Algebraic Representations for Faster Predictions in Convolutional Neural Networks

Jan Verschelde[†]
joint with Johnny Joyce

University of Illinois at Chicago
Department of Mathematics, Statistics, and Computer Science
http://www.math.uic.edu/~jan
https://github.com/janverschelde
janv@uic.edu

The 26th Workshop on Computer Algebra in Scientific Computing
2-6 September 2024, Rennes, France

[†]Supported by the National Science Foundation, DMS 1854513 and a 2023 Simons Travel Award.

Outline

1. Introduction
   - convolutional neural networks
   - machine learning and algebraic geometry
   - skip connections
2. Making Predictions with CNNs
   - convolutions as transformation matrices
   - removing skip connections with a homotopy
3. Computational Experiments
   - equipment, software, data
   - precomputing CNNs
   - removing skip connections

Algebraic Representations for CNNs

1. Introduction
   - convolutional neural networks
   - machine learning and algebraic geometry
   - skip connections
2. Making Predictions with CNNs
   - convolutions as transformation matrices
   - removing skip connections with a homotopy
3. Computational Experiments
   - equipment, software, data
   - precomputing CNNs
   - removing skip connections

convolutional neural networks

A *neural network* is defined by weights $W^{(k)}$ and biases $B^{(k)}$, for $k = 1, 2, \ldots, L$, in

$$Y = W^{(L)}(\cdots(W^{(2)}(W^{(1)}X + B^{(1)}) + B^{(2)})\cdots) + B^{(L)}$$

in the L layers between the input and output vectors, X and Y.

In a *Convolutional Neural Network* (CNN), convolutions connect the layers, applied to tasks such as image classification.

To *train* a neural network, the weights and biases are computed, given a collection of input and output vectors.

machine learning and algebraic geometry

Almost all learning machines are singular,
is the title of a paper published by S. Watanabe, IEEE, 2007.

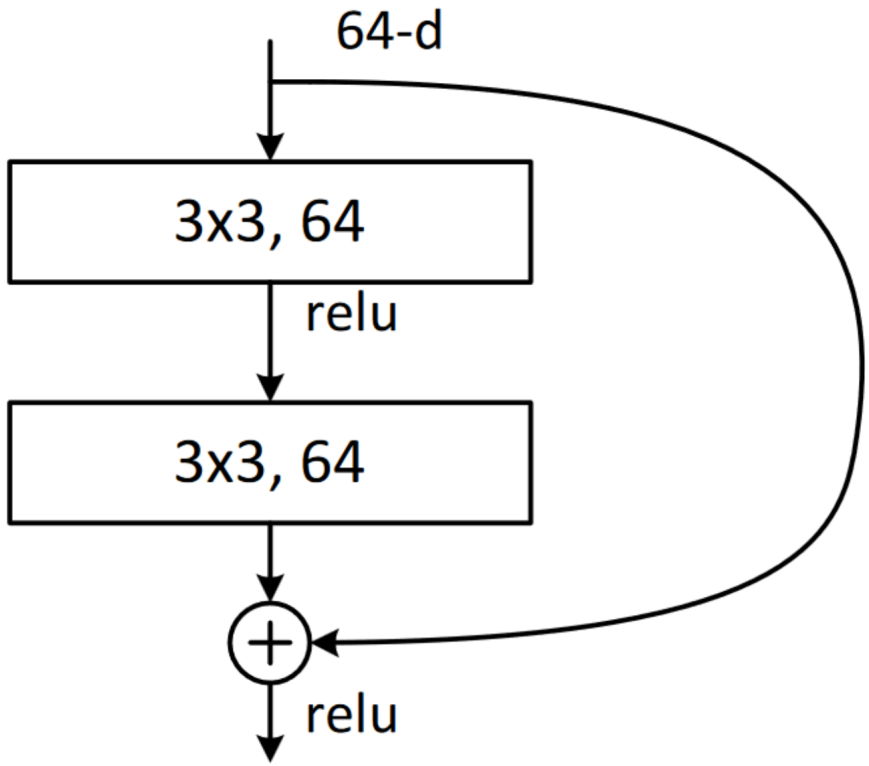
In the book *Algebraic Geometry and Statistical Learning Theory*, Cambridge UP, 2009, S. Watanabe proposes the resolution of singularities to compute the learning coefficient of singular machines.

