Advancing Data Loss Prevention With Neural Networks: Detecting & Prioritizing Incidents

3rd Annual North Carolina Cybersecurity Symposium

Samuel Cameron

February 22, 2024

The opinions and content presented in this talk are solely my own and do not represent the views or positions of my employer.

Purpose Of This Talk

Apply the CNN to DLP

- Potential DLP gaps
- The theory behind applying
- The method of applying
- What to expect
- Case Study

Examine CNN for text classification

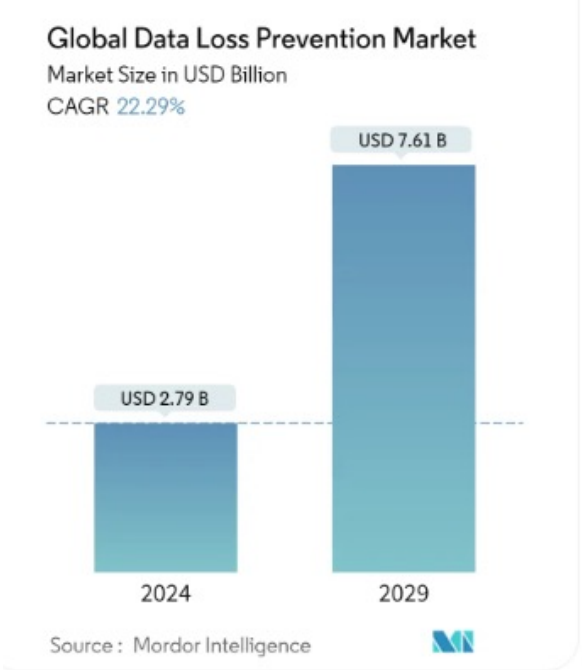
- What is CNN
- How are they used?
- How can they be used?
- Research studies
- Architecture for the model I will present

Tips and tricks for finetuning

- What parameters to tune
- How to get more for less
- How to deploy effectively

Brief discussion on DLP

DLP Market



Study Period	2019 - 2029
Market Size (2024)	USD 2.79 Billion
Market Size (2029)	USD 7.61 Billion
CAGR (2024 - 2029)	22.29 %
Fastest Growing Market	Asia Pacific
Largest Market	North America

Major Players

*Disclaimer: Major Players sorted in no particular order

Mordor Intelligence Research & Advisory. (2023, June). Data Loss Prevention Market Size & Share Analysis - Growth Trends & Forecasts (2024 - 2029). Mordor Intelligence. Retrieved February 20, 2024, from https://www.mordorintelligence.com/industry-reports/data-loss-prevention-market

DLP Limitations *(non-exhaustive list)*

1. Complexity of Implementation
2. False Positives/Negatives
3. Policy Management
4. Performance Impact
5. Advanced Threats
6. User Behavior
7. Integration with Other Systems
8. Maintenance and Upkeep
9. Legal and Privacy Concerns
10. Limited by Scope

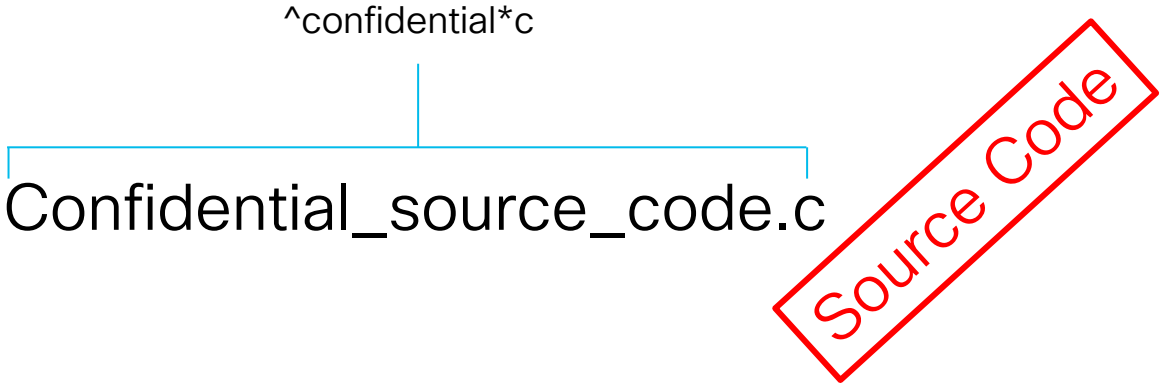
DLP Limitations *(non-exhaustive list)*

1. Complexity of Implementation
2. False Positives/Negatives
3. Policy Management
4. Performance Impact
5. Advanced Threats
6. User Behavior
7. Integration with Other Systems
8. Maintenance and Upkeep
9. Legal and Privacy Concerns
10. Limited by Scope

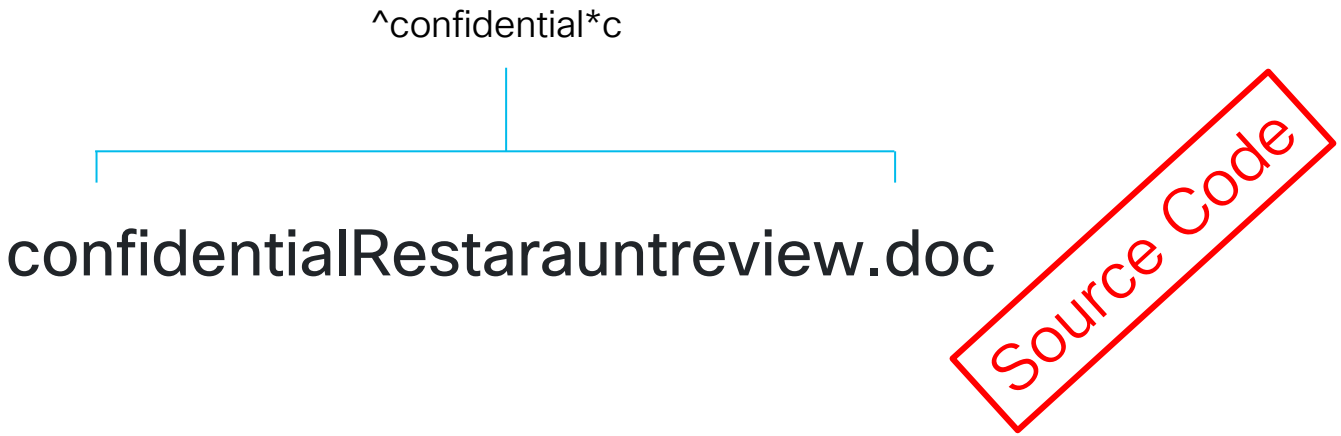
Example Limitation

Example Limitation: Regular Expressions (False Positives)

- File name matches ✓
- Content Inspection ✓



Example Limitation: Regular Expressions (False Positves)



It’s likely that your DLP product won’t function in a vacuum — you’ll probably need other tools, too.

