From Sparse Modeling to Sparse Communication

André Martins

Erice International School on Complexity

Ettore Majorana Foundation and Centre for Scientific Culture, Erice, Italy

April 27, 2024

Our Amazing Team

SARDINE: Structure AwaRe moDelIng for Natural LanguagE

Our Amazing Team

SARDINE: Structure AwaRe moDelIng for Natural LanguagE

DeepSPIN & DECOLLAGE

- ERC starting grant (2018–23) and consolidator grant (2023–28)
- Goal: put together *deep learning* and *structured prediction* for *natural language processing*
- More details: https://deep-spin.github.io

From Sparse Modeling ...

- Mostly used with linear models, lots of work in the 2000s
- Main idea: embed a sparse regularizer (e.g. ℓ_1-norm) in the learning objective
- Irrelevant features get zero weight and can be discarded
- Extensions to structured sparsity (group-lasso, fused-lasso, etc.)

... to Sparse Communication:

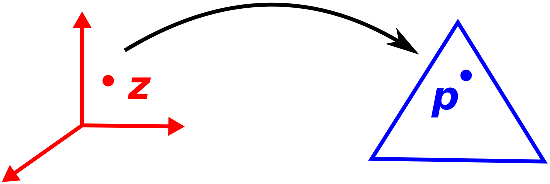
- Mostly used with neural networks, most work after 2015
- Main idea: sparse neuron activations (biological plausibility)
- Predictions are triggered by a few neurons only (input-dependent)
- Example: ReLUs, dropout, sparse attention mechanisms

This Talk

An inventory of transformations that capture **sparsity** and **structure**:

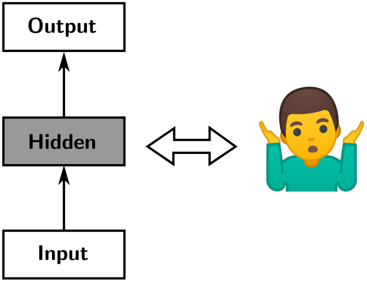
- All differentiable (efficient forward and backward propagation)
- Can be used at hidden (attention) or output layers (loss)
- Can make a bridge between the continuous and discrete worlds
- Effective in several natural language processing tasks.

Building block:

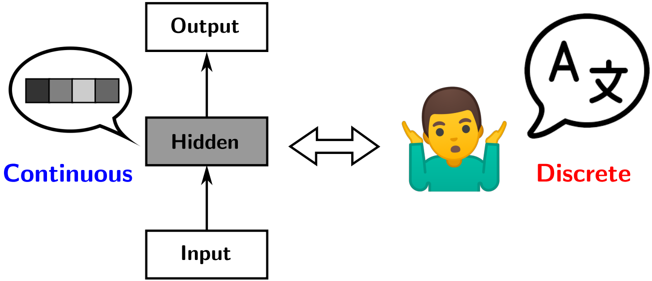


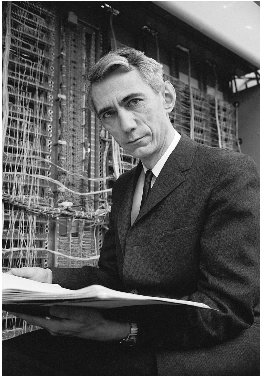
Sparse transformations from the Euclidean space to the simplex $\triangle$.

Machine-Human Communication



Machine-Human Communication





The Bell System Technical Journal

Vol. XXVII July, 1948 No. 3

A Mathematical Theory of Communication

By C. E. SHANNON

INTRODUCTION

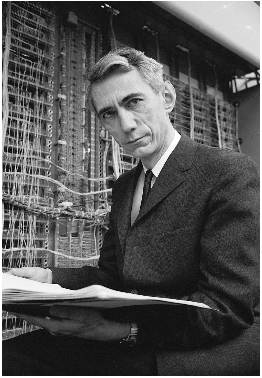
PART I: DISCRETE NOISELESS SYSTEMS

PART II: THE DISCRETE CHANNEL WITH NOISE

PART III: MATHEMATICAL PRELIMINARIES

PART IV: THE CONTINUOUS CHANNEL

PART V: THE RATE FOR A CONTINUOUS SOURCE



The Bell System Technical Journal

Vol. XXVII July, 1948 No. 3

A Mathematical Theory of Communication

By C. E. SHANNON

INTRODUCTION

PART I: DISCRETE NOISELESS SYSTEMS

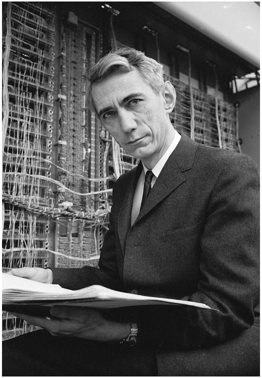
PART II: THE DISCRETE CHANNEL WITH NOISE

$\sum$

PART III: MATHEMATICAL PRELIMINARIES

PART IV: THE CONTINUOUS CHANNEL

PART V: THE RATE FOR A CONTINUOUS SOURCE



The Bell System Technical Journal

Vol. XXVII *July, 1948* *No. 3*

A Mathematical Theory of Communication

By C. E. SHANNON

INTRODUCTION

$\boxed{\begin{array}{c}\text{PART I: DISCRETE NOISELESS SYSTEMS} \\ \text{PART II: THE DISCRETE CHANNEL WITH NOISE}\end{array}}$ $\sum$

$\boxed{\begin{array}{c}\text{PART III: MATHEMATICAL PRELIMINARIES} \\ \text{PART IV: THE CONTINUOUS CHANNEL} \\ \text{PART V: THE RATE FOR A CONTINUOUS SOURCE}\end{array}}$ $\int$

$\sum$ vs. $\int$

Commonly we have to opt between discrete or continuous models:

- Language is symbolic and *discrete*
- Neural networks use (and learn) *continuous* representations

We should look at what happens in-between!

Sparsity might help with this, but...

$\sum$ vs. $\int$

Commonly we have to opt between discrete or continuous models:

- Language is symbolic and *discrete*
- Neural networks use (and learn) *continuous* representations

We should look at what happens in-between!

Sparsity might help with this, but...

... sparse probabilities are understudied and often excluded from theory:

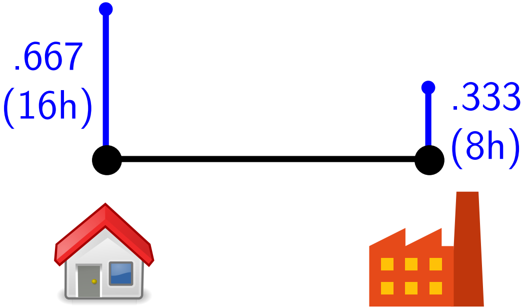
- Hammersley-Clifford theorem in graphical models
- Pitman-Koopman-Darmois theorem (sufficient statistics and exponential families)
- Log-likelihood is $-\infty$ if estimated probability is 0.

Motivating Example: John's Life

John splits his day as follows: he works 8h/day, and stays home 16h/day.

Motivating Example: John's Life

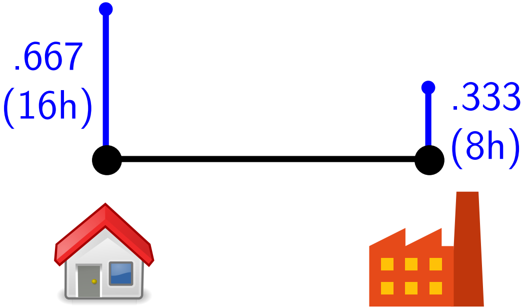
John splits his day as follows: he works 8h/day, and stays home 16h/day.



Motivating Example: John's Life

John splits his day as follows: he works 8h/day, and stays home 15h/day.

He is in transit 1h/day to commute to work and back.



Motivating Example: John's Life

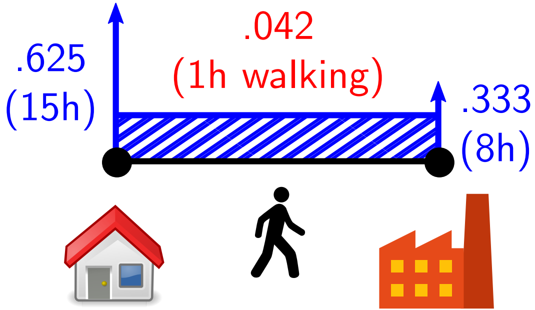
John splits his day as follows: he works 8h/day, and stays home 15h/day.

He is in transit 1h/day to commute to work and back.

Motivating Example: John's Life

John splits his day as follows: he works 8h/day, and stays home 15h/day.

He is in transit 1h/day to commute to work and back.



Is John's location a *discrete* or *continuous* random variable?

Motivating Example: John's Life

John splits his day as follows: he works 8h/day, and stays home 15h/day.

He is in transit 1h/day to commute to work and back.

Is John's location a *discrete* or *continuous* random variable? It's mixed.

Outline

Recap: Softmax and Argmax

Softmax exponentiates and normalizes:

$$\mathrm{softmax}(\boldsymbol{z}) = \frac{\exp(\boldsymbol{z})}{\sum_{k=1}^{K} \exp(z_k)}$$

- **Fully dense:** $\mathrm{softmax}(\boldsymbol{z}) > 0, \forall \boldsymbol{z}$
- Used both as a loss function (cross-entropy) and for attention.

Recap: Softmax and Argmax

Softmax exponentiates and normalizes:

$$\text{softmax}(\boldsymbol{z}) = \frac{\exp(\boldsymbol{z})}{\sum_{k=1}^{K} \exp(z_k)}$$

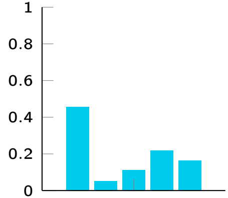
- **Fully dense:** $\text{softmax}(\boldsymbol{z}) > 0, \forall \boldsymbol{z}$
- Used both as a loss function (cross-entropy) and for attention.

