Sparsely Activated Layers for Text Classifiers

Wilker Aziz
Institute for Logic, Language, and Computation

SEA @ IvI

Text classifiers

Let's consider a general text classifier these days

Text classifiers

Let's consider a general text classifier these days

$$Y|x \sim \mathrm{Cat}(f(x;\theta))$$

- x is some (high-dimensional) input text
  e.g. a sentence, short paragraph, pair of texts
- y is a K-valued label
  e.g. sentiment, logical entailment
- $f(\cdot;\theta)$ maps from text to a K-dimensional probability vector
  e.g. a NN encoder and a softmax output layer

We call this an *observation model*

Parameter estimation

Given N i.i.d. observations, a step in the direction

$$\boldsymbol{\nabla}_\theta \log \mathrm{Cat}(y^{(s)}|f(x^{(s)};\theta))$$

takes us closer to a local optimum of the log-likelihood function.

Parameter estimation

Given N i.i.d. observations, a step in the direction

$$\nabla_\theta \log \mathrm{Cat}(y^{(s)}|f(x^{(s)};\theta))$$

takes us closer to a local optimum of the log-likelihood function.

As long as we keep everything about f fully differentiable
it can be as fancy as we like!

Fancy f

In sentiment classification
f is usually a bidirectional recurrent encoder

In natural language inference (aka textual entailment)
f compares two sentences using attention mechanisms

Fancy f

In sentiment classification
f is usually a bidirectional recurrent encoder

In natural language inference (aka textual entailment)
f compares two sentences using attention mechanisms

But, make f too fancy and

1. it may overfit
2. it may not scale
3. we can never tell what the classifier is doing

Fancy f

In sentiment classification
f is usually a bidirectional recurrent encoder

In natural language inference (aka textual entailment)
f compares two sentences using attention mechanisms

But, make f too fancy and

1. it may overfit
2. it may not scale
3. we can never tell what the classifier is doing

In this talk I will focus on (3)
collaboration with Joost Bastings and Ivan Titov

Outline

Outline

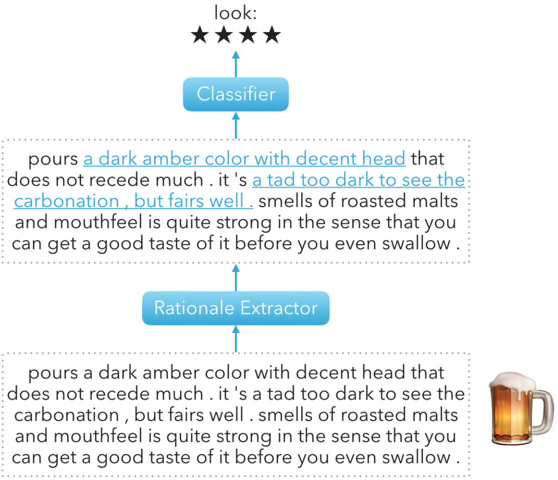
A step towards transparency

We give a NN lots of data to crunch and it makes decisions for us

- why certain decisions take place?
- based on what evidence?
- can we take a peek at what correlations a NN is likely exploiting?

Rationale

What if we classified based on a compact view of the input?



Lei et al. (2016) called this view a *rationale*

Inducing latent rationales

I will denote this view by $x \odot z$

- think of $z = \langle z_1, \ldots, z_n \rangle$ as an elementwise mask
  it selects what parts of the input $x = \langle x_1, \ldots, x_n \rangle$ are available for classification

Inducing latent rationales

I will denote this view by $x \odot z$

- think of $z = \langle z_1, \ldots, z_n \rangle$ as an elementwise mask
  it selects what parts of the input $x = \langle x_1, \ldots, x_n \rangle$ are available for classification

We want to *learn* what to select, thus we introduce a *latent model*

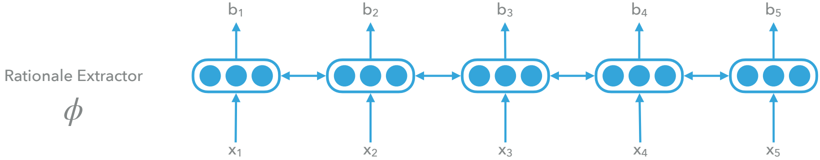
$$Z_i | x \sim \mathrm{Bern}(g_i(x; \phi))$$
$$Y | x, z \sim \mathrm{Cat}(f(x \odot z; \theta))$$

and have a NN $g(x; \phi)$ parameterise n Bernoulli selectors

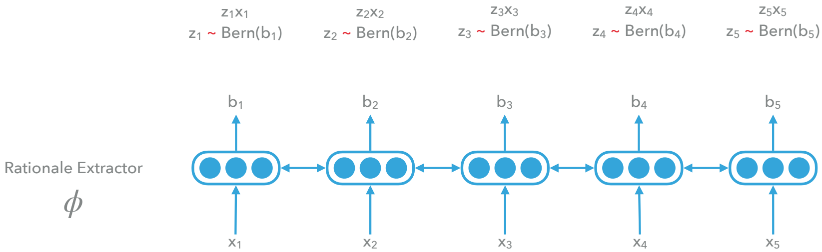
Latent rationales with Bernoulli selectors

