Differentiable Adaptive Sparsity For Neural Networks

Vlad Niculae

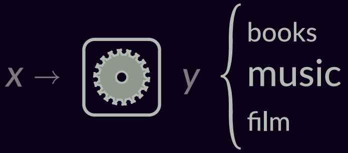
Instituto de Telecomunicações

https://vene.ro

Choosing Between K Options

A building block in many ML tasks!

multi-class classification

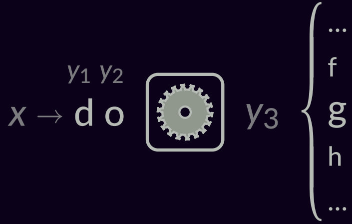


Choosing Between *K* Options

A building block in many ML tasks!

multi-class classification

sequence generation



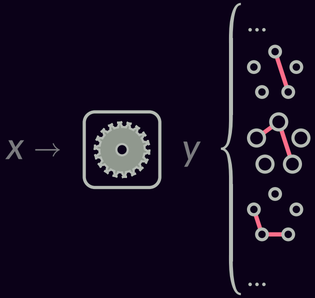
Choosing Between *K* Options

A building block in many ML tasks!

multi-class classification

sequence generation

structured output prediction



Choosing Between *K* Options

A building block in many ML tasks!

multi-class classification
sequence generation
structured output prediction

} output

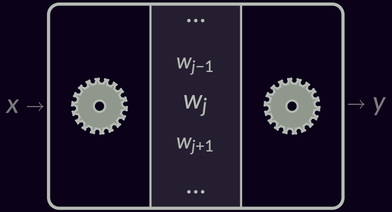
Choosing Between K Options

A building block in many ML tasks!

multi-class classification
sequence generation
structured output prediction

} output

neural attention



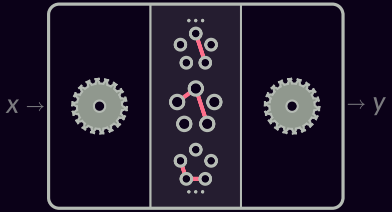
Choosing Between *K* Options

A building block in many ML tasks!

multi-class classification
sequence generation
structured output prediction

} output

neural attention
structured hidden layers



Choosing Between *K* Options

A building block in many ML tasks!

multi-class classification		
sequence generation	}	output
structured output prediction		
neural attention	}	hidden
structured hidden layers		

Choosing Between *K* Options

A building block in many ML tasks!

multi-class classification
sequence generation
structured output prediction
} output

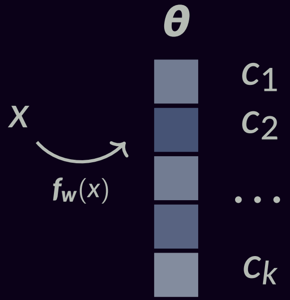
neural attention
structured hidden layers
} hidden

Deterministic **sparse & structured mappings** and losses via a general, constructive framework.

Outline

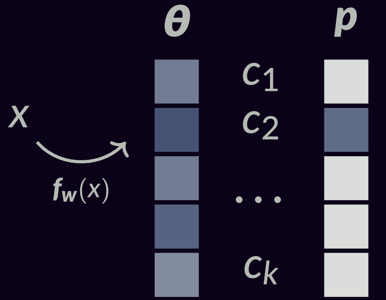
1. Warm-Up: Well-Known Losses and Mappings
2. Regularized Prediction Functions
3. Fenchel-Young Losses
4. Sparse Sequence-to-Sequence Models
5. Adaptively Sparse Transformers
6. Sparse Structured Prediction

Perceptron & Argmax



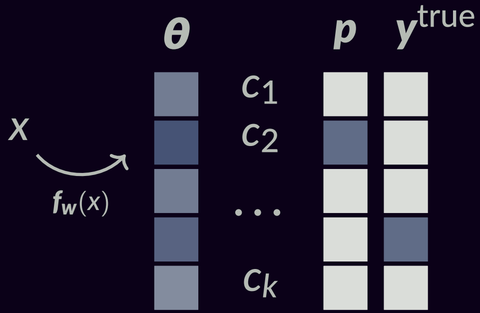
Perceptron & Argmax

$\boldsymbol{p} := \mathrm{argmax}(\boldsymbol{\theta})$



- very sparse predictions

Perceptron & Argmax



$$\boldsymbol{p} := \operatorname{argmax}(\boldsymbol{\theta})$$

$$L(\boldsymbol{\theta}; \boldsymbol{y}^{\text{true}}) = \langle \boldsymbol{\theta}, \boldsymbol{p} \rangle - \langle \boldsymbol{\theta}, \boldsymbol{y}^{\text{true}} \rangle$$

$$\partial_{\boldsymbol{\theta}} L(\boldsymbol{\theta}; \boldsymbol{y}^{\text{true}}) \ni \boldsymbol{p} - \boldsymbol{y}^{\text{true}}$$

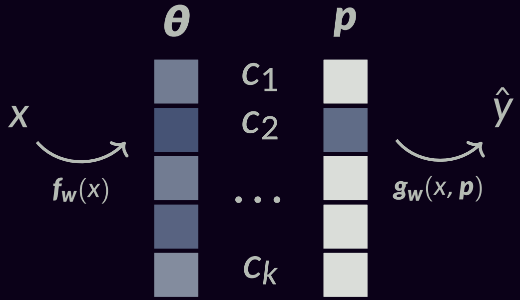
- very sparse predictions
- famous update rule

Perceptron & Argmax

$$\boldsymbol{p} := \operatorname{argmax}(\boldsymbol{\theta})$$

$$L(\boldsymbol{\theta}; \boldsymbol{y}^{\text{true}}) = \langle \boldsymbol{\theta}, \boldsymbol{p} \rangle - \langle \boldsymbol{\theta}, \boldsymbol{y}^{\text{true}} \rangle$$

$$\partial_{\boldsymbol{\theta}} L(\boldsymbol{\theta}; \boldsymbol{y}^{\text{true}}) \ni \boldsymbol{p} - \boldsymbol{y}^{\text{true}}$$



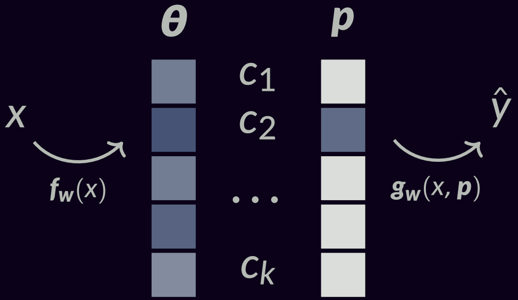
- very sparse predictions
- famous update rule

Perceptron & Argmax

$$\boldsymbol{p} := \operatorname{argmax}(\boldsymbol{\theta})$$

$$L(\boldsymbol{\theta}; \boldsymbol{y}^{\text{true}}) = \langle \boldsymbol{\theta}, \boldsymbol{p} \rangle - \langle \boldsymbol{\theta}, \boldsymbol{y}^{\text{true}} \rangle$$

$$\partial_{\boldsymbol{\theta}} L(\boldsymbol{\theta}; \boldsymbol{y}^{\text{true}}) \ni \boldsymbol{p} - \boldsymbol{y}^{\text{true}}$$



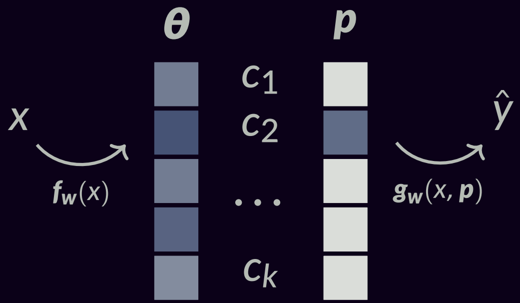
- very sparse predictions
- famous update rule
- can't use as hidden layer: $\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{\theta}} = \mathbf{0}$ *a.e.*

Perceptron & Argmax

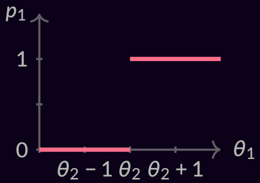
$$\boldsymbol{p} := \operatorname{argmax}(\boldsymbol{\theta})$$

$$L(\boldsymbol{\theta}; \boldsymbol{y}^{\text{true}}) = \langle \boldsymbol{\theta}, \boldsymbol{p} \rangle - \langle \boldsymbol{\theta}, \boldsymbol{y}^{\text{true}} \rangle$$

$$\partial_{\boldsymbol{\theta}} L(\boldsymbol{\theta}; \boldsymbol{y}^{\text{true}}) \ni \boldsymbol{p} - \boldsymbol{y}^{\text{true}}$$

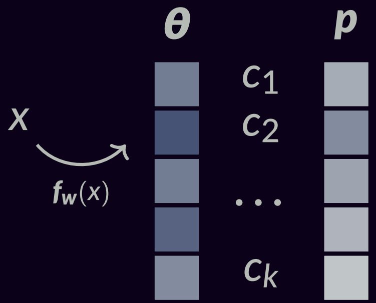


- very sparse predictions
- famous update rule
- can't use as hidden layer: $\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{\theta}} = \mathbf{0}$ *a.e.*



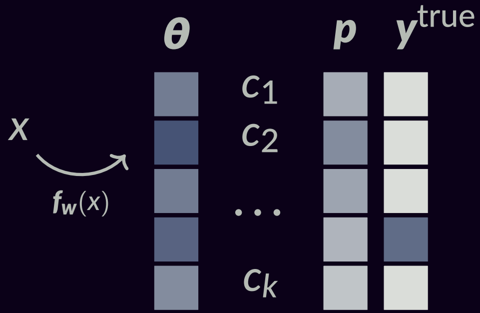
Logistic Regression & Softmax

$\boldsymbol{p} := \text{softmax}(\boldsymbol{\theta})$



- dense predictive distribution (Gibbs)

Logistic Regression & Softmax



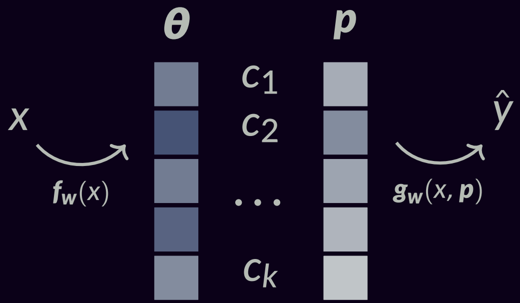
$$\boldsymbol{p} := \operatorname{softmax}(\boldsymbol{\theta})$$

$$L(\boldsymbol{\theta}; \boldsymbol{y}^{\text{true}}) = \log \sum_j \exp \theta_j - \langle \boldsymbol{\theta}, \boldsymbol{y}^{\text{true}} \rangle$$

$$\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}; \boldsymbol{y}^{\text{true}}) = \boldsymbol{p} - \boldsymbol{y}^{\text{true}}$$

- dense predictive distribution (Gibbs)
- loss gradient:
  expected − *observed* statistics

Logistic Regression & Softmax



$$\boldsymbol{p} := \text{softmax}(\boldsymbol{\theta})$$

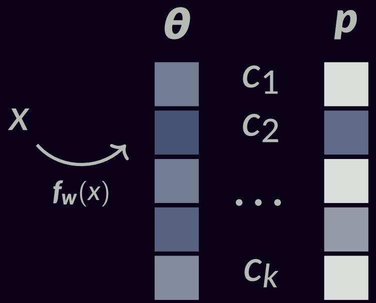
$$L(\boldsymbol{\theta}; \boldsymbol{y}^{\text{true}}) = \log \sum_j \exp \theta_j - \langle \boldsymbol{\theta}, \boldsymbol{y}^{\text{true}} \rangle$$

$$\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}; \boldsymbol{y}^{\text{true}}) = \boldsymbol{p} - \boldsymbol{y}^{\text{true}}$$

- dense predictive distribution (Gibbs)
- loss gradient:
  expected − *observed* statistics
- soft hidden layers: $\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{\theta}} = \text{diag}(\boldsymbol{p}) - \boldsymbol{p}\boldsymbol{p}^\top$
  (neural attention)

Sparsemax

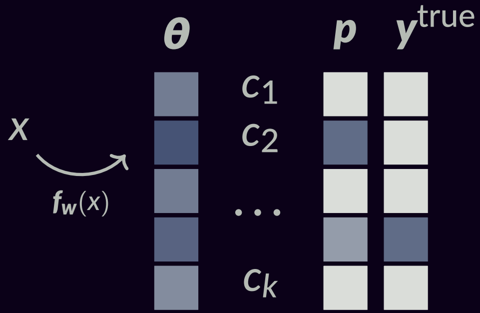
$$\boldsymbol{p} := \text{sparsemax}(\boldsymbol{\theta}) = \text{proj}_{\triangle}(\boldsymbol{\theta})$$



- sparse predictive distribution

(Martins and Astudillo, 2016)

Sparsemax



$$\boldsymbol{p} := \text{sparsemax}(\boldsymbol{\theta}) = \text{proj}_{\triangle}(\boldsymbol{\theta})$$

$$L(\boldsymbol{\theta}, \boldsymbol{y}^{\text{true}}) = \quad ?$$

$$\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \boldsymbol{y}^{\text{true}}) = \boldsymbol{p} - \boldsymbol{y}^{\text{true}}$$

- sparse predictive distribution
- reverse-engineer loss from gradient
  expected − *observed* statistics

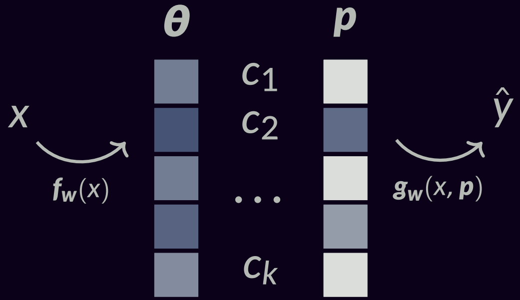
(Martins and Astudillo, 2016)

Sparsemax

$$\boldsymbol{p} := \text{sparsemax}(\boldsymbol{\theta}) = \text{proj}_{\triangle}(\boldsymbol{\theta})$$

$$L(\boldsymbol{\theta}, \boldsymbol{y}^{\text{true}}) = \ ?$$

$$\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \boldsymbol{y}^{\text{true}}) = \boldsymbol{p} - \boldsymbol{y}^{\text{true}}$$



- sparse predictive distribution
- reverse-engineer loss from gradient
  expected − *observed* statistics
- sparse attention by deriving $\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{\theta}}$

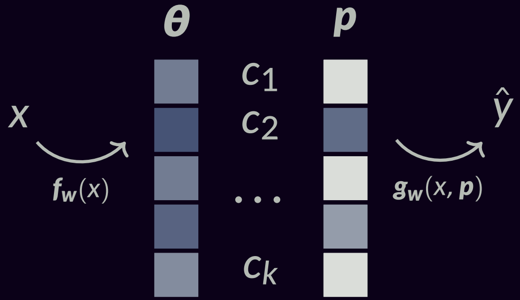
(Martins and Astudillo, 2016)

Sparsemax

$$\boldsymbol{p} := \text{sparsemax}(\boldsymbol{\theta}) = \text{proj}_{\triangle}(\boldsymbol{\theta})$$

$$L(\boldsymbol{\theta}, \boldsymbol{y}^{\text{true}}) = \;\; ?$$

$$\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \boldsymbol{y}^{\text{true}}) = \boldsymbol{p} - \boldsymbol{y}^{\text{true}}$$



- sparse predictive distribution
- reverse-engineer loss from gradient
  expected − *observed* statistics
- sparse attention by deriving $\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{\theta}}$

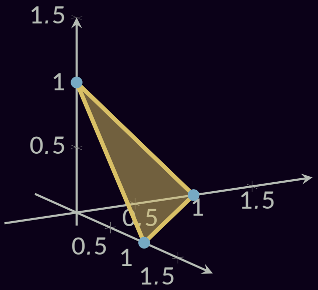
where do softmax-like functions come from?

A Softmax Origin Story 🦸🏽‍♀️

A Softmax Origin Story 🦸🏽‍♀️

First, some background.

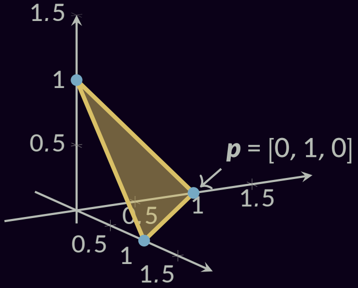
The simplex $\Delta := \{\boldsymbol{p} \in \mathbb{R}^k \ : \ \boldsymbol{p} \geq \boldsymbol{0}, \ \sum_j p_j = 1\}$



A Softmax Origin Story 🦸‍♀️

First, some background.

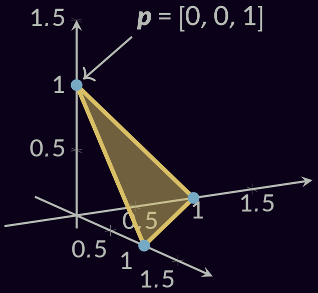
The simplex $\triangle := \{\boldsymbol{p} \in \mathbb{R}^k \; : \; \boldsymbol{p} \geq \boldsymbol{0}, \; \sum_j p_j = 1\}$



A Softmax Origin Story 🦸‍♀️

First, some background.

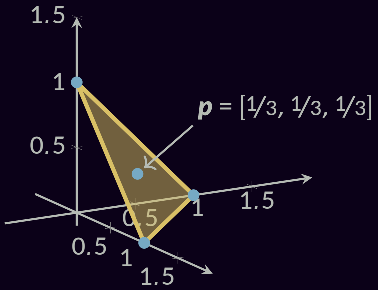
The simplex $\triangle := \{\boldsymbol{p} \in \mathbb{R}^k \ : \ \boldsymbol{p} \geq \boldsymbol{0}, \ \sum_j p_j = 1\}$



A Softmax Origin Story 🦸🏽‍♀️

First, some background.

The simplex $\Delta := \{\boldsymbol{p} \in \mathbb{R}^k \;:\; \boldsymbol{p} \geq \boldsymbol{0},\; \sum_j p_j = 1\}$

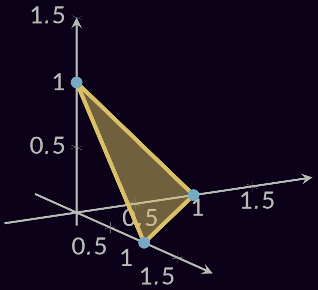


A Softmax Origin Story 🦸🏽‍♀️

First, some background.

The simplex $\triangle := \{\boldsymbol{p} \in \mathbb{R}^k \ : \ \boldsymbol{p} \geq \mathbf{0}, \ \sum_j p_j = 1\}$

Extended value functions $f : \mathbb{R}^k \to \mathbb{R} \cup \{\infty\}$



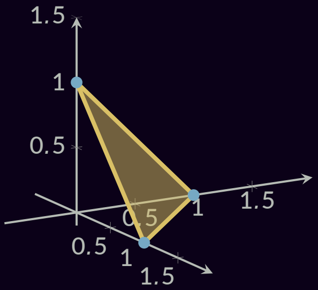
A Softmax Origin Story 🦸🏽‍♀️

First, some background.

The simplex $\Delta := \{\boldsymbol{p} \in \mathbb{R}^k \ : \ \boldsymbol{p} \geq \mathbf{0}, \ \sum_j p_j = 1\}$

Extended value functions $f : \mathbb{R}^k \to \mathbb{R} \cup \{\infty\}$

$\mathrm{dom}(f)$:= points where f is finite



A Softmax Origin Story

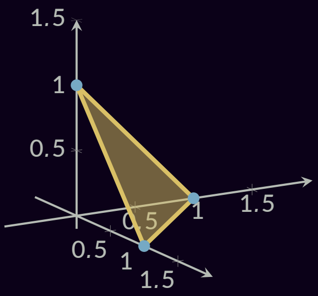
First, some background.

