HOPFIELD NETWORKS IS ALL YOU NEED

Deep Learning: Classics and Trends

Johannes Brandstetter
Amsterdam Machine Learning Lab
Institute for Machine Learning Linz

Universiteit van Amsterdam

Johannes Kepler University Linz

HOPFIELD NETWORKS IS ALL YOU NEED

Hubert Ramsauer[*] **Bernhard Schäfl**[*] **Johannes Lehner**[*] **Philipp Seidl**[*]
Michael Widrich[*] **Thomas Adler**[*] **Lukas Gruber**[*] **Markus Holzleitner**[*]
Milena Pavlović[‡,§] **Geir Kjetil Sandve**[§] **Victor Greiff**[‡] **David Kreil**[†]
Michael Kopp[†] **Günter Klambauer**[*] **Johannes Brandstetter**[*] **Sepp Hochreiter**[*,†]

[*]ELLIS Unit Linz, LIT AI Lab, Institute for Machine Learning,
Johannes Kepler University Linz, Austria
[†]Institute of Advanced Research in Artificial Intelligence (IARAI)
[‡]Department of Immunology, University of Oslo, Norway
[§]Department of Informatics, University of Oslo, Norway

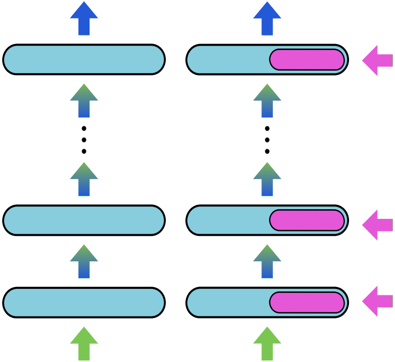
ACCEPTED AT ICLR conference 2021

Overview

- How to equip Deep Learning architectures with memories:
  - Motivation for continuous modern Hopfield Networks
  - Properties of continuous modern Hopfield Networks
  - Relation to Transformers
- New layers for Deep Learning architectures
  - New Hopfield layers
- New Hopfield layers at work

Deep Learning with Memories

- The goal is to integrate **associative memories** into **Deep Learning architectures**.
- Deep Learning that goes beyond convolutional and recurrent networks.



Deep Learning with Associative Memories

- Association of sets
- Pattern search in sets
- Pooling operations
- **Memories (LSTM, GRU)**
- Learning prototypes
- **Transformer attention**
- Sequence-to-sequence
- Point sets
- **Multiple instance learning**
- k-nearest neighbor of set

Modern Hopfield Networks as tool to equip Deep Learning architectures with memory.

Classical Hopfield Networks

- **Hopfield Networks** (Hopfield 1982)
- N binary patterns $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ with $\boldsymbol{x}_i \in \{-1, 1\}^d$
- Weight matrix $\boldsymbol{W}$ stores the N binary patterns: $\boldsymbol{W} = \sum_i^N \boldsymbol{x}_i \boldsymbol{x}_i^T$
- State (query) pattern $\boldsymbol{\xi} \in \{-1, 1\}^d$.
- Update rule $\boldsymbol{\xi}^{\text{new}} = \operatorname{sgn}(\boldsymbol{W}\boldsymbol{\xi} - \boldsymbol{b})$ with threshold $\boldsymbol{b}$ minimizes the energy function:

$$E = -\frac{1}{2}\boldsymbol{\xi}^T \boldsymbol{W} \boldsymbol{\xi} + \boldsymbol{\xi}^T \boldsymbol{b} = -\frac{1}{2}\sum_{i=1}^N (\boldsymbol{\xi}^T \boldsymbol{x}_i)^2 + \boldsymbol{\xi}^T b \tag{1}$$

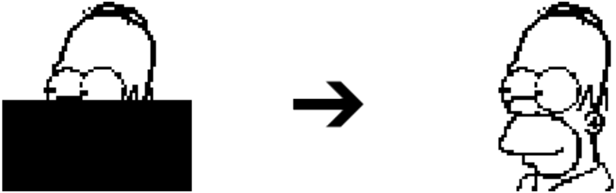
- Convergence is reached if $\boldsymbol{\xi}^{\text{new}} = \boldsymbol{\xi}$.

