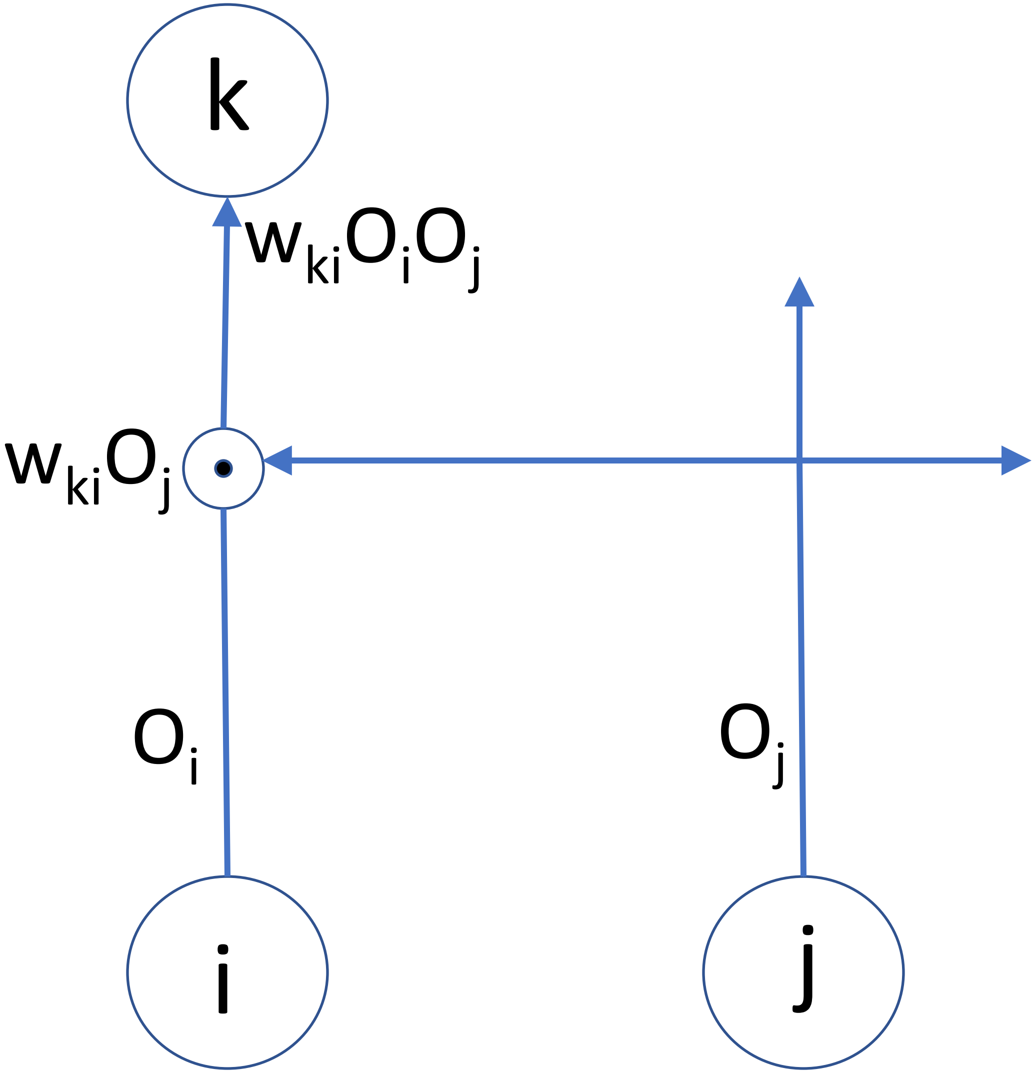
Mechanism

	Activation (S)	Output (O)	Weight (w)
Addition	Activation Attention (SM)		
Multiplication		Output Gating	Synaptic Gating

