Pattern Recognition
Lecture 05-1

Deep Learning Advanced

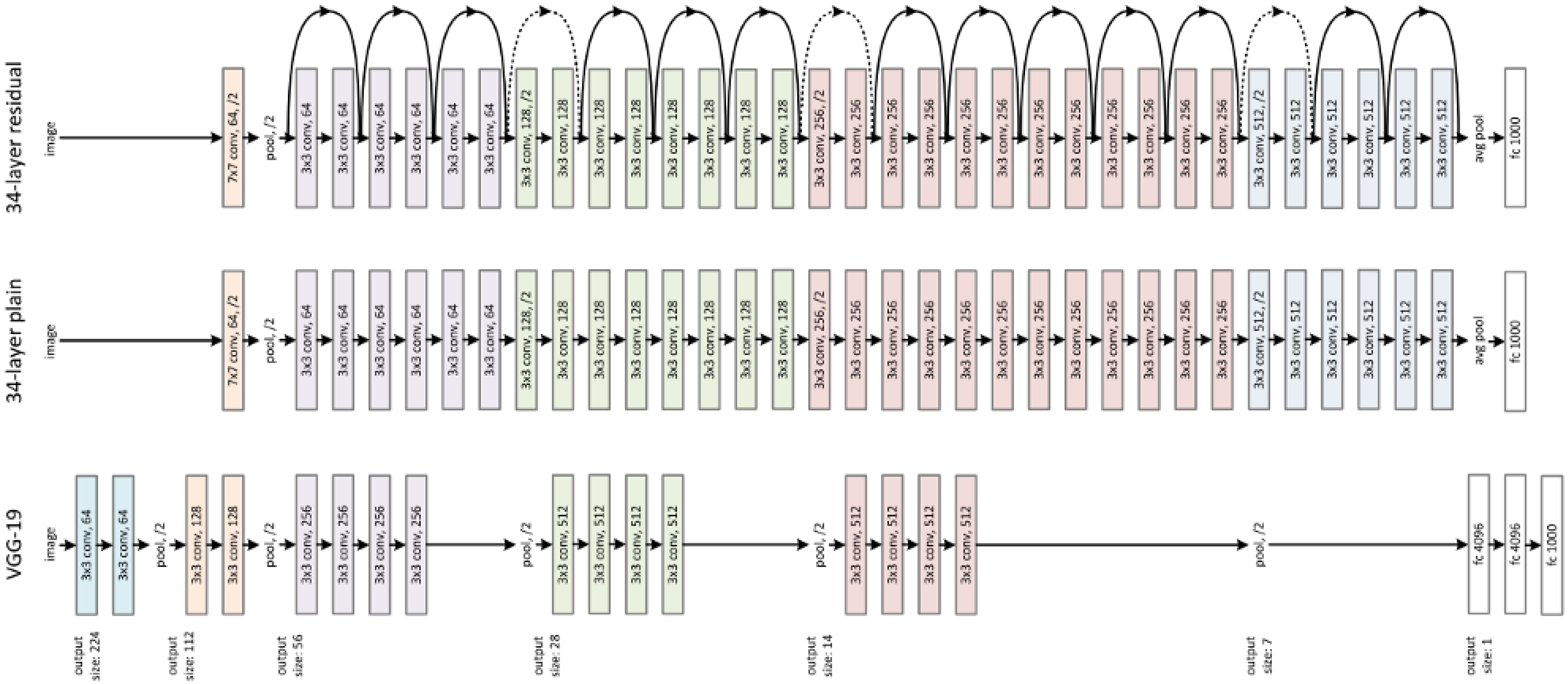
Prof. Jongwon Choi
Chung-Ang University
Fall 2022

This Class

- Deep Learning Advanced – 2
  - **Residual Network**
  - **Probabilistic Deep Learning**
  - **Variational Auto-encoder**

