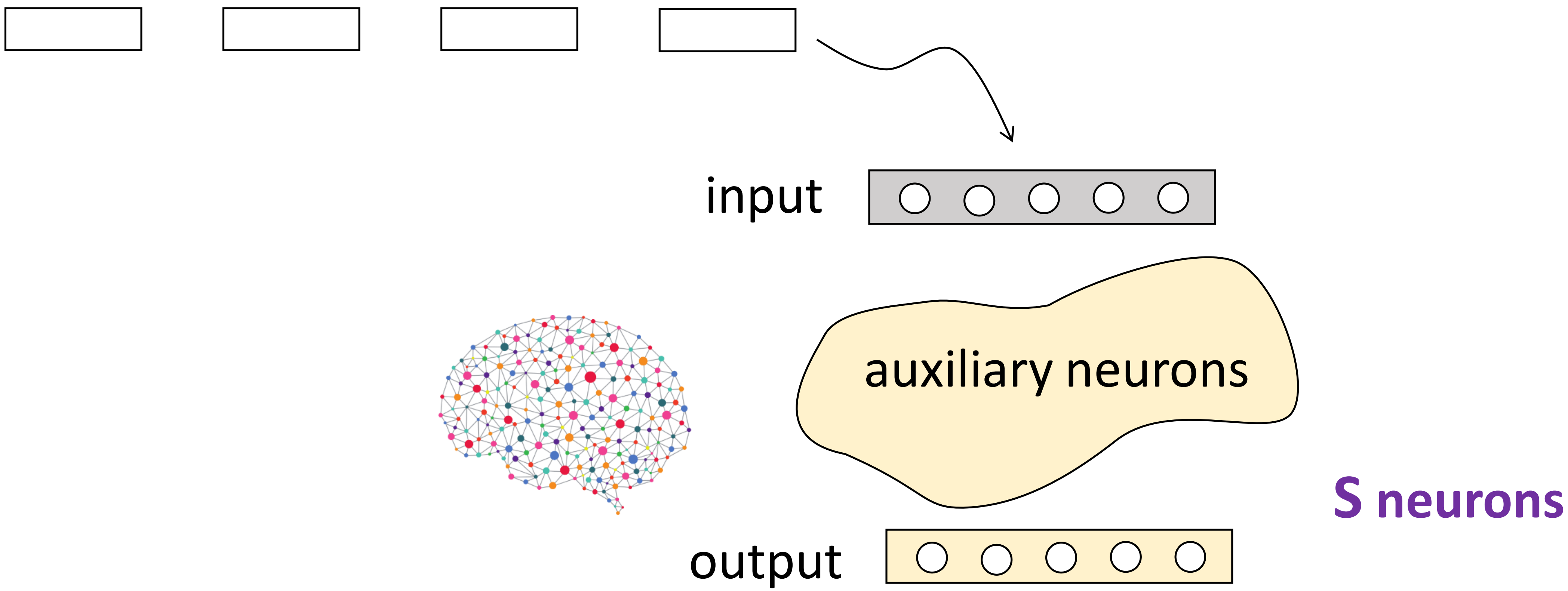
Neural Networks Through the Lens of Streaming Algorithms