Classical Hopfield Networks

Weight matrix $\boldsymbol{W} = \sum_{i}^{1} \boldsymbol{x}_i \boldsymbol{x}_i^T$

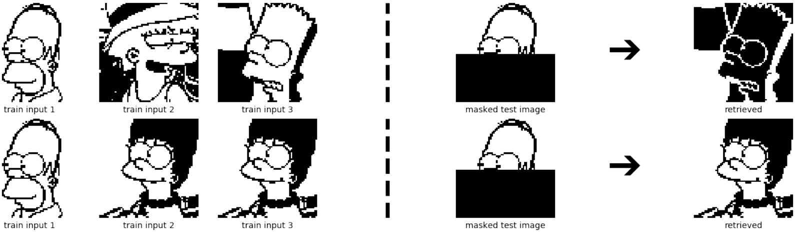
to store pattern

Update rule $\boldsymbol{\xi}^{\text{new}} = \operatorname{sgn}\left(\boldsymbol{W}\boldsymbol{\xi} - \boldsymbol{b}\right)$ to retrieve pattern



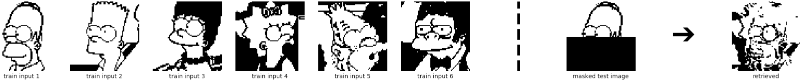
Classical Hopfield Networks

Undesired retrieval



Classical Hopfield Networks

Spurious minima: patterns are correlated

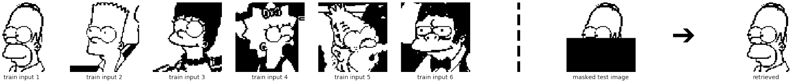


Modern Hopfield Networks

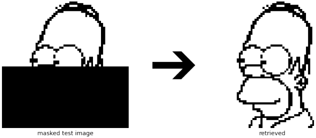
- **Krotov & Hopfield (2016)**
- $E = \sum_{i}^{N} F(\boldsymbol{\xi}^T \boldsymbol{x}_i)$, where $F(z) = z^a$ is the interaction function.
- For $a = 2$, we obtain the classical Hopfield Networks:
  $E = \frac{1}{2} \sum_{i}^{N} (\boldsymbol{\xi}^T \boldsymbol{x}_i)^2$
- **Storage capacity** is **polynomial in d**:
  - Storage means that patterns are fixed points of the update rule.

Modern Hopfield Networks

- **Demircigil et al. (2017)**
- $E = \sum_{i}^{N} F(\boldsymbol{\xi}^T \boldsymbol{x}_i)$, where $F(z) = \exp(z)$ is the interaction function.
- **Storage capacity** is **exponential in d**.
- **Convergence** / **retrieval** after one update.

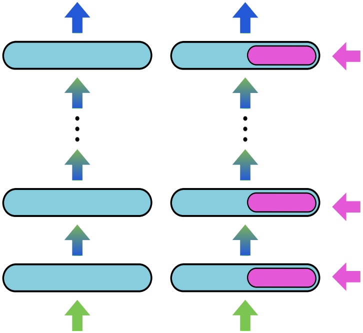


Modern Hopfield Networks



New Energy Function

- Modern Hopfield Networks are **binary**.
- We want to extend them towards **Continuous Hopfield Networks**:
  - **Differentiability** for gradient descent in Deep Networks.
  - **Retrieval with one update** to activate the layer.
  - **High storage capacity** for complex systems.