x_1 x_2 x_3 x_4 x_5

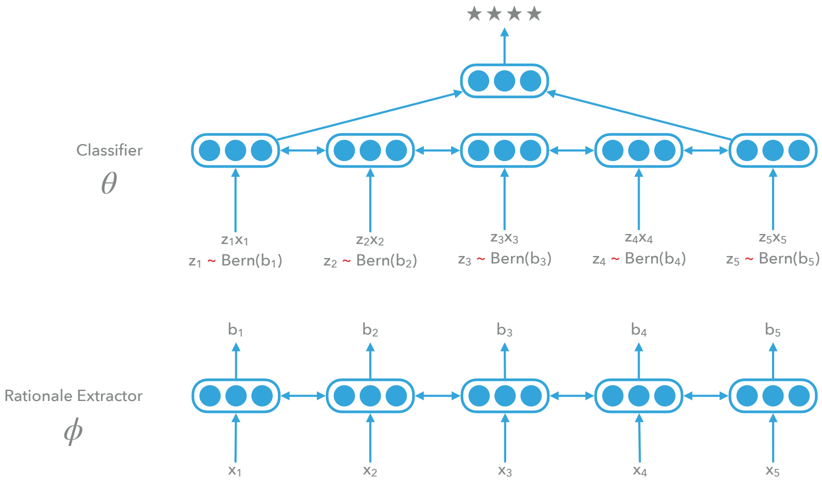
Latent rationales with Bernoulli selectors



Latent rationales with Bernoulli selectors

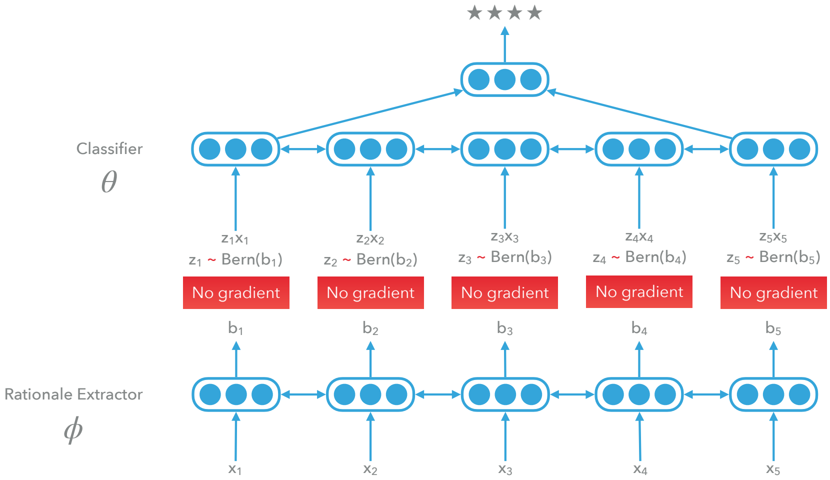


Latent rationales with Bernoulli selectors



Latent rationales with Bernoulli selectors

Requires gradient estimation via REINFORCE!



Outline

A rectified distribution

Consider a Gaussian variable $\epsilon \sim \mathcal{N}(0, 1)$

Rectified Gaussians in machine learning (Socci et al. 1998, Winn and Bishop 2005)

A rectified distribution

Consider a Gaussian variable $\epsilon \sim \mathcal{N}(0, 1)$

- What is the probability of sampling **exactly** 0?

Rectified Gaussians in machine learning (Socci et al. 1998, Winn and Bishop 2005)

A rectified distribution

Consider a Gaussian variable $\epsilon \sim \mathcal{N}(0, 1)$

- What is the probability of sampling **exactly** 0? 0!

Rectified Gaussians in machine learning (Socci et al. 1998, Winn and Bishop 2005)

A rectified distribution

Consider a Gaussian variable $\epsilon \sim \mathcal{N}(0, 1)$

- What is the probability of sampling **exactly** 0? 0!
- What is the probability of sampling a *negative number*?

Rectified Gaussians in machine learning (Socci et al. 1998, Winn and Bishop 2005)

A rectified distribution

Consider a Gaussian variable $\epsilon \sim \mathcal{N}(0, 1)$

- What is the probability of sampling **exactly** 0? 0!
- What is the probability of sampling a *negative number*? 0.5 or alternatively, $\Phi(0) = \int_{-\infty}^{0} \mathcal{N}(\epsilon|0, 1)\mathrm{d}\epsilon$

Rectified Gaussians in machine learning (Socci et al. 1998, Winn and Bishop 2005)

A rectified distribution

Consider a Gaussian variable $\epsilon \sim \mathcal{N}(0, 1)$

- What is the probability of sampling **exactly** 0? 0!
- What is the probability of sampling a *negative number*?
  0.5 or alternatively, $\Phi(0) = \int_{-\infty}^{0} \mathcal{N}(\epsilon|0, 1)\mathrm{d}\epsilon$

Consider the variable

$$\epsilon \sim \mathcal{N}(0, 1)$$
$$h = \max(0, \epsilon)$$

Rectified Gaussians in machine learning (Socci et al. 1998, Winn and Bishop 2005)

A rectified distribution

Consider a Gaussian variable $\epsilon \sim \mathcal{N}(0, 1)$

- What is the probability of sampling **exactly** 0? 0!
- What is the probability of sampling a *negative number*?
  0.5 or alternatively, $\Phi(0) = \int_{-\infty}^{0} \mathcal{N}(\epsilon|0, 1)\mathrm{d}\epsilon$

Consider the variable

$$\epsilon \sim \mathcal{N}(0, 1)$$
$$h = \max(0, \epsilon)$$

- What's the probability of sampling ϵ exactly 0?

