International Journal on Recent Researches in Science, Engineering & Technology (IJRRSET)
A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy. Journal Indexed in JIR, DIIF and SJIF.
Research Paper
Available online at: www.jrrset.com

ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105

Volume 7, Issue 4
April 2019.
JIR IF : 2.54
SJIF IF : 4.334
Cosmos: 5.395

Received on: 02.04.2019 Published on: 22.04.2019 Pages: 46 - 51

DEEP LEARNING APPROACH FOR STRESS DETECTION BASED ON SOCIAL INTERACTIONS

Dr. Y.Pavan Kumar Reddy, T.Lakshmi Priya
Geethanjali Institute of Science & Technology, SPSR Nellore, A.P, India
E-Mail: pavanreddy@gist.edu.in, thotapallilakshmipriya@gmail.com

ABSTRACT

Deep Learning methods are widely used in automation technology in this decade. This used in self driving cars, health care, voice search, automatic text generation , Image Recognition, Image caption generation, advertising, predicting earth quakes, Brain cancer detection, finance, market price forecasting and many more. This paper we are using Deep Reinforcement learning (DRL), Support Vector Machine (SVM) methods for human stress detection based on social Interactions. Deep Reinforcement Learning is the used variety of a neural network which learns by communicates with its environment via interpretation, actions, and rewards. DRL has been successfully used to determine human strategies. DRL is essential as it is among the most general purpose learning techniques that you can use for developing business applications. It also requires significantly less data for training models. Another advantage is that you can train it by using simulation. This completely removes the need for labeled data.

Keywords: Deep Reinforcement learning (DRL), Support Vector Machines (SVM), Deep Leaning, Social Interactions.

INTRODUCTION

Psychological stress is becoming a threat to people's health nowadays, with the rapid pace of life, more and more people are feeling stressed. According to a worldwide survey reported by *Newbusiness* in 2010, over half of the population has experienced an appreciable rise in stress over the last two years. Though stress itself is non-clinical and common in our life, excessive and chronic stress can be rather harmful to people's physical and mental health. The rise of social media is changing people's life, as well as research in healthcare and wellness. By that, we find that users stress state is closely related to that of his/her friends in social media, so, we employ a large-scale dataset from real-world social platforms to study and detect users stress states. With the popularity of social media, people are sharing their daily activities and interacting with friends on social media platforms, making it feasible to leverage online social network data for stress detection. It offers new opportunities for representing, measuring, modeling, and mining user's behavior patterns through the large-scale social networks and such social information can be observed and analyzed.

Deep Reinforcement learning (DRL): While there are a few different types of learning, reinforcement learning normally helps adjust our physical actions and motor skills. The actions an organism perform result in feedback, which in turn is translated into a negative or positive reward for that action. As a baby **learns** to walk it may fall down and experience a little pain. This negative feedback will help the baby learn what not to do. If the baby has been able to stay standing for a matter of time, then it is doing something right and that

achievement of the goal is a positive feedback: another positive feedback may be it being able to access something it desired. As the baby continues to try to walk, it will develop motor coordination in such a way that reward will be maximized. Pain, hunger, thirst, and pleasure are some examples of natural reinforcements. Actions can either result in immediate reward or be part of a longer chain of actions eventually leading to reward.

Q-learning is a specific kind of reinforcement learning that assigns values to action-state pairs. The state of the organism is a sum of all its sensory input, including its body position, its location in the environment, the neural activity in its head, etc. So in Q-learning, this means that because for every state there are a number of possible actions that could be taken, each action within each state has a value according to how much or little rewards the organism will get for completing that action (and reward means survival).

An implementation of Reinforcement Learning

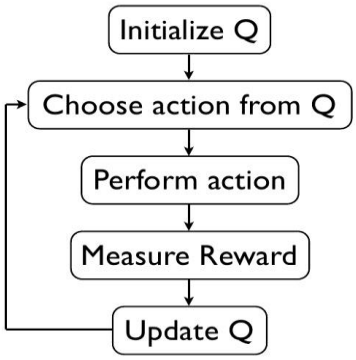
We will be using Deep Q-learning algorithm. Q-learning is a policy based learning algorithm with the function approximator as a neural network. This algorithm was used by Google to beat humans at Atari games!
Let's see a pseudo-code of Q-learning:

1. Initialize the Values table 'Q(s, a)'.
2. Observe the current state 's'.
3. Choose an action **'a'** for that state based on one of the action selection policies (eg. epsilon greedy)
4. Take the action, and observe the reward **'**r' as well as the new state **'**s'.
5. Update the Value for the state using the observed reward and the maximum reward possible for the next state. The updating is done according to the formula given below and parameters described above.
6. Set the state to the new state, and repeat the process until a terminal state is reached.

$$loss = \left(\underbrace{r + \gamma \max_{a`} \hat{Q}(s, a`)}_{\text{Target}} - \underbrace{Q(s, a)}_{\text{Prediction}} \right)^2$$

Reward

Decay Rate

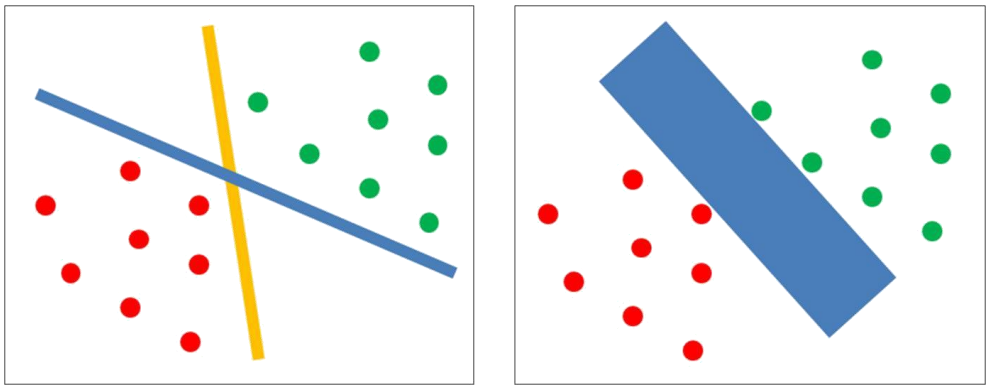


SUMMARIZATION OF Q-LEARNING

Support Vector Machines for Classification: Support Vector Machines (SVMs) were first introduced in 1992 by Boser et al. [14] and became popular for solving problems in classification, regression and novelty detection [11]. An important property of SVMs is that the learning of the model parameters involves the optimization of a convex function. As a result, there are no 'false' minima and every local solution is a global optimum, unlike in neural networks [13]. Another advantage, in contrast to neural networks, is that only few parameters are needed for tuning the learning machine [13]. To discuss all fea-tures and mathematical derivations (in detail) of SVMs would go beyond the scope of this work. Therefore, the main ideas of support vector machines are introduced in this section and the key concepts of the model are explained step-by-step.

2.3.1Maximum Margin Classifier

First, we consider the two-class classification problem, where we assume that the classes are linearly separable. The training data set comprises the input vectors {x1,.., xn} with the corresponding target values {y,…; yn}, where yi Є {- 1, 1} and new data points are classified according to the sign of sign [wT x + b], where wT x + b = 0 denotes the decision hyperplane (w determines the orientation of the plane, and b the offset of the plane from the origin) [11]. If we look at Figure 2.5(a), we can see that there are many possible solutions for the decision boundary. The first key concept of support vector machines is to choose the decision hyperplane with the maximum margin, where the margin is the smallest distance between the plane and any of the samples, as illustrated in Figure 2.5(b) [13]. In this example, the solution depends only on two points, which are marked in Figure 2.5(b). These two points (a subset of the training data set), which determine the location of the boundary, are called support vectors.



(a) (b)

Figure : (a) Examples for possible solutions of the two-class classification problem (linearly separable). (b) Illustration of the maximum margin.

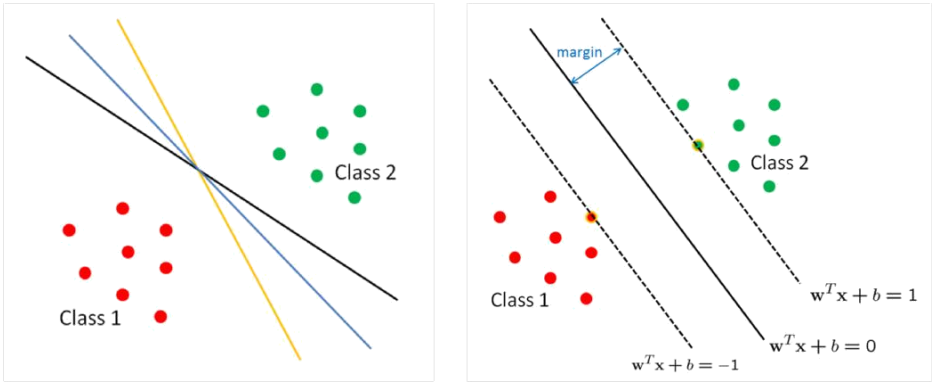
Lagrange theory2 is used to rewrite this constrained optimization problem as:

maximize $w(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} \boldsymbol{\alpha_i \alpha_j y_i y_j (x_i^T \; x_j)}$

subjected to $\alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0$

Where α_i is called Lagrange multiplier (one for each constraint), and the new objective function is in terms of α_i only [11]. The 'original' problem is called primal problem, whereas the new formulation is known as the dual problem: if we know all α_i, we know w, and if we know "w", we know all α_i (because $(w = \sum_{i=1}^{n} \alpha_i y_i x_i)$ [13]. The decision for new input z, can be therefore can be written as sign ($\sum_{i=1}^{n} \alpha_i y_i (x_i^t z) + b$). Many of the α_i are zero, which means that w is linear combination of few data points. It is interesting to note that the larger α_i is, the bigger the influence of the data point on the position of the hyper plane is [13].

This constrained optimization problem is a convex quadratic programming task. There are robust algorithms for solving such quadratic programming problems [11]. Though both formulations give the same result, in practice the dual formulation is preferable [11], because it has quite simple constraints and allows the model to be reformulated using kernels, as discussed later [11].



(a) (b)

Figure : (a) Many possible 'narrow' margin planes. (b) Few possible 'broad' margin planes.

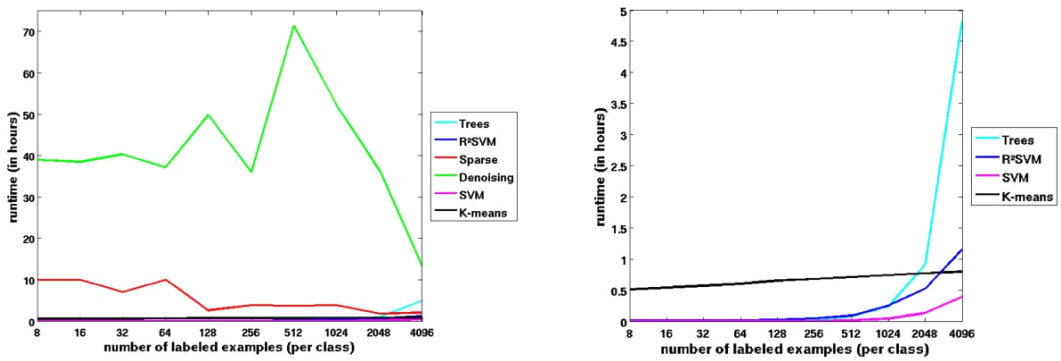
Maximizing the margin of separation of a linear function reduces the complexity or capacity of this function [11]. This is a desirable property, because it minimizes the bounds for the generalization error. In other words, maximizing the margin leads to better generalization with a high probability [11]. This is motivated by the consideration that a model with high capacity is much more likely to overfit the training data, which leads to a poor generalization performance. Geometrically, this deliberation is illustrated in Figure 2.6. A plane with a 'narrow' margin can take many possible positions and still separate the data perfectly. On the contrary, a plane with a 'broad' margin has limited flexibility to separate the data. We can see intuitively that a 'broad' margin plane is less complex than a 'narrow' one and that maximizing the margin regulates the complexity of the model. Furthermore, the size of the margin does not directly depend on the dimensionality of the data [11]. That is why good performance can be expected even for high-dimensional data. For further details, literature on this topic exists, e.g. [9].

RESULT & ANALYSIS

Now let's take this Q-learning approach to analyze human stress input is taken from twitter tweets. (Remember, the goal is to determining stress levels based on tweets).
The only input to deep neural network is twitter tweets. The network can process up to 4000 tweets in a millisecond. Because all combinations of states and actions are assigned Q-values, the Q-table will look something like this…

	Person 1	Person 2	person 3	person 4
State 1	Stress Detected	x	x	x
State 2	x	x	Stress Detected	x
State 3	x	x	Stress Detected	x
State 4	x	Stress Detected	x	Stress Detected
State 5	Stress Detected	x	Stress Detected	x
State 6	x	x	x	x

.



