



# Training very deep networks

Srivastava, Rupesh Kumar, Klaus Greff, and Jürgen Schmidhuber.

# Abstract

Theoretical and empirical evidence indicates that the depth of neural networks is crucial for their success.

However, training becomes more difficult as **depth increases**.

**Highway networks** allow unimpeded information flow across many layers on information highways

- use adaptive **gating units** to regulate the information flow.
- Even with hundreds of layers, highway networks can be trained directly through simple gradient descent.

# Introduction

In fact, deep networks can represent certain function classes far more efficiently than shallow ones.

To deal with the difficulties of training deep networks, some researchers have focused on developing better optimizers.

Skip connections between layers or to output layers have long been used in neural networks, more recently with the explicit aim to improve the flow of information.

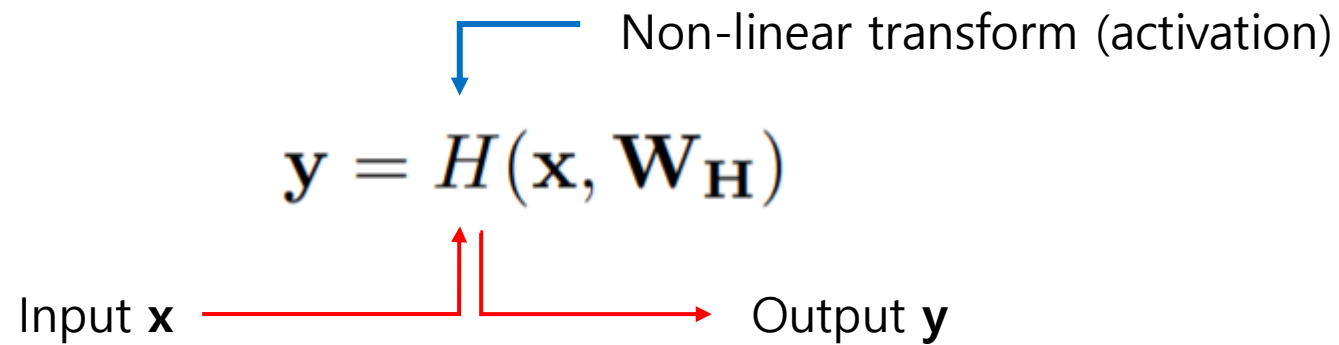
LSTM-inspired adaptive gating mechanism that allows for computation paths along which information can flow across many layers without attenuation.

-> **information highways.**

# Highway Networks

## 1. A plain feedforward neural network

- L layers where the l-th layer ( $l \in \{1, 2, \dots, L\}$ ) applies a non-linear transformation  $\mathbf{H}$  (parameterized by  $\mathbf{W}_{H,l}$ ) on its input  $\mathbf{x}_l$  to produce its output  $\mathbf{y}_l$ .



# Highway Networks

## 2. Highway network

$$y = H(\mathbf{x}, \mathbf{W}_H) \cdot T(\mathbf{x}, \mathbf{W}_T) + \mathbf{x} \cdot C(\mathbf{x}, \mathbf{W}_C).$$

Diagram illustrating the Highway network equation:

- The term  $T(\mathbf{x}, \mathbf{W}_T)$  is labeled "Transform gate" with a blue arrow pointing to it.
- The term  $C(\mathbf{x}, \mathbf{W}_C)$  is labeled "Carry gate" with a blue arrow pointing to it.

We set  $C = 1 - T$ ,

$$y = H(\mathbf{x}, \mathbf{W}_H) \cdot T(\mathbf{x}, \mathbf{W}_T) + \mathbf{x} \cdot (1 - T(\mathbf{x}, \mathbf{W}_T)).$$

# Highway Networks

## 2. Highway network

$$\mathbf{y} = \begin{cases} \mathbf{x}, & \text{if } T(\mathbf{x}, \mathbf{W}_T) = 0, \\ H(\mathbf{x}, \mathbf{W}_H), & \text{if } T(\mathbf{x}, \mathbf{W}_T) = 1. \end{cases}$$

$$\frac{d\mathbf{y}}{d\mathbf{x}} = \begin{cases} \mathbf{I}, & \text{if } T(\mathbf{x}, \mathbf{W}_T) = 0, \\ H'(\mathbf{x}, \mathbf{W}_H), & \text{if } T(\mathbf{x}, \mathbf{W}_T) = 1. \end{cases}$$