Rectified Gaussians in machine learning (Socci et al. 1998, Winn and Bishop 2005)

A rectified distribution

Consider a Gaussian variable $\epsilon \sim \mathcal{N}(0, 1)$

- What is the probability of sampling **exactly** 0? 0!
- What is the probability of sampling a *negative number*?
  0.5 or alternatively, $\Phi(0) = \int_{-\infty}^{0} \mathcal{N}(\epsilon|0, 1)\mathrm{d}\epsilon$

Consider the variable

$$\epsilon \sim \mathcal{N}(0, 1)$$
$$h = \max(0, \epsilon)$$

- What's the probability of sampling ϵ exactly 0? 0!

Rectified Gaussians in machine learning (Socci et al. 1998, Winn and Bishop 2005)

A rectified distribution

Consider a Gaussian variable $\epsilon \sim \mathcal{N}(0, 1)$

- What is the probability of sampling **exactly** 0? 0!
- What is the probability of sampling a *negative number*?
  0.5 or alternatively, $\Phi(0) = \int_{-\infty}^{0} \mathcal{N}(\epsilon|0, 1)\mathrm{d}\epsilon$

Consider the variable

$$\epsilon \sim \mathcal{N}(0, 1)$$
$$h = \max(0, \epsilon)$$

- What's the probability of sampling ϵ exactly 0? 0!
- What's the probability of sampling h exactly 0?

Rectified Gaussians in machine learning (Socci et al. 1998, Winn and Bishop 2005)

A rectified distribution

Consider a Gaussian variable $\epsilon \sim \mathcal{N}(0, 1)$

- What is the probability of sampling **exactly** 0? 0!
- What is the probability of sampling a *negative number*?
  0.5 or alternatively, $\Phi(0) = \int_{-\infty}^{0} \mathcal{N}(\epsilon|0, 1)\mathrm{d}\epsilon$

Consider the variable

$$\epsilon \sim \mathcal{N}(0, 1)$$
$$h = \max(0, \epsilon)$$

- What's the probability of sampling ϵ exactly 0? 0!
- What's the probability of sampling h exactly 0? 0.5!

Rectified Gaussians in machine learning (Socci et al. 1998, Winn and Bishop 2005)

A rectified distribution

Consider a Gaussian variable $\epsilon \sim \mathcal{N}(0, 1)$

- What is the probability of sampling **exactly** 0? 0!
- What is the probability of sampling a *negative number*?
  0.5 or alternatively, $\Phi(0) = \int_{-\infty}^{0} \mathcal{N}(\epsilon|0, 1)\mathrm{d}\epsilon$

Consider the variable

$$\epsilon \sim \mathcal{N}(0, 1)$$
$$h = \max(0, \epsilon)$$

- What's the probability of sampling ϵ exactly 0? 0!
- What's the probability of sampling h exactly 0? 0.5!
- Where is the $\max$ non-differentiable?

Rectified Gaussians in machine learning (Socci et al. 1998, Winn and Bishop 2005)

A rectified distribution

Consider a Gaussian variable $\epsilon \sim \mathcal{N}(0, 1)$

- What is the probability of sampling **exactly** 0? 0!
- What is the probability of sampling a *negative number*? 0.5 or alternatively, $\Phi(0) = \int_{-\infty}^{0} \mathcal{N}(\epsilon|0, 1)\mathrm{d}\epsilon$

Consider the variable

$$\epsilon \sim \mathcal{N}(0, 1)$$
$$h = \max(0, \epsilon)$$

- What's the probability of sampling ϵ exactly 0? 0!
- What's the probability of sampling h exactly 0? 0.5!
- Where is the $\max$ non-differentiable? At $\epsilon = 0$

Rectified Gaussians in machine learning (Socci et al. 1998, Winn and Bishop 2005)

A rectified distribution

Consider a Gaussian variable $\epsilon \sim \mathcal{N}(0, 1)$

- What is the probability of sampling **exactly** 0? 0!
- What is the probability of sampling a *negative number*?
  0.5 or alternatively, $\Phi(0) = \int_{-\infty}^{0} \mathcal{N}(\epsilon|0, 1)\mathrm{d}\epsilon$

Consider the variable

$$\epsilon \sim \mathcal{N}(0, 1)$$
$$h = \max(0, \epsilon)$$

- What's the probability of sampling ϵ exactly 0? 0!
- What's the probability of sampling h exactly 0? 0.5!
- Where is the $\max$ non-differentiable? At $\epsilon = 0$
- Will we ever sample $\epsilon = 0$?