New Energy Function / New Update Rule

$$\boxed{E = -\mathrm{lse}(\beta, \boldsymbol{X}^T \boldsymbol{\xi}) + \frac{1}{2}\boldsymbol{\xi}^T\boldsymbol{\xi} + \beta^{-1}\log N + \frac{1}{2}M^2}$$

- N **stored (key) patterns** $\boldsymbol{x}_i \in \mathbb{R}^d$ are from a d-dimensional space
- Pattern matrix $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$
- Largest pattern $M = \max_i ||\boldsymbol{x}_i||$
- **State (query) pattern** $\boldsymbol{\xi}$
- $\mathrm{lse}(\beta, \boldsymbol{a}) = \beta^{-1} \log \left(\sum_{i=1}^{N} \exp(\beta a_i) \right)$

$$\boxed{\boldsymbol{\xi}^{\mathrm{new}} = f(\boldsymbol{\xi}) = \boldsymbol{X}\,\mathrm{softmax}(\beta \boldsymbol{X}^T \boldsymbol{\xi})}$$

Properties of New Energy Function

- The new energy function generalizes the energy of binary modern Hopfield Networks (Demircigil et al. 2017) to continuous valued patterns.
- Important properties are kept:
  - **Exponential storage capacity** (Theorem 3 in the paper)
  - **Retrieval after one update** (Theorem 4 in the paper)
- Additionally, **global convergence to a local minimum** proven (Theorem 2 in the paper).

Convergence to Stationary Points

Theorem of Convergence to Stationary Point.

For the iteration with the update rule we have $\mathrm{E}\left(\boldsymbol{\xi}^t\right) \to \mathrm{E}(\boldsymbol{\xi}^*) = \mathrm{E}^*$ *as* $t \to \infty$*, for some stationary point* $\boldsymbol{\xi}^*$*. Furthermore,* $\left\|\boldsymbol{\xi}^{t+1} - \boldsymbol{\xi}^t\right\| \to 0$ *and either* $\{\boldsymbol{\xi}^t\}_{t=0}^{\infty}$ *converges or, in the other case, the set of limit points of* $\{\boldsymbol{\xi}^t\}_{t=0}^{\infty}$ *is a connected and compact subset of* $\mathcal{L}(\mathrm{E}^*)$*, where* $\mathcal{L}(a) = \{\boldsymbol{\xi} \in \mathcal{L} \mid \mathrm{E}(\boldsymbol{\xi}) = a\}$ *and* $\mathcal{L}$ *is the set of stationary points of the iteration. If* $\mathcal{L}(\mathrm{E}^*)$ *is finite, then any sequence* $\{\boldsymbol{\xi}^t\}_{t=0}^{\infty}$ *generated by the iteration converges to some* $\boldsymbol{\xi}^* \in \mathcal{L}(\mathrm{E}^*)$*.*

- **All limit points** of any sequence generated by the iteration
  $\boldsymbol{\xi}^{\text{new}} = f(\boldsymbol{\xi}) = \boldsymbol{X}\text{softmax}(\beta \boldsymbol{X}^T \boldsymbol{\xi})$
  are **stationary points** (**local minima or saddle points**) of the energy function E.

Exponential storage capacity

- First we have to define, **storing**/**retrieving** patterns with a modern Hopfield Network:

Definition of Retrieved and Stored Patterns.

We assume that around every pattern $\boldsymbol{x}_i$ a sphere S_i is given. We say $\boldsymbol{x}_i$ *is stored* if there is a single fixed point $\boldsymbol{x}_i^* \in \mathrm{S}_i$ to which all points $\boldsymbol{\xi} \in \mathrm{S}_i$ converge, and $\mathrm{S}_i \cap \mathrm{S}_j = \emptyset$ for $i \neq j$. We say $\boldsymbol{x}_i$ *is retrieved* for a given ϵ if iteration (update rule) gives a point $\tilde{\boldsymbol{x}}_i$ that is at least ϵ-close to the single fixed point $\boldsymbol{x}_i^* \in \mathrm{S}_i$. The retrieval error is $\|\tilde{\boldsymbol{x}}_i - \boldsymbol{x}_i\|$.

Exponential storage capacity

Theorem of Exponential Storage Capacity.