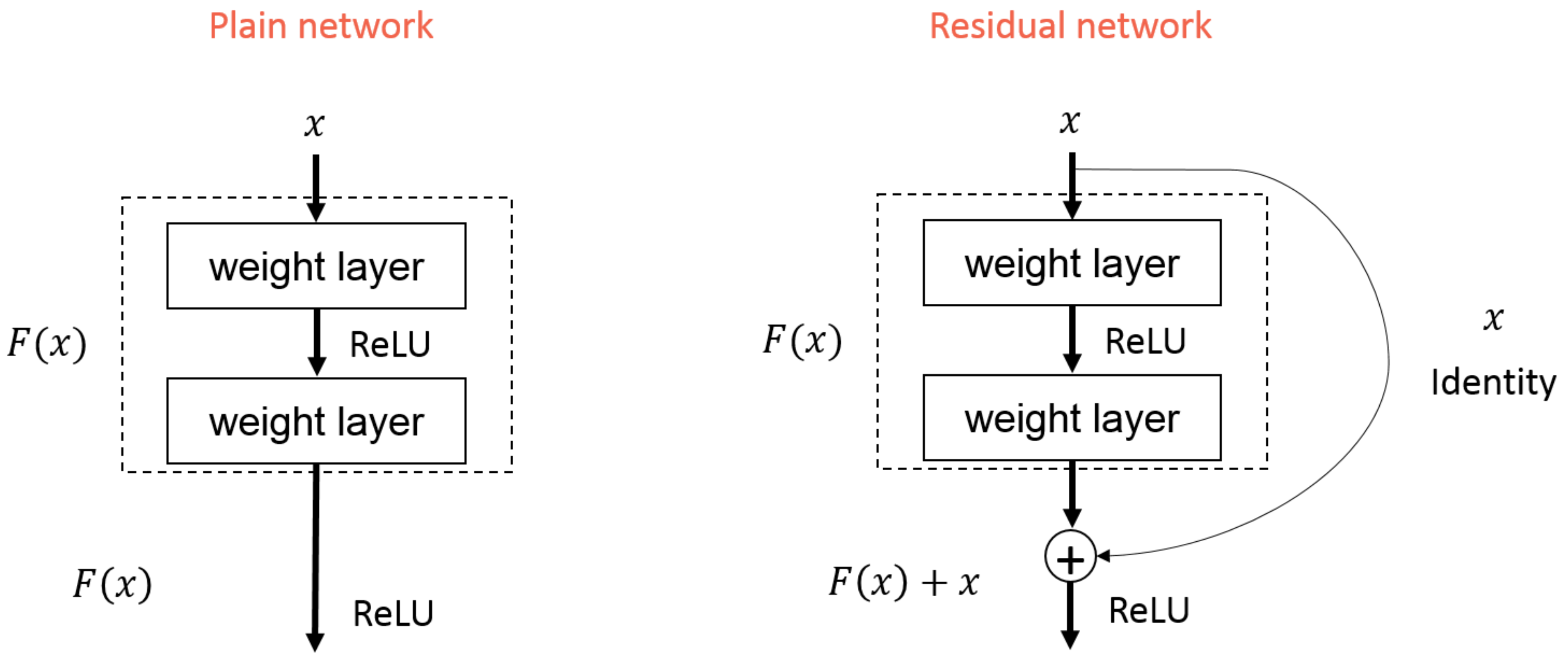
Residual Network

- Kaiming He et al., “Deep Residual Learning for Image Recognition”, CVPR2016



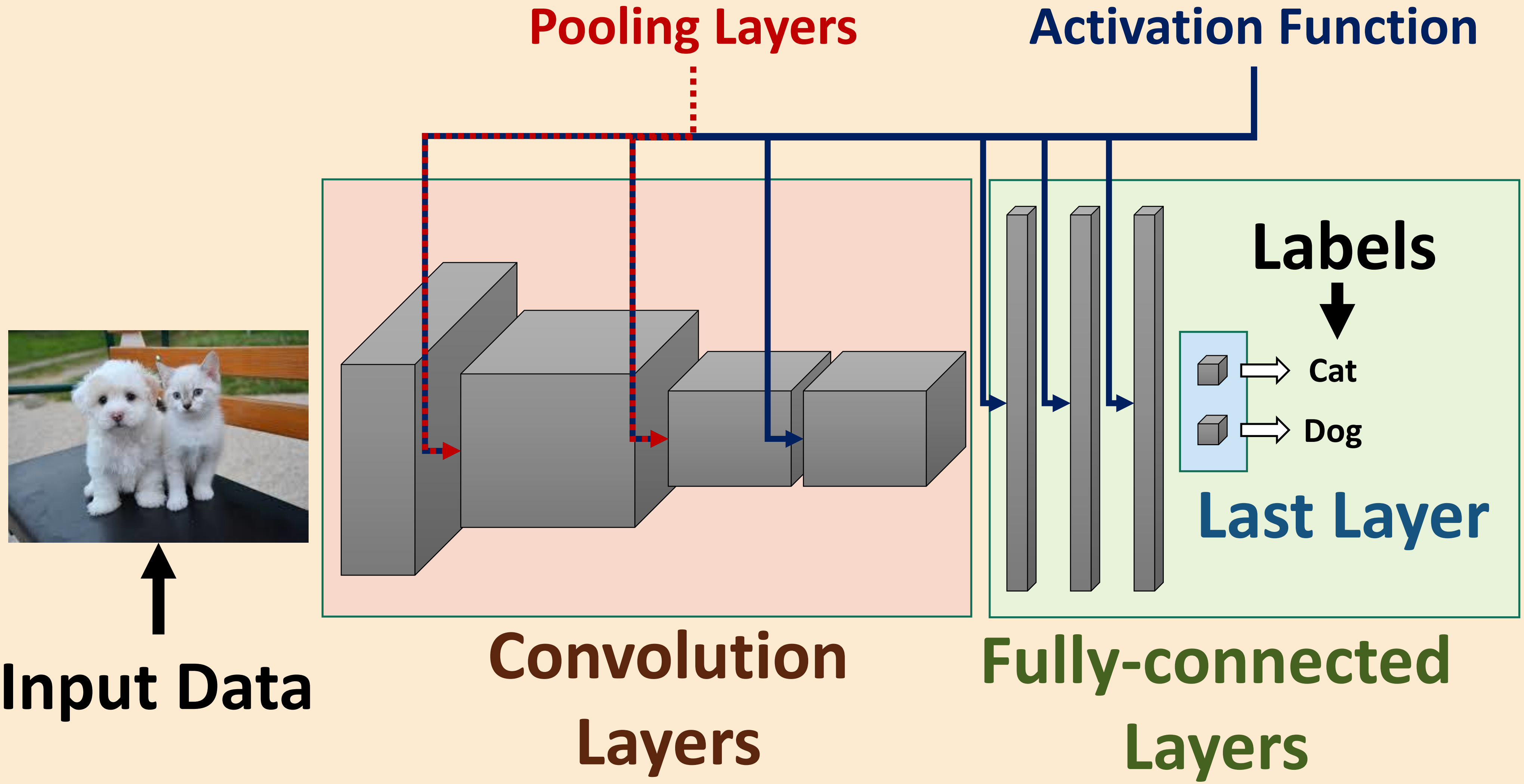
Residual Network

- Using a skip connection from the input of a block



Why Do We Need It??

Supervised Deep Learning - Architecture



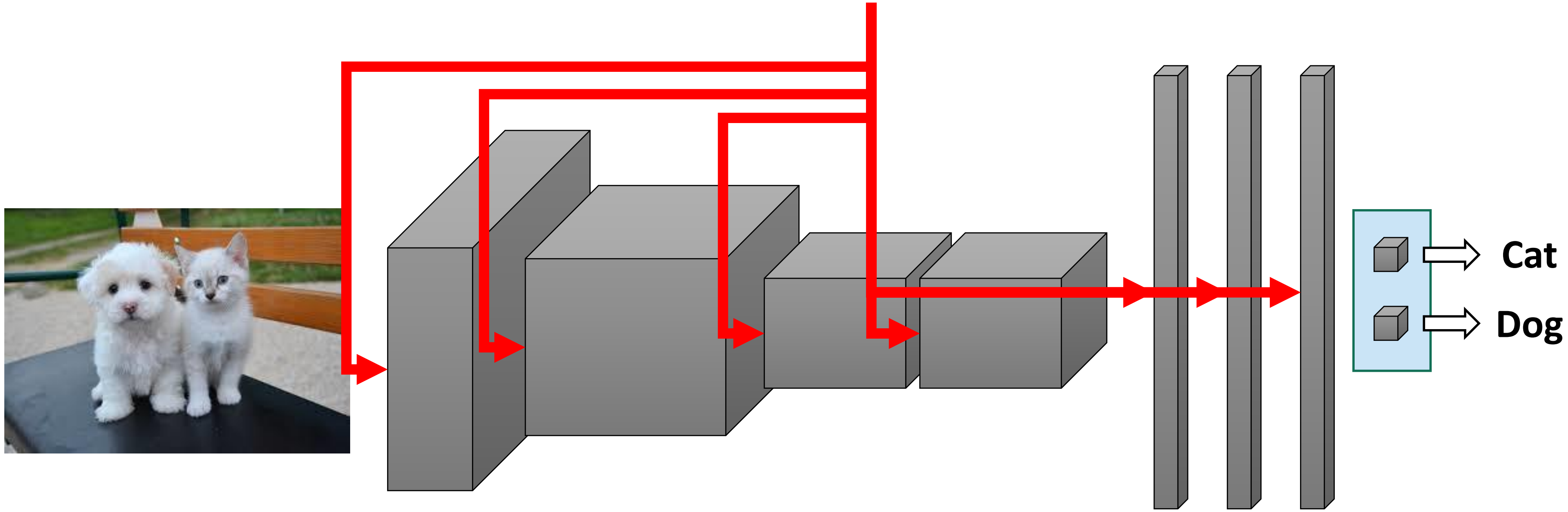
Supervised Deep Learning - Training

- **Gradient Descent on the remaining layers – Chain Rule!!**
  - Weight initialization – Gaussian random (Xavier's initialization)
  - for the iterative update: $w_{L-1}^{t+1} = w_{L-1}^{t} - \alpha^t \nabla f(w_{L-1}^{t})$
    - $\nabla_{\mathbf{w}_{L-1}^t} f_{CE}(\tilde{\mathbf{y}},\ \mathbf{y},\ \mathbf{x}) = \frac{\partial f_{CE}}{\partial \mathbf{w}_{L-1}^t} = \frac{\partial \mathbf{x}_{L-1}^t}{\partial \mathbf{w}_{L-1}^t} \times \frac{\partial f_{CE}}{\partial \mathbf{x}_{L-1}^t}$
    - $\nabla_{\mathbf{w}_{L-2}^t} f_{CE}(\tilde{\mathbf{y}},\ \mathbf{y},\ \mathbf{x}) = \frac{\partial f_{CE}}{\partial \mathbf{w}_{L-2}^t} = \frac{\partial \mathbf{x}_{L-2}^t}{\partial \mathbf{w}_{L-2}^t} \times \frac{\partial \mathbf{x}_{L-1}^t}{\partial \mathbf{x}_{L-2}^t} \times \frac{\partial f_{CE}}{\partial \mathbf{x}_{L-1}^t}$
    - $\nabla_{\mathbf{w}_{L-3}^t} f_{CE}(\tilde{\mathbf{y}},\ \mathbf{y},\ \mathbf{x}) = \frac{\partial f_{CE}}{\partial \mathbf{w}_{L-3}^t} = \frac{\partial \mathbf{x}_{L-3}^t}{\partial \mathbf{w}_{L-3}^t} \times \frac{\partial \mathbf{x}_{L-2}^t}{\partial \mathbf{x}_{L-3}^t} \times \frac{\partial \mathbf{x}_{L-1}^t}{\partial \mathbf{x}_{L-2}^t} \times \frac{\partial f_{CE}}{\partial \mathbf{x}_{L-1}^t}$

Xavier's Initialization

- $W \sim \mathcal{N}\left(0, \sqrt{\frac{2}{\#(input)+\#(output)}}\right)$

$$X \sim \mathcal{N}(0,1)$$



Supervised Deep Learning - Training

- $\nabla_{\mathbf{w}_{l^o}^t} f_{CE}(\tilde{\mathbf{y}},\ \mathbf{y},\ \mathbf{x}) = \frac{\partial f_{CE}}{\partial \mathbf{w}_l^t} = \frac{\partial \mathbf{x}_{\boldsymbol{l}}^t}{\partial \mathbf{w}_{\boldsymbol{l}}^t} \times \boxed{\prod_{l=l^o}^{L-2} \frac{\partial \mathbf{x}_{l+1}^t}{\partial \mathbf{x}_l^t}} \times \frac{\partial f_{CE}}{\partial \mathbf{x}_{L-1}^t}$

- Xavier's Initialization
  - $W \sim \mathcal{N}\left(0, \sqrt{\frac{2}{\#(input)+\#(output)}}\right) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$

- $\nabla_{\mathbf{w}_{l^o}^t} f_{CE}(\tilde{\mathbf{y}},\ \mathbf{y},\ \mathbf{x}) = \frac{\partial f_{CE}}{\partial \mathbf{w}_l^t} = \frac{\partial \mathbf{x}_{\boldsymbol{l}}^t}{\partial \mathbf{w}_{\boldsymbol{l}}^t} \times \prod_{l=l^o}^{L-2} \mathbf{W}_l \times \frac{\partial f_{CE}}{\partial \mathbf{x}_{L-1}^t}$

