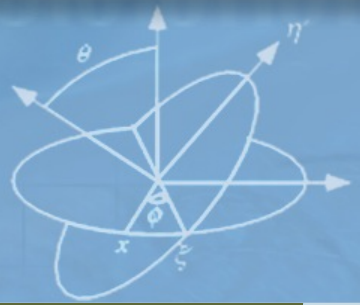
JHU vision lab

Mathematics of Deep Learning

René Vidal
Herschel Seder Professor of Biomedical Engineering
Director of the Mathematical Institute for Data Science
Johns Hopkins University



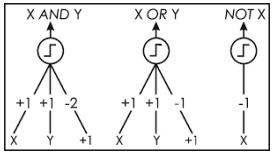
THE DEPARTMENT OF BIOMEDICAL ENGINEERING
The Whitaker Institute at Johns Hopkins

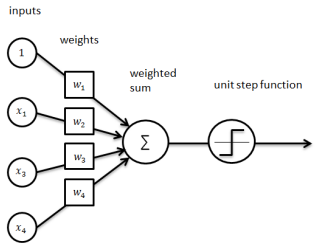
JOHNS HOPKINS
MATHEMATICAL INSTITUTE
for DATA SCIENCE

Brief History of Neural Networks

Beginnings

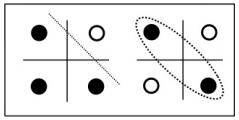
- Thresholded Logic Unit — 1943 — S. McCulloch - W. Pitts
- Perceptron — 1957 — R. Rosenblatt
- Adaline — 1960 — B. Widrow - M. Hoff





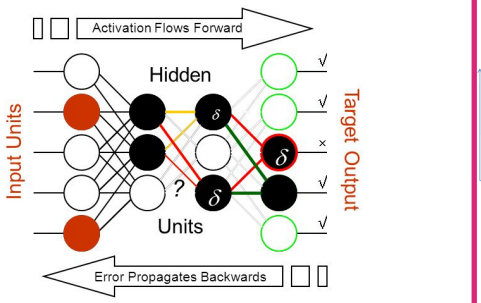
1st Neural Winter

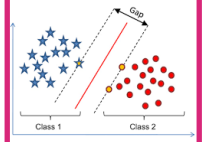
- XOR Problem — 1969 — M. Minsky - S. Papert



2nd Neural Winter

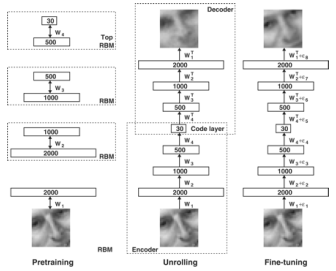
- Multilayer Backprop — 1982 — P. Werbos
- CNNs — 1986 — D. Rumelhart - G. Hinton - R. Williams
- LSTMs — 1997 — Y. Lecun J. Schmidhuber
- SVMs — 1995 — C. Cortes - V. Vapnik





