UPPSALA
UNIVERSITET

On Deep Learning for Low-Dimensional Representations

Daniel Gedon

Uppsala University

Disputation

Uppsala, June 14, 2024

Deep Learning for Low-Dimensional Representation

Fundamental
- **Paper I**
- Paper II
- **Paper III**

Applications
- **Paper VI**
- Paper V
- **Paper IV**

I. Uncertainty Estimation with Recursive Feature Machines, *UAI 2024*

II. Invertible Kernel PCA with Random Fourier Features, *IEEE Signal Processing Letters 2023*

III. No Double Descent in Principal Component Regression: A High-Dimensional Analysis, *ICML 2024*

IV. Deep State Space Model for Nonlinear System Identification, *SYSID 2021*

V. First Steps Towards Self-Supervised Pretraining of the 12-Lead ECG, *CinC 2021*

VI. Development and Validation of Deep Learning ECG-Based Prediction of Myocardial Infarction in Emergency Department Patients, *Scientific Reports 2022*

Part I. Problem formulation

Part II. Fundamental ML

Part III. Applications

UPPSALA UNIVERSITET

1. **Machine Learning**
   - What is it?
   - When do we need it?

2. **Low-dimensional representations**
   - What are representations?
   - Why are they low-dimensional?

$$\Downarrow$$

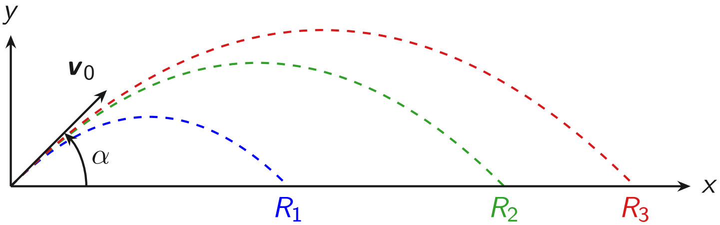
Part II: Machine learning for low-dimensional data structure

Part III: Low-dimensional representations for deep learning application

Scenario:

Real-world observations $\rightarrow$ model $\rightarrow$ $\boxed{\text{model}}$ $\rightarrow$ explanation / prediction

Example 1: Trajectory motion

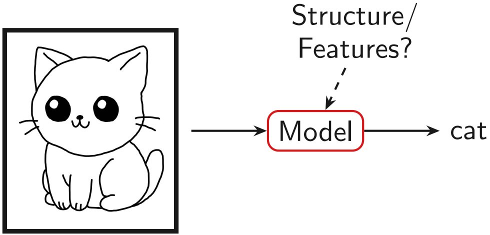


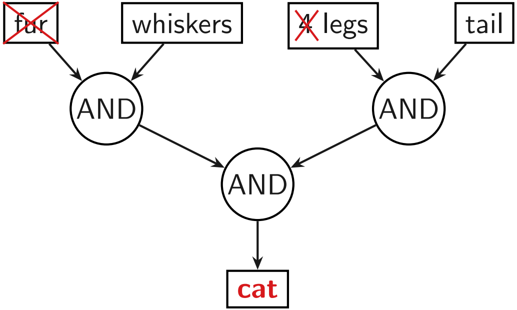
Model $f : \boldsymbol{v}_0 \mapsto R$ given α, g: $$R = f(\boldsymbol{v}_0) = \frac{\boldsymbol{v}_0^2 \sin 2\alpha}{g}$$

$\rightarrow$ Occam's razor / Principle of parsimony

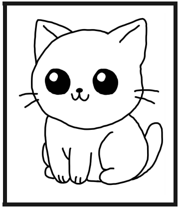
UPPSALA UNIVERSITET

Example 2: Image classification





$\Rightarrow$ Fails in many real-world scenarios



which

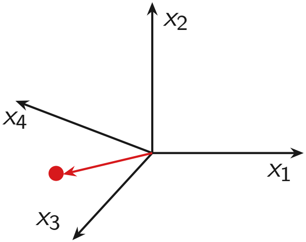
features?

