Deep Restricted Kernel Machines using Conjugate Feature Duality

Johan A.K. Suykens
KU Leuven, ESAT-STADIUS
Kasteelpark Arenberg 10
B-3001 Leuven (Heverlee), Belgium
Email: johan.suykens@esat.kuleuven.be

March 7, 2016

Abstract

The aim of this paper is to propose a theory of *Deep Restricted Kernel Machines* offering new foundations for deep learning with kernel machines. From the viewpoint of deep learning it is partially related to restricted Boltzmann machines, which are characterized by visible and hidden units in a bipartite graph without hidden-to-hidden connections, and deep learning extensions as deep belief networks and deep Boltzmann machines. From the viewpoint of kernel machines it includes least squares support vector machines for classification and regression, kernel PCA, matrix SVD, and Parzen-type models. A key element in this paper is to characterize these kernel machines in terms of so-called *conjugate feature duality*. It is shown how this is related to the energy form in restricted Boltzmann machines, with continuous variables in a nonprobabilistic setting. In this new framework of so-called *Restricted Kernel Machine* (RKM) representations the role of dual variables becomes played by the hidden features. Deep Restricted Kernel Machines (Deep RKM) are obtained by coupling the Restricted Kernel Machines. It leads to a learning algorithm that consists of forward and backward phases. It also reveals how kernel fusion takes place at the different levels. The method is illustrated for Deep Restricted Kernel Machines consisting of three levels with a LS-SVM regression or classification level and two additional kernel PCA levels.

Keywords. deep learning, kernel methods, least squares support vector machines, kernel PCA, restricted Boltzmann machines, duality.

1 Introduction

Deep learning has become an important method of choice in several application fields such as computer vision, speech recognition, language processing and others [20]. Among the existing techniques in deep learning are deep belief networks, deep Boltzmann machines, convolutional neural networks, stacked autoencoders with pretraining and finetuning [4, 15, 16, 20, 21, 30, 33, 38]. On the other hand also support vector machines (SVM) and kernel based methods have made a large impact in a very wide range of application fields, together with finding strong foundations in optimization and learning theory [5, 8, 26, 35, 44, 49, 50]. Therefore one can pose the question: which synergies or common foundations could be developed between these different directions? One has started already to explore such synergies e.g. in kernel methods for deep learning [7], deep Gaussian processes [9, 29], convolutional kernel networks [22], multi-layer support vector machines [52], mathematics of the neural response [36], and others.

In this paper we intend to present a new theory of *Deep Restricted Kernel Machines* (Deep RKM) offering foundations for deep learning with kernel machines. It partially relates to restricted Boltzmann machines which are used within deep belief networks [15, 16]. In restricted Boltzmann machines (RBM) one considers a specific type of Markov random field, characterized by a bipartite graph consisting of a layer of visible units and another layer of hidden units [4, 11, 16, 30]. In RBMs, which are related to harmoniums [37, 51], one has no connections between the hidden units [15]. In deep belief networks the hidden units of a layer are mapped to a next layer in order to create a deep architecture. In RBM one considers stochastic binary variables [1, 14], and extensions have been made to Gaussian-Bernouilli variants [30]. On the other hand Hopfield networks [17] take continuous values and a class of Hamiltonian neural networks has been studied in [10]. One has also studied discriminative RBMs where the class labels are considered at the level of visible units [11, 18]. In all these methods the energy function plays an important role, also in energy based learning methods [19].

Representation learning issues are considered to be important in deep learning [3]. The method that is proposed in this paper will make a link to restricted Boltzmann machines by characterizing several kernel machines by means of so-called *conjugate feature duality*. Duality is important in the context of support vector machines [5, 8, 49, 44, 46], optimization [6, 27] and also in mathematics and physics in general. Here we will consider hidden features conjugated to some part of the unknown variables. This part of the formulation is linked to a restricted Boltzmann machine energy expression, though with continuous variables in a nonprobabilistic setting. In this way one can express a model both in its primal representation and in its dual representation, and give an interpretation of visible and hidden units, in analogy with RBM. The primal representation contains the feature map, while the dual model representation is expressed in terms of the kernel function and the conjugated features. The class of kernel machines discussed in this paper includes least squares support vector machines (LS-SVM) for classification and regression, kernel

principal component analysis (kernel PCA), matrix singular value decomposition (matrix SVD), and Parzen-type models. These have been previously conceived within a primal and Lagrange dual setting in [42, 44, 45, 47, 48]. Other examples are kernel spectral clustering [2, 23], kernel canonical correlation analysis [44] and several others, which will not be addressed in this paper but can be the subject of future work. In this paper we will give a different characterization for these models. It is based on a property of quadratic forms, which can be verified through the Schur complement form. The property relates to a specific case of Legendre-Fenchel duality [27]. Also note that in classical mechanics converting a Lagrangian into Hamiltonian formulation is by Legendre transformation [12].

The kernel machines with conjugate feature representations are used then as building blocks to obtain the Deep RKM by coupling the RKMs. The Deep RKM becomes unrestricted after coupling the RKMs. The approach is explained for a model with 3 levels, where in level 1 a LS-SVM regression is taken, while in level 2 and 3 kernel PCA levels. The objective of the Deep RKM is then the sum of the objectives of the RKMs from the different levels. The conjugate features of level 1 are taken as input of level 2, and subsequently the features of level 2 as input for level 3. The characterization of the stationary points leads to solving a set of nonlinear equations in the unknowns, which is computationally expensive. However, for the case of linear kernels in level 2 and 3 it leads to an algorithm with forward and backward phases propagating over the different levels. It also reveals how kernel fusion is taking place over the different levels. Note that also the classical backpropagation algorithm for training multilayer feedforward neural networks [28] has a forward and a backward phase with error backpropagation. In the application of dynamic backpropagation to multilayer recurrent neural networks it was possible to impose conditions for stability in order to learn stable multilayer recurrent networks [40].

In this paper we make a distinction between *levels* and *layers*. We use the terminology of *levels* to indicate the depth of the model. The terminology of *layers* is used here in connection to the feature map. It was shown in [41] how a multilayer perceptron can be trained by a support vector machine method. It is done by defining the hidden layer to be equal to the feature map. In this way the hidden layer is treated at the feature map and at the kernel parameters level. Also in [44] it has been explained that in support vector machine and least squares support vector machine models, one can have a neural networks interpretation both in the primal and in the dual. The number of hidden units in the primal equals the dimension of the feature space, while in the dual representation it equals the number of support vectors.

This paper is organized as follows. In Section 2 the context of this paper is outlined with a brief introductory part on restricted Boltzmann machines, support vector machines, least squares support vector machines, kernel PCA and singular value decomposition. In Section 3 we explain how these kernel machines can be characterized by conjugate feature duality. In Section 4 deep restricted kernel machines are explained for 3 levels consisting of a LS-SVM regression level and two additional kernel PCA levels. Illustrations with numerical examples are given in Section 5. Finally, Section 6 concludes the paper.

2 Preliminaries and Context

2.1 Restricted Boltzmann machines (RBM)

A restricted Boltzmann machine is a specific type of Markov random field, characterized by a bipartite graph consisting of a layer of visible units and another layer of hidden units [4, 11, 16, 30], without hidden-to-hidden connections. Both the visible and hidden variables, denoted by v and h respectively, have stochastic binary units with value 0 or 1. A joint state $\{v, h\}$ is defined for these visible and hidden variables with energy (see Figure 1)

$$E(v, h; \theta) = -v^T W h - c^T v - a^T h \tag{1}$$

where $\theta = \{W, c, a\}$ are the model parameters, W is an interaction weight matrix and c, a contain bias terms.

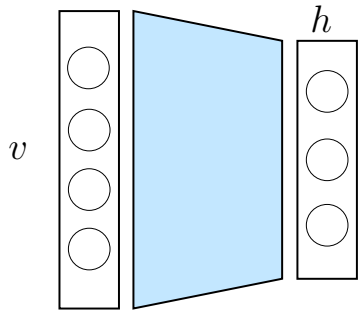

Figure 1: Restricted Boltzmann machine (RBM) consisting of a layer of visible units v and a layer of hidden units h. They are interconnected through the interaction matrix W, depicted in blue color.

One obtains then the joint distribution over the visible and hidden units as

$$P(v, h; \theta) = \frac{1}{Z(\theta)} \exp(-E(v, h; \theta)) \tag{2}$$

with the partition function

$$Z(\theta) = \sum_v \sum_h \exp(-E(v, h; \theta))$$

for normalization.

Thanks to the specific bipartite structure one can obtain an explicit expression for the marginalization $P(v; \theta) = \frac{1}{Z(\theta)} \sum_h \exp(-E(v, h; \theta))$. The conditional distributions are obtained as

$$\begin{aligned} P(h|v; \theta) &= \prod_j p(h_{(j)}|v) \\ P(v|h; \theta) &= \prod_i p(v_{(i)}|h) \end{aligned} \tag{3}$$

where $p(h_{(j)} = 1|v) = \sigma(\sum_i W_{ij} v_{(i)} + a_j)$ and $p(v_{(i)} = 1|h) = \sigma(\sum_j W_{ij} h_{(j)} + d_i)$ with $\sigma(x) = 1/(1+\exp(-x))$ the logistic function. Here $v_{(i)}$ and $h_{(j)}$ denote the i-th visible unit and the j-th hidden unit, respectively.

Because exact maximum likelihood for this model is intractable a contrastive divergence algorithm is used with the following update equation for the weights

$$\Delta W = \alpha(\mathbb{E}_{P_{\text{data}}}(vh^T) - \mathbb{E}_{P_T}(vh^T)) \tag{4}$$

with learning rate α and $\mathbb{E}_{P_{\text{data}}}$ the expectation wrt the data distribution $P_{\text{data}}(h, v; \theta) = P(h|v; \theta)P_{\text{data}}(v)$ where $P_{\text{data}}(v)$ denotes the empirical distribution. Furthermore $\mathbb{E}_{P_T}$ is a distribution defined by running a Gibbs chain for T steps initialized at the data. Often one takes $T = 1$, while $T \to \infty$ recovers the maximum likelihood approach [30].

In Boltzmann machines there are besides visible-to-hidden, also visible-to-visible, and hidden-to-hidden interaction terms with

$$E(v, h; \theta) = -v^T W h - \frac{1}{2} v^T L v - \frac{1}{2} h^T G h \tag{5}$$

and $\theta = \{W, L, G\}$ as explained in [31].