Rectified Gaussians in machine learning (Socci et al. 1998, Winn and Bishop 2005)

A rectified distribution

Consider a Gaussian variable $\epsilon \sim \mathcal{N}(0, 1)$

- What is the probability of sampling **exactly** 0? 0!
- What is the probability of sampling a *negative number*?
  0.5 or alternatively, $\Phi(0) = \int_{-\infty}^{0} \mathcal{N}(\epsilon|0, 1)\mathrm{d}\epsilon$

Consider the variable

$$\epsilon \sim \mathcal{N}(0, 1)$$
$$h = \max(0, \epsilon)$$

- What's the probability of sampling ϵ exactly 0? 0!
- What's the probability of sampling h exactly 0? 0.5!
- Where is the $\max$ non-differentiable? At $\epsilon = 0$
- Will we ever sample $\epsilon = 0$? No :D

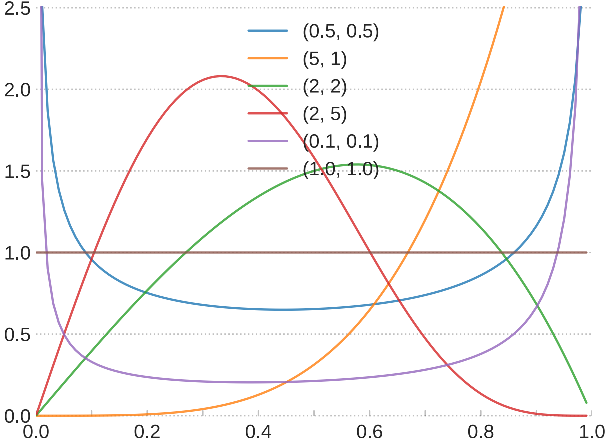
Rectified Gaussians in machine learning (Socci et al. 1998, Winn and Bishop 2005)

HardKumaraswamy

We propose a distribution that

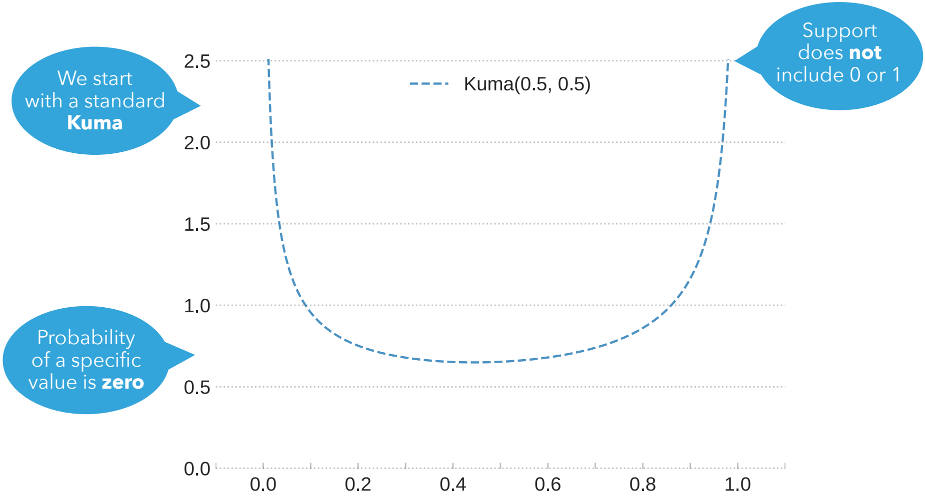
- gives support to the **closed** interval $[0, 1]$
- and assign non-zero probability to outcomes 0 and 1
  $\mathbb{P}(z \in \{0\}) > 0$ and $\mathbb{P}(z \in \{1\}) > 1$

HardKumaraswamy



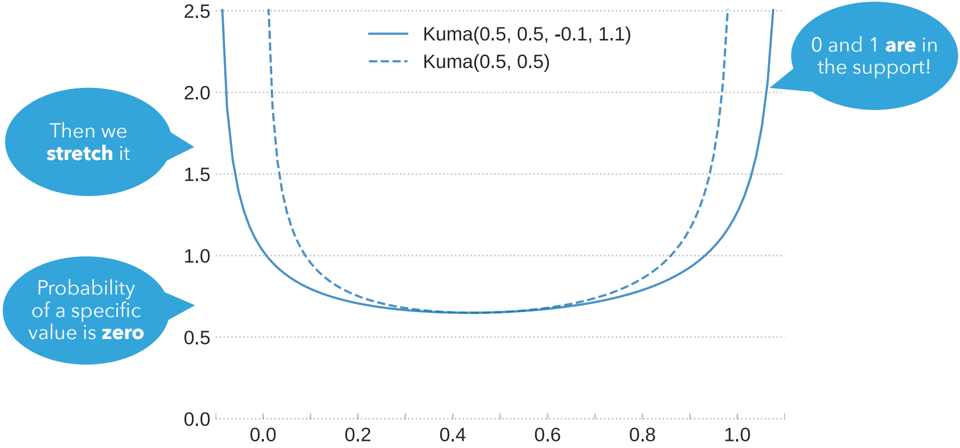
Kumaraswamy distribution (Kumaraswamy 1980) in machine learning (Nalisnick and Smyth 2016)

HardKumaraswamy



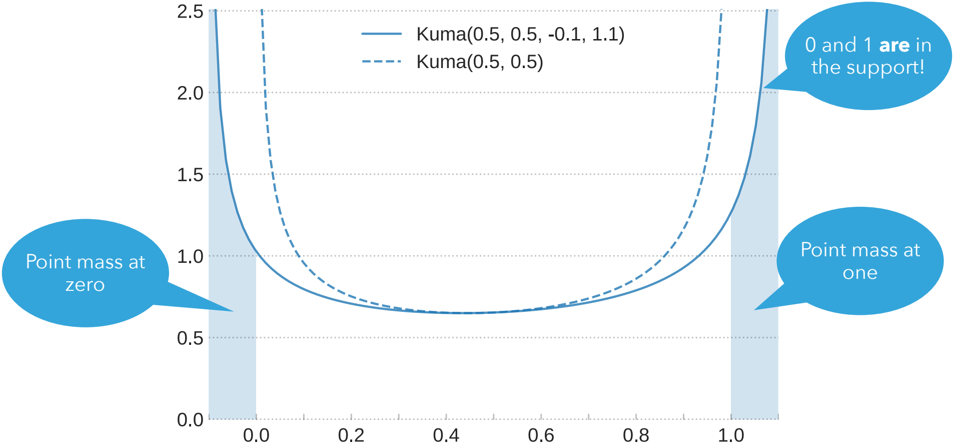
Kumaraswamy distribution (Kumaraswamy 1980) in machine learning (Nalisnick and Smyth 2016)