The simplex $\triangle := \{\boldsymbol{p} \in \mathbb{R}^k \ : \ \boldsymbol{p} \geq \boldsymbol{0}, \ \sum_j p_j = 1\}$

Extended value functions $f : \mathbb{R}^k \to \mathbb{R} \cup \{\infty\}$

$\mathrm{dom}(f)$:= points where f is finite

Indicator function: $\iota_{\mathcal{S}}(\boldsymbol{x}) = \begin{cases} 0, & \boldsymbol{x} \in \mathcal{S} \\ \infty, & \boldsymbol{x} \notin \mathcal{S} \end{cases}$



A Softmax Origin Story 🦸🏽‍♀️

First, some background.

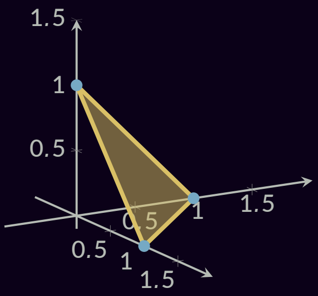
The simplex $\triangle := \{\boldsymbol{p} \in \mathbb{R}^k \ : \ \boldsymbol{p} \geq \boldsymbol{0}, \ \sum_j p_j = 1\}$

Extended value functions $f : \mathbb{R}^k \to \mathbb{R} \cup \{\infty\}$

$\operatorname{dom}(f) :=$ points where f is finite

Indicator function: $\iota_{\mathcal{S}}(\boldsymbol{x}) = \begin{cases} 0, & \boldsymbol{x} \in \mathcal{S} \\ \infty, & \boldsymbol{x} \notin \mathcal{S} \end{cases}$

($f + \iota_{\mathcal{S}}$ is f restricted to $\mathcal{S}$)



A Softmax Origin Story

First, some background.

The simplex $\triangle := \{\boldsymbol{p} \in \mathbb{R}^k \ : \ \boldsymbol{p} \geq \boldsymbol{0}, \ \sum_j p_j = 1\}$

Extended value functions $f : \mathbb{R}^k \rightarrow \mathbb{R} \cup \{\infty\}$

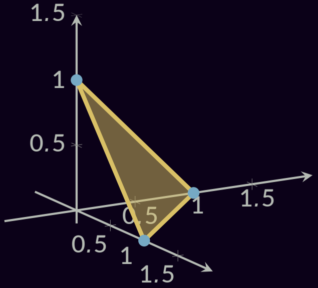
$\mathrm{dom}(f)$:= points where f is finite

Indicator function: $\iota_{\mathcal{S}}(\boldsymbol{x}) = \begin{cases} 0, & \boldsymbol{x} \in \mathcal{S} \\ \infty, & \boldsymbol{x} \notin \mathcal{S} \end{cases}$

($f + \iota_{\mathcal{S}}$ is f restricted to $\mathcal{S}$)

Fenchel conjugate of $f : \mathbb{R}^k \rightarrow \mathbb{R} \cup \{\infty\}$:

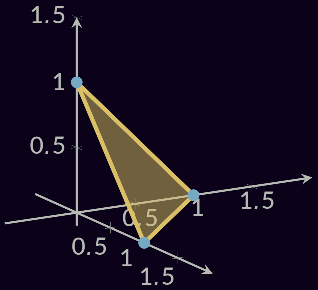
$$f^*(\boldsymbol{x}) := \sup_{\boldsymbol{p} \in \mathrm{dom}(f)} \langle \boldsymbol{p}, \boldsymbol{x} \rangle - f(\boldsymbol{p})$$



A Softmax Origin Story 🦸🏽‍♀️

Let $\Omega = \iota_\triangle$. Then,

$$\Omega^*(\boldsymbol{\theta}) = \max_{\boldsymbol{p} \in \triangle} \langle \boldsymbol{p}, \boldsymbol{\theta} \rangle$$



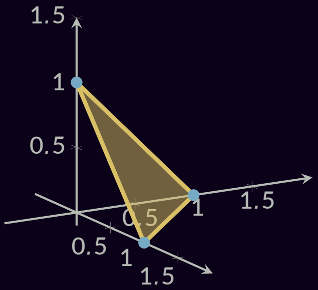
(Danskin, 1966)

A Softmax Origin Story 🦸🏽‍♀️

Let $\Omega = \iota_\triangle$. Then,

$$\Omega^*(\boldsymbol{\theta}) = \max_{\boldsymbol{p}\in\triangle}\langle \boldsymbol{p}, \boldsymbol{\theta}\rangle$$

$$\partial\Omega^*(\boldsymbol{\theta}) = \operatorname*{argmax}_{\boldsymbol{p}\in\triangle}\langle \boldsymbol{p}, \boldsymbol{\theta}\rangle$$



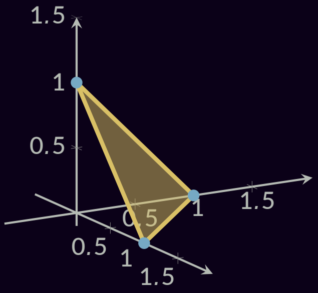
(Dantzig et al., 1955)

A Softmax Origin Story 🦸🏽‍♀️

Let $\Omega = \iota_\triangle$. Then,

$$\Omega^*(\boldsymbol{\theta}) = \max_{\boldsymbol{p}\in\triangle} \langle \boldsymbol{p}, \boldsymbol{\theta} \rangle = \max(\boldsymbol{\theta})$$

$$\partial\Omega^*(\boldsymbol{\theta}) = \operatorname*{argmax}_{\boldsymbol{p}\in\triangle} \langle \boldsymbol{p}, \boldsymbol{\theta} \rangle \ni \operatorname{argmax}(\boldsymbol{\theta})$$

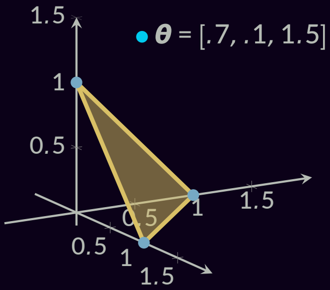


A Softmax Origin Story 🦸🏽‍♀️

Let $\Omega = \iota_\triangle$. Then,

$$\Omega^*(\boldsymbol{\theta}) = \max_{\boldsymbol{p}\in\triangle} \langle \boldsymbol{p}, \boldsymbol{\theta} \rangle = \max(\boldsymbol{\theta})$$

$$\partial\Omega^*(\boldsymbol{\theta}) = \operatorname*{argmax}_{\boldsymbol{p}\in\triangle} \langle \boldsymbol{p}, \boldsymbol{\theta} \rangle \ni \operatorname{argmax}(\boldsymbol{\theta})$$

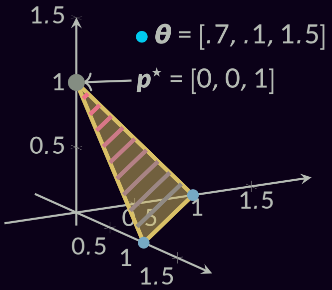


A Softmax Origin Story 🦸‍♀️

Let $\Omega = \iota_\triangle$. Then,

$$\Omega^*(\boldsymbol{\theta}) = \max_{\boldsymbol{p}\in\triangle} \langle \boldsymbol{p}, \boldsymbol{\theta} \rangle = \max(\boldsymbol{\theta})$$

$$\partial\Omega^*(\boldsymbol{\theta}) = \operatorname*{argmax}_{\boldsymbol{p}\in\triangle} \langle \boldsymbol{p}, \boldsymbol{\theta} \rangle \ni \operatorname{argmax}(\boldsymbol{\theta})$$



$$\operatorname*{argmax}_{\boldsymbol{p}\in\triangle} \langle \boldsymbol{p}, \boldsymbol{\theta} \rangle = \{\boldsymbol{p}^\star\}$$

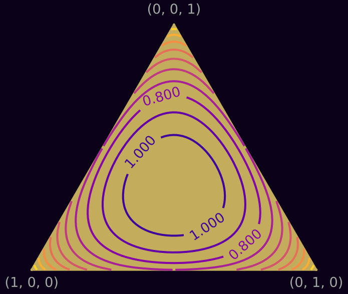
A Softmax Origin Story 🦸🏽‍♀️

Let $\Omega = \iota_\triangle$. Then,

$$\Omega^*(\boldsymbol{\theta}) = \max_{\boldsymbol{p} \in \triangle} \langle \boldsymbol{p}, \boldsymbol{\theta} \rangle \qquad = \max(\boldsymbol{\theta})$$

$$\partial\Omega^*(\boldsymbol{\theta}) = \operatorname*{argmax}_{\boldsymbol{p} \in \triangle} \langle \boldsymbol{p}, \boldsymbol{\theta} \rangle \qquad \ni \operatorname{argmax}(\boldsymbol{\theta})$$

Shannon entropy of $\boldsymbol{p}$ $\quad \mathsf{H}_1(\boldsymbol{p}) := -\sum_j p_j \log p_j$



A Softmax Origin Story 🦸🏽‍♀️

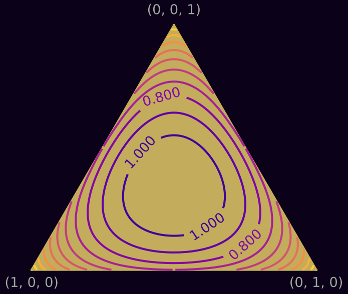
Let $\Omega = \iota_\triangle$. Then,

$$\Omega^*(\boldsymbol{\theta}) = \max_{\boldsymbol{p}\in\triangle} \langle \boldsymbol{p}, \boldsymbol{\theta} \rangle = \max(\boldsymbol{\theta})$$

$$\partial\Omega^*(\boldsymbol{\theta}) = \operatorname{argmax}_{\boldsymbol{p}\in\triangle} \langle \boldsymbol{p}, \boldsymbol{\theta} \rangle \ni \operatorname{argmax}(\boldsymbol{\theta})$$

Shannon entropy of $\boldsymbol{p}$ $\quad \mathrm{H}_1(\boldsymbol{p}) := -\sum_j p_j \log p_j$

Let $\Omega = -\mathrm{H}_1(\boldsymbol{p}) + \iota_\triangle$. Then,



A Softmax Origin Story 🦸🏽‍♀️

Let $\Omega = \iota_\triangle$. Then,

$$\Omega^*(\boldsymbol{\theta}) = \max_{\boldsymbol{p}\in\triangle} \langle \boldsymbol{p}, \boldsymbol{\theta} \rangle = \max(\boldsymbol{\theta})$$

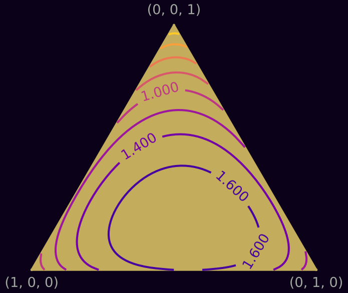
$$\partial\Omega^*(\boldsymbol{\theta}) = \operatorname*{argmax}_{\boldsymbol{p}\in\triangle} \langle \boldsymbol{p}, \boldsymbol{\theta} \rangle \ni \operatorname{argmax}(\boldsymbol{\theta})$$

Shannon entropy of $\boldsymbol{p}$ $\quad \mathrm{H}_1(\boldsymbol{p}) := -\sum_j p_j \log p_j$

Let $\Omega = -\mathrm{H}_1(\boldsymbol{p}) + \iota_\triangle$. Then,

$$\Omega^*(\boldsymbol{\theta}) = \max_{\boldsymbol{p}\in\triangle} \langle \boldsymbol{p}, \boldsymbol{\theta} \rangle + \mathrm{H}_1(\boldsymbol{p}) = \operatorname{logsumexp}(\boldsymbol{\theta})$$

$$\nabla\Omega^*(\boldsymbol{\theta}) = \operatorname*{argmax}_{\boldsymbol{p}\in\triangle} \langle \boldsymbol{p}, \boldsymbol{\theta} \rangle + \mathrm{H}_1(\boldsymbol{p}) = \operatorname{softmax}(\boldsymbol{\theta})$$



(Boyd and Vandenberghe, 2004)

A Softmax Origin Story 🦸🏽‍♀️

Let $\Omega = \iota_\triangle$. Then,

$$\Omega^*(\boldsymbol{\theta}) = \max_{\boldsymbol{p}\in\triangle}\langle \boldsymbol{p}, \boldsymbol{\theta}\rangle = \max(\boldsymbol{\theta})$$

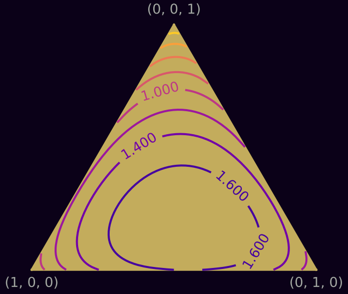
$$\partial\Omega^*(\boldsymbol{\theta}) = \operatorname{argmax}_{\boldsymbol{p}\in\triangle}\langle \boldsymbol{p}, \boldsymbol{\theta}\rangle \ni \operatorname{argmax}(\boldsymbol{\theta})$$

Shannon entropy of $\boldsymbol{p}$ $\quad \mathsf{H}_1(\boldsymbol{p}) := -\sum_j p_j \log p_j$

Let $\Omega = -\mathsf{H}_1(\boldsymbol{p}) + \iota_\triangle$. Then,

$$\Omega^*(\boldsymbol{\theta}) = \max_{\boldsymbol{p}\in\triangle}\langle \boldsymbol{p}, \boldsymbol{\theta}\rangle + \mathsf{H}_1(\boldsymbol{p}) = \operatorname{logsumexp}(\boldsymbol{\theta})$$

$$\nabla\Omega^*(\boldsymbol{\theta}) = \operatorname{argmax}_{\boldsymbol{p}\in\triangle}\langle \boldsymbol{p}, \boldsymbol{\theta}\rangle + \mathsf{H}_1(\boldsymbol{p}) = \operatorname{softmax}(\boldsymbol{\theta})$$



Softmax is an entropy-regularized argmax!

Outline

1. Warm-Up: Well-Known Losses and Mappings
2. Regularized Prediction Functions
3. Fenchel-Young Losses
4. Sparse Sequence-to-Sequence Models
5. Adaptively Sparse Transformers
6. Sparse Structured Prediction

(Niculae and Blondel, 2017)

Regularized Prediction Functions

A family of softmax-like mappings

$$\boldsymbol{\pi}_\Omega(\boldsymbol{\theta}) = \operatorname*{argmax}_{\boldsymbol{p} \in \operatorname{dom}(\Omega)} \langle \boldsymbol{p}, \boldsymbol{\theta} \rangle - \Omega(\boldsymbol{p}) \quad = \nabla \Omega^*(\boldsymbol{\theta})$$

(Niculae and Blondel, 2017)

Regularized Prediction Functions

A family of softmax-like mappings

$$\boldsymbol{\pi}_\Omega(\boldsymbol{\theta}) = \underset{\boldsymbol{p}\in\mathrm{dom}(\Omega)}{\mathrm{argmax}} \langle \boldsymbol{p}, \boldsymbol{\theta} \rangle - \Omega(\boldsymbol{p}) \quad = \nabla\Omega^*(\boldsymbol{\theta})$$

Let $\mathrm{dom}(\Omega) = \triangle$. We recover

(Niculae and Blondel, 2017)

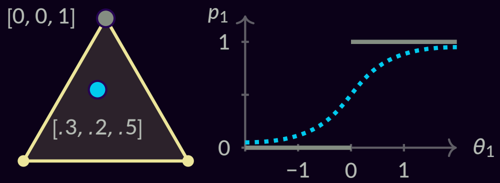
Regularized Prediction Functions

A family of softmax-like mappings

$$\boldsymbol{\pi}_\Omega(\boldsymbol{\theta}) = \underset{\boldsymbol{p}\in\mathrm{dom}(\Omega)}{\mathrm{argmax}}\ \langle \boldsymbol{p}, \boldsymbol{\theta}\rangle - \Omega(\boldsymbol{p}) \quad = \nabla\Omega^*(\boldsymbol{\theta})$$

Let $\mathrm{dom}(\Omega) = \triangle$. We recover

- argmax: $\Omega(\boldsymbol{p}) = 0$
- softmax: $\Omega(\boldsymbol{p}) = -\mathrm{H}_1(\boldsymbol{p}) = \sum_j p_j \log p_j$



(Niculae and Blondel, 2017)

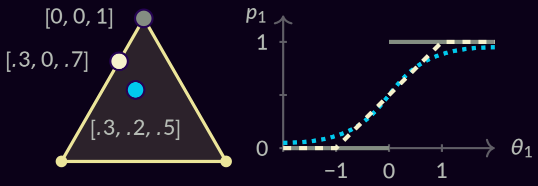
Regularized Prediction Functions

A family of softmax-like mappings

$$\boldsymbol{\pi}_\Omega(\boldsymbol{\theta}) = \underset{\boldsymbol{p} \in \mathrm{dom}(\Omega)}{\mathrm{argmax}} \langle \boldsymbol{p}, \boldsymbol{\theta} \rangle - \Omega(\boldsymbol{p}) \quad = \nabla\Omega^*(\boldsymbol{\theta})$$

Let $\mathrm{dom}(\Omega) = \triangle$. We recover

- argmax: $\Omega(\boldsymbol{p}) = 0$
- softmax: $\Omega(\boldsymbol{p}) = -\mathsf{H}_1(\boldsymbol{p}) = \sum_j p_j \log p_j$
- sparsemax: $\Omega(\boldsymbol{p}) = -\mathsf{H}_2(\boldsymbol{p}) = 1/2 \sum_j p_j(p_j - 1)$



(Niculae and Blondel, 2017)

Regularized Prediction Functions

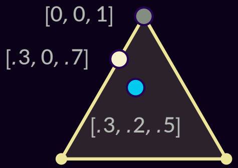
Regularization brings:

(Niculae and Blondel, 2017)

Regularized Prediction Functions

Regularization brings:

- improved uncertainty handling: (predictions become hedged bets)

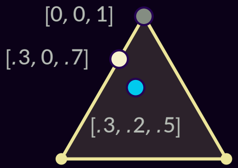


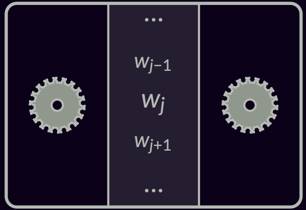
(Niculae and Blondel, 2017)

Regularized Prediction Functions

Regularization brings:

- improved uncertainty handling: (predictions become hedged bets)
- smoothing effect (Nesterov, 2005; Kakade et al., 2009)
  Ω strongly convex $\Rightarrow \Omega^*$ smooth,
  $\Rightarrow \boldsymbol{\pi}_\Omega$ differentiable almost everywhere



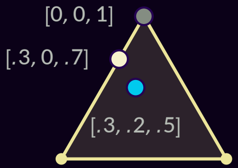


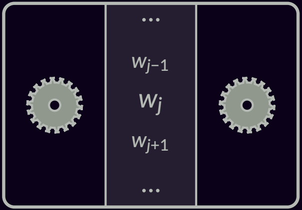
(Niculae and Blondel, 2017)

Regularized Prediction Functions

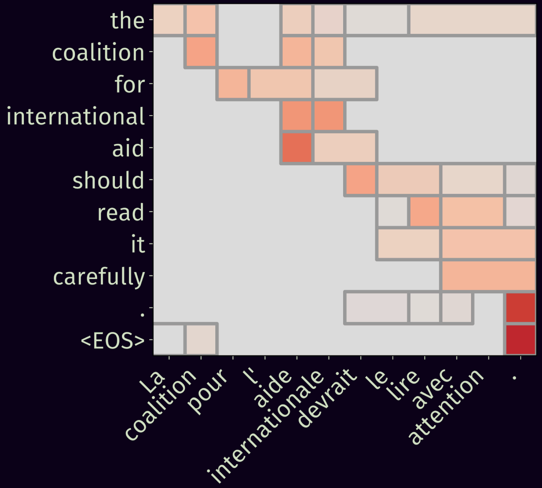
Regularization brings:

- improved uncertainty handling:
  (predictions become **hedged bets**)
- smoothing effect (Nesterov, 2005; Kakade et al., 2009)
  Ω strongly convex $\Rightarrow \Omega^*$ smooth,
  $\Rightarrow \boldsymbol{\pi}_\Omega$ **differentiable** almost everywhere
- ability to add **inductive bias**





(Niculae and Blondel, 2017)



fusedmax: $\Omega(\boldsymbol{p}) = -\mathrm{H}_2(\boldsymbol{p}) + \sum_{j=1}^{k} |p_i - p_{i-1}|$

Outline

1. Warm-Up: Well-Known Losses and Mappings
2. Regularized Prediction Functions
3. Fenchel-Young Losses
4. Sparse Sequence-to-Sequence Models
5. Adaptively Sparse Transformers
6. Sparse Structured Prediction

$$\text{perceptron} \iff \text{argmax}$$
$$\text{logistic regression} \iff \text{softmax}$$

What motivates this connection?

