Analyzing Polticial Bias through A User-Friendly Interface

Colin Lightfoot

12/8/2017

Overview

Problem

1. Many news outlets report stories with biased undertones.
2. Some hide facts to present a specific argument.
3. The average person lacks the time to see multi-faceted stories.

Problem

1. Many news outlets report stories with biased undertones.
2. Some hide facts to present a specific argument.
3. The average person lacks the time to see multi-faceted stories.

Problem

1. Many news outlets report stories with biased undertones.
2. Some hide facts to present a specific argument.
3. The average person lacks the time to see multi-faceted stories.

Solution

1. Researchers have begun to use neural networks for textual analyses.
2. Recent studies have found neural networks to classify text accurately.
3. Use proven neural network techniques to create a useful political classifier.

Solution

1. Researchers have begun to use neural networks for textual analyses.
2. Recent studies have found neural networks to classify text accurately.
3. Use proven neural network techniques to create a useful political classifier.

Solution

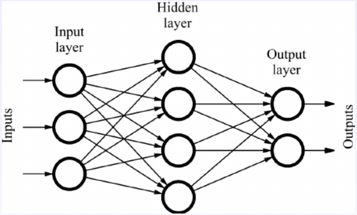
1. Researchers have begun to use neural networks for textual analyses.
2. Recent studies have found neural networks to classify text accurately.
3. Use proven neural network techniques to create a useful political classifier.

Related Work

- Iyyer et al. created political bias classifiers using Recursive Neural Network (RNN) models with Long Short Term Memory (LSTM) nodes with a high degree of accuracy.[1]
- Stanford researchers Arkajyoti Misra and Sanjib Basak showed that RNNs with LSTM nodes can also predict implicit political bias.[2]

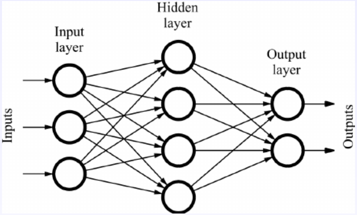
Introduction to Neural Networks

- Algorithms modeled after the brain.
- Organized into layers with interconnected nodes.
- Weighted edges connect nodes.
- Layers are successively computed based off computations from the lower layers.
- Nodes transform input using an activation function and then outputs result.



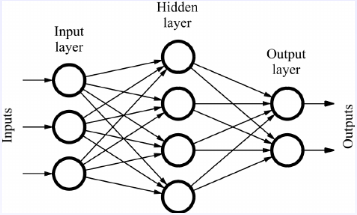
Introduction to Neural Networks

- Algorithms modeled after the brain.
- Organized into layers with interconnected nodes.
- Weighted edges connect nodes.
- Layers are successively computed based off computations from the lower layers.
- Nodes transform input using an activation function and then outputs result.



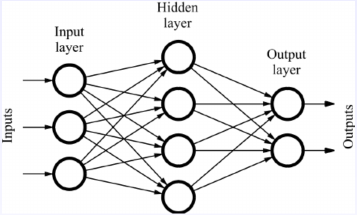
Introduction to Neural Networks

- Algorithms modeled after the brain.
- Organized into layers with interconnected nodes.
- Weighted edges connect nodes.
- Layers are successively computed based off computations from the lower layers.
- Nodes transform input using an activation function and then outputs result.

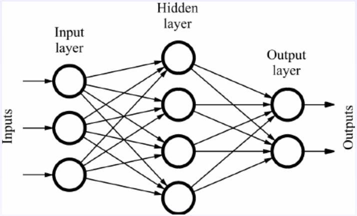


Introduction to Neural Networks

- Algorithms modeled after the brain.
- Organized into layers with interconnected nodes.
- Weighted edges connect nodes.
- Layers are successively computed based off computations from the lower layers.
- Nodes transform input using an activation function and then outputs result.

