Interpretable Deep Canonical Correlation Analysis Based Neural Networks

Yunlyu Wang
Matri.-Nr.:6898496

Master Thesis
Electrical System Engineering
University of Paderborn

First Examiner: Prof. Dr. Peter Schreier
Second Examiner: Prof. Dr.-Ing. Reinhold Häb-Umbach
Supervisors: Dr.-Ing. Tanuj Hasija, Maurice Kuschel

UNIVERSITÄT PADERBORN

Contents

Contents

Introduction

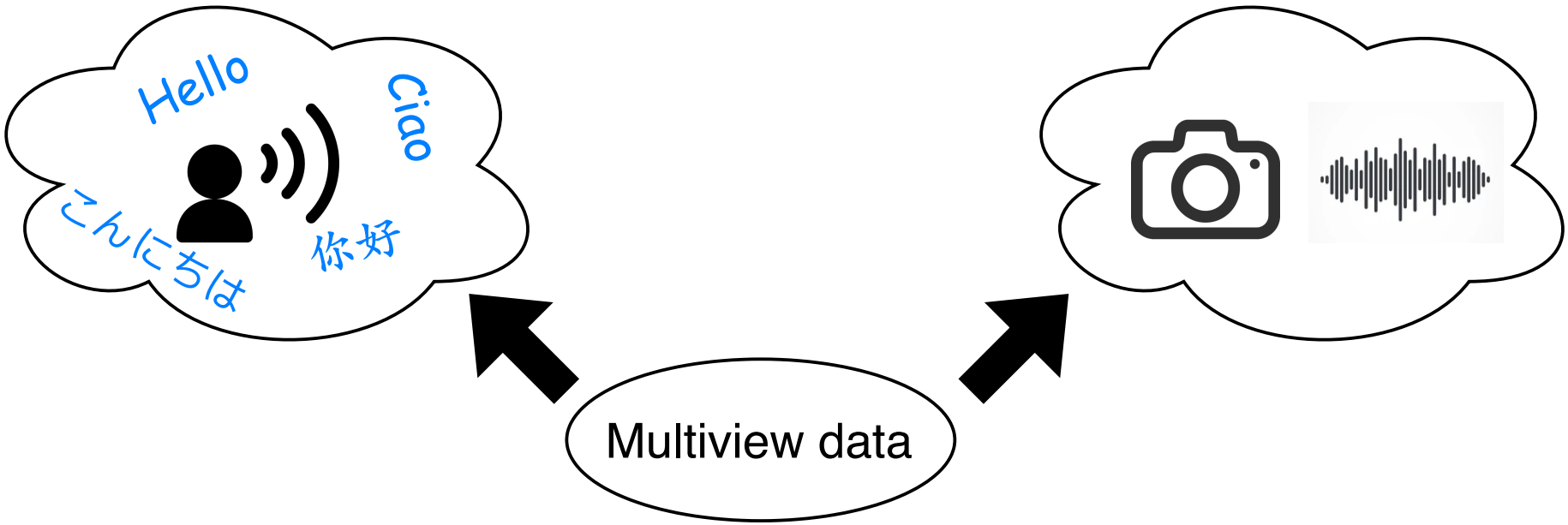
- Multi-view analysis is popular in machine learning

Introduction

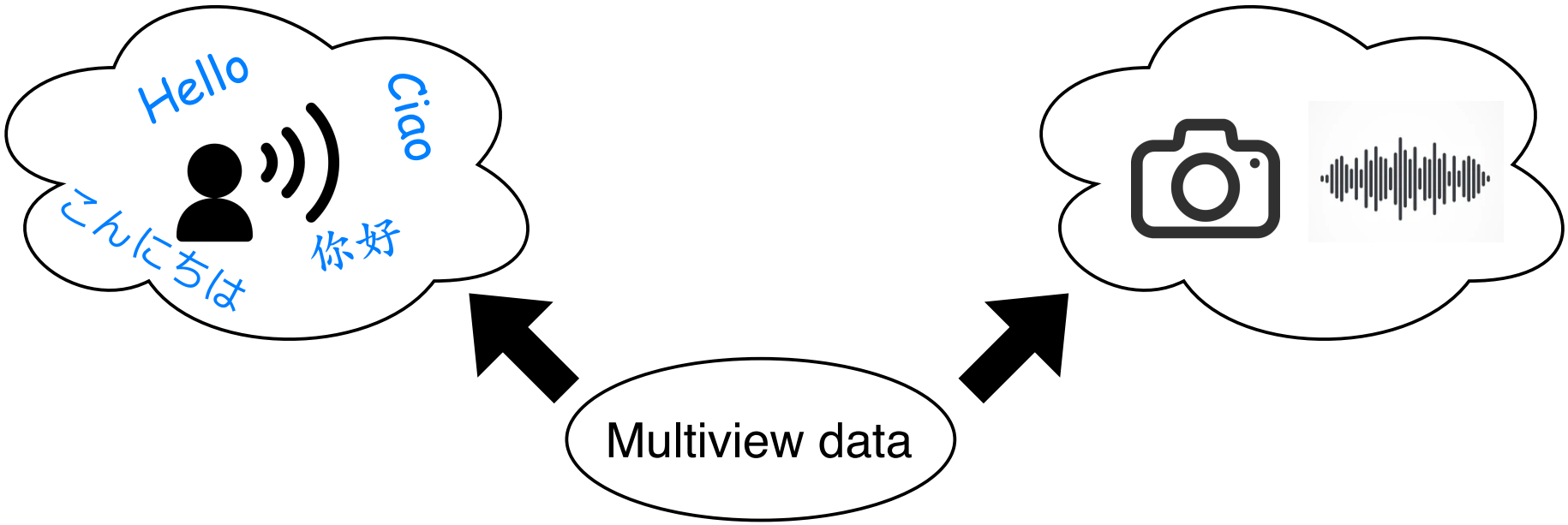
- Multi-view analysis is popular in machine learning
- Multi-view data acquires the data from different views

Introduction

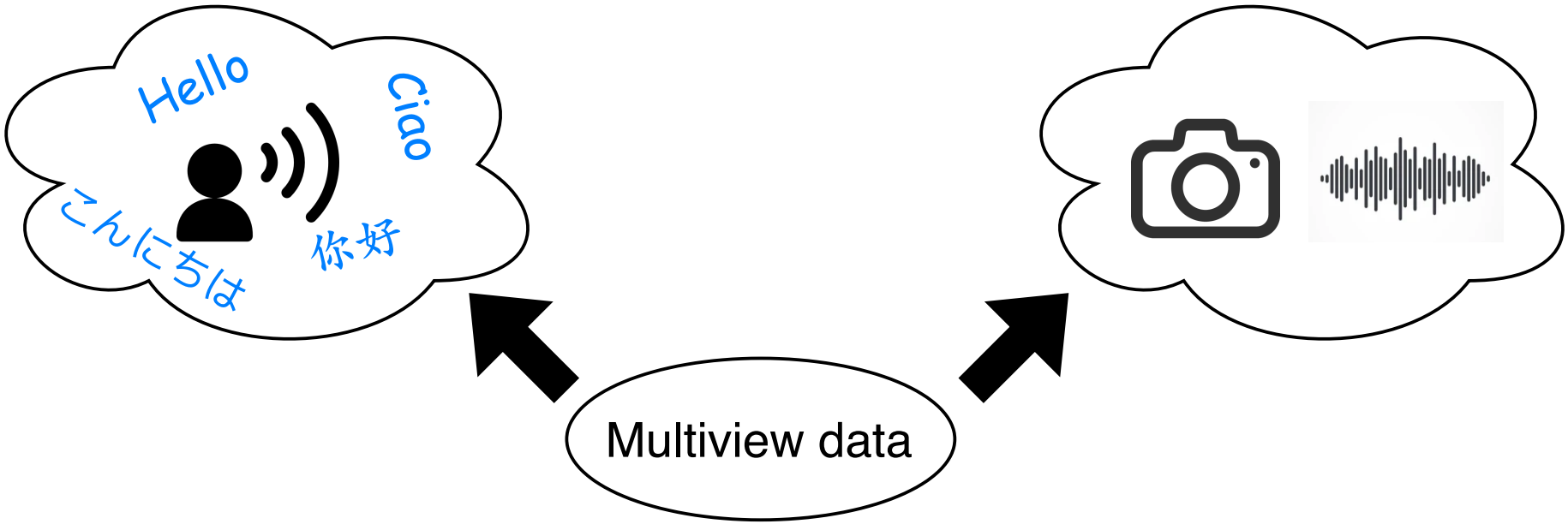
- Multi-view analysis is popular in machine learning
- Multi-view data acquires the data from different views



Introduction



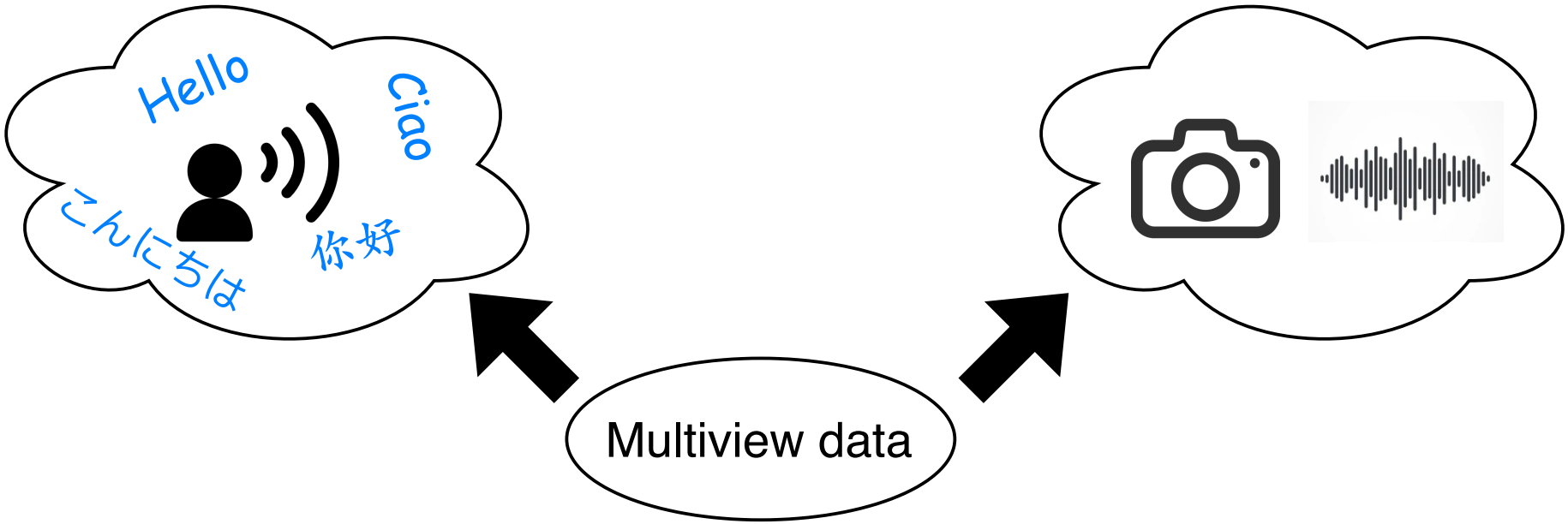
Introduction



Characteristics

- more information
- unsupervised learning
- to learn better representations

Introduction



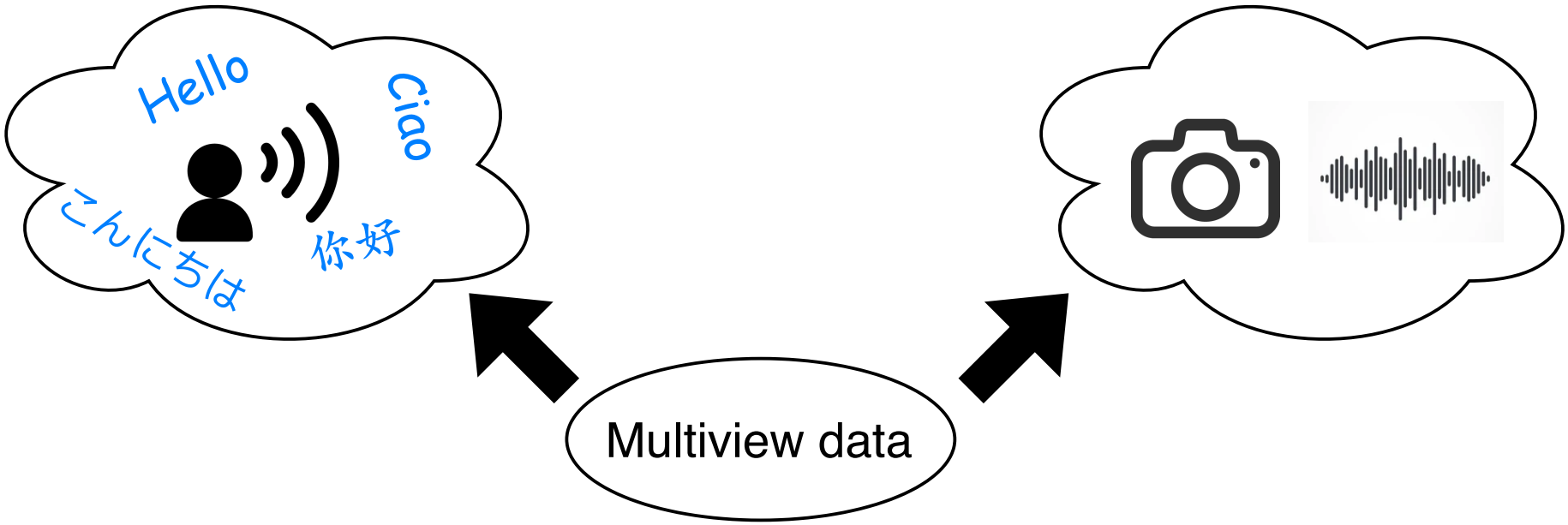
Characteristics

- more information
- unsupervised learning
- to learn better representations

Applications

- dimensionality reduction
- multi-label classification
- image denoising

Introduction



Multiview analysis

- correlation-maximization approaches

Contents

CCA

Correlation-maximization approaches:

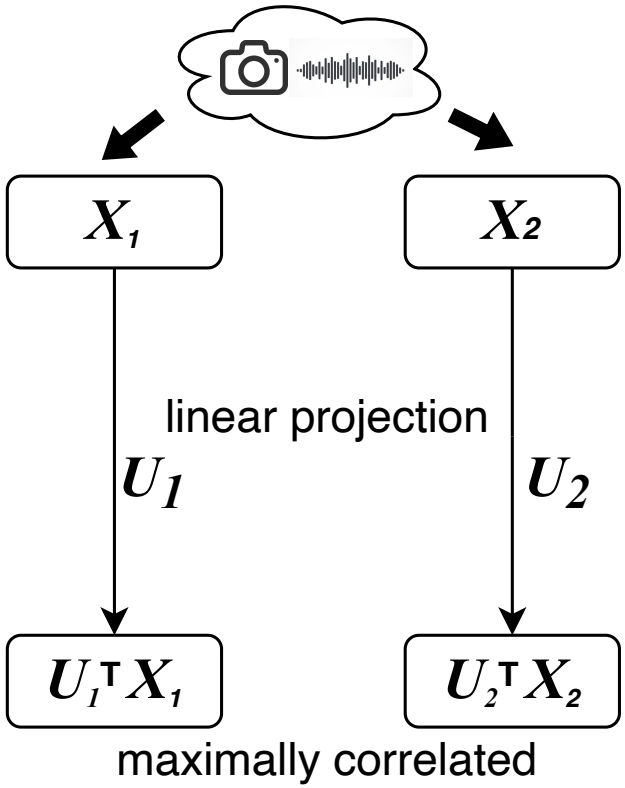
- Canonical Correlation Analysis(CCA) [1]

[1] Harold Hotelling. Relations between two sets of variates. In Breakthroughs in statistics, pages 162–190. Springer, 1992.

CCA

Correlation-maximization approaches:

- Canonical Correlation Analysis(CCA) [1]



[1] Harold Hotelling. Relations between two sets of variates. In Breakthroughs in statistics, pages 162–190. Springer, 1992.

CCA

The objective of CCA

- $\boldsymbol{X}_1 \in \mathbb{R}^{n_1 \times N}$,$\boldsymbol{X}_2 \in \mathbb{R}^{n_2 \times N}$ with covariances $\boldsymbol{\Sigma}_{11}, \boldsymbol{\Sigma}_{22}$ and cross-covariance $\boldsymbol{\Sigma}_{12}$. CCA finds the linear projections that:

$$\begin{aligned} \max_{\boldsymbol{U}_1, \boldsymbol{U}_2} \ & \operatorname{Tr}(\boldsymbol{U}_1^\intercal \boldsymbol{\Sigma}_{12} \boldsymbol{U}_2) \\ \text{s.t. } & \boldsymbol{U}_1^\intercal \boldsymbol{\Sigma}_{11} \boldsymbol{U}_1 = \boldsymbol{I} \ , \\ & \boldsymbol{U}_2^\intercal \boldsymbol{\Sigma}_{22} \boldsymbol{U}_2 = \boldsymbol{I} \ , \\ & \boldsymbol{u}_1^{(i)\intercal} \boldsymbol{X}_1 \boldsymbol{X}_2^\intercal \boldsymbol{u}_2^{(j)} = \boldsymbol{0}, \text{for } i \neq j \ . \end{aligned}$$

- When covariances matrices are ill-conditioned, they can be placed with their regularized version, e.g.

$$\boldsymbol{\Sigma}_{11} = \frac{1}{N} \boldsymbol{X}_1 \boldsymbol{X}_1^\intercal + r_1 \boldsymbol{I}$$

where r_1 is regularization parameter

CCA

The solution of CCA

Define:

$$\boldsymbol{T} \triangleq \boldsymbol{\Sigma}_{11}^{-1/2}\boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1/2}$$

Then take singular value decomposition of $\boldsymbol{T}$:

$$\boldsymbol{T} = \boldsymbol{F}\boldsymbol{K}\boldsymbol{G}^{\intercal}$$

Then the optimal objective value is:

$$(\boldsymbol{U}_1^*, \boldsymbol{U}_2^*) = (\boldsymbol{\Sigma}_{11}^{-\frac{1}{2}}\boldsymbol{F},\ \boldsymbol{\Sigma}_{22}^{-\frac{1}{2}}\boldsymbol{G})$$

DCCA

Correlation-maximization approaches:

- Canonical Correlation Analysis(CCA)

DCCA

Correlation-maximization approaches:

- Canonical Correlation Analysis(CCA) $\rightarrow$ linear transformation

DCCA

Correlation-maximization approaches:

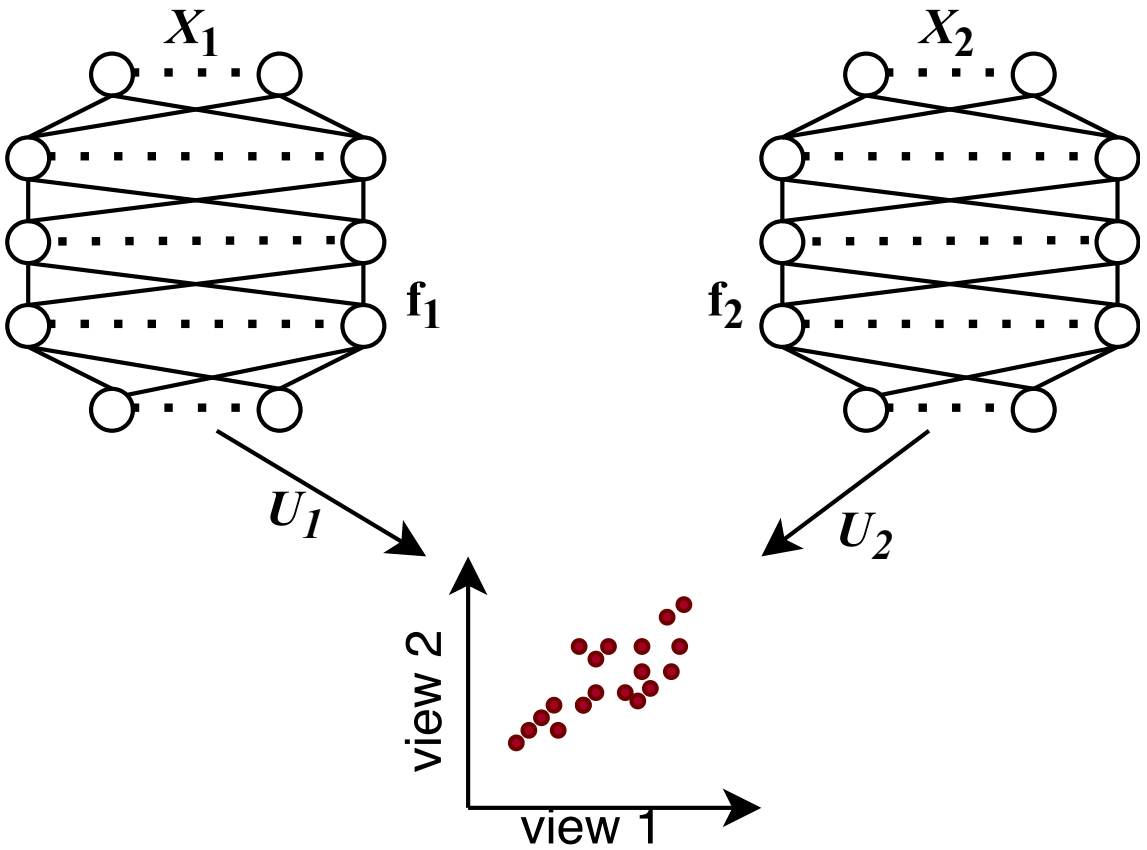
- Canonical Correlation Analysis(CCA) $\rightarrow$ linear transformation
- Deep Canonical Correlation Analysis(DCCA) [2]

[2] Andrew, Galen, et al. "Deep canonical correlation analysis." International conference on machine learning. PMLR, 2013.

DCCA

Correlation-maximization approaches:

- Canonical Correlation Analysis(CCA) $\rightarrow$ linear transformation
- Deep Canonical Correlation Analysis(DCCA) [2]



[2] Andrew, Galen, et al. "Deep canonical correlation analysis." International conference on machine learning. PMLR, 2013.

DCCA

The objective of DCCA

There are $\boldsymbol{X}_1 \in \mathbb{R}^{n\times N}$,$\boldsymbol{X}_2 \in \mathbb{R}^{n\times N}$, $\mathbf{f}_1$ and $\mathbf{f}_2$ denote mapping implemented by DNNs, with a corresponding set of learnable parameters, $\mathbf{W}_{\mathbf{f}_1}$ and $\mathbf{W}_{\mathbf{f}_2}$

$$\begin{aligned}\max_{\mathbf{W}_{\mathbf{f}_1},\mathbf{W}_{\mathbf{f}_2},\boldsymbol{U}_1,\boldsymbol{U}_2} \ & \frac{1}{N}\operatorname{Tr}(\boldsymbol{U}_1^{\intercal}\mathbf{f}_1(\boldsymbol{X}_1)\mathbf{f}_2(\boldsymbol{X}_2)^{\intercal}\boldsymbol{U}_2)\\ \text{s.t. } & \boldsymbol{U}_1^{\intercal}(\frac{1}{N}\mathbf{f}_1(\boldsymbol{X}_1)\mathbf{f}_1(\boldsymbol{X}_1)^{\intercal}+r_1\boldsymbol{I})\boldsymbol{U}_1=\boldsymbol{I}\ ,\\ & \boldsymbol{U}_2^{\intercal}(\frac{1}{N}\mathbf{f}_2(\boldsymbol{X}_2)\mathbf{f}_2(\boldsymbol{X}_2)^{\intercal}+r_2\boldsymbol{I})\boldsymbol{U}_2=\boldsymbol{I}\ ,\\ & \boldsymbol{u}_1^{(i)\intercal}\mathbf{f}_1(\boldsymbol{X}_1)\mathbf{f}_2(\boldsymbol{X}_2)^{\intercal}\boldsymbol{u}_2^{(j)}=0, \text{for } i\neq j\ .\end{aligned}$$

where $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$ are the CCA projections that project the DNN outputs, r_1 and r_2 are regularization parameters for sample covariance estimation.