Yael Hitron, Cameron Musco and Merav Parter

UMASS AMHERST

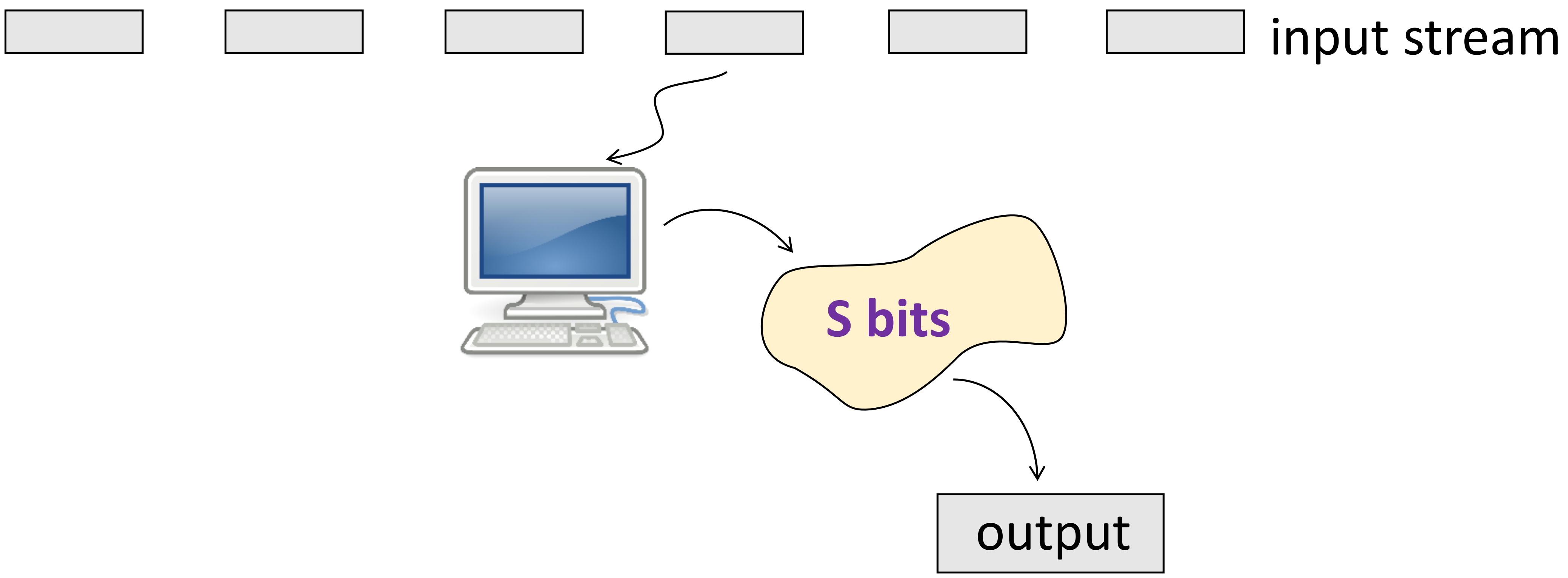
מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

Neural algorithms



Process large scale data with limited number of neurons

Streaming algorithms



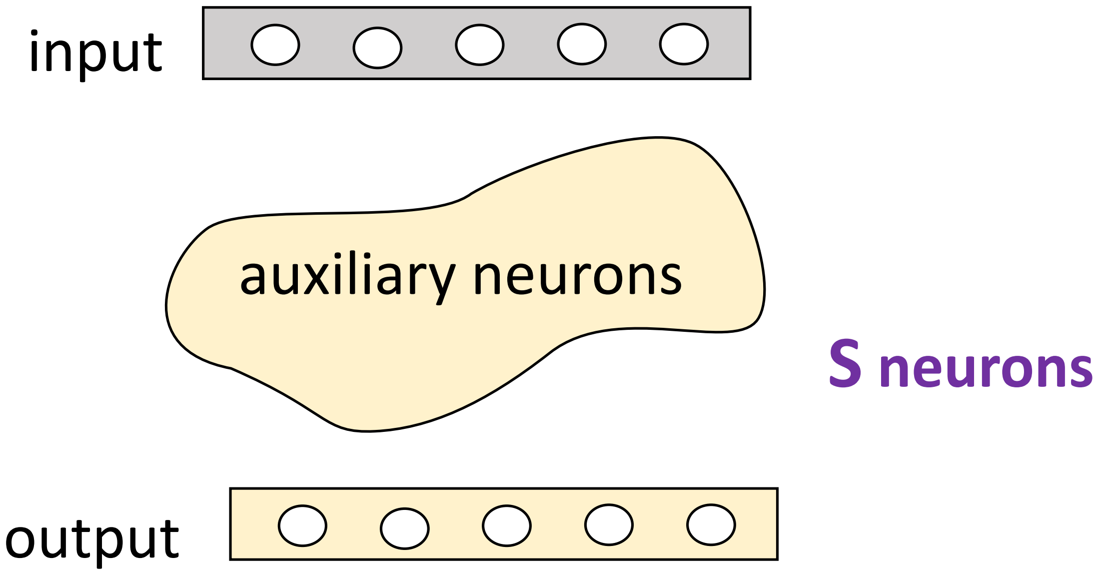
Process large scale input stream with limited space

