Learning Deep Dynamical Models from Image Pixels

Niklas Wahlström[1], Thomas B. Schön[2], Marc P. Deisenroth[3]

[1]Department of Electrical Engineering, Linköping University, Sweden

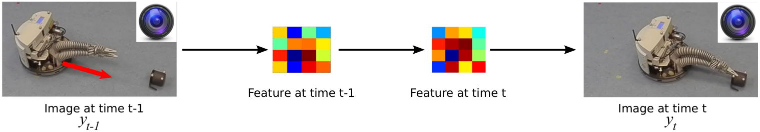
[2]Department of Information Technology, Uppsala University, Sweden

[3]Department of Computing, Imperial College London, UK

LINKÖPING UNIVERSITY

Motivation

- **Vision:** fully autonomous systems that learn by themselves from raw pixel data.
- **This paper:** Modeling of high-dimensional pixel data
- **Strategy:** A deep dynamical model is proposed that contains a low-dimensional dynamical model.

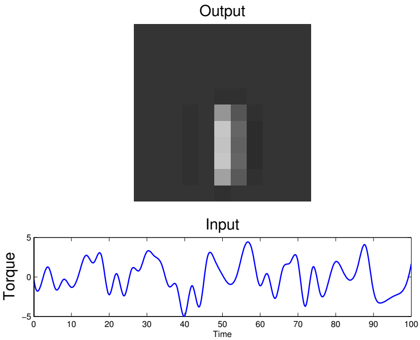


Problem Formulation

Problem formulation: Modeling of high-dimensional pixel data

Example: Video stream of a pendulum

- **Input:** Torque of a pendulum
- **Output:** Pixel values of an 11×11 image

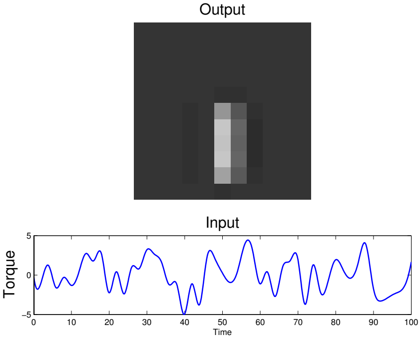


Problem Formulation

Problem formulation: Modeling of high-dimensional pixel data

Example: Video stream of a pendulum

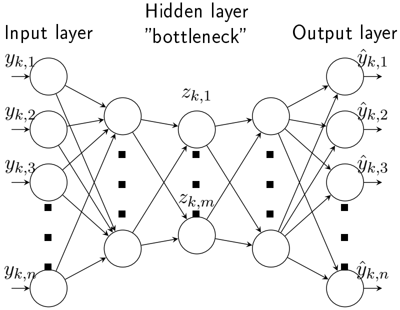
- **Input:** Torque of a pendulum
- **Output:** Pixel values of an 11×11 image



The Autoencoder

Notation:

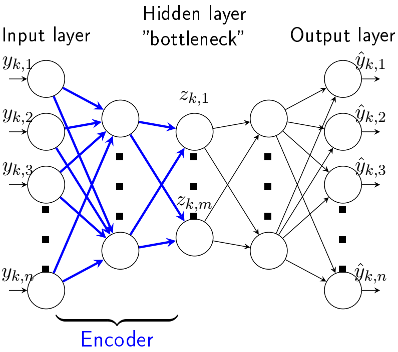
- $\mathbf{y}_k$ - High-dim. observations
- $\mathbf{z}_k$ - Low-dim. features



The Autoencoder

Notation:

- $\mathbf{y}_k$ - High-dim. observations
- $\mathbf{z}_k$ - Low-dim. features



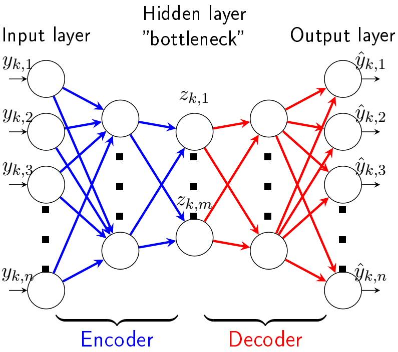
Model components:

