CVPR 2021 Tutorial

Normalization Techniques in Deep Learning: Methods, Analyses and Applications

Lei Huang

Beihang University, Beijing, China

北京航空航天大学人工智能研究院
Institute of Artificial Intelligence, Beihang University

2021-06-19

软件开发环境国家重点实验室
State Key Laboratory of Software Development Environment

Deep Neural Networks

Computer vision

Image classification

Segmentation

.....

3D vision

Object detection

Object tracking

Natural language processing

Sentiment Analysis

Knowledge Graph

.....

Question answering

Language model

Machine translation

Speech and audio processing

.....

Speech recognition

Speech Synthesis

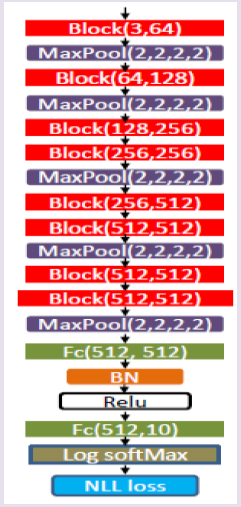
.....

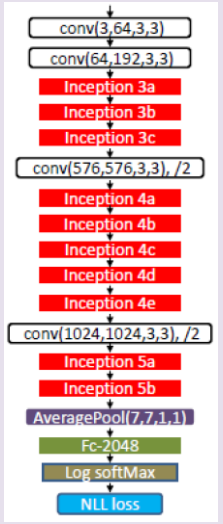
Speech Enhancement

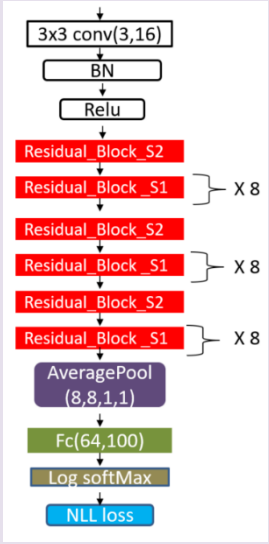
Speech coding

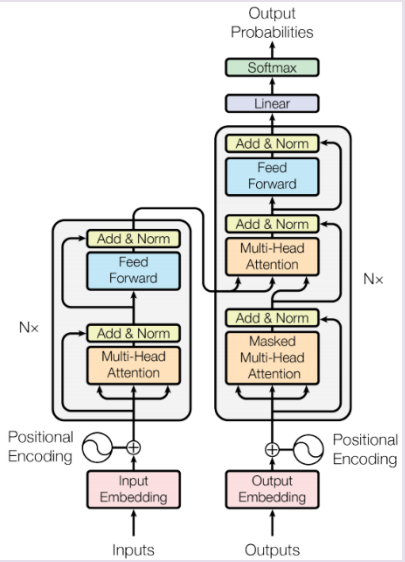
Speaker identification

Deep Neural Networks (DNNs)



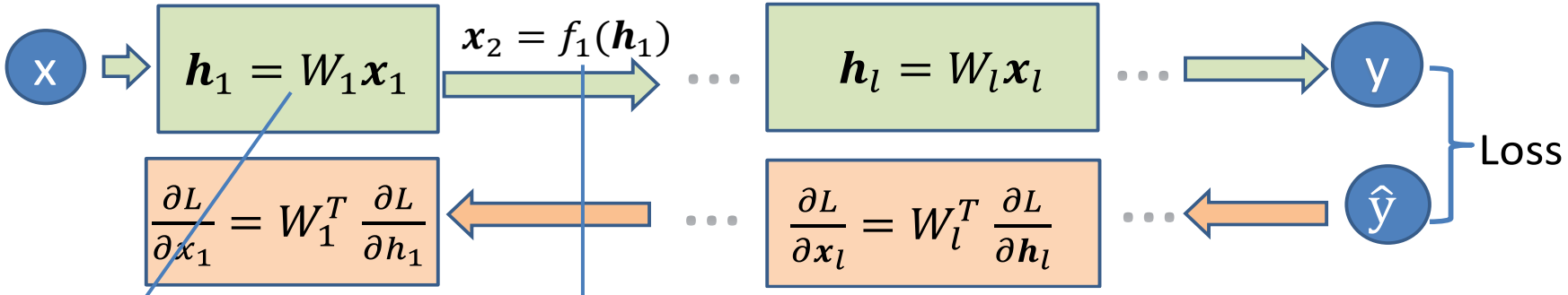






Milestones of DNNs

- The gradient explosion or vanishing problem



Linear

nonlinear

Layer-wise initialization

ReLU

2006, Science

Citation: 15200+

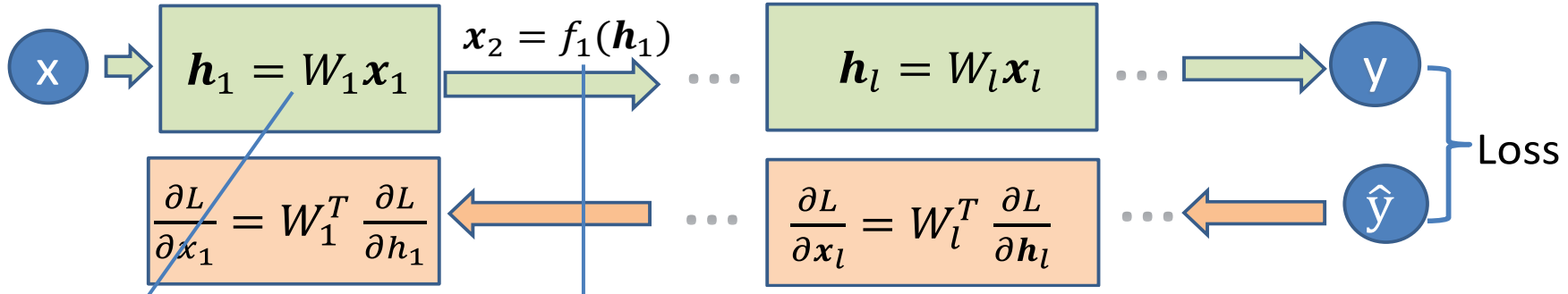
2010, ICML

Citation: 13000+

[Hinton and Salakhutdinov, 2006; Nair and Hinton, 2010; Ioffe and Szegedy, 2015]