2.2 Least Squares Support Vector Machines and related Kernel Machines

2.2.1 SVM and LS-SVM

Assume a binary classification problem with training data $\{(x_i, y_i)\}_{i=1}^N$ with input data $x_i \in \mathbb{R}^d$ and corresponding class labels $y_i \in \{-1, 1\}$. A support vector machine classifier takes the form

$$\hat{y} = \text{sign}[w^T \varphi(x) + b]$$

where the feature map $\varphi(\cdot) : \mathbb{R}^d \to \mathbb{R}^{n_f}$ maps the data from the input space to a high dimensional feature space and $\hat{y}$ is the estimated class label for a given input point $x \in \mathbb{R}^d$. The training problem for this SVM classifier [5, 8, 49] is

$$\begin{array}{ll} \min\limits_{w,b,\xi} & \frac{1}{2} w^T w + \mathsf{c} \sum\limits_{i=1}^{N} \xi_i \\ \text{subject to} & y_i[w^T \varphi(x_i) + b] \geq 1 - \xi_i, \quad i = 1, ..., N \\ & \xi_i \geq 0, \quad i = 1, ..., N \end{array} \tag{6}$$

where the objective function makes a trade-off between minimization of the regularization term (corresponding to maximization of the margin $2/\|w\|_2$) and the amount of misclassifications, controlled by the regularization constant $\mathsf{c} > 0$. The slack variables ξ_i are needed to tolerate misclassifications on the training data, in order to avoid that one would overfit

the data. The following dual problem in the Lagrange multipliers α_i is obtained, related to the first set of constraints:

$$\begin{array}{ll} \max\limits_{\alpha} & -\frac{1}{2}\sum\limits_{i,j=1}^{N} y_i y_j\, K(x_i,x_j)\,\alpha_i\alpha_j + \sum\limits_{j=1}^{N}\alpha_j \\ \text{subject to} & \sum_{i=1}^{N}\alpha_i y_i = 0 \\ & 0 \leq \alpha_i \leq \mathsf{c},\ i=1,...,N. \end{array} \tag{7}$$

Here a positive definite kernel K is used with $K(x,z) = \varphi(x)^T\varphi(z) = \sum_{j=1}^{n_f}\varphi_j(x)\varphi_j(z)$. The SVM classifier is expressed in the dual as

$$\hat{y} = \text{sign}[\sum_{i\in\mathcal{S}_{SV}} \alpha_i\, y_i\, K(x_i,x) + b], \tag{8}$$

where $\mathcal{S}_{SV}$ denotes the set of support vectors, corresponding to the non-zero α_i values. Common choices are e.g. to take a linear $K(x_i,x_j) = x_i^T x_j$, polynomial $K(x_i,x_j) = (\nu + x_i^T x_j)^d$ with $\nu \geq 0$ or Gaussian radial basis function kernel $K(x_i,x_j) = \exp(-\|x_i - x_j\|_2^2/\sigma^2)$.

The LS-SVM classifier [42] is a modification to it

$$\begin{array}{ll} \min\limits_{w,b,e_i} & \frac{1}{2}w^T w + \gamma\,\frac{1}{2}\sum\limits_{i=1}^{N} e_i^2 \\ \text{subject to} & y_i[w^T\varphi(x_i) + b] = 1 - e_i, \quad i=1,...,N, \end{array} \tag{9}$$

where the value 1 in the constraints is taken as a target value, instead of a threshold value. This implicitly corresponds to a regression on the class labels ± 1. From the Lagrangian $\mathcal{L}(w,b,e;\alpha) = \frac{1}{2}w^T w + \gamma\,\frac{1}{2}\sum_{i=1}^{N} e_i^2 - \sum_{i=1}^{N}\alpha_i\{y_i[w^T\varphi(x_i) + b] - 1 + e_i\}$ one takes the conditions for optimality $\partial\mathcal{L}/\partial w = 0$, $\partial\mathcal{L}/\partial b = 0$, $\partial\mathcal{L}/\partial e_i = 0$, $\partial\mathcal{L}/\partial\alpha_i = 0$. Writing the solution in α, b gives the square linear system

$$\left[\begin{array}{c|c} \Omega + I/\gamma & y_{1:N} \\ \hline y_{1:N}^T & 0 \end{array}\right]\left[\begin{array}{c} \alpha \\ \hline b \end{array}\right] = \left[\begin{array}{c} 1_N \\ \hline 0 \end{array}\right] \tag{10}$$

where $\Omega_{ij} = y_i y_j\,\varphi(x_i)^T\varphi(x_j) = y_i y_j\, K(x_i,x_j)$ and $y_{1:N} = [y_1;...;y_N]$, $1_N = [1;...;1]$ with as classifier in the dual

$$\hat{y} = \text{sign}[\sum_{i=1}^{N}\alpha_i y_i K(x_i,x) + b]. \tag{11}$$

This formulation has also been extended to multiclass problems [43, 44].

In the LS-SVM regression formulation [44] one performs ridge regression in the feature space with an additional bias term b

$$\begin{array}{ll} \min\limits_{w,b,e_i} & \frac{1}{2}w^T w + \gamma\,\frac{1}{2}\sum\limits_{i=1}^{N} e_i^2 \\ \text{subject to} & y_i = w^T\varphi(x_i) + b + e_i, \quad i=1,...,N \end{array} \tag{12}$$

which gives

$$\left[\begin{array}{c|c} K + I/\gamma & 1_N \\ \hline 1_N^T & 0 \end{array}\right] \left[\begin{array}{c} \alpha \\ \hline b \end{array}\right] = \left[\begin{array}{c} y_{1:N} \\ \hline 0 \end{array}\right] \tag{13}$$

with the predicted output

$$\hat{y} = \sum_{i=1}^{N} \alpha_i K(x_i, x) + b \tag{14}$$

where $K_{ij} = K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$. The classifier formulation can also be transformed into the regression formulation, by multiplying the constraints in (9) by the class labels and considering new error variables [44]. In the zero bias term case this corresponds to kernel ridge regression [32], which is also related to function estimation in reproducing kernel Hilbert spaces, regularization networks and Gaussian processes, within a different setting [25, 50, 26, 44].

2.2.2 Kernel PCA and matrix SVD

Within the setting of using equality constraints and L_2 loss functions, typical for LS-SVMs, one can characterize the kernel PCA problem [34] as follows, as shown in [44, 45]

$$\begin{array}{ll} \min\limits_{w,b,e_i} & \frac{1}{2} w^T w - \gamma \frac{1}{2} \sum\limits_{i=1}^{N} e_i^2 \\ \text{subject to} & e_i = w^T \varphi(x_i) + b, \quad i = 1, ..., N. \end{array} \tag{15}$$

From the KKT conditions one obtains the following in the Lagrange multipliers α_i

$$K^{(c)} \alpha = \lambda \alpha \quad \text{with} \quad \lambda = 1/\gamma \tag{16}$$

where $K_{ij}^{(c)} = (\varphi(x_i) - \hat{\mu}_\varphi)^T (\varphi(x_j) - \hat{\mu}_\varphi)$ are the elements of the centered kernel matrix $K^{(c)}$, $\hat{\mu}_\varphi = (1/N) \sum_{i=1}^{N} \varphi(x_i)$ and $\alpha = [\alpha_1; ....; \alpha_N]$. In (15) maximizing instead of minimizing also leads to (16). The centering of the kernel matrix is obtained as a result of taking a bias term b in the model. The γ value is treated at a selection level and is chosen so as to correspond to $\lambda = 1/\gamma$ where λ are eigenvalues of $K^{(c)}$. In the zero bias term case $K^{(c)}$ becomes the kernel matrix $K = [\varphi(x_i)^T \varphi(x_j)]$. Also kernel spectral clustering [2, 46] was obtained in this setting by considering a weighted version of the L_2 loss part, weighted by the inverse of the degree matrix of the graph in the clustering problem.

It has been recently shown in [48] that matrix SVD can be obtained from the following primal problem:

$$\begin{array}{ll} \min\limits_{w,v,e,r} & -w^T v + \gamma \frac{1}{2} \sum\limits_{i=1}^{N} e_i^2 + \gamma \frac{1}{2} \sum\limits_{j=1}^{M} r_j^2 \\ \text{subject to} & e_i = w^T \varphi(x_i), \ i = 1, ..., N \\ & r_j = v^T \psi(z_j), \ j = 1, ..., M \end{array} \tag{17}$$

where $\{x_i\}_{i=1}^N$ and $\{z_j\}_{j=1}^M$ are data sets related to two data sources, which in the matrix SVD [13, 39] case correspond to the sets of rows and columns of the given matrix. Here one has two feature maps $\varphi(\cdot)$ and $\psi(\cdot)$. After taking the Lagrangian and the necessary conditions for optimality, the dual problem in the Lagrange multipliers α_i and β_j, related to the first and second set of constraints, results into

$$\begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \lambda \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \tag{18}$$

where $A = [\varphi(x_i)^T \psi(z_j)]$ denotes the matrix with ij-th entry $\varphi(x_i)^T \psi(z_j)$, $\lambda = 1/\gamma$ correspond to non-zero eigenvalues and $\alpha = [\alpha_1; ....; \alpha_N]$, $\beta = [\beta_1; ....; \beta_M]$. For a given matrix A, by choosing the linear feature maps $\varphi(x_i) = C^T x_i$, $\psi(z_j) = z_j$ with a compatibility matrix C that satisfies $ACA = A$, this eigenvalue problem corresponds to the SVD of matrix A [48] in connection to Lanczos' decomposition theorem. One can also see that for a symmetric matrix the two data sources coincide and the objective of (17) reduces then to the kernel PCA objective (15) [48], involving only one feature map instead of two feature maps.

3 Restricted Kernel Machines (RKM) and Conjugate Feature Duality

3.1 LS-SVM regression as a restricted kernel machine: linear case

A training data set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ is assumed to be given with input data $x_i \in \mathbb{R}^d$ and output data $y_i \in \mathbb{R}^p$ (now with p outputs), where the data are assumed to be i.i.d. and drawn from an unknown but fixed underlying distribution $P(x, y)$, which is a common assumption made in statistical learning theory [49].

