Schedae Informaticae Vol. 25 (2016): 37–47
doi: 10.4467/20838476SI.16.003.6184

tfml 2017
theoretical foundations
of machine learning, Kraków

On Certain Limitations of Recursive Representation Model

Stanisław Jastrzębski, Igor Sieradzki
Faculty of Mathematics and Computer Science
Jagiellonian University, Łojasiewicza 6, 30-348 Kraków, Poland
e-mail: *{stanislaw.jastrzebski, igor.sieradzki}@uj.edu.pl*

Abstract. There is a strong research effort towards developing models that can achieve state-of-the-art results without sacrificing interpretability and simplicity. One of such is recently proposed Recursive Random Support Vector Machine (R^2SVM) model, which is composed of stacked linear models. R^2SVM was reported to learn deep representations outperforming many strong classifiers like Deep Convolutional Neural Network. In this paper we try to analyze it both from theoretical and empirical perspective and show its important limitations. Analysis of similar model Deep Representation Extreme Learning Machine (DrELM) is also included. It is concluded that models in its current form achieves lower accuracy scores than Support Vector Machine with Radial Basis Function kernel.

Keywords: support vector machines, random recursive support vector machine, extreme learning machine, representation learning, stacked generalization

1. Introduction

Successes of deep architectures often comes at the cost of interpretability and fitting complexity [1, 2]. Popular techniques used to battle this problem include random projections, which are a basic building block of Extreme Learning Machines [3, 4],

Received: 11 December 2016 / Accepted: 30 December 2016

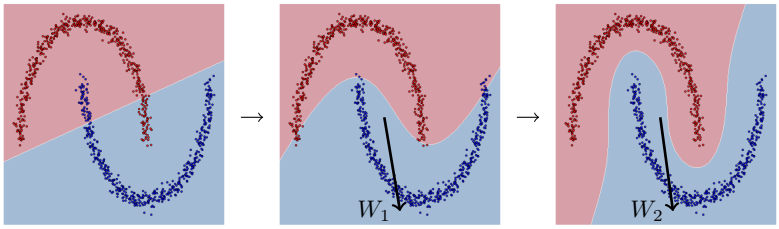

Figure 1. Visualization of R^{2}SVM model on two moon dataset. Random projections were manually adjusted.

where before classification data is projected into a higher dimensional space using random non-linear transformation. Such classifiers are stacked to form a deep architecture that can achieve state of the art results [5]. In this paper we analyze Random Recursive SVM (R^{2}SVM) model proposed by Vinyals et al. [6], which recursively transforms data using predictions from linear layers (see Figure 1). We will also cover similar model called Deep Representation Extreme Learning Machine (DrELM) [7]. Both models are following "stacked generalization" introduced by Wolpert et al. [8].

R^{2}SVM uses as linear classifier Support Vector Machines (SVM), model proposed by Vapnik [9], one of the most successful classifiers of the last decade mostly thanks to its well motivated regularization method in the linear case and its efficient delinearization. While shallow learners, like SVM, are usually outperformed by Deep Learning models, combining strengths of principled shallow models and Deep Learning ones is an active field [10].

The most important advantages of R^{2}SVM and DrELM include their interpretability and scalability, while at the same time learning deep representation, as reported by authors of the models. Indeed both linear SVM and non–linear ELM can be fitted in $\mathcal{O}(N)$ time and only single fit is performed for each layer. Both models optimize a convex objective, that has a global minimum. The original papers did not include theoretical discussion of the obtained results and its limitations, which are the main topic of the paper.

R^{2}SVM and DrELM

First we introduce the model in an informal discussion. R^{2}SVM consists of multiple layers transforming recursively input dataset. Let's focus on the binary case, where each layer fits a hyperplane separating the dataset into two subspaces. The two groups are then moved in random opposite directions proportionally to distance of the hyperplane. The main idea behind the model is to separate those groups, which should improve classification performance at later stages. Transformed dataset with applied non-linearity (which prevents some degeneration of the method) is passed to the next layer.

