Least Squares Support Vector Machines and Deep Learning

Johan Suykens

KU Leuven, ESAT-Stadius and Leuven.AI Institute
Kasteelpark Arenberg 10, B-3001 Leuven, Belgium
Email: johan.suykens@esat.kuleuven.be
http://www.esat.kuleuven.be/stadius/

IEEE WCCI - IJCNN 2024 Yokohama, July 2024



Overview

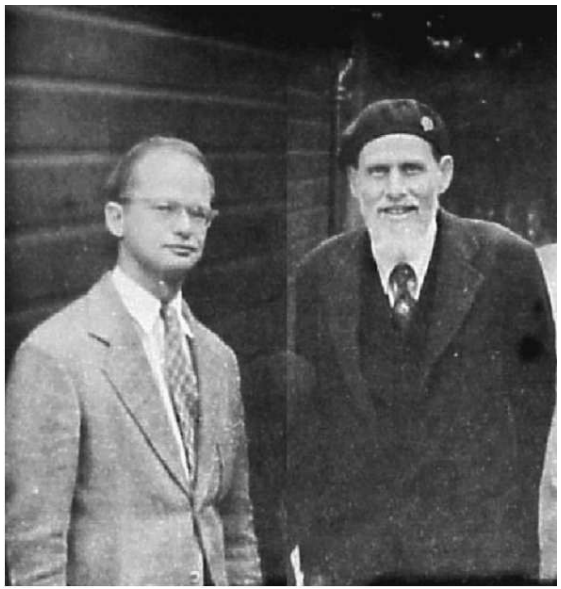
- Introduction and Motivation
- Least Squares Support Vector Machines (LS-SVM) as core models
- LS-SVM and Deep Learning:

  $\rightarrow$ Kernel SVD & Self-Attention (Transformers in AI)

  $\rightarrow$ From RBM to RKM, Gen-RKM & Deep Kernel Machines
- Future challenges
- Conclusions

Introduction

McCulloch & Pitts model of a neuron (1943)



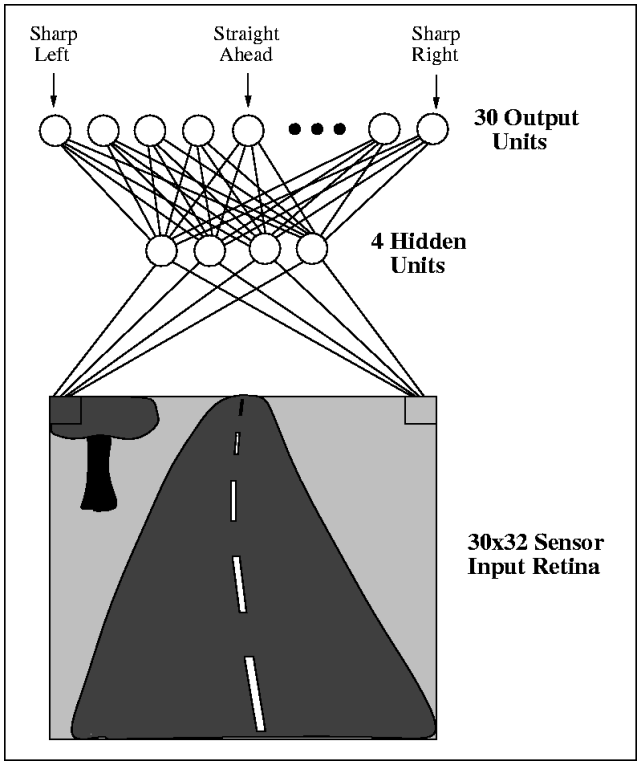
"A logical calculus of the ideas immanent in nervous activity",
Bulletin of Mathematical Biophysics, Vol. 5, pp. 115-133 (1943)

Warren McCulloch (neurophysiologist, cybernetician, engineer)
Walter Pitts (logician, cognitive psychologist)

source: www.historyofinformation.com

Neural networks as universal approximators

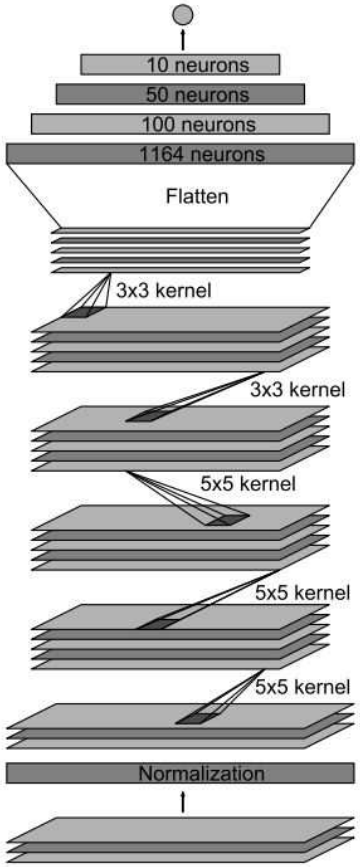
One hidden layer is sufficient for universal approximation:





ALVINN (Autonomous Land Vehicle In a Neural Network)
[Pomerleau, Neural Computation 1991]

Deep feature learning



Output: vehicle control

Fully-connected layer
Fully-connected layer
Fully-connected layer

Convolutional feature map 64@1x18

Convolutional feature map 64@3x20

Convolutional feature map 48@5x22

Convolutional feature map 36@14x47

Convolutional feature map 24@31x98

Normalized input planes 3@66x200

Input planes 3@66x200

(27 million connections)



Waymo / Google Self-Driving Car



Tesla Autopilot



Uber



nuTonomy

from: [selfdrivingcars.mit.edu (Lex Fridman et al.), 2017]

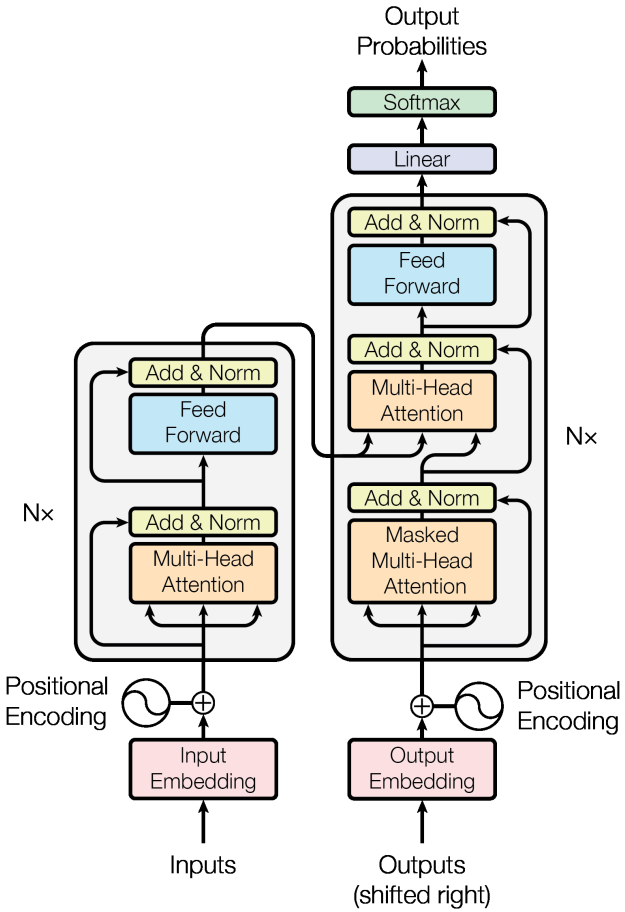
GANs

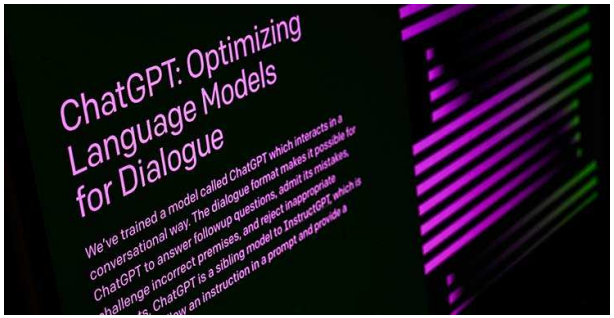


Fig. 13.1 Four years of face generation using generative models

[figure Ye 2022]

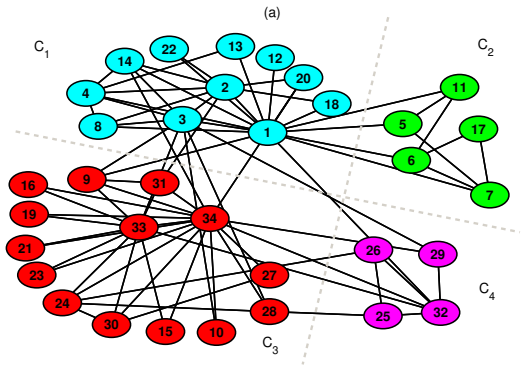
Transformers





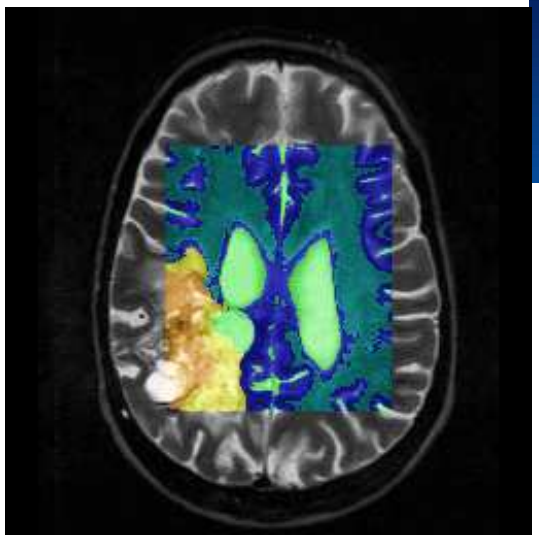
Data-driven world

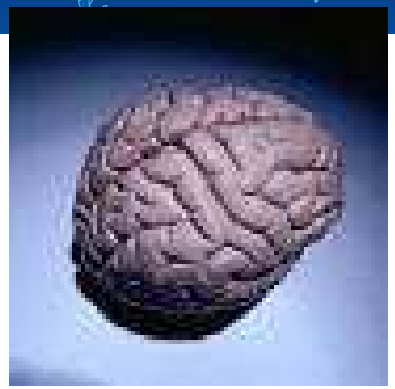








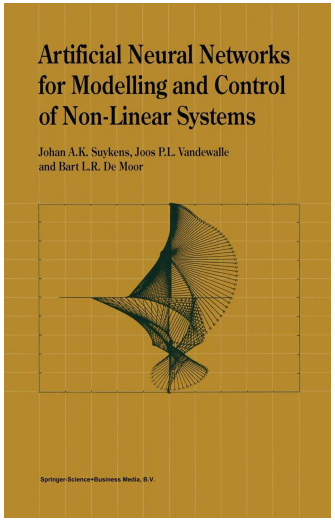


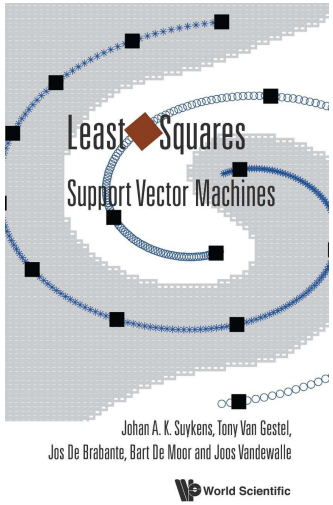


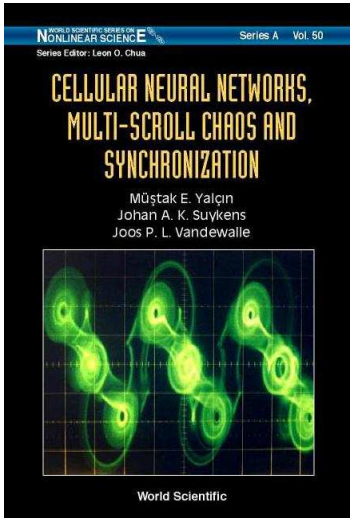


Challenges

- general methodology
- unifying frameworks
- mathematical foundations
- scalability
- robustness
- interpretability