Residual Network

- Using a skip connection from the input of a block
  - $\nabla_{\mathbf{w}_{l^o}^t} f_{CE}(\tilde{\mathbf{y}},\ \mathbf{y},\ \mathbf{x}) = \frac{\partial f_{CE}}{\partial \mathbf{w}_l^t} = \frac{\partial \mathbf{x}_l^t}{\partial \mathbf{w}_l^t} \times \prod_{l=l^o}^{L-2} \frac{\partial \mathbf{x}_{l+1}^t}{\partial \mathbf{x}_l^t} \times \frac{\partial f_{CE}}{\partial \mathbf{x}_{L-1}^t}$
  - becomes
  - $\nabla_{\mathbf{w}_{l^o}^t} f_{CE}(\tilde{\mathbf{y}},\ \mathbf{y},\ \mathbf{x}) = \frac{\partial f_{CE}}{\partial \mathbf{w}_l^t} = \frac{\partial \mathbf{x}_l^t}{\partial \mathbf{w}_l^t} \times \prod_{l=l^o}^{L-2} (\mathbf{W}_l + 1) \times \frac{\partial f_{CE}}{\partial \mathbf{x}_{L-1}^t}$
  - instead of
  - $\nabla_{\mathbf{w}_{l^o}^t} f_{CE}(\tilde{\mathbf{y}},\ \mathbf{y},\ \mathbf{x}) = \frac{\partial f_{CE}}{\partial \mathbf{w}_l^t} = \frac{\partial \mathbf{x}_l^t}{\partial \mathbf{w}_l^t} \times \prod_{l=l^o}^{L-2} \mathbf{W}_l \times \frac{\partial f_{CE}}{\partial \mathbf{x}_{L-1}^t}$

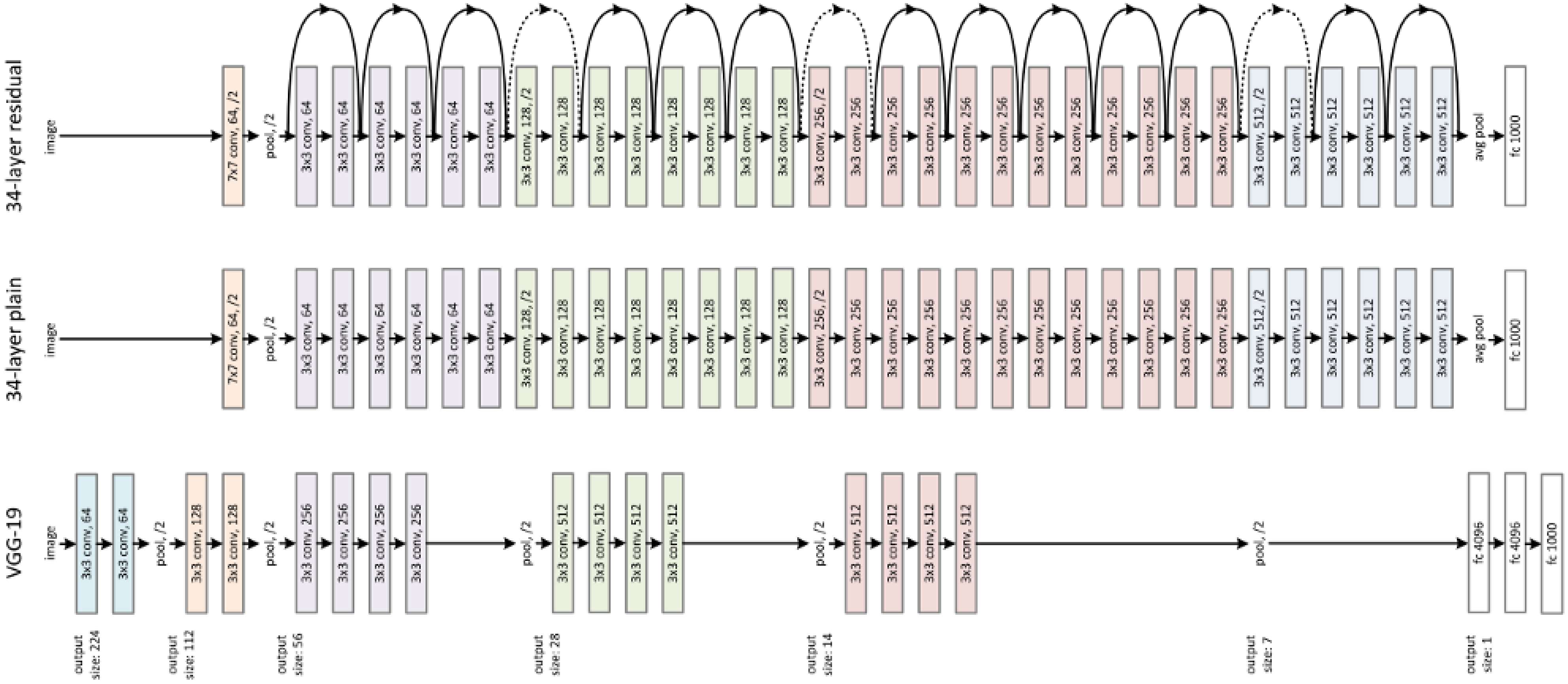
n from the input of a block

$= \frac{\partial \mathbf{x}_l^t}{\partial \mathbf{w}_l^t} \times \prod_{l=l^o}^{L-2} \frac{\partial \mathbf{x}_{l+1}^t}{\partial \mathbf{x}_l^t} \times \frac{\partial f_{CE}}{\partial \mathbf{x}_{L-1}^t}$

$= \frac{\partial \mathbf{x}_l^t}{\partial \mathbf{w}_l^t} \times \prod_{l=l^o}^{L-2} (\mathbf{W}_l + 1) \times \frac{\partial f_{CE}}{\partial \mathbf{x}_{L-1}^t}$

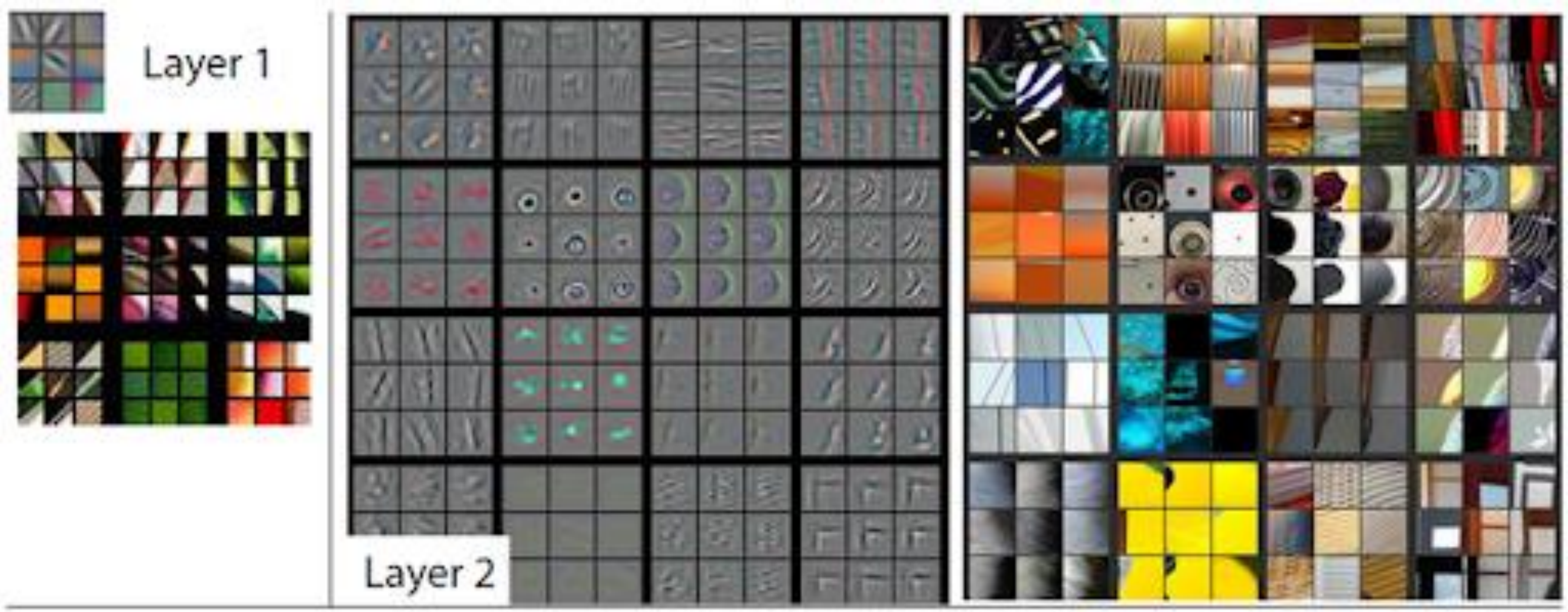
$= \frac{\partial \mathbf{x}_l^t}{\partial \mathbf{w}_l^t} \times \prod_{l=l^o}^{L-2} \mathbf{W}_l \times \frac{\partial f_{CE}}{\partial \mathbf{x}_{L-1}^t}$