We will explain now how LS-SVM regression can be linked to the energy form expression of a restricted Boltzmann machine, with an interpretation in terms of hidden and visual units. In view of these connections with RBMs and the fact that there will be no hidden-to-hidden connections, we will call it a Restricted Kernel Machine (RKM) representation, when this particular interpretation of the model is made. For LS-SVM regression, the part in the RKM interpretation that will take a similar form as the RBM energy function is

$$\begin{aligned} R_{\text{RKM}}(v, h) &= -v^T \tilde{W} h \\ &= -(x^T W h + b^T h - y^T h) \\ &= e^T h \end{aligned} \tag{19}$$

with a vector of hidden units $h \in \mathbb{R}^p$ and a vector of visible units $v \in \mathbb{R}^{n_v}$ with $n_v = d+1+p$

equal to

$$v = \begin{bmatrix} x \\ 1 \\ -y \end{bmatrix} \text{ and } \tilde{W} = \begin{bmatrix} W \\ b^T \\ I_p \end{bmatrix} \tag{20}$$

and $e = y - \hat{y}$ with $\hat{y} = W^T x + b$ the estimated output vector for a given input vector x where $e, y, \hat{y} \in \mathbb{R}^p$, $W \in \mathbb{R}^{d \times p}$, $b \in \mathbb{R}^p$. Note that b is treated as part of the interconnection matrix by adding a constant 1 within the vector v, which is also frequently done in the area of neural networks [40]. While in RBM the units are binary valued, in the RKM they are continuous valued. The notation "R" in $R_{\text{RKM}}(v,h)$ refers to the fact that the expression is restricted, i.e. there are no hidden-to-hidden connections.

For the training problem, the sum is taken over the training data $\{(x_i, y_i)\}_{i=1}^N$ with

$$\begin{aligned} R_{\text{RKM}}^{\text{train}} &= \sum_{i=1}^N R_{\text{RKM}}(v_i, h_i) \\ &= -\sum_{i=1}^N (x_i^T W h_i + b^T h_i - y_i^T h_i) \\ &= \sum_{i=1}^N e_i^T h_i. \end{aligned} \tag{21}$$

Note that we will adopt the following notation $h_{(j),i}$ to denote the value of the j-th unit for the i-th data point, and $e_i, h_i \in \mathbb{R}^p$ for $i = 1, ..., N$.

We start now from the LS-SVM regression training problem (12) but for the multiple outputs case. We express the objective in terms of $R_{\text{RKM}}^{\text{train}}$ and show how the hidden units can be introduced. Defining $\lambda = 1/\gamma > 0$ we obtain

$$\begin{aligned} J &= \frac{\eta}{2}\text{Tr}(W^T W) + \frac{1}{2\lambda}\sum_{i=1}^N e_i^T e_i \text{ s.t. } e_i = y_i - W^T x_i - b, \forall i \\ &\geq \sum_{i=1}^N e_i^T h_i - \frac{\lambda}{2}\sum_{i=1}^N h_i^T h_i + \frac{\eta}{2}\text{Tr}(W^T W) \text{ s.t. } e_i = y_i - W^T x_i - b, \forall i \\ &= \sum_{i=1}^N (y_i^T - x_i^T W - b^T) h_i - \frac{\lambda}{2}\sum_{i=1}^N h_i^T h_i + \frac{\eta}{2}\text{Tr}(W^T W) \triangleq \underline{J} \\ &= R_{\text{RKM}}^{\text{train}} - \frac{\lambda}{2}\sum_{i=1}^N h_i^T h_i + \frac{\eta}{2}\text{Tr}(W^T W) \end{aligned} \tag{22}$$

where λ, η are positive regularization constants and the first term corresponds to $R_{\text{RKM}}^{\text{train}}$. $\underline{J}$ denotes the lower bound on J. [1] This is based on the property that for two arbitrary

[1] Note that also the term $-\frac{\lambda}{2}\sum_{i=1}^N h_i^T h_i$ appears. This would in a Boltzmann machine energy correspond to matrix G equal to the identity matrix. The term $\frac{\eta}{2}\text{Tr}(W^T W)$ is an additional regularization term.

vectors e, h one has

$$\frac{1}{2\lambda}e^T e \geq e^T h - \frac{\lambda}{2}h^T h, \ \forall e, h \in \mathbb{R}^p. \tag{23}$$

The maximal value of the rhs in (23) is obtained for $h = e/\lambda$ which follows from $\partial(e^T h - \frac{\lambda}{2}h^T h)/\partial h = 0$ and $\partial^2(e^T h - \frac{\lambda}{2}h^T h)/\partial h^2 = -\lambda I < 0$. The maximal value that can be obtained for the rhs equals the lhs $\frac{1}{2\lambda}e^T e$. The property (23) can also be verified by writing it in the quadratic form

$$\frac{1}{2}\left[e^T \ h^T\right]\left[\begin{array}{cc} \frac{1}{\lambda}I & I \\ I & \lambda I \end{array}\right]\left[\begin{array}{c} e \\ h \end{array}\right] \geq 0, \ \forall e, h \in \mathbb{R}^p, \tag{24}$$

which holds. This follows immediately from the Schur complement form[2] which results into the condition $\frac{1}{2}(\lambda I - I(\lambda I)I \geq 0)$, which holds. Writing (23) as

$$\frac{1}{2\lambda}e^T e + \frac{\lambda}{2}h^T h \geq e^T h \tag{25}$$

gives a property that is also known in Legendre-Fenchel duality for the case of a quadratic function [27]. Furthermore it also follows from (23) that

$$\frac{1}{2\lambda}e^T e = \max_h (e^T h - \frac{\lambda}{2}h^T h). \tag{26}$$

We will call the method of introducing the hidden features h_i into (22) *conjugate feature duality*, where the hidden features h_i are conjugated to the e_i. Here $R^{\text{train}}_{\text{RKM}} = \sum_i e_i^T h_i$ will be called an *inner pairing* between the e_i and the hidden features h_i (see Figure 2).

We proceed now by looking at the stationary points of $\underline{J}(h_i, W, b)$: [3]

$$\left\{\begin{array}{lcl} \frac{\partial \underline{J}}{\partial h_i} = 0 & \Rightarrow & y_i = W^T x_i + b + \lambda h_i, \ \forall i \\ \frac{\partial \underline{J}}{\partial W} = 0 & \Rightarrow & W = \frac{1}{\eta}\sum_i x_i h_i^T \\ \frac{\partial \underline{J}}{\partial b} = 0 & \Rightarrow & \sum_i h_i = 0. \end{array}\right. \tag{27}$$

The first condition yields $h_i = e_i/\lambda$ which means that the maximal value of $\underline{J}$ is reached. Also note the similarity between the condition $W = \frac{1}{\eta}\sum_i x_i h_i^T$ and the expression (4) in

[2]This states that for a matrix $Q = \left[\begin{array}{cc} A & B \\ B^T & C \end{array}\right]$, one has $Q \geq 0$ if and only if $A > 0$ and the Schur complement $C - B^T A^{-1} B \geq 0$. [6]

[3]The following properties are used throughout this paper: $\frac{\partial \text{Tr}(X^T B X)}{\partial X} = BX + B^T X, \frac{\partial a^T X b}{\partial X} = ab^T, \frac{\partial a^T X^T b}{\partial X} = ba^T, \frac{\partial \text{Tr}(X^T A)}{\partial X} = A, \frac{\partial \text{Tr}(AX^T)}{\partial X} = A, \frac{\partial \text{Tr}(XA)}{\partial X} = A^T, \frac{\partial x^T a}{\partial x} = \frac{\partial a^T x}{\partial x} = a, a^T a = \text{Tr}(aa^T)$ for matrices A, B, X and vectors a, b.[24]

the contrastive divergence algorithm. Elimination of h_i from this set of conditions gives the solution in W, b:

$$\left[\begin{array}{c|c} \sum_i x_i x_i^T + \lambda\eta I_d & \sum_i x_i \\ \hline \sum_i x_i^T & N \end{array}\right]\left[\begin{array}{c} W \\ \hline b^T \end{array}\right] = \left[\begin{array}{c} \sum_i x_i y_i^T \\ \hline \sum_i y_i^T \end{array}\right]. \tag{28}$$

Elimination of W from the set of conditions gives the solution in h_i, b:

$$\left[\begin{array}{c|c} \frac{1}{\eta}[x_i^T x_j] + \lambda I_N & 1_N \\ \hline 1_N^T & 0 \end{array}\right]\left[\begin{array}{c} H^T \\ \hline b^T \end{array}\right] = \left[\begin{array}{c} Y^T \\ \hline 0 \end{array}\right] \tag{29}$$

with $[x_i^T x_j]$ denoting the matrix with ij-entry $x_i^T x_j$, $H = [h_1 ... h_N] \in \mathbb{R}^{p\times N}$, $Y = [y_1 ... y_N] \in \mathbb{R}^{p\times N}$. From this square linear system one can solve $\{h_i\}$ and b. 1_N denotes a vector of all ones of size N, I_N the identity matrix of size $N \times N$.

It is remarkable to see here that the hidden features h_i take the same role as the Lagrange dual variables α_i in the LS-SVM formulation based on Lagrange duality (13) when taking $\eta = 1$ and $p = 1$. For the estimated values $\hat{y}_i$ on the training data one can express the model then in terms of W, b or in terms of h_i, b. In the restricted kernel machine interpretation of the LS-SVM regression one has the following primal and dual model representations:

$$\begin{array}{ccc} & \nearrow & (P)_{\text{RKM}}: \quad \hat{y} = W^T x + b \\ \mathcal{M} & & \\ & \searrow & (D)_{\text{RKM}}: \quad \hat{y} = \frac{1}{\eta}\sum_j h_j x_j^T x + b \end{array} \tag{30}$$

evaluated at a point x, where the primal representation is in terms of W, b and the dual representation is in the hidden features h_i.

3.2 Nonlinear case

The extension to the general nonlinear case goes by replacing x_i by $\varphi(x_i)$ where $\varphi(x_i) : \mathbb{R}^d \to \mathbb{R}^{n_f}$ denotes the feature map, with n_f the dimension of the feature space. Therefore the objective function for the RKM interpretation becomes

$$\underline{J} \;=\; \sum_{i=1}^{N}(y_i^T - \varphi(x_i)^T W - b^T)h_i - \frac{\lambda}{2}\sum_{i=1}^{N} h_i^T h_i + \frac{\eta}{2}\text{Tr}(W^T W) \tag{31}$$

with vector of visible units $v \in \mathbb{R}^{n_v}$ with $n_v = n_f + 1 + p$ equal to

$$v = \left[\begin{array}{c} \varphi(x) \\ 1 \\ -y \end{array}\right]. \tag{32}$$

Following the same approach as in the linear case, one obtains then as solution in the primal

$$\left[\begin{array}{c|c} \sum_j \varphi(x_j)\varphi(x_j)^T + \lambda\eta I_{n_f} & \sum_j \varphi(x_j) \\ \hline \sum_j \varphi(x_j)^T & N \end{array}\right]\left[\begin{array}{c} W \\ \hline b^T \end{array}\right] = \left[\begin{array}{c} \sum_j \varphi(x_j) y_j^T \\ \hline \sum_j y_j^T \end{array}\right]. \tag{33}$$

In the conjugate feature dual one obtains the same linear system as (29) but with the positive definite kernel $K(x_i, x_j) = \varphi(x_i)^T\varphi(x_j)$ instead of the linear kernel $x_i^T x_j$:

$$\left[\begin{array}{c|c} \frac{1}{\eta}K + \lambda I_N & 1_N \\ \hline 1_N^T & 0 \end{array}\right]\left[\begin{array}{c} H^T \\ \hline b^T \end{array}\right] = \left[\begin{array}{c} Y^T \\ \hline 0 \end{array}\right]. \tag{34}$$

We will also employ the notation $[K(x_i, x_j)]$ to denote the kernel matrix K with ij-th entry equal to $K(x_i, x_j)$.