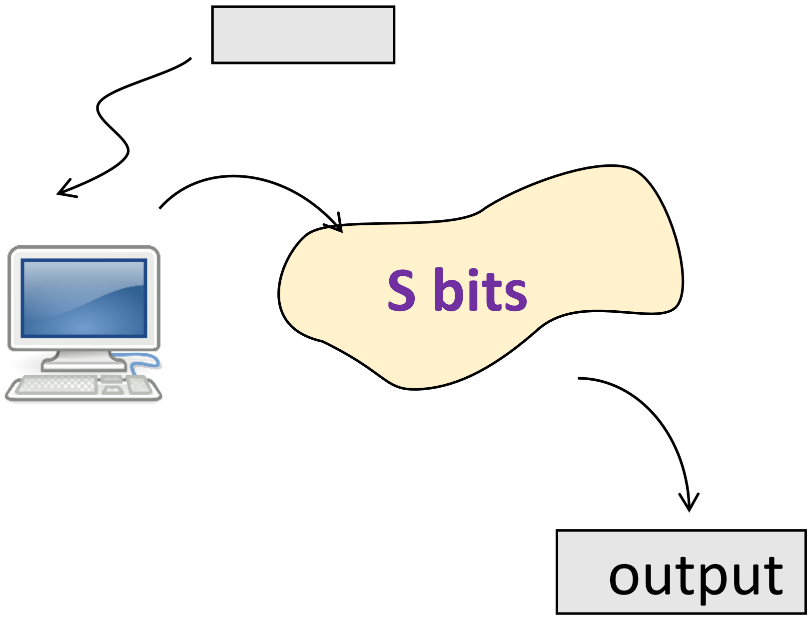
Streaming vs neural algorithms

Neural algorithms

?