Argmax can be written as:

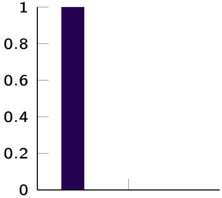
$$\begin{aligned} \text{argmax}(\boldsymbol{z}) &:= \arg\max_{\boldsymbol{p} \in \triangle} \boldsymbol{z}^\top \boldsymbol{p} \\ &= \lim_{\tau \to 0^+} \text{softmax}(\boldsymbol{z}/\tau) \qquad \text{(temperature trick)} \end{aligned}$$

- Retrieves a **one-hot vector** for the highest scored index.

softmax($\boldsymbol{z}$)

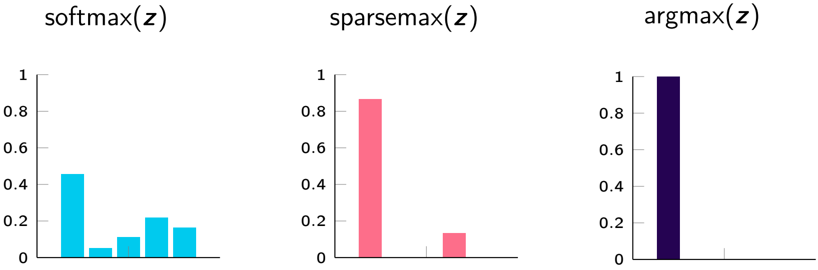


argmax($\boldsymbol{z}$)



(Same $\boldsymbol{z} = [1.0716, -1.1221, -0.3288, 0.3368, 0.0425]$)

- Argmax is an extreme case of sparsity, but it is **discontinuous**.
- Is there a **sparse** and **differentiable** alternative?



(Same $\boldsymbol{z} = [1.0716, -1.1221, -0.3288, 0.3368, 0.0425]$)

- Argmax is an extreme case of sparsity, but it is **discontinuous**.
- Is there a **sparse** and **differentiable** alternative?

Sparsemax (Martins and Astudillo, 2016, ICML)

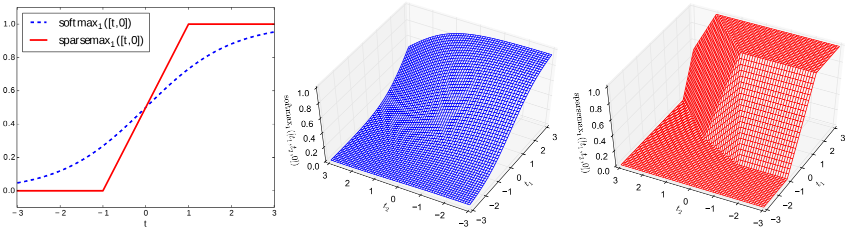
Euclidean projection of $\boldsymbol{z}$ onto the probability simplex $\triangle$:

$$\begin{aligned}\operatorname{sparsemax}(\boldsymbol{z}) &:= \arg\min_{\boldsymbol{p}\in\triangle} \|\boldsymbol{p}-\boldsymbol{z}\|^2 \\ &= \arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{z}^\top \boldsymbol{p} - \frac{1}{2}\|\boldsymbol{p}\|^2.\end{aligned}$$

- Likely to hit the boundary of the simplex, in which case sparsemax($\boldsymbol{z}$) becomes sparse (hence the name)
- End-to-end differentiable
- Forward pass: $O(K \log K)$ or $O(K)$, **(almost) as fast as softmax**
- Backprop: sublinear, **better than softmax!**

Sparsemax in 2D and 3D

(Martins and Astudillo, 2016, ICML)



- Sparsemax is piecewise linear, but asymptotically similar to softmax.

Ω-Regularized Argmax (Niculae and Blondel, 2017, NeurIPS)

For convex Ω, define the Ω**-regularized argmax transformation**:

$$\operatorname{argmax}_{\Omega}(\boldsymbol{z}) := \arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{z}^\top \boldsymbol{p} - \Omega(\boldsymbol{p})$$

- **Argmax** corresponds to **no regularization**, $\Omega \equiv 0$
- **Softmax** amounts to **entropic regularization**, $\Omega(\boldsymbol{p}) = \sum_{i=1}^{K} p_i \log p_i$
- **Sparsemax** amounts to ℓ_2**-regularization**, $\Omega(\boldsymbol{p}) = \frac{1}{2}\|\boldsymbol{p}\|^2$

Is there something in-between?

Entmax (Peters et al., 2019a, ACL)

Parametrized by $\alpha \geq 0$:

$$\Omega_\alpha(\boldsymbol{p}) := \begin{cases} \frac{1}{\alpha(\alpha-1)}\left(\sum_{i=1}^K p_i^\alpha - 1\right) & \text{if } \alpha \neq 1 \\ \sum_{i=1}^K p_i \log p_i & \text{if } \alpha = 1. \end{cases}$$

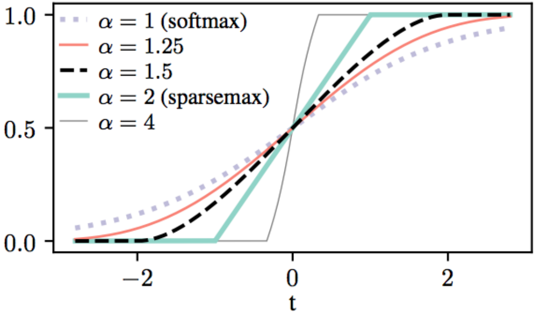
Related to **Tsallis generalized entropies** (Tsallis, 1988).

- **Argmax** corresponds to $\alpha \to \infty$
- **Softmax** amounts to $\alpha \to 1$
- **Sparsemax** amounts to $\alpha = 2$.

Key result: always sparse for $\alpha > 1$, sparsity increases with α

- Forward pass for general α can be done with a bissection algorithm
- Backward pass runs in sublinear time.

Entmax in 2D (Peters et al., 2019a, ACL)



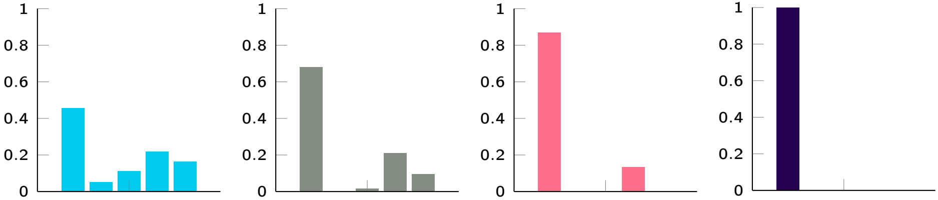
$\alpha = 1.5$ is a sweet spot!

- Efficient exact algorithm (nearly as fast as softmax), smooth, and good empirical performance.

Pytorch code: `https://github.com/deep-spin/entmax`

Sparse Transformations (Peters et al., 2019a, ACL)

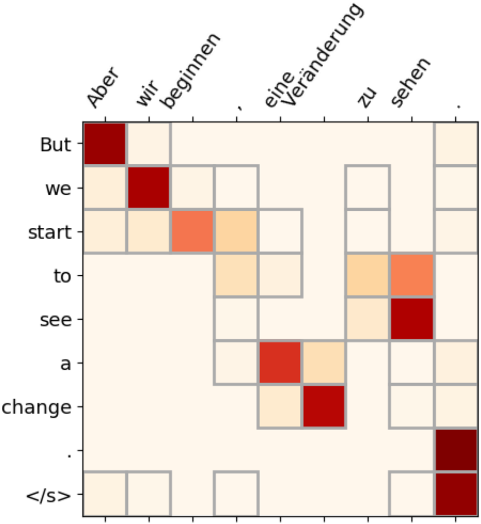
$\alpha = 1$	$\alpha = 1.5$	$\alpha = 2$	$\alpha = \infty$
$\mathrm{softmax}(\boldsymbol{z})$	$1.5\text{-}\mathrm{entmax}(\boldsymbol{z})$	$\mathrm{sparsemax}(\boldsymbol{z})$	$\mathrm{argmax}(\boldsymbol{z})$



(Same $\boldsymbol{z} = [1.0716, -1.1221, -0.3288, 0.3368, 0.0425]$)

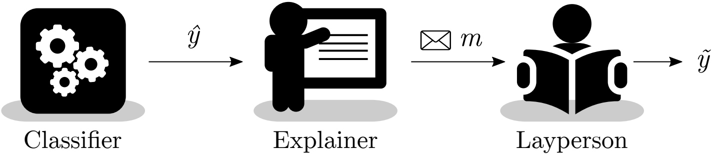
Example: Sparse Attention for Machine Translation

- **Selects** source words when generating a target word (sparse alignments)
- Better interpretability
- Can also model fertility: **constrained** sparsemax
  (Malaviya et al., 2018, ACL)
- Can also learn α (**adaptively sparse** transformers):
  (Correia et al., 2019, EMNLP)



Example: Sparse Attention for Explainability

(Treviso and Martins, 2020, BlackboxNLP)



- A classifier makes a prediction
- An "explainer" (embedded or not in the classifier) generates a sparse **message** that explains the classifier's decision
- The layperson receives the message and tries to guess the classifier's prediction (also called *simulatability*, *forward simulation/prediction*)
- Communication success rate: how often the two predictions match?
- Follow-up: Scaffold Maximizing Training (Fernandes et al., 2022, NeurIPS)

LP-SparseMAP (Niculae et al., 2018, ICML) (Niculae and Martins, 2020, ICML)

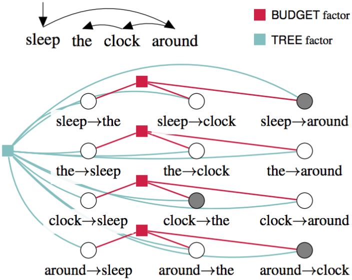
Generalizes sparsemax to **structures**.

