DEGREE PROJECT IN TECHNOLOGY,
FIRST CYCLE, 15 CREDITS
STOCKHOLM, SWEDEN 2016

Detecting Trends on Twitter

The Effect of Unsupervised Pre-Training

SANDRA BÄCKSTRÖM

JOHAN FREDIN HASLUM

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF COMPUTER SCIENCE AND COMMUNICATION

KTH Computer Science
and Communication

Detecting Trends on Twitter

The Effect of Unsupervised Pre-Training

SANDRA BÄCKSTRÖM
JOHAN FREDIN HASLUM

Degree Project in Computer Science, DD143X

Supervisor: Pawel Herman

Examiner: Örjan Ekeberg

Royal Institute of Technology

Stockholm, Sweden 2016

KTH Computer Science
and Communication

Hitta Twittertrender

Effekten av oövervakad förträning

SANDRA BÄCKSTRÖM
JOHAN FREDIN HASLUM

Examensarbete inom datalogi, grundnivå, DD143X

Handledare: Pawel Herman

Examinator: Örjan Ekeberg

Kungliga Tekniska Högskolan

Stockholm, Sverige 2016

Abstract

Unsupervised pre-training has recently emerged as a method for initializing supervised machine learning methods. Foremost it has been applied to artificial neural networks (ANN). Previous work has found unsupervised pre-training to increase accuracy and be an effective method of initialization for ANNs[2].

This report studies the effect of unsupervised pre-training when detecting Twitter trends. A Twitter trend is defined as a topic gaining popularity.

Previous work has studied several machine learning methods to analyse Twitter trends. However, this thesis studies the efficiency of using a multi-layer perceptron classifier (MLPC) with and without Bernoulli restricted Boltzmann machine (BRBM) as an unsupervised pre-training method. Two relevant factors studied are the number of hidden layers in the MLPC and the size of the available dataset for training the methods.

This thesis has implemented a MLPC that can detect trends at an accuracy of 85%. However, the experiments conducted to test the effect of unsupervised pre-training were inconclusive. No benefit could be concluded when using BRBM pre-training for the Twitter time series data.

Referat

Oövervakade förträning (OF) är ett område inom maskininlärning som används för att initialisera övervakade metoder. Tidigare studier har visat på att OF har varit en effektiv metod för att initialisera artificiella neurala nätverk (ANN). Denna initialiseringsmetod har haft positiv inverkan på den övervakade metodens precision[2].

Denna rapport studerar OFs påverkan när en övervakad metod används för att hitta trender i ett Twitterdataset. En Twittertrend definieras av ett ämnes ökning i popularitet.

Tidigare har flera studier analyserat olika maksininlärnings metoders applicerbarhet på Twitter tidsserie data. Dock har ingen studie fokuserat på användningen av OF och ANNs på denna typ av data, något denna rapport ämnar göra. Effekten av att kombinera en Bernoulli restricted Boltzmann machine (BRBM) med en multi-layer perceptron classifier (MLPC) jämförs med en modell vilken endast använder MLCP. Två relevanta faktorer som också studeras är hur storleken på datasetet som tränar metoderna påverkar deras relativa precision, samt hur antalet gömda lager i MLPC påverkar respektive metod.

Denna studie har implementerat en MLPC som kan hitta trender med 85% säkerhet. Dock har experimenten för OF inte lyckats bekräfta någon fördel med OF vid tillämpning på Twitter tidsserie data.

Contents

1 Introduction

In today's society most people communicate through social media every day, ranging from ordinary people to politicians and corporations. Social media forums can be utilized to reach extensive crowds around the world if used strategically.

An example of such a social media is Twitter, which is a widely used platform with a large worldwide user base of 310 million monthly active users. When a topic or an event becomes trending on Twitter, it reaches out to an international crowd of 1 billion unique people per month[17].

Twitter is a social media platform where users post short messages. Other users can then like and retweet their tweets, which means that they share the other person's tweet with their followers. Users can follow each other and no mutual following is required[19]. This permits certain users to have millions of followers, whereas other people only have a dozen, if any at all. To enable users to find tweets related to a certain subject, a short sequence of characters starting with a "#" is used. These sequence of characters are called hashtags and allow users to participate with and view all related tweets.

In 2015, the trends ranged from world wide events and tragedies to social phenomenons[21]. These trends included #FIFAWWC (FIFA World Cup) and #ParisAttacks as well as #TheDress[1]. When a hashtag becomes trending on Twitter it happens fast, the Paris attacks is an example of how a previously non-existent hashtag's activity suddenly goes through the roof[16]. However, a group of hashtags can behave differently and still all be classified as trends.

Finding and detecting trends on Twitter is not only useful for companies or people that wish to pick up on the latest trends, but also to Twitter themselves as a company. Twitter makes approximately 85% of its revenue based on selling advertisements[5]. Twitter has developed an algorithm that finds trending hashtags, both global and local trends[18]. Then it is possible for Twitter to strategically use the trending hashtags to place its advertisements such that it reaches out to a large crowd.

Previous work has been done in the area of trend detection in Twitter's feed. In

[1]The dress was a viral phenomenon when people saw a photo of a dress and saw it either as being colored blue and black or white and gold.

2012, Nikolov et al. developed an algorithm that found trends in the Twitter data feed, before Twitter's algorithms was able to do so. The research team was able to do this by experimenting with different machine learning methods and arrive at an algorithm that used the frequencies of hashtags to predict an upcoming trend[6].

Machine learning is a relatively new area within computer science that uses data-driven learning to train programs without literally programming them to do so. Methods within machine learning are used for image recognition and self-driving cars among others[13]. In the context of detecting trends, machine learning algorithms are trained to look at existing Twitter data and then be able to find trends in new data.

Within machine learning, there are two categories of methods. The first one is supervised learning that trains the algorithm by providing a training set of input data and the expected outcome. The second one is unsupervised learning that takes in a training set of input data, but no data of the expected outcome. This makes unsupervised learning different, as the algorithm has to find a structure and patterns in the given data without knowing what the output is supposed to be[13].