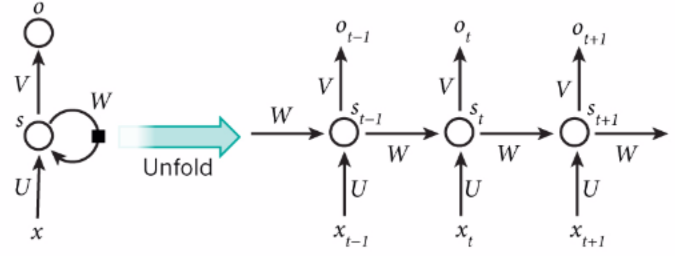


- Algorithms modeled after the brain.
- Organized into layers with interconnected nodes.
- Weighted edges connect nodes.
- Layers are successively computed based off computations from the lower layers.
- Nodes transform input using an activation function and then outputs result.



Recurrent Neural Networks

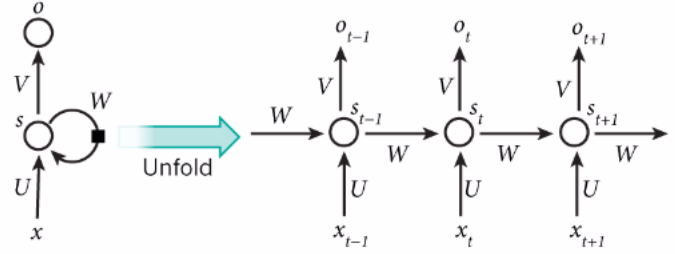
- Designed to make use of sequential information.
- Perform same computations for every element in a sequence.
- The output is dependent on the model's previous recurrences.
- RNNs can have dependent output since nodes hold previous calculations.



An unfolded recursive neural network

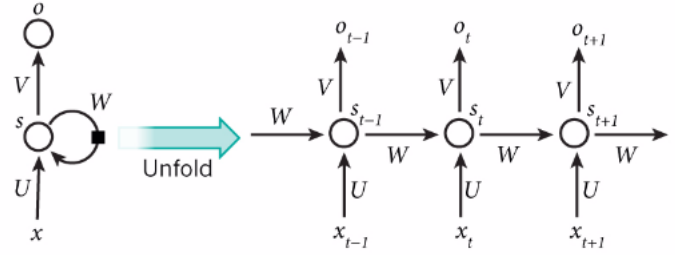
Recurrent Neural Networks

- Designed to make use of sequential information.
- Perform same computations for every element in a sequence.
- The output is dependent on the model's previous recurrences.
- RNNs can have dependent output since nodes hold previous calculations.



An unfolded recursive neural network

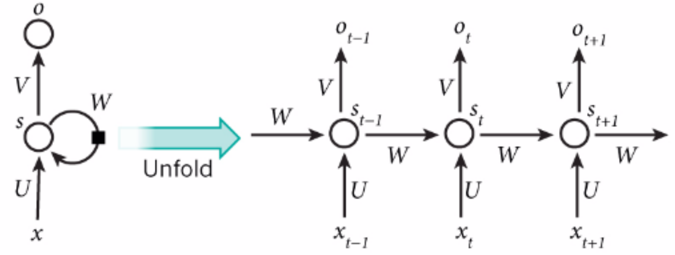
- Designed to make use of sequential information.
- Perform same computations for every element in a sequence.
- The output is dependent on the model's previous recurrences.
- RNNs can have dependent output since nodes hold previous calculations.



An unfolded recursive neural network

Recurrent Neural Networks

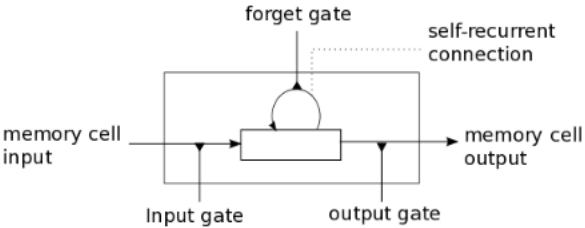
- Designed to make use of sequential information.
- Perform same computations for every element in a sequence.
- The output is dependent on the model's previous recurrences.
- RNNs can have dependent output since nodes hold previous calculations.



