Title
Fake news classification using transfer learning

Abstract

Introduction

Fake news has become one of our age's biggest issues. It has corrupted both online and offline debate, and one can even go as far as to suggest that, to date, fake news presents a clear and present danger to the community. "False reports" are used to symbolize untrue rumour or misinformation encompassing propaganda conversed with the help of conventional medium such as publishing houses, television and also through non-conventional medium such as social networking sites. The universal reasons of spreading such kind of reports engage confusing the readers; damaging the status of some individual, or for the attainment of some aggrandizement. Fake news is considered as furthermost intimidation to democratic system, liberated discussion and the Western sort. Counterfeit reports are capable of producing damaging effects on persons and the civilization. Because of the fake news individuals may be deceived and admit bogus attitude. Counterfeit reports can alter the people's reaction towards factual reports. The reliability of whole report environment can be damaged through extensive broadcasting of bogus information. Thus, the recognition of bogus information on social networking sites is necessary. False reports are deliberately printed to misinform customers; therefore the recognition normal news becomes difficult. The exploration of supplementary data from dissimilar perceptions is normal and essential for the development of an effectual and sensible bogus reports recognition scheme.

False reports discovery has freshly fascinated a mounting attention from the common community as well as researchers because of the online increasing of propaganda movement predominantly in medium outlets like social networking sites, supplies, report blogs, and online correspondents. Some techniques offer immense assurance for developer for the construction of schemes which can involuntarily perceive counterfeit reports. Though, recognition of false information is a demanding duty to achieve because of its requirement of mock-up for the summarization of reports. For the classification of false report, a comparison is performed between genuine and fraudulent information. Furthermore, the assignment of evaluation of proposed news with the genuine report is an intimidating chore because of its prejudice and estimation. A dissimilar method for the detection of bogus information is from stance recognition .attitude recognition is the procedure of mechanically noticing the association amid two portions of content. A way is explored in this research for the prediction of the deportment in association with information editorial and information caption couple. Relying on the similarity between same news editorial text and caption, the posture amid them can be depicted as “ready’, ‘not ready’, ‘discussion’ or ‘neutral’. A lot of experiments are conducted with some conventional machine learning approaches for the settlement of a baseline and then comparison is performed between these results to the state of-the art deep networks for the classification of the attitude amid editorial corpse and caption. A number of phases are involved in the recognition of false reports. The ground of false information recognition is a comparatively novel region of investigation. Hence, a small number of unrestricted data samples are available. A primarily new data sample is collected by the researcher through compilation of publicly obtainable information editorials. Information pre-processing prepares unprocessed information for supplementary dispensation. The conventional information pre-processing technique initiates with data which is implicitly prepared for investigation without any response and convey the method of information compilation. In tokenization, the stream of content is breached into language, idioms, cryptogram or additional significant rudiments described as tokens. The aspiration of this technique is the examination of the vocabulary in a statement. The stream of symbols is converted into input for supplementary dispensation like parsing or passage withdrawal. Stemming is the procedure of converting the alternative patterns of a statement into an ordinary depiction called stem. For example, the words: “presentation”, “presented”, “presenting” could all be summarized to an ordinary illustration called “present”. This procedure is utilized globally in content dispensation for performing data repossession.

Related work

Katherine Clayton, et al. (2019) evaluated the effectiveness of different technologies using which the false stories posted on various social networking sites like Facebook could be identified [12]. The belief in false news is reduced to a moderate level by the “Disputed” and “Rated false” tags. It has seen that “Disputed” approach provided higher accuracy results in previous approaches. However, this research showed that the belief in misinformation for reduced to greater extent by the “Rated False” mechanism. Simulation results achieved showed that the effective approaches derived from this work proved to be highly beneficial when applied in real time scenarios. Atodiresei, et al. (2018) suggested a scheme for the identification of fraudulent twitter followers and fraudulent twitter reports [13]. The presented approach returned a pattern of figures about the authenticity of tweets. By the means of tweeter groups and tweeter content, the projected approach did not achieve its main objective yet. Recognizing certainly that the presented report was false or not relied exclusively on its reputation in the similar social media platform was not a good suggestion. For the identification of false reports, individual possessions were used by the face book for the investigation of reputed news. Adware, et al. (2018) defined an easy and efficient methodology for the clients which allowed them for the installation of a simple technique into their individual account [14]. The presented tool also allowed users to utilize this tool for identification and elimination of probable click enticements. The proposed approach showed tremendous performance in the recognition of false report origins. Girgis, et al. (2018) stated that the main aim of this study was the development of a classifier for the prediction of fraudulent reports [15]. The fakeness of a report was based on the text. Thus for the encounter of false report issue, a completely deep neural approach with the outlook of RNN scheme representation and LSTMs was implemented in the proposed study. The tested results depicted that GRU showed best accuracy in comparison with several other approaches. Al-Ash, et al. (2018) proposed a research work for the modelling of vectors for the accommodation of false reports features [16]. The vector modelling was performed before the supplementary progression by speech approaches with the help of the Indonesian communication. The major objective of the projected research was the identification of false reports. With the help of support vector machine approach, the frequency was conversed in the form of tenfold cross corroboration. A much admirable performance had been shown by the vector demonstration which utilized the phrase frequency. Vedova, et al. (2018) proposed a new ML false reports recognition methodology that combined the characteristics of public and reports texts [17]. The development of the projected approach was performed with the utilization of content relied and public relied methodology. For the validation of proposed approach, a number of experiments were carried out. The tested results verified the authenticity of the proposed approach. The combination of both of these approaches was relied on threshold imperative. The threshold rule was capable of capturing the distinct assistances of the proposed approaches and also performed better in comparison with several other methodologies. In future, the training of classifier with opinion reality in other speeches will be performed for the extension of proposed approach in several other nations.