Residual Network



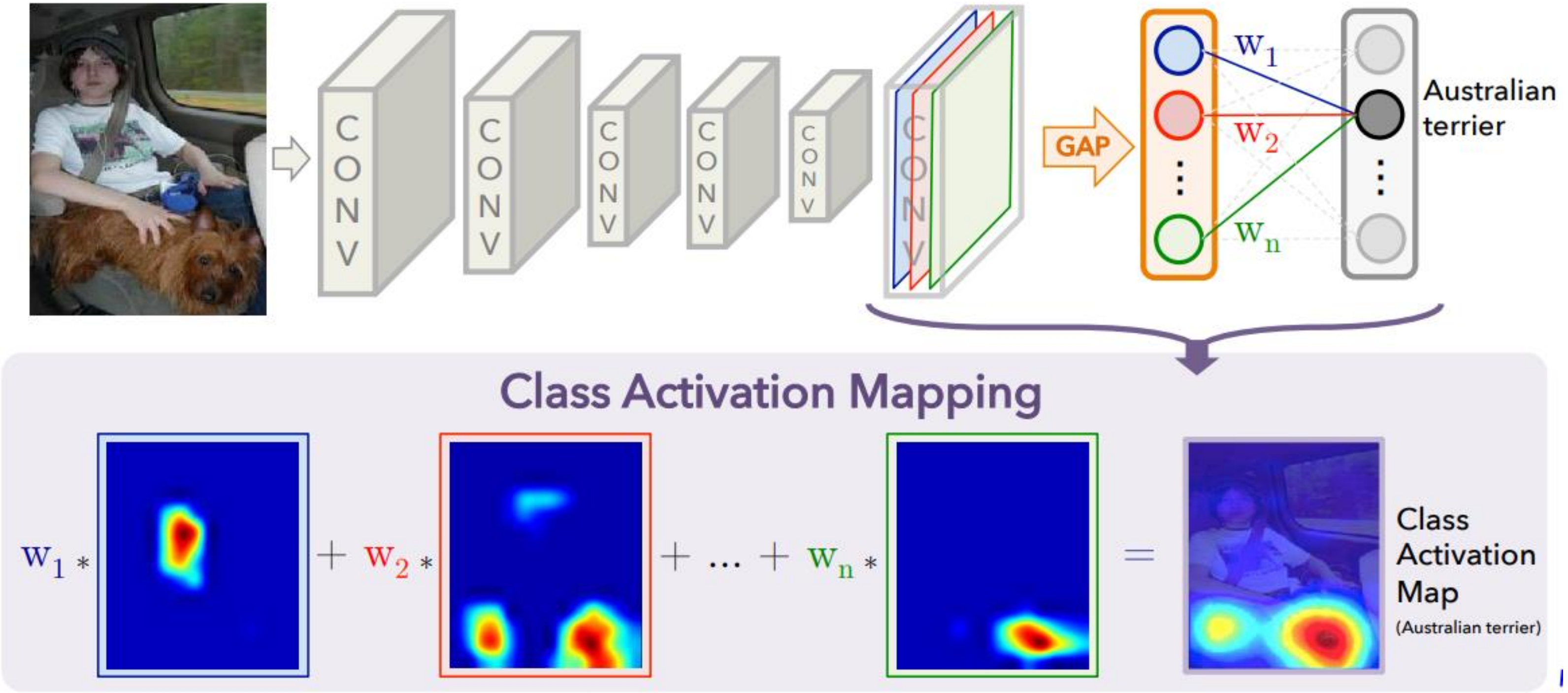
ZFNet & Class Activation Map

- **ZFNet –** “Visualizing and Understanding Convolutional Networks”, ECCV2014
  - Works well for AlexNet & VGGNet (Plain CNN)
  - A little bit weak visualization for ResNet (Due to the skip connection)



ZFNet & Class Activation Map

- **CAM –** "Learning Deep Features for Discriminative Localization", CVPR2016
  - Weak-supervised spatial attention for NN (Plain & Residual NN)
  - GAP-based model approaches (ResNet)

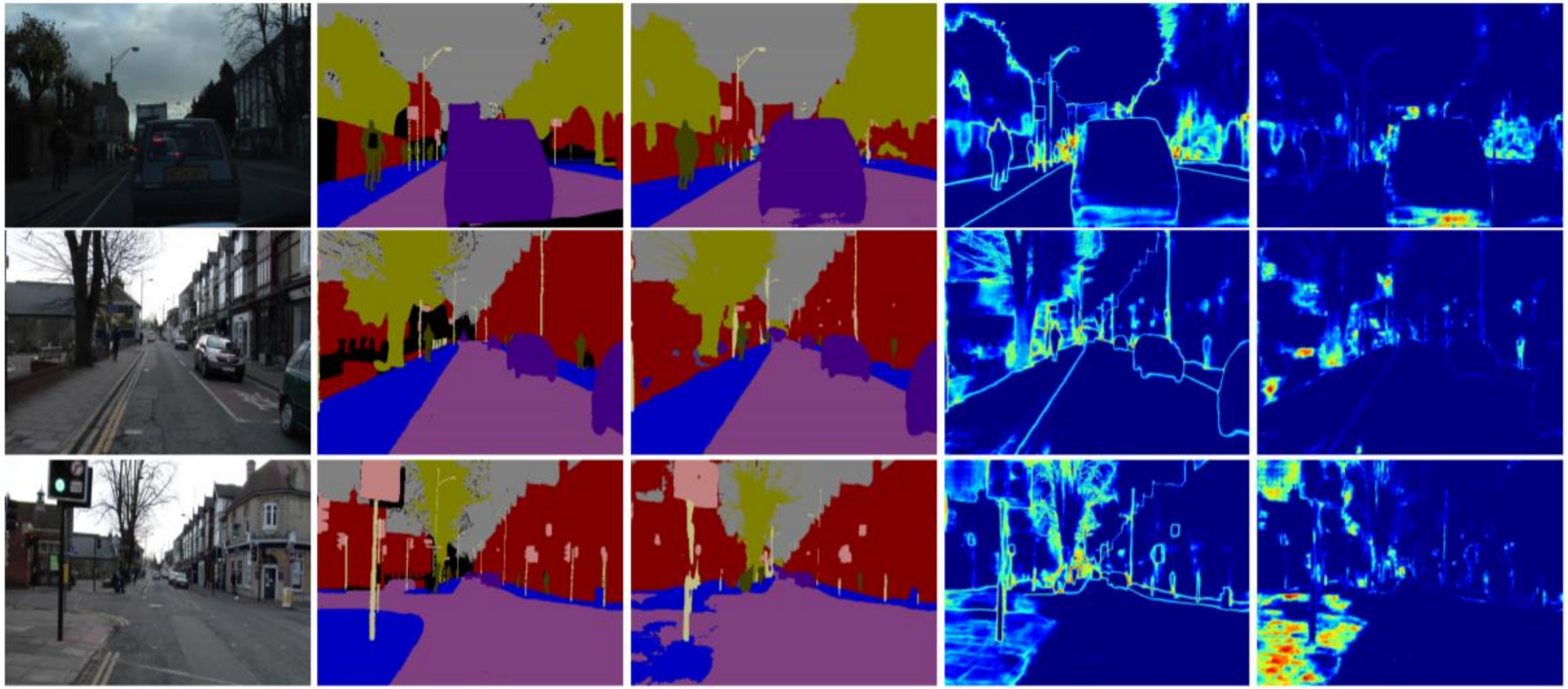


Probabilistic Deep Learning

- Deep network based on the probabilistic distributions
- There are various types of probabilistic DL
  - Probabilistic output
  - Probabilistic hidden vectors
  - Mixtured model

Bayesian Deep Learning

- NN with Probabilistic Output
  - “What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?”, NIPS2017



(a) Input Image (b) Ground Truth (c) Semantic Segmentation (d) Aleatoric Uncertainty (e) Epistemic Uncertainty

Bayesian Deep Learning

- NN with Probabilistic Output
  - “What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?”, NIPS2017
    - A single network to transform the input x, with its head split to predict both y as well as σ.