Jenna Phipps

https://www.esecurityplanet.com/products/data-loss-prevention-dlp-solutions/

The Convolutional Neural Network (CNN)

Convolutional Neural Network

Google

Pinterest

NETFLIX

amazon

airbnb

Adobe

Research

https://arxiv.org/abs/2203.05173

TextConvoNet: A Convolutional Neural Network based Architecture for Text Classification

Sanskar Soni, Satyendra Singh Chouhan, Santosh Singh Rathore

***Abstract*—In recent years, deep learning-based models have significantly improved the Natural Language Processing (NLP) tasks. Specifically, the Convolutional Neural Network (CNN), initially used for computer vision, has shown remarkable performance for text data in various NLP problems. Most of the existing CNN-based models use 1-dimensional convolving filters (*n-gram* detectors), where each filter specialises in extracting *n-grams* features of a particular input word embedding. The input word embeddings, also called sentence matrix, is treated as a matrix where each row is a word vector. Thus, it allows the model to apply one-dimensional convolution and only extract *n-gram* based features from a sentence matrix. These features can be termed as *intra-sentence n-gram features*. To the extent of our knowledge, all the existing CNN models are based on the aforementioned concept. In this paper, we present a CNN-based architecture *TextConvoNet* that not only extracts the intra-sentence n-gram features but also captures the inter-sentence n-gram features in input text data. It uses an alternative approach for input matrix representation and applies a two-dimensional multi-scale convolutional operation on the input. To evaluate the performance of *TextConvoNet*, we perform an experimental study on five text classification datasets. The results are evaluated by using various performance metrics. The experimental results show that the presented *TextConvoNet* outperforms state-of-the-art machine learning and deep learning models for text classification purposes.**

***Index Terms*—Text classification, CNN, Multi-dimensional Convolution, Deep learning.**

I. Introduction

Natural language processing (NLP) involves computational processing and understanding of the natural/human languages. It involves various tasks that rely on various statistics and data-driven computation techniques [1]. One of the important tasks in NLP is text classification. It is a classical problem where the prime objective is to classify (assign labels or tags) to the textual contents [2]. Textual contents can either be sentences, paragraphs, or queries [2], [3]. There are many real-world applications of text classification such as sentiment analysis [4], news classification [5], intent classification [6], categories: Rule-based, Data-Driven based (Machine Learning/Deep Learning-based approaches), and Hybrid approaches. Rule-based approaches classify text into different categories using a set of pre-defined rules. However, it requires complete domain knowledge [8], [9]. Alternatively, machine learning-based approaches have proven to be significantly effective in recent years. All the machine learning approaches work in two stages: first, they extract some handcrafted features from the text. Next, these features are fed into a machine learning model. For extracting the handcrafted features, a bag of words, *n-grams* based model, term frequency, and inverse document frequency (TF-IDF) and their extensions were popularly used. For the second stage, many classical Machine Learning algorithms such as Support Vector Machine (SVM), Decision Tree (DT), Conditional Probability-based such as Naïve Bayes, and other Ensemble-based approaches are used [10], [11], [12], [13].

Recently, some of the Deep Learning methods, specifically RNN (Recurrent Neural Network) and CNN (Convolutional Neural Network), have shown remarkable results in text classification [14], [15], [16], [17], [18], [19], [20]. CNN-based models are trained to recognize patterns in text, such as key phrases. Most CNN-based models utilize one-dimensional (1-D) convolution followed by a one-dimensional max-pooling operation to extract a feature vector from the input word embeddings. This feature vector is fed into the classification layer as an input for classification purposes. Input word embedding is a word matrix where each row represents a word vector. Therefore, one-dimensional (1-D) convolution extracts *n-gram* based features by performing convolution operation on two or more than two-word vectors at a time.

However, improving text classification results by utilizing the *n-gram* features in between different sentences using convolution operation still remains an open research question for all the researchers. Furthermore, the input matrix structure also remains a point to ponder, which could be revamped to apply multidimensional convolution. This paper presents,

Research

https://arxiv.org/pdf/1509.01626.pdf

Character-level Convolutional Networks for Text Classification*

Xiang Zhang **Junbo Zhao** **Yann LeCun**
Courant Institute of Mathematical Sciences, New York University
719 Broadway, 12th Floor, New York, NY 10003
{xiang, junbo.zhao, yann}@cs.nyu.edu

Abstract

This article offers an empirical exploration on the use of character-level convolutional networks (ConvNets) for text classification. We constructed several large-scale datasets to show that character-level convolutional networks could achieve state-of-the-art or competitive results. Comparisons are offered against traditional models such as bag of words, n-grams and their TFIDF variants, and deep learning models such as word-based ConvNets and recurrent neural networks.

1 Introduction

Text classification is a classic topic for natural language processing, in which one needs to assign predefined categories to free-text documents. The range of text classification research goes from designing the best features to choosing the best possible machine learning classifiers. To date, almost all techniques of text classification are based on words, in which simple statistics of some ordered word combinations (such as n-grams) usually perform the best [12].

On the other hand, many researchers have found convolutional networks (ConvNets) [17] [18] are useful in extracting information from raw signals, ranging from computer vision applications to speech recognition and others. In particular, time-delay networks used in the early days of deep learning research are essentially convolutional networks that model sequential data [1] [31].

