ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1.
Available from http://www.i6doc.com/en/.

Scalable Hybrid Deep Neural Kernel Networks

Siamak Mehrkanoon*[1], Andreas Zell[2], Johan A.K. Suykens[1]

1- Department of Electrical Engineering ESAT-STADIUS,
KU Leuven, B-3001 Leuven, Belgium
2- Computer Science Department, University of Tuebingen,
Sand 1, 72076 Tuebingen, Germany
*Corresponding author, e-mail: siamak.mehrkanoon@esat.kuleuven.be

Abstract. This paper introduces a novel hybrid deep neural kernel framework. The proposed deep learning model follows a combination of neural networks based architecture and a kernel based model. In particular, here an explicit feature map, based on random Fourier features, is used to make the transition between the two architectures more straightforward as well as making the model scalable to large datasets by solving the optimization problem in the primal. The introduced framework can be considered as the first building block for the development of even deeper models and more advanced architectures. Experimental results show a significant improvement over shallow models on several medium to large scale real-life datasets.

1 Deep Learning Models

Conventional machine learning techniques were limited in processing natural data in their raw forms and a lot of domain experts were required in transforming raw data into meaningful features or representations. Deep Learning is a class of machine learning techniques that belongs to the family of representation learning models [1]. It has attracted many researchers due to its success in revolutionizing many application domains ranging from auditory to vision sensory signal processing. Deep learning based models deal with complex tasks by learning from subtasks. In particular, several nonlinear modules are stacked in hierarchical architectures to learn multiple levels of representation (hierarchical features) from the raw input data. Each module transforms the representation at one level into a slightly more abstract representation at a higher level, i.e. the higher-level features are defined in terms of lower-level ones. Deep learning architectures have grown significantly, resulting in different models such as stacked denoising autoencoders [2], Restricted Boltzmann Machines [3], Convolutional Neural Networks [4, 5] among others.

Recent works in machine learning have highlighted the superiority of deep architectures over shallow architectures in terms of accuracy in several application domains [1, 6]. But, no free lunch, training deep neural networks involves difficult nonlinear optimizations and demands huge amount of training data. Most of the developed deep learning models are based on artificial neural networks (ANN) architecture, whereas deep kernel based models have not yet been explored in great detail. Authors in [7] introduced a convex deep learning model

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1. Available from http://www.i6doc.com/en/.

via normalized kernels. A family of positive-definite kernel functions that mimic the computation in multilayer neural networks is given in [8].

This paper is organized as follows. In Section 2 the existing connections between artificial neural networks and kernel based models are explored. Section 3 introduces the proposed hybrid deep neural kernel architecture that can take advantage of the best of two worlds by bridging between the ANN and kernel based models. Section 4 reports the experimental results. Conclusions and future works are drawn in Section 5.

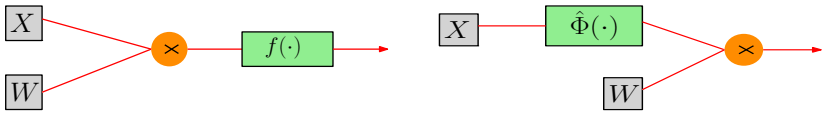
2 ANNs vs Kernel Architecture

In neural networks based approaches, the input is projected into a new space via multiplication with a weight matrix followed by applying a nonlinear activation function. If we consider the described operations in a module, then one can stack together several of these modules and form a deep architecture. In this way, different configurations of stacking as well as wiring used in the entire networks can cover different modeling strategies. On the other hand, in kernel based approaches one often works with the primal-dual setting. In the primal formulation, one projects the input data into an infinite dimensional space using implicit feature mapping. The projected data points are then mapped to a target space by means of an inner product with a weight matrix (primal decision variables). Alternatively one could also work with an explicit feature map and project the data into a finite dimensional feature space where each of its elements can be approximated. In the case of an implicit feature mapping, the projection space corresponds to a hidden layer in a neural network with infinite number of neurons [9]. Whereas using an explicit feature map the connection with neural network architectures becomes even more closer. More precisely, the dimension of the explicit feature map corresponds to the number of hidden units in the hidden layer of a neural network architecture.