(Niculae, Martins, Blondel, and Cardie, 2018; Blondel, Martins, and Niculae, 2019)

Fenchel-Young Losses

$$L_\Omega(\boldsymbol{\theta}; \boldsymbol{y}^{\text{true}}) := \Omega^*(\boldsymbol{\theta}) + \Omega(\boldsymbol{y}^{\text{true}}) - \langle \boldsymbol{\theta}, \boldsymbol{y}^{\text{true}} \rangle$$

Ω: a regularizer

$\boldsymbol{y}^{\text{true}} \in \text{dom}(\Omega)$: target (e.g. $\boldsymbol{e}_k$)

$\boldsymbol{\theta} \in \mathbb{R}^d$: prediction scores

(Niculae, Martins, Blondel, and Cardie, 2018; Blondel, Martins, and Niculae, 2019)

Fenchel-Young Losses

$$L_\Omega(\boldsymbol{\theta}; \boldsymbol{y}^{\text{true}}) := \Omega^*(\boldsymbol{\theta}) + \Omega(\boldsymbol{y}^{\text{true}}) - \langle \boldsymbol{\theta}, \boldsymbol{y}^{\text{true}} \rangle$$

Ω: a regularizer

$\boldsymbol{y}^{\text{true}} \in \text{dom}(\Omega)$: target (e.g. $\boldsymbol{e}_k$)

$\boldsymbol{\theta} \in \mathbb{R}^d$: prediction scores

Based on the FY inequality:

$$\Omega^\star(\boldsymbol{\theta}) + \Omega(\boldsymbol{p}) \geq \langle \boldsymbol{\theta}, \boldsymbol{p} \rangle$$

Properties:

1. Non-negativity:

$$L_\Omega(\boldsymbol{\theta}; \boldsymbol{y}^{\text{true}}) \geq 0$$

(Niculae, Martins, Blondel, and Cardie, 2018; Blondel, Martins, and Niculae, 2019)

Fenchel-Young Losses

$$L_\Omega(\boldsymbol{\theta}; \boldsymbol{y}^{\text{true}}) := \Omega^*(\boldsymbol{\theta}) + \Omega(\boldsymbol{y}^{\text{true}}) - \langle \boldsymbol{\theta}, \boldsymbol{y}^{\text{true}} \rangle$$

Ω: a regularizer

$\boldsymbol{y}^{\text{true}} \in \text{dom}(\Omega)$: target (e.g. $\boldsymbol{e}_k$)

$\boldsymbol{\theta} \in \mathbb{R}^d$: prediction scores

Based on the FY inequality:

$$\Omega^\star(\boldsymbol{\theta}) + \Omega(\boldsymbol{p}) \geq \langle \boldsymbol{\theta}, \boldsymbol{p} \rangle$$

Properties:

1. Non-negativity:

$$L_\Omega(\boldsymbol{\theta}; \boldsymbol{y}^{\text{true}}) \geq 0$$

2. Zero loss:

$$L_\Omega(\boldsymbol{\theta}; \boldsymbol{y}^{\text{true}}) = 0 \iff \boldsymbol{\pi}_\Omega(\boldsymbol{\theta}) = \boldsymbol{y}^{\text{true}}$$

The natural loss for the mapping $\boldsymbol{\pi}_\Omega$.

(Niculae, Martins, Blondel, and Cardie, 2018; Blondel, Martins, and Niculae, 2019)

Fenchel-Young Losses

$$L_\Omega(\boldsymbol{\theta}; \boldsymbol{y}^{\text{true}}) := \Omega^*(\boldsymbol{\theta}) + \Omega(\boldsymbol{y}^{\text{true}}) - \langle \boldsymbol{\theta}, \boldsymbol{y}^{\text{true}} \rangle$$

Ω: a regularizer

$\boldsymbol{y}^{\text{true}} \in \text{dom}(\Omega)$: target (e.g. $\boldsymbol{e}_k$)

$\boldsymbol{\theta} \in \mathbb{R}^d$: prediction scores

Based on the FY inequality:

$$\Omega^\star(\boldsymbol{\theta}) + \Omega(\boldsymbol{p}) \geq \langle \boldsymbol{\theta}, \boldsymbol{p} \rangle$$

The natural loss for the mapping $\boldsymbol{\pi}_\Omega$.

Properties:

1. Non-negativity:

$$L_\Omega(\boldsymbol{\theta}; \boldsymbol{y}^{\text{true}}) \geq 0$$

2. Zero loss:

$$L_\Omega(\boldsymbol{\theta}; \boldsymbol{y}^{\text{true}}) = 0 \iff \boldsymbol{\pi}_\Omega(\boldsymbol{\theta}) = \boldsymbol{y}^{\text{true}}$$

3. Convex and differentiable:

$$\nabla_{\boldsymbol{\theta}} L_\Omega(\boldsymbol{\theta}; \boldsymbol{y}^{\text{true}}) = \boldsymbol{\pi}_\Omega(\boldsymbol{\theta}) - \boldsymbol{y}^{\text{true}}$$

(Blondel, Martins, and Niculae, 2019)

Well-Known Fenchel-Young Losses

	$\text{dom}(\Omega)$	$\Omega(\boldsymbol{p})$	$\boldsymbol{\pi}_\Omega(\boldsymbol{\theta})$
Perceptron	$\triangle^k$	0	$\text{argmax}(\boldsymbol{\theta})$
Logistic Regression	$\triangle^k$	$-\text{H}_1(\boldsymbol{p})$	$\text{softmax}(\boldsymbol{\theta})$
Sparsemax	$\triangle^k$	$-\text{H}_2(\boldsymbol{p})$	$\text{sparsemax}(\boldsymbol{\theta})$

(Blondel, Martins, and Niculae, 2019)

Well-Known Fenchel-Young Losses

	$\mathrm{dom}(\Omega)$	$\Omega(\boldsymbol{p})$	$\boldsymbol{\pi}_\Omega(\boldsymbol{\theta})$
Perceptron	$\triangle^k$	0	$\mathrm{argmax}(\boldsymbol{\theta})$
Logistic Regression	$\triangle^k$	$-\mathrm{H}_1(\boldsymbol{p})$	$\mathrm{softmax}(\boldsymbol{\theta})$
Sparsemax	$\triangle^k$	$-\mathrm{H}_2(\boldsymbol{p})$	$\mathrm{sparsemax}(\boldsymbol{\theta})$
Hinge (SVM)	$\triangle^k$	$\langle \boldsymbol{p}, \boldsymbol{y}^{\mathrm{true}} - \mathbf{1} \rangle$	$\mathrm{argmax}(\mathbf{1} - \boldsymbol{y}^{\mathrm{true}} + \boldsymbol{\theta})$

Well-Known Fenchel-Young Losses

	$\mathrm{dom}(\Omega)$	$\Omega(\boldsymbol{p})$	$\boldsymbol{\pi}_\Omega(\boldsymbol{\theta})$
Perceptron	$\triangle^k$	0	$\mathrm{argmax}(\boldsymbol{\theta})$
Logistic Regression	$\triangle^k$	$-\mathrm{H}_1(\boldsymbol{p})$	$\mathrm{softmax}(\boldsymbol{\theta})$
Sparsemax	$\triangle^k$	$-\mathrm{H}_2(\boldsymbol{p})$	$\mathrm{sparsemax}(\boldsymbol{\theta})$
Hinge (SVM)	$\triangle^k$	$\langle \boldsymbol{p}, \boldsymbol{y}^{\mathrm{true}} - \mathbf{1} \rangle$	$\mathrm{argmax}(\mathbf{1} - \boldsymbol{y}^{\mathrm{true}} + \boldsymbol{\theta})$
Squared	$\mathbb{R}^k$	$\frac{1}{2}\lVert\boldsymbol{p}\rVert^2$	$\boldsymbol{\theta}$

(Blondel, Martins, and Niculae, 2019)

Well-Known Fenchel-Young Losses

	$\mathrm{dom}(\Omega)$	$\Omega(\boldsymbol{p})$	$\boldsymbol{\pi}_\Omega(\boldsymbol{\theta})$
Perceptron	$\triangle^k$	0	$\mathrm{argmax}(\boldsymbol{\theta})$
Logistic Regression	$\triangle^k$	$-\mathrm{H}_1(\boldsymbol{p})$	$\mathrm{softmax}(\boldsymbol{\theta})$
Sparsemax	$\triangle^k$	$-\mathrm{H}_2(\boldsymbol{p})$	$\mathrm{sparsemax}(\boldsymbol{\theta})$
Hinge (SVM)	$\triangle^k$	$\langle \boldsymbol{p}, \boldsymbol{y}^{\mathrm{true}} - \mathbf{1} \rangle$	$\mathrm{argmax}(\mathbf{1} - \boldsymbol{y}^{\mathrm{true}} + \boldsymbol{\theta})$
Squared	$\mathbb{R}^k$	$\frac{1}{2}\lVert\boldsymbol{p}\rVert^2$	$\boldsymbol{\theta}$
One-vs-all	$[0, 1]^k$	$-\sum_j \mathrm{H}_1([p_i, 1 - p_i])$	$\mathrm{sigmoid}(\boldsymbol{\theta})$

(Blondel, Martins, and Niculae, 2019)

Well-Known Fenchel-Young Losses

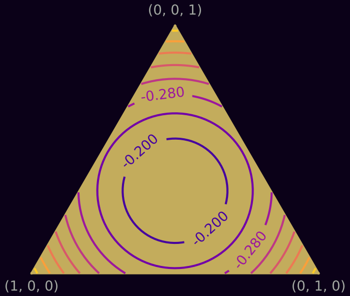
	$\text{dom}(\Omega)$	$\Omega(\boldsymbol{p})$	$\boldsymbol{\pi}_\Omega(\boldsymbol{\theta})$
Perceptron	$\triangle^k$	0	$\text{argmax}(\boldsymbol{\theta})$
Logistic Regression	$\triangle^k$	$-\text{H}_1(\boldsymbol{p})$	$\text{softmax}(\boldsymbol{\theta})$
Sparsemax	$\triangle^k$	$-\text{H}_2(\boldsymbol{p})$	$\text{sparsemax}(\boldsymbol{\theta})$
Hinge (SVM)	$\triangle^k$	$\langle \boldsymbol{p}, \boldsymbol{y}^{\text{true}} - \mathbf{1} \rangle$	$\text{argmax}(\mathbf{1} - \boldsymbol{y}^{\text{true}} + \boldsymbol{\theta})$
Squared	$\mathbb{R}^k$	$\frac{1}{2}\|\boldsymbol{p}\|^2$	$\boldsymbol{\theta}$
One-vs-all	$[0, 1]^k$	$-\sum_j \text{H}_1([p_i, 1 - p_i])$	$\text{sigmoid}(\boldsymbol{\theta})$
... and more!			

(DeGroot, 1962; Grünwald and Dawid, 2004)

Generalized Entropies

A function H($\boldsymbol{p}$) quantifying uncertainty in $\boldsymbol{p} \in \Delta^k$:

1. H($\boldsymbol{p}$) = 0 if $\boldsymbol{p} \in \{\boldsymbol{e}_k\}$
2. H strictly concave
3. H($\boldsymbol{p}$) = H($\boldsymbol{Pp}$) (permutation-invariant)



Tsallis entropies, Rényi entropies, norm entropies, etc.

(Tsallis, 1988)

Tsallis Entropies

$$H_\alpha(\boldsymbol{p}) = \frac{1}{\alpha(\alpha-1)} \sum_j (p_j - p_j^\alpha)$$

$\alpha \to 1$	Shannon
$\alpha = 2$	Gini
$\alpha \to \infty$	0

(Tsallis, 1988)

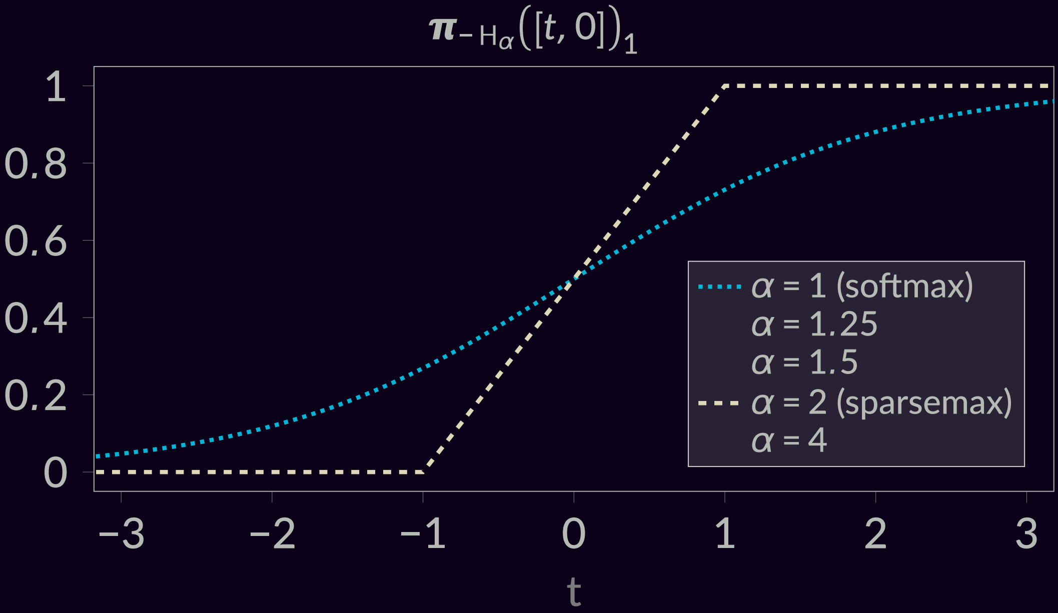
Tsallis Entropies

$$H_\alpha(\boldsymbol{p}) = \frac{1}{\alpha(\alpha-1)} \sum_j (p_j - p_j^\alpha)$$

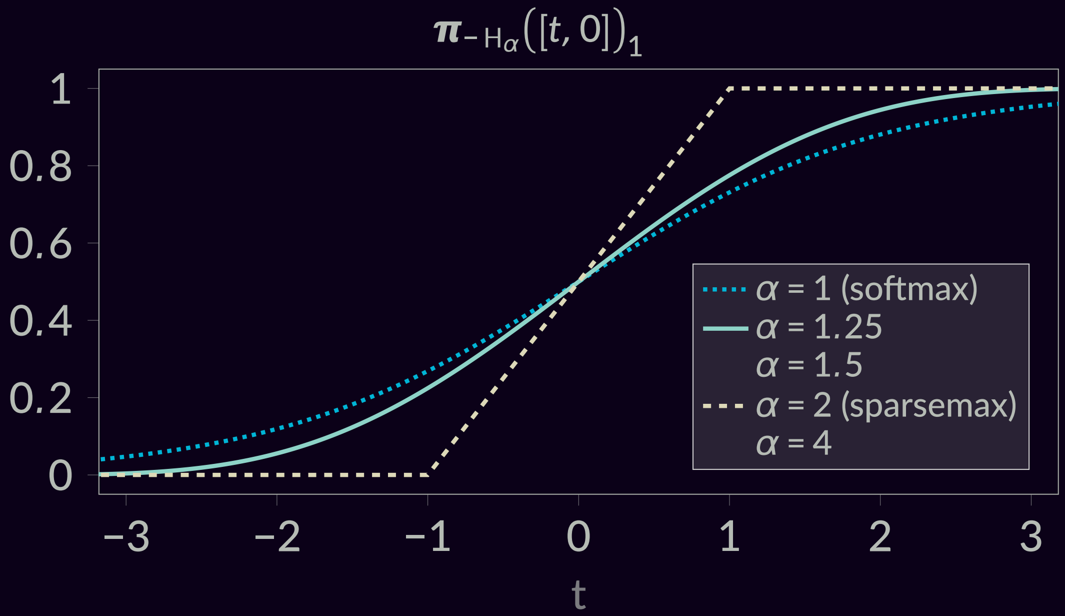
$\alpha \to 1$	Shannon
$\alpha = 2$	Gini
$\alpha \to \infty$	0

generate Tsallis α-entmax mappings & losses!