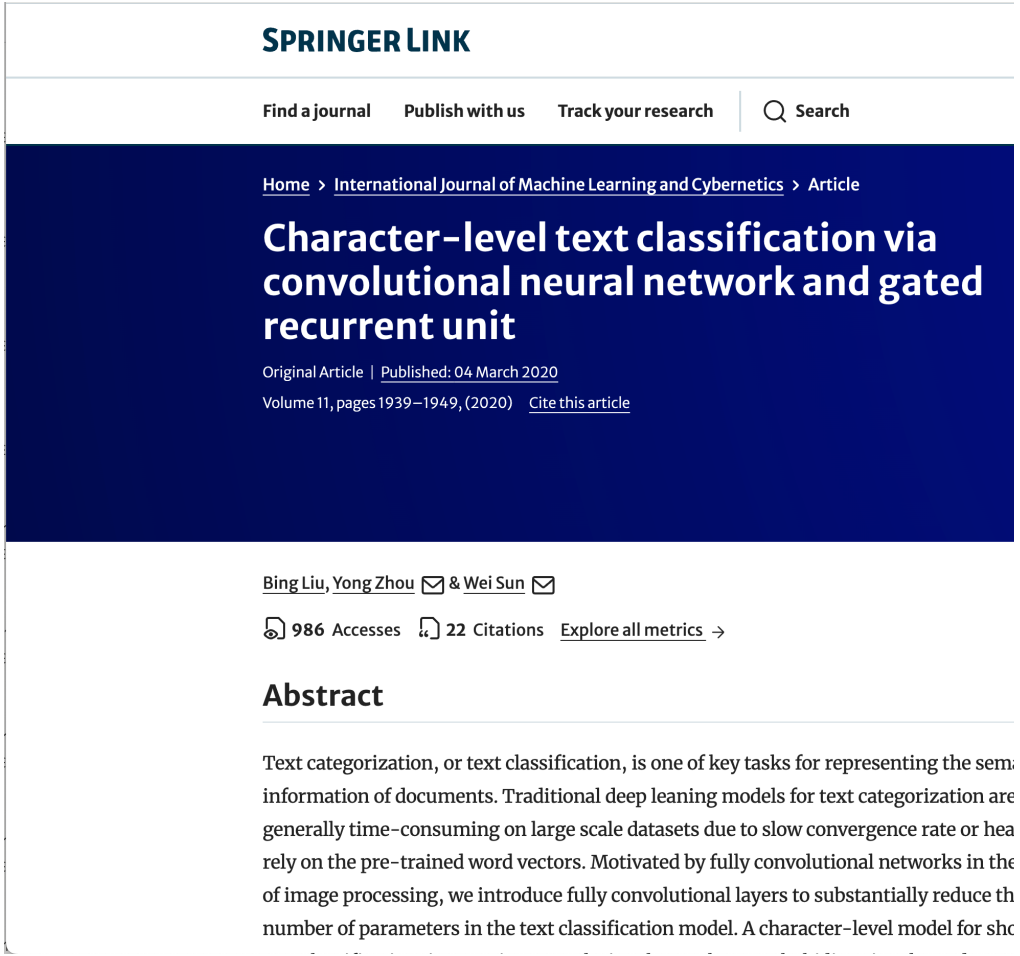
In this article we explore treating text as a kind of raw signal at character level, and applying temporal (one-dimensional) ConvNets to it. For this article we only used a classification task as a way to exemplify ConvNets' ability to understand texts. Historically we know that ConvNets usually require large-scale datasets to work, therefore we also build several of them. An extensive set of comparisons is offered with traditional models and other deep learning models.

Applying convolutional networks to text classification or natural language processing at large was explored in literature. It has been shown that ConvNets can be directly applied to distributed [6] [16] or discrete [13] embedding of words, without any knowledge on the syntactic or semantic structures of a language. These approaches have been proven to be competitive to traditional models.

There are also related works that use character-level features for language processing. These in-

Research Cont...

https://link.springer.com/article/10.1007/s13042-020-01084-9



SPRINGER LINK

Find a journal | Publish with us | Track your research | Search

Home > International Journal of Machine Learning and Cybernetics > Article

Character-level text classification via convolutional neural network and gated recurrent unit

Original Article | Published: 04 March 2020

Volume 11, pages 1939–1949, (2020) Cite this article

Bing Liu, Yong Zhou & Wei Sun

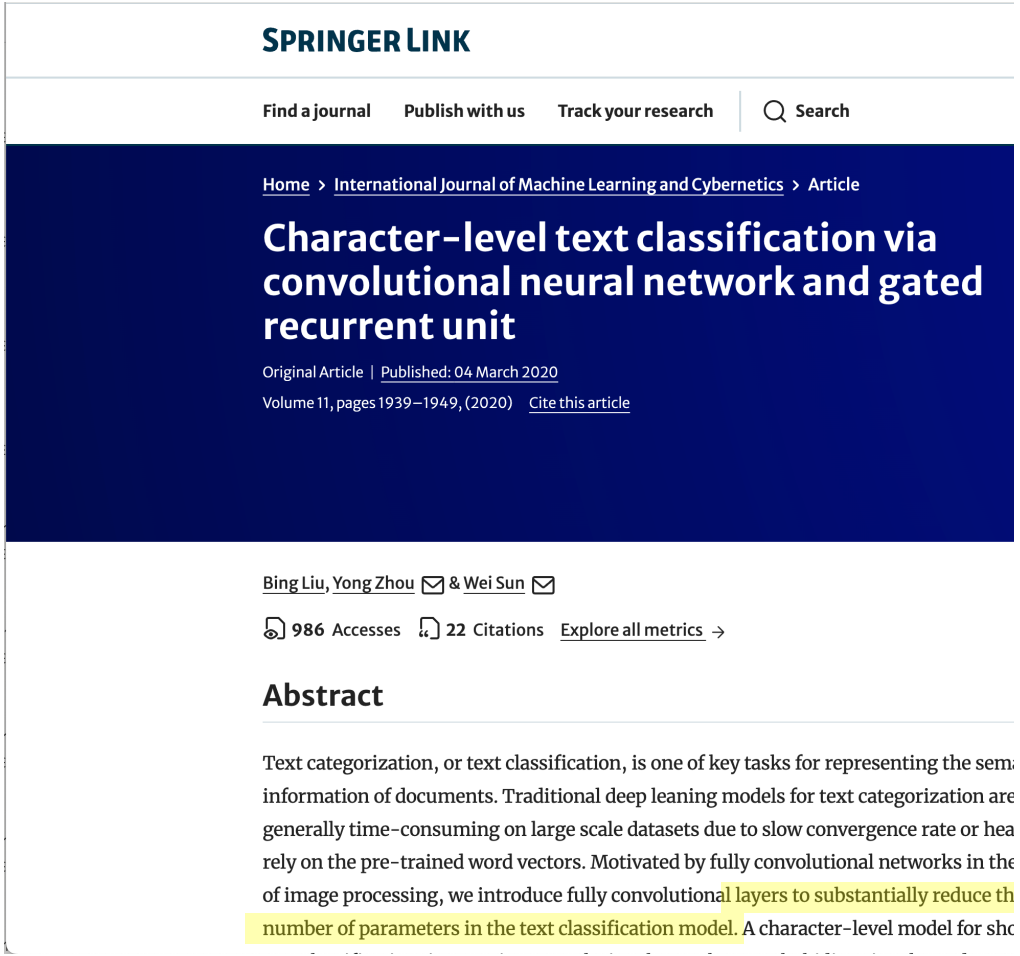
986 Accesses 22 Citations Explore all metrics →

Abstract

Text categorization, or text classification, is one of key tasks for representing the sem
information of documents. Traditional deep leaning models for text categorization are
generally time-consuming on large scale datasets due to slow convergence rate or hea
rely on the pre-trained word vectors. Motivated by fully convolutional networks in the
of image processing, we introduce fully convolutional layers to substantially reduce th
number of parameters in the text classification model. A character-level model for sho

