Visual Analytics of the Machine Learning Process

Mengchen Liu, Tsinghua University

Supervisor: Shixia Liu

Machine learning is hot

Machine Learning is Hard

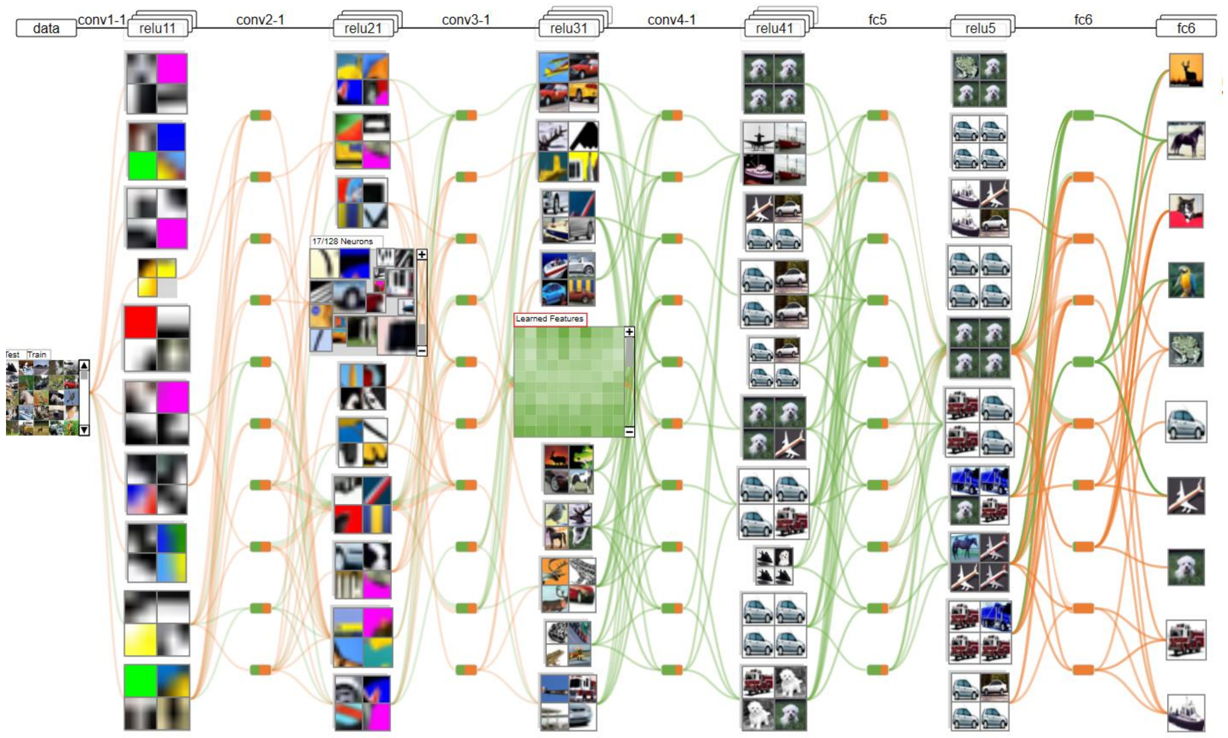
- A large amount of data is needed
  - MSCOCO: 330K images

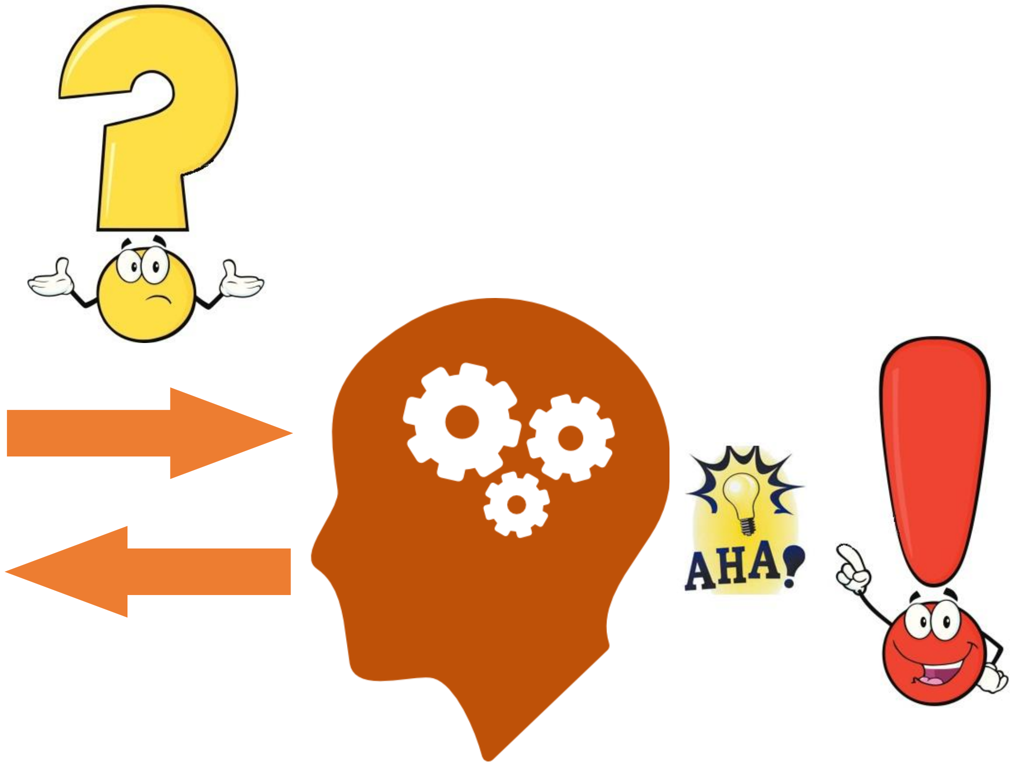
- Machine learning models are often treated as “black boxes”
  - Dozens of layers, millions of parameters

- Successful training a machine learning model needs time, skill.
  - Trial-and-error
  - Single trial
    - AlphaGo (2016): 1 week on 50 GPUs using Google Cloud

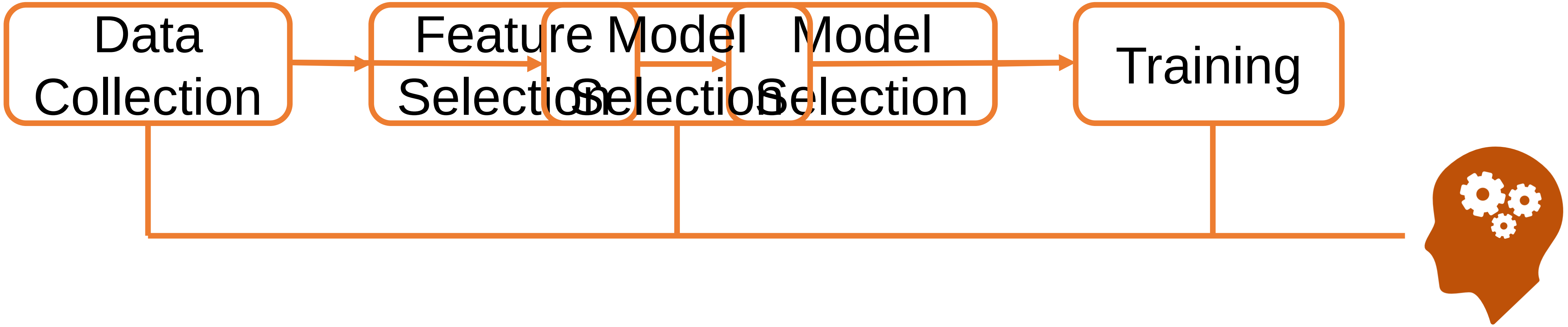
- ……

Machine Learning + Visualization



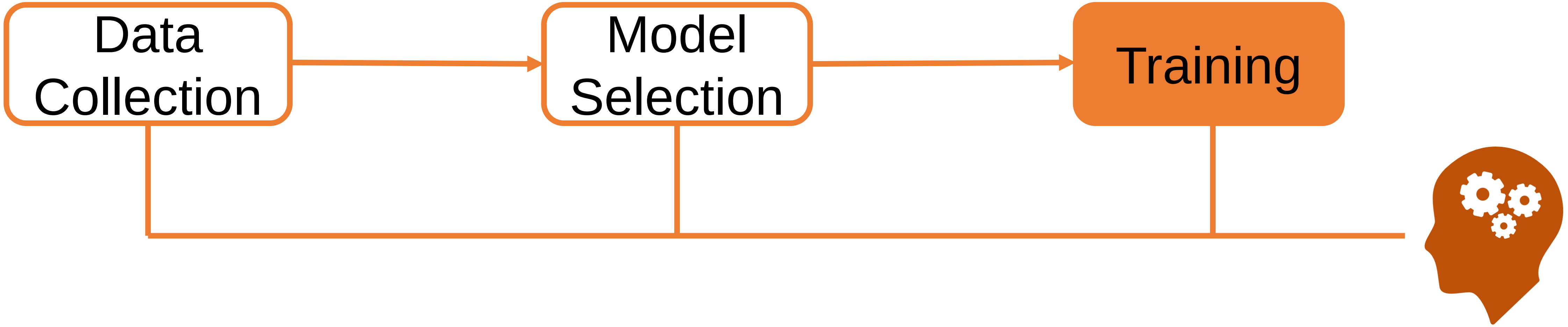


Machine Learning + Visualization



Deep Learning. Goodfellow et al.,
Pattern Classification. Duda et al.

Machine Learning + Visualization



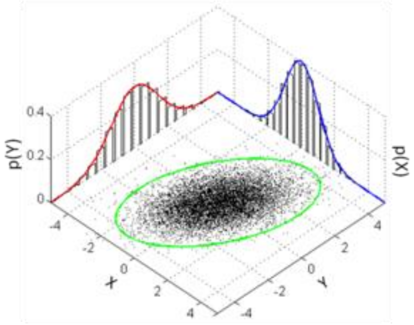
Deep Learning. Goodfellow et al.,
Pattern Classification. Duda et al.

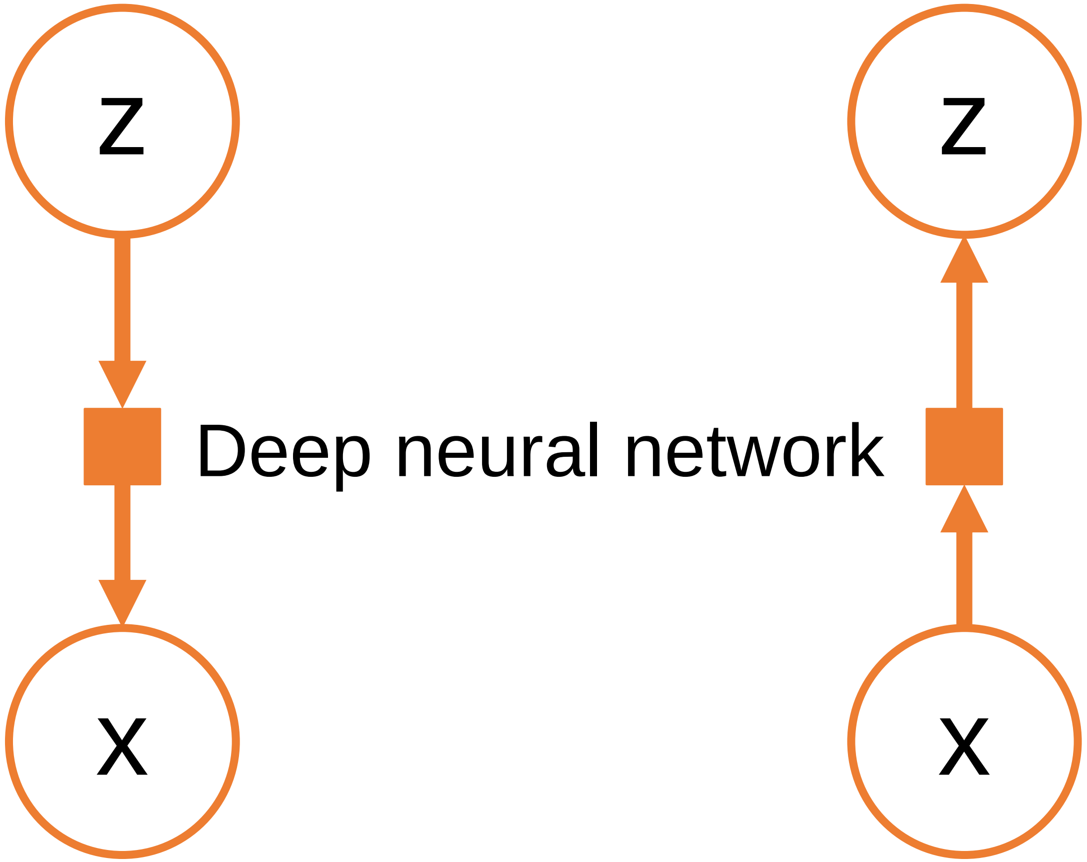
Analyzing the Training Processes of Deep Generative Networks

Mengchen Liu, Jiaxin Shi, Kelei Cao, Jun Zhu, Shixia Liu

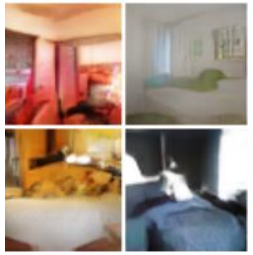
TVCG 2018

Deep Generative Model (DGM)





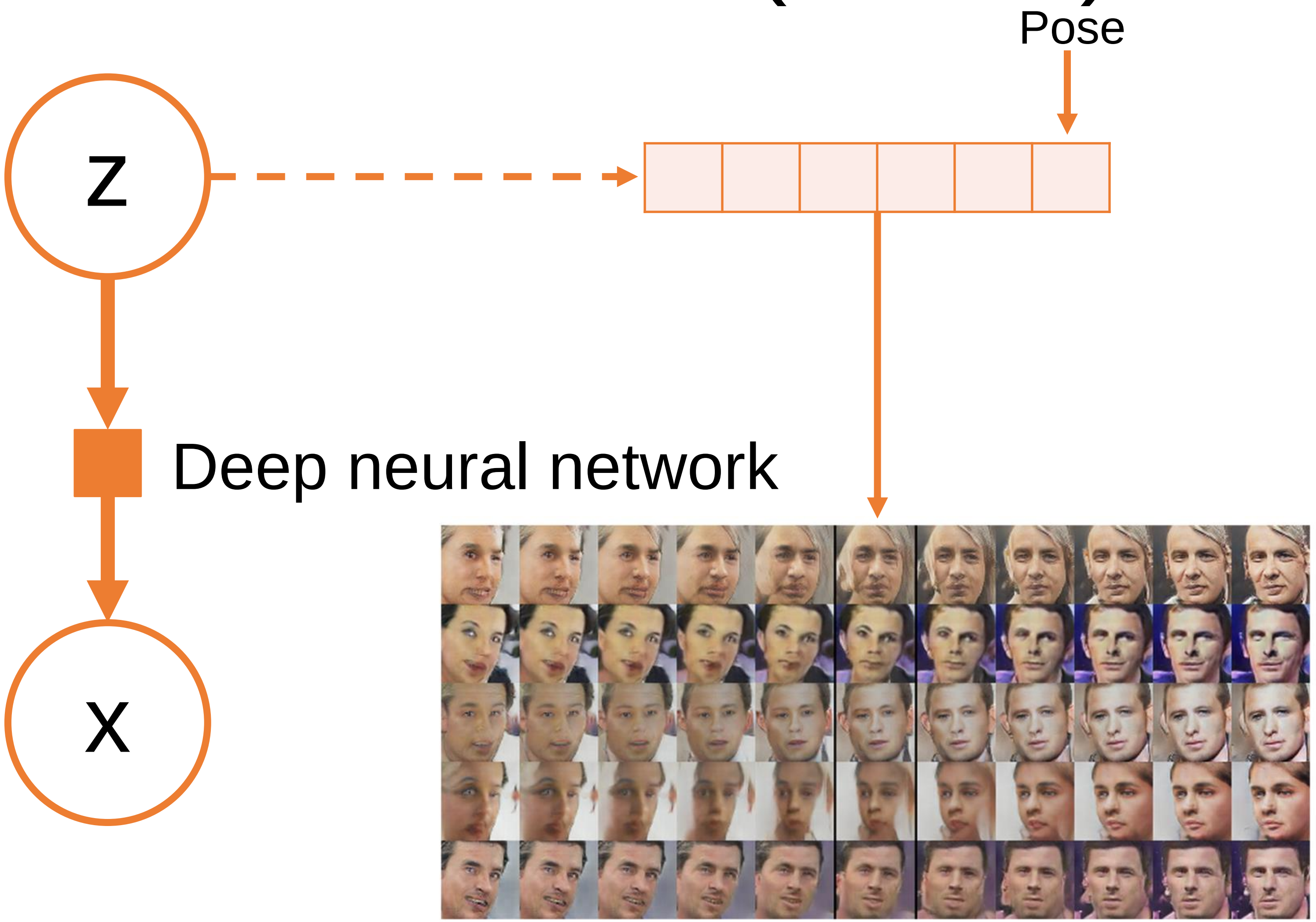
Smiling face



Unsupervised / semi-supervised

Supervised

Deep Generative Models (DGMs)