DCCAE

Correlation-maximization approaches:

- Canonical Correlation Analysis(CCA)
- Deep Canonical Correlation Analysis(DCCA)

DCCAE

Correlation-maximization approaches:

- Canonical Correlation Analysis(CCA)
- Deep Canonical Correlation Analysis(DCCA)
  - shared representation $\rightarrow^{?}$ reconstruct original views

DCCAE

Correlation-maximization approaches:

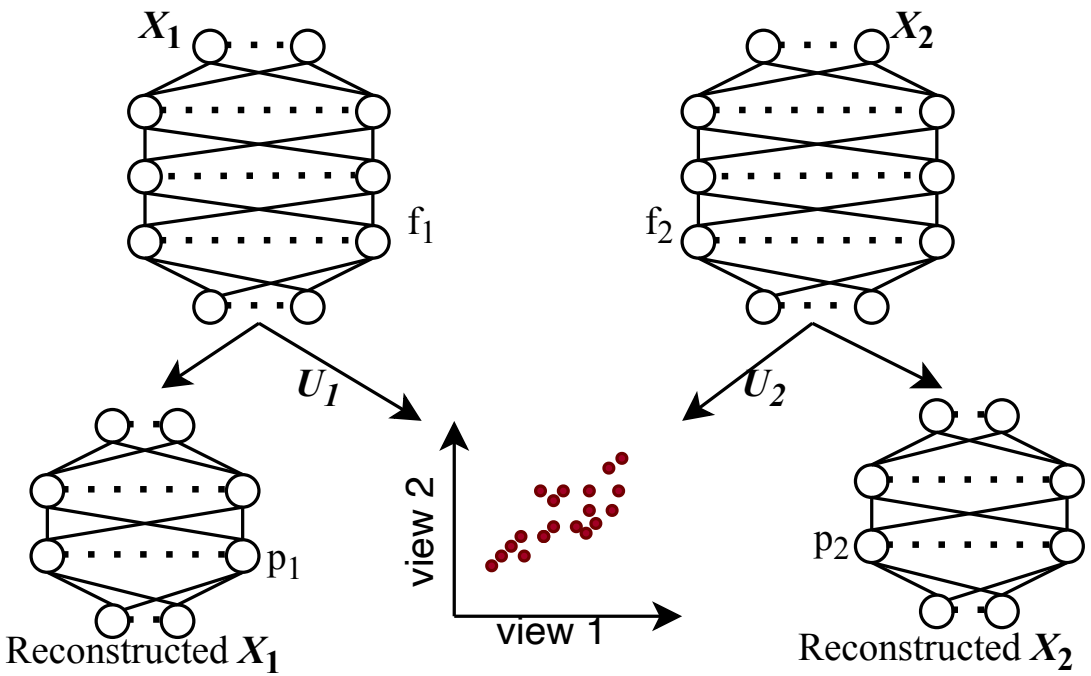
- Canonical Correlation Analysis(CCA)
- Deep Canonical Correlation Analysis(DCCA)
  - shared representation $\rightarrow^{?}$ reconstruct original views
- Deep Canonically Correlated Autoencoders(DCCAE)[3]

[3] Wang, Weiran, et al. " On deep multi-view representation learning." International conference on machine learning. PMLR, 2015.

DCCAE

Correlation-maximization approaches:

- Canonical Correlation Analysis(CCA)
- Deep Canonical Correlation Analysis(DCCA)
  - shared representation $\rightarrow^{?}$ reconstruct original views
- Deep Canonically Correlated Autoencoders(DCCAE)[3]



[3] Wang, Weiran, et al. " On deep multi-view representation learning." International conference on machine learning. PMLR, 2015.

DCCAE

The objective of DCCAE

There are two view data $\boldsymbol{X}_1 \in \mathbb{R}^{n\times N}$, $\boldsymbol{X}_2 \in \mathbb{R}^{n\times N}$. $\mathbf{f}_1$, $\mathbf{f}_2$ are the two DNNs of finding the maximal correlated outputs, and $\mathbf{p}_1$, $\mathbf{p}_2$ denote the two DNNs of reconstruction, with a corresponding set of learnable parameters, $\mathbf{W}_{\mathbf{f}_1}$, $\mathbf{W}_{\mathbf{f}_2}$, $\mathbf{W}_{\mathbf{p}_1}$ and $\mathbf{W}_{\mathbf{p}_2}$.

$$\min_{\mathbf{W}_{\mathbf{f}_1},\mathbf{W}_{\mathbf{f}_2},\mathbf{W}_{\mathbf{p}_1},\mathbf{W}_{\mathbf{p}_2},\boldsymbol{U}_1,\boldsymbol{U}_2} \quad -\frac{1}{N}\operatorname{Tr}(\boldsymbol{U}_1^\intercal \mathbf{f}_1(X_1)\mathbf{f}_2(X_2)^\intercal \boldsymbol{U}_2)$$
$$+\frac{\lambda}{N}\sum_{i=1}^{N}(\|\boldsymbol{x}_{1i}-\mathbf{p}_1(\mathbf{f}_1(\boldsymbol{x}_{1i}))\|^2+\|\boldsymbol{x}_{2i}-\mathbf{p}_2(\mathbf{f}_2(\boldsymbol{x}_{2i}))\|^2)$$
$$\text{s.t. the same constraints in DCCA ,}$$

where $\lambda > 0$ is a trade-off parameter.

Motivation

Disadvantages

Motivation

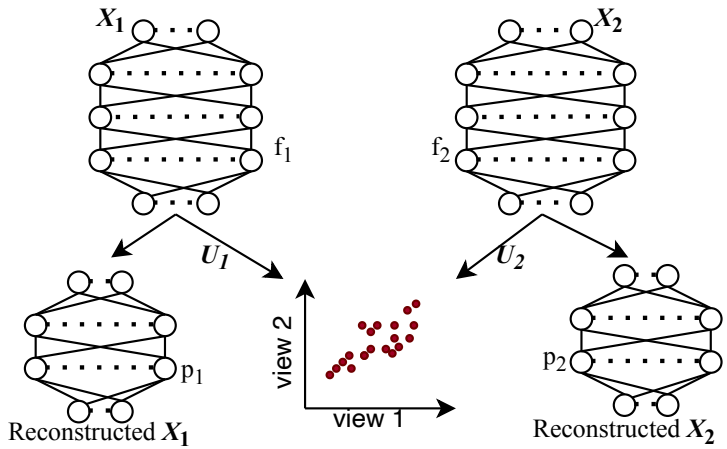
Disadvantages

- DCCAE based techniques

Motivation

Disadvantages

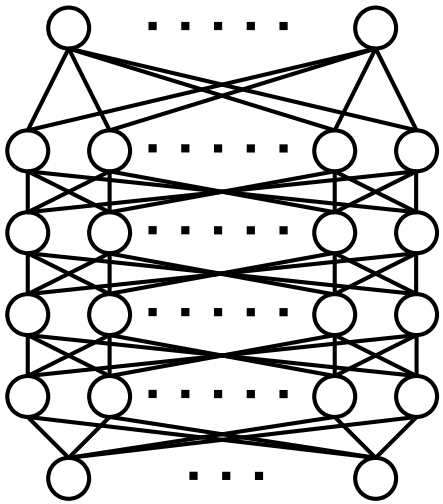
- DCCAE based techniques
  - feedforward neural network



Motivation

Disadvantages

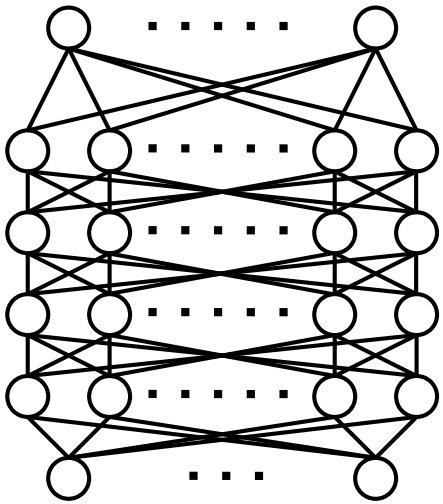
- DCCAE based techniques
  - feedforward neural network
  - not suitable for images



Motivation

Disadvantages

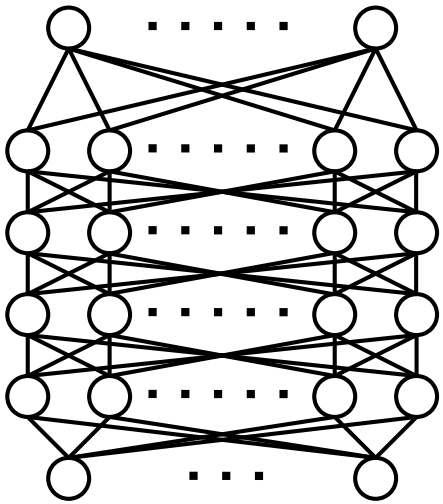
- DCCAE based techniques
  - feedforward neural network
  - not suitable for images
- Correlation $\rightarrow 1 \Rightarrow$ Overfitting



Motivation

Disadvantages

- DCCAE based techniques
  - feedforward neural network
  - not suitable for images
- Correlation $\rightarrow 1 \Rightarrow$ Overfitting
- Not interpretable



Motivation

Disadvantages

- DCCAE based techniques
  - feedforward neural network
  - not suitable for images
- Correlation $\rightarrow 1 \Rightarrow$ Overfitting
- Not interpretable

This thesis proposes

- CNN-based DCCAE model

Motivation

Disadvantages

- DCCAE based techniques
  - feedforward neural network
  - not suitable for images
- Correlation $\rightarrow 1 \Rightarrow$ Overfitting
- Not interpretable

This thesis proposes

- CNN-based DCCAE model
- Analysing overfitting

Motivation

Disadvantages

- DCCAE based techniques
  - feedforward neural network
  - not suitable for images
- Correlation $\rightarrow 1 \Rightarrow$ Overfitting
- Not interpretable

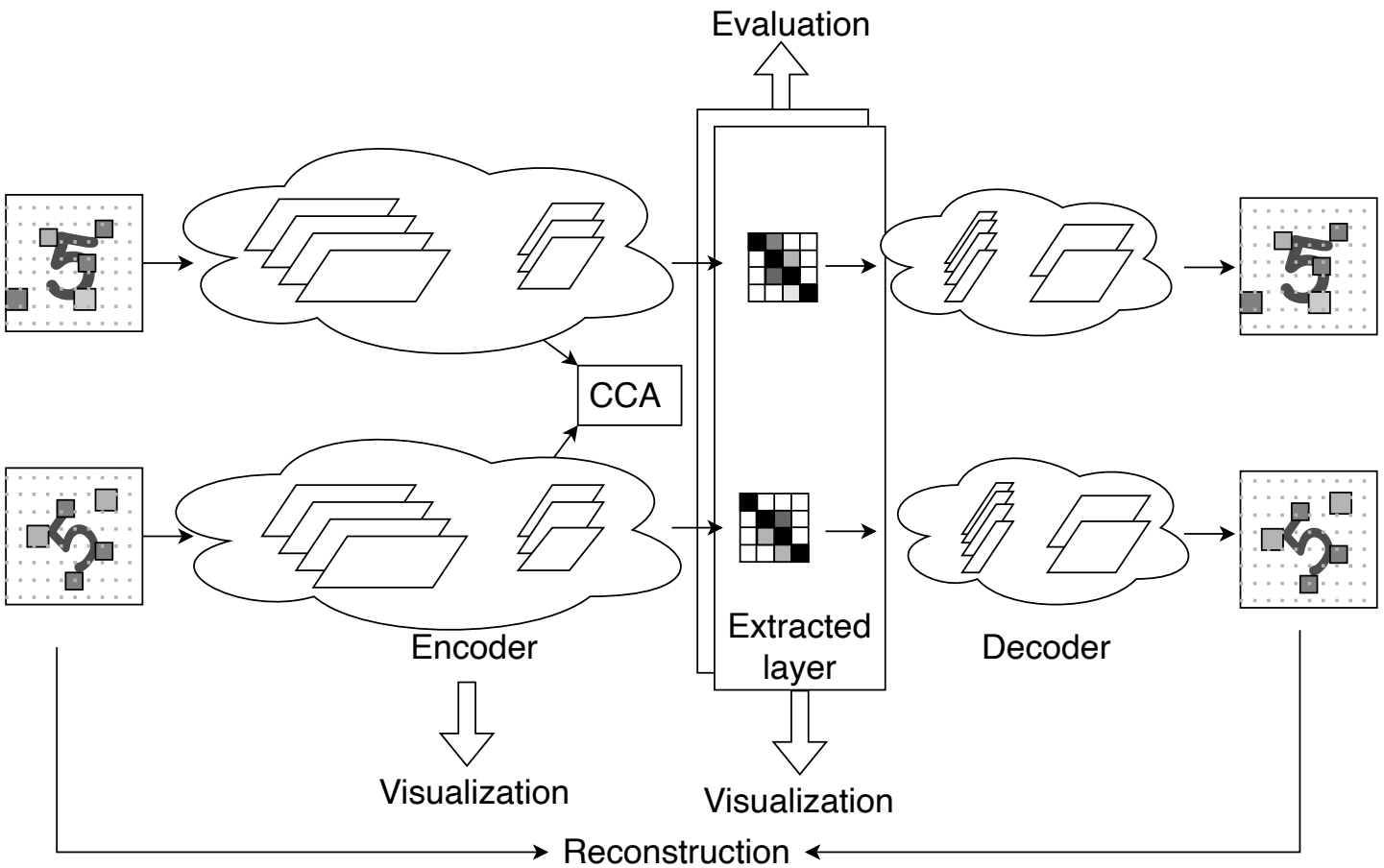
This thesis proposes

- CNN-based DCCAE model
- Analysing overfitting
- Interpretability

Contents

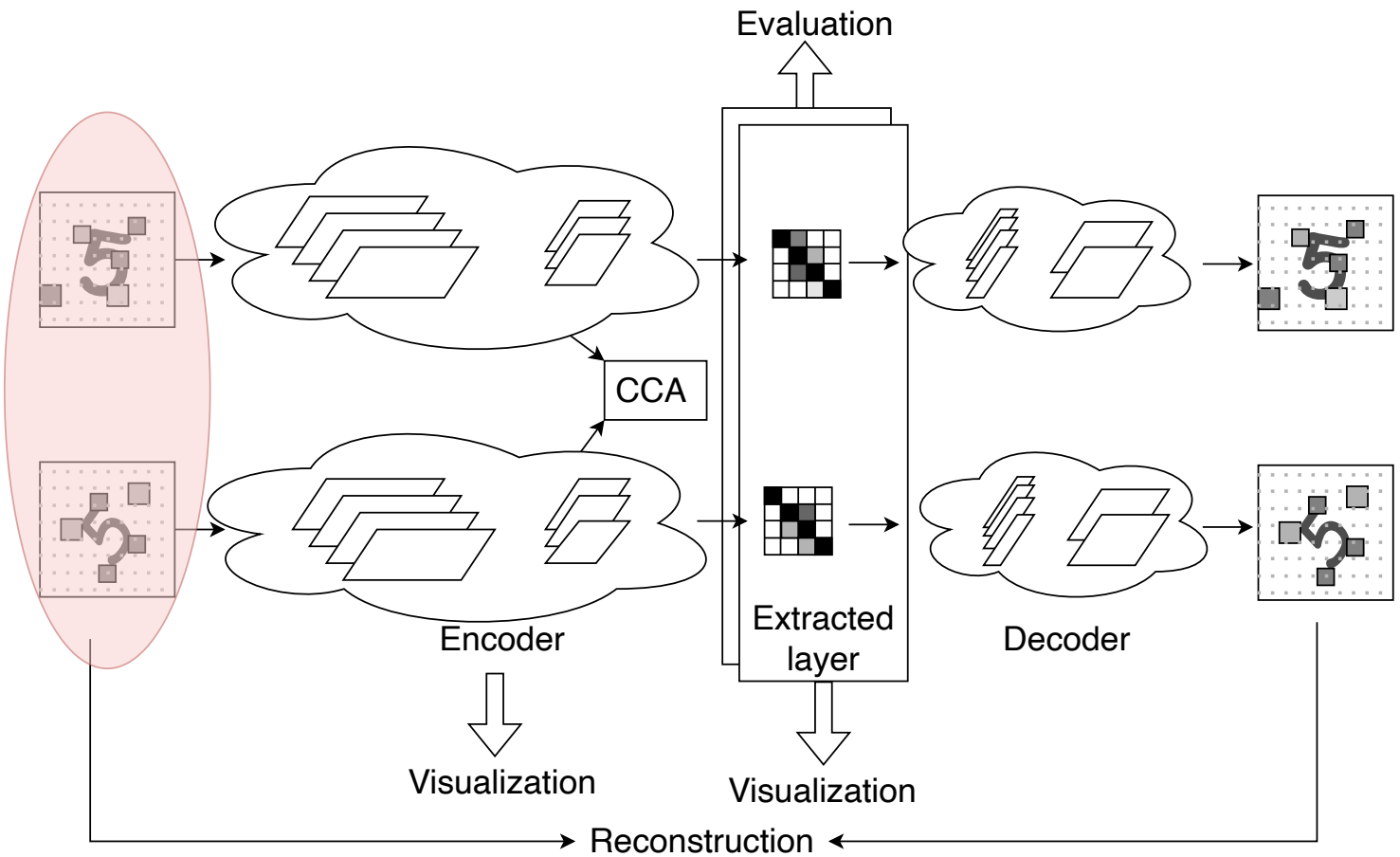
Overview

The CNN-based DCCAE network model



Proposed Pipeline

The CNN-based DCCAE network model



Data Generation

- Inputs: MNIST dataset

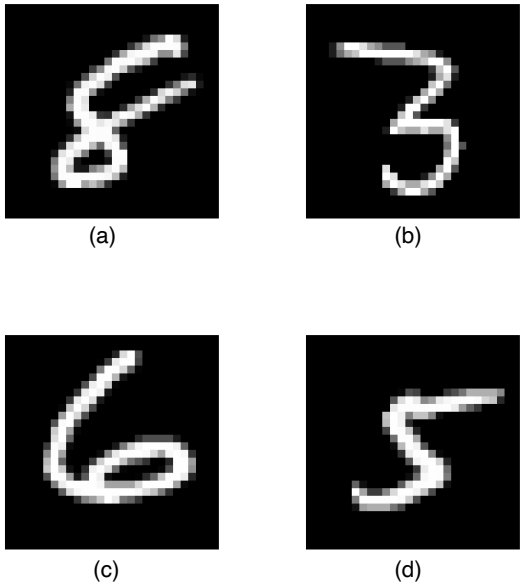


Figure: MNIST

Data Generation

- Inputs: MNIST dataset with some box noise

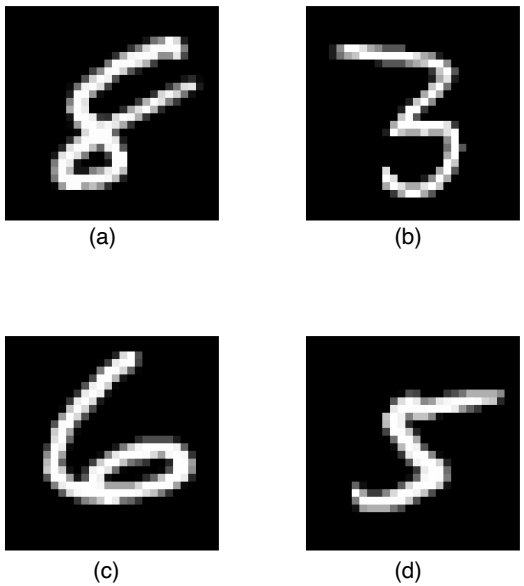


Figure: MNIST

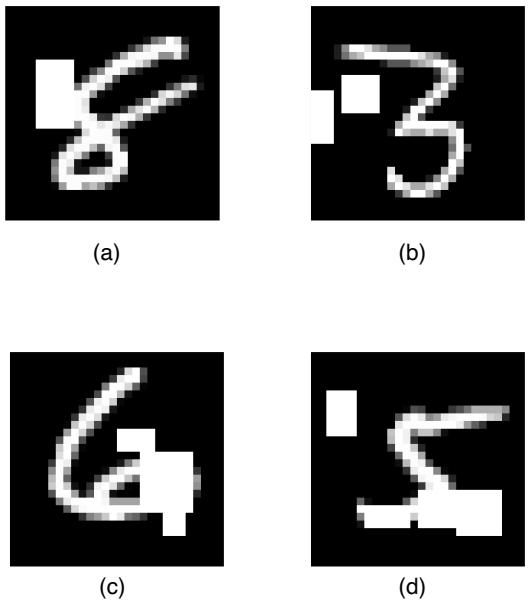


Figure: MNIST with Box Nosie

Data Generation

- Inputs: MNIST dataset with some box noise

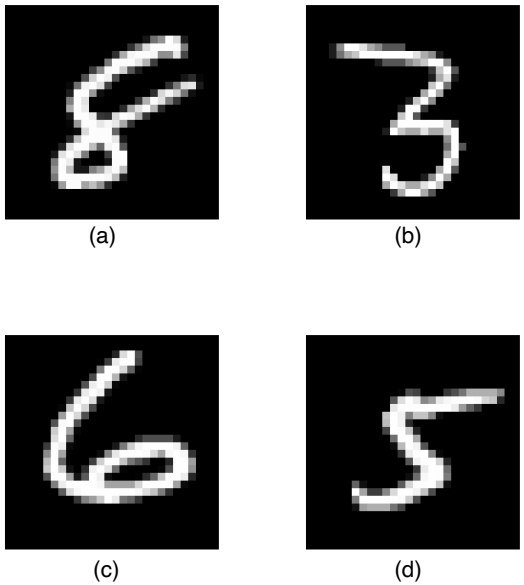


Figure: MNIST

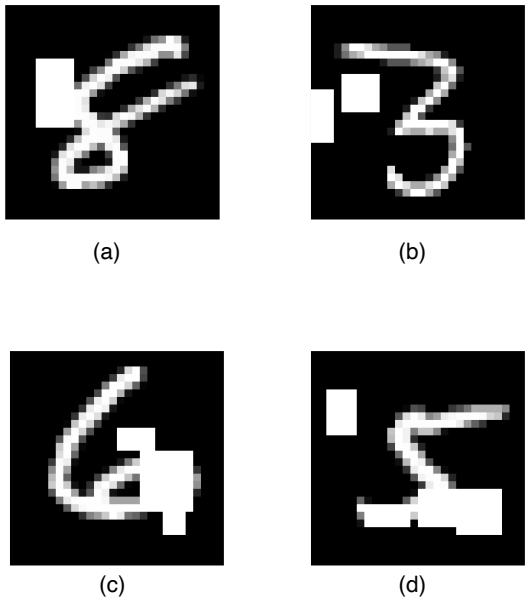


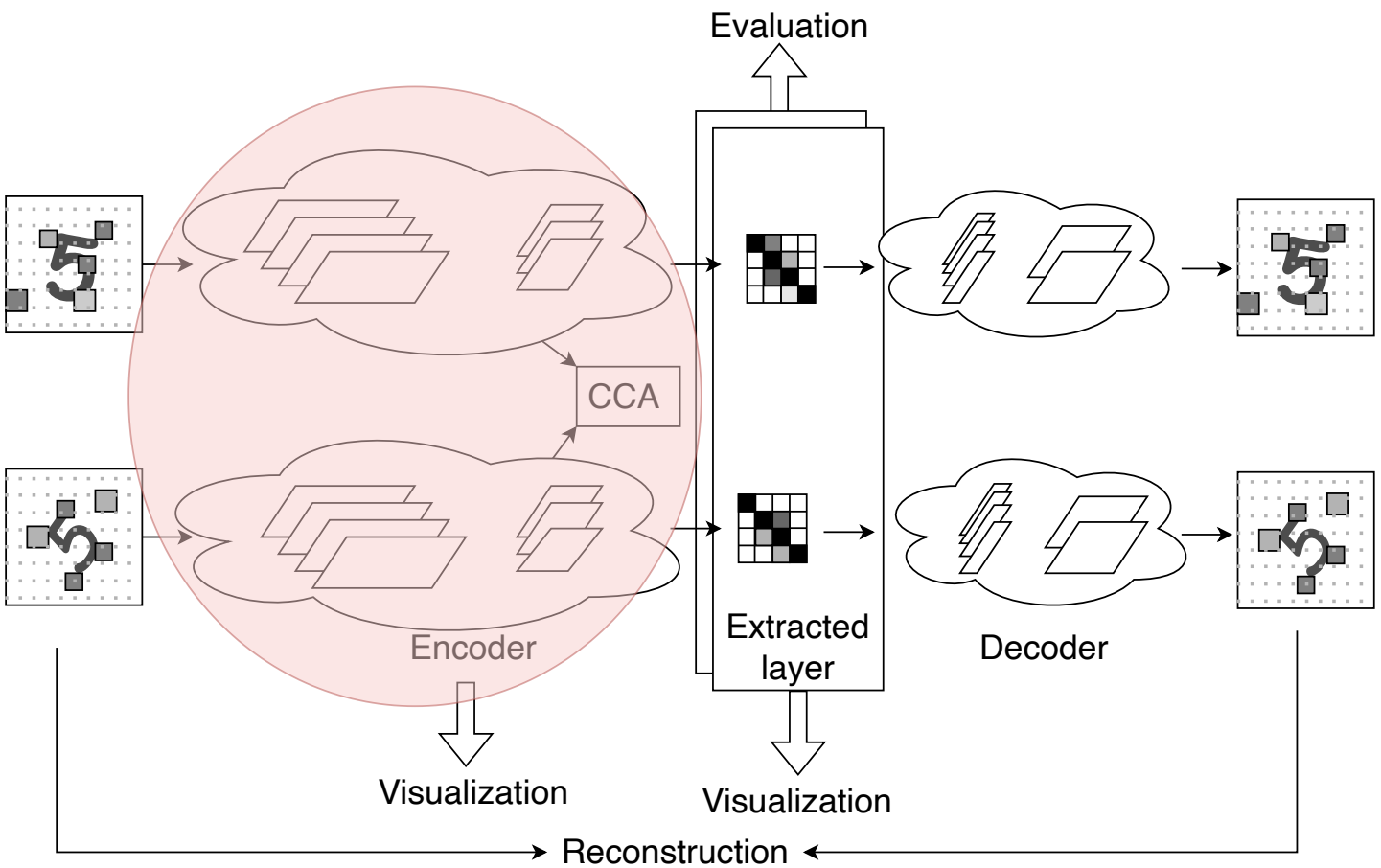
Figure: MNIST with Box Nosie

The character of the data

- The number of boxes is customizable
- The size and the position of boxes are random in each examples of both views

