# Hopfield Networks is All You Need

----------- Author: Hubert Ramsauer et al. -----------

----------- Presented by Tangrui Li -----------

tuo90515@temple.edu

# $x$ is all you need.

## Attention Is All You Need

**Ashish Vaswani*** Google Brain avaswani@google.com

**Noam Shazeer*** Google Brain noam@google.com

**Niki Parmar*** Google Research nikip@google.com

**Jakob Uszkoreit*** Google Research usz@google.com

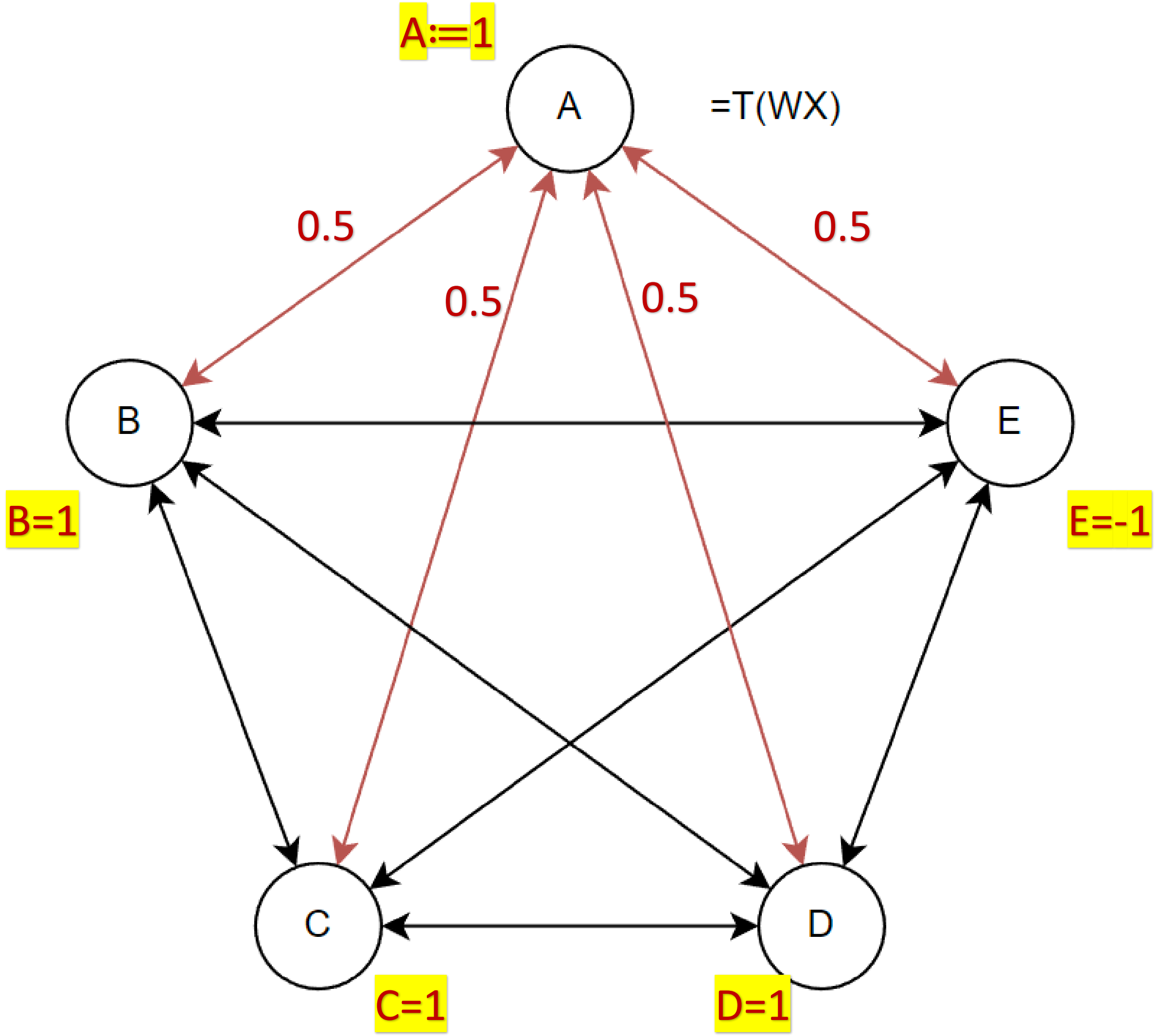
**Llion Jones*** Google Research llion@google.com