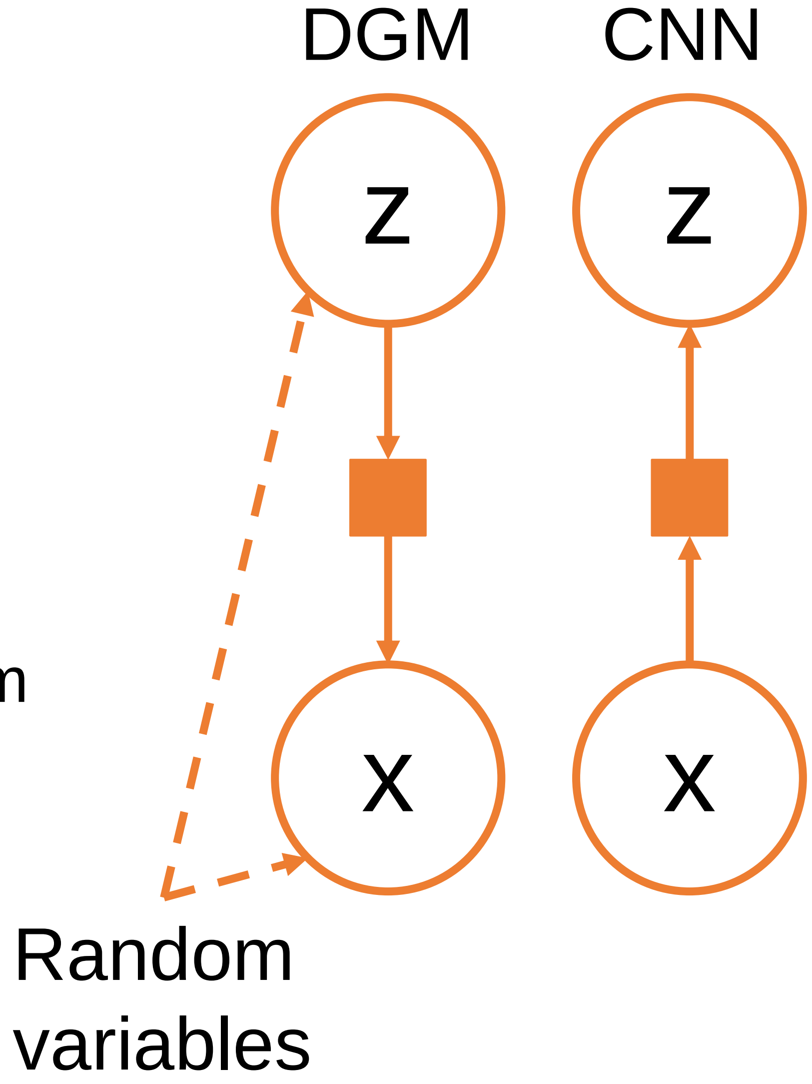


By generative adversarial network (GAN)

Training a DGM is Hard

- DGM: both deterministic functions and random variables (x, z)
  - Convolutional neural network (CNN): deterministic functions (e.g., convolution and pooling)
- DGM: a top-down generative process and a bottom-up Bayesian inference process
  - CNN: a bottom-up process: input at the bottom layer → high-level features → outputs

Visualization



Challenge 1

- Handle a large amount of time series data
  - Typical time series data: activation/gradient/weight changes over time
  - Millions of activations/gradients/weights in a DGM

Contribution 1

- Handle a large amount of time series data
  - Activation/gradient/weight changes over time
  - Millions of activations/gradients/weights in a DGM

- Line chart + **blue noise polyline sampling** algorithm
  - Each time series as a polyline
  - Select polyline samples with blue-noise properties
  - Preserve outliers and reduce visual clutter

Challenge 2

- Identify the root cause of a failed training process
  - It’s often difficult to locate the specific neurons leading to the training failure
  - Neurons influence each other

Contribution 2

- Identify the root cause of a failed training process
  - It’s often difficult to locate the specific neurons leading to the training failure
  - Neurons influence each other

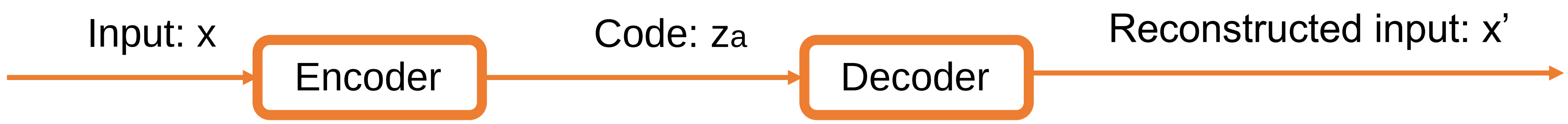
- A **credit assignment** algorithm
  - Explain how other neurons contribute to the output of the neuron causing a training problem

DGMTracker

- Better understand and diagnose the training process of a DGM

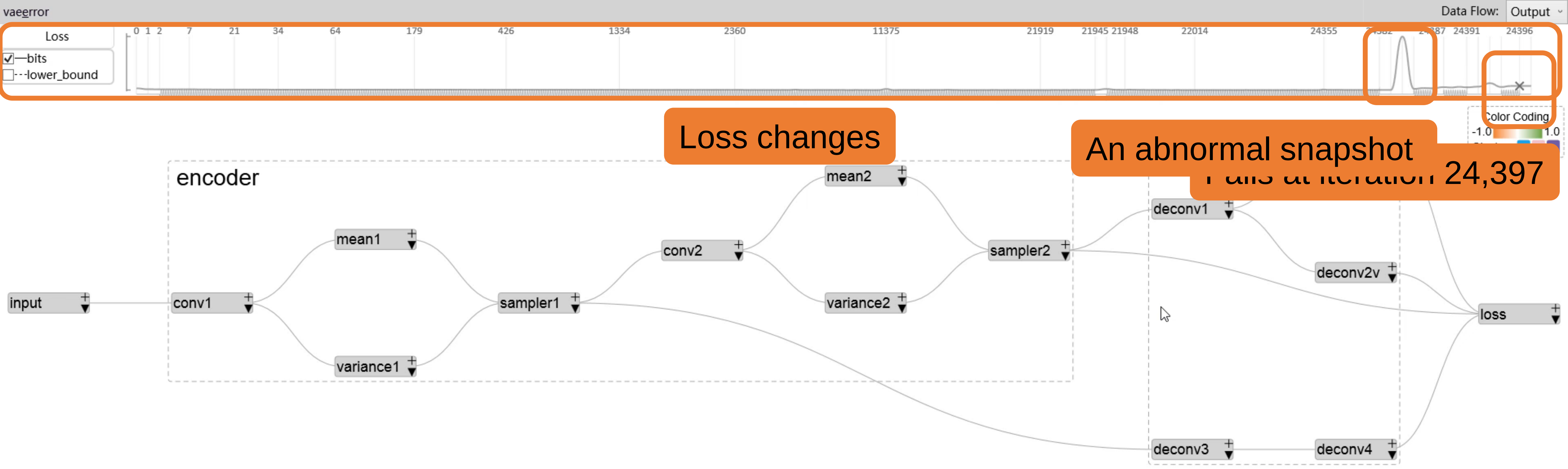
Case Study: Debugging a Failed Training Process of a Variational Autoencoder (VAE)

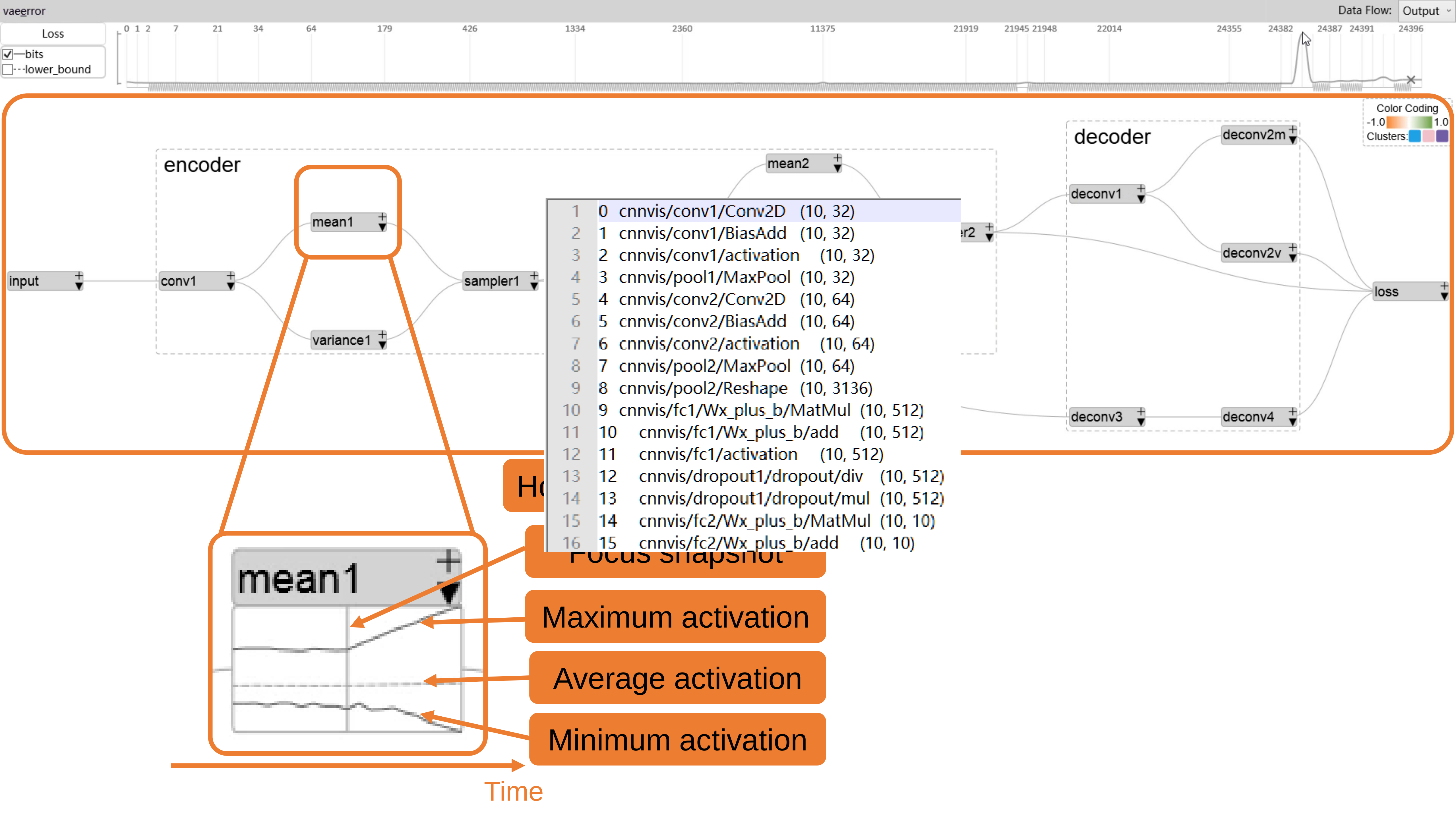
- Autoencoder
  - Reconstruct their input with minimum information loss

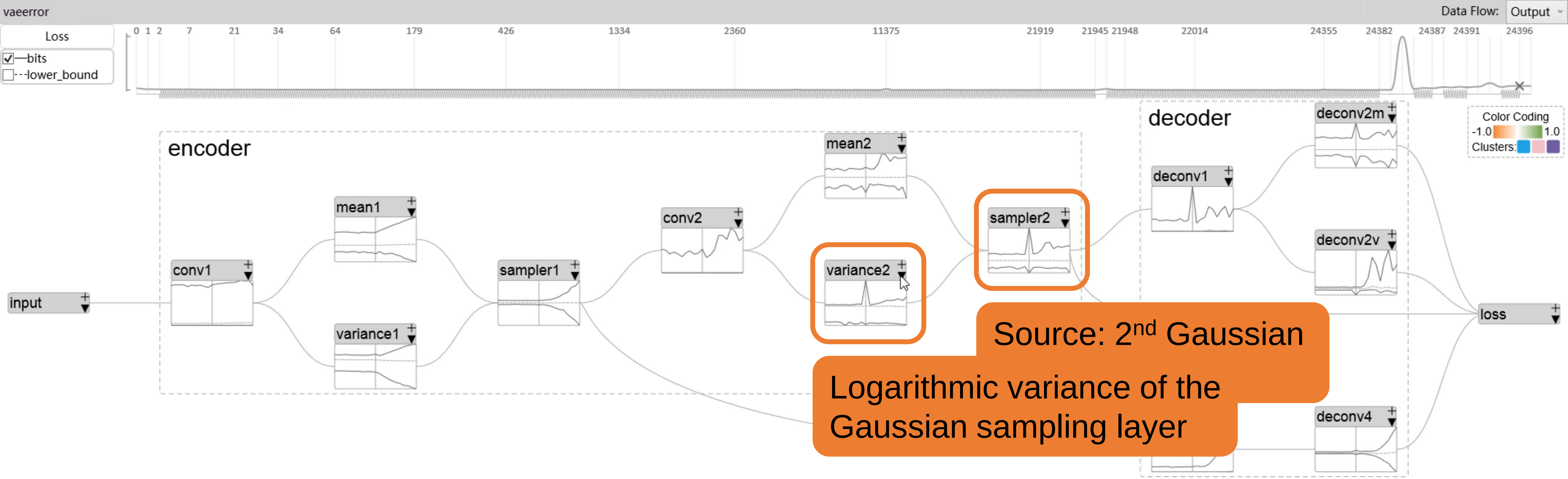


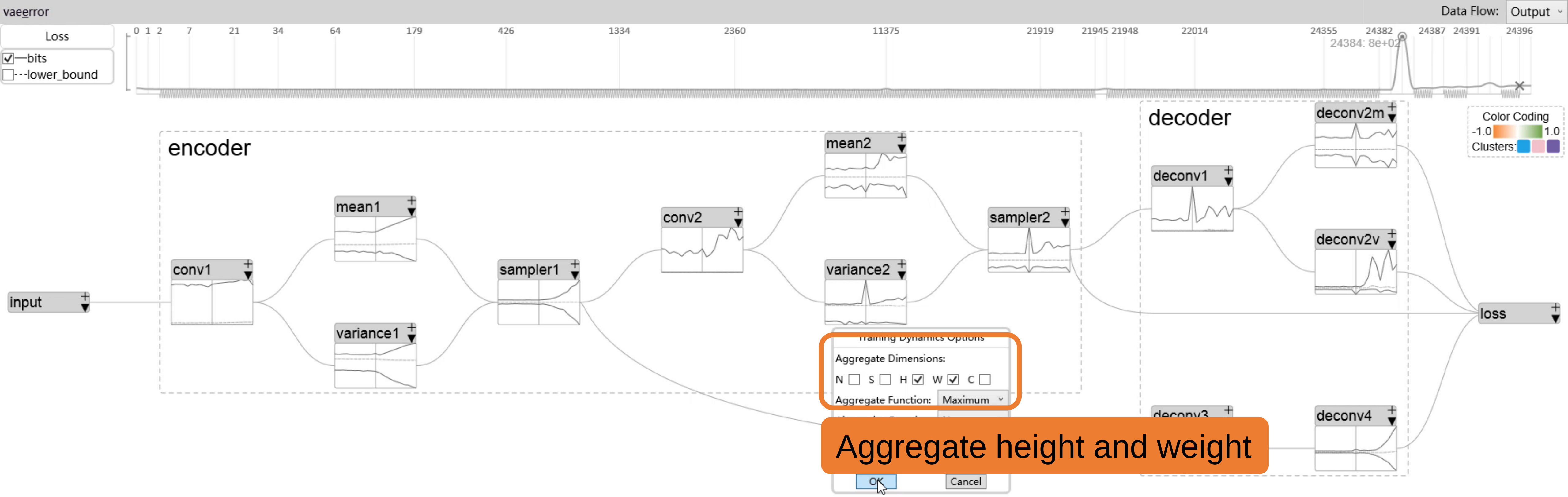
- Variational autoencoder
  - Probabilistic version of an autoencoder
  - z_v : a vector of random variables
  - z_a : a vector of real numbers

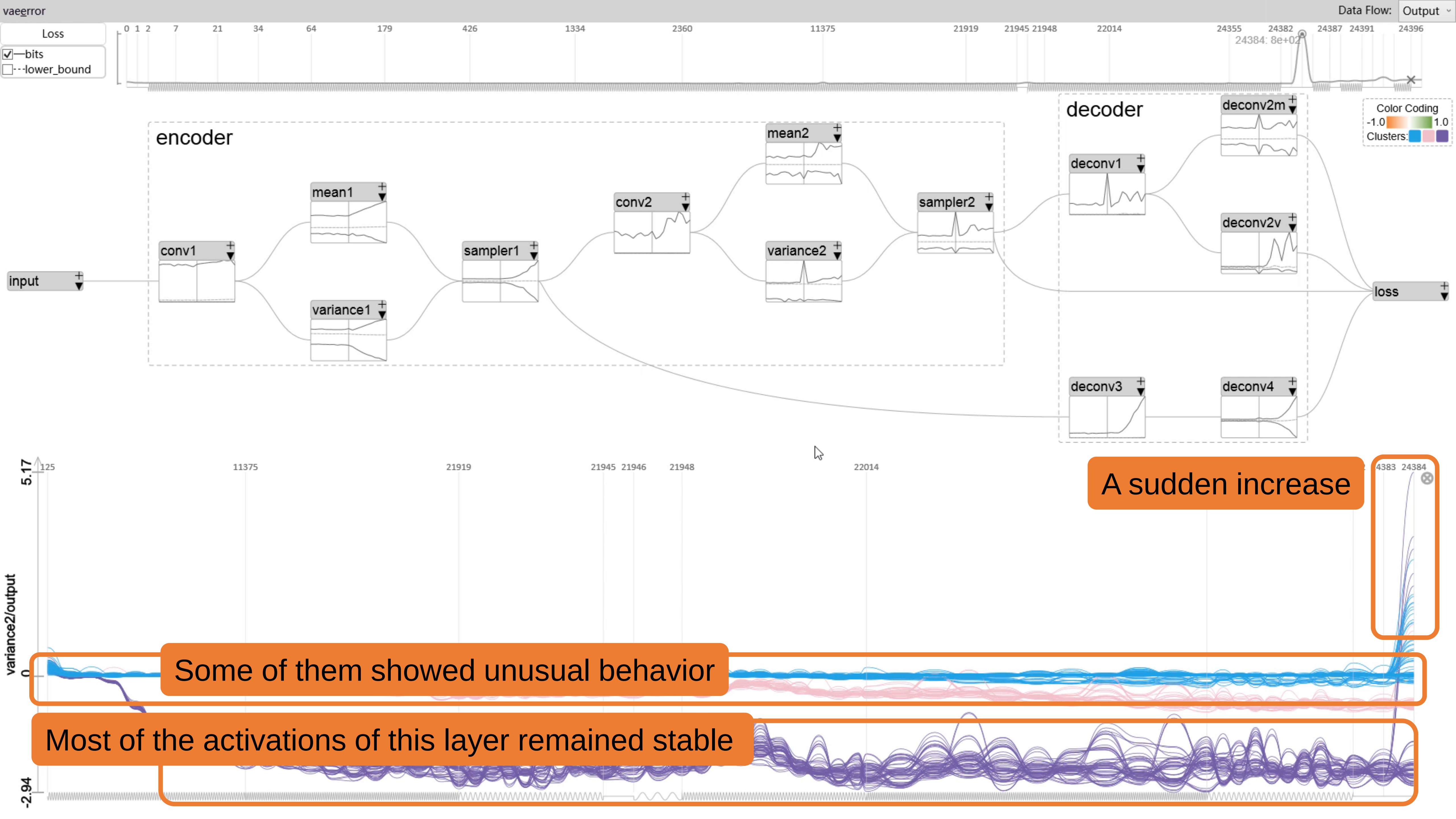
Dataset: CIFAR10 dataset
Loss = NaN (10k-30k iterations)
An example case: fails at 24,397


vaeerror
Data Flow: Output
Loss
bits
lower_bound
0 1 2 7 21 34 64 179 426 1334 2360 11375 21919 21945 21948 22014 24355 24391 24396
Loss changes
An abnormal snapshot
24,397
Color Coding
-1.0
1.0
encoder
input
conv1
mean1
variance1
sampler1
conv2
mean2
variance2
sampler2
deconv1
deconv2v
deconv3
deconv4
loss