Multiplication



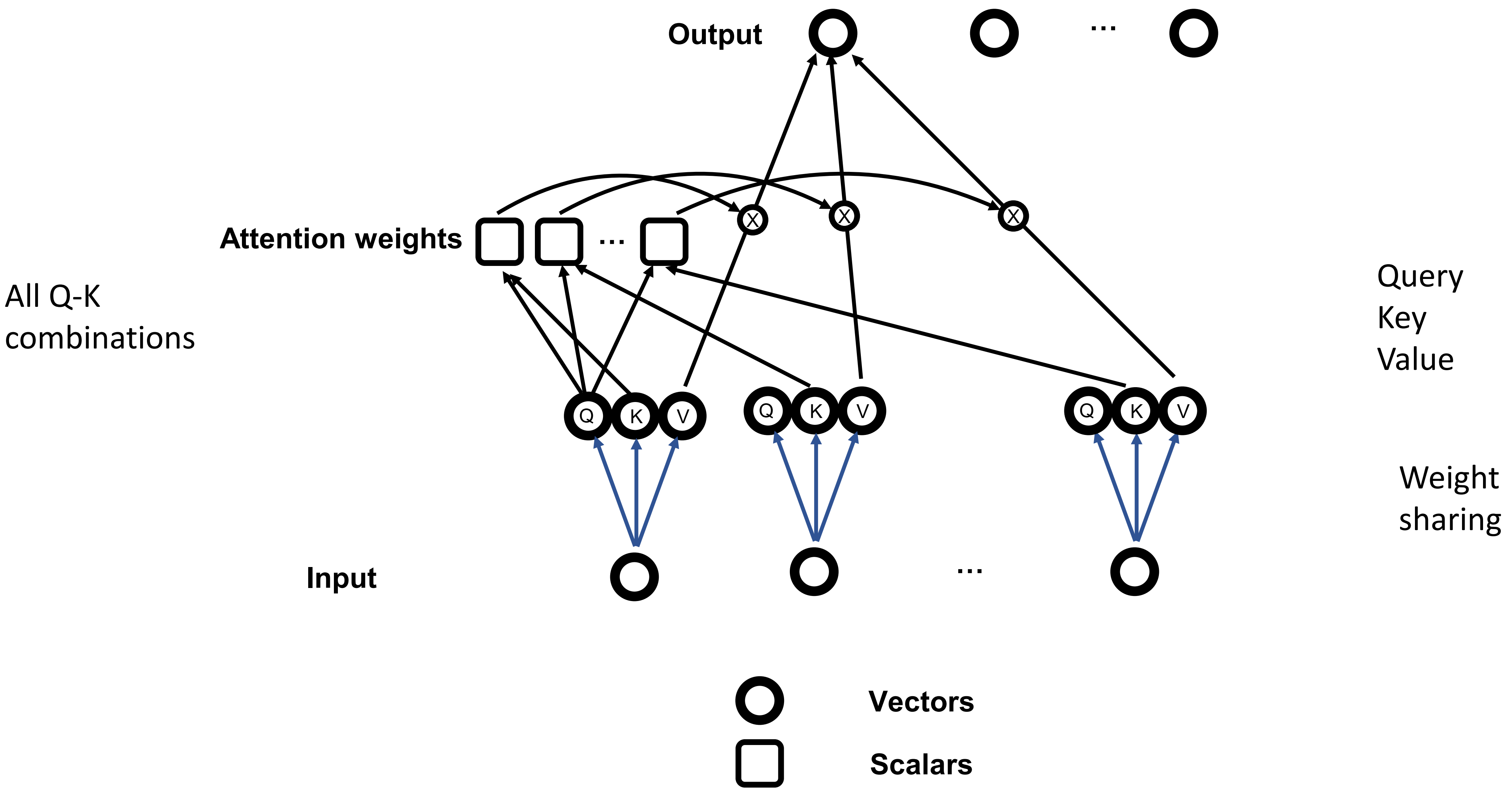
Output Gating



Synaptic Gating

RoadMap

1. Introduction to Attention and the Standard Model
2. A Taxonomy of Attention Mechanisms (Quarks)
3. Transformers and Attention
4. Applications of Attention
5. Mathematical Theory of Attention



Output
...
Attention weights
...
X
X
X
All Q-K
combinations
Query
Key
Value
Q
K
V
Q
K
V
Q
K
V
Weight
sharing
Input
...
Vectors
Scalars