Formally let's consider training dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where $\mathbf{x}_i \in \mathbb{R}^d$ is a feature vector and $y_i \in \{1, 2, \ldots, K\}$ is class label. R^{2}SVM is a recursive model, where each layer transforms dataset using all previous outputs. Let's denote $\mathbf{X}_i \in \mathbb{R}^{N \times d}$ as representation output of the i$^{\text{th}}$ layer (see Fig. 2), $\mathbf{O}_i \in \mathbb{R}^{N \times K}$ is a vector of distances from hyperplanes and $\mathbf{W}_i^j \in \mathbb{R}^{K \times d}$ is a matrix containing random vectors as rows that maps output of i$^{\text{th}}$ layer to input of j$^{\text{th}}$ layer. Each random vector is drawn from standard normal distribution. Then we can write

$$\mathbf{X}_{i+1} = \sigma\left(\mathbf{X} + \alpha \sum_{j=1}^{i} \mathbf{O}_j \mathbf{W}_i^j\right), \tag{1}$$

where $\mathbf{X}$ is the original dataset, α controls size of applied transformation and σ is element–wise sigmoid, see Figure 2 for diagram.

In the case of DrELM we have a slight modification. Most importantly classifier used is Extreme Learning Machine (with linear activation function).

Authors also propose to use only the last output ($\mathbf{O}_i$):

$$\mathbf{X}_{i+1} = \sigma\left(\mathbf{X} + \alpha \mathbf{O}_i \mathbf{W}_i\right), \tag{2}$$

where σ is a sigmoid function.

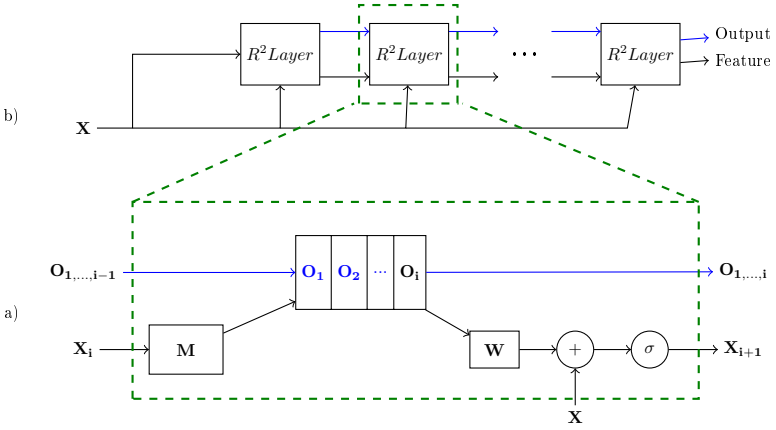

Figure 2. a) Each R^2 layer calculates new transformation based on the output of the previous layer. b) Model consists of repeated R^2 layers.

2. Theoretical analysis

To simplify notation we will use R^{2}M to denote both R^{2}SVM and DrELM models, where M denotes any classifier, so DrELM is denoted as R^{2}ELM. We start the discussion with analysis of R^{2}M behavior on the well–known spiral dataset. Despite its

simplicity model struggles to find correct random projections. We have exhaustively searched possible hypothesis space of a three layered R^{2}SVM and concluded that this dataset cannot be separated by the given model[1]. For simpler two–moon dataset approximately 1% of the runs separate the classes. Interestingly replicating dimensions (by stacking copies) of the datasets drastically improves performance, see Figure 3.

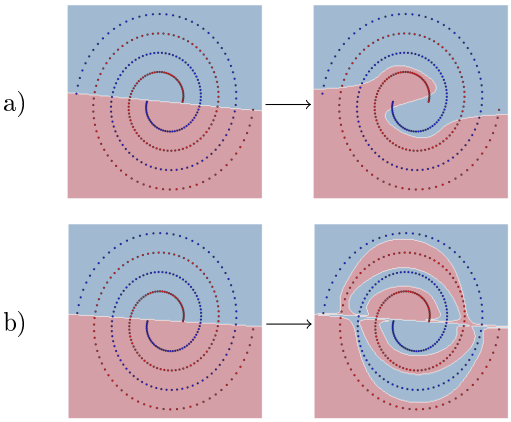

Figure 3. Visualization of 3 layered R^{2}SVM. a) Best run without replicating dimensions. b) Best run with replicated dimensions.