Proposed work

In the fake news classification task we tried 4 models, there are

1. Siamese CNN
2. Siamese RNN
3. Siamese LSTM
4. VNet (proposed model)

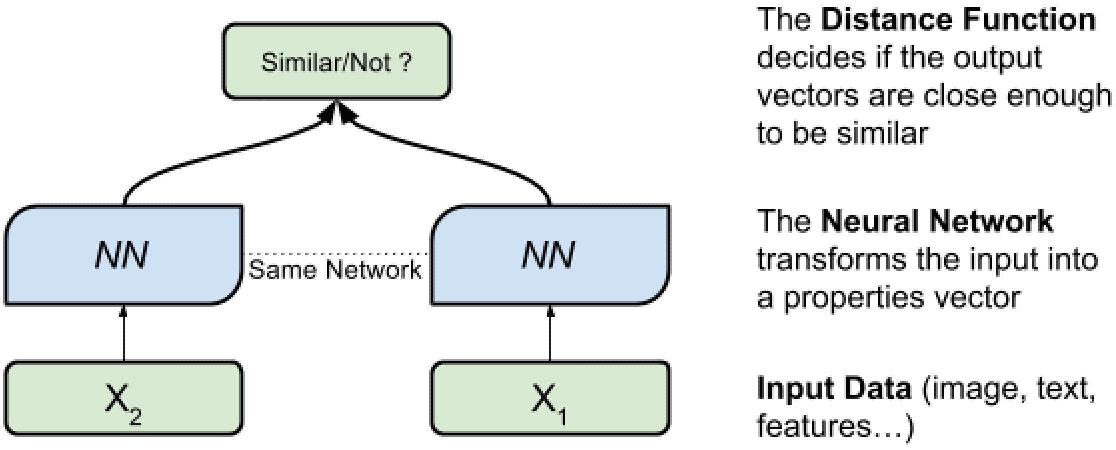
Before researching this we just embed the each news headlines and the actual context with word embedding algorithm called GloVe, which converts the natural language to dense vector space (n-dimensional)

Siamese Neural Networks

Siamese network definition from Wikipedia,

"***Siamese neural network*** *is an artificial neural network that use the same weights while working in tandem on two different input vectors to compute comparable output vectors. Often one of the output vectors is precomputed, thus forming a baseline against which the other output vector is compared*".

Siamese networks are neural networks containing two or more identical sub network components. A siamese network may look like this:


Similar/Not ?
NN
Same Network
NN
X₂
X₁
The **Distance Function** decides if the output vectors are close enough to be similar
The **Neural Network** transforms the input into a properties vector
Input Data (image, text, features…)

It is important that not only the architecture of the sub networks is identical, but the weights have to be shared among them as well for the network to be called "siamese". The main idea behind siamese networks is that they can learn useful data descriptors that can be further used to compare between the inputs of the respective sub networks. Hereby, inputs can be anything from numerical data in this case the sub networks are usually formed by fully-connected layers), image data with CNNs as sub networks) or even sequential data such as sentences or time signals (with RNNs as sub networks).

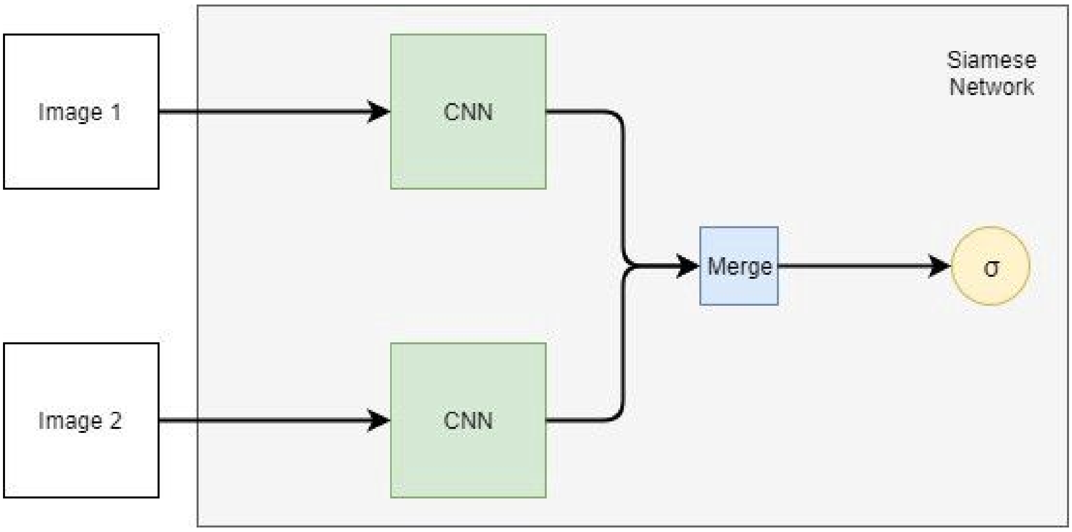
Siamese Neural networks in Fake news classification:

- The task of classification of the fake news is that very important thing today among public. The fake news classification is the task of identifying the fake/real relationship between the News headline and the actual whole context of the news.
- In the Siamese neural network, the news headline is given as one input and the actual content will be given as another input at the same time, in the output layer the fake/real probability will be given.
- The siamese net that finds the actual relation between the two inputs with respect to the output value.
- In the siamese net, we used three state of the art layers
    - 1D CNN
    - RNN
    - LSTM

1D CNN Siamese Network

In Siamese Neural network we use 1D CNN layer instead of dense layer is called as 1D CNN Siamese Network. This network has the following components (Note: each word will be Embedded, and the created embedded vectors will be passed as inputs)

1. Headline
2. Actual Content of the news



What is convolution?

In purely mathematical terms, convolution is a function derived from two given functions by integration which expresses how the shape of one is modified by the other. That can sound baffling as it is, but to make matters worse, we can take a look at the convolution formula:

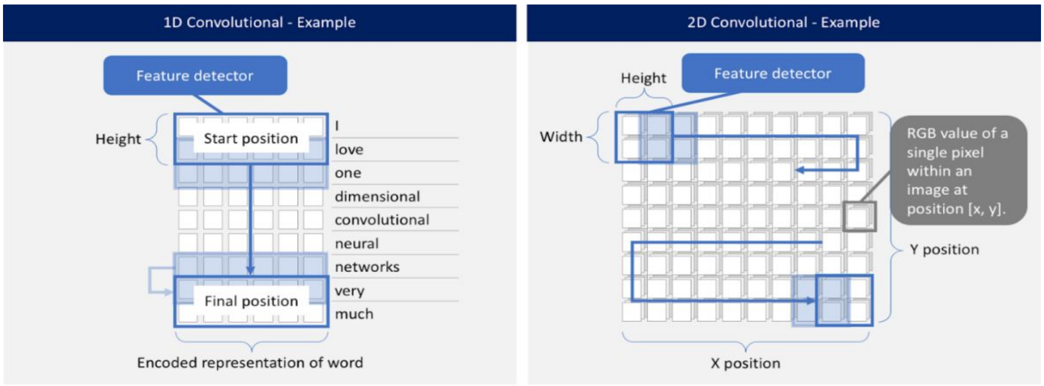
$$(f * g)(t) \overset{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau)\, g(t - \tau)\, d\tau$$

If you don't consider yourself to be quite the math buff, there is no need to worry since this course is based on a more intuitive approach to the concept of convolutional neural networks, not a mathematical or a purely technical one.

1D CNN layer

The convolutional layer is the core building block of a CNN. The layer's parameters consist of a set of learnable filters (or kernels), which have a small receptive field, but extend through the full depth of the input volume. During the forward pass, each filter is convolved across the width and height of the input volume, computing the dot product between the entries of the filter and the input and producing a 2-dimensional activation map of that filter. As a result, the network learns filters that activate when it detects some specific type of feature at some spatial position in the input.

Stacking the activation maps for all filters along the depth dimension forms the full output volume of the convolution layer. Every entry in the output volume can thus also be interpreted as an output of a neuron that looks at a small region in the input and shares parameters with neurons in the same activation map.