Research Cont...

https://link.springer.com/article/10.1007/s13042-020-01084-9



SPRINGER LINK

Find a journal | Publish with us | Track your research | Search

Home > International Journal of Machine Learning and Cybernetics > Article

Character-level text classification via convolutional neural network and gated recurrent unit

Original Article | Published: 04 March 2020

Volume 11, pages 1939–1949, (2020) Cite this article

Bing Liu, Yong Zhou & Wei Sun

986 Accesses 22 Citations Explore all metrics →

Abstract

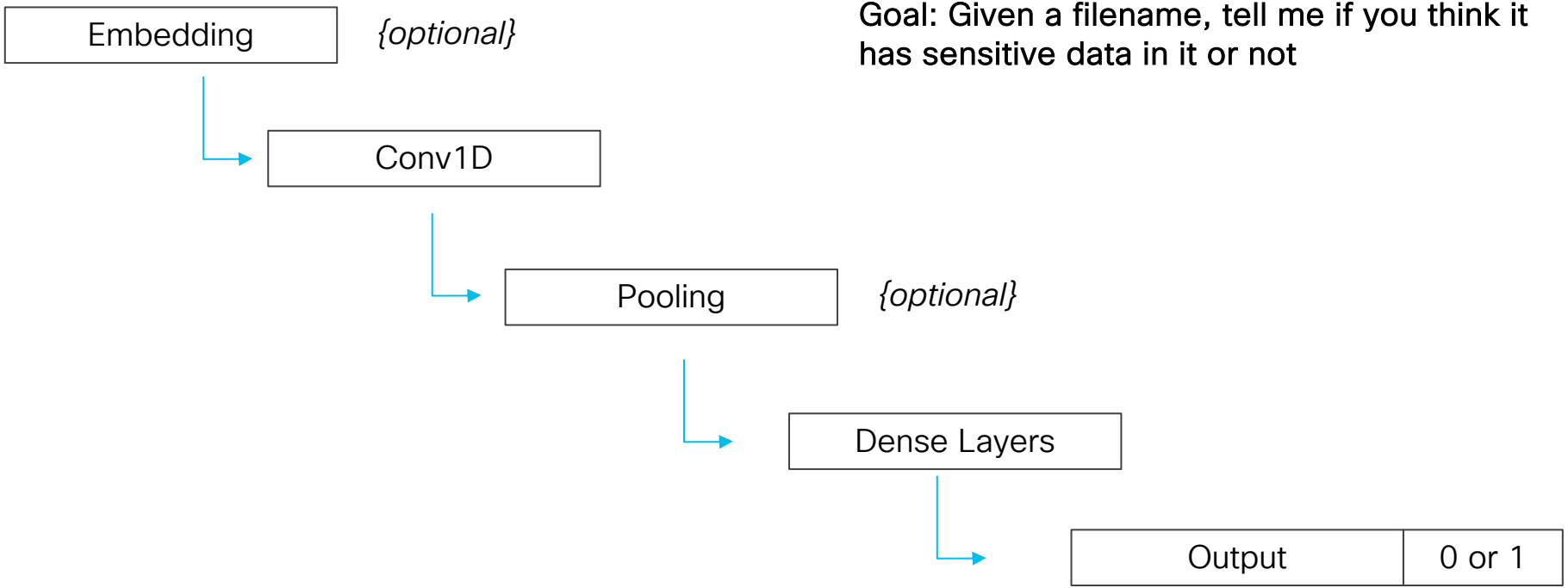
Text categorization, or text classification, is one of key tasks for representing the sem
information of documents. Traditional deep leaning models for text categorization are
generally time-consuming on large scale datasets due to slow convergence rate or hea
rely on the pre-trained word vectors. Motivated by fully convolutional networks in the
of image processing, we introduce fully convolutional layers to substantially reduce th
number of parameters in the text classification model. A character-level model for sho

Assumptions

- You have some sense of what your sensitive data is or where its at
- Logging, or some stream of data is available
- You have training data or can make it

Architecture for our model

The CNN - Our Architecture



Goal: Given a filename, tell me if you think it has sensitive data in it or not

```
model = Sequential()
model.add(Embedding(alphabet_size + 1, embed_dim, input_length=max_len))
model.add(Conv1D(num_filters, kernel_size, activation='relu'))
model.add(MaxPooling1D(pool_size=pool_size))
model.add(Dropout(0.2))  # add dropout
model.add(Conv1D(num_filters, kernel_size, activation='relu'))
model.add(MaxPooling1D(pool_size=pool_size))
model.add(Conv1D(num_filters, kernel_size, activation='relu'))
model.add(GlobalMaxPooling1D())
model.add(Dense(hidden_dim1, activation='relu'))
model.add(Dropout(0.2))  # add dropout
model.add(Dense(hidden_dim2, activation='relu'))
model.add(Dense( units: 1, activation='sigmoid'))
```

https://github.com/phantomOPBro/3rd-Annual-North-Carolina-Cybersecurity-Symposium.git

Characteristics of the CNN

- Process input like a 1D picture
- Kernel – how many characters to look at at a time
- Filters – How many patterns to detect
- Generally, less parameters than MOST other text classification models
- Fast training time
- Processing time and power are minimal

Example Flow Through the CNN

confidential_source_code.c

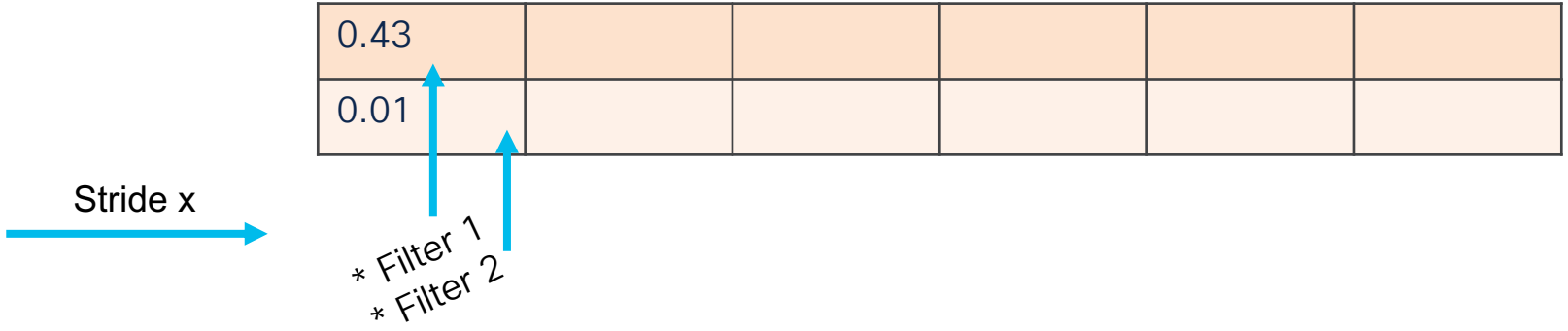
Assuming your tokenization alphabet looks like this:

alphabet = "abcdefghijklmnopqrstuvwxyz0123456789-,;.!?:'\"/\\|_@#$%^&*~`+-=<>()[]{}"

