Reward Augmented Maximum Likelihood (RAML) for Neural Structured Prediction

Mohammad Norouzi   Samy Bengio   Zhifeng Chen   Navdeep Jaitly   Mike Schuster   Yonghui Wu   Dale Schuurmans

Problem

Structured output prediction: learning a mapping from inputs to complex multivariate outputs ($\mathbf{x} \rightarrow \mathbf{y}$)

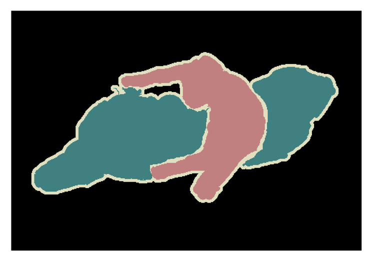
Given a dataset of input-output pairs,

$$\mathcal{D} \equiv \{ (\mathbf{x}^{(i)}, \mathbf{y}^{*(i)}) \}_{i=1}^{N} ,$$

learn a conditional distribution $p_\theta(\mathbf{y} \mid \mathbf{x})$ consistent with $\mathcal{D}$.

- Image captioning
- Semantic segmentation

x → **y**



- Machine translation

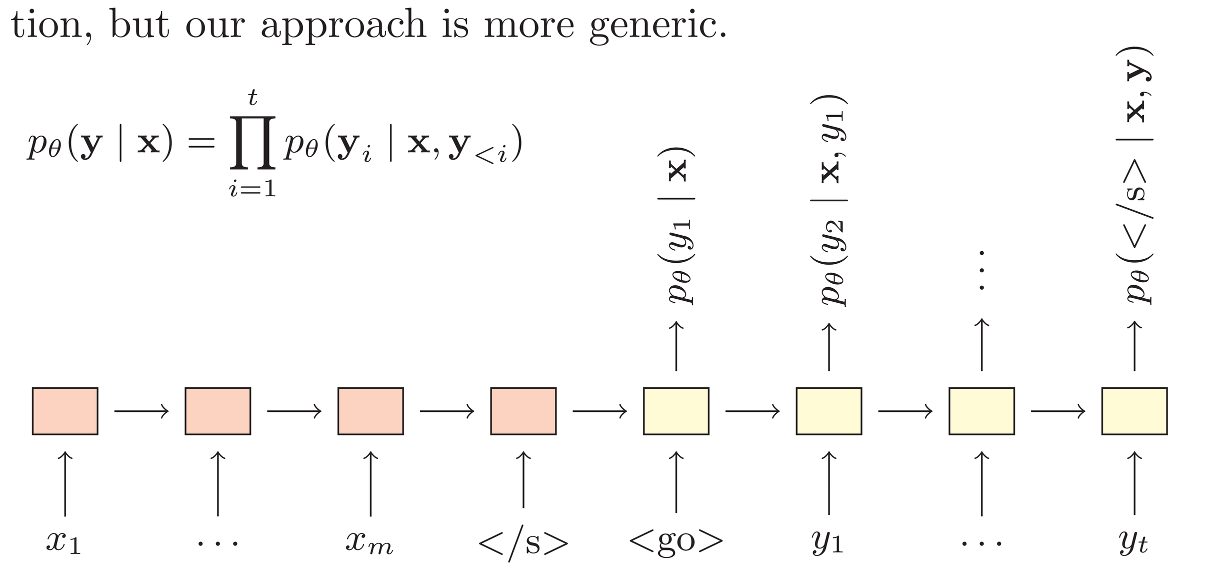
As diets change, people get bigger but plane seating has not radically changed.	→	Avec les changements dans les habitudes alimentaires, les gens grossissent, mais les sièges dans les avions n'ont pas radicalement changé.