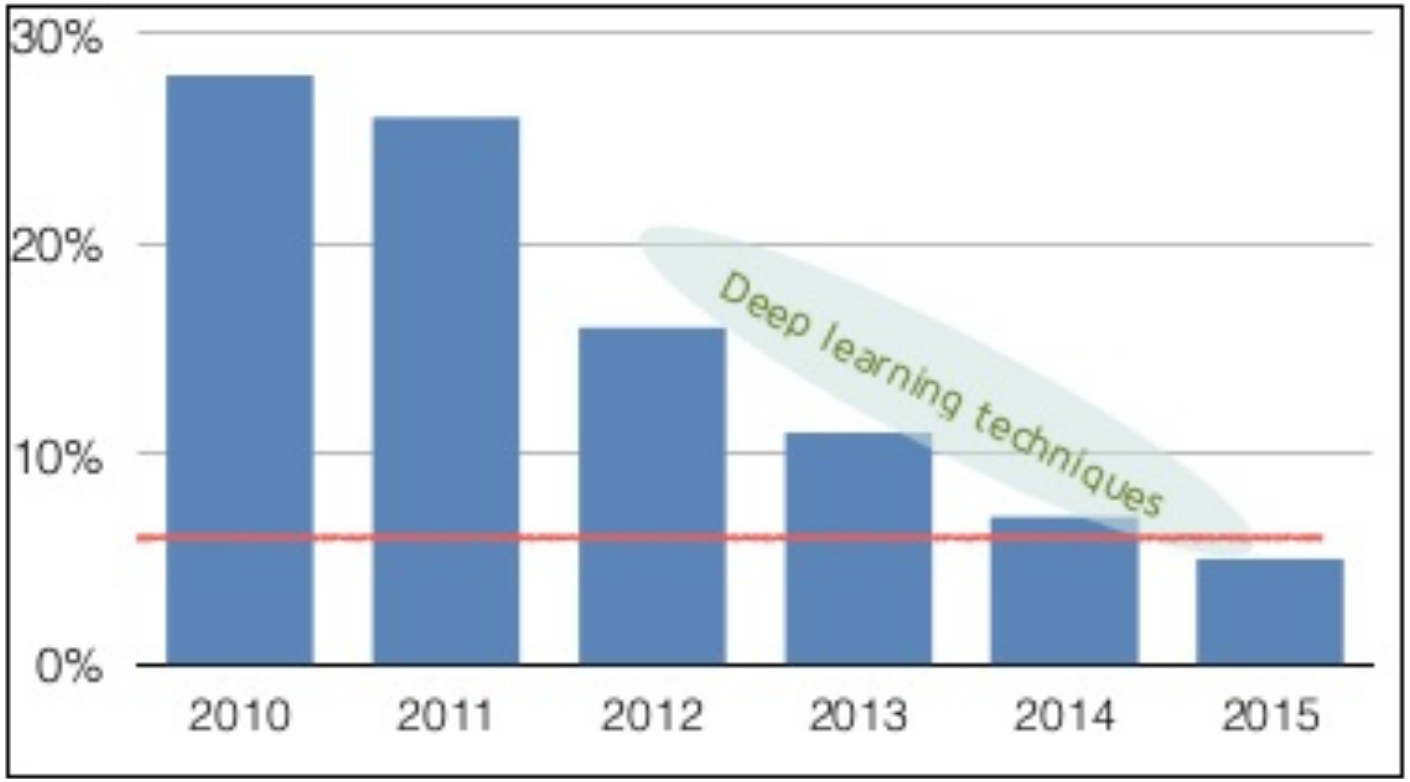
GPU Era

- Deep Nets — 2006 — R. Salakhutdinov - J. Hinton - A. Krizhevsky - I. Sutskever
- Alex Net — 2012



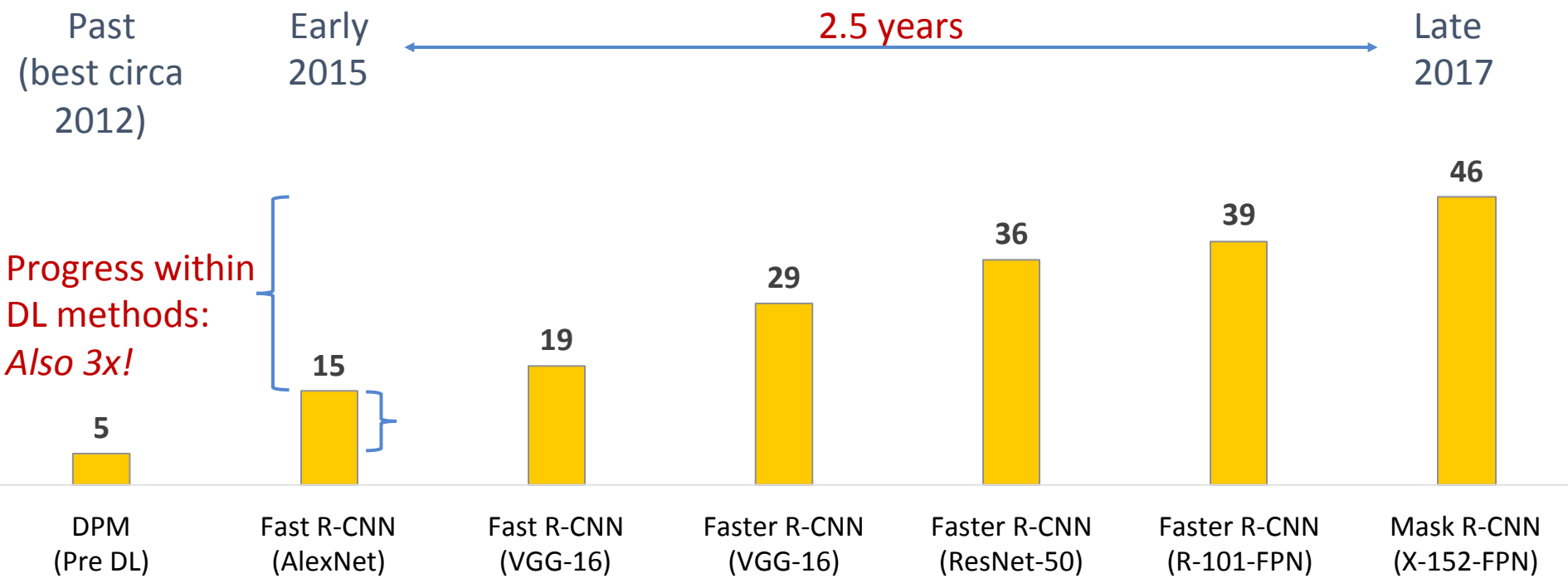
1940 | 1950 | 1960 | 1970 | 1980 | 1990 | 2000 | 2010