An incomplete list of related studies:

- Wei, Zhang, Cousseau, Ozeki, Amari, Neural computation, 2008.
- S. Lin, PhD Thesis, Berkeley, 2011.
- Zhang, Naitzat, Lim, PMLR, 2018.
- Mehta, Chen, Tang, Hauenstein, IEEE, 2022.
- Kohn, Merkh, Montúfar, Trager, SIAGA, 2022.

skip connections in ResNet

A building block from `ResNet34`:



Picture from `ResNet` [He, Zhang, Ren, Sun, IEEE, 2016].

`ResNet34` has 34 layers of 3-by-3 convolutions.

skip connections

At the 2018 International Conference on Learning Representations, Orhan and Pitkow presented *skip connections eliminate singularities.*

Skip connections (or shortcut connections, or residual connections) add the output of a particular layer to the input of a later layer.

`ResNet` [He, Zhang, Ren, Sun, IEEE, 2016]
uses deep residual learning for image recognition.

Skip connections allow for the training process to be an easier optimization problem, mitigating the "vanishing gradients" problem.

We propose removing skip connections through a homotopy.

Algebraic Representations for CNNs

1. Introduction
   - convolutional neural networks
   - machine learning and algebraic geometry
   - skip connections

2. Making Predictions with CNNs
   - convolutions as transformation matrices
   - removing skip connections with a homotopy

3. Computational Experiments
   - equipment, software, data
   - precomputing CNNs
   - removing skip connections

resampling images

Our focus is on applying convolutional neural networks to classify images, represented as one dimensional vectors.

Resampling and padding allows for the loss in size when applying convolutions and adding skip connections.

$$B = \begin{bmatrix} a_{1,1} & a_{1,1} & a_{1,2} & a_{1,2} \\ a_{2,1} & a_{2,1} & a_{2,2} & a_{2,2} \\ a_{2,1} & a_{2,1} & a_{2,2} & a_{2,2} \end{bmatrix}$$

is a *resampling* of the vector A:

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{2,1} & a_{2,2} \end{bmatrix}^T$$

The image A is resampled into B by nearest-neighbor interpolation.

padding images

Padding introduces a border around the edges of an image.

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & a_{1,1} & a_{1,2} & 0 \\ 0 & a_{2,1} & a_{2,2} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

is a *padding* of the vector A:

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{2,1} & a_{2,2} \end{bmatrix}^T$$

Any convolution with kernel size [2,2] on C
gives an output of the same size as the original image.

With resampling and padding, convolutional neural networks can be constructed with arbitrarily many skip connections.

skip connections

Starting at the input X: a network is defined recursively:

$$\begin{aligned} f^{(0)}(X) &= X \\ f^{(i)}(X) &= W^{(i)} P^{(i)} \left(\sum_{k=0}^{i-1} t^{(k,i-1)} R^{(k,i-1)} f^{(k)}(X) \right) + B^{(i)} \end{aligned}$$

for all $i \in \{1, \ldots, L\}$, $Y = f^{(L)}(X)$, where

- $t^{(k,i)}$ are the weights of the skip connections, $\sum_{k=0}^{i} t^{(k,i)} = 1$,
  $t^{(k,i)} = 0$ when there is no skip connection;
- $R^{(k,i-1)}$ is a resampling matrix; and
- $P^{(i)}$ is a padding matrix.

making predictions faster

Let $f_W^{(i)}(X) := W^{(i)} P^{(i)} \left(\sum_{k=0}^{i-1} t^{(k,i-1)} R^{(k,i-1)} f^{(k)}(X) \right)$ be a network with skip connections will all bias terms removed.

- Let $\mathbb{1}_{hw}$ be the identity matrix of size $hw \times hw$.
- Let $\mathbf{0}$ to be the zero vector of length hw.

Then the following holds:

$$f(X) = f_W^{(L)}(\mathbb{1}_{hw})X + f^{(L)}(\mathbf{0}),$$

where f is the map given by a CNN with L layers and arbitrarily many skip connections, and where X is an $h \times w$ input matrix that has been reshaped into a vector of length hw.

removing skip connections with a homotopy

Rename $t^{(k,i-1)}$, the value of a skip connection, into t.