**Aidan N. Gomez*** † University of Toronto aidan@cs.toronto.edu

**Łukasz Kaiser*** Google Brain lukaszkaiser@google.com

**Illia Polosukhin*** ‡ illia.polosukhin@gmail.com

# Classical Binary Hopfield Networks




- The value of $A$, say $V_A$

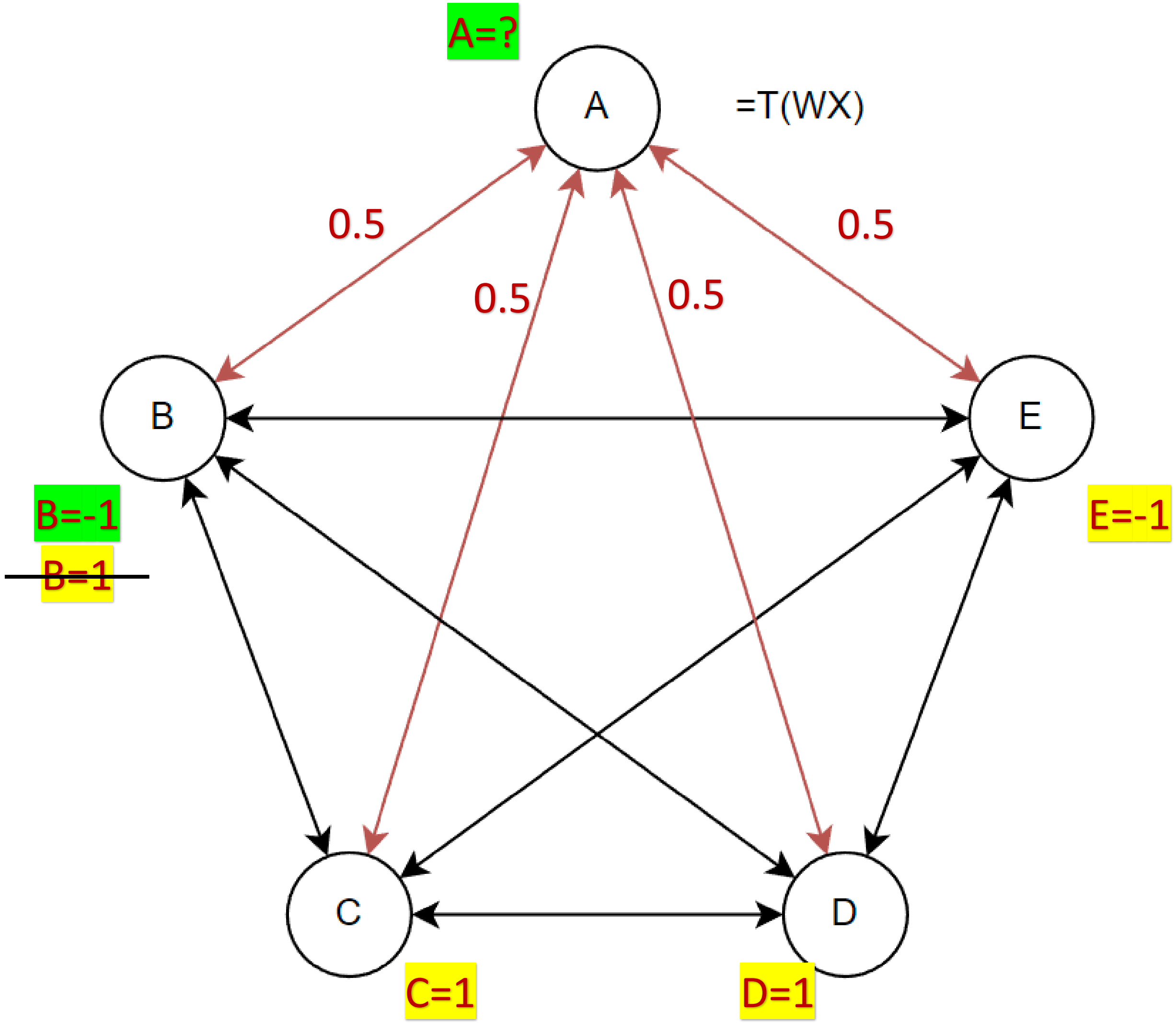
  $$= T(W_{AB}V_B + W_{AC}V_C + W_{AD}V_D + W_{AE}V_E)$$

  $$= T(0.5 + 0.5 + 0.5 - 0.5) = T(1)$$

- In which $T(\cdot)$ is the thresholding function (e.g., $Sgn(\cdot)$). As a result, $V_A = 1$, which is consistent with the **GIVEN** value.