Database Vocabulary

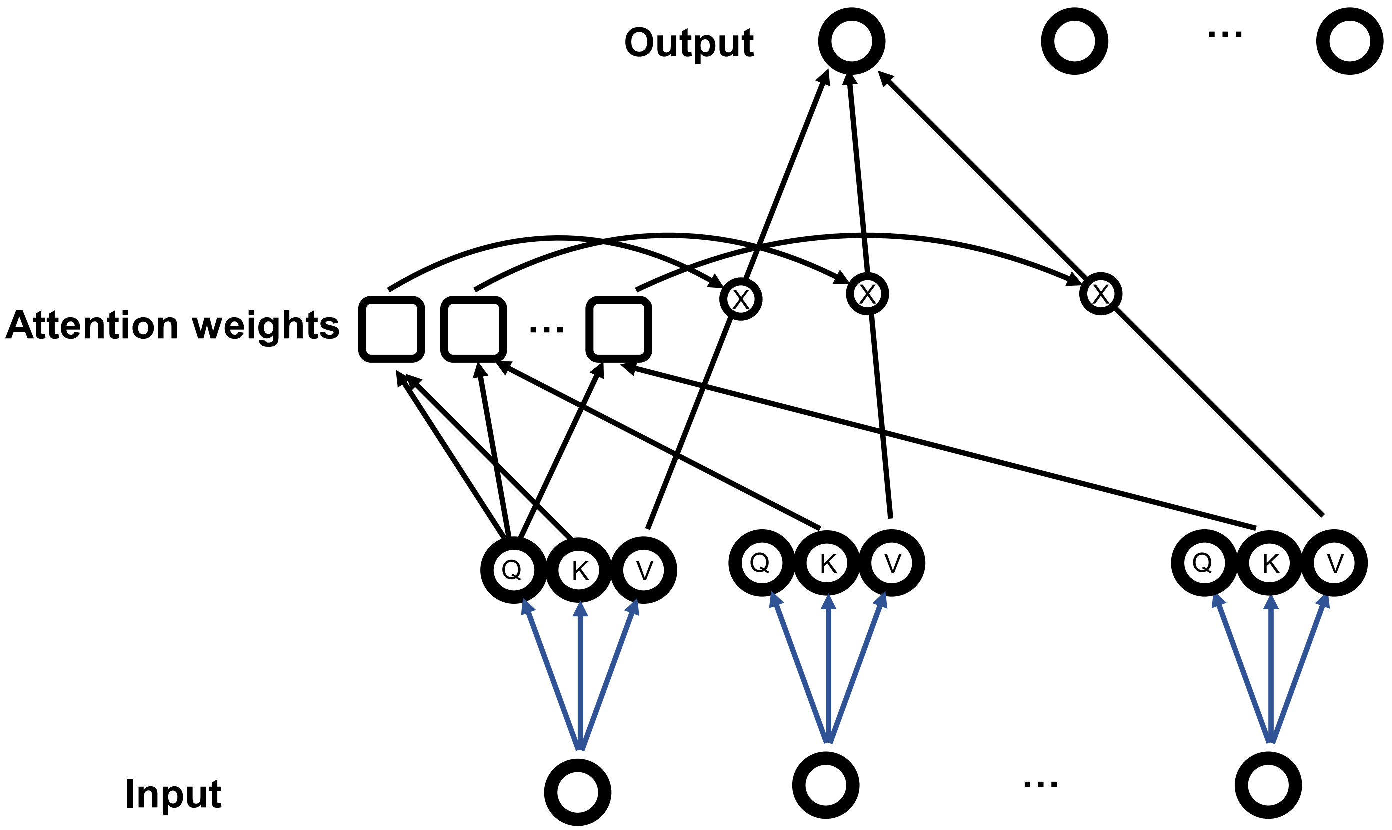
Key

Student ID	Driver License #	Address	First Name	Last Name
	123456			
	123789			
	123770			
	123775			

Values= Rows Contents

Query: 123770?

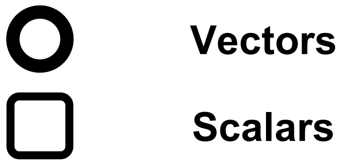
All Q-K
combinations



Query
Key
Value

Weight
sharing

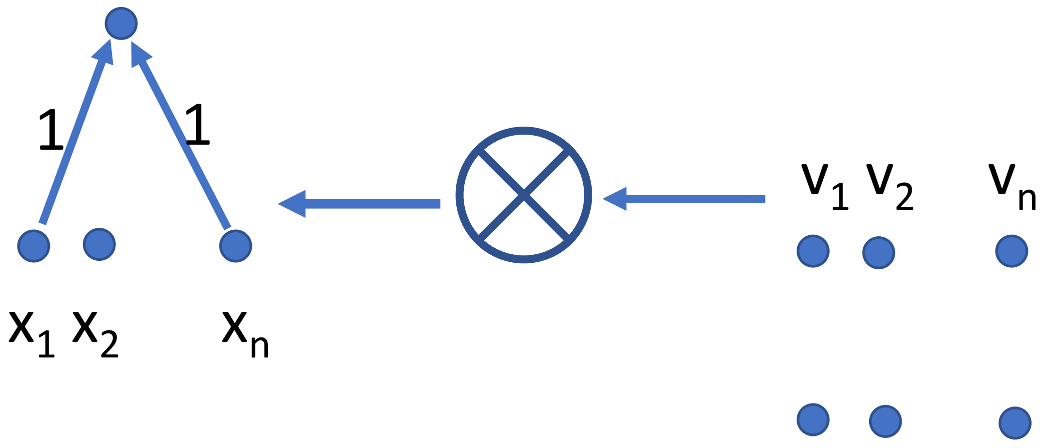
Fundamental Property: Invariant to permutation of the inputs (!).



