Machine Learning in Scientific Computing

CECAM/CSM/IRTG SCHOOL 2018



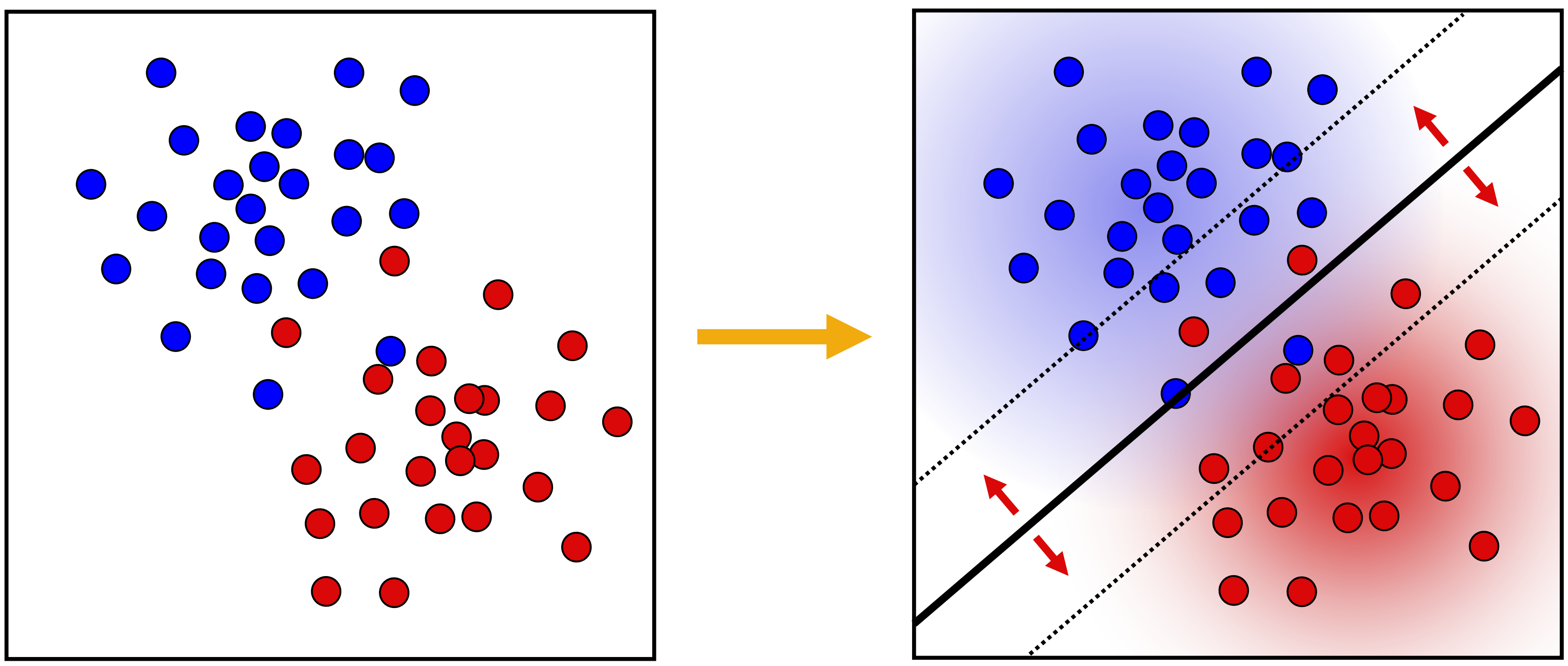
Lecture 3.2

Deep Learning

Michael Wand · Institut für Informatik, JGU Mainz · michael.wand@uni-mainz.de

Neural Network Building Blocks

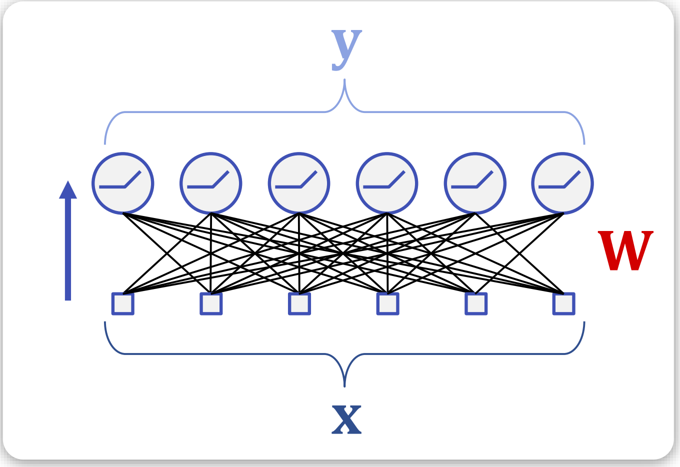
Support Vector Machines



training set

separating hyperplane, minimal penetration of margin (L_1)

Architectural Building Blocks



Fully connected network layer

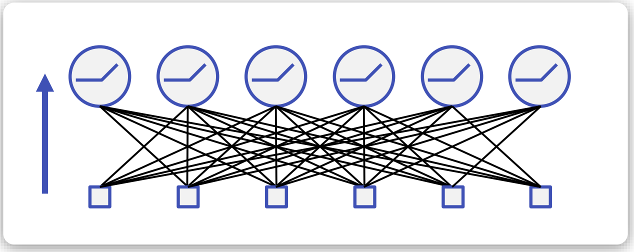
$$layer: \mathbb{R}^n \rightarrow \mathbb{R}^m$$

$$\mathbf{y} = layer(\mathbf{x}) = nonLinearity(\mathbf{W}\mathbf{x})$$

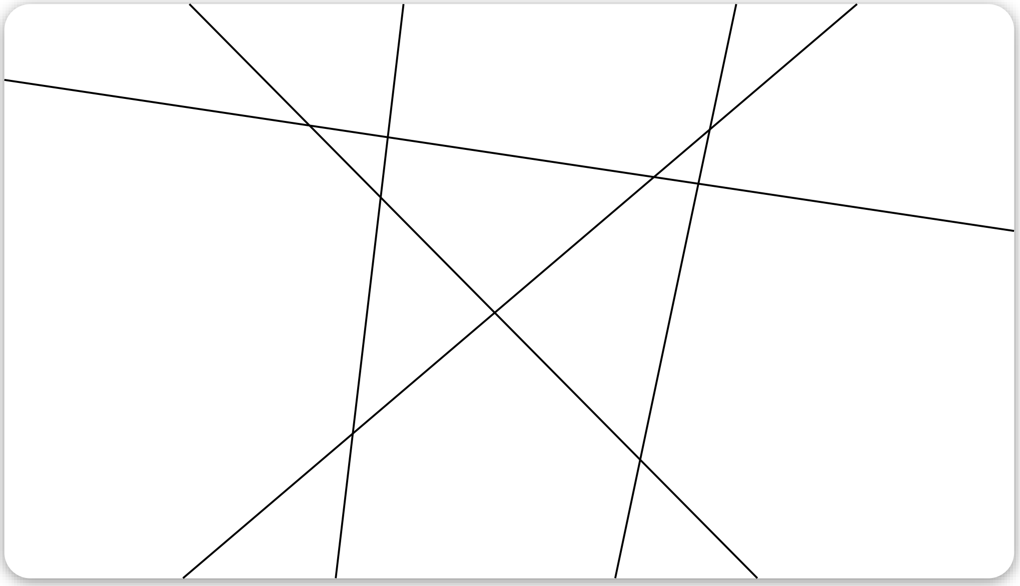
$$y_j = nonLinearity\left(\sum_{i=1}^{n} w_{ij} x_i\right)$$

$$nonLinearity(y) = \begin{cases} \max(y, 0) \;\; (\text{"relu"}) \\ \tanh(y) \\ \quad \dots \end{cases}$$

Architectural Building Blocks

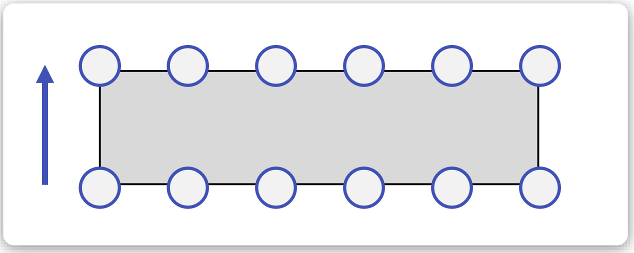


Fully connected network layer



Interpretation: ReLu-Layer = Arrangement of Hyperplanes

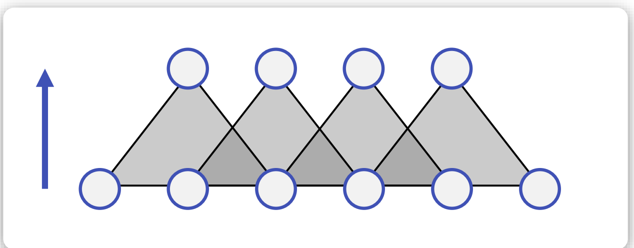
Architectural Building Blocks



Fully connected network layer

global connection / global dependencies

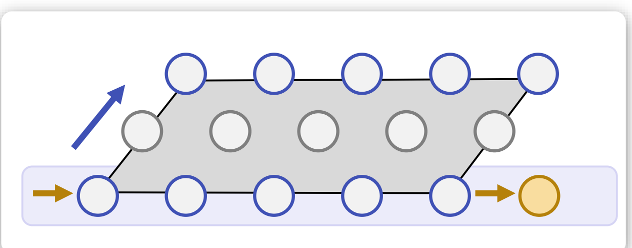
e.g: feature classification



Convolutional neural network

local connection / local correlations

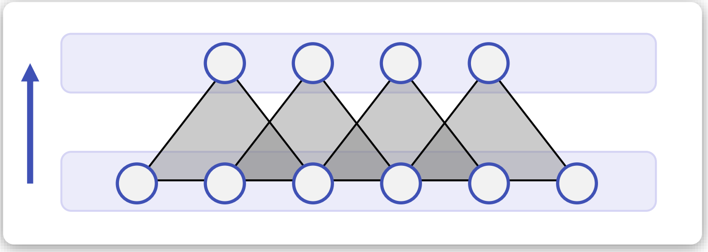
e.g.: image/audio/text data



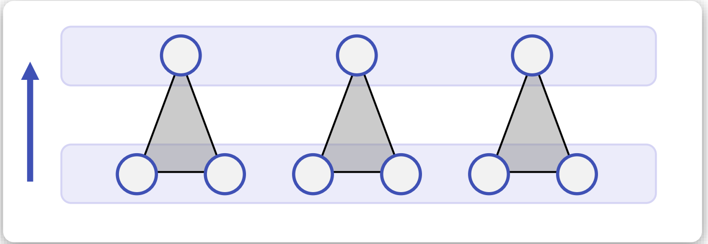
Recurrent neural networks

Markov-chain models with memory

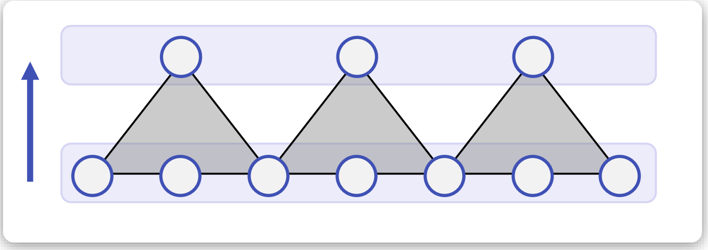
Convolutional Building Blocks



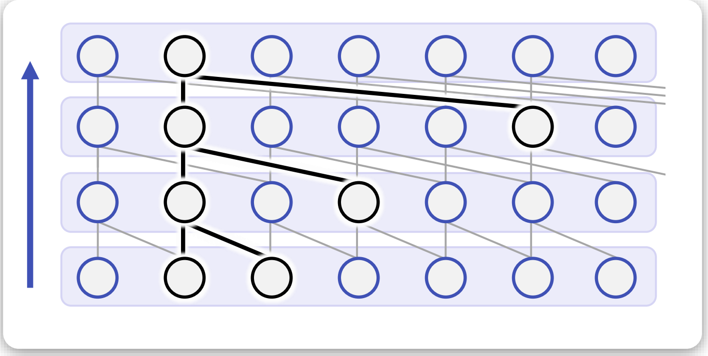
Convolutional neural network
local connection / local correlations



Pooling layer
reduce resolution (half, third, ...)