However, inspecting the mathematical expressions of ANNs and kernel based models with explicit feature mapping, reveals the differences and similarities between the two frameworks, see Fig. 1. In ANNs, the nonlinear activation function is applied on the weighted sum of the given input instance. Whereas in kernel based approaches, the nonlinear feature map is directly applied on the given input instance and then the target value is estimated by means of a weighted sum of the projected sample. When an explicit feature map is used, the optimal values for the model parameters, i.e. weights and biases, can be learned in the primal. In contrast to the ANN module shown in Fig. 1(a), a single module of kernel based model is linear in the weight matrix W, therefore convex optimization techniques can be applied to obtain an optimal values of W. In addition, compared to a single ANN module, in practice, it has a better capability for learning nonlinear decision boundaries with a satisfactory generalization performance. It is worth noting that the matrix of the hidden layer shown in Fig. 1(a) can also be treated at the tuning parameter level (see [10]).

In today's applications, addressing large scale problems is inevitable. ANN

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1.
Available from http://www.i6doc.com/en/.



(a) (b)

Fig. 1: Computational graph corresponding to (a) A single module of a neural network architecture. (b) A single module of kernel based models with explicit feature mapping.

based models can rely on stochastic gradient descent algorithm for obtaining the optimized model parameters when the size of the data set is large. In kernel based approaches, on the other hand, one can for instance work with low rank approximation of the kernel matrix and avoid building and storing the entire kernel matrix which is not computationally efficient.

3 Proposed Deep Hybrid Model

Consider training data points $\mathcal{D} = \{x_1, ..., x_n\}$, where $\{x_i\}_{i=1}^n \in \mathbb{R}^d$, the labels $\{y_i\}_{i=1}^n$ and the total number of classes is Q. This section introduces a new deep architecture configuration that bridges the ANN and kernel based models in the primal level. The proposed model is suitable for both regression and classification. To this end, for the kernel based model counterpart we employ an explicit feature mapping. In the literature, several methodologies are introduced to scale up the kernel based models for dealing with large scale problems. For instance Greedy basis selection techniques [11], incomplete Cholesky decomposition [12], Nyström methods [13, 14] aim at providing a low-rank approximation of the kernel matrix. In particular the Nyström approximation method as well as a reduced kernel technique have been previously successfully applied in the context of large scale semi-supervised learning for providing an approximation of the feature map and solving the optimization problem in the primal (see [15]).

Here we use random Fourier features to compute an explicit feature map and build a module that can be cast into a neural networks architecture. Random Fourier features has recently been introduced in the field of kernel methods by exploiting the classical Bochner's theorem in harmonic analysis [16]. The Bochner's theorem states that a continuous kernel $K(x, y) = K(x - y)$ on $\mathbb{R}^d$ is positive definite if and only if K is the Fourier transform of a non-negative measure. If a shift-invariant kernel k is properly scaled, its Fourier transform $p(\xi)$ is a proper probability distribution. This property is used to approximate kernel functions with linear projections on D random features as follows [16]: $K(x - y) = \int_{\mathbb{R}^d} p(\xi) e^{j\xi^T(x-y)} d\xi = \mathbb{E}_\xi[z_\xi(x) z_\xi(y)^*]$, where $z_\xi(x) = e^{j\xi^T x}$. Here $z_\xi(x) z_\xi(y)^*$ is an unbiased estimate of $K(x, y)$ when ξ is drawn from $p(\xi)$ (see [16]). To obtain a real-valued random feature for K, one can replace the $z_\xi(x)$ by the mapping $z_\xi(x) = \cos(\xi^T x)$. The random Fourier features $\hat{\varphi}(x)$, for the sample x, are then defined as $\hat{\varphi}(x) = \frac{1}{\sqrt{D}}[z_{\xi_1}(x), \ldots, z_{\xi_D}(x)]^T \in \mathbb{R}^D$. Here $\frac{1}{\sqrt{D}}$ is used as normalization factor to reduce the variance of the estimate and

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1.
Available from http://www.i6doc.com/en/.

$\xi_1, \ldots, \xi_D \in \mathbb{R}^d$ are sampled from $p(\xi)$. For a Gaussian kernel, they are drawn from a Normal distribution $\mathcal{N}(0, \sigma^2 I_d)$. Assuming that an explicit feature map is given, we formulate a two layer hybrid architecture as follows (see also Fig. 2):

$$h_1 = W_1 x + b_1, \quad h_2 = \hat{\varphi}(h_1), \quad S = W_2 h_2 + b_2,$$

where $W_1 \in \mathbb{R}^{d_1 \times d}$ and $W_2 \in \mathbb{R}^{Q \times d_2}$ are weight matrices and the bias vectors are denoted by b_1 and b_2.