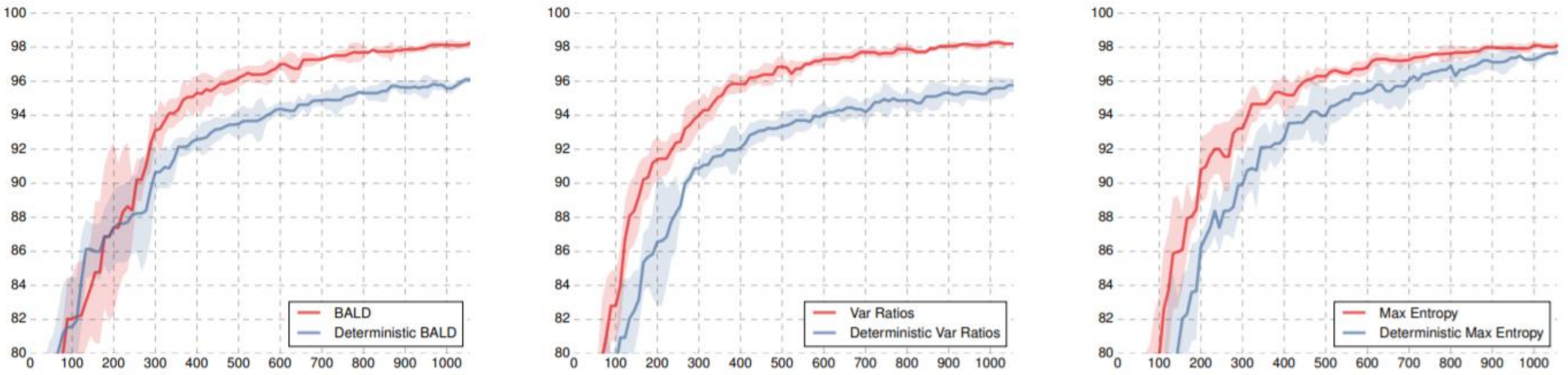
$$[\hat{\mathbf{y}}, \hat{\sigma}^2] = \mathbf{f}^{\widehat{\mathbf{W}}}(\mathbf{x})$$

    - We don’t know the uncertainty! => Unsupervised learning of the uncertainty

$$\mathcal{L}_{BNN}(\theta) = \frac{1}{D} \sum_i \frac{1}{2} \hat{\sigma}_i^{-2} ||\mathbf{y}_i - \hat{\mathbf{y}}_i||^2 + \frac{1}{2} \log \hat{\sigma}_i^2$$

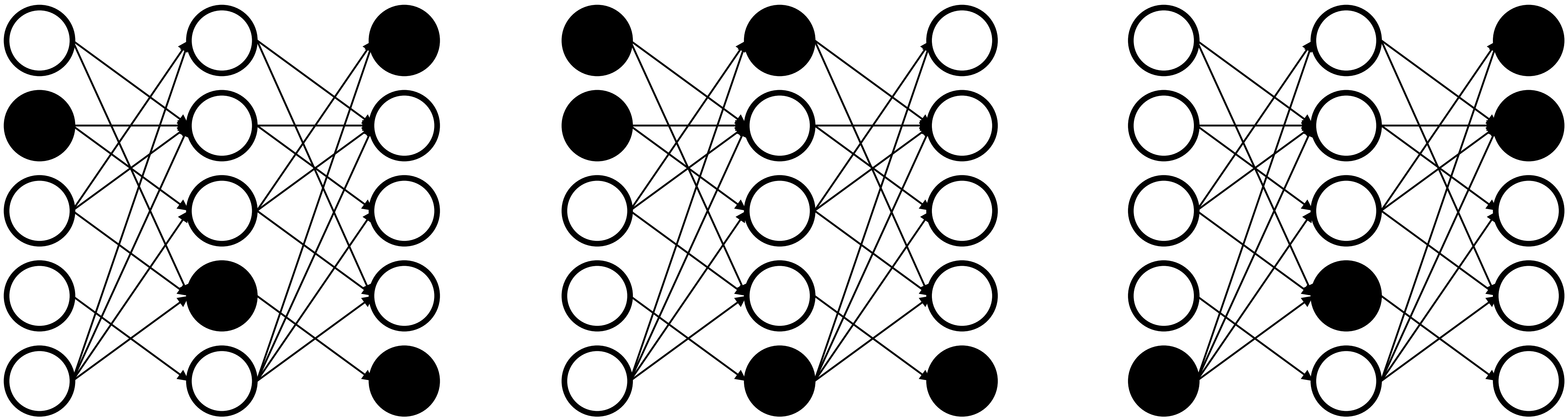
MC-DROPOUT

- Simply estimate the uncertainty by using the dropout in test phase
  - “Deep Bayesian Active Learning with Image Data”, ICML2017



MC-DROPOUT

- Simply estimate the uncertainty by using the dropout in test phase
  - “Deep Bayesian Active Learning with Image Data”, ICML2017
    - Dropout : Set the randomly chosen neuron to 0



Update 1 **Update 2** **Update 3**

MC-DROPOUT

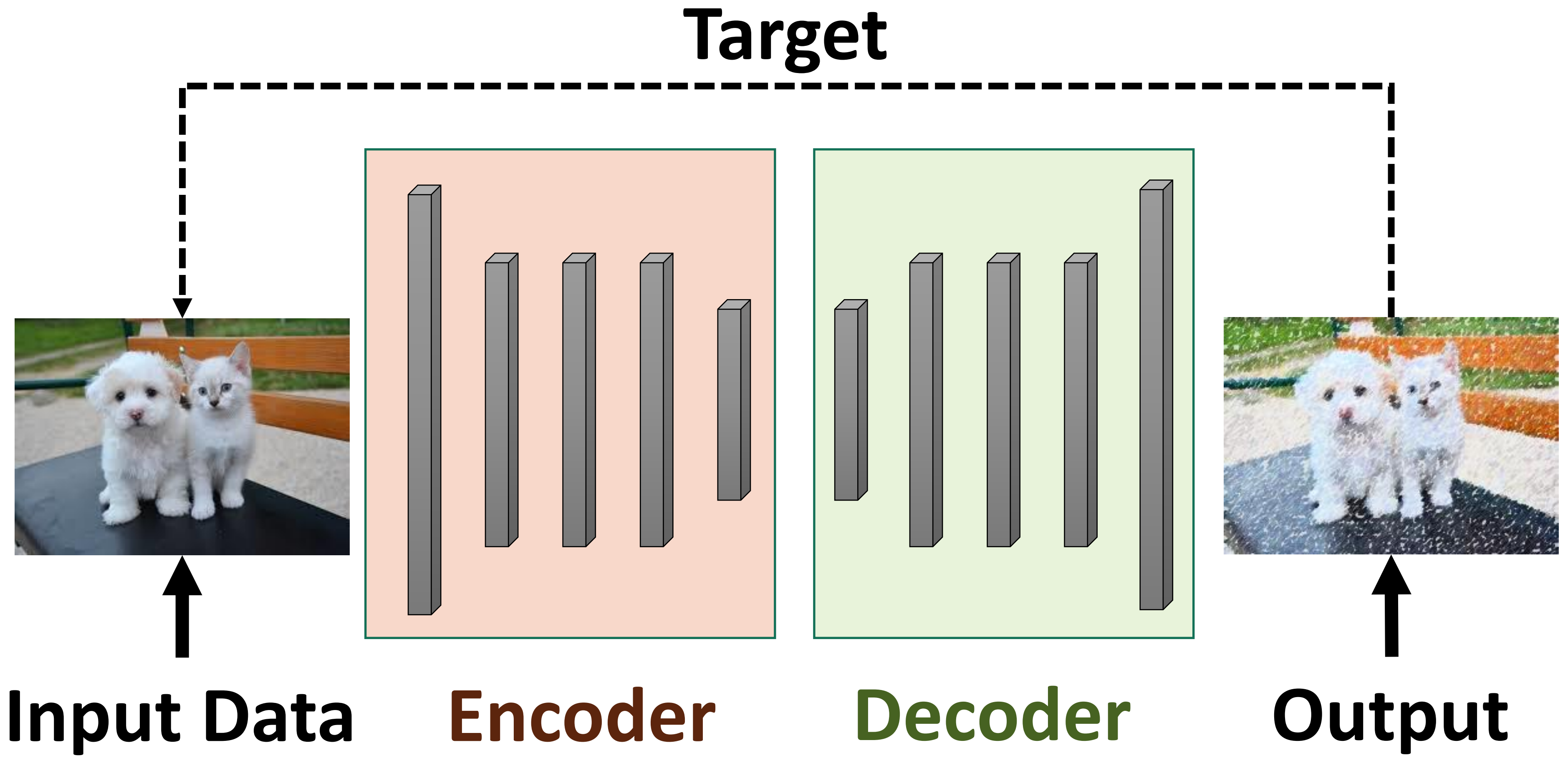
- Simply estimate the uncertainty by using the dropout in test phase
  - “Deep Bayesian Active Learning with Image Data”, ICML2017
    - Dropout : Set the randomly chosen neuron to 0

