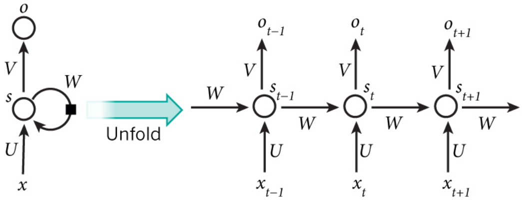
Noise-Based Regularizers for Recurrent Neural Networks

Adji Bousso Dieng

COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

Recurrent Neural Networks Are Awesome!

→ Yes, they are.



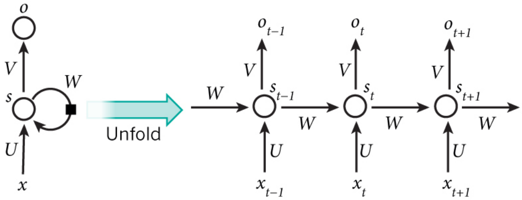
$$\mathbf{s}_t = f_\Psi(\mathbf{x}_t, \mathbf{s}_{t-1}) \text{ and } \mathbf{o}_t = V^\top \mathbf{s}_t \text{ where } \Psi = \{U, W\}$$

→ Recurrence + Parameter sharing $\implies \mathbf{s}_t = r(\mathbf{x}_t, ..., \mathbf{x}_1; \Psi)$

→ Very powerful model class for sequences

Hold On... RNNs Overfit Easily Though

→ Yes, unfortunately.

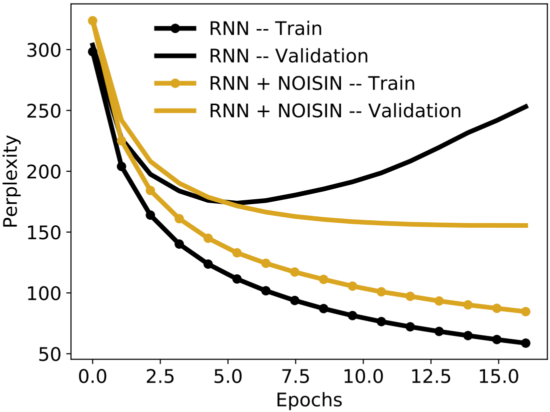


→ But we know what happens when they overfit:

"When a neural network overfits badly during training, its hidden states depend very heavily on each other."
– Hinton, 2012

So We Can Just Inject Noise And... Boom!

→ Absolutely. Noise injection helps the RNN learn better.



Hold On... How Do I Inject The Noise?

$\rightarrow$ Simple. Just follow this generative process:

$$\epsilon_t \sim \varphi(\cdot; \mu, \gamma);\ \mathbf{z}_t = g_\Psi(\mathbf{x}_t, \mathbf{z}_{t-1}, \epsilon_t);\ \text{and } \mathbf{o}_t = V^\top \mathbf{z}_t$$

$\rightarrow$ Make sure you choose g_Ψ such that $\mathbf{z}_t$ is *unbiased*,

$$\mathbb{E}_{p(\mathbf{z}_t(\epsilon_{1:t}) \mid \mathbf{z}_{t-1})}\left[\mathbf{z}_t(\epsilon_{1:t})\right] = f_\Psi(\mathbf{x}_t, \mathbf{z}_{t-1}) \ (\textit{weak unbiasedness})$$
$$\mathbb{E}_{p(\mathbf{z}_t(\epsilon_{1:t}) \mid \mathbf{z}_{t-1})}\left[\mathbf{z}_t(\epsilon_{1:t})\right] = \mathbf{s}_t \ (\textit{strong unbiasedness})$$

$\rightarrow$ This ensures that the underlying RNN is preserved.

$\rightarrow$ For example you can use:

$$\mathbf{z}_t = f_\Psi(\mathbf{x}_t, \mathbf{z}_{t-1}) \odot \epsilon_t$$

$\rightarrow$ Dropout is also noise injection. However, Dropout is *biased*.

Ok. Tell Me More...

→ Alright. This procedure is called NOISIN.

→ Uses backpropagation through time on a lower bound to the log marginal likelihood of the data

$$\mathcal{L} = \sum_{t=1}^{T} E_{p(\epsilon_{1:t})}\Big[\log p(\mathbf{x}_{t+1}|\mathbf{z}_t(\epsilon_{1:t}))\Big]$$

→ This averages the predictions of infinitely many RNNs
(a.k.a ensemble method)

→ It also has ties to Empirical Bayes.

→ Using NOISIN is as easy as fitting the original RNN.

→ You can use any noise distribution as long as you scale it to have unbounded variance.

Ok Well... Any Results?

→ On language modeling benchmarks, NOISIN improves over Dropout by as much as 12.2% on the Penn Treebank and 9.4% on the Wikitext-2 dataset.

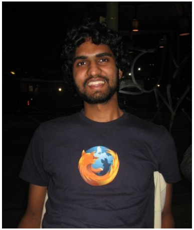
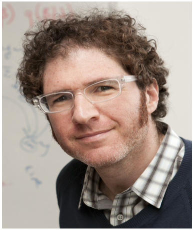
→ See below for the Penn Treebank.

	Medium			Large		
Method	γ	Dev	Test	γ	Dev	Test
None	$--$	115	109	$--$	123	123
Gaussian	1.10	76.2	71.8	1.37	73.2	69.1
Gamma	1.06	78.2	74.5	1.39	73.6	69.5
Bernoulli	0.41	75.7	71.4	0.33	72.8	68.3
Beta	1.07	76.0	71.4	1.50	74.4	70.2

	Medium			Large		
Method	γ	Dev	Test	γ	Dev	Test
Dropout (D)	$--$	80.2	77.0	$--$	78.6	75.3
D + Gaussian	0.53	73.4	70.4	0.92	70.0	66.1
D + Gamma	0.38	73.5	70.3	0.92	71.1	68.2
D + Bernoulli	0.80	73.3	70.1	0.50	70.0	66.1
D + Beta	0.20	73.0	69.2	0.70	70.0	66.2

RNNs + NOISIN Are Awesome!

Collaborators

+ Rajesh Ranganath
+ Jaan Altosaar
+ David Blei