Proposed Pipeline

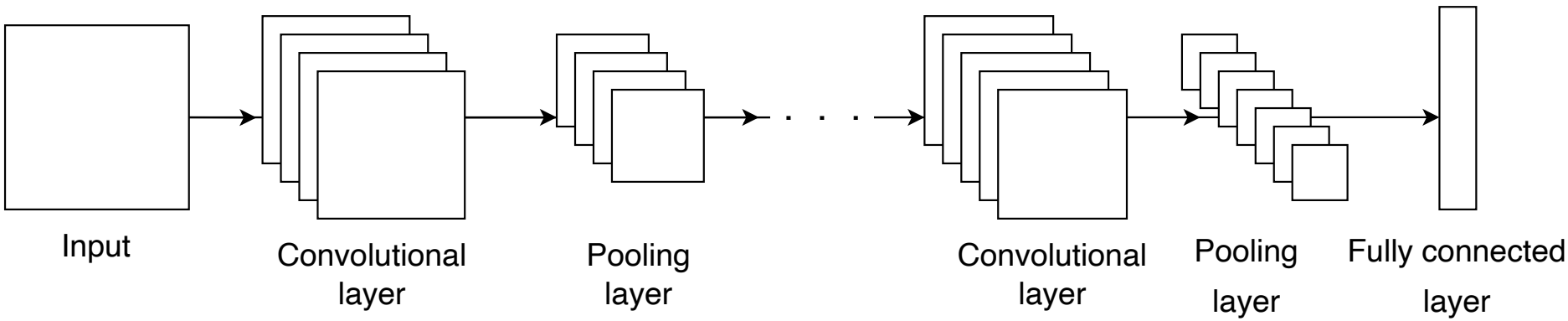
The CNN-based DCCAE network model



Convolution Neural Network

Convolution neural network [4] is mainly used for applications in image and speech recognition

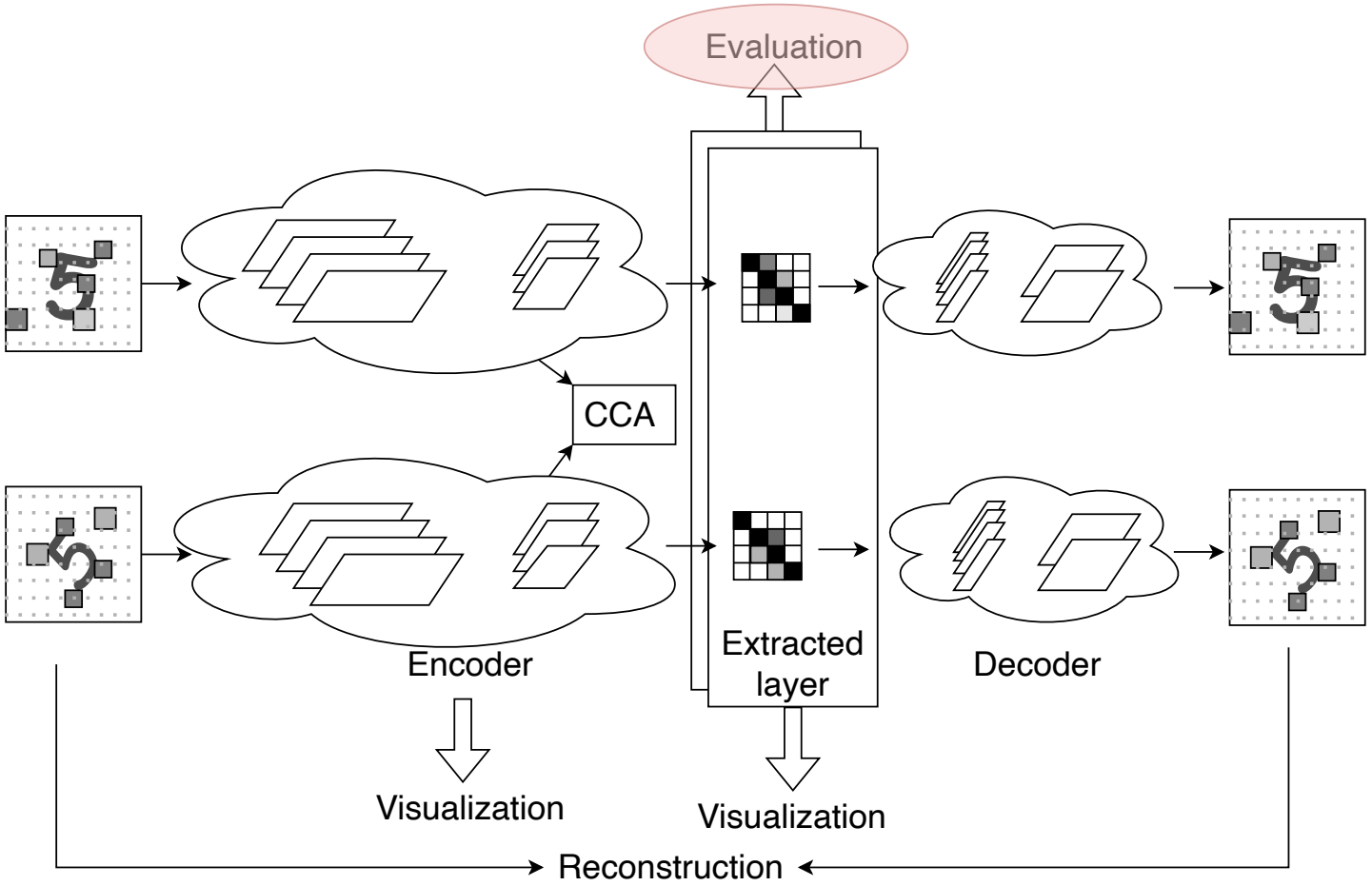
- small number of weights (shared)
- reduce high dimensionality
- without losing image information



[4] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient- based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

Proposed Pipeline

The CNN-based DCCAE network model



Evaluation

Accuracy of the classification $\Longrightarrow$ Performance of the model.

Evaluation

Accuracy of the classification $\Longrightarrow$ Performance of the model.

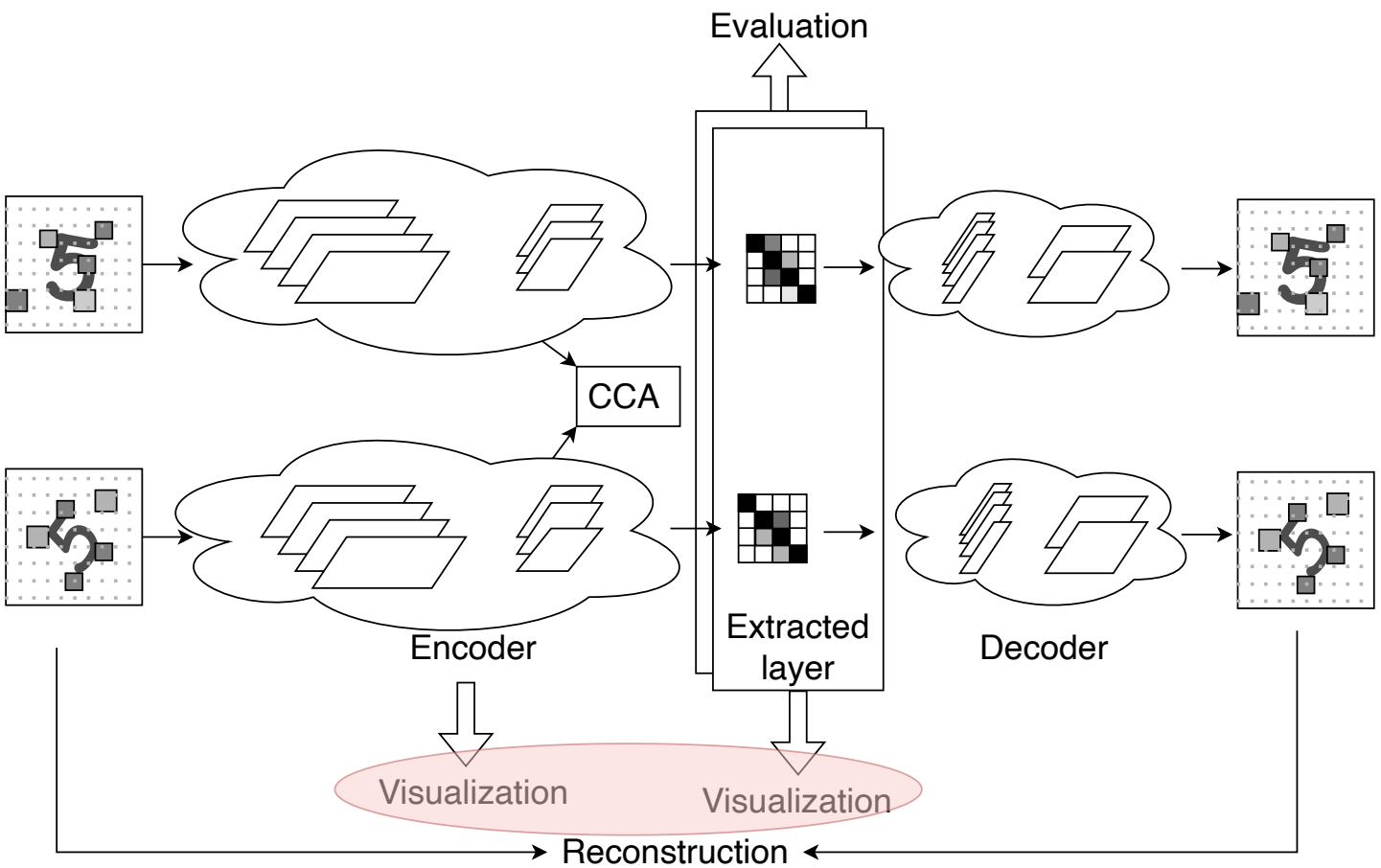
- Support vector machines (SVM) [5]
- K-means clustering [6]

[5] Vladimir N Vapnik. Support vector networks. Machine learning, 20(3):273–297, 1995.

[6] Anil K Jain and Richard C Dubes. Algorithms for clustering data. Prentice-Hall, Inc., 1988.

Proposed Pipeline

The CNN-based DCCAE network model



Visualization

Visualization of neural network $\Longrightarrow$ which parts of input the network learns

Visualization

Visualization of neural network $\Longrightarrow$ which parts of input the network learns

- Saliency Map [7]
- SmoothGrad [8]
- Gradient-weighted Class Activation Mapping(GradCAM) [9]

[7] Niels JS Morch, Ulrik Kjems, Lars Kai Hansen, Claus Svarer, Ian Law, Benny Lautrup, Steve Strother, and Kelly Rehm. Visualization of neural networks using saliency maps. In Proceedings of ICNN'95-International Conference on Neural Networks, volume 4, pages 2085–2090. IEEE, 1995.

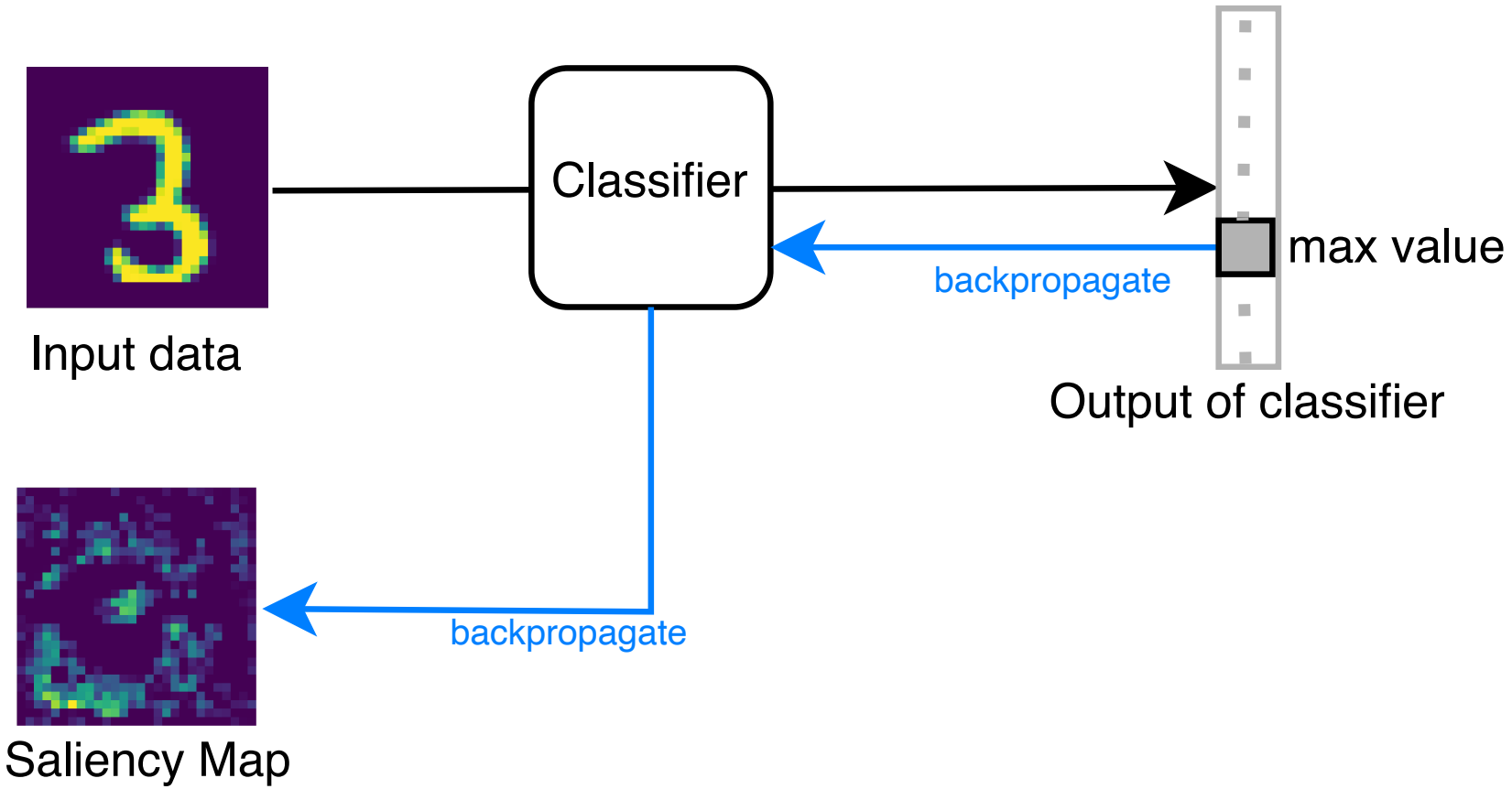
[8] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. arXiv preprint arXiv:1706.03825, 2017.

[9] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE international conference on computer vision, pages 618–626, 2017.

Contents

Saliency Map [7]

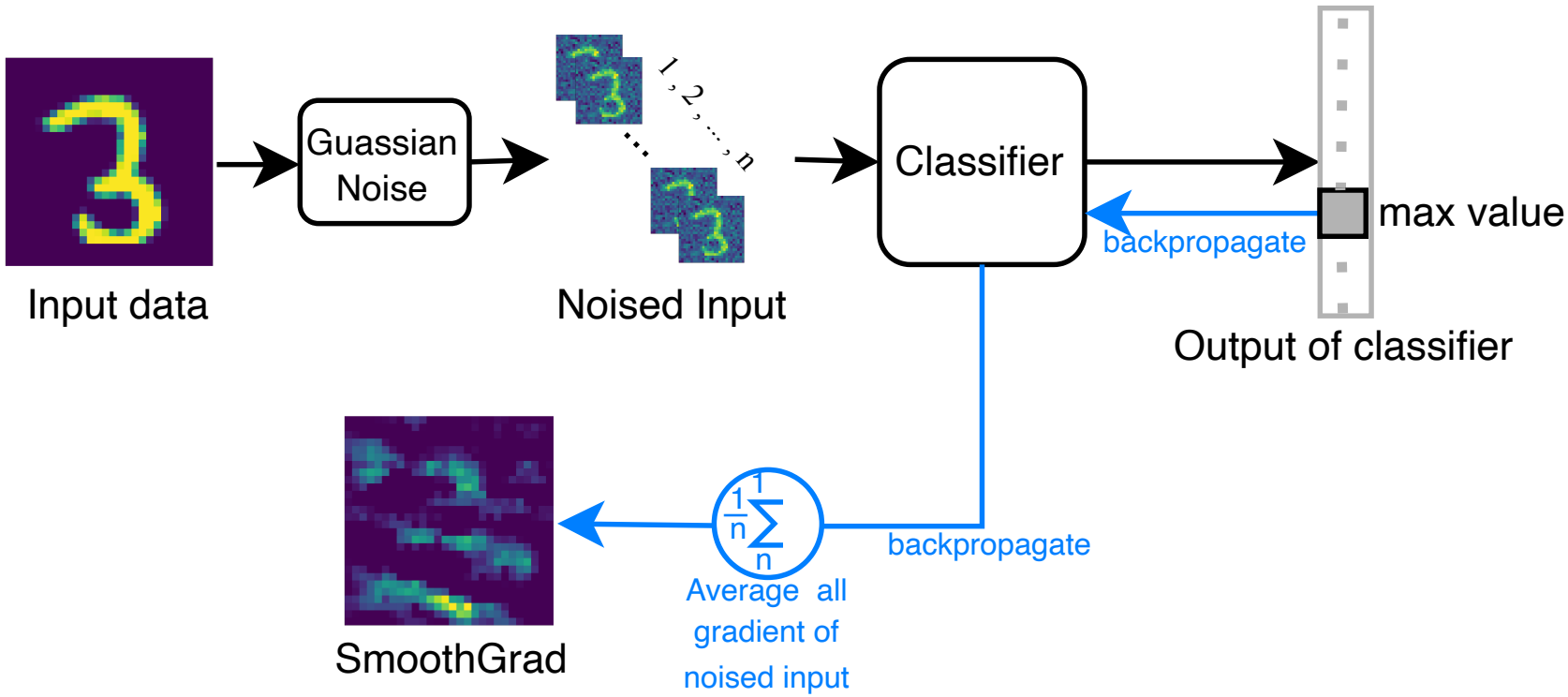
Workflow



[7] Niels JS Morch, Ulrik Kjems, Lars Kai Hansen, Claus Svarer, Ian Law, Benny Lautrup, Steve Strother, and Kelly Rehm. Visualization of neural networks using saliency maps. In Proceedings of ICNN'95-International Conference on Neural Networks, volume 4, pages 2085–2090. IEEE, 1995.

SmoothGrad [8]

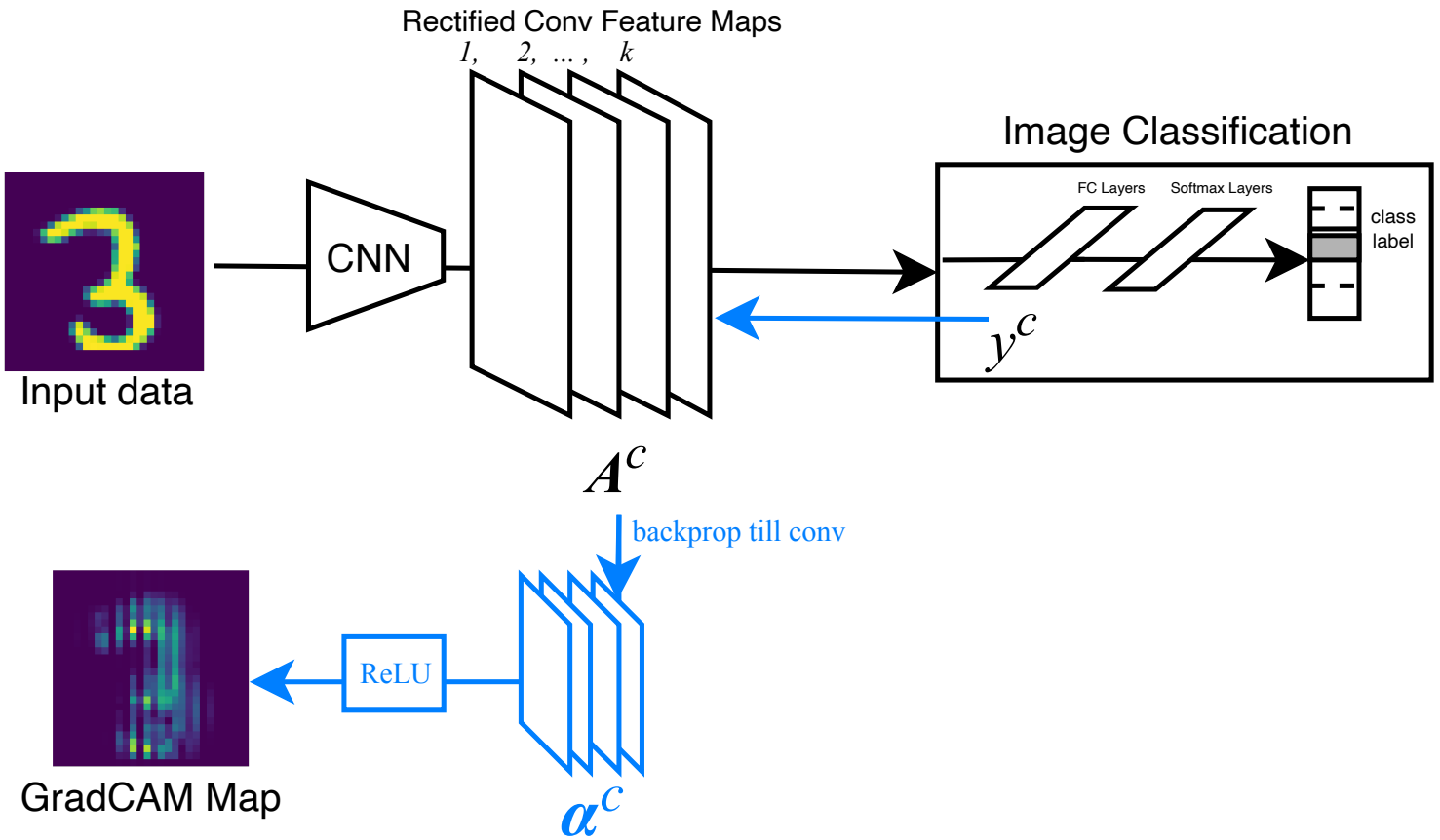
Workflow



[8] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. arXiv preprint arXiv:1706.03825, 2017.

GradCAM [9]

Workflow



[9] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE international conference on computer vision, pages 618–626, 2017.