Works both as output and hidden layer.

Can handle logic variables and constraints through a factor graph.

Returns **sparse** and **differentiable combination of structures**.

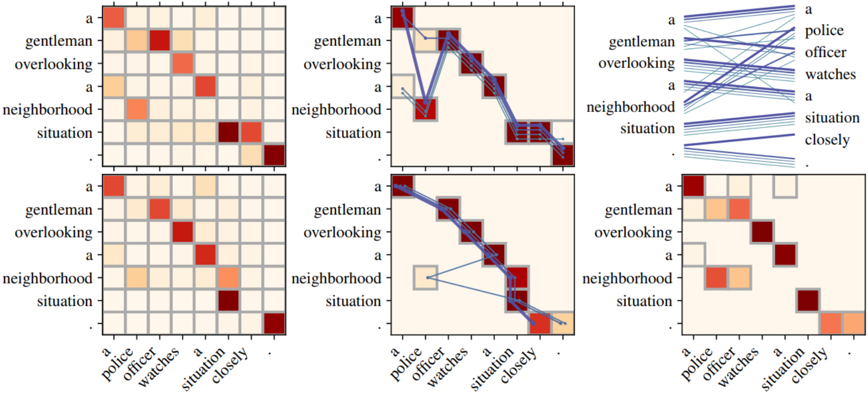
Efficient forward/backprop (requires only a MAP oracle).




```
fg = TorchFactorGraph()
u = fg.variable_from(arc_scores)
fg.add(DepTree(u))
for k in range(n):
    fg.add(Budget(u[:, k], budget=5))
fg.solve()
```

Example: Latent Structured Alignments in SNLI

(Niculae et al., 2018)



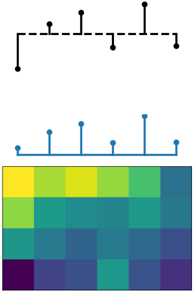
(a) softmax (b) sequence (c) matching

Sparse and Continuous Attention

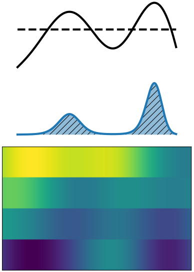
(Martins et al., 2020a, 2022a, NeurIPS, JMLR)

- So far: attention over a **finite set** (words, pixel regions, etc.)
- We generalize attention to *arbitrary sets*, possibly continuous.
- Applications: VQA; long-range ∞-former (Martins et al., 2022b, ACL)





$\Longrightarrow$



From Discrete to Continuous Attention

(Martins et al., 2020a, NeurIPS)

(Bahdanau et al., 2015, ICLR)

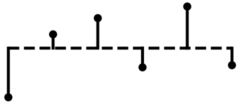
Finite set $S = \{1, \ldots, K\}$

From Discrete to Continuous Attention

(Martins et al., 2020a, NeurIPS)

(Bahdanau et al., 2015, ICLR)

Finite set $S = \{1, \ldots, K\}$



Three ingredients:

- Score vector $\boldsymbol{z} \in \mathbb{R}^K$

From Discrete to Continuous Attention

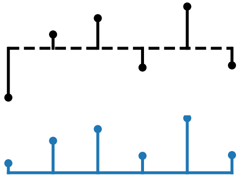
(Martins et al., 2020a, NeurIPS)

(Bahdanau et al., 2015, ICLR)

Finite set $S = \{1, \ldots, K\}$

Three ingredients:

- Score vector $\boldsymbol{z} \in \mathbb{R}^K$
- Transformation from $\boldsymbol{z}$ to probability vector $\boldsymbol{p} \in \triangle^K$



From Discrete to Continuous Attention

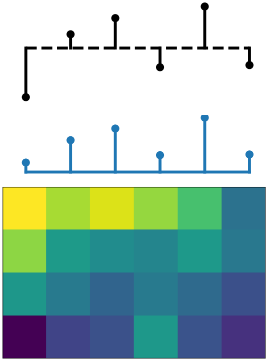
(Martins et al., 2020a, NeurIPS)

(Bahdanau et al., 2015, ICLR)

Finite set $S = \{1, \ldots, K\}$

Three ingredients:

- Score vector $\boldsymbol{z} \in \mathbb{R}^K$
- Transformation from $\boldsymbol{z}$ to probability vector $\boldsymbol{p} \in \triangle^K$
- Value matrix $V \in \mathbb{R}^{D \times K}$



From Discrete to Continuous Attention

(Martins et al., 2020a, NeurIPS)

(Bahdanau et al., 2015, ICLR)

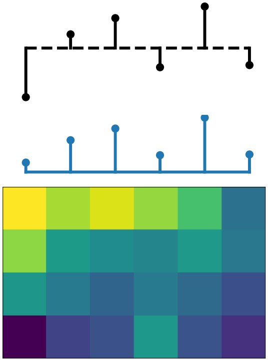
Finite set $S = \{1, \ldots, K\}$

Three ingredients:

- Score vector $\boldsymbol{z} \in \mathbb{R}^K$
- Transformation from $\boldsymbol{z}$ to probability vector $\boldsymbol{p} \in \triangle^K$
- Value matrix $V \in \mathbb{R}^{D \times K}$

Output:

- Weighted average $V\boldsymbol{p} \in \mathbb{R}^D$



From Discrete to Continuous Attention

(Martins et al., 2020a, NeurIPS)

Our work:

Measure space S (*e.g.* continuous)

Three ingredients:

- ~~Score vector $\boldsymbol{z} \in \mathbb{R}^K$~~
- ~~Transformation from $\boldsymbol{z}$ to probability vector $\boldsymbol{p} \in \triangle^K$~~
- ~~Value matrix $V \in \mathbb{R}^{D \times K}$~~

Output:

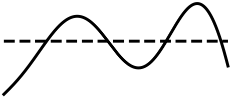
- ~~Weighted average $V\boldsymbol{p} \in \mathbb{R}^D$~~

From Discrete to Continuous Attention

(Martins et al., 2020a, NeurIPS)

Our work:

Measure space S (*e.g.* continuous)



Three ingredients:

- Score *function* $f : S \to \mathbb{R}$
- ~~Transformation from~~ $\cancel{\boldsymbol{z}}$ ~~to probability vector~~ $\cancel{\boldsymbol{p} \in \triangle^K}$
- ~~Value matrix~~ $\cancel{V \in \mathbb{R}^{D \times K}}$

Output:

- ~~Weighted average~~ $\cancel{V\boldsymbol{p} \in \mathbb{R}^D}$

From Discrete to Continuous Attention

(Martins et al., 2020a, NeurIPS)

Our work:

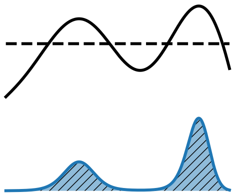
Measure space S (*e.g.* continuous)

Three ingredients:

- Score *function* $f : S \to \mathbb{R}$
- Transformation from f to *density* $p : S \to \mathbb{R}_+, \int_S p = 1$
- ~~Value matrix $V \in \mathbb{R}^{D\times K}$~~

Output:

- ~~Weighted average $V\boldsymbol{p} \in \mathbb{R}^D$~~



From Discrete to Continuous Attention

(Martins et al., 2020a, NeurIPS)

Our work:

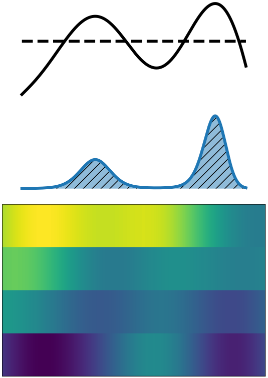
Measure space S (*e.g.* continuous)

Three ingredients:

- Score *function* $f : S \to \mathbb{R}$
- Transformation from f to *density* $p : S \to \mathbb{R}_+$, $\int_S p = 1$
- Value *function* $V : S \to \mathbb{R}^D$

Output:

- ~~Weighted average $V\boldsymbol{p} \in \mathbb{R}^D$~~



From Discrete to Continuous Attention

(Martins et al., 2020a, NeurIPS)

Our work:

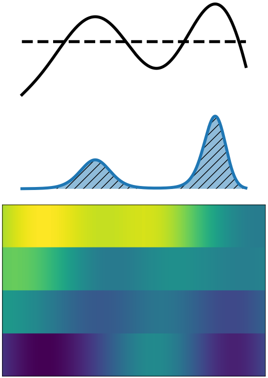
Measure space S (*e.g.* continuous)

Three ingredients:

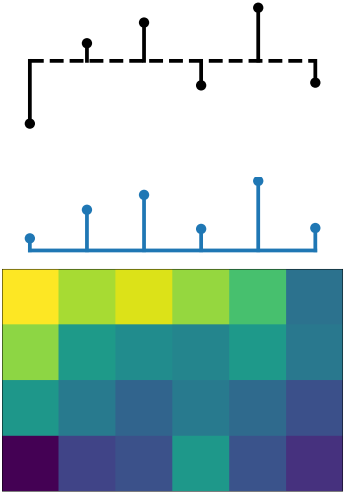
- Score *function* $f : S \to \mathbb{R}$
- Transformation from f to *density* $p : S \to \mathbb{R}_+$, $\int_S p = 1$
- Value *function* $V : S \to \mathbb{R}^D$

Output:

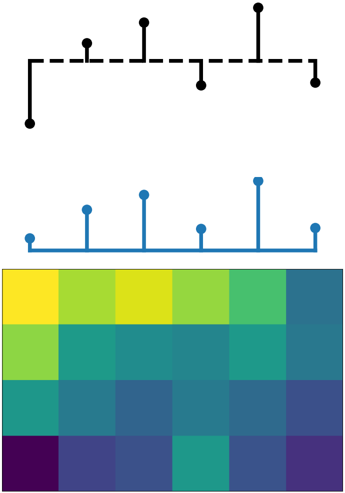
- $\mathbb{E}_p[V(t)] = \int_S p(t)V(t) \in \mathbb{R}^D$



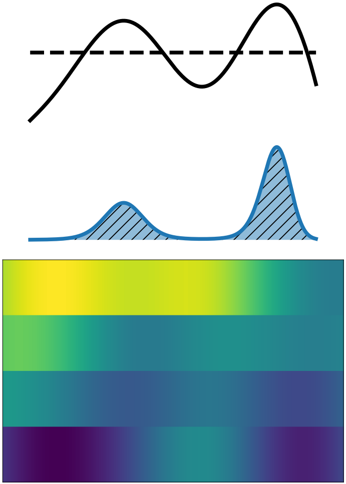
From Discrete to Continuous Attention



From Discrete to Continuous Attention



$\Longrightarrow$

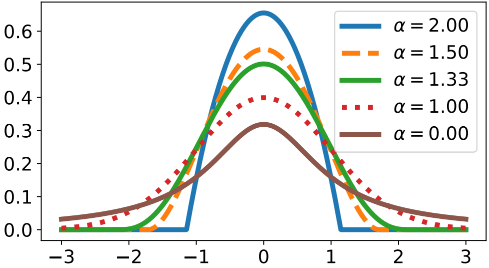


Sparse and Continuous

How to generalize the concept of **sparsity** to non-finite (e.g. continuous) domains S?