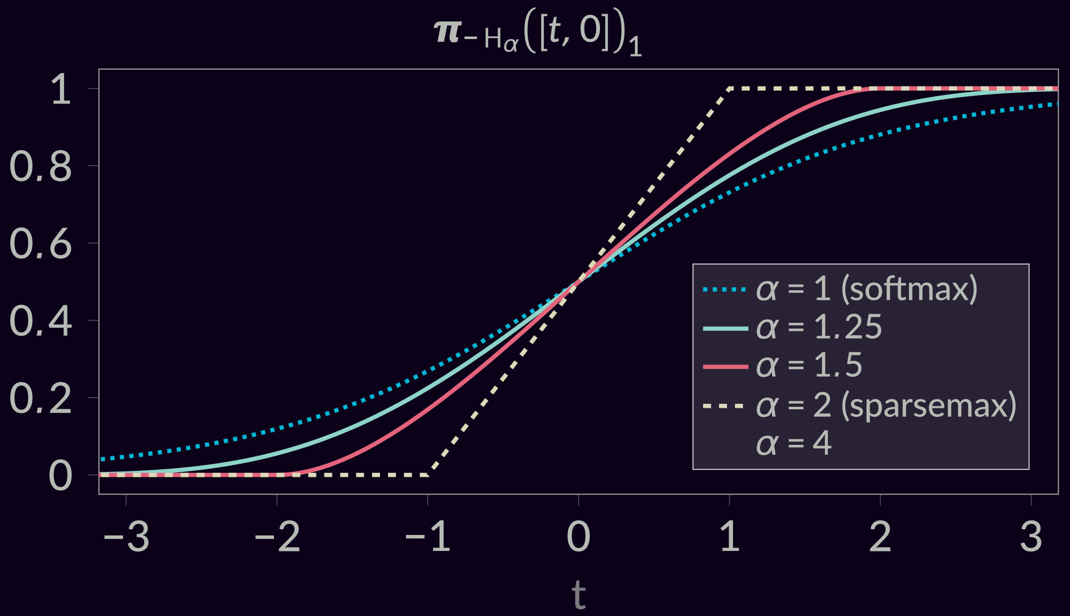
(Blondel, Martins, and Niculae, 2019)



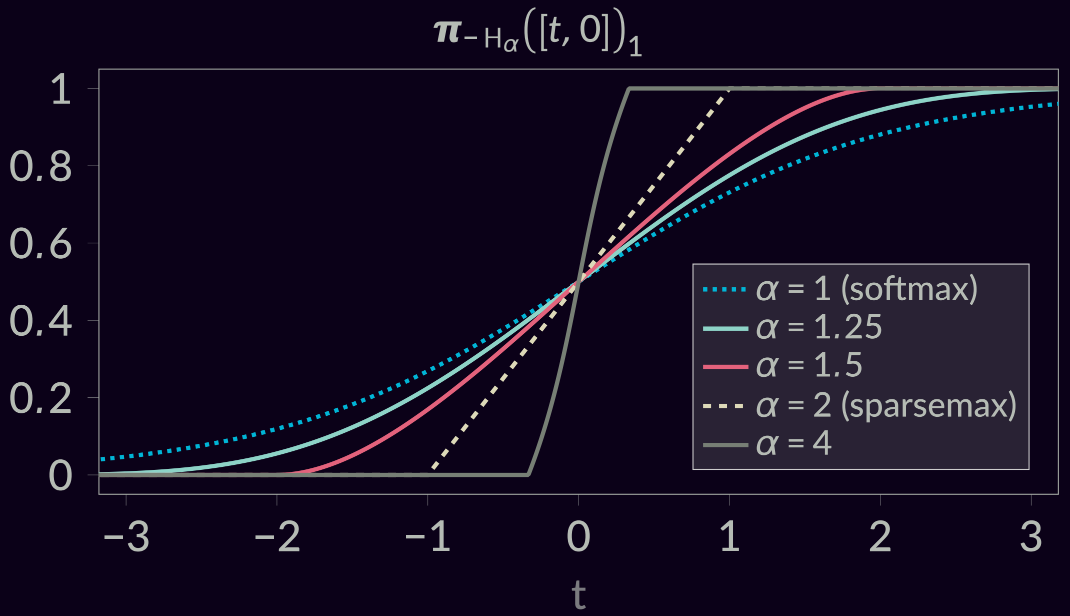
(Blondel, Martins, and Niculae, 2019)



(Blondel, Martins, and Niculae, 2019)



(Blondel, Martins, and Niculae, 2019)

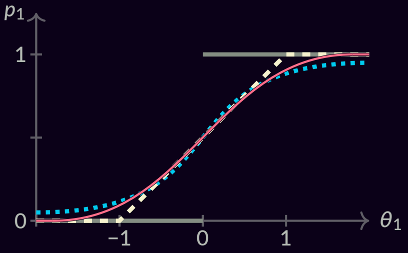

$\pi_{-H_\alpha}([t, 0])_1$
1
0.8
0.6
0.4
0.2
0
−3
−2
−1
0
1
2
3
t
α = 1 (softmax)
α = 1.25
α = 1.5
α = 2 (sparsemax)
α = 4

(Blondel, Martins, and Niculae, 2019)

Properties of α-entmax Mappings & Losses

$\boldsymbol{\pi}_{-\mathsf{H}_\alpha}$ is sparse for $\alpha > 1$

(Novel general condition:
$\boldsymbol{\pi}_\Omega$ is sparse iff. $\partial\Omega(\boldsymbol{p}) \neq \varnothing$ for any $\boldsymbol{p} \in \triangle$)



(Blondel, Martins, and Niculae, 2019)

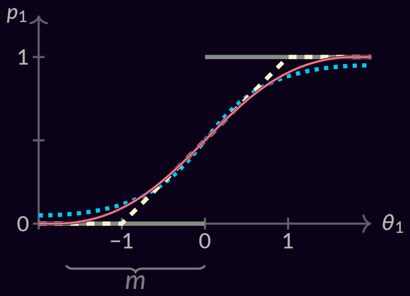
Properties of α-entmax Mappings & Losses

$\boldsymbol{\pi}_{-\mathsf{H}_\alpha}$ is sparse for $\alpha > 1$

(Novel general condition:
$\boldsymbol{\pi}_\Omega$ is sparse iff. $\partial\Omega(\boldsymbol{p}) \neq \varnothing$ for any $\boldsymbol{p} \in \triangle$)

$L_{-\mathsf{H}_\alpha}$ has the margin property:

$$\theta_k \geq \underbrace{1/\alpha{-}1}_{m} + \max_{j \neq k} \theta_j \Rightarrow L_{-\mathsf{H}_\alpha}(\boldsymbol{\theta}; \boldsymbol{e}_k) = 0$$



(Equivalence result between sparsity and margins)

(Peters, Niculae, and Martins, 2019)

Computing α-entmax

$$\boldsymbol{\pi}_{-\mathsf{H}_\alpha}(\boldsymbol{\theta}) := \underset{\boldsymbol{p}\in\triangle}{\operatorname{argmax}} \langle \boldsymbol{p}, \boldsymbol{\theta} \rangle + \mathsf{H}_\alpha(\boldsymbol{p})$$

(Peters, Niculae, and Martins, 2019)

Computing α-entmax

$$\boldsymbol{\pi}_{-\mathrm{H}_\alpha}(\boldsymbol{\theta}) := \operatorname*{argmax}_{\boldsymbol{p}\in\triangle} \langle \boldsymbol{p}, \boldsymbol{\theta}\rangle + \mathrm{H}_\alpha(\boldsymbol{p})$$

Solution has the form:

$$\boldsymbol{\pi}_{-\mathrm{H}_\alpha}(\boldsymbol{\theta}) = [(\alpha - 1)\boldsymbol{\theta} - \tau\mathbf{1}]_+^{1/\alpha-1}$$

(Peters, Niculae, and Martins, 2019)

Computing α-entmax

$$\boldsymbol{\pi}_{-\mathrm{H}_\alpha}(\boldsymbol{\theta}) := \underset{\boldsymbol{p}\in\triangle}{\operatorname{argmax}} \langle \boldsymbol{p}, \boldsymbol{\theta} \rangle + \mathrm{H}_\alpha(\boldsymbol{p})$$

Solution has the form:

$$\boldsymbol{\pi}_{-\mathrm{H}_\alpha}(\boldsymbol{\theta}) = [(\alpha - 1)\boldsymbol{\theta} - \tau\mathbf{1}]_+^{1/\alpha-1}$$

Algorithms:

bisection

- approximate; bracket $\tau \in [\tau_{\text{lo}}, \tau_{\text{hi}}]$
- gain 1 bit per $\mathcal{O}(d)$ iteration
- `float32` has 23 mantissa bits

(Peters, Niculae, and Martins, 2019)

Computing α-entmax

$$\boldsymbol{\pi}_{-\mathsf{H}_\alpha}(\boldsymbol{\theta}) := \operatorname*{argmax}_{\boldsymbol{p}\in\triangle} \langle \boldsymbol{p}, \boldsymbol{\theta}\rangle + \mathsf{H}_\alpha(\boldsymbol{p})$$

Solution has the form:

$$\boldsymbol{\pi}_{-\mathsf{H}_\alpha}(\boldsymbol{\theta}) = [(\alpha - 1)\boldsymbol{\theta} - \tau\mathbf{1}]_+^{1/\alpha-1}$$

Algorithms:

bisection

- approximate; bracket $\tau \in [\tau_{\text{lo}}, \tau_{\text{hi}}]$
- gain 1 bit per $\mathcal{O}(d)$ iteration
- `float32` has 23 mantissa bits

sort-based

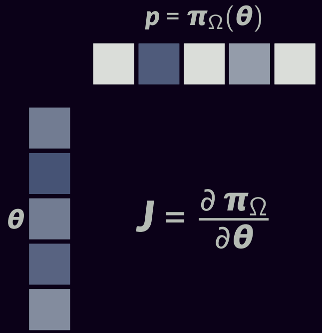
- exact algorithm, $\mathcal{O}(d \log d)$
- available only for $\alpha \in \{1.5, 2\}$
- For $\alpha = 2$, known since Held et al. (1974)!

(Peters, Niculae, and Martins, 2019)

Backward Pass

(general result)

$$\boldsymbol{\pi}_\Omega(\boldsymbol{\theta}) = \underset{\boldsymbol{p}\in\triangle}{\operatorname{argmax}} \langle \boldsymbol{p}, \boldsymbol{\theta} \rangle - \Omega(\boldsymbol{p}) \quad = \nabla\Omega^*(\boldsymbol{\theta})$$



$$\boldsymbol{J} = \frac{\partial\, \boldsymbol{\pi}_\Omega}{\partial \boldsymbol{\theta}}$$

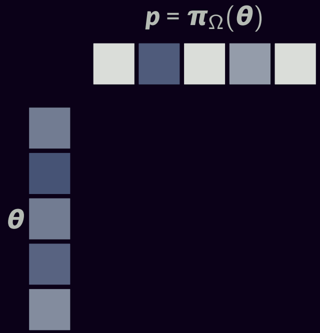
(Peters, Niculae, and Martins, 2019)

Backward Pass

(general result)

$$\boldsymbol{\pi}_\Omega(\boldsymbol{\theta}) = \underset{\boldsymbol{p}\in\triangle}{\operatorname{argmax}} \langle \boldsymbol{p}, \boldsymbol{\theta}\rangle - \Omega(\boldsymbol{p}) \quad = \nabla\Omega^*(\boldsymbol{\theta})$$

- $\boldsymbol{J}$ symmetric ($=\nabla\nabla\Omega^*$).



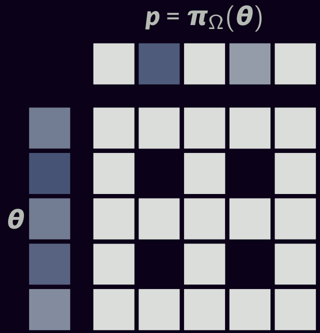
(Peters, Niculae, and Martins, 2019)

Backward Pass

(general result)

$$\boldsymbol{\pi}_\Omega(\boldsymbol{\theta}) = \underset{\boldsymbol{p}\in\triangle}{\operatorname{argmax}} \langle \boldsymbol{p}, \boldsymbol{\theta}\rangle - \Omega(\boldsymbol{p}) \quad = \nabla\Omega^*(\boldsymbol{\theta})$$

- $\boldsymbol{J}$ symmetric ($=\nabla\nabla\Omega^*$).
- $(\boldsymbol{J})_{ij} = 0$ if $p_i = 0$ or $p_j = 0$.



(Peters, Niculae, and Martins, 2019)

Backward Pass

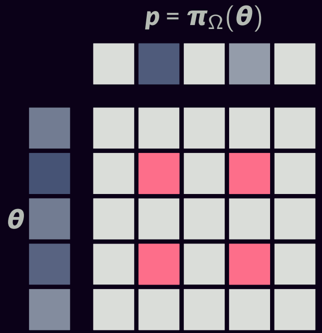
(general result)

$$\boldsymbol{\pi}_\Omega(\boldsymbol{\theta}) = \underset{\boldsymbol{p}\in\triangle}{\operatorname{argmax}} \langle \boldsymbol{p}, \boldsymbol{\theta}\rangle - \Omega(\boldsymbol{p}) \quad = \nabla\Omega^*(\boldsymbol{\theta})$$

- $\boldsymbol{J}$ symmetric $(=\nabla\nabla\Omega^*)$.
- $(\boldsymbol{J})_{ij} = 0$ if $p_i = 0$ or $p_j = 0$.
- Let $(\boldsymbol{H})_{ij} = \frac{\partial^2\Omega}{\partial p_i \partial p_j}(\boldsymbol{p})$ for nonzero i, j.

  Let $\boldsymbol{S} = \boldsymbol{H}^{-1}$ and $\boldsymbol{s} = \boldsymbol{1S}$.

  Then, $\bar{\boldsymbol{J}} = \boldsymbol{S} - \frac{1}{\langle \boldsymbol{1}, \boldsymbol{s}\rangle} \boldsymbol{s}\boldsymbol{s}^\top$.



(Peters, Niculae, and Martins, 2019)

Backward Pass

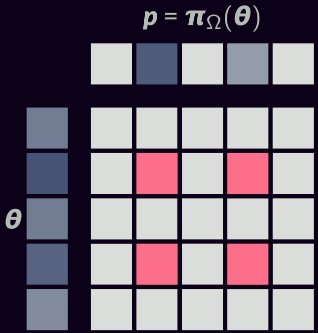
(general result)

$$\boldsymbol{\pi}_\Omega(\boldsymbol{\theta}) = \underset{\boldsymbol{p}\in\triangle}{\operatorname{argmax}} \langle \boldsymbol{p}, \boldsymbol{\theta}\rangle - \Omega(\boldsymbol{p}) \quad = \nabla\Omega^*(\boldsymbol{\theta})$$

- $\boldsymbol{J}$ symmetric ($=\nabla\nabla\Omega^*$).
- $(\boldsymbol{J})_{ij} = 0$ if $p_i = 0$ or $p_j = 0$.
- Let $(\boldsymbol{H})_{ij} = \frac{\partial^2\Omega}{\partial p_i \partial p_j}(\boldsymbol{p})$ for nonzero i, j.

  Let $\boldsymbol{S} = \boldsymbol{H}^{-1}$ and $\boldsymbol{s} = \boldsymbol{1S}$.

  Then, $\bar{\boldsymbol{J}} = \boldsymbol{S} - \frac{1}{\langle \boldsymbol{1}, \boldsymbol{s}\rangle} \boldsymbol{s}\boldsymbol{s}^\top$.
- For $-\mathrm{H}_\alpha$, $\boldsymbol{S} = \operatorname{diag}(\bar{\boldsymbol{p}}^{2-\alpha})$.



Outline

1. Warm-Up: Well-Known Losses and Mappings
2. Regularized Prediction Functions
3. Fenchel-Young Losses
4. Sparse Sequence-to-Sequence Models
5. Adaptively Sparse Transformers
6. Sparse Structured Prediction

(Bahdanau et al., 2015)

Sequence-to-Sequence With Attention

United Nations elections end today

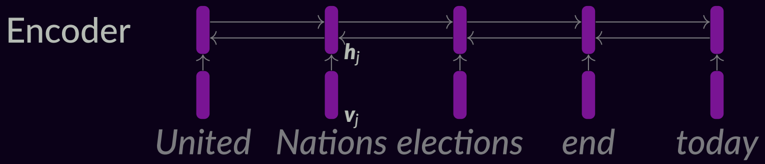
(Bahdanau et al., 2015)

Sequence-to-Sequence With Attention

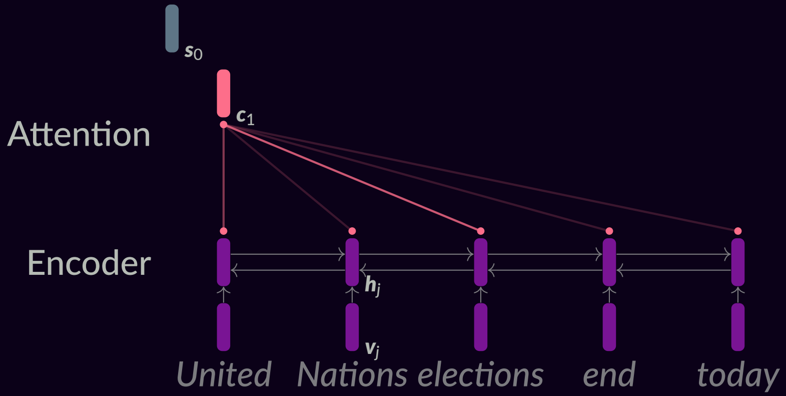
Encoder

(Bahdanau et al., 2015)

Sequence-to-Sequence With Attention



Sequence-to-Sequence With Attention



attention weights computed with *softmax*:

for some decoder state $\boldsymbol{s}_t$, compute contextually weighted average of input $\boldsymbol{c}_t$:

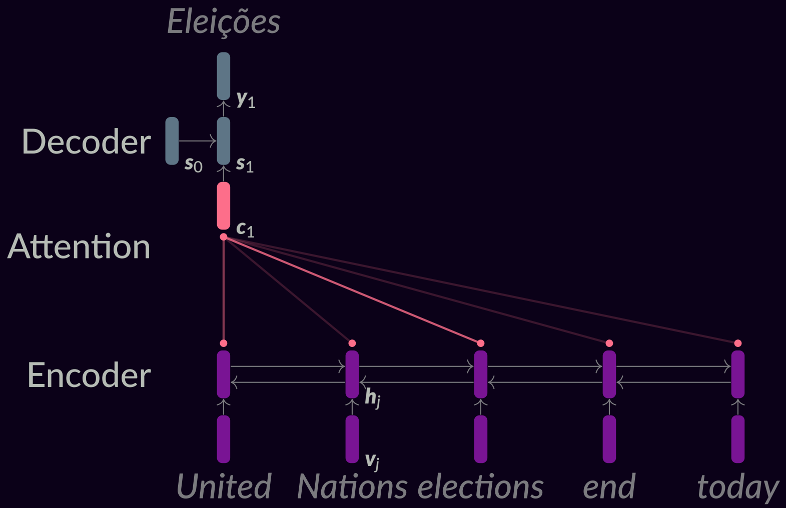
$$\theta_j = \boldsymbol{s}_t^\top \boldsymbol{W}^{(a)} \boldsymbol{h}_j$$

$$\boldsymbol{p} = \text{softmax}(\boldsymbol{\theta})$$

$$\boldsymbol{c}_t = \sum_j p_j \boldsymbol{h}_j$$

(Bahdanau et al., 2015)

Sequence-to-Sequence With Attention



predictive probability
(also using *softmax*!)

$$u_t = \tanh(\boldsymbol{W}^{(u)}[\boldsymbol{s}_t; \boldsymbol{c}_t])$$

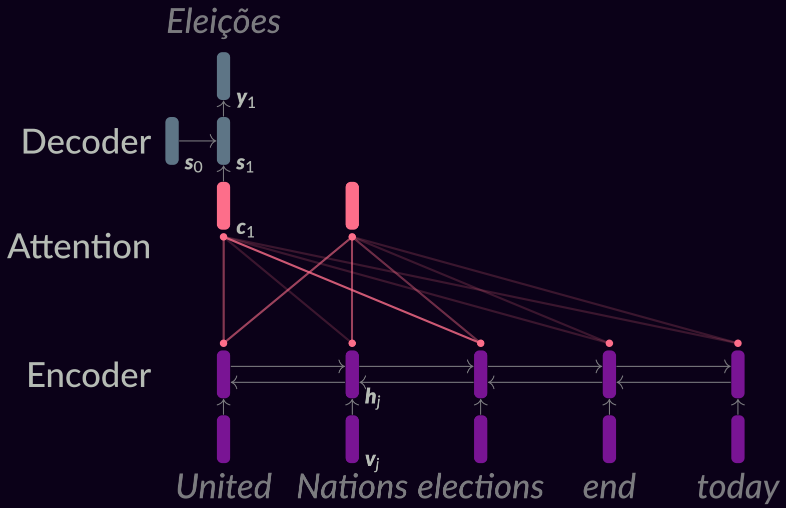
$$P(y_t \mid y_{1:t-1}, x) = \text{softmax}(\boldsymbol{V}\boldsymbol{u}_t)$$