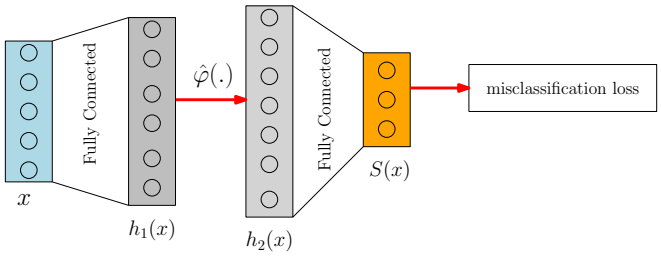


Fig. 2: Deep hybrid neural kernel network architecture

The dimensions of the hidden variables h_1 and h_2 are user defined parameters. Here we assume that the input data are first projected to a d_1 dimensional space in the first layer followed by a d_2 dimensional space in the second layer. Any misclassification loss function can be employed in the misclassification loss layer. Here the cross-entropy loss, also known as softmax function, is used which equips the results with a probabilistic interpretation by minimizing the negative log likelihood of the correct class. The formulation of the proposed method is as follows:

$$\min_{W_1, W_2, b_1, b_2} J(W_1, W_2, b_1, b_2) = \frac{1}{2} Tr(W_1 W_1^T) + \frac{1}{2} Tr(W_2 W_2^T) + \frac{\gamma_1}{n} \sum_{i=1}^{n} L(x_i, y_i). \quad (1)$$

Here the cross-entropy loss, i.e. $L(x_i, y_i) = -\log\left(\frac{\exp(s_i^{y_i})}{\sum_{j=1}^{Q} \exp(s_i^{y_j})}\right)$, is used where s_i^j denotes the score that is assigned to j-th class for the instance x_i and y_i is the true class membership for the instance x_i. The first two terms in the objective function are regularization terms over the model parameters. The last term in (1) aims at minimizing the negative log likelihood of the correct class. The stochastic gradient descent algorithm is used to train the model in (1). After obtaining the model parameter W_1 and W_2, the score variable for the test point x_{test} can be computed as follows: $S_{\text{test}} = W_2\ \tilde{\varphi}(W_1 x_{\text{test}} + b1) + b_2$. The final class label for the test point x_{test} is computed by $\hat{y}_{\text{test}} = \underset{\ell=1,\ldots,Q}{\text{argmax}}(S_{\text{test}})$.

4 Experimental Results

In this section experimental results on several real-life datasets taken from UCI machine learning repository and KEEL-dataset are reported. All the experiments are performed on a laptop computer with Intel Core i7 CPU and 16 GB

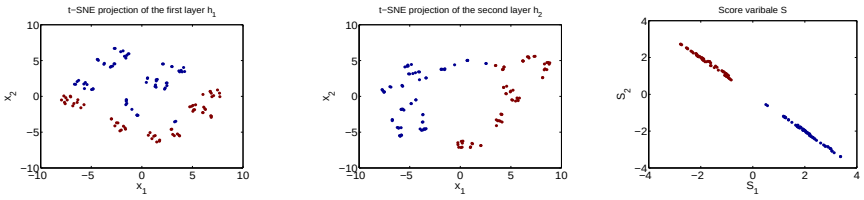
ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1.
Available from http://www.i6doc.com/en/.

RAM under Matlab 2014a. In all the experiments, the given dataset is randomly partitioned to 80% training and 20% test sets respectively. The performance of the proposed deep model is compared with that of the shallow LS-SVM with implicit and explicit feature mapping which correspond to solving the problem in the dual and primal respectively. In order to have a fair comparison, the dimension of the explicit random Fourier feature in both deep and shallow models are kept the same. The hyper-parameters are tuned using 10-fold cross-validation. The obtained results obtained by averaging over 10 simulation runs are tabulated in Table 1, which shows that in most of the cases studied here the proposed hybrid deep model improves the accuracy over the shallow structure models. The t-SNE visualization [17] of the obtained projections in the first and second layers as well as the score variables for the Monk2 dataset are also depicted in Fig. 3 which shows the evolution of the features in the network.

Table 1: The average accuracy of the proposed deep model and the shallow LS-SVM with implicit and explicit feature maps on several real-life datasets.