We then set the input to the i^{th} layer to:

$$t \cdot x^{(i-1)} + (1-t) \cdot x^{(j)}, \quad t \in [0,1).$$

$t = 0.5$ is the standard value for a network with skip connections.

Then $f(x,t)$ is a homotopy between skip connection models and models without skip connections, at $f(x,0)$.

Training the network with nonzero value for t has the benefit of a more navigable loss landscape in early training rounds.

Algebraic Representations for CNNs

1. Introduction
   - convolutional neural networks
   - machine learning and algebraic geometry
   - skip connections
2. Making Predictions with CNNs
   - convolutions as transformation matrices
   - removing skip connections with a homotopy
3. Computational Experiments
   - equipment, software, data
   - precomputing CNNs
   - removing skip connections

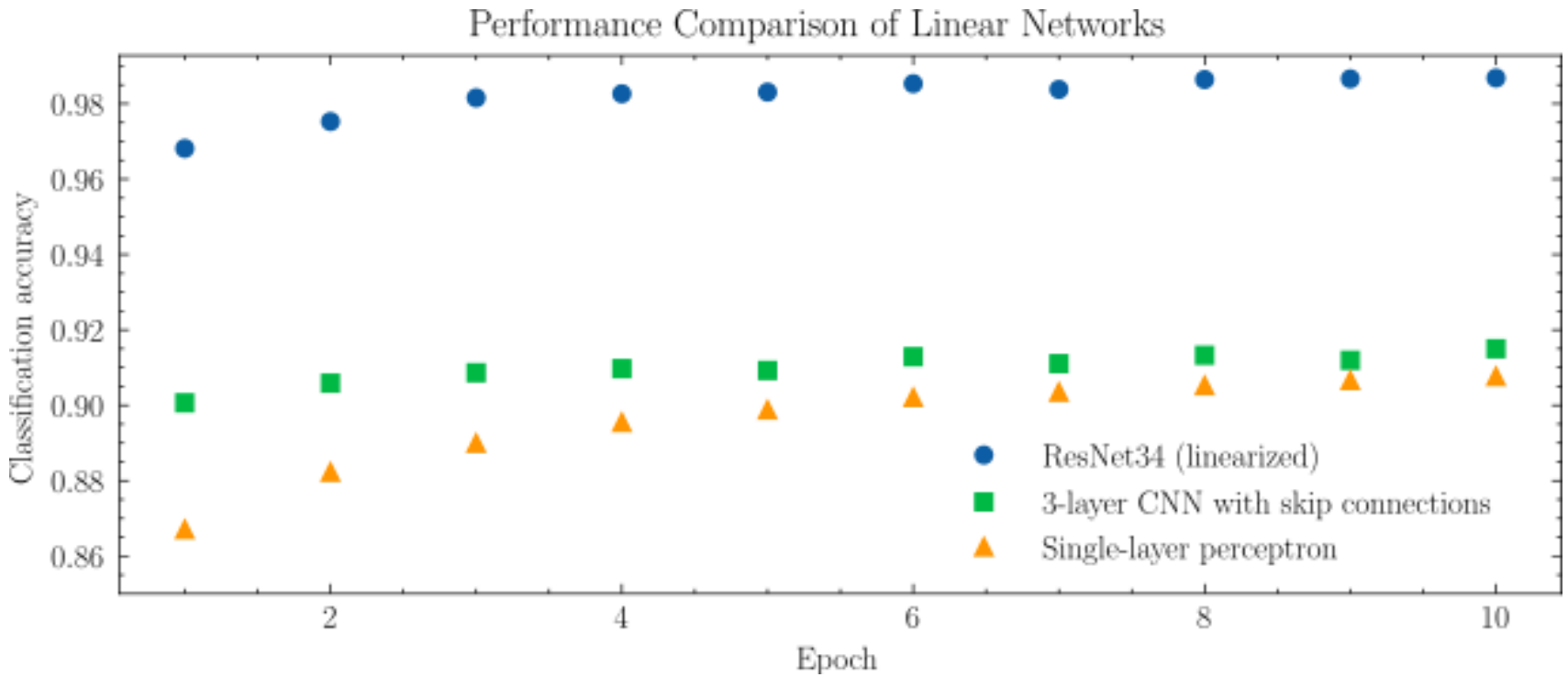
equipment, software, data

- The MNIST database of handwritten digits was used, with
  - 60,000 labeled images for the training set, and
  - 10,000 labeled images for the validation set.
- All models are built in PyTorch, and
  run on an NVIDIA GeForce RTX 2060 Mobile GPU.
- `ResNet34`, with 34 convolutional layers, is used.
- The Jupyter Notebook file used is at
  https://github.com/johnnyvjoyce/simplify-skip-connections

precomputing CNNs

Classification accuracy for varying values of t (skip connection strength) in `ResNet34`:

- 10 epochs were performed over the MNIST dataset in each trial,
- the mean result over 5 trials was taken for each point shown.