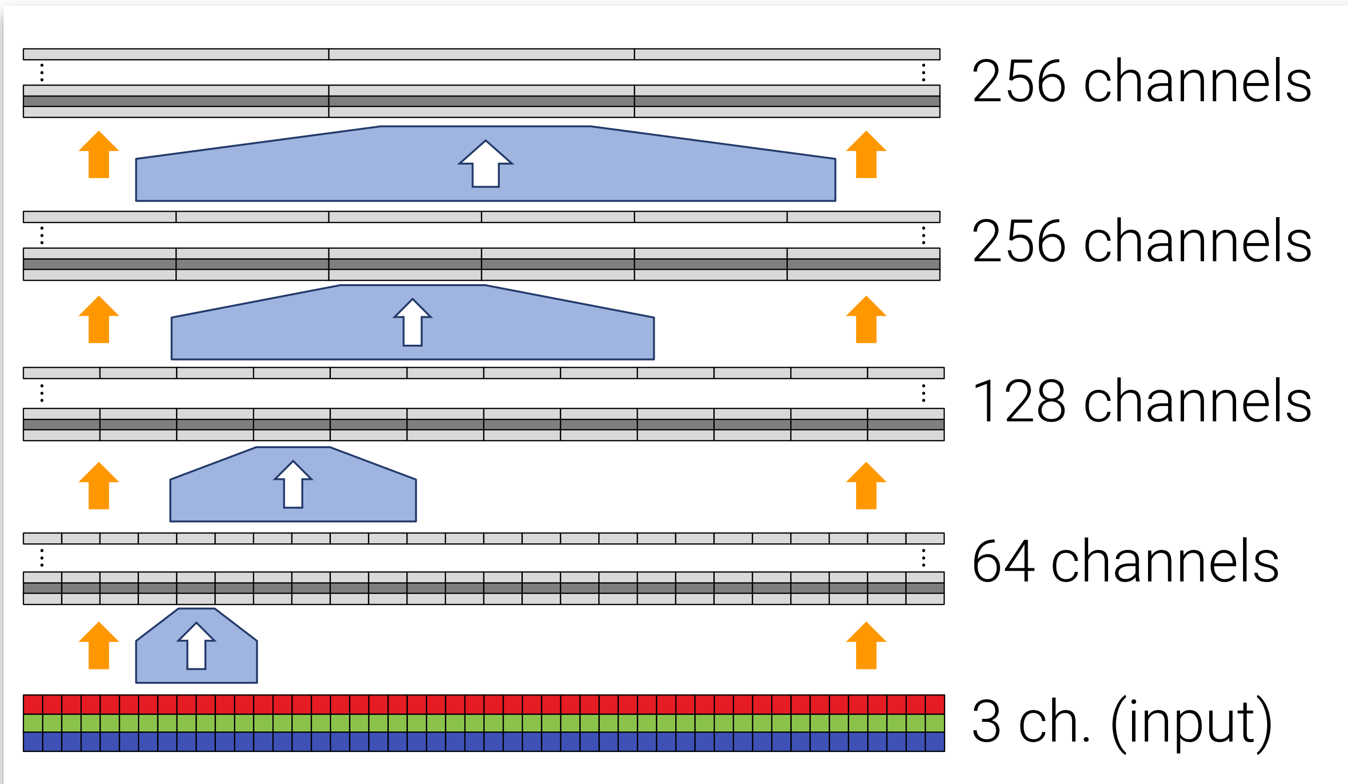


Convolution with stride
reduce resolution, learned filters



Dilated networks
aggregate context, same resolution

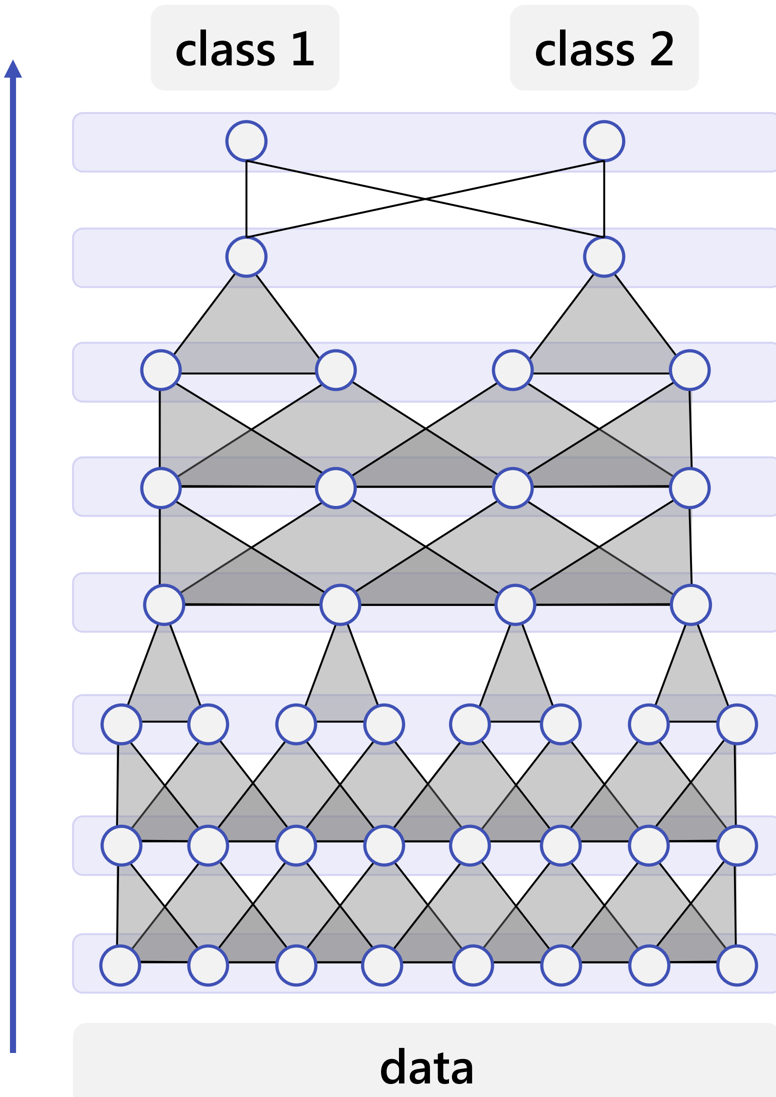
Typical ConvNet



Coarse Graining

(model correlations at different scales)

Image Classification



Fully Connected

Pooling / striding (typ. 2x2)

Convolution (typ. 3x3, residual)

Convolution (typ. 3x3, residual)

many layers

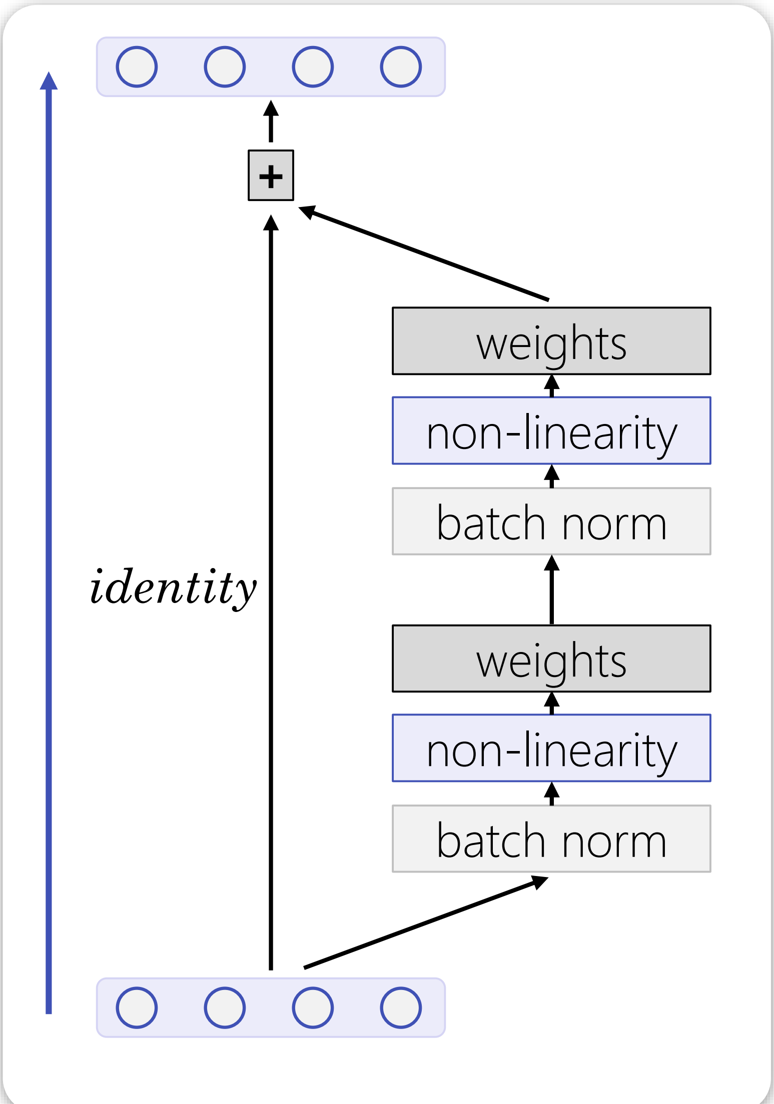
Pooling / striding (typ. 2x2)

Convolution (typ. 3x3, residual)

Convolution (typ. 3x3, residual)

many layers

Architectural Building Blocks

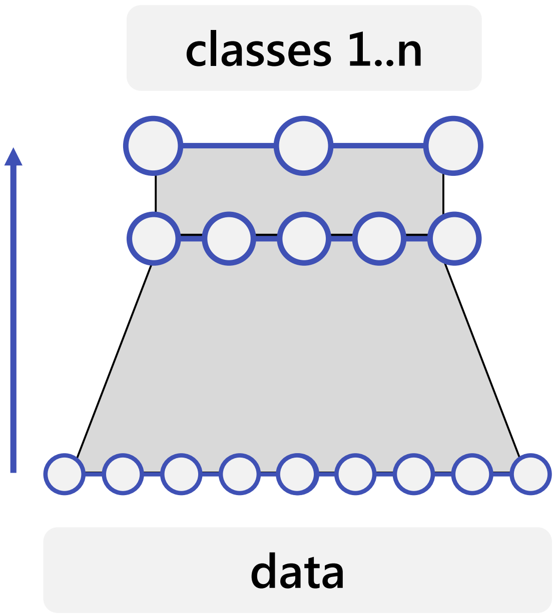


Residual networks

Allows very deep networks

Identity mapping as default

Image Classification



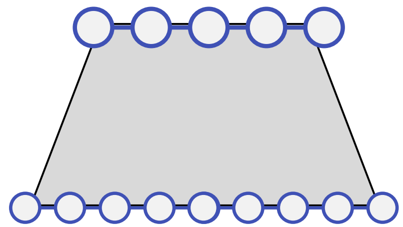
Fully connected
(typ. global av. pooling + 1 layer)

Convolution + pooling or striding
(typ. 20-100 layers)