- Speech recognition

Model

We use autoregressive sequence to sequence models with attention, but our approach is more generic.

$$p_\theta(\mathbf{y} \mid \mathbf{x}) = \prod_{i=1}^{t} p_\theta(\mathbf{y}_i \mid \mathbf{x}, \mathbf{y}_{<i})$$



- At inference, beam search finds $\widehat{\mathbf{y}}(\mathbf{x}) \approx \operatorname{argmax}_{\mathbf{y}} \; p_\theta(\mathbf{y} \mid \mathbf{x})$.
- As the reward signal, *BLEU* score or negative *edit distance* measure the quality of the predictions: $\sum_{(\mathbf{x},\mathbf{y}^*)} r(\widehat{\mathbf{y}}(\mathbf{x}), \mathbf{y}^*)$

Related Work

- [*Szegedy* et al., CVPR'16] Rethinking the Inception
  Label smoothing can be thought as a special case of our method

Some alternative methods all of which require either sampling or inference from the model during training:

- [*S. Bengio* et al., NIPS'15] Schedule sampling
- [*Ranzato* et al., ICLR'16] Sequence level training REINFORCE for machine translation
- [*Wiseman & Rush*, EMNLP'16] Beam search optimization

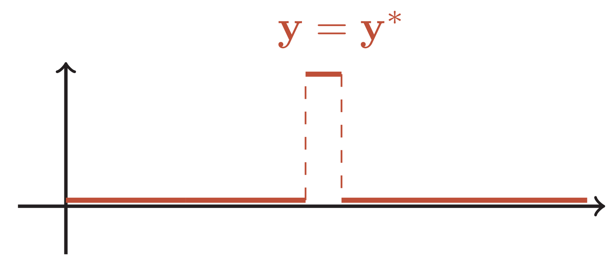
ML

Conditional log-likelihood:

$$\mathcal{O}_{\mathrm{ML}}(\boldsymbol{\theta}) = \sum_{(\mathbf{x},\mathbf{y}^*)} \log p_\theta(\mathbf{y}^* \mid \mathbf{x})$$
$$= \sum_{(\mathbf{x},\mathbf{y}^*)} -D_{\mathrm{KL}}\big(\mathbb{1}[\mathbf{y} = \mathbf{y}^*] \;\|\; p_\theta(\mathbf{y} \mid \mathbf{x})\big)$$

- There is no notion of reward (*e.g.* BLEU score, edit distance).
- All of the negative outputs $\mathbf{y} \neq \mathbf{y}^*$ are equally penalized.

Optimal $p_\theta(\mathbf{y} \mid \mathbf{x})$:



RL

Entropy regularized expected reward (with a regularizer τ):

$$\mathcal{O}_{\mathrm{RL}}(\boldsymbol{\theta}; \tau) = \sum_{(\mathbf{x},\mathbf{y}^*)} \Big[\underbrace{\tau \mathbb{H}(p_\theta(\mathbf{y} \mid \mathbf{x}))}_{\text{entropy}} + \underbrace{\sum_{\mathbf{y} \in \mathcal{Y}} p_\theta(\mathbf{y} \mid \mathbf{x}) \, r(\mathbf{y}, \mathbf{y}^*)}_{\text{expected reward}} \Big]$$

- To optimize $\mathcal{O}_{\mathrm{RL}}$, one uses REINFORCE, *e.g.* [*Ranzato* et al.], to compute $\nabla_{\boldsymbol{\theta}} \mathcal{O}_{\mathrm{RL}}$ by sampling from $p_\theta(\mathbf{y} \mid \mathbf{x})$.
- The gradients are high variance. The training is slow.
- One needs to bootstrap training from an ML trained model.
- REINFORCE ignores direct supervision after initialization.

Key observation

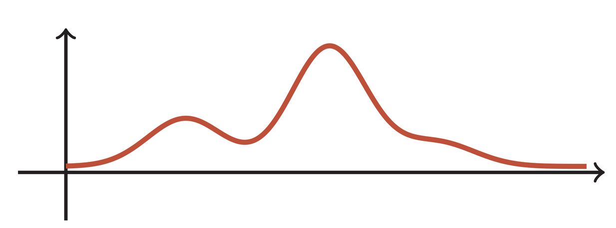
One can re-express $\mathcal{O}_{\mathrm{RL}}$ as:

$$\mathcal{O}_{\mathrm{RL}}(\boldsymbol{\theta}; \tau) = \sum_{(\mathbf{x},\mathbf{y}^*)} -\tau D_{\mathrm{KL}}\big(p_\theta(\mathbf{y} \mid \mathbf{x}) \;\|\; \underbrace{q_\tau(\mathbf{y} \mid \mathbf{y}^*)}_{\text{exponentiated payoff}}\big) + \text{const}$$