Unsupervised learning methods can also be helpful in other contexts, such as unsupervised pre-training. Supervised training methods aim to find a global minimum (fitting the problem with the smallest error), but the global minimum is not guaranteed to be reached every time. Unsupervised pre-training can help initialize the supervised method such that when it begins, it is more prone to reach the global minimum[2].

1.1 Problem Definition

This thesis aims to explore the effects unsupervised pre-training has on supervised training methods in the case of twitter trend detection.Two methods will be used in experiments from Twitter data to see if the unsupervised pre-training helps improve the accuracy of a supervised training classification method. The methods' accuracy will be examined based on the true positive rate (TPR) and the overall accuracy. The methods will use time series data based on the frequency of a certain hashtag in a set time span.

The supervised method that will be used is a Multi-layer Perceptron Classifier

(MLPC), which is a form of artificial neural network (ANN) using supervised training. The method that will be used to pre-train the MLPC is an unsupervised ANN model called Bernoulli Restricted Boltzmann Machine (BRBM).

Relevant factors such as the training set size and number of hidden layers in the MLPC will also be examined. The research question for this thesis follows:

How does unsupervised pre-training affect the performance of a Multi-layer Perceptron Classifier(MLPC) in detecting trends on Twitter? How does the performance depend on the size of available training data and the number of hidden layers in the MLPC?

1.2 Scope and Constraints

This thesis will only study at trends that are represented by hashtags and their activity in the Twitter feed. As Twitter provides the current trends continuously, it is possible to calculate the accuracy of the methods when detecting trends. This excludes trends that are not represented by a hashtag, such as words or topics written in the text of a tweet without the "#" symbol.

Even though only the hashtags of a tweet will be analysed, other factors and properties of a tweet may be useful for trend detection. As seen later in the background, methods in similar previous work has looked at several properties of tweets when analysing trends. However, the time series data of the frequencies is of most importance and is what will be studied, excluding any other properties of a tweet.

The accuracy of any method used may be impacted by the size of the training set. Therefore this thesis will examine the chosen methods using training sets of different sizes. However, the maximum size of the training set will be limited to a gathered sample of the Twitter feed. Since this thesis is applying ANN methods, the effect of the number of hidden layers will also be examined. No other factors that may affect the accuracy of a Twitter trend detection algorithm will be accounted for in this work.

This thesis will examine ANN models MLPC and BRBM and no other methods.

1.3 Thesis Overview

In the following second section, previous work related to this thesis and relevant background information is presented. The third section presents the procedure of the thesis. Motivation of approach and experiments, as well as explanation of the dataset and assumptions made are all addressed here. The fourth section presents the results acquired by experiments previously explained. The results are discussed and analysed in the fifth section. The sixth section summarizes the discussion and presents conclusions.

2 Background

2.1 Twitter

Twitter is an online social network in the shape of a micro-blog platform, which first started in 2006. It allows users to write posts no longer than 140 characters, called tweets. A user can choose to broadcast his or her tweets privately amongst selected followers or publicly for all of twitter to see. In March 2016 about 500 million tweets were produced per day by users ranging from politicians, celebrities and other public figures. This huge amount of data makes it hard to find related tweets, to solve this Twitter uses hashtags[20]. This makes it possible for users to explore tweets containing a particular hashtag. When a hashtag is tweeted more frequently, it may be classified as trending by Twitter's own trend detection algorithm[18].

2.2 Trends

Twitter defines trends as a topic that is emerging in popularity. It thereby excludes topics that constantly have a relative high frequency for an extended period of time[18].

Therefore a trend is the equivalent of an emerging hashtag in this context, or as Naaman et al. proposes, an activity burst (increase in frequency). The question of defining a trend then comes down to how strong the burst has to be for the developed algorithm to catch the emerging topic. Two different trending topics may have activities increase at a different rate. Therefore the activity burst that defines a trend must be within a specified range[10]. However this only applies to trends above a certain activity level, as low activity trends may change rapidly and errors may occur.

2.3 Related Work

In this section we will discuss the previous work done focusing on twitter trends, detection using supervised learning and previous work on unsupervised pre-training.

2.3.1 Supervised Learning

Supervised learning is a technique within machine learning for which an algorithm is trained using a training set with a known outcome in order to use the algorithm to predict the outcome of another set of data. The training set consists of input data and corresponding outcome. Larger training set usually result in greater accuracy, as there is more data to train the algorithm with and thereby refine it[8]. There are several methods within supervised learning that have been used in previous work in the area of Twitter trends.

The most relevant work done on the subject of Twitter trend prediction was done focusing on nonparametric time series classification, using a nearest neighbor model to solve the problem. It evaluates each hashtag by creating a frequency time series and comparing it to previously trending hashtags in the training set. This approach was able to predict Twitter trends 79% of the time[6]. However, this method was only applied to a small data set of about 1000 hashtags. The small size of the training set posts questions regarding if a larger training set could further improve the accuracy of such an algorithm.

Zongyang et al. proposes methods to predict the future popularity of hashtags. The problem is approached as a classification task and five different methods are used (Naïve bayes, k-nearest neighbors, decision trees, support vector machines and logistic regression). These are applied to two sets of features, content and contextual. Contextual features include user's followers and how many times a tweet have been retweeted, and this type of data can be used to create a social graph. Content features focus on the actual text (including hashtags) of the tweet. The use of contextual features was proven to be the most effective option[22].

Brennan et al. explores the possibilities of suggesting current trends based on only tweet content while excluding hashtags. Similarly to previous work above, they divide tweet data into two parts: content and context. Foremost focusing on the word frequency (content) and then surrounding factors (contextual), the study includes twitter followers, retweets and other contextual data. A Naïve Bayes classifier, which has been slightly modified, is applied to their data set of around 50,000 tweets. The use of this classification algorithm is shown to be most effective when combining both content and contextual data, successfully classifying tweets

as containing a trend or not with an accuracy of 75% - 85%[4].

This suggest that not only the hashtags contained in a tweet can be used to predict or detect trends, but also the relational data surrounding a tweet. This is a fact that can be useful in combination with previously mentioned methods. However the relatively small data set and the computationally expensive tasks raises the question whether their methods are effective in larger scale problems.