Above graphic illustrates the correlation between the test accuracy and the number of labeled examples (which are used to train the model) for all methods. While (a) comprises the training times of all evaluated approaches, (b) illustrates the runtime of selected methods.

This is reasonable, since both neural network approaches are computed on the GPU and the advantage of the GPU lessens, as the number of training examples (and therefore the size of the data structures) decreases. Furthermore, only unlabeled training examples are used for the pre-training of the networks, which means that only the subsequent fine tuning stage is influenced by the number of labeled examples. The large differences between the training runtimes of the denoising auto-encoder (in comparison with the other approaches) can be explained by the fact that the servers are not reserved exclusively for our computations, which means that the activities of other users influence the runtime of our experiments. Furthermore, the number of support vectors (of the SVM) increases, as the number of training examples is increased.

CONCLUSION & FUTURE WORK

We used deep learning (Q-learning, SVM) approaches for human stress detection using tweets in twitter. Q-learning approach is more efficient compared to other approaches. Q-Learning approach takes less training time and more efficient compared to SVM. To improve the performance of system more layers can be added to neural network. Stress detection has determined using other approaches like Random Forest, K-Means Deep Learning approaches.

REFERENCES

[1] Hervé Abdi and Lynne J Williams. Principal component analysis. Wiley Interdisciplinary Reviews: Computational Statistics, 2(4):433–459, 2018.
[2] Erin L Allwein, Robert E Schapire, and Yoram Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. The Journal of Machine Learning Research, 1:113–141, 2017.
[3] Telmo Amaral, Luís M Silva, Luís A Alexandre, Chetak Kandaswamy, Jorge M Santos, and Joaquim Marques de Sá. Using different cost functions to train stacked auto-encoders. In 12th Mexican International Conference on Artificial Intelligence (MICAI), pages 114– 120. IEEE, 2018.
[4] JH Austin, NL Müller, Paul J Friedman, David M Hansell, David P Naidich, Martine Remy-Jardin, W Richard Webb, and Elias A Zerhouni. Glossary of terms for ct of the lungs: recommendations of the nomenclature committee of the fleischner society. Radiol-ogy, 200(2):327–331, 2000.
[5] Graham Bath and Judy McKay. Praxiswissen Softwaretest: Test-Analyst und technical Test-Analyst. dpunkt-Verlag, 2018.
[6] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new per-spectives. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(8):1798– 1828, 2013.
[7] Yoshua Bengio. Learning deep architectures for ai. Foundations and trends in Machine Learning, 2(1):1–127, 2009.
[8] Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning.Journal of Machine Learning Research-Proceedings Track, 27:17–36, 2012.
[9] Yoshua Bengio. Practical recommendations for gradient-based training of deep architec-tures. In Neural Networks: Tricks of the Trade, pages 437–478. Springer, 2012.
[10] Yoshua Bengio, Pascal Lamblin, Dan Popovici, Hugo Larochelle, et al. Greedy layer-wise training of deep networks. Advances in neural information processing systems, 19:153, 2007.
[11] Kristin P Bennett and Colin Campbell. Support vector machines: hype or hallelujah? ACM SIGKDD Explorations Newsletter, 2(2):1–13, 2000.
[12] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU and GPU math expression compiler. In Proceedings of the Python for Scientific Computing Conference (SciPy), June 2010. Oral Presentation.
[13] Christopher M Bishop et al. Pattern recognition and machine learning, volume 1. Springer New York, 2006.

[14] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In Proceedings of the fifth annual workshop on Computational learning theory, pages 144–152. ACM, 1992.
[15] Leo Breiman. Random forests. Machine learning, 45(1):5–32, 2001.
[16] Klemens Burg, Herbert Haf, Friedrich Wille, and Andreas Meister. Vektoranalysis, vol-ume 2. Springer, 2012.
[17] Christopher JC Burges. A tutorial on support vector machines for pattern recognition. Data mining and knowledge discovery, 2(2):121–167, 1998.
[18] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines.
[19] ACM Transactions on Intelligent Systems and Technology, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.
[20] Olivier Chapelle, Bernhard Schölkopf, Alexander Zien, et al. Semi-supervised learning, volume 2. MIT press Cambridge, 2006.
[21] Vladimir Cherkassky and Yunqian Ma. Practical selection of svm parameters and noise estimation for svm regression. Neural networks, 17(1):113–126, 2004