[2, 14, 13, 5, 8, 3, 4, 13, 19, 8, 0, 11, 37, 18, 14, 20, 17, 2, 4, 37, 2, 14, 3, 4, 36, 2]

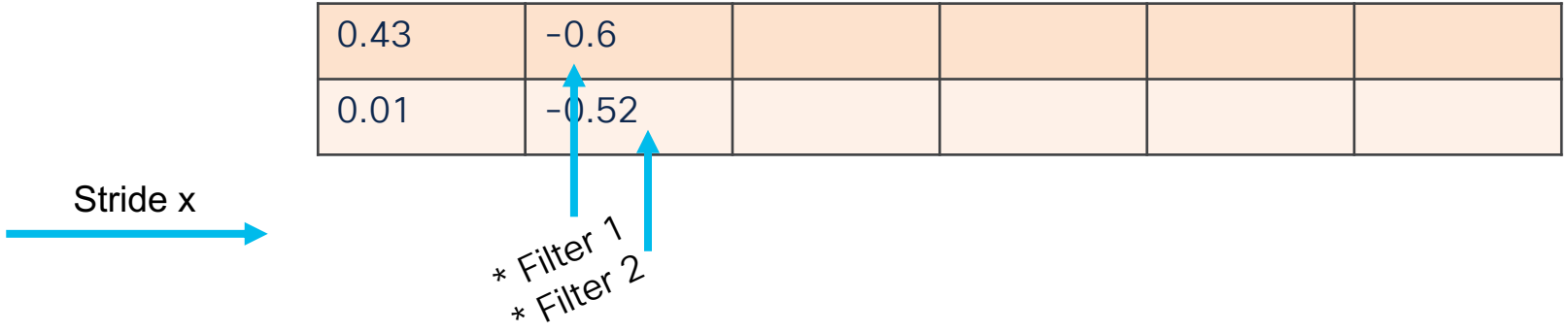
Assuming the embedding dimension is 2 (in our model it is 10 but for simplicity I show 2 here)

[[0.1, 0.2], [0.3, 0.4], [0.5, 0.6], [0.7, 0.8], [0.9, 1.0], [0.2, 0.3], [0.4, 0.5], [0.5, 0.6], [0.6, 0.7],
[0.9, 1.0], [0.1, 0.2], [0.2, 0.3], [0.3, 0.4], [0.4, 0.5], [0.3, 0.4], [0.5, 0.6], [0.6, 0.7], [0.1, 0.2],
[0.4, 0.5], [0.3, 0.4], [0.1, 0.2], [0.3, 0.4], [0.2, 0.3], [0.4, 0.5], [0.7, 0.8], [0.1, 0.2]]

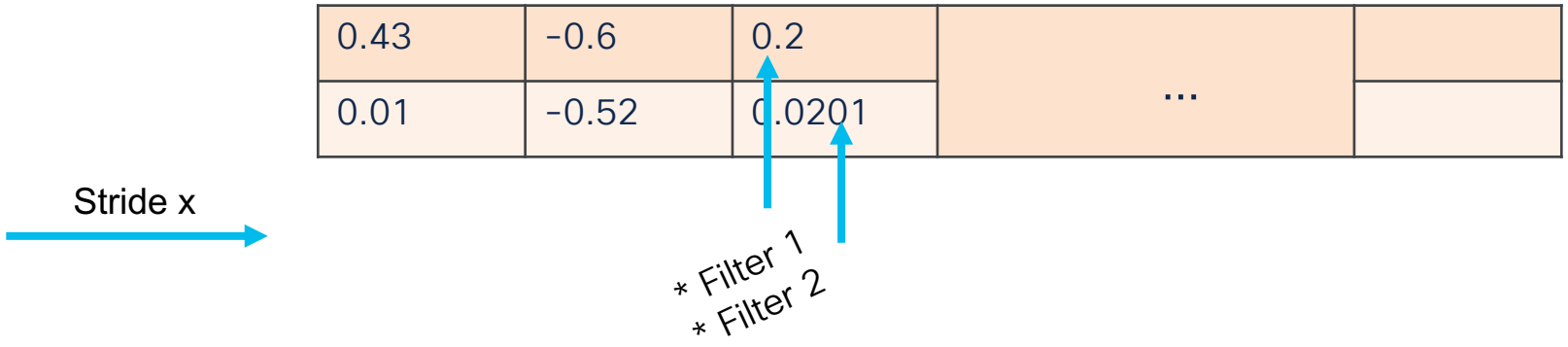


Kernel

[[0.1, 0.2], [0.3, 0.4], [0.5, 0.6], [0.7, 0.8], [0.9, 1.0], [0.2, 0.3], [0.4, 0.5], [0.5, 0.6], [0.6, 0.7], [0.9, 1.0], [0.1, 0.2], [0.2, 0.3], [0.3, 0.4], [0.4, 0.5], [0.3, 0.4], [0.5, 0.6], [0.6, 0.7], [0.1, 0.2], [0.4, 0.5], [0.3, 0.4], [0.1, 0.2], [0.3, 0.4], [0.2, 0.3], [0.4, 0.5], [0.7, 0.8], [0.1, 0.2]]



[[0.1, 0.2], [0.3, 0.4], [0.5, 0.6], [0.7, 0.8], [0.9, 1.0], [0.2, 0.3], [0.4, 0.5], [0.5, 0.6], [0.6, 0.7],
[0.9, 1.0], [0.1, 0.2], [0.2, 0.3], [0.3, 0.4], [0.4, 0.5], [0.3, 0.4], [0.5, 0.6], [0.6, 0.7], [0.1, 0.2],
[0.4, 0.5], [0.3, 0.4], [0.1, 0.2], [0.3, 0.4], [0.2, 0.3], [0.4, 0.5], [0.7, 0.8], [0.1, 0.2]]



[[0.1, 0.2], [0.3, 0.4], [0.5, 0.6], [0.7, 0.8], [0.9, 1.0], [0.2, 0.3], [0.4, 0.5], [0.5, 0.6], [0.6, 0.7], [0.9, 1.0], [0.1, 0.2], [0.2, 0.3], [0.3, 0.4], [0.4, 0.5], [0.3, 0.4], [0.5, 0.6], [0.6, 0.7], [0.1, 0.2], [0.4, 0.5], [0.3, 0.4], [0.1, 0.2], [0.3, 0.4], [0.2, 0.3], [0.4, 0.5], [0.7, 0.8], [0.1, 0.2]]