- plain layer consists of multiple computing units such that the  $i$ -th unit computes  $y_i = H_i(x)$ ,
- a highway network consists of multiple blocks such that the  $i$ -th block computes a block state  $H_i(x)$  and transform gate output  $T_i(x)$ .
- Finally, it produces the block output  $y_i = H_i(x) * T_i(x) + x_i * (1 - T_i(x))$ , which is connected to the next layer

# Highway Networks

## Constructing Highway Networks

- the dimensionality of  $x$ ,  $y$ ,  $H(x, W_H)$  and  $T(x, W_T)$  be the same.
- To change the size of the intermediate representation,
  1. one can replace  $x$  with  $x'$  obtained by suitably sub-sampling or zero padding  $x$ .
  2. Another alternative is to use a plain layer (without highways) to change dimensionality, which is the strategy we use in this study.

## Convolutional highway layers

We used the same sized receptive fields for both, and zero-padding to ensure that the block state and transform gate feature maps match the input size.

# Highway Networks

## Training Deep Highway Networks


transform gate defined as  $T(x) = \sigma(W_T x + b_T)$ , where  $W_T$  is the weight matrix and  $b_T$  the bias vector for the transform gates

we found that a **negative bias initialization** for the transform gates was sufficient for training to proceed in very deep networks for various zero-mean initial distributions of  $W_H$  and different activation functions used by H



# Experiments

All networks were trained using SGD with momentum. a simpler commonly used strategy was employed where the learning rate starts at a value  $\lambda$  and decays according to a fixed schedule by a factor  $\gamma$ ,  $\lambda$ ,  $\gamma$  and the schedule were selected once based on validation set


$$\alpha = \alpha_0 e^{-kt}$$

All convolutional highway networks utilize the rectified linear activation function (Relu) to compute the block state H.

To provide a better estimate of the variability of classification results due to random initialization, we report our results in the format Best (mean  $\pm$  std) based on 5 runs wherever available.

# Experiments

## Optimization

To support the hypothesis that highway networks do not suffer from increasing depth, we conducted a series of rigorous optimization experiments, comparing them to plain networks with normalized initialization.

All networks are thin:

each layer has 50 blocks for highway networks and 71 units for plain networks

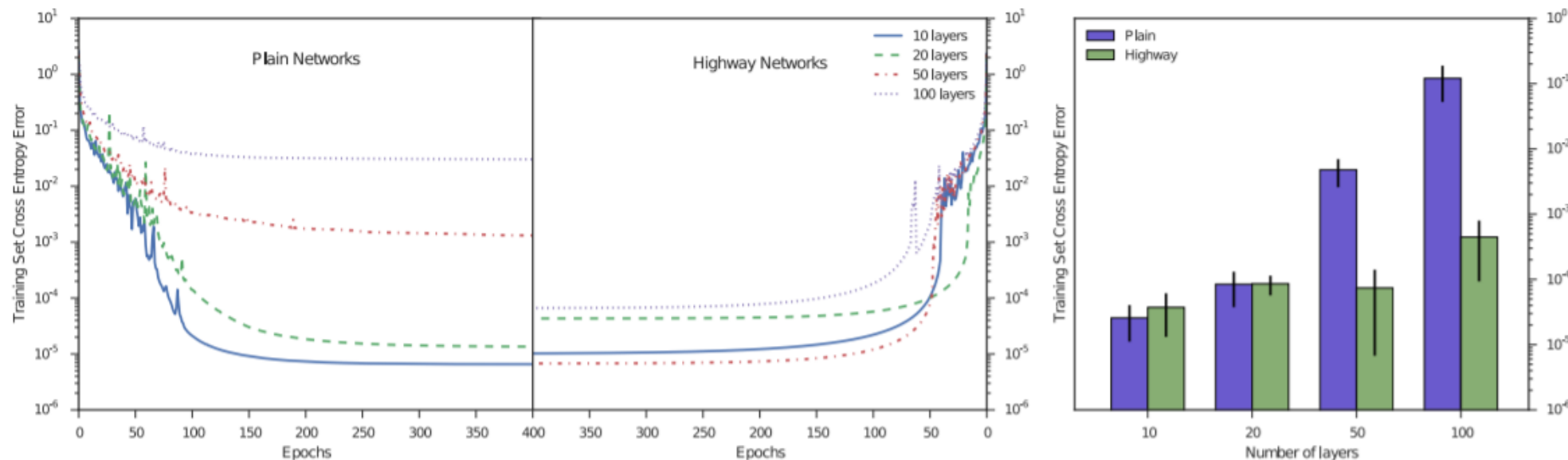
- the first layer is a fully connected plain layer followed by 9, 19, 49, or 99 fully connected plain or highway layers.
- Finally, the network output is produced by a softmax layer

# Experiments

Plain networks become much harder to optimize with increasing depth, while highway networks with up to 100 layers can still be optimized well

The training curves for the best hyperparameter settings obtained for each network depth.