Central Building Block: Regression

Trained with Examples

Outputs

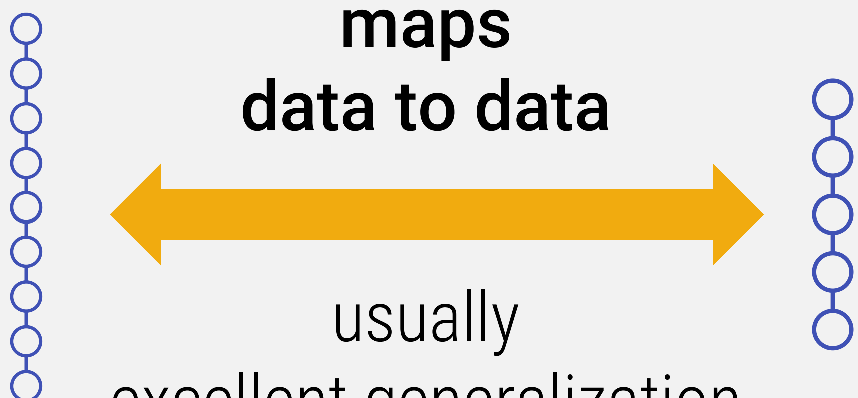


Inputs

Prediction $\in \mathbb{R}^m$

General Regressor

Data $\in \mathbb{R}^n$

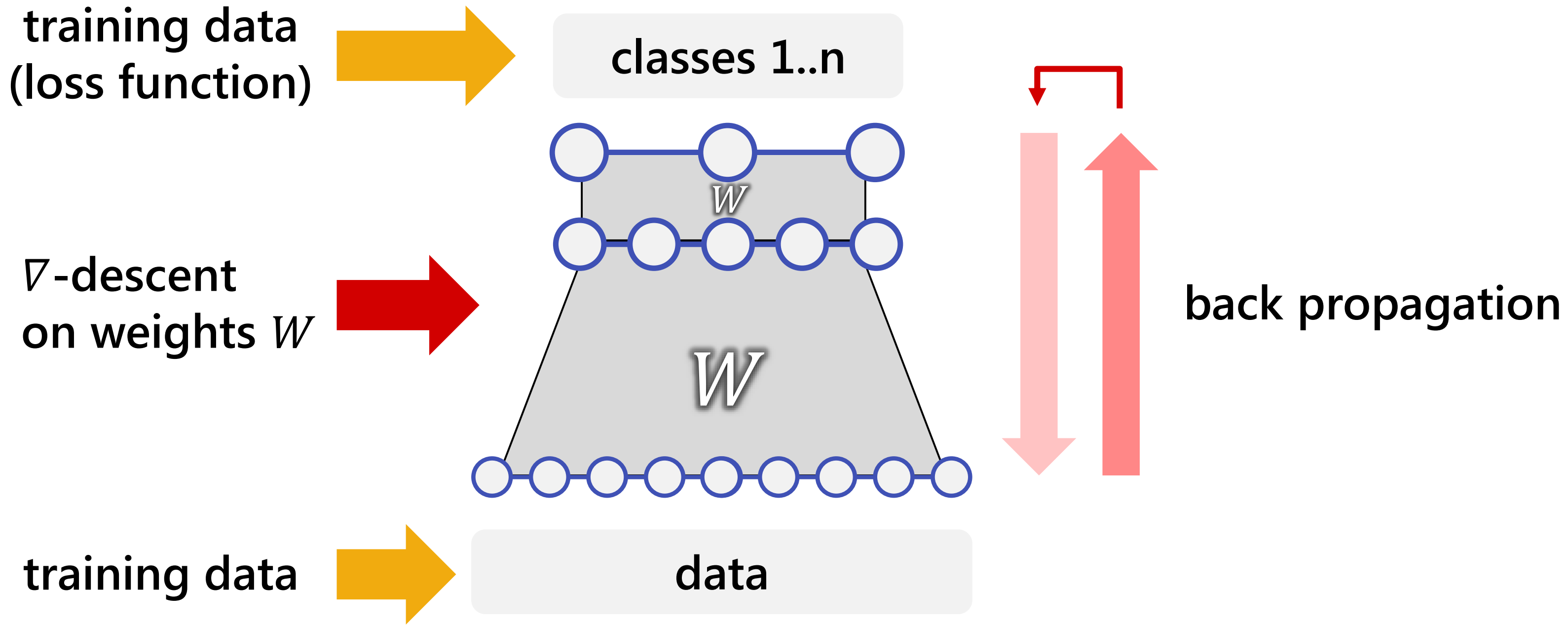


maps data to data

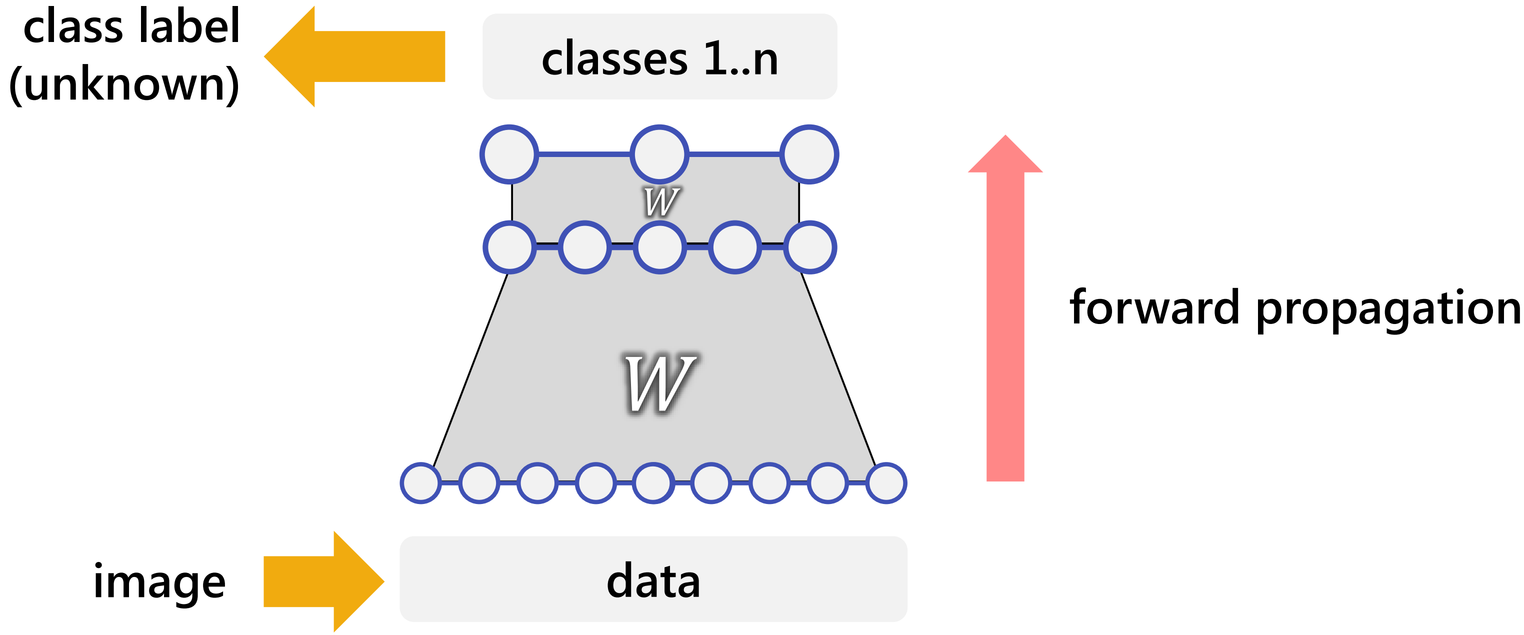
usually
excellent generalization
(*not clear why*)

Training & Inference

Discriminative Training



Inference


class label
(unknown)
classes 1..n
W
W
forward propagation
image
data

Loss Function

Notation

- Neural network f, weights W, input $\mathbf{x}$, output $\mathbf{y}$
- Supervised, training data: $(\mathbf{x}_i, \mathbf{y}_i)_{i=1..n}$
- $f_W(\mathbf{x}) = \mathbf{y}$

Different Loss Functions

Regression:

- Least squares $(f_W(\mathbf{x}_i) - \mathbf{y}_i)^2$

Classification

- One-Hot-Vectors $\mathbf{y}_i$
- Cross Entropy:

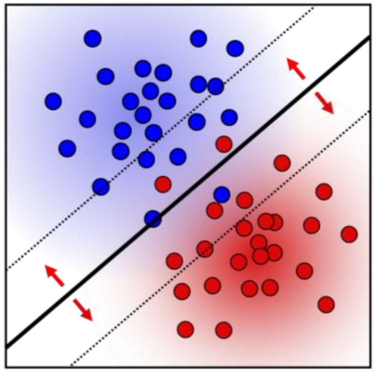
$$H\left(\text{softmax}\left(f_W(\mathbf{x}_i)\right), \mathbf{y}_i\right)$$

- Max-Margin:

$$\text{margin}(f_W(\mathbf{x}_i), \mathbf{y}_i)$$

Softmax:

$$\text{softmax}(\mathbf{y}) = \begin{pmatrix} \frac{e^{-y_1}}{\sum_{i=1}^{n} e^{-y_i}} \\ \vdots \\ \frac{e^{-y_n}}{\sum_{i=1}^{n} e^{-y_i}} \end{pmatrix}$$



Cross Entropy as Maximum-Likelihood

$$\arg\min_{W} KL\left(\mathbf{y}_i \parallel f_W(\mathbf{x}_i)\right)$$

Cross-Entropy Loss

$$= \arg\min_{W} \sum_{k=1}^{n_l} [\mathbf{y}_i]_k \log_2 \frac{[\mathbf{y}_i]_k}{[f_W(\mathbf{x}_i)]_k}$$

$$= \arg\min_{W} \Big(H\big(\mathbf{y}_i, f_W(\mathbf{x}_i)\big) - H(\mathbf{y}_i)\Big)$$

$$= \arg\min_{W} \Big(H\big(\mathbf{y}_i, f_W(\mathbf{x}_i)\big)\Big)$$

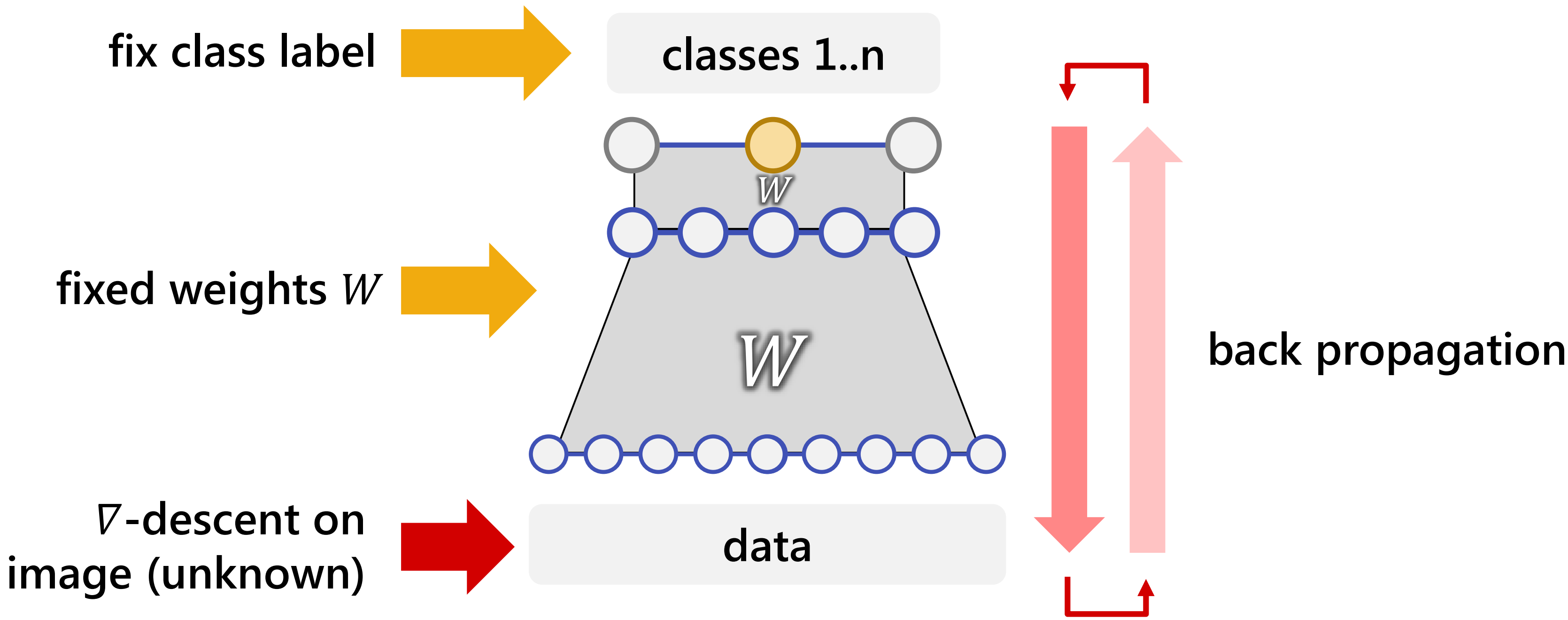
$$= \arg\min_{W} \sum_{k=1}^{n_l} [\mathbf{y}_i]_k \log_2 [f_W(\mathbf{x}_i)]_k$$

$$= \arg\min_{W} \log_2 [f_W(\mathbf{x}_i)]_{k=l_i}$$

Class likelihood (maximization)