0.43	-0.6	0.2	...	0.205
0.01	-0.52	0.0201		0.12

Stride x

* Filter 1
* Filter 2

[[0.1, 0.2], [0.3, 0.4], [0.5, 0.6], [0.7, 0.8], [0.9, 1.0], [0.2, 0.3], [0.4, 0.5], [0.5, 0.6], [0.6, 0.7], [0.9, 1.0], [0.1, 0.2], [0.2, 0.3], [0.3, 0.4], [0.4, 0.5], [0.3, 0.4], [0.5, 0.6], [0.6, 0.7], [0.1, 0.2], [0.4, 0.5], [0.3, 0.4], [0.1, 0.2], [0.3, 0.4], [0.2, 0.3], [0.4, 0.5], [0.7, 0.8], [0.1, 0.2]]

Kernel

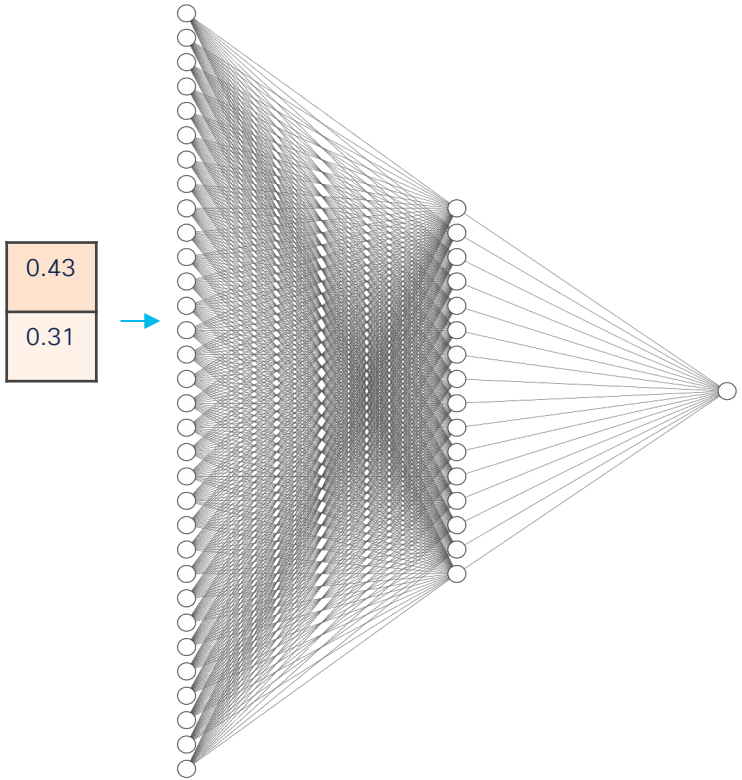
MAX Pooling

0.43	-0.6	0.2	0.4	-0.9	0.205
0.01	-0.52	0.0201	0.2	0.31	0.12

MAX Pooling

0.43	-0.6	0.2	0.4	-0.9	0.205
0.01	-0.52	0.0201	0.2	0.31	0.12

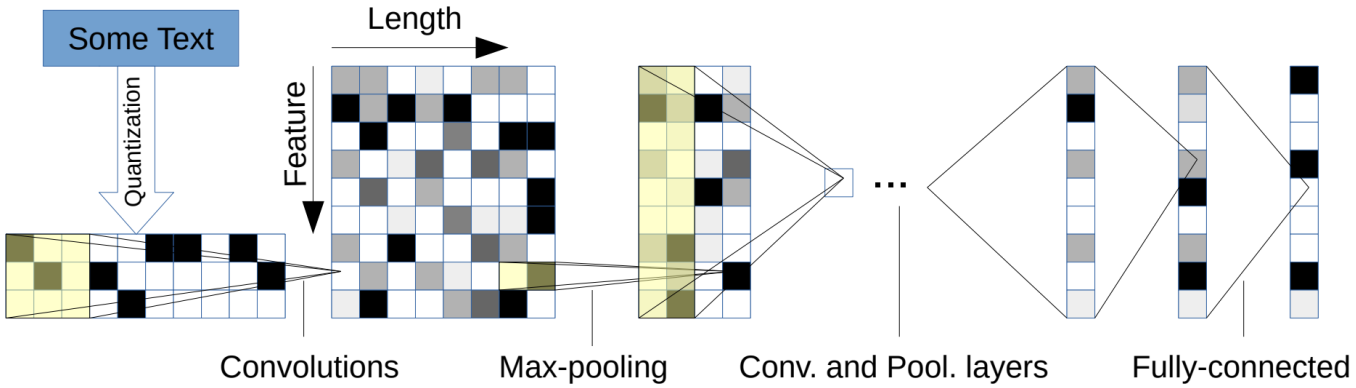
Dense layers and output



0.62 percent chance that this filename is sensitive (belongs to class 1)