Mean performance of top 10 (out of 100) hyperparameter settings.



Comparison of optimization of plain networks and highway networks of various depths.

# Experiments

## Pilot Experiments on MNIST digit classification

As a sanity check for the generalization capability of highway networks, we trained 10-layer convolutional highway networks on MNIST, using two architectures, each with 9 convolutional layers followed by a soft-max output. The number of filter maps (width) was set to 16 and 32.

Network	Highway Networks		Maxout [20]	DSN [24]
	10-layer (width 16)	10-layer (width 32)		
No. of parameters	39 K	151 K	420 K	350 K
Test Accuracy (in %)	99.43 (99.4 $\pm$ 0.03)	99.55 (99.54 $\pm$ 0.02)	99.55	99.61

Test set classification accuracy for pilot experiments on the MNIST dataset.

# Experiments

## Experiments on CIFAR-10 and CIFAR-100 object recognition

**Fitnet training** : Maxout networks can cope much better with increased depth than those with traditional activation functions.

However, recently reported that training on CIFAR-10 through plain backpropagation was only possible for maxout networks with a depth up to 5 layers when the number of parameters was limited to ~250K and the number of multiplications to ~30M.

Training of deeper networks was only possible through the use of a two-stage training procedure and addition of soft targets produced from a pre-trained shallow teacher network -> **hint-based training**.

# Experiments

## Experiments on CIFAR-10 and CIFAR-100 object recognition

We found that it was easy to train highway networks with numbers of parameters and operations comparable to those of fitnets in a single stage using SGD

Network	No. of Layers	No. of Parameters	Accuracy (in %)
Fitnet Results (reported by Romero et. al.[25])			
Teacher	5	~9M	90.18
Fitnet A	11	~250K	89.01
Fitnet B	19	~2.5M	91.61
Highway networks			
Highway A (Fitnet A)	11	~236K	89.18
Highway B (Fitnet B)	19	~2.3M	<b>92.46 (92.28±0.16)</b>
Highway C	32	~1.25M	91.20