The primal and dual model representations are expressed in terms of the feature map and kernel function, respectively

$$\begin{array}{ccc} & \nearrow & (P)_{\mathrm{RKM}}: \quad \hat{y} = W^T\varphi(x) + b \\ \mathcal{M} & & \\ & \searrow & (D)_{\mathrm{RKM}}: \quad \hat{y} = \frac{1}{\eta}\sum_j h_j K(x_j, x) + b. \end{array} \tag{35}$$

One can define the feature map either in an implicit or in an explicit way. When employing a positive definite kernel function $K(\cdot,\cdot)$, according to the Mercer theorem, there exists a feature map φ such that $K(x_i, x_j) = \varphi(x_i)^T\varphi(x_j)$ holds. On the other hand one could also explicitly define an expression for φ and construct the kernel function according to $K(x_i, x_j) := \varphi(x_i)^T\varphi(x_j)$. For multilayer perceptrons it was shown in [41] that the hidden layer can be chosen as the feature map. We can let it correspond to

$$\varphi_{\mathrm{FF}}(x) = \sigma(U_q\sigma(...\sigma(U_2\sigma(U_1 x + \beta_1) + \beta_2)...) + \beta_q) \tag{36}$$

related to a feedforward (FF) neural network with multi layers, with hidden layer matrices $U_1, U_2, ..., U_q$ and bias term vectors $\beta_1, \beta_2, ..., \beta_q$. By construction one obtains then $K_{\mathrm{FF}}(x_i, x_j) := \varphi_{\mathrm{FF}}(x_i)^T\varphi_{\mathrm{FF}}(x_j)$. Note that the activation function σ might be different also for each of the hidden layers. A common choice is a sigmoid or hyperbolic tangent function. Within the context of this paper $U_1, ..., U_q, \beta_1, ..., \beta_q$ are treated then at the feature map level and the kernel parameter level.

As explained in [44] one can also give a neural network interpretation both to the primal and to the dual representation, with a number of hidden units equal to the dimension of the feature space for the primal representation and the number of support vectors in the dual representation, respectively. For the case of a Gaussian RBF kernel one has a one hidden layer interpretation then with an infinite amount of hidden units in the primal, while in the dual the number of hidden units equals the number of support vectors.

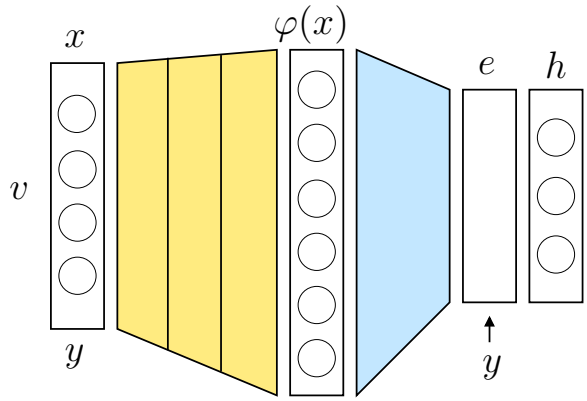

Figure 2: Restricted kernel machine (RKM) representation for regression: the feature map $\varphi(x)$ maps the input vector x to a feature space (possibly by multi layers, depicted in yellow color) and the hidden features are obtained through an inner pairing $e^T h$ where $e = y - \hat{y}$ compares the given output vector y with the predictive model output vector $\hat{y} = W^T\varphi(x) + b$, where the interconnection matrix W is depicted in blue color.

3.3 Classifier formulation

In the multi-class case the LS-SVM classifier constraints are

$$D_{y_i}(W^T\varphi(x_i) + b) = 1_p - e_i, i = 1, ..., N \tag{37}$$

where $y_i \in \{-1, 1\}^p, e_i \in \mathbb{R}^p$ with p outputs encoding the classes and diagonal matrix $D_{y_i} = \text{diag}\{\ y_{(1),i}, ..., y_{(p),i}\}$.

In this case, starting from the LS-SVM classifier objective, one obtains

$$\begin{aligned} J &= \frac{\eta}{2}\text{Tr}(W^TW) + \frac{1}{2\lambda}\sum_{i=1}^{N} e_i^T e_i \ \text{ s.t. } e_i = 1_p - D_{y_i}(W^T\varphi(x_i) + b), \forall i \\ &\geq \sum_{i=1}^{N} e_i^T h_i - \frac{\lambda}{2}\sum_{i=1}^{N} h_i^T h_i + \frac{\eta}{2}\text{Tr}(W^TW) \ \text{ s.t. } e_i = 1_p - D_{y_i}(W^T\varphi(x_i) + b), \forall i \\ &= \sum_{i=1}^{N} \left(1_p^T - (\varphi(x_i)^TW + b^T)D_{y_i}\right) h_i - \frac{\lambda}{2}\sum_{i=1}^{N} h_i^T h_i + \frac{\eta}{2}\text{Tr}(W^TW) \triangleq \underline{J}. \end{aligned} \tag{38}$$

The stationary points of $\underline{J}(h_i, W, b)$ are given by

$$\begin{cases} \dfrac{\partial \underline{J}}{\partial h_i} = 0 \ \Rightarrow \ 1_p = D_{y_i}(W^T\varphi(x_i) + b) + \lambda h_i, \ \forall i \\ \dfrac{\partial \underline{J}}{\partial W} = 0 \ \Rightarrow \ W = \dfrac{1}{\eta}\displaystyle\sum_i \varphi(x_i)h_i^T D_{y_i} \\ \dfrac{\partial \underline{J}}{\partial b} = 0 \ \Rightarrow \ \displaystyle\sum_i D_{y_i} h_i = 0. \end{cases} \tag{39}$$

The solution in the conjugate features follows then from the linear system:

$$\left[\begin{array}{c|c} \frac{1}{\eta}K + \lambda I_N & 1_N \\ \hline 1_N^T & 0 \end{array}\right] \left[\begin{array}{c} H_D^T \\ \hline b^T \end{array}\right] = \left[\begin{array}{c} Y^T \\ \hline 0 \end{array}\right] \tag{40}$$

with $H_D = [D_{y_1}h_1 ... D_{y_N}h_N]$.

The primal and dual model representations are expressed in terms of the feature map and the kernel function, respectively

$$\begin{array}{ccc} & & (P)_{\text{RKM}}: \quad \hat{y} = \text{sign}[W^T\varphi(x) + b] \\ \mathcal{M} & \nearrow & \\ & \searrow & \\ & & (D)_{\text{RKM}}: \quad \hat{y} = \text{sign}[\frac{1}{\eta}\sum_j D_{y_j}h_j K(x_j, x) + b]. \end{array} \tag{41}$$

3.4 Kernel PCA

In the kernel PCA case we start from the objective in (15) and introduce the conjugate hidden features:

$$\begin{aligned} J &= \frac{\eta}{2}\text{Tr}(W^TW) - \frac{1}{2\lambda}\sum_{i=1}^N e_i^Te_i \text{ s.t. } e_i = W^T\varphi(x_i), \forall i \\ &\leq -\sum_{i=1}^N e_i^Th_i + \frac{\lambda}{2}\sum_{i=1}^N h_i^Th_i + \frac{\eta}{2}\text{Tr}(W^TW) \text{ s.t. } e_i = W^T\varphi(x_i), \forall i \\ &= -\sum_{i=1}^N \varphi(x_i)^TWh_i + \frac{\lambda}{2}\sum_{i=1}^N h_i^Th_i + \frac{\eta}{2}\text{Tr}(W^TW) \triangleq \overline{J} \\ &= -R_{\text{RKM}}^{\text{train}} + \frac{\lambda}{2}\sum_{i=1}^N h_i^Th_i + \frac{\eta}{2}\text{Tr}(W^TW) \end{aligned} \tag{42}$$

where the upper bound $\overline{J}$ is introduced now by relying on the same property as used in the regression/classification case: $\frac{1}{2\lambda}e^Te + \frac{\lambda}{2}h^Th \geq e^Th$, but employed in a different way. Note that

$$-\frac{1}{2\lambda}e^Te = \min_h(-e^Th + \frac{\lambda}{2}h^Th). \tag{43}$$

The minimal value for the rhs is obtained for $h = e/\lambda$ which equals the lhs in that case.

We then proceed by characterizing the stationary points of $\overline{J}(h_i, W)$:

$$\left\{\begin{array}{lcl} \frac{\partial \overline{J}}{\partial h_i} = 0 & \Rightarrow & W^T\varphi(x_i) = \lambda h_i, \ \forall i \\ \frac{\partial \overline{J}}{\partial W} = 0 & \Rightarrow & W = \frac{1}{\eta}\sum_i \varphi(x_i)h_i^T. \end{array}\right. \tag{44}$$

Note that the first condition yields $h_i = e_i/\lambda$. Therefore the minimum value of $\overline{J}$ is reached. Elimination of W gives the following solution in the conjugated features

$$\frac{1}{\eta} K H^T = H^T \Lambda \tag{45}$$

where $H = [h_1 ... h_N] \in \mathbb{R}^{s \times N}$ and $\Lambda = \text{diag}\{\lambda_1, ..., \lambda_s\}$ with $s \leq N$ the number of selected components. One can verify that the solutions corresponding to the different eigenvectors h_i and their corresponding eigenvalues λ_i all lead to the value $J = 0$.

The primal and dual model representations are

$$\begin{array}{ccc} & \nearrow & (P)_{\text{RKM}}: \quad \hat{e} = W^T \varphi(x) \\ \mathcal{M} & & \\ & \searrow & (D)_{\text{RKM}}: \quad \hat{e} = \frac{1}{\eta} \sum_j h_j K(x_j, x). \end{array} \tag{46}$$

Here the number of hidden units equals s with $h \in \mathbb{R}^s$ and the visible units $v \in \mathbb{R}^{n_f}$ with $v = \varphi(x)$, and $R_{\text{RKM}}(v, h) = -v^T W h$.