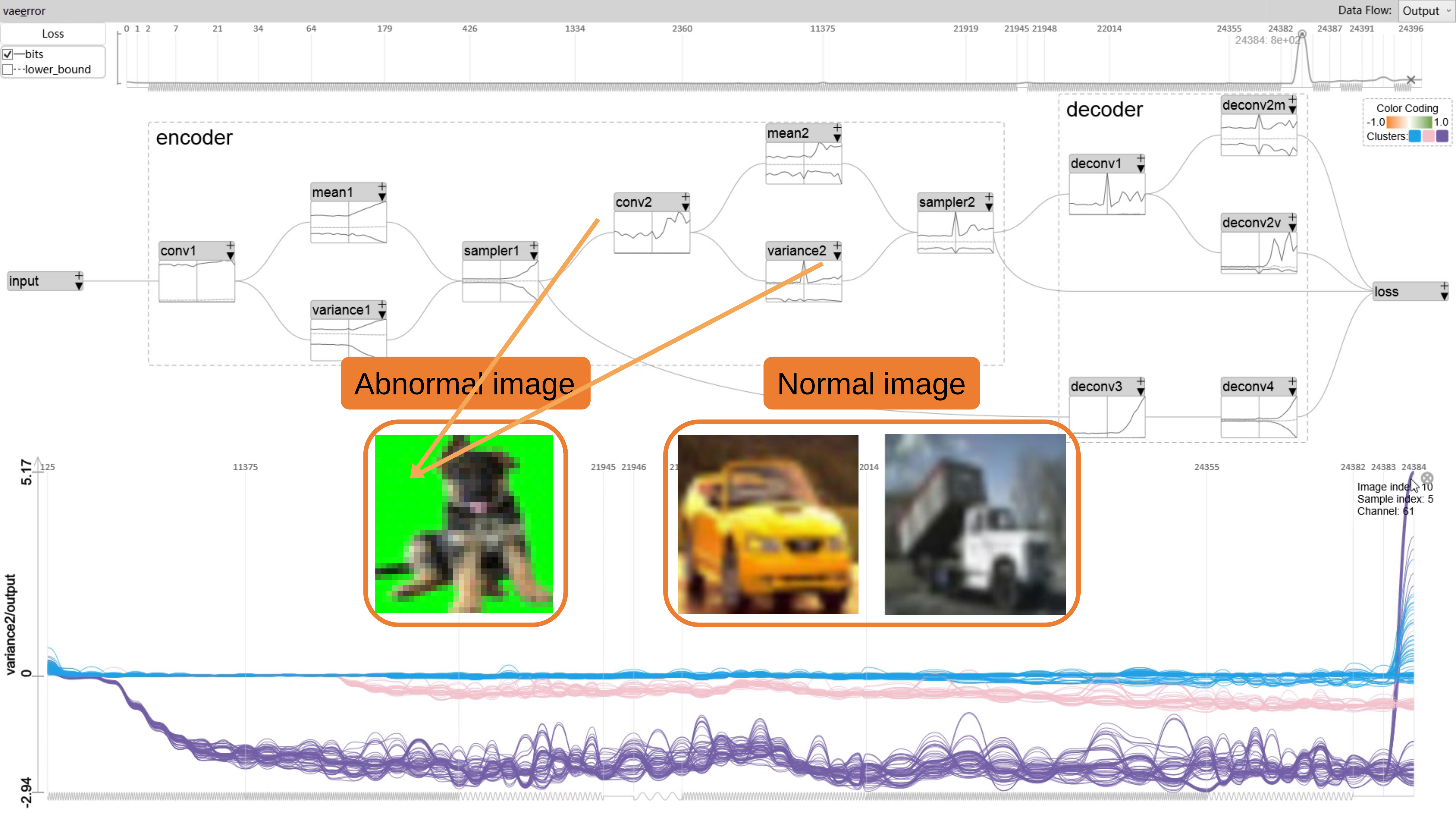

vaegrror
Data Flow: Output
Loss
bits
lower_bound
0 1 2 7 21 34 64 179 426 1334 2360 11375 21919 21945 21948 22014 24355 24382 24387 24391 24396
Color Coding
-1.0
1.0
Clusters:
encoder
decoder
input
conv1
mean1
variance1
sampler1
mean2
deconv1
deconv2m
deconv2v
deconv3
deconv4
loss
1 0 cnnvis/conv1/Conv2D (10, 32)
2 1 cnnvis/conv1/BiasAdd (10, 32)
3 2 cnnvis/conv1/activation (10, 32)
4 3 cnnvis/pool1/MaxPool (10, 32)
5 4 cnnvis/conv2/Conv2D (10, 64)
6 5 cnnvis/conv2/BiasAdd (10, 64)
7 6 cnnvis/conv2/activation (10, 64)
8 7 cnnvis/pool2/MaxPool (10, 64)
9 8 cnnvis/pool2/Reshape (10, 3136)
10 9 cnnvis/fc1/Wx_plus_b/MatMul (10, 512)
11 10 cnnvis/fc1/Wx_plus_b/add (10, 512)
12 11 cnnvis/fc1/activation (10, 512)
13 12 cnnvis/dropout1/dropout/div (10, 512)
14 13 cnnvis/dropout1/dropout/mul (10, 512)
15 14 cnnvis/fc2/Wx_plus_b/MatMul (10, 10)
16 15 cnnvis/fc2/Wx_plus_b/add (10, 10)
mean1
Maximum activation
Average activation
Minimum activation
Time


vaeerror
Data Flow: Output
Loss
bits
lower_bound
0 1 2 7 21 34 64 179 426 1334 2360 11375 21919 21945 21948 22014 24355 24382 24387 24391 24396
encoder
decoder
input
conv1
mean1
variance1
sampler1
conv2
mean2
variance2
sampler2
deconv1
deconv2m
deconv2v
deconv4
loss
Color Coding
-1.0
1.0
Clusters:
Source: 2nd Gaussian
Logarithmic variance of the Gaussian sampling layer


vaeerror
Data Flow: Output
Loss
bits
lower_bound
0 1 2 7 21 34 64 179 426 1334 2360 11375 21919 21945 21948 22014 24355 24382 24387 24391 24396
24384: 8e+02
encoder
decoder
input
conv1
mean1
variance1
sampler1
conv2
mean2
variance2
sampler2
deconv1
deconv2m
deconv2v
deconv3
deconv4
loss
Color Coding
-1.0
1.0
Clusters:
Training Dynamics Options
Aggregate Dimensions:
N S H W C
Aggregate Function: Maximum
OK
Cancel
Aggregate height and weight

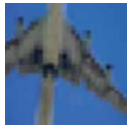


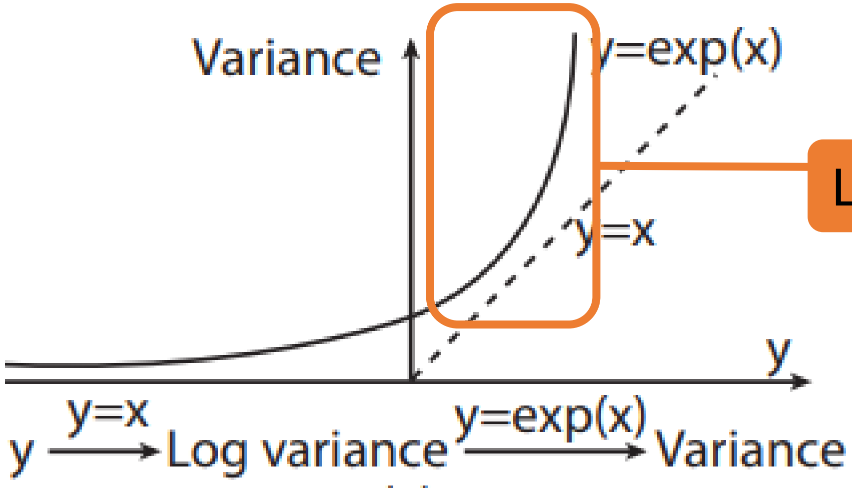
vaeerror
Data Flow: Output
Loss
bits
lower_bound
encoder
decoder
input
conv1
mean1
variance1
sampler1
conv2
mean2
variance2
sampler2
deconv1
deconv2m
deconv2v
deconv3
deconv4
loss
Color Coding
-1.0
1.0
Clusters:
variance2/output
5.17
0
-2.94
A sudden increase
Some of them showed unusual behavior
Most of the activations of this layer remained stable



vaeerror
Data Flow: Output
Loss
bits
lower_bound
encoder
decoder
input
conv1
mean1
variance1
sampler1
conv2
mean2
variance2
sampler2
deconv1
deconv2m
deconv2v
deconv3
deconv4
loss
Color Coding
-1.0
1.0
Clusters:
Abnormal image
Normal image
variance2/output
5.17
0
-2.94
Image index: 10
Sample index: 5
Channel: 61

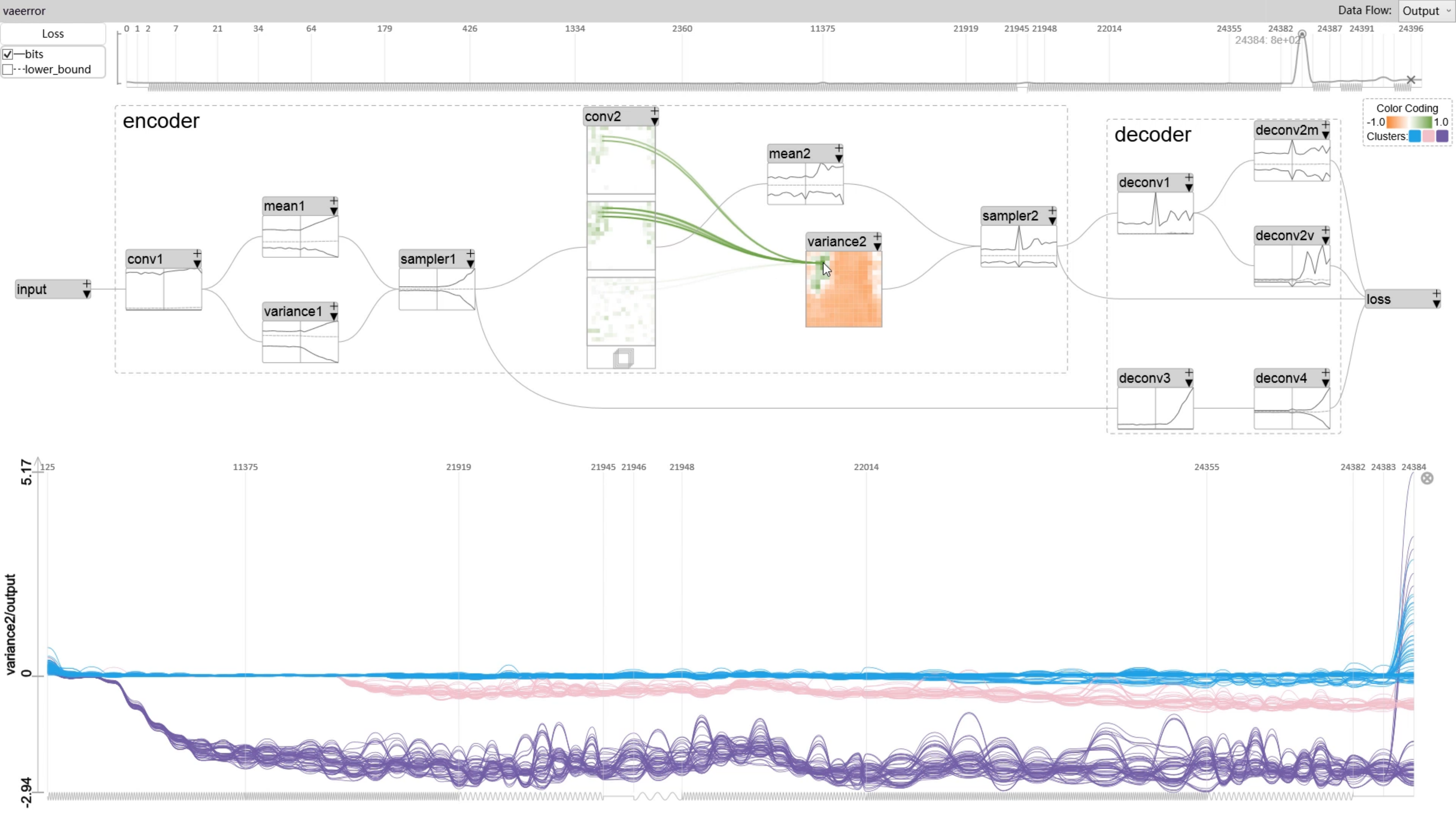
Solution

- Trial 1:
  - Replacing  with , but the network failed again
  - By the same analysis, we find another "bad" image 
- Trial 2:

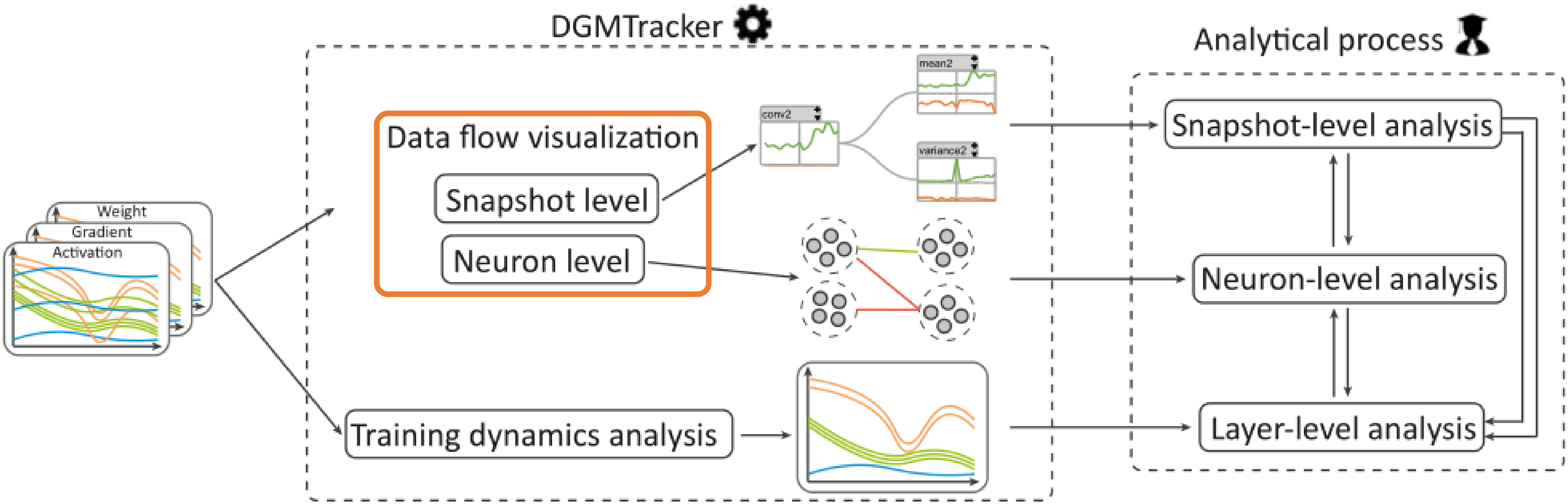


Much smoother

Large variance → large samples → large increase in loss


vaeerror
Data Flow: Output
Loss
bits
lower_bound
24384: 8e+02
encoder
conv2
mean2
mean1
conv1
sampler1
variance2
sampler2
variance1
input
decoder
deconv2m
deconv1
deconv2v
loss
deconv3
deconv4
Color Coding
-1.0
1.0
Clusters:
variance2/output
5.17
0
-2.94