We assume a failure probability $0 < p \leq 1$ *and randomly chosen patterns on the sphere with radius* $M := K\sqrt{d-1}$. *We define* $a := \frac{2}{d-1}(1 + \ln(2\beta K^2 p(d-1)))$, $b := \frac{2K^2\beta}{5}$, *and* $c := \frac{b}{W_0(\exp(a + \ln(b)))}$, *where* W_0 *is the upper branch of the Lambert* W *function, and ensure* $c \geq \left(\frac{2}{\sqrt{p}}\right)^{\frac{4}{d-1}}$. *Then with probability* $1-p$, *the number of random patterns that can be stored is*

$$N \geq \sqrt{p}\, c^{\frac{d-1}{4}} .$$

Therefore it is proven for $c \geq 3.1546$ *with* $\beta = 1$, $K = 3$, $d = 20$ *and* $p = 0.001$ *(*$a + \ln(b) > 1.27$*) and proven for* $c \geq 1.3718$ *with* $\beta = 1$, $K = 1$, $d = 75$, *and* $p = 0.001$ *(*$a + \ln(b) < -0.94$*).*

- **Exponential storage capacity** in the dimension d of the patterns ($\boldsymbol{x}_i \in \mathbb{R}^d$)

Retrieval with one update

- The update rule retrieves patterns **with one update** for well separated patterns, that is, patterns with large Δ_i:

Theorem of Retrieval with One Update.

With query $\boldsymbol{\xi}$, after one update the distance of the new point $f(\boldsymbol{\xi})$ to the fixed point $\boldsymbol{x}_i^*$ is exponentially small in the separation Δ_i. The precise bounds using the Jacobian $\mathrm{J} = \frac{\partial f(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}$ and its value J^m in the mean value theorem are:

$$
\begin{aligned}
\| f(\boldsymbol{\xi}) \ - \ \boldsymbol{x}_i^* \| \ &\leq \| \mathrm{J}^m \|_2 \ \| \boldsymbol{\xi} \ - \ \boldsymbol{x}_i^* \| \ , \\
\| \mathrm{J}^m \|_2 \ &\leq 2 \ \beta \ N \ M^2 \ (N - 1) \\
&\qquad \exp(- \ \beta \ (\Delta_i \ - \ 2 \ \max\{ \| \boldsymbol{\xi} \ - \ \boldsymbol{x}_i \|, \| \boldsymbol{x}_i^* \ - \ \boldsymbol{x}_i \| \} \ M)) \ .
\end{aligned}
$$

For given ϵ and sufficient large Δ_i, we have $\left\| f(\boldsymbol{\xi}) \ - \ \boldsymbol{x}_i^* \right\| < \epsilon$, that is, retrieval with one update.

$$
\boxed{\Delta_i := \min_{j, j \neq i} \left(\boldsymbol{x}_i^T \boldsymbol{x}_i \ - \ \boldsymbol{x}_i^T \boldsymbol{x}_j \right) \ = \ \boldsymbol{x}_i^T \boldsymbol{x}_i \ - \ \max_{j, j \neq i} \boldsymbol{x}_i^T \boldsymbol{x}_j}
$$

- The **retrieval error decreases exponentially** with the separation Δ_i (Theorem 5 in the paper).

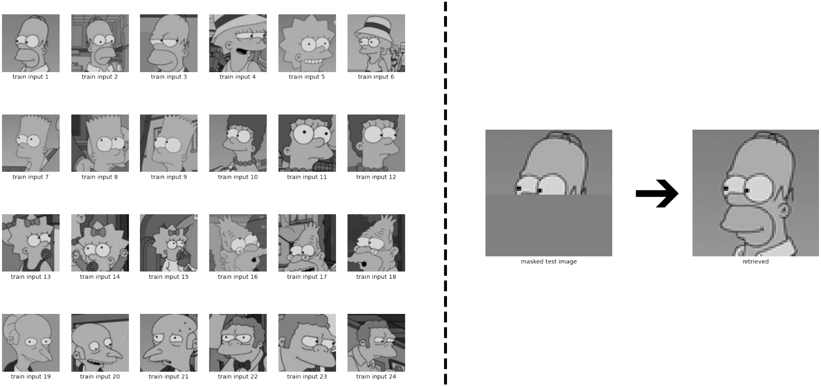
Global Fixed Point and Metastable States

- If **no pattern $x_i \in \mathbb{R}^d$ is well separated**, then the iterate converges to a **global fixed point** close to the **arithmetic mean** of the vectors (**softmax is close to uniform**).
- **Metastable states**:
  - Some vectors are **similar to each other**,
  - but well separated from other vectors.
  - Fixed point near the similar patterns (metastable state).
  - Iterates that start near the metastable state converge to it.