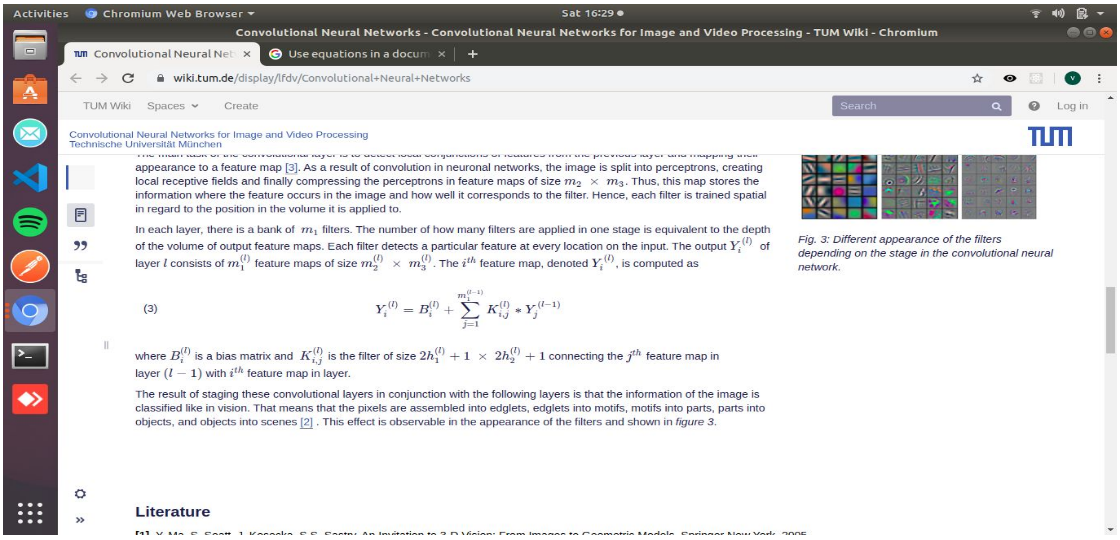
In this example for natural language processing, a sentence is made up of 9 words. Each word is a vector that represents a word as a low dimensional representation. The feature detector will always cover the whole word. The height determines how many words are considered when training the feature detector. In our example, the height is two. In this example the feature detector will iterate through the data 8 times.

In this example for computer vision, each pixel within the image is represented by its x- and y position as well as three values (RGB). The feature detector has a dimension of 2 x 2 in our example. The feature detector will now slide both horizontally and vertically across the image.

(In Natural language processing, a sentence is made up of 9 words .Each word is a vector that represents a word as a low dimensional representation .The feature detector will always cover the whole word. The height determines how many words are considered when training the feature detector. In this example the height is two, the feature detector will iterate through the data 8 times)

The main task of the convolutional layer is to detect local conjunctions of features from the previous layer and mapping their appearance to a feature map. As a result of convolution in neuronal networks, the image is split into perceptron's, creating local receptive fields and finally compressing the perceptron's in feature maps of size m2 × m3. Thus, this map stores the information where the feature occurs in the image and how well it corresponds to the filter. Hence, each filter is trained spatial in regard to the position in the volume it is applied to.

Each layer, there is a bank of m1 filters. The number of how many filters are applied in one stage is equivalent to the depth of the volume of output feature maps. Each filter detects a particular feature at every location on the input. The output Yi (l) of layer (l) consists of m1 (l) feature map size m2 (l) x m3 (l)
The feature map is denoted as Yi


Activities
Chromium Web Browser
Sat 16:29
Convolutional Neural Networks - Convolutional Neural Networks for Image and Video Processing - TUM Wiki - Chromium
Convolutional Neural Net
Use equations in a docum
wiki.tum.de/display/lfdv/Convolutional+Neural+Networks
TUM Wiki
Spaces
Create
Search
Log in
Convolutional Neural Networks for Image and Video Processing
Technische Universität München

The main task of the convolutional layer is to detect local conjunctions of features from the previous layer and mapping their appearance to a feature map [3]. As a result of convolution in neuronal networks, the image is split into perceptrons, creating local receptive fields and finally compressing the perceptrons in feature maps of size $m_2 \times m_3$. Thus, this map stores the information where the feature occurs in the image and how well it corresponds to the filter. Hence, each filter is trained spatial in regard to the position in the volume it is applied to.

In each layer, there is a bank of m_1 filters. The number of how many filters are applied in one stage is equivalent to the depth of the volume of output feature maps. Each filter detects a particular feature at every location on the input. The output $Y_i^{(l)}$ of layer l consists of $m_1^{(l)}$ feature maps of size $m_2^{(l)} \times m_3^{(l)}$. The i^{th} feature map, denoted $Y_i^{(l)}$, is computed as

$$Y_i^{(l)} = B_i^{(l)} + \sum_{j=1}^{m_1^{(l-1)}} K_{i,j}^{(l)} * Y_j^{(l-1)} \tag{3}$$

where $B_i^{(l)}$ is a bias matrix and $K_{i,j}^{(l)}$ is the filter of size $2h_1^{(l)} + 1 \times 2h_2^{(l)} + 1$ connecting the j^{th} feature map in layer $(l-1)$ with i^{th} feature map in layer.

The result of staging these convolutional layers in conjunction with the following layers is that the information of the image is classified like in vision. That means that the pixels are assembled into edglets, edglets into motifs, motifs into parts, parts into objects, and objects into scenes [2] . This effect is observable in the appearance of the filters and shown in *figure 3*.

Fig. 3: Different appearance of the filters depending on the stage in the convolutional neural network.

Literature

[1] Y. Ma, S. Soatt, J. Kosecka, S.S. Sastry. An Invitation to 3-D Vision: From Images to Geometric Models. Springer New York, 2005

Where B is bias matrix K is the filter that connecting to the jth feature map in the (l-1) layer with i th feature map in layer.