Impact of Deep Learning in Computer Vision



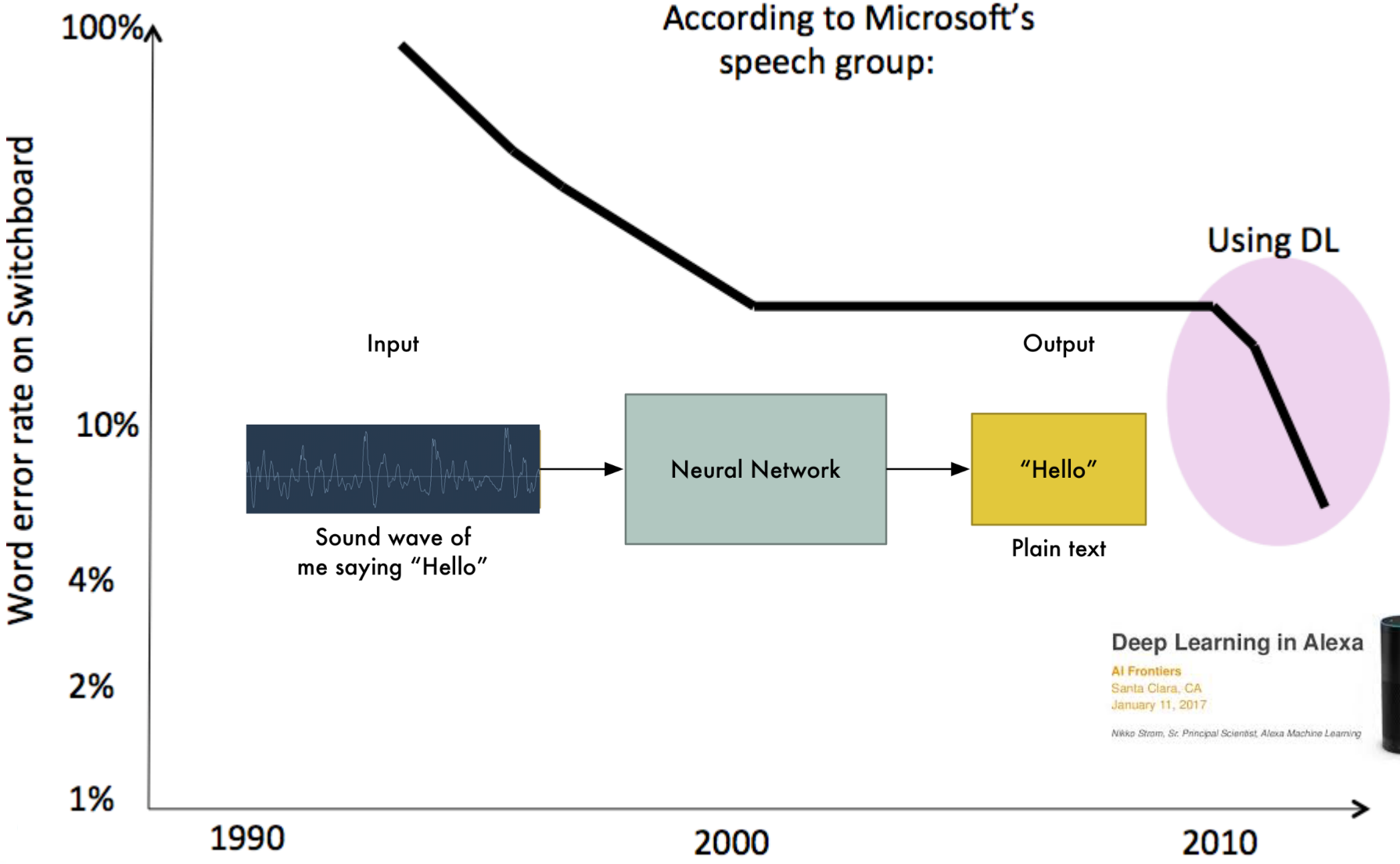
Impact of Deep Learning in Computer Vision