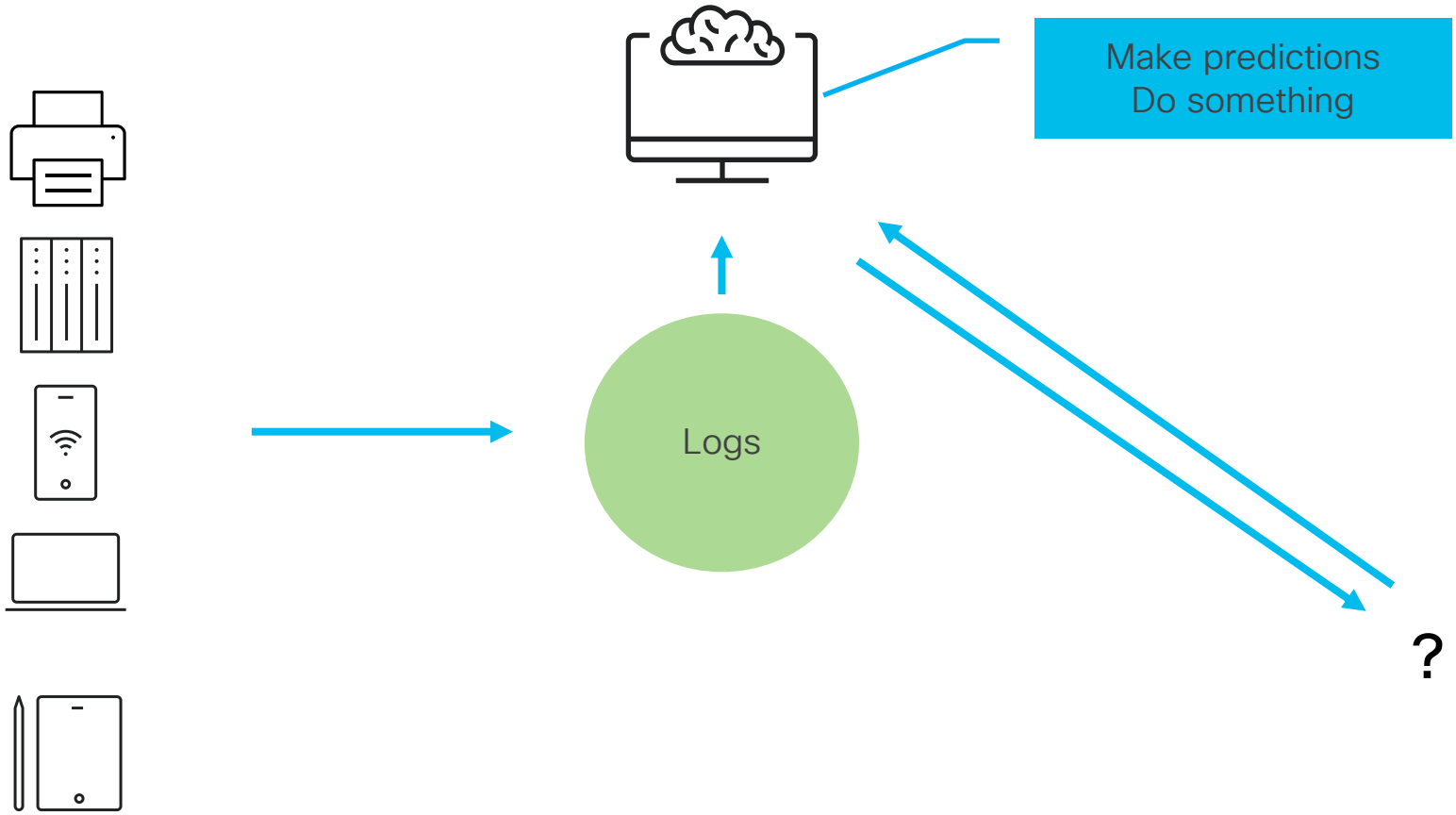
This model is similar to the one constructed from the research re: **Xiang Zhang, Junbo Zhao, Yann LeCun. "Character-level Convolutional Networks for Text Classification."**

Architecture from the research



Xiang Zhang, Junbo Zhao, Yann LeCun. “Character-level Convolutional Networks for Text Classification.” *arXiv preprint arXiv:1509.01626*, 2015

Application


Make predictions
Do something
Logs
?

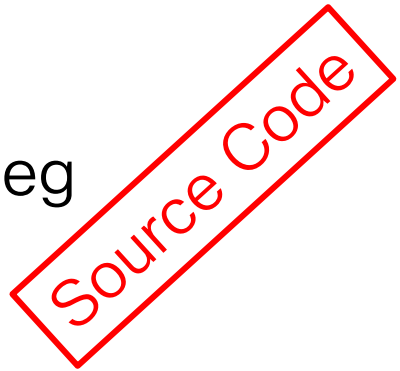
Alerting

For samples where we have confidence (>0.7) and the destination is something we know we do not sanction, raise an incident.

Continuous Synthetic Eyes

For samples that we are not completely confident (0.5-0.69) create an observation or contact someone to review if they have time.

Confidential_source_code001.jpeg



Wrap Up

More coverage

As discussed, DLP is not a one tool fits all approach. Deploying models for monitoring can help fill in the gaps.

Less Cost (comparatively)

The CNN model architecture discussed today is computationally cheap and easy to deploy

Better Upkeep

Though upkeep will always be a factor, like in anything else, these models can be retrained or fine-tuned with minimal effort if the pipeline is there.

Thank you for your time

Connect With Me

https://www.linkedin.com/in/samuel-cameron-cyber-security/

https://github.com/phantomOPBro

samuelrcameron@yahoo.com

Results When Testing

Model/Stats	1	2	3	4	5	6	7	8
Loss	1.474581480026250	0.148036181926727	0.440393894910812	0.308385491371155	0.205238685011864	0.079444326460361	0.059998337179422	0.250516206026077
Accuracy	0.885355293750763	0.987043559551239	0.913231253623962	0.984295248985291	0.986258327960968	0.991755008697510	0.992540240287781	0.945033371448517
Time to train (2.5m data points) (rounded)	15 min	28 min	30 min	25 min	27 min	30 min	30 min	28 min

Adjustments

Layers

I experimented with adding more Conv1D layers, adding in dropout layers, taking out the embedding layer, changing from maxpooling to averagepooling