Milestones of DNNs

- The gradient explosion or vanishing problem



Linear

nonlinear

Normalization

Layer-wise initialization

ReLU

Batch Normalization (BN)

2006, Science

Citation: 15200+

2010, ICML

Citation: 13000+

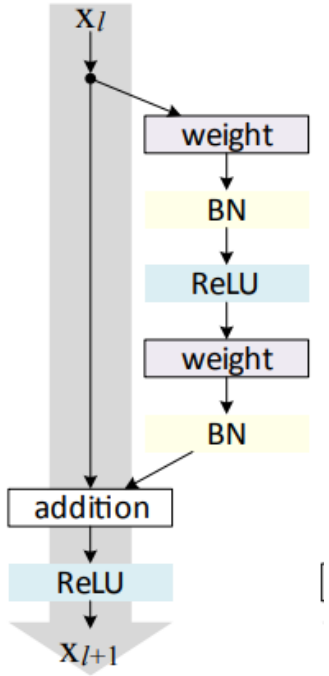
2015, ICML

Citation: 28000+

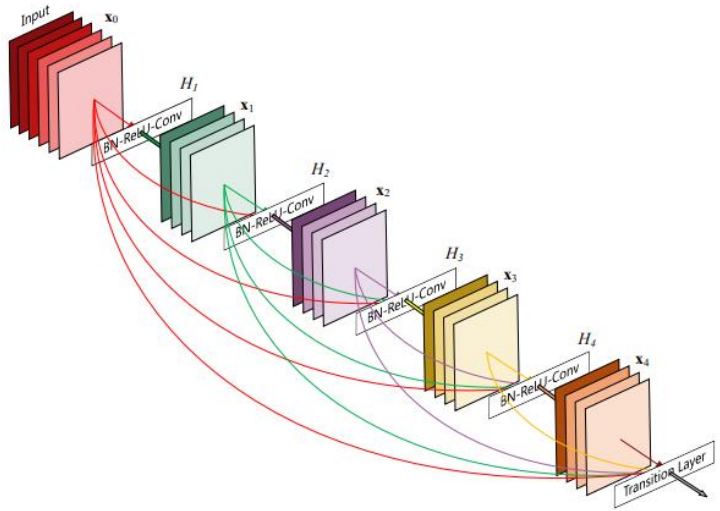
[Hinton and Salakhutdinov, 2006; Nair and Hinton, 2010; Ioffe and Szegedy, 2015]

BN: Key module of DNNs

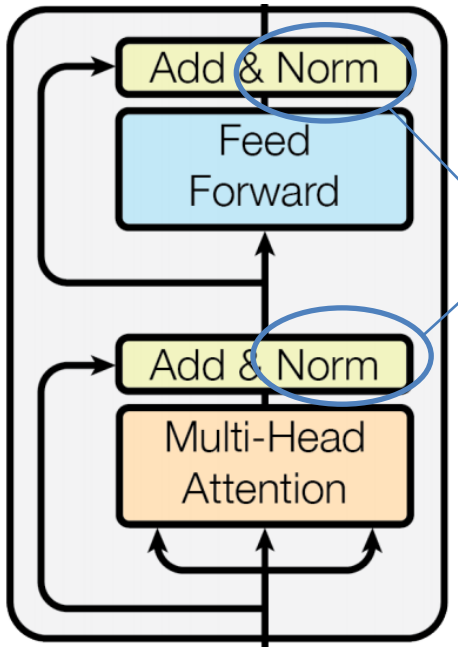
- Key module in current the state-of-the-art network architectures



Residual Network,
CVPR 2016
Citation: 81000+



DenseNet,
CVPR 2017
Citation: 16000+



Layer normalization

Transformer,
NeurIPS 2017
Citation: 22000+

[He et al, 2016; Ba et al, 2016; Huang et al, 2017; Vaswani et al, 2017]

Recently Papers Involving Normalization

- Statistics of publications on main conferences between 2020 to 2021
  - ICLR, AAAI, CVPR, ICML, ECCV, NeurIPS: 77+

	2020	2021
ICLR	**4+**	**6+**
AAAI	**7+**	**6+**
CVPR	**11+**	**15+**
ICML	**5+**	**7+**
ECCV	**7+**	
NeurIPS	**9+**	

Recently Papers Involving Normalization

- Statistics of publications on main conferences between 2020 to 2021
  - ICLR, AAAI, CVPR, ICML, ECCV, NeurIPS: 77+

	2020	2021
ICLR	**4+**	**6+**
AAAI	**7+**	**6+**
CVPR	**11+**	**15+**
ICML	**5+**	**7+**
ECCV	**7+**	
NeurIPS	**9+**	

***normalization/whitening

1.Rethink BN in ***
2.Towards understanding ***normalization from ***

1.***normalization for ***(tasks)
2.Rethink ***normalization in *** (tasks)

Some questions

- Why so many normalization variants? What are the main motivations behind them? And how can we present a taxonomy?

- How can we reduce the gap between empirical success of normalization techniques and our theoretical understanding of them?

- What recent advances have been made in designing/tailoring normalization techniques for different tasks, and what are the main insights behind them?