Variational Inversion

Variational Inversion



Google‘s „Deep Dream (Inceptionism)“ Algorithms

Image: Daniel Strecker

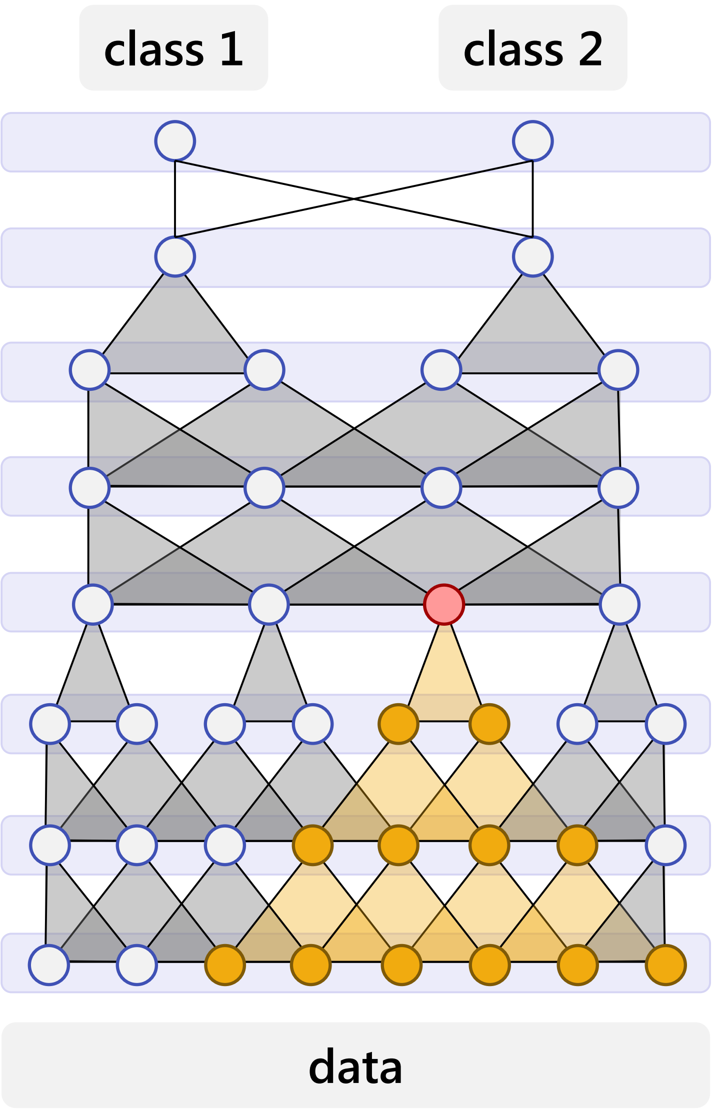
Google's „Deep Dream (Inceptionism)" Algorithms

Image: Daniel Strecker

Better Generative Image Models

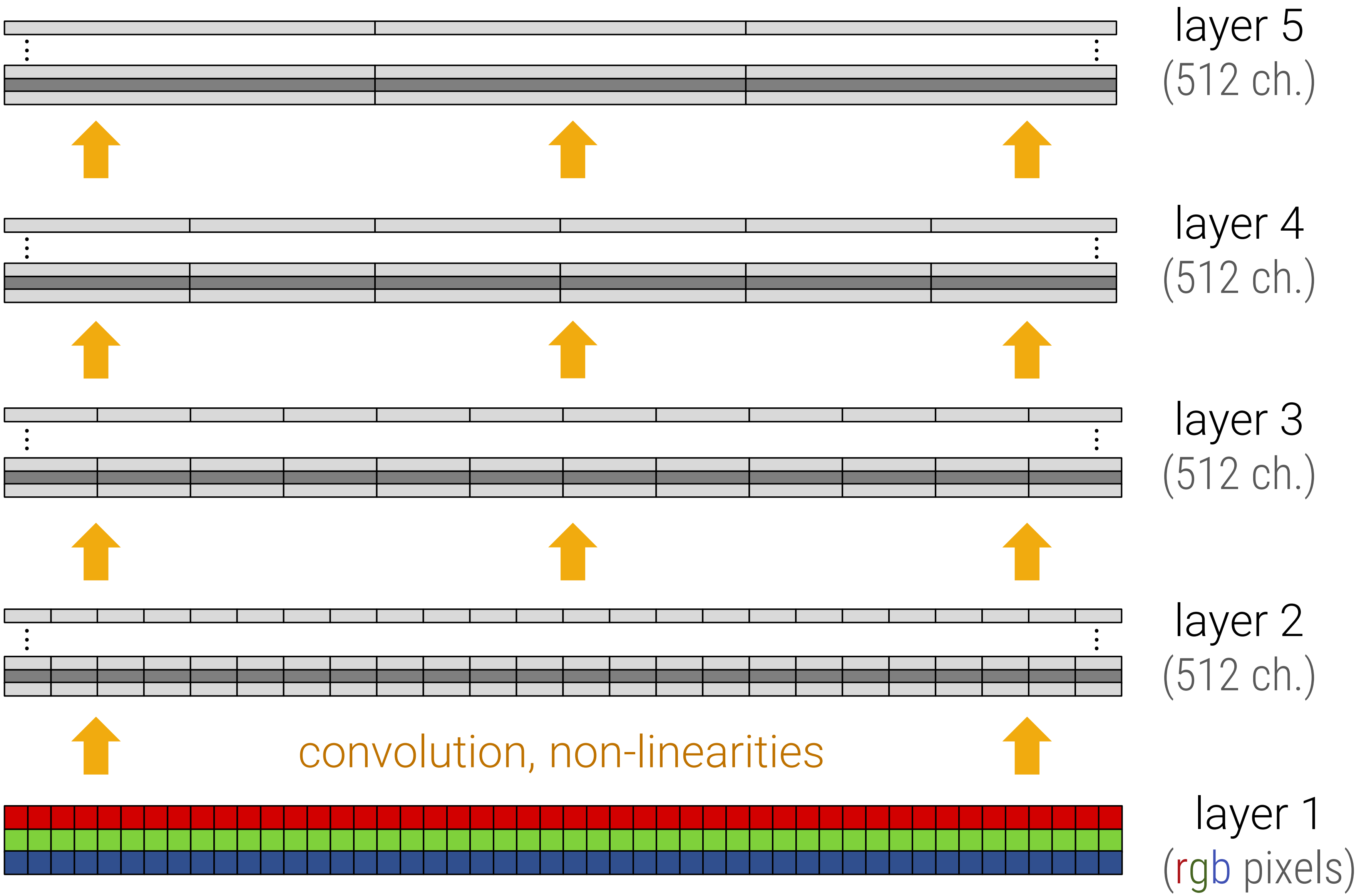
Convnets?

MRF Model is Build-In

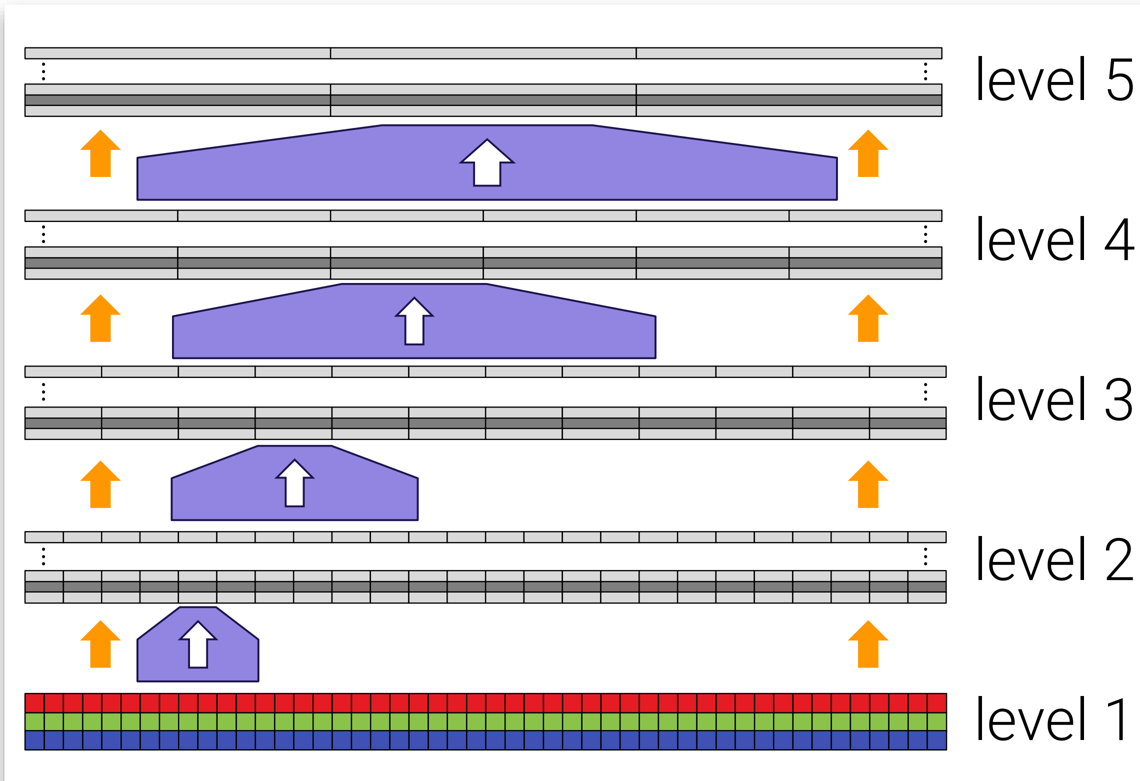


ConvNets are Markovian

VGG-Network (Discriminative!)



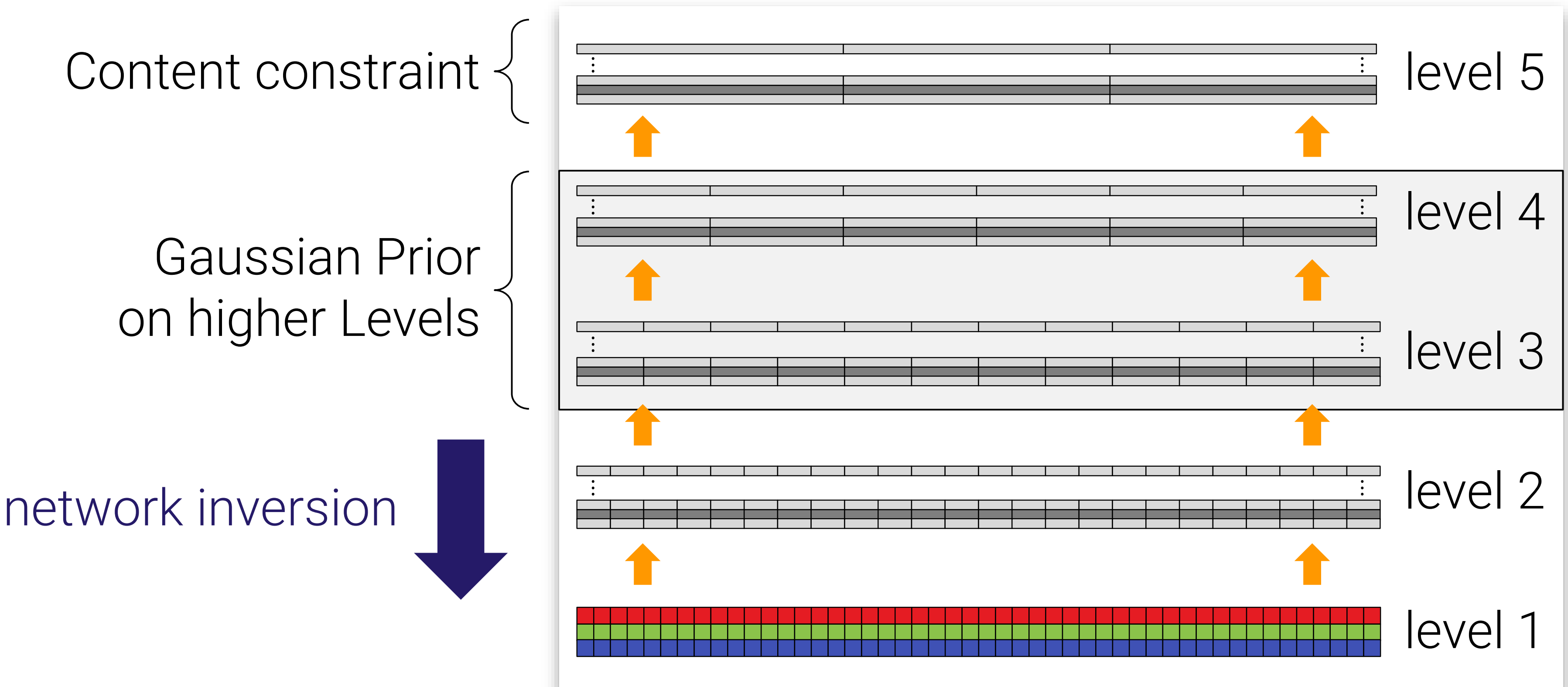
Discriminative Network



Coarse Graining

(model correlations at different scales)