$P(y_1 \mid x)$

.70	Eleições
.11	Os
.10	As
.09	Nações
	...
10^{-6}	Amsterdam

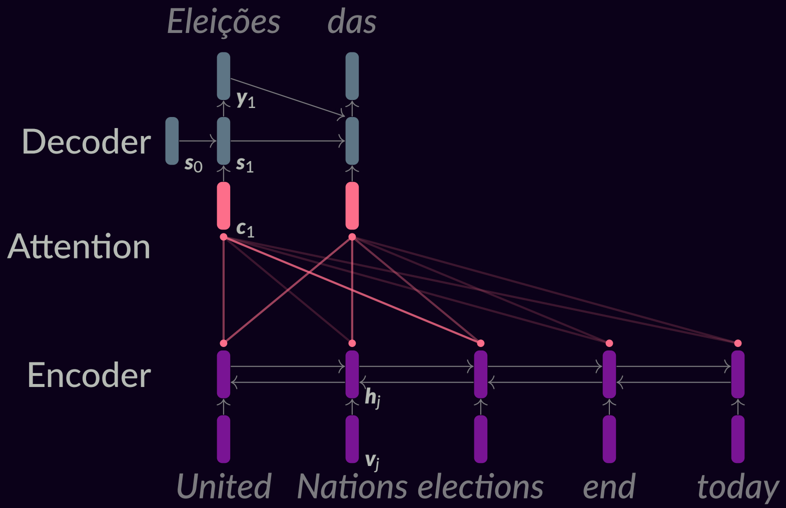
(Bahdanau et al., 2015)

Sequence-to-Sequence With Attention



(Bahdanau et al., 2015)

Sequence-to-Sequence With Attention



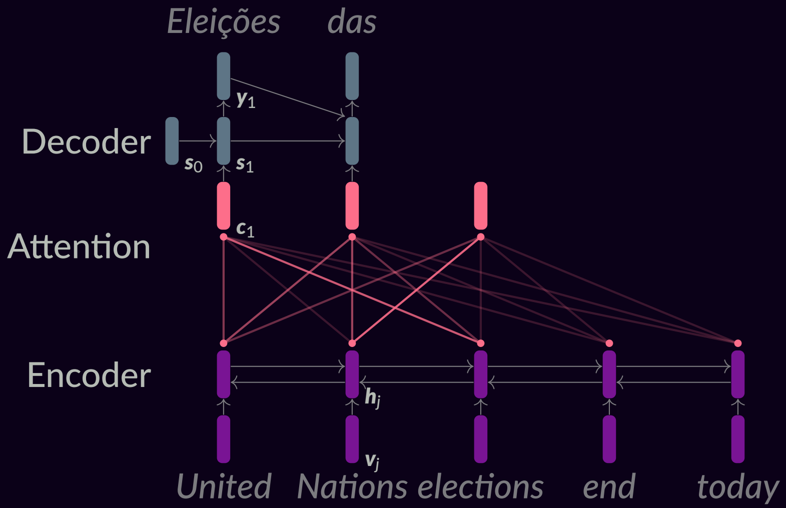
predictive probability

$P(y_2 \mid y_1, x)$

.40	das
.30	para
.20	ás
	...
10^{-7}	esquerda

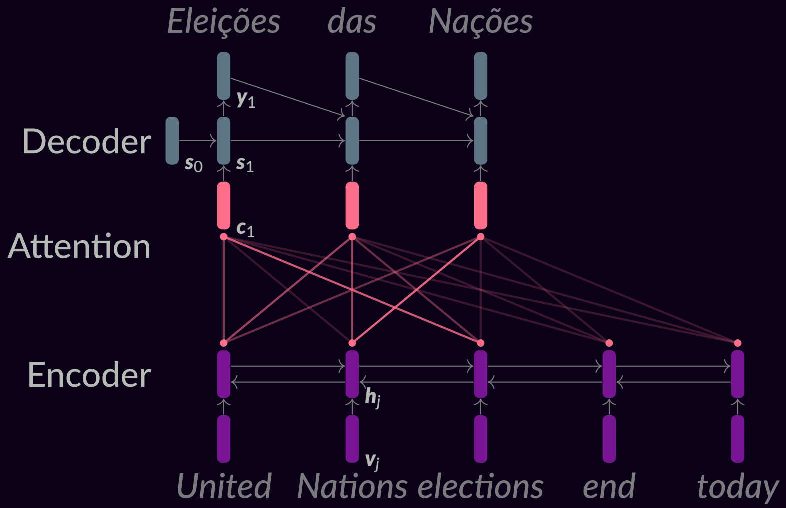
(Bahdanau et al., 2015)

Sequence-to-Sequence With Attention



(Bahdanau et al., 2015)

Sequence-to-Sequence With Attention



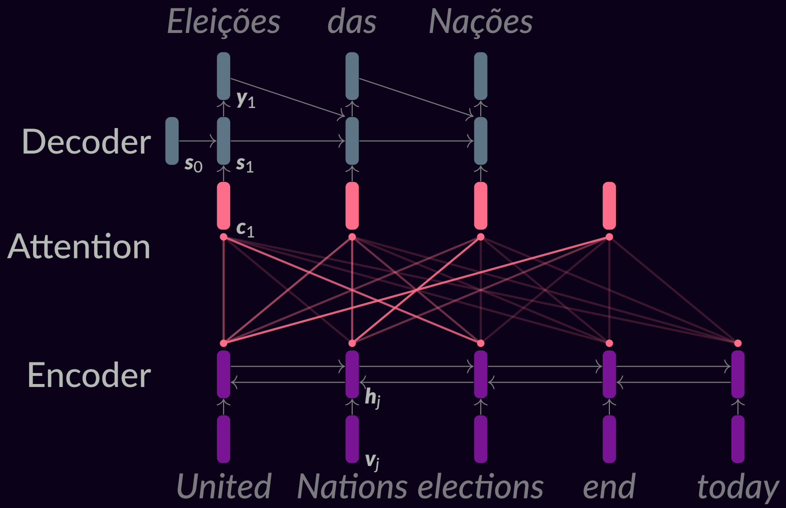
predictive probability

$P(y_3 \mid y_2, y_1, x)$

.80 Nações
.11 Representações
.03 assembleias
...
10^{-8} resultados

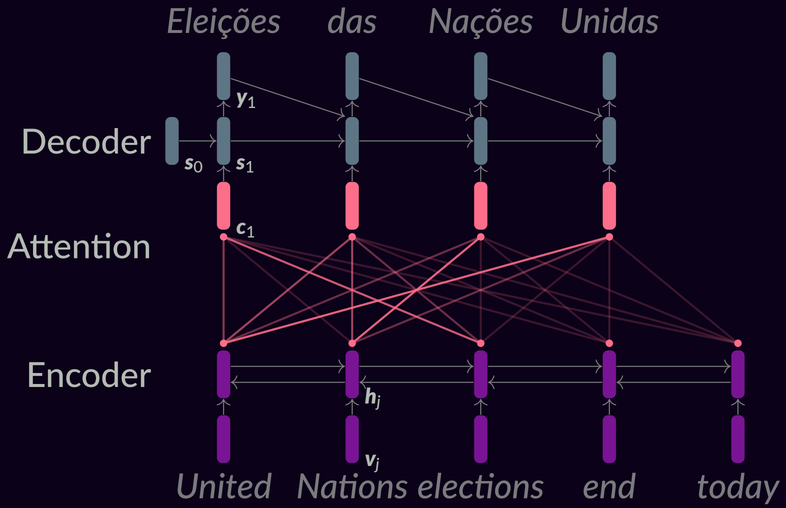
(Bahdanau et al., 2015)

Sequence-to-Sequence With Attention



(Bahdanau et al., 2015)

Sequence-to-Sequence With Attention



predictive probability

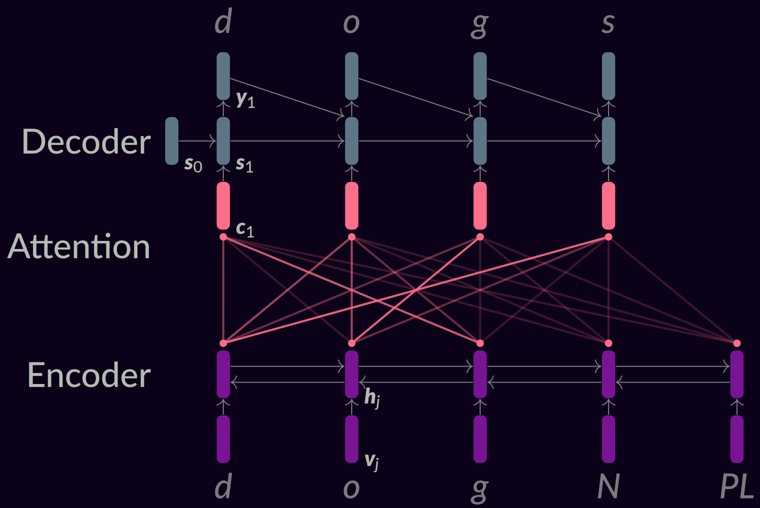
$P(y_4 \mid y_3, y_2, y_1, x)$

.90	Unidas
.05	Shopping
.01	,
	...
10^{-5}	aquático

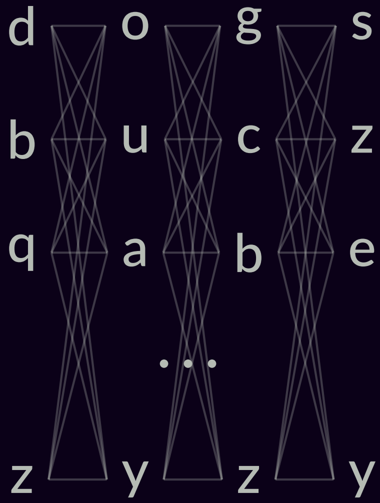
(Bahdanau et al., 2015)

Sequence-to-Sequence With Attention

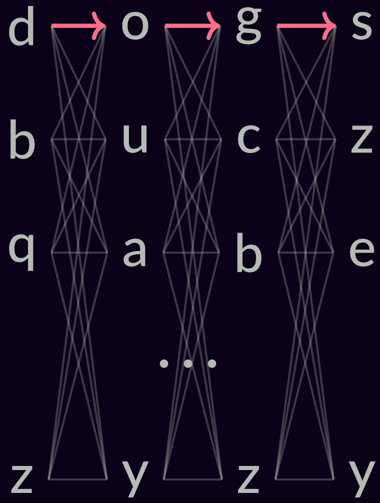
morphological inflection!



The Space of Outputs

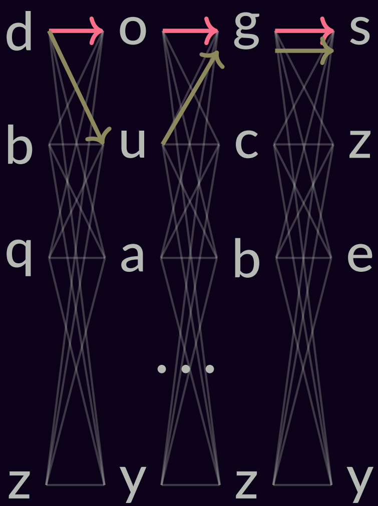


The Space of Outputs



$p(\cdot) = 0.60$

The Space of Outputs



$p(\cdot) = 0.60$

$p(\cdot) = 0.13$

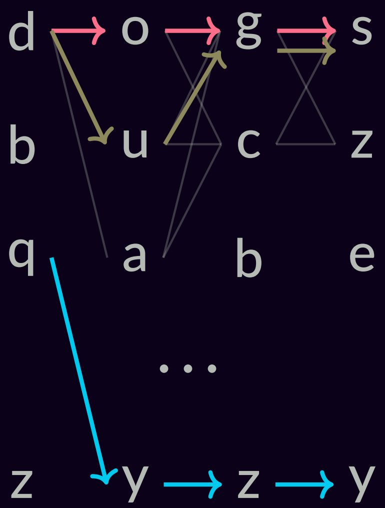
The Space of Outputs

$p(\cdot) = 0.60$

$p(\cdot) = 0.13$

$p(\cdot) = 10^{-4}$

The Space of Outputs: Made Sparse!



$p(\cdot) = 0.70$

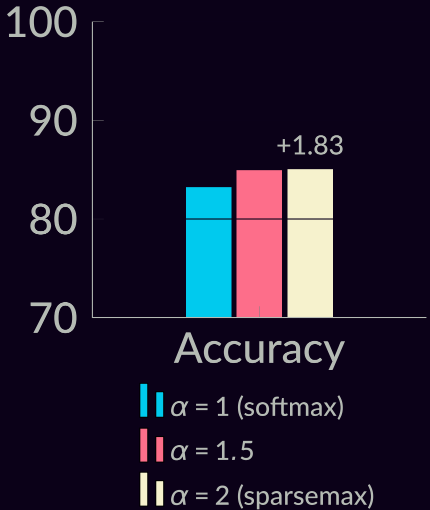
$p(\cdot) = 0.20$

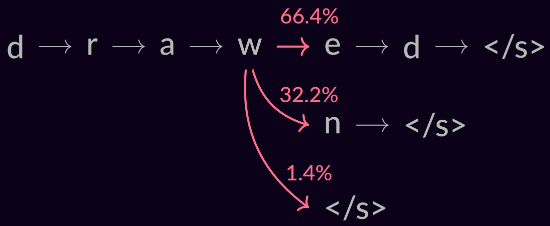
$p(\cdot) = 0$!!

(Peters, Niculae, and Martins, 2019)

Morphological Inflection

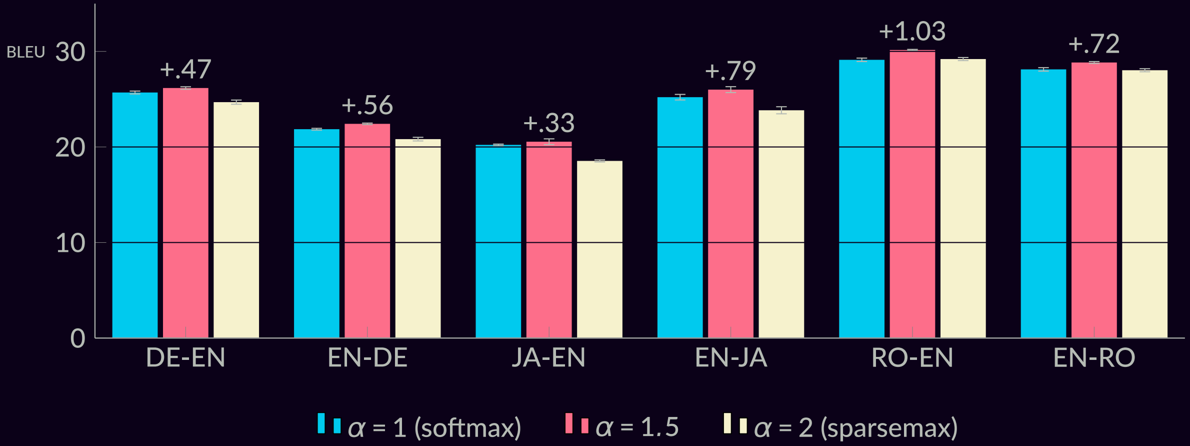
SIGMORPHON 2018 data, shared multi-lingual model.





(Peters, Niculae, and Martins, 2019)

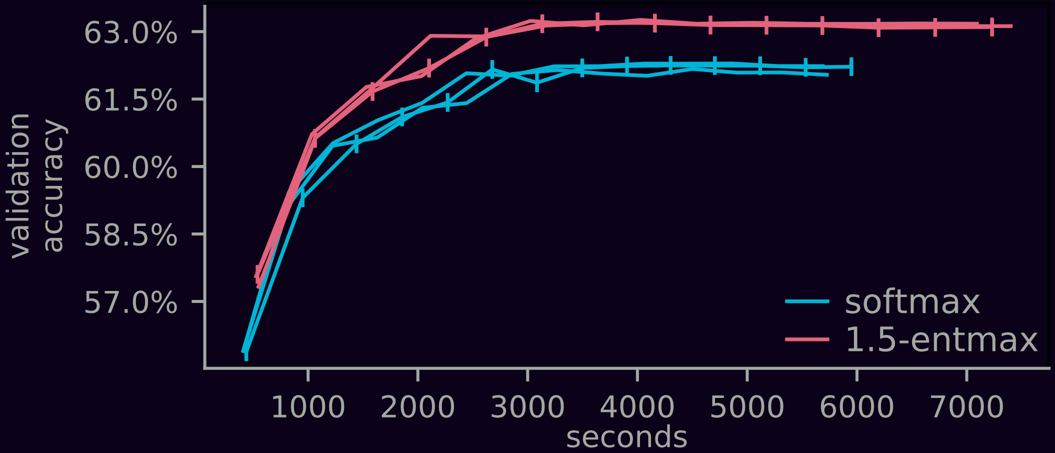
Neural Machine Translation



(Peters, Niculae, and Martins, 2019)

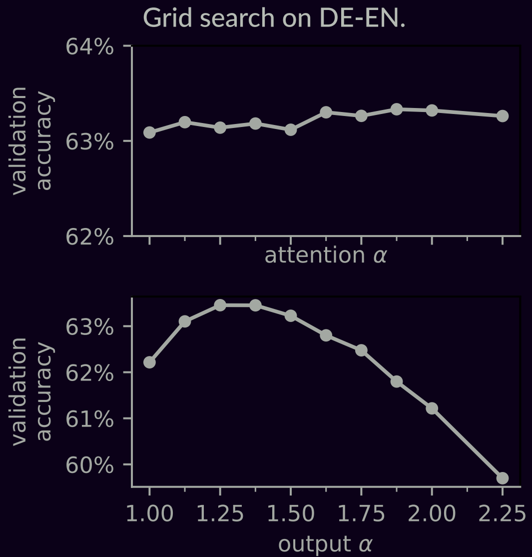
Sparse Mappings Don't Slow Down Training

Training timing on three DE-EN runs.
Ticks = passes over data.



(Peters, Niculae, and Martins, 2019)

Impact of Fine Tuning α



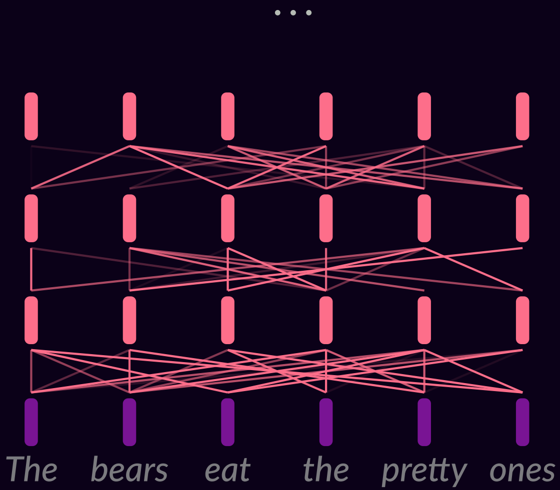
Outline

1. Warm-Up: Well-Known Losses and Mappings
2. Regularized Prediction Functions
3. Fenchel-Young Losses
4. Sparse Sequence-to-Sequence Models
5. Adaptively Sparse Transformers
6. Sparse Structured Prediction

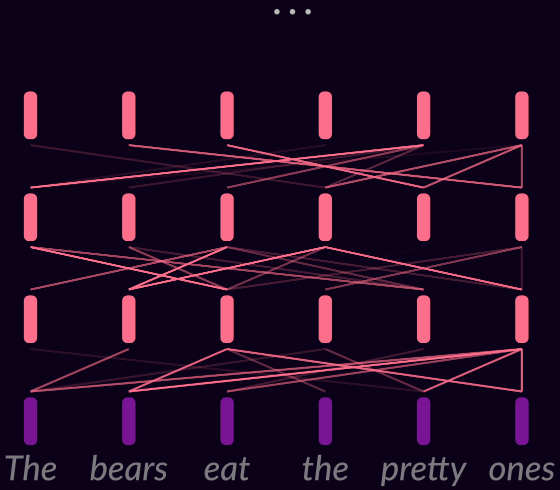
(Vaswani et al., 2017)

Transformers: Deep Self-Attention

Layered multi-head attention instead of LSTMs



Sparse Transformers



Adaptively Sparse Transformers

Transformers have $6 \times 4 \times 3$ attention heads: maybe *not all* should be sparse.