A study by Kong et al. can be seen as a summary of the above mentioned studies. It uses contextual, content and time series data to examine the importance of each feature set. Their study shows that the most important factor when predicting trends in hashtags is the time series data. The longer a hashtag had been studied, and therefore more time series data had been collected, the prediction accuracy significantly increased. The accuracy ranged from 5.6% in newly discovered hashtags (small amount of time series data), but reached values of up to 72% for hashtags closer to its burst (large amount of time series data)[3].

To summarize, the time series data is of great importance when analysing trends on Twitter. Other factors, content and contextual, are refinement methods that does not perform nearly as well as time series on their own.

2.3.2 Unsupervised Pre-Training

Unsupervised learning is similar to supervised learning methods. However, the training set consists of input data only, without any known outcome. Therefore the unsupervised learning algorithms function differently as they try to find structure and underlying patterns in the input data[9].

No previous work has been done in the area of using unsupervised learning on Twitter trends or similar topics. However, research has been done in the last couple of years on unsupervised methods used in unsupervised pre-training, mostly focusing on ANN[2].

Unsupervised learning methods can be used for unsupervised pre-training for initilizing supervised learning methods. This type of pre-training mostly focuses on how it can be used to improve gradient descent in ANN. Unsupervised pre-training helps the supervised learning algorithm to find the global minimum rather than a local minimum, by initializing the variables in the supervised learning model using

an unsupervised learning method. More specifically it has been used to improve the choice of initial variables when training these networks, rather than initializing the variables randomly. Research proves that it does not only perform better than using random initialization, but also better than refined initialization methods[1].

The same study suggests that unsupervised pre-training is preferred in situations where there are a lot of local maximums and minimums. It must not always result in a better prediction accuracy, but algorithms using unsupervised pre-training generally performs better than random initialization of the supervised learning model[1].

The effects of unsupervised pre-training has been studied further by Erhan et al. concludes that unsupervised pre-training is more efficient on ANN structures with several hidden layers rather than few. An ANN with only one hidden layer using unsupervised pre-training is shown to perform worse than the same algorithm without unsupervised pre-training. The study also shows that a 3-layer algorithm without unsupervised pre-training performs worse than an equivalent 1-layer algorithm also without unsupervised pre-training[2]. This suggests that unsupervised pre-training is neither necessary nor beneficial for 1-layer deep learning algorithms. However, for several layer ANN algorithms, unsupervised pre-training helps improve the accuracy of the deep learning algorithm.

Erhan et al. also examines the effect of large training sets for two algorithms where one is using unsupervised pre-training. Using large training sets, popular belief is that unsupervised pre-training might not be necessary as the algorithm is trained to greater precision due to the larger training set. However, study states that the benefit of unsupervised pre-training does not stop as the training set increases. In fact, pre-training enables the algorithm to take advantage of the larger training set[2].

To summarize, unsupervised pre-training does help improve the accuracy of supervised learning methods in certain cases. Mainly so by finding the optimum set of initialization parameters for the supervised learning algorithm. However, this is not suitable for all cases, especially when an ANN model has few layers.

2.4 Artificial Neural Networks (ANN)

Most studies examining unsupervised pre-training examine deep architectures, such as ANNs. These also state that deep architectures are needed for unsupervised pre-training to yield result and that the deep architectures need the unsupervised pre-training for best result[2]. There are methods explicitly recommended for unsupervised pre-training, one of which is Restricted Boltzmann Machines (RBM) that is an ANN used for unsupervised pre-training[12].

ANN is a type of deep learning architecture. What this essentially means is that there are several layers of neurons in a network that are trained to be activated in order to predict an outcome of given input data. The structure below in figure 1 is an example of an ANN with two hidden layers with respectively four and three neurons each. As previously mentioned in the background, several hidden layers is required to obtain the effects of unsupervised pre-training.

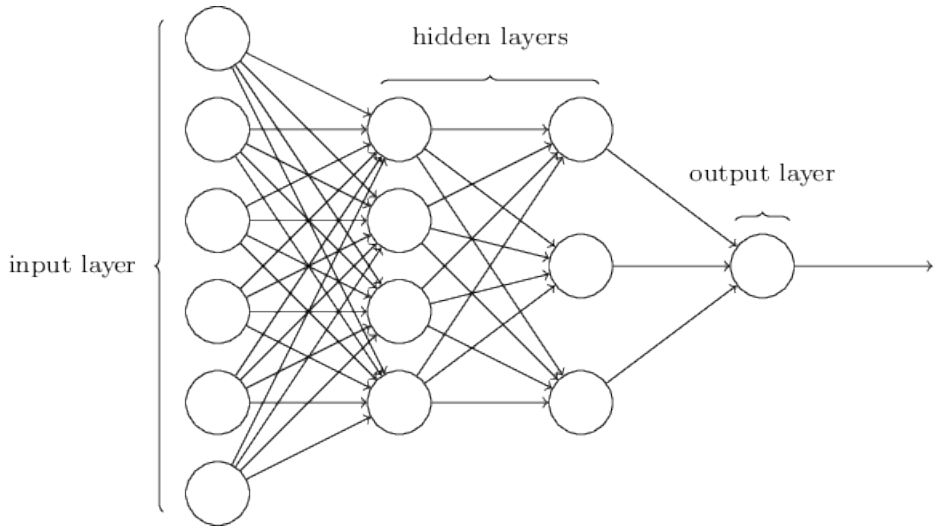

Figure 1: An ANN with 2 hidden layers with 4 and 3 units respectively[14].

As seen previously in the background, time series data is the most effective type of data to use when detecting trends. Time series are also studied in several different areas, one of them is the financial market. For quite obvious reasons, people are interested to predict the changes in the stock market. ANN have been successful in analysing and predicting financial time series data. Although this thesis does not concern the financial market, the fact that ANNs have shown

promising results in time series data is of importance[15]. To summarize, ANNs are beneficial for time series data analysis, which makes it relevant for the scope of this thesis. Furthermore, since ANNs benefit from unsupervised pre-training, ANN models is of importance for the scope of this thesis.

