Kernel-Based Approaches for Sequence Modeling: Connections to Neural Methods

Kevin J Liang*, Guoyin Wang*, Yitong Li, Ricardo Henao, Lawrence Carin

{kevin.liang, guoyin.wang, yitong.li, ricardo.henao, lcarin}@duke.edu

Duke University

Introduction

- There have been recent efforts connecting neural methods with kernel machines.
  - Feature mapping $x \to \phi_\theta(x)$ with associated kernel $k_\theta(x, x') = \phi_\theta(x)^T \phi_\theta(x')$
- Viewing neural networks from a kernel machines perspective provides theoretical underpinnings, explaining why they tend to work well (*e.g.* invariance, stability).
- We extend this analysis to *recurrent* neural networks (RNNs), showing how certain natural assumptions lead to popular neural models (or close variants), such as the LSTM, CNN, GCNN, and RAN.
- Experiments on document classification, natural language modeling, and local field potential analysis demonstrate that the models derived from kernel methods perform on par with or slightly better than traditional neural methods.

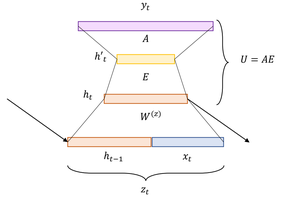
Recurrent Kernel Networks

- Recurrent **Neural** Network

$$h_t = f(W^{(x)} x_t + W^{(h)} h_{t-1} + b)$$
$$y_t = U h_t$$

or[1]

$$z_t = [x_t, h_{t-1}]$$
$$h_t = f(W^{(z)} z_t + b)$$
$$h'_t = E h_t$$
$$y_t = A h'_t$$



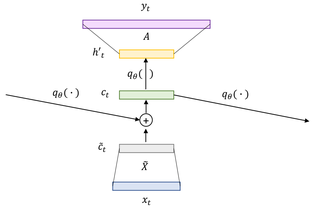
[1]Factorizing $U = AE$:

- Recurrent **Kernel** Network

$$h_t = f(W^{(z)} z_t + b)$$
$$e_i = f\big(W^{(z)} \tilde{z}_i + b\big)$$
$$h'_t = e_i^T h_t = k_\theta(\tilde{z}_i, z_t)$$
$$y_t = A h'_t$$

or[2]

$$\tilde{c}_t = \tilde{X} x_t$$
$$c_t = \tilde{c}_t + q_\theta(c_{t-1})$$
$$h'_t = q_\theta(c_t)$$
$$y_t = A h'_t$$



[2]Noting $k_\theta(\tilde{z}_i, z_t) = q_\theta(\tilde{x}_i^T x_t + q_\theta(\tilde{x}_i^T x_{t-1} + \ldots))$:

Linear Kernel and Dynamic Gates

- Assuming a linear kernel and injecting additional feedback:

$$h'_t = c_t, \qquad c_t = \sigma_i^2 \tilde{c}_t + \sigma_f^2 c_{t-1}, \qquad \tilde{c}_t = \tilde{X} x_t + \tilde{H} h'_{t-1}$$

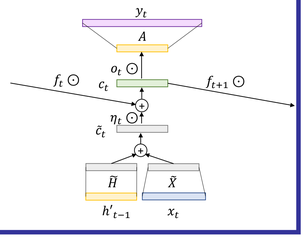
where σ_i^2 and σ_f^2 are static gating elements.

- Replacing σ_i^2 and σ_f^2 with dynamic gates and introducing an output gate:

$$h'_t = o_t \odot c_t, \qquad c_t = \eta_t \odot \tilde{c}_t + f_t \odot c_{t-1}, \qquad \tilde{c}_t = W_c z'_t$$
$$o_t = \sigma(W_o z'_t + b_o), \qquad \eta_t = \sigma\big(W_\eta z'_t + b_\eta\big), \qquad f_t = \sigma\big(W_f z'_t + b_f\big)$$

where $z'_t = [x_t, h'_{t-1}]$.

- As a result of these assumptions, we have thus derived a model closely related to Long Short-Term Memory (LSTM), from kernel machines.