Contents

Experiments and Results

Experiments

- Comparison between two network structures
- Visualizations evaluation
- Overfitting analysis

Two Network Structures

DNN model structure

Layers	Output Shape	Activation function
Input	(None, 784)	
Layer 1(Dense)	(None, 1024)	ReLU
Layer 2(Dense)	(None, 1024)	ReLU
Layer 3(Dense)	(None, 1024)	ReLU
Output layer(Dense)	(None, 10)	None

Two Network Structures

DNN model structure

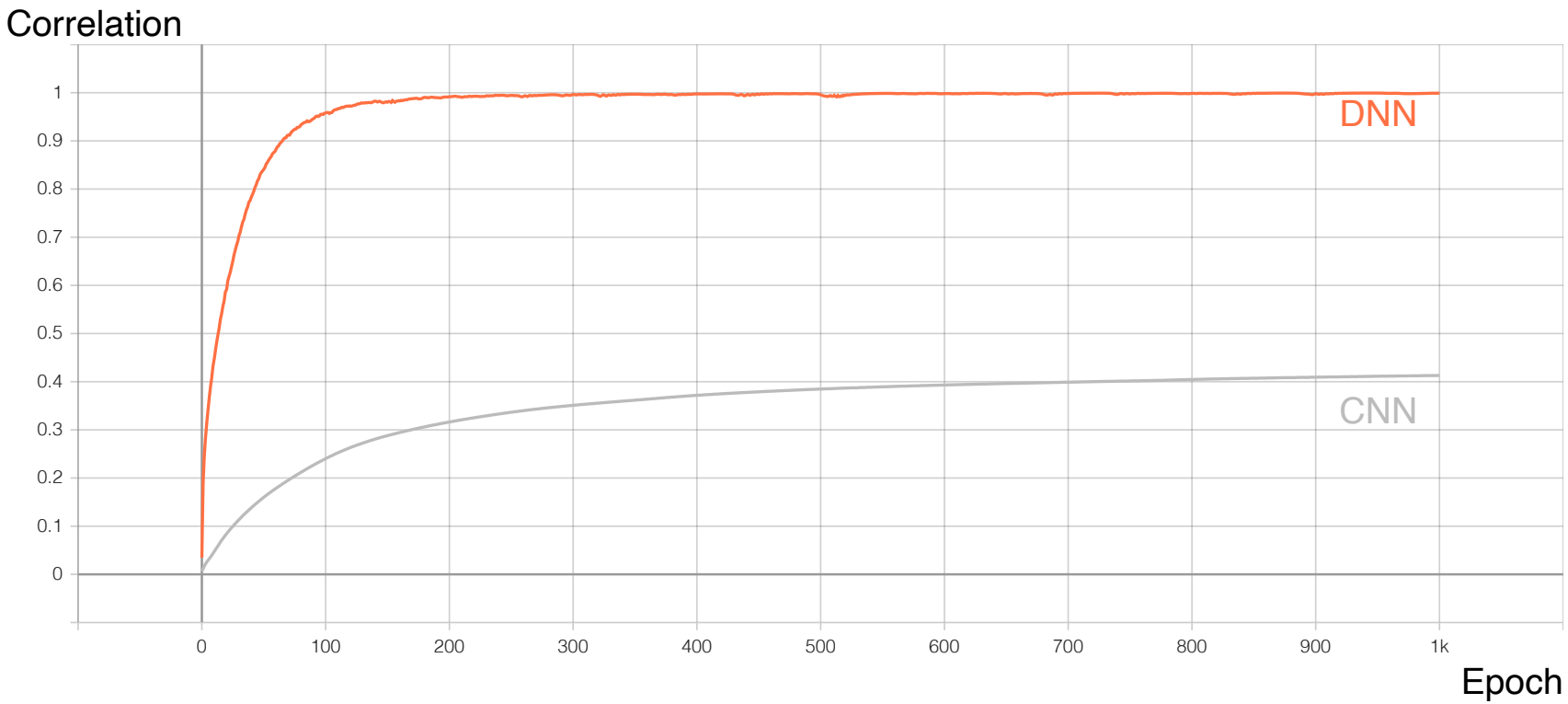
Layers	Output Shape	Activation function
Input	(None, 784)	
Layer 1(Dense)	(None, 1024)	ReLU
Layer 2(Dense)	(None, 1024)	ReLU
Layer 3(Dense)	(None, 1024)	ReLU
Output layer(Dense)	(None, 10)	None

CNN model structure

Layers	Output Shape	Filters	Kernel / Pooling	Stride	Padding	Activation function
Input	(None, 28,28,1)					
Layer 1(Conv2D)	(None, 28,28,8)	8	(5,5)	(1,1)	same	ReLU
Layer 2(MaxPooling2D)	(None, 14,14,8)		(2,2)	(2,2)	no padding	
Layer 3(Conv2D)	(None, 14,14,6)	6	(5,5)	(1,1)	same	ReLU
Layer 4(MaxPooling2D)	(None, 7,7,6)		(2,2)	(2,2)	no padding	
Layer 5(Conv2D)	(None, 7,7,4)	4	(5,5)	(1,1)	same	ReLU
Layer 6(MaxPooling2D)	(None, 3,3,4)		(2,2)	(2,2)	no padding	
Layer 7(Conv2D)	(None, 3,3,4)	2	(5,5)	(1,1)	same	ReLU
Output layer((Flatten))	(None, 18)					