New Modern Hopfield Networks

$$\boxed{\boldsymbol{\xi}^{\text{new}} = f(\boldsymbol{\xi}) = \boldsymbol{X}\,\text{softmax}(\beta \boldsymbol{X}^T \boldsymbol{\xi})}$$

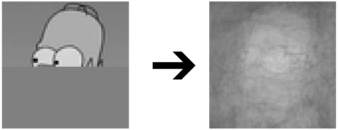
$$\boxed{\beta = 8}$$



New Modern Hopfield Networks

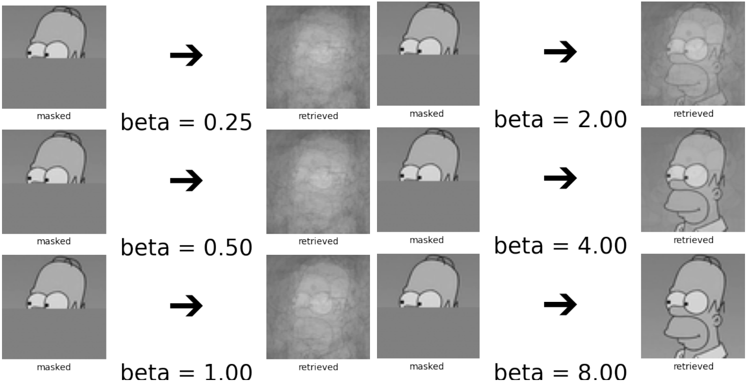
$$\boxed{\boldsymbol{\xi}^{\text{new}} = f(\boldsymbol{\xi}) = \boldsymbol{X}\,\text{softmax}(\beta \boldsymbol{X}^T \boldsymbol{\xi})}$$

$$\boxed{\beta = 0.5}$$



masked test image → retrieved

New Modern Hopfield Networks



New Update Rule = Transformer Attention

- Hopfield update:

$$\boxed{\boldsymbol{\xi}^{\text{new}} = f(\boldsymbol{\xi}) = \boldsymbol{X}\,\text{softmax}(\beta \boldsymbol{X}^T \boldsymbol{\xi})}$$

- Transformer attention:

$$\boxed{\text{softmax}(1/\sqrt{d_k}\,\boldsymbol{Q}\boldsymbol{K}^T)\boldsymbol{V}}$$

$$\boldsymbol{y}_i \in \mathbb{R}^{d_y}$$
$$\boldsymbol{x}_i = \boldsymbol{W}_K^T \boldsymbol{y}_i \in \mathbb{R}^{d_k},\ \boldsymbol{W}_K \in \mathbb{R}^{d_y \times d_k}$$
$$\boldsymbol{\xi}_i = \boldsymbol{Q}_K^T \boldsymbol{y}_i \in \mathbb{R}^{d_k},\ \boldsymbol{W}_Q \in \mathbb{R}^{d_y \times d_k}$$
$$\boldsymbol{Y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N)^T \in \mathbb{R}^{N \times d_y}$$
$$\boldsymbol{X}^T = \boldsymbol{K} = \boldsymbol{Y}\boldsymbol{W}_K \in \mathbb{R}^{N \times d_k}$$
$$\boldsymbol{\Xi}^T = \boldsymbol{Q} = \boldsymbol{Y}\boldsymbol{W}_Q \in \mathbb{R}^{N \times d_k}$$
$$\boldsymbol{V} = \boldsymbol{Y}\boldsymbol{W}_K\boldsymbol{W}_V = \boldsymbol{X}^T\boldsymbol{W}_V \in \mathbb{R}^{N \times d_v}$$
$$\boldsymbol{W}_V \in \mathbb{R}^{d_k \times d_v}$$
$$\beta = \frac{1}{\sqrt{d_k}}$$