Generalization to n-grams

- Consider the generalization:

$$h_t = f(W^{(x_0)} x_t + \cdots W^{(x_{-n+1})} x_{t-n+1} + W^{(h)} h_{t-1} + b)$$

- As before, we may express:

$$e_i = f(W^{(x_0)} \tilde{x}_{i,0} + \cdots W^{(x_{-n+1})} \tilde{x}_{i,-n+1} + W^{(h)} \tilde{h}_i + b)$$

- We thus have an n-gram generalization of the LSTM:

$$h'_t = o_t \odot c_t, \qquad c_t = \eta_t \odot \tilde{c}_t + f_t \odot c_{t-1}, \qquad \tilde{c}_t = \widetilde{\boldsymbol{X}} \cdot X_t + \widetilde{H} h'_{t-1}$$
$$o_t = \sigma\big(\widetilde{\boldsymbol{X}}_o \cdot X_t + \widetilde{W}_o h'_{t-1} + b_o\big), \eta_t = \sigma\big(\widetilde{\boldsymbol{X}}_\eta \cdot X_t + \widetilde{W}_\eta h'_{t-1} + b_\eta\big), f_t = \sigma\big(\widetilde{\boldsymbol{X}}_f \cdot X_t + \widetilde{W}_f h'_{t-1} + b_f\big)$$

- Assuming $f_t = 0$ and constant values for o_t and η_t, the n-gram LSTM reduces to a CNN.

Experiments

Model	Parameters	Input	Cell	Output
LSTM	$(nm+d)(4d)$	$z'_t = [x_t, h'_{t-1}]$	$c_t = \eta_t \odot \tanh(\tilde{c}_t) + f_t \odot c_{t-1}$	$h'_t = o_t \odot \tanh(c_t)$
RKM-LSTM	$(nm+d)(4d)$	$z'_t = [x_t, h'_{t-1}]$	$c_t = \eta_t \odot \tilde{c}_t + f_t \odot c_{t-1}$	$h'_t = o_t \odot c_t$
RKM-CIFG	$(nm+d)(3d)$	$z'_t = [x_t, h'_{t-1}]$	$c_t = (1 - f_t) \odot \tilde{c}_t + f_t \odot c_{t-1}$	$h'_t = o_t \odot c_t$
Linear Kernel w/ o_t	$(nm+d)(2d)$	$z'_t = [x_t, h'_{t-1}]$	$c_t = \sigma_i^2 \tilde{c}_t + \sigma_f^2 c_{t-1}$	$h'_t = o_t \odot c_t$
Linear Kernel	$(nm+d)(d)$	$z'_t = [x_t, h'_{t-1}]$	$c_t = \sigma_i^2 \tilde{c}_t + \sigma_f^2 c_{t-1}$	$h'_t = \tanh(c_t)$
Gated CNN	$(nm)(2d)$	$z'_t = x_t$	$c_t = \sigma_i^2 \tilde{c}_t$	$h'_t = o_t \odot c_t$
CNN	$(nm)(d)$	$z'_t = x_t$	$c_t = \sigma_i^2 \tilde{c}_t$	$h'_t = \tanh(c_t)$

Table 1: Model variant comparison, for 1-gram inputs.

Model	Parameters		AGNews		DBpedia		Yahoo!		Yelp Full	
	1-gram	3-gram	1-gram	3-gram	1-gram	3-gram	1-gram	3-gram	1-gram	3-gram
LSTM	720K	1.44M	91.82	**92.46**	98.98	98.97	**77.74**	77.72	**66.27**	66.37
RKM-LSTM	720K	1.44M	91.76	92.28	98.97	99.00	77.70	77.72	65.92	**66.43**
RKM-CIFG	540K	1.08M	**92.29**	92.39	**98.99**	**99.05**	77.71	**77.91**	65.93	65.92
Linear Kernel w/ o_t	360K	720K	92.07	91.49	98.96	98.94	77.41	77.53	65.35	65.94
Linear Kernel	180K	360K	91.62	91.50	98.65	98.77	76.93	76.53	61.18	62.11
Gated CNN	180K	540K	91.54	91.78	98.37	98.77	72.92	76.66	60.25	64.30
CNN	90K	270K	91.20	91.53	98.17	98.52	72.51	75.97	59.77	62.08

Table 2: Document classification accuracy, for 1-gram and 3-gram models.

Model	PTB		Wikitext-2	
	PPL valid	PPL test	PPL valid	PPL test
LSTM	61.2	58.9	68.74	65.68
RKM-LSTM	**60.3**	**58.2**	**67.85**	**65.22**
RKM-CIFG	61.9	59.5	69.12	66.03
Linear Kernel w/ o_t	72.3	69.7	84.23	80.21

Table 3: Language model perplexity (PPL) on validation and test sets of Penn Treebank and Wikitext-2.

Model	n-gram LSTM	RKM-LSTM	RKM-CIFG	Linear Kernel w/ o_t	Linear Kernel	Gated CNN	CNN
Accuracy	**80.24**	79.02	77.58	76.11	73.13	76.02	73.40

Table 4: Mean leave-one-out classification accuracies for mouse LFP data.

Implementations can be found at https://github.com/kevinjliang/kernels2rnns