HardKumaraswamy



Kumaraswamy distribution (Kumaraswamy 1980) in machine learning (Nalisnick and Smyth 2016)

HardKumaraswamy



Kumaraswamy distribution (Kumaraswamy 1980) in machine learning (Nalisnick and Smyth 2016)

Sampling a HardKumaraswamy variable

If $Z \sim \mathrm{HKuma}(a, b, l, r)$

Louizos et al. (2017) proposed this *stretch-and-rectify* technique using Binary Concrete variables (Maddison et al. 2017, Jang et al. 2017) in the context of Bayesian NNs.

Sampling a HardKumaraswamy variable

If $Z \sim \mathrm{HKuma}(a, b, l, r)$

$$u \sim \mathcal{U}(0, 1) \qquad \text{Fixed random source}$$

Louizos et al. (2017) proposed this *stretch-and-rectify* technique using Binary Concrete variables (Maddison et al. 2017, Jang et al. 2017) in the context of Bayesian NNs.

Sampling a HardKumaraswamy variable

If $Z \sim \mathrm{HKuma}(a, b, l, r)$

$$u \sim \mathcal{U}(0, 1) \qquad \text{Fixed random source}$$

$$k = \underbrace{(1 - (1 - u)^{1/b})^{1/a}}_{\text{inverse cdf}} \qquad \sim \mathrm{Kuma}(a, b)$$

Louizos et al. (2017) proposed this *stretch-and-rectify* technique using Binary Concrete variables (Maddison et al. 2017, Jang et al. 2017) in the context of Bayesian NNs.

Sampling a HardKumaraswamy variable

If $Z \sim \mathrm{HKuma}(a, b, l, r)$

$$
\begin{aligned}
u &\sim \mathcal{U}(0, 1) && \text{Fixed random source} \\
k &= \underbrace{(1 - (1 - u)^{1/b})^{1/a}}_{\text{inverse cdf}} && \sim \mathrm{Kuma}(a, b) \\
t &= \underbrace{l + (r - l)k}_{\text{stretch}} && \sim \mathrm{Kuma}(a, b, l, r)
\end{aligned}
$$

Louizos et al. (2017) proposed this *stretch-and-rectify* technique using Binary Concrete variables (Maddison et al. 2017, Jang et al. 2017) in the context of Bayesian NNs.

Sampling a HardKumaraswamy variable

If $Z \sim \mathrm{HKuma}(a, b, l, r)$

$$
\begin{aligned}
u &\sim \mathcal{U}(0,1) && \text{Fixed random source} \\
k &= \underbrace{(1-(1-u)^{1/b})^{1/a}}_{\text{inverse cdf}} && \sim \mathrm{Kuma}(a, b) \\
t &= \underbrace{l + (r-l)k}_{\text{stretch}} && \sim \mathrm{Kuma}(a, b, l, r) \\
z &= \underbrace{\min(1, \max(0, t))}_{\text{rectify}} && \sim \mathrm{HKuma}(a, b, l, r)
\end{aligned}
$$

Louizos et al. (2017) proposed this *stretch-and-rectify* technique using Binary Concrete variables (Maddison et al. 2017, Jang et al. 2017) in the context of Bayesian NNs.

Sampling a HardKumaraswamy variable

If $Z \sim \mathrm{HKuma}(a, b, l, r)$

$$
\begin{aligned}
u &\sim \mathcal{U}(0,1) && \text{Fixed random source} \\
k &= \underbrace{(1-(1-u)^{1/b})^{1/a}}_{\text{inverse cdf}} && \sim \mathrm{Kuma}(a, b) \\
t &= \underbrace{l + (r-l)k}_{\text{stretch}} && \sim \mathrm{Kuma}(a, b, l, r) \\
z &= \underbrace{\min(1, \max(0, t))}_{\text{rectify}} && \sim \mathrm{HKuma}(a, b, l, r)
\end{aligned}
$$

- Is this differentiable wrt a, b?

Louizos et al. (2017) proposed this *stretch-and-rectify* technique using Binary Concrete variables (Maddison et al. 2017, Jang et al. 2017) in the context of Bayesian NNs.