    - Increase the ratio of neuron for dropout,
    - Remain the dropout process even on the test

    - The training phase takes longer time and is sometimes unstable
    - In testing phase, we can estimate the sampling-based probabilistic distribution

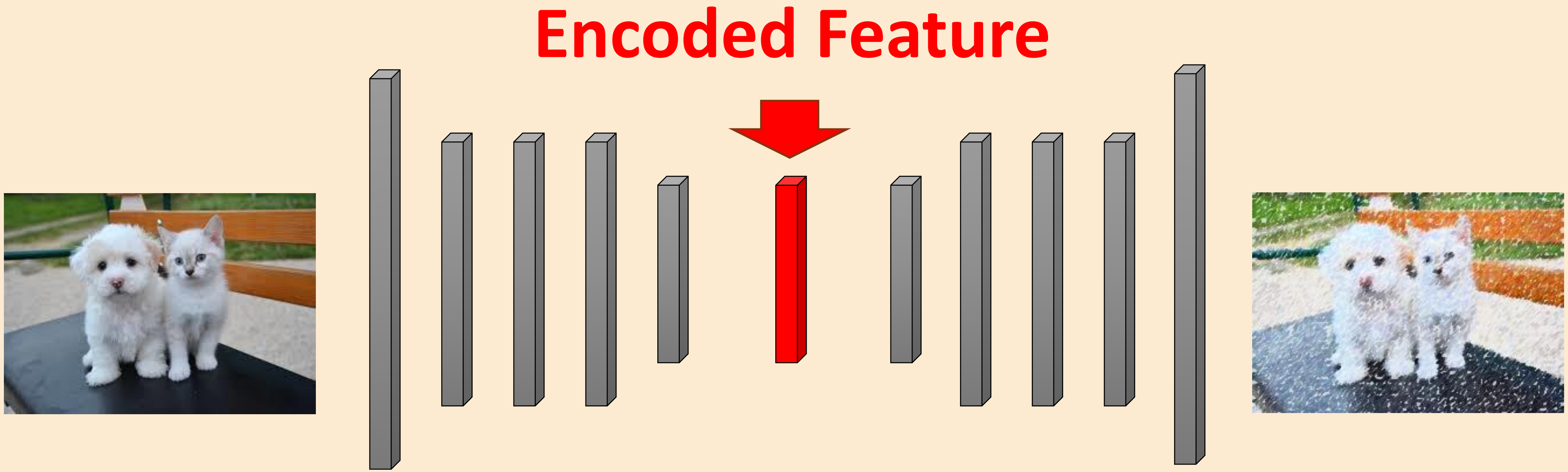
Variational Auto-encoder

- Similar architecture with auto-encoder



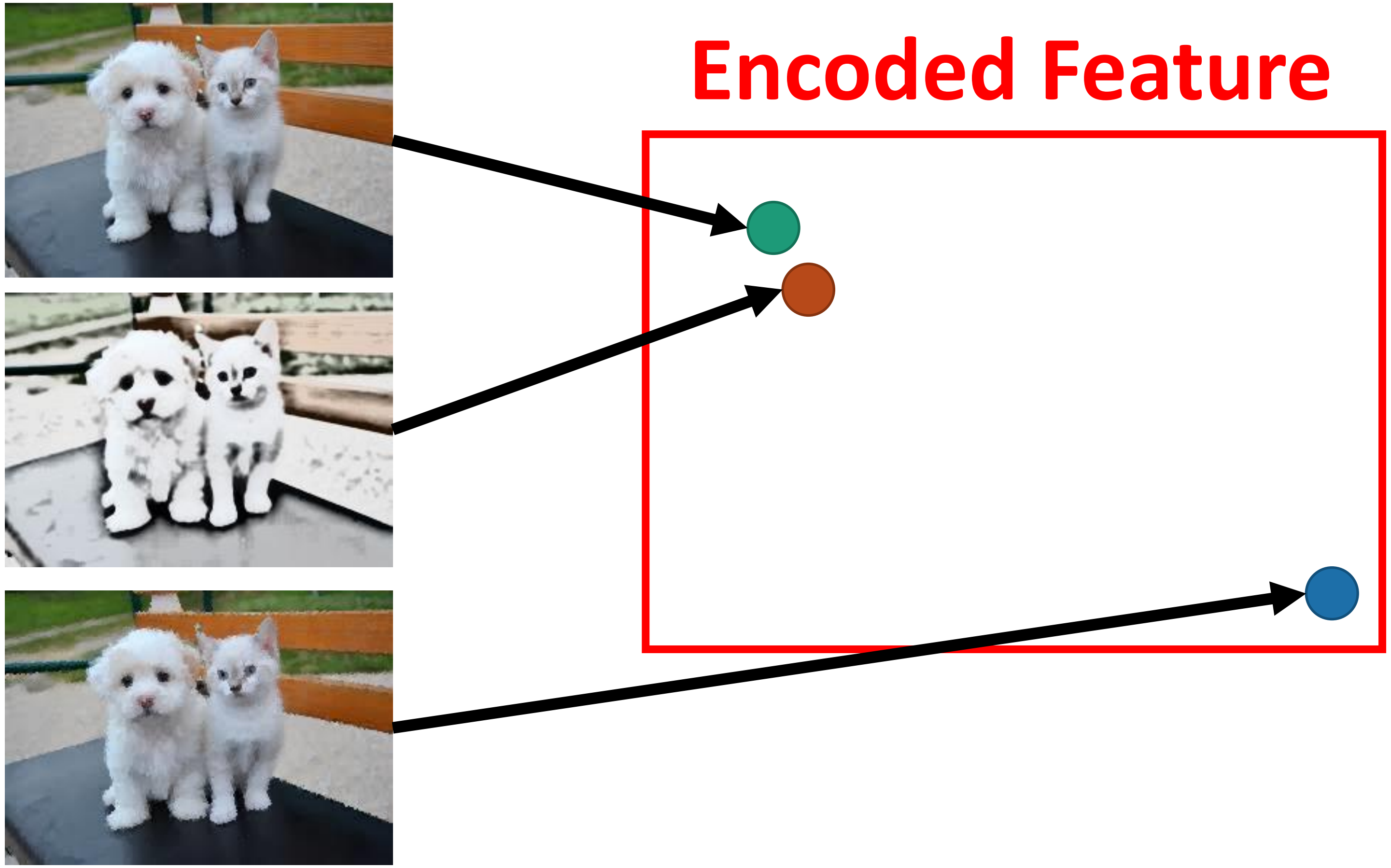
Auto-encoder - Architecture

- When the encoded feature is smaller than the input data,
- the information of input data is compressed in the encoded feature
- because the input data should be reconstructed from that!