Adaptively Sparse Transformers

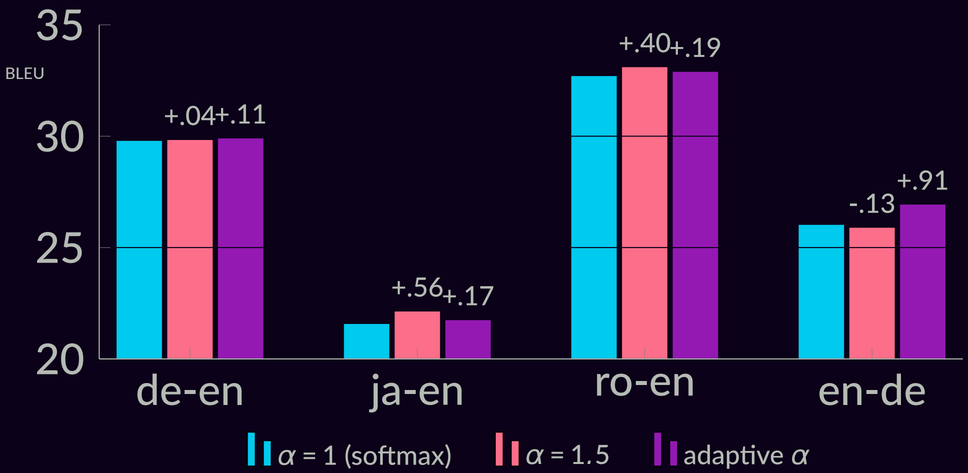
Transformers have $6 \times 4 \times 3$ attention heads: maybe *not all* should be sparse.

Let each attention head learn its α!

$$\frac{\partial \pi_{-\mathrm{H}_\alpha}}{\partial \alpha}$$

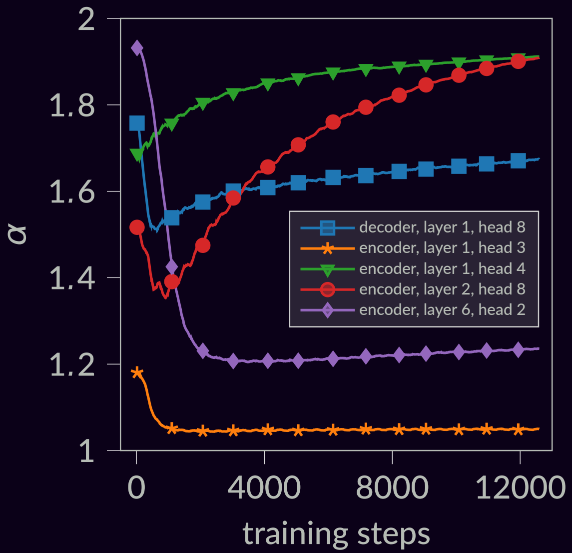
(Correia, Niculae, and Martins, 2019)

Neural Machine Translation



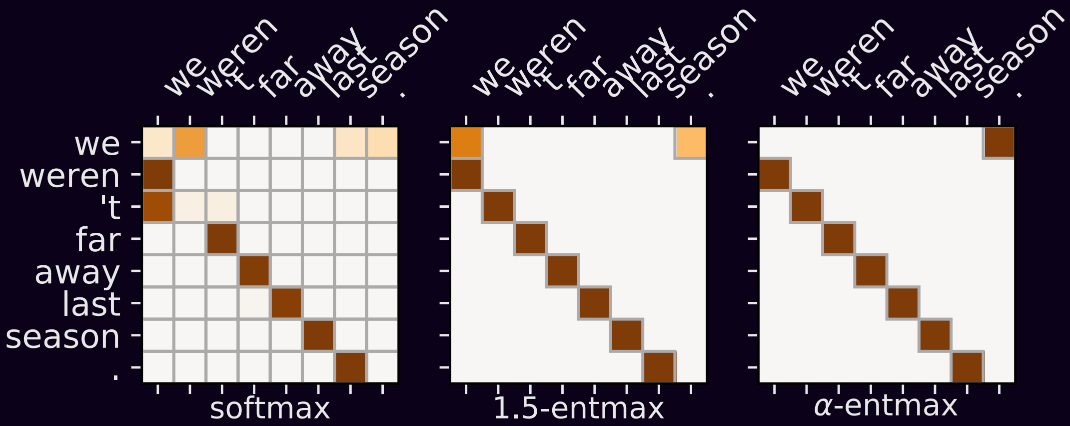
(Correia, Niculae, and Martins, 2019)

Trajectories of α During Training



(Correia, Niculae, and Martins, 2019)

Previous Position Head

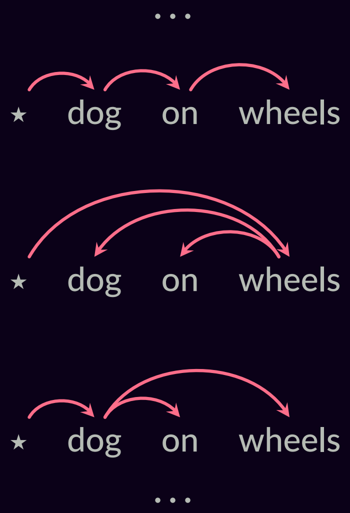


Learned $\alpha = 1.91$.

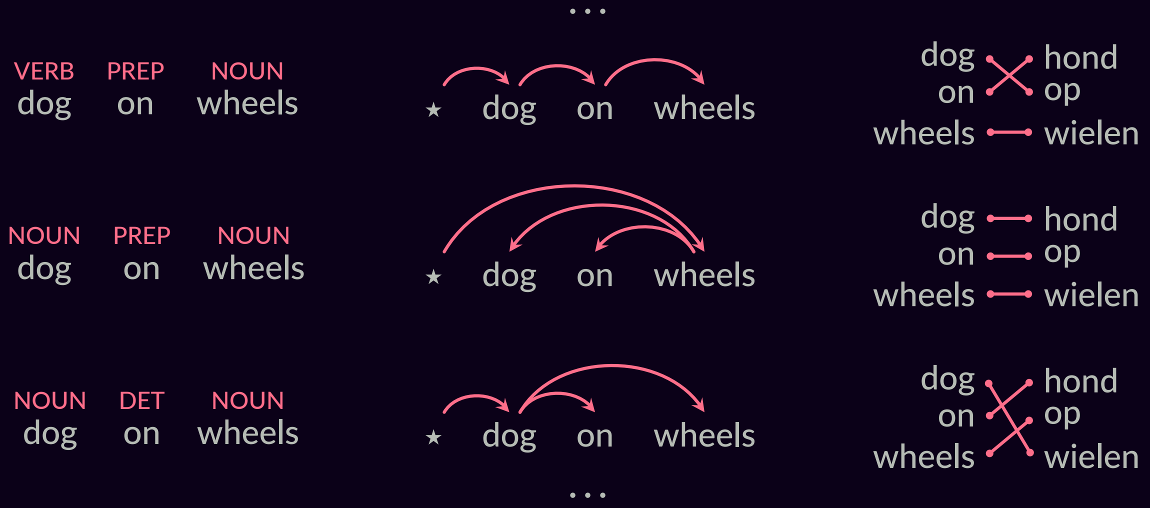
Outline

1. Warm-Up: Well-Known Losses and Mappings
2. Regularized Prediction Functions
3. Fenchel-Young Losses
4. Sparse Sequence-to-Sequence Models
5. Adaptively Sparse Transformers
6. Sparse Structured Prediction

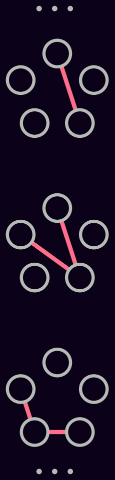
Structured Prediction



Structured Prediction



Structured Prediction



Factorization Into Parts

$$\boldsymbol{\theta} = A^\top \boldsymbol{\eta}$$

Factorization Into Parts

$$\boldsymbol{\theta} = A^\top \boldsymbol{\eta}$$

$$\boldsymbol{A} = \begin{array}{r} \star\to\text{dog} \\ \text{on}\to\text{dog} \\ \text{wheels}\to\text{dog} \\ \hline \star\to\text{on} \\ \text{dog}\to\text{on} \\ \text{wheels}\to\text{on} \\ \hline \star\to\text{wheels} \\ \text{dog}\to\text{wheels} \\ \text{on}\to\text{wheels} \end{array} \left[\begin{array}{ccccc} 1 & & 0 & 0 & \\ 0 & & 1 & 1 & \\ 0 & & 0 & 0 & \\ \hline 0 & & 1 & 1 & \\ 1 & \dots & 0 & 0 & \dots \\ 0 & & 0 & 0 & \\ \hline 0 & & 0 & 0 & \\ 0 & & 1 & 0 & \\ 1 & & 0 & 1 & \end{array}\right] \quad \boldsymbol{\eta} = \left[\begin{array}{r} .1 \\ .2 \\ -.1 \\ \hline .3 \\ .8 \\ .1 \\ \hline -.3 \\ .2 \\ -.1 \end{array}\right]$$

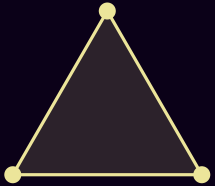
Factorization Into Parts

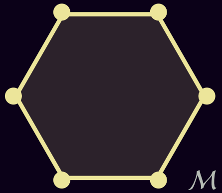
$$\boldsymbol{\theta} = A^\top \boldsymbol{\eta}$$

$$
\boldsymbol{A} = \begin{array}{r}
\star\to\text{dog} \\ \text{on}\to\text{dog} \\ \text{wheels}\to\text{dog} \\ \hline \star\to\text{on} \\ \text{dog}\to\text{on} \\ \text{wheels}\to\text{on} \\ \hline \star\to\text{wheels} \\ \text{dog}\to\text{wheels} \\ \text{on}\to\text{wheels}
\end{array}
\left[\begin{array}{ccccc}
1 & & 0 & 0 & \\
0 & & 1 & 1 & \\
0 & & 0 & 0 & \\
\hline
0 & & 1 & 1 & \\
1 & \dots & 0 & 0 & \dots \\
0 & & 0 & 0 & \\
\hline
0 & & 0 & 0 & \\
0 & & 1 & 0 & \\
1 & & 0 & 1 &
\end{array}\right]
\quad
\boldsymbol{\eta} = \left[\begin{array}{r}
.1 \\ .2 \\ -.1 \\ \hline .3 \\ .8 \\ .1 \\ \hline -.3 \\ .2 \\ -.1
\end{array}\right]
$$

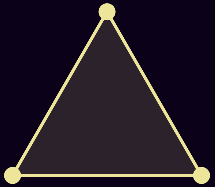
$$
\boldsymbol{A} = \begin{array}{r}
\text{dog—hond} \\ \text{dog—op} \\ \text{dog—wielen} \\ \hline \text{on—hond} \\ \text{on—op} \\ \text{on—wielen} \\ \hline \text{wheels—hond} \\ \text{wheels—op} \\ \text{wheels—wielen}
\end{array}
\left[\begin{array}{ccccc}
1 & & 0 & 0 & \\
0 & & 1 & 1 & \\
0 & & 0 & 0 & \\
\hline
0 & & 0 & 0 & \\
1 & \dots & 0 & 0 & \dots \\
0 & & 1 & 1 & \\
\hline
0 & & 1 & 0 & \\
0 & & 0 & 0 & \\
1 & & 0 & 1 &
\end{array}\right]
\quad
\boldsymbol{\eta} = \left[\begin{array}{r}
.1 \\ .2 \\ -.1 \\ \hline .3 \\ .8 \\ .1 \\ \hline -.3 \\ .2 \\ -.1
\end{array}\right]
$$

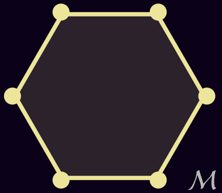
$$\mathcal{M} := \operatorname{conv}\left\{\boldsymbol{a}_h : h \in \mathcal{H}\right\}$$

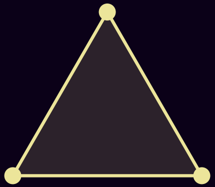


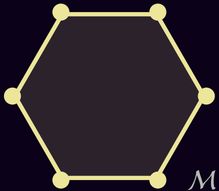


$$\mathcal{M} := \operatorname{conv}\left\{\boldsymbol{a}_h : h \in \mathcal{H}\right\}$$
$$= \left\{\boldsymbol{A}\boldsymbol{p} : \boldsymbol{p} \in \triangle\right\}$$





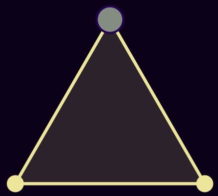


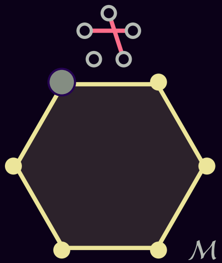


- **argmax** $\operatorname{argmax}_{\boldsymbol{p} \in \triangle} \langle \boldsymbol{p}, \boldsymbol{\theta} \rangle$

argmax $\operatorname{argmax}_{\boldsymbol{p} \in \triangle} \langle \boldsymbol{p}, \boldsymbol{\theta} \rangle$

MAP $\operatorname{argmax}_{\boldsymbol{\mu} \in \mathcal{M}} \langle \boldsymbol{\mu}, \boldsymbol{\eta} \rangle$

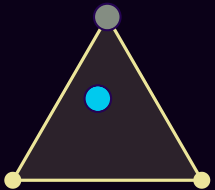


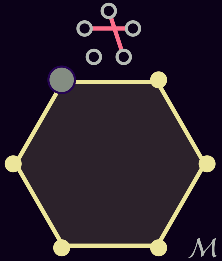


argmax $\underset{\boldsymbol{p}\in\triangle}{\operatorname{argmax}}\langle\boldsymbol{p},\boldsymbol{\theta}\rangle$

softmax $\underset{\boldsymbol{p}\in\triangle}{\operatorname{argmax}}\langle\boldsymbol{p},\boldsymbol{\theta}\rangle + \mathrm{H}(\boldsymbol{p})$

MAP $\underset{\boldsymbol{\mu}\in\mathcal{M}}{\operatorname{argmax}}\langle\boldsymbol{\mu},\boldsymbol{\eta}\rangle$



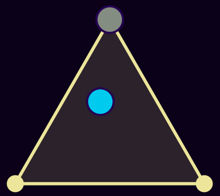


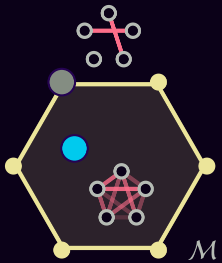
argmax $\arg\max_{\boldsymbol{p} \in \triangle} \langle \boldsymbol{p}, \boldsymbol{\theta} \rangle$

softmax $\arg\max_{\boldsymbol{p} \in \triangle} \langle \boldsymbol{p}, \boldsymbol{\theta} \rangle + \mathrm{H}(\boldsymbol{p})$

MAP $\arg\max_{\boldsymbol{\mu} \in \mathcal{M}} \langle \boldsymbol{\mu}, \boldsymbol{\eta} \rangle$

marginals $\arg\max_{\boldsymbol{\mu} \in \mathcal{M}} \langle \boldsymbol{\mu}, \boldsymbol{\eta} \rangle + \widetilde{\mathrm{H}}(\boldsymbol{\mu})$





Algorithms for specific structures

	Best structure (MAP)	Marginals
Sequence tagging	Viterbi (Rabiner, 1989)	Forward-Backward (Rabiner, 1989)
Constituent trees	CKY (Kasami, 1966; Younger, 1967) (Cocke and Schwartz, 1970)	Inside-Outside (Baker, 1979)
Temporal alignments	DTW (Sakoe and Chiba, 1978)	Soft-DTW (Cuturi and Blondel, 2017)
Dependency trees	Max. Spanning Arborescence (Chu and Liu, 1965; Edmonds, 1967)	Matrix-Tree (Kirchhoff, 1847)
Assignments	Kuhn-Munkres (Kuhn, 1955; Jonker and Volgenant, 1987)	#P-complete (Valiant, 1979; Taskar, 2004)

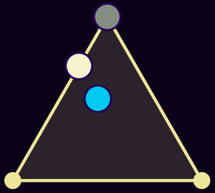
argmax $\operatorname{argmax}_{\boldsymbol{p} \in \triangle} \langle \boldsymbol{p}, \boldsymbol{\theta} \rangle$

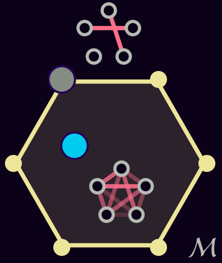
softmax $\operatorname{argmax}_{\boldsymbol{p} \in \triangle} \langle \boldsymbol{p}, \boldsymbol{\theta} \rangle + \mathrm{H}(\boldsymbol{p})$

sparsemax $\operatorname{argmax}_{\boldsymbol{p} \in \triangle} \langle \boldsymbol{p}, \boldsymbol{\theta} \rangle - 1/2 \|\boldsymbol{p}\|^2$

MAP $\operatorname{argmax}_{\boldsymbol{\mu} \in \mathcal{M}} \langle \boldsymbol{\mu}, \boldsymbol{\eta} \rangle$

marginals $\operatorname{argmax}_{\boldsymbol{\mu} \in \mathcal{M}} \langle \boldsymbol{\mu}, \boldsymbol{\eta} \rangle + \widetilde{\mathrm{H}}(\boldsymbol{\mu})$

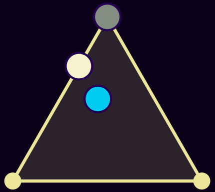


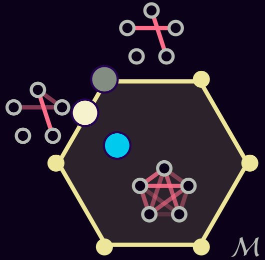


(Niculae, Martins, Blondel, and Cardie, 2018)

- **argmax** $\underset{p \in \triangle}{\operatorname{argmax}} \langle \boldsymbol{p}, \boldsymbol{\theta} \rangle$
- **softmax** $\underset{p \in \triangle}{\operatorname{argmax}} \langle \boldsymbol{p}, \boldsymbol{\theta} \rangle + \mathrm{H}(\boldsymbol{p})$
- **sparsemax** $\underset{p \in \triangle}{\operatorname{argmax}} \langle \boldsymbol{p}, \boldsymbol{\theta} \rangle - 1/2 \|\boldsymbol{p}\|^2$

- **MAP** $\underset{\mu \in \mathcal{M}}{\operatorname{argmax}} \langle \boldsymbol{\mu}, \boldsymbol{\eta} \rangle$
- **marginals** $\underset{\mu \in \mathcal{M}}{\operatorname{argmax}} \langle \boldsymbol{\mu}, \boldsymbol{\eta} \rangle + \widetilde{\mathrm{H}}(\boldsymbol{\mu})$
- **SparseMAP** $\underset{\mu \in \mathcal{M}}{\operatorname{argmax}} \langle \boldsymbol{\mu}, \boldsymbol{\eta} \rangle - 1/2 \|\boldsymbol{\mu}\|^2$