softmax $\in \mathbb{R}^N$ is a row vector

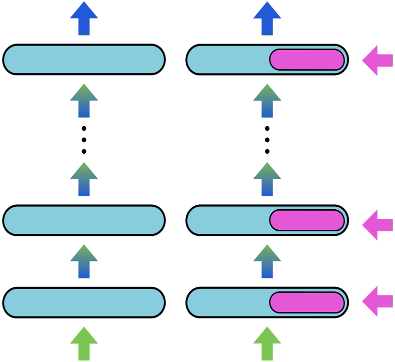
$\boldsymbol{y}_i \in \mathbb{R}^d$ is a data vector

$\boldsymbol{x}_i \in \mathbb{R}^d$ is stored (key) pattern

$\boldsymbol{\xi}_i \in \mathbb{R}^d$ is state (query) pattern

Deep Learning with Memories

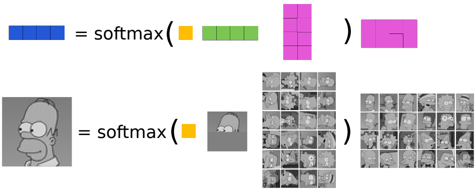
- The goal is to integrate **associative memories** into **Deep Learning architectures**.
- With **Modern continuous Hopfield Networks** we now have a tool to do that.

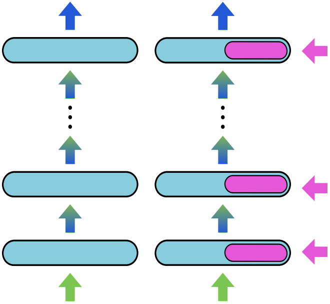


Deep Learning with Memories

- The goal is to integrate **associative memories** into **Deep Learning architectures**.
- With **Modern continuous Hopfield Networs** we now have a tool to do that.

$$\boldsymbol{Z} = \mathrm{softmax}\left(\beta \, \boldsymbol{R} \, \boldsymbol{Y}^T\right) \boldsymbol{Y}$$



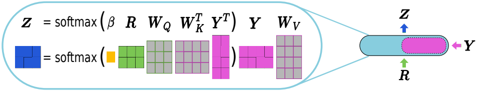


New Hopfield Layers

- **Hopfield**:

  Propagate R and Y

  Transformer attention

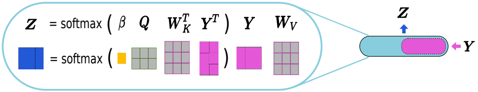


- **HopfieldPooling**:

  Propagate Y

  Multiple instances

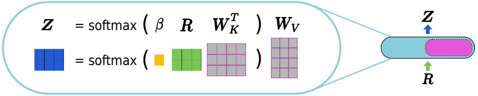
  Sequences (LSTMs)



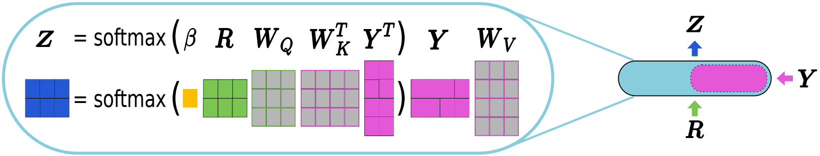
- **HopfieldLayer**:

  Propagate R

  SVM, k-NN

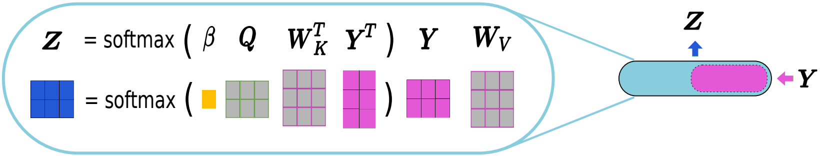


Layer Hopfield



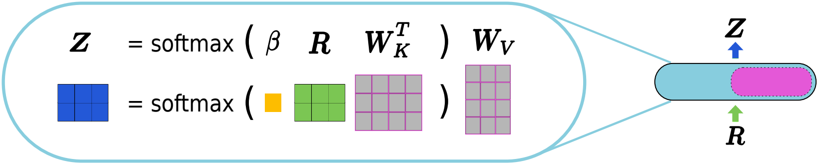
- Association of **raw state (query) patterns** R and **raw stored (key) patterns** Y
- Association of two sets R and Y
- This layer works for:
  - Transformer attention (associates keys and queries)
  - Sequence-to-sequence learning
  - Point set operations
  - Retrieval-based methods