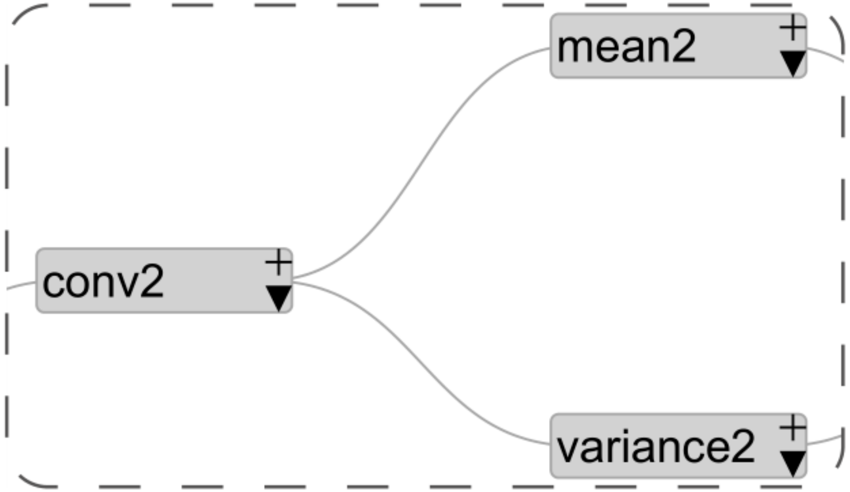
DGMTracker



Weight
Gradient
Activation
DGMTracker
Data flow visualization
Snapshot level
Neuron level
conv2
mean2
variance2
Training dynamics analysis
Analytical process
Snapshot-level analysis
Neuron-level analysis
Layer-level analysis

Snapshot-Level Visualization

- How data flows through a network

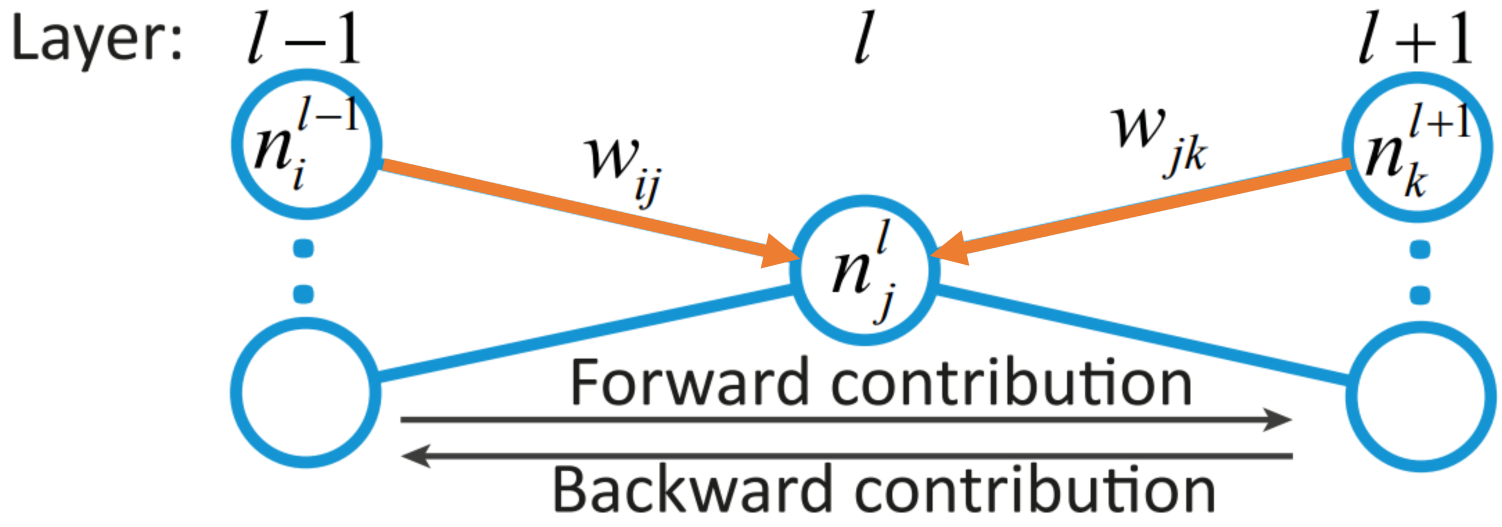


DAG visualization for DGM structure

Line chart for representing data flow

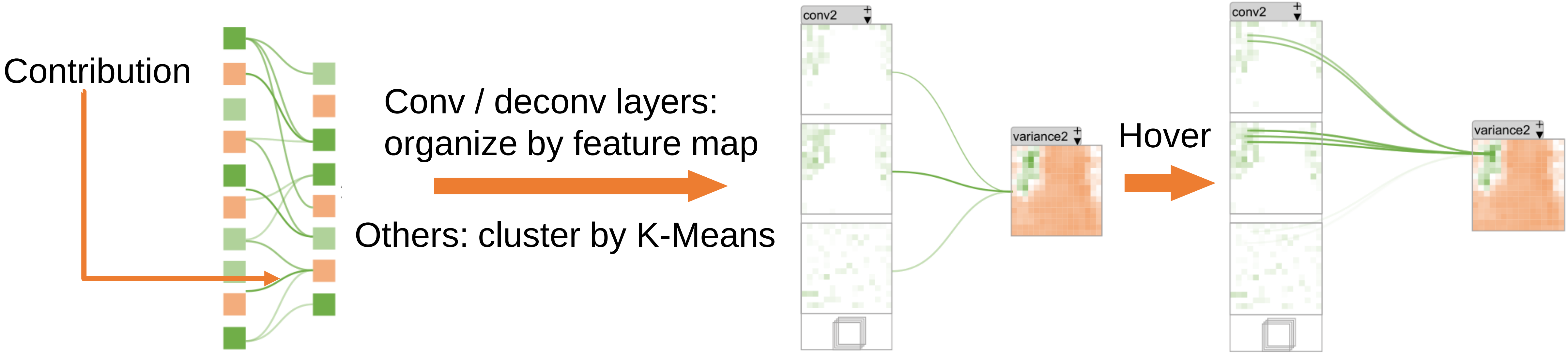
Neuron Level Visualization

- Computing and presenting how other neurons contribute to the output of the neuron being explored
  - Forward contribution: based on Layer-wise Relevance Propagation (LRP)
  - Backward contribution: based on backpropagation (BP)

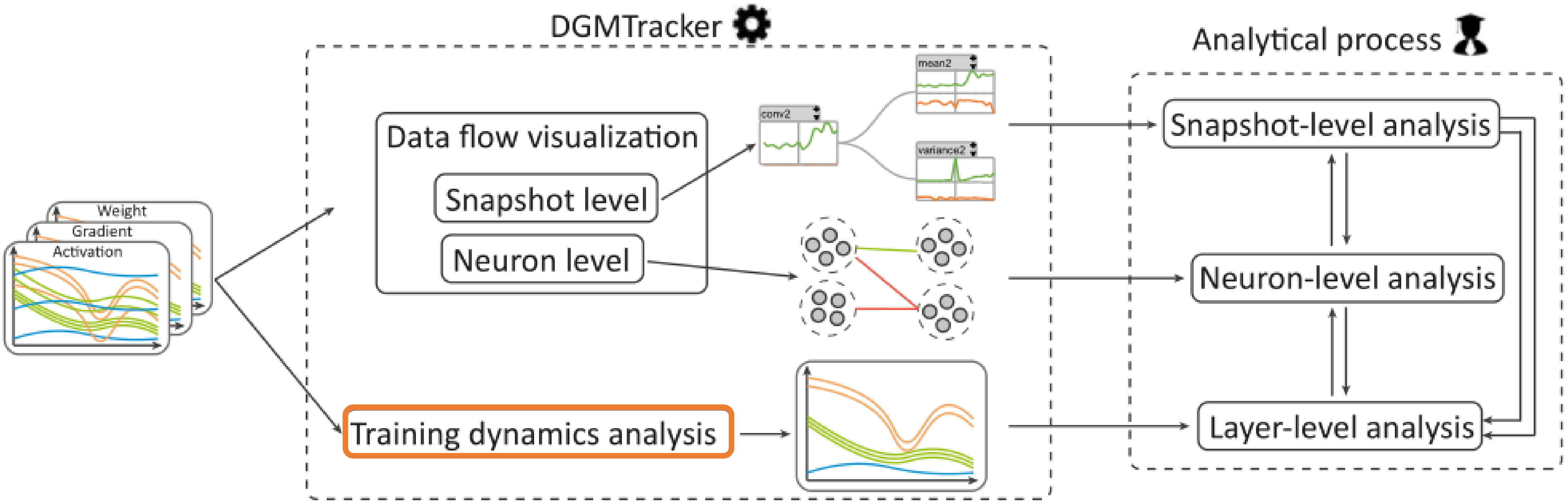


Neuron Level Visualization

- Computing and presenting how other neurons contribute to the output of the neuron being explored



DGMTracker



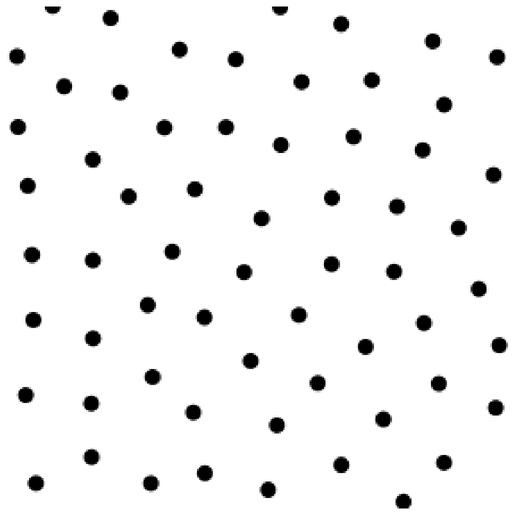
DGMTracker
Analytical process
Weight
Gradient
Activation
Data flow visualization
Snapshot level
Neuron level
Training dynamics analysis
conv2
mean2
variance2
Snapshot-level analysis
Neuron-level analysis
Layer-level analysis

Training Dynamics Analysis

- Employ a line chart to visually convey the training dynamics
  - Training dynamics: activation/gradient/weight changes over time
- Challenge
  - Visual clutter caused by a large amount of time series data
- Solution: ? polyline sampling
  - Both reduce visual clutter and preserve outliers

Blue Noise Sampling

- The selected samples have blue-noise properties
  - The selected samples are located **randomly** and **uniformly** in the space
- Compared with traditional random sampling
  - Low sampling rate in the high-density regions
    - Reduce visual clutter
  - High sampling rate in the low-density regions
    - Preserve outliers



Blue noise sampling example

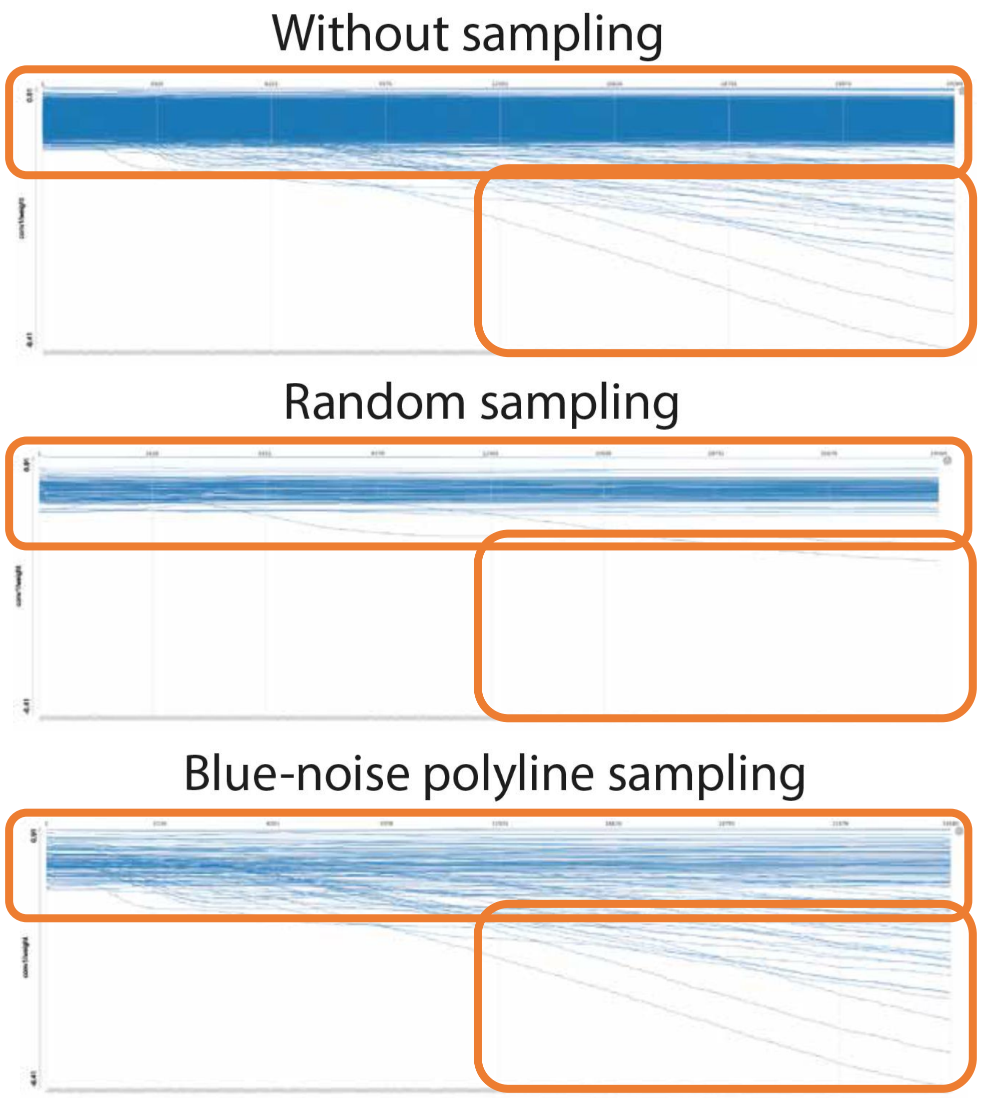
Blue-Noise Polyline Sampling

- State-of-the-art: blue-noise line segment sampling [Sun et al., 2013]
  - Sample a line segment → if the distance with others is large enough, accept, or reject
- Intuitive method
  - Selecting line segment samples with blue-noise properties
  - Selecting polylines that contain the selected line segments as samples
- Solution: select “complete” polylines
  - How to select a polyline sample
  - How to compute the distance between two polylines

Blue-Noise Polyline Sampling (cont'd)

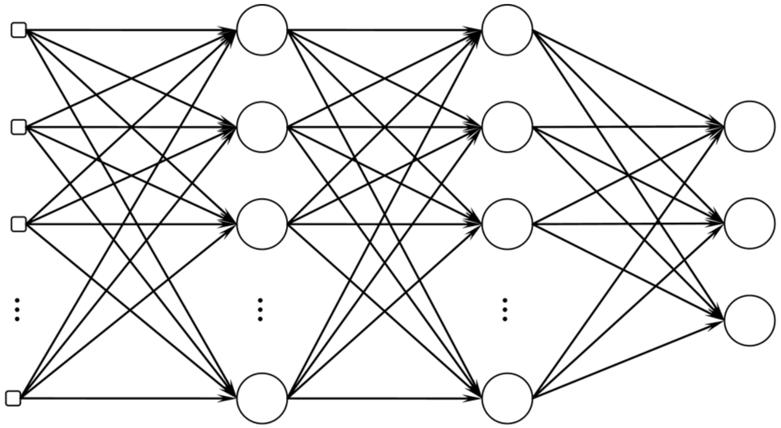
- How to select a polyline sample
  - Solution: select the polyline that can make the samples the most balanced in direction
- How to compute the distance between two polylines
  - Solution: sum of distances between corresponding line segments

$$d(L_1, L_2) = \frac{1}{N_S} \sum_{i=1}^{N_S} d_C(s_1^i, s_2^i)$$

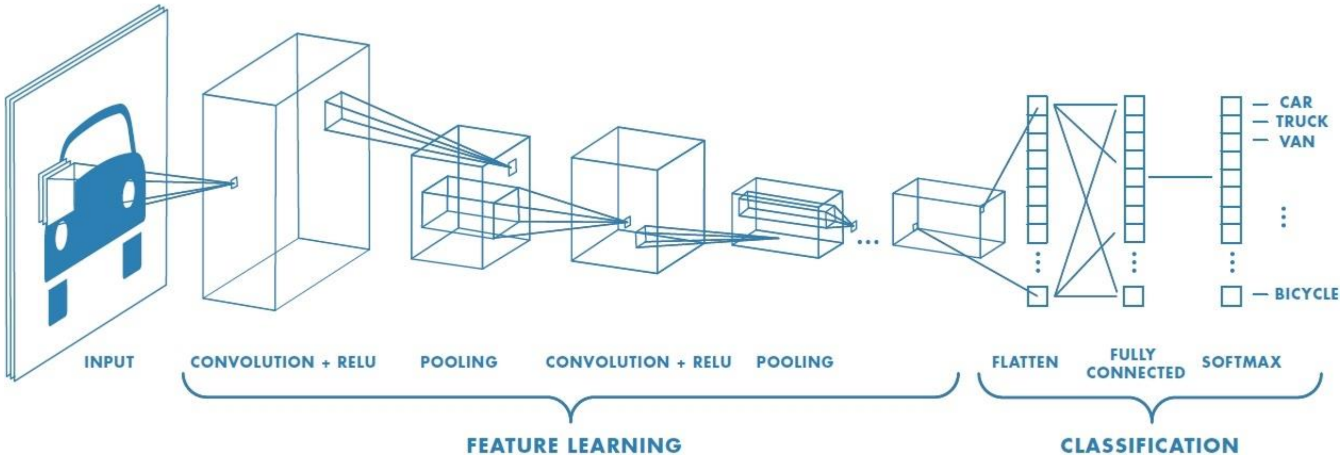


Generalization

- DGMTracker can be directly extended to other models, such as CNNs and MLPs
  - Often a base component of a DGM



Multiplayer perceptrons (MLPs)



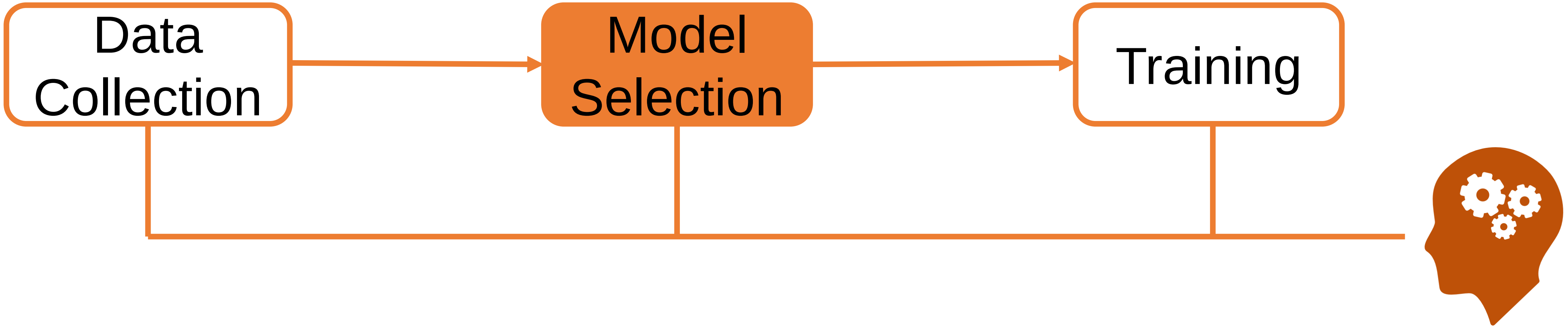
Convolutional neural networks (CNNs)

Conclusion and Future Work

- We have developed a visual analytics tool, DGMTracker, to facilitate machine learning experts in better understanding and diagnosing DGMs.