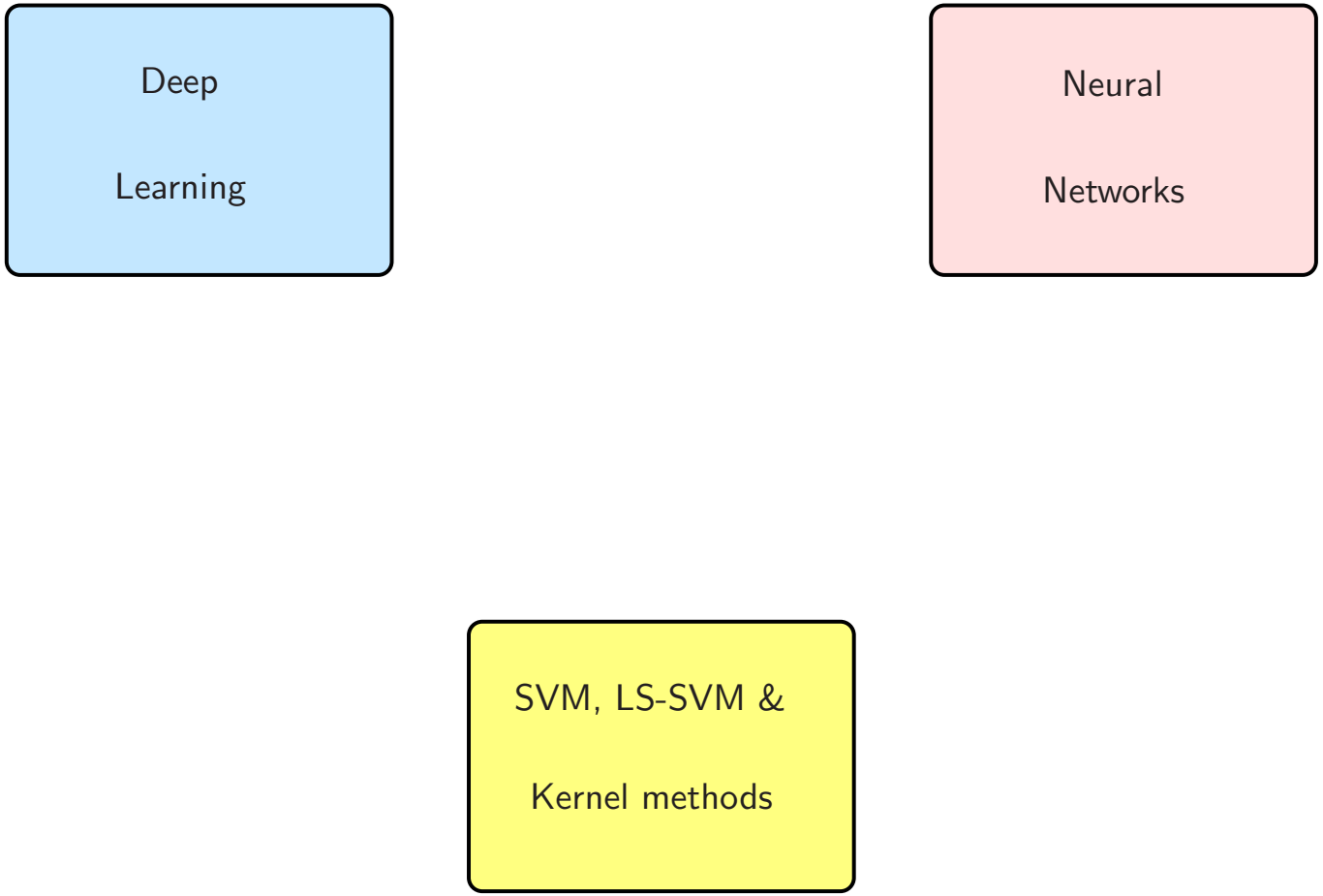


J.A.K. Suykens, J.P.L. Vandewalle, B.L.R. De Moor, *Artificial Neural Networks for Modeling and Control of Non-Linear Systems*, Springer, 1996 *(stability theory for multilayer recurrent networks; NLq theory)*

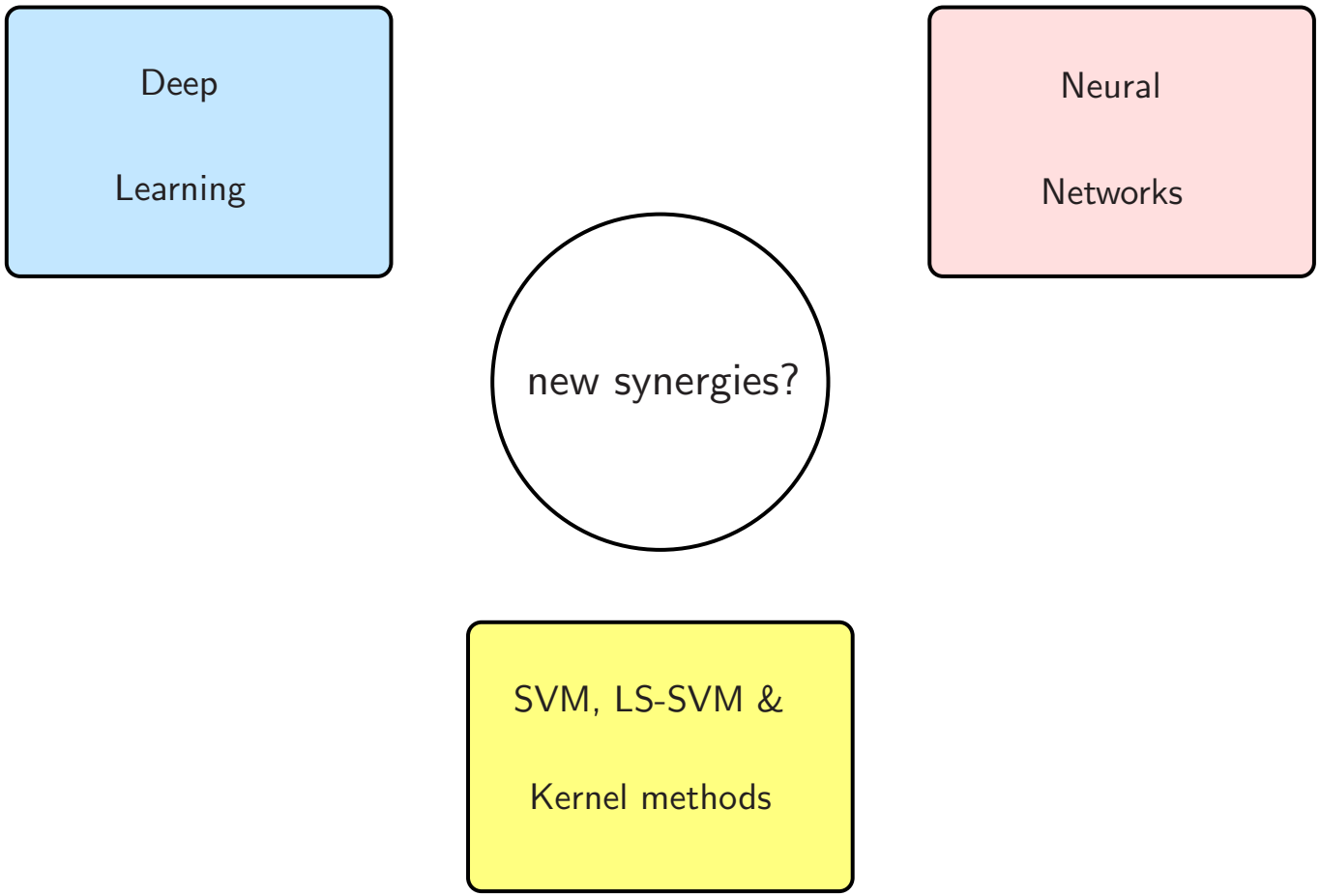
J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, *Least Squares Support Vector Machines*, World Scientific, Singapore, 2002 (THIS TALK)

M.E. Yalcin, J.A.K. Suykens, J.P.L. Vandewalle, *Cellular Neural Networks, Multi-Scroll Chaos and Synchronization*, World Scientific Series on Nonlinear Science, 2005 *(Lur'e systems, nonlinear circuits, recurrent networks; synchronization, stability, complex behaviour)*

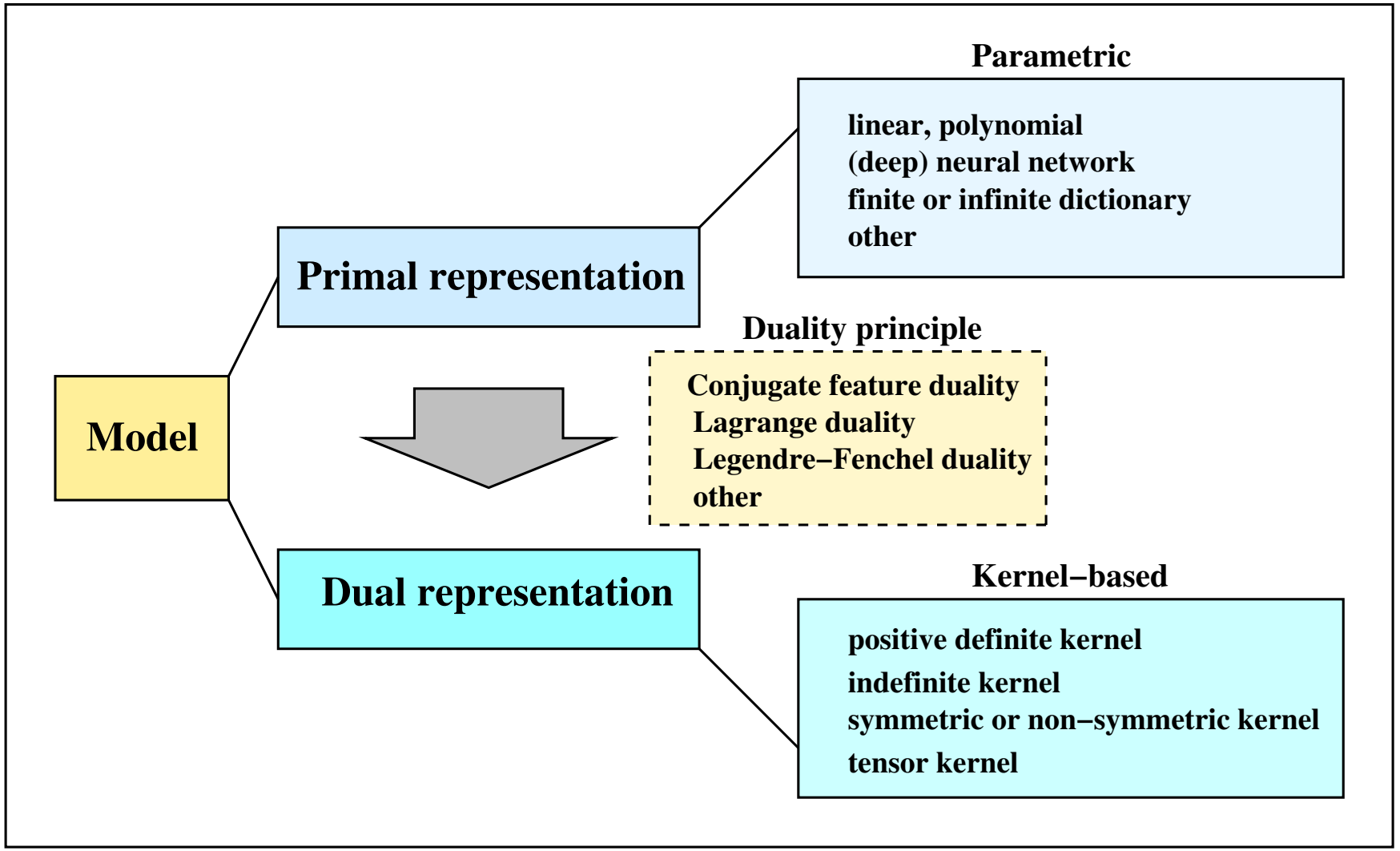
New Synergies



New Synergies



Towards a unifying picture



[Suykens 2017]

LS-SVM as core models

SVM and LS-SVM classifier

- **SVM Primal problem:** [Vapnik, 1995; Cortes & Vapnik, 1995]

$$\min_{w,b,\xi} \ \frac{1}{2}w^T w + c\sum_{i=1}^{N}\xi_i \ \text{ s.t. } \ \begin{cases} y_i[w^T\varphi(x_i)+b] \geq 1-\xi_i \\ \xi_i \geq 0, \quad i=1,...,N \end{cases}$$

  Dual problem: convex QP problem

- **LS-SVM Primal problem:** [Suykens & Vandewalle, 1999]

$$\min_{w,b,e} \ \frac{1}{2}w^T w + \frac{1}{2}\gamma\sum_{i=1}^{N}e_i^2 \ \text{ s.t. } \quad y_i[w^T\varphi(x_i)+b] = 1-e_i$$

  Dual problem: linear system of equations

Feature map and kernel

Primal and dual representation:

$$\text{Model} \begin{cases} \nearrow & (P): \quad \hat{y} = w^T \varphi(x) + b \\ \searrow & (D): \quad \hat{y} = \sum_i \alpha_i K(x_i, x) + b \end{cases}$$

Mercer theorem (one can **either** choose φ **or** positive definite K):

$$K(x, z) = \varphi(x)^T \varphi(z)$$

Feature map φ, Kernel function $K(x, z)$ (e.g. linear, polynomial, RBF, ...)