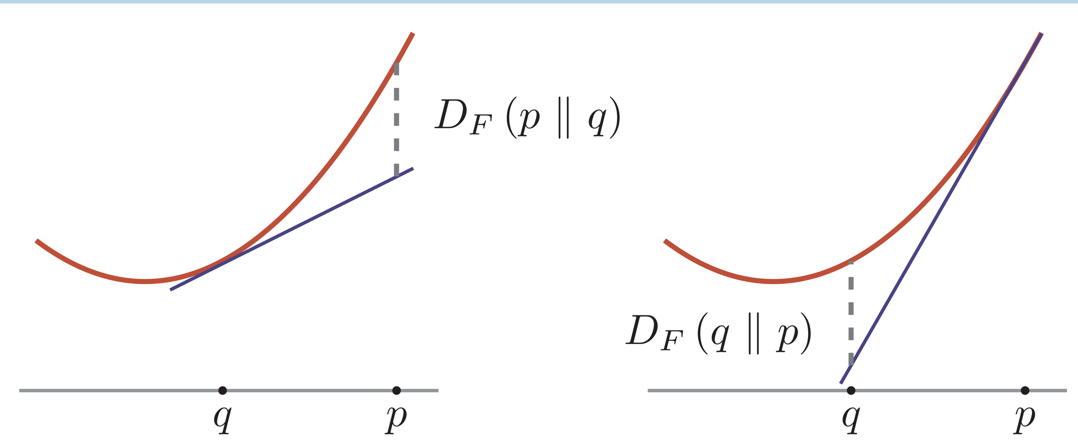
The exponentiated payoff distribution globally maximizes $\mathcal{O}_{\mathrm{RL}}$:

$$q_\tau(\mathbf{y} \mid \mathbf{y}^*) = \frac{1}{Z} \exp\{ r(\mathbf{y}, \mathbf{y}^*) / \tau \}$$

Optimal $p_\theta(\mathbf{y} \mid \mathbf{x})$:



KL as a Bregman divergence



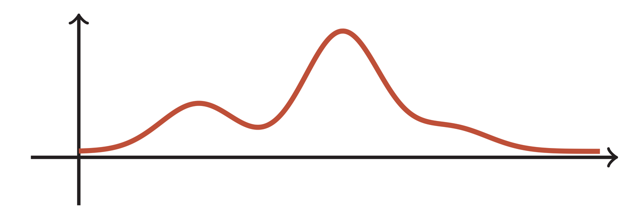
RAML

We propose *reward augmented* conditional log-likelihood:

$$\mathcal{O}_{\mathrm{RAML}}(\boldsymbol{\theta}; \tau) = \sum_{(\mathbf{x},\mathbf{y}^*)} \sum_{\mathbf{y} \in \mathcal{Y}} q_\tau(\mathbf{y} \mid \mathbf{y}^*) \log p_\theta(\mathbf{y} \mid \mathbf{x})$$
$$= \sum_{(\mathbf{x},\mathbf{y}^*)} -D_{\mathrm{KL}}\big(q_\tau(\mathbf{y} \mid \mathbf{y}^*) \;\|\; p_\theta(\mathbf{y} \mid \mathbf{x})\big) + \text{const}$$

- Similar to ML, in the direction of KL.
  Similar to RL, in the optimal conditional distribution
- There is a notion of reward captured in q_τ.

Optimal $p_\theta(\mathbf{y} \mid \mathbf{x})$
$\propto \exp\{ r(\mathbf{y}, \mathbf{y}^*) / \tau \}$:



- The temperature τ controls the concentration of q_τ.
  As $\tau \rightarrow 0$, then $q_\tau(\mathbf{y} \mid \mathbf{y}^*) \rightarrow \mathbb{1}[\mathbf{y} = \mathbf{y}^*]$.
- This objective is convex in the softmax weights.