(Niculae, Martins, Blondel, and Cardie, 2018)

Generic Algorithm for SparseMAP

$$\boldsymbol{\mu}^\star = \operatorname*{argmax}_{\boldsymbol{\mu} \in \mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta} - 1/2 \|\boldsymbol{\mu}\|^2$$

(Niculae, Martins, Blondel, and Cardie, 2018)

Generic Algorithm for SparseMAP

$$\boldsymbol{\mu}^\star = \operatorname*{argmax}_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta} - 1/2\|\boldsymbol{\mu}\|^2$$

linear constraints
(alas, exponentially many!)

quadratic objective

(Niculae, Martins, Blondel, and Cardie, 2018)

Generic Algorithm for SparseMAP

linear constraints
(alas, exponentially many!)

$$\boldsymbol{\mu}^\star = \underset{\boldsymbol{\mu} \in \mathcal{M}}{\operatorname{argmax}} \boldsymbol{\mu}^\top \boldsymbol{\eta} - 1/2 \|\boldsymbol{\mu}\|^2$$

quadratic objective

Conditional Gradient

(Frank and Wolfe, 1956; Lacoste-Julien and Jaggi, 2015)

(Niculae, Martins, Blondel, and Cardie, 2018)

Generic Algorithm for SparseMAP

$$\boldsymbol{\mu}^\star = \underset{\boldsymbol{\mu} \in \mathcal{M}}{\operatorname{argmax}} \, \boldsymbol{\mu}^\top \boldsymbol{\eta} - 1/2 \|\boldsymbol{\mu}\|^2$$

linear constraints
(alas, exponentially many!)

quadratic objective

Conditional Gradient

(Frank and Wolfe, 1956; Lacoste-Julien and Jaggi, 2015)

- select a new corner of $\mathcal{M}$

Generic Algorithm for SparseMAP

linear constraints
(alas, exponentially many!)

$$\boldsymbol{\mu}^\star = \underset{\boldsymbol{\mu} \in \mathcal{M}}{\operatorname{argmax}}\, \boldsymbol{\mu}^\top \boldsymbol{\eta} - 1/2 \|\boldsymbol{\mu}\|^2$$

quadratic objective

Conditional Gradient

(Frank and Wolfe, 1956; Lacoste-Julien and Jaggi, 2015)

- select a new corner of $\mathcal{M}$

$$\boldsymbol{a}_{y^\star} = \underset{\boldsymbol{\mu} \in \mathcal{M}}{\operatorname{argmax}}\, \boldsymbol{\mu}^\top \underbrace{\left(\boldsymbol{\eta} - \boldsymbol{\mu}^{(t-1)}\right)}_{\tilde{\boldsymbol{\eta}}}$$

(Niculae, Martins, Blondel, and Cardie, 2018)

Generic Algorithm for SparseMAP

linear constraints
(alas, exponentially many!)

$$\boldsymbol{\mu}^\star = \operatorname*{argmax}_{\boldsymbol{\mu} \in \mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta} - 1/2 \|\boldsymbol{\mu}\|^2$$

quadratic objective

Conditional Gradient

(Frank and Wolfe, 1956; Lacoste-Julien and Jaggi, 2015)

- select a new corner of $\mathcal{M}$
- update the (sparse) coefficients of $\boldsymbol{p}$
  - Update rules: vanilla, away-step, pairwise

Generic Algorithm for SparseMAP

linear constraints
(alas, exponentially many!)

$$\boldsymbol{\mu}^\star = \operatorname*{argmax}_{\boldsymbol{\mu} \in \mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta} - 1/2 \|\boldsymbol{\mu}\|^2$$

quadratic objective

Conditional Gradient

(Frank and Wolfe, 1956; Lacoste-Julien and Jaggi, 2015)

- select a new corner of $\mathcal{M}$
- update the (sparse) coefficients of $\boldsymbol{p}$
    - Update rules: vanilla, away-step, pairwise
    - Quadratic objective: **Active Set**

      (Nocedal and Wright, 1999, Ch. 16.4 & 16.5)

      (Wolfe, 1976; Vinyes and Obozinski, 2017)

(Niculae, Martins, Blondel, and Cardie, 2018)

Generic Algorithm for SparseMAP

linear constraints
(alas, exponentially many!)

$$\boldsymbol{\mu}^\star = \operatorname*{argmax}_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta} - 1/2\|\boldsymbol{\mu}\|^2$$

quadratic objective

Conditional Gradient

(Frank and Wolfe, 1956; Lacoste-Julien and Jaggi, 2015)

- select a new corne
- update the (sparse)
  - Update rules: van
  - Quadratic objective: **Active Set**
    (Nocedal and Wright, 1999, Ch. 16.4 & 16.5)
    (Wolfe, 1976; Vinyes and Obozinski, 2017)

Active Set achieves
finite & **linear** convergence!

(Niculae, Martins, Blondel, and Cardie, 2018)

Generic Algorithm for SparseMAP

$$\mu^\star = \underset{\mu \in \mathcal{M}}{\operatorname{argmax}}\, \mu^\top \eta - 1/2 \|\mu\|^2$$

linear constraints
(alas, exponentially many!)

quadratic objective

Conditional Gradient

(Frank and Wolfe, 1956; Lacoste-Julien and Jaggi, 2015)

- select a new corner of $\mathcal{M}$
- update the (sparse) coefficients of $\boldsymbol{p}$
  - Update rules: vanilla, away-step, pairwise
  - Quadratic objective: **Active Set**

    (Nocedal and Wright, 1999, Ch. 16.4 & 16.5)

    (Wolfe, 1976; Vinyes and Obozinski, 2017)

Backward pass

$\frac{\partial \mu}{\partial \eta}$ is sparse;
precomputed in forward pass!

Generic Algorithm for SparseMAP

linear constraints
(alas, exponentially many!)

$$\boldsymbol{\mu}^\star = \operatorname*{argmax}_{\boldsymbol{\mu} \in \mathcal{M}} \boldsymbol{\mu}^\top \boldsymbol{\eta} - 1/2 \|\boldsymbol{\mu}\|^2$$

quadratic objective

Conditional Gradient

(Frank and Wolfe, 1956; Lacoste-Julien and Jaggi, 2015)

- select a new corner of $\mathcal{M}$
- update the (sparse) coefficients of $\boldsymbol{p}$
  - Update rules: vanilla, away-step, pairwise
  - Quadratic objective: **Active Set**

    (Nocedal and Wright, 1999, Ch. 16.4 & 16.5)

    (Wolfe, 1976; Vinyes and Obozinski, 2017)

Backward pass

$\frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{\eta}}$ is sparse;
precomputed in forward pass!

(Niculae, Martins, Blondel, and Cardie, 2018)

Generic Algorithm for SparseMAP

linear constraints
(alas, exponentially many!)

$$\mu^\star = \operatorname*{argmax}_{\mu \in \mathcal{M}} \mu^\top \eta - 1/2\|\mu\|^2$$

quadratic objective

Conditi **pass**

(Frank and Wolfe, 1956

Completely modular: just add MAP

- select a new c
- update the (sparse) coefficients of $\boldsymbol{p}$
  - Update rules: vanilla, away-step, pairwise
  - Quadratic objective: **Active Set**

    (Nocedal and Wright, 1999, Ch. 16.4 & 16.5)

    (Wolfe, 1976; Vinyes and Obozinski, 2017)

$\frac{\partial \mu}{\partial \eta}$ is sparse;
precomputed in forward pass!

(Niculae, Martins, Blondel, and Cardie, 2018)

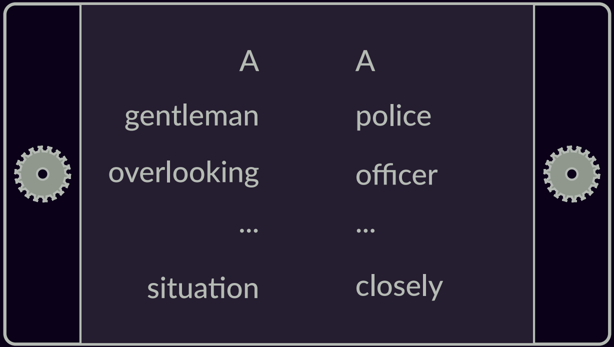
Sparse Structured Attention for Alignments

NLI

premise: A gentleman overlooking a neighborhood situation.
hypothesis: A police officer watches a situation closely.

input

(P, H)



output

- entails
- contradicts
- neutral

(Model: ESIM (Chen et al., 2017))

(Niculae, Martins, Blondel, and Cardie, 2018)

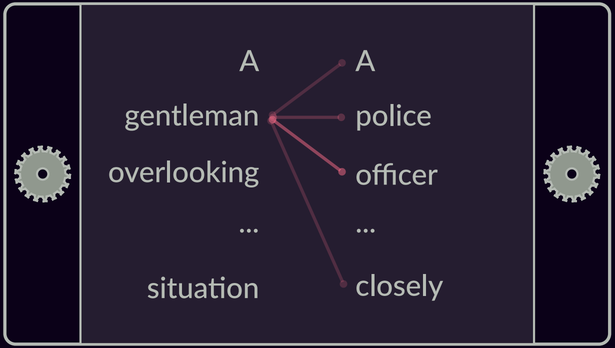
Sparse Structured Attention for Alignments

NLI

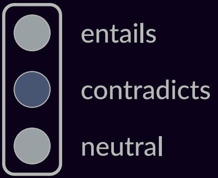
premise: A gentleman overlooking a neighborhood situation.
hypothesis: A police officer watches a situation closely.

input

(P, H)



output



(Model: ESIM (Chen et al., 2017))

(Niculae, Martins, Blondel, and Cardie, 2018)

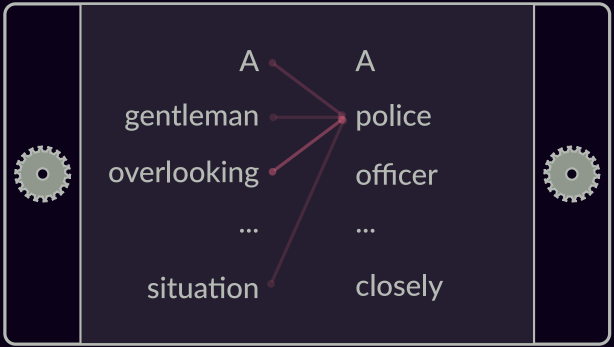
Sparse Structured Attention for Alignments

NLI

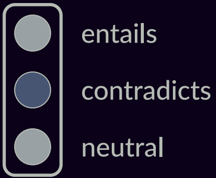
premise: A gentleman overlooking a neighborhood situation.
hypothesis: A police officer watches a situation closely.

input

(P, H)



output



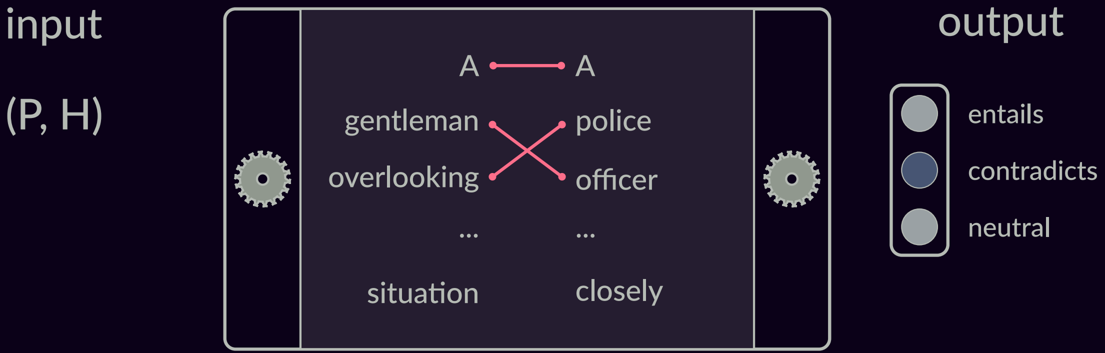
(Model: ESIM (Chen et al., 2017))

(Niculae, Martins, Blondel, and Cardie, 2018)

Sparse Structured Attention for Alignments

NLI

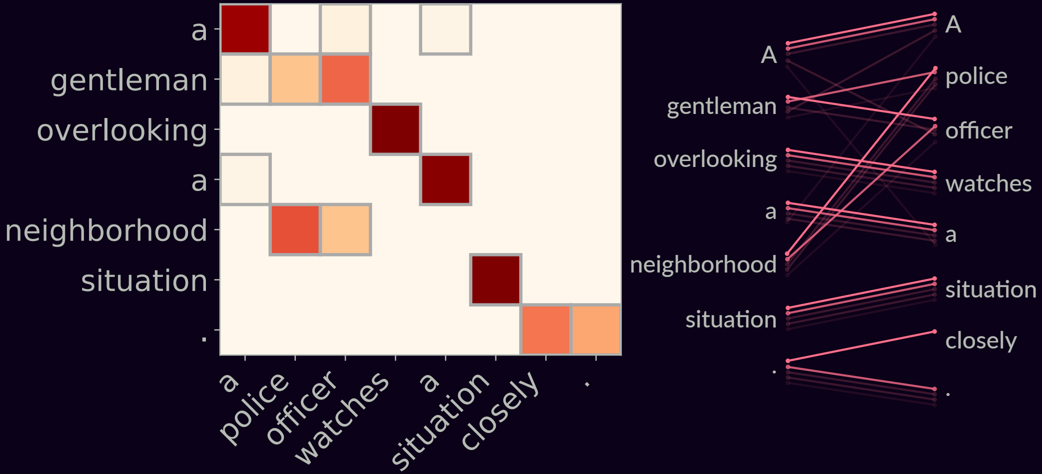
premise: A gentleman overlooking a neighborhood situation.
hypothesis: A police officer watches a situation closely.



(Proposed model: global matching)

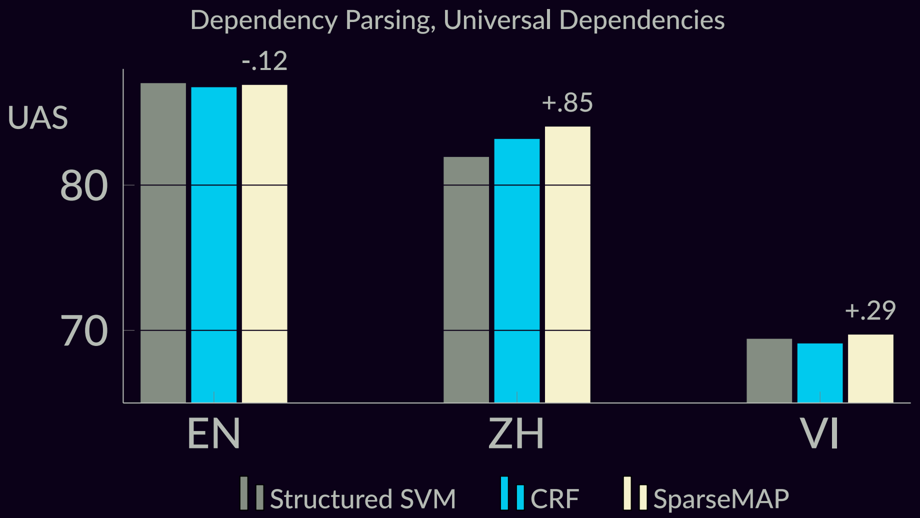
(Niculae, Martins, Blondel, and Cardie, 2018)

Sparse Structured Attention for Alignments



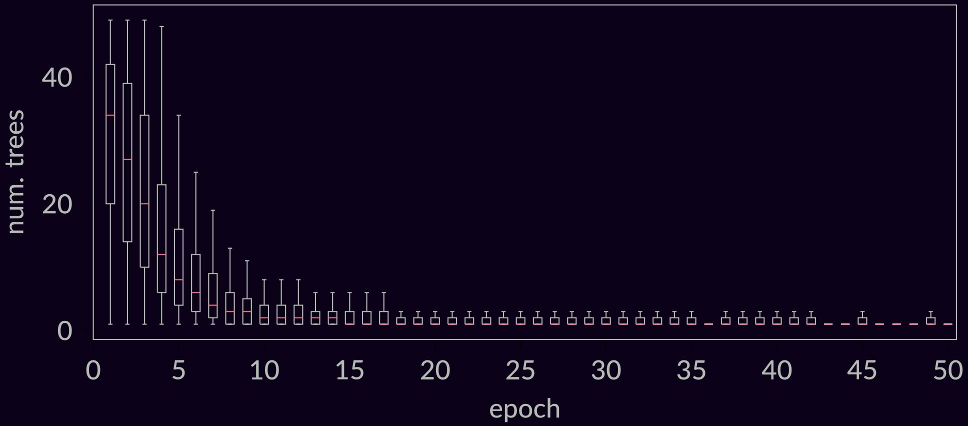
(Niculae, Martins, Blondel, and Cardie, 2018)

Sparse Structured Output Prediction



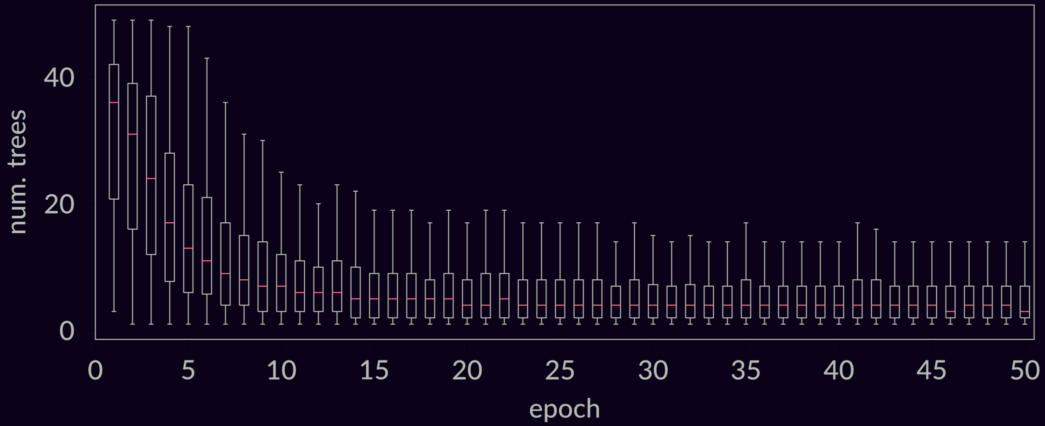
Sparse Structured Output Prediction

Training



Sparse Structured Output Prediction

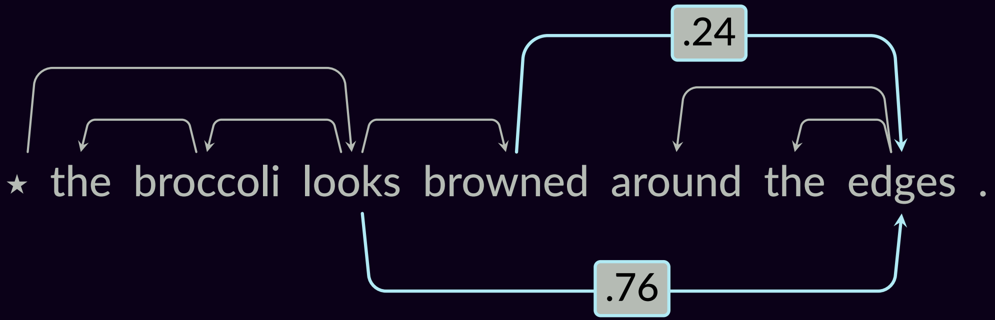
Validation: 25% unambiguous, 66% ≤ 5



(Niculae, Martins, Blondel, and Cardie, 2018)

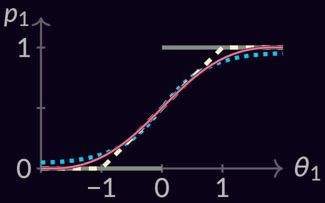
Sparse Structured Output Prediction

Inference captures linguistic ambiguity!