- SVMs: feature map and positive definite kernel [Cortes & Vapnik, 1995]
- Neural networks: hidden layer as feature map [Suykens & Vandewalle, IEEE-TNN 1999]
- Least squares support vector machines [Suykens et al., 2002]: L_2 loss and regularization

Least Squares Support Vector Machines: "core models"

- Regression

$$\min_{w,b,e} w^T w + \gamma \sum_i e_i^2 \;\; \text{s.t.} \;\; y_i = w^T \varphi(x_i) + b + e_i, \;\; \forall i$$

- Classification

$$\min_{w,b,e} w^T w + \gamma \sum_i e_i^2 \;\; \text{s.t.} \;\; y_i(w^T \varphi(x_i) + b) = 1 - e_i, \;\; \forall i$$

- Kernel pca $(V = I)$, Kernel spectral clustering $(V = D^{-1})$

$$\min_{w,b,e} -w^T w + \gamma \sum_i v_i e_i^2 \;\; \text{s.t.} \;\; e_i = w^T \varphi(x_i) + b, \;\; \forall i$$

- Kernel canonical correlation analysis/partial least squares

$$\min_{w,v,b,d,e,r} w^T w + v^T v + \nu \sum_i (e_i - r_i)^2 \text{ s.t. } \left\{ \begin{array}{rcl} e_i & = & w^T \varphi^{(1)}(x_i) + b \\ r_i & = & v^T \varphi^{(2)}(y_i) + d \end{array} \right.$$

[Suykens & Vandewalle, NPL 1999; Suykens et al., 2002; Alzate & Suykens, 2010]

other LS-SVM developments

- robustly (re)weighted versions, sparse versions, Bayesian versions, fixed-size methods for large scale applications, primal and dual estimation

- recurrent models, time-series prediction, correlated errors, optimal control, PDE/ODE/DAE approximate solutions

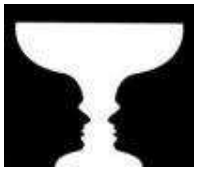
- system identification: Wiener-Hammerstein, state space, partially linear

- missing values, prediction intervals, variable selection, survival analysis, multi-view models, domain adaptation, indefinite kernels, constraints

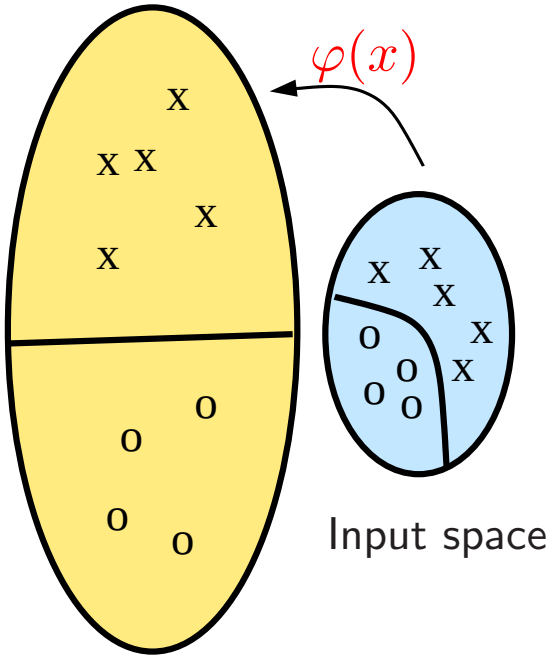
- LS-SVMlab **software** toolbox

  Applications: electric load forecasting, pollution modelling, weather forecasting, community detection, financial time-series, credit scoring, bankruptcy prediction, industrial machines maintenance, structural health monitoring, chemometrics, brain tumour classification, ovarian tumor diagnosis, biomedical data fusion, ...

  [see e.g. LS-SVMlab website $>$ toolbox & publications]



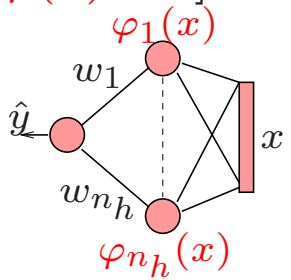
SVMs and neural networks



Primal space

Parametric

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



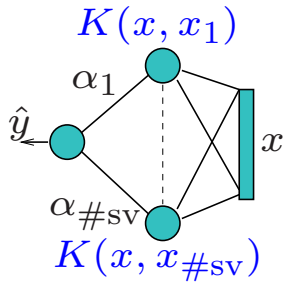
Parametric

$K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$ **("Kernel trick")**

Dual space

Nonparametric

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



Non–parametric

[Suykens et al., 2002]

Kernels

Wide range of positive definite kernel functions possible:

- linear $K(x, z) = x^T z$
- polynomial $K(x, z) = (\eta + x^T z)^d$
- radial basis function $K(x, z) = \exp(-\|x - z\|_2^2/\sigma^2)$
- splines, wavelets
- string kernel, Fisher kernel
- graph kernels, kernels from graphical models
- kernels for dynamical systems
- data fusion kernels
- other

[Schölkopf & Smola, 2002; Shawe-Taylor & Cristianini, 2004; Jebara et al., 2004; other]

Function estimation in RKHS, RKKS, RKBS; Gaussian processes

[Wahba 1990; Steinwart & Christmann 2008; Cucker & Zhou 2007; Rasmussen & Williams 2006]

Self-attention and transformers within the LS-SVM framework

Singular Value Decomposition (SVD)

- **Singular Value Decomposition (SVD)** of $A \in \mathbb{R}^{N \times M}$

$$A = U \Sigma V^T$$

with $U^T U = I_N$, $V^T V = I_M$, $\Sigma = \mathrm{diag}(\sigma_1, ..., \sigma_p) \in \mathbb{R}^{N \times M}$.

- **Early history overview** by [Stewart, 1993]:
  - early contributions by Beltrami (1873), Jordan (1874), Eckart & Young (1936), Lanczos (1958), and several others.
  - related work on integral equations: Schmidt (1907)

SVD - classical variational principle

- extrema of the bilinear form

$$f(u,v) = u^T A v \;\; \text{subject to} \;\; \|u\|^2 = \|v\|^2 = 1.$$

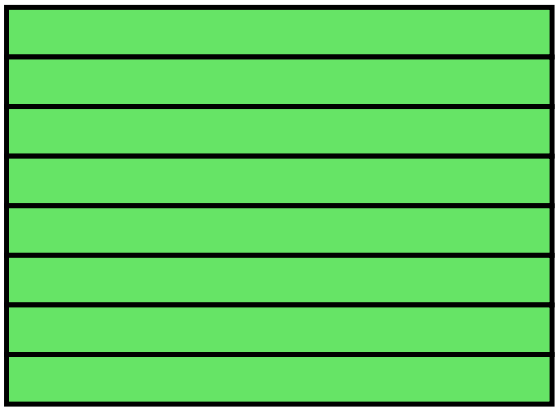
- solutions follow from the eigenvalue decomposition

$$\begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \lambda \begin{bmatrix} u \\ v \end{bmatrix}$$

where $\lambda \in \{\pm\sigma_1, \pm\sigma_2, ..., \pm\sigma_p, 0\}$ with multiplicity $M - N$ for the zero eigenvalue (assuming $M > N$ and non-zero σ_i values).

SVD within the LS-SVM setting (1)

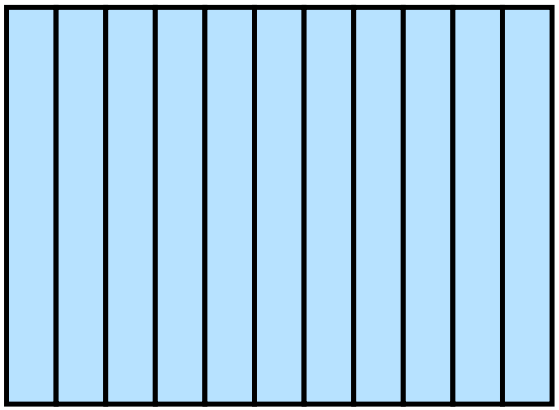
matrix A



$\{x_i\}$

SVD within the LS-SVM setting (1)

matrix A



$\{z_j\}$

SVD within the LS-SVM setting (2)

- **Obtain two sets of data points** (rows and columns):

$$x_i = A^T \epsilon_i, \quad z_j = A\varepsilon_j$$

for $i = 1, ..., N$, $j = 1, ..., M$ where $\epsilon_i, \varepsilon_j$ are standard basis vectors of dimension N and M.

- **Compatible feature maps:** $\varphi : \mathbb{R}^M \to \mathbb{R}^N$, $\psi : \mathbb{R}^N \to \mathbb{R}^N$ where

$$\begin{array}{rcl} \varphi(x_i) & = & C^T x_i = C^T A^T \epsilon_i \\ \psi(z_j) & = & z_j = A\varepsilon_j \end{array}$$

with $C \in \mathbb{R}^{M \times N}$ a **compatibility matrix**.

[Suykens, SVD revisited: A new variational principle, compatible feature maps and nonlinear extensions, ACHA, 2016]

SVD within the LS-SVM setting (3)

- **Primal problem** (new variational principle):

$$\min_{w,v,e,r} \; -w^T v + \tfrac{1}{2}\gamma \sum_{i=1}^{N} e_i^2 + \tfrac{1}{2}\gamma \sum_{j=1}^{M} r_j^2 \text{ subject to } \quad e_i = w^T\varphi(x_i),\ i = 1,...,N \quad r_j = v^T\psi(z_j),\ j = 1,...,M$$

  with $e_i, r_j \in \mathbb{R}$ and $w, v \in \mathbb{R}^N$.

- Lagrangian $\mathcal{L} = J - \sum_i \alpha_i \left(e_i - w^T\varphi(x_i)\right) - \sum_j \beta_j \left(r_j - v^T\psi(z_j)\right)$

- Optimality conditions $\frac{\partial \mathcal{L}}{\partial w} = 0, \frac{\partial \mathcal{L}}{\partial v} = 0, \frac{\partial \mathcal{L}}{\partial e_i} = 0, \frac{\partial \mathcal{L}}{\partial r_j} = 0, \frac{\partial \mathcal{L}}{\partial \alpha_i} = 0, \frac{\partial \mathcal{L}}{\partial \beta_j} = 0$

- Eliminate w, v, e, r; collect solutions corresponding to non-zero $\lambda = \frac{1}{\gamma}$:

$$\begin{bmatrix} 0 & K \\ K^T & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \lambda \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \text{ with kernel matrix } K_{ij} = \varphi(x_i)^T\psi(z_j).$$

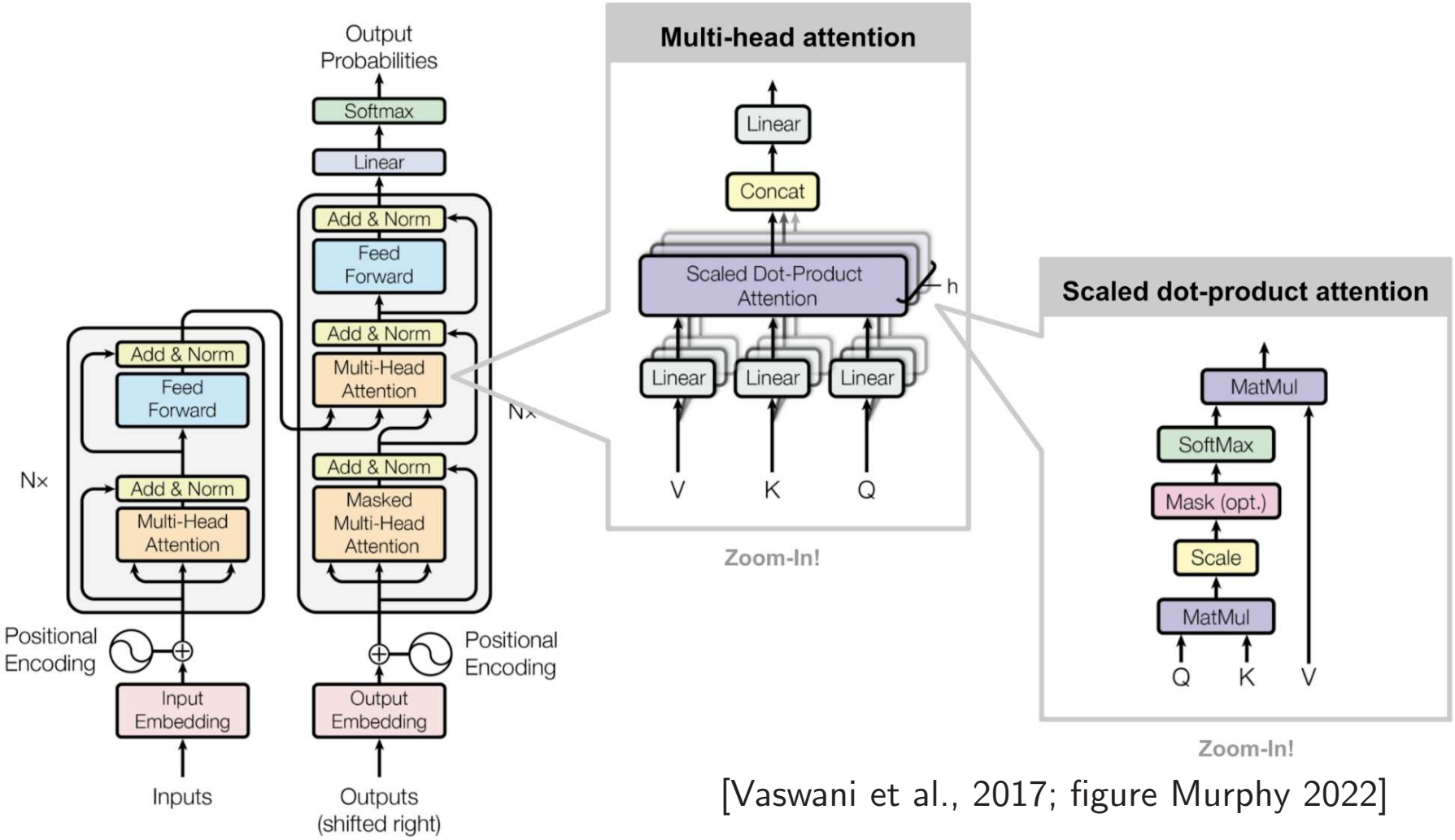
Primal and dual model representations

- Primal and dual model representations

$$\mathcal{M} \begin{cases} \nearrow & (P): \quad e_i = w^T\varphi(x_i) \\ & \qquad\quad r_j = v^T\psi(z_j) \\ \searrow & (D): \quad e_i = \sum_j \beta_j \psi(z_j)^T\varphi(x_i) \\ & \qquad\quad r_j = \sum_i \alpha_i \varphi(x_i)^T\psi(z_j) \end{cases}$$

- One can consider an out-of-sample extension, corresponding to adding a row or a column to matrix A.