- **The value of a part is calculated by the other values.**

# Classical Binary Hopfield Networks



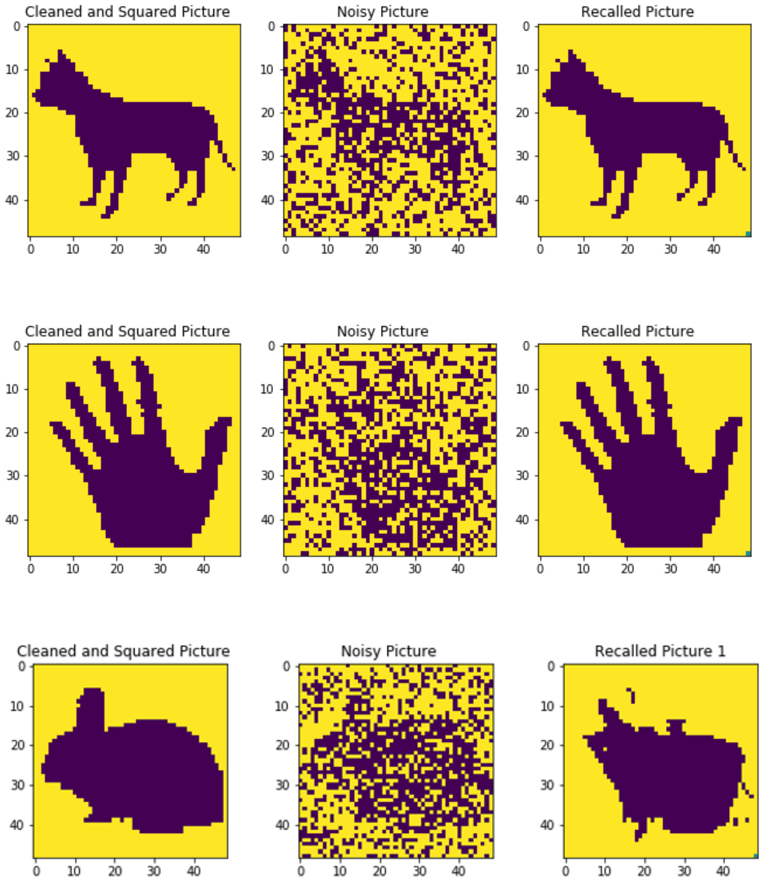

- When a part

$$= T(W_{AB}V$$

$$= T(-0.5$$

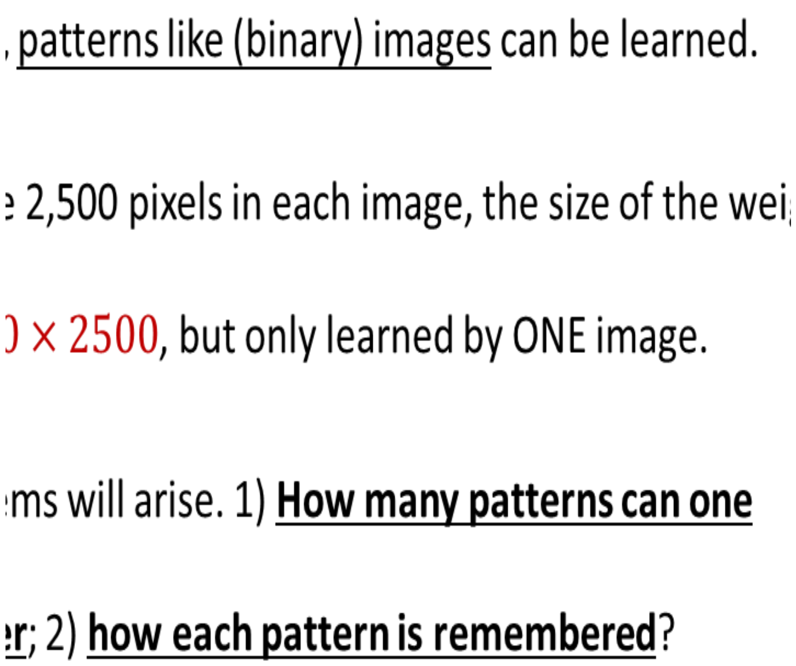
- It could be **re**

appropriate v



# Classical Binary Hopfield Networks

- More complicated, <u>patterns like (binary) images</u> can be learned.
- Note that there are 2,500 pixels in each image, the size of the weight matrix will be $2500 \times 2500$, but only learned by ONE image.
- Two natural problems will arise. 1) **<u>How many patterns can one network remember</u>**; 2) **<u>how each pattern is remembered</u>**?




https://github.com/nosratullah/hopfieldNeuralNetwork

# Global Stable Patterns

- Classical binary Hopfield networks are energy-based models (EBMs) with an energy function like:

$$E = -X^T W X$$

  which is a convex function. $\nabla E = -2WX$ (a linear system).