Outline

01. Motivations of Normalization Techniques

02. Introduction of Normalization Methods

03. Analyses of Normalization

04. Applications of Normalization

Disclaimers

- Inevitably miss important related work
- Citations are only representative examples
- There are no consistent understandings of normalization, this is one

Survey paper: "**Normalization Techniques in Training DNNs: Methodology, Analysis and Application**," arXiv:2009.12836

Paper list related: https://github.com/huangleiBuaa/NormalizationSurvey

Outline

01. Motivations of Normalization Techniques

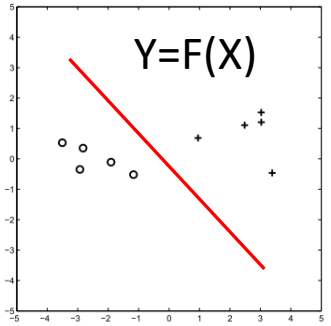
02. Introduction of Normalization Methods

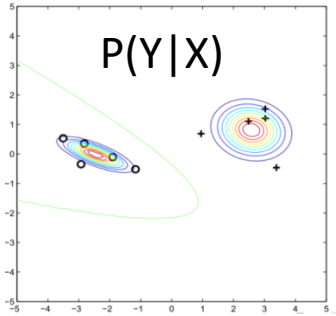
03. Analyses of Normalization

04. Applications of Normalization

Supervised Learning

- Dataset D={X, Y}
  - Input: X
  - Output: Y
  - Learning: $Y = F(X)$ or $P(Y|X)$
- Main types of learning models
  - Non-parametric model
    - $Y=F(X; x_1, x_2 \ldots x_n)$
  - Parametric model
    - $Y=F(X; \theta)$
- Training (Fitting) : $\min_\theta \mathcal{L} = E_D \, \ell(F(X; \theta), \hat{Y})$
- Generalization



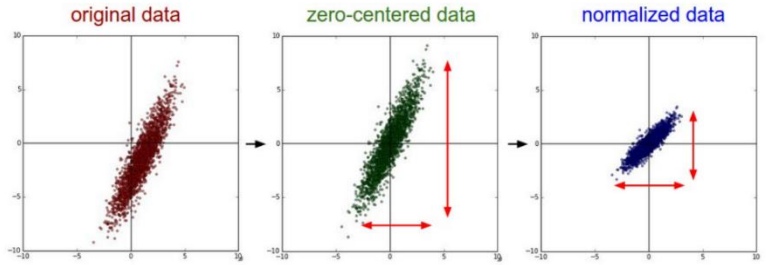


Normalization

- Definition of normalization
  - In statistics: adjustments of values or distributions in statistics
  - In image processing: changing the range of pixel intensity values
  - In data processing: general reduction of data to canonical form

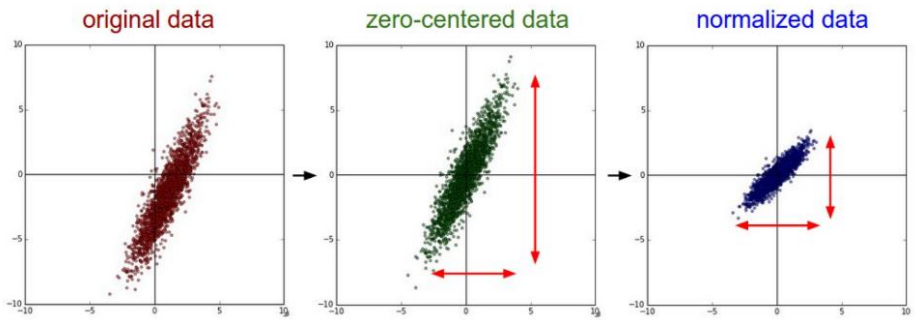
- Definition of normalization in this tutorial
  - Given a set of data $\mathbb{D} = \{\mathrm{x}^{(i)}\}_{i=1}^{N}$, the normalization operation is a function $\Phi: \mathrm{x} \longmapsto \hat{\mathrm{x}}$, which ensures that the transformed data $\hat{\mathbb{D}} = \{\hat{\mathrm{x}}^{(i)}\}_{i=1}^{N}$ has certain statistical properties.

$$\hat{\mathbf{x}} = \frac{\mathbf{x}-\boldsymbol{\mu}}{\boldsymbol{\sigma}}$$



Motivation of Normalizing Input

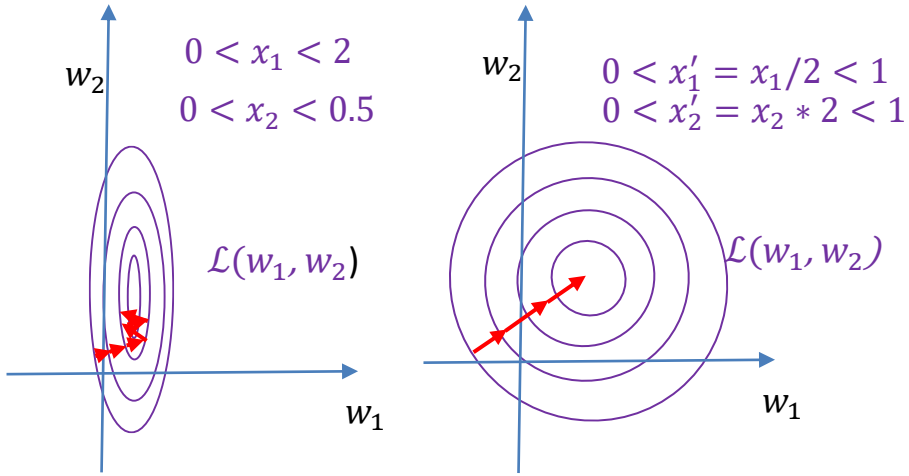
- Improve the effects of learning
  - Non-parameter models (KNN, Kernel SVM)
  - Distance/ Similarity



- Improve optimization efficiency
  - Parametric model (logistic regression)
  - Update parameters iteratively

$$\theta_{t+1} \leftarrow \theta_t - \eta \frac{\partial \mathcal{L}}{\partial \theta_t}$$