Transformer



[Vaswani et al., 2017; figure Murphy 2022]

Self-attention with Asymmetric Kernel

Let $\{x_i \in \mathbb{R}^d\}_{i=1}^N$ be the input data sequence. In **self-attention** [Vaswani et al., 2017], the **queries**, **keys** and **values** are

$$q(x_i) = W_q x_i,\ k(x_i) = W_k x_i,\ v(x_i) = W_v x_i,$$

where $W_q \in \mathbb{R}^{d_q \times d}, W_k \in \mathbb{R}^{d_k \times d}, W_v \in \mathbb{R}^{d_v \times d}$ (commonly $d_q = d_k$). The attention scores are $a(x_i, x_j) = \langle q(x_i), k(x_j) \rangle / \sqrt{d_k} = \langle W_q x_i, W_k x_j \rangle / \sqrt{d_k}$. In the canonical self-attention, the softmax activation is then applied, yielding the attention weights:

$$\kappa(x_i, x_j) = \mathrm{softmax}(\langle W_q x_i, W_k x_j \rangle) / \sqrt{d_k},\ i, j = 1, ..., N$$

which is an **asymmetric kernel function**, i.e. $\kappa(x_i, x_j) \neq \kappa(x_j, x_i)$. The attention output $o_i \in \mathbb{R}^{d_v}$ in each head is

$$o_i = \sum_{j=1}^{N} v(x_j) \kappa(x_i, x_j) = \sum_{j=1}^{N} v(x_j) K_{ij}, \quad i = 1, ..., N.$$

Primal-dual representations for self-attention

- **Self-attention** primal problem [Chen et al., NeurIPS 2023]:

$$\begin{aligned} \max_{W_e, W_r, e_i, r_j} \quad & J = \frac{1}{2}\sum_{i=1}^{N} e_i^T \Lambda e_i + \frac{1}{2}\sum_{j=1}^{N} r_j^T \Lambda r_j - \mathrm{Tr}(W_e^T W_r) \\ s.t. \quad & e_i = (f(X)^T W_e)^T \phi_q(x_i),\ i = 1, ..., N, \\ & r_j = (f(X)^T W_r)^T \phi_k(x_j),\ j = 1, ..., N \end{aligned}$$

  where ϕ_q, ϕ_k are **feature maps** related to queries and keys, respectively. Λ denotes a diagonal matrix with positive diagonal elements. The dual problem gives a shifted eigenvalue problem, related to **Kernel SVD**.

- In primal-attention [Chen et al., 2023] the attention outputs are

$$o_i = [e_i; r_i] = [W_e^T f(X) g_q(q(x_i)); W_r^T f(X) g_k(k(x_i))]$$

  where $K_{ij} = \langle g_q(q(x_i)), g_k(k(x_j)\rangle = \langle \phi_q(x_i), \phi_k(x_j)\rangle$.

Training of Transformers

Training of **Transformers** based on **primal-attention** [Chen et al., 2023]:

$$\min L + \eta \sum_l J_l^2$$

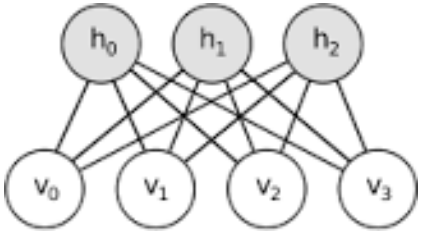
with $\eta > 0$, L a task-oriented loss (e.g. cross-entropy for classification tasks) and J_l are the primal objectives of the several attention blocks with the primal-attention scheme. Here J_l is of the form

$$J = \frac{1}{2}\sum_{i=1}^{N} e_i^T \Lambda e_i + \frac{1}{2}\sum_{j=1}^{N} r_j^T \Lambda r_j - \mathrm{Tr}(W_e^T W_r).$$

It exploits the property that for all components evaluated in the primal objective J one has $J = 0$ (hence the term J_l^2 in the objective for training the Transformer). In this way **low-rank representations** can be obtained, together with **efficient training in the primal**.

From LS-SVM framework to RKM representations, and beyond

Restricted Boltzmann Machines (RBM)



- Markov random field, bipartite graph, stochastic binary units
  Layer of visible units v and layer of hidden units h
  No hidden-to-hidden connections
- Energy:

$$E(v, h; \theta) = -v^T W h - c^T v - a^T h \quad \text{with} \quad \theta = \{W, c, a\}$$

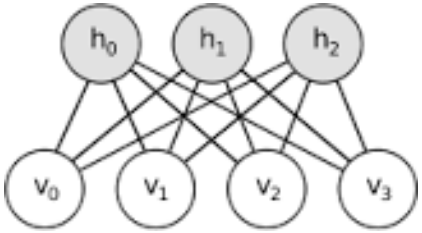
Joint distribution:

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

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

[Hinton, Osindero, Teh, Neural Computation 2006]

Restricted Boltzmann Machines (RBM)



- Markov random field, bipartite graph, stochastic binary units
  Layer of visible units v and layer of hidden units h
  No hidden-to-hidden connections
- Energy:

$$E(v, h; \theta) = -v^T W h - c^T v - a^T h \quad \text{with} \quad \theta = \{W, c, a\}$$

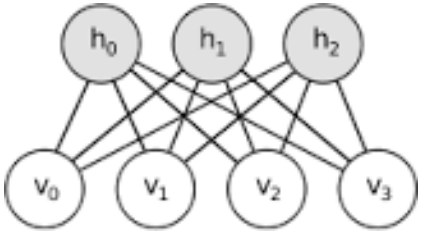
Joint distribution:

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

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

[Hinton, Osindero, Teh, Neural Computation 2006]

Restricted Boltzmann Machines (RBM)



- Markov random field, bipartite graph, stochastic binary units
  Layer of visible units v and layer of hidden units h
  No hidden-to-hidden connections
- Energy:

$$E(v, h; \theta) = -(v^T W)h - c^T v - a^T h \quad \text{with} \quad \theta = \{W, c, a\}$$

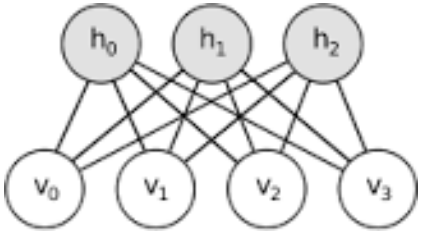
Joint distribution:

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

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

[Hinton, Osindero, Teh, Neural Computation 2006]

Restricted Boltzmann Machines (RBM)



- Markov random field, bipartite graph, stochastic binary units
  Layer of visible units v and layer of hidden units h
  No hidden-to-hidden connections
- Energy:

$$E(v, h; \theta) = -v^T(Wh) - c^T v - a^T h \quad \text{with} \quad \theta = \{W, c, a\}$$

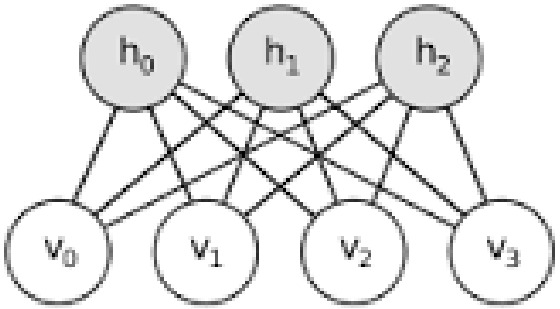
Joint distribution:

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

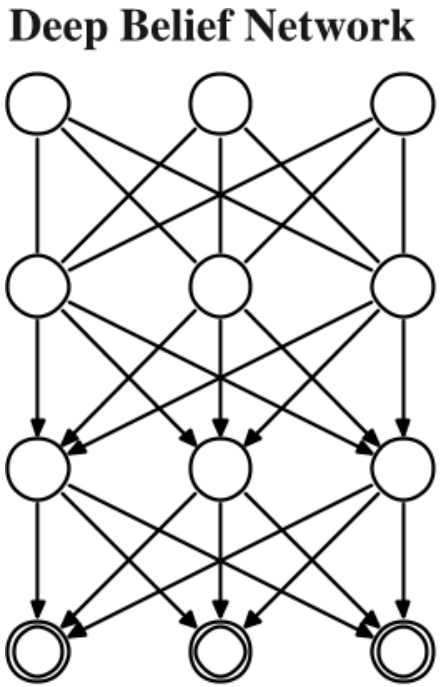
with partition function $Z(\theta) = \sum_v \sum_h \exp(-E(v, h; \theta))$

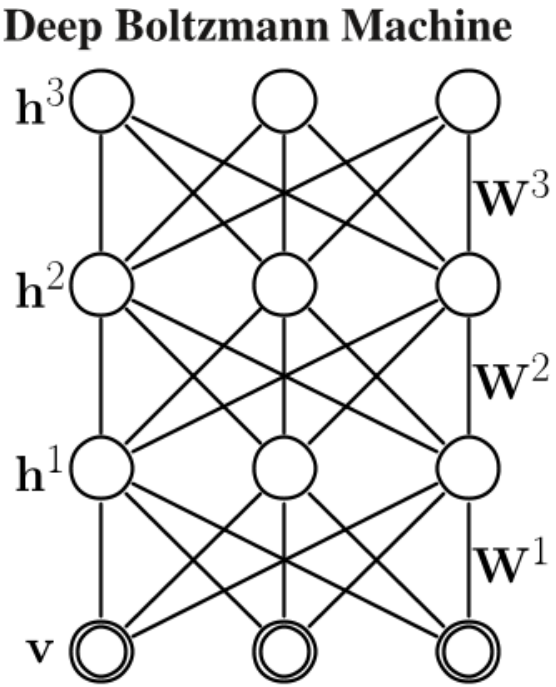
[Hinton, Osindero, Teh, Neural Computation 2006]

RBM and deep learning



$p(v, h)$



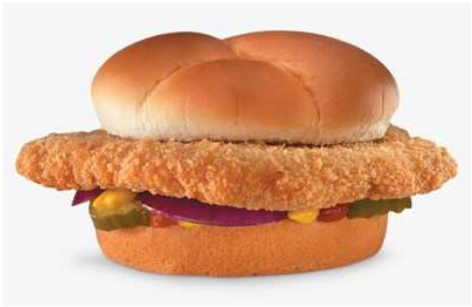


$p(v, h^1, h^2, h^3, ...)$

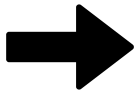
[Hinton et al., 2006; Salakhutdinov, 2015]

in other words ...

"sandwich"



$$E = -v^T W h$$

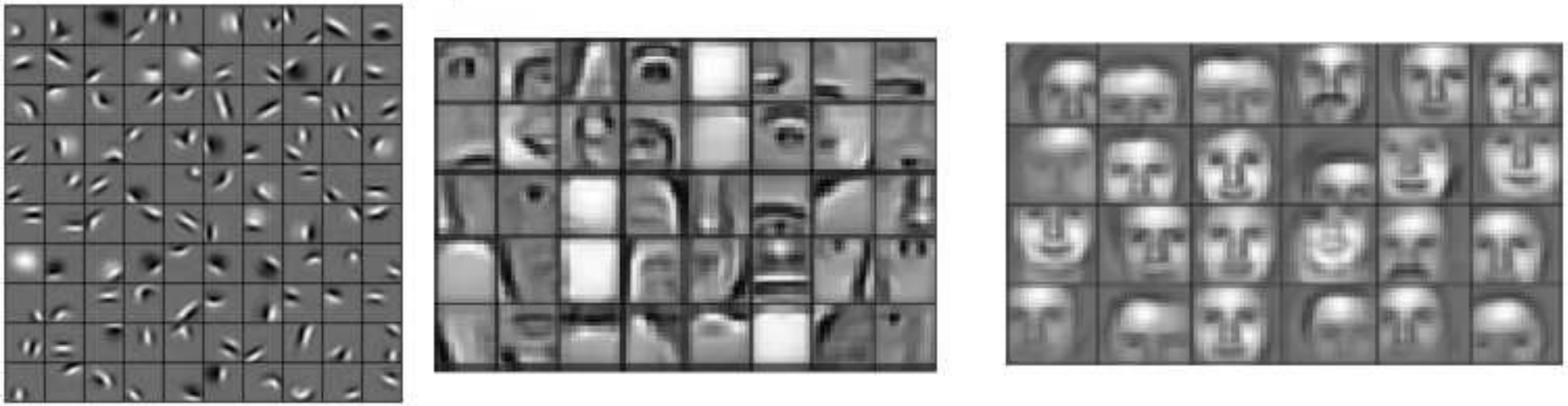


"deep sandwich"



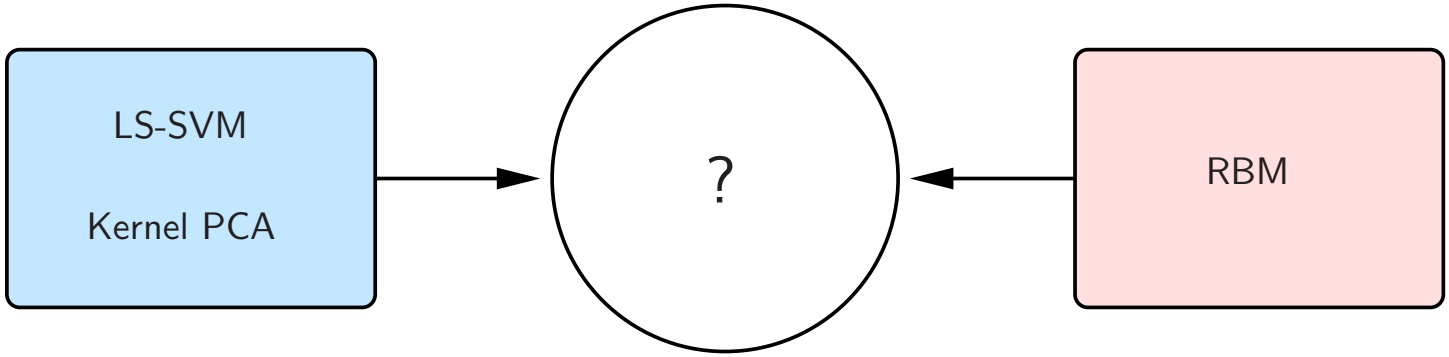
$$E = -v^T W^1 h^1 - {h^1}^T W^2 h^2 - {h^2}^T W^3 h^3$$