- When $\nabla E = 0$, there will be infinite one point with the **GLOBAL** minimal energy. But due to the "**binary**" requirement, this point might be unreachable, and so more than one patterns can be remembered.

# Local Stable Patterns

- Attractors (energy minimums) are not necessarily global minimums. Local minimums will also work.

$$X = [1, -1, -1, 1, \ldots, -1, -1]$$

If a part of $X$ is flipped, its energy should be larger when $X$ is an attractor.

- It is also possible for two local minimums overlap. If $[1, -1]$ is an attractor, the energy of $[-1, -1]$, $[1, 1]$ should be larger. But for $[-1, 1]$, the same case. So, for $[1, 1]$, it has two local energy minimum, which will make both patterns ($[1, -1], [-1, 1]$) not retrievable. As proved, only $\mathbf{0.138}\boldsymbol{N}$ ($N$ is the number of neurons) patterns can be remembered and retrieved with no errors, which is not a large number.

Krotov, Dmitry, and John J. Hopfield. "Dense associative memory for pattern recognition." *Advances in neural information processing systems* 29 (2016).

# Polynomial Energy Function

- The reason why $0.138N$ is the bound is because the gradient of the energy function is too "**flat**". So, polynomial energy functions are proposed:

$$E(X) = -(WX)^n$$

- And this limit is pushed to

$$\frac{1}{2(2n-3)!!} \cdot \frac{N^{n-1}}{\ln(N)}$$

which is an exponential function of $N$.

Krotov, Dmitry, and John J. Hopfield. "Dense associative memory for pattern recognition." *Advances in neural information processing systems* 29 (2016).

# Exponential Energy Function

- Naturally, people will think when $n \to \infty$, what will the energy function be like? As proved, an exponential energy function will work.

$$E(X) = -e^{WA}$$

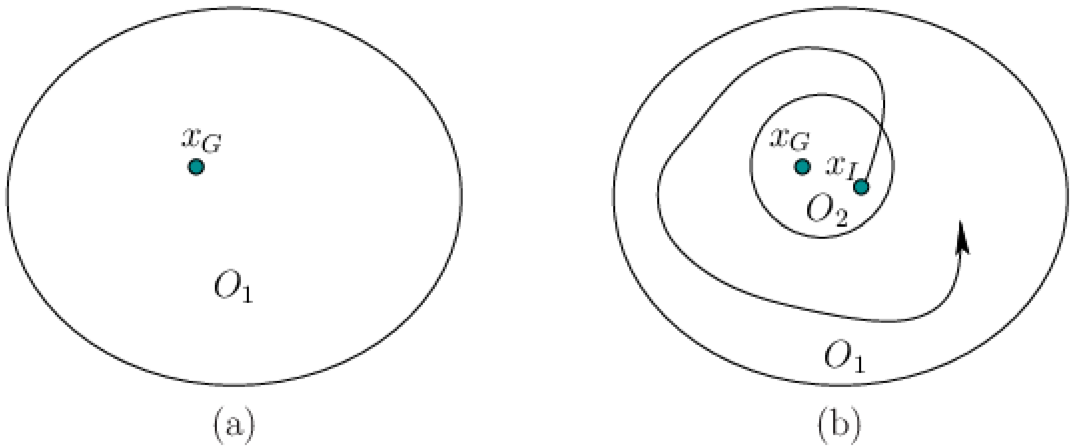
- This leads to the energy function ($lse$ function, log-sum-exp) used in this work, in which $\beta, c$ are constants, $W^T W$ is the regularization.

$$E(X) = -\beta^{-1}\log\left(\Sigma e^{\beta W X}\right) + W^T W + c$$

Krotov, Dmitry, and John J. Hopfield. "Dense associative memory for pattern recognition." *Advances in neural information processing systems* 29 (2016).

# Continuous Hopfield Networks

- In binary conditions, we define "attractor" by flipping each digit. But for continuous conditions, we need a new way to analyzing attractors.
- This leads to the Lyapunov analyzing.




http://lavalle.pl/planning/node801.html

# Hebbian Learning & Self-Attention

- Classical Hopfield Networks are often learned by Hebbian Learning. The idea is that "Neurons that fire together, wire together. – *Donald Hebb*". Say the weights prefers similar parts, which is similar with self-attention.
- Self attention.

$$Attention = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

$$Q = W_Q X, K = W_K X, V = W_V X$$

Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

# Hebbian Learning & Self-Attention

- In which $Q$ stands for Query, $K$ stands for Key, and $V$ stands for Value, which are linear transformations of $X$.
- This could be interpreted as "for several features of $X$ ($\boldsymbol{K}$), whether some features ($\boldsymbol{Q}$) are similar, and this similarity ($\boldsymbol{QK^T}/\sqrt{\boldsymbol{d_k}}$) can help get features of $X$ ($\boldsymbol{V}$)".
- $Attention \propto XX^T$, somehow like the Hebbian learning. In this paper, the parameter updating strategy is literally a simplified self-attention.

$$W_{new} = X \cdot \mathrm{softmax}(W^T X)$$

# Hopfield Networks in NNs

- When Hopfield networks (remember and retrieve patterns) and self-attention (distinct $Q, K, V$) are considered together, Hopfield NN layers are created.