Sampling a HardKumaraswamy variable

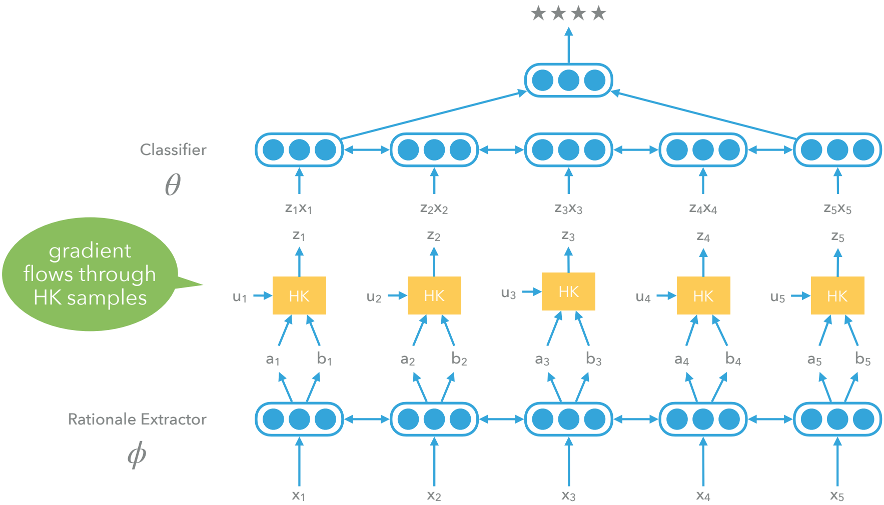
If $Z \sim \mathrm{HKuma}(a, b, l, r)$

$$
\begin{aligned}
u &\sim \mathcal{U}(0,1) && \text{Fixed random source} \\
k &= \underbrace{(1-(1-u)^{1/b})^{1/a}}_{\text{inverse cdf}} && \sim \mathrm{Kuma}(a,b) \\
t &= \underbrace{l + (r-l)k}_{\text{stretch}} && \sim \mathrm{Kuma}(a,b,l,r) \\
z &= \underbrace{\min(1, \max(0, t))}_{\text{rectify}} && \sim \mathrm{HKuma}(a,b,l,r)
\end{aligned}
$$

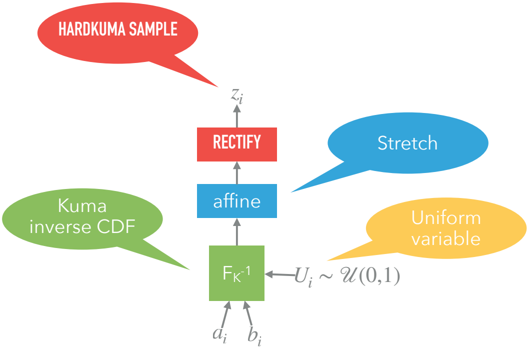
- Is this differentiable wrt a, b?
  Yes, reparameterised gradients are available!

Louizos et al. (2017) proposed this *stretch-and-rectify* technique using Binary Concrete variables (Maddison et al. 2017, Jang et al. 2017) in the context of Bayesian NNs.

Latent rationales with HardKuma selectors



Latent rationales with HardKuma selectors



Strictly positive shape parameters $a_i, b_i = g_i(x; \phi)$

Promoting sparsity

Short selections: penalise expected number of non-zero selectors

$$\boxed{\mathbb{E}_{P(z|x,\phi)}\left[\sum_{i=1}^{n}\mathbb{I}[z_i \neq 0]\right]}$$

Relaxed L_0 due to Louizos et al. (2017)

Promoting sparsity

Short selections: penalise expected number of non-zero selectors

$$\boxed{\mathbb{E}_{P(z|x,\phi)}\left[\sum_{i=1}^{n} \mathbb{I}[z_i \neq 0]\right]}$$

Coherent groups: penalise expected number of zero-to-nonzero and nonzero-to-zero changes

$$\boxed{\mathbb{E}_{P(z|x,\phi)}\left[\sum_{i=1}^{n-1} \mathbb{I}[z_i = 0, z_{i+1} \neq 0]\right]} + \boxed{\mathbb{E}_{P(z|x,\phi)}\left[\sum_{i=1}^{n-1} \mathbb{I}[z_i \neq 0, z_{i+1} = 0]\right]}$$

Relaxed L_0 due to Louizos et al. (2017)

Promoting sparsity

Short selections: penalise expected number of non-zero selectors

$$\boxed{\mathbb{E}_{P(z|x,\phi)}\left[\sum_{i=1}^{n} \mathbb{I}[z_i \neq 0]\right]}$$

Coherent groups: penalise expected number of zero-to-nonzero and nonzero-to-zero changes

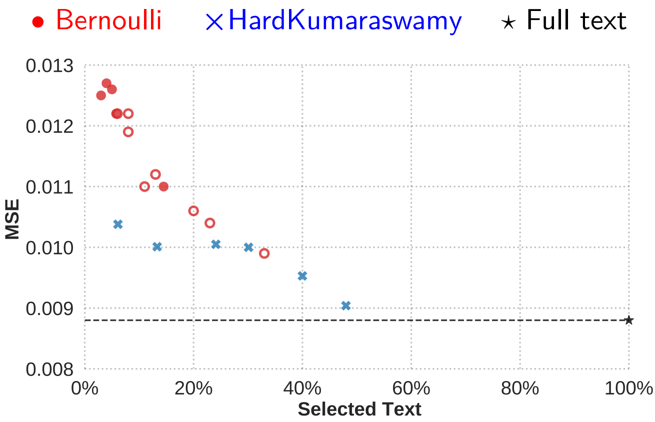
$$\boxed{\mathbb{E}_{P(z|x,\phi)}\left[\sum_{i=1}^{n-1} \mathbb{I}[z_i = 0, z_{i+1} \neq 0]\right]} + \boxed{\mathbb{E}_{P(z|x,\phi)}\left[\sum_{i=1}^{n-1} \mathbb{I}[z_i \neq 0, z_{i+1} = 0]\right]}$$

