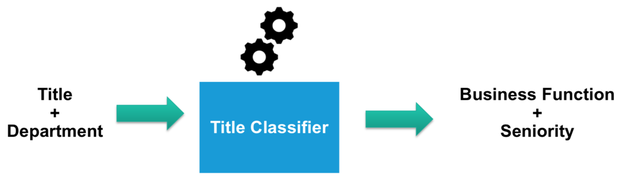
Text Classification of Active Directory Data with Long Short-term Memory Networks

Cole Juracek
Duke University, Statistical Science

Title Classification for Cybersecurity

People: Brian Jones (Manager), Zachary Abzug (Mentor), Aaron Liu (Co-intern)

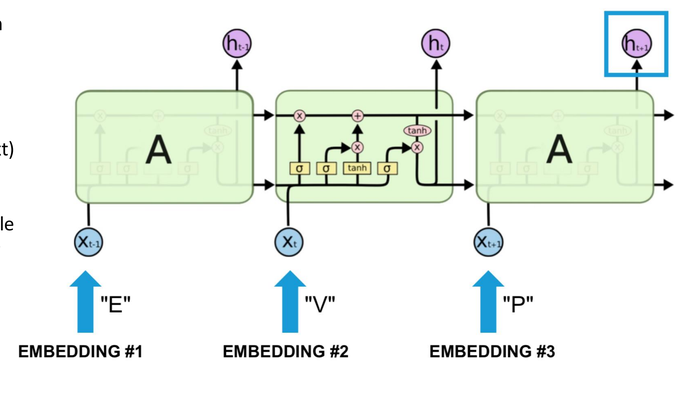
Introduction



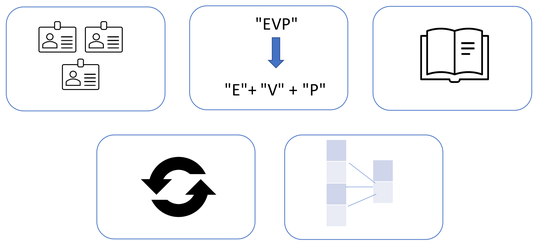
- Proofpoint is a cybersecurity company providing software as a service
- Clients interested in how their people are being attacked. Insight on *groups* of employees more actionable than at the individual level
- The **Title Classifier** seeks to classify an employee's title and department into a normalized business function and seniority

Methods

- *Problem:* Supervised multi-class classification problem in natural language processing
- *Solution:* **Long short-term memory (LSTM)** neural network (right)
  - Extension of recurrent neural network
  - Designed for sequential / variable length input (text)
  - Address vanishing gradient found in vanilla recurrent neural networks
- Cell state (top line) represents long-term memory while the hidden state (bottom line) interacts with input for updates
- Final hidden state used for classification



High-level flowchart of classifier



1. Batch of title-department pairs obtained from the training data
2. Tokenization of input at character-level
   - Word and subword level also considered
3. Tokens converted into real-valued vectors (embeddings) to be processed by network
4. Embeddings sequentially fed into LSTM layer, updating hidden and cell states at each step
5. Final set of dense layers used to bring output to appropriate dimensionality

- Steps for a single weight update:

1. Perform a "forward pass" of the batch through the network with current weights
2. Calculate loss using known labels
3. Use loss to calculate an estimate of the loss gradient with respect to weights via *backpropagation*
4. Update weights using gradient and Adam optimizer

- **Cross-entropy** loss function used:

$$\text{loss}(x, class) = -\log\left(\frac{\exp(x[class])}{\sum_j \exp(x[j])}\right) = -x[class] + \log\left(\sum_j \exp(x[j])\right)$$

- $X \in \mathbb{R}^C$ is the unnormalized output of the final dense layer
- Convert to distribution via inner softmax. Perfect classification places all probability mass on the observed class label

Application

Generic Business Function	
Administration	Product Dev / Services
Bizdev / Strategic	Supply Chain
Finance	Purchasing
HR	Sales
IT	Facilities
Legal	Customer Service
Marketing	Unsure

Seniority
Executive
VP
Manager / Director
Employee
Part-time Employee
Unsure

- 3463 hand-labeled examples
- Problems with Data
  - Missing departments (13%)
  - Foreign languages
  - Acronyms
  - Cross-functional mapping
- Trained over 5 epochs until validation loss stopped decreasing

References: [1] Porteous, D. Newman, A. Ihler, A. Asuncion, P. Smyth, and M. Welling. Fast collapsed gibbs sampling for latent dirichlet allocation. In SIGKDD, pages 569–577. ACM, 2008.
[2] Christopher Olah's blog: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