Attention Enables Computing the Dot Product of the Activities of Two Layers of the Same Size (output or synaptic gating)

gated output

O= $\sum x_i v_i$



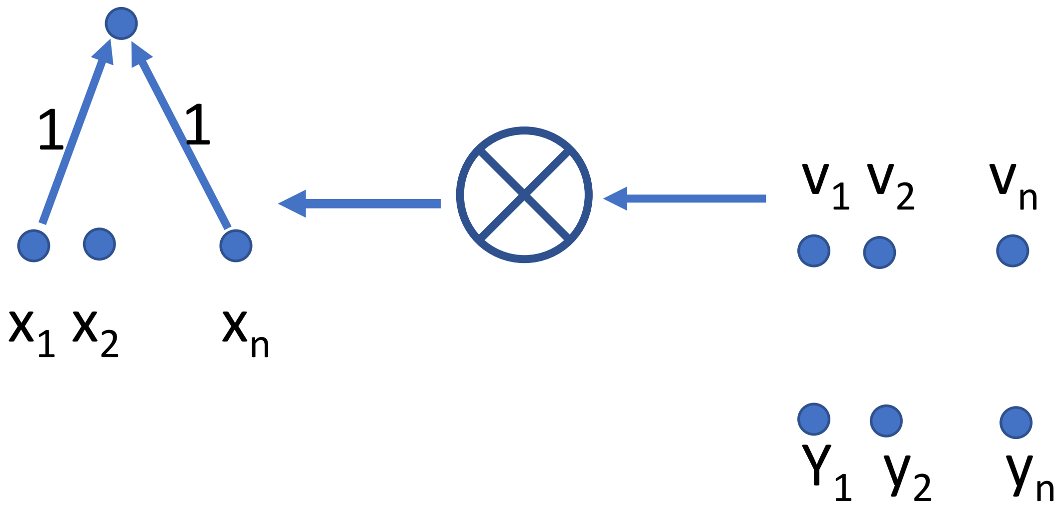
pairwise gating layer

[Can be used to derive alternative proof of universal approximation properties for SM + attention]

Softmax Attention=Dot Product with Softmax (output or synaptic gating)