Tractable and differentiable function of ϕ

Relaxed L_0 due to Louizos et al. (2017)

Outline

BeerAdvocate



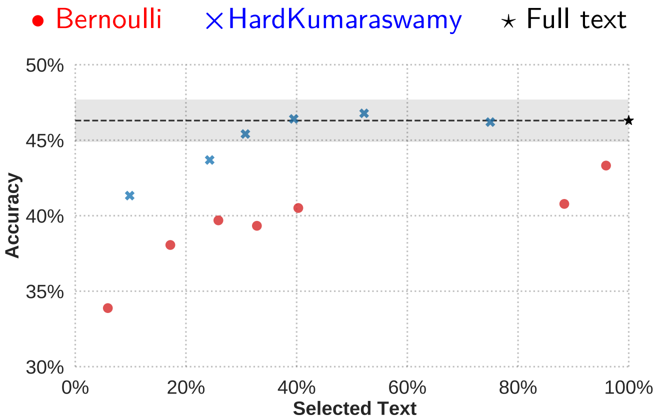
Regression to sentiment score $[0, 1]$

BeerAdvocate

Method	Target rate	Look		Smell		Taste	
		% Precision	% Selected	% Precision	% Selected	% Precision	% Selected
Attention (Lei et al.)	Threshold	80.6	13	88.4	7	65.3	7
Bernoulli (Lei et al.)	Grid	96.3	14	95.1	7	80.2	7
Bernoulli *(reimpl.)*	Grid	94.8	13	95.1	7	80.5	7
HardKuma	Lagrange	**98.1**	13	**96.8**	7	**89.8**	7

Regression to sentiment score $[0, 1]$

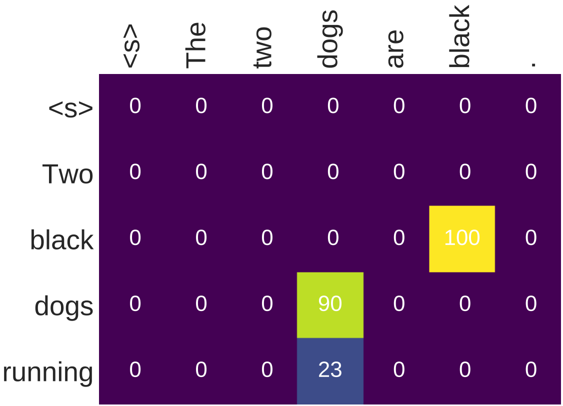
Stanford sentiment classification



Classification from very negative to very positive

Stanford natural language inference

Entailment



Stanford natural language inference

Contradiction

	<s>	Three	cats	race	on	a	track	.
<s>	0	0	0	0	0	0	0	0
Three	0	84	0	0	0	0	0	0
dogs	0	0	100	0	0	0	18	0
racing	0	0	0	87	0	0	43	0
on	0	0	0	0	0	0	0	0
racetrack	0	0	33	48	0	0	73	0

Stanford natural language inference

	Accuracy	
	Dev	Test
LSTM (Bowman et al. 2016)	–	80.6
DA (Parikh et al. 2016)	–	86.3
DA *(reimplementation)*	86.9	86.5
DA with HardKuma attention	86.0	85.5

1% drop with 8.6% of non-zero attention cells

Outline

Remarks

Distributions that mix discrete and continuous behaviour are typically used to sparsify models (i.e. parameters)

We show how to use them to construct differentiable sparse layers

- for sentiment classification (sparse rationale)
- and natural language inference (sparse attention)

Other applications we are looking into include

- adjacency in a graph
- keys/values in memory networks

Remarks

Distributions that mix discrete and continuous behaviour are typically used to sparsify models (i.e. parameters)

We show how to use them to construct differentiable sparse layers

- for sentiment classification (sparse rationale)
- and natural language inference (sparse attention)

Other applications we are looking into include

- adjacency in a graph
- keys/values in memory networks

Thanks!

References I

Samuel R. Bowman, Jon Gauthier, Abhinav Rastogi, Raghav Gupta, Christopher D. Manning, and Christopher Potts. A fast unified model for parsing and sentence understanding. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1466–1477. Association for Computational Linguistics, 2016. doi: 10.18653/v1/P16-1139. URL `http://aclweb.org/anthology/P16-1139`.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *International Conference on Learning Representations*, 2017.

MichaelI. Jordan, Zoubin Ghahramani, TommiS. Jaakkola, and LawrenceK. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.

References II

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

Ponnambalam Kumaraswamy. A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2):79–88, 1980.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117. Association for Computational Linguistics, 2016. doi: 10.18653/v1/D16-1011. URL http://aclweb.org/anthology/D16-1011.

Christos Louizos, Max Welling, and Diederik P Kingma. Learning sparse neural networks through l_0 regularization. *arXiv preprint arXiv:1712.01312*, 2017.

References III

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continous relaxation of discrete random variables. *International Conference on Learning Representations*, 2017.

Eric Nalisnick and Padhraic Smyth. Stick-breaking variational autoencoders. *arXiv preprint arXiv:1605.06197*, 2016.

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255. Association for Computational Linguistics, 2016. doi: 10.18653/v1/D16-1244. URL `http://aclweb.org/anthology/D16-1244`.