2.4.1 Restricted Boltzmann Machines (RBM)

RBMs are based on unsupervised learning using a probabilistic model, which is a form of unsupervised ANN. The RBM learns the probability distribution of its input data and by making groups of features. It is used to detect noticeable regularities in the input training data. This has shown to be a good structure for initializing ANNs and has therefore become a popular method for unsupervised pre-training of deep architectures[12].

The Bernoulli RBM (BRBM) which is a common type of RBM using data in the [0, 1]range. The BRBM takes advantage of a bipartite graph structure, thus eliminating connections between neurons in the same layer. Each neuron is connected to all neurons in the next layer and have the following probability to be activated:

$$P(v_i = 1|\mathbf{h}) = \sigma(\sum_j \omega_{ij} h_j + b_i) \tag{1}$$

$$P(h_i = 1|\mathbf{v}) = \sigma(\sum_i \omega_{ij} v_i + c_j) \tag{2}$$

Sigma (σ) is the sigmoid function, and the variables in the sum is: v_i a neuron in the visible layer and h_i in the hidden, w_{ij} the weight of the connection between h_i and v_j, b_i and c_j being the corresponding bias. These are fitted by the use of Stochastic Maximum Likelihood (SML) learning. Using small batches of data, the gradient is calculated according to:

$$logP(v) = log \sum_h e^{-E(\nu,h)} - log \sum_{x,y} e^{-E(x,y)} \tag{3}$$

E(v,h) is the sum of the weights over the all adjacent neurons[12]. BRBM's use of SML has been proven the most efficient when training RBMs, furthermore the

SML algorithm ensures that the data representation is maintained on the same scale as the input data, which is important if the BRBM is to be used for pre-training[12].

2.4.2 Multi-layer Perceptron (MLP)

A multi-layer perceptron (MLP) is a supervised learning model based on the ANN structure. The MLP that is used for classification problems (MLPC) is trained using backpropagation. The backpropagation training algorithm uses gradient descent to minimize the loss function in the network. MLPs can be used for nonlinear models, which is beneficial for the use of time series data[11].

The MLP is sensitive for different scaling in input and is recommended to implement with scaled input, similar to how the BRBM uses values in range [0, 1]. The MLP work well with the BRBM as the weights from the trained BRBM can be transferred to the MLPC due to their similar structure[11].

3 Method

The process of this thesis started with background research, followed by data collection. Once the data was collected and formatted, it was possible to start testing the two ANN methods to acquire results.

3.1 Data Collection

To be able to train the methods to detect trending hashtags, Twitter data must be gathered. Twitter provides access to streaming live data, which allows a portion of the Twitter feed to be collected. In order to implement supervised learning, Twitter's trend data must also be gathered.

The data is gathered by streaming the live feed of tweets and trends into SQL databases. The format of the gathered data and the database structure are demonstrated in Appendix A.

A total of 1,4 million tweets were collected over a timespan of 60 hours. Since only hashtag frequency will be analysed, only tweets containing hashtags are gathered. The gathered data gives a total frequency of 400 tweets per minute. Considering that the entire Twitter feed consists of 500 million tweets per 24 hours, the data gathered is only about 1‰ of the actual Twitter feed[7].

The reason for only collecting the content of the tweet and a timestamp is to enable use of time series data formatting, as research done in the background suggests this to be the most efficient way to detect trends on Twitter. Even though contextual data (e.g. creating social network graphs to help detect trends) would be interesting to analyze, this is not done as Twitter provides limited access to contextual data.

3.2 Data Formatting

In order to use the data gathered in the MLPC and BRBM, the data must be formatted. First of all, the methods require input data X and output data Y, although the unsupervised methods only require input data X. That creates two tasks, create the X and the Y data.

Since the gathered data is based on content (hashtags) and timestamps for each tweet, the data is constructed in tables of sequential data. The X data is constructed as a table with columns consisting of time slots of five minutes. Every row represents a hashtag and the columns are filled with it's frequency (number of tweets with that hashtag in that time slot) during each time slot.

The Y data table has one column, each row in it matching the corresponding hashtag in the X data. The value in this Y data table states whether the hashtag trended sometime during any of the timeslots. An example of these data tables can be seen in Appendix B. This type of data formatting was chosen essentially since it well represents the change in frequency for hashtags, which further is what characterizes a trend.

The data gathered ranges from tweets that have a low frequency, e.g. is only detected once in one time slot and has a frequency of 0 in the other time slots. To narrow down the data collected, all rows in the X data table need to have at least one time slot in which it has a higher frequency than 2. This restriction is imposed on the data to focus on the hashtags that are more likely to be a trend than not. This lower limit of a frequency of 2 is chosen to eliminate hashtags written once by a single person. This lower bound was chosen due to the limited access to the Twitter feed.

Since the BRBM only works with data in the [0,1] range, the data had to be normalised to meet that criteria. Normalisation is further known to increase accuracy in various machine learning methods. Therefore the X data is normalised such that every value in the table is in the range [0, 1], but still has the same relative value. Further, the y vector values are binary values 0 or 1, the 1 stating that the hashtag data represents the time series data of a trend and 0 not representing a trend. Since the Y vector has two possible values, this is a classification problem.

3.3 Experiments

In order to see how the unsupervised pre-training may affect the supervised training method, there are several tests that need to run. These tests are based on the two methods, MLPC and MLPC pre-trained by BRBM, and the methods are created and tested by accessing a library called Sci-Kit Learn. The two relevant factors,

size of the training set and the number of hidden layers used in the MLPC will also be examined.

3.3.1 Sci-Kit Learn

It is noteworthy that the tests are carried out using a open source library called Sci-Kit Learn. This library has developed methods amongst machine learning and algorithms that can be implemented and adjusted. This library is used as it has the built in methods of the MLPC and BRBM. The library is mentioned in an article in the Journal of Machine Learning where it states that Sci-Kit Learn has reliable and useful implementation of machine learning methods[15].

3.3.2 Experiments to be Conducted

Brute force style testing has to be done to find the optimal initializations of both the MLPC and BRBM. Both methods can use different algorithms and learning rates, among many other initialization factors found in the Sci-Kit Learn library. This testing is done for each method to optimize performance on the data before conducting the experiments.