1. Encoder: $\mathbf{z}_k = \mathbf{g}^{-1}(\mathbf{y}_k; \boldsymbol{\theta}_\mathsf{E})$

The Autoencoder

Notation:

- $\mathbf{y}_k$ - High-dim. observations
- $\mathbf{z}_k$ - Low-dim. features



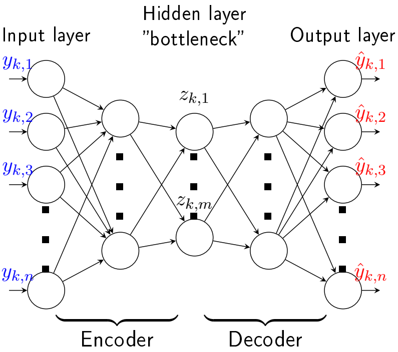
Model components:

1. Encoder: $\mathbf{z}_k = \mathbf{g}^{-1}(\mathbf{y}_k; \boldsymbol{\theta}_\mathsf{E})$
2. Decoder: $\widehat{\mathbf{y}}_k^\mathsf{R} = \mathbf{g}(\mathbf{z}_k; \boldsymbol{\theta}_\mathsf{D})$

The Autoencoder

Notation:

- $\mathbf{y}_k$ - High-dim. observations
- $\mathbf{z}_k$ - Low-dim. features



Model components:

1. Encoder: $\mathbf{z}_k = \mathbf{g}^{-1}(\mathbf{y}_k; \boldsymbol{\theta}_\mathsf{E})$
2. Decoder: $\widehat{\mathbf{y}}_k^\mathsf{R} = \mathbf{g}(\mathbf{z}_k; \boldsymbol{\theta}_\mathsf{D})$

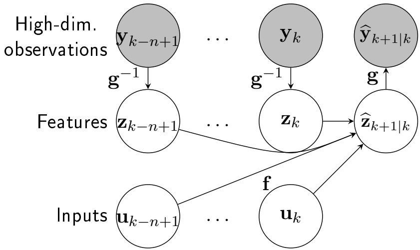
Reconstruction error:

$$V_\mathsf{R}(\boldsymbol{\theta}_\mathsf{E}, \boldsymbol{\theta}_\mathsf{D}) = \sum_{k=1}^{N} \|\mathbf{y}_k - \widehat{\mathbf{y}}_k^\mathsf{R}(\boldsymbol{\theta}_\mathsf{E}, \boldsymbol{\theta}_\mathsf{D})\|^2$$

Deep Dynamical Model

Notation:

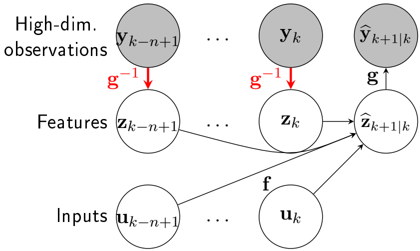
- $\mathbf{y}_k$ - High-dim. observations
- $\mathbf{z}_k$ - Low-dim. features
- $\mathbf{u}_k$ - Inputs



Deep Dynamical Model

Notation:

- $\mathbf{y}_k$ - High-dim. observations
- $\mathbf{z}_k$ - Low-dim. features
- $\mathbf{u}_k$ - Inputs



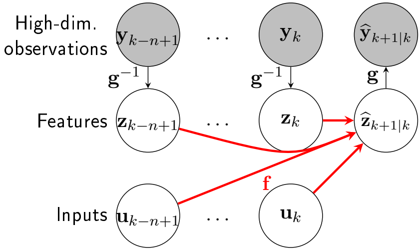
Model components:

1. Encoder: $\mathbf{z}_k = \mathbf{g}^{-1}(\mathbf{y}_k; \boldsymbol{\theta}_\mathsf{E})$

Deep Dynamical Model

Notation:

- $\mathbf{y}_k$ - High-dim. observations
- $\mathbf{z}_k$ - Low-dim. features
- $\mathbf{u}_k$ - Inputs