- A density with base measure μ is **sparse** iff $\mu(S \setminus \mathrm{supp}(p)) > 0$.

Examples with $S = \mathbb{R}$: q-Gaussians for $\alpha = 2 - q > 1$



q	α	
0	2	Epanechnikov
2/3	4/3	triweight
1/2	3/2	biweight
1	1	Gaussian
2	0	Cauchy

These can be generalized to $S = \mathbb{R}^K$ (later)

Ω-Regularized Prediction Map (Ω-RPM)

Transforms **score function** f into **probability density** $p \equiv \hat{p}_\Omega[f]$:

$$\hat{p}_\Omega[f] = \operatorname*{argmax}_p \mathbb{E}_p[f(t)] - \Omega(p), \qquad \Omega \text{ convex regularizer.}$$

Ω-Regularized Prediction Map (Ω-RPM)

Transforms **score function** f into **probability density** $p \equiv \hat{p}_\Omega[f]$:

$$\hat{p}_\Omega[f] = \operatorname*{argmax}_p \mathbb{E}_p[f(t)] - \Omega(p), \qquad \Omega \text{ convex regularizer.}$$

$-\Omega_\alpha$ **Tsallis α-entropy** $\Longrightarrow$ α**-entmax** (deformed exponential family):

$$\hat{p}_{\Omega_\alpha}[f](t) = \begin{cases} \exp(f(t) - \tau) & \text{if } \alpha = 1 \\ (1 + (\alpha - 1)(f(t) - \tau))_+^{\frac{1}{\alpha-1}} & \text{if } \alpha \neq 1 \end{cases}$$

This is the q**-exponential function**, with $q = 2 - \alpha$.

Particular cases: (continuous) softmax ($\alpha = 1$) and sparsemax ($\alpha = 2$).

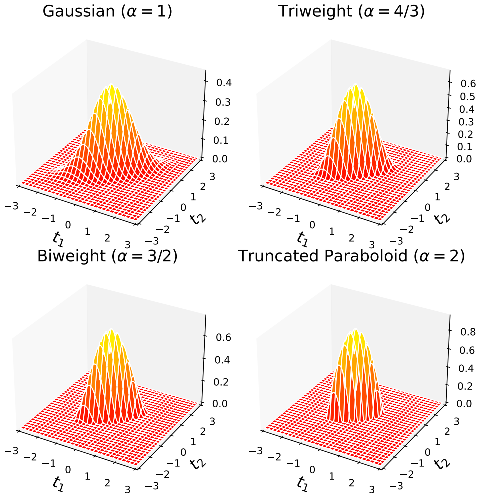
Blondel et al. (2020a, JMLR), Martins and Astudillo (2016, ICML), Peters et al. (2019b, ACL)
Tsallis (1988, "Possible Generalization of Boltzmann-Gibbs Statistics", J. of Stat. Phys.)

Example: Multivariate q-Gaussians

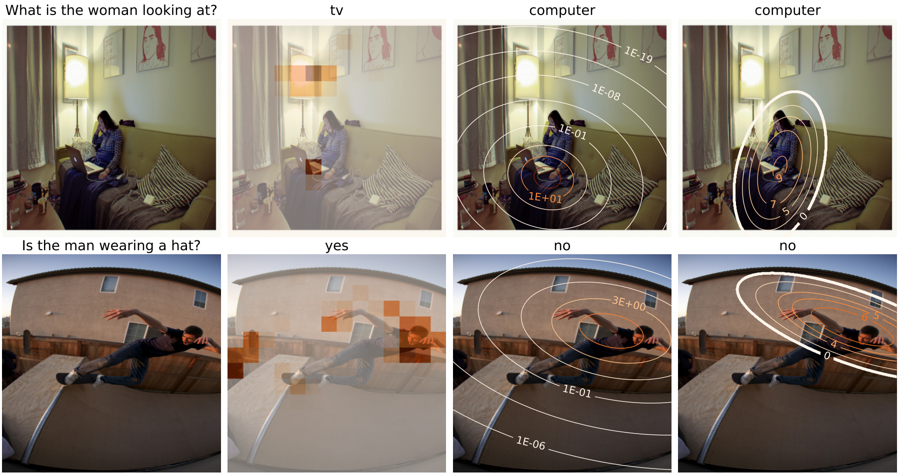
Quadratic score function: $f(t) = -\frac{1}{2}(t-\mu)^\top \Sigma^{-1}(t-\mu)$

Nice properties:

- They're instances of **elliptical distributions**
- Efficient to sample from (Beta + sphere trick)
- FY loss (Bregman) and Wasserstein distances computable in closed form
- For $\alpha = 2 - q = \frac{k}{k-1}$ with $k \in \mathbb{Z}$, attention and its gradient have closed form for the 1D case.



Example: Visual Question Answering



(original image) (discrete attention) (continuous softmax) (continuous sparsemax)

Outline

Loss Functions

Define the training objective to fit the model to the data.

Assess compatibility between:

- groundtruth $\boldsymbol{y}$ (e.g. response variable)
- model output $\boldsymbol{z}$ (e.g. last layer of a neural network).

Examples:

- squared loss in regression ($\boldsymbol{y} \in \mathbb{R}^K$, $\boldsymbol{z} \in \mathbb{R}^K$):

$$L(\boldsymbol{z}, \boldsymbol{y}) = \frac{1}{2}\|\boldsymbol{z} - \boldsymbol{y}\|^2$$

- cross-entropy loss in logistic regression ($\boldsymbol{y} \in \triangle$, $\boldsymbol{z} \in \mathbb{R}^K$):

$$\begin{aligned} L(\boldsymbol{z}, \boldsymbol{y}) &= -\sum_i y_i \log[\mathrm{softmax}(\boldsymbol{z})]_i \\ &= -\sum_i y_i z_i + \log \sum_i \exp(z_i). \end{aligned}$$

Entmax Losses

- Entmax can also be used as a loss in the **output layer** (to replace logistic/cross-entropy loss)
- However, not expressed as a log-likelihood (which could lead to $\log(0)$ problems due to sparsity)
- Instead, we build a entmax loss inspired by **Fenchel-Young losses**.

Recap: Ω-Regularized Argmax (Niculae and Blondel, 2017, NeurIPS)

For convex Ω, define the **Ω-regularized argmax transformation**:

$$\mathrm{argmax}_{\Omega}(\boldsymbol{z}) := \arg\max_{\boldsymbol{p}\in\triangle} \boldsymbol{z}^\top \boldsymbol{p} - \Omega(\boldsymbol{p})$$

- **Argmax** corresponds to **no regularization**, $\Omega \equiv 0$
- **Softmax** amounts to **entropic regularization**, $\Omega(\boldsymbol{p}) = \sum_{i=1}^{K} p_i \log p_i$
- **Sparsemax** amounts to **ℓ_2-regularization**, $\Omega(\boldsymbol{p}) = \frac{1}{2}\|\boldsymbol{p}\|^2$

All these are particular cases of α-entmax (Peters et al., 2019a, ACL).

Fenchel-Young Losses (Blondel et al., 2020b, JMLR)

Assess compatibility between groundtruth $\boldsymbol{y} \in \triangle$ and scores $\boldsymbol{z} \in \mathbb{R}^K$

Convex conjugate $\Omega^*(\boldsymbol{z}) := \max_{\boldsymbol{p} \in \triangle} \boldsymbol{z}^\top \boldsymbol{p} - \Omega(\boldsymbol{p})$

$$L_\Omega(\boldsymbol{z}, \boldsymbol{y}) := \Omega^*(\boldsymbol{z}) + \Omega(\boldsymbol{y}) - \boldsymbol{z}^\top \boldsymbol{y}$$

Recover **cross-entropy loss**: $\Omega(\boldsymbol{p}) = \sum_i p_i \log p_i \Rightarrow \Omega^*(\boldsymbol{z}) = \log \sum_i \exp(z_i)$.

Fenchel-Young Losses (Blondel et al., 2020b, JMLR)

Assess compatibility between groundtruth $\boldsymbol{y} \in \triangle$ and scores $\boldsymbol{z} \in \mathbb{R}^K$

Convex conjugate $\Omega^*(\boldsymbol{z}) := \max_{\boldsymbol{p} \in \triangle} \boldsymbol{z}^\top \boldsymbol{p} - \Omega(\boldsymbol{p})$

$$L_\Omega(\boldsymbol{z}, \boldsymbol{y}) := \Omega^*(\boldsymbol{z}) + \Omega(\boldsymbol{y}) - \boldsymbol{z}^\top \boldsymbol{y}$$

Recover **cross-entropy loss**: $\Omega(\boldsymbol{p}) = \sum_i p_i \log p_i \Rightarrow \Omega^*(\boldsymbol{z}) = \log \sum_i \exp(z_i)$.