COCO Object Detection Average Precision (%)



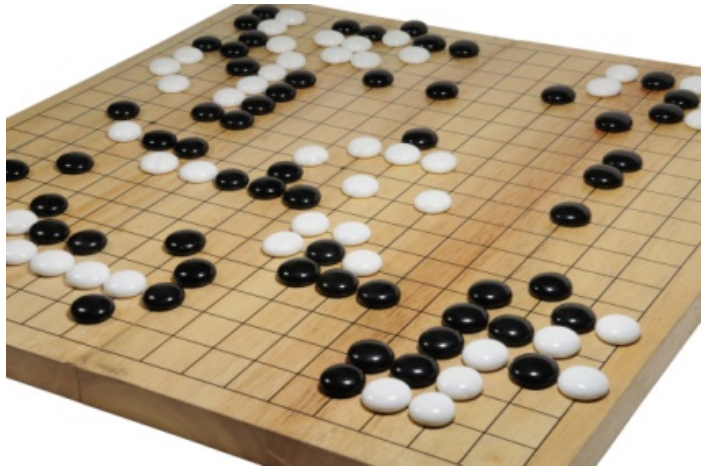
Slide Courtesy of Ross Girshick, ECCV18

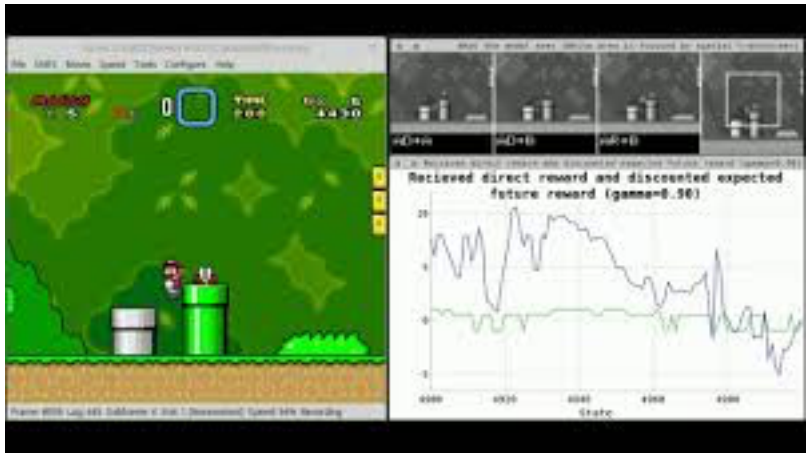
Impact of Deep Learning in Speech Recognition



Impact of Deep Learning in Game Playing

- **AlphaGo**: the first computer program to ever beat a professional player at the game of Go [1]





- Similar deep reinforcement learning strategies developed to play **Atari Breakout**, **Super Mario**

Silver et al. Mastering the game of Go with deep neural networks and tree search, Nature 2016
Artificial intelligence learns Mario level in just 34 attempts, https://www.engadget.com/2015/06/17/super-mario-world-self-learning-ai/, https://github.com/aleju/mario-ai