High-Level Content Control



Picasso

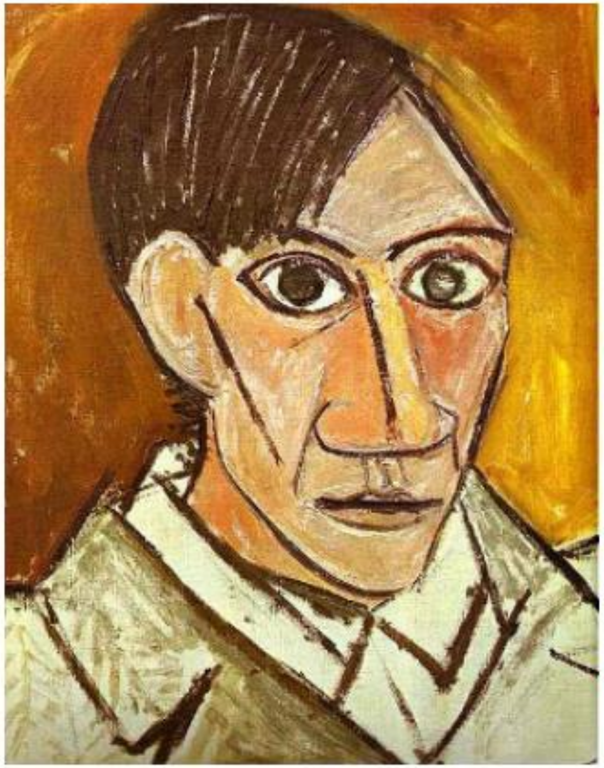
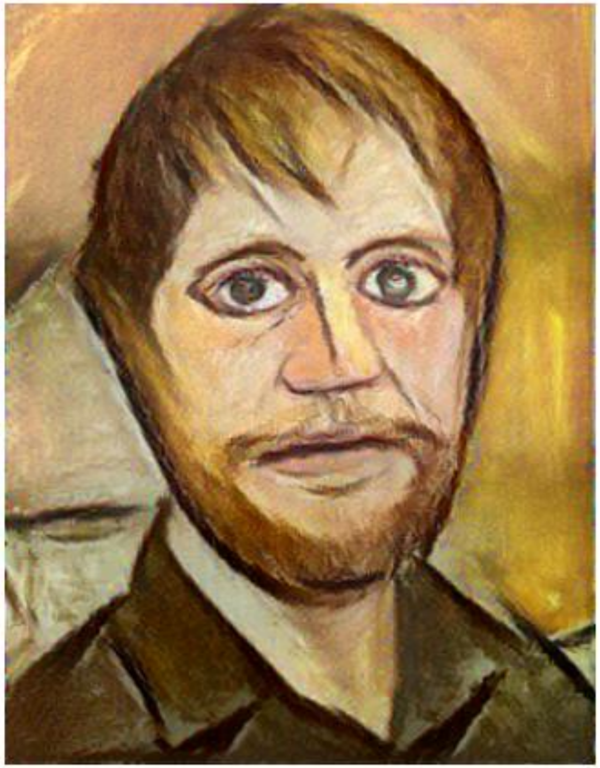
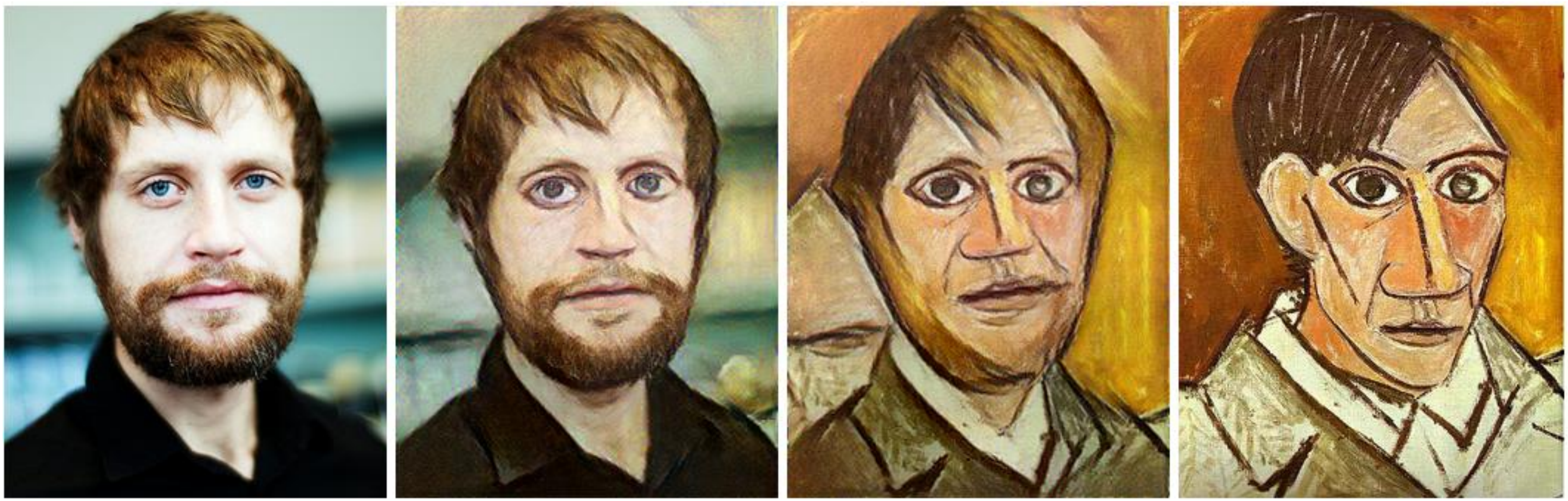




image sources: flickr user Christopher Michel, Pablo Picasso (self-portrait 1907)

Picasso



That was cute.
How do we get it right?

Learning generative models

Wishlist

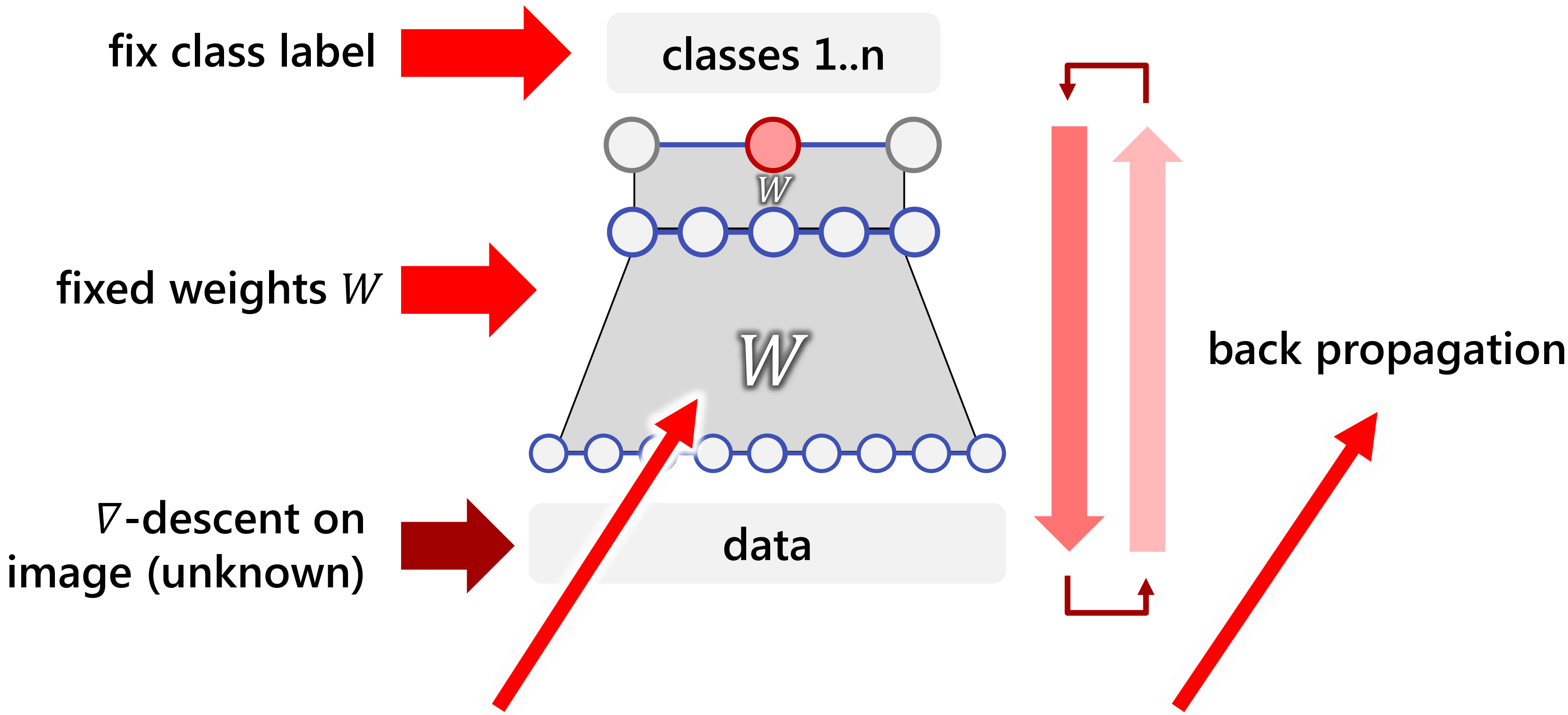
- Generalization
  - Broad object class rather than single image
  - “Faces”, “Dogs”, “Living Room Scenes”
- Explicit generative statistical model

Technical options

- (Variational) Autoencoders
- Generative adversarial networks

Generative Networks

So Far: Variational Inversion



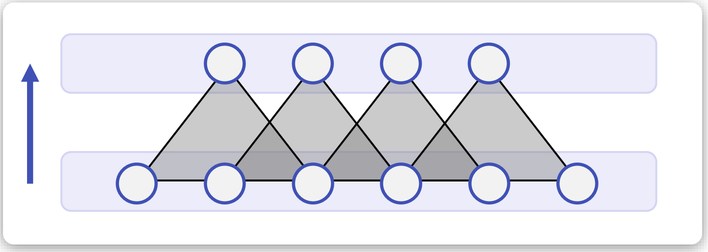
pretrained for image classification!

(why should we do this?)

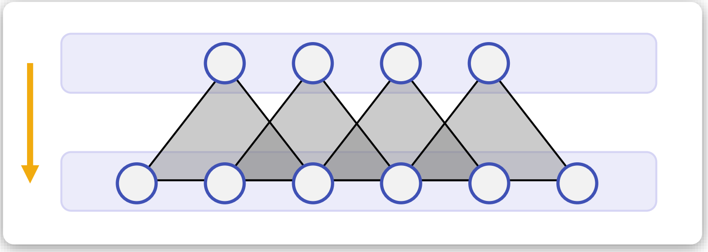
variational optimization very expensive

(~2min for 256^2 pix)

How to Create the Data?