Nicholas D. Socci, Daniel D. Lee, and H. Sebastian Seung. The rectified gaussian distribution. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural Information Processing Systems 10*, pages 350–356. MIT Press, 1998.

References IV

John Winn and Christopher M Bishop. Variational message passing. *Journal of Machine Learning Research*, 6(Apr):661–694, 2005.

Controlled sparsity

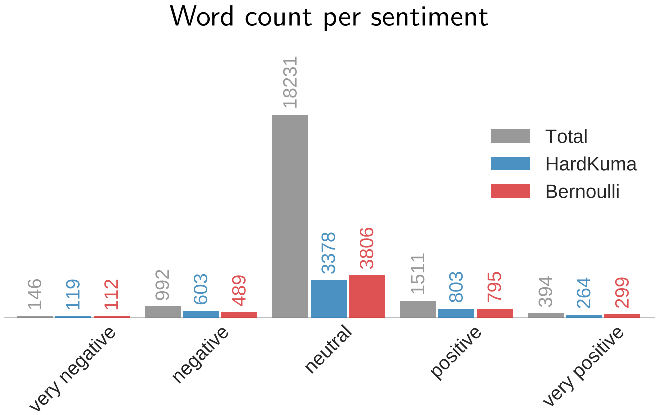
We specify target values t for the sparsity-inducing penalties $R(\phi)$ and employ Langrangian relaxation

$$\min_{\lambda} \max_{\phi,\theta} \mathcal{L}(\phi, \theta) - \lambda^\top (R(\phi) - t)$$

where $\mathcal{L}(\theta, \phi)$ is a lowerbound on the log-likelihood function

A simple form of variational inference (Jordan et al. 1999), an instance of a VAE (Kingma and Welling 2014).

Sentiment words



Reparameterised gradients

$$\frac{\partial \mathcal{L}}{\partial u} = \frac{\partial \mathcal{L}}{\partial h} \times \frac{\partial h}{\partial t} \times \frac{\partial t}{\partial k} \times \frac{\partial k}{\partial u}$$

$$\begin{aligned} k &= F_K^{-1}(u; a, b) \\ t &= l + (r - l)k \\ h &= \min(1, \max(0, t)) \end{aligned}$$

ELBO

We need to marginalise all possible latent assignments:

$$\log P(y|x,\theta,\phi) = \log \sum_z P(z|x,\phi) P(y|x \odot z, \theta)$$

but there 2^n of those!

Let's derive a lowerbound

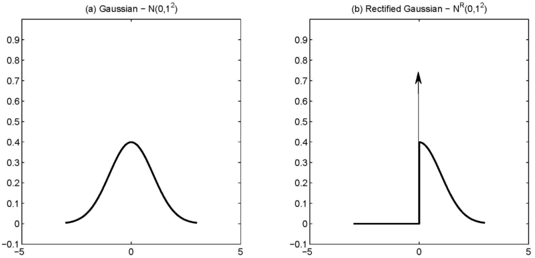
$$\log P(y|x,\theta,\phi) \overset{\text{JI}}{\geq} \underbrace{\sum_z P(z|x,\phi) \log P(y|x \odot z, \theta)}_{\mathcal{L}(\theta,\phi|x,y)}$$

and work with gradient estimates instead

$$\begin{aligned}
\boldsymbol{\nabla}_\theta \mathcal{L}(\theta,\phi|x,y) &= \mathbb{E}_{P(z|x,\phi)}[\boldsymbol{\nabla}_\theta \log P(y|x \odot z, \theta)] \\
\boldsymbol{\nabla}_\phi \mathcal{L}(\theta,\phi|x,y) &= \mathbb{E}_{P(z|x,\phi)}[\log P(y|x \odot z, \theta) \boldsymbol{\nabla}_\phi \log P(z|x,\phi)]
\end{aligned}$$

Rectified Gaussian

As we know the cdf of a Gaussian variable
we can collapse some of the probability mass to a single point



This variable mixes discrete and continuous behaviour.

Images from Wikipedia
Rectified Gaussians in machine learning (Socci et al. 1998, Winn and Bishop 2005)

Distribution function

For the rectified Gaussian

$$f_H(h) = F_\epsilon(0|\mu,\sigma)\delta(h) + (1 - F_\epsilon(0|\mu,\sigma))\mathcal{N}(h|\mu,\sigma^2)\mathbf{1}_{\mathbb{R}_{>0}}(h)$$

For the Hard Kumaraswamy

$$\begin{aligned}
f_H(h;a,b,l,r) &= \mathbb{P}(h=0)\delta(h) + \mathbb{P}(h=1)\delta(h-1) \\
&\quad + \mathbb{P}(0<h<1) f_T(h;a,b,l,r)\mathbf{1}_{(0,1)}(h) \\
f_T(t;a,b,l,r) &= f_K\left(\tfrac{t-l}{r-l};a,b\right)\frac{1}{(r-l)} \\
F_T(t;a,b,l,r) &= f_K\left(\tfrac{t-l}{r-l};a,b\right) \\
f_K(k;a,b) &= abk^{a-1}(1-k^a)^{b-1} \\
F_K(k;a,b) &= 1-(1-k^a)^b \\
F_K^{-1}(u;a,b) &= \left(1-(1-u)^{1/b}\right)^{1/a}
\end{aligned}$$