- Future work
  - Offline analysis → online analysis
  - Employ pattern mining techniques to disclose interesting patterns
  - Further reduce the amount of data needed for analysis: information theory
    - Originally 6TB → currently 6GB

Machine Learning + Visualization



Deep Learning. Goodfellow et al.,
Pattern Classification. Duda et al.

Towards Better Analysis of Deep Convolutional Neural Networks

Mengchen Liu, Jiaxin Shi, Zhen Li, Chongxuan Li, Jun Zhu, Shixia Liu

TVCG 2017

Challenges

One key factor: two deep CNNs

Convolution

Policy network

Value network

Why do the neural networks work?

May 2017

DeepMind challenge match

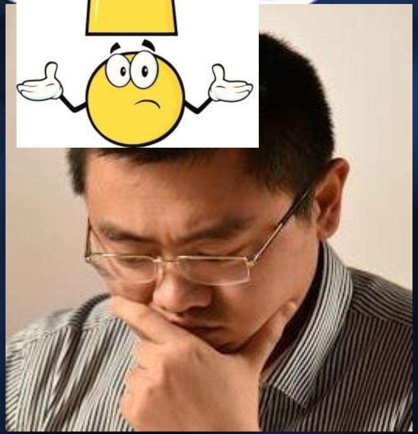
Jie Ke (9p)

Beats

Beats

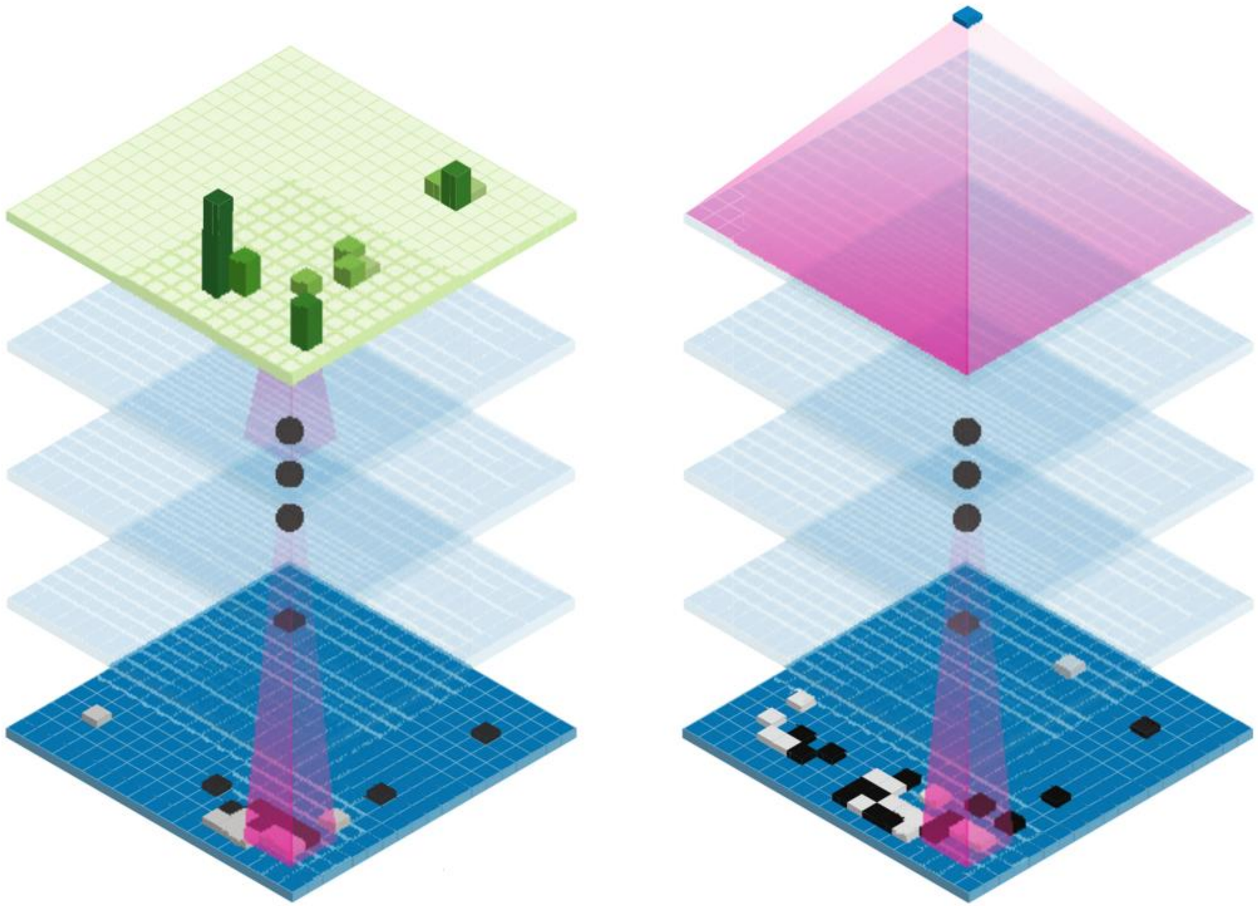
Oct 2015

Nature match



Fan Hui (2p)

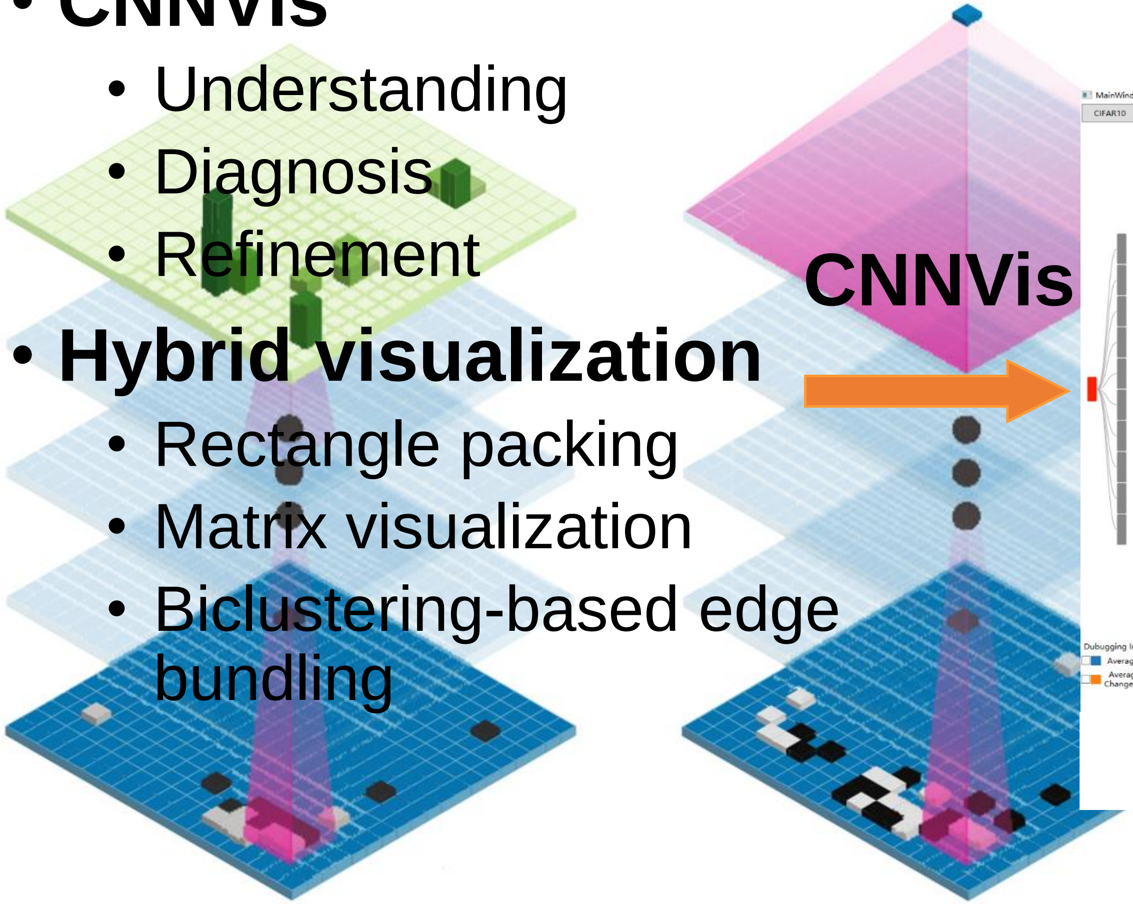
Challenges



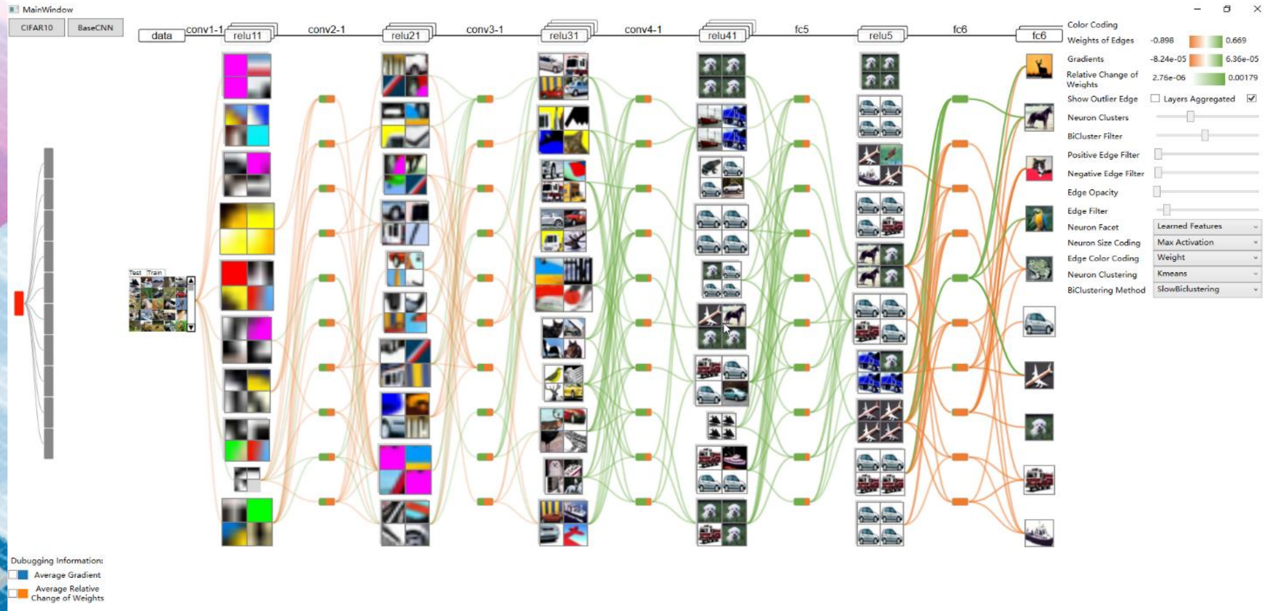
- The size of a CNN is large
  - Tens or hundreds of layers (depth)
  - Thousands of neurons in each layer (width)
  - Millions of connections between neurons
- Many functional components
  - Their values and roles are not well understood

Contributions

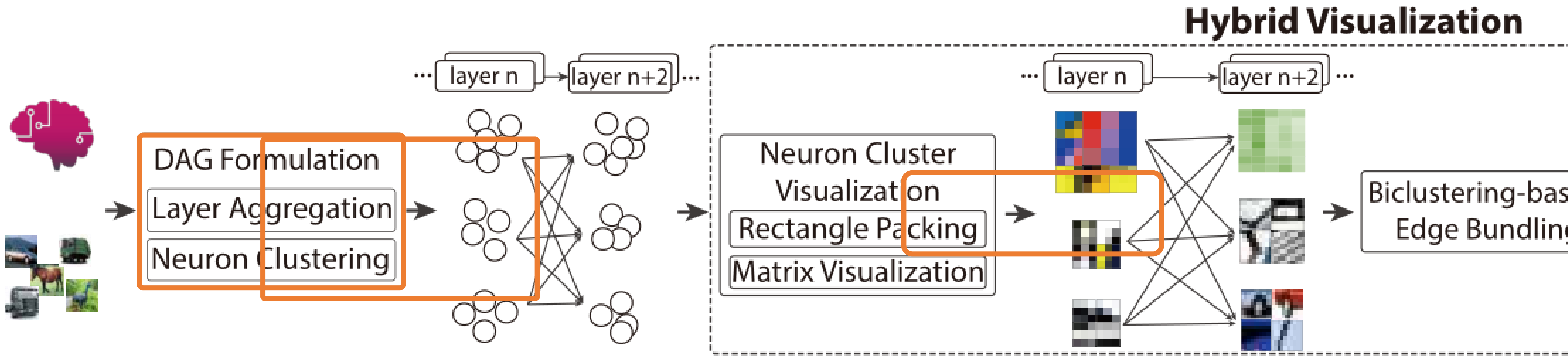
- **CNNVis**
  - Understanding
  - Diagnosis
  - Refinement
- **Hybrid visualization**
  - Rectangle packing
  - Matrix visualization
  - Biclustering-based edge bundling