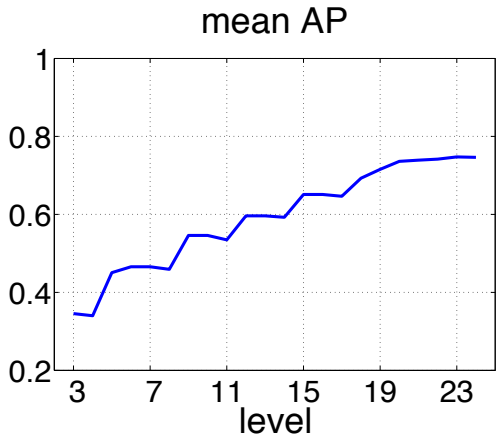
Why These Improvements in Performance?

- Features are learned rather than hand-crafted

[1] Razavian, Azizpour, Sullivan, Carlsson, CNN Features off-the-shelf: an Astounding Baseline for Recognition. CVPRW'14.
[2] Hahnloser, Sarpeshkar, Mahowald, Douglas, Seung. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. Nature, 405(6789):947–951, 2000.

Why These Improvements in Performance?

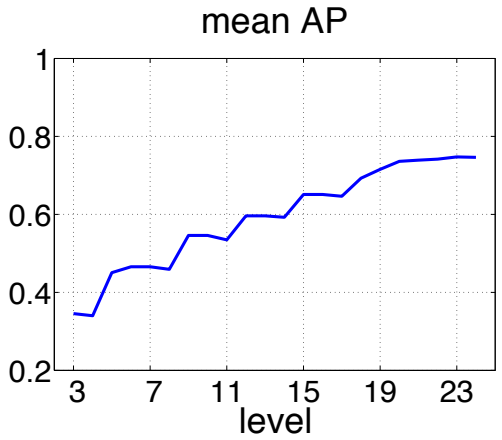
- Features are learned rather than hand-crafted
- More layers capture more invariances [1]



[1] Razavian, Azizpour, Sullivan, Carlsson, CNN Features off-the-shelf: an Astounding Baseline for Recognition. CVPRW'14.
[2] Hahnloser, Sarpeshkar, Mahowald, Douglas, Seung. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. Nature, 405(6789):947–951, 2000.

Why These Improvements in Performance?

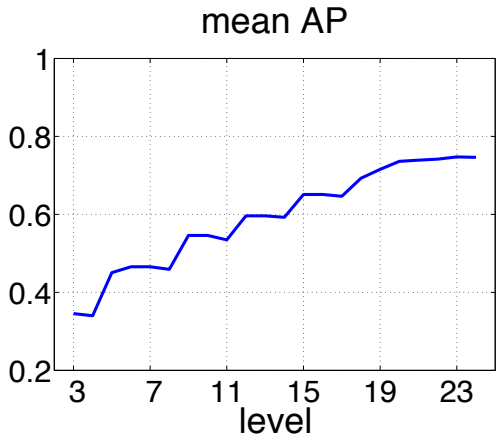
- Features are learned rather than hand-crafted
- More layers capture more invariances [1]
- More data to train deeper networks



[1] Razavian, Azizpour, Sullivan, Carlsson, CNN Features off-the-shelf: an Astounding Baseline for Recognition. CVPRW'14.
[2] Hahnloser, Sarpeshkar, Mahowald, Douglas, Seung. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. Nature, 405(6789):947–951, 2000.

Why These Improvements in Performance?

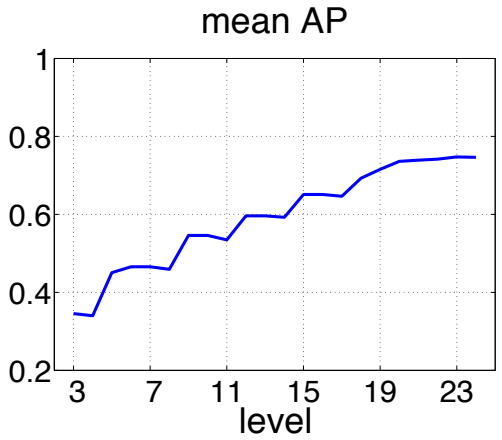
- Features are learned rather than hand-crafted
- More layers capture more invariances [1]
- More data to train deeper networks
- More computing (GPUs)



[1] Razavian, Azizpour, Sullivan, Carlsson, CNN Features off-the-shelf: an Astounding Baseline for Recognition. CVPRW'14.
[2] Hahnloser, Sarpeshkar, Mahowald, Douglas, Seung. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. Nature, 405(6789):947–951, 2000.