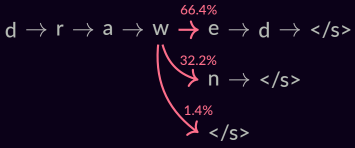
Summary: Fenchel-Young losses and mappings, a framework for:

insight into sparsity & margins



sparse attention weights

sparse output space



Next steps: sparsity in stochastic and generative models.

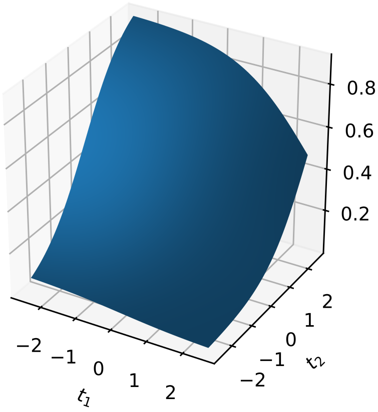
vlad@vene.ro github.com/deep-spin/entmax · github.com/vene/sparsemap https://vene.ro

Extra slides

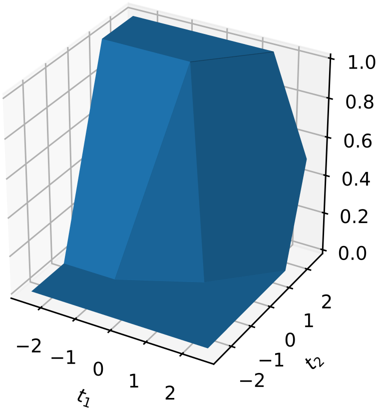
Acknowledgements

This work was supported by the European Research Council (ERC StG DeepSPIN 758969) and by the Fundação para a Ciência e Tecnologia through contract UID/EEA/50008/2013.

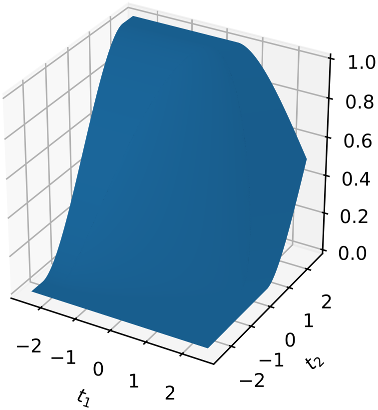
Some icons by Dave Gandy and Freepik via flaticon.com.



softmax



sparsemax



1.5-entmax

Expressions for Margins

- Main result: $L_{-\mathrm{H}}(\boldsymbol{\theta}, \boldsymbol{e}_k)$ has margin m iff. $m\boldsymbol{e}_k \in \partial(-\mathrm{H})(\boldsymbol{e}_k)$.
- If H twice-differentiable, $m_{\mathrm{H}} = \nabla_j \mathrm{H}(\boldsymbol{e}_k) - \nabla_k \mathrm{H}(\boldsymbol{e}_k)$.
- If $\mathrm{H} = \sum_j h(p_j)$ separable, $m_{\mathrm{H}} = h'(0) - h'(1)$.

Relation With Bregman Divergences

- Bregman divergences are defined in primal space: $B_\Omega : \operatorname{dom}\Omega \times \operatorname{dom}\Omega \to \mathbb{R}_+$

$$B_\Omega(\boldsymbol{y}||\boldsymbol{p}) := \Omega(\boldsymbol{y}) - \Omega(\boldsymbol{p}) = \langle \nabla\Omega(\boldsymbol{p}), \boldsymbol{y} - \boldsymbol{p}\rangle$$

- FY losses are in **mixed** space: $L_\Omega : \operatorname{dom}(\Omega^\star) \times \operatorname{dom}(\Omega) \to \mathbb{R}_+$
- Denoting $\boldsymbol{\theta} = \nabla\Omega(\boldsymbol{p})$ gives $B_\Omega(\boldsymbol{y}||\boldsymbol{p}) = L_\Omega(\boldsymbol{\theta}; \boldsymbol{y})$.
- However, starting from $\boldsymbol{\theta}$, $B_\Omega(\boldsymbol{y}|| \boldsymbol{\pi}_\Omega(\boldsymbol{\theta}))$ not always convex.
  ("link function" approach).

(Danskin, 1966; Prop. B.25 in Bertsekas, 1999)

Danskin's Theorem

Let $\phi : \mathbb{R}^k \times \mathcal{Z} \to \mathbb{R}$, $\mathcal{Z} \subset \mathbb{R}^k$ compact.

$$\partial \max_{\boldsymbol{z} \in \mathcal{Z}} \phi(\boldsymbol{x}, \boldsymbol{z}) = \operatorname{conv}\{\nabla_{\boldsymbol{x}} \phi(\boldsymbol{x}, \boldsymbol{z}^\star) \mid \boldsymbol{z}^\star \in \operatorname*{argmax}_{\boldsymbol{z} \in \mathcal{Z}} \phi(\boldsymbol{x}, \boldsymbol{z})\}.$$

Example: maximum of a vector

(Danskin, 1966; Prop. B.25 in Bertsekas, 1999)

Danskin's Theorem

Let $\phi : \mathbb{R}^k \times \mathcal{Z} \to \mathbb{R}$, $\mathcal{Z} \subset \mathbb{R}^k$ compact.

$$\partial \max_{\boldsymbol{z} \in \mathcal{Z}} \phi(\boldsymbol{x}, \boldsymbol{z}) = \operatorname{conv}\{\nabla_{\boldsymbol{x}} \phi(\boldsymbol{x}, \boldsymbol{z}^\star) \mid \boldsymbol{z}^\star \in \operatorname*{argmax}_{\boldsymbol{z} \in \mathcal{Z}} \phi(\boldsymbol{x}, \boldsymbol{z})\}.$$

Example: maximum of a vector

$$\begin{aligned}
\partial \max_{j \in [d]} \theta_j &= \partial \max_{\boldsymbol{p} \in \triangle} \boldsymbol{p}^\top \boldsymbol{\theta} \\
&= \partial \max_{\boldsymbol{p} \in \triangle} \phi(\boldsymbol{p}, \boldsymbol{\theta}) \\
&= \operatorname{conv}\{\nabla_{\boldsymbol{\theta}} \phi(\boldsymbol{p}^\star, \boldsymbol{\theta})\} \\
&= \operatorname{conv}\{\boldsymbol{p}^\star\}
\end{aligned}$$

(Danskin, 1966; Prop. B.25 in Bertsekas, 1999)

Danskin's Theorem

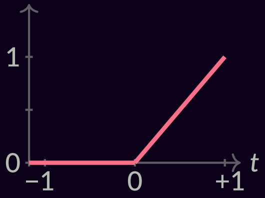
Let $\phi : \mathbb{R}^k \times \mathcal{Z} \to \mathbb{R}$, $\mathcal{Z} \subset \mathbb{R}^k$ compact.

$$\partial \max_{\boldsymbol{z} \in \mathcal{Z}} \phi(\boldsymbol{x}, \boldsymbol{z}) = \operatorname{conv}\left\{ \nabla_{\boldsymbol{x}} \phi(\boldsymbol{x}, \boldsymbol{z}^\star) \mid \boldsymbol{z}^\star \in \operatorname*{argmax}_{\boldsymbol{z} \in \mathcal{Z}} \phi(\boldsymbol{x}, \boldsymbol{z}) \right\}.$$

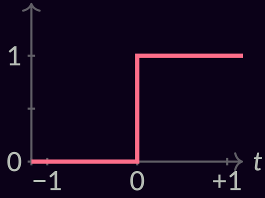
Example: maximum of a vector

$$\begin{aligned} \partial \max_{j \in [d]} \theta_j &= \partial \max_{\boldsymbol{p} \in \triangle} \boldsymbol{p}^\top \boldsymbol{\theta} \\ &= \partial \max_{\boldsymbol{p} \in \triangle} \phi(\boldsymbol{p}, \boldsymbol{\theta}) \\ &= \operatorname{conv}\left\{ \nabla_{\boldsymbol{\theta}} \phi(\boldsymbol{p}^\star, \boldsymbol{\theta}) \right\} \\ &= \operatorname{conv}\left\{ \boldsymbol{p}^\star \right\} \end{aligned}$$

$\boldsymbol{\theta} = [t, 0]$

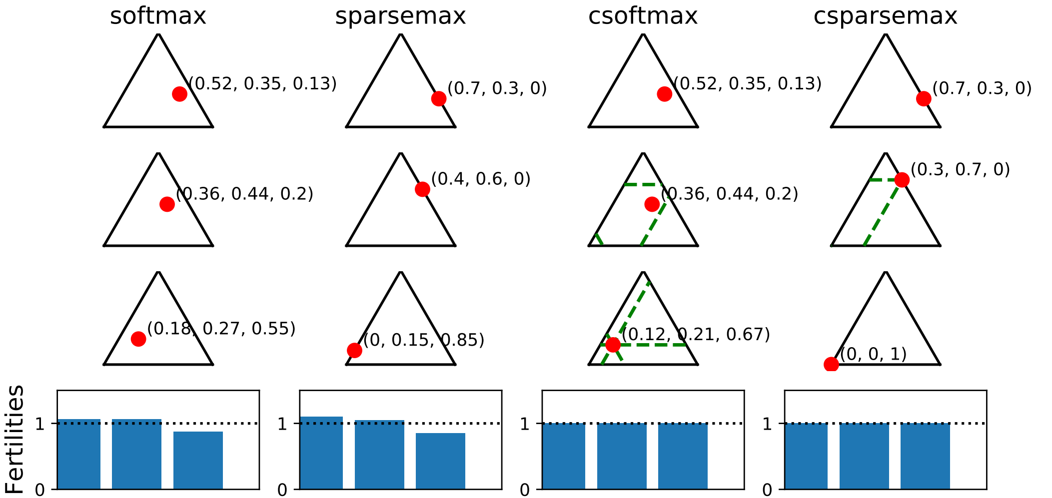


$\max_j \theta_j$

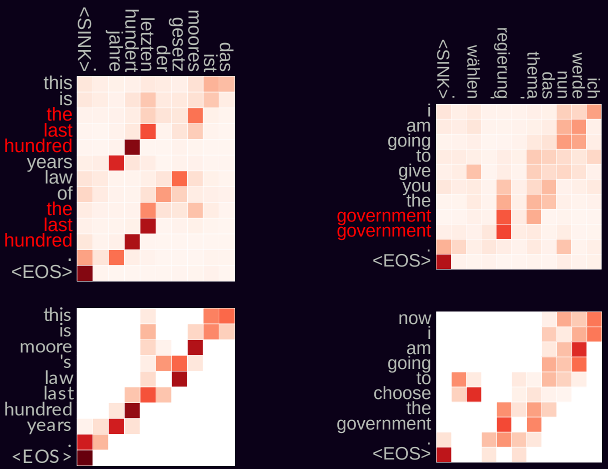


$\{ g_1 \mid \boldsymbol{g} \in \partial \max_j \theta_j \}$

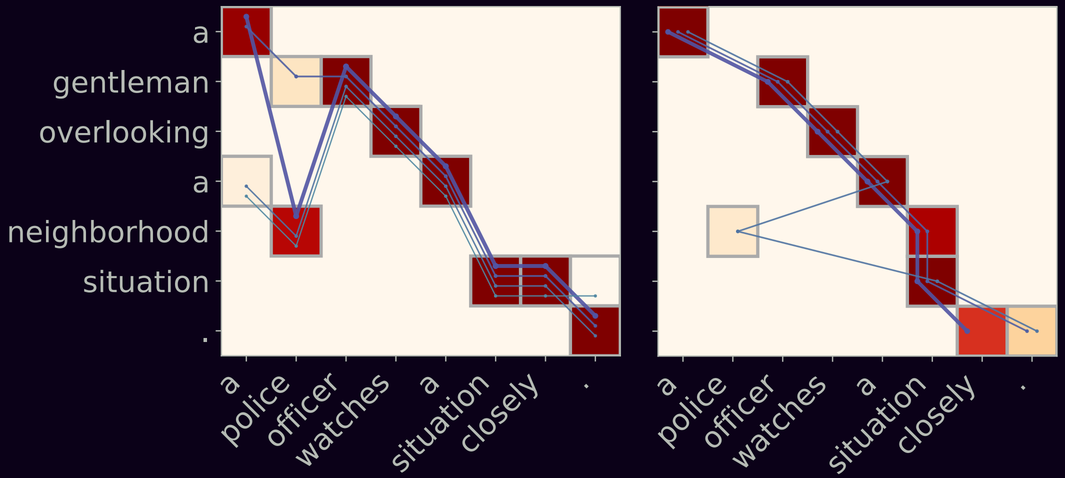
Example: Source Sentence with Three Words



e.g., fertility constraints for NMT



constrained softmax: (Martins and Kreutzer, 2017) **constrained sparsemax:** (Malaviya et al., 2018)


a
gentleman
overlooking
a
neighborhood
situation
.
a
police
officer
watches
a
situation
closely
.
a
police
officer
watches
a
situation
closely
.

References I

- Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2015). "Neural machine translation by jointly learning to align and translate". In: *Proc. of ICLR*.
- Baker, James K (1979). "Trainable grammars for speech recognition". In: *The Journal of the Acoustical Society of America* 65.S1, S132–S132.
- Bertsekas, Dimitri P (1999). *Nonlinear Programming*. Athena Scientific Belmont.
- Blondel, Mathieu, André FT Martins, and Vlad Niculae (2019). "Learning classifiers with Fenchel-Young losses: Generalized entropies, margins, and algorithms". In: *Proc. AISTATS*.
- Boyd, Stephen and Lieven Vandenberghe (2004). *Convex Optimization*. Cambridge University Press.
- Chen, Qian, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen (2017). "Enhanced LSTM for natural language inference". In: *Proc. of ACL*.
- Chu, Yoeng-Jin and Tseng-Hong Liu (1965). "On the Shortest Arborescence of a Directed Graph". In: *Science Sinica* 14, pp. 1396–1400.
- Cocke, William John and Jacob T Schwartz (1970). *Programming languages and their compilers*. Courant Institute of Mathematical Sciences.
- Correia, Gonçalo M., Vlad Niculae, and André FT Martins (2019). "Adaptively Sparse Transformers". In: *Proc. EMNLP*.
- Cuturi, Marco and Mathieu Blondel (2017). "Soft-DTW: a differentiable loss function for time-series". In: *Proc. of ICML*.

References II

Danskin, John M (1966). "The theory of max-min, with applications". In: *SIAM Journal on Applied Mathematics* 14.4, pp. 641–664.

Dantzig, George B, Alex Orden, and Philip Wolfe (1955). "The generalized simplex method for minimizing a linear form under linear inequality restraints". In: *Pacific Journal of Mathematics* 5.2, pp. 183–195.

DeGroot, Morris H (1962). "Uncertainty, information, and sequential experiments". In: *The Annals of Mathematical Statistics*, pp. 404–419.

Edmonds, Jack (1967). "Optimum branchings". In: *J. Res. Nat. Bur. Stand.* 71B, pp. 233–240.

Frank, Marguerite and Philip Wolfe (1956). "An algorithm for quadratic programming". In: *Nav. Res. Log.* 3.1-2, pp. 95–110.

Grünwald, Peter D and A Philip Dawid (2004). "Game theory, maximum entropy, minimum discrepancy and robust Bayesian decision theory". In: *Annals of Statistics*, pp. 1367–1433.

Held, Michael, Philip Wolfe, and Harlan P Crowder (1974). "Validation of subgradient optimization". In: *Mathematical Programming* 6.1, pp. 62–88.

Jonker, Roy and Anton Volgenant (1987). "A shortest augmenting path algorithm for dense and sparse linear assignment problems". In: *Computing* 38.4, pp. 325–340.

Kakade, Sham, Shai Shalev-Shwartz, and Ambuj Tewari (2009). "On the duality of strong convexity and strong smoothness: Learning applications and matrix regularization". In: *Tech Report*.

References III

Kasami, Tadao (1966). “An efficient recognition and syntax-analysis algorithm for context-free languages”. In: *Coordinated Science Laboratory Report no. R-257*.

Kirchhoff, Gustav (1847). “Ueber die Auflösung der Gleichungen, auf welche man bei der Untersuchung der linearen Vertheilung galvanischer Ströme geführt wird”. In: *Annalen der Physik* 148.12, pp. 497–508.

Kuhn, Harold W (1955). “The Hungarian method for the assignment problem”. In: *Nav. Res. Log.* 2.1-2, pp. 83–97.

Lacoste-Julien, Simon and Martin Jaggi (2015). “On the global linear convergence of Frank-Wolfe optimization variants”. In: *Proc. of NeurIPS*.

Malaviya, Chaitanya, Pedro Ferreira, and André FT Martins (2018). “Sparse and constrained attention for neural machine translation”. In: *Proc. of ACL*.

Martins, André FT and Ramón Fernandez Astudillo (2016). “From softmax to sparsemax: A sparse model of attention and multi-label classification”. In: *Proc. of ICML*.

Martins, André FT and Julia Kreutzer (2017). “Learning What’s Easy: Fully Differentiable Neural Easy-First Taggers”. In: *Proc. of EMNLP*, pp. 349–362.

Nesterov, Yurii (2005). “Smooth minimization of non-smooth functions”. In: *Mathematical Programming* 103.1, pp. 127–152.

Niculae, Vlad and Mathieu Blondel (2017). “A regularized framework for sparse and structured neural attention”. In: *Proc. of NeurIPS*.

References IV

Niculae, Vlad, André FT Martins, Mathieu Blondel, and Claire Cardie (2018). "SparseMAP: Differentiable sparse structured inference". In: *Proc. of ICML*.

Nocedal, Jorge and Stephen Wright (1999). *Numerical Optimization*. Springer New York.

Peters, Ben, Vlad Niculae, and André FT Martins (2019). "Sparse sequence-to-sequence models". In: *Proc. ACL*.

Rabiner, Lawrence R. (1989). "A tutorial on Hidden Markov Models and selected applications in speech recognition". In: *P. IEEE* 77.2, pp. 257–286.

Sakoe, Hiroaki and Seibi Chiba (1978). "Dynamic programming algorithm optimization for spoken word recognition". In: *IEEE Trans. on Acoustics, Speech, and Sig. Proc.* 26, pp. 43–49.

Taskar, Ben (2004). "Learning structured prediction models: A large margin approach". PhD thesis. Stanford University.

Tsallis, Constantino (1988). "Possible generalization of Boltzmann-Gibbs statistics". In: *Journal of Statistical Physics* 52, pp. 479–487.

Valiant, Leslie G (1979). "The complexity of computing the permanent". In: *Theor. Comput. Sci.* 8.2, pp. 189–201.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin (2017). "Attention Is All You Need". In: *Proc. of NeurIPS*.

Vinyes, Marina and Guillaume Obozinski (2017). " Fast column generation for atomic norm regularization". In: *Proc. of AISTATS*.

References V

Wolfe, Philip (1976). "Finding the nearest point in a polytope". In: *Mathematical Programming* 11.1, pp. 128–149.

Younger, Daniel H (1967). "Recognition and parsing of context-free languages in time n^3". In: *Information and Control* 10.2, pp. 189–208.