Convolutional Deep Belief Networks

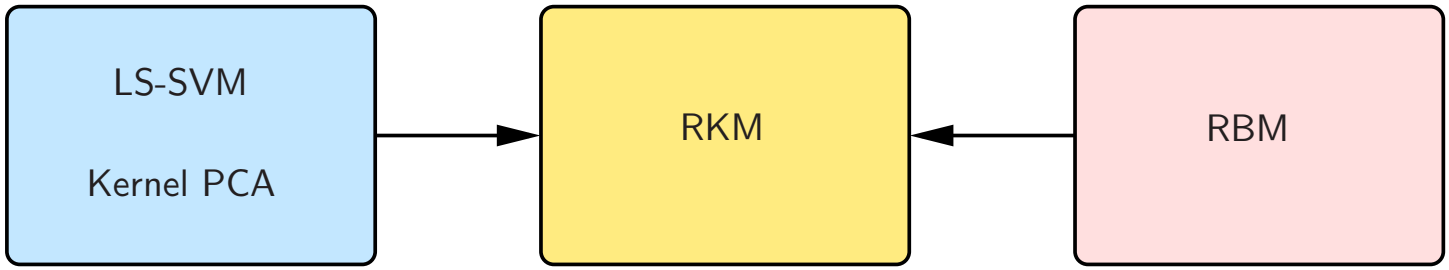


Unsupervised Learning of Hierarchical Representations with Convolutional Deep Belief Networks [Lee et al. 2011]

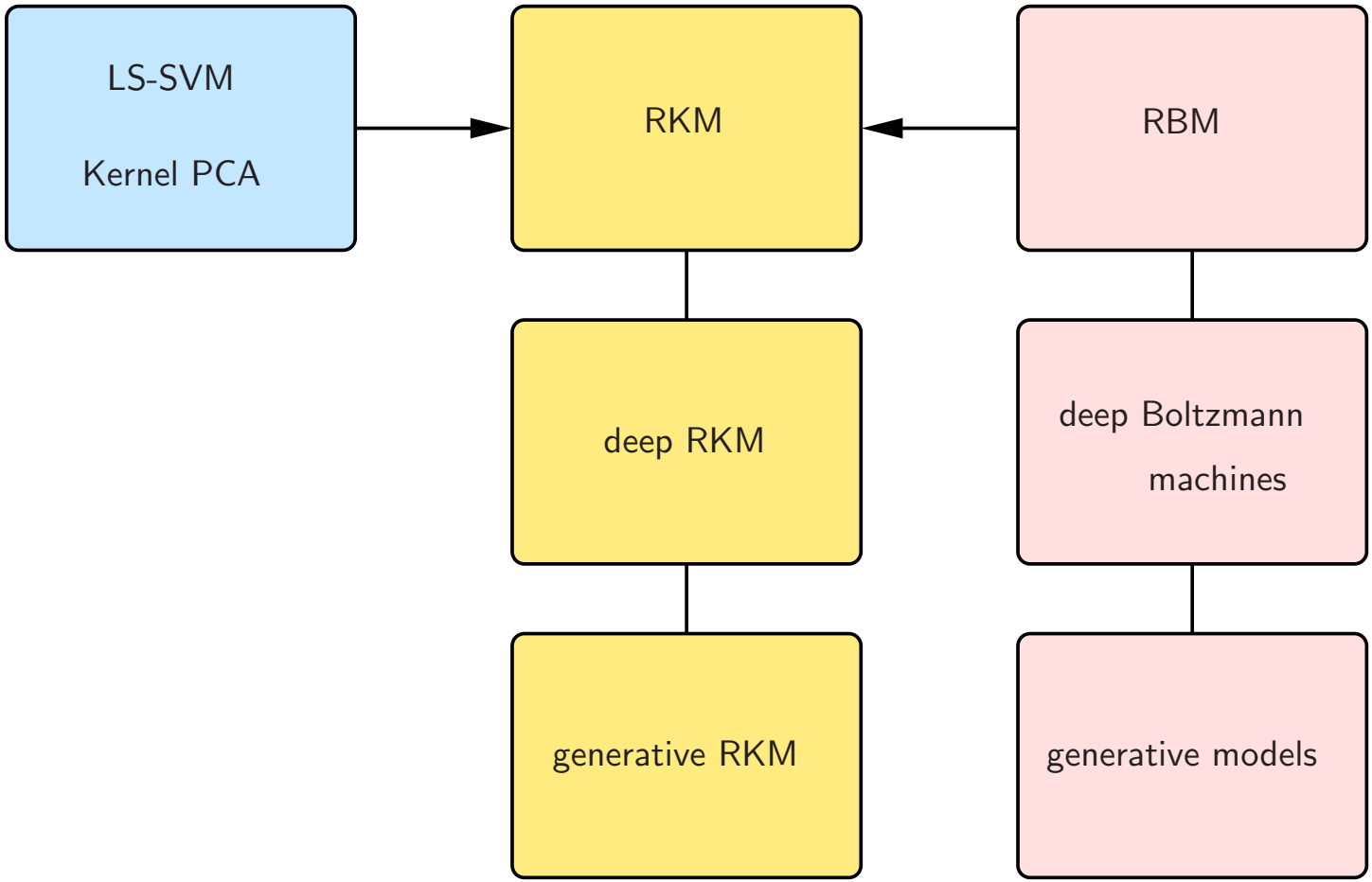
Connecting LSSVM/KPCA & RBM



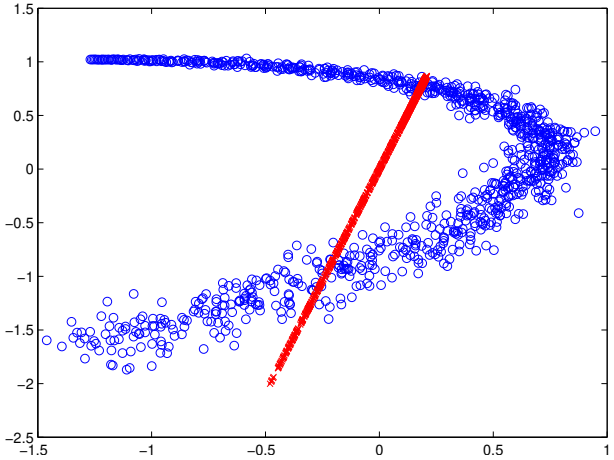
Connecting LSSVM/KPCA & RBM



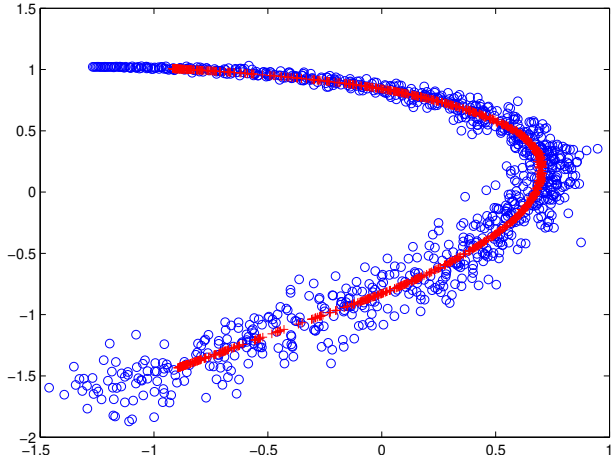
Connecting LSSVM/KPCA & RBM



Kernel principal component analysis (KPCA)



linear PCA



kernel PCA (RBF kernel)

Kernel PCA [Schölkopf et al., 1998]:
take eigenvalue decomposition of the kernel matrix

$$\begin{bmatrix} K(x_1, x_1) & \dots & K(x_1, x_N) \\ \vdots & & \vdots \\ K(x_N, x_1) & \dots & K(x_N, x_N) \end{bmatrix}$$

(applications in dimensionality reduction and denoising)

Kernel PCA: classical LS-SVM approach

- Primal problem: [Suykens et al., 2002]: **model-based** approach

$$\min_{w,b,e} \ \frac{1}{2}w^T w - \frac{1}{2}\gamma \sum_{i=1}^{N} e_i^2 \ \text{ s.t. } \ e_i = w^T \varphi(x_i) + b, \ i = 1, ..., N.$$

- Dual problem (Lagrange duality) corresponds to kernel PCA

$$\Omega^{(c)}\alpha = \lambda\alpha \ \text{ with } \ \lambda = 1/\gamma$$

with $\Omega_{ij}^{(c)} = (\varphi(x_i) - \hat{\mu}_\varphi)^T(\varphi(x_j) - \hat{\mu}_\varphi)$ the *centered kernel matrix* and $\hat{\mu}_\varphi = (1/N)\sum_{i=1}^{N} \varphi(x_i)$.

- Interpretation:
  1. pool of candidate components (objective function equals zero)
  2. select relevant components

- Robust and sparse versions [Alzate & Suykens, 2008]

From KPCA to RKM representation (1)



Model:

$e = W^T \varphi(x)$

objective J
$=$ regularization term $\mathrm{Tr}(W^T W)$
- $(\frac{1}{\lambda})$ variance term $\sum_i e_i^T e_i$

$\downarrow$ use property $e^T h \leq \frac{1}{2\lambda} e^T e + \frac{\lambda}{2} h^T h$

RKM representation:

$e = \sum_j h_j K(x_j, x)$

obtain $J \leq \overline{J}(h_i, W)$
solution from stationary points of $\overline{J}$:
$\frac{\partial \overline{J}}{\partial h_i} = 0$, $\frac{\partial \overline{J}}{\partial W} = 0$

From KPCA to RKM representation (2)

- Objective [Suykens, Neural Computation 2017]

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

- Stationary points of $\overline{J}(h_i, W)$:

$$
\begin{cases}
\dfrac{\partial \overline{J}}{\partial h_i} = 0 \quad \Rightarrow \quad W^T\varphi(x_i) = \lambda h_i, \ \forall i \\
\dfrac{\partial \overline{J}}{\partial W} = 0 \quad \Rightarrow \quad W = \dfrac{1}{\eta}\sum_i \varphi(x_i) h_i^T
\end{cases}
$$

From KPCA to RKM representation (3)

- Elimination of W gives the eigenvalue decomposition:

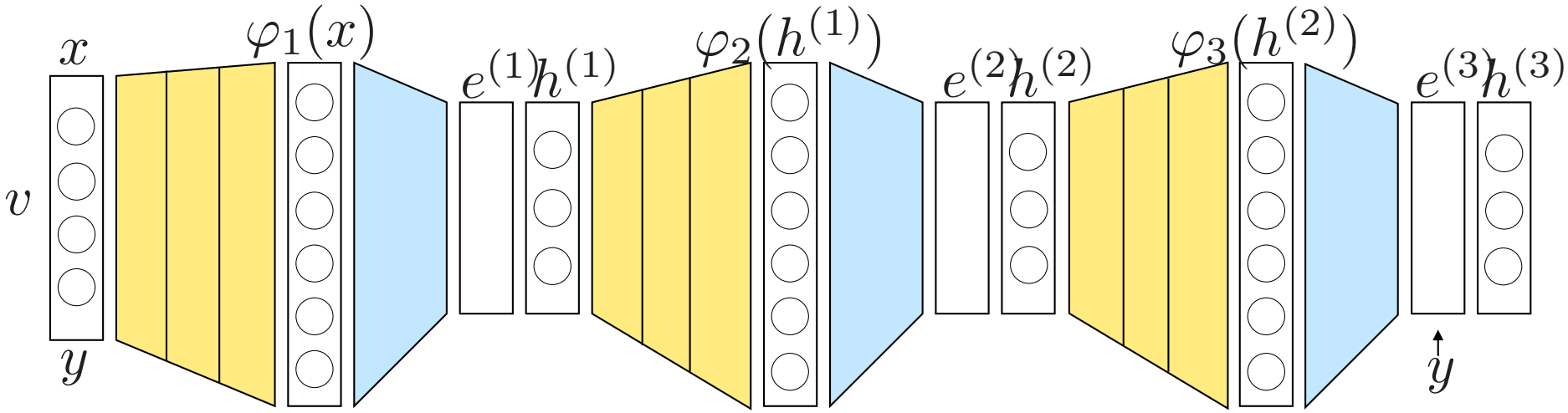
$$\frac{1}{\eta} K H^T = H^T \Lambda$$

where $H = [h_1 ... h_N] \in \mathbb{R}^{s \times N}$ and $\Lambda = \mathrm{diag}\{\lambda_1, ..., \lambda_s\}$ with $s \leq N$

- Primal and dual model representations

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

Deep RKM: example



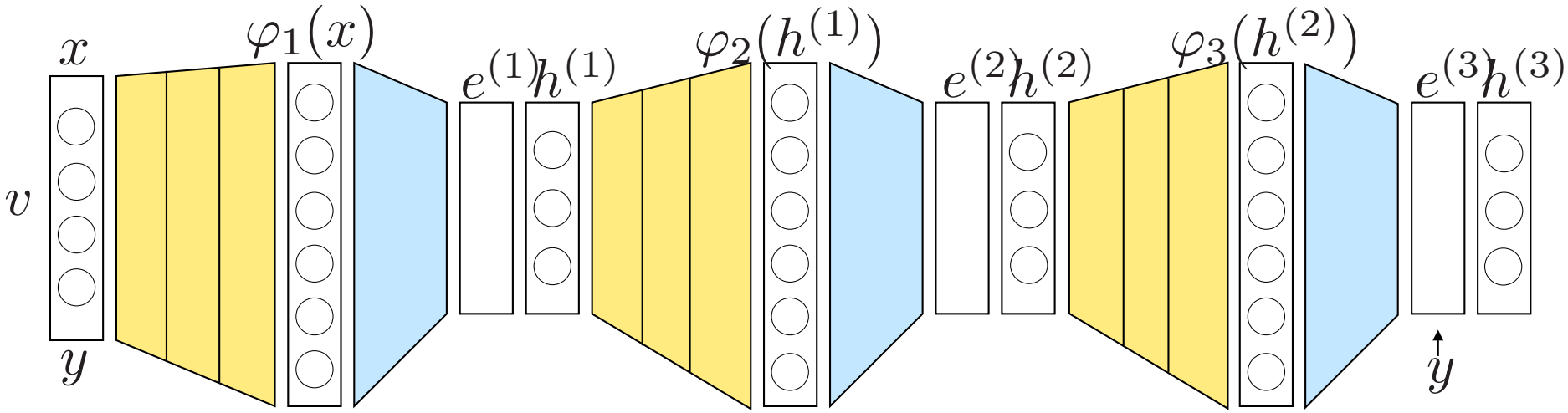
Deep RKM: KPCA + KPCA + LSSVM [Suykens, Neural Computation 2017]

Coupling of RKMs by taking sum of the objectives

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

Multiple *levels* and multiple *layers* per level.

in more detail ...