RAML Optimization

Training with RAML is efficient and easy to implement.

- Given a training case $(\mathbf{x}^{(i)}, \mathbf{y}^{*(i)})$, first sample $\widetilde{\mathbf{y}} \sim q_\tau(\mathbf{y} \mid \mathbf{y}^{*(i)})$ then optimize $\log p_\theta(\widetilde{\mathbf{y}} \mid \mathbf{x}^{(i)})$. These samples can be cashed.

$$\nabla_{\boldsymbol{\theta}} \mathcal{O}_{\mathrm{RAML}}(\boldsymbol{\theta}; \tau) = \sum_{(\mathbf{x},\mathbf{y}^*)} \mathbb{E}_{\widetilde{\mathbf{y}} \sim q(\mathbf{y} \mid \mathbf{y}^*; \tau)} \big[\nabla_{\boldsymbol{\theta}} \log p_\theta(\widetilde{\mathbf{y}} \mid \mathbf{x}) \big] .$$

- By contrast, in REINFORCE ($\tau = 0$), one samples from p_θ:

$$\nabla_{\boldsymbol{\theta}} \mathcal{O}_{\mathrm{RL}}(\boldsymbol{\theta}) = \sum_{(\mathbf{x},\mathbf{y}^*)} \mathbb{E}_{\widetilde{\mathbf{y}} \sim p_\theta(\mathbf{y} \mid \mathbf{x})} \big[\nabla_{\boldsymbol{\theta}} \log p_\theta(\widetilde{\mathbf{y}} \mid \mathbf{x}) \cdot r(\widetilde{\mathbf{y}}, \mathbf{y}^*) \big]$$

- RAML is a form of data augmentation on the targets based on the reward signal.
- We just sample one augmentation $\widetilde{\mathbf{y}}$ per input $\mathbf{x}$ per iteration.

Sampling from Exponentiated Payoff

Stratified sampling: first select a particular reward value, and then sample an output with that reward value.

- If reward is negative Hamming distance, $r(\mathbf{y}, \mathbf{y}^*) = -D_{\mathrm{H}}(\mathbf{y}, \mathbf{y}^*)$ one can draw exact samples from $q_\tau(\mathbf{y} \mid \mathbf{y}^*)$.

$$\text{if } \; \mathcal{Y} \equiv \{1, \ldots, v\}^m, \quad \text{then } \; r(\mathbf{y}, \mathbf{y}^*) \in \{0, \ldots, -m\}$$

It is easy to count $\{\mathbf{y} \in \mathcal{Y} \mid r(\mathbf{y}, \mathbf{y}^*) = k\}$: $\binom{m}{k}(v-1)^k$.
Summing over k, one can compute the normalization factor.

- For negative edit distance, an approximate sampler is proposed.
- Generally, one can resort to importance sampling and MCMC. Samples from $q_\tau(\mathbf{y} \mid \mathbf{y}^*)$ can be pre-computed and stored.