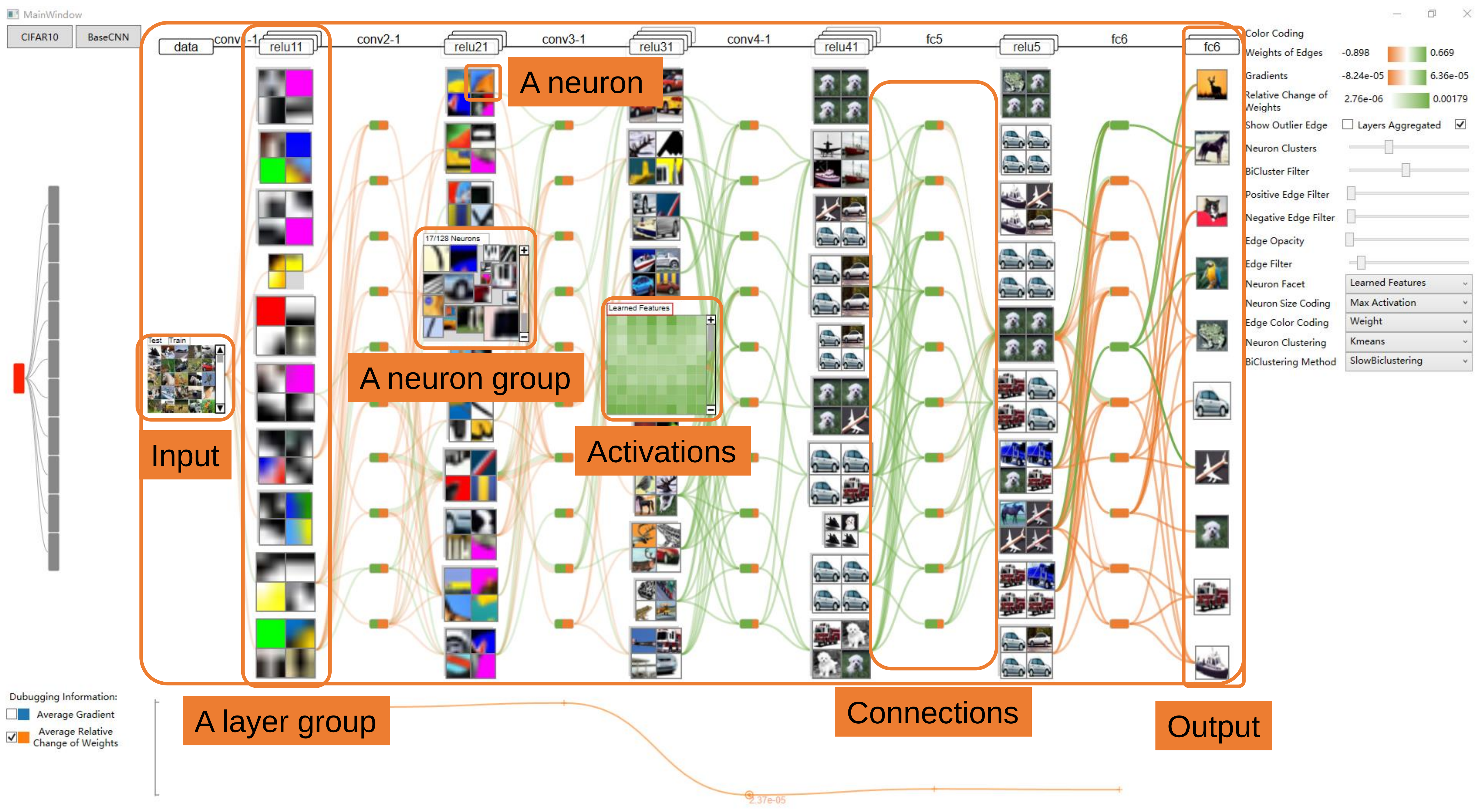


CNNVis

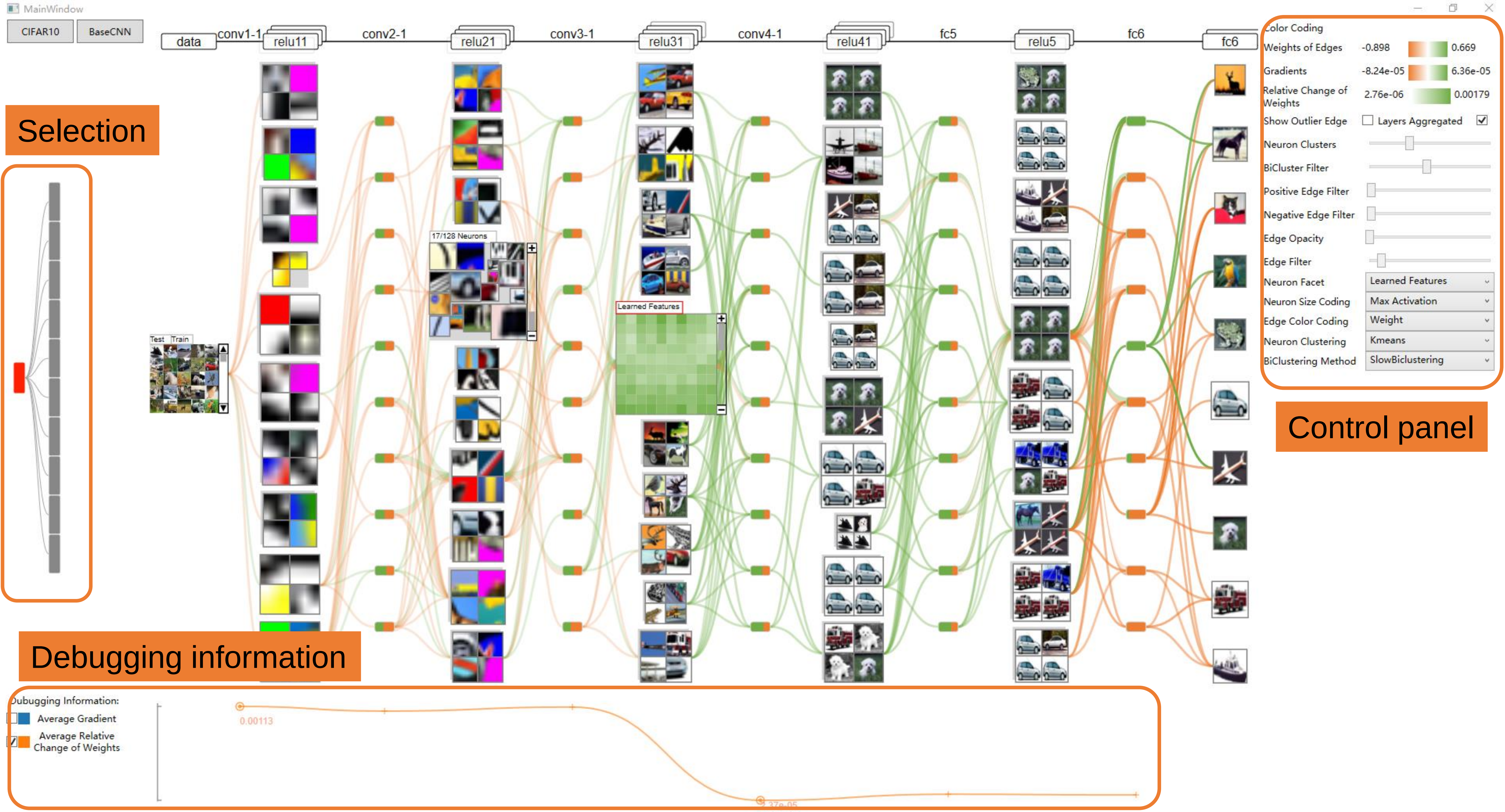


CNNVis Overview



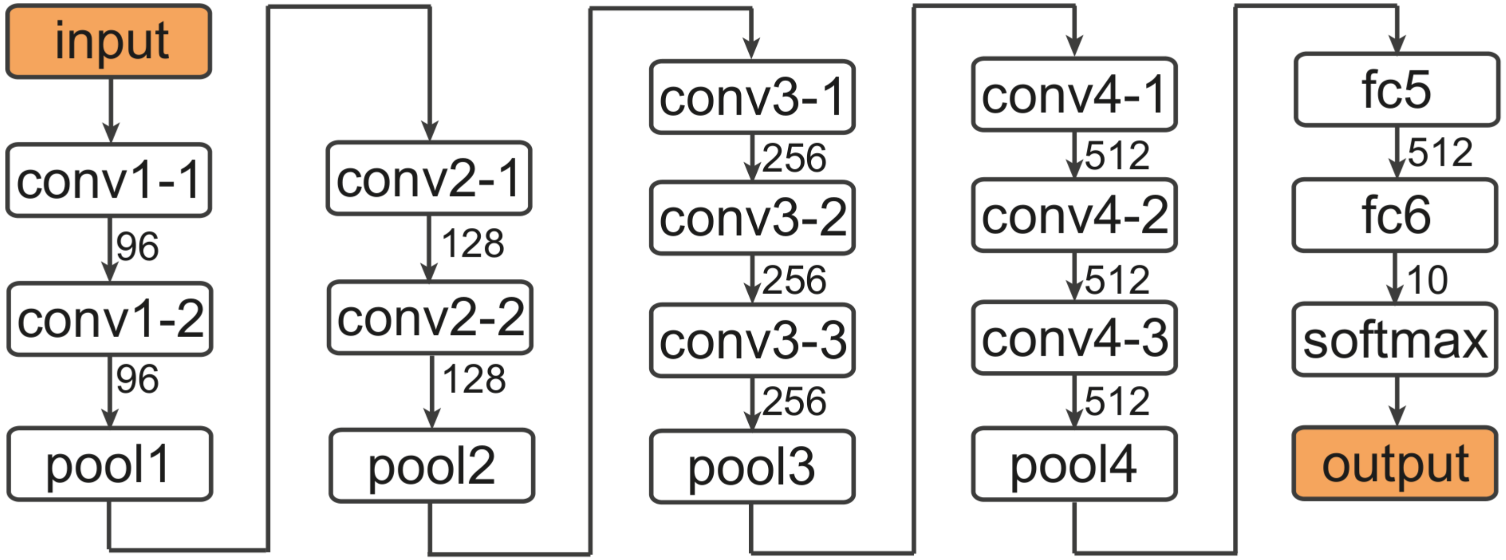


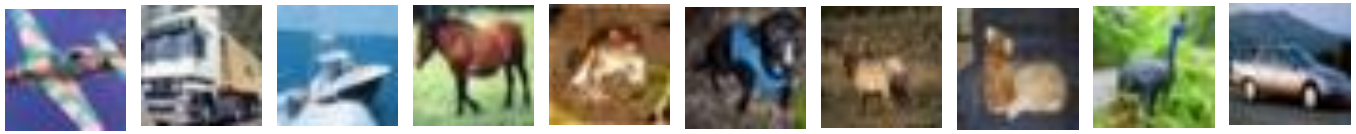
MainWindow
CIFAR10
BaseCNN
data
relu11
conv2-1
relu21
conv3-1
relu31
conv4-1
relu41
fc5
relu5
fc6
fc6
A neuron
17/128 Neurons
A neuron group
Learned Features
Activations
Test
Train
Input
Connections
Output
A layer group
Color Coding
Weights of Edges
-0.898
0.669
Gradients
-8.24e-05
6.36e-05
Relative Change of Weights
2.76e-06
0.00179
Show Outlier Edge
Layers Aggregated
Neuron Clusters
BiCluster Filter
Positive Edge Filter
Negative Edge Filter
Edge Opacity
Edge Filter
Neuron Facet
Learned Features
Neuron Size Coding
Max Activation
Edge Color Coding
Weight
Neuron Clustering
Kmeans
BiClustering Method
SlowBiclustering
Dubugging Information:
Average Gradient
Average Relative Change of Weights
2.37e-05

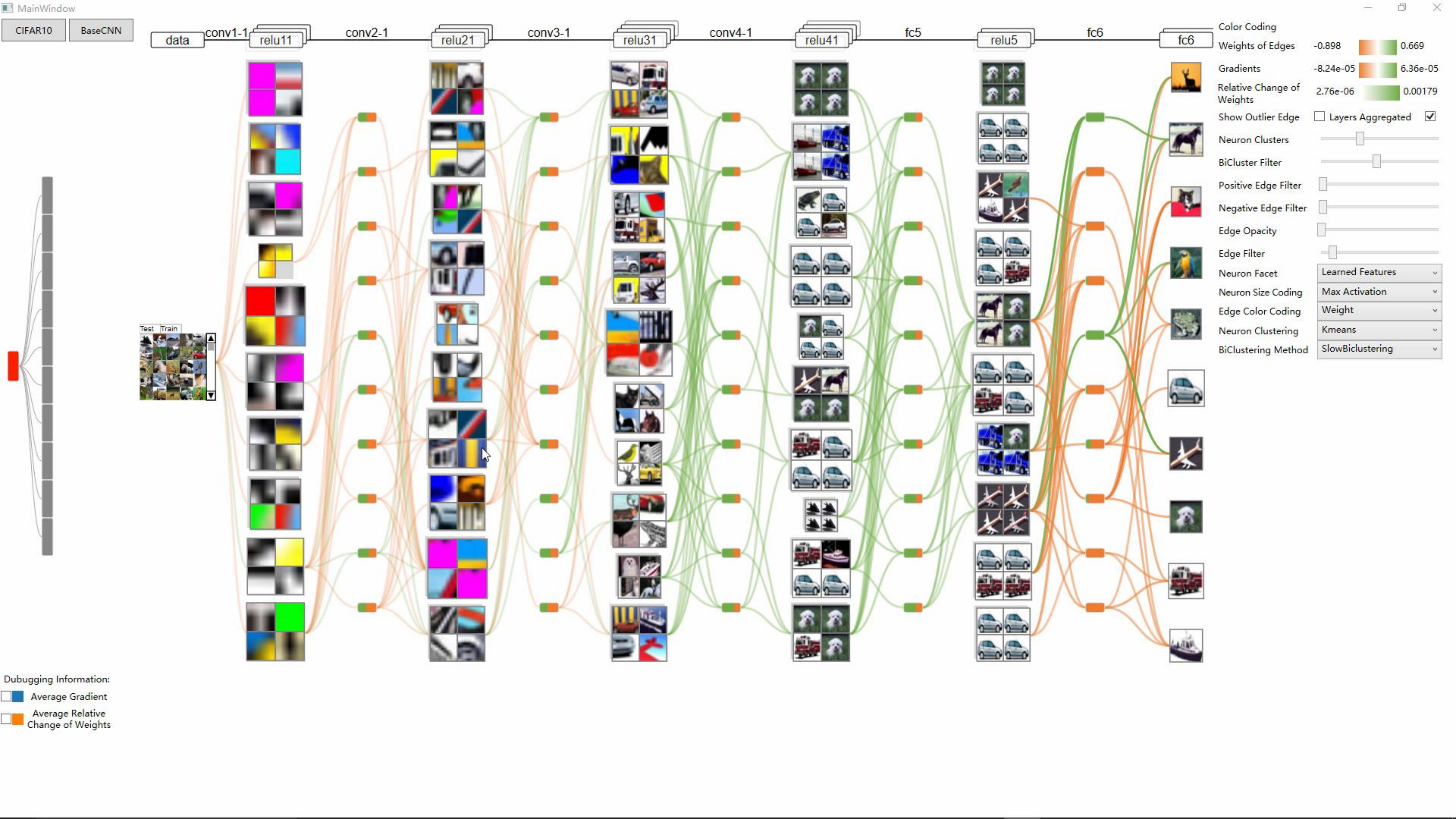

MainWindow
CIFAR10
BaseCNN
data
conv1-1
relu11
conv2-1
relu21
conv3-1
relu31
conv4-1
relu41
fc5
relu5
fc6
fc6
Selection
Test
Train
17/128 Neurons
Learned Features
Color Coding
Weights of Edges
-0.898
0.669
Gradients
-8.24e-05
6.36e-05
Relative Change of Weights
2.76e-06
0.00179
Show Outlier Edge
Layers Aggregated
Neuron Clusters
BiCluster Filter
Positive Edge Filter
Negative Edge Filter
Edge Opacity
Edge Filter
Neuron Facet
Learned Features
Neuron Size Coding
Max Activation
Edge Color Coding
Weight
Neuron Clustering
Kmeans
BiClustering Method
SlowBiclustering
Control panel
Debugging information
Debugging Information:
Average Gradient
Average Relative Change of Weights
0.00113

Case Study: Understanding

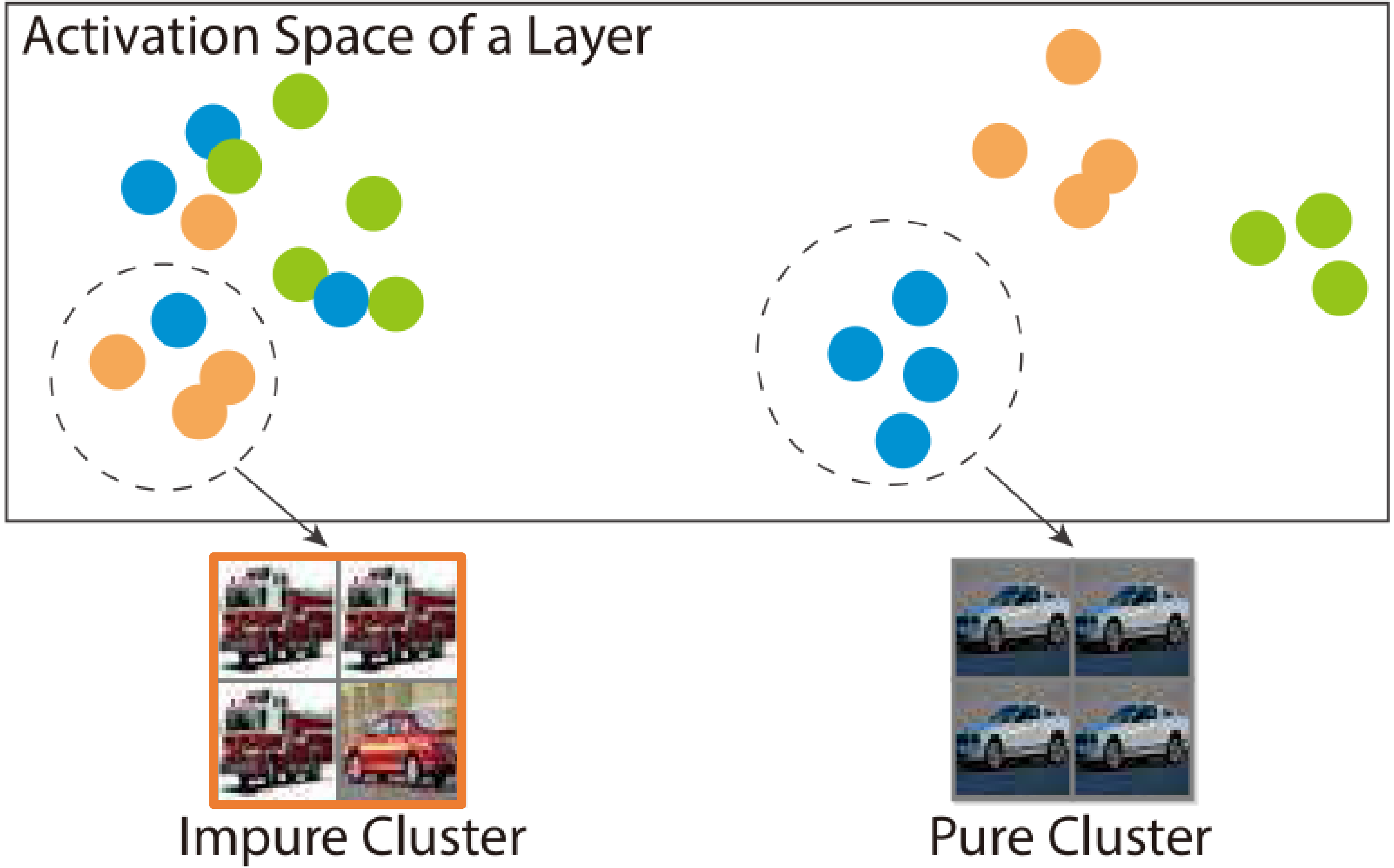
- Network: BaseCNN (Inspired by VGG)