In this analysis we focus on model using only previous layer predictions, as in R^{2}ELM model. It simplifies analysis and does not decrease performance as is both conjectured by R^{2}ELM authors and proven empirically in this paper.

Recall that transformation used in R^{2}M for any linear M can be written as

$$\mathbf{X}_{i+1} = \sigma\left(\mathbf{X} + \alpha \mathbf{O}_i \mathbf{W}_i\right) = \sigma\left(\mathbf{X} + \alpha \sum \langle \mathbf{x}, \mathbf{v}_i \rangle\right) + \mathbf{b} = \sigma\left(\mathbf{X} + T(\mathbf{x})\right),$$

where by $T(\mathbf{x})$ we denote displacement function applied to vector given by $T(\mathbf{x}) = \sum_{i=1}^{K}(\langle \mathbf{x}, \mathbf{v}_i \rangle + \mathbf{b}_i)\mathbf{w}_i$. Note that we can write it as $T(\mathbf{x}) = \mathbf{x}\left(\sum_{i=1}^{K} \mathbf{v}_i \mathbf{w}_i^T\right) + \mathbf{b}$. Consequently

$$T(\mathbf{x}) = \mathbf{x}\left(\sum_{i=1}^{K} \mathbf{v}_i \mathbf{w}_i^T\right) + \mathbf{b} = \mathbf{x}\mathbf{A} + \mathbf{b}.$$

For linearly independent hyperplanes $rank(\mathbf{A}) = K$. Thus this displacement operator is a degenerated affine transformation. Whole layer could be interpreted as a two layer neural network with skip connections or a single layer neural network, see Figure 4.

[1] We tested all hyperplanes with a rotation step. Original paper reported the experiment using a deeper R^{2}SVM.

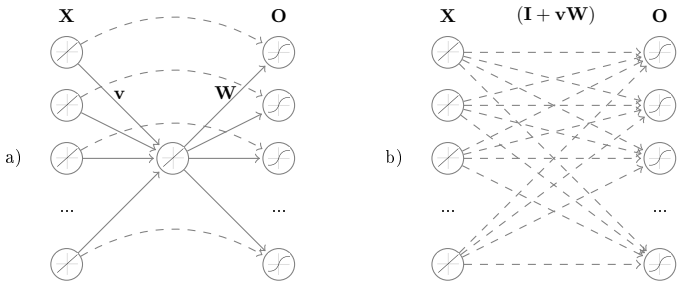

Figure 4. Interpretation of R^2SVM for binary case. a) As 2 layered neural network with skip connections. b) As fully connected single layer neural network ($\mathbf{x} + T(\mathbf{x}) = \mathbf{x}(\mathbf{I} + \mathbf{A})$).

2.1. Classifier importance

One of our claims is that R^2M performance is characterized by the intersection of found hyperplanes. This should be clarified by the following observation. Let $T_{V,W}$ be a transformation defined by hyperplanes, $V_1, \ldots, V_K$ and random vectors $\mathbf{w}_1, \ldots, \mathbf{w}_K$. From now simplified notation is used, where V_i is i^{th} hyperplane and $\mathbf{v}_i$ is its normal.

Observation 1 *Let $V_1, \ldots, V_K$ and $V_1', \ldots, V_K'$ denote hyperplanes then $\bigcap_i V_i = \bigcap_i V_i^{'} \Rightarrow \exists W'$, such that $T_{V',W} = T_{V,W'}$.*

Proof. Let's simplify by skipping bias in the equation for T (which is equivalent to adding constant dimension to x), then