Streaming algorithms





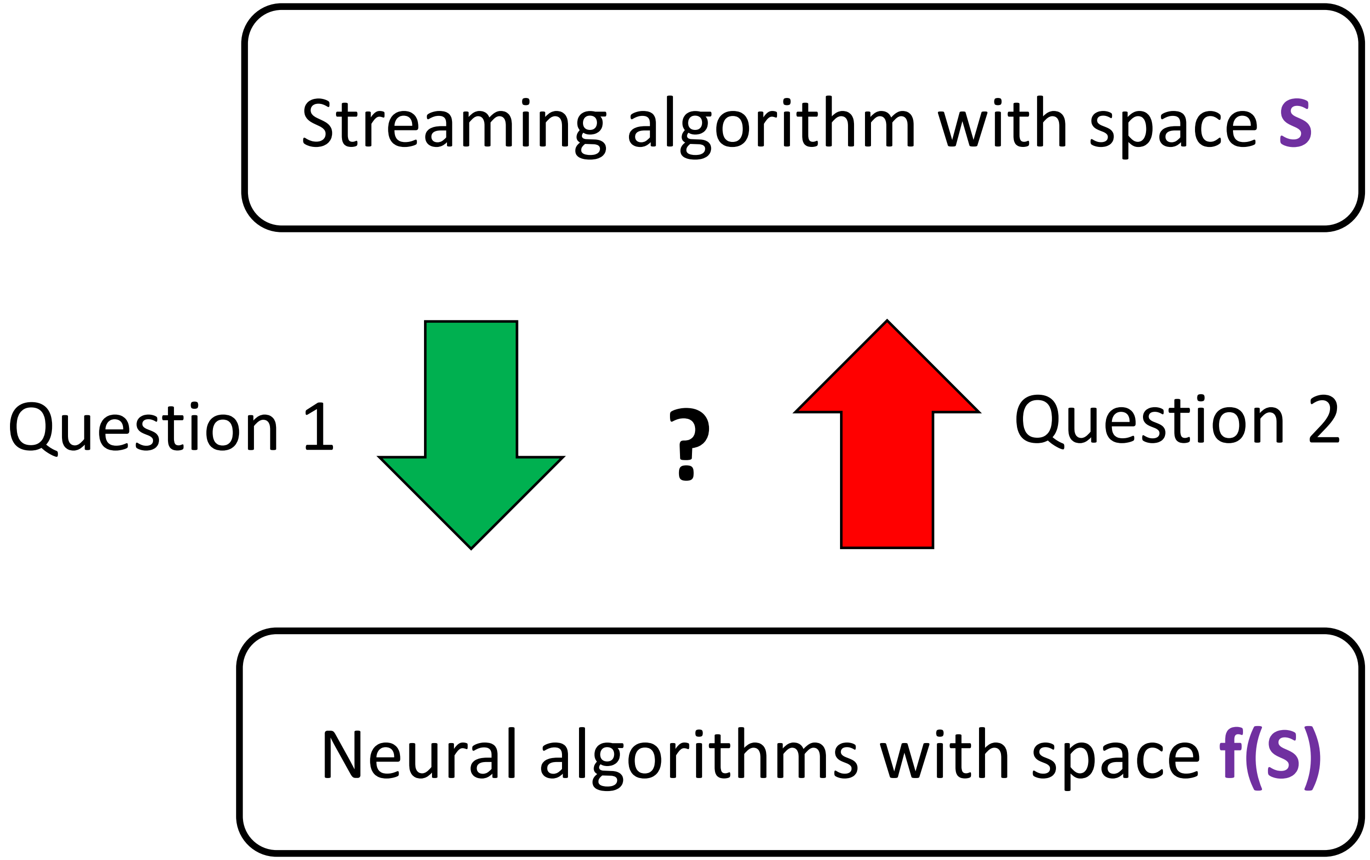
Streaming vs neural algorithms

Streaming algorithm with space **S**

Question 1 ↓ **?**

Neural algorithms with space **f(S)**

Streaming vs neural algorithms

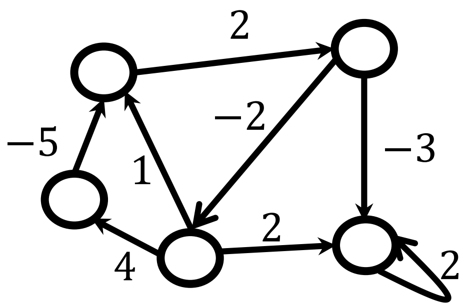


Spiking neural network [Maass, Neural Networks 97]

Directed weighted graph

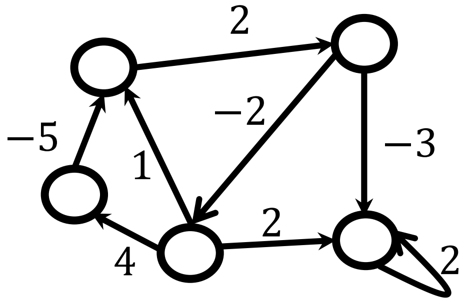
Nodes – neurons

Weighted edges – synaptic connections



Spiking neural network [Maass, Neural Networks 97]

<u>*Directed weighted graph*</u>

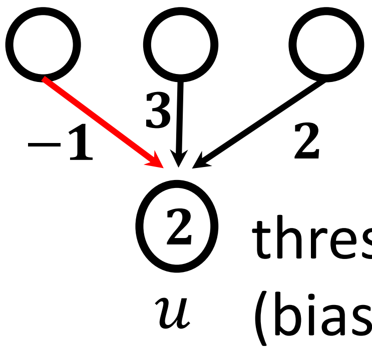


Nodes – neurons

Weighted edges – synaptic connections

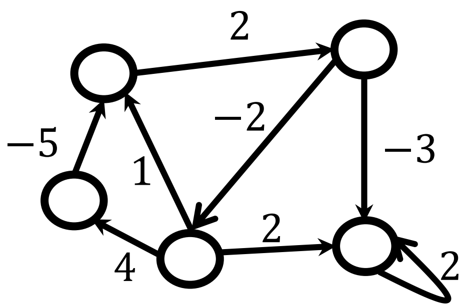
<u>Neuron</u>: (prob.)Threshold gate

- *Deterministic neuron*: Fires if $W - b \geq 0$
- *Probabilistic (spiking) neuron*: Fires with probability $\frac{1}{1+e^{-(W-b)}}$



threshold **b** (bias)