CIFAR-10 test set accuracy of convolutional highway networks

# Experiments

## Comparison to State-of-the-art Methods

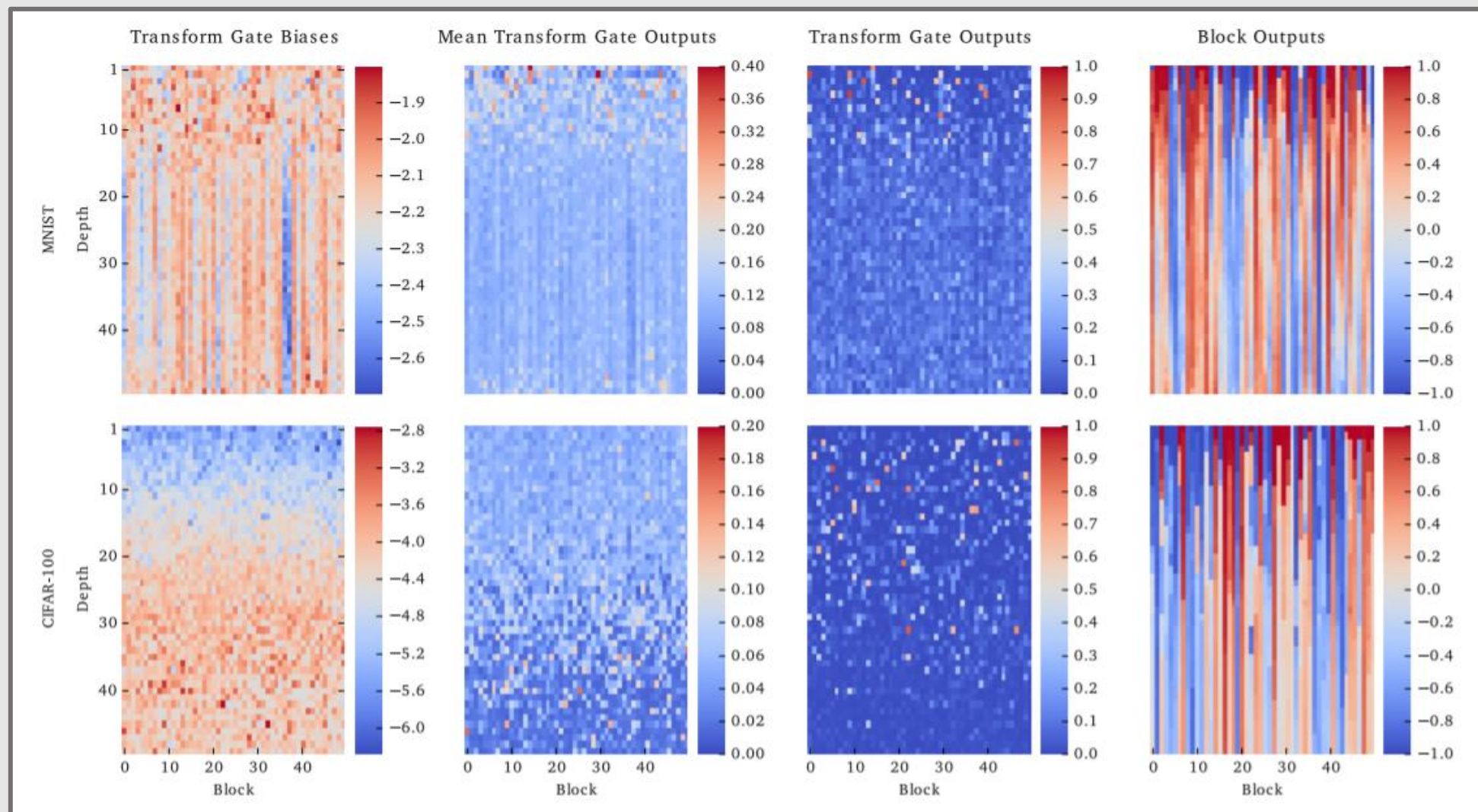
- Since our aim is only to demonstrate that deeper networks can be trained without sacrificing ease of training or generalization ability, we only performed experiments in the more common setting of global contrast normalization, small translations and mirroring of images.
- We replaced the fully connected layer used in the networks in the previous section with a convolutional layer with a receptive field of size one and a global average pooling layer.

Network	CIFAR-10 Accuracy (in %)	CIFAR-100 Accuracy (in %)
Maxout [20]	90.62	61.42
dasNet [36]	90.78	66.22
NiN [35]	91.19	64.32
DSN [24]	92.03	65.43
All-CNN [37]	<b>92.75</b>	66.29
Highway Network	92.40 (92.31 $\pm$ 0.12)	<b>67.76 (67.61<math>\pm</math>0.15)</b>