gated output

O=$\text{sum}_i\ v_i\ x_i$

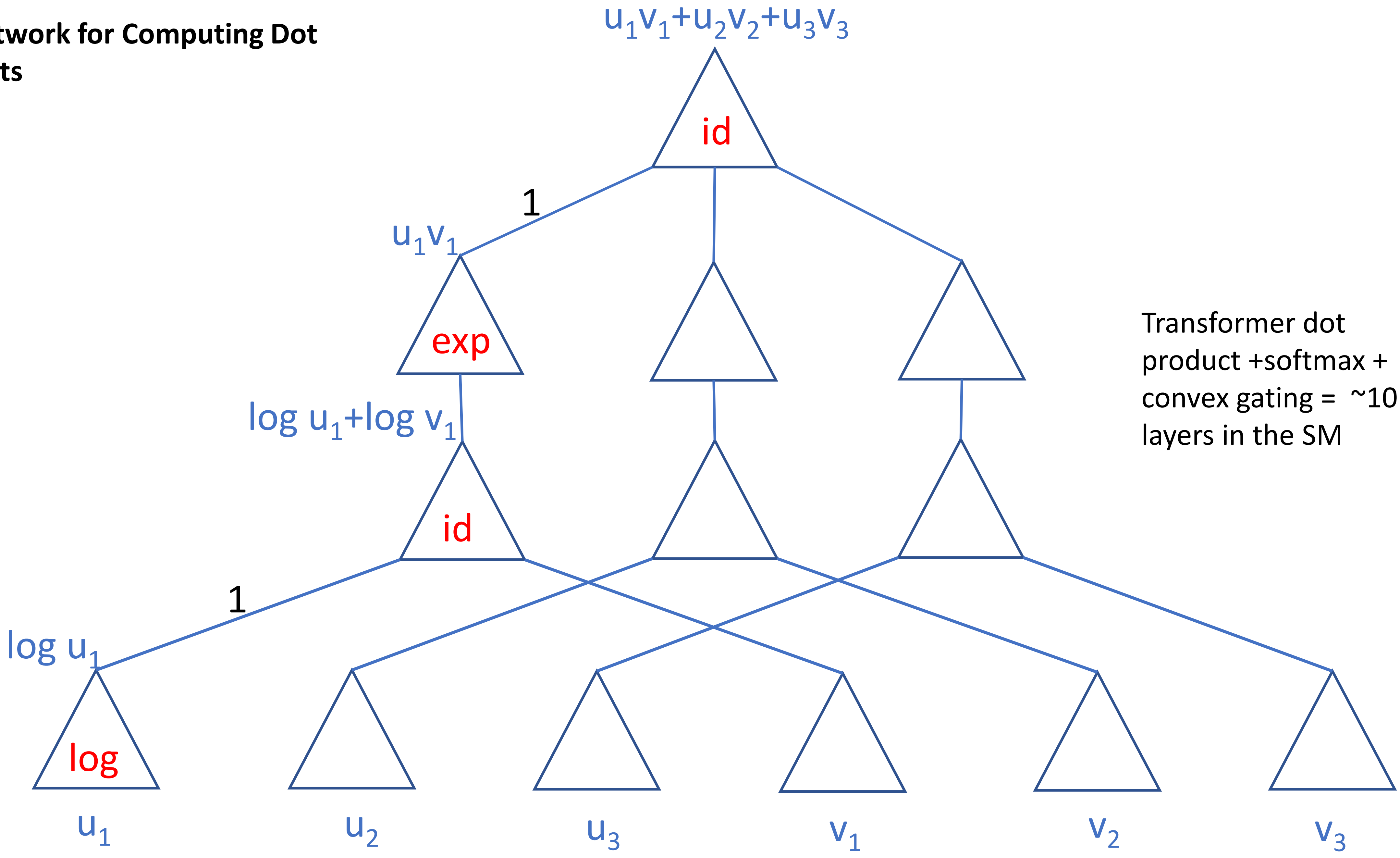


gating layer:

softmax unit

vi=exp y_i / sum_j exp v_j

SM Network for Computing Dot Products



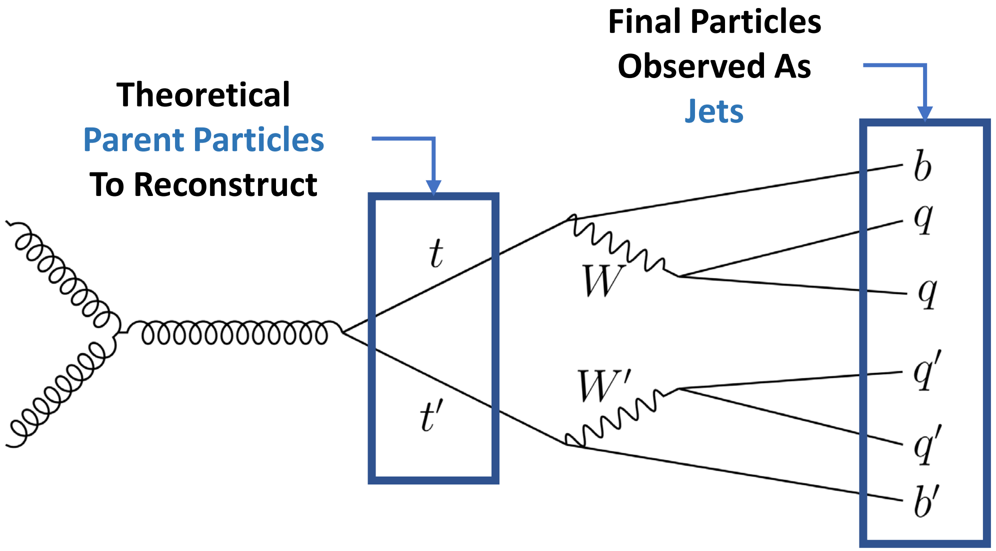
Transformer dot product +softmax + convex gating = ~10 layers in the SM

RoadMap

1. Introduction to Attention and the Standard Model
2. A Taxonomy of Attention Mechanisms (Quarks)
3. Transformers and Attention
4. Applications of Attention
5. Mathematical Theory of Attention
6. Large Language Models

SPANet Jet-Parton Matching in LHC Top Quark Decays

- Primary (all-hadronic) decay channel produces six particles - two qqb triplets with opposite charge – originating from the top – antitop particle pair which we wish to reconstruct.
- After these particles are produced, they are propagated and measured by the detector as ***jets***.
- Along with the jets from each of the particles, there may be additional jets from other decay products.



$$\{j_1, j_2, j_3, j_4, j_5, j_6, j_7, j_8\}$$

Match Jets to Particle Labels

$$\{b, q', \emptyset, q', b', \emptyset, q, q\}$$

Garbage Jets

This is a difficult matching problem: Observing the jets from the detector, can you determine which jets belong to which particles?
Effective matching requires exploiting the symmetries in this problem!