An unfolded recursive neural network

LSTM Nodes

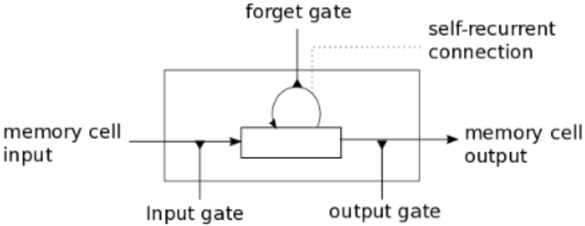
- Retains standard node properties.
- Prevents exploding and vanishing gradient dilemmas during edges' weight training.
- Has a chance to "forget" its current state.
- Otherwise remembers state for next recurrence.



A simple LSTM node.

LSTM Nodes

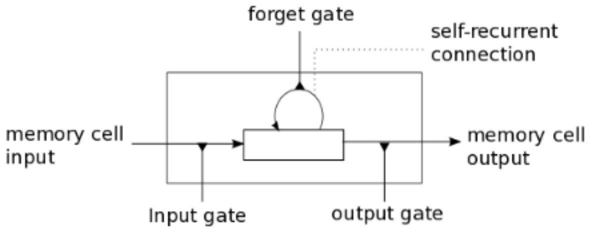
- Retains standard node properties.
- Prevents exploding and vanishing gradient dilemmas during edges' weight training.
- Has a chance to "forget" its current state.
- Otherwise remembers state for next recurrence.



A simple LSTM node.

LSTM Nodes

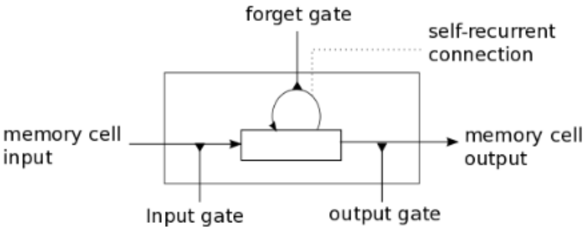
- Retains standard node properties.
- Prevents exploding and vanishing gradient dilemmas during edges' weight training.
- Has a chance to "forget" its current state.
- Otherwise remembers state for next recurrence.



A simple LSTM node.

LSTM Nodes

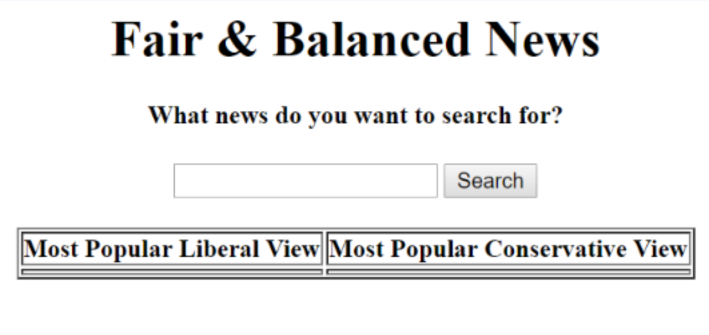
- Retains standard node properties.
- Prevents exploding and vanishing gradient dilemmas during edges' weight training.
- Has a chance to "forget" its current state.
- Otherwise remembers state for next recurrence.



A simple LSTM node.

Implementation

- User searches a news topic.
- Web scraper searches queried article results.
- Articles are run through RNN.
- Cycle repeats until RNN finds both a liberal and a conservative article.



The user interface for my project.

Implementation

- User searches a news topic.
- Web scraper searches queried article results.
- Articles are run through RNN.
- Cycle repeats until RNN finds both a liberal and a conservative article.