Properties:

- $L_\Omega(\boldsymbol{z}, \boldsymbol{y}) \geq 0$ (automatic from **Fenchel-Young inequality**)
- $L_\Omega(\boldsymbol{z}, \boldsymbol{y}) = 0$ iff $\boldsymbol{y} = \operatorname{argmax}_\Omega(\boldsymbol{z})$
- L_Ω is convex and differentiable with $\nabla L_\Omega(\boldsymbol{z}, \boldsymbol{y}) = \operatorname{argmax}_\Omega(\boldsymbol{z}) - \boldsymbol{y}$

Also called “mixed-type Bregman divergences” (Amari, 2016).

Definition: Loss Margin

Some loss functions (e.g. the **hinge loss** in SVMs) are associated to the concept of **margin**.

A loss function $L(\boldsymbol{z}, \boldsymbol{y})$ has a margin if there is finite $m \geq 0$ such that

$$\forall i \in [K], \quad L(\boldsymbol{z}, \boldsymbol{e}_i) = 0 \Leftrightarrow z_i - \max_{j \neq i} z_j \geq m.$$

The smallest such m is called the margin of L.

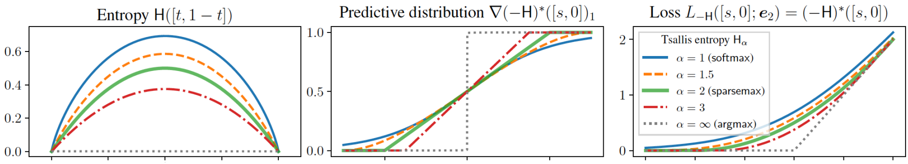
If L_Ω is a Fenchel-Young loss, this condition is equivalent to

$$\text{argmax}_\Omega(\boldsymbol{z}) = \boldsymbol{e}_i.$$

Corollary: cross-entropy loss does **not** have a margin.

Entmax Transformations and Losses

(Blondel et al., 2020b, JMLR)

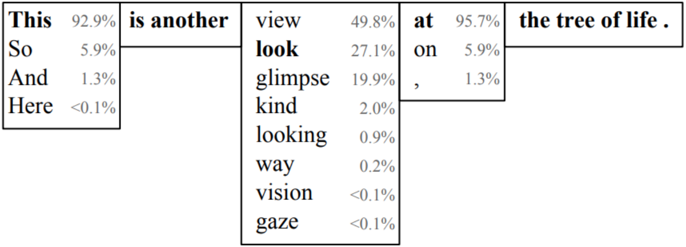


- Key result: for all $\alpha > 1$, α-entmax transformations are **sparse** and lead to losses with **margins**!
- The **margin** m is related to the **slope** of the entropy in the simplex corners! ($m = \frac{1}{\alpha - 1}$ for entmax losses.)
- See paper for details!

Pytorch code: `https://github.com/deep-spin/entmax`

Example: Machine Translation

(Peters et al., 2019a, ACL) (Peters and Martins, 2021, NAACL)



(Source: "Dies ist ein weiterer Blick auf den Baum des Lebens.")

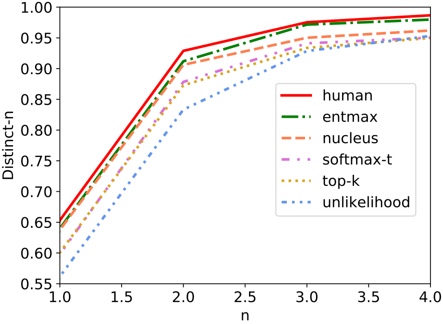
- Only a few words get non-zero probability at each time step
- Auto-completion when several words in a row have probability 1
- Useful for predictive translation.

Entmax Sampling (Martins et al., 2020b, EMNLP)

Use the entmax loss for training language models.

At test time, sample from this sparse distribution.

Better quality with less repetitions than other methods:



Outline

What are Hopfield Networks? (Amari, 1972; Hopfield, 1982)

A model of recurrent neural network:

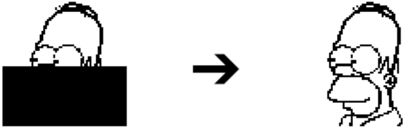
- Named after John Hopfield, an American physicist and neuroscientist.
- Designed for associative memory and associative recall.
- Stores and recalls patterns, making it useful for pattern recognition.
- Inspired by the human brain's associative memory.
  - Our brain stores and retrieves information not through explicit memory addresses but by associating content with memories.
  - **Content-based recall**: in the brain, seeing or hearing a partial cue can trigger the recall of associated memories.

Hopfield Networks (Amari, 1972; Hopfield, 1982)

A model of associative memory:

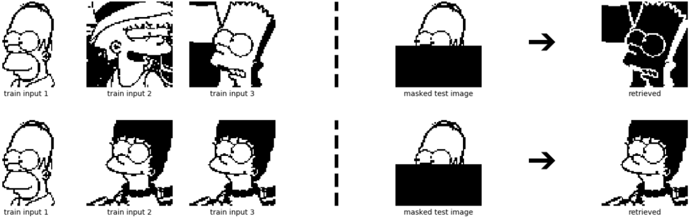
- Memory patterns $\boldsymbol{X} = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_N] \in \{\pm 1\}^{N \times D}$, query $\boldsymbol{q} \in \{\pm 1\}^D$
- Energy $E(\boldsymbol{q}) = -\frac{1}{2}\|\boldsymbol{X}\boldsymbol{q}\|^2$
- Hopfield dynamics $\boldsymbol{q}_{t+1} = \mathrm{sign}(\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{q}_t)$
- Memory patterns are attractors (but many spurious attractors)
- Memory capacity is only $N \lesssim 0.138D = \mathcal{O}(D)$.

Hopfield Networks



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

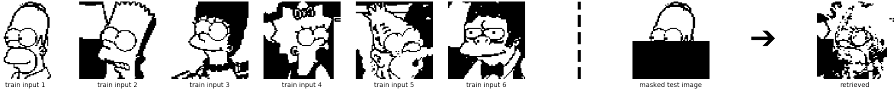
What if we store more than one pattern?



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

Hopfield Networks

What if we try with even more?



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

There are alternative energies with much better capacity (Krotov and Hopfield, 2016; Demircigil et al., 2017; Ramsauer et al., 2020).

Caveat: needs to store patterns explicitly.

Dense Associative Memories

(Krotov and Hopfield, 2016; Demircigil et al., 2017)

- Krotov et al. proposed a new energy function:

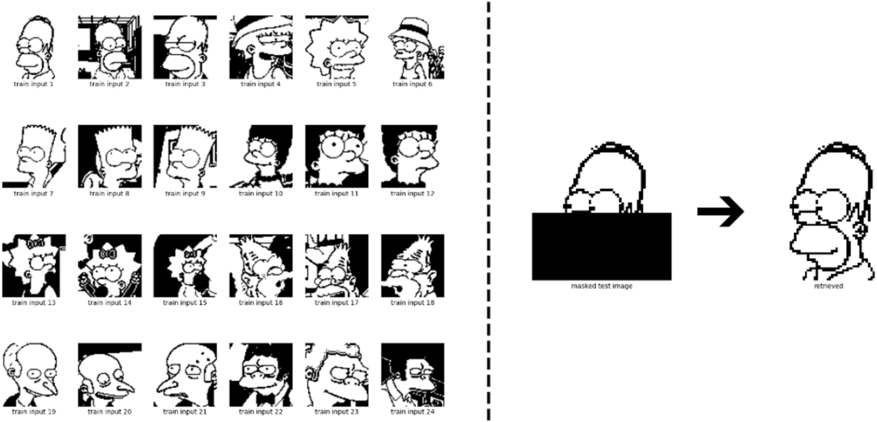
$$E(\boldsymbol{q}) = -F(\boldsymbol{X}\boldsymbol{q}) \qquad\qquad F(\boldsymbol{x}) = \begin{cases} \boldsymbol{x}^n & \text{if } \boldsymbol{x} \geq 0 \\ 0 & \text{if } \boldsymbol{x} < 0 \end{cases}$$

- Hopfield dynamics $\boldsymbol{q}_{t+1} = \operatorname{sign}(\boldsymbol{X}^\top \operatorname{spow}(\boldsymbol{X}\boldsymbol{q}_t, n-1))$
- Demircigil et al. further expanded the energy function:

$$E(\boldsymbol{q}) = -\exp(\boldsymbol{X}\boldsymbol{q})$$

- Hopfield dynamics $\boldsymbol{q}_{t+1} = \operatorname{sign}(\boldsymbol{X}^\top \exp(\boldsymbol{X}\boldsymbol{q}_t))$
- Memory capacity is now $\mathcal{O}(\exp(D))$

Example


train input 1
train input 2
train input 3
train input 4
train input 5
train input 6
train input 7
train input 8
train input 9
train input 10
train input 11
train input 12
train input 13
train input 14
train input 15
train input 16
train input 17
train input 18
train input 19
train input 20
train input 21
train input 22
train input 23
train input 24
masked test image
retrieved

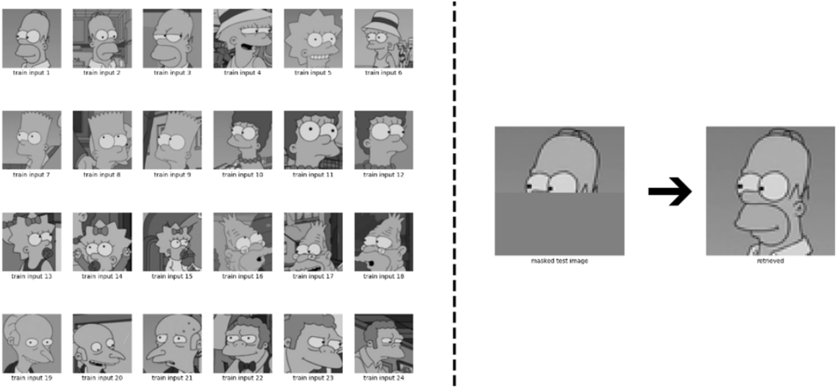
Modern Hopfield Networks (Ramsauer et al., 2020)