SPANet Complete Architecture

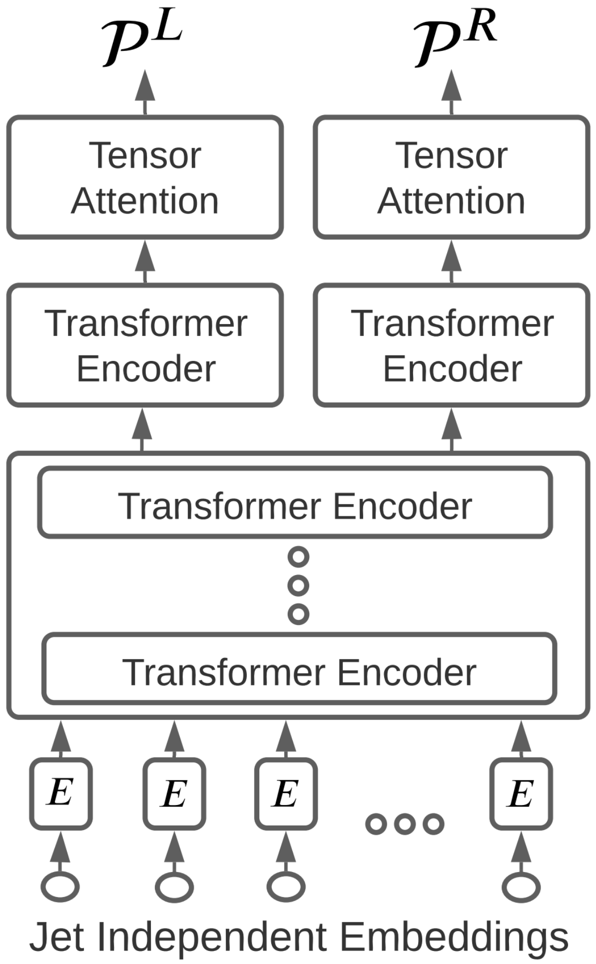
Construct an architecture following the structure of the original Feynman Diagram with attention as its core operation.

Attention Use

Symmetric jet matching

Particle-level encoding

Event-level context-aware encoding



Tensor attention to predict the most likely assignment of jets associated with each particle.

Split the information stream into a finite collection of "particles".

Heavily employ attention in several sections within our network for **context-aware permutation-invariant** learning.

Input is unsorted set of jet 4-momentum vectors.

SPANet Results

- We compare *SPANet* to a classical permutation-based method based on χ^2 probability of assignments.
- *SPANet* uses attention to match all top-quarks while the χ^2 method needs to compute many jet-permutations.
- *SPANet* **reduces the runtime** from $O(N^6)$ to $O(N^3)$ while **increasing efficiency** by $\sim 30\%$ across the board.

	χ^2 Efficiency			SPA-NET Efficiency		
N_{jets}	ϵ^{event}	ϵ_2^{top}	ϵ_1^{top}	ϵ^{event}	ϵ_2^{top}	ϵ_1^{top}
6	61.8%	65.0%	24.2%	80.7%	84.1%	56.7%
7	40.8%	50.4%	24.6%	66.8%	75.7%	56.2%
≥ 8	23.2%	35.5%	20.2%	52.3%	66.2%	52.9%
Inclusive	**37.7%**	**47.0%**	**23.0%**	**63.7%**	**73.5%**	**55.2%**

Runtime on 8 jet events

χ^2 : 369 *ms* per event

Spatter : 4.4 *ms* per event

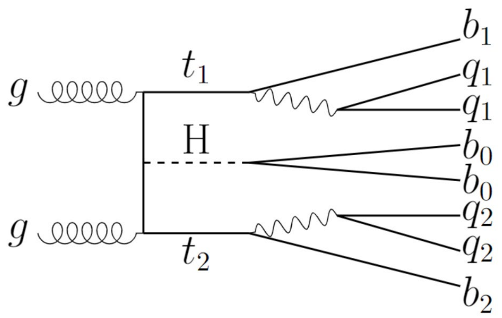
Alexander Shmakov

Michael James Fenton, Alexander Shmakov, Ta-Wei Ho, Shih-Chieh Hsu, Daniel Whiteson, and Pierre Baldi. Permutationless many-jet event reconstruction with symmetry preserving attention networks. *Physical Review* D, in press.