$Y$ is used twice, with two weight matrices.
This is not only for retrieving, but also for transformation.

$$Z = \mathrm{softmax}(RW_K^T Y^T) Y W_V$$

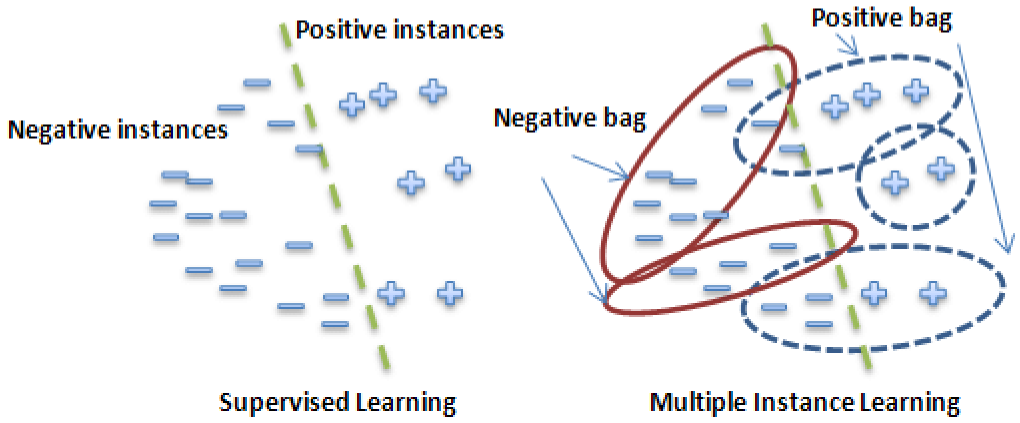
- In self-attention, we have $X$ and its three linear transformations $Q, K, V$. But here $R$ and $Y$ as two inputs can be different.

- Based on whether $R, Y$ are trainable, 3 types of Hopfield NN layers are proposed: 1) ***Hopfield***, with $R, Y$ both trainable, 2) ***Hopfield pooling***, with $Y$ trainable, $R$ fixed, 3) ***Hopfield layer***, with $R$ trainable, $Y$ fixed.

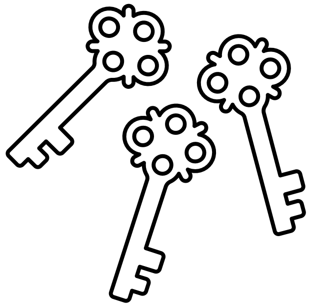
# Hopfield Networks in NNs

- ***Hopfield***, with $R, Y$ both trainable. This is self-attention.
- ***Hopfield pooling***, with $Y$ trainable, $R$ fixed. In this case, queries are fixed, if more inputs are similar with the queries, the result will be an average of these queries. (pattern pooling)
- ***Hopfield layer***, with $R$ trainable, $Y$ fixed. In this case, the keys are fixed, which can be pre-learned (fixed) patterns, or simply instances, and this layer will perform like KNN.