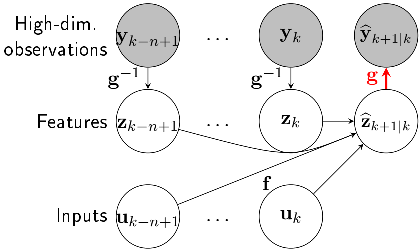
Model components:

1. Encoder: $\mathbf{z}_k = \mathbf{g}^{-1}(\mathbf{y}_k; \boldsymbol{\theta}_\mathsf{E})$
2. Prediction model: $\widehat{\mathbf{z}}_{k+1|k} = \mathbf{f}(\mathbf{z}_k, \mathbf{u}_k, \ldots, \mathbf{z}_{k-n+1}, \mathbf{u}_{k-n+1}; \boldsymbol{\theta}_\mathsf{P})$

Deep Dynamical Model

Notation:

- $\mathbf{y}_k$ - High-dim. observations
- $\mathbf{z}_k$ - Low-dim. features
- $\mathbf{u}_k$ - Inputs



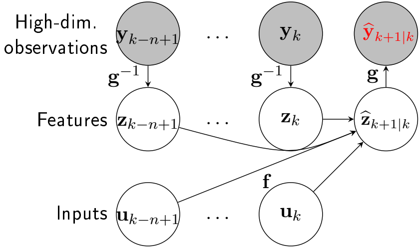
Model components:

1. Encoder: $\mathbf{z}_k = \mathbf{g}^{-1}(\mathbf{y}_k; \boldsymbol{\theta}_\mathsf{E})$
2. Prediction model: $\widehat{\mathbf{z}}_{k+1|k} = \mathbf{f}(\mathbf{z}_k, \mathbf{u}_k, \dots, \mathbf{z}_{k-n+1}, \mathbf{u}_{k-n+1}; \boldsymbol{\theta}_\mathsf{P})$
3. Decoder: $\widehat{\mathbf{y}}^\mathsf{P}_{k+1|k} = \mathbf{g}(\widehat{\mathbf{z}}_{k+1|k}; \boldsymbol{\theta}_\mathsf{D})$

Deep Dynamical Model

Notation:

- $\mathbf{y}_k$ - High-dim. observations
- $\mathbf{z}_k$ - Low-dim. features
- $\mathbf{u}_k$ - Inputs



Model components:

1. Encoder: $\mathbf{z}_k = \mathbf{g}^{-1}(\mathbf{y}_k; \boldsymbol{\theta}_\mathsf{E})$
2. Prediction model: $\widehat{\mathbf{z}}_{k+1|k} = \mathbf{f}(\mathbf{z}_k, \mathbf{u}_k, \dots, \mathbf{z}_{k-n+1}, \mathbf{u}_{k-n+1}; \boldsymbol{\theta}_\mathsf{P})$
3. Decoder: $\widehat{\mathbf{y}}^\mathsf{P}_{k+1|k} = \mathbf{g}(\widehat{\mathbf{z}}_{k+1|k}; \boldsymbol{\theta}_\mathsf{D})$

Prediction error:

$$V_\mathsf{P}(\boldsymbol{\theta}_\mathsf{E}, \boldsymbol{\theta}_\mathsf{D}, \boldsymbol{\theta}_\mathsf{P}) = \sum_{k=n}^{N-1} \|\mathbf{y}_{k+1} - \widehat{\mathbf{y}}^\mathsf{P}_{k+1|k}(\boldsymbol{\theta}_\mathsf{E}, \boldsymbol{\theta}_\mathsf{D}, \boldsymbol{\theta}_\mathsf{P})\|^2$$

Training

Key ingredient: The reconstruction error and the prediction error are minimized *simultaneously*!