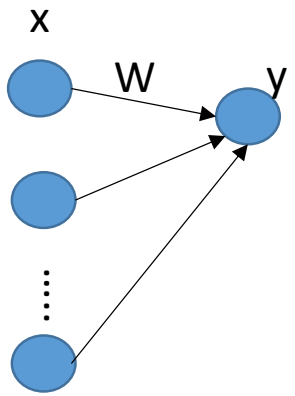
$$y = w_1 x_1 + w_2 x_2 + b,\ \mathcal{L} = (y - \hat{y})^2$$
$$\theta = \{w_1, w_2\}$$



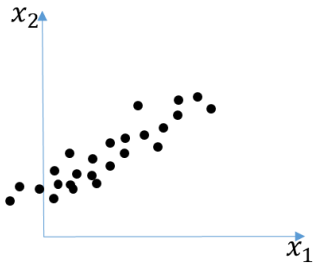
Normalizing Input Benefits Optimization

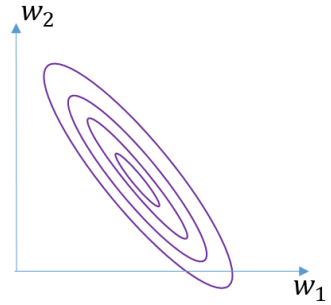
- Linear regression: $\mathcal{L}(F(\mathbf{x}),\ \hat{y}) = \frac{1}{2}(W\mathbf{x} - \hat{y})^2$

$$\mathcal{L}(W) = \frac{1}{2}(W^T \mathrm{C} \mathrm{W} - 2A^T W + b)$$



  - Where C=$\frac{1}{N}\sum_{i=1}^{N} \mathbf{x}^T\ \mathbf{x}$ is the covariance matrix
  - Gradient: $\frac{\partial \mathcal{L}}{\partial W}$=$\sum_{i=1}^{N} \mathbf{x}(y - \hat{y})$
  - Hessian matrix: $\boldsymbol{H}$=$\frac{\partial \mathcal{L}^2}{\partial W \partial W} = C$

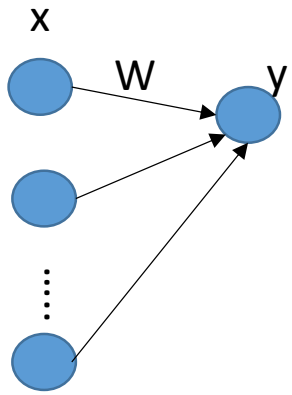




[LeCun et al, NIPS 1990; LeCun et al, NN 1998; Wiesler et al, NIPS 2011]

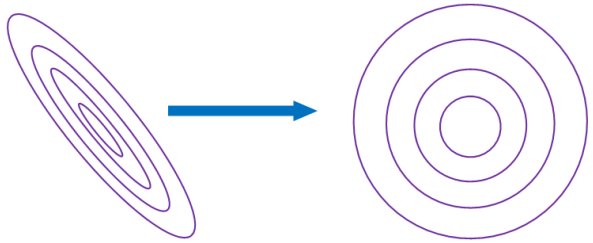
Normalizing Input Benefits Optimization

- Learning dynamics are controlled by the spectrum of curvature matrix (Hessian $\boldsymbol{H}$)
  - $\lambda_{max}(\boldsymbol{H})$:
    - Optimal learning rate: $\eta = \frac{1}{\lambda_{max}(\boldsymbol{H})}$
    - Diverge if $\eta > \frac{2}{\lambda_{max}(\boldsymbol{H})}$:
  - Condition number $\kappa = \frac{\lambda_{max}(\boldsymbol{H})}{\lambda_{min}(\boldsymbol{H})}$ control the iterations required for convergence



- Hessian of multiple output:

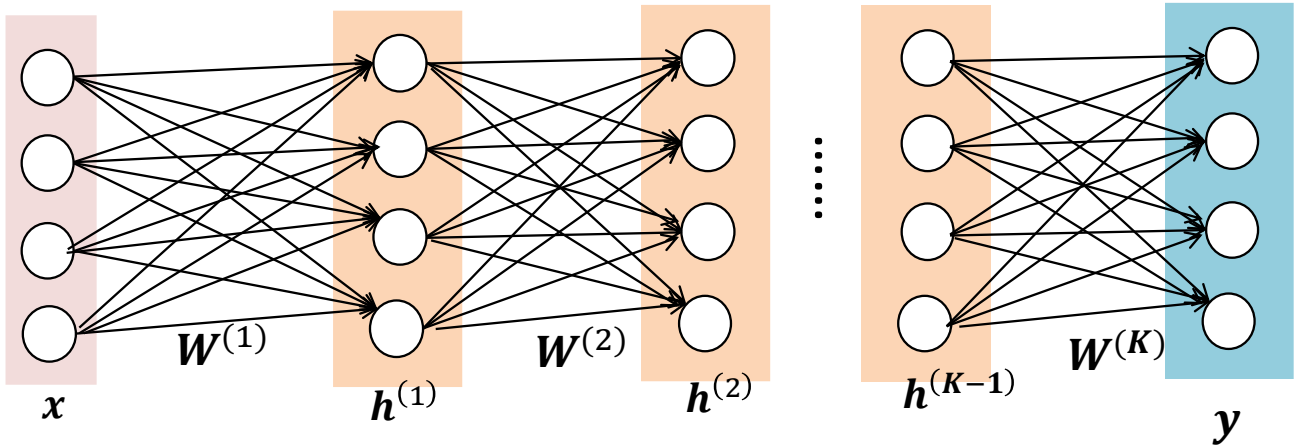
$$\boldsymbol{H} = \mathbb{E}_{\mathcal{D}}\left(\boldsymbol{x}\boldsymbol{x}^{\boldsymbol{T}}\right) \otimes \boldsymbol{I}$$



Second Order Properties of Error Surfaces: Learning Time and Generalization [Lecun et al 1990.]