3.5 Singular value decomposition

For the SVD case we start from the objective in (17) and introduce the conjugated hidden features. The model is characterized now by matrices W, V:

$$\begin{aligned} J &= -\frac{\eta}{2}\text{Tr}(V^T W) + \frac{1}{2\lambda}\sum_{i=1}^{N} e_i^T e_i + \frac{1}{2\lambda}\sum_{j=1}^{M} r_j^T r_j \\ &\quad \text{s.t. } e_i = W^T\varphi(x_i), \forall i \ \& \ r_j = V^T\psi(z_j), \forall j \\ &\geq \sum_{i=1}^{N} e_i^T h_{e_i} - \frac{\lambda}{2}\sum_{i=1}^{N} h_{e_i}^T h_{e_i} + \sum_{j=1}^{M} r_j^T h_{r_j} - \frac{\lambda}{2}\sum_{j=1}^{M} h_{r_j}^T h_{r_j} - \frac{\eta}{2}\text{Tr}(V^T W) \\ &\quad \text{s.t. } e_i = W^T\varphi(x_i), \forall i \ \& \ r_j = V^T\psi(z_j), \forall j \\ &= \sum_{i=1}^{N} \varphi(x_i)^T W h_{e_i} - \frac{\lambda}{2}\sum_{i=1}^{N} h_{e_i}^T h_{e_i} + \sum_{j=1}^{M} \psi(z_j)^T V h_{r_j} - \frac{\lambda}{2}\sum_{j=1}^{M} h_{r_j}^T h_{r_j} - \frac{\eta}{2}\text{Tr}(V^T W) \triangleq \underline{J}. \end{aligned} \tag{47}$$

In this case $R^{\text{train}}_{\text{RKM}} = \sum_{i=1}^{N} \varphi(x_i)^T W h_{e_i} + \sum_{j=1}^{M} \psi(z_j)^T V h_{r_j}$. The stationary points of $\underline{J}(h_{e_i}, W, h_{r_j}, V)$ are given by:

$$\begin{cases} \frac{\partial \underline{J}}{\partial h_{e_i}} = 0 & \Rightarrow \quad W^T \varphi(x_i) = \lambda h_{e_i}, \ \forall i \\ \frac{\partial \underline{J}}{\partial W} = 0 & \Rightarrow \quad V = \frac{1}{\eta} \sum_i \varphi(x_i) h_{e_i}^T \\ \frac{\partial \underline{J}}{\partial h_{r_j}} = 0 & \Rightarrow \quad V^T \psi(z_j) = \lambda h_{r_j}, \ \forall j \\ \frac{\partial \underline{J}}{\partial V} = 0 & \Rightarrow \quad W = \frac{1}{\eta} \sum_j \psi(z_j) h_{r_j}^T. \end{cases} \tag{48}$$

Elimination of W, V gives the solution in the conjugated dual features h_{e_i}, h_{r_j}:

$$\left[\begin{array}{c|c} 0 & \frac{1}{\eta}[\varphi(x_i)^T \psi(z_j)] \\ \hline \frac{1}{\eta}[\psi(z_j)^T \varphi(x_i)] & 0 \end{array}\right] \left[\begin{array}{c} H_e^T \\ \hline H_r^T \end{array}\right] = \left[\begin{array}{c} H_e^T \\ \hline H_r^T \end{array}\right] \Lambda \tag{49}$$

with $H_e = [h_{e_1} ... h_{e_N}] \in \mathbb{R}^{s \times N}$, $H_r = [h_{r_1} ... h_{r_M}] \in \mathbb{R}^{s \times M}$ and $\Lambda = \text{diag}\{\lambda_1, ..., \lambda_s\}$ with $s \leq N + M$ a specified number of non-zero eigenvalues.

The primal and dual model representations are

$$\begin{array}{rll} & (P)_{\text{RKM}}: & \hat{e} = W^T \varphi(x) \\ \nearrow & & \hat{r} = V^T \psi(z) \\ \mathcal{M} & & \\ \searrow & & \\ & (D)_{\text{RKM}}: & \hat{e} = \frac{1}{\eta} \sum_j h_{r_j} \psi(z_j)^T \varphi(x) \\ & & \hat{r} = \frac{1}{\eta} \sum_i h_{e_i} \varphi(x_i)^T \psi(z). \end{array} \tag{50}$$

which corresponds to matrix SVD in the case of linear compatible feature maps and if an additional compatibility condition holds [48].

3.6 Kernel pmf

For the case of kernel probability mass function (kernel pmf) estimation [47], we start from the objective

$$J = \sum_{i=1}^{N} (p_i - \varphi(x_i)^T w) h_i - \sum_{i=1}^{N} p_i + \frac{\eta}{2} w^T w \tag{51}$$

in the unknowns $w \in \mathbb{R}^{n_f}$, $p_i \in \mathbb{R}$ and $h_i \in \mathbb{R}$. In [47] it was explained how a similar formulation is related to the probability rule in quantum measurement for a complex valued model.

The stationary points are characterized by

$$\begin{cases} \dfrac{\partial \underline{J}}{\partial h_j} = 0 & \Rightarrow \quad p_i = w^T \varphi(x_i), \ \forall i \\ \dfrac{\partial \underline{J}}{\partial w} = 0 & \Rightarrow \quad w = \dfrac{1}{\eta} \displaystyle\sum_i \varphi(x_i) h_i \\ \dfrac{\partial \underline{J}}{\partial p_i} = 0 & \Rightarrow \quad h_i = 1, \ \forall i. \end{cases} \tag{52}$$

The regularization constant η can be chosen to normalize $\sum_i p_i = 1$ ($p_i \geq 0$ is achieved by the choice of an appropriate kernel function [47]), which gives then the kernel pmf obtained in [47]. This results in the representations

$$\mathcal{M} \begin{cases} \nearrow & (P)_{\text{RKM}}: \quad p_i = w^T \varphi(x_i) \\ \searrow & (D)_{\text{RKM}}: \quad p_i = \dfrac{1}{\eta} \displaystyle\sum_j K(x_j, x_i). \end{cases} \tag{53}$$

4 Deep Restricted Kernel Machines

We explain now how deep restricted kernel machines can be obtained, based on the insights of the previous sections. We will focus here on the case of three levels where

- Level 1 consists of LS-SVM regression with given input data x_i and output data y_i and is characterized by conjugated features $h_i^{(1)}$;
- Level 2 consists of kernel PCA by taking $h_i^{(1)}$ as input and is characterized by conjugated features $h_i^{(2)}$;
- Level 3 consists of kernel PCA by taking $h_i^{(2)}$ as input and is characterized by conjugated features $h_i^{(3)}$.

We look then for the stationary points of

$$J_{\text{deep}} = \underline{J}_1 + \overline{J}_2 + \overline{J}_3 \tag{54}$$

where the objective $J_{\text{deep}}(h_i^{(1)}, W_1, b, h_i^{(2)}, W_2, h_i^{(3)}, W_3)$ for the Deep RKM consists of the sum of the objectives of levels 1,2,3 given by $\underline{J}_1$, $\overline{J}_2$, $\overline{J}_3$, respectively. Deep RKM consists of coupling the RKMs.

This becomes

$$\begin{aligned} J_{\text{deep}} &= \sum_{i=1}^{N}(y_i^T - \varphi_1(x_i)^T W_1 - b^T)h_i^{(1)} - \frac{\lambda_1}{2}\sum_{i=1}^{N} h_i^{(1)^T} h_i^{(1)} + \frac{\eta_1}{2}\text{Tr}(W_1^T W_1) \\ &-\sum_{i=1}^{N}\varphi_2(h_i^{(1)})^T W_2 h_i^{(2)} + \frac{\lambda_2}{2}\sum_{i=1}^{N} h_i^{(2)^T} h_i^{(2)} + \frac{\eta_2}{2}\text{Tr}(W_2^T W_2) \\ &-\sum_{i=1}^{N}\varphi_3(h_i^{(2)})^T W_3 h_i^{(3)} + \frac{\lambda_3}{2}\sum_{i=1}^{N} h_i^{(3)^T} h_i^{(3)} + \frac{\eta_3}{2}\text{Tr}(W_3^T W_3), \end{aligned} \tag{55}$$

with the following inner pairings at the three levels

$$\begin{aligned} \text{Level 1}: &\quad \sum_{i=1}^{N} e_i^{(1)^T} h_i^{(1)} = \sum_{i=1}^{N}(y_i^T - \varphi_1(x_i)^T W_1 - b^T)h_i^{(1)} \\ \text{Level 2}: &\quad \sum_{i=1}^{N} e_i^{(2)^T} h_i^{(2)} = \sum_{i=1}^{N}\varphi_2(h_i^{(1)})^T W_2 h_i^{(2)} \\ \text{Level 3}: &\quad \sum_{i=1}^{N} e_i^{(3)^T} h_i^{(3)} = \sum_{i=1}^{N}\varphi_3(h_i^{(2)})^T W_3 h_i^{(3)}. \end{aligned} \tag{56}$$

The level 1 part has feature map $\varphi_1 : \mathbb{R}^d \to \mathbb{R}^{n_{f_1}}$, $W_1 \in \mathbb{R}^{n_{f_1} \times p}$, the level 2 part $\varphi_2 : \mathbb{R}^p \to \mathbb{R}^{n_{f_2}}$, $W_2 \in \mathbb{R}^{n_{f_2} \times s^{(2)}}$, and the level 3 part $\varphi_3 : \mathbb{R}^{s^{(2)}} \to \mathbb{R}^{n_{f_3}}$, $W_3 \in \mathbb{R}^{n_{f_3} \times s^{(3)}}$.

Note that in J_{deep} the sum of the three inner pairing terms is similar to the energy in deep Boltzmann machines [30, 31] for the particular case of linear feature maps $\varphi_1, \varphi_2, \varphi_3$ and symmetric interaction terms. For the special case of linear feature maps one has

$$U_{\text{deep}} = -v^T \tilde{W}_1 h^{(1)} - h^{(1)^T} W_2 h^{(2)} - h^{(2)^T} W_3 h^{(3)} \tag{57}$$

which takes a same form as [30](eq.(29)), with $\tilde{W}_1$ defined in the sense of (19). The "U" in U_{deep} refers to the fact that the Deep RKM is unrestricted after coupling because of the hidden-to-hidden connections between layer 1 and 2 and between layer 2 and 3, while the uncoupled RKMs are restricted.