Electrocardiogram (ECG)

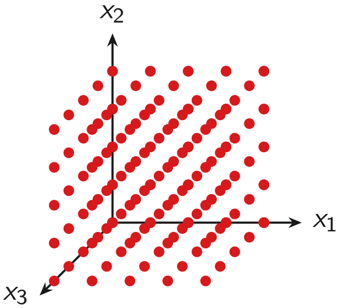
Input $\longrightarrow$ Model / DNN $\longrightarrow$ Prediction

-0.34
0.78
0.21
-0.09
0.89
-0.17

high-dimensional features $\Rightarrow$



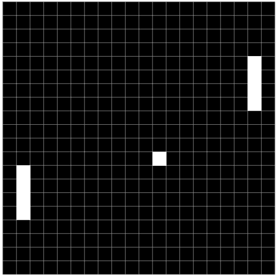
Low-dimensional representations



1-dim space takes volume: $V = s^1$ 2-dim space takes volume: $V = s^2$ 3-dim space takes volume: $V = s^3$ p-dim space takes volume: $V = s^p$

$\rightarrow$ exponentially more samples needed!

Example: Atari game Pong



Problem: 20×20 pixels $\rightarrow$ 400 dimensions

Solution: only need x/y-position of ball

$$\boldsymbol{z} = [11, 8]^\top \text{ with } \mathcal{Z} \in \mathbb{R}^2$$

$\rightarrow$ Obtain useful low-dim. representation

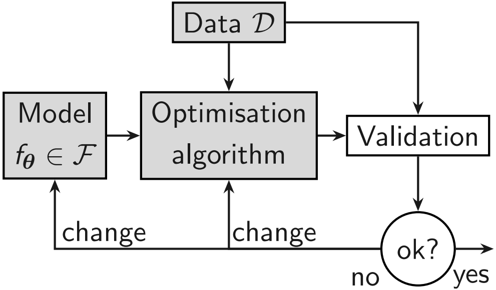
Part I. Problem formulation

Part II. Fundamental ML

Part III. Applications

UPPSALA UNIVERSITET

Overview



$\mathcal{D} = (\boldsymbol{x}, y) \mid \boldsymbol{x} \in \mathcal{X}, y \in \mathcal{Y}$

Parameterised model $f_{\boldsymbol{\theta}} : \mathcal{X} \mapsto \mathcal{Y}$

L layers $f_{\boldsymbol{\theta}} = f^L_{\boldsymbol{\theta}_L} \circ \cdots \circ f^2_{\boldsymbol{\theta}_2} \circ f^1_{\boldsymbol{\theta}_1}$

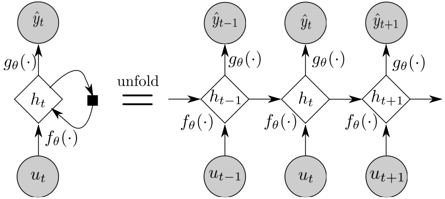
Empirical risk minimisation

$$\arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), y_i)$$

Architectures for low-dim. representations

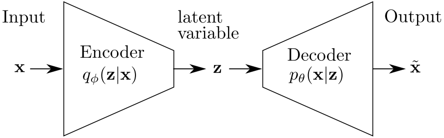
Recurrent neural network:

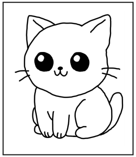
→ past information in state h



Variational autoencoder:

→ bottleneck layer





Similar data

$\downarrow$

Same prediction

Define kernel function $k(\boldsymbol{x}, \boldsymbol{x}') \mapsto \mathbb{R}$

Prediction: $f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i k(\boldsymbol{x}, \boldsymbol{x}_i)$

Which kernel to choose?