Towards Normalizing Activations of DNNs

- Difficulty of analysis for DNNs
  - Nonlinear model
  - X is only linearly connected by $W^{(1)}$; Optimization is over θ, not $W^{(1)}$ only



$$\theta = \{W^{(1)}, W^{(2)}, \dots, W^{(K)}\}$$

- What we can exploit?
  - Layer-wise structure
  - $h^{(i)}$ is linearly connected by $W^{(i+1)}$

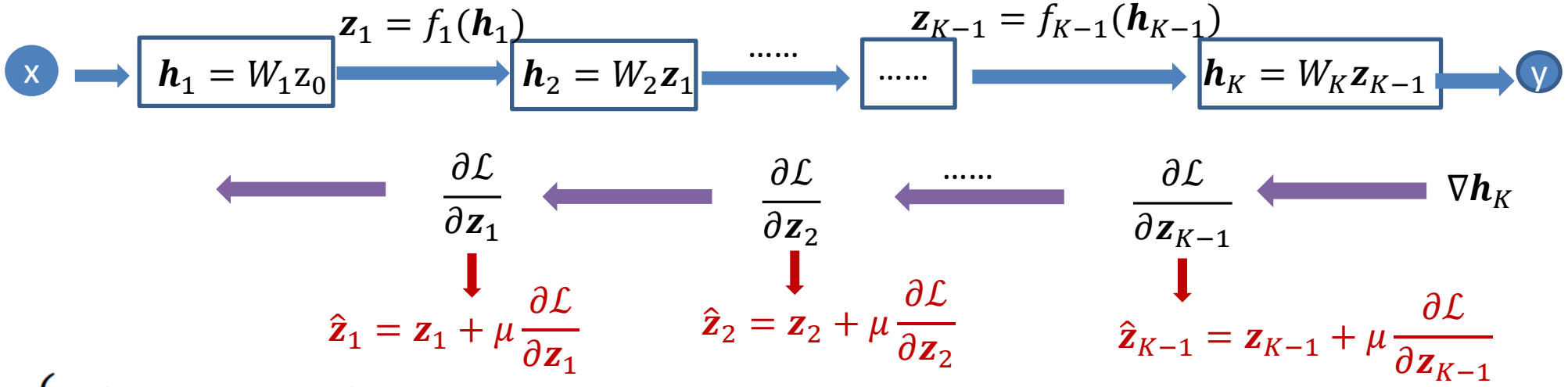
→ Normalizing Activations

Intuitive Motivation for Normalizing Activation

- Proximal back-propagation

$$\widetilde{\mathcal{L}}(\theta, \mathbf{z}) = \mathcal{L}(\mathbf{y}, f_K(\mathbf{W}_K, \mathbf{z}_{K-1})) + \sum_{k=1}^{K-1} \frac{\lambda}{2} \|\mathbf{z}_k - f_k(\mathbf{W}_k, \mathbf{z}_{k-1}))\|^2$$

- Back-match propagation



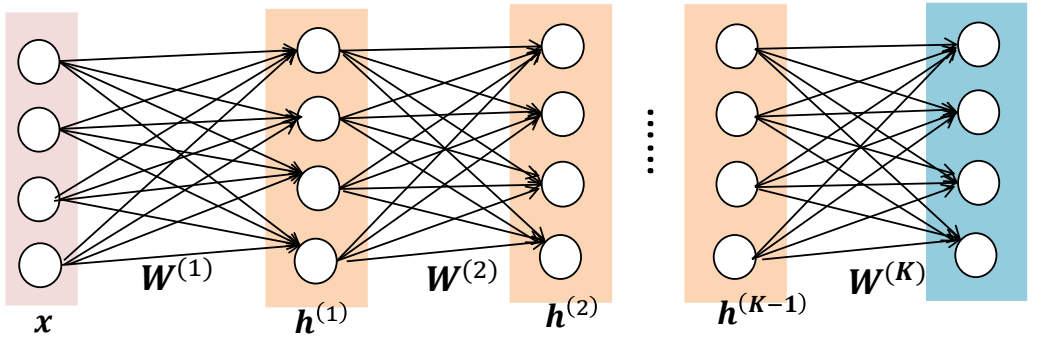
$$\begin{cases} \mathcal{L}(\mathbf{y}, \mathbf{W}_K \mathbf{z}_{K-1}), & for\ k = K \\ \frac{1}{2} \|\hat{\mathbf{z}}_k - \mathbf{W}_k \mathbf{z}_{k-1}\|^2, & for\ k = K-1, \ldots, 1. \end{cases}$$

[Frerix et al , ICLR 2018; Zhang et al, NeurIPS 2018; Huang et al, ECCV 2020]

Theoretic Analysis for Normalizing Activation

- Fisher Information Matrix (FIM): $\boldsymbol{F} = \mathbb{E}_{p(x),\, q(y|x)} \left(\frac{\partial \ell}{\partial \theta}^T \frac{\partial \ell}{\partial \theta}\right)$
- Foundation: approximating FIM using the Kronecker product (K-FAC)
  - Assumption1: weight-gradients in different layers are assumed to be uncorrelated
  - Assumption2: the input and output-gradient in each layer are approximated as independent

$$\underset{\mathbb{R}^{Kd^2 \times Kd^2}}{\mathbf{F}} = diag\left(\underset{\mathbb{R}^{d^2 \times d^2}}{F_1}, \ldots, F_K\right)$$



$$F_k = \mathbb{E}_{p(\mathbf{x}),\ q(\mathbf{y}|\mathbf{x})}\left((\mathbf{x}_k \mathbf{x}_k^T) \otimes \left(\frac{\partial \ell}{\partial \mathbf{h}_k}^T \frac{\partial \ell}{\partial \mathbf{h}_k}\right)\right) \approx \underset{\mathbb{R}^{d \times d}}{\mathbb{E}_{p(\mathbf{x})}(\mathbf{x}_k \mathbf{x}_k^T)} \otimes \mathbb{E}_{q(\mathbf{y}|\mathbf{x})}\left(\frac{\partial \ell}{\partial \mathbf{h}_k}^T \frac{\partial \ell}{\partial \mathbf{h}_k}\right)$$