$$(\widehat{\boldsymbol{\theta}}_{\mathsf{E}}, \widehat{\boldsymbol{\theta}}_{\mathsf{D}}, \widehat{\boldsymbol{\theta}}_{\mathsf{P}}) = \underset{\boldsymbol{\theta}_{\mathsf{E}}, \boldsymbol{\theta}_{\mathsf{D}}, \boldsymbol{\theta}_{\mathsf{P}}}{\arg\min}\; V_{\mathsf{R}}(\boldsymbol{\theta}_{\mathsf{E}}, \boldsymbol{\theta}_{\mathsf{D}}) + V_{\mathsf{P}}(\boldsymbol{\theta}_{\mathsf{E}}, \boldsymbol{\theta}_{\mathsf{D}}, \boldsymbol{\theta}_{\mathsf{P}})$$

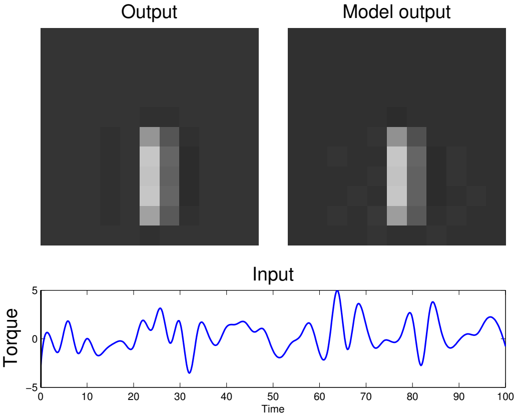
$$V_{\mathsf{R}}(\boldsymbol{\theta}_{\mathsf{E}}, \boldsymbol{\theta}_{\mathsf{D}}) = \sum_{k=1}^{N} \|\mathbf{y}_k - \widehat{\mathbf{y}}_k^{\mathsf{R}}(\boldsymbol{\theta}_{\mathsf{E}}, \boldsymbol{\theta}_{\mathsf{D}})\|^2,$$

$$V_{\mathsf{P}}(\boldsymbol{\theta}_{\mathsf{E}}, \boldsymbol{\theta}_{\mathsf{D}}, \boldsymbol{\theta}_{\mathsf{P}}) = \sum_{k=n}^{N-1} \|\mathbf{y}_{k+1} - \widehat{\mathbf{y}}_{k+1|k}^{\mathsf{P}}(\boldsymbol{\theta}_{\mathsf{E}}, \boldsymbol{\theta}_{\mathsf{D}}, \boldsymbol{\theta}_{\mathsf{P}})\|^2.$$

Results

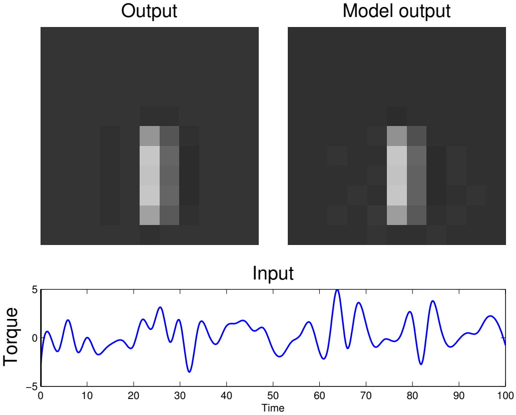
Experiment: Pendulum

- Layers in encoder/decoder: 4
- Latent dim.: $\dim(\mathbf{z}) = 1$
- Order of prediction model: $n = 4$



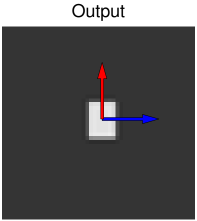
Experiment: Pendulum

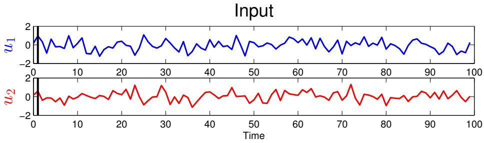
- Layers in encoder/decoder: 4
- Latent dim.: $\dim(\mathbf{z}) = 1$
- Order of prediction model: $n = 4$



Experiment: Agent in a Planar System

- **Input:** Offset in x−dir. (u_1) and y−dir. (u_2)
- **Output:** Pixel values of a 51×51 image
- **Latent dim.:** dim($\mathbf{z}$)=2