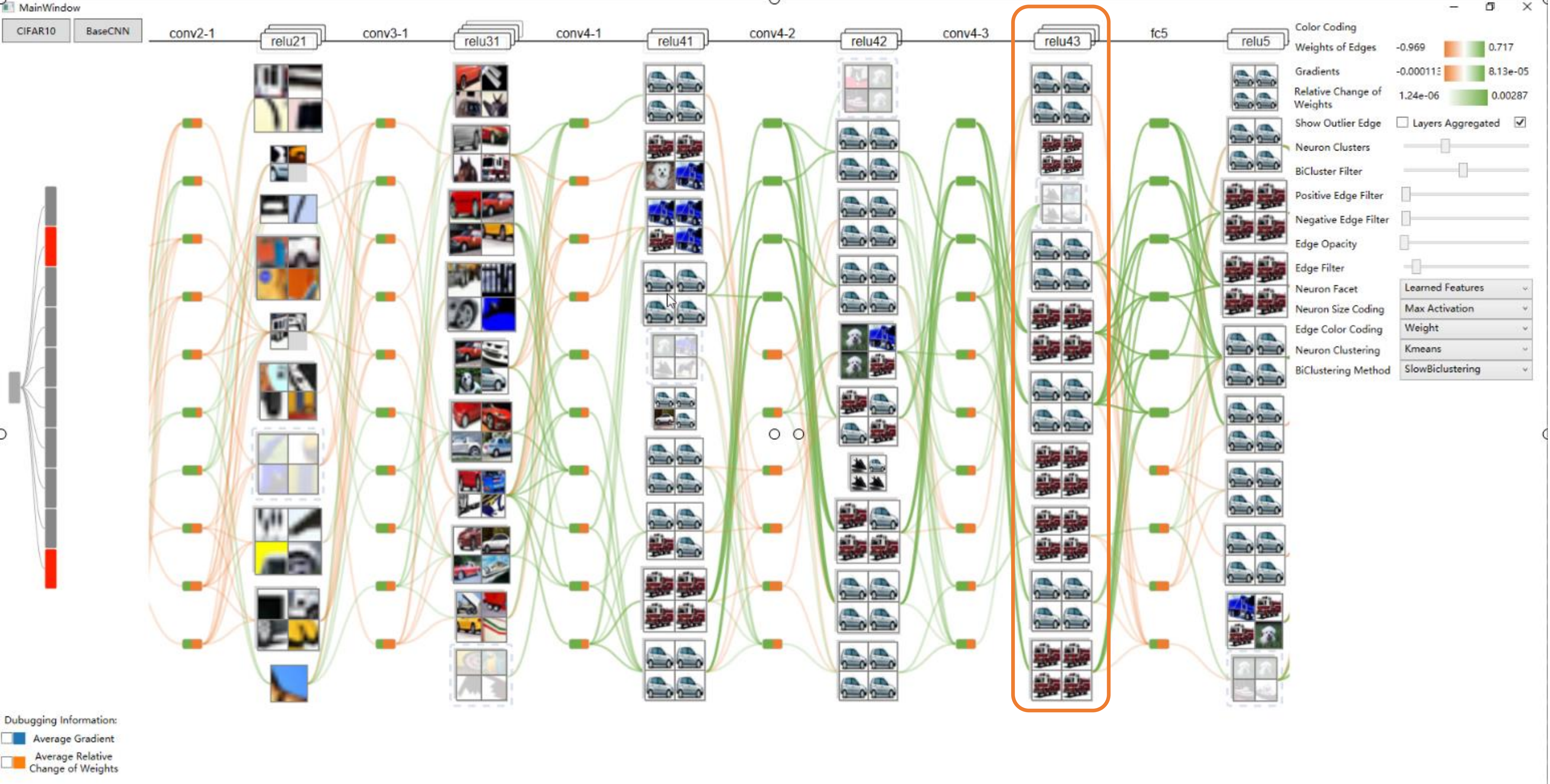


- Dataset: CIFAR10 
- Performance: 11.33% error rate


MainWindow
CIFAR10
BaseCNN
data
conv1-1
relu11
conv2-1
relu21
conv3-1
relu31
conv4-1
relu41
fc5
relu5
fc6
fc6
Color Coding
Weights of Edges -0.898 0.669
Gradients -8.24e-05 6.36e-05
Relative Change of Weights 2.76e-06 0.00179
Show Outlier Edge
Layers Aggregated
Neuron Clusters
BiCluster Filter
Positive Edge Filter
Negative Edge Filter
Edge Opacity
Edge Filter
Neuron Facet
Learned Features
Neuron Size Coding
Max Activation
Edge Color Coding
Weight
Neuron Clustering
Kmeans
BiClustering Method
SlowBiclustering
Test
Train
Dubugging Information:
Average Gradient
Average Relative Change of Weights


MainWindow
CIFAR10
BaseCNN
Color Coding
Weights of Edges
Gradients
Relative Change of Weights
Show Outlier Edge
Layers Aggregated
Neuron Clusters
BiCluster Filter
Positive Edge Filter
Negative Edge Filter
Edge Opacity
Edge Filter
Neuron Facet
Learned Features
Neuron Size Coding
Max Activation
Edge Color Coding
Weight
Neuron Clustering
Kmeans
BiClustering Method
SlowBiclustering
Dubugging Information:
Average Gradient
Average Relative Change of Weights


Activation Space of a Layer
Impure Cluster
Pure Cluster


MainWindow
CIFAR10
BaseCNN
conv2-1
relu21
conv3-1
relu31
conv4-1
relu41
conv4-2
relu42
conv4-3
relu43
fc5
relu5
Color Coding
Weights of Edges -0.969 0.717
Gradients -0.000113 8.13e-05
Relative Change of Weights 1.24e-06 0.00287
Show Outlier Edge
Layers Aggregated
Neuron Clusters
BiCluster Filter
Positive Edge Filter
Negative Edge Filter
Edge Opacity
Edge Filter
Neuron Facet Learned Features
Neuron Size Coding Max Activation
Edge Color Coding Weight
Neuron Clustering Kmeans
BiClustering Method SlowBiclustering
Dubugging Information:
Average Gradient
Average Relative Change of Weights

Network Depth

	Error	#ConvLayers	#Layers
ShallowCNN	11.94%	7	30
BaseCNN	11.33%	10	40
DeepCNN	14.77%	20	70


MainWindow
CIFAR10
ShallowCNN
Color Coding
Weights of Edges
Gradients
Relative Change of Weights
Show Outlier Edge
Layers Aggregated
Neuron Clusters
BiCluster Filter
Positive Edge Filter
Negative Edge Filter
Edge Opacity
Edge Filter
Neuron Facet
Learned Features
Neuron Size Coding
Max Activation
Edge Color Coding
Weight
Neuron Clustering
Kmeans
BiClustering Method
SlowBiclustering
Dubugging Information:
Average Gradient
Average Relative Change of Weights


MainWindow
CIFAR10
DeepCNN
Color Coding
Weights of Edges
Gradients
Relative Change of Weights
Show Outlier Edge
Layers Aggregated
Neuron Clusters
BiCluster Filter
Positive Edge Filter
Negative Edge Filter
Edge Opacity
Edge Filter
Neuron Facet
Learned Features
Neuron Size Coding
Max Activation
Edge Color Coding
Weight
Neuron Clustering
Kmeans
BiClustering Method
SlowBiclustering
Dubugging Information:
Average Gradient
Average Relative Change of Weights

Network Width

	Error	#Params	Training loss	Testing loss
BaseCNN×4	12.33%	4.22M	0.04	0.51
BaseCNN×2	11.47%	2.11M	0.07	0.43
BaseCNN	11.33%	1.05M	0.16	0.40
BaseCNN×0.5	12.61%	0.53M	0.34	0.40
BaseCNN×0.25	17.39%	0.26M	0.65	0.53

Overfitting

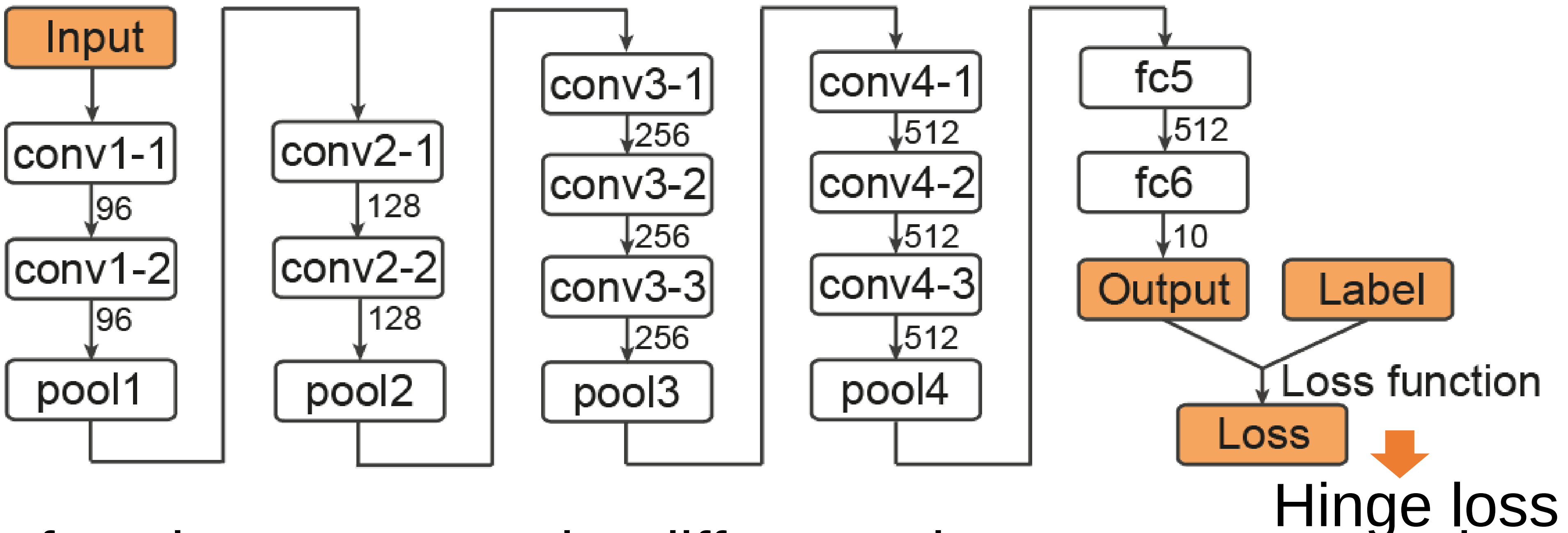
Underfitting


MainWindow
CIFAR10
BaseCNNx0.25
Color Coding
Weights of Edges
Gradients
Relative Change of Weights
Show Outlier Edge
Layers Aggregated
Neuron Clusters
BiCluster Filter
Positive Edge Filter
Negative Edge Filter
Edge Opacity
Edge Filter
Neuron Facet
Learned Features
Neuron Size Coding
Max Activation
Edge Color Coding
Weight
Neuron Clustering
Kmeans
BiClustering Method
SlowBiclustering
Dubugging Information:
Average Gradient
Average Relative Change of Weights

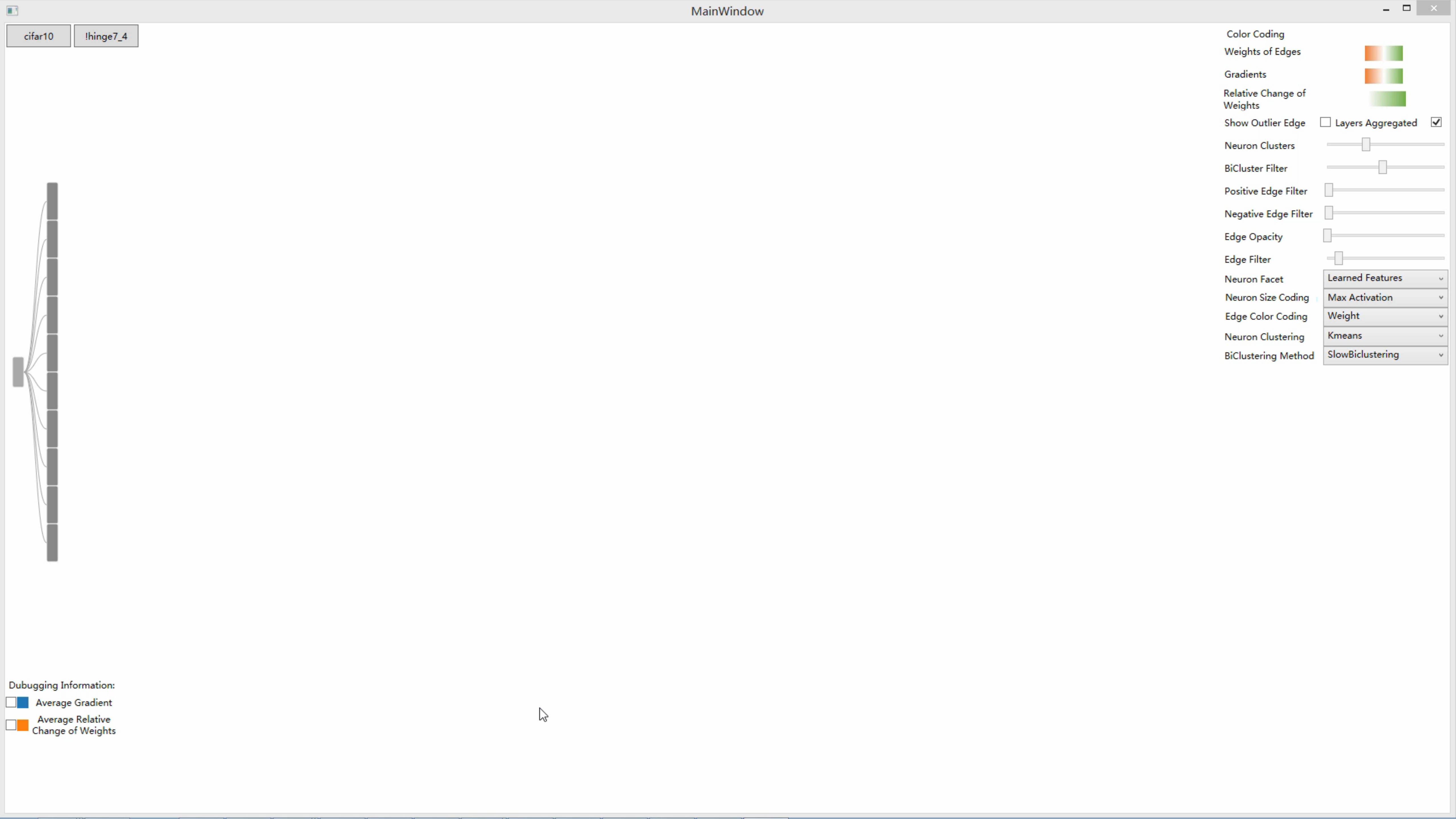

MainWindow
CIFAR10
BaseCNNx4
Color Coding
Weights of Edges
Gradients
Relative Change of Weights
Show Outlier Edge
Layers Aggregated
Neuron Clusters
BiCluster Filter
Positive Edge Filter
Negative Edge Filter
Edge Opacity
Edge Filter
Neuron Facet
Learned Features
Neuron Size Coding
Max Activation
Edge Color Coding
Weight
Neuron Clustering
Kmeans
BiClustering Method
SlowBiclustering
Dubugging Information:
Average Gradient
Average Relative Change of Weights