Optimizing Neural Networks with Kronecker-factored Approximate Curvature [Martens and Grosse, 2015]

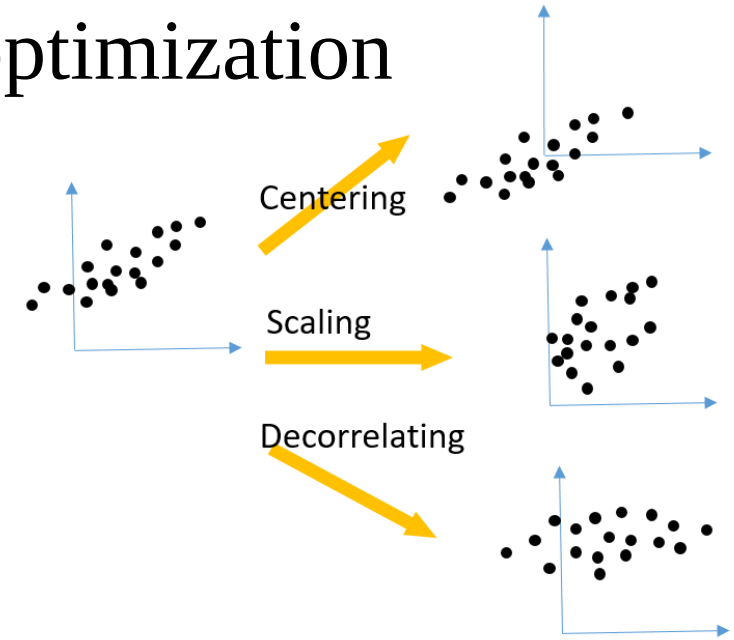
Construct Well-Conditioned Landscape

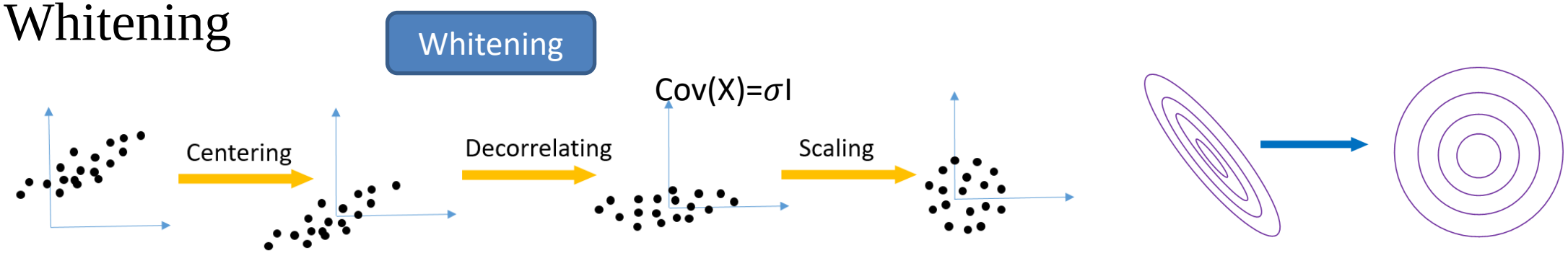
- Denoting $\Sigma_x = \mathbb{E}_{p(x)}(xx^T)$ and $\Sigma_{\nabla h} = \mathbb{E}_{p(x),q(y|x)}(\frac{\partial \ell}{\partial h}^T \frac{\partial \ell}{\partial h})$
- Criteria
  - 1. The statistics of the layer input (e.g., Σ_x) and output-gradient ($\Sigma_{\nabla h}$) across different layers are equal (across layer)
  - 2. Σ_x and $\Sigma_{\nabla h}$ are well conditioned (in layer)
- Initialization techniques: designed to satisfy Criteria 1 and/or 2 during initialization
  - Arxiv-Init [Glorot and Bengio, 2010], He-Init [He et al, 2015]: for Criteria 1
  - Orthogonal Initialization [Saxe et al, 2014] : for Criteria 1 and 2
- General goals of "normalization" in DNNs: Controlling the distribution of the activations/output-gradients during training.

Normalization Operation

- Basic normalization operations benefits the optimization
  - Centering
  - Scaling
  - Decorrelating
- Combine above
  - Standardization
  - Whitening