- Linear: $k_{lin}(\boldsymbol{x}, \boldsymbol{x}') := \langle \boldsymbol{x}, \boldsymbol{x}' \rangle$
- RBF: $k_{rbf}(\boldsymbol{x}, \boldsymbol{x}') := \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|^2}{2\ell^2}\right)$
- Mahalanobis:

$$k_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{x}') := \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|_{\boldsymbol{M}}}{\ell}\right)$$

$$\|\boldsymbol{x} - \boldsymbol{x}'\|_{\boldsymbol{M}}^2 = (\boldsymbol{x} - \boldsymbol{x}')^\top \boldsymbol{M}(\boldsymbol{x} - \boldsymbol{x}')$$

Recursive Feature Machines

Kernel Machine

$$f_{\boldsymbol{M}}(x) = k_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{X})\alpha$$

$$k_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{z}) = \exp(-1/\gamma \|\boldsymbol{x} - \boldsymbol{z}\|_{\boldsymbol{M}})$$

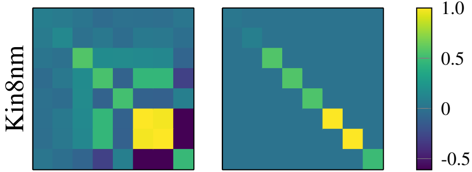
Feature Matrix

$$\boldsymbol{M} = \frac{1}{n} \sum_{p=1}^{n} \nabla f_{\boldsymbol{M}}(\boldsymbol{x}_p) \nabla f_{\boldsymbol{M}}(\boldsymbol{x}_p)^T$$

Why Average Gradient Outer Product?

$$\boldsymbol{W}_l^\top \boldsymbol{W}_l \propto \frac{1}{n} \sum_{i=1}^{n} \nabla f^l(\tilde{\boldsymbol{x}}_i) \nabla f^l(\tilde{\boldsymbol{x}}_i)^\top$$

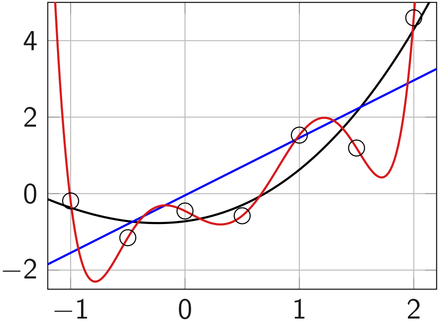
Why low-dimensional?

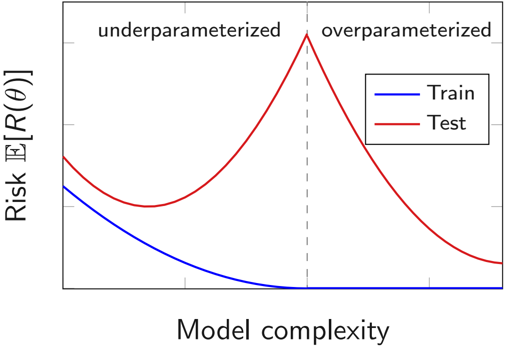


Overparameterization (1/3)

How well do machine learning models perform? $\rightarrow$ Generalisation risk

Example: regression



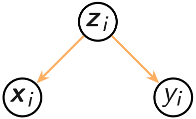


Model complexity:

- Parameters $p <$ number of training data points n: underparameterized
- Parameters $p >$ number of training data points n: overparameterized

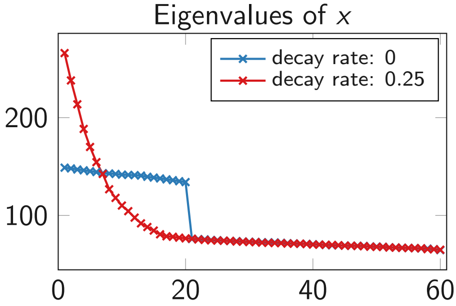
Overparameterization (2/3)