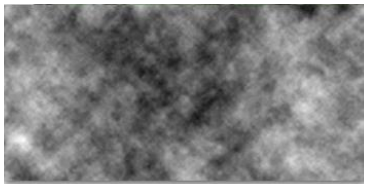


Convolutional network
Discriminative network

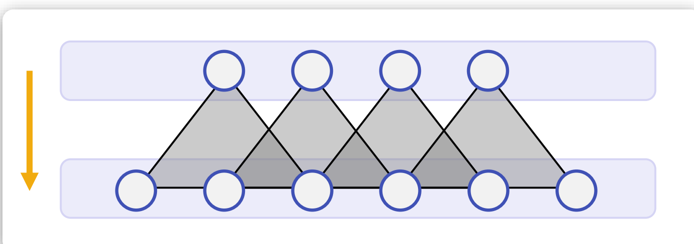


Convolutional network
Generative network

How to Create the Data?



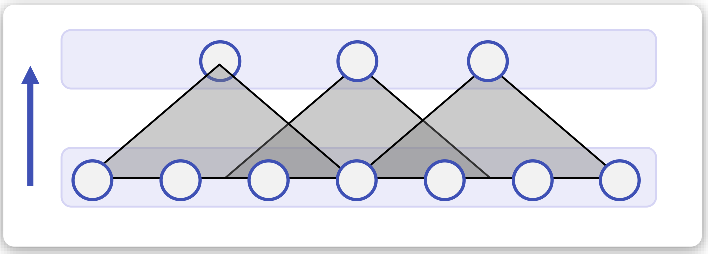
Input
(for example noise)



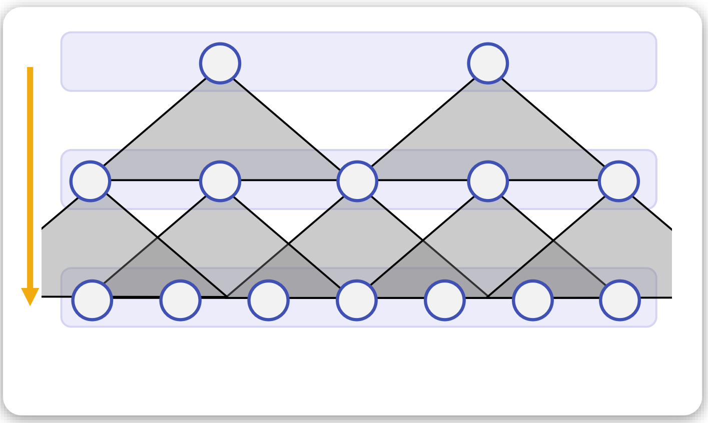
Convolutional network
Generative network

Synthesis
(after many layers...)

How to Create the Data?



Convolution with stride
(un)-pooling effect

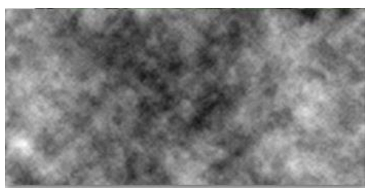


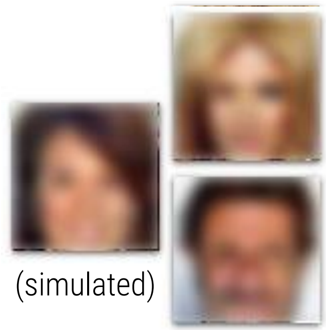
Fully convolution generator
many layers
some with stride

Great! How do we train it?

Autoencoder

Auto-Encoder: Non-linear PCA



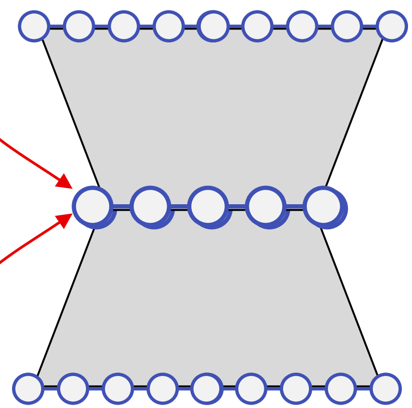


(simulated)

Need Entropy Source
(→ Variational Autoencoder)

Reconstruction $\in \mathbb{R}^n$

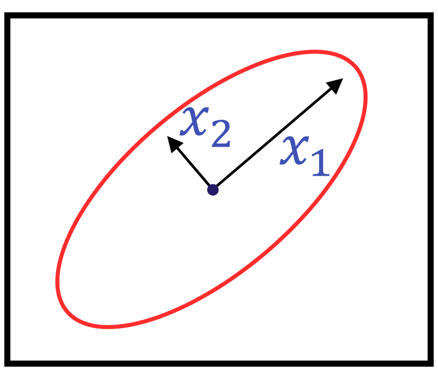
Latend, Coarse-Grained Representation $\in \mathbb{R}^m$



Auto-Encoder

Original Data $\in \mathbb{R}^n$

least squares loss



Training Data
(tons of it)

Nature: Correlated Noise

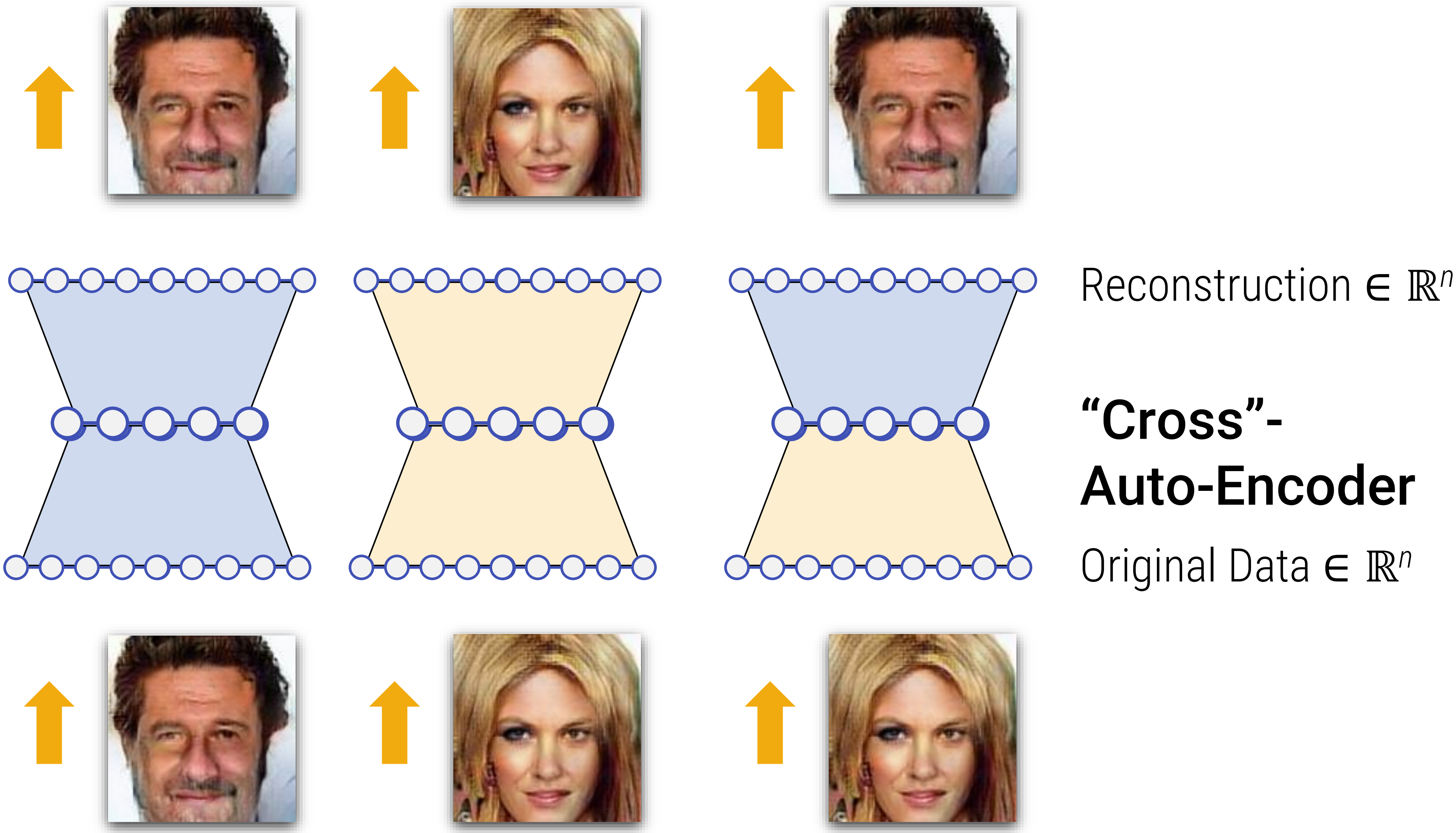
Samples

[CelebA data set]

Natural phenomena

- Most of the information (entropy) is noise
- Correlated noise, not independent pixel noise

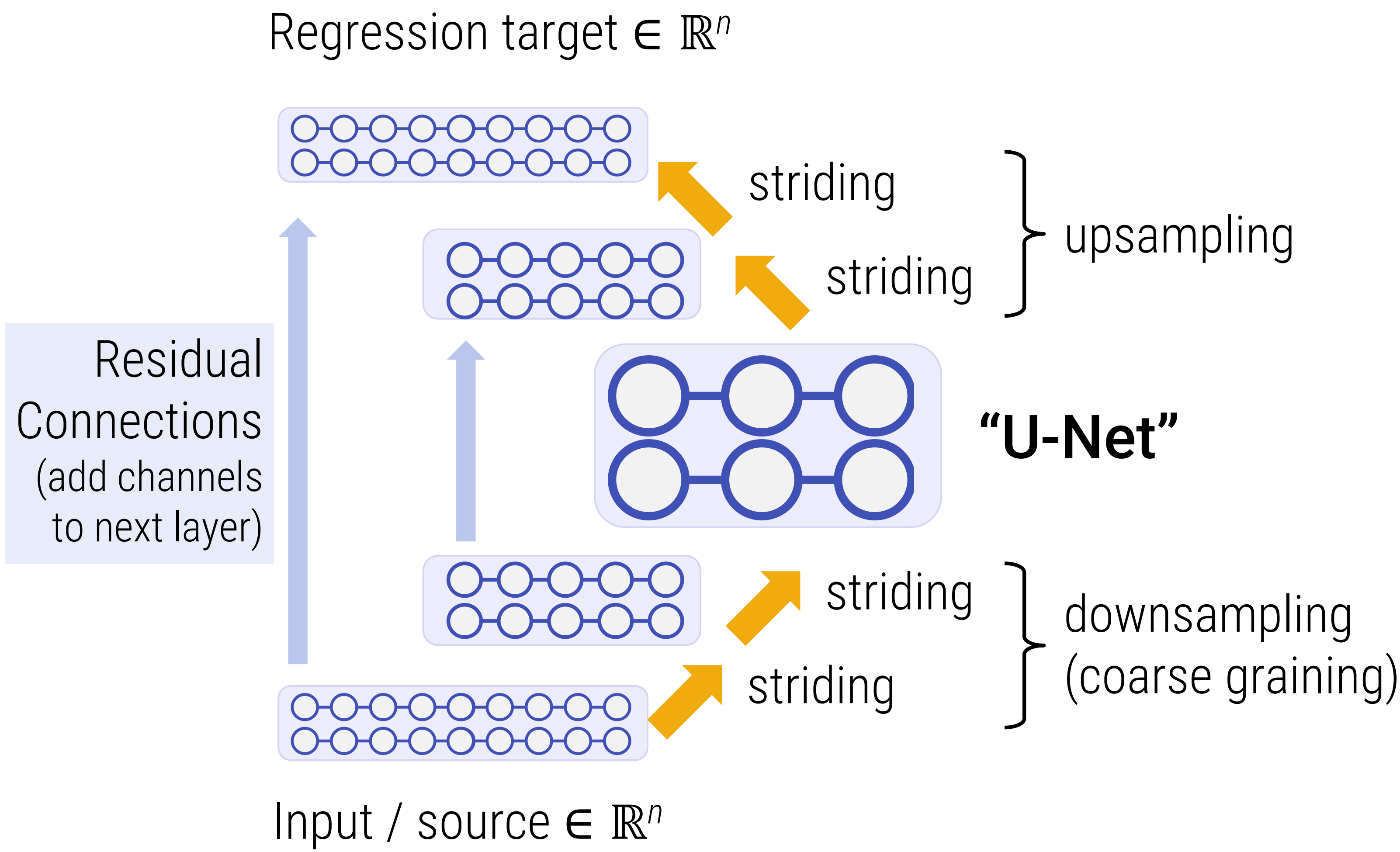
Cross Auto-Encoder



“Deep Fakes”

https://www.reddit.com/r/SFWdeepfakes/comments/7tcddy/nicolas_cage_deepfaked_into_superman/