The stationary points of $J_{\text{deep}}(h_i^{(1)}, W_1, b, h_i^{(2)}, W_2, h_i^{(3)}, W_3)$ are given by

$$\begin{cases} \frac{\partial J_{\text{deep}}}{\partial h_i^{(1)}} = 0 \;\; \Rightarrow \;\; y_i - W_1^T \varphi_1(x_i) - b = \lambda_1 h_i^{(1)} + \frac{\partial}{\partial h_i^{(1)}}[\varphi_2(h_i^{(1)})^T W_2 h_i^{(2)}], \;\; \forall i \\ \frac{\partial J_{\text{deep}}}{\partial W_1} = 0 \;\; \Rightarrow \;\; W_1 = \frac{1}{\eta_1} \sum_i \varphi_1(x_i) h_i^{(1)^T} \\ \frac{\partial J_{\text{deep}}}{\partial b} = 0 \;\; \Rightarrow \;\; \sum_i h_i^{(1)} = 0 \\ \frac{\partial J_{\text{deep}}}{\partial h_i^{(2)}} = 0 \;\; \Rightarrow \;\; W_2^T \varphi_2(h_i^{(1)}) = \lambda_2 h_i^{(2)} - \frac{\partial}{\partial h_i^{(2)}}[\varphi_3(h_i^{(2)})^T W_3 h_i^{(3)}], \;\; \forall i \\ \frac{\partial J_{\text{deep}}}{\partial W_2} = 0 \;\; \Rightarrow \;\; W_2 = \frac{1}{\eta_2} \sum_i \varphi_2(h_i^{(1)}) h_i^{(2)^T} \\ \frac{\partial J_{\text{deep}}}{\partial h_i^{(3)}} = 0 \;\; \Rightarrow \;\; W_3^T \varphi_3(h_i^{(2)}) = \lambda_3 h_i^{(3)}, \;\; \forall i \\ \frac{\partial J_{\text{deep}}}{\partial W_3} = 0 \;\; \Rightarrow \;\; W_3 = \frac{1}{\eta_3} \sum_i \varphi_3(h_i^{(2)}) h_i^{(3)^T}. \end{cases} \tag{58}$$

By elimination of W_1, W_2, W_3 one obtains the following set of nonlinear equations in the conjugated features $h_i^{(1)}, h_i^{(2)}, h_i^{(3)}$ and b:

$$\begin{cases} y_i = \frac{1}{\eta_1} \sum_j h_j^{(1)} K_1(x_j, x_i) + b + \lambda_1 h_i^{(1)} + \frac{1}{\eta_2} \sum_j \frac{\partial K_2(h_i^{(1)}, h_j^{(1)})}{\partial h_i^{(1)}} h_j^{(2)^T} h_i^{(2)}, \;\; \forall i \\ \sum_i h_i^{(1)} = 0 \\ \frac{1}{\eta_2} \sum_j h_j^{(2)} K_2(h_j^{(1)}, h_i^{(1)}) = \lambda_2 h_i^{(2)} - \frac{1}{\eta_3} \sum_j \frac{\partial K_3(h_i^{(2)}, h_j^{(2)})}{\partial h_i^{(2)}} h_j^{(3)^T} h_i^{(3)}, \;\; \forall i \\ \frac{1}{\eta_3} \sum_j h_j^{(3)} K_3(h_j^{(2)}, h_i^{(2)}) = \lambda_3 h_i^{(3)}, \;\; \forall i. \end{cases} \tag{59}$$

Solving this set of nonlinear equations is computationally expensive. However, for the case of taking linear kernels K_2 and K_3, corresponding to linear kernel PCA levels at level 2 and 3, (59) simplifies to

$$\begin{cases} y_i = \sum_j h_j^{(1)} \left(\frac{1}{\eta_1} K_1(x_j, x_i) + \frac{1}{\eta_2} K_{\text{lin}}(h_j^{(2)}, h_i^{(2)}) \right) + b + \lambda_1 h_i^{(1)}, \;\; \forall i \\ \sum_i h_i^{(1)} = 0 \\ \sum_j h_j^{(2)} \left(\frac{1}{\eta_2} K_{2,\text{lin}}(h_j^{(1)}, h_i^{(1)}) + \frac{1}{\eta_3} K_{\text{lin}}(h_j^{(3)}, h_i^{(3)}) \right) = \lambda_2 h_i^{(2)}, \;\; \forall i \\ \sum_j h_j^{(3)} \left(\frac{1}{\eta_3} K_{3,\text{lin}}(h_j^{(2)}, h_i^{(2)}) \right) = \lambda_3 h_i^{(3)}, \;\; \forall i \end{cases} \tag{60}$$

where $K_{\text{lin}}, K_{2,\text{lin}}, K_{3,\text{lin}}$ denote linear kernels. In matrix-vector notation this corresponds to solving

$$\begin{cases} \text{Level 1}: & \left[\begin{array}{c|c} \frac{1}{\eta_1}[K_1(x_j,x_i)] + \frac{1}{\eta_2}[K_{\text{lin}}(h_j^{(2)},h_i^{(2)})] + \lambda_1 I_N & 1_N \\ \hline 1_N^T & 0 \end{array}\right] \left[\begin{array}{c} H_1^T \\ \hline b^T \end{array}\right] = \left[\begin{array}{c} Y^T \\ \hline 0 \end{array}\right] \\ \text{Level 2}: & \left(\frac{1}{\eta_2}[K_{2,\text{lin}}(h_j^{(1)},h_i^{(1)})] + \frac{1}{\eta_3}[K_{\text{lin}}(h_j^{(3)},h_i^{(3)})]\right) H_2^T = H_2^T \Lambda_2 \\ \text{Level 3}: & \left(\frac{1}{\eta_3}[K_{3,\text{lin}}(h_j^{(2)},h_i^{(2)})]\right) H_3^T = H_3^T \Lambda_3. \end{cases} \tag{61}$$

Here we denote $H_1 = [h_1^{(1)}...h_N^{(1)}]$, $H_2 = [h_1^{(2)}...h_N^{(2)}]$, $H_3 = [h_1^{(3)}...h_N^{(3)}]$. One sees that at level 1 and 2 a data fusion is taking place between K_1 and K_{lin} and between $K_{2,\text{lin}}$ and K_{lin}, where $\frac{1}{\eta_1}, \frac{1}{\eta_2}, \frac{1}{\eta_3}$ are specifying the relative weight given to each of these kernels. In this way one can choose for emphasizing or de-emphasizing the levels with respect to each other.

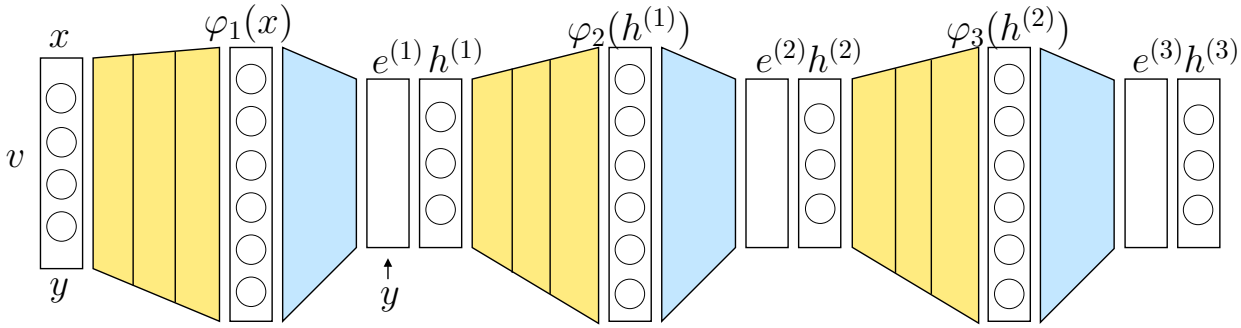

Figure 3: Deep restricted kernel machine (Deep RKM) consisting of 3 levels with LS-SVM regression in level 1 and kernel PCA in level 2 and level 3. The conjugate features of a certain layer are taken as input for the next layer.

In order to solve level 1 as a linear system one needs the input/output data $X = [x_1...x_N] \in \mathbb{R}^{d\times N}, Y = [y_1...y_N] \in \mathbb{R}^{p\times N}$, but one would also need the knowledge of $h_i^{(2)}$. Therefore an initialization phase is required. One can initialize $h_i^{(2)}$ as zero or at random at level 1, obtain then H_1 and propagate it to level 2. At level 2, after initializing H_3, one finds H_2, which is then propagated to level 3 where one computes H_3. After this *forward phase* one can go backward from level 3 to level 1 in a *backward phase*.

Schematically this gives the following heuristic algorithm:

$$\begin{array}{c}\textbf{Forward phase } (\text{level } 1 \rightarrow \text{level } 3)\\ \begin{array}{ll} & H_2, H_3 \text{ initialization}\\ \text{Level } 1: & H_1 := f_1(X, Y, H_2)\\ \text{Level } 2: & H_2 := f_2(H_1, H_3)\\ \text{Level } 3: & H_3 := f_3(H_2)\end{array}\\ \textbf{Backward phase } (\text{level } 3 \rightarrow \text{level } 1)\\ \begin{array}{ll}\text{Level } 2: & H_2 := f_2(H_1, H_3)\\ \text{Level } 1: & H_1 := f_1(X, Y, H_2).\end{array}\end{array} \tag{62}$$

One can further repeat the Forward and Backward phases a number of times, without the initialization step.

As predictive model for this Deep RKM case, we have

$$\mathcal{M} \begin{array}{ll}\nearrow & (P)_{\text{DeepRKM}}: \quad \hat{y} = W_1^T \varphi_1(x) + b\\ & \\ \searrow & (D)_{\text{DeepRKM}}: \quad \hat{y} = \dfrac{1}{\eta_1} \displaystyle\sum_j h_j^{(1)} K_1(x_j, x) + b\end{array} \tag{63}$$

where $h_j^{(1)}$ results from the learning process with forward-backward phases.

5 Illustrative examples

5.1 Synthetic data set: regression example

In this example we compare a basic LS-SVM regression with Deep RKM consisting of 3 levels with LSSVM + KPCA + KPCA, where a Gaussian RBF kernel $K(x_i, x_j) = \exp(-\|x_i - x_j\|_2^2/\sigma^2)$ is used in the LSSVM level and linear kernels in the KPCA levels. Training, validation and test data sets are generated from the following true underlying function

$$f(x) = \sin(0.3x) + \cos(0.5x) + \sin(2x) \tag{64}$$

where zero mean Gaussian noise with standard deviation 0.1, 0.5, 1 and 2 is added to the function values for the different data sets. In this example we have a single input and single output $d = p = 1$. Training data (with noise) are generated in the interval $[-10, 10]$ with steps 0.1, validation data (with noise) in $[-9.77, 9.87]$ with steps 0.11 and test data (noiseless) in $[-9.99, 9.99]$ with steps 0.07. In the experiments 100 realizations

noise	basic	deep (1+1)	deep (7+2)
0.1	$0.0019 \pm 4.3\,10^{-4}$	$0.0018 \pm 4.2\,10^{-4}$	$0.0019 \pm 4.4\,10^{-4}$
0.5	0.0403 ± 0.0098	0.0374 ± 0.0403	0.0397 ± 0.0089
1	0.1037 ± 0.0289	0.0934 ± 0.0269	0.0994 ± 0.0301
2	0.3368 ± 0.0992	0.2902 ± 0.0875	0.3080 ± 0.0954

Table 1: Comparison between basic LS-SVM regression and deep RKM on the synthetic data set, for different noise levels.

for the noise are made, for which the mean and standard deviation of the results are shown in Table 1. The tuning parameters are selected based on the validation set, which are σ, γ for the RBF kernel in the basic LS-SVM model and σ, λ_1, η_2, η_3 ($\eta_1 = 1$ has been chosen) for the complete Deep RKM. The number of forward-backward passes in the Deep RKM is chosen equal to 10. For Deep RKM we take the following two choices for the number of components $s^{(2)}, s^{(3)}$ in the KPCA levels: 1 and 1, 7 and 2 for level 2 and level 3 respectively.