Layer Hopfield Pooling



- Queries Q and **raw stored (key) patterns Y**
- Result is mapped by W_V.
- Fixed pattern search of Q in Y:
  - **Pooling** of Y guided by Q.
  - **Memories** of sequence or set Y
- This layer can potentially substitute:
  - Pooling
  - LSTMs / GRUs applied to Y
  - Multiple instance learning, patterns search
  - 2-D position encoding: convolutions

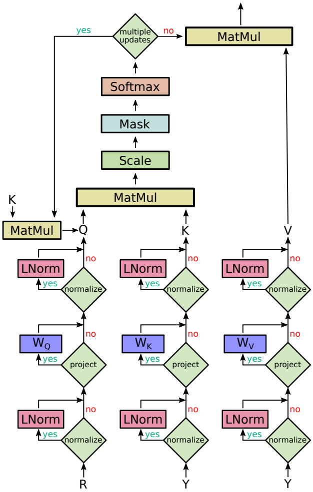
Layer HopfieldLayer



- **Raw state (query) patterns** R and stored patterns $\boldsymbol{W}_K$
- This layer can potentially substitute:
  - k**-nearest neighbor** if $\boldsymbol{W}_K$ are training data
  - **SVM** if prototypes $\boldsymbol{W}_K$ are support vectors
  - **Similarity-based** if $\boldsymbol{W}_K$ are training data
  - **Learning vector quantization (LVQ)** if $\boldsymbol{W}_K$ are the cluster centers

Additional Functionalities

- **Multiple updates** to control how precise fixed points are found.
- **Variable β**: kind of fixed point / size of metastable states:
  - β controls over how many patterns is averaged.
  - Relevant in combination with the learning rate to steer learning.
- **Controlling the storage capacity** via the associative dimension.
- **Pattern normalization** by layernorm (controls fixed point dynamics).



Experiments

- We have already successfully applied Hopfield layers to a **wide range of tasks**:
  - Natural Language Processing
  - Multiple instance learning problems (MIL)
  - Small classification tasks (UCI)
  - Drug design problems

Experiments NLP

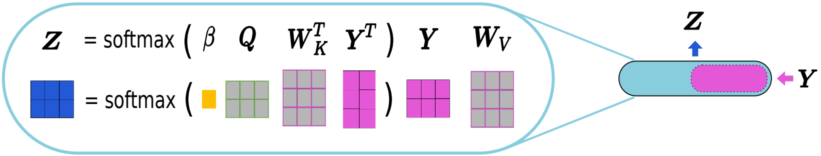
Minimal number k required to sum up the softmax values to 0.90: k indicates the **size of a metastable** state.

- **Very large metastable state** or global fixed point (layer 1)
- **Large metastable state** (layers 3, 4, 5)
- **Medium metastable state** (layers 10, 11, 12). Information collected that is required for the task.
- **Small metastable state** or fixed point close to a single patters (layers 6, 7, and 8)

Experiments: MIL

Multiple Instance Learning (MIL):

- **Memory** of new modern Hopfield Network is promising for MIL.
- **HopfieldPooling** as Hopfield layer in Deep Learning architectures.



Datasets:

1. Immune Repertoire Classification
2. MIL benchmark datasets

Immune Repertoire Classification

Multiple Instance Learning:

- Extract **few patterns** from a large set of sequences, the repertoire, that are indicative for the respective immune status.
- About **300,000 instances** per immune repertoire.
- One of the largest MIL tasks ever conducted.
- **HopfieldPooling** outperformed all other methods.