# Experiments



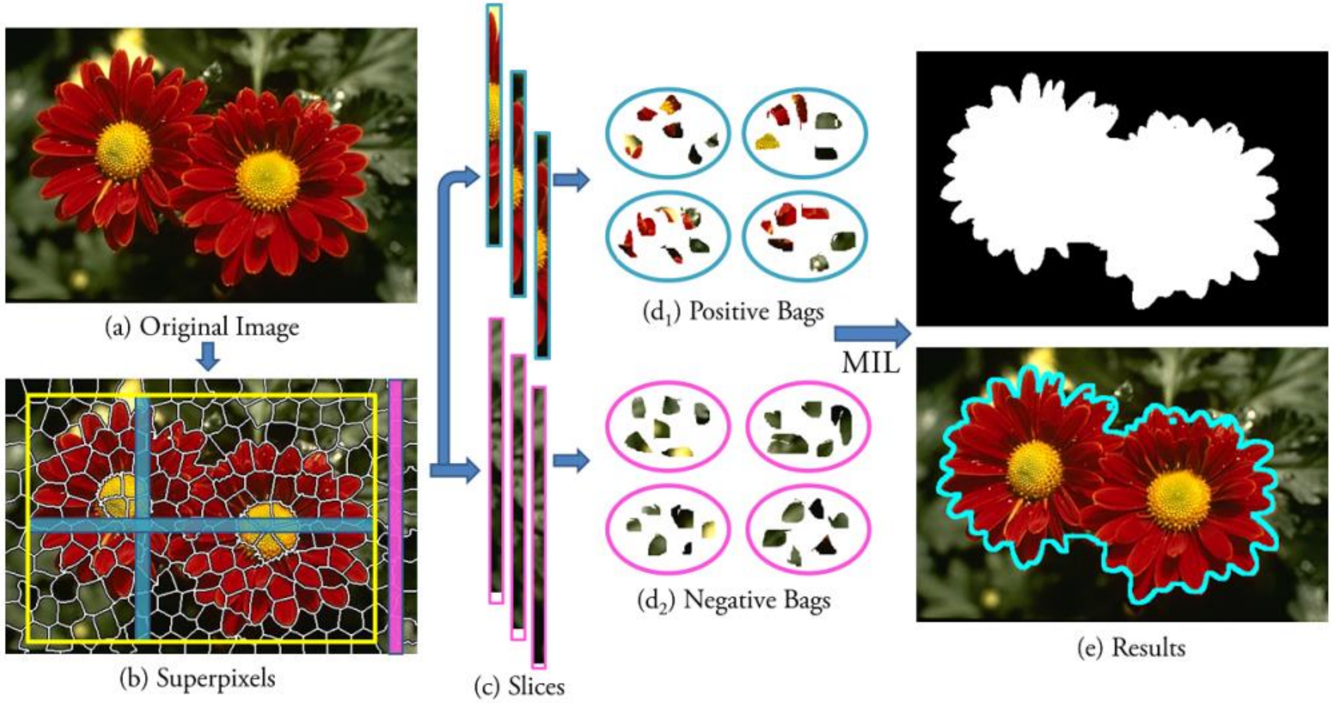

- Multi instance learning (MIL) classification.
- MIL is a type of questions with instances capsuled as bags. There are many instances, but they are not assumed uniformly distributed in the sample space but grouped in bags.
- MIL classification requires to 1) find the bag of one label and 2) the instance in that bag with the label.
- E.g., keychains. To open a lock, which keychain and which key will I need?



https://www.researchgate.net/figure/Comparison-of-supervised-learning-and-multi-instance-learning-The-classifier-is-learned_fig1_224265580

# Experiments

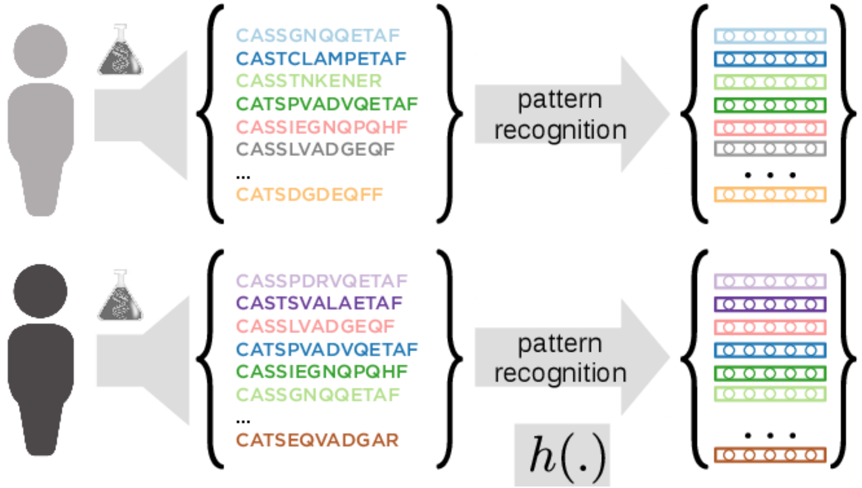
- This case is also applied for semantic segmentation tasks.
- The major challenge of MIL tasks is that these bags share different distributions and so finding a unified model modeling all instances (usually a lot instances) is very hard.




Wu, Jiajun, et al. "Milcut: A sweeping line multiple instance learning paradigm for interactive image segmentation." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014.