$$T_{V,W} = \sum \mathbf{v}_i \mathbf{w}_i^T.$$

First we show that that $\ker T = \bigcap_i V_i$ if random vectors W are linearly independent. Left inclusion is obvious. Assume that $\mathbf{x} \in \ker T$ and $\mathbf{x} \notin \bigcap_i V_i$, but that implies $\exists \alpha_i : \sum_{i=1}^{K} \alpha_i \mathbf{w}_i = 0$. That would mean that random vectors W are not linearly independent. If the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_K$ are linearly independent then $\dim(\ker T_{V,W}) = N - K$, because $T(\mathbf{x}) = 0 \Leftrightarrow \forall_{i \in \{1,\ldots,K\}}\ \mathbf{x} \perp \mathbf{v}_i$. In general we have $\dim(\ker T_{V,W}) \leq N - K$. We choose basis for V $\{\mathbf{e}_1, \ldots, \mathbf{e}_L\}, L \leq K$. $T_{V',W} = T_{V,W'}$ is equivalent to the set of $N \times L$ linear equations with $N \times K$ parameters:

$$T_{V',W}(\mathbf{e}_i) = T_{V,W'}(\mathbf{e}_i), \quad i = 1, \ldots, L$$

which has an unique solution. □

The observation suggests that multiclass classification is not crucial, as one can for instance rotate all hyperplanes or perform orthogonalization and still be able to find an equivalent model. The observation does not include any results on distribution of W and because of that only hints at this possibility that doing multiclass classification in the middle layers might not be optimal.

2.2. Model interpretation

In this section we derive an interpretation of the model. Let $\mathbf{x} \in \mathbf{X}$ is a data point. Let $T_{V,W}(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b}$ be affine displacement operator. Clearly $\ker T$ is also an affine subspace, as it is intersection of K hyperplanes (affine subspaces). Denote by $P_{\ker T}(\mathbf{x})$ a projection onto this space, then

$$T'_{U,W}(\mathbf{x}) = T_{V,W}(\mathbf{x} - P_{\ker T}(\mathbf{x})).$$

We can now easily simplify further discussion. Let us first fix origin inside $\ker T$ and then subtract projection as follows: $\mathbf{y} = \mathbf{x} - P_{\ker T_{V,W}}$. We can now assume that space is a vector space, in which of course $\ker T'$ is a linear subspace.

Denote by $\mathbf{u}_1, \ldots, \mathbf{u}_K$ transformed vectors $\mathbf{v}_1, \ldots, \mathbf{v}_K$. Let $B = span(U)$ be subspace spanned by $\mathbf{u}_1, \ldots, \mathbf{u}_K$. We can proof the following observation

Observation 2 *Let $U^\star$ be set of optimally separating hyperplanes in B. Then exists W' such that $T_{U,W} = T_{U^\star,W'}$*

Proof. All vectors from $U^\star$ are linear combinations of vectors from U, $\mathbf{u}_i^\star \in U^\star \Leftrightarrow \mathbf{u}_i = \sum_{j=1}^{L} \alpha_j \mathbf{u}_j$. From this follows that $\bigcap_i U_i = \bigcap_i U_i^\star$. We can find W' using Observation 1. □

Observation 2 suggests that if B subspace is well separable, then model should perform considerably well, given that W' is not degenerated. Finding easily separable B space is unfortunately not equivalent to maximizing multiclass accuracy. One can for instance simply add new hyperplane that is not a good classifier, but divides space in a useful way. If the dataset is almost linearly separable then if V consist of hyperplanes separating classes then B space is likely to be linearly separable; however *the opposite* might not hold. This suggests that the method might not be well suited for datasets for which linear classifiers performs poorly.

It seems also natural that both models should work well when data manifold has smaller dimensionality than space, as it makes more likely that random projections are separating classes. We pose the following hypotheses, that are validated in the empirical section:

Hypothesis 1 *R^2M with L layers performs weakly if at any of the $1, \ldots, L-1$ layers data representation leads to weak linear classifier.*

Hypothesis 2 *R^2M work better if dataset resides on lower dimensional manifold.*

3. Empirical analysis

Tested models were Random Recursive Support Vector Machine (R^2SVM), Deep Representation Extreme Learning Machine (R^2ELM), Support Vector Machine with

Radial Basis Kernel (SVM+RBF), Extreme Learning Machine with Sigmoid Kernel (ELM+SIG) and Linear Support Vector Machine (SVM).