The first test aims to find the optimal number of hidden layers in the MLPC. This test has to be conducted both using the MLPC and the MLPC pre-trained by BRBM. The other experiments will use the optimal number of hidden layers from this experiment.

Another factor that is to be analyzed is the size of the training set. By varying the size of the training set on both the MLPC and the MLPC pre-trained with BRBM, it may be possible to determine how the size of the training set affects the accuracy of the methods and if they behave differently.

The final test intends to compare the accuracy of the MLPC against the MLPC pre-trained using BRBM. The most efficient initializations of both the MLPC and BRBM are used for best results, including the results of the previously mentioned experiments.

3.3.3 Evaluation

When conducting the experiments above, it is interesting to evaluate the methods by the overall accuracy. Another dimension of evaluation can also be added by looking at the TPR. The TPR and overall accuracy can be calculated:

$$TPR = \frac{TP}{TP + FN} \tag{4}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{5}$$

In the equations above the TP stands for true positive (number of true positive predictions), FN for false negative, TN for true negative and FP for false positive. TP, FN, TN and FP all togehter stand for all of the predictions made by al algorithm.

In the nature of this thesis it may be more important to detect hashtags that actually become trends, although this depends on what algorithm behavior is preferred. The different behaviors are either optimal overall accuracy or optimal TPR, as the two do not have to correlate perfectly. Therefore both the overall accuracy and the TPR will be examined when comparing the MLPC and the MLPC pretrained by BRBM.

4 Results

In all of the tests conducted, the result varies. Therefore the tests are run multiple times and shown usingbox plots to better represent the results and their variation.

4.1 Number of Hidden Layers in MLPC

The figures below demonstrate the correct classification rate and TPR for the MLPC with and without BRBM pre-training.

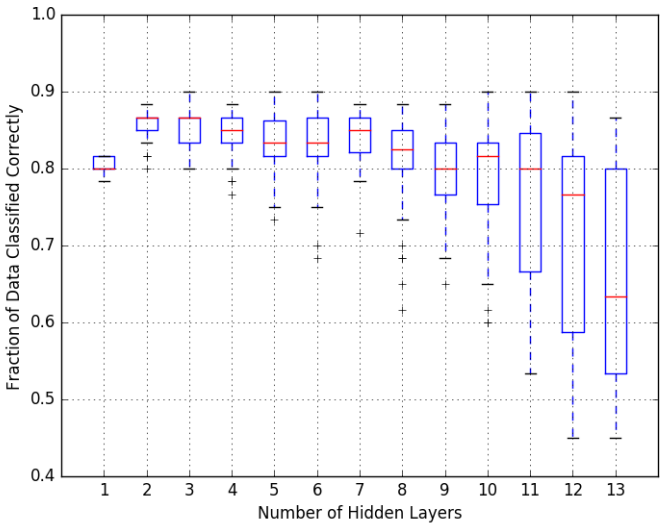

Figure 2: The figure represents correct classification rates when detecting trends in the dataset, using different number of hidden layers in a MLPC. Variation over the y-axis is represents the different outcomes when using the same number of hidden layers 50 times.

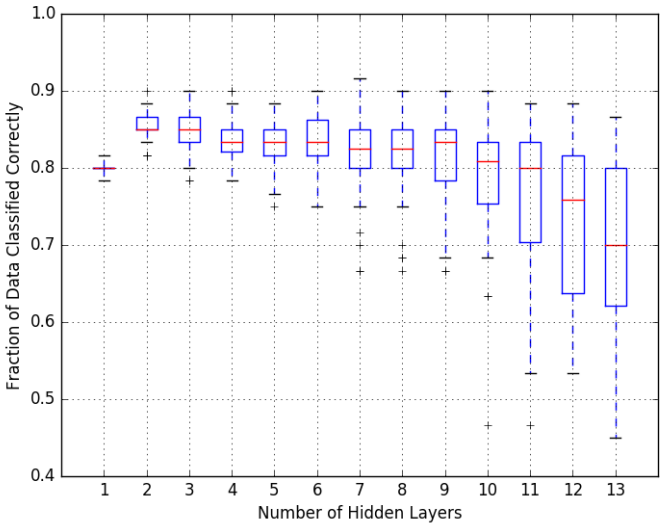

Figure 3: The figure represents correct classification rates when detecting trends in the dataset, using different number of hidden layers in a BRBM pre-trained MLPC. Variation over the y-axis is represents the different outcomes when using the same number of hidden layers 50 times.

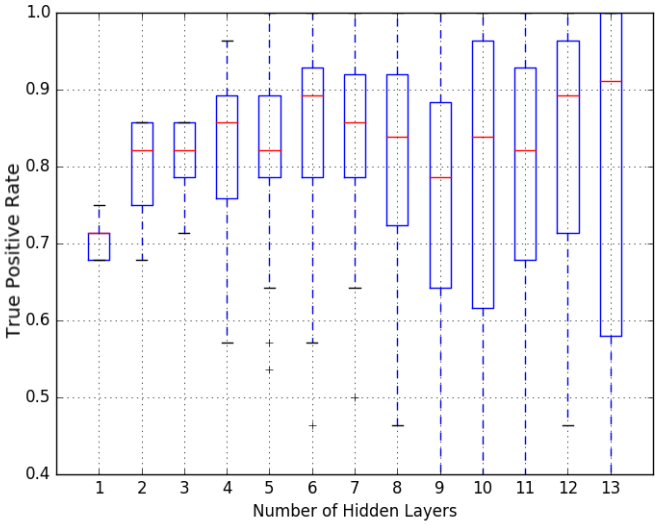

Figure 4: The figure represents TPRs when detecting trends in the dataset, using different number of hidden layers in a MLPC. Variation over the y-axis is represents the different outcomes when using the same number of hidden layers 50 times.