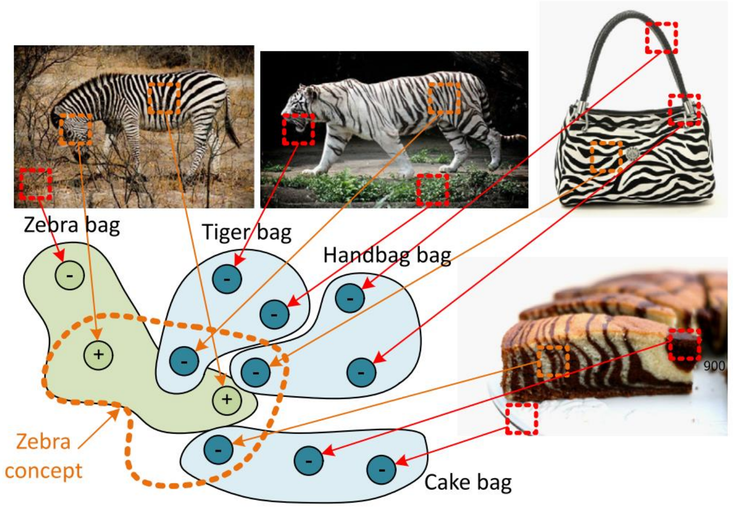
# Immune Repertoire Classification (big data)

- This task needs to find few patterns from a large number of sequences, and different sequences are enclosed in bags (each individual). This dataset has 300,000 instances in each repertoire.
- With sequences restored, patterns are queried and learned. The work proposed achieves AUC 0.832+0.022, compared with 0.825+0.022 from SVM.




M. Widrich, B. Schäfl, M. Pavlovi´c, H. Ramsauer, L. Gruber, M. Holzleitner, J. Brandstetter, G. K. Sandve, V. Greiff, S. Hochreiter, and G. Klambauer. Modern Hopfield networks and attention for immune repertoire classification. ArXiv, 2007.13505, 2020a.

# Image Classification (small data)




- An image classification problem.
- With 1,391 elephant images with 230 patterns, 1,320 fox images with 230 patterns, 1,220 tiger images with 230 features, and 2,002 UCSB breast cancer images with 58 objects.

| Method | tiger | fox | elephant | UCSB |
|---|---|---|---|---|
| Hopfield (ours) | $\mathbf{91.3 \pm 0.5}$ | $64.05 \pm 0.4$ | $\mathbf{94.9 \pm 0.3}$ | $\mathbf{89.5 \pm 0.8}$ |
| Path encoding (Küçükaşcı & Baydoğan, 2018) | $91.0 \pm 1.0^{a}$ | $71.2 \pm 1.4^{a}$ | $94.4 \pm 0.7^{a}$ | $88.0 \pm 2.2^{a}$ |
| MInD (Cheplygina et al., 2016) | $85.3 \pm 1.1^{a}$ | $70.4 \pm 1.6^{a}$ | $93.6 \pm 0.9^{a}$ | $83.1 \pm 2.7^{a}$ |
| MILES (Chen et al., 2006) | $87.2 \pm 1.7^{b}$ | $\mathbf{73.8 \pm 1.6}^{a}$ | $92.7 \pm 0.7^{a}$ | $83.3 \pm 2.6^{a}$ |
| APR (Dietterich et al., 1997) | $77.8 \pm 0.7^{b}$ | $54.1 \pm 0.9^{b}$ | $55.0 \pm 1.0^{b}$ | — |
| Citation-kNN (Wang, 2000) | $85.5 \pm 0.9^{b}$ | $63.5 \pm 1.5^{b}$ | $89.6 \pm 0.9^{b}$ | $70.6 \pm 3.2^{a}$ |
| DD (Maron & Lozano-Pérez, 1998) | $84.1^{b}$ | $63.1^{b}$ | $90.7^{b}$ | — |

Carbonneau, Marc-André, et al. "Multiple instance learning: A survey of problem characteristics and applications." Pattern Recognition 77 (2018): 329-353.

# Small Datasets on UCI MLR Benchmarks

- Hopfield networks can also work well on small datasets, since it can simply remember each data instance and perform KNN-like functionalities.
- This work collects 75 small datasets (within 1,000 instances) and 45 large datasets (larger than 1,000 instances) on UCI MLR.
- Since there are many datasets for each ML schema to test, the author **ranks** each method as the following. Say Hopfield related methods are the best.

| Method | avg. rank diff. | $p$-value |
|---|---|---|
| Hopfield (ours) | $\mathbf{-3.92}$ | — |
| SVM | $-3.23$ | 0.15 |
| SNN | $-2.85$ | 0.10 |
| RandomForest | $-2.79$ | 0.05 |
| ... | ... | ... |
| Stacking | 8.73 | 1.2e$-$11 |

# Discussion

- Patterns are remembered and retrieved in a distributed manner. Compositionality issues.
- Though the author provides an efficient way of updating (exponential retrieving error decreasing while training) the parameters of Hopfield layers, it is still in the training process of a NN, not real-time.
- When $R, Y$ are not both trainable, fixed pre-learned data needs to be provided by the user. Can only be used to remember individual training instances due to interpretability issues.