Comparison between Two Network Structures

Correlation

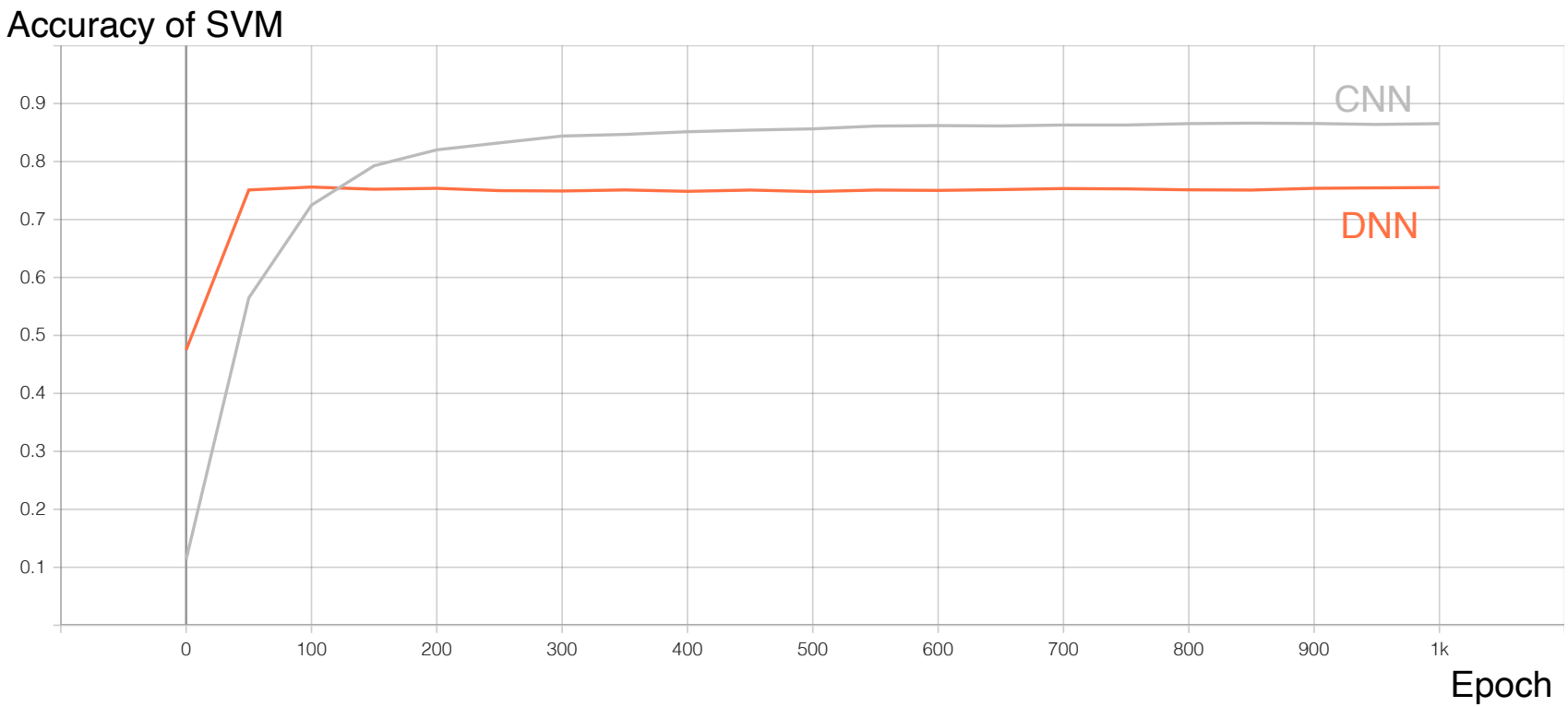


Setting

- Input: 4 boxes MNIST

Comparison between Two Network Structures

Accuracy



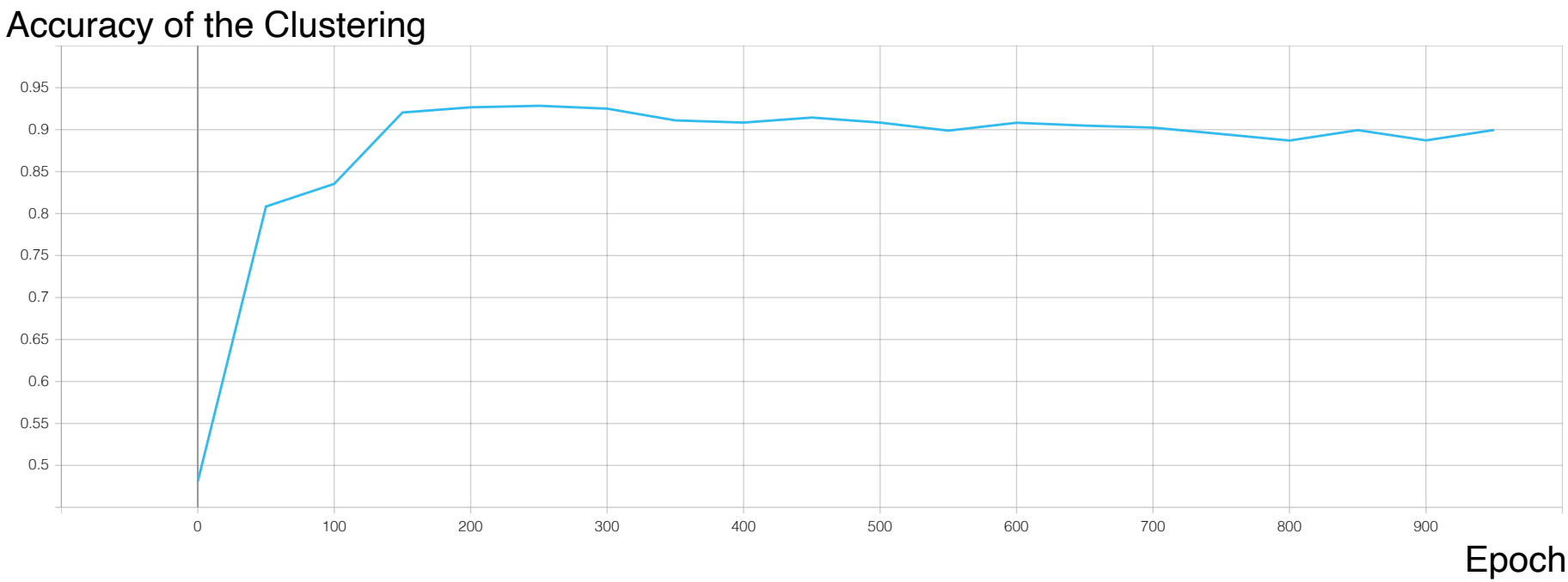
Setting

- Input: 4 boxes MNIST

Visualization Evaluation

Visualization Evaluation

DNN model

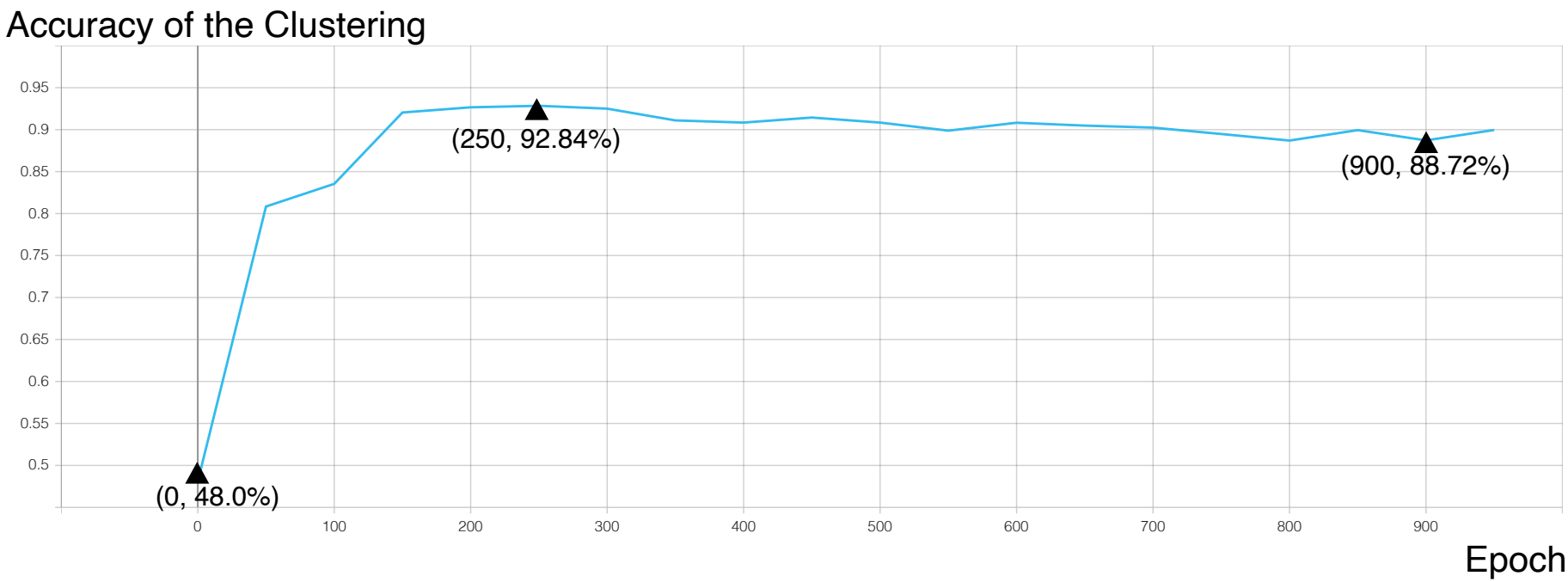


Setting

- Input: 1 box MNIST

Visualization Evaluation

DNN model

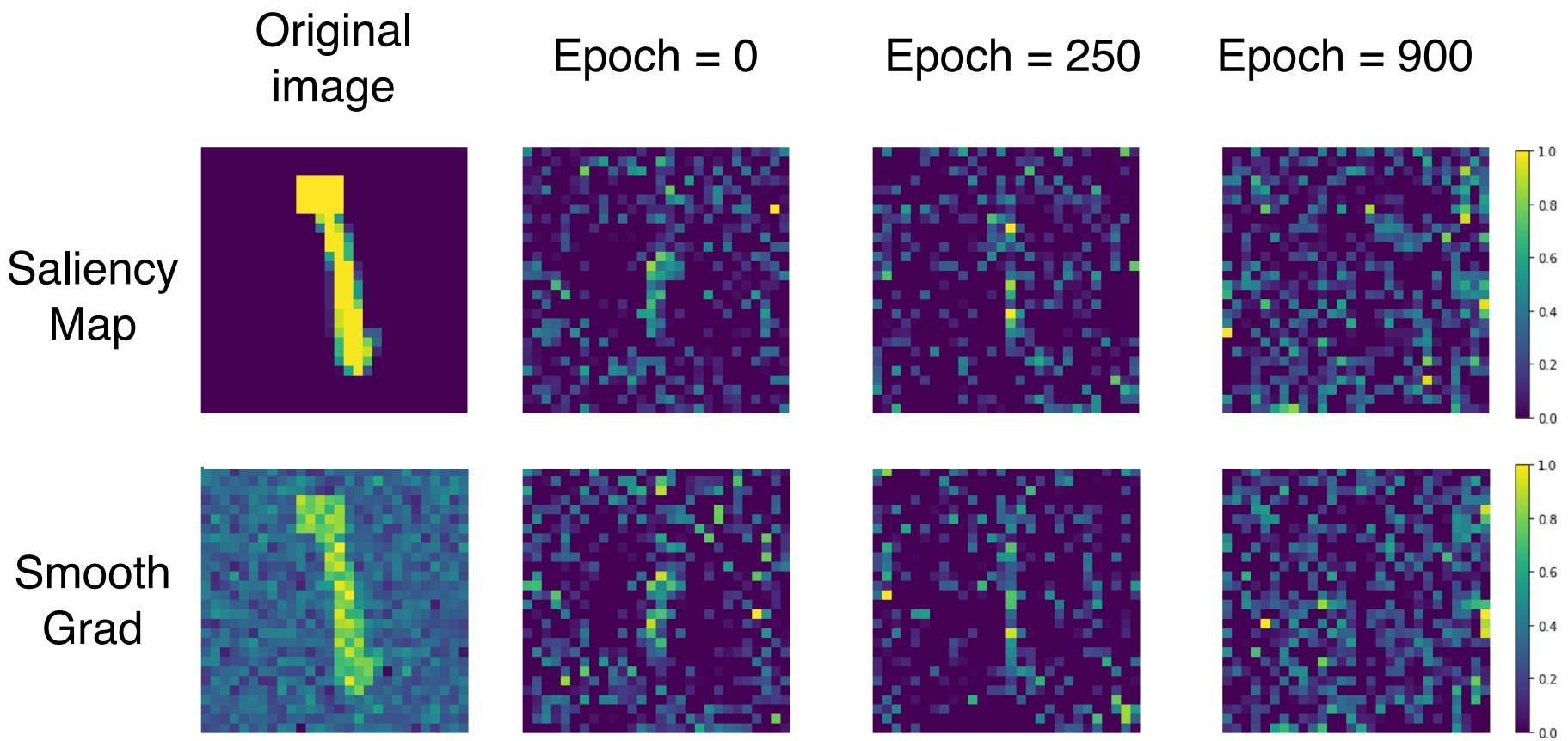


Setting

- Input: 1 box MNIST

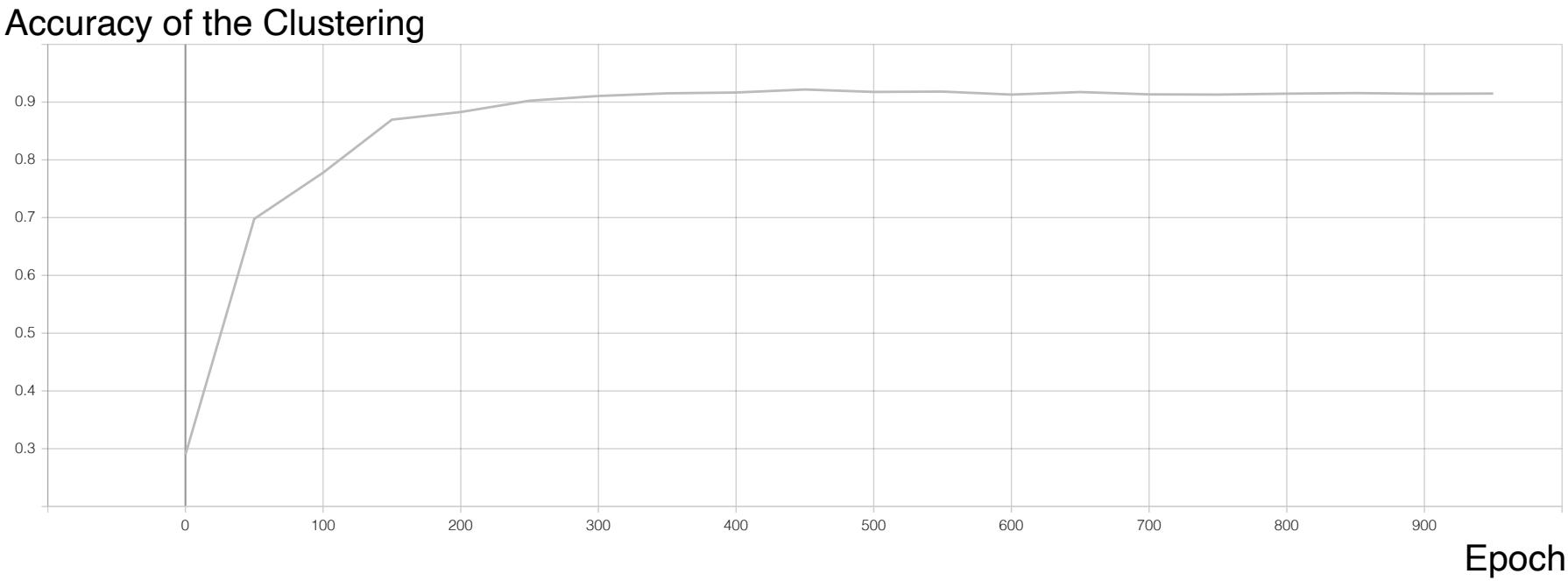
Visualization Evaluation

DNN model



Visualization Evaluation

CNN model

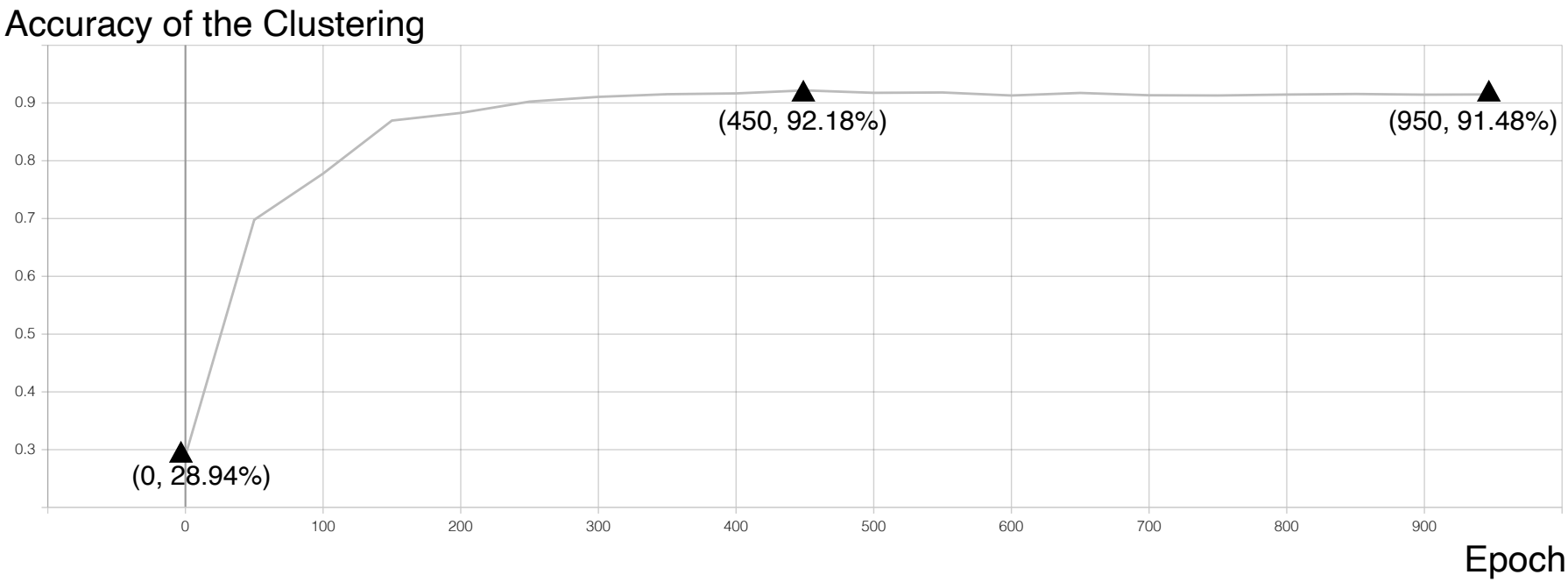


Setting

- Input: 1 box MNIST

Visualization Evaluation

CNN model

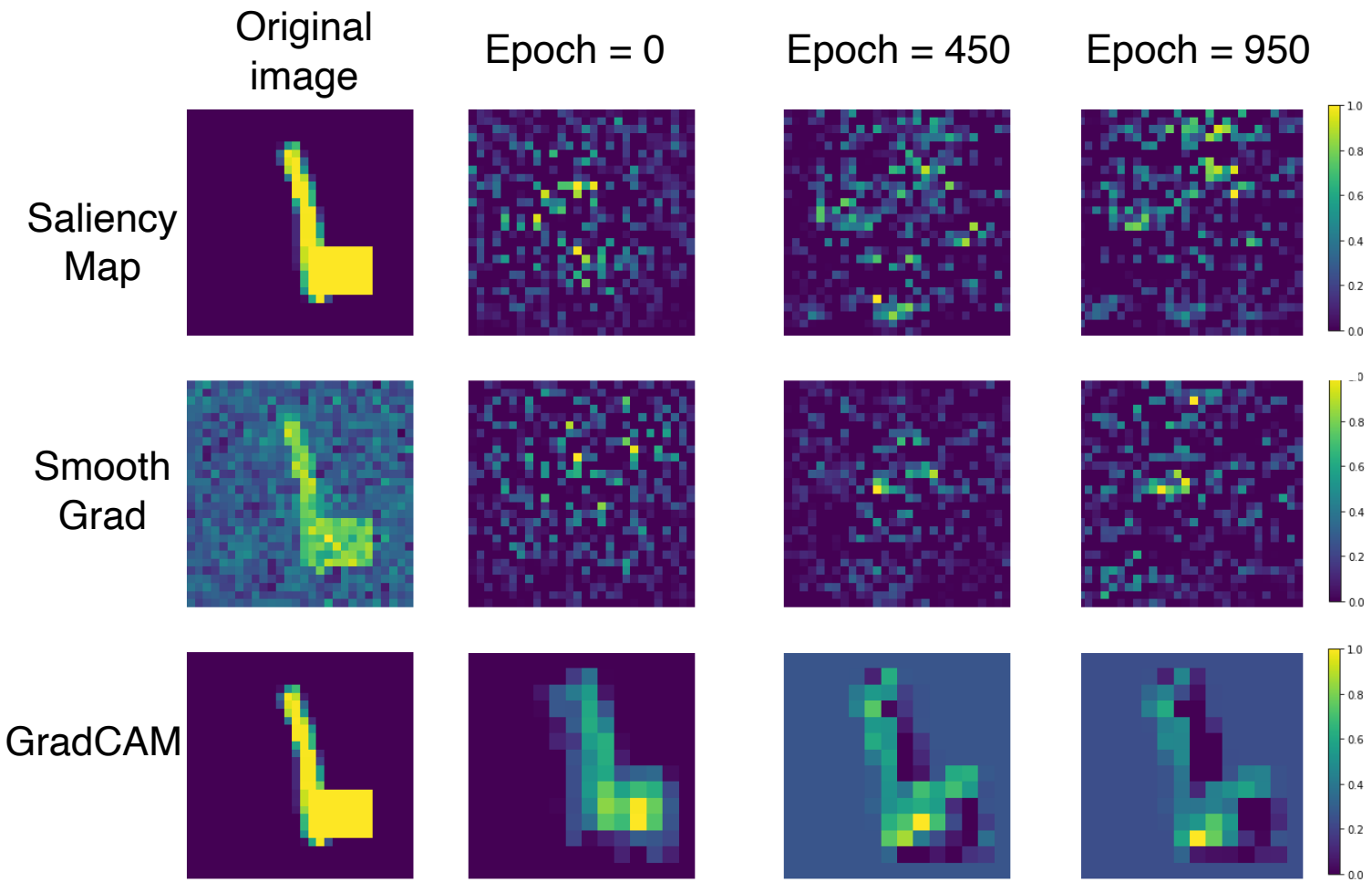


Setting

- Input: 1 box MNIST

Visualization Evaluation

CNN model



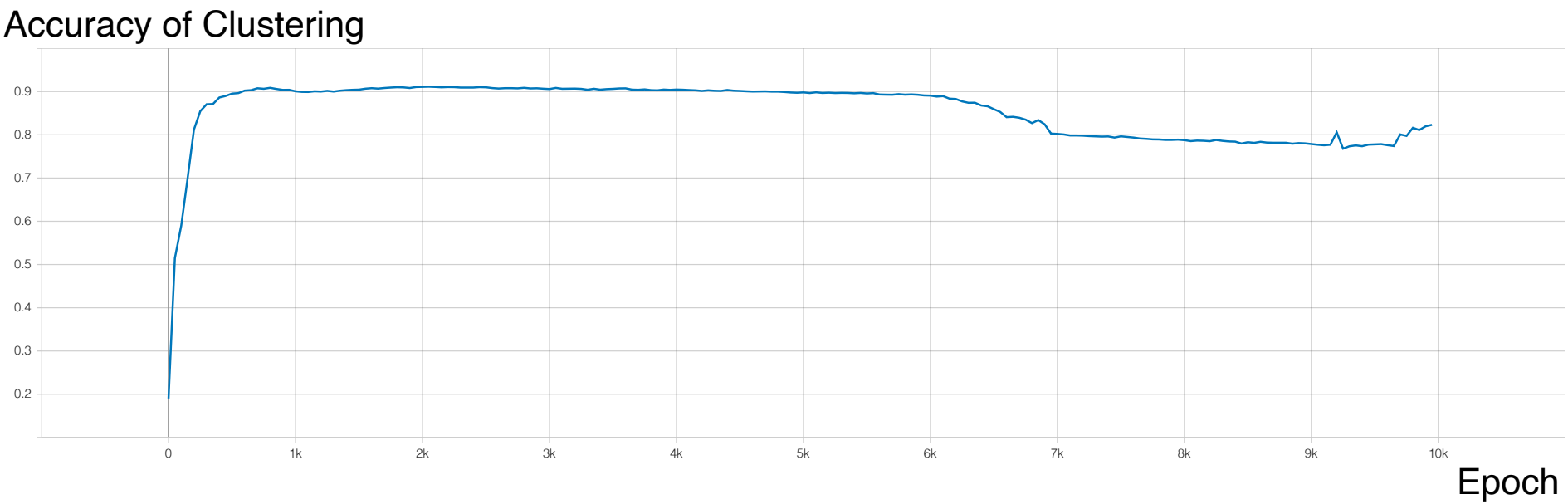
Overfitting Evaluation

Setting

- Input: 2 boxes MNIST

Overfitting Evaluation

CNN model

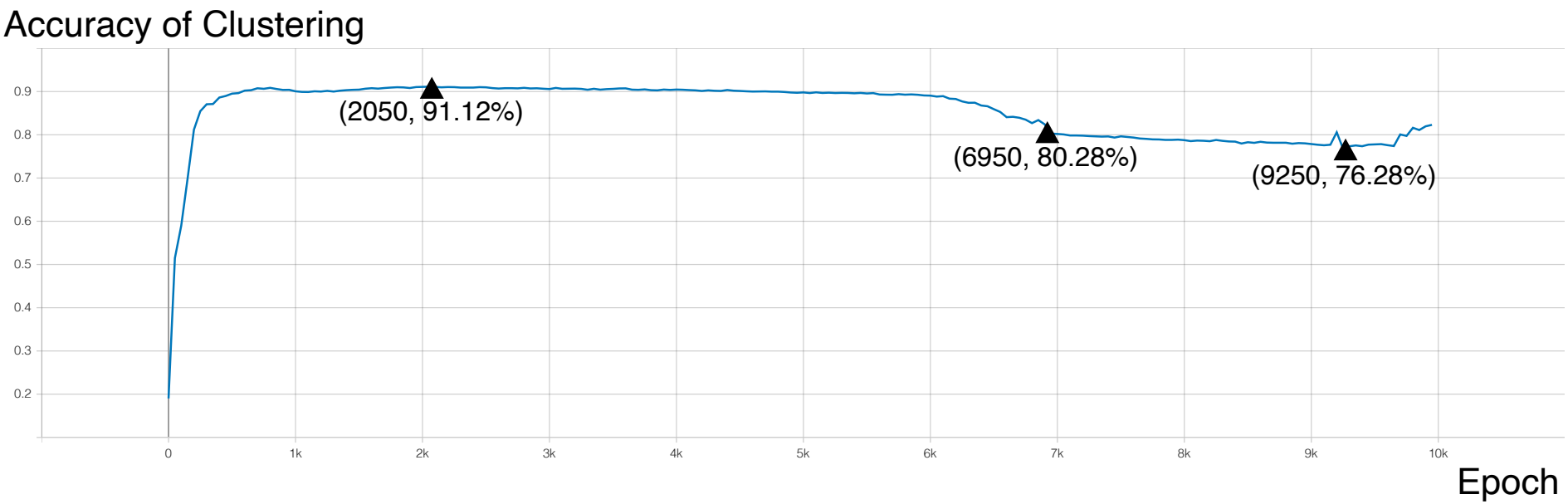


Setting

- Input: 2 boxes MNIST

Overfitting Evaluation

CNN model

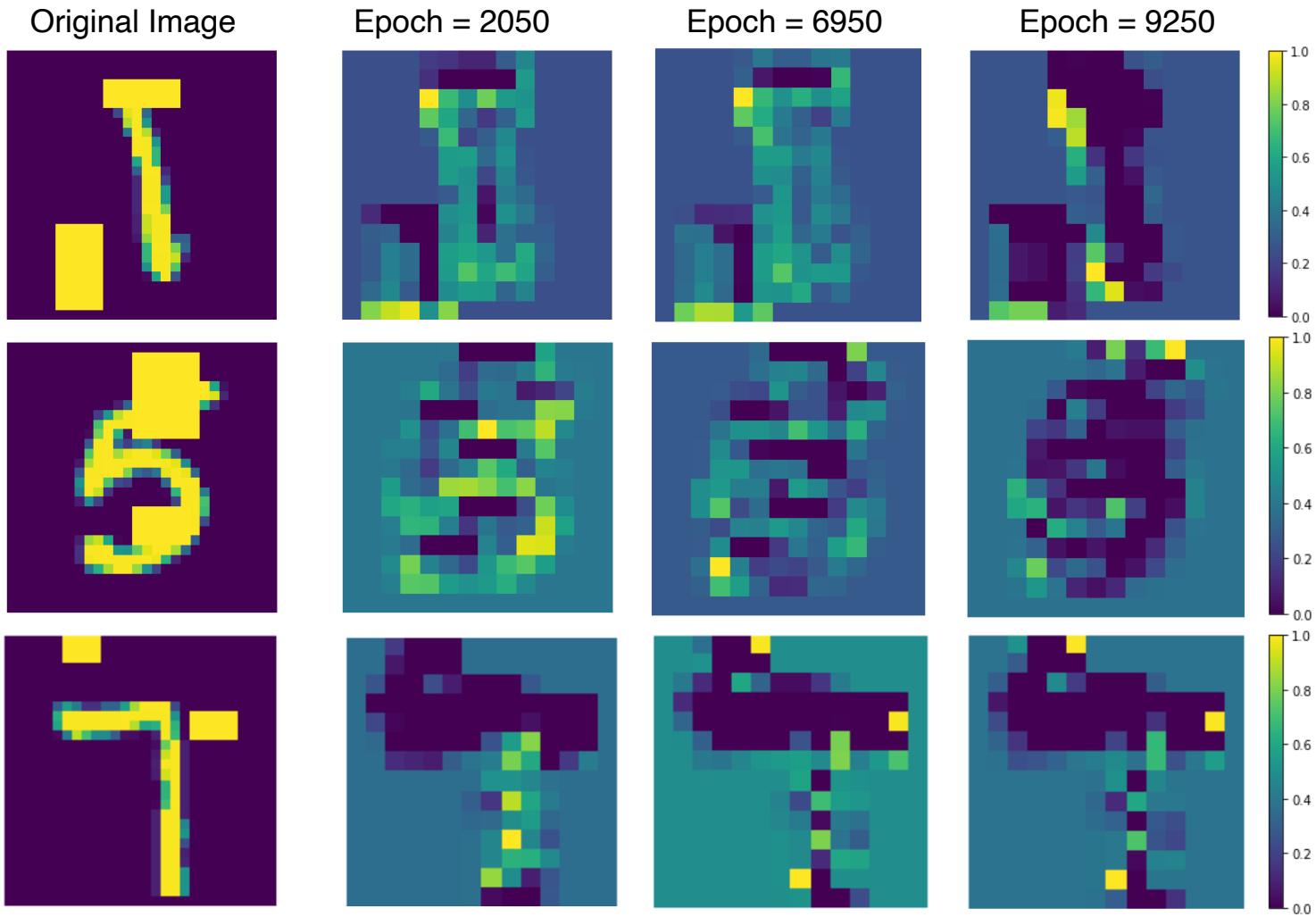


Setting

- Input: 2 boxes MNIST

Overfitting Evaluation

CNN model



Overfitting Evaluation

- Quantitative metric $\Rightarrow$ Digit attention

Overfitting Evaluation

- Quantitative metric $\Rightarrow$ Digit attention

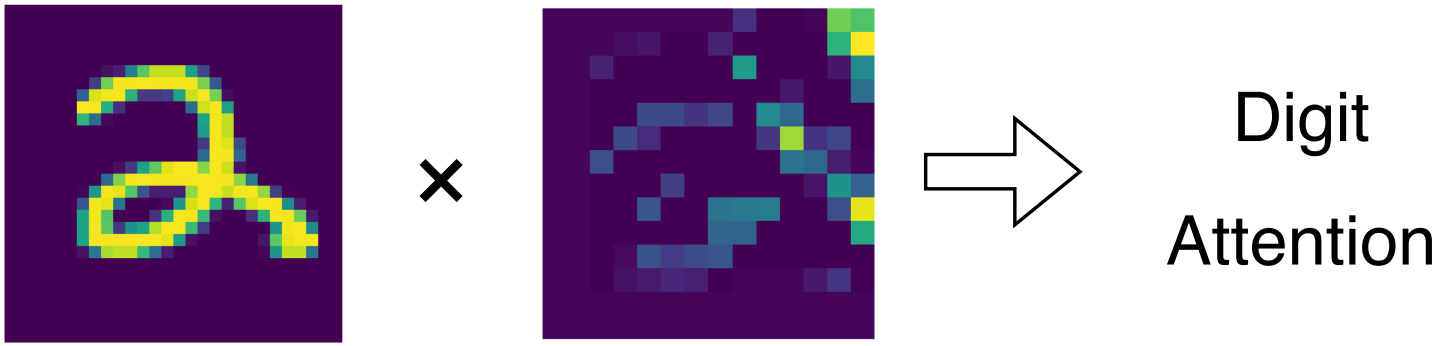
Digit Attention

Overfitting Evaluation

- Quantitative metric ⇒ Digit attention

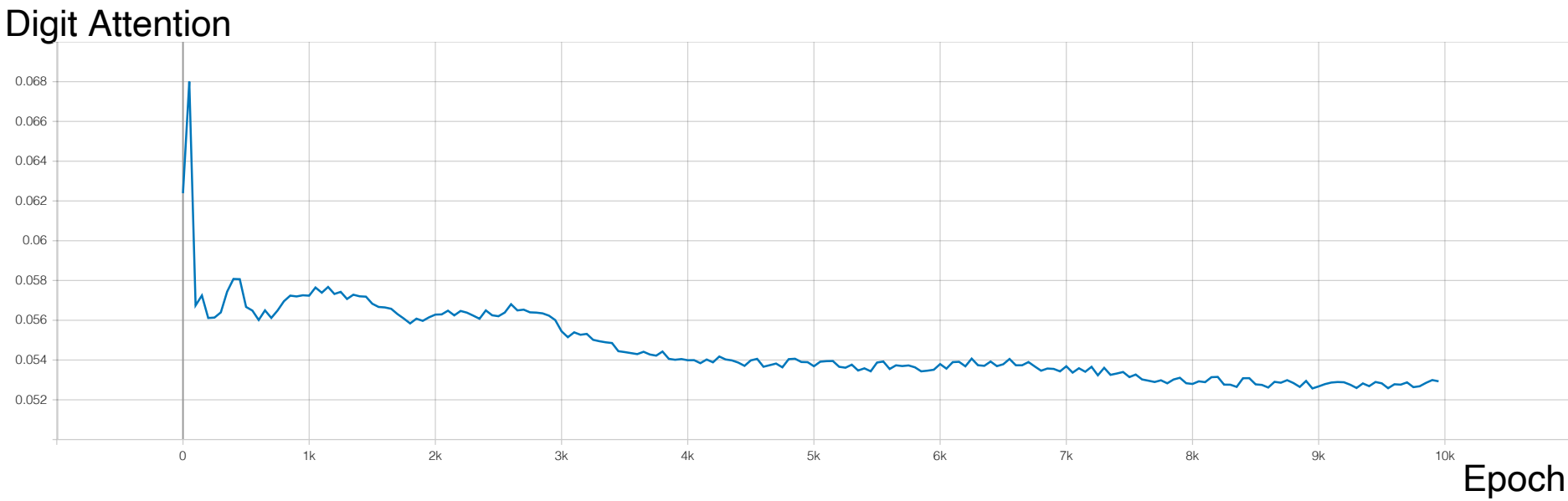
Digit Attention

Digit Attention



Overfitting Evaluation

CNN model

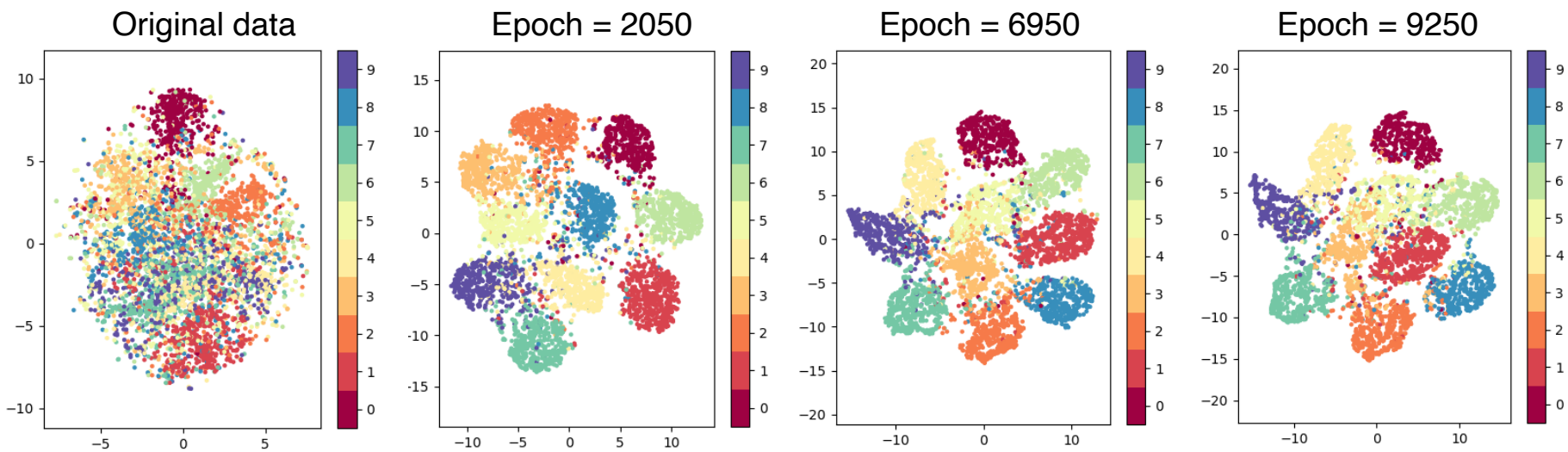


Setting

- Input: 2 boxes MNIST

Overfitting Evaluation

CNN model



Setting

- Input: 2 boxes MNIST

Contents

Conclusion

Conclusion

Contribution

- Traditional DCCAE model

Conclusion

Contribution

- Traditional DCCAE model

√ CNN-based DCCAE model
  - outperformed traditional DCCAE

Conclusion

Contribution

- Traditional DCCAE model

✓ CNN-based DCCAE model
  - outperformed traditional DCCAE

Contribution

- Correlation $\rightarrow 1 \Rightarrow$ Overfitting

Conclusion

Contribution

- Traditional DCCAE model

✓ CNN-based DCCAE model
  - outperformed traditional DCCAE

Contribution

- Correlation $\rightarrow 1 \Rightarrow$ Overfitting

✓ Analysing overfitting
  - more accurate representations
  - less likely to overfitting

Conclusion

Contribution

- Traditional DCCAE model

✓ CNN-based DCCAE model
  - outperformed traditional DCCAE

Contribution

- Correlation $\rightarrow 1 \Rightarrow$ Overfitting

✓ Analysing overfitting
  - more accurate representations
  - less likely to overfitting

Contribution

- Not interpretable

Conclusion

Contribution

- Traditional DCCAE model

✓ CNN-based DCCAE model
  - outperformed traditional DCCAE

Contribution

- Correlation $\rightarrow 1 \Rightarrow$ Overfitting

✓ Analysing overfitting
  - more accurate representations
  - less likely to overfitting

Contribution

- Not interpretable

✓ Interpretability
  - Saliency Map
  - SmoothGrad
  - GradCAM

Outlook

Outlook

Limitation

- Visualization results $\rightarrow$ Not ideal

Outlook

Limitation

- Visualization results $\rightarrow$ Not ideal
- Saliency Map, SmoothGrad $\rightarrow$ Extra classifier

Outlook

Limitation

- Visualization results $\rightarrow$ Not ideal
- Saliency Map, SmoothGrad $\rightarrow$ Extra classifier

Outlook

- More visualization techniques

Outlook

Limitation

- Visualization results $\rightarrow$ Not ideal
- Saliency Map, SmoothGrad $\rightarrow$ Extra classifier

Outlook

- More visualization techniques
  - Guided Backpropagation

Outlook

Limitation

- Visualization results $\rightarrow$ Not ideal
- Saliency Map, SmoothGrad $\rightarrow$ Extra classifier

Outlook

- More visualization techniques
  - Guided Backpropagation
- More quantitative metrics
  - Attention corporate more techniques
  - Relate with the correlation

This endeavour would not have been possible without the help of my supervisors:

Dr.-Ing. Tanuj Hasija and Mr. Maurice Kuschel

I am truly grateful for their support and guidance.

Thank you!

CCA

Moreover, notice that the solution is independent with the scalar of the projections $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$. For example, if $u_1 = \alpha \boldsymbol{u}_1$ where α is a scalar,

$$\begin{aligned}\frac{\alpha \boldsymbol{u}_1^\intercal \boldsymbol{\Sigma}_{12} \boldsymbol{u}_2}{\sqrt{\alpha \boldsymbol{u}_1^\intercal \boldsymbol{\Sigma}_{11} \alpha \boldsymbol{u}_1 \boldsymbol{u}_2^\intercal \boldsymbol{\Sigma}_{22} \boldsymbol{u}_2}} &= \frac{\alpha \boldsymbol{u}_1^\intercal \boldsymbol{\Sigma}_{12} \boldsymbol{u}_2}{\sqrt{\alpha^2 \boldsymbol{u}_1^\intercal \boldsymbol{\Sigma}_{11} \boldsymbol{u}_1 \boldsymbol{u}_2^\intercal \boldsymbol{\Sigma}_{22} \boldsymbol{u}_2}} \\ &= \frac{\boldsymbol{u}_1^\intercal \boldsymbol{\Sigma}_{12} \boldsymbol{u}_2}{\sqrt{\boldsymbol{u}_1^\intercal \boldsymbol{\Sigma}_{11} \boldsymbol{u}_1 \boldsymbol{u}_2^\intercal \boldsymbol{\Sigma}_{22} \boldsymbol{u}_2}} \, .\end{aligned}$$

CCA

For solving the (8), there are two ways to solve it.