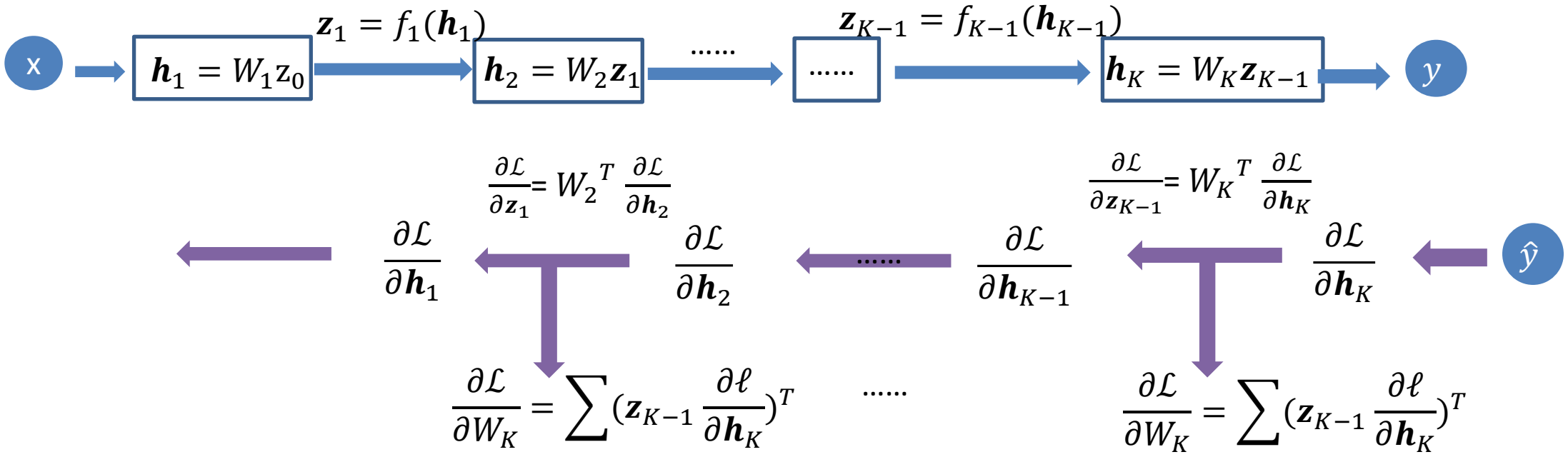
$$\boldsymbol{H} = \mathbb{E}_{\mathcal{D}}\left(\boldsymbol{x}\boldsymbol{x}^{\boldsymbol{T}}\right) \otimes \boldsymbol{I}$$



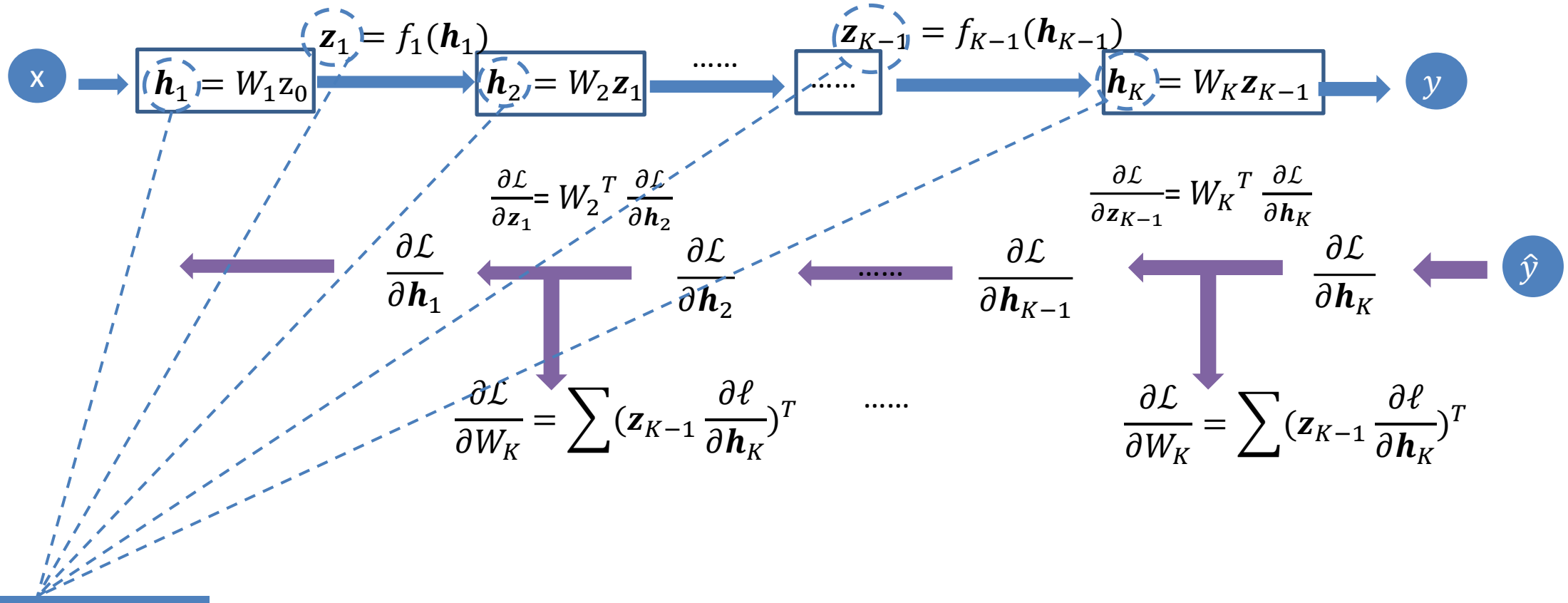


Efficient Back-propagation[Lecun et al, 1998.]