Test set accuracy of convolutional highway networks



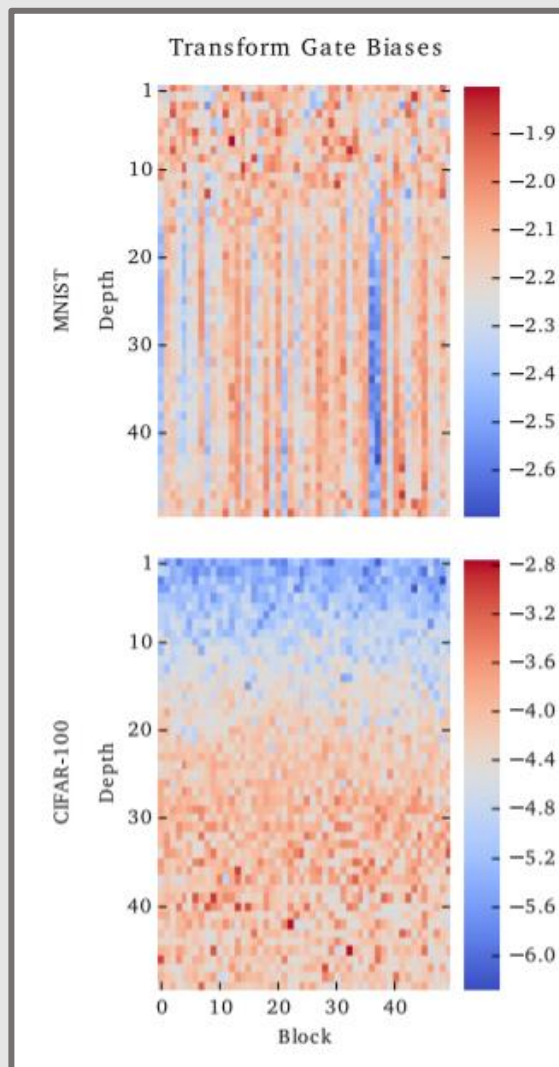
# Analysis



Visualization of best 50 hidden-layer highway networks trained on MNIST and CIFAR-100

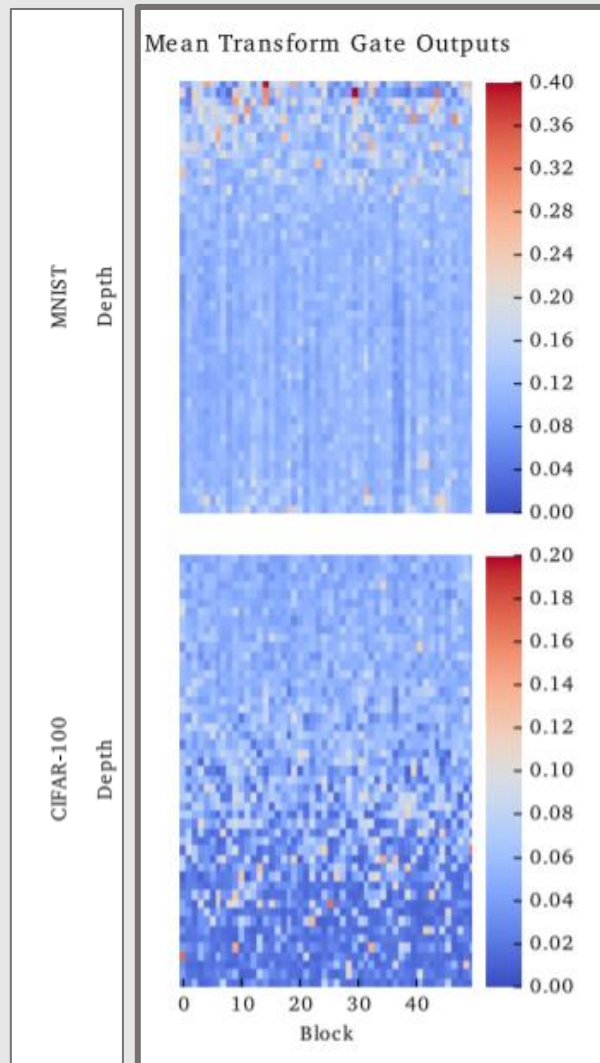


# Analysis



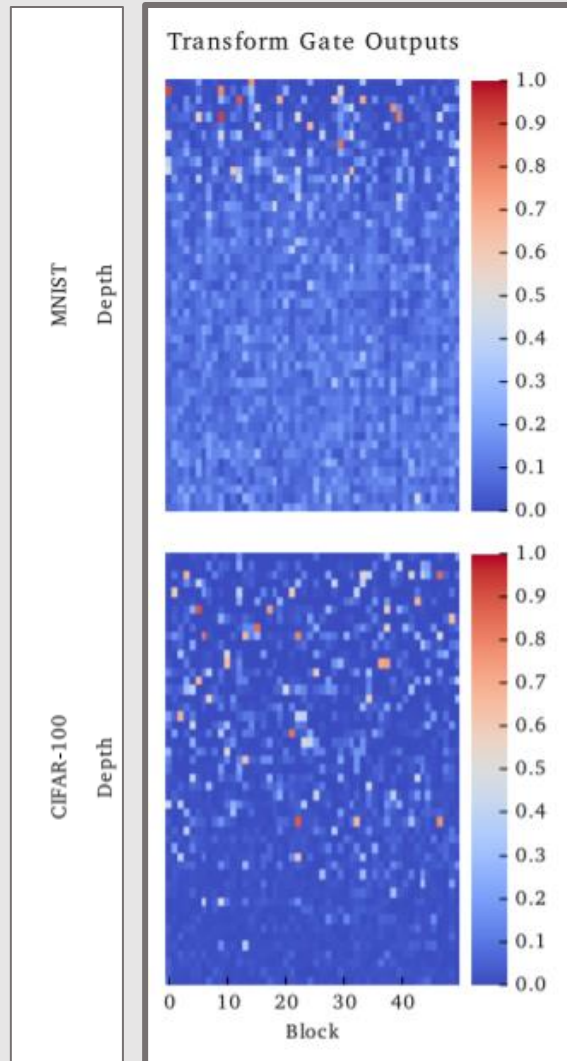
Contrary to our expectations most biases decreased further during training. For the CIFAR-100 network the biases increase with depth forming a gradient.