- Eigenvalue Decomposition (EVD)

$$\begin{aligned} \Sigma_{11}^{-1}\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\boldsymbol{U}_1 &= \lambda_1^2\boldsymbol{U}_1 \\ \Sigma_{22}^{-1}\Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}\boldsymbol{U}_2 &= \lambda_2^2\boldsymbol{U}_2 \end{aligned} \tag{1}$$

- Singular Value Decomposition (SVD)

$$\begin{aligned} \Sigma_{11}^{-1/2}\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\Sigma_{11}^{-1/2}\boldsymbol{U}_1 &= \eta_1\boldsymbol{U}_1 \\ \Sigma_{22}^{-1/2}\Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}\Sigma_{22}^{-1/2}\boldsymbol{U}_2 &= \eta_2\boldsymbol{U}_2 \end{aligned} \tag{2}$$

DeepCCA

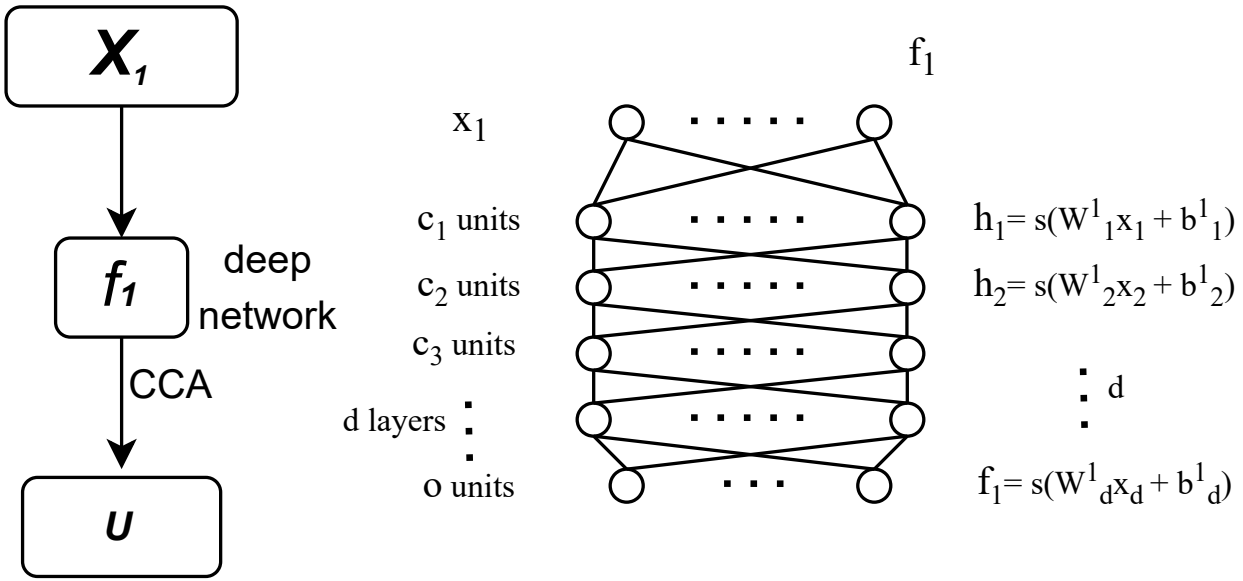


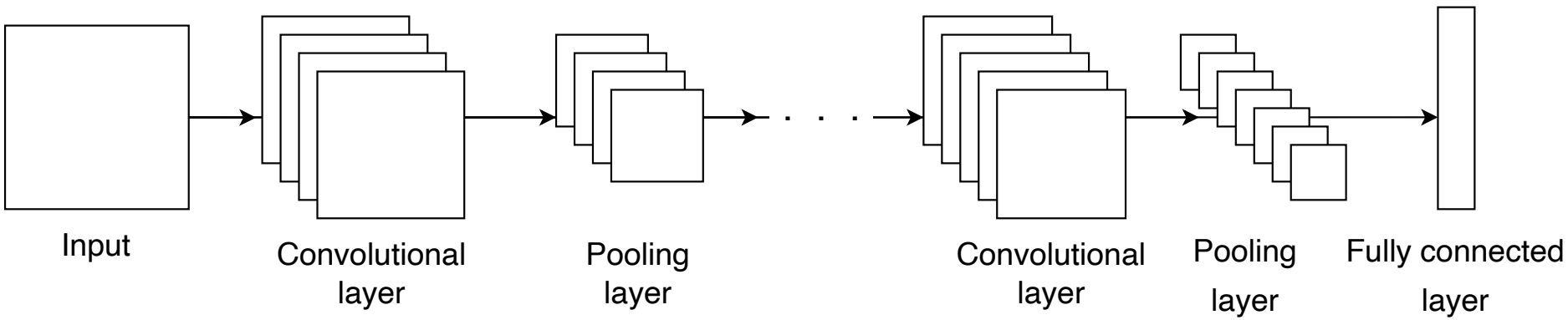
Figure: A schematic of deep CCA

Given an instance x_2 of the second view, the representation $f_2(x_2)$ is computed the same way, with different parameters $\boldsymbol{W}_l^2$ and b_l^2.

Convolution Neural Network

A simple convolutional neural network structure

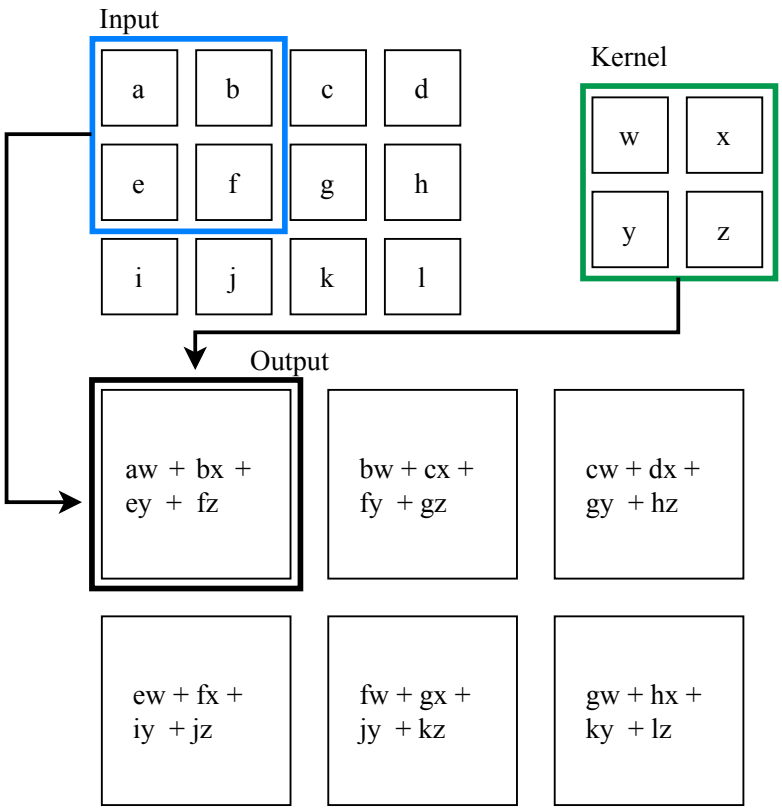
- Convolutional layers
  - Convolution operation
  - Padding and strides
- Pooling layers
- A fully connected layer



Convolution Neural Network

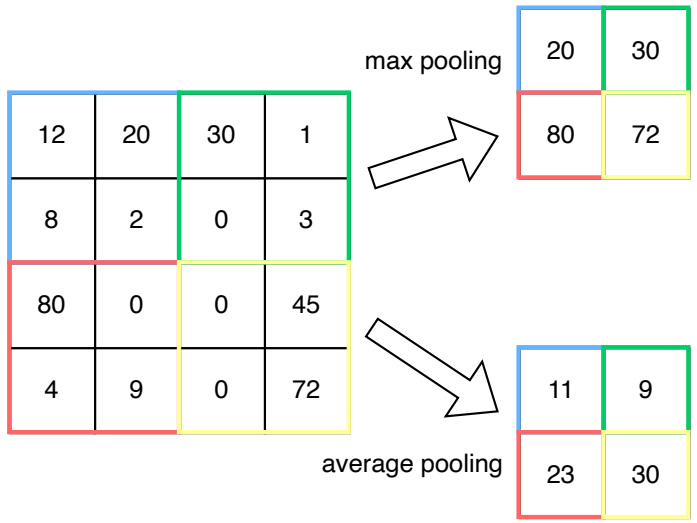
Convolutional layers

- Convolution operation
- Padding and strides



Pooling layers

- Average pooling
- Max pooling



Saliency Map

Saliency Map

Given a gray scalar image $\boldsymbol{I}_1 \in \mathbb{R}^{w \times h}$ and the corresponding class c_1, the output can be calculated by the classifier function $\boldsymbol{S}_c$ and the predicted label $class(\boldsymbol{I}_1)$ is the index of the maximal value in the output.

$$class(\boldsymbol{I}_1) = \underset{c}{\operatorname{argmax}}\, \boldsymbol{S}_c(\boldsymbol{I}_1) \tag{3}$$

To get the Saliency Map $\boldsymbol{M} \in \mathbb{R}^{w \times h}$, first it has to get the derivative $\boldsymbol{\omega}$ of $class(\boldsymbol{I}_1)$ with the respect to the image $\boldsymbol{I}_1$ by backpropagation

$$\begin{aligned} \boldsymbol{\omega} &= \frac{\partial class(\boldsymbol{I}_1)}{\partial \boldsymbol{I}_1} \\ &= \frac{\partial \operatorname{argmax}_c \boldsymbol{S}_c(\boldsymbol{I}_1)}{\partial \boldsymbol{I}_1} \,. \end{aligned} \tag{4}$$

Saliency Map

Saliency Map

The number of elements in $\boldsymbol{\omega}$ is equal to the number of pixels in $\boldsymbol{I}_1$. Therefore the Saliency Map $\boldsymbol{M}$ can be taken directly from the absolute value of the $\boldsymbol{\omega}$ with corresponding pixels.

$$\boldsymbol{M}(i,j) = |\boldsymbol{\omega}_{p(i,j)}| \ , \tag{5}$$

where the $p(i,j)$ denotes the index of the element of $\boldsymbol{\omega}$ corresponding to the image $\boldsymbol{I}_1$ in the i-th row and j-th column.
If the input $\boldsymbol{I}_1$ is a multi-channel image, e.g. **RGB** picture. The Saliency Map $\boldsymbol{M}$ can take the maximal value of the channels h to be obtained.

$$\boldsymbol{M}(i,j) = \max_h |\boldsymbol{\omega}_{p(i,j,h)}| \ , \tag{6}$$

where the $p(i,j,h)$ represents the index of the pixel in the (i,j) position and the c color channel of image $\boldsymbol{I}_1$.

SmoothGrad

SmoothGrad

Adding noise has a de-noise effect

$$\tilde{\boldsymbol{I}} = \frac{1}{n}\sum_{1}^{n}(\hat{\boldsymbol{I}}_1 + \cdots + \hat{\boldsymbol{I}}_n) \ , \tag{7}$$

where $\hat{\boldsymbol{I}}_n$ denotes the noised $\boldsymbol{I}$ for $n = 1, 2, ..., n$.

$$\begin{aligned} \hat{\boldsymbol{I}}_1 &= \boldsymbol{I}_1 + \mathcal{N}(0, \sigma^2) \ , \\ \hat{\boldsymbol{I}}_2 &= \boldsymbol{I}_1 + \mathcal{N}(0, \sigma^2) \ , \\ &\vdots \\ \hat{\boldsymbol{I}}_n &= \boldsymbol{I}_1 + \mathcal{N}(0, \sigma^2) \ , \end{aligned} \tag{8}$$

where $\mathcal{N}(0, \sigma^2)$ represents Gaussian noise with standard deviation σ.

SmoothGrad

SmoothGrad

Then using each noised input can calculate the corresponding noised gradient $\hat{\boldsymbol{\omega}}_1, \hat{\boldsymbol{\omega}}_2, ..., \hat{\boldsymbol{\omega}}_n$ by backpropagation:

$$\hat{\boldsymbol{\omega}}_n = \frac{\partial class(\hat{\boldsymbol{I}}_n)}{\partial \hat{\boldsymbol{I}}_n} = \frac{\partial \operatorname{argmax}_c \boldsymbol{S}_c(\hat{\boldsymbol{I}}_n)}{\partial \hat{\boldsymbol{I}}_n} \ . \tag{9}$$

Taking the de-noised thinking in (7) to calculate the smooth gradient of input $\boldsymbol{I}$

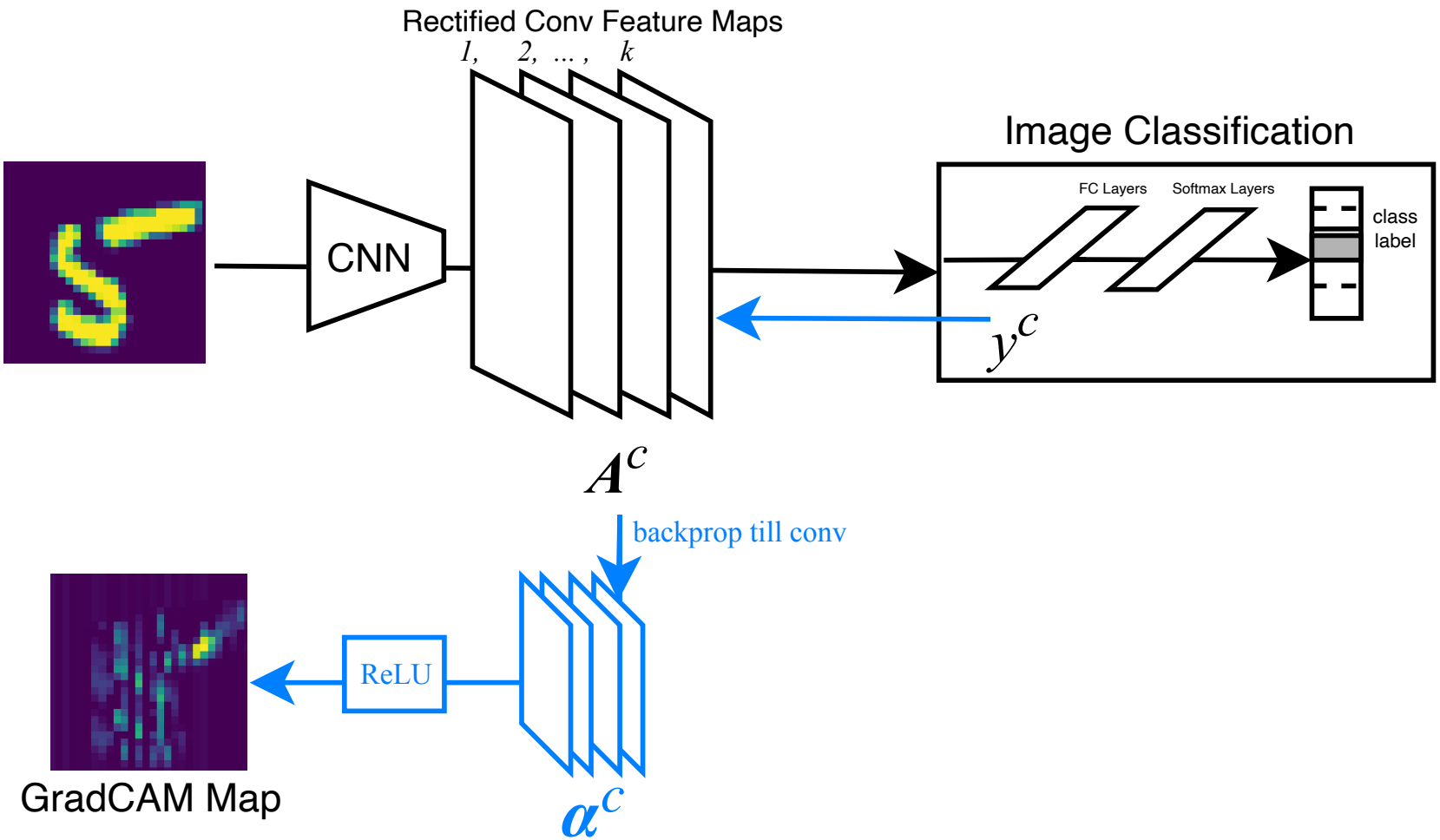
$$\tilde{\boldsymbol{\omega}} = \frac{1}{n} \sum_{1}^{n} (\hat{\boldsymbol{\omega}}_1 + \hat{\boldsymbol{\omega}}_2 + \cdots + \hat{\boldsymbol{\omega}}_n) \ . \tag{10}$$

SmoothGrad map can be computed:

$$\begin{aligned} \tilde{\boldsymbol{M}}(i,j) &= |\hat{\boldsymbol{\omega}}_{p(i,j)}| \ , \\ \tilde{\boldsymbol{M}}(i,j) &= \max_h |\hat{\boldsymbol{\omega}}_{p(i,j,h)}| \ , \text{ if the input is multi-channel image} \end{aligned} \tag{11}$$

GradCAM

GradCAM



Vsiualization

GradCAM

It first has to compute the gradient of the score for class c, $\boldsymbol{y}^c$(before the softmax), with respect to feature map activation $\boldsymbol{A}^k$ of a convolutional layer,

$$\frac{\partial \boldsymbol{y}^c}{\partial \boldsymbol{A}^k}$$

These gradients then are global-average-pooled over the width and height dimensions to obtain the neuron importance weights α_k^c.

$$\alpha_k^c = \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial \boldsymbol{y}^c}{\partial A_{ij}^k}}_{\text{gradient via backprop}}, \tag{12}$$

where A_{ij}^k represents the element value in $\boldsymbol{A}^k$ with corresponding pixel index (i, j) and Z is the sum of all pixels.

Vsiualization

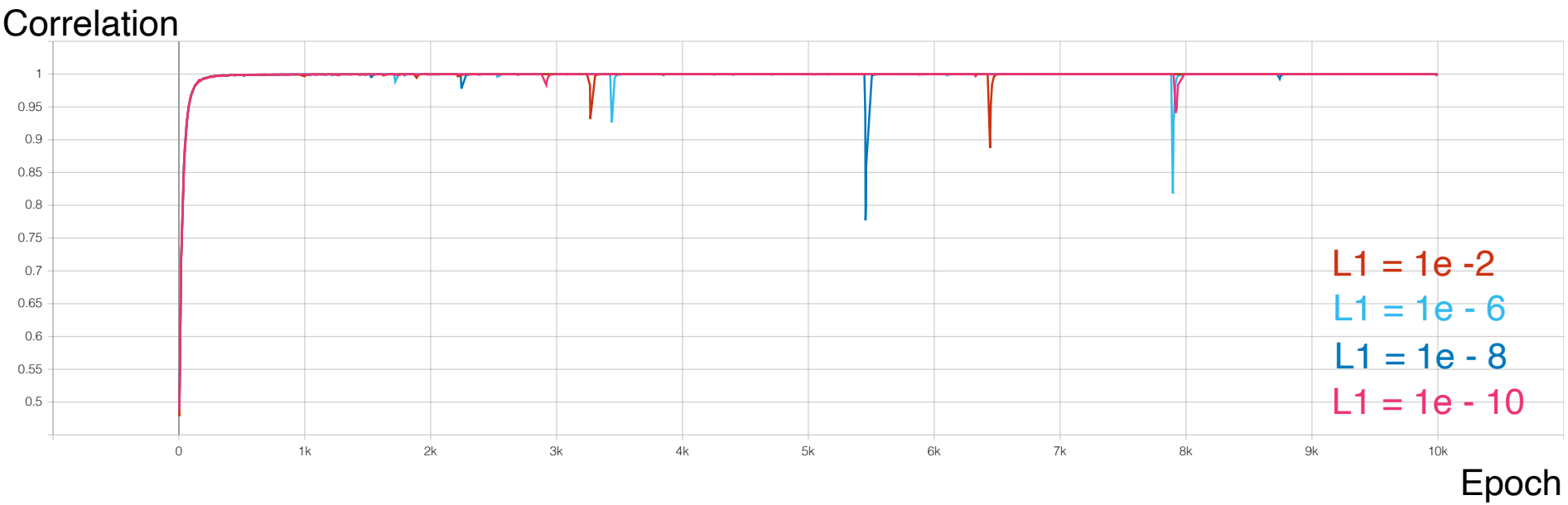
GradCAM

Then GradCAM map $\boldsymbol{L}^c_{\text{GradCAM}}$ can be a weighted combination of forward activation maps then be the answers after through a ReLU.

$$\boldsymbol{L}^c_{\text{GradCAM}} = \text{ReLU} \underbrace{\left(\sum_k \alpha^c_k \boldsymbol{A}^k \right)}_{\text{linear combination}} \tag{13}$$

L1-norm regularization

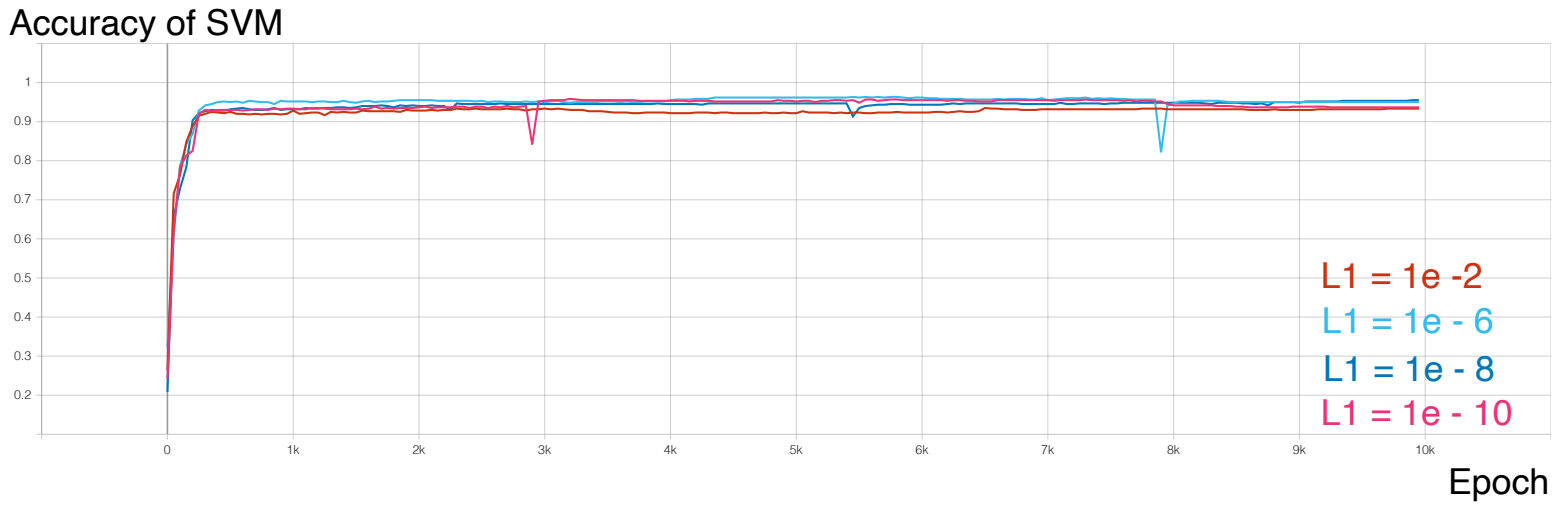
DNN model



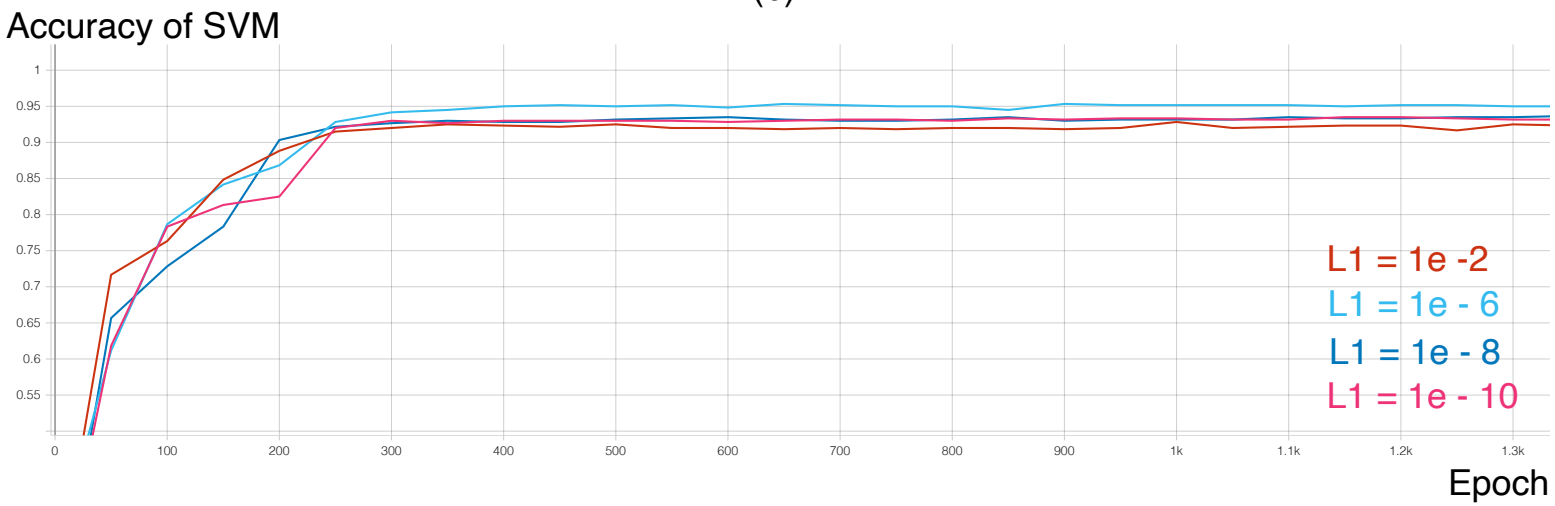
L1-norm regularization

DNN model

(a)