U-Net

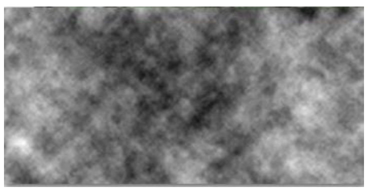


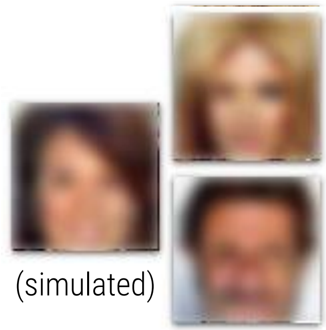
Regression target ∈ $\mathbb{R}^n$
striding
striding
upsampling
Residual Connections (add channels to next layer)
"U-Net"
striding
striding
downsampling (coarse graining)
Input / source ∈ $\mathbb{R}^n$

Example: Segmentation

https://towardsdatascience.com/training-road-scene-segmentation-on-cityscapes-with-supervisely-tensorflow-and-unet-1232314781a8

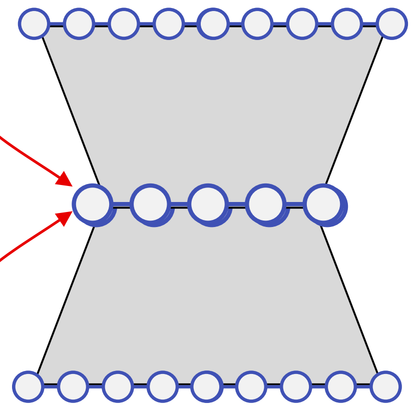
Auto-Encoder: Non-linear PCA





(simulated)

Need Entropy Source
(→ Variational Autoencoder)



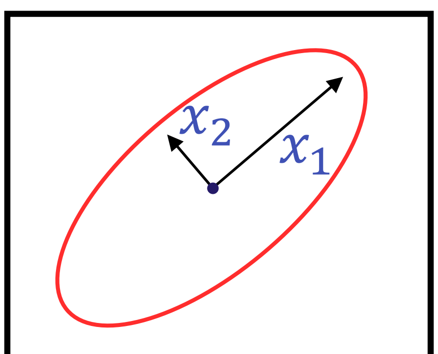
Reconstruction $\in \mathbb{R}^n$

Auto-Encoder

Original Data $\in \mathbb{R}^n$

Latend, Coarse-Grained Representation $\in \mathbb{R}^m$

least squares loss

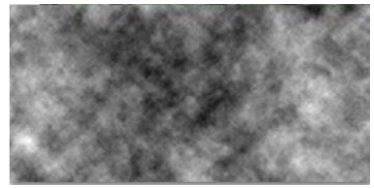


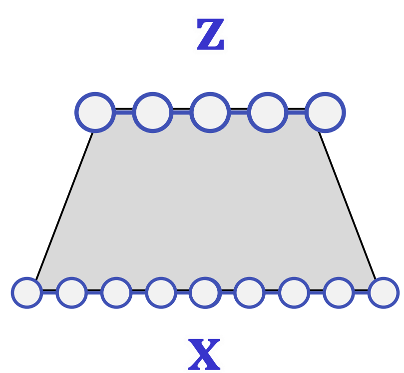
Training Data
(tons of it)

Probabilistic Generative Models

Learning Schemes

Gaussian Noise $\in \mathbb{R}^m$

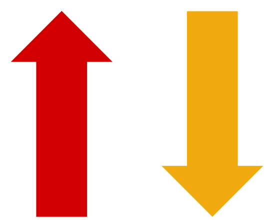




Generative Network

Original Data $\in \mathbb{R}^n$

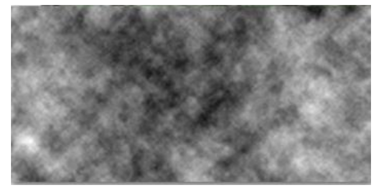
Training Data

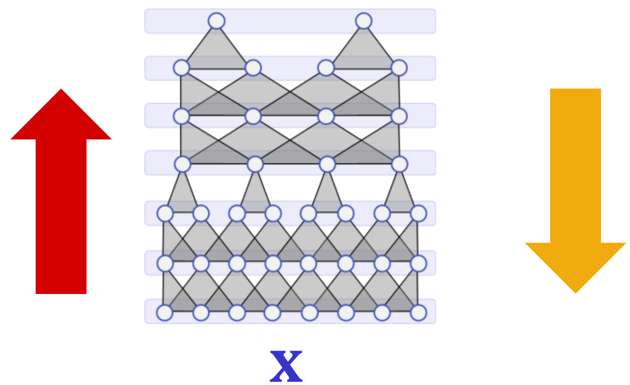


New Samples

Learning Schemes

Gaussian Noise $\in \mathbb{R}^m$





Original Data $\in \mathbb{R}^n$

Generative Network

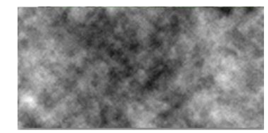
Training Data

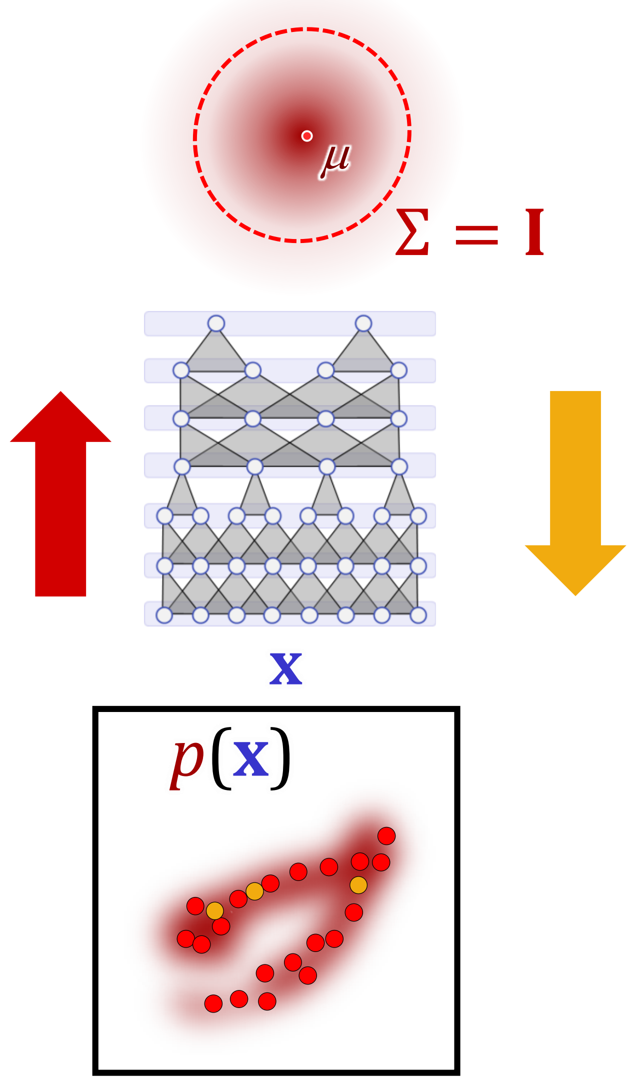
New Samples

Learning Schemes

Gaussian Noise $\in \mathbb{R}^m$

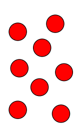
$N_{\mathbf{0},\mathbf{I}}(\mathbf{z})$

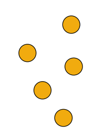




Generative Network

Original Data $\in \mathbb{R}^n$

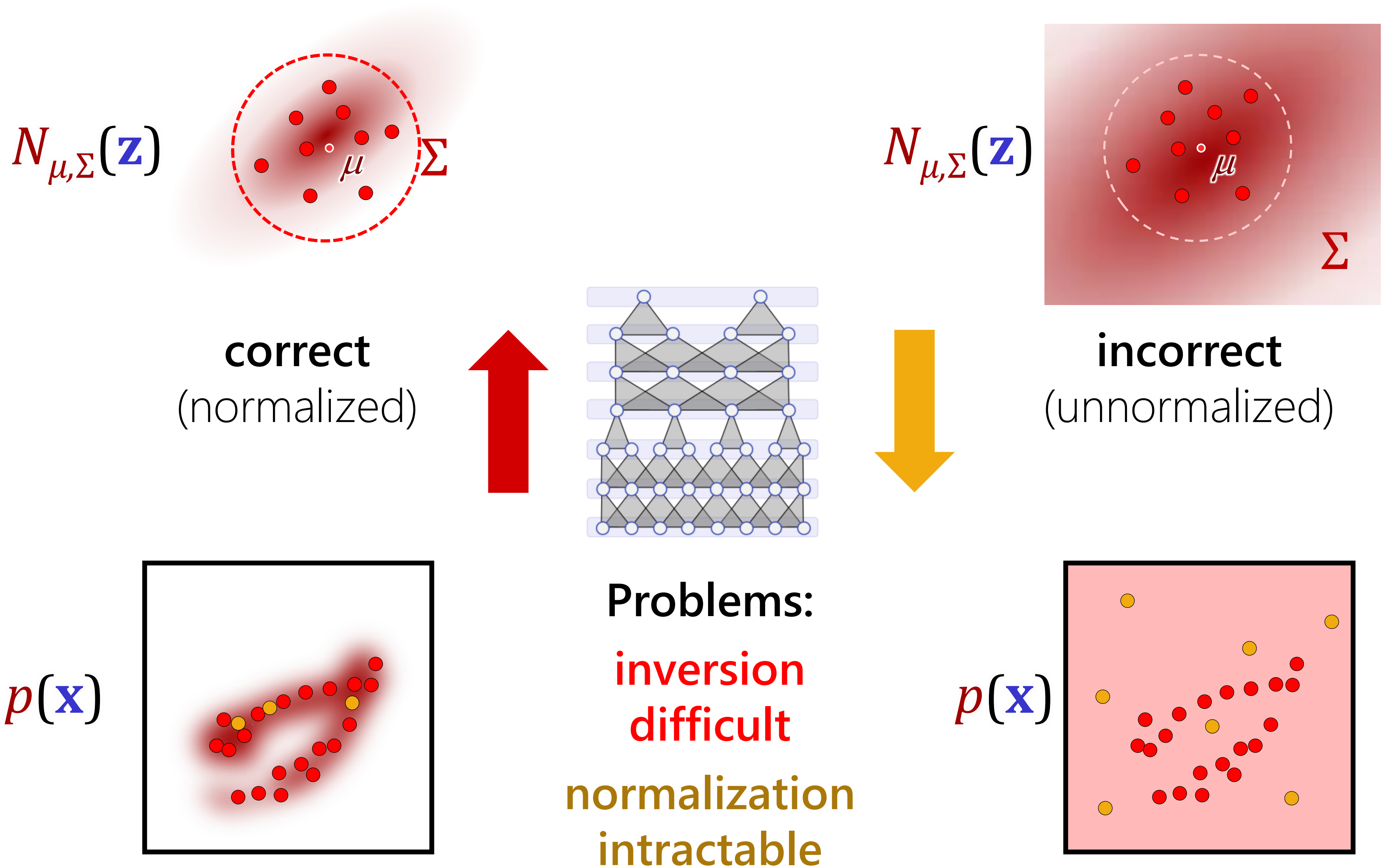




Training Data

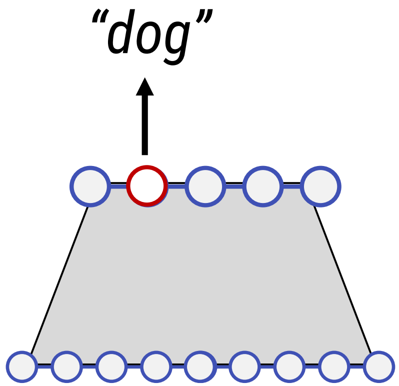
New Samples

Problem: Need Normalized Density!