Neural network

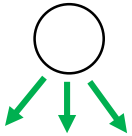


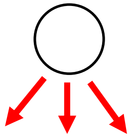
Network dynamics:

- Discrete synchronous rounds (Markov chain)
- Firing states in round $\boldsymbol{i}$ depends on firing states in round $\boldsymbol{i-1}$

Biological constraint:

- Two types of neurons: Excitatory (all positive) and Inhibitory (all negative)



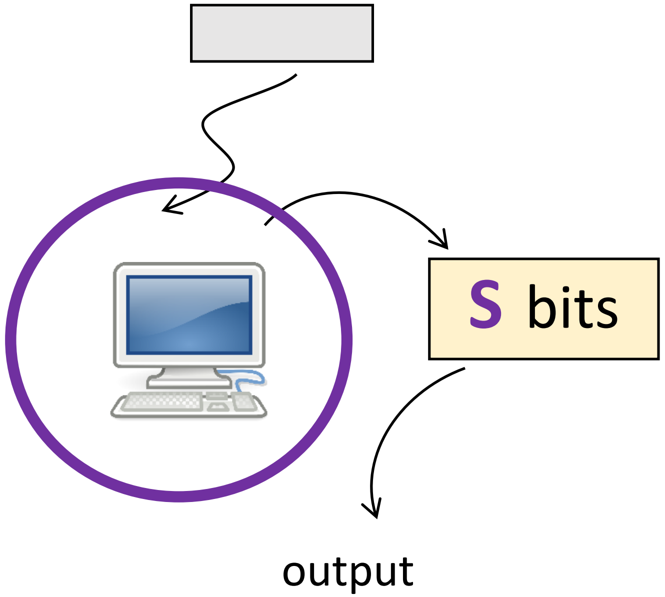


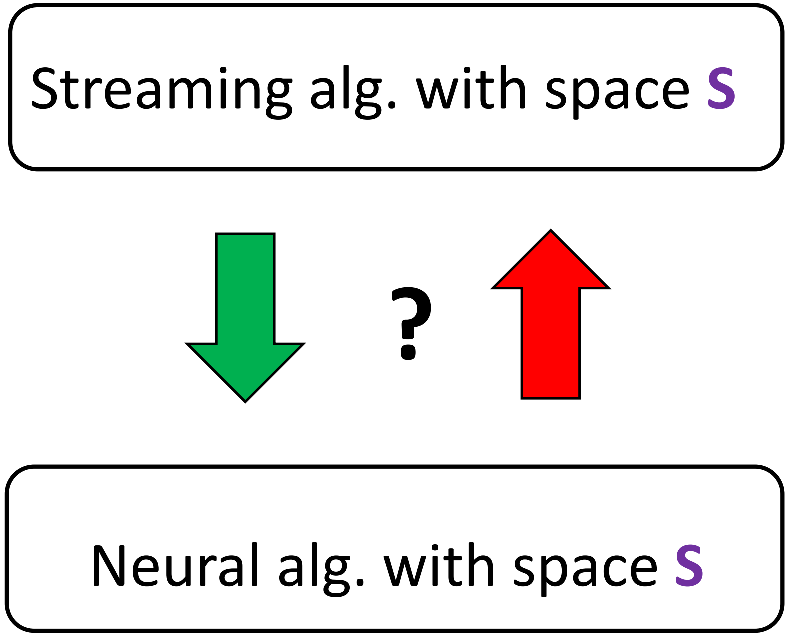
Randomness in neural networks

Two (potential) sources of randomness:

- Probabilistic neurons
- Random edge weights

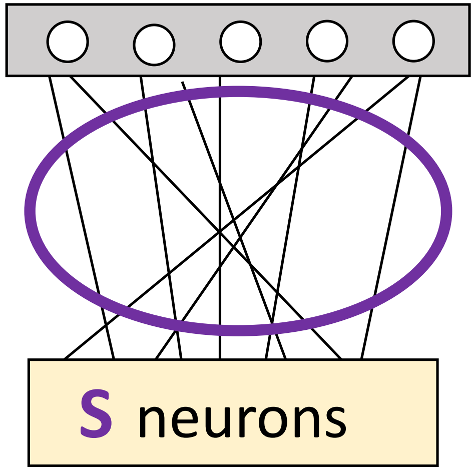
The challenge

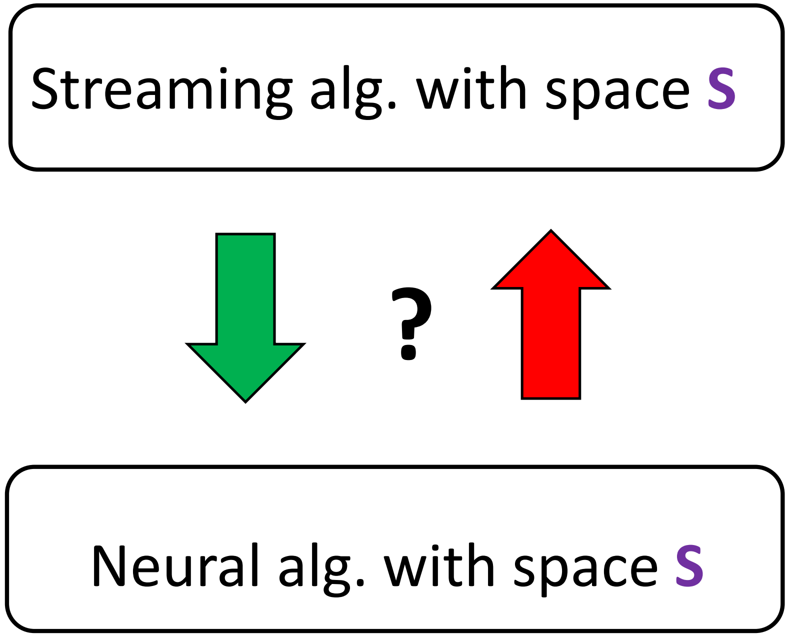




1. In the streaming model, do not pay for algorithm description

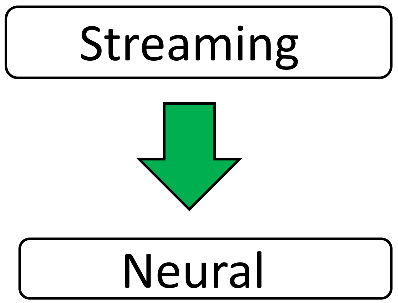
The challenge





2. In neural network, do not pay for random edge weights

Positive results



	Streaming space	Neural space
Approximate median $(1+\epsilon)$ approx. w.p $1-\delta$	$\tilde{O}(1/\epsilon)$ [Cormode, Muthukrishnan 05]	$\boldsymbol{\tilde{O}(1/\epsilon)}$
Distinct elements $(1+\epsilon)$ approx. w.p $1-\delta$	$\tilde{O}(1/\epsilon^2)$ [Blasiok 18]	$\boldsymbol{\tilde{O}(1/\epsilon^2)}$

Additional results:

Count-min-sketch, pairwise independent hash functions

$\tilde{O}$ hides polylog(n, δ)

Positive results: linear sketching

<u>Linear sketching algorithm</u>: the state of the algorithm at time $\boldsymbol{t}$ is a *linear function* of the updates seen up to time $\boldsymbol{t}$.

<u>Input</u> insertion/deletion elements in [n], (turnstile model)

- The state of the algorithm: $\mathbf{A} \cdot \bar{\mathbf{z}},$

  for $r \times n$ *sketching matrix* $\boldsymbol{A}$, frequencies vector $\bar{\boldsymbol{z}} \in \mathbb{Z}^{\boldsymbol{n}}$

- **<u>Examples:</u>** heavy hitters, ℓ_p estimation
- Li, Nguyen and Woodruff (STOC 14): streaming → linear sketching

Positive results: linear sketching

Theorem: For any integer sketching matrix **A** of size $\boldsymbol{r} \times \boldsymbol{n}$, each entry $|A \cdot \overline{x}| \leq \ell$, there exists a neural sketching network with $\boldsymbol{O(r \cdot log\, \ell)}$ auxiliary neurons

the output neurons encodes the state of the sketching algorithms $A \cdot \overline{z}$