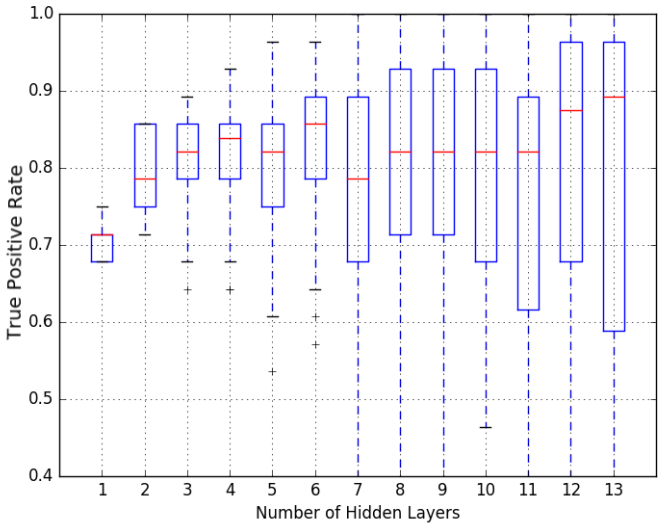

Figure 5: The figure represents TPRs when detecting trends in the dataset, using different number of hidden layers in a BRBM pre-trained MLPC. Variation over the y-axis is represents the different outcomes when using the same number of hidden layers 50 times.

By looking at the figures above, it is possible to see that the number of layers that yield the best results is using three. Further the figures also suggest that seven and above hidden layers decrease the overall accuracy and TPR. Generally the MLPC has similar accuracy with and without pre-training, only small differences. The TPR and overall accuracy is similarly low for both the pre-trained and non pre-trained MLPC.

4.2 Size of Training Set

The two figures below test if the MLPC with and without BRBM pre-training behave differently as the size of the training set differs. The x-axis on the two figures measure the fraction of the dataset used for testing, such that the sum of the testing and training set is always equal to 1.

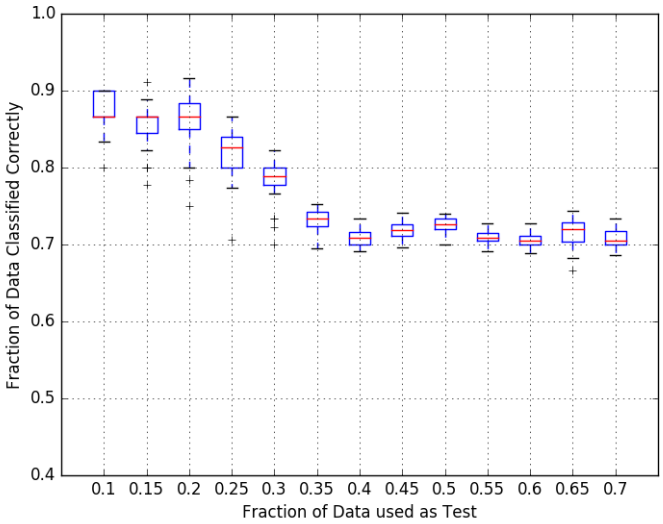

Figure 6: The figure above demonstrates how the accuracy of a MLPC differs as the size of the testing set increases (size of training set decreases).

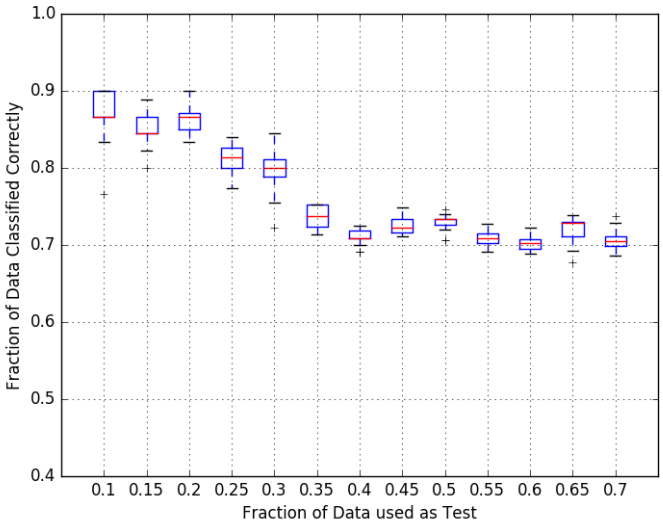

Figure 7: The figure above demonstrates how the accuracy of a BRBM pre-trained MLPC differs as the size of the testing set increases (size of training set decreases).

From the two figures above, it is possible to tell that pre-trained MLPC does not differ significantly from the MLPC when the trainingset size differs. However, it is possible to see a general trend that a large training set improves accuracy.

4.3 MLPC vs. BRBM Pre-Trained MLPC

The figure below takes into account the two previous results and does a test on the MLPC with and without BRBM pre-training.

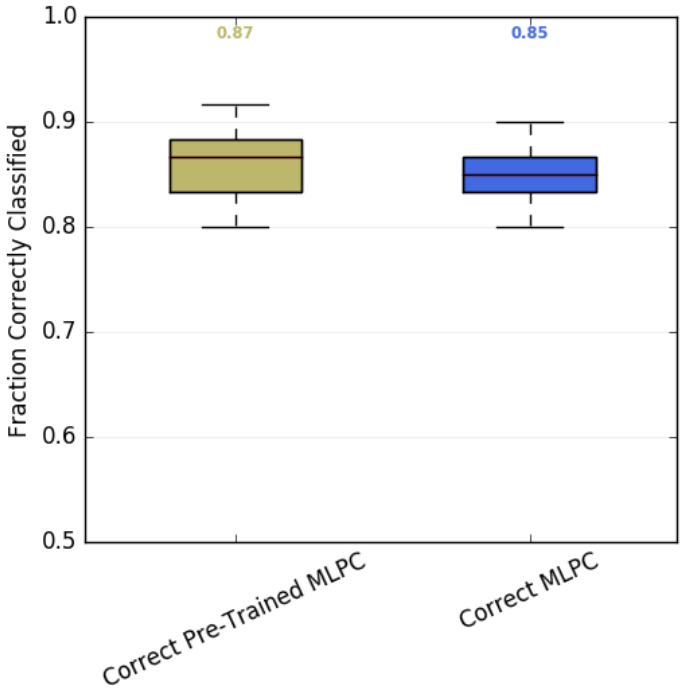

Figure 8: The figure represents the accuracy of a BRBM pre-trained MLPC and a MLPC. Variation over the y-axis is represents the different outcomes when running the methods 50 times.

