Salka

A Deep Learning Toolkit for Unsupervised Anomaly Detection in Computer Networks

Elliott Skomski, Yusheng Jiang, and Ashima Shrivastava
{skomsks,jiangy2,shrivaa}@wwu.edu

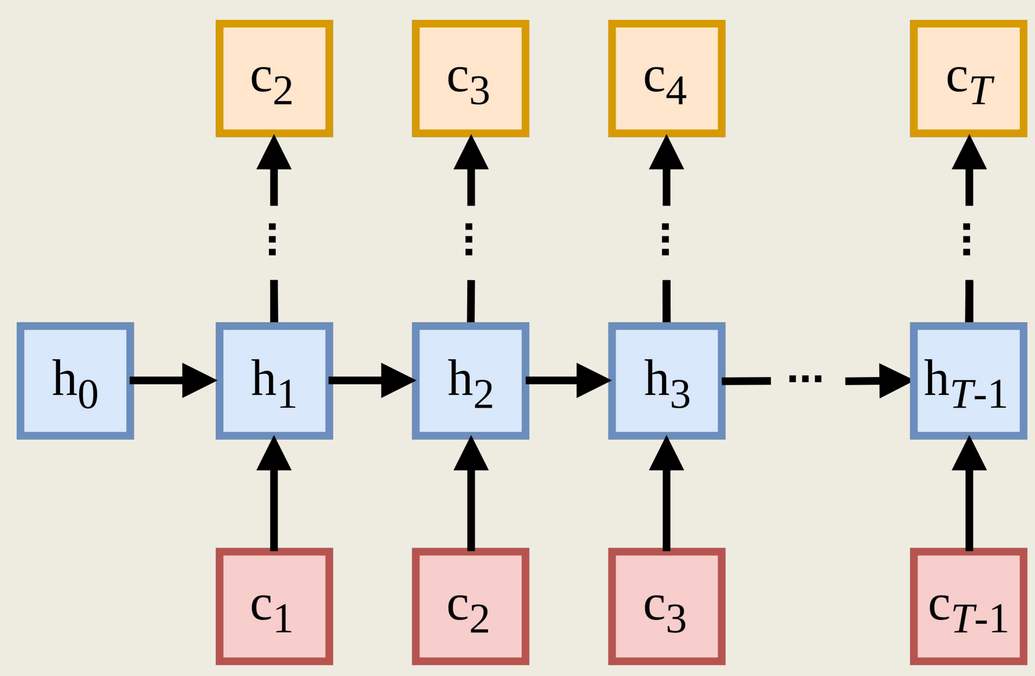
Advised by Dr. Michael Tsikerdekis for CSCI 597Q, Spring 2019

Introduction

Motivation: Computer network logs are large, difficult to comb through to detect anomalous, potentially malicious activity.

Goal: Reduce the burden incurred by security analysts by developing an easy–to–use, free, and open–source anomaly detection system to identify suspicious activity and filter these events for analysts.

Approach: Apply recent innovations in unsupervised deep learning to develop effective anomaly detection models.



Recurrent neural network language model.
Based on event model proposed in [1]. Model uses recurrent connections between hidden representations of tokens to infer the next token at each position in an event log line. Optionally supports bidirectional connections and residual connections between recurrent layers.

Methods

- Based on [1], we make use of deep learning to train *unsupervised language models* for anomaly detection.
  - Idea: learn probability of sentence as product of probabilities of each word given all previous words:

$$P(W) = \prod_{t=1}^{T} P(c_t \mid c_1, c_2, \dots, c_{t-1})$$

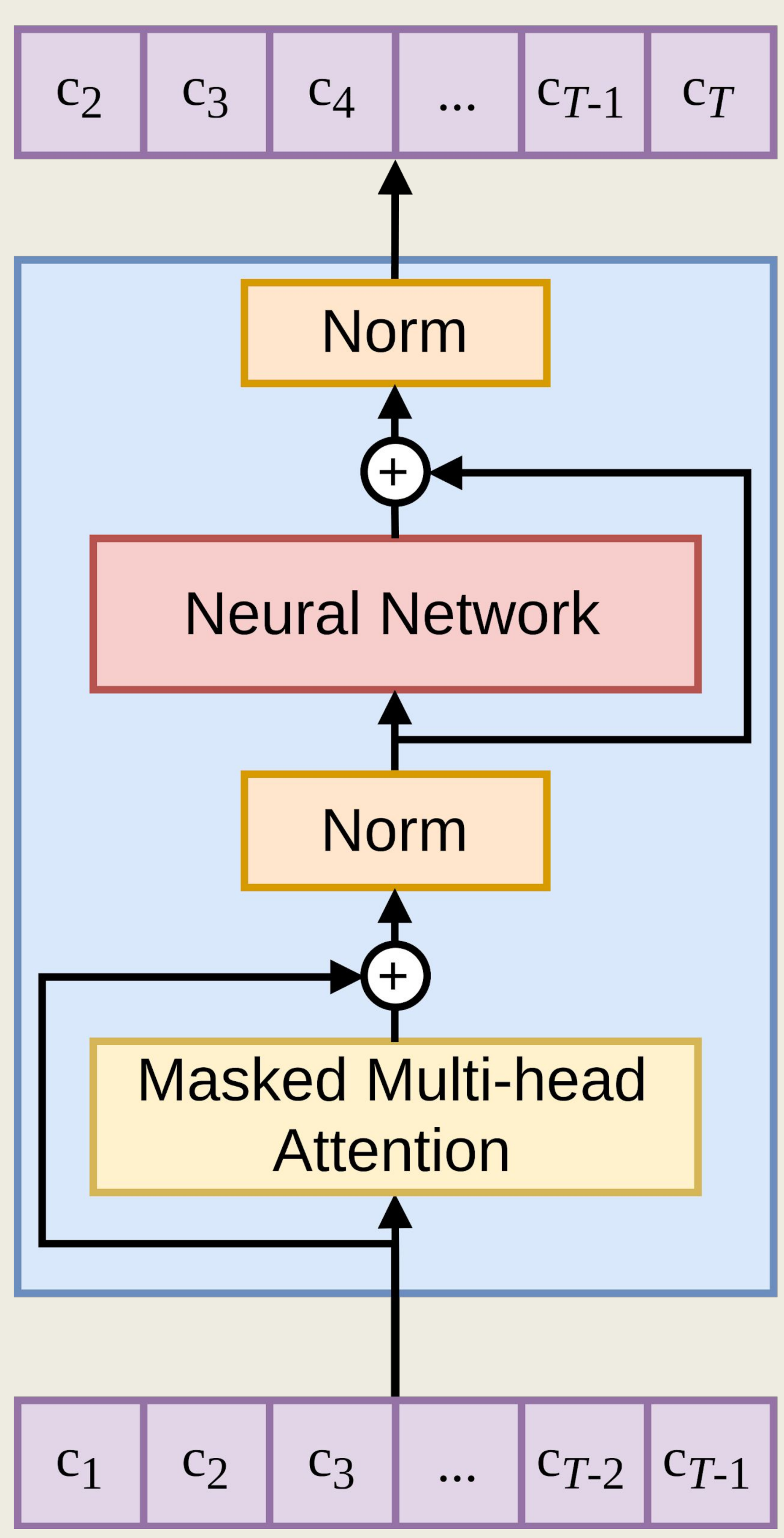
  - Models are trained to minimize cross–entropy between predicted and expected words:

$$H(c_t, \hat{c}_t) = -\sum_i c_{t,i} \log \hat{c}_{t,i}$$

  - If model fails to reconstruct event tokens incrementally, event incurs high loss.
  - This loss can be used to score and rank events by their anomalousness.
  - Models are trained incrementally and asynchronously:

```
while w < number of time windows in log:
   train on data from window w
   evaluate on data from window w + 1
   w = w + 1
```

- We further extend this framework with recent innovations in deep learning and language modeling.



Transformer decoder language model.
Based on model proposed in [2], extended by [3, 4]. Model uses masked self–attention heads and a position–wise neural network to infer tokens in an event log line. Includes residual connections and layer normalization for increased training stability. Figure derived from [3].

References

[1] Tuor, Aaron Randall, Ryan Baerwolf, Nicolas Knowles, Brian Hutchinson, Nicole Nichols, and Robert Jasper. "Recurrent Neural Network Language Models for Open Vocabulary Event–Level Cyber Anomaly Detection." *In Workshops at the Thirty–Second AAAI Conference on Artificial Intelligence*. 2018.
[2] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *In Advances in Neural Information Processing Systems*, pp. 5998–6008. 2017.
[3] Radford, Alec, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. "Improving language understanding by generative pre–training." URL https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf (2018).
[4] Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. "Language models are unsupervised multitask learners." URL https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf (2019).
[5] Paszke, Adam, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. "Automatic differentiation in PyTorch." (2017).
[6] Sennrich, Rico, Barry Haddow, and Alexandra Birch. "Neural machine translation of rare words with subword units." *arXiv preprint arXiv:1508.07909* (2015).
[7] Brown, Andy, Aaron Tuor, Brian Hutchinson, and Nicole Nichols. "Recurrent neural network attention mechanisms for interpretable system log anomaly detection." *arXiv preprint arXiv:1803.04967* (2018).

Features

Models

- Recurrent neural network:
  - As in [1], model uses recurrent connections across time steps.
  - Supports different types of RNN cells and bidirectional recurrent connections.
- Transformer decoder:
  - Non–recurrent, attention–based architecture based on [2].
  - OpenAI's GPT/GPT–2 [3,4] models have demonstrated excellent language modeling capabilities.
- Both models support weight tying of token embeddings and residual connections.

All models are developed using PyTorch [5].

Data Processing

- Support for character– and word–level log line tokenization, as well as byte–pair encoding [6].
- Parses and featurizes CSV logs on–the–fly.
- Can buffer large CSV logs by arbitrary time windows for asynchronous training.
- Easily configured using a YAML file.

Analysis

- Event scores are GZIP–compressed for easy storage and transfer.
- Toolkit will soon include analysis tools for ranking and flagging events based on user and network score statistics.

This toolkit is open source and available for free at `github.com/eskomski/salka`

Future Work

- Extend recurrent model to include attention mechanisms as in [7].
- Evaluate models on different datasets.
- Incorporate user and network metadata into model training data.
- Add simple analysis tools.
- Include more model variants.

Acknowledgements

We would like to thank Aaron Tuor, Ryan Baerwolf, Nicolas Knowles, Andy Brown, Samuel Kaplan, Robin Cosbey, Josh Loehr, Brian Barragan–Cruz, Brian Hutchinson, Nicole Nichols, Robert Jasper, and Sean Robinson for their contributions to Safekit and the AIMSAFE project which inspired and informed this work.