For Deep RKM the optimal values for $\frac{1}{\eta_2}, \frac{1}{\eta_3}$ are 10^5 which means that the level 2 and 3 kernel PCA levels receive higher weight in the kernel fusion terms. As seen in this table Deep RKM improves over the basic LS-SVM regression in this example. The optimal values for (σ, λ_1) are $(1, 0.001)$ for noise level 0.1, $(1, 0.01)$ for noise level 0.5, $(1, 0.4)$ for noise level 1. $(1, 1)$ for noise level 2. Figure 4 illustrates a typical example of the regression with the true function indicated by '*' (in color), a noisy training data sample by '*', the predicted output of basic LS-SVM on test data by '+' and the predicted output of Deep RKM by 'o'.

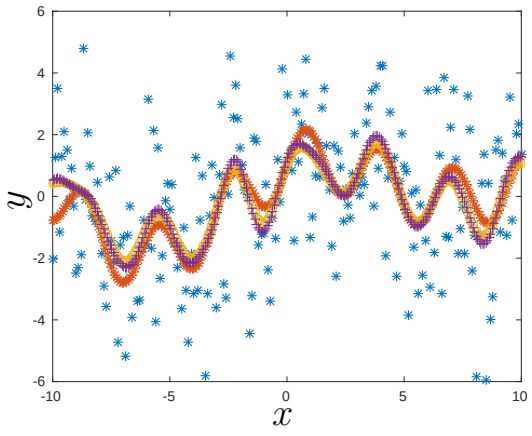

Figure 4: Regression example on synthetic data with input x and output y, showing basic LS-SVM and Deep RKM predictions in comparison to test data.

5.2 Multiclass example

In this example the USPS Handwritten Digits data set is taken, with the Matlab data set as prepared at http://www.cs.nyu.edu/∼roweis/data.html. It contains 8-bit grayscale images of digit "0" through "9" with 1100 examples of each class. These data are used without additional scaling or preprocessing. We compare a basic LS-SVM model (with primal representation $\hat{y} = W^T\varphi(x)+b$ and $W \in \mathbb{R}^{n_f \times p}, b \in \mathbb{R}^p$ with $p = 10$, i.e. one output per class, and RBF kernel) with Deep RKM consisting of LSSVM + KPCA + KPCA with RBF kernel in level 1 and linear kernels in level 2 and 3 (with number of selected components $s^{(2)}$, $s^{(3)}$ in level 2 and 3). In level 1 of Deep RKM the same type of model is taken as in the basic LS-SVM model. In this way we intend to study the effect of the two additional KPCA layers. The dimensionality of the input data is $d = 256$.

Two training set sizes were taken ($N = 2000$ and $N = 4000$ data points, i.e. 200 and 400 examples per class), 2000 data points (200 per class) for validation and 5000 data (500 per class) for testing. The tuning parameters are selected based on the validation set, which are σ, γ for the RBF kernel in the basic LS-SVM model and σ, λ_1, η_2, η_3 ($\eta_1 = 1$ has been chosen) for Deep RKM. The number of forward-backward passes in the Deep RKM is chosen equal to 2. The results are shown for the case of 2000 training data in Figure 5, showing the results on training, validation and test data with the predicted class labels and the predicted output values for the different classes.

For the case $N = 2000$ the selected values were $\sigma^2 = 45$, $s^{(2)} = 10$, $s^{(3)} = 1$, $\lambda_1 = 10^{-6}$, $\frac{1}{\eta_2} = \frac{1}{\eta_3} = 10^6$. The misclassification error on the test data set is 3.18% for the Deep RKM, while 3.26% for the basic LS-SVM (with $\sigma^2 = 45$ and $\gamma = 1/\lambda_1$). For the case $N = 4000$ the selected values were $\sigma^2 = 45$, $s^{(2)} = 1$, $s^{(3)} = 1$, $\lambda_1 = 10^{-6}$, $\frac{1}{\eta_2} = \frac{1}{\eta_3} = 10^6$. The misclassification error on the test data set is 2.12% for the Deep RKM, while 2.14% for the basic LS-SVM (with $\sigma^2 = 45$ and $\gamma = 1/\lambda_1$). This illustrates that for Deep RKM the level 2 and 3 are given high relative importance through the selection of large $\frac{1}{\eta_2}$, $\frac{1}{\eta_3}$ values.

The matrices W_1, W_2, W_3 for the different levels are $W_1 = \frac{1}{\eta_1}\sum_i \varphi_1(x_i){h_i^{(1)}}^T$, $W_2 = \frac{1}{\eta_2}\sum_i h_i^{(1)}{h_i^{(2)}}^T = \frac{1}{\eta_2}H_1H_2^T$, $W_3 = \frac{1}{\eta_3}\sum_i h_i^{(2)}{h_i^{(3)}}^T = \frac{1}{\eta_3}H_2H_3^T$, where $W_2 \in \mathbb{R}^{p\times s^{(2)}}$ and $W_3 \in \mathbb{R}^{s^{(2)}\times s^{(3)}}$. In Figure 6 the interconnection weight strengths for the matrices W_2 and W_3 (absolute values) are visualized for different cases of $s^{(2)}$ and $s^{(3)}$. For the first level an approximate linear model of the form $\hat{\overline{y}} = \overline{W_1}^T x + b$ was taken where as target data the predicted values of the Deep RKM were used. In this way one obtains an approximate linear model with $\overline{W}_1 \in \mathbb{R}^{d\times p}$, estimated by ordinary least squares, solely for the purpose of visualization. This is done because the RBF model part has an infinite dimensional feature map φ_1 in the primal.

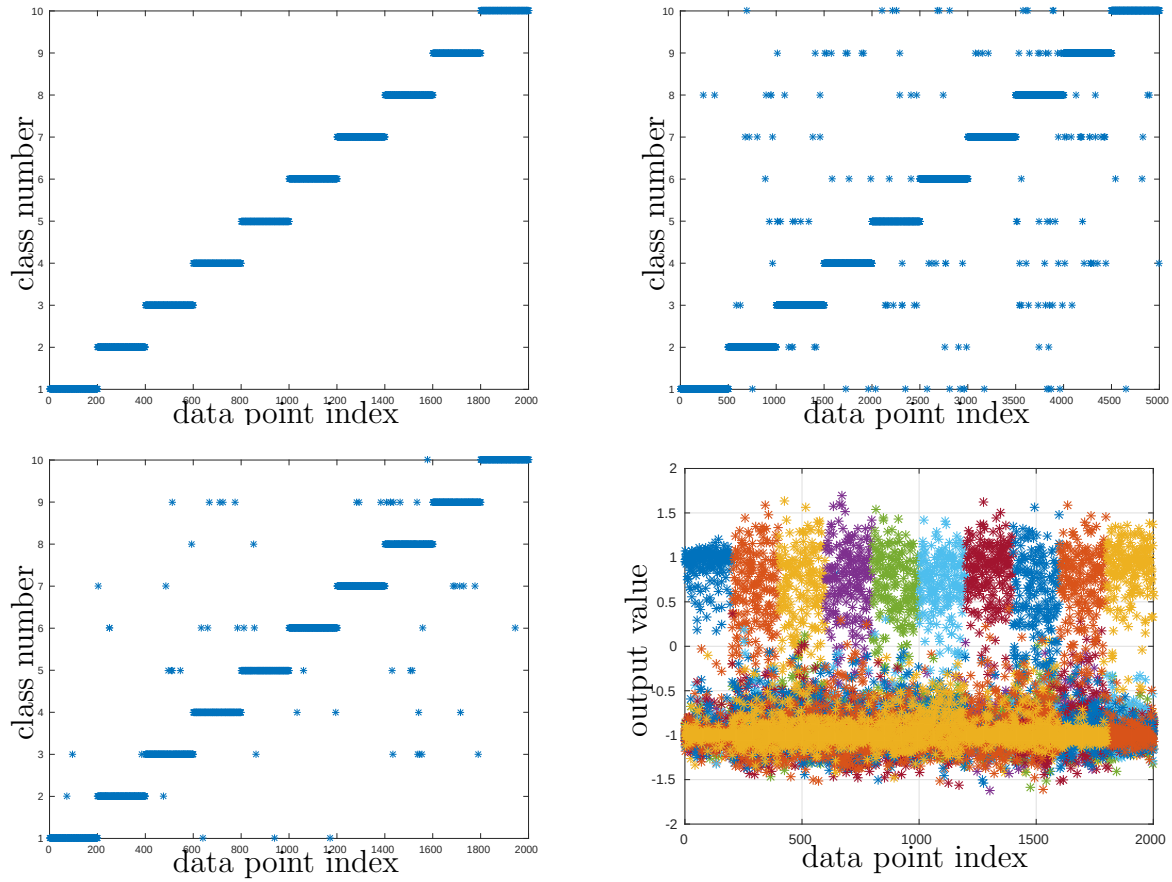

Figure 5: Deep RKM on USPS Handwritten Digits data set: (left-top) training data results (2000 data), (left-bottom) validation data results (2000 data), (right-top) test data results (5000 data), (right-bottom) output values for the 10 different classes on the validation set.