Data generator

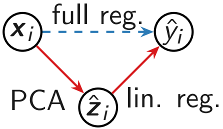


$$\boldsymbol{x}_i = \boldsymbol{W}_{r_w} \boldsymbol{z}_i + \boldsymbol{e}_i$$

$$y_i = \boldsymbol{\theta}^\top \boldsymbol{z}_i + \varepsilon_i$$



Model

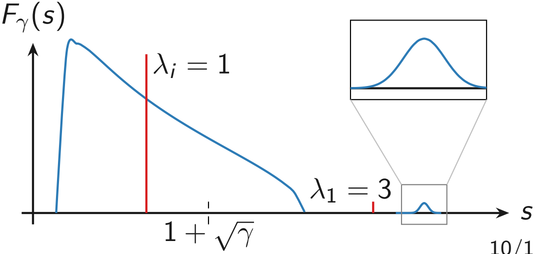


SVD: $\boldsymbol{X} \approx \hat{\boldsymbol{U}} \hat{\Sigma} \hat{\boldsymbol{V}}^\top,$

PCA: $\hat{\boldsymbol{z}}_i = \hat{\boldsymbol{V}}^\top \boldsymbol{x}_i,$

lin. reg. $\hat{y}_i = \hat{\boldsymbol{\theta}}^\top \hat{\boldsymbol{z}}_i.$

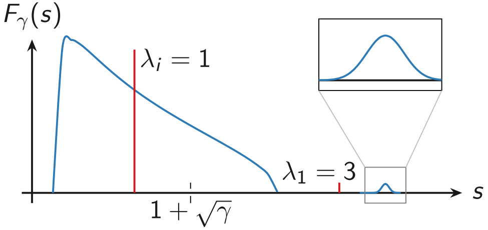
High-dim. random matrix theory



UPPSALA UNIVERSITET

High dimensional analysis based on random matrix theory

Eigenvalue shift:



Risk:

$$R(\hat{\boldsymbol{\theta}}) = \mathbb{E}_{(\boldsymbol{x}_0, y_0)\sim\mathcal{D}}\left[(y_0 - \hat{y}_0(\boldsymbol{x}_0))^2\right]$$

Asymptotic PCR risk:

$$\mathbb{E}_\nu\left[R(\hat{\theta})\right] \to \mathrm{Bias}_\gamma(\hat{\theta})^2 + \mathrm{Var}_\gamma(\hat{\theta}) + \sigma^2$$

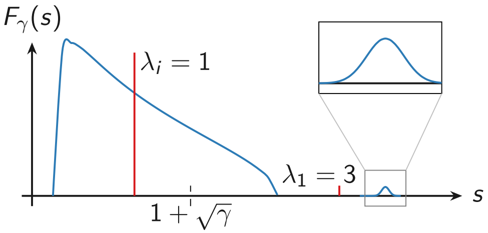
With the following:

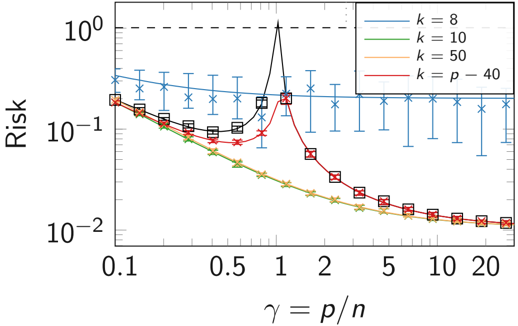
$$\mathrm{Bias}_\gamma(\hat{\theta})^2 = \bar{\beta}^\top\big(\Lambda_d - \Lambda_d\boldsymbol{P} - \boldsymbol{P}\Lambda_d + \boldsymbol{P} + \boldsymbol{P}r_w^2\boldsymbol{C}_z\boldsymbol{P}\big)\bar{\beta}$$

$$\mathrm{Var}_\gamma(\hat{\theta}) = \frac{\sigma^2}{n}\left(\mathrm{Tr}\left[(\boldsymbol{P}r_w^2\boldsymbol{C}_z + \boldsymbol{I})\frac{1}{\mu(\Lambda,\gamma)}\right] + (p-d)\int_c^{(1+\sqrt{\gamma})^2}\frac{1}{s}dF_\gamma(s)\right)$$

High dimensional analysis based on random matrix theory

Eigenvalue shift:





Risk:

$$R(\hat{\boldsymbol{\theta}}) = \mathbb{E}_{(\boldsymbol{x}_0, y_0)\sim\mathcal{D}}\left[(y_0 - \hat{y}_0(\boldsymbol{x}_0))^2\right]$$

Asymptotic PCR risk:

$$\mathbb{E}_\nu\left[R(\hat{\theta})\right] \to \mathrm{Bias}_\gamma(\hat{\theta})^2 + \mathrm{Var}_\gamma(\hat{\theta}) + \sigma^2$$

Part I. Problem formulation

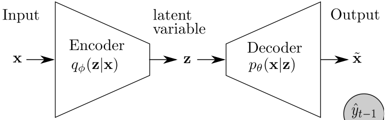
Part II. Fundamental ML

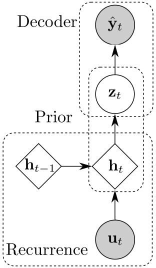
Part III. Applications

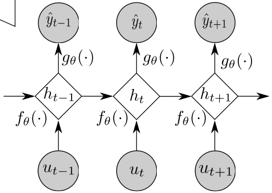
Dynamical Systems (1/3)

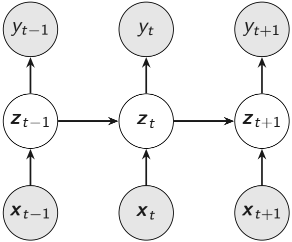
Modelling of systems:

- Fully-connected networks
- (Temporal) Convolutional networks
- Recurrent neural networks
- Latent variable models
  - Autoencoder
  - Variational autoencoder
  - **Deep state-space models**
- Energy-based models









Temporal extension of VAE $\rightarrow$ time-varying prior for latent $\boldsymbol{z}$

Model definition:

- Joint model distribution

$$p_{\theta}(y_{1:T}, \boldsymbol{z}_{1:T} | \boldsymbol{x}_{1:T}, \boldsymbol{z}_0) = \prod_{t=1}^{T} p_{\theta}(y_t | \boldsymbol{z}_t) p_{\theta}(\boldsymbol{z}_t | \boldsymbol{z}_{t-1}, \boldsymbol{x}_t)$$

- Prior and decoder

$$p_{\theta}(\boldsymbol{z}_t | \boldsymbol{z}_{t-1}, \boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{z}_t | \mu_t^{\text{prior}}, \sigma_t^{\text{prior}} \boldsymbol{I})$$
$$p_{\theta}(y_t | \boldsymbol{z}_t) = \mathcal{N}(y_t | \mu_t^{\text{dec}}, \sigma_t^{\text{dec}} \boldsymbol{I})$$

- For training: variational encoder distribution

$$q_{\phi}(\boldsymbol{z}_{1:T} | y_{1:T}, \boldsymbol{x}_{1:T}, \boldsymbol{z}_0) = \prod_{t=1}^{T} q_{\phi}(\boldsymbol{z}_t | \boldsymbol{z}_{t-1}, y_{t:T}, \boldsymbol{x}_{t:T})$$

$\rightarrow$ Requires nonlinear smoothing step for training

Temporal extension of VAE $\rightarrow$ time-varying prior for latent $\boldsymbol{z}$

Model definition:

- Joint model distribution

$$p_\theta(y_{1:T}, \boldsymbol{z}_{1:T}, \boldsymbol{h}_{1:T} | \boldsymbol{x}_{1:T}, \boldsymbol{h}_0) = \prod_{t=1}^{T} p_\theta(y_t|\boldsymbol{z}_t) p_\theta(\boldsymbol{z}_t|\boldsymbol{h}_t) p_\theta(\boldsymbol{h}_t|\boldsymbol{h}_{t-1}, \boldsymbol{x}_t)$$

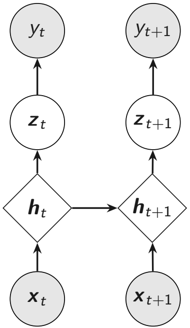
- Prior and decoder

$$p_\theta(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, \boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{z}_t|\mu_t^{\text{prior}}, \sigma_t^{\text{prior}} \boldsymbol{I})$$
$$p_\theta(y_t|\boldsymbol{z}_t) = \mathcal{N}(y_t|\mu_t^{\text{dec}}, \sigma_t^{\text{dec}} \boldsymbol{I})$$