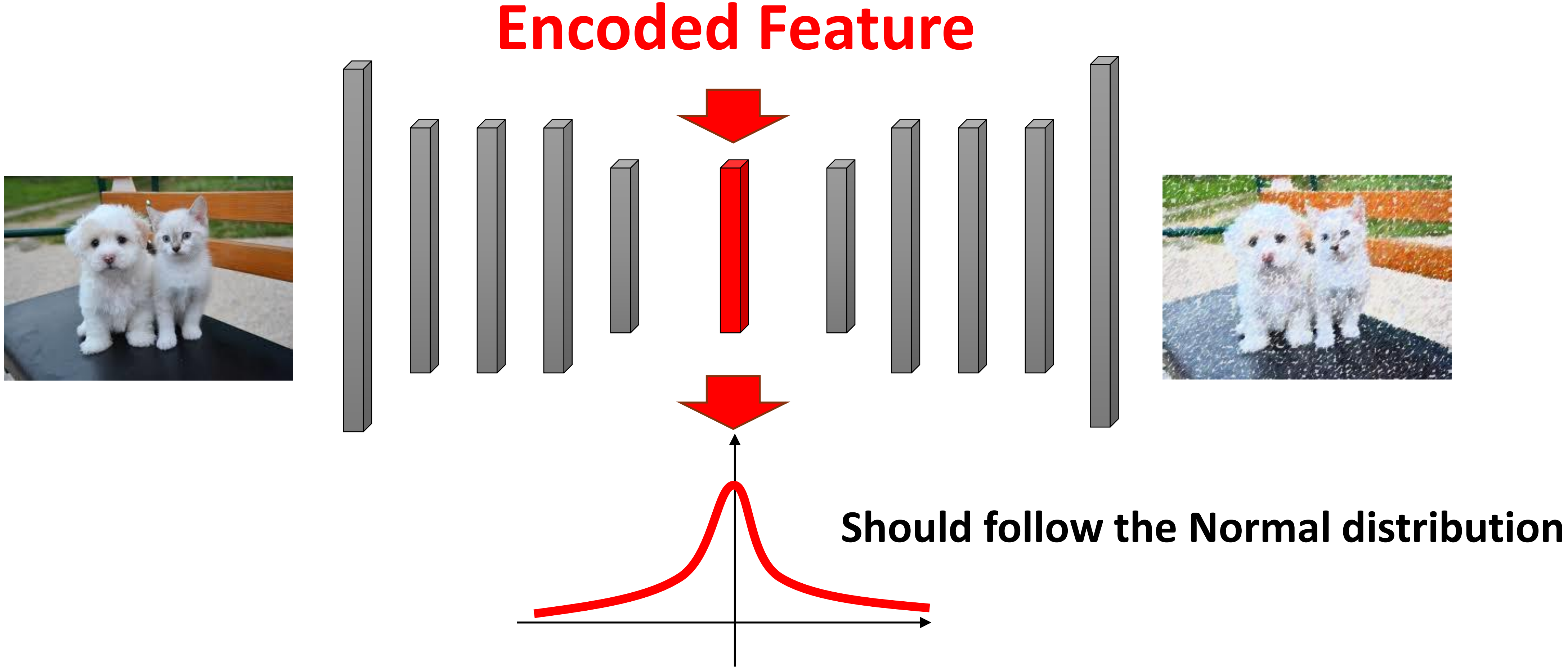
Variational Auto-encoder

- Limitation of the conventional auto-encoder



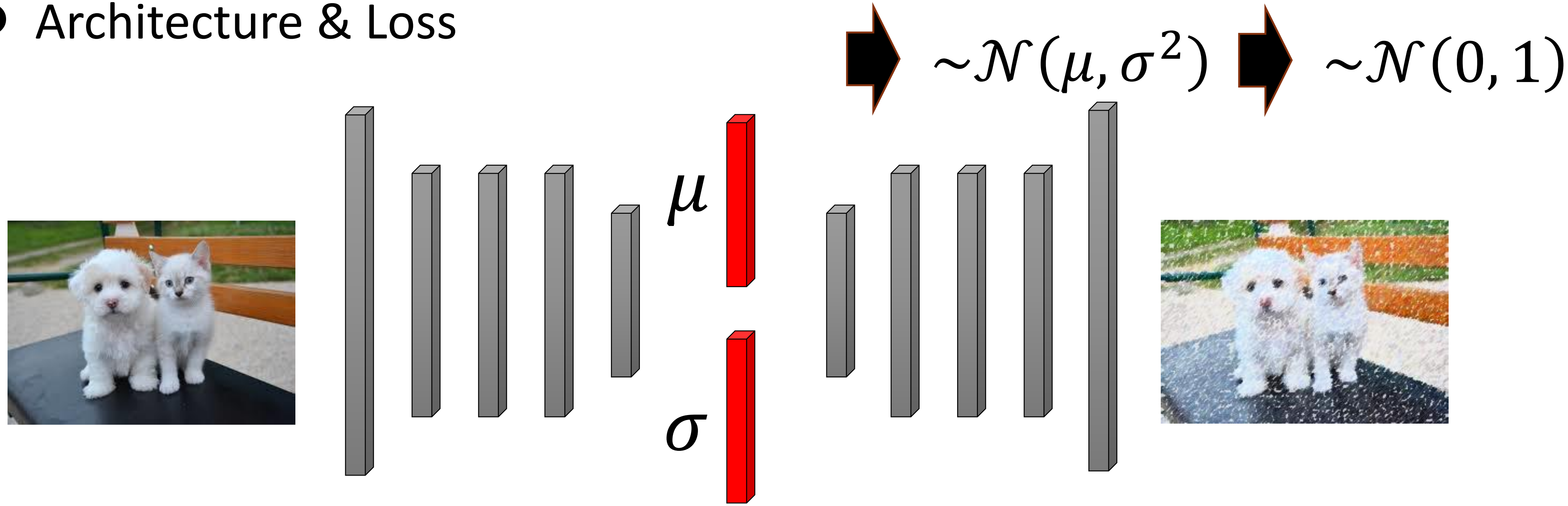
Variational Auto-encoder

- Architecture



Variational Auto-encoder

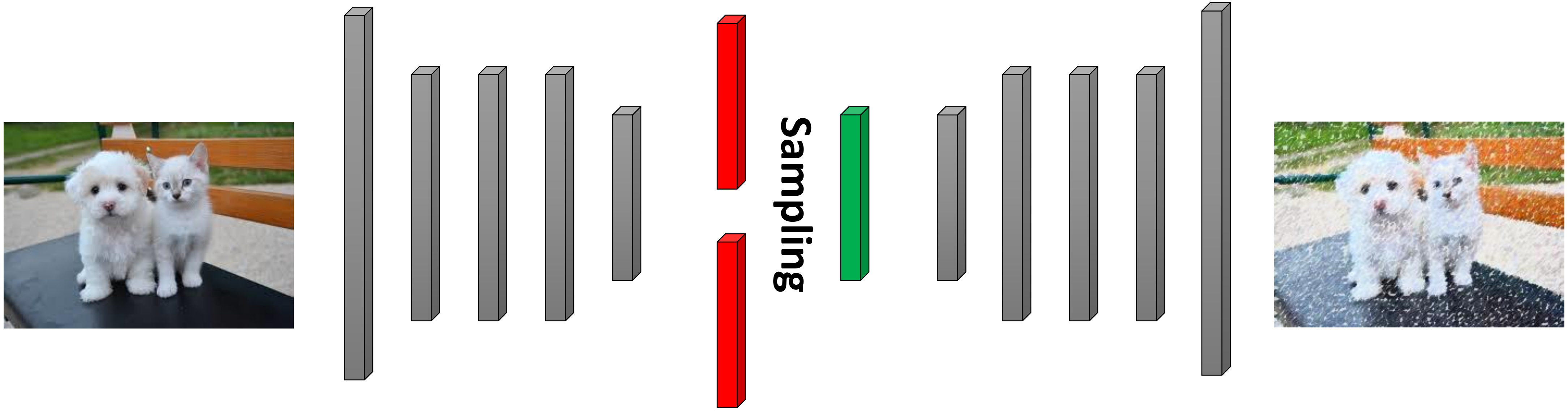
- Architecture & Loss



$$L = -E_{z\sim q(z|x)}[\log p(x|z)] + D_{KL}\big(q(z|x)||p(z)\big)$$

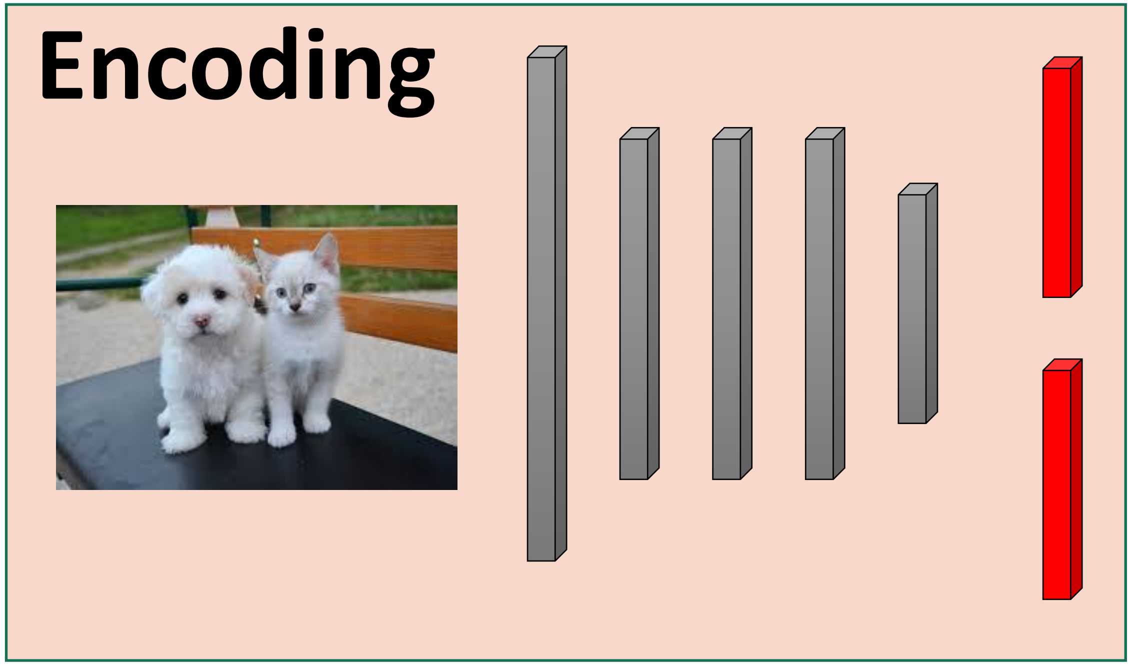
Variational Auto-encoder

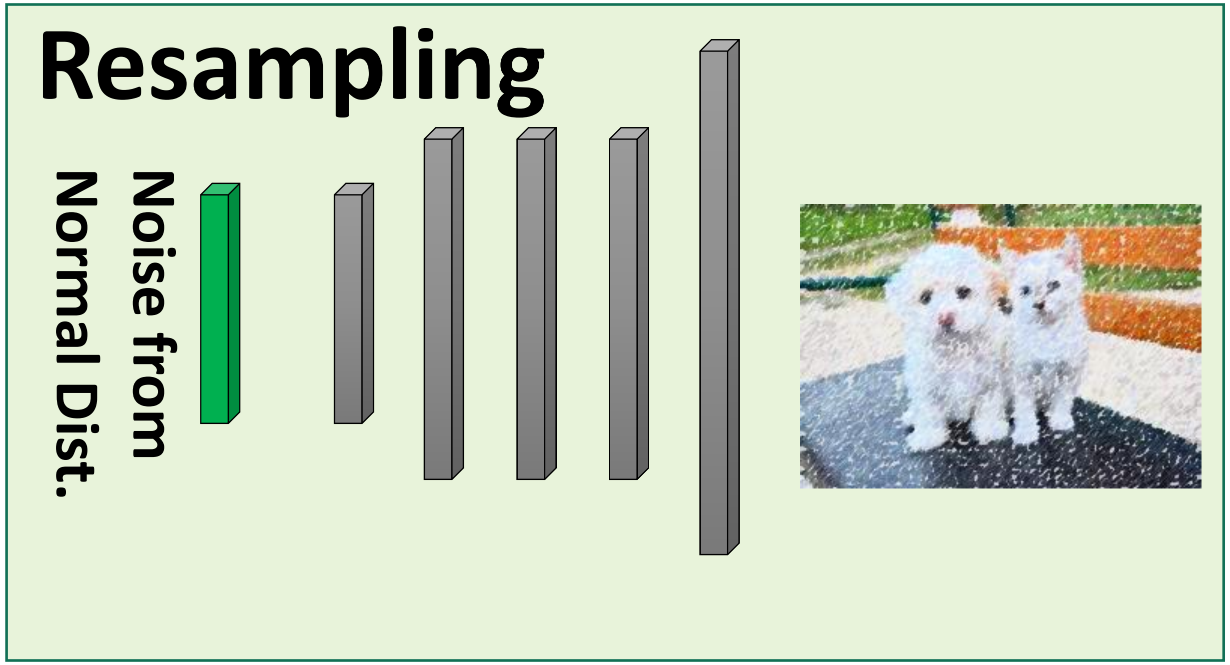
- Training Phase
  - Estimate mean & variance
  - Randomly sample a hidden variable according to the distribution
  - Reconstruct the target image



Variational Auto-encoder

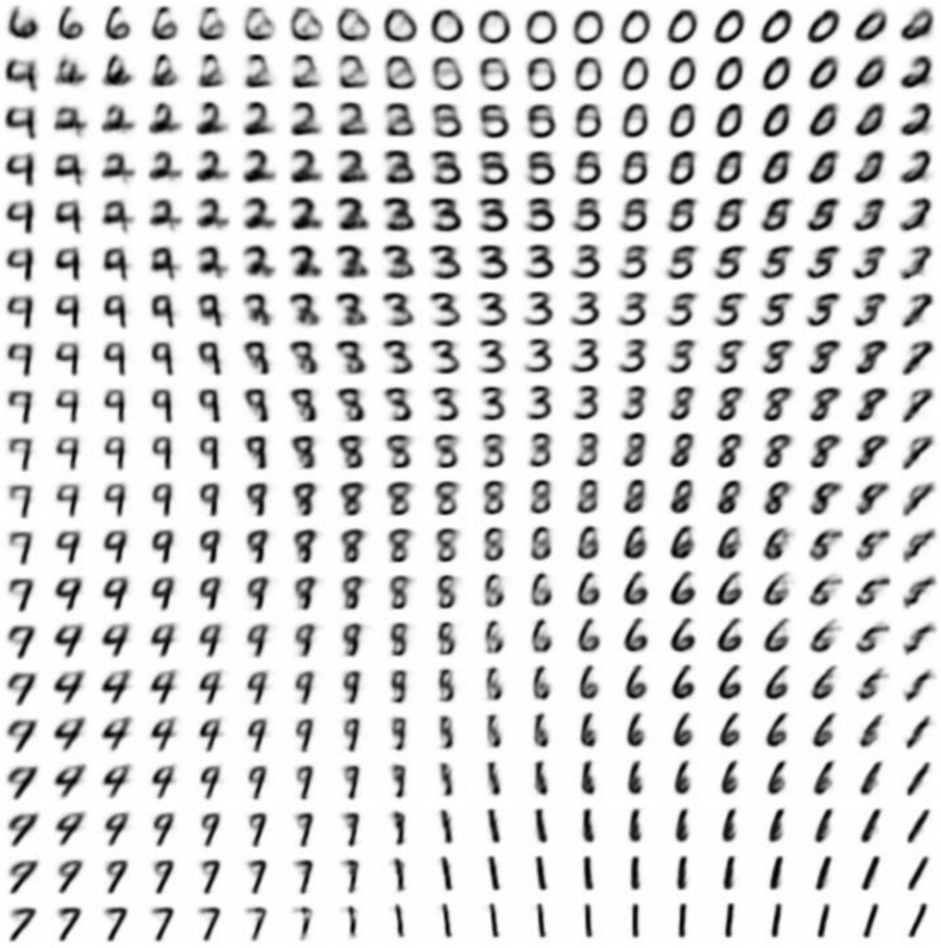
- Inference Phase