Additionally we added randomized version of R^2SVM called Fixed Prediction R^2SVM ($R^2$1+SVM), where middle layers are using constant prediction equal to one, i.e. $O_i = 1$[2]. By adding this model we evaluate if R^2SVM is not just using additional space besides dataset manifold to increase separability. Interesting results reported in theoretical section inspired us to add modifications of $R^2$1+SVM and R^2SVM, in which models use tripled version of the dataset (R^2_*1+SVM, R^2_*SVM).

3.1. Evaluation

We conducted our experiments on datasets taken from UCI repository [11] and LIBSVM dataset repository [12]. Summary of the datasets is presented in Table 1. To make sure we make a fair comparison we tested extensive grid of parameters for R^2SVM and R^2ELM. The grid tested all combinations (640) of the following parameters:

1. *recurrent*: If set to true model is reusing all the previous predictions as in R^2SVM model.
2. *scale*: If set to true model is performing scaling in every layer.
3. *fit*: If set to true model performs approximate regularization parameter C fitting in each layer.
4. α: Controls size of applied transformation. 8 values ranging from 0.1 to 2.0.
5. *depth*: Depth of the model. 10 values from 1 to 10.

Code used for experimentation is accessible online[3]. Experiments were conducted in Python using scikit-learn and LIBSVM package [12, 13]. All of the experiments were done using 5-fold stratified cross validation. Every experiment for R^2SVM models (R^2SVM, R^2_*SVM, ...) is repeated three times (with different seed) and average accuracy of the best performing set of hyperparameters is reported.

3.2. Results

Results are reported in Table 2. We would like to focus on several outcomes. Experiments clearly confirm that R^2M is weaker or comparable to SVM+RBF model in classification accuracy. This does not contradict results reported by authors of the model, as their work did not include extensive testing. Additionally, datasets

[2] Similar results were obtained by several other random versions of R^2SVM, for instance using random hyperplanes.

[3] https://github.com/gmum/r2-learner

Table 1. Experiments datasets summary. N – number of examples, d – number of dimensions, K – number of classes, M – manifold dimension estimation using PCA.

name	N	d	K	M	name	N	d	K	M
australian	690	14	2	1	liver	345	6	2	3
bank	1372	4	2	3	pendigits	10992	16	10	9
breast cancer	683	10	2	1	satimage	10870	36	6	6
crashes	540	20	2	1	segment	2310	19	7	7
diabetes	768	8	2	2	sonar	208	60	2	28
fourclass	862	2	2	2	splice	1000	60	2	55
german	1000	24	2	3	svmguide2	391	20	3	15
glass	214	9	6	6	svmguide4	612	10	6	1
heart	270	13	2	3	vehicle	846	18	4	6
indian	583	10	2	3	vowel	990	10	11	8
ionosphere	351	34	2	24	wine	178	4	3	2
iris	150	4	3	2					

tested in this work are characterized by rather low dimensionality and high number of classes, which differs from set tested in the original papers. It is also consistent with our theoretical understanding of the model.

Both hypotheses are confirmed by the empirical results. Hypothesis 1 specifically stated that $\mathrm{R}^2\mathrm{M}$ performance is correlated with some measure of weakness of linear classifier. It is clearly visible for highly linearly non–separable datasets, e.g. *glass* or *vowel*. We did a heuristic hypothesis test by measuring correlation of $\frac{\mathrm{acc}_{SVM+RBF}}{\mathrm{acc}_{R^2SVM}}$ and $\frac{\mathrm{acc}_{SVM}-\mathrm{acc}_{SVM+RBF}}{\mathrm{acc}_{SVM+RBF}}$, where acc is the best accuracy on given dataset. We obtained approximately 0.9 Spearman's correlation coefficient, which supports the hypothesis.

Second hypothesis can be validated similarly, we calculate correlation between acc_{R^2SVM} and difference between dataset dimensionality and manifold estimation. We report Spearman's correlation coefficient equal to 0.7, which also confirms correlation, however weaker.

Additional result is the fact that tripling dataset dimensionality by replicating data improves accuracy. It is equivalent to increasing hidden layer sizes in neural network interpretation of $\mathrm{R}^2\mathrm{M}$ model. In that case R^2_*1+SVM model, which is not fitting linear models in the middle layers, is performing comparatively to R^2SVM.