NeurIPS2020 Spotlight Paper “Modern Hopfield Networks and Attention for Immune Repertoire Classification”

MIL Benchmark Datasets

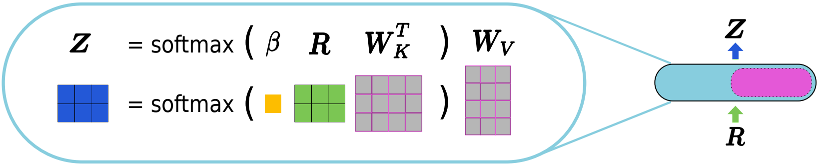
- MIL datasets Elephant, Fox and Tiger for image annotation:
  - Color images consist of a set of segments (1391; 1320; 1220)
  - Segment has 230 color, texture and shape descriptors
- UCSB breast cancer classification (cancerous or normal):
  - 2000 instances across 58 input objects
  - Instance: patch of a histopathological image

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

Small UCI Benchmark Collection

Small datasets of the UCI Benchmark Collection (UCI):

- Deep Learning struggles with **small datasets**.
- Layer **HopfieldLayer** can store the training data.
- Enables similarity-based or nearest neighbor methods.
- 121 UCI datasets: 75 “small datasets” with less than 1000 samples
- **Hopfield Networks outperform all other methods**.



Experiments Drug Design

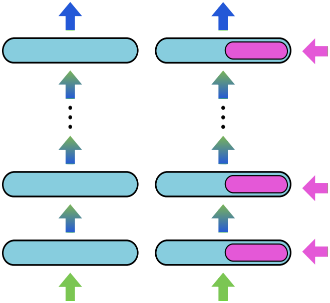
Four main areas of modeling tasks in drug design:

- **New anti-virals** (HIV) by the Drug Therapeutics Program (DTP)
- **New protein inhibitors**: human β-secretase (BACE) inhibitors
- **Metabolic effects** as blood-brain barrier permeability (BBBP)
- **Side effects** from the Side Effect Resource (SIDER)

Model	HIV	BACE	BBBP	SIDER
SVM	0.822 ± 0.020	0.893 ± 0.020	0.919 ± 0.028	0.630 ± 0.021
XGBoost	0.816 ± 0.020	0.889 ± 0.021	$\mathbf{0.926 \pm 0.026}$	0.642 ± 0.020
RF	0.820 ± 0.016	0.890 ± 0.022	$\mathbf{0.927 \pm 0.025}$	0.646 ± 0.022
GCN	$\mathbf{0.834 \pm 0.025}$	0.898 ± 0.019	0.903 ± 0.027	0.634 ± 0.026
GAT	0.826 ± 0.030	0.886 ± 0.023	0.898 ± 0.033	0.627 ± 0.024
DNN	0.797 ± 0.018	0.890 ± 0.024	0.898 ± 0.033	0.627 ± 0.024
MPNN	0.811 ± 0.031	0.838 ± 0.027	0.879 ± 0.037	0.598 ± 0.031
Attentive FP	0.822 ± 0.026	0.876 ± 0.023	0.887 ± 0.032	0.623 ± 0.026
Hopfield (ours)	0.815 ± 0.023	$\mathbf{0.902 \pm 0.023}$	0.910 ± 0.026	$\mathbf{0.672 \pm 0.019}$

Deep Learning with Memories

- The goal is to integrate **associative memories** into **Deep Learning architectures**.
- With **Modern continuous Hopfield Networs** we have a tool to do that.
- Deep Learning that goes beyond Convolutional and Recurrent Networks.
- Operations: pooling, memory, association, and attention mechanisms
- Can substitute: SVM, k-nearest neighbors, LVQ



Material

ICLR2021 paper: https://arxiv.org/abs/2008.02217

Blog post: https://ml-jku.github.io/hopfield-layers/

Software: https://github.com/ml-jku/hopfield-layers/

Video (Yannic Kilcher):
https://www.youtube.com/watch?v=nv6oFDp6rNQ