- For training: variational encoder distribution

$$q_\phi(\boldsymbol{z}_{1:T}, \boldsymbol{h}_{1:T} | y_{1:T}, \boldsymbol{x}_{1:T}, \boldsymbol{h}_0) = \prod_{t=1}^{T} q_\phi(\boldsymbol{z}_t|y_t, \boldsymbol{h}_t) p_\theta(\boldsymbol{h}_t|\boldsymbol{h}_{t-1}, \boldsymbol{x}_t)$$

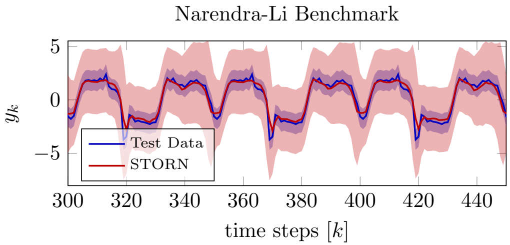
$\rightarrow$ Solution: De-couple state $\boldsymbol{z}_t$ with hidden state $\boldsymbol{h}_t$

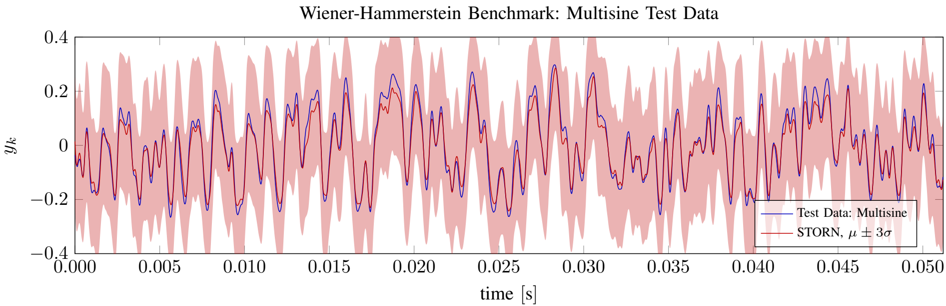


UPPSALA UNIVERSITET

$$\begin{bmatrix} x_{k+1}^{(1)} \\ x_{k+1}^{(2)} \end{bmatrix} = \begin{bmatrix} \left(\frac{x_k^{(1)}}{1+(x_k^{(1)})^2} + 1\right) \sin(x_k^{(2)}) \\ x_k^{(2)} \cos(x_k^{(2)}) + x_k^{(1)} \exp(-\frac{(x_k^{(1)})^2+(x_k^{(2)})^2}{8}) + \dots \\ \dots + \frac{u_k^3}{1+u_k^2+0.5\cos(x_k^{(1)}+x_k^{(2)})} \end{bmatrix},$$

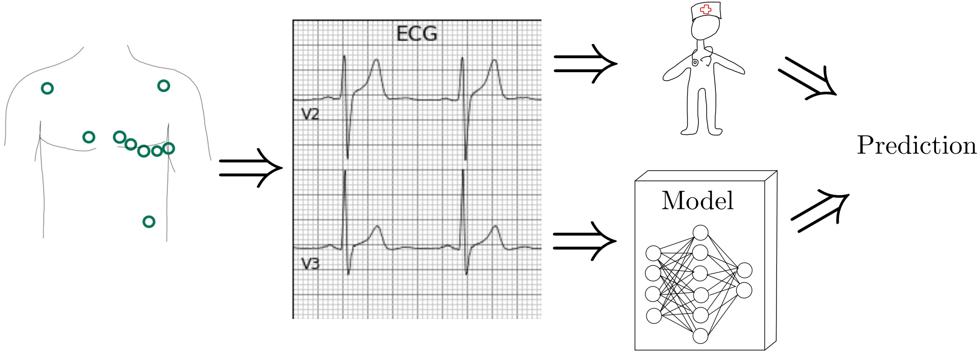
$$y_k = \frac{x_k^{(1)}}{1+0.5\sin(x_k^{(2)})} + \frac{x_k^{(2)}}{1+0.5\sin(x_k^{(1)})} + e_k .$$