Table 2. Main results of the experiments. Both accuracy and standard deviation is reported.

dataset	R^2SVM		R^2ELM		ELM+SIG		SVM+RBF		SVM		$R^2$1+SVM		R^2_*SVM		*SVM+RBF		R^2_*1+SVM	
australian	0.87	±0.01	0.87	±0.01	**0.88**	± 0.02	0.87	±0.01	0.86	±0.01	0.86	±0.01	0.87	±0.01	0.86	±0.01	0.86	±0.01
bank	1.00	±0.01	0.97	±0.01	**1.00**	± 0.00	**1.00**	± 0.00	0.99	±0.01	1.00	±0.01	1.00	±0.01	**1.00**	± 0.00	**1.00**	± 0.00
breast_cancer	0.97	±0.01	0.97	±0.01	0.97	±0.01	0.97	±0.01	0.97	±0.01	0.97	±0.01	0.97	±0.01	0.97	±0.01	**0.97**	± 0.01
crashes	0.95	±0.01	0.95	±0.01	0.93	±0.01	**0.96**	± 0.02	0.96	±0.01	0.95	±0.01	0.95	±0.01	0.96	±0.01	0.95	±0.01
diabetes	0.78	±0.01	0.77	±0.01	**0.78**	± 0.03	0.78	±0.01	0.78	±0.01	0.76	±0.01	0.76	±0.01	0.78	±0.01	0.77	±0.01
fourclass	0.79	±0.01	0.78	±0.01	0.99	±0.01	**1.00**	± 0.00	0.77	±0.01	0.74	±0.01	0.81	±0.01	**1.00**	± 0.00	0.77	±0.01
german	0.77	±0.01	**0.77**	± 0.00	0.75	±0.01	0.76	±0.01	0.76	±0.01	0.73	±0.01	0.73	±0.01	0.76	±0.01	0.72	±0.01
glass	0.64	±0.01	0.64	±0.01	0.71	±0.01	**0.73**	± 0.05	0.62	±0.01	0.59	±0.01	0.69	±0.01	0.72	±0.01	0.65	±0.01
heart	0.84	±0.01	**0.85**	± 0.00	0.83	±0.01	0.85	±0.01	0.84	±0.01	0.85	±0.01	0.85	±0.01	0.85	±0.01	0.85	±0.01
indian	0.73	±0.01	0.72	±0.01	**0.73**	± 0.01	0.72	±0.01	0.72	±0.01	0.71	±0.01	0.71	±0.01	0.72	±0.01	0.71	±0.01
ionosphere	0.89	±0.01	0.92	±0.01	0.91	±0.01	**0.96**	± 0.02	0.89	±0.01	0.91	±0.01	0.89	±0.01	0.93	±0.01	0.92	±0.01
iris	0.97	±0.01	0.96	±0.01	**0.98**	± 0.02	**0.98**	± 0.03	0.95	±0.01	0.97	±0.01	**0.98**	± 0.01	0.97	±0.01	0.98	±0.01
liver	0.70	±0.01	0.69	±0.01	0.74	±0.01	**0.75**	± 0.04	0.71	±0.01	0.66	±0.01	0.70	±0.01	0.73	±0.01	0.71	±0.01
pendigits	0.94	±0.01	0.87	±0.01	0.97	±0.01	**1.00**	± 0.00	0.93	±0.01	0.94	±0.01	0.99	±0.01	1.00	±0.01	0.97	±0.01
satimage	0.89	±0.01	0.88	±0.01	0.89	±0.01	0.93	±0.01	0.82	±0.01	0.87	±0.01	0.92	±0.01	**0.97**	± 0.00	0.90	±0.01
segment	0.96	±0.01	0.94	±0.01	0.93	±0.01	0.97	±0.01	0.93	±0.01	0.93	±0.01	**0.97**	± 0.01	0.97	±0.01	0.94	±0.01
sonar	0.75	±0.01	0.77	±0.01	0.83	±0.01	**0.89**	± 0.05	0.76	±0.01	0.81	±0.01	0.76	±0.01	0.82	±0.01	0.84	±0.01
splice	0.81	±0.01	0.82	±0.01	0.76	±0.01	0.88	±0.01	0.80	±0.01	0.81	±0.01	0.81	±0.01	**0.88**	± 0.02	0.88	±0.01
svmguide2	0.83	±0.01	0.84	±0.01	0.82	±0.01	0.85	±0.01	0.84	±0.01	0.82	±0.01	0.83	±0.01	**0.85**	± 0.02	0.84	±0.01
svmguide4	0.85	±0.01	0.80	±0.01	0.76	±0.01	0.87	±0.01	0.81	±0.01	0.73	±0.01	**0.90**	± 0.01	0.87	±0.01	0.82	±0.01
vehicle	0.81	±0.01	0.80	±0.01	0.82	±0.01	**0.86**	± 0.01	0.78	±0.01	0.78	±0.01	0.82	±0.01	0.85	±0.01	0.82	±0.01
vowel	0.62	±0.01	0.54	±0.01	0.83	±0.01	0.99	±0.01	0.47	±0.01	0.49	±0.01	0.87	±0.01	**1.00**	± 0.00	0.72	±0.01
wine	0.83	±0.01	0.83	±0.01	**0.87**	± 0.04	0.86	±0.01	0.83	±0.01	0.84	±0.01	0.84	±0.01	0.84	±0.01	0.86	±0.01