- Operates on continuous space, $\boldsymbol{X} \in \mathbb{R}^{N \times D}$, $\boldsymbol{q} \in \mathbb{R}^{D}$
- Energy:

$$E(\boldsymbol{q}) = -\beta^{-1} \log \sum_{i=1}^{N} \exp(\beta \boldsymbol{x}_i^\top \boldsymbol{q}) + \frac{1}{2} \|\boldsymbol{q}\|^2.$$

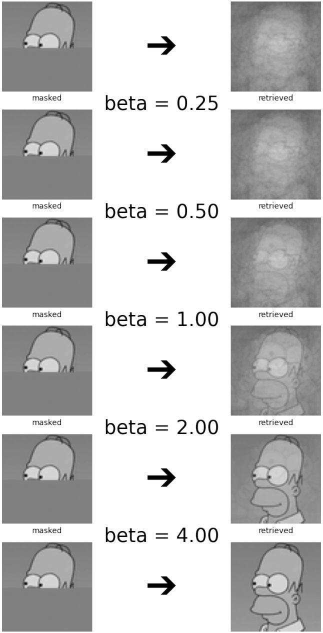
- Hopfield dynamics $\boldsymbol{q}_{t+1} = \boldsymbol{X}^\top \mathrm{softmax}(\beta \boldsymbol{X} \boldsymbol{q}_t)$
- Similar to self-attention in transformers!
- Memory patterns are close to attractors (but there can be some spurious attractors)
- Memory capacity is $\mathcal{O}(\exp(D))$ (but retrieval is only approximate)

Modern Hopfield Networks



If some stored patterns are similar to each other, then a metastable state near the similar patterns appears.

Sensitivity to Temperature



Research Questions

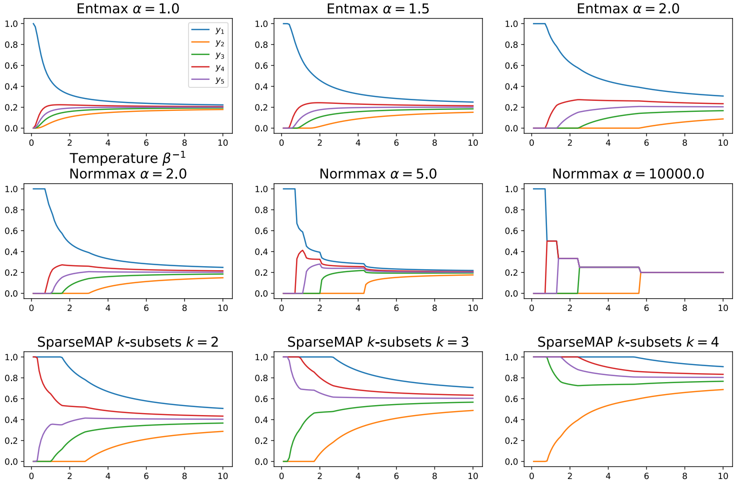
- Can we design high capacity, continuous-space Hopfield networks with exact retrieval?
- Can we make them less sensitive to temperature?
- Can we extend them to handle structure?

Research Questions

- Can we design high capacity, continuous-space Hopfield networks with exact retrieval?
- Can we make them less sensitive to temperature?
- Can we extend them to handle structure?

Yes, if we use sparse transformations and Fenchel-Young losses!

Sparse and Structured Transformations



Regularization path of sparse and structured transformations.
Shown is $\text{argmax}_{\Omega}(\beta \boldsymbol{z})$ as a function of the temperature β^{-1} where
$\boldsymbol{z} = [1.0716, -1.1221, -0.3288, 0.3368, 0.0425]^\top$.

Sparse Modern Hopfield Networks (Hu et al., 2023; Santos et al., 2024)

- Hopfield-Fenchel-Young energy, induced by convex Ω:

$$E(\boldsymbol{q}) = -\beta^{-1}\Omega^*(\beta \boldsymbol{X}\boldsymbol{q}) + \frac{1}{2}\|\boldsymbol{q}\|^2$$
$$= -L_{\beta^{-1}\Omega}(\boldsymbol{X}\boldsymbol{q}; \boldsymbol{u}) + \frac{1}{2}\|\boldsymbol{q} - \boldsymbol{X}^\top \boldsymbol{u}\|^2 + \text{const.}$$

- Includes MHNs as a particular case
- Hopfield dynamics $\boldsymbol{q}_{t+1} = \boldsymbol{X}^\top \operatorname{argmax}_\Omega(\beta \boldsymbol{X}\boldsymbol{q})$
- If $\operatorname{argmax}_\Omega$ is a sparse transformation, memory patterns are exactly attractors (but there can be some spurious attractors)
- Memory capacity is still $\mathcal{O}(\exp(D))$ (but retrieval can be exact)

Exact Convergence to Single Pattern

Define separation of pattern $\boldsymbol{x}_i$ from data (Ramsauer et al., 2020):

$$\Delta_i = \boldsymbol{x}_i^\top \boldsymbol{x}_i - \max_{j \neq i} \boldsymbol{x}_i^\top \boldsymbol{x}_j.$$

Assume L_Ω is a FY loss with margin m, and let $\boldsymbol{x}_i$ be a memory pattern outside the convex hull of the other memory patterns. Then,

- $\boldsymbol{x}_i$ is a stationary point of the energy iff $\Delta_i \geq \frac{1}{m\beta}$.
- if the initial query satisfies $\boldsymbol{q}_0^\top(\boldsymbol{x}_i - \boldsymbol{x}_j) \geq \frac{1}{m\beta}$ for all $j \neq i$, then the update rule converges to $\boldsymbol{x}_i$ exactly in one iteration.
- if the patterns are normalized and $\Delta_i \geq \frac{1}{m\beta} + 2M\epsilon$, then any $\boldsymbol{q}_0$ ϵ-close to $\boldsymbol{x}_i$ ($\|\boldsymbol{q}_0 - \boldsymbol{x}_i\| \leq \epsilon$) will converge to $\boldsymbol{x}_i$ in one iteration.

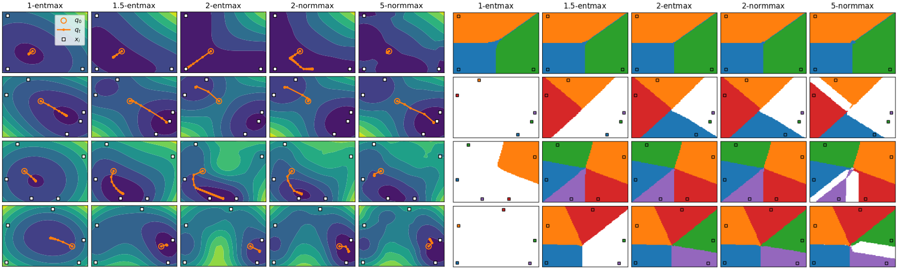
Margins: $m = 1$ for normmax and $m = \frac{1}{\alpha - 1}$ for α-entmax.

Storage Capacity with Exact Retrieval

Assume patterns are randomly placed on the sphere with uniform distribution. Then, with probability $1 - p$, the HFY network can store and exactly retrieve $N = \mathcal{O}(\sqrt{p}\zeta^{\frac{D-1}{2}})$ patterns in one iteration under a ϵ-perturbation if

$$\epsilon \leq \frac{M}{2}\left(1 - \cos\frac{1}{\zeta}\right) - \frac{m}{2\beta M}.$$

Example: Hopfield Dynamics and Basis of Attraction



As α increases:

- α-entmax converges more often to a single pattern.
- α-normmax tends to converge towards an attractor which is a uniform average of some patterns.

Structured Hopfield Networks

Similar guarantees for structured sparse transformations:

- Hopfield dynamics $\boldsymbol{q}_{t+1} = \boldsymbol{X}^\top \mathrm{SparseMAP}(\beta \boldsymbol{X} \boldsymbol{q}_t)$

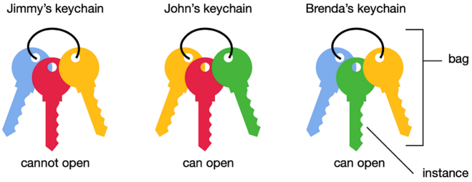
Examples of structural constraints:

- ***k*-subsets**:
  - Retrieve subsets of k patterns, e.g., to take into account a k-ary relation among patterns or to perform top-k retrieval.
- **sequential *k*-subsets**:
  - Promote consecutive memory items to be both (or none) retrieved.

Other structures (trees, graphs, matchings, ...) are possible.