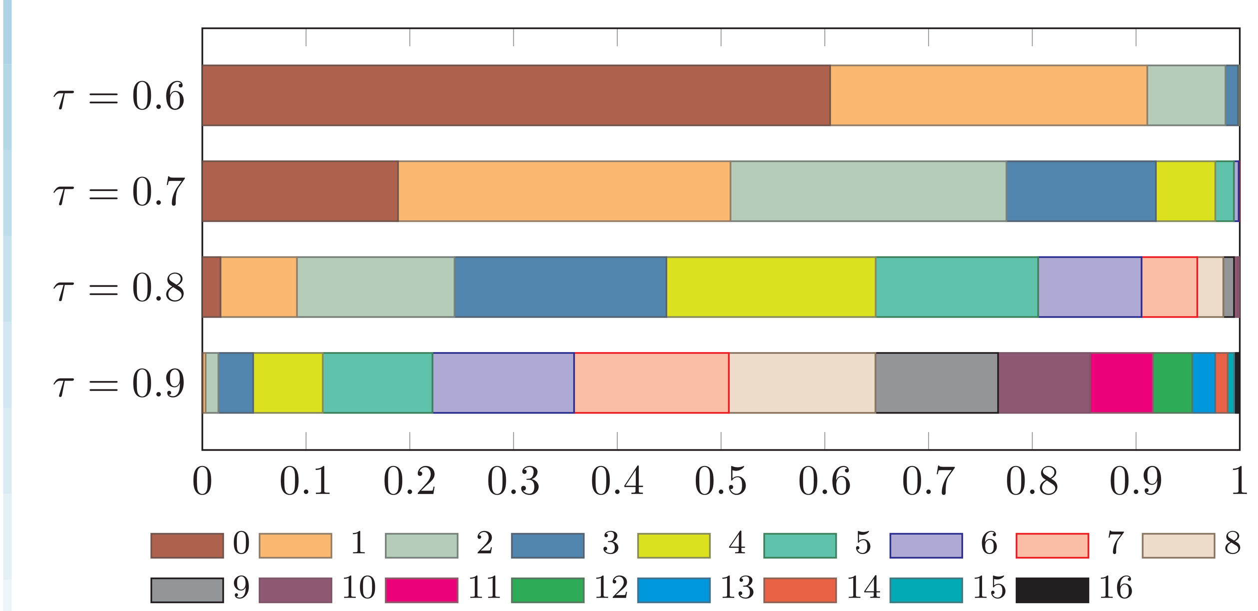
TIMIT Speech Recognition

Phone error rates (PER) for different methods on TIMIT dev & test sets. Average (min, max) PER for 4 training runs:

Method	Dev set	Test set
ML baseline	20.87 (−0.2, +0.3)	22.18 (−0.4, +0.2)
RAML, $\tau = 0.60$	19.92 (−0.6, +0.3)	21.65 (−0.5, +0.4)
RAML, $\tau = 0.65$	19.64 (−0.2, +0.5)	21.28 (−0.6, +0.4)
RAML, $\tau = 0.70$	18.97 (−0.1, +0.1)	21.28 (−0.5, +0.4)
RAML, $\tau = 0.75$	18.44 (−0.4, +0.4)	20.15 (−0.4, +0.4)
RAML, $\tau = 0.80$	18.27 (−0.2, +0.1)	19.97 (−0.1, +0.2)
RAML, $\tau = 0.85$	18.10 (−0.4, +0.3)	19.97 (−0.3, +0.2)
RAML, $\tau = 0.90$	**18.00** (−0.4, +0.3)	**19.89** (−0.4, +0.7)
RAML, $\tau = 0.95$	18.46 (−0.1, +0.1)	20.12 (−0.2, +0.1)
RAML, $\tau = 1.00$	18.78 (−0.6, +0.8)	20.41 (−0.2, +0.5)

Fraction of number of edits

Fraction of number of edits for a sequence of length 20:



At $\tau = 0.9$, augmentations with 5 to 9 edits are sampled with a probability > 0.1.

Machine Translation (WMT En→Fr)

Tokenized BLEU score on WMT'14 English to French:

Method	Average BLEU	Best BLEU
ML baseline	36.50	36.87
RAML, $\tau = 0.75$	36.62	36.91
RAML, $\tau = 0.80$	36.80	37.11
RAML, $\tau = 0.85$	**36.91**	**37.23**
RAML, $\tau = 0.90$	36.69	37.07
RAML, $\tau = 0.95$	36.57	36.94

The RAML approach with different τ considerably improves upon the maximum likelihood baseline.

Follow-up work: UREX

- Is RAML applicable to RL with unknown reward landscapes?

Improving Policy Gradient by
Exploring Under-appreciated Rewards. (arXiv:1611.09321)

The key idea is to sample from $p_\theta(y)$ and perform importance correction given $\exp\{r(y)/\tau\} / p_\theta(y)$.