# Analysis



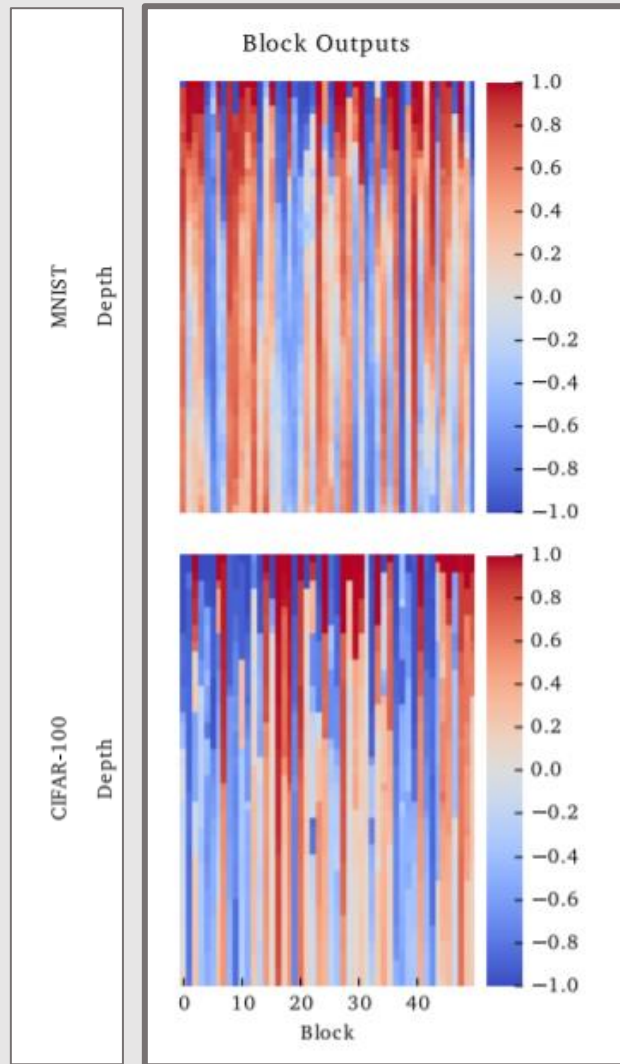
Curiously this gradient is inversely correlated with the average activity of the transform gates. This indicates that the strong negative biases at low depths are not used to shut down the gates, but to make them more **selective**

# Analysis



The transform gate activity for a single example is very **sparse**

# Analysis



The block outputs and visualizes the concept of "**information highways**."

Most of the outputs stay constant over many layers forming a pattern of stripes.