Medical Applications (1/2)

Cardiovascular diseases: $\approx$18 million deaths in 2019 (32%)



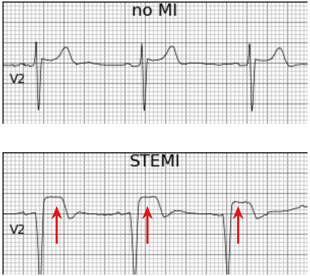
ECG is a major diagnostic tool:

- low-cost, safe, quick, non-invasive
- Can detect arrhythmias, myocardial infarctions, cardiomyopathy, ...

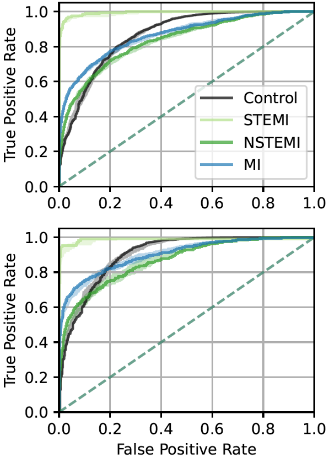
Self-supervised (**V**), NSTEMI (**VI**), atrial fibrillation, electrolyte regression, Chagas, multiclass challenge.

Myocardial Infarctions

- ST-elevation MI
  → ECG
- non-ST-elevation MI
  → blood test

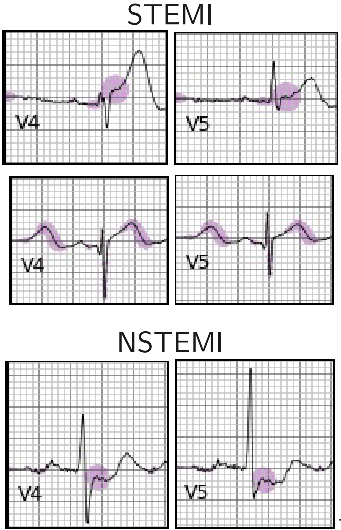


ROC curve

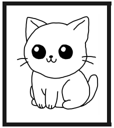


Top: temporal split.
Bottom: random split.

Grad-CAM

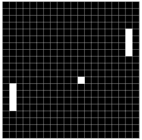


Conclusion



Data variations $\rightarrow$ machine learning

Real world data: low-dimensional



Kernel Machine

$f_M(x) = k_M(\boldsymbol{x}, \boldsymbol{X})\alpha$

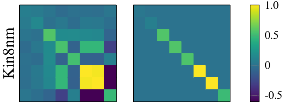
$k_M(\boldsymbol{x}, \boldsymbol{z}) = \exp(-1/\gamma \|\boldsymbol{x} - \boldsymbol{z}\|_M)$

Feature Matrix

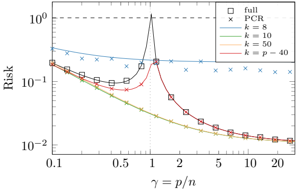
$\boldsymbol{M} = \frac{1}{n}\sum_{p=1}^{n} \nabla f_M(\boldsymbol{x}_p) \nabla f_M(\boldsymbol{x}_p)^T$

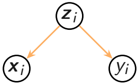
Recursive Feature Machine

$\rightarrow$ uncertainty quantification



High-dimensional risk analysis

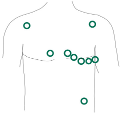




$u(t) \longrightarrow$ Dynamical System $\longrightarrow y(t)$

Deep state-space model

Electrocardiogram modelling



Thank you!

Daniel Gedon, Uppsala University

E-mail: `daniel.gedon@it.uu.se`
Web: `dgedon.github.io`
Twitter / $\mathbb{X}$: `@danigedon`