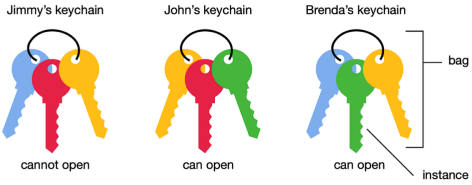
Example: Multiple Instance Learning

MIL



Example: Multiple Instance Learning

MIL



K-MIL

At least K positive instances

Example: Multiple Instance Learning

Methods	MNIST				MIL benchmarks		
	K=1	K=2	K=3	K=5	Fox	Tiger	Elephant
1-entmax (softmax)	98.4 ± 0.2	94.6 ± 0.5	91.1 ± 0.5	89.0 ± 0.3	66.4 ± 2.0	87.1 ± 1.6	92.6 ± 0.6
1.5-entmax	97.6 ± 0.8	96.0 ± 0.9	90.4 ± 1.1	92.4 ± 1.4	66.3 ± 2.0	87.3 ± 1.5	92.4 ± 1.0
2.0-entmax (sparsemax)	97.9 ± 0.2	96.7 ± 0.5	92.9 ± 0.9	91.6 ± 1.0	66.1 ± 0.6	$\mathbf{87.7 \pm 1.4}$	91.8 ± 0.6
2.0-normmax	97.9 ± 0.3	96.6 ± 0.6	93.9 ± 0.7	92.4 ± 0.7	66.1 ± 2.5	86.4 ± 0.8	92.4 ± 0.7
5.0-normmax	98.2 ± 0.5	97.2 ± 0.3	95.8 ± 0.4	93.2 ± 0.5	66.4 ± 2.3	85.5 ± 0.6	93.0 ± 0.7
SparseMAP, $k = 2$	$\mathbf{98.7 \pm 0.3}$	$\mathbf{97.9 \pm 0.2}$	92.2 ± 0.5	92.2 ± 0.5	66.8 ± 2.7	85.4 ± 0.6	$\mathbf{93.1 \pm 1.0}$
SparseMAP, $k = 3$	97.2 ± 0.4	96.2 ± 1.2	$\mathbf{97.3 \pm 0.4}$	92.0 ± 0.6	$\mathbf{68.0 \pm 2.6}$	85.0 ± 0.5	90.8 ± 0.7
SparseMAP, $k = 5$	97.5 ± 0.9	97.4 ± 0.5	94.5 ± 0.8	$\mathbf{96.2 \pm 1.1}$	65.5 ± 1.9	79.8 ± 1.2	89.9 ± 0.6

- Normmax consistently performs well across datasets, likely because of its adaptability to near-uniform metastable states of varying sizes.
- When $K > 1$, the k-subsets method works best with $k = K$.

See Santos et al. (2024) for more experiments (e.g. text rationalization).

Outline

Mixed Distributions (Farinhas et al., 2022, ICLR)

- We saw how to obtain sparse probability distributions.
- How can we use them to bridge the gap between *discrete* and *continuous* domains?
- We'll see how next.

Back to John's Life

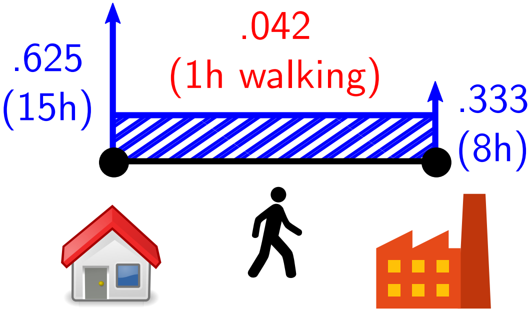
John splits his day as follows: he works 8h/day, and stays home 15h/day.

He is in transit 1h/day to commute to work and back.

Back to John's Life

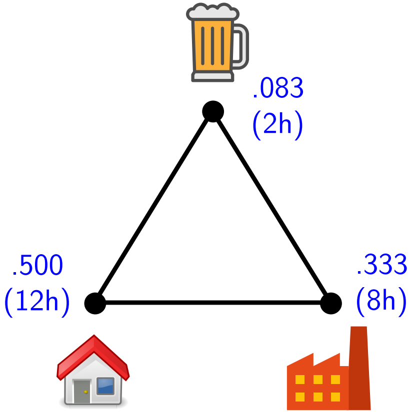
John splits his day as follows: he works 8h/day, and stays home 15h/day.

He is in transit 1h/day to commute to work and back.

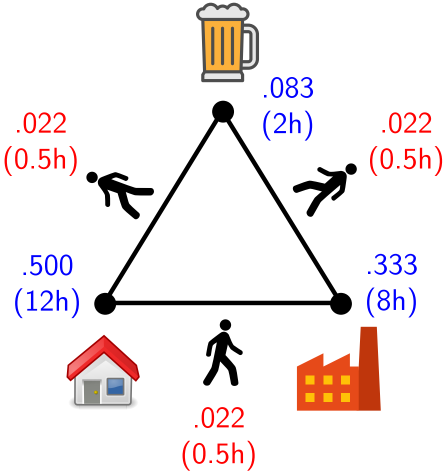


That's a sad life!

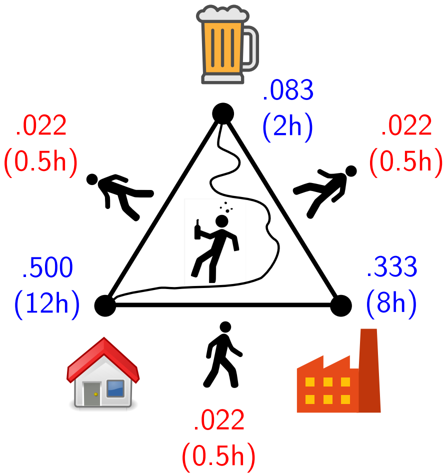
After work, John spends 2h in the pub with friends.



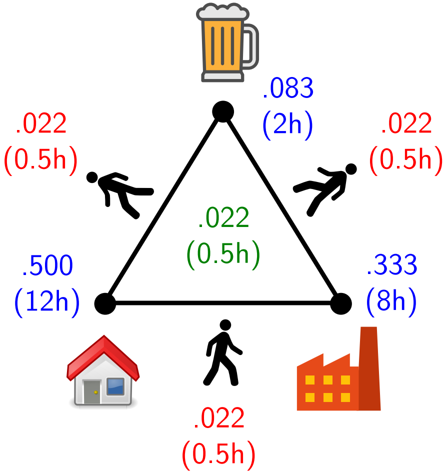
After work, John spends 2h in the pub with friends.



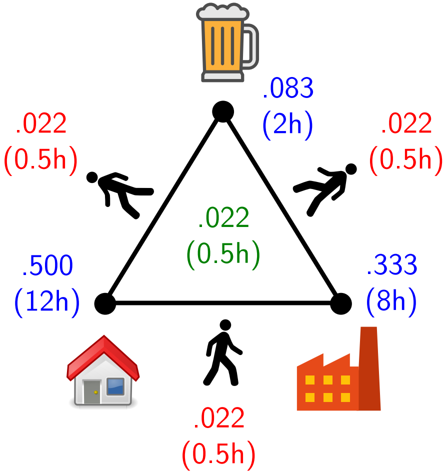
After work, John spends 2h in the pub with friends.



After work, John spends 2h in the pub with friends.



After work, John spends 2h in the pub with friends.



We need a way to represent this probability mass in vertices, edges, face.

Densities over the simplex $\triangle$

We denote by $\mathrm{ri}(\triangle)$ the relative interior of $\triangle$.

Common densities on the simplex:

- Dirichlet distribution
- Logistic-Normal (a.k.a. Gaussian-Softmax)
- Concrete (a.k.a. Gumbel-Softmax)

None of these place any probability mass on the boundary $\triangle \setminus \mathrm{ri}(\triangle)$.

Truncated Densities in the Binary Case ($K = 2$)

When $K = 2$, the simplex is isomorphic to unit interval, $\triangle_1 \simeq [0, 1]$.

A point in $\triangle_1$ can be represented as $\boldsymbol{y} = [y, 1 - y]$.

Truncated densities have been proposed for $K = 2$:

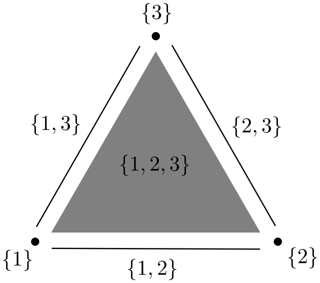
- Binary Hard Concrete **(Louizos et al., 2018)**
- Rectified Gaussian **(Hinton and Ghahramani, 1997; Palmer et al., 2017)**

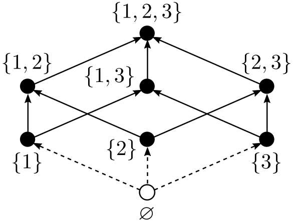
We generalize them for $K > 2$.

Our Approach: Face Stratification

How to extend these "truncated densities" to $K > 2$?

Our solution relies on the face lattice of the simplex:





0-faces are vertices, 1-faces are edges, ..., the $(K-1)$-face is $\triangle$ itself.

We define a direct sum measure on the stratified $\triangle$ and define probability densities w.r.t. this base measure.

Mixed Random Variables (Farinhas et al., 2022, ICLR)

Discrete RVs assign probability only to 0-faces (vertices of $\triangle$).

Continuous RVs assign probability only to the maximal face ($\mathrm{ri}(\triangle)$).

Mixed RVs generalize both: can assign probability to all faces of $\triangle$.

Mixed Random Variables (Farinhas et al., 2022, ICLR)

Discrete RVs assign probability only to 0-faces (vertices of $\triangle$).

Continuous RVs assign probability only to the maximal face ($\mathrm{ri}(\triangle)$).

Mixed RVs generalize both: can assign probability to all faces of $\triangle$.

They can be defined via:

- Their face probability mass function $P_F(f) = \Pr\{\boldsymbol{y} \in \mathrm{ri}(f)\}$, $f \in \mathcal{F}$.
- Their face-conditional densities $p_{Y|F}(\boldsymbol{y} \mid f)$, for $f \in \mathcal{F}$, $\boldsymbol{y} \in \mathrm{ri}(f)$.

The probability of a set $A \subseteq \triangle$ is given by:

$$\Pr\{\boldsymbol{y} \in A\} = \sum_{f \in \mathcal{F}} P_F(f) \int_{A \cap \mathrm{ri}(f)} p_{Y|F}(\boldsymbol{y} \mid f).$$

Extrinsic vs Intrinsic (Farinhas et al., 2022, ICLR)

Two ways of characterizing mixed RVs:

- Extrinsic characterizaton: start with a distribution over $\mathbb{R}^K$ and then apply a deterministic transformation to project it to $\triangle$
- Intrinsic characterizaton: specify a mixture of distributions directly over the faces of $\triangle$, by specifying P_F and $p_{Y|F}$ for each $f \in \mathcal{F}$

K-D Hard Concrete (Farinhas et al., 2022, ICLR)

Uses an extrinsic characterization, via "stretch-and-project."