				Shallow LS-SVM	
Dataset	n	d	**Hybrid Deep Model**	**Primal**	**Dual**
Australian	690	14	$\mathbf{0.85 \pm 0.01}$	0.81 ± 0.01	0.83 ± 0.01
Sonar	208	60	$\mathbf{0.75 \pm 0.04}$	0.69 ± 0.07	0.70 ± 0.05
Titanic	2201	3	$\mathbf{0.78 \pm 0.01}$	0.77 ± 0.02	$\mathbf{0.78 \pm 0.01}$
Monk2	432	6	$\mathbf{1.00 \pm 0.00}$	0.93 ± 0.05	0.95 ± 0.02
Balance	625	4	$\mathbf{0.96 \pm 0.01}$	0.93 ± 0.02	0.94 ± 0.01
Magic	19,020	10	$\mathbf{0.86 \pm 0.04}$	0.84 ± 0.01	0.84 ± 0.01
Covertype	581,012	54	$\mathbf{0.85 \pm 0.03}$	0.78 ± 0.01	N.A
SUSY	5,000,000	18	$\mathbf{0.80 \pm 0.02}$	0.78 ± 0.01	N.A

Note: "N.A" stands for Not Applicable due the large size of the dataset.



(a) (b) (c)

Fig. 3: Monk2 dataset. (a) and (b) t-SNE projections of the test data in the hidden layers h_1 and h_2. (c) The obtained score variables S for the test data.

5 Conclusions and Future Works

In this paper a new hybrid deep neural kernel network model is introduced. The similarities and differences between single artificial neural networks module and its kernel counterpart are discussed in detail. We showed how hybridization of kernel based models with explicit feature mapping and neural networks can lead to a new deep architecture, taking advantage of the two worlds. The proposed

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1. Available from http://www.i6doc.com/en/.

model can be considered as the first building block for deeper models with more advanced architectures. Our future work is devoted to studying several choices of misclassification loss functions as well as the extension of the proposed framework to semi-supervised learning with deep architecture.

Acknowledgments
The authors acknowledge support of ERC AdG A-DATADRIVE-B (290923), KUL: GOA/10/09 MaNet, CoE PFV/10/002 (OPTEC), BIL12/11T; FWO: G.0377.12, G.088114N, G0A4917N; IUAPP7/19 DYSCO.

References

[1] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[2] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010.

[3] Ruslan Salakhutdinov and Geoffrey E Hinton. Deep Boltzmann Machines. In *AISTATS*, volume 1, page 3, 2009.

[4] Steve Lawrence, C Lee Giles, Ah Chung Tsoi, and Andrew D Back. Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 8(1):98–113, 1997.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[6] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[7] Özlem Aslan, Xinhua Zhang, and Dale Schuurmans. Convex deep learning via normalized kernels. In *Advances in Neural Information Processing Systems*, pages 3275–3283, 2014.

[8] Youngmin Cho and Lawrence K Saul. Kernel methods for deep learning. In *Advances in neural information processing systems*, pages 342–350, 2009.

[9] J A K Suykens, T Van Gestel, J De Brabanter, B De Moor, and J Vandewalle. *Least squares support vector machines*. Singapore: World Scientific Pub. Co., 2002.

[10] Johan AK Suykens and Joos Vandewalle. Training multilayer perceptron classifiers based on a modified support vector method. *IEEE Transactions on Neural Networks*, 10(4):907–911, 1999.

[11] Alex J Smola and Bernhard Schölkopf. Sparse greedy matrix approximation for machine learning. *17th ICML, Stanford, 2000*, pages 911–918, 2000.

[12] Francis R Bach and Michael I Jordan. Predictive low-rank decomposition for kernel methods. In *Proceedings of the 22nd ICML*, pages 33–40. ACM, 2005.

[13] Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. Sampling methods for the Nyström method. *The Journal of Machine Learning Research*, 13(1):981–1006, 2012.

[14] Christopher Williams and Matthias Seeger. Using the Nyström method to speed up kernel machines. In *Proceedings of the 14th Annual Conference on Neural Information Processing Systems*, number EPFL-CONF-161322, pages 682–688, 2001.

[15] Siamak Mehrkanoon and Johan A K Suykens. Large scale semi-supervised learning using KSC based model. In *Proc. of International Joint Conference onNeural Networks (IJCNN)*, pages 4152–4159, 2014.

[16] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2007.

[17] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.