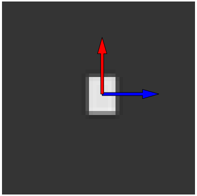




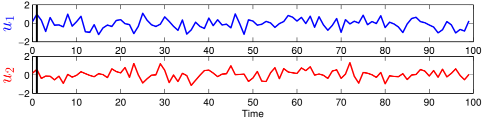
Experiment: Agent in a Planar System

- **Input:** Offset in x−dir. (u_1) and y−dir. (u_2)
- **Output:** Pixel values of a 51×51 image
- **Latent dim.:** dim($\mathbf{z}$)=2

Output

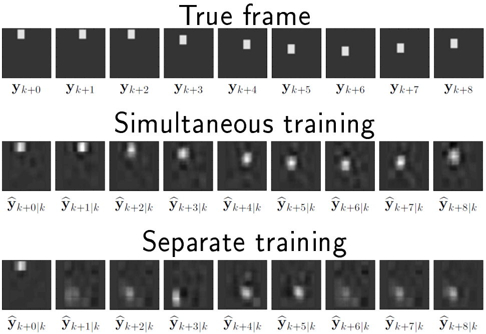


Input



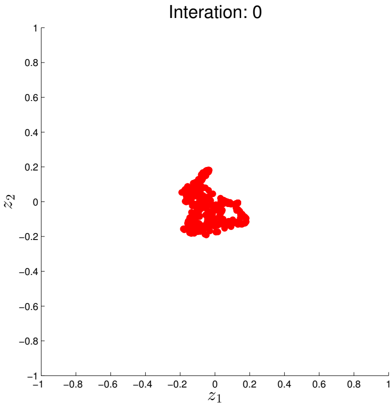
Experiment: Agent in a Planar System

Separate vs. Simultaneous Training

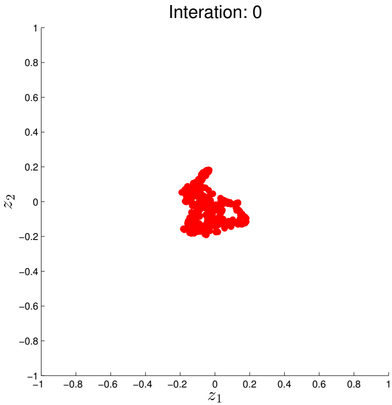


Experiment: Agent in a Planar System

Simultaneous Training

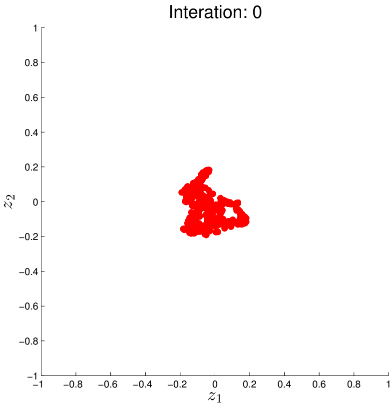


Separate Training

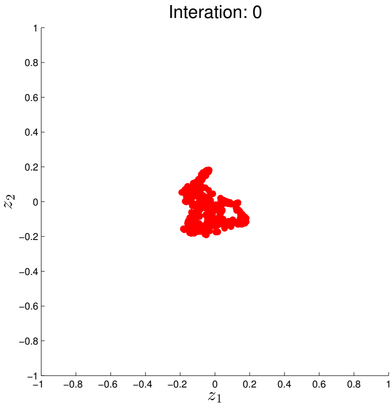


Experiment: Agent in a Planar System

Simultaneous Training

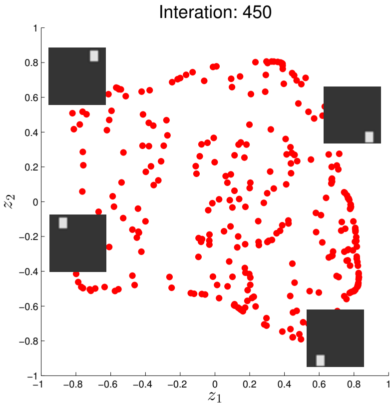


Separate Training

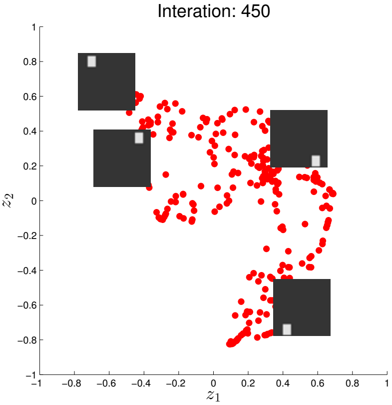


Experiment: Agent in a Planar System

Simultaneous Training

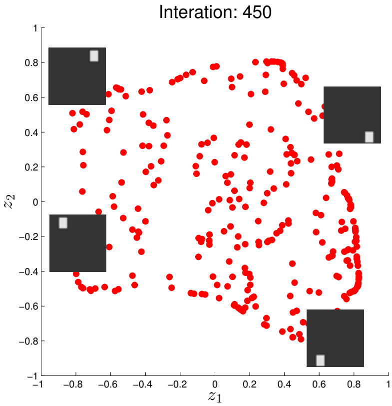


Separate Training

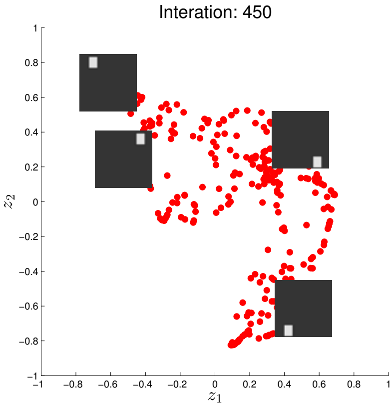


Experiment: Agent in a Planar System

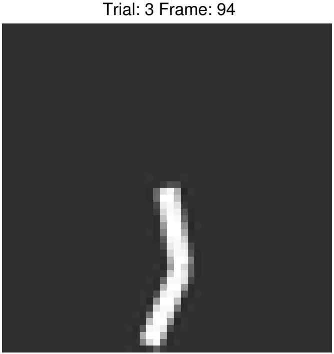
Simultaneous Training