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

Primal and dual model representations

$$\mathcal{M} \begin{cases} \nearrow & (P)_{\text{DeepRKM}}: \quad \begin{aligned} \hat{e}^{(1)} &= W_1^T \varphi_1(x) \\ \hat{e}^{(2)} &= W_2^T \varphi_2(\Lambda_1^{-1} \hat{e}^{(1)}) \\ \hat{y} &= W_3^T \varphi_3(\Lambda_2^{-1} \hat{e}^{(2)}) + b \end{aligned} \\ \searrow & (D)_{\text{DeepRKM}}: \quad \begin{aligned} \hat{e}^{(1)} &= \frac{1}{\eta_1} \sum_j h_j^{(1)} K_1(x_j, x) \\ \hat{e}^{(2)} &= \frac{1}{\eta_2} \sum_j h_j^{(2)} K_2(h_j^{(1)}, \Lambda_1^{-1} \hat{e}^{(1)}) \\ \hat{y} &= \frac{1}{\eta_3} \sum_j h_j^{(3)} K_3(h_j^{(2)}, \Lambda_2^{-1} \hat{e}^{(2)}) + b \end{aligned} \end{cases}$$

The framework can be used for training deep feedforward neural networks and deep kernel machines [Suykens, 2017].

(Other approaches: e.g. kernels for deep learning [Cho & Saul, 2009], mathematics of the neural response [Smale et al., 2010], deep gaussian processes [Damianou & Lawrence, 2013], convolutional kernel networks [Mairal et al., 2014], multi-layer support vector machines [Wiering & Schomaker, 2014])

"Super-objective": objective for training and generating

- RBM energy function

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

with model parameters $\theta = \{W, c, a\}$

- RKM "super-objective" function (for training and for generating)

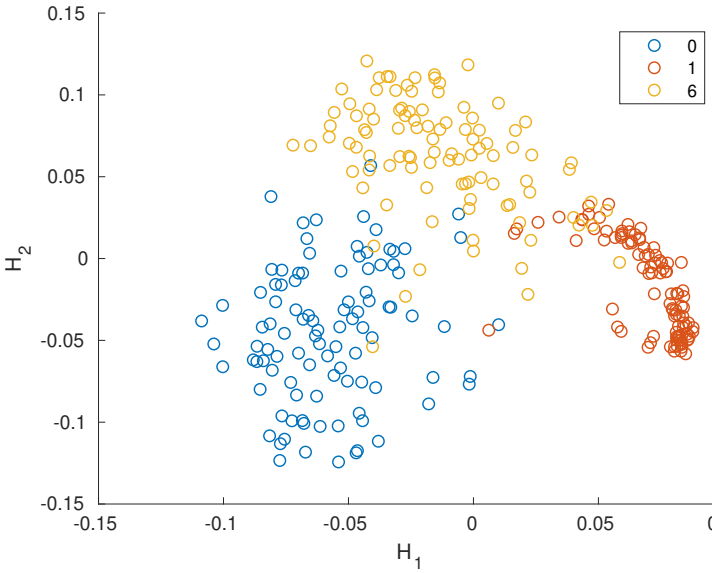
$$\bar{J}(v, h, W) = -v^{\mathrm{T}} W h + \frac{\lambda}{2} h^{\mathrm{T}} h + \frac{\eta}{2} \mathrm{Tr}(W^{\mathrm{T}} W) + \frac{1}{2} v^{\mathrm{T}} v$$

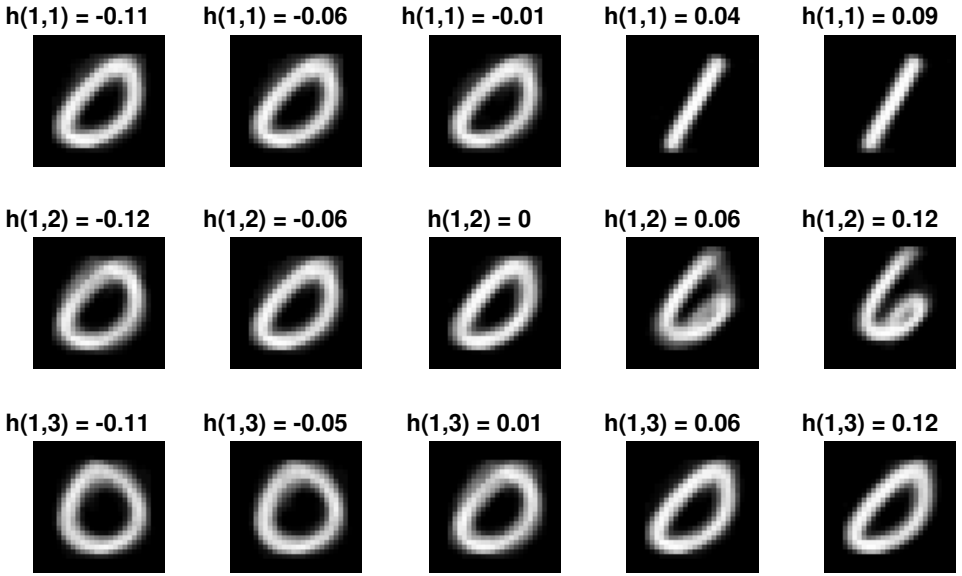
Training:	clamp v	$\rightarrow$	$\bar{J}(h, W)$
Generating:	clamp h, W	$\rightarrow$	$\bar{J}(v)$

[Schreurs & Suykens, ESANN 2018; Achten et al., ICML 2023 workshop]

Latent space exploration

hidden units: exploring the **whole continuum**:

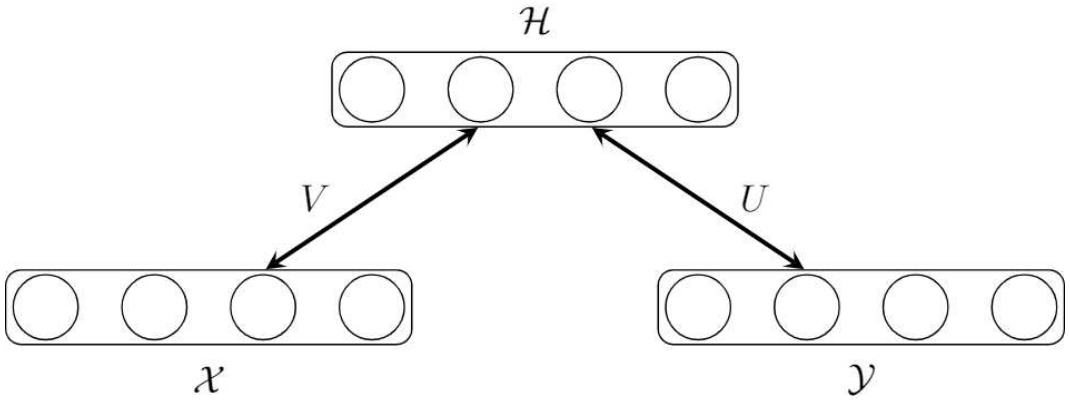




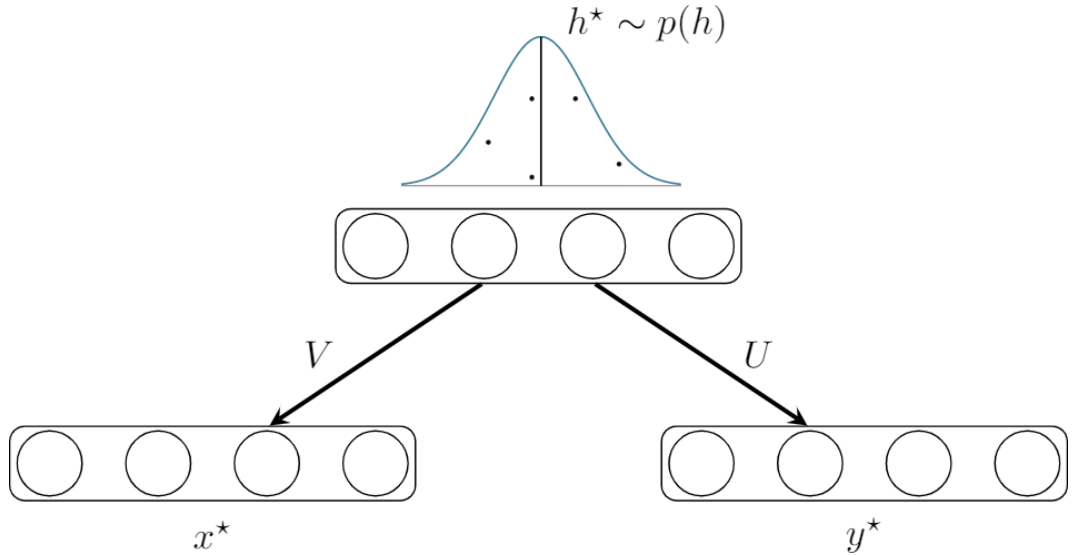
[figures by Joachim Schreurs]

Multi-view Generative RKM (Gen-RKM)

Train:

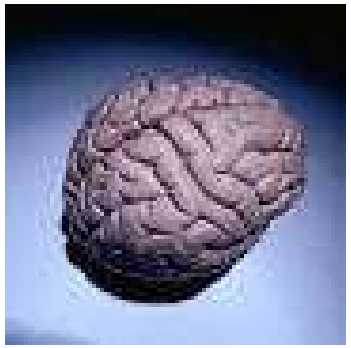


Generate:

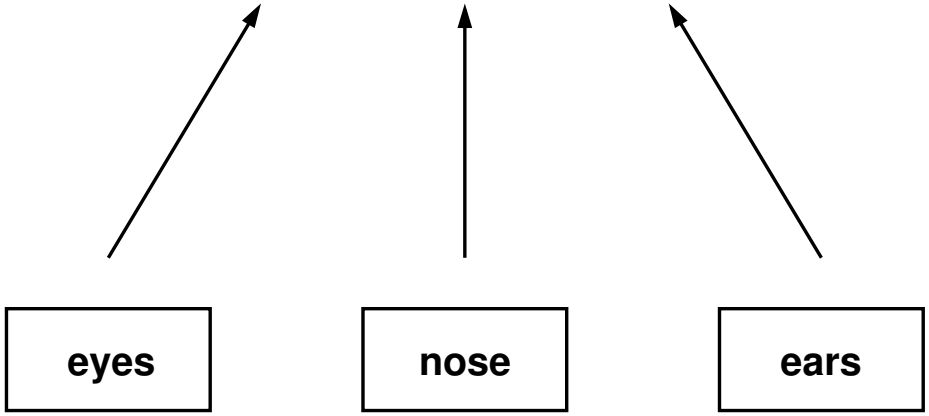


[Pandey, Schreurs & Suykens, Neural Networks, 2021]

Analogy with the human brain



shared feature representation



multimodal data

Multi-view Gen-RKM

The objective

$$J_{\text{train}}(\boldsymbol{h}_i, \boldsymbol{U}, \boldsymbol{V}) = \sum_{i=1}^{N} \left(-\phi_1(\boldsymbol{x}_i)^T \boldsymbol{U} \boldsymbol{h}_i - \phi_2(\boldsymbol{y}_i)^T \boldsymbol{V} \boldsymbol{h}_i + \frac{\lambda}{2} \boldsymbol{h}_i^T \boldsymbol{h}_i\right) + \frac{\eta_1}{2}\text{Tr}(\boldsymbol{U}^T \boldsymbol{U}) + \frac{\eta_2}{2}\text{Tr}(\boldsymbol{V}^T \boldsymbol{V})$$

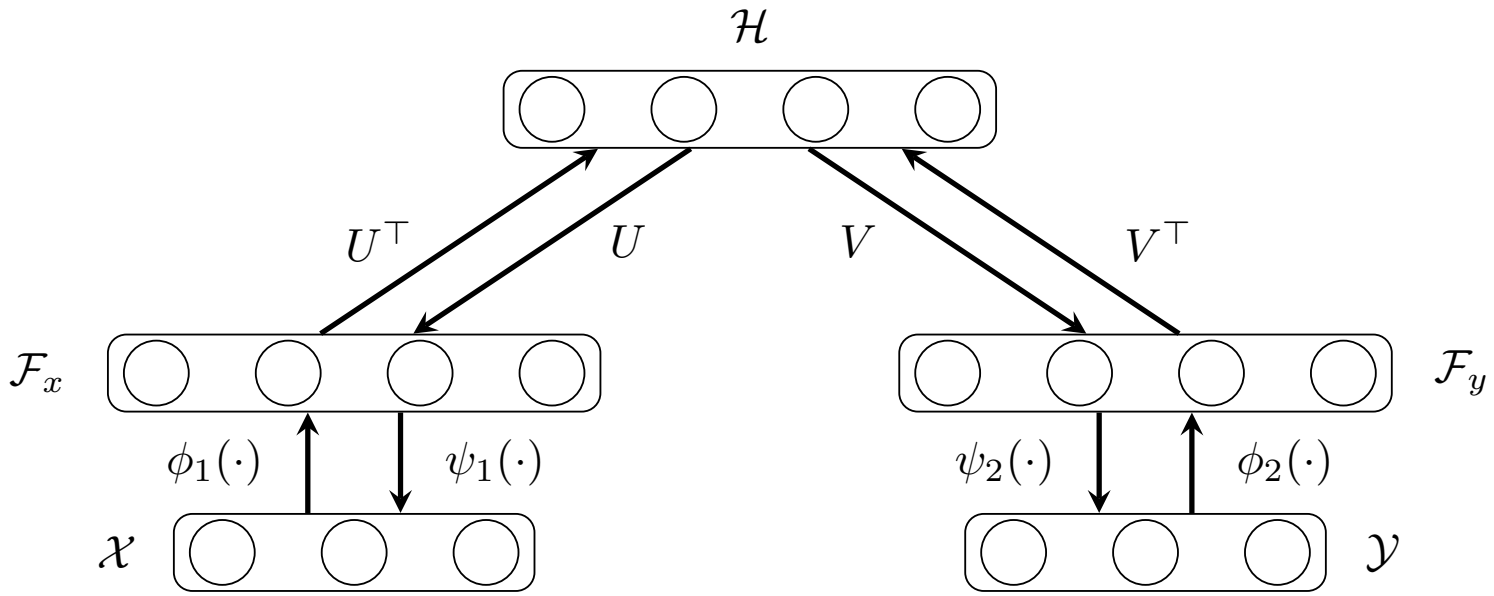
results for **training** into the eigenvalue problem

$$(\frac{1}{\eta_1}\boldsymbol{K}_1 + \frac{1}{\eta_2}\boldsymbol{K}_2)\boldsymbol{H}^T = \boldsymbol{H}^T \boldsymbol{\Lambda}$$

with $\boldsymbol{H} = [\boldsymbol{h}_1 ... \boldsymbol{h}_N]$ and kernel matrices $\boldsymbol{K}_1, \boldsymbol{K}_2$ related to ϕ_1, ϕ_2.