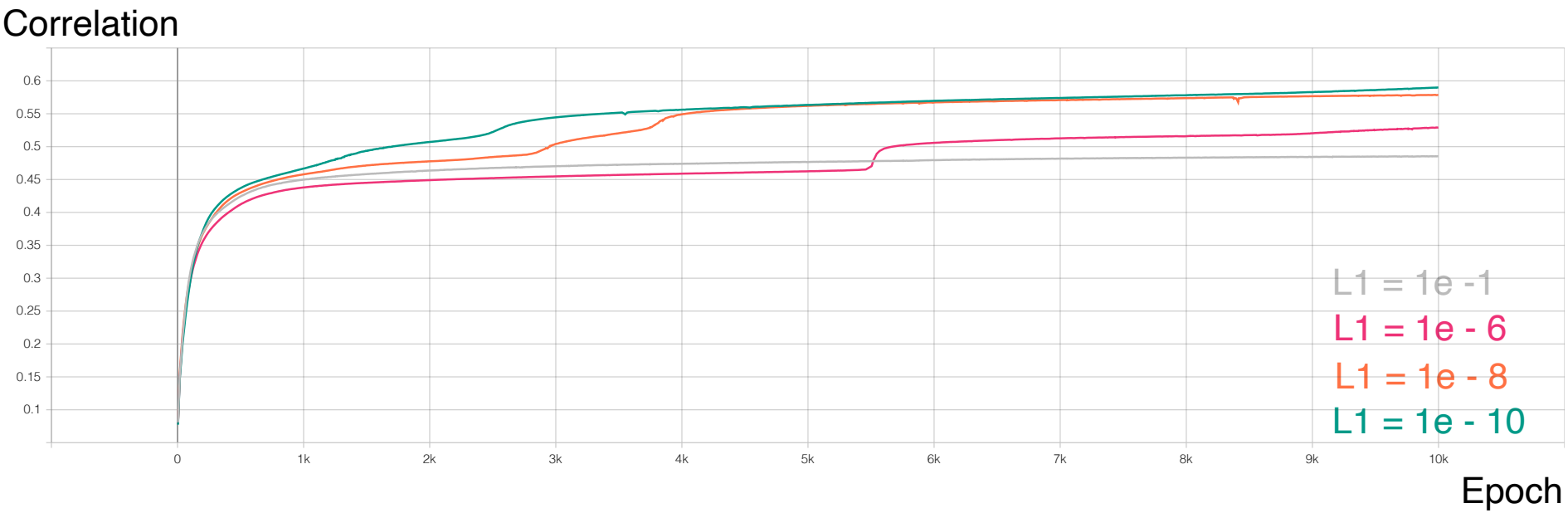


(b)



L1-norm regularization

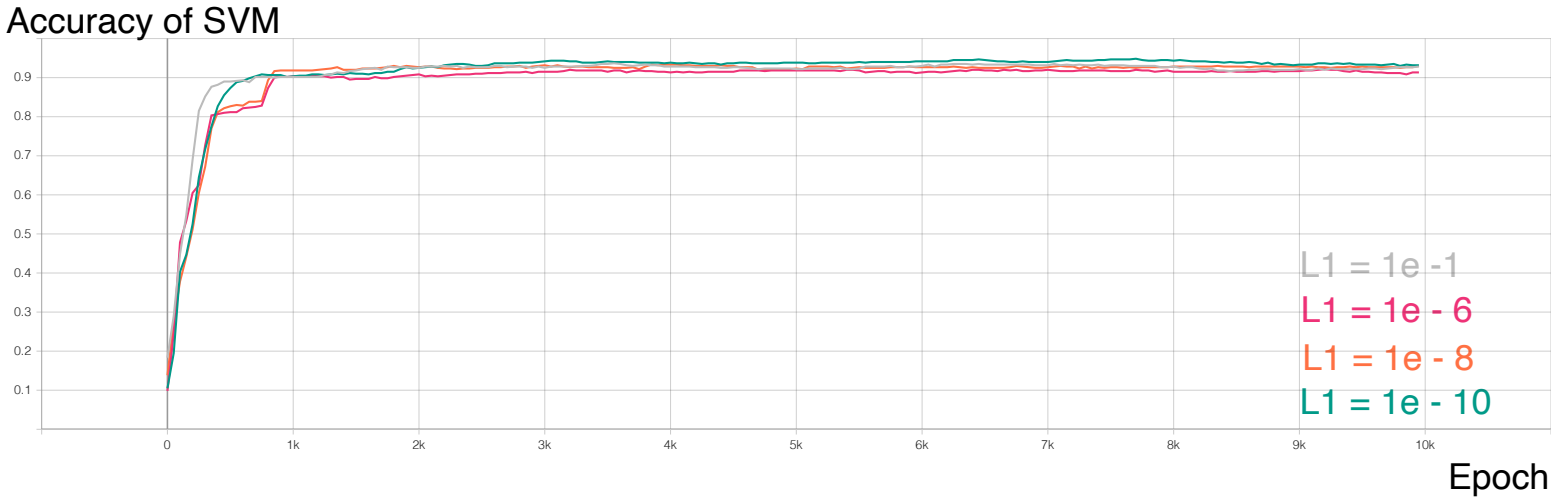
CNN model



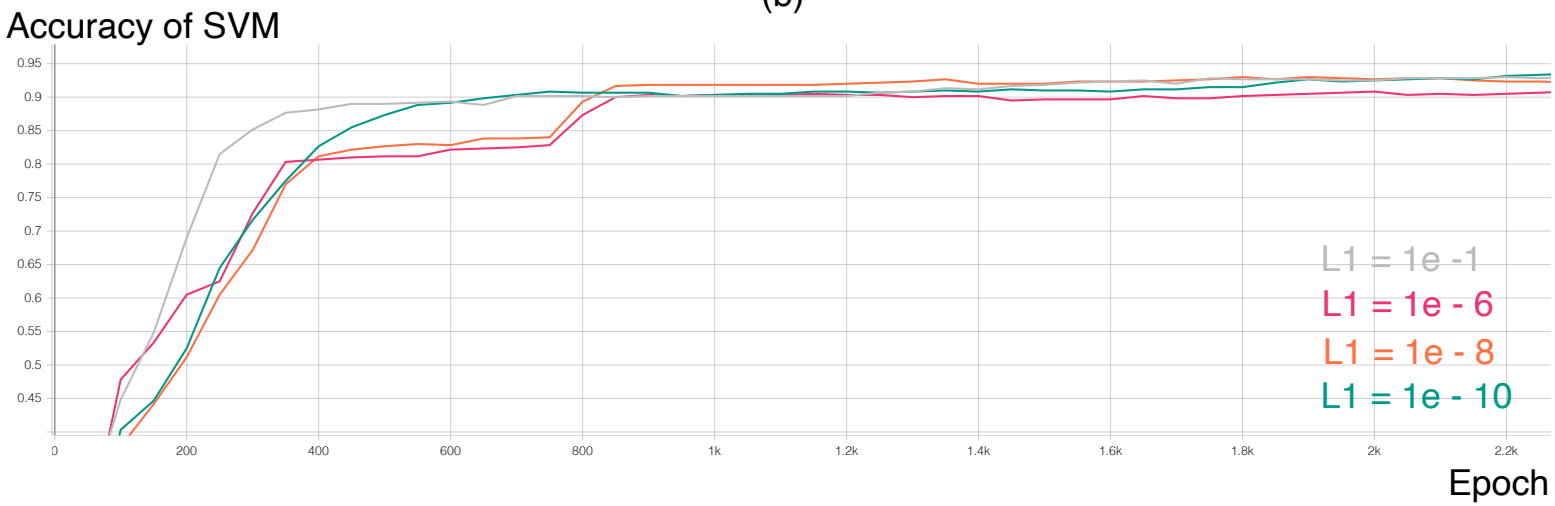
L1-norm regularization

CNN model

(a)

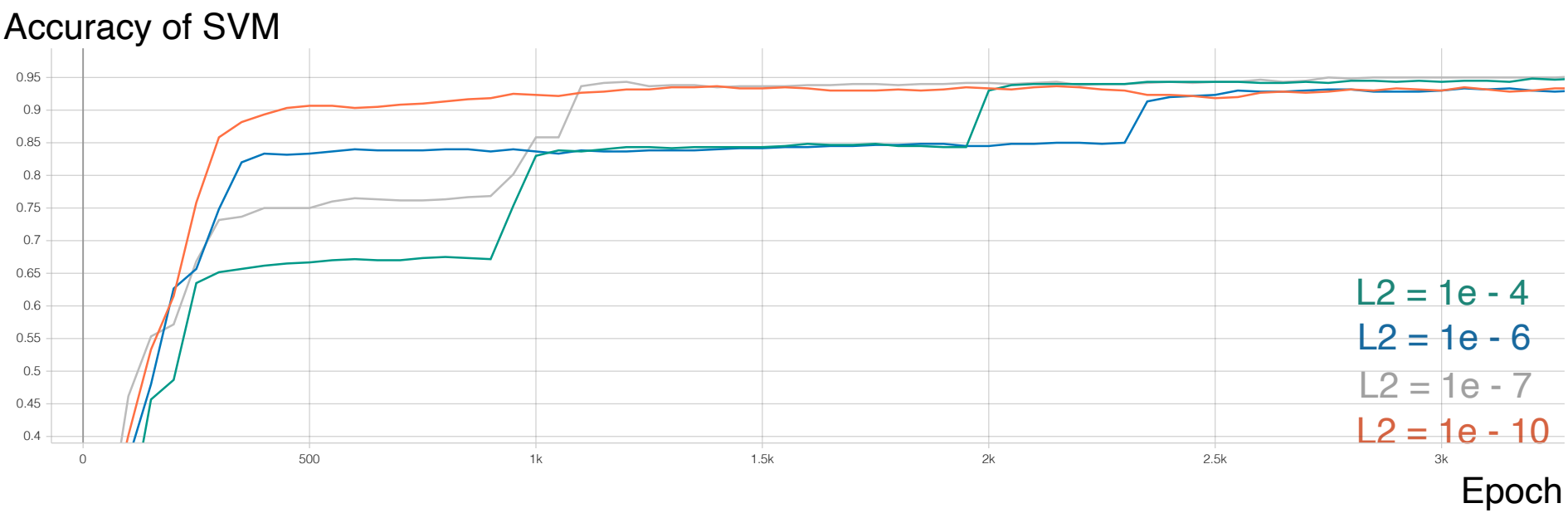


(b)



L2-norm regularization

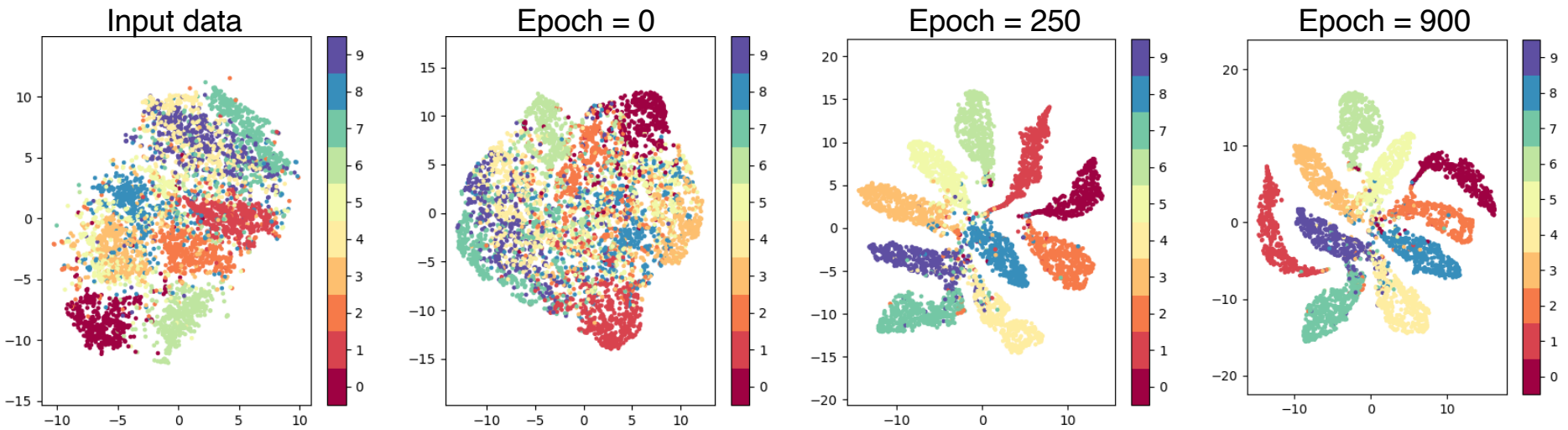
CNN model



The comparable optimal hyper-parameters setting is $L2 = 1e - 6$

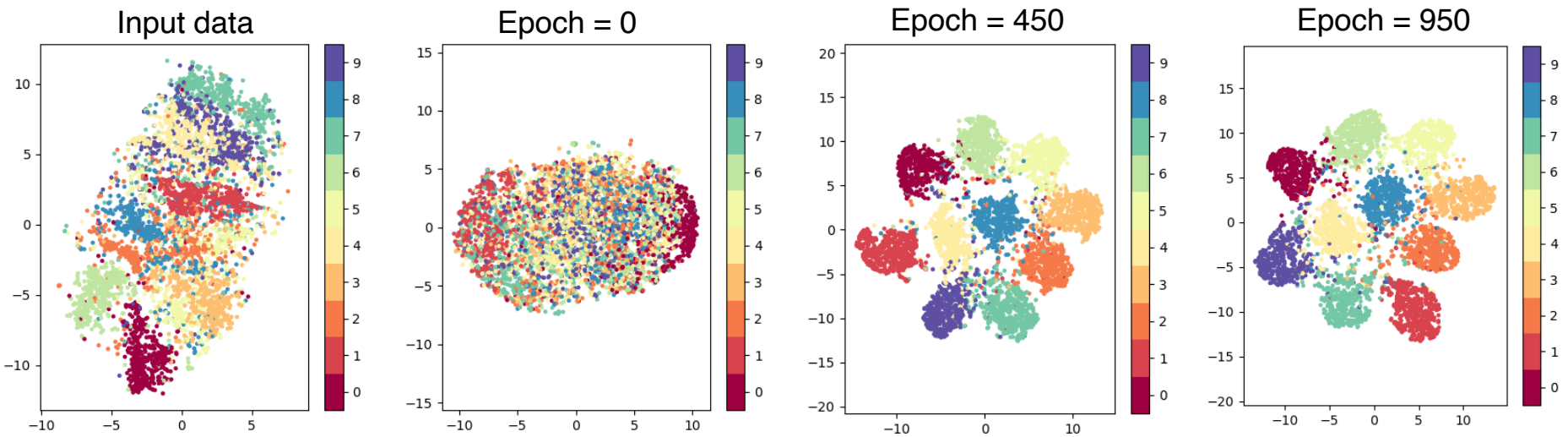
Visualization Evaluation

DNN model



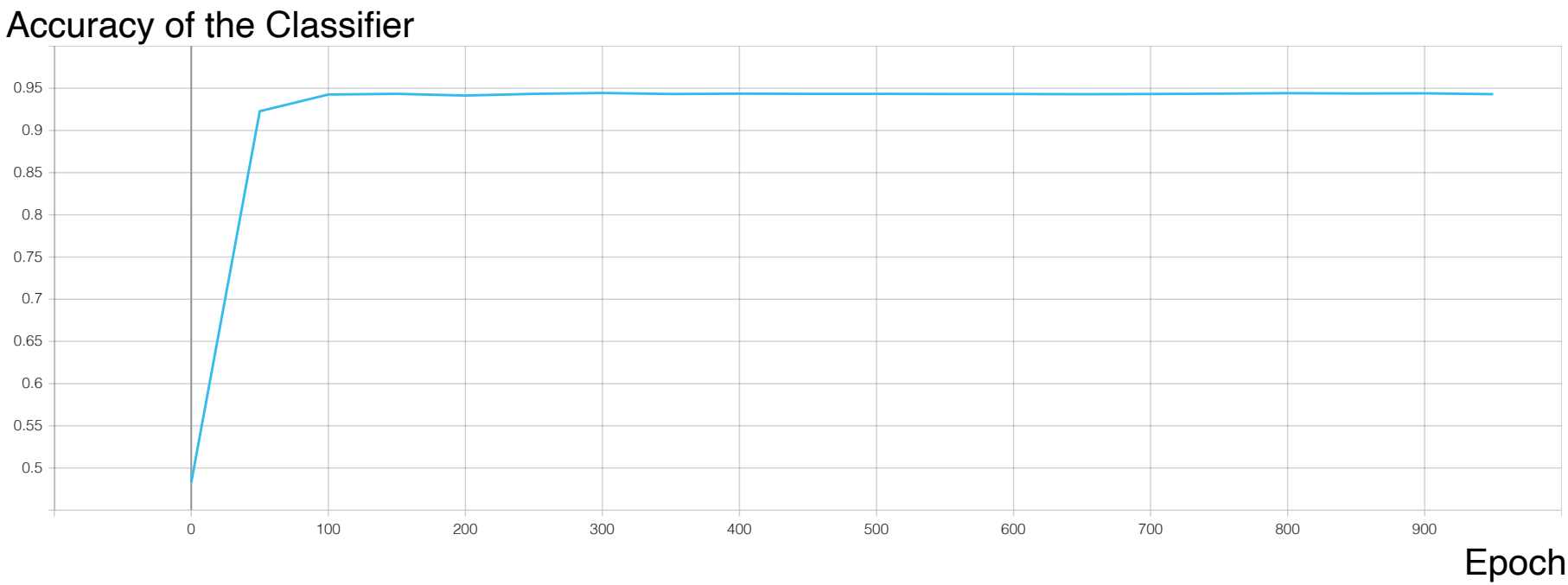
Visualization Evaluation

CNN model



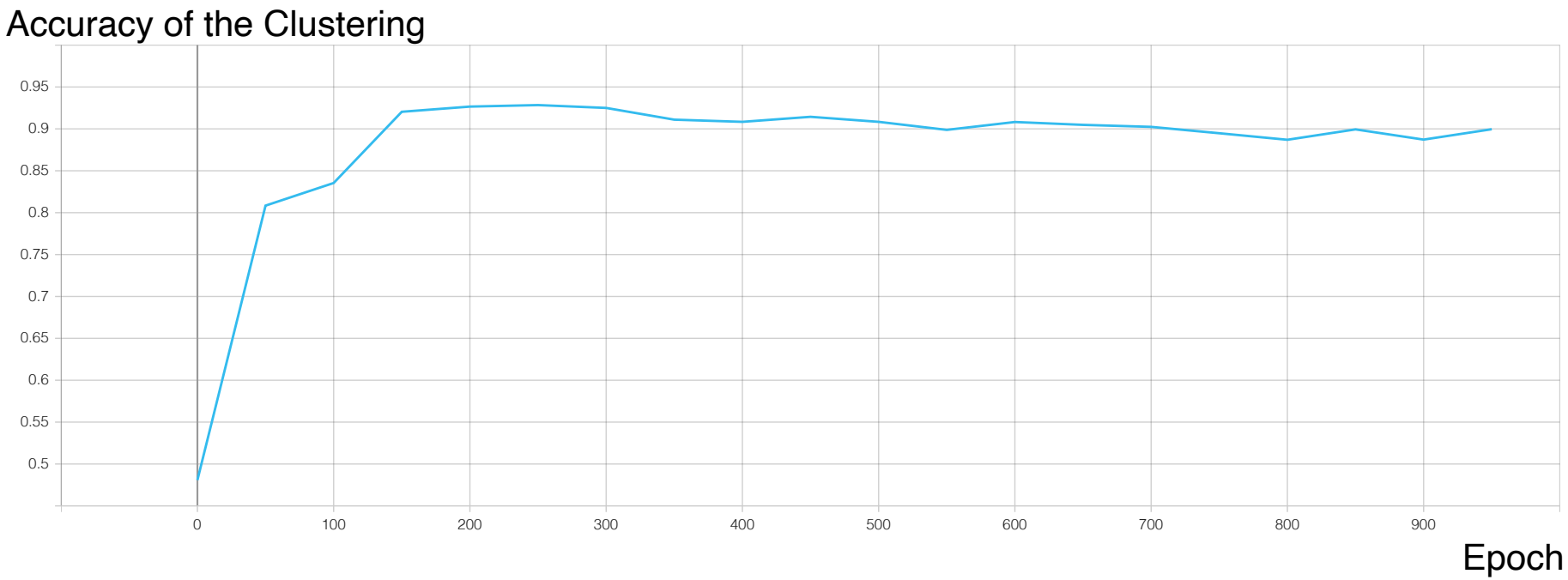
Overfitting Evaluation

CNN model



Overfitting Evaluation

CNN model



- k-means clustering[10]

[10]Anil K Jain and Richard C Dubes. Algorithms for clustering data. Prentice-Hall, Inc., 1988.