Separate Training



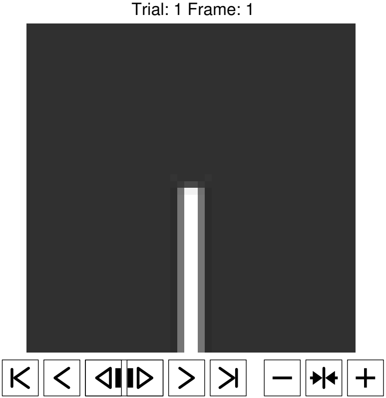
Application: Control of Two-Link Arm from Pixels Only



- Model predictive control
- Ref. value: $\mathbf{z}_{\text{ref}} = \mathbf{g}^{-1}(\mathbf{y}_{\text{ref}})$
- Model iteratively improved

J.-A. M. Assael, N. Wahlström, T. B. Schön, and M. P. Deisenroth.
Data-Efficient Learning of Feedback Policies from Image Pixels using Deep Dynamical Models.
Sept. 2015, ArXiv e-prints 1510.02173

Application: Control of Two-Link Arm from Pixels Only



- Model predictive control
- Ref. value: $\mathbf{z}_{\text{ref}} = \mathbf{g}^{-1}(\mathbf{y}_{\text{ref}})$
- Model iteratively improved

J.-A. M. Assael, N. Wahlström, T. B. Schön, and M. P. Deisenroth.
Data-Efficient Learning of Feedback Policies from Image Pixels using Deep Dynamical Models.
Sept. 2015, ArXiv e-prints 1510.02173

Conclusions

- Model for high-dimensional pixel data
- Simultaneous training is crucial
- Application: Control based on pixel data only

Niklas Wahlström
niklas.wahlstrom@liu.se

www.liu.se

LINKÖPING
UNIVERSITY