SPANet Upcoming Results

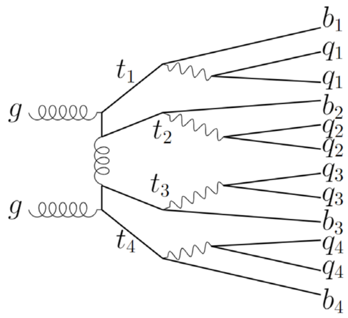
- General formulation allows us to extend this technique to virtually any possible event at the LHC.
- Split particle paths and symmetric attention may be extended to match jets in **incomplete events** – where one or more particles are missing due to detector loss, allowing us to use more training data.
- Extended this technique to two other, more complicated, events at the LHC: ttH and $tttt$.
- $tttt$ Event is so complex and large that the χ^2 method **cannot be tractably computed**!

ttH



	N_{jets}	Event Fraction	SPA-NET Efficiency Event	Higgs	Top	χ^2 Efficiency Event	Higgs	Top
All Events	== 8	0.261	0.370	0.497	0.540	0.056	0.193	0.092
	== 9	0.313	0.343	0.492	0.514	0.053	0.160	0.102
	≥ 10	0.313	0.294	0.472	0.473	0.031	0.150	0.056
	Inclusive	**0.972**	**0.330**	**0.485**	**0.502**	**0.045**	**0.164**	**0.081**
Complete Events	== 8	0.042	0.532	0.657	0.663	0.040	0.220	0.135
	== 9	0.070	0.422	0.601	0.596	0.019	0.152	0.079
	≥ 10	0.115	0.306	0.545	0.523	0.004	0.126	0.073
	Inclusive	**0.228**	**0.383**	**0.583**	**0.572**	**0.016**	**0.153**	**0.087**

tttt



	N_{jets}	Event Fraction	SPA-NET Efficiency Event	Top Quark
All Events	== 12	0.219	0.276	0.484
	== 13	0.304	0.247	0.474
	≥ 14	0.450	0.198	0.450
	Inclusive	**0.974**	**0.231**	**0.464**
Complete Events	== 12	0.005	0.350	0.617
	== 13	0.016	0.249	0.567
	≥ 14	0.044	0.149	0.504
	Inclusive	**0.066**	**0.191**	**0.529**

Alexander Shmakov, Michael James Fenton, Ta-Wei Ho, Shih-Chieh Hsu, Daniel Whiteson, Pierre Baldi. SPANet: Generalized Permutationless Set Assignment for Particle Physics using Symmetry Preserving Attention. *SciPost Physics*, in press.

x5000

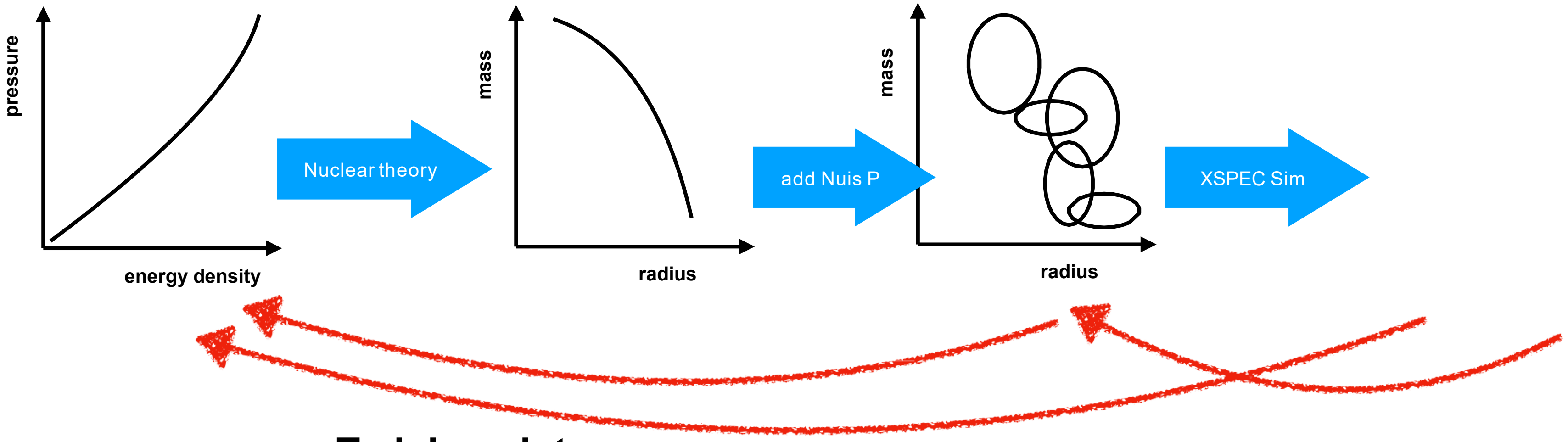
The problem

Equation of State (2 params)

Mass-Radius relation (2 params: M,R)

Star parameters
2 params: (mass, radius)
3 nuis params (dist, temp, dust)

Neutron Star X-ray data
(1024 chan,80% empty)



<u>Training data</u>

Fixed EOS, sample of (M,R) pairs
For each M,R pair, add 3 nuisance param
generate sample spectra