[Pandey, Schreurs & Suykens, Neural Networks, 2021

Multi-view Gen-RKM



Gen-RKM schematic representation modeling a common subspace $\mathcal{H}$ between two data sources $\mathcal{X}$ and $\mathcal{Y}$. The ϕ_1, ϕ_2 are the feature maps ($\mathcal{F}_x$ and $\mathcal{F}_y$ represent the feature-spaces) corresponding to the two data sources. While ψ_1, ψ_2 represent the pre-image maps. The interconnection matrices U, V model dependencies between latent variables and the mapped data sources.

[Pandey, Schreurs & Suykens, Neural Networks, 2021

Gen-RKM: using an explicit (CNN) feature map

Parametrized feature maps: $\phi_{\boldsymbol{\theta}}(\cdot)$, $\psi_{\boldsymbol{\zeta}}(\cdot)$ (e.g. CNN and transposed CNN).

Overall objective function, using a stabilization mechanism [Suykens, 2017]:

$$\begin{aligned}\min_{\boldsymbol{\theta}_1,\boldsymbol{\theta}_2,\boldsymbol{\zeta}_1,\boldsymbol{\zeta}_2} \mathcal{J}_c \;=\; & \mathcal{J}_{\text{train}} + \tfrac{c_{\text{stab}}}{2}\mathcal{J}_{\text{train}}^2 \\ & + \tfrac{c_{\text{acc}}}{2N}\left(\sum_{i=1}^{N}\left[\mathcal{L}_1(\boldsymbol{x}_i^\star, \psi_{1_{\boldsymbol{\zeta}_1}}(\phi_{1_{\boldsymbol{\theta}_1}}(\boldsymbol{x}_i^\star))) + \mathcal{L}_2(\boldsymbol{y}_i^\star, \psi_{2_{\boldsymbol{\zeta}_2}}(\phi_{2_{\boldsymbol{\theta}_2}}(\boldsymbol{y}_i^\star)))\right]\right)\end{aligned}$$

with reconstruction errors

$$\begin{aligned}\mathcal{L}_1(\boldsymbol{x}_i^\star, \psi_{1_{\boldsymbol{\zeta}_1}}(\phi_{1_{\boldsymbol{\theta}_1}}(\boldsymbol{x}_i^\star))) &= \tfrac{1}{N}\|\boldsymbol{x}_i^\star - \psi_{1_{\boldsymbol{\zeta}_1}}(\phi_{1_{\boldsymbol{\theta}_1}}(\boldsymbol{x}_i^\star))\|_2^2, \\ \mathcal{L}_2(\boldsymbol{y}_i^\star, \psi_{2_{\boldsymbol{\zeta}_2}}(\phi_{2_{\boldsymbol{\theta}_2}}(\boldsymbol{y}_i^\star))) &= \tfrac{1}{N}\|\boldsymbol{y}_i^\star - \psi_{2_{\boldsymbol{\zeta}_2}}(\phi_{2_{\boldsymbol{\theta}_2}}(\boldsymbol{y}_i^\star))\|_2^2\end{aligned}$$

and with $\Phi_{\boldsymbol{x}} = [\phi_1(\boldsymbol{x}_1), \ldots, \phi_1(\boldsymbol{x}_N)], \Phi_{\boldsymbol{y}} = [\phi_2(\boldsymbol{y}_1), \ldots, \phi_2(\boldsymbol{y}_N)]$, U, V from

$$\begin{bmatrix} \frac{1}{\eta_1}\Phi_{\boldsymbol{x}}\Phi_{\boldsymbol{x}}^\top & \frac{1}{\eta_1}\Phi_{\boldsymbol{x}}\Phi_{\boldsymbol{y}}^\top \\ \frac{1}{\eta_2}\Phi_{\boldsymbol{y}}\Phi_{\boldsymbol{x}}^\top & \frac{1}{\eta_2}\Phi_{\boldsymbol{y}}\Phi_{\boldsymbol{y}}^\top \end{bmatrix} \begin{bmatrix} U \\ V \end{bmatrix} = \begin{bmatrix} U \\ V \end{bmatrix} \Lambda.$$

Gen-RKM: latent space exploration



CelebA reconstructed images by bilinear-interpolation in latent space
[Pandey, Schreurs & Suykens, Neural Networks 2021]

Gen-RKM - traversals along principal components

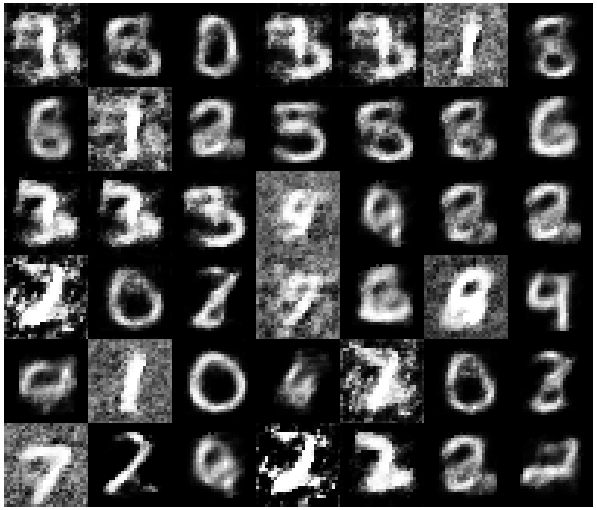


Disentangled Representation Learning and Generation with Manifold Optimization

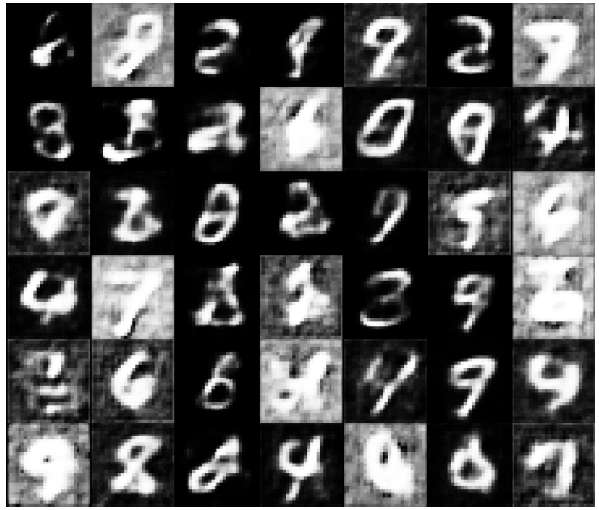
[Pandey, Fanuel, Schreurs & Suykens, Neural Computation, 2022]

Software: see https://www.esat.kuleuven.be/stadius/E/software.php

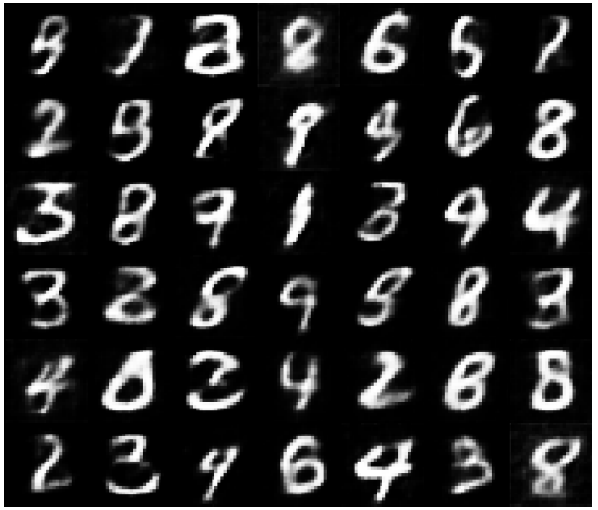
Robust Gen-RKM - Robust generation



VAE



Gen-RKM

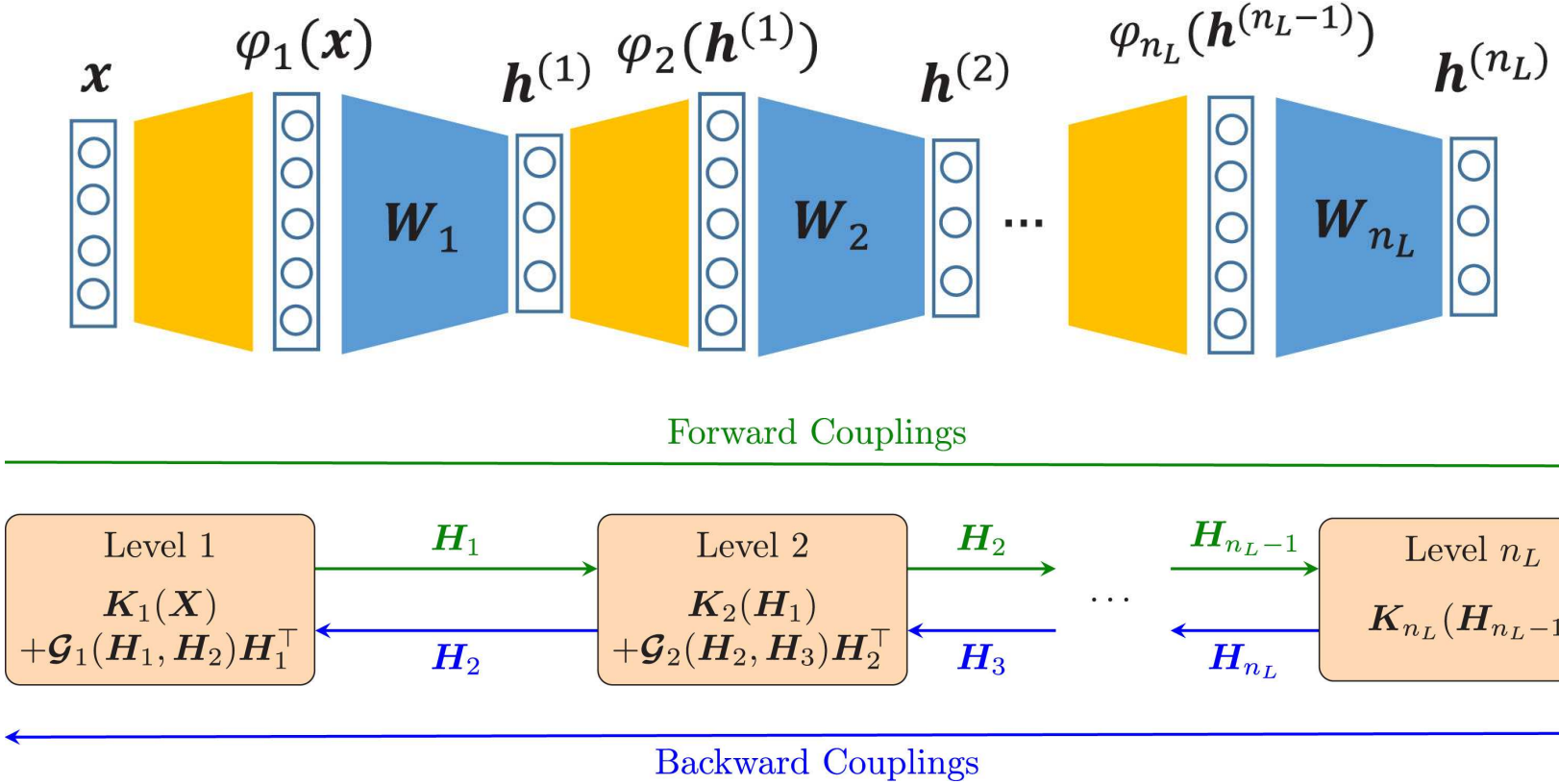


Robust Gen-RKM

Weighted conjugate feature duality: $\frac{1}{2\lambda}e^T De + \frac{\lambda}{2}h^T D^{-1}h \geq e^T h$