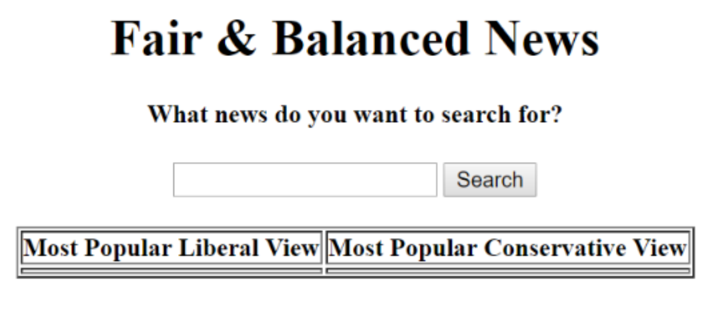
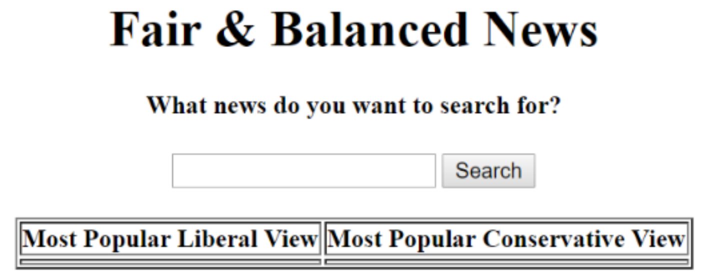


Figure 7: The user interface for my project.

Implementation

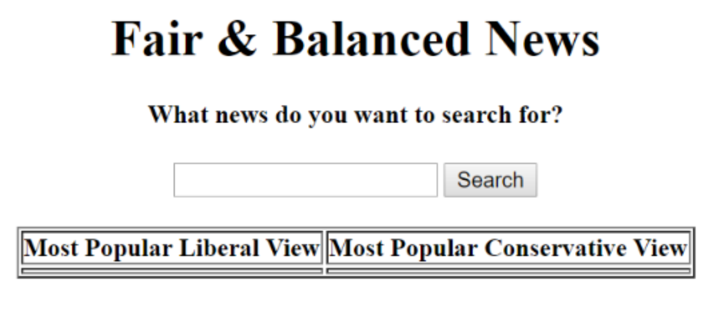
- User searches a news topic.
- Web scraper searches queried article results.
- Articles are run through RNN.
- Cycle repeats until RNN finds both a liberal and a conservative article.



The user interface for my project.

Implementation

- User searches a news topic.
- Web scraper searches queried article results.
- Articles are run through RNN.
- Cycle repeats until RNN finds both a liberal and a conservative article.



The user interface for my project.

F1 Score

Precision is measured as

$$\frac{true\ \ positives}{false\ \ positives + true\ \ positives},$$

while recall is measured as

$$\frac{true\ \ positives}{false\ \ negatives + true\ \ positives}.$$

Measuring the amount of false positives and false negatives shows model's ability to not predict Type I and II errors.
The formula for calculating the F1 Score is:

$$F1\ \ Score = 2 \times \frac{precision \times recall}{precision + recall}.$$

Evaluation

- F1 Score: 0.90
- Overall Accuracy: 57.49%
- Conservative Accuracy: 57.065%
- Liberal Accuracy: 65.625%

Evaluation

- F1 Score: 0.90
- Overall Accuracy: 57.49%
- Conservative Accuracy: 57.065%
- Liberal Accuracy: 65.625%

Evaluation

- F1 Score: 0.90
- Overall Accuracy: 57.49%
- Conservative Accuracy: 57.065%
- Liberal Accuracy: 65.625%

Evaluation

- F1 Score: 0.90
- Overall Accuracy: 57.49%
- Conservative Accuracy: 57.065%
- Liberal Accuracy: 65.625%

Possible Noise

Possible Noise:

- Various-sized datasets
- Complexity of language
- Small sentence sizes

Future Work

- Continue tuning RNN
- Collect larger datasets
- Scrape more news outlets
- Run against other algorithms
- Better user interface

Conclusion

Thank you professors Deverick and Lewis, as well as my friends and family for helping me complete this project! I was able to apply knowledge learned throughout my time here to create a useful and easy-to-use webpage that helps people understand the political atmosphere surrounding a news event.

References

Mohit Iyyer, Peter Enns, Jordan Boyd-Graber, Philip Resnik. *Political Ideology Detection Using Recursive Neural Networks* http://www.aclweb.org/anthology/P/P14/P14-1105.pdf

Arkajyoti Misra and Sanjib Basak. *Political Bias Analysis* https://cs224d.stanford.edu/reports/MisraBasak.pdf