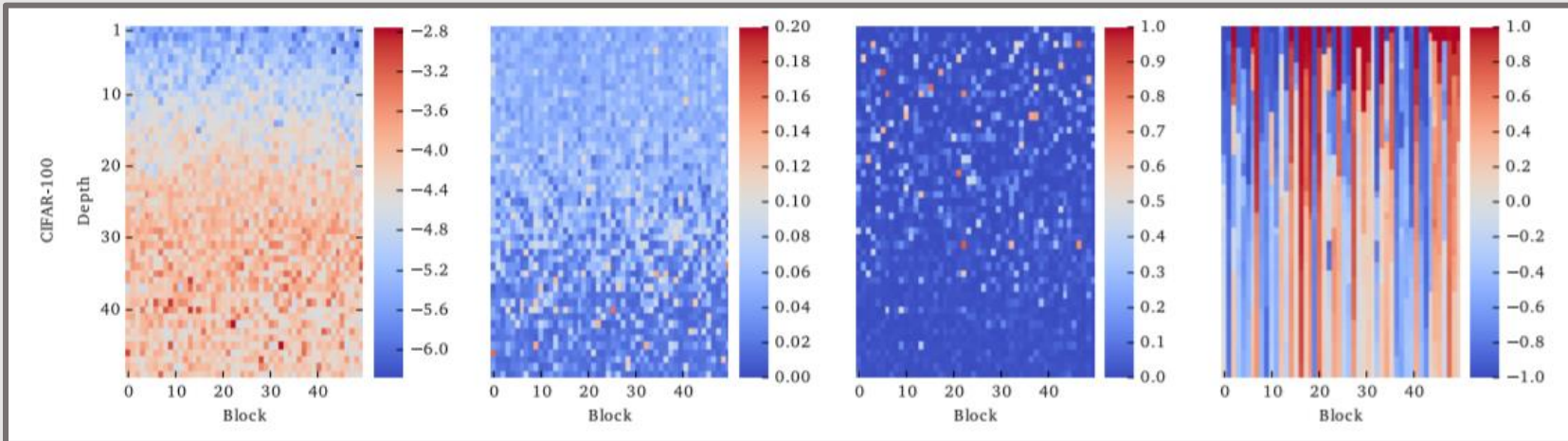
Most of the change in outputs happens in the early layers

# Experiments

## Routing of Information

- One possible advantage of the highway architecture over hard-wired shortcut connections is that the network can learn to dynamically adjust the routing of the information based on the current input. This begs the question:

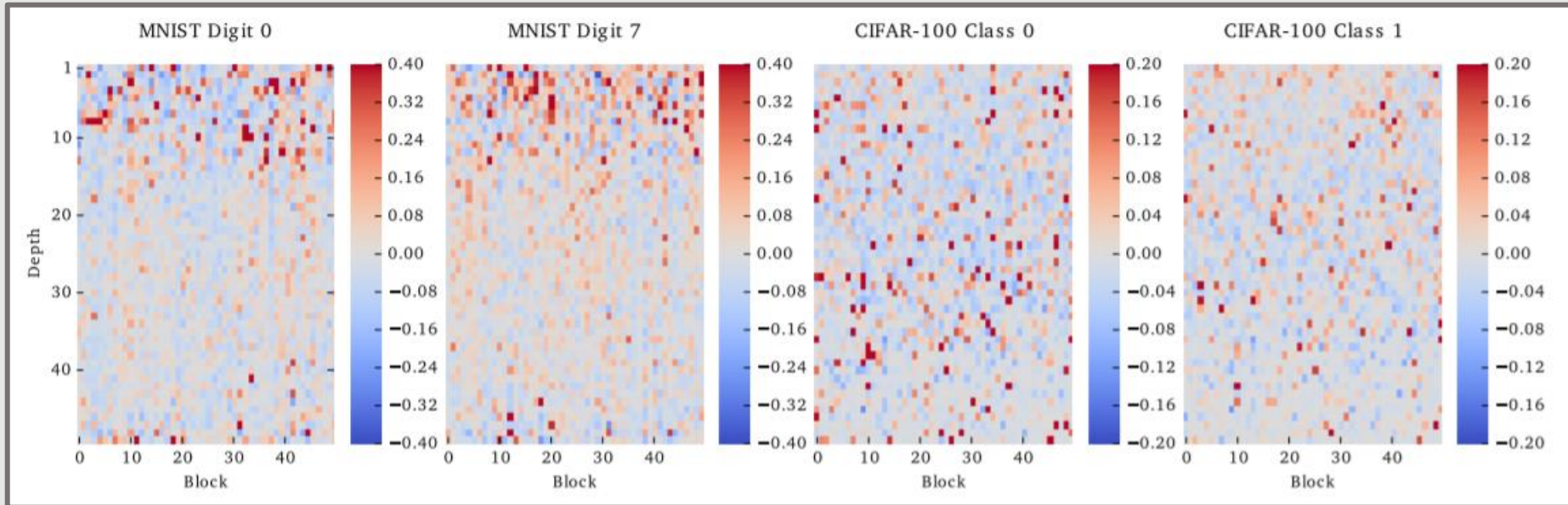
1. does this behaviour manifest itself in trained networks
2. do they just learn a static routing that applies to all inputs similarly



Most transform gates are active on average, while they show very selective activity for the single example. This implies that for each sample only a few blocks perform transformation but different blocks are utilized by different samples.