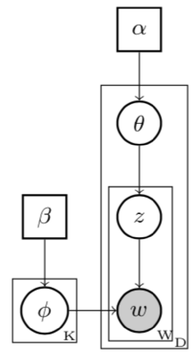
Latent Dirichlet Allocation with Collapsed Gibbs Sampling (Porteous et al., 2008[1])

Collaborators: Pierre Gardan

Introduction

- Latent Dirichlet Allocation (LDA) is a generative, probabilistic model for discrete data (text). We assume that words are drawn according to document specific mixture distributions of topics. The mixture distributions of words per topic and topic per document are described via Dirichlet distributions
- Primary use case explored is corpus summarization: how can we describe the underlying topics in a collection of documents?
- The exact posterior is difficult to obtain samples from. We implement a collapsed Gibbs sampler to obtain a chain of latent topics, then use these to obtain estimates of the remaining parameters

Methods



Notation	Interpretation
Alpha	Controls sparsity of topic distributions in documents
Beta	Controls sparsity of word distributions in topics
Phi	Word mixtures for a given topic
Theta	Topic mixtures for a given document
z	Latent topic for a given word
w	Observed word

- A topic is simply a distribution over words
- Model each document as a mixture over K topics
- Model contents of document as multinomial over W word vocabulary

Generative process for word w_{ij}

1. Draw document-specific topic mixtures $\theta_j \sim Dirichlet(\alpha)$
2. Draw topic-specific word mixtures $\phi_k \sim Dirichlet(\beta)$
3. Draw latent topic $z_{ij} \sim Categorical(\theta_j)$
4. Draw word given topic as $w_{ij} \sim Categorical(\phi_{z_{ij}})$

- Marginal joint distribution of **z, w** obtained by integrating theta and phi over full joint distribution:

$$p(\mathbf{z}, \mathbf{w}) = \int_{\boldsymbol{\theta}} \int_{\boldsymbol{\phi}} \prod_{k=1}^{K} p(\phi_k|\beta) \prod_{j=1}^{D} p(\theta_j|\alpha) \prod_{i=1}^{n_j} p(z_{ij}|\theta_j) p(w_{ij}|\phi_{z_{ij}}) d\boldsymbol{\phi} d\boldsymbol{\theta}$$

Algorithm 1 Collapsed Gibbs Sampler

```
z_ij = [] for all i, j
for iter = 1 to niter do
    for j = 1 to D do
        for i = 1 to n_j do
            Densities = []
            for k = 1 to K do
                Append p(z_ij = k) to densities
            end for
            Normalize densities by densities/ = sum(densities)
            New topic = weighted sample from densities
            Append new topic to z_ij
        end for
    end for
end for
z_ij = Mode(z_ij) for all i, j
```

- Use to construct *collapsed* Gibbs sampler on topic values alone
- N_{wkj}: Number of times word w is assigned to topic k in document j

$$p(z_{ij} = k|\mathbf{z}^{\neg ij}, \mathbf{w}, \alpha, \beta) = \frac{1}{Z} a_{kj} b_{wk}$$

$$a_{kj} = N_{kj}^{\neg ij} + \alpha \quad b_{wk} = \frac{N_{wk}^{\neg ij} + \beta}{N_k^{\neg ij} + W\beta}$$

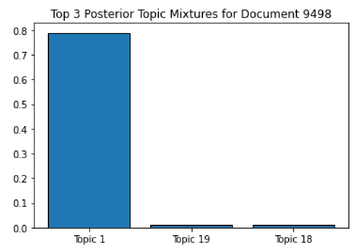
- Recover posterior estimates of other parameters via:

$$\hat{\theta}_{kj} = \frac{N_{kj} + \alpha}{N_j + K\alpha} \quad \hat{\phi}_{wk} = \frac{N_{wk} + \beta}{N_k + W\beta}$$

Application

- Pre-processing
  - Tokenization
  - Lowercasing
  - Removal of stop words
  - Lemmatization
- Datasets
  - 20 Newsgroups
    - D = 18000 documents
    - K = 20 topics
  - Reuters-21578 (subset)
    - D = 1000 documents
    - K = 5 topics

Interpreted Topic	Matched Topic	Top 5 Words by Posterior Probability
Computers / Cryptography	Sci.crypt	('key', 0.056), ('chip', 0.023), ('encryption', 0.020), ('bit', 0.015), ('system', 0.015)
Sports	Rec.sport	('game', 0.043), ('team', 0.033), ('year', 0.027), ('play', 0.026), ('player', 0.021)
Religion	Soc.religion .christian	('god', 0.0497, ('jesus', 0.0219), ('believe', 0.0212), ('bible', 0.0128), ('life', 0.0124)



Document 9498: *"It depends on the algorithm used. 128-bit secret keys for RSA are definitively not secure enough. Regards, [omitted]"*. We observe a dominant topic 1 (cryptography), and observe sparsity in the other topics