4. Summary

In this paper we analyzed rigorously R^2SVM and R^2ELM models both from theoretical and empirical point of view. Basing on our theoretical observations we have proven empirically two hypotheses, stating that R^2SVM and R^2ELM perform weakly if any of the layer data representation leads to weak linear classifier and that R^2SVM and R^2ELM rely on dataset residing on a much lower dimensional manifold. In summary it suggests that R^2SVM and R^2ELM might not be always learning useful representations, which is further confirmed by weak results in comparison with Support Vector Machine with RBF kernel. Future research should focus on making sure R^2SVM is more broadly applicable.

Acknowledgements

We would like to thank Wojciech Czarnecki, member of our research group, for support and guidance that he continuously expressed during our work on this paper.

5. References

[1] Bengio Y., *Learning deep architectures for AI.* Foundations and Trends in Machine Learning, 2009, 2.

[2] Podolak I.T., Roman A., *Theoretical foundations and experimental results for a hierarchical classifier with overlapping clusters.* Computational Intelligence, 2013, 29 (2), pp. 357–388.

[3] Czarnecki W.M., Tabor J., *Extreme entropy machines: Robust information theoretic classification.* arXiv preprint arXiv:1501.05279, 2015.

[4] Huang G.B., Zhu Q.Y., Siew C.K., *Extreme learning machine: Theory and applications.* Neurocomputing, 2006, 70 (1–3), pp. 489–501.

[5] Tissera M., McDonnell M., *Deep extreme learning machines for classification.* In: *Proceedings of ELM-2014 Volume 1.* vol. 3 of *Proceedings in Adaptation, Learning and Optimization.* Springer International Publishing 2015, pp. 345–354.

[6] Vinyals O., Jia Y., Deng L., Darrell T., *Learning with recursive perceptual representations.* In: *Advances in Neural Information Processing Systems 25.* Curran Associates, Inc. 2012, pp. 2825–2833.

[7] Yu W., Zhuang F., He Q., Shi Z., *Learning deep representations via extreme learning machines.* Neurocomputing, 2015, 149, pp. 308–315.

[8] Wolpert D.H., *Stacked generalization.* Neural Networks, 1992, 5, pp. 241–259.

[9] Cortes C., Vapnik V., *Support-vector networks.* Machine Learning, 1995, 20 (3), pp. 273–297.

[10] Tang Y., *Deep learning using linear support vector machines.* In: *In ICML*, 2013.

[11] Lichman M., *UCI machine learning repository*, 2013.

[12] Chang C.C., Lin C.J., *LIBSVM: A library for support vector machines.* ACM Transactions on Intelligent Systems and Technology, 2011, 2, pp. 27:1–27:27.

[13] Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., Vanderplas J., Passos A., Cournapeau D., Brucher M., Perrot M., Duchesnay E., *Scikit-learn: Machine learning in Python.* Journal of Machine Learning Research, 2011, 12, pp. 2825–2830.