Generative story:

$$Y \sim \mathsf{HardConcrete}(\boldsymbol{z}, \lambda, \tau) \quad \Leftrightarrow \quad \begin{array}{l} Y' \sim \mathsf{Concrete}(\boldsymbol{z}, \lambda) \\ Y = \mathsf{sparsemax}(\tau Y'), \quad \text{with } \tau \geq 1. \end{array}$$

- Recovers the binary Hard Concrete for $K = 2$
- The larger τ, the higher the tendency to hit a non-maximal face of the simplex and induce sparsity.

Gaussian-Sparsemax (Farinhas et al., 2022, ICLR)

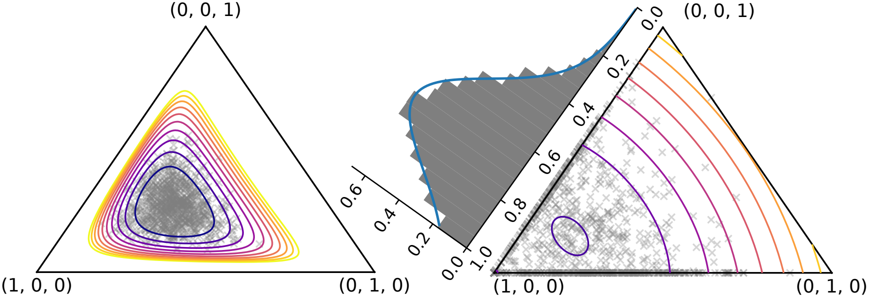
Uses an extrinsic characterization, by sampling from a Gaussian and projecting.

Generative story:

$$Y \sim \mathrm{GaussianSparsemax}(\boldsymbol{z}, \Sigma) \quad \Leftrightarrow \quad \begin{array}{l} N \sim \mathcal{N}(0, \mathrm{I}) \\ Y = \mathrm{sparsemax}(\boldsymbol{z} + \Sigma^{1/2} N). \end{array}$$

- Sparsemax counterpart of the Logistic-Normal.
- Can assign nonzero probability mass to the boundary of the simplex.
- When $K = 2$, we recover the double-sided rectified Gaussian.
- For $K > 2$, an intrinsic representation can be expressed via the orthant probability of multivariate Gaussians.

Logistic-Normal vs Gaussian-Sparsemax (Farinhas et al., 2022, ICLR)



Logistic-Normal (left) assigns zero probability to all faces but $\mathrm{ri}(\triangle)$

Gaussian-Sparsemax (right) is a mixed distribution: it assigns probability to the *full* simplex, including its boundary.

Information Theory of Mixed Random Variables

(Farinhas et al., 2022, ICLR)

"Direct sum" entropy using $\mu^{\oplus}$ as the base measure:

$$\begin{aligned} H^{\oplus}(Y) &:= H(F) + H(Y \mid F) \\ &= \underbrace{-\sum_{f \in \mathcal{F}} P_F(f) \log P_F(f)}_{\text{discrete entropy}} + \sum_{f \in \mathcal{F}} P_F(f) \underbrace{\left(-\int_f p_{Y|F}(\boldsymbol{y} \mid f) \log p_{Y|F}(\boldsymbol{y} \mid f) \right)}_{\text{differential entropy}}. \end{aligned}$$

- Average length of the optimal code where f must be encoded losslessly and where $\boldsymbol{y}|_f$ has a predefined bit precision N
- Max-ent is written as a generalized Laguerre polynomial (see paper)
  - e.g. $\log_2(2 + 2^N)$ for $K = 2$ (vs. $\log_2(2) = 1$ in the purely discrete case)
- KL divergence and mutual information defined similarly.

Experiment: Emergent Communication

The first agent needs to communicate a code to the second agent that represents a given image.

Given the code, the second agent needs to identify the correct image among 16 possibilities. (Random guess is $1/16 = 6.25\%$.)

Success average and standard error over 10 runs:

Method	Success (%)	Nonzeros ↓
Gumbel-Softmax	78.84 $\pm$8.07	256
Gumbel-Softmax ST	49.96 $\pm$9.51	1
K-D Hard Concrete	76.07 $\pm$7.76	21.43 $\pm$17.56
Gaussian-Sparsemax	**80.88** $\pm$0.50	1.57 $\pm$0.02

(See paper for more experiments with VAEs on FashionMNIST and MNIST.)

Outline

Conclusions

- Transformations from real numbers to distributions are ubiquitous
- We introduced new transformations that handle **sparsity**, **constraints**, and **structure**
- All are differentiable and their gradients are efficient to compute
- Can be used as hidden layers or as output layers (Fenchel-Young losses)
- Mixed distributions are in-between the discrete and continuous worlds
- Sparse communication potentially useful as a path for explainability.

Thank you!

References I

Amari, S.-I. (1972). Learning patterns and pattern sequences by self-organizing nets of threshold elements. *IEEE Transactions on computers*, 100(11):1197–1206.

Amari, S.-i. (2016). *Information geometry and its applications*, volume 194. Springer.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *Proc. ICLR*.

Blondel, M., Martins, A. F., and Niculae, V. (2020a). Learning with fenchel-young losses. *Journal of Machine Learning Research*, 21(35):1–69.

Blondel, M., Martins, A. F. T., and Niculae, V. (2020b). Learning with fenchel-young losses. *Journal of Machine Learning Research*, 21(35):1–69.

Correia, G., Niculae, V., and Martins, A. F. T. (2019). Adaptively sparse transformers. In *Proceedings of the Empirical Methods for Natural Language Processing*.

Demircigil, M., Heusel, J., Löwe, M., Upgang, S., and Vermet, F. (2017). On a model of associative memory with huge storage capacity. *Journal of Statistical Physics*, 168:288–299.

Farinhas, A., Aziz, W., Niculae, V., and Martins, A. F. (2022). Sparse communication via mixed distributions. In *Proc. of ICLR*.

Fernandes, P., Treviso, M., Pruthi, D., Martins, A. F., and Neubig, G. (2022). Learning to scaffold: Optimizing model explanations for teaching. *arXiv preprint arXiv:2204.10810*.

Hinton, G. E. and Ghahramani, Z. (1997). Generative models for discovering sparse distributed representations. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 352(1358):1177–1190.

References II

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558.

Hu, J. Y.-C., Yang, D., Wu, D., Xu, C., Chen, B.-Y., and Liu, H. (2023). On sparse modern hopfield model. In *Advances in Neural Information Processing Systems*.

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

Louizos, C., Welling, M., and Kingma, D. P. (2018). Learning sparse neural networks through l_0 regularization. In *International Conference on Learning Representations*.

Malaviya, C., Ferreira, P., and Martins, A. F. T. (2018). Sparse and Constrained Attention for Neural Machine Translation. In *Proc. of the Annual Meeting of the Association for Computation Linguistics*.

Martins, A. and Astudillo, R. (2016). From softmax to sparsemax: A sparse model of attention and multi-label classification. In Balcan, M. F. and Weinberger, K. Q., editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1614–1623, New York, New York, USA. PMLR.

Martins, A., Farinhas, A., Treviso, M., Niculae, V., Aguiar, P., and Figueiredo, M. (2020a). Sparse and continuous attention mechanisms. *Advances in Neural Information Processing Systems*, 33.

Martins, A. F. T., Treviso, M., Farinhas, A., Aguiar, P. M., Figueiredo, M. A., Blondel, M., and Niculae, V. (2022a). Sparse continuous distributions and Fenchel-Young losses. *Journal of Machine Learning Research (to appear)*, 23(257):1–74.

Martins, P. H., Marinho, Z., and Martins, A. F. (2020b). Sparse text generation. In *Empirical Methods for Natural Language Processing*.

References III

Martins, P. H., Marinho, Z., and Martins, A. F. T. (2022b). ∞-former: Infinite memory transformer. In *Proc. of NAACL-HLT*.

Niculae, V. and Blondel, M. (2017). A regularized framework for sparse and structured neural attention. *arXiv preprint arXiv:1705.07704*.

Niculae, V. and Martins, A. F. (2020). Lp-sparsemap: Differentiable relaxed optimization for sparse structured prediction. In *International Conference on Machine Learning*.

Niculae, V., Martins, A. F. T., Blondel, M., and Cardie, C. (2018). SparseMAP: Differentiable Sparse Structured Inference. In *Proc. of the International Conference on Machine Learning*.

Palmer, A. W., Hill, A. J., and Scheding, S. J. (2017). Methods for stochastic collection and replenishment (scar) optimisation for persistent autonomy. *Robotics and Autonomous Systems*, 87:51–65.

Peters, B. and Martins, A. F. (2021). Smoothing and shrinking the sparse seq2seq search space. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2642–2654.

Peters, B., Niculae, V., and Martins, A. F. T. (2019a). Sparse sequence-to-sequence models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Peters, B., Niculae, V., and Martins, A. F. T. (2019b). Sparse sequence-to-sequence models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1504–1519, Florence, Italy. Association for Computational Linguistics.

Ramsauer, H., Schäfl, B., Lehner, J., Seidl, P., Widrich, M., Adler, T., Gruber, L., Holzleitner, M., Pavlović, M., Sandve, G. K., et al. (2020). Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217*.

References IV

Santos, S., Niculae, V., McNamee, D., and Martins, A. F. (2024). Sparse and structured hopfield networks. *arXiv preprint arXiv:2402.13725*.

Treviso, M. and Martins, A. F. (2020). The explanation game: Towards prediction explainability through sparse communication. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 107–118.

Tsallis, C. (1988). Possible generalization of boltzmann-gibbs statistics. *Journal of Statistical Physics*, 52:479–487.