General Picture of Normalization in DNNs

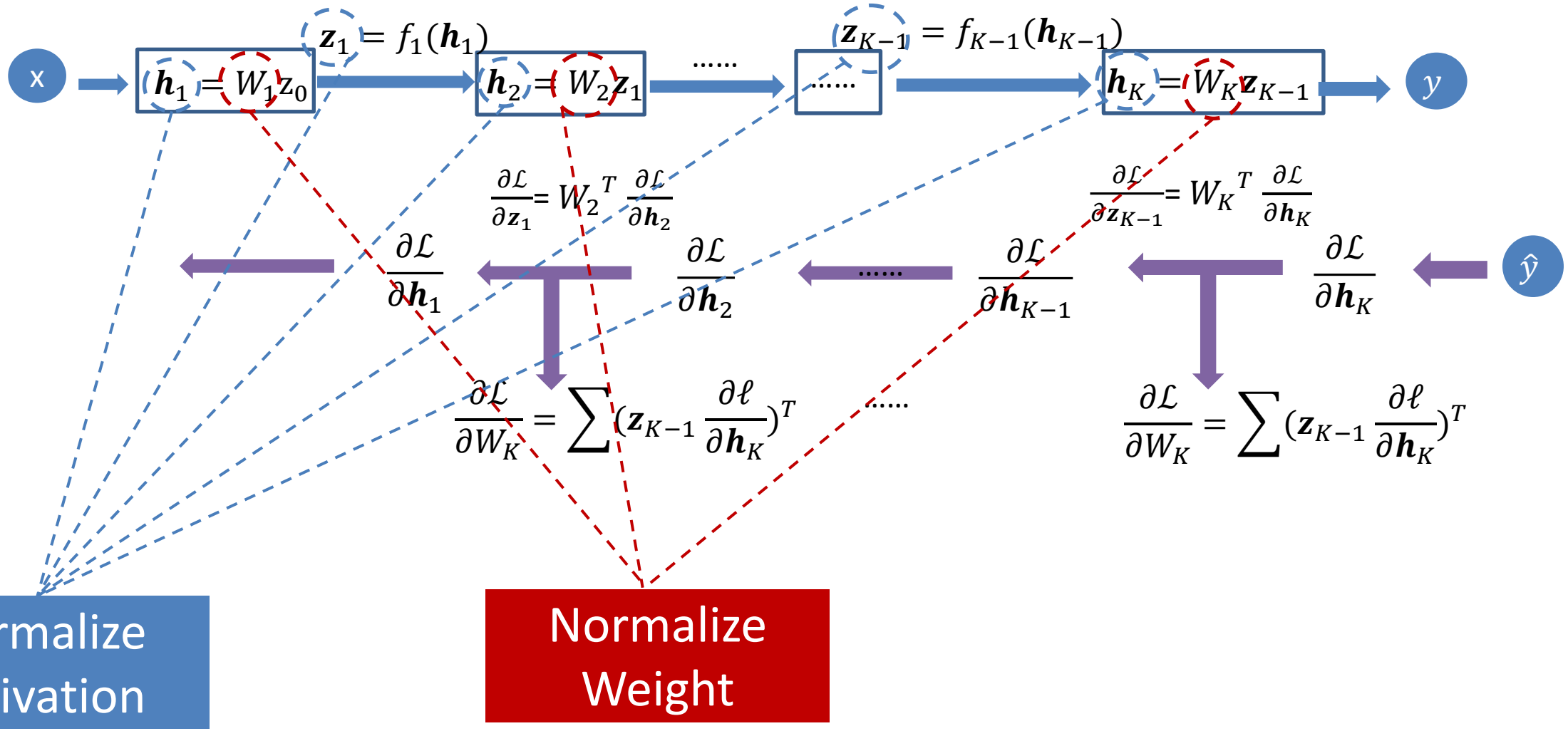


General Picture of Normalization in DNNs

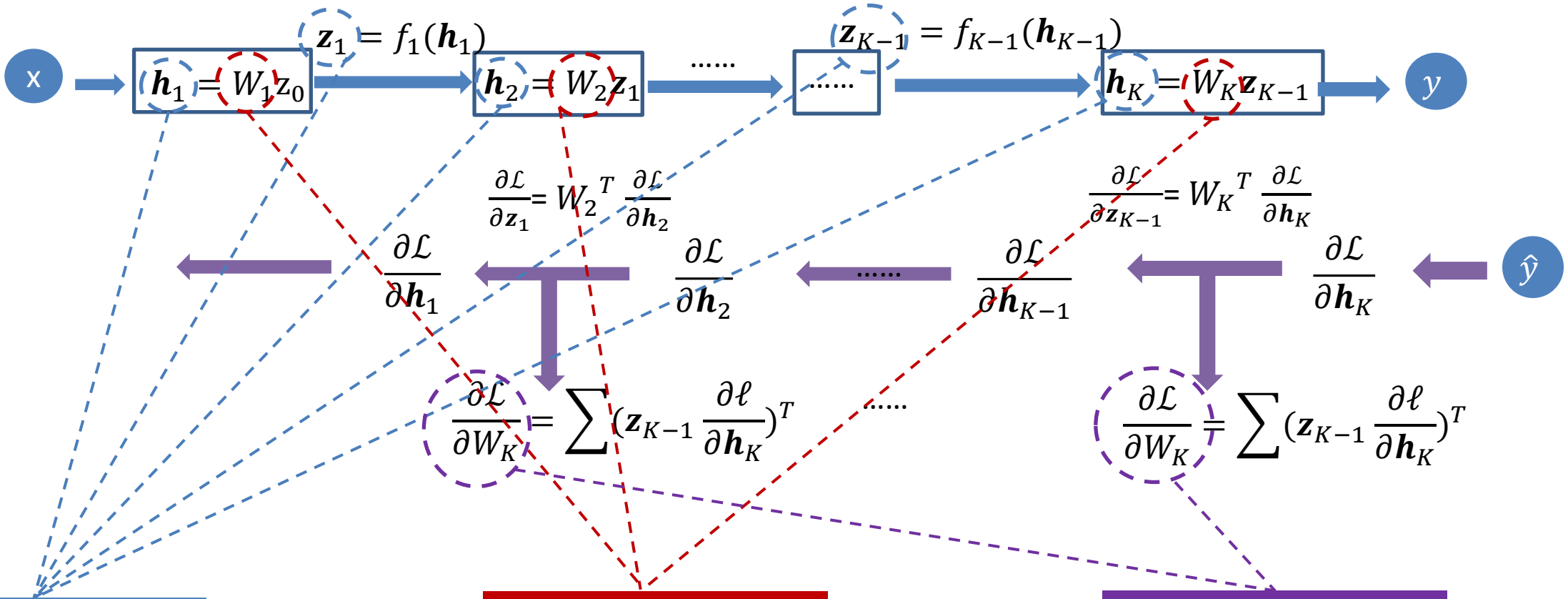


Normalize Activation

General Picture of Normalization in DNNs



General Picture of Normalization in DNNs



Normalize Activation

Normalize Weight

Normalize Gradient