Max pooling layer:

A pooling layer is often used after a CNN layer in order to reduce the complexity of the output and prevent over fitting of the data.

Relu Layer

ReLU is the abbreviation of rectified linear unit, which applies the non-saturating activation function f (x) = max (0 , x) It effectively removes negative values from an activation map by setting them to zero.[59] It increases the nonlinear properties of the decision function and of the overall network without affecting the receptive fields of the convolution layer.

Fully connected layer:

The output of the previous layer will be flatten and connected to dense layers

Loss Layer

The **"loss layer"** specifies how training penalizes the deviation between the predicted (output) and true labels and is normally the final layer of a neural network. Various loss functions appropriate for different tasks may be used.

Softmax:

$$f_j(z) = \frac{e^{z_j}}{\sum_k e^{z_k}}$$

Cross-entropy loss, or log loss, measures the performance of a classification model whose output is a probability value between 0 and 1. Cross-entropy loss increases as the predicted probability diverges from the actual label. So predicting a probability of .012 when the actual observation label is 1 would be bad and result in a high loss value. A perfect model would have a log loss of 0.

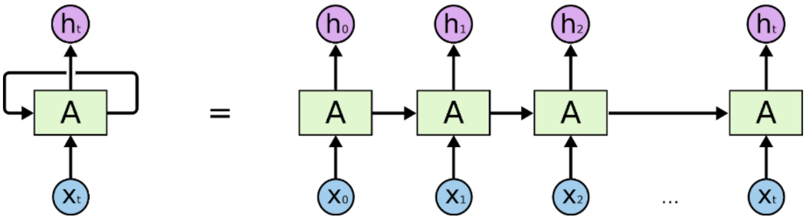
Cross entropy

$$L_i = -\log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right)$$

$$H(p,q) = -\sum_x p(x)\,\log q(x)$$

2 Siamese RNN for fake news classification:

In Siamese Neural network we use recurrent units in both input section. It’s as same as siamese cnn but we use RNN instead 1d CNN layer
Recurrent Neural Networks (RNN) are for handling sequential data. **RNNs share parameters across different positions**/ index of time/ time steps of the sequence, which makes it possible to generalize well to examples of different sequence length. RNN is usually a better alternative to position-independent classifiers and sequential models that treat each position differently.



The basic equations that defines the above RNN is shown below

$$\begin{aligned}\boldsymbol{a}^{(t)} &= \boldsymbol{b} + \boldsymbol{W}\boldsymbol{h}^{(t-1)} + \boldsymbol{U}\boldsymbol{x}^{(t)} \\ \boldsymbol{h}^{(t)} &= \tanh(\boldsymbol{a}^{(t)}) \\ \boldsymbol{o}^{(t)} &= \boldsymbol{c} + \boldsymbol{V}\boldsymbol{h}^{(t)} \\ \hat{\boldsymbol{y}}^{(t)} &= \operatorname{softmax}(\boldsymbol{o}^{(t)}) \qquad (10.6)\end{aligned}$$

The total loss for a given sequence of x values paired with a sequence of y values would then be just the sum of the losses over all the time steps. For example, if L(t) is the negative log-likelihood
Of y (t) given x (1), x (t), then sum them up you get the loss for the sequence as shown in (10.7):

$$L(\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(\tau)}\}, \{\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(\tau)}\}) = \sum_t L^{(t)} = \sum_t -\log \hat{y}^{(t)}_{y^{(t)}}. \qquad (10.7)$$

3. Siamese LSTM for fake news classification:

LSTM can be used to solve problems faced by the RNN model. So, it can be used to solve:

1. Long term dependency problem in RNNs.
2. Vanishing Gradient & Exploding Gradient.

*The heart of a LSTM network is its cell or say cell state which provides a bit of memory to the LSTM so it can remember the past.*i.e the cell state may remember the gender of the subject in a given input sequence so that the proper pronoun or verb can be used.
LSTM is made up of Gates:
In single LSTM we will have 3 gates:
1) Input Gate.
2) Forget Gate.
3) Output Gate.

The recurrent neural network (RNN) is used for the text classification that follows the sequential model. Problem in the RNN is the "vanishing gradient ", to overcome this we use the LSTM networks.
An LSTM network is a recurrent neural network that has LSTM cell blocks in place of our standard neural network layers. Main components of LSTM cells are 1) input gate, 2) the forget gate and 3) the output gate. The detailed about the LSTM cells depict in the diagram.

New word/sequence value input xt being concatenated to the previous output from the cell ht-1.