  - 1. Encoding : Obtain the probabilistic distribution by the encoder
  - 2. Resampling : Reconstruct from the random noise of normal distribution





Variational Auto-encoder

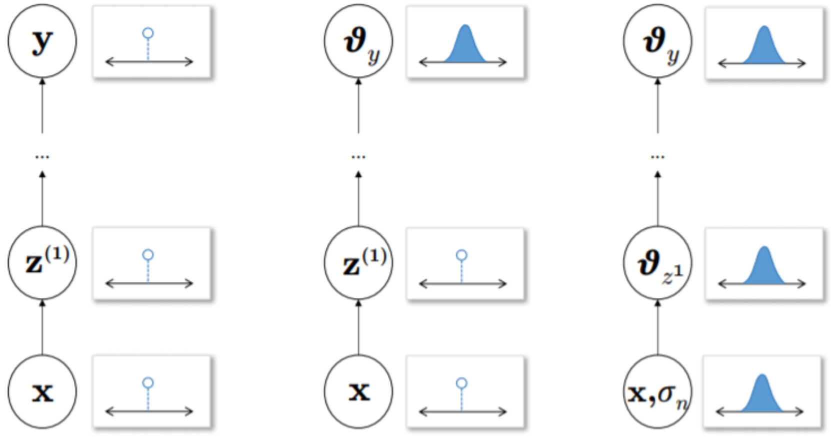
(a) Learned Frey Face manifold



(b) Learned MNIST manifold

Mixtured Model

- Estimate the last probabilistic distribution based on the probabilistic distributions of hidden variables
- The relationship between the layers becomes very complex
- Many methods to simplify the relationship have been proposed
  - Ex. “Lightweight Probabilistic Deep Networks”, CVPR2018



(a) Determ. (b) ProbOut (c) ADF

Summary

- Deep Learning Advanced
  - **Residual Network**
  - **Probabilistic Deep Learning**
  - **Variational Auto-encoder**