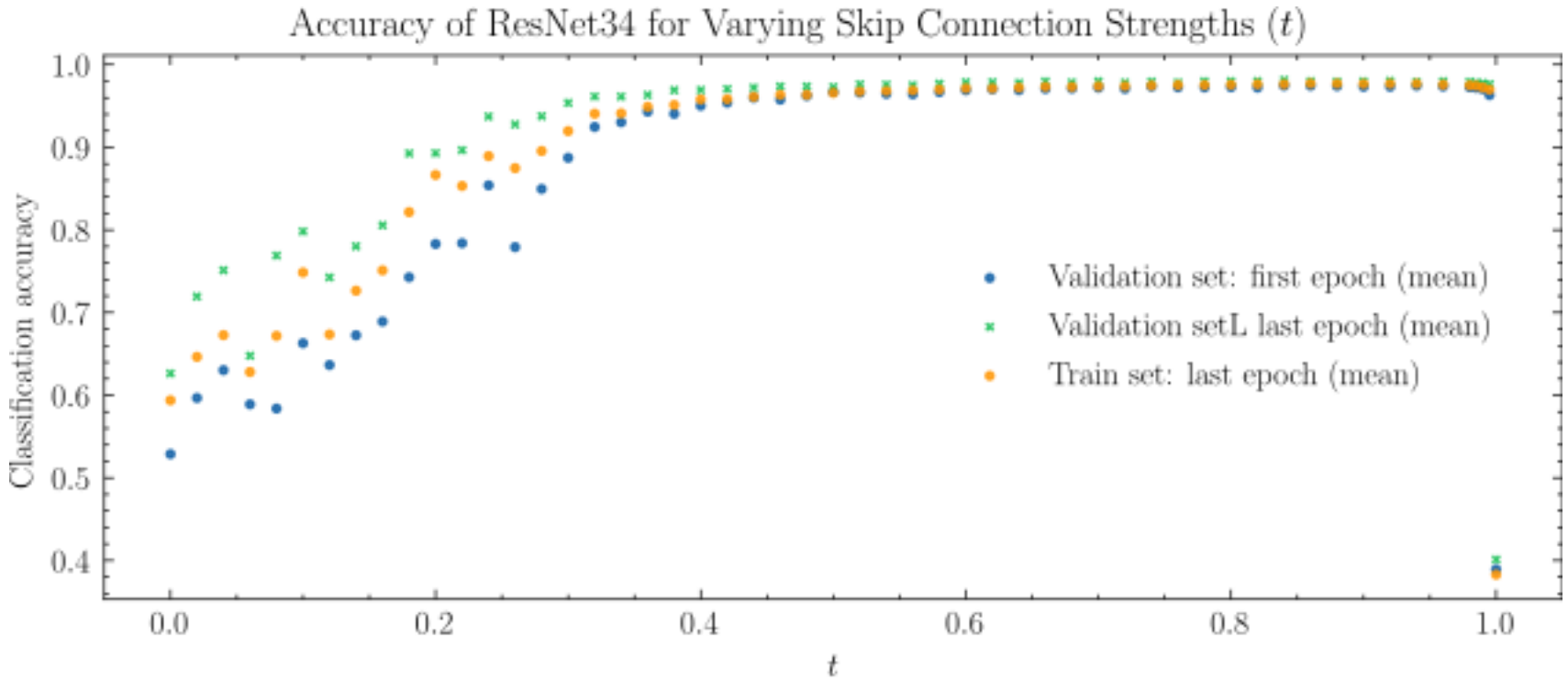


Prediction time is reduced from 104.91 μs to 2.06 μs.

removing skip connections

Classification accuracy for varying values of t (skip connection strength) in `ResNet34`:

- 2 epochs were performed over the MNIST dataset in each trial,
- ithe mean result over 5 trials was taken for each point shown.



static versus varying t values

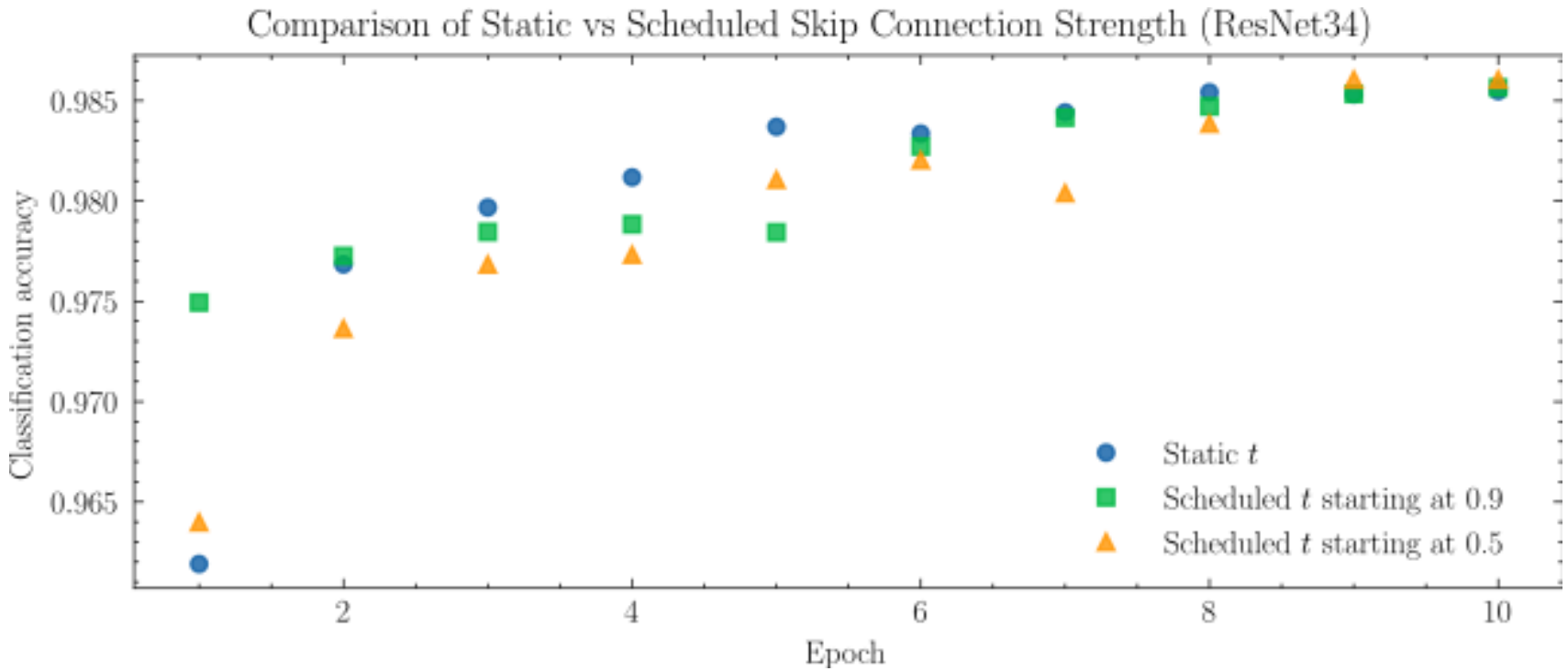
Classification accuracy of `ResNet34` on MNIST validation set with

(a) fixed t (skip connection strength), compared to

(b) scheduled values of t that decrease to 0,

(c) scheduled values starting at 0.5.

Each point shown is the mean over 5 trials.



conclusions

Convolutional neural networks with arbitrarily many skip connections can be represented as maps that can be pre-computed, allowing for prediction times that require resources invariant to the model's depth.

Furthermore,

- In one experiment, a linearized version of `ResNet34`,
  which achieves vastly superior accuracy on unseen data over a single-layer perceptron,
  requires the same resources as a single-layer perceptron when making predictions, a 98% prediction time improvement.
- A method for removing skip connections was introduced.

  Our experiments show that this does not sacrifice any significant amount of accuracy on unseen data, yet provides a 22–46% improvement in prediction time.