Overfitting Evaluation

New performance metric

- box attention

box attention

An original image $\boldsymbol{I}$ has the corresponding GradCAM map $\boldsymbol{G} \in \mathbb{R}^{m_g \times n_g}$ and the box filter is $\boldsymbol{B} \in \mathbb{R}^{m \times n}$. First scale the $\boldsymbol{G}$ in the same size of $\boldsymbol{I}$ by a linear transformation,

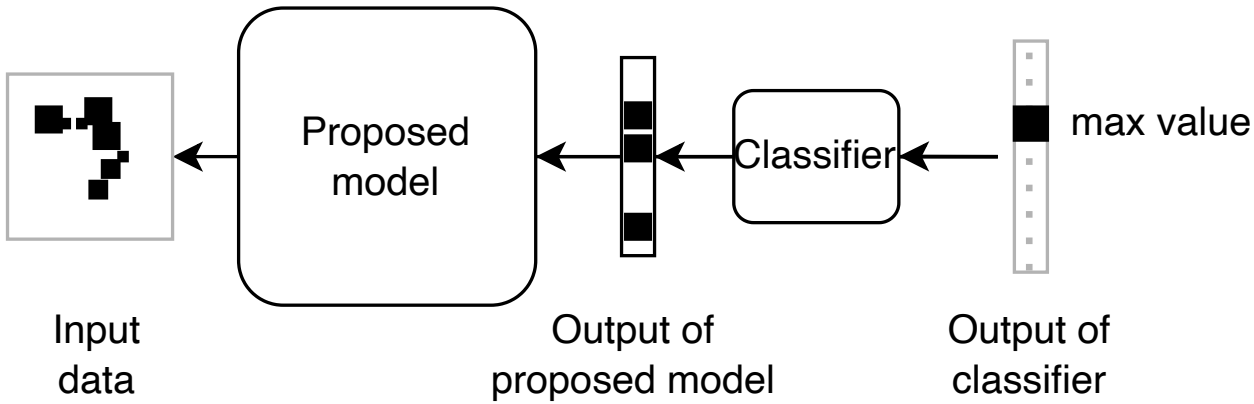
$$\boldsymbol{G}'(i,j) = \frac{m-n}{m_g - n_g}\boldsymbol{G}(i,j) \ ,$$

where $\boldsymbol{G}'$ is the scaled GradCAM map and (i,j) represents the pixel value in the i-th row and j-th column. Then box attention b is defined as the average result of scaled GradCAM map times the box filter,

$$b = \mathrm{E}(\boldsymbol{G}' \cdot \boldsymbol{B}) \ . \tag{14}$$

Visualization Techniques

Saliency Map and SmoothGrad workflow in the proposed model



Setting

- Gaussian noise $\mathcal{N}(0, 0.4^2)$

Experiments and Results

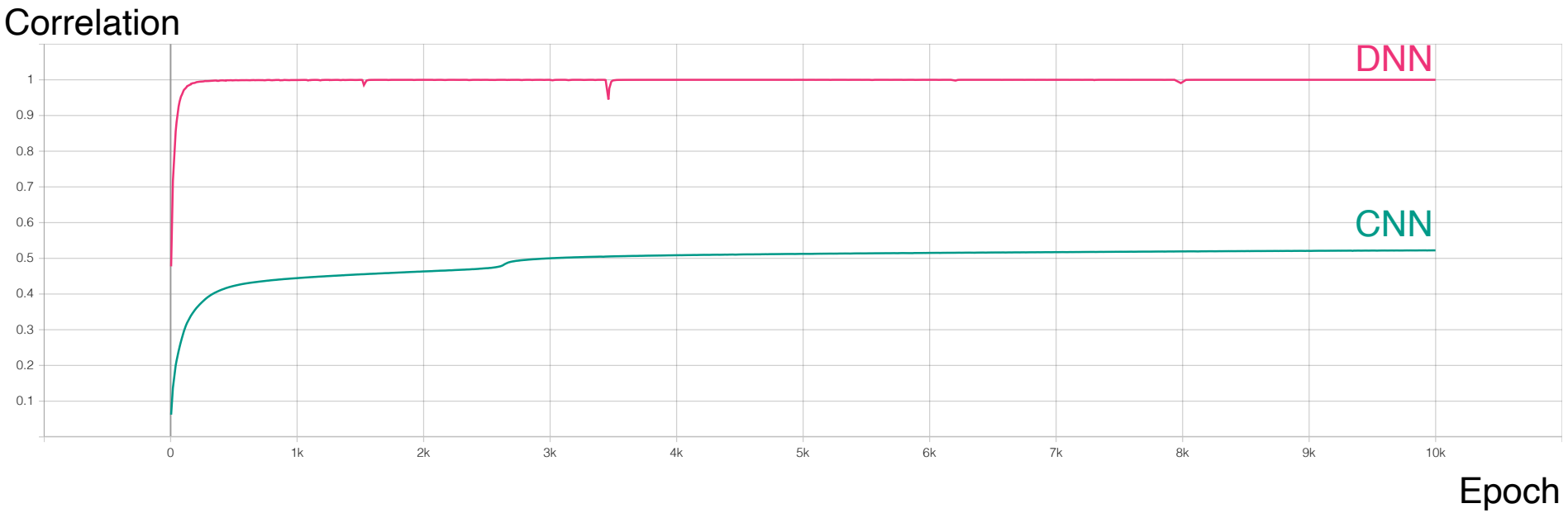
Experiments

- Comparison between Two Network Structures
- Hyper-parameter settings
  - L1-norm
  - L2-norm
- Visualizations evaluation
- Overfitting analysis

Optimal setting

- The comparable optimal hyper-parameters setting is $L2 = 1e - 6$

L2-norm



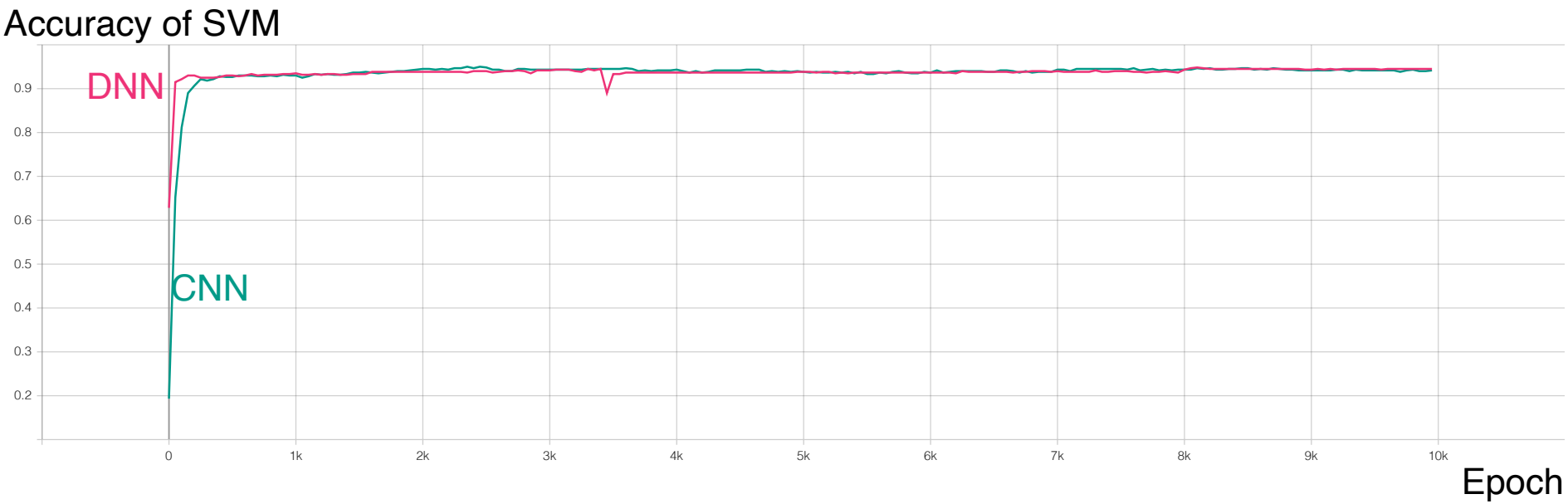
Setting

- Input: 1 box MNIST

Optimal setting

- The comparable optimal hyper-parameters setting is $L2 = 1e - 6$

L2-norm

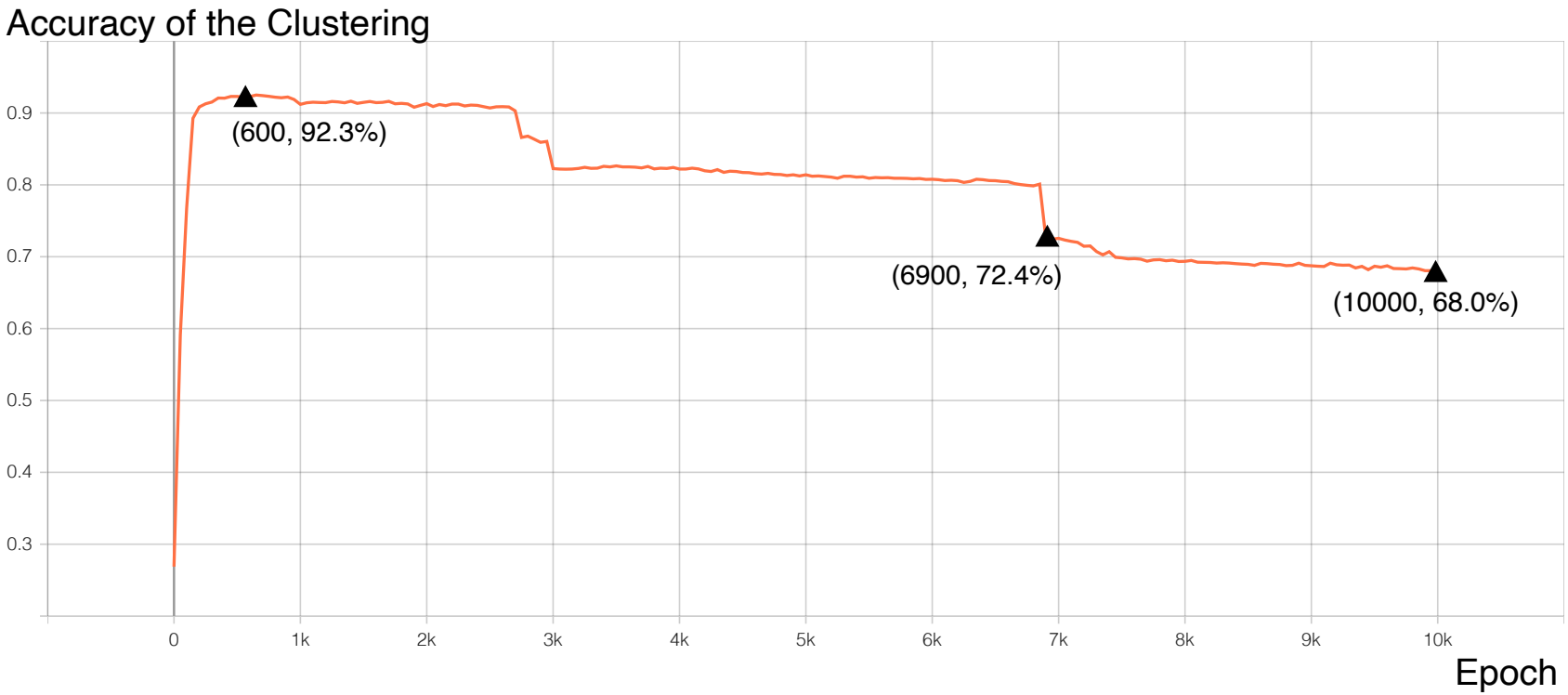


Setting

- Input: 1 box MNIST

Overfitting Evaluation

CNN model

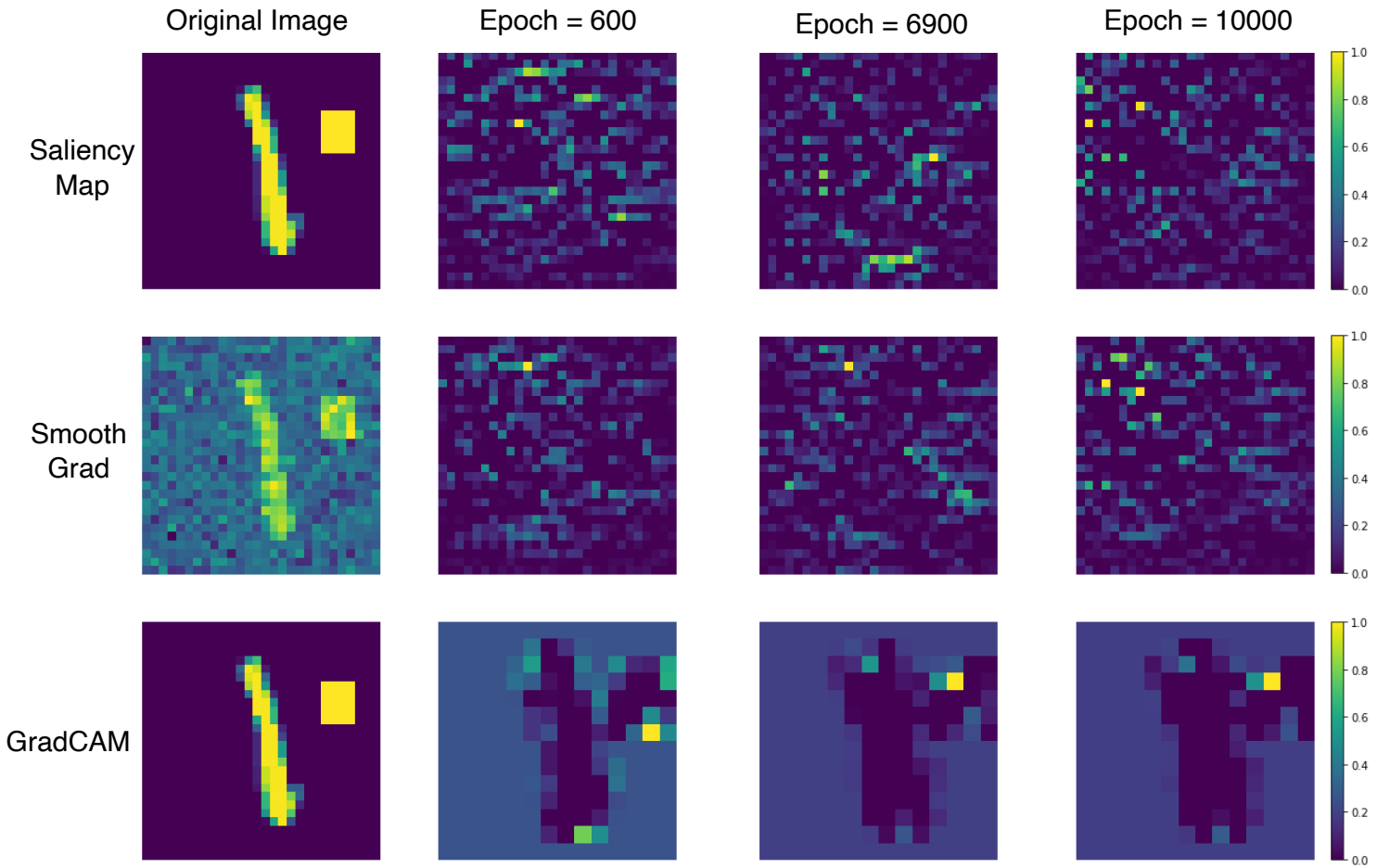


Setting

- Input: 1 box MNIST

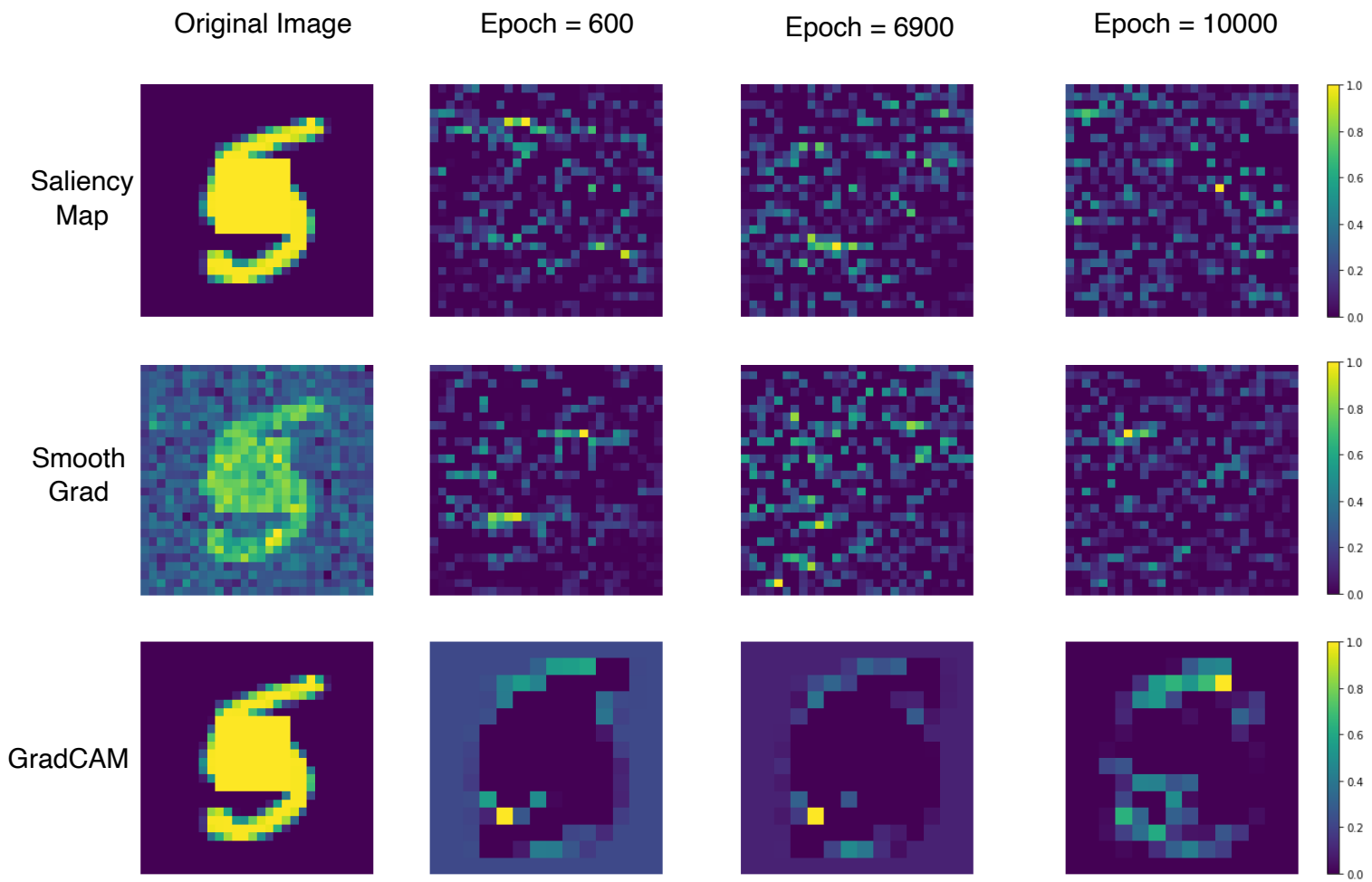
Overfitting Evaluation

CNN model



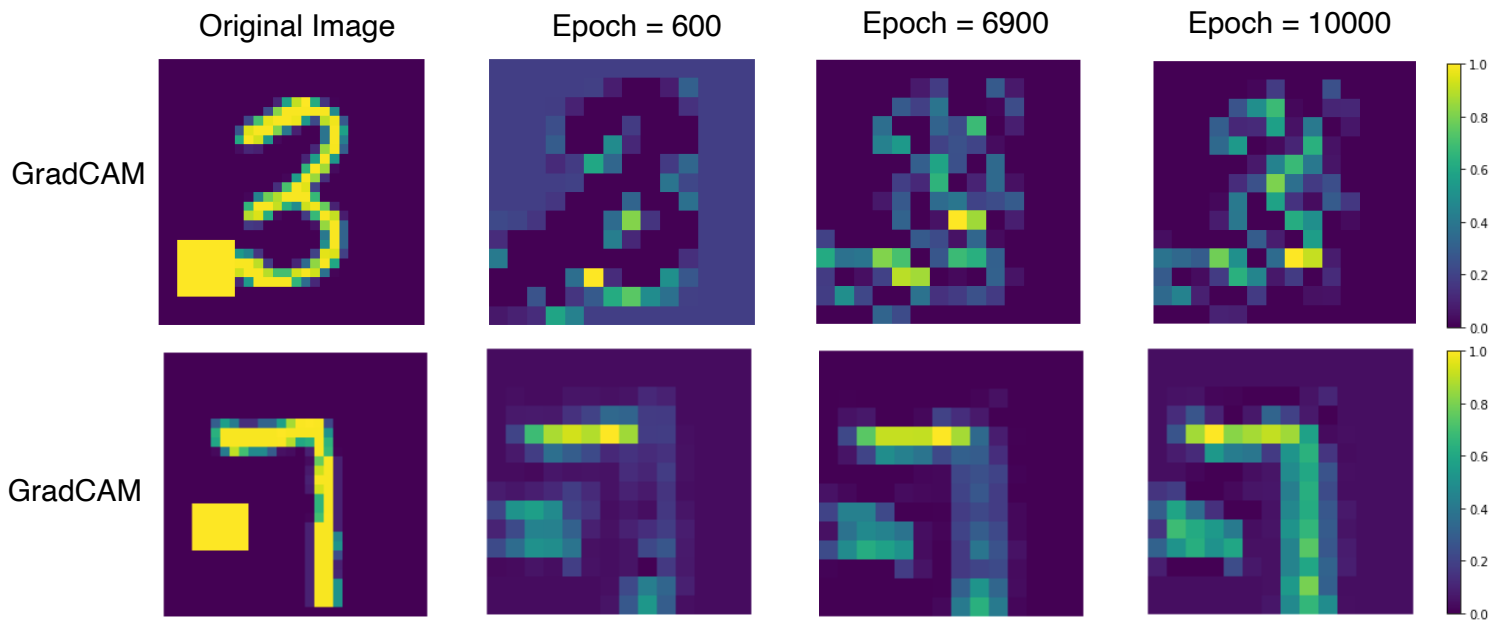
Overfitting Evaluation

CNN model



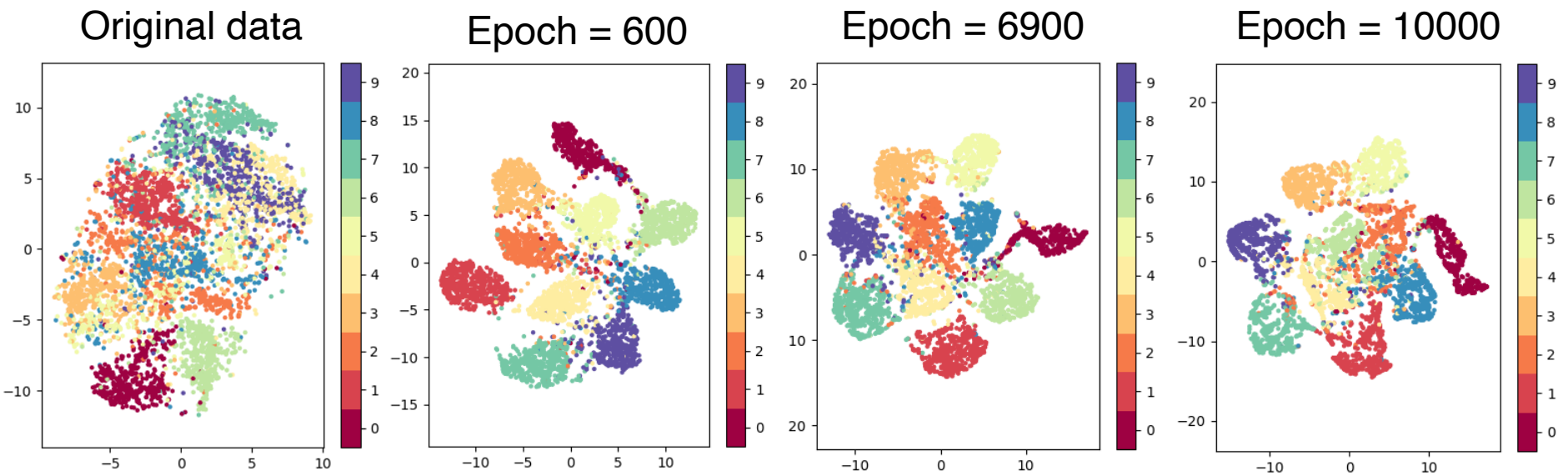
Overfitting Evaluation

CNN model



Overfitting Evaluation

CNN model

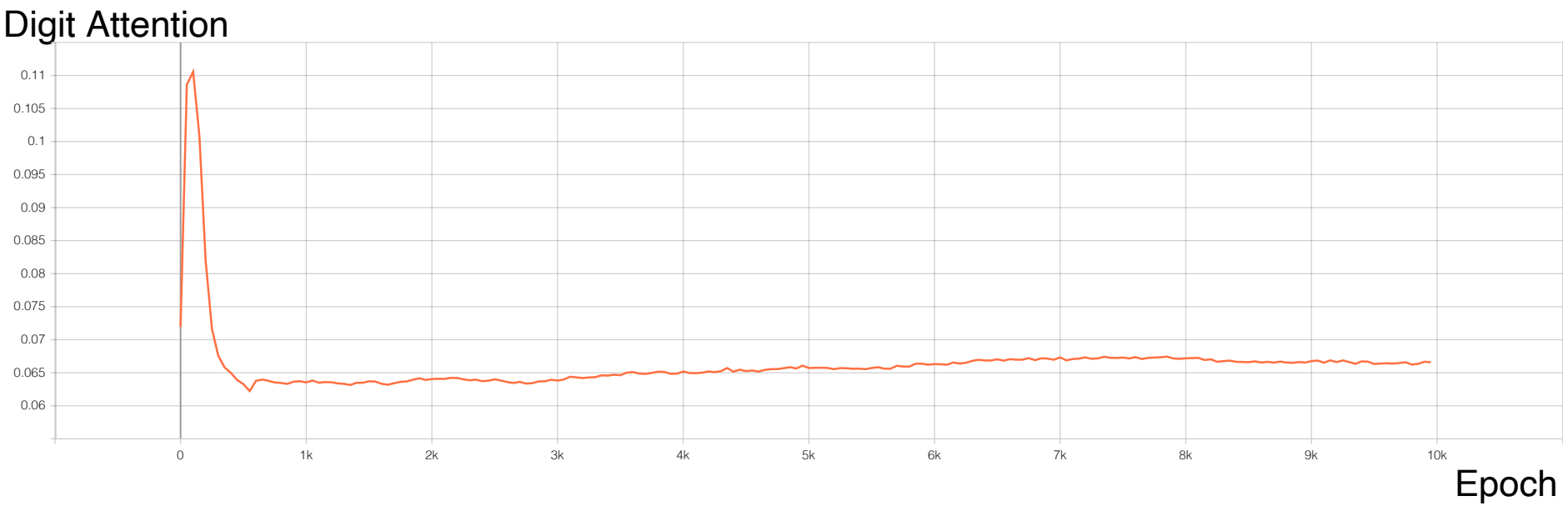


Setting

- Input: 1 box MNIST

Overfitting Evaluation

CNN model



Setting

- Input: 1 box MNIST