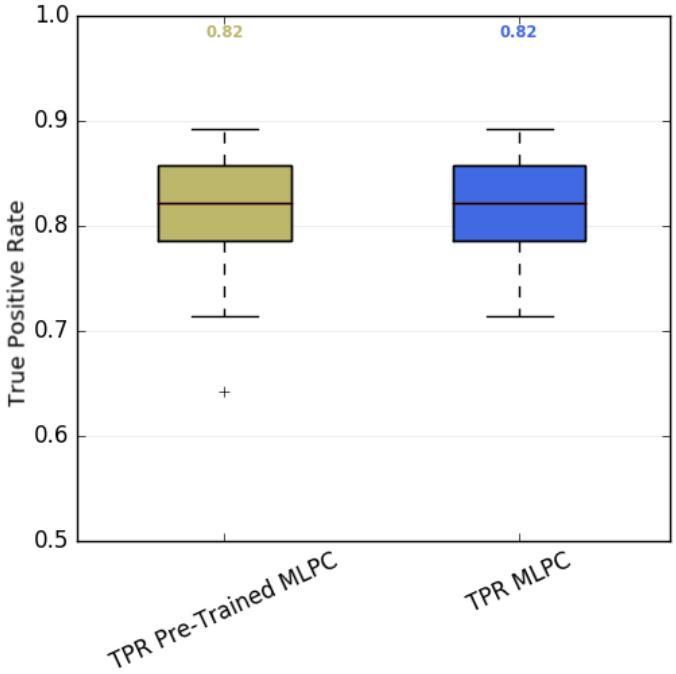

Figure 9: The figure represents the TPR of a BRBM pre-trained MLPC and a MLPC. Variation over the y-axis is represents the different outcomes when running the methods 50 times.

The two figures above suggests that a pre-trained MLPC has a slightly higher overall accuracy than a non pre-trained MLPC. However, the difference is only by 2%. Their TPRs are the same, despite the pre-trained MLPC having a higher overall accuracy.

5 Discussion

The results show that the BRBM pre-trained MLPC outperformed the MLPC on average by 2%. While this number suggests that pre-training does increase performance, the difference is small. In fact it is smaller than the range between the first and third quartile for both of the methods. Therefore the data does not show distinguishable enough results to confidently conclude that pre-training does improve the performance of a MLPC.

Furthermore the results of the optimal choice of hidden layers showed no tendency for one model to outperform the other. This suggests that there is no significant difference between MLPC with or without BRBM pre-training. This contradicts research presented in the background. There may be several reasons as to why pre-training has none to low effect on the MLPC. One reason may be that unsupervised pre-training is not beneficial for time series data, as no previous study has examined that combination.

For both the BRBM pre-trained MLPC and MLPC methods yielded results that correctly classified more than 70% of the data for all iterations using one to five layers. However when exceeding seven layers the spread of correct classification rates increase significantly. This seems to be the limit at which both the methods can no longer methodically converge towards a useful solution. Furthermore, Erhan et al. concluded that unsupervised pre-training has no effect on ANNs with one hidden layer. Similarly no effect was shown in this experiment as both methods performed identically using one hidden layer.

An interesting trend for the both the BRBM pre-trained MLPC and the MLPC is that the model have a lower true positive rate than correct classification rate. Since the data consists of about 50% trends, the methods therefore are more willing to predict trends as non-trends more frequently than the opposite. One possible reason for this is the possibility that non-trends often follow a more recognizable chain of events, while the frequencies for trends might vary more and therefore be harder to detect.

When comparing the accuracy between the BRBM pre-trained MLPC and MLPC they follow the same trend of improvement as the available training set increases in size. In other words, the size of the training data does not affect them

differently and neither method show any tendency to result in better outcome than the other when only a small data set is available. Therefore there is no obvious reason to prefer a pre-trained MLPC to counteract the negative effect of a small dataset, based on the results from this Twitter time series dataset.

When evaluating the performance of the methods, both correct classification and TPR are of interest. The trade-off between properly classifying all trends while not incorrectly classifying non-trends as trends is important. Depending on what the intended use of the method is, the importance of true positive rate vs. false positive rate may differ. The experiment uncovered that the TPR is not perfectly correlated with the correct classification rate. This means that when deciding how many layers to use in the MLPC, the optimal number may be different depending on the purpose of the trend detection. This thesis has valued both TPR and the overall accuracy.

Despite that the experiments show no advantage of unsupervised pre-training, the overall accuracy of both pre-trained and non pre-trained MLPC is high. Similarly to Brennan et al., who also carried out experiments in trend detection (although focusing on contextual data), both the experiments yielded an accuracy of 80-85%. Even though the unsupervised pre-training yield no apparent results, the MLPC performed well and was able to achieve similar accuracy as the methods used by Brennan et al.

5.1 Limitations

When conducting the experiments the dataset is one apparent limiting factor to consider. The stream that Twitter provides for developers is a sample of the entire stream. This means that the data that has been collected for the dataset is not the entire Twitter stream, in fact only 1‰ of it. When Twitter classifies trends it has access to the entire Twitter feed to make those decisions.

When running the tests, the data gathered is assumed to be an accurate representation of the Twitter stream. However, there is no guarantee that all relevant hashtags are represented equally in the gathered dataset. Therefore the lack of a complete dataset is a limitation that may affect the accuracy of the methods used. Despite this, the comparasion between MLPC and BRBM pre-trained

MLPC should be valid as both are used on the same dataset.

Furthermore, the lack of total access to the Twitter feed data, the resulting dataset was not as large as initially hoped for. As seen in the result section, as the size of the training set approaches the maximal amount of available training data, the accuracy has not converged. If there was a larger training set, the accuracy may converge, which suggests that the dataset used can be considered small. If further study is to be done in this subject, it would be beneficial and interesting to see how a large dataset (large enough for the accuracy to converge) may affect the results.