Features

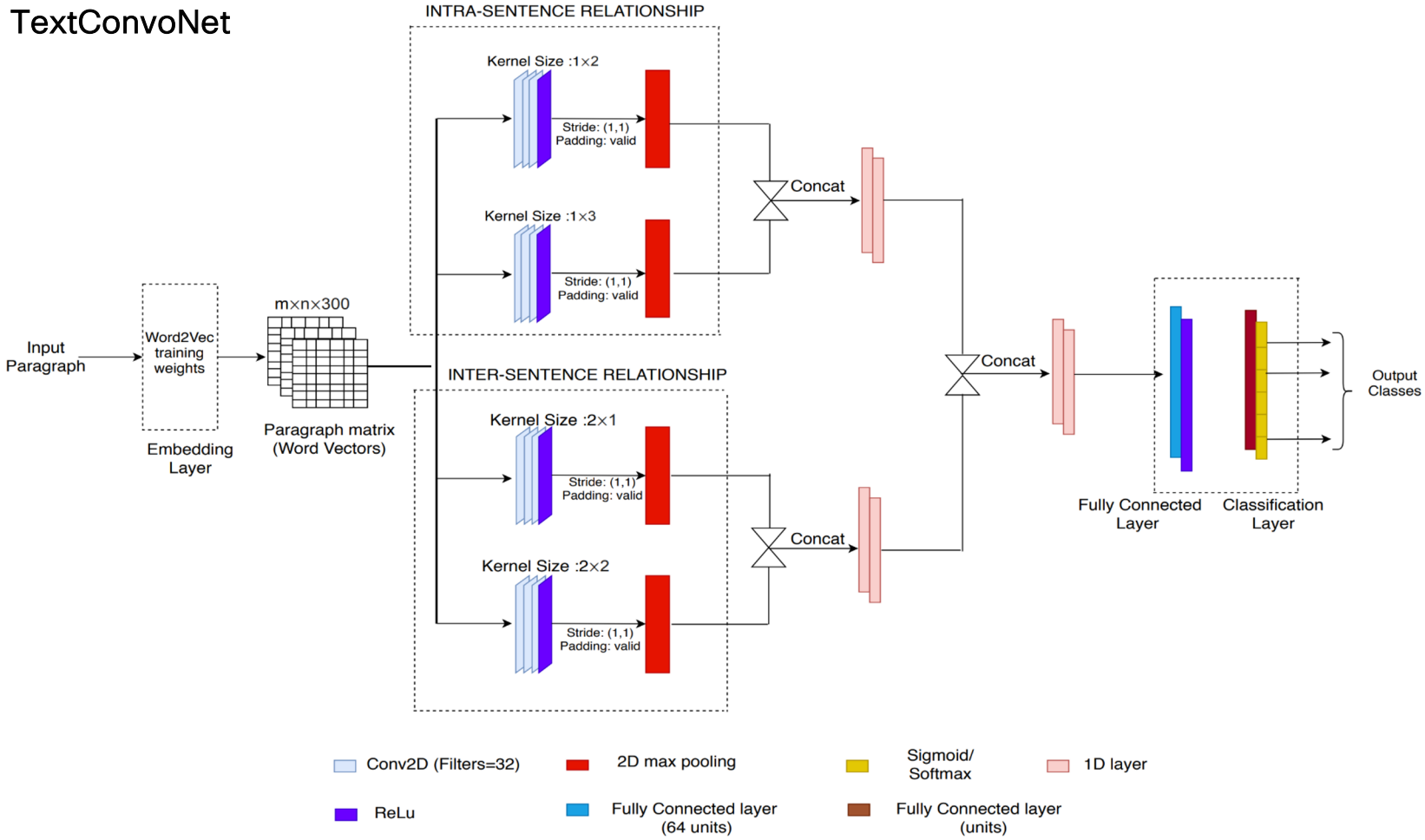
I added in some additional features such as the length of the filename, length of the filename without the extension, and the file extension

Parameters

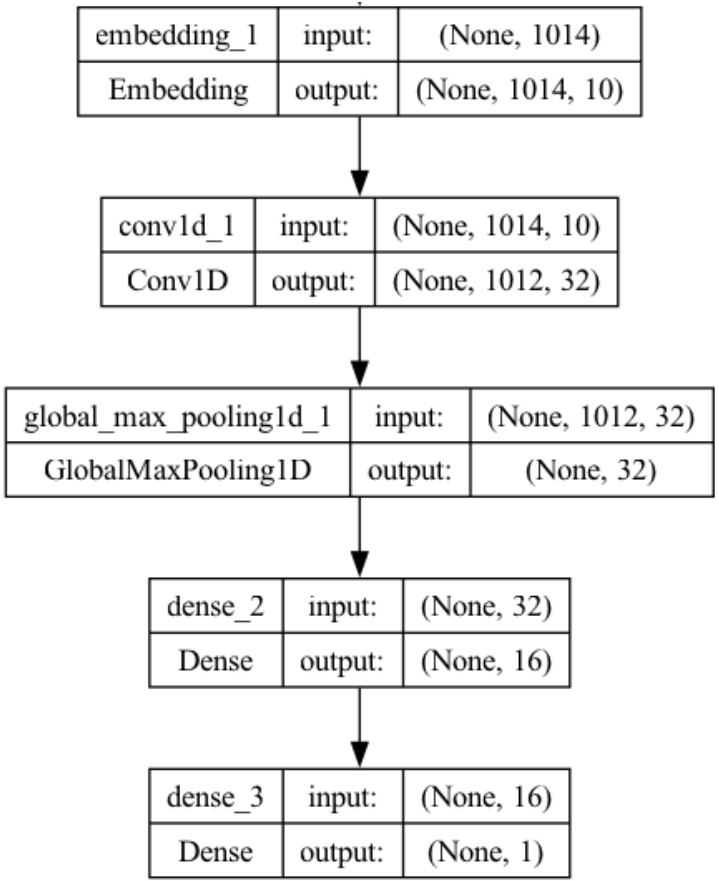
This is the main tweak throughout training. I adjusted the window size for the kernel, the number of filters applied, the embedding layer dimension, the max length of the sequences going into the embedding.

Useful Links and Graphics

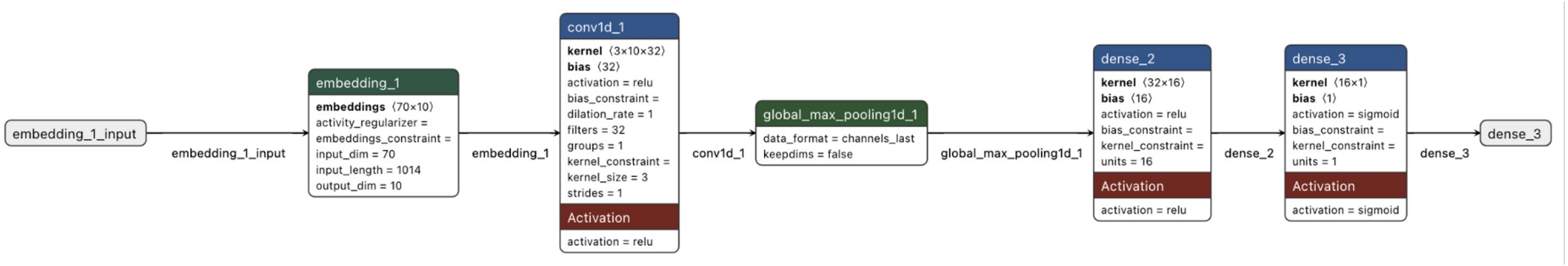
TextConvoNet



Output from tensorboard



Another View of the net



By the numbers

Ponemon Institute (2022) found that 56% of incidents were attributable to insider *negligence* whereas 26% were attributable to *malicious* insiders.

https://*citrap*.scholasticahq.com/#fn5

Negligent employees and credential thieves are the root causes of most insider incidents. Fifty-seven percent of respondents say the insider incidents involved employee negligence and 51 percent say a malicious outsider stole data by compromising insider credentials or accounts

2022 Cost of Insider Threats Global Report

https://www.proofpoint.com/sites/default/files/threat-reports/pfpt-us-tr-the-cost-of-insider-threats-ponemon-report.pdf

Sixty-five percent of respondents say email is where employees store their organizations most sensitive data such as personally identifiable information (PII), intellectual property (IP) and other critical business information.

2022 Cost of Insider Threats Global Report

https://www.proofpoint.com/sites/default/files/threat-reports/pfpt-us-tr-the-cost-of-insider-threats-ponemon-report.pdf