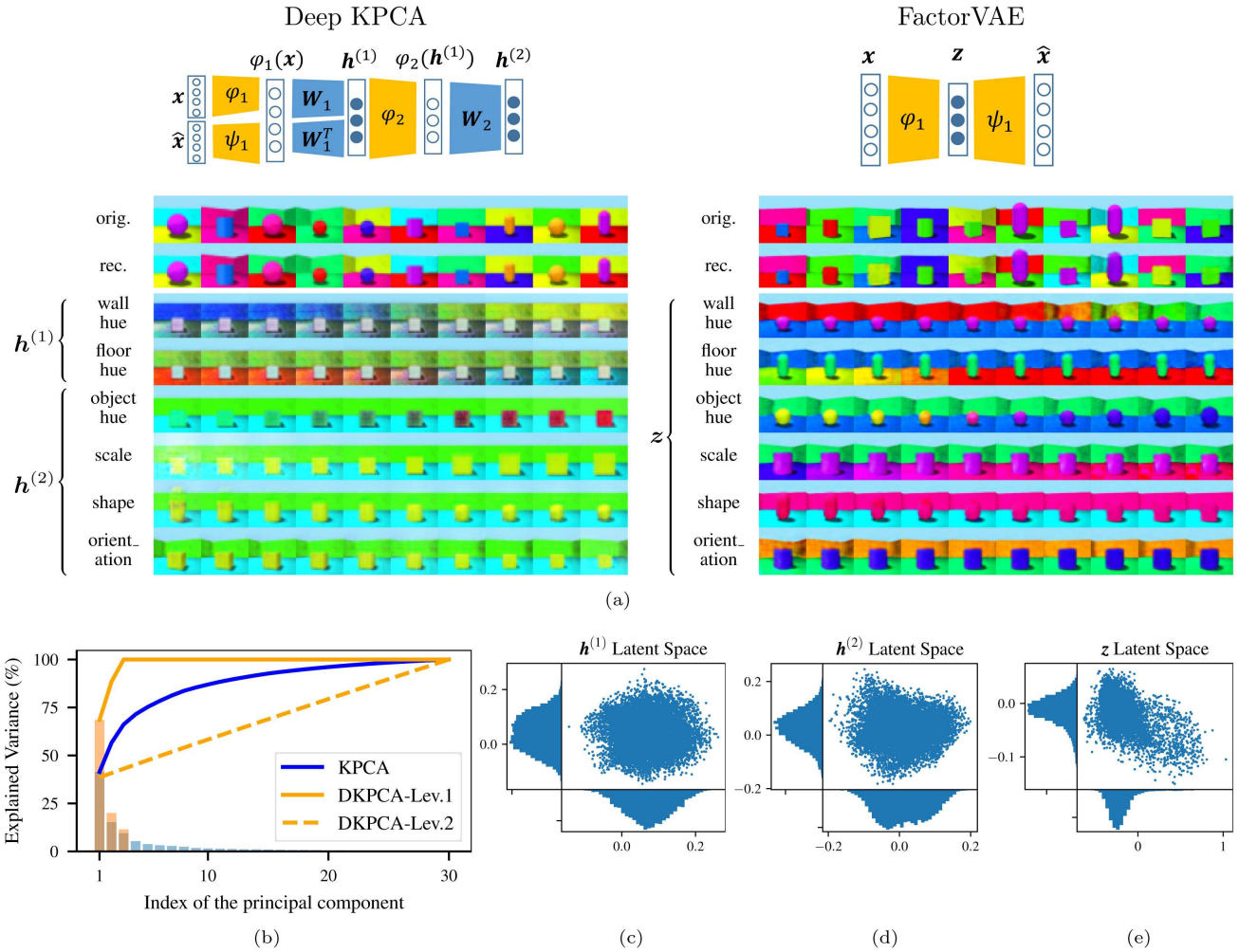
[Pandey, Schreurs & Suykens, LOD 2020]

Deep KPCA based on RKM representations



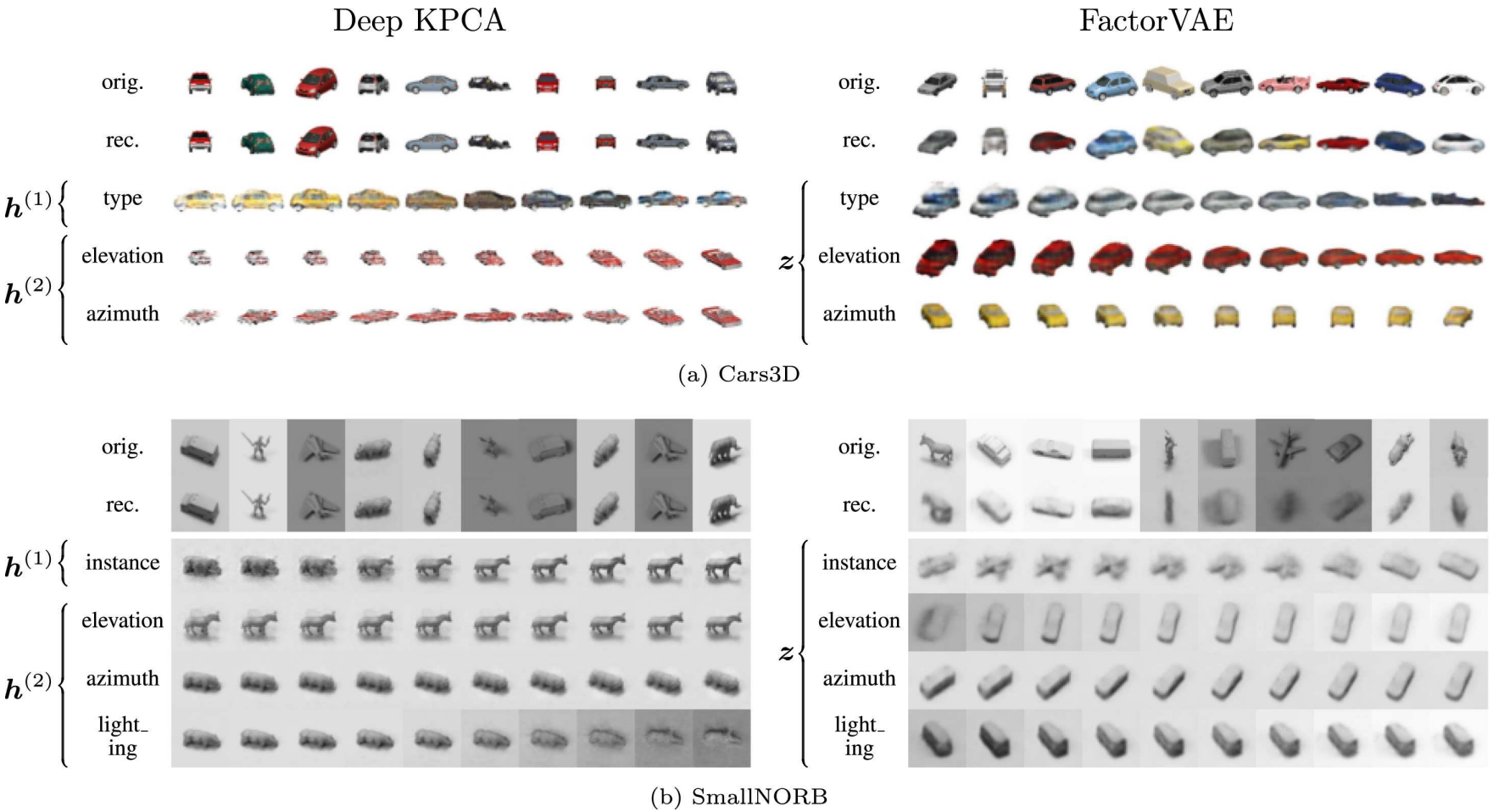
[Tonin, Tao, Patrinos & Suykens, Neural Networks 2024]

Towards "white box": deep eigenvalues/eigenvectors (1)



[Tonin, Tao, Patrinos & Suykens, Neural Networks 2024]

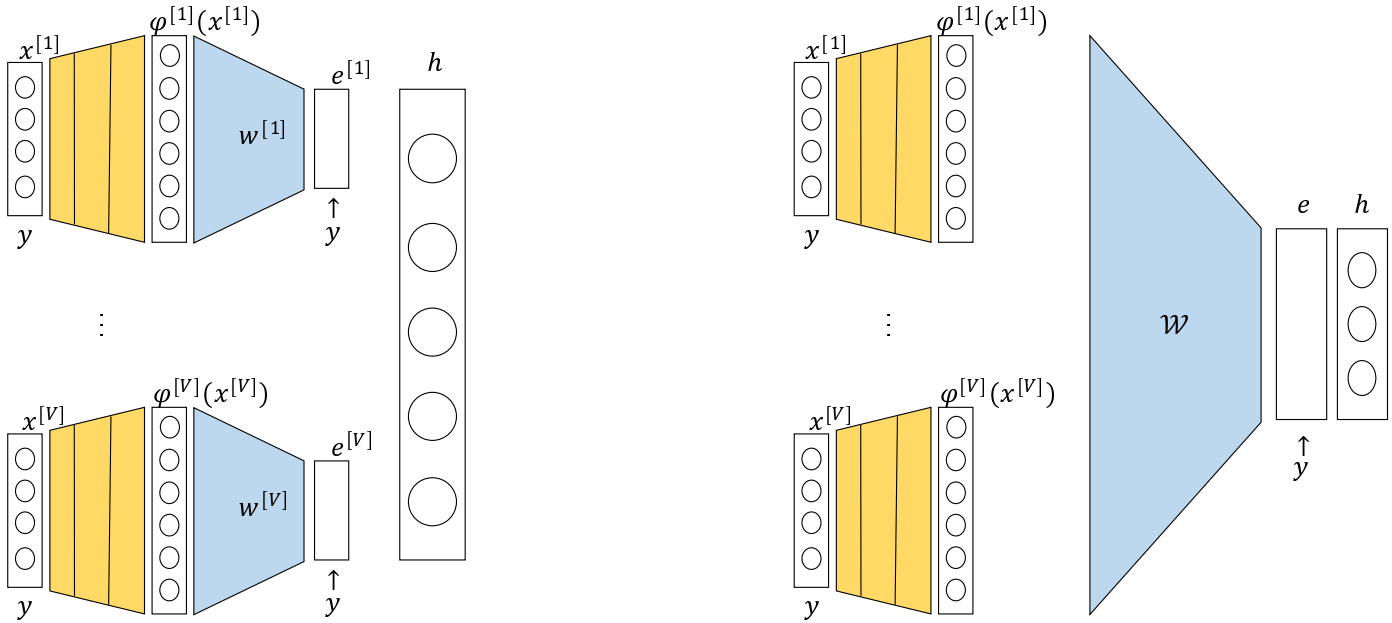
Towards "white box": deep eigenvalues/eigenvectors (2)



[Tonin, Tao, Patrinos & Suykens, Neural Networks 2024]

Tensor-based RKM for Multi-view KPCA

$$\min \langle \mathcal{W}, \mathcal{W} \rangle - \sum_{i=1}^{N} \left\langle \Phi_{(i)}, \mathcal{W} \right\rangle h_i + \lambda \sum_{i=1}^{N} h_i^2 \quad \text{with} \quad \Phi_{(i)} = \varphi^{[1]}(x_i^{[1]}) \otimes \ldots \otimes \varphi^{[V]}(x_i^{[V]})$$



[Houthuys & Suykens, ICANN 2018]

Recurrent RKM

- Objective for **Recurrent RKM** [Pandey et al., 2022]:

$$J = -\sum_t \varphi(x_t)^T W h_t - \sum_t \sum_l h_{t-l}^T h_t + \frac{1}{2} h_t^T \Lambda h_t + \frac{1}{2} \mathrm{Tr} W^T W$$

- Results into a sequence $\{h_t\}$ in the latent space from which the future of the time-series is predicted.

- $\{h_t\}$ follows from the solution to an eigenvalue problem.

- Related work on recurrent temporal RBM [Sutskever et al., 2008; Osogami, 2019]

[Pandey, De Meulemeester, De Plaen, De Moor, Suykens, ESANN 2022]

Future Challenges and Conclusions

- LS-SVM as **core models** in supervised and unsupervised learning, and beyond
- **primal and dual** model representations, **parametrized feature maps** or **kernel-based**
- combining powerful deep learning models with kernel-based setting
- **new connections with Transformers and RBMs** within LS-SVM framework
- new schemes for **generative models** and **deep kernel machines**
- **multi-view** and **tensor-based** schemes
- robustness, disentanglement, latent space exploration

Acknowledgements (1)

- Current and former co-workers at ESAT-STADIUS:

  S. Achten, C. Alzate, Y. Chen, J. De Brabanter, K. De Brabanter, B. De Cooman, L. De Lathauwer, H. De Meulemeester, B. De Moor, H. De Plaen, Ph. Dreesen, M. Espinoza, T. Falck, M. Fanuel, Y. Feng, B. Gauthier, B. Hamers, F. He, L. Hoegaerts, X. Huang, L. Houthuys, V. Jumutc, Z. Karevan, A. Lambert, R. Langone, F. Liu, L. Lukas, R. Mall, S. Mehrkanoon, M. Orchel, A. Pandey, P. Patrinos, K. Pelckmans, S. RoyChowdhury, S. Salzo, J. Schreurs, M. Signoretto, Q. Tao, F. Tonin, J. Vandewalle, T. Van Gestel, S. Van Huffel, C. Varon, D. Winant, Y. Yang, X. Zeng and others

- Many other people for joint work, discussions, invitations, organizations

- Support from ERC AdG E-DUALITY, ERC AdG A-DATADRIVE-B, KU Leuven, OPTEC, IUAP DYSCO, FWO projects, IWT, Flanders AI, Leuven.AI

Acknowledgements (2)







Acknowledgements (3)



ERC Advanced Grant E-DUALITY

Exploring duality for future data-driven modelling

Thank you