Generative Adversarial Networks (GANs)

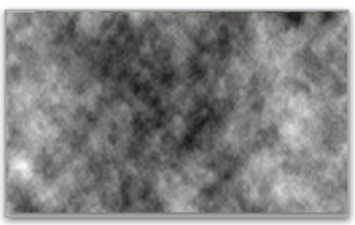
Learning Schemes



Labels, e.g. [cat?, dog?, bunny?] $\in [0,1]^m$

Discriminator

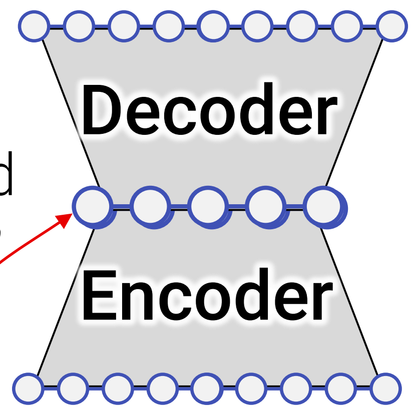
Data $\in \mathbb{R}^n$



Learning Schemes

(blur simulated)

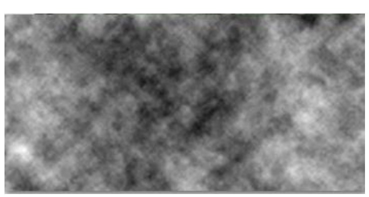
Reconstructed Data ∈ $\mathbb{R}^n$



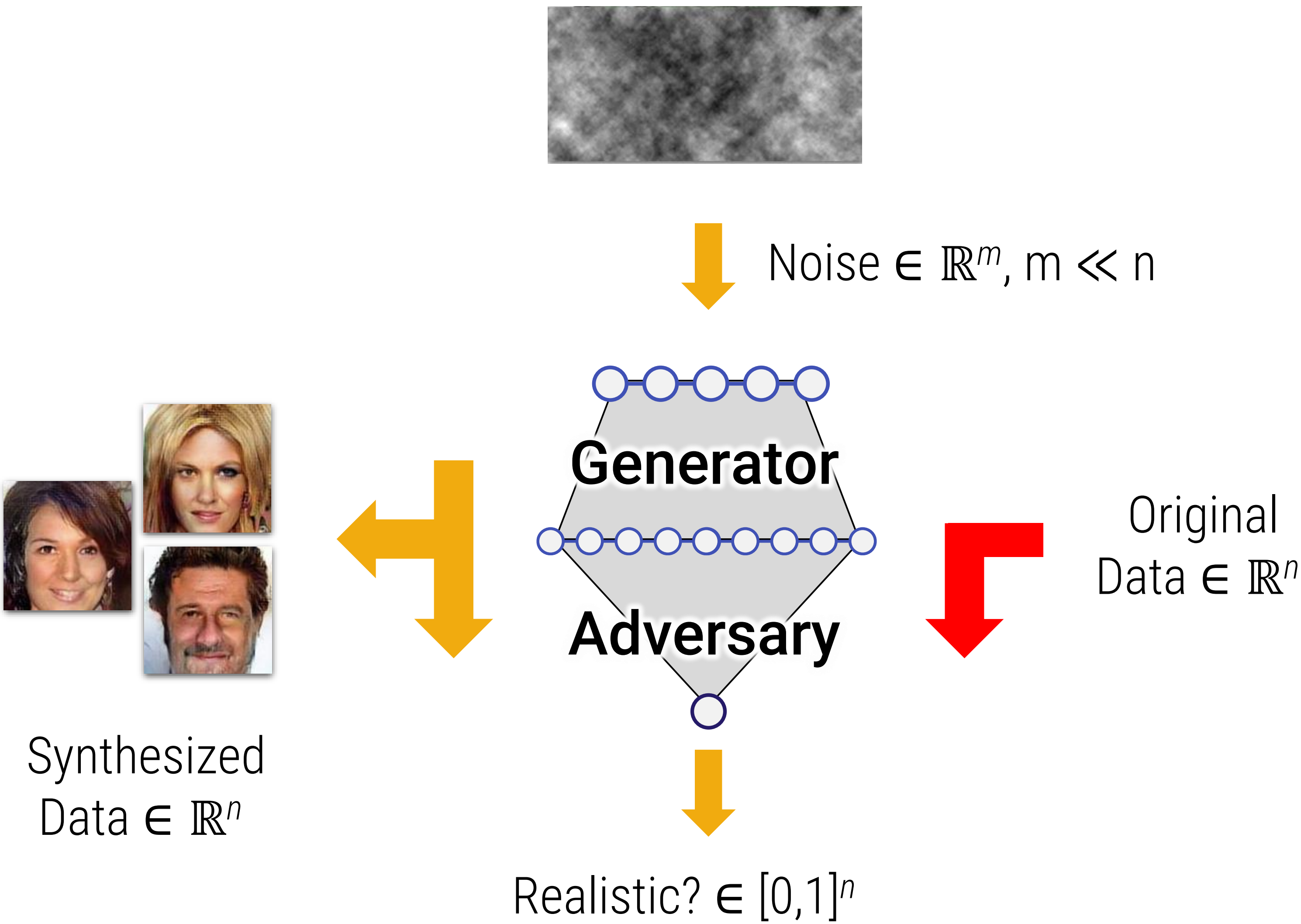
Latend, coarse-grained Representation ∈ $\mathbb{R}^m$

Auto-Encoder (Generative Model)

Original Data ∈ $\mathbb{R}^n$



Learning Schemes



[Chuan Li / CelebA Data]

Optimization

Training

- Discriminator tries to distinguish real / fake
  - Maximize prediction accuracy
- Generator tries to fool discriminator
  - Minimizes prediction accuracy
- Minimax game
- Nash equilibrium at true distribution

Issues

- Gradient descent instead of min-max
- Training unstable

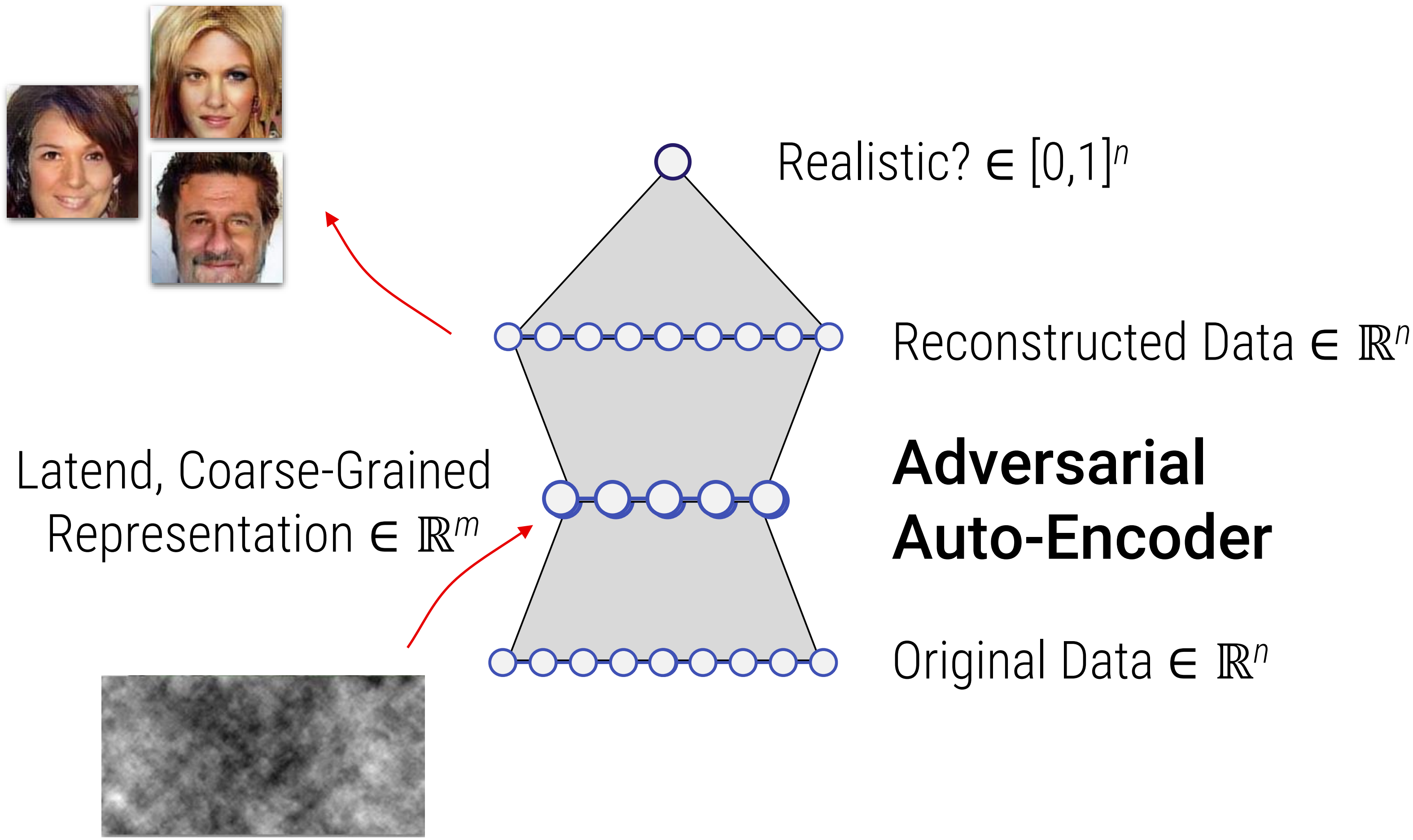
Mode Collapse Problem

Good result
(W-DCGAN)

Mode collapse
(Std. GAN)

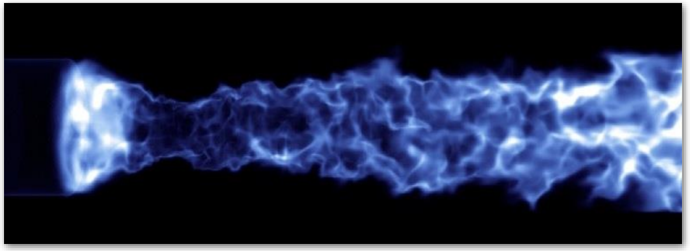
Results from [Arjovsky et al. 2017]

Simple Solution

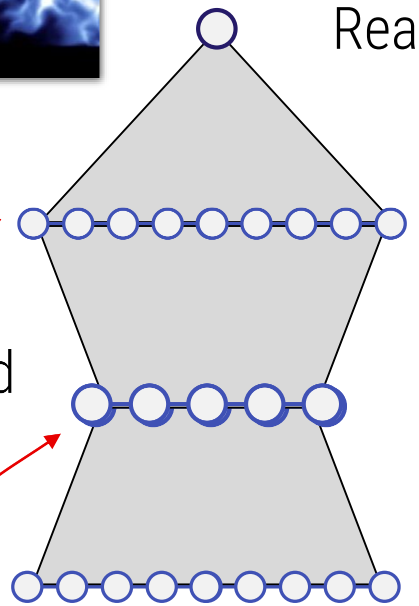


[Chuan Li / CelebA Data]

Learning Schemes



Details



Realistic? $\in [0,1]^n$

Reconstructed Data $\in \mathbb{R}^n$

Latend, Coarse-Grained Representation $\in \mathbb{R}^m$

Adversarial Auto-Encoder (Unexplained Entropy)

Original Data $\in \mathbb{R}^n$

Coarse Phenomenon

[Simulation Img: Andreas Kempf/Wikipedia]

MGANs: Noise → Images

[MGANs: Chuan Li; Training data: CelebA, University of Hong Kong]