# Experiments

For MNIST digits 0 and 7 substantial differences can be seen within the first 15 layers, while for CIFAR class numbers 0 and 1 the differences are sparser and spread out over all layers.



**Visualization showing the extent to which the mean transform gate activity for certain classes differs from the mean activity over all training samples**



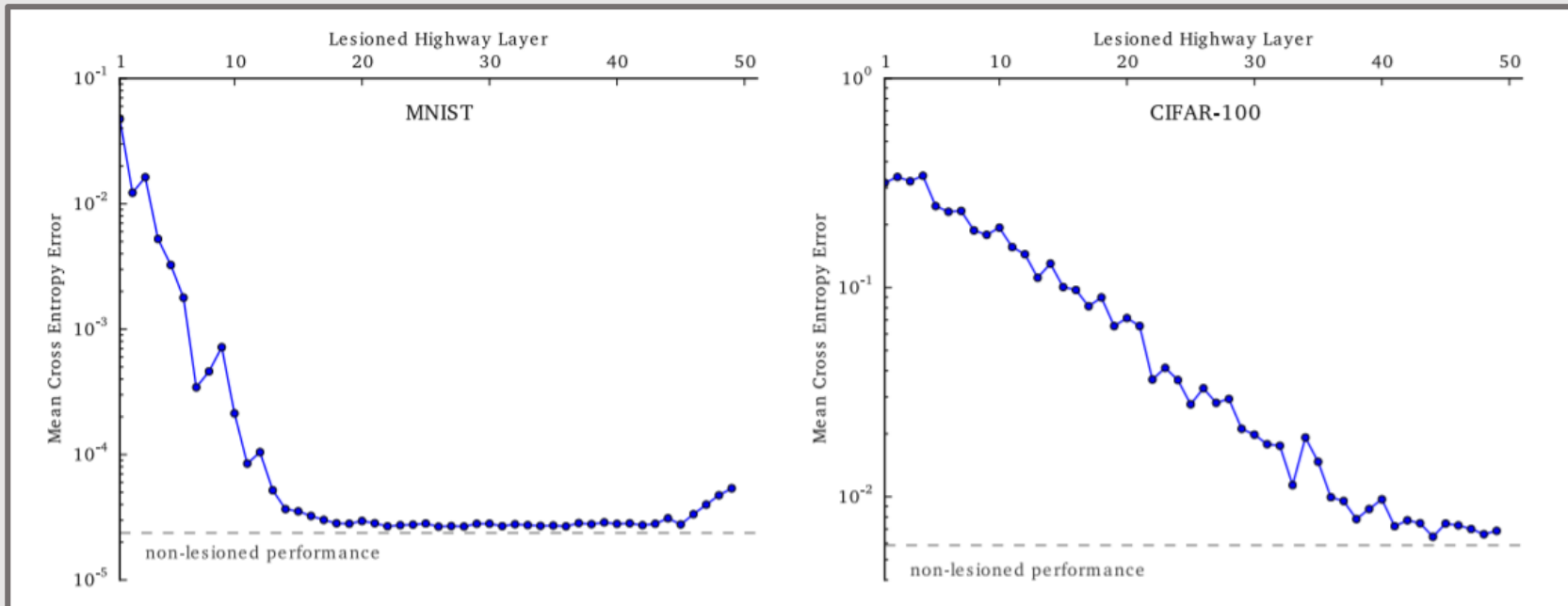
# Experiments

## Layer Importance

- Since we bias all the transform gates towards being closed, in the beginning every layer mostly copies the activations of the previous layer.
- Does training indeed change this behaviour, or is the final network still essentially equivalent to a network with a much fewer layers?
- we investigated the extent to which lesioning a single layer affects the total performance of trained networks .

# Experiments

- it can be seen that the error rises significantly if any one of the early layers is removed, but layers 15 – 45 seem to have close to no effect on the final performance.
- Different picture for the CIFAR-100 dataset with performance degrading noticeably when removing any of the first  $\approx 40$  layers.
- This suggests that for complex problems a highway network can learn to utilize all of its layers, while for simpler problems like MNIST it will keep many of the unneeded layers idle





# Discussion

- Very deep highway networks can directly be trained with simple gradient descent methods due to their specific architecture.
- The additional parameters required by the gating mechanism help in routing information through the use of multiplicative connections, responding differently to different inputs, unlike fixed skip connections