1. The first step for this combined input is for it to be squashed via a *tanh* layer.
2. The second step is that this input is passed through an *input gate*.
    a. An input gate is a layer of sigmoid activated nodes whose output is multiplied by the squashed input.
    b. These input gate sigmoid can act to “kill off” any elements of the input vector that aren’t required.
    c. A sigmoid function outputs values between 0 and 1, so the weights connecting the input to these nodes can be trained to output values close to zero to “switch off” certain input values.
    d. In conversely, outputs close to 1 to “pass through” other values.
2. The third step in the flow of data through this cell is the internal state / forget gate loop.

a. LSTM cells have an internal state variable St. This variable, lagged one time step St-1is *added* to the input data to create an effective layer of recurrence.

b. This *addition* operation, instead of a multiplication operation, helps to reduce the risk of vanishing gradients.

c. This recurrence loop is controlled by a forget gate – this works the same as the input gate, but instead helps the network learn which state variables should be “remembered” or “forgotten”.

2. Finally, we have an output layer *tan* squashing function, the output of which is controlled by an *output gate.* This gate determines which values are actually allowed as an output from the cell ht

Input Gate

1. Mathematical expression of *tanh* activation function

$g=tanh (bg+xtUg+ht-1Vg)$

Ug -> weights for the input

Vg -> weights for the previous cell output,

bg -> input bias.

1. Sigmoid activated nodes. It will multiply element-wise by the output of the *input gate,*

$$i=\sigma(bi+xtUi+ht-1Vi)$$

1. The output of the input section of the LSTM cell is then given by:

$$g\circ i$$

∘ -> operator expresses element-wise multiplication.

Forget gate and state loop

1. The internal state / forget gate loop is expressed as:

$$f=\sigma(bf+xtUf+ht-1Vf)$$

1. The output of the element-wise product of the previous state and the forget gate is expressed as

$$st-1\circ \mathbf{f}$$

1. . The output from the forget gate / state loop stage is:

$$st-1\circ f+g\circ i$$

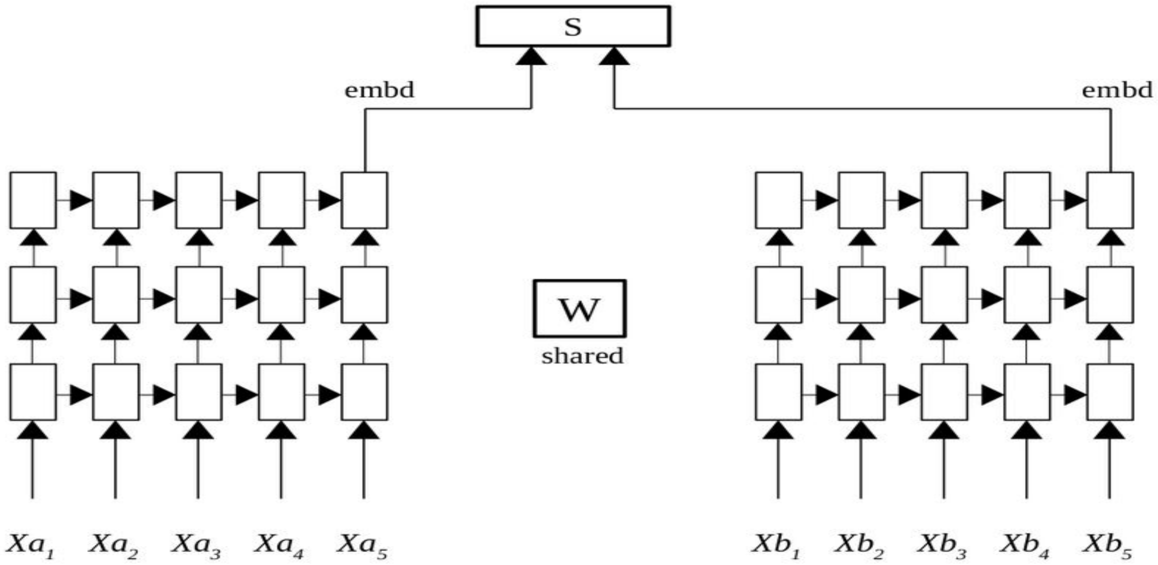
Output gate

1. The output gate is expressed as:

$$o = \sigma(bo+xtUo+ht-1Vo)$$

final output of the cell , with the *tanh* squashing, can expressed as
$ht =tanh(st) \circ o$