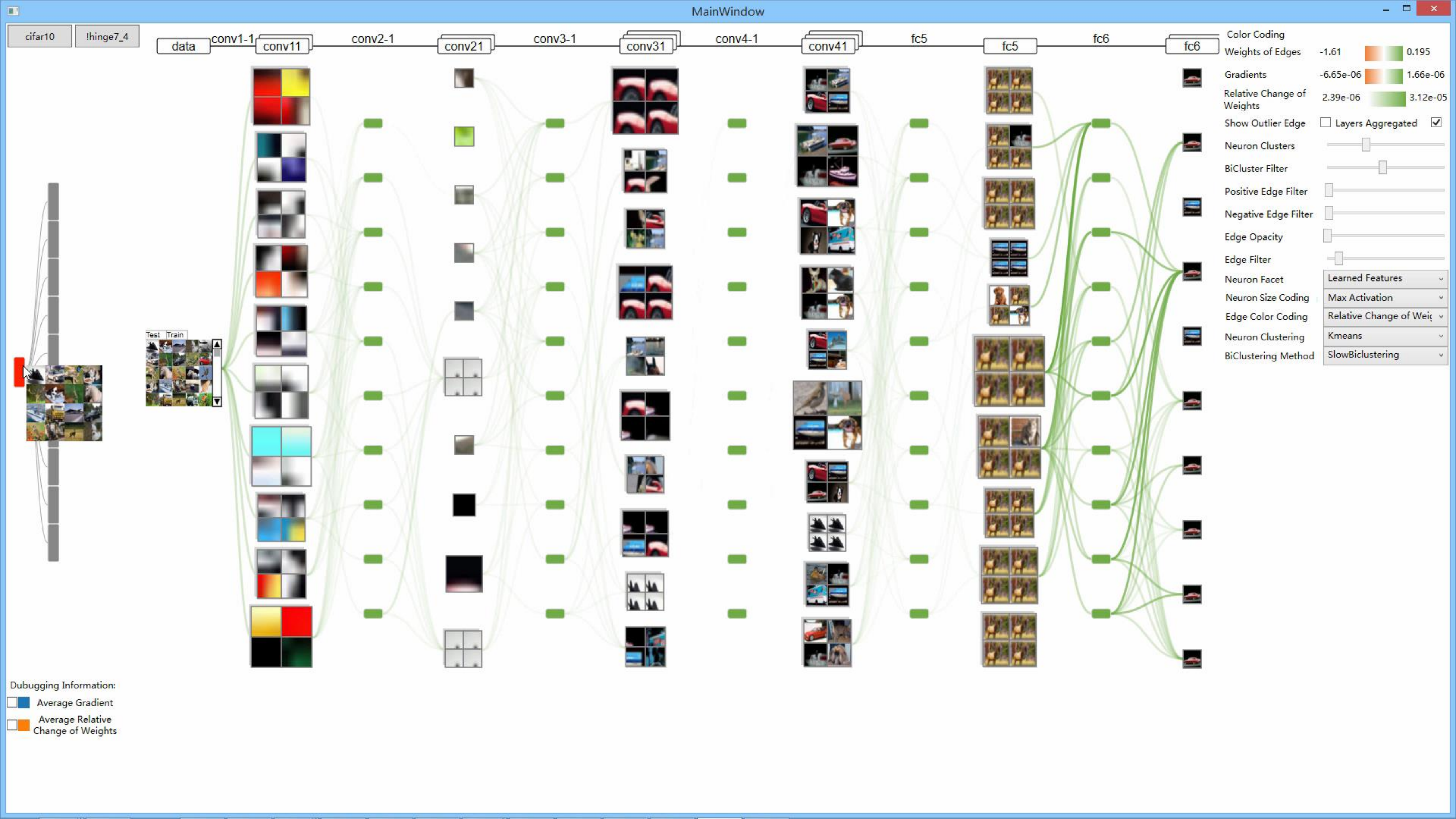
Case Study: Training Diagnosis

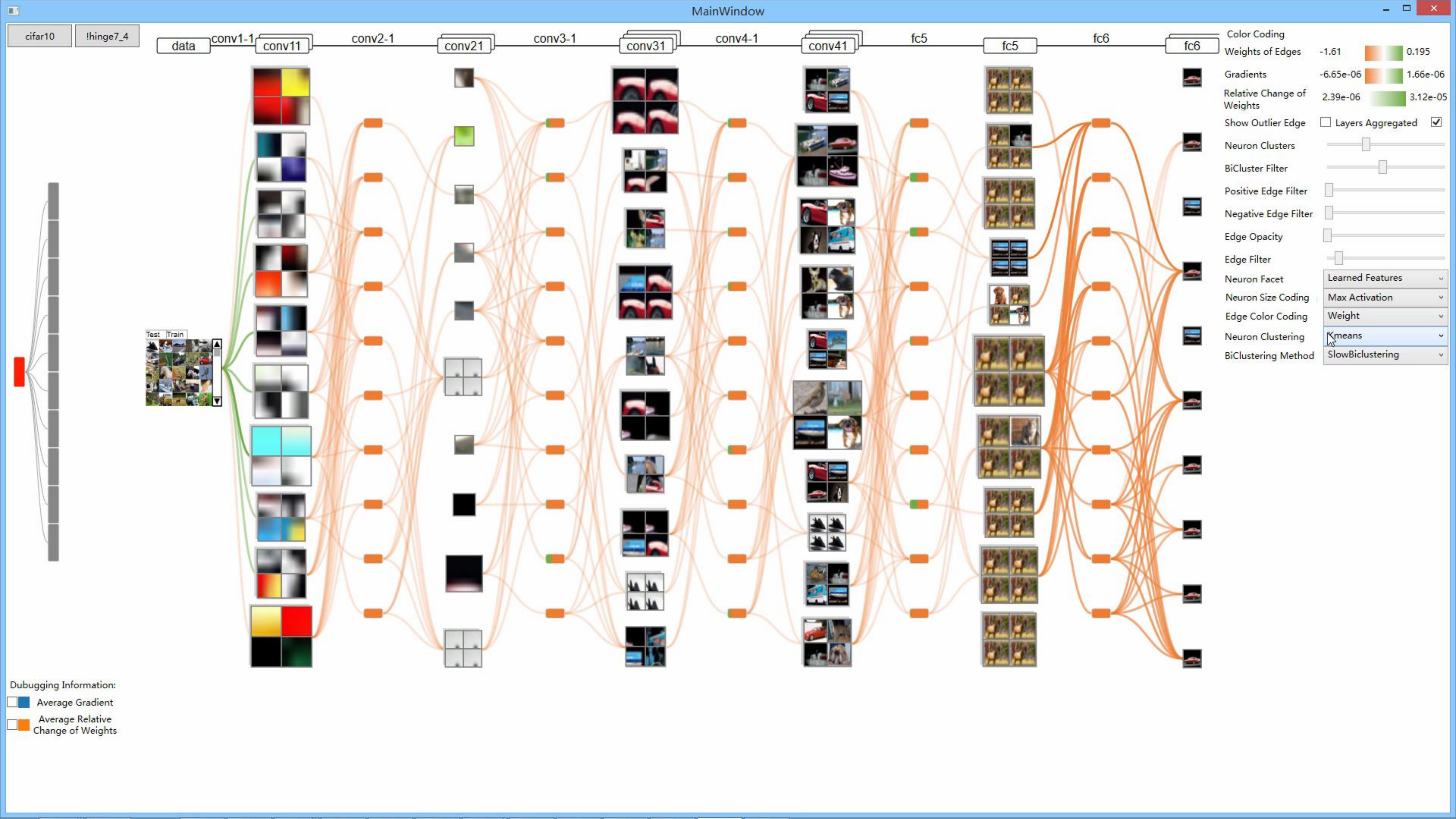
- Network:

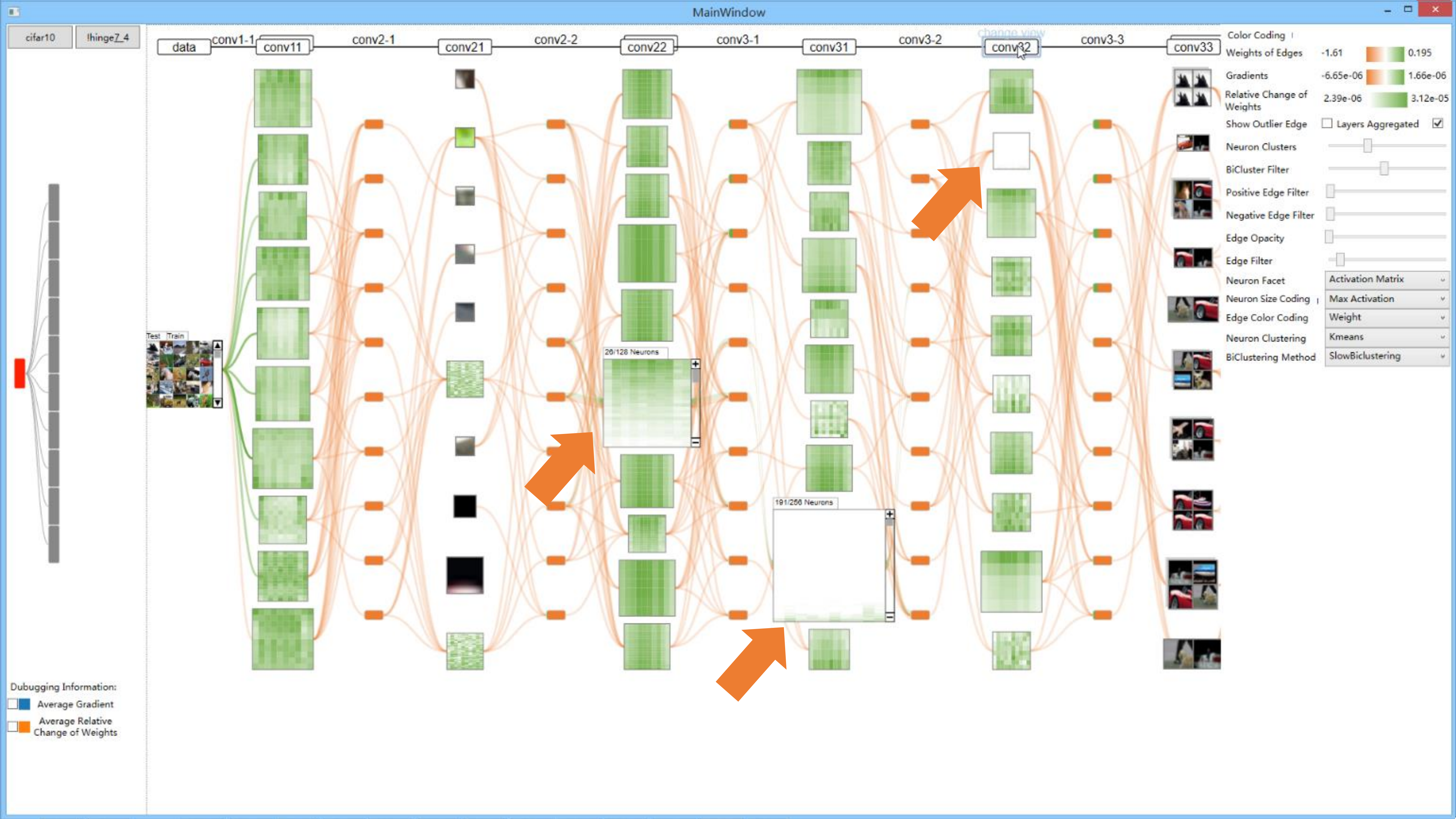


- Loss function: measure the difference between output and true labels
- Training stuck at loss=2.0 (far from achieving good accuracy)

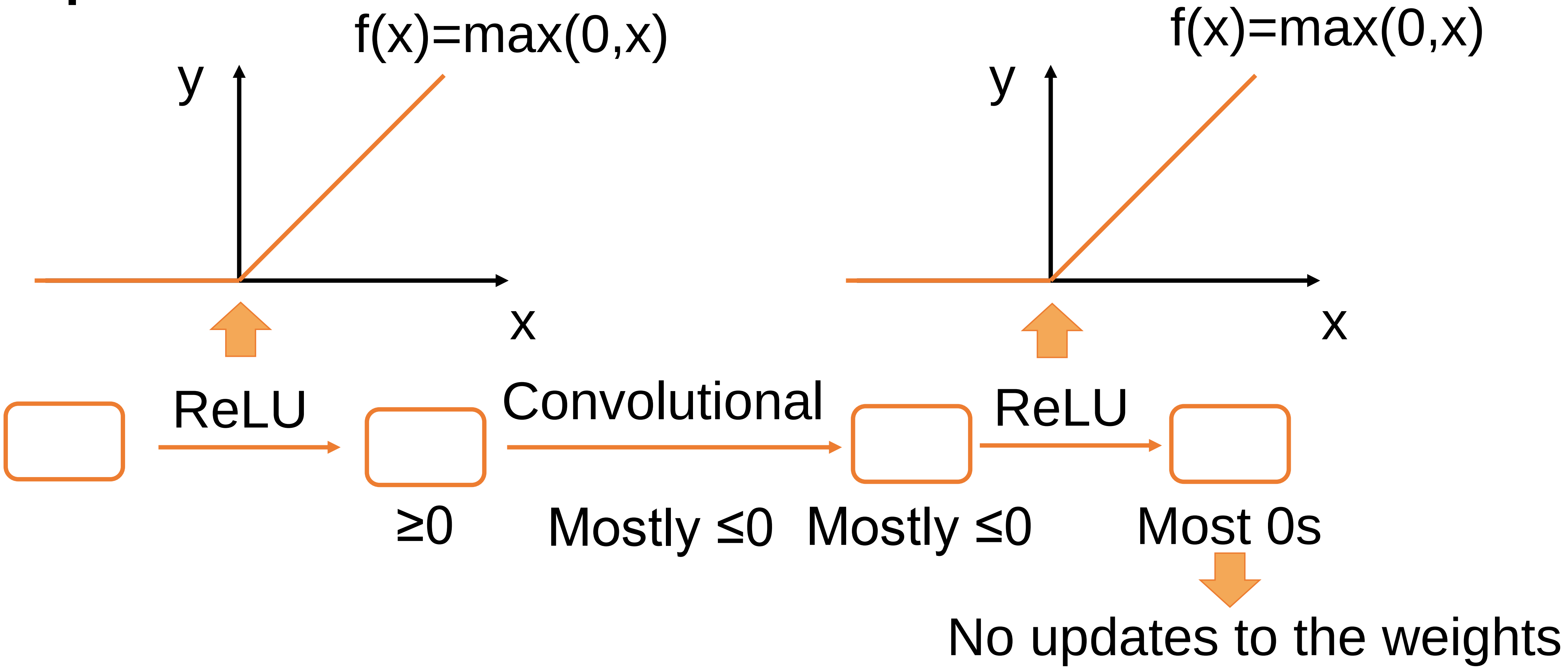

MainWindow
cifar10
!hinge7_4
Color Coding
Weights of Edges
Gradients
Relative Change of Weights
Show Outlier Edge
Layers Aggregated
Neuron Clusters
BiCluster Filter
Positive Edge Filter
Negative Edge Filter
Edge Opacity
Edge Filter
Neuron Facet
Learned Features
Neuron Size Coding
Max Activation
Edge Color Coding
Weight
Neuron Clustering
Kmeans
BiClustering Method
SlowBiclustering
Dubugging Information:
Average Gradient
Average Relative Change of Weights


MainWindow
cifar10
!hinge7_4
data
conv1-1
conv11
conv2-1
conv21
conv3-1
conv31
conv4-1
conv41
fc5
fc5
fc6
fc6
Test
Train
Color Coding
Weights of Edges
-1.61
0.195
Gradients
-6.65e-06
1.66e-06
Relative Change of Weights
2.39e-06
3.12e-05
Show Outlier Edge
Layers Aggregated
Neuron Clusters
BiCluster Filter
Positive Edge Filter
Negative Edge Filter
Edge Opacity
Edge Filter
Neuron Facet
Learned Features
Neuron Size Coding
Max Activation
Edge Color Coding
Relative Change of Weig
Neuron Clustering
Kmeans
BiClustering Method
SlowBiclustering
Dubugging Information:
Average Gradient
Average Relative Change of Weights

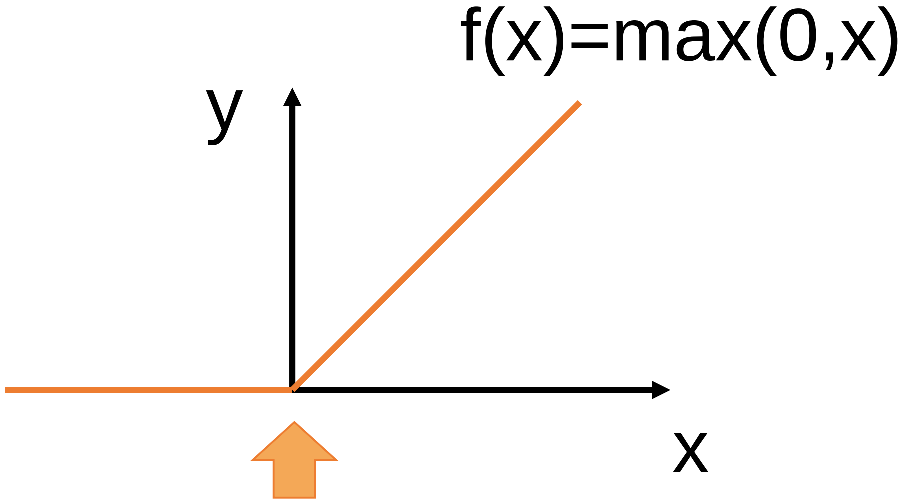

MainWindow
cifar10
!hinge7_4
data
conv1-1
conv11
conv2-1
conv21
conv3-1
conv31
conv4-1
conv41
fc5
fc5
fc6
fc6
Test
Train
Color Coding
Weights of Edges -1.61 0.195
Gradients -6.65e-06 1.66e-06
Relative Change of Weights 2.39e-06 3.12e-05
Show Outlier Edge
Layers Aggregated
Neuron Clusters
BiCluster Filter
Positive Edge Filter
Negative Edge Filter
Edge Opacity
Edge Filter
Neuron Facet
Learned Features
Neuron Size Coding
Max Activation
Edge Color Coding
Weight
Neuron Clustering
Kmeans
BiClustering Method
SlowBiclustering
Dubugging Information:
Average Gradient
Average Relative Change of Weights


MainWindow
cifar10
lhinge7_4
data
conv1-1
conv11
conv2-1
conv21
conv2-2
conv22
conv3-1
conv31
conv3-2
conv32
conv3-3
conv33
change view
Test
Train
26/128 Neurons
191/256 Neurons
Color Coding
Weights of Edges
-1.61
0.195
Gradients
-6.65e-06
1.66e-06
Relative Change of Weights
2.39e-06
3.12e-05
Show Outlier Edge
Layers Aggregated
Neuron Clusters
BiCluster Filter
Positive Edge Filter
Negative Edge Filter
Edge Opacity
Edge Filter
Neuron Facet
Activation Matrix
Neuron Size Coding
Max Activation
Edge Color Coding
Weight
Neuron Clustering
Kmeans
BiClustering Method
SlowBiclustering
Dubugging Information:
Average Gradient
Average Relative Change of Weights

Explanation



Solution



No updates to the weights

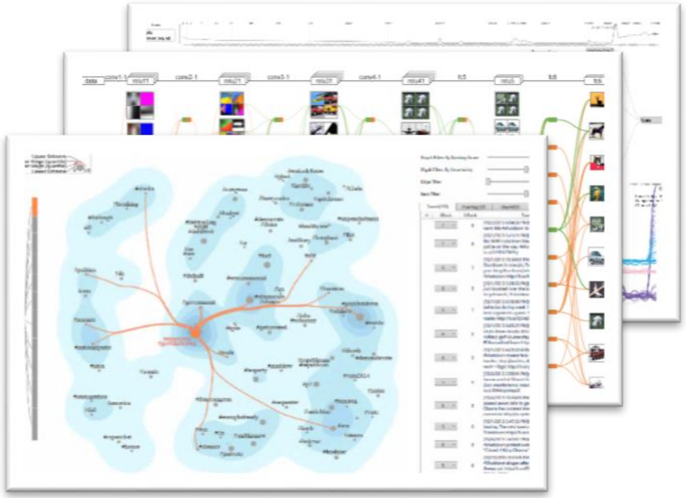
Solution: batch-normalization

Discussion

- CNNVis cannot visualize deep models that cannot be formulated as DAGs
  - RNNs
- The scalability of activation matrix is limited
  - Number of columns = Number of classes
- There is a learning curve associated with the system
  - Neuron cluster and neuron

Future Work

- Provide an integrated system
  - End-to-end development



- Interactive feature selection
  - Manual feature construction / selection is still needed (tabular data)

Name	Thread pitch (mm)	Minor diameter tolerance	Nominal diameter (mm)	Head shape	Price for 50 screws	Available at factory outlet?	Number in stock	Flat or Phillips head?
M4	0.7	4g	4	Pan	$10.08	Yes	276	Flat
M5	0.8	4g	5	Round	$13.89	Yes	183	Both
M6	1	5g	6	Button	$10.42	Yes	1043	Flat
M8	1.25	5g	8	Pan	$11.98	No	298	Phillips
M10	1.5	6g	10	Round	$16.74	Yes	488	Phillips
M12	1.75	7g	12	Pan	$18.26	No	998	Flat
M14	2	7g	14	Round	$21.19	No	235	Phillips
M16	2	8g	16	Button	$23.57	Yes	292	Both
M18	2.1	8g	18	Button	$25.87	No	664	Both
M20	2.4	8g	20	Pan	$29.09	Yes	486	Both
M24	2.55	9g	24	Round	$33.01	Yes	982	Phillips
M28	2.7	10g	28	Button	$35.66	No	1067	Phillips
M36	3.2	12g	36	Pan	$41.32	No	434	Both
M50	4.5	15g	50	Pan	$44.72	No	740	Flat

Thank You!

Q & A