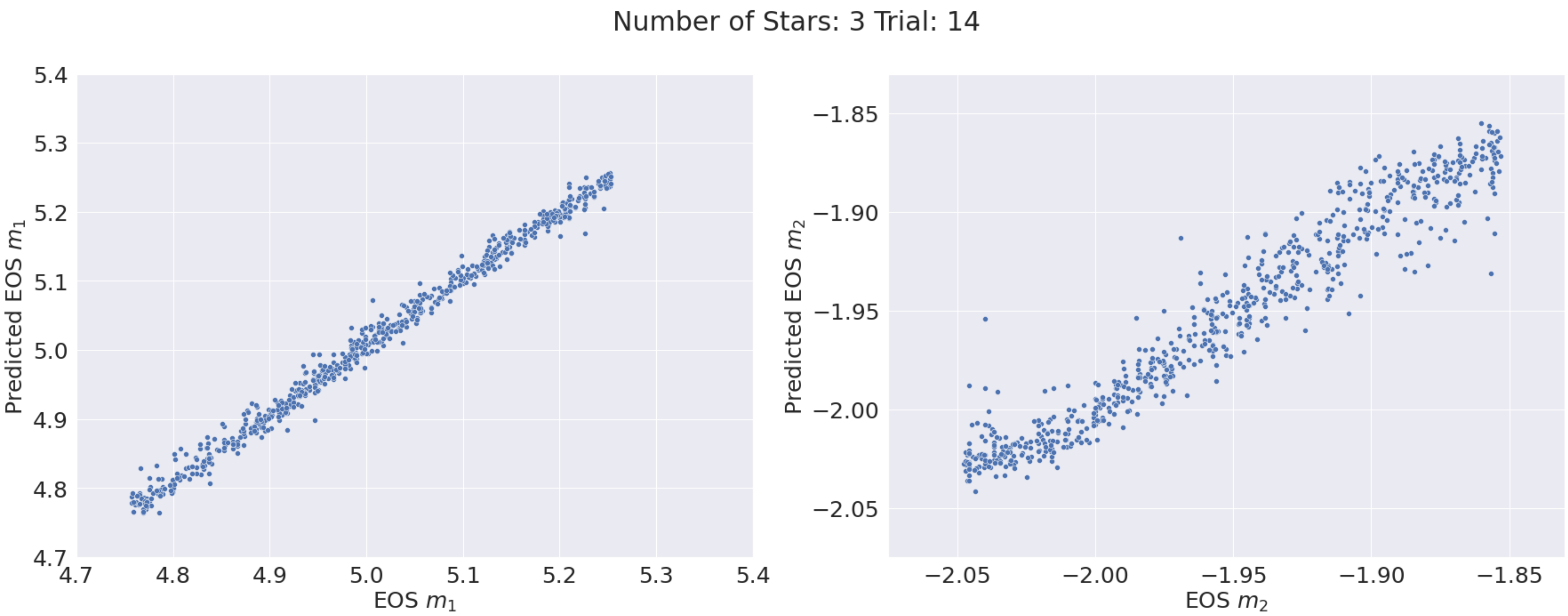
<u>Inference</u>

End-to-end: spectra -> EOS
Also might try: spectra-> star
star -> EOS

Jordan Ott

MLLR method: ttps://arxiv.org/abs/2002.04699

Prediction of EOS coefficients from 3 Stars (M,R)



Delaney Farrell, Pierre Baldi, Jordan Ott, Aishik Ghosh, Andrew W. Steiner, Atharva Kavitkar, Lee Lindblom, Daniel Whiteson, and Fridolin Weber. Deducing Neutron Star Equation of State Parameters Directly From Telescope Spectra with Uncertainty-Aware Machine Learning. *Journal of Cosmology and Astroparticle Physics*, 02, 016, (2023).

Outline

1. The Overparameterization “Problem”
2. Applications of Transformers in Physics
3. AI Concerns and Safety

Two Key Ideas

1. AI Safety and NI Safety
2. CERN-AI

-
-

Two Key Ideas

1. AI Safety and NI Safety
2. CERN-AI

Parallels between NI and AI Safety

Natural Intelligence Safety	Artificial Intelligence Safety
Evolution	Modular architectures, safety modules
Examples (parents, teachers, role models)	RL from Human Feedback (RLFH)
Principles (e.g. 10 commandments)	Constitutional AI
Law	AI laws
Societal	Agentic
Enforcement (e.g., police, lie detectors)	Enforcement (e.g., police^2, fake detectors)
Enforcement (e.g., military, WMD)	Enforcement (e.g. military^2, killer switches)

Two Key Ideas

1. AI Safety and NI Safety
2. CERN-AI

AI Concerns

1. AI Today and its Neuroscience Origins
2. The AI-Driven Hospital
3. AI Safety and Concerns

•

Two Key Ideas

1. AI Safety and NI Safety
2. CERN-AI