Architecture of Siamese LSTM

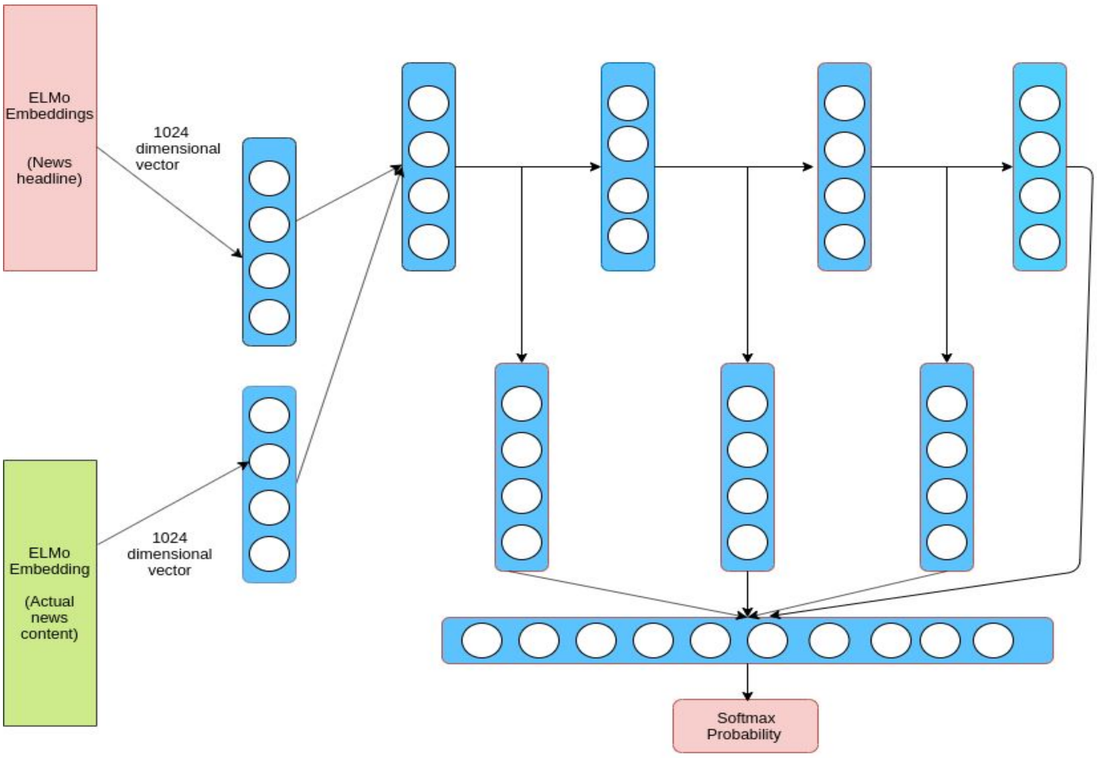


Xa - HEADLINE

Xb - NEWS CONTENT

S- LATER LAYERS (FC layer, output layer)

VNet the proposed architecture



VNet

In the Vnet (the proposed architecture) we used transfer learning model ELMo (Embedding's from Language Models).for embedding the news headline and the actual content

Transfer learning

Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task. It is a popular approach in deep learning where pre-trained models are used as the starting point on computer vision and natural language processing tasks given the vast compute and time resources required to develop neural network models on these problems and from the huge jumps in skill that they provide on related problems.

ELMo embedding's:

Unlike word embedding's such as word2vec and Glove, the ELMo vector assigned to a token or word is actually a function of the entire sentence containing that word. Therefore, the same word can have different word vectors under different contexts.

Result analysis

Conclusion and Future Work

References

1. Nir Kshetri, Jeffrey Voas, “The Economics of Fake News”, IEEE, IT Professional, 2017, Volume: 19, Issue: 6, Pages: 8 – 12

2. Roger Musson, “Views: The frost report: fake news is nothing new”, 2017, IEEE, Astronomy & Geophysics, Volume: 58, Issue: 3, Pages: 3.10 - 3.10

3. Hal Berghel, “Oh, What a Tangled Web: Russian Hacking, Fake News, and the 2016 US Presidential Election”, IEEE, Computer, 2017, Volume: 50, Issue: 9, Pages: 87 - 91

4. Hal Berghel, “Alt-News and Post-Truths in the "Fake News" Era”, IEEE, Computer, 2017, Volume: 50, Issue: 4, Pages: 110 – 114

5. Hal Berghel, “Lies, Damn Lies, and Fake News”, IEEE, Computer, 2017, Volume: 50, Issue: 2, Pages: 80 - 85

6. Sneha Singhania, Nigel Fernandez, “3HAN: A Deep Neural Network for Fake News Detection”, 2017, Neural Information Processing: 24th International Conference, ICONIP 2017, Guangzhou, China

7. Shashank Gupta, Raghuveer Thirukovalluru, Manjira Sinha, Sandya Mannarswamy, “CIMTDetect: A Community Infused Matrix-Tensor Coupled Factorization Based Method for Fake News Detection”, 2018, IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)

8. Stefan Helmstetter, Heiko Paulheim, “Weakly Supervised Learning for Fake News Detection on Twitter”, 2018, IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)

9. Akshay Jain, Amey Kasbe, “Fake News Detection”, 2018, IEEE International Students' Conference on Electrical, Electronics and Computer Sciences

10. Chandra Mouli Madhav Kotteti, Xishuang Dong, Na Li, Lijun Qian, “Fake News Detection Enhancement with Data Imputation”, 2018, IEEE 16th Int. Conf. on Dependable, Autonomic & Secure Comp., 16th Int. Conf. on Pervasive Intelligence & Comp., 4th Int. Conf. on Big Data Intelligence & Comp., and 3rd Cyber Sci. & Tech. Cong.