6 Conclusions

In this paper a theory of deep restricted kernel machines has been proposed. It is obtained by introducing a notion of conjugate feature duality where the conjugate features correspond to hidden features. Existing kernel machines such as least squares support vector machines for classification and regression, kernel PCA, matrix SVD and Parzen-type models are considered as building blocks within a Deep RKM, and are characterized through the conjugate feature duality. By means of the inner pairing one achieves a link with the energy expression of restricted Boltzmann machines, though with continuous variables in a nonprobabilistic setting. It also provides an interpretation of visible and hidden units. Therefore this paper connects on the one hand to deep learning methods, while on the other hand also to least squares support vector machines and kernel methods. In this way the insights and foundations achieved in these different research areas could possibly mutually

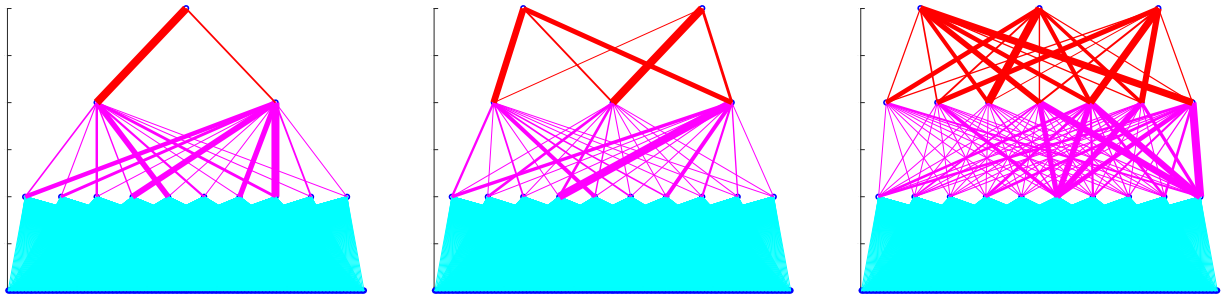

Figure 6: Deep RKM on USPS data: visualization of interconnection matrices $\overline{W}_1, W_2, W_3$. Level 1: LSSVM (bottom layer), Level 2: kernel PCA (middle layer), Level 3: kernel PCA (top layer). The three figures show different numbers of selected components $s^{(2)}, s^{(3)}$ in level 2 and 3: (left) 2 and 1 components, (middle) 3 and 2 components, (right) 7 and 3 components.

reinforce each other in the future. Many future work is possible in different directions, including e.g. the development of methods for big data, the extension to other loss functions and regularization schemes, treating multimodal data, different coupling schemes, models for clustering and semisupervised learning, and several others.

Acknowledgments. The research leading to these results has received funding from the European Research Council (FP7/20072013) / ERC AdG A-DATADRIVE-B (290923) under the European Union's Seventh Framework Programme. This paper reflects only the authors' views, the Union is not liable for any use that may be made of the contained information; Research Council KUL: GOA/10/09 MaNet, CoE PFV/10/002 (OPTEC), BIL12/11T; PhD/Postdoc grants; Flemish Government: FWO: PhD/Postdoc grants, projects: G.0377.12 (Structured systems), G.088114N (Tensor based data similarity); IWT: PhD/Postdoc grants, projects: SBO POM (100031); iMinds Medical Information Technologies SBO 2014; Belgian Federal Science Policy Office: IUAP P7/19 (DYSCO, Dynamical systems, control and optimization, 20122017).

References

[1] D.H. Ackley, G.E. Hinton, T.J. Sejnowski, A learning algorithm for Boltzmann machines, Cognitive Science, 9, 147-169, 1985

[2] C. Alzate, J.A.K. Suykens, Multiway Spectral Clustering with Out-of-Sample Extensions through Weighted Kernel PCA, IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(2) (2010) 335–347.

[3] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35, 1798-1828 (2013).

[4] Y. Bengio, Learning Deep Architectures for AI, Now, 2009.

[5] B.E. Boser, I.M. Guyon, V.N. Vapnik, A training algorithm for optimal margin classifiers, Proceedings COLT '92 Proceedings of the fifth annual workshop on Computational learning theory pp 144-152, 1992.

[6] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004

[7] Y. Cho, L.K. Saul, Kernel Methods for Deep Learning, Advances in Neural Information Processing Systems 22 (NIPS 2009).

[8] C. Cortes, V. Vapnik, Support vector networks, Machine Learning, 20 (1995) 273–297.

[9] A.C. Damianou, N.D. Lawrence, Deep Gaussian Processes, Proceedings of the 16th International Conference on Artificial Intelligence and Statistics (AISTATS) 2013

[10] Ph. De Wilde, Class of Hamiltonian neural networks, Phys. Rev. E 47, 1392-1396, 1993.

[11] A. Fischer, C. Igel, Training Restricted Boltzmann Machines: An Introduction. Pattern Recognition 47, pp. 25-39, 2014

[12] H. Goldstein, C. Poole, J. Safko, Classical Mechanics, Addison-Wesley, 2002.

[13] G.H. Golub, C.F. Van Loan, Matrix Computations, Johns Hopkins University Press, Baltimore, MD (1989)

[14] J. Hertz, A. Krogh, R.G. Palmer, Introduction to the theory of neural computation, Lecture notes Vol I, Santa Fe institute, Addison-Wesley, 1991.

[15] G.E. Hinton, What kind of graphical model is the brain? In Proc. 19th International Joint Conference on Artificial intelligence 1765-1775 (2005).

[16] G.E. Hinton, S. Osindero, Y.-W. Teh, A fast learning algorithm for deep belief nets. Neural Computation 18, 1527-1554 (2006).

[17] J.J. Hopfield, Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences USA, 79, 2554-2558, 1982.

[18] H. Larochelle, Y. Bengio, Classification using discriminative restricted Boltzmann machines. Proceedings of the 25th international conference on Machine learning - ICML 2008. p. 536, 2008

[19] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, F.-J. Huang, A Tutorial on Energy-Based Learning, in Bakir, G. and Hofman, T. and Schölkopf, B. and Smola, A. and Taskar, B. (Eds), Predicting Structured Data, MIT Press, 2006.

[20] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature, 521, 436-444, 2015

[21] H. Lee, R. Grosse. R. Ranganath, A.Y. Ng., Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations, ICML 2009 Proceedings of the 26th Annual International Conference on Machine Learning pp. 609-616, 2009.

[22] J. Mairal, P. Koniusz, Z. Harchaoui and C. Schmid. Convolutional Kernel Networks. Adv. Neural Information Processing Systems (NIPS). 2014

[23] R. Mall, R. Langone, J.A.K. Suykens, Multilevel Hierarchical Kernel Spectral Clustering for Real-Life Large Scale Complex Networks, PLOS One, 9(6) (2014) e99966.

[24] K.B. Petersen, M.S. Pedersen, The Matrix Cookbook, Technical University of Denmark, 2012

[25] T. Poggio, F. Girosi, Networks for approximation and learning, Proceedings of the IEEE, 78(9), 1481-1497, 1990.

[26] C.E. Rasmussen, C. Williams, Gaussian Processes for Machine Learning, MIT Press, 2006.

[27] R.T. Rockafellar, Conjugate Duality and Optimization, CBMS-NSF Regional Conference Series in Applied Mathematics, 1987

[28] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors. Nature, 323, 533-536 (1986).

[29] R. Salakhutdinov, G.E. Hinton, Using deep belief nets to learn covariance kernels for Gaussian processes, Advances in Neural Information Processing Systems 20 (NIPS 2007)

[30] R. Salakhutdinov, Learning Deep Generative Models, Annu. Rev. Stat. Appl. 2015. 2:361-85.

[31] R. Salakhutdinov, G.E. Hinton, Deep Boltzmann Machines, International conference on artificial intelligence and statistics, 448-455, 2009

[32] C. Saunders, A. Gammerman, V. Vovk, Ridge regression learning algorithm in dual variables, Proc. of the 15th Int. Conf. on Machine Learning (ICML-98), Madison-Wisconsin, 515–521, 1998.

[33] J. Schmidhuber, Deep learning in neural networks: An overview, Neural Networks, Vol 61, pp.85-117, 2015.

[34] B. Schölkopf, A. Smola, K.-R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Computation, 10 (1998) 1299–1319.

[35] B. Schölkopf, A. Smola, Learning with Kernels, MIT Press, Cambridge, MA, 2002.

[36] S. Smale, L. Rosasco, J. Bouvrie, A. Caponnetto, T. Poggio, Mathematics of the neural response, Foundations of Computational Mathematics, 10 (1), 67-91, 2010.

[37] P. Smolensky. Information processing in dynamical systems: foundations of harmony theory. Chapter 6 in D.E. Rumelhart and J.L. McClelland, editors, Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations. McGraw-Hill, New York, 1986

[38] N. Srivastava, R. Salakhutdinov, Multimodal Learning with Deep Boltzmann Machines, Journal of Machine Learning Research 15 (2014) 2949-2980.

[39] G.W. Stewart, On the early history of the singular value decomposition, SIAM Review, 35 (4) (1993), pp. 551566

[40] J.A.K. Suykens, J. Vandewalle, B. De Moor, Artificial Neural Networks for Modeling and Control of Non-Linear Systems, Springer, 1995

[41] J.A.K. Suykens, J. Vandewalle, Training multilayer perceptron classifiers based on a modified support vector method, IEEE Transactions on Neural Networks, vol. 10, no. 4, Jul. 1999, pp. 907-911.

[42] J.A.K. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural Processing Letters, vol. 9, no. 3, Jun. 1999, pp. 293-300

[43] J.A.K. Suykens, J. Vandewalle, Multiclass Least Squares Support Vector Machines, in Proc. of the International Joint Conference on Neural Networks (IJCNN'99), Washington DC, USA, Jul. 1999

[44] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, Least Squares Support Vector Machines, World Scientific, Singapore, 2002.

[45] J.A.K. Suykens, T. Van Gestel, J. Vandewalle, B. De Moor, A support vector machine formulation to PCA analysis and its kernel version, IEEE Transactions on Neural Networks, 14(2) (2003) 447–450.

[46] J.A.K. Suykens, C. Alzate, K. Pelckmans, Primal and dual model representations in kernel-based learning, Statistics Surveys, 4 (2010) 148–183.

[47] J.A.K. Suykens, Generating quantum-measurement probabilities from an optimality principle, Physical Review A, 87(5) (2013) 052134.

[48] J.A.K. Suykens, SVD revisited: a new variational principle, compatible feature maps and nonlinear extensions, Applied and Computational Harmonic Analysis, in press, doi:10.1016/j.acha.2015.09.004

[49] V. Vapnik, Statistical Learning Theory, John Wiley & Sons, New York, 1998.

[50] G. Wahba, Spline Models for Observational Data, SIAM, 1990

[51] M. Welling, M. Rosen-Zvi, G.E. Hinton, Exponential family harmoniums with an application to information retrieval, Advances in neural information processing systems, 2004

[52] M.A. Wiering, L.R.B. Schomaker, Multi-Layer Support Vector Machines, Chapter 20 in: Regularization, Optimization, Kernels, and Support Vector Machines, J.A.K. Suykens, M. Signoretto, A. Argyriou (Eds.), Chapman & Hall/CRC, Machine Learning & Pattern Recognition Series, Boca Raton US, pp.457-476, 2014