# Toward a Broad AI

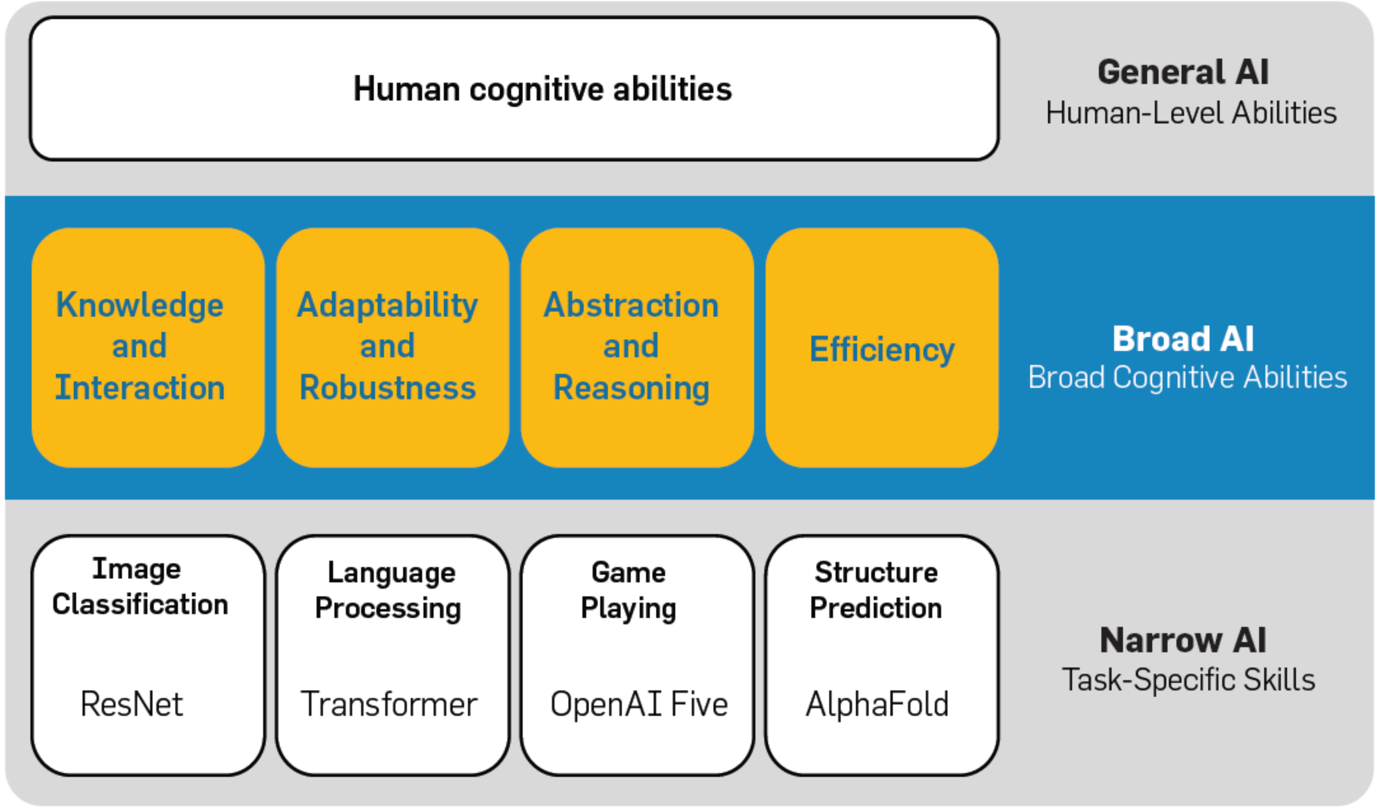
----------- Author: Sepp Hochreiter et al. -----------

----------- Presented by Tangrui Li -----------

tuo90515@temple.edu

# Definition

- Definition, a more abstract way with considerably enhanced and broader capabilities for skill acquisition and problem solving.
- Still capability-oriented.
- But different emphasis.



# Properties Should Equipped

- Knowledge transfer and interaction.
- Adaptability (through few-shot learning and self-supervising) and robustness.
- Abstraction and advanced reasoning.
- Efficiency.

# Properties Should Equipped

- By its own sensory perceptions.
- Context and short/long term memory (through the Hopfield networks mentioned).
- GNN (neural-symbolic reasoning).