11. Shaban Shabani, Maria Sokhn, “Hybrid Machine-Crowd Approach for Fake News Detection”, 2018 IEEE 4th International Conference on Collaboration and Internet Computing

12. Katherine Clayton, Spencer Blair, Jonathan A. Busam, Samuel Forstner, John Glance, Guy Green, Anna Kawata, Akhila Kovvuri, Jonathan Martin, Evan Morgan, Morgan Sandhu, and Rachel Sang, Rachel Scholz Bright, Austin T. Welch, Andrew G. Wolf, Amanda Zhou, Brendan Nyhan, “Real Solutions for Fake News? Measuring the Efectiveness of General Warnings and Fact Check Tags in Reducing Belief in False Stories on Social Media”, 2019, Springer Science, Business Media, LLC, part of Springer Nature

13. Costel-Sergiu Atodiresei, Alexandru Tănăselea, Adrian Iftene , “Identifying Fake News and Fake Users on Twitter”, 2018, International Conference on Knowledge Based and Intelligent Information and Engineering Systems,Belgrade, Serbia

14. Monther Aldwairi, Ali Alwahedi, “Detecting Fake News in Social Media Networks”, 2018, the 9th International Conference on Emerging Ubiquitous Systems and Pervasive Networks

15. Sherry Girgis, Eslam Amer, Mahmoud Gadallah, “Deep Learning Algorithms for Detecting Fake News in Online Text”, 2018, 13th International Conference on Computer Engineering and Systems (ICCES)

16. Herley Shaori Al-Ash, Wahyu Catur Wibowo, “Fake News Identification Characteristics Using Named Entity Recognition and Phrase Detection”, 2018, 10th International Conference on Information Technology and Electrical Engineering (ICITEE)

17. Marco L. Della Vedova, Eugenio Tacchini, Stefano Moret, Gabriele Ballarin, “Automatic Online Fake News Detection Combining Content and Social Signals”, 2018, Procedding of the 22nd Conference of Fruct Associtio.

18. M.Vasuki, J. Arthi, K. Kayalvizhi,” Decision Making Using Sentiment Analysisfrom Twitter”, 2014, International Journal of Innovative Research in Computerand Communication Engineering, Vol. 2, Issue 12

19. Hassan Saif, YulanHe, Miriam Fernandez, and Harith Alani," Semantic Patterns for Sentiment Analysis of Twitter", 2014, ISWC Part II, LNCS 8797, pp. 324–340
20. SanthiChinthala, Ramesh Mande, SuneethaManne, and SindhuraVemuri," Sentiment Analysis on Twitter Streaming Data", 2015, Emerging ICT for Bridging the Future – Volume 1, pp- 470- 481
21. Hassan Saif, Yulan He, and Harith Alani," Semantic Sentiment Analysis of Twitter", 2012, ISWC Part I, LNCS 7649, pp. 508–524
22. Syed Akib Anwar Hridoy, M. TahmidEkram, Mohammad Samiul Islam, Faysal Ahmed and Rashedur M. Rahman," Localized twitter opinion mining usingsentiment analysis", 2015, Anwar Hridoy et al. Decis. Analysis, vol. 4, issue 65 pp- 015-016
23. Xing Fang and Justin Zhan," Sentiment analysis using product review data", 2015, Springer, volume 5 issue 7, pp- 015-020
24. Khaled Ahmed, Neamat El Tazi, Ahmad HanyHossny," Sentiment Analysis Over Social Networks: AnOverview", 2015, IEEE, vol. 9, iss. 8, pp- 97-110
25. Aldo Hernández, Victor Sanchez, Gabriel Sánchez, Héctor Pérez,Jesús Olivares, Karina Toscano, Mariko Nakano and Victor Martinez," Security Attack Prediction Based onUser Sentiment Analysis of Twitter Data", 2016, IEEE, vol. 56, pp.45
26. Dan Cao, LiutongXu. Analysis of Complex Network Methods for Extractive Automatic Text Summarization.2016 2nd IEEE International Conference on Computer and Communications, vol. 9, iss. 8, pp- 97-110, 2016.
27. RasimAlguliyev, RamizAliguliyev, NijatIsazade. A Sentence Selection Model and HLO Algorithm for Extractive Text Summarization, IEEE, vol. 9, iss. 8, pp- 97-110, 2016.
28. NarendraAndhale, L.A. Bewoor. An Overview of Text Summarization Techniques. IEEE, vol. 9, iss. 8, pp- 97-110, 2016.

Reference: https://arxiv.org/abs/1802.05365

- After giving news headline as one input to separate embedding layer and news content as one as another input to the separate embedding layer the output of the both layers will be merged (shown in diagram)
- The later layers that follows unique architecture that is completely different from the usual fully connected neural network
- Each layer that may have important thing to tell but will be missed when we transfer the output to the later layer when we follow linear structure
- But the VNet capture the important features and merged them together before the output layer