6 Conclusion

This thesis was able to implement a MLPC that achieved an accuracy of 85% when detecting trends in a Twitter dataset. This is on par with previous work studying Twitter trend detection. However, the study of the effects of unsupervised pre-training using BRBM on a MLPC was inconclusive. The unsupervised pre-training had neither a positive nor negative effect on the MLPC's performance regardless of how many hidden layers were used. Further, changing the size of the training set affected both methods equally, thus there is no clear advantage of pre-training a MLPC if the training set is small.

A limitation in this study is the possibly unrepresentative dataset. Therefore this thesis refrain from drawing any definite conclusion regarding the effect of unsupervised pre-training on Twitter time series data using a MLPC. A larger and more representative dataset is vital for further studies in this topic.

Bibliography

[1] Dumitru Erhan et al. "The Difficulty of Training Deep Architectures and the Effect of Unsupervised Pre-Training". In: *Journal of Machine Learning Research* 5 (2009).

[2] Dumitru Erhan et. al. "Why Does Unsupervised Pre-training Help Deep Learning?" In: *Journal of Machine Learning Research* 11 (Oct. 2010).

[3] Shoubin Kong et al. *Predicting Bursts and Popularity of Hashtags in Real-Time*. Dept. of CS&T, 2014. URL: http://www-personal.umich.edu/~qmei/pub/sigir2014-kong.pdf.

[4] Michael Brennan and Rachel Greenstadt. *Coalescing Twitter Trends: The Under-Utilization of Machine Learning in Social Media*. Tech. rep. Drexel University - Department of Computer Science, 2011.

[5] Pia Gadkari. *How does Twitter make money?* Business, 2013. URL: http://www.bbc.com/news/business-24397472.

[6] Stanislav Nikolov George H. Chen and Devavrat Shah. *A Latent Source Model for Nonparametric Time Series Classification*. URL: http://arxiv.org/pdf/1302.3639.pdf.

[7] InternetLiveStats.com. *Twitter Usage Statistics*. 2016. URL: http://www.internetlivestats.com/twitter-statistics/#trend.

[8] *Machine learning technique for building predictive models from known input and response data*. The MathWorks, Inc. URL: http://www.mathworks.com/discovery/supervised-learning.html.

[9] *Machine learning technique for finding hidden patterns or intrinsic structures in data*. Math Works, Inc., 2016. URL: http://www.mathworks.com/discovery/unsupervised-learning.html.

[10] Hila Becker Mor Naaman and Luis Gravano. *Hip and Trendy: Characterizing Emerging Trends on Twitter*. Tech. rep. Columbia University Computer Science, 2011.

[11] *Neural network models (supervised)*. scikit-learn developers, 2016. URL: http://scikit-learn.org/dev/modules/neural_networks_supervised.html#.

[12] *Neural network models (unsupervised)*. scikit-learn developers, 2016. URL: http://scikit-learn.org/stable/modules/neural_networks.html.

[13] Andrew Ng. *Machine Learning*. Online course. Stanford, 2016. URL: https://www.coursera.org/learn/machine-learning.

[14] Michael Nielsen. *Neural Networks and Deep Learning*. URL: %5C%5Chttp://neuralnetworksanddeeplearning.com/images/tikz11.png.

[15] Bogdan Oancea and Stefan Cristian Ciucu. *Time Series Forecasting Using Neural Networks*. URL: http://arxiv.org/pdf/1401.1333.pdf.

[16] Statweestics. *Tweets statistics for the hashtag 'parisattacks'*. Apr. 2016. URL: http://vps1.statweestics.com/.

[17] Inc. Twitter. *Company Facts*. 2016. URL: https://about.twitter.com/company.

[18] Inc. Twitter. *FAQs about trends on Twitter*. 2016. URL: https://support.twitter.com/articles/101125.

[19] Inc. Twitter. *Help Center - Getting Started With Twitter*. 2016. URL: https://support.twitter.com/articles/215585?lang=en.

[20] Inc. Twitter. *Using Hashtags on Twitter*. 2016. URL: https://support.twitter.com/articles/49309.

[21] Alexandra Valasek. *This #YearOnTwitter*. Dec. 2015. URL: https://blog.twitter.com/2015/this-yearontwitter.

[22] Aixin Sun Zongyang Ma and Gao Cong. “On Predicting the Popularity of Newly Emerging Hashtags in Twitter”. In:
Journal of the American Society for Information Science and Technology (2013). URL:http://www.ntu.edu.sg/home/axsun/paper/sun$_j asist13a.pdf$.

Appendix

Appendix A

Twitter Data Collected

The data gathered from Twitter's API are placed in two SQL databases demonstrated below. Also, the tables shows what properties are collected from each tweet and likewise for the trend data.

Tweet Property	Description
Id_str *primary key*	Unique ID for the tweet
created_at	Timestamp for when tweet was created (time and date)
hashtags	List of hashtags in tweet, empty if none
text	Text written in tweet

Table 1: The table shows and explains what properties of a tweet that is collected and placed in the SQL database. It also shows how the SQL database is structured.

Trend Property	Description
Time *primary key*	Time for current trends (time and date)
trends	List of all currently trending hashtags classified as trends by Twitter

Table 2: The table shows and explains what properties of the current trends are collected and placed in a SQL database.

Appendix B

Formatted Twitter Data

The two figures below gives an example of how the data is formatted and structured (no real data is used in the two figures below). The data is divided into two tables, X and Y. The X data is the input data handed to the MLPC and BRBM and the Y data is the output that the MLPC is trained to predict.

X Data

Hashtag/Timeslot	0	1	2	3	4
#1	1	1	1	0	10
#2	2	4	8	16	64
#3	1	2	16	64	512
#4	4	5	0	2	3
#5	5	5	1	6	5

Table 3: The table above provides an example of what the X data may look like. Rows represent hashtags and columns represent each time slot. The table data represents the frequency of each hashtag in a given time period.

Y Data
0
1
1
0
0

Table 4: The table above represents what the Y may look like. Each row corresponds directly to the row in the X data, stating whether or not it trended during any of the time periods. These values are always binary, 0 representing a non-trend and 1 representing a trend.