Why These Improvements in Performance?

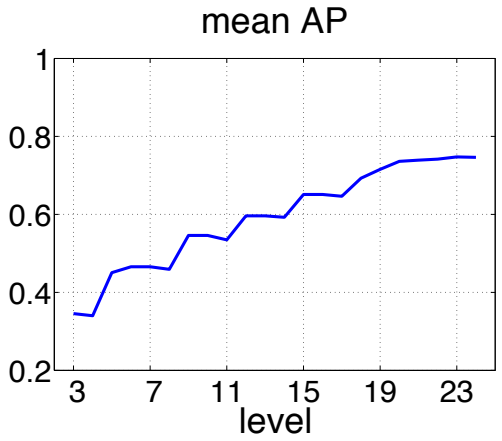
- Features are learned rather than hand-crafted
- More layers capture more invariances [1]
- More data to train deeper networks
- More computing (GPUs)
- Better regularization: Dropout

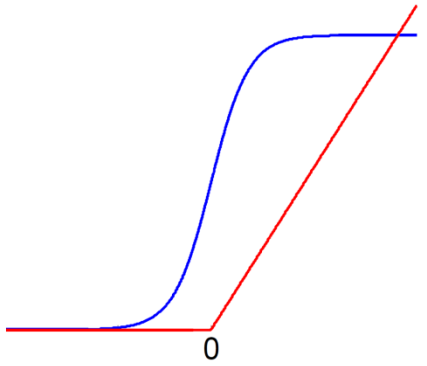


[1] Razavian, Azizpour, Sullivan, Carlsson, CNN Features off-the-shelf: an Astounding Baseline for Recognition. CVPRW'14.
[2] Hahnloser, Sarpeshkar, Mahowald, Douglas, Seung. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. Nature, 405(6789):947–951, 2000.

Why These Improvements in Performance?

- Features are learned rather than hand-crafted
- More layers capture more invariances [1]
- More data to train deeper networks
- More computing (GPUs)
- Better regularization: Dropout
- New nonlinearities
  - Max pooling, Rectified linear units (ReLU) [2]

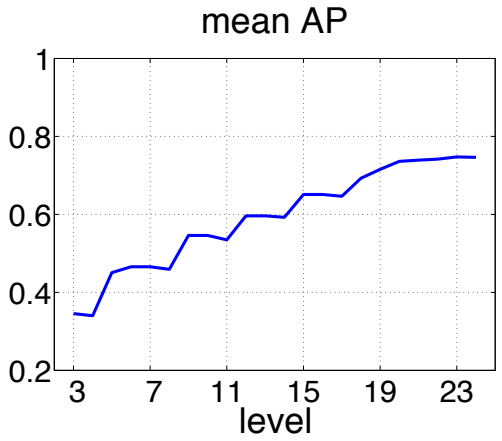


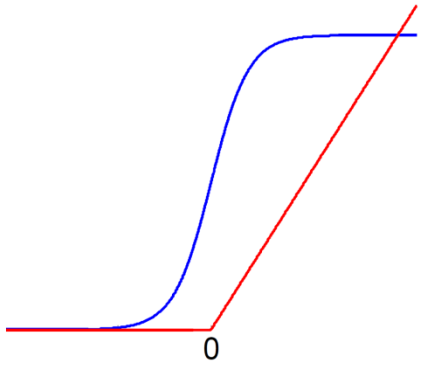


[1] Razavian, Azizpour, Sullivan, Carlsson, CNN Features off-the-shelf: an Astounding Baseline for Recognition. CVPRW'14.
[2] Hahnloser, Sarpeshkar, Mahowald, Douglas, Seung. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. Nature, 405(6789):947–951, 2000.

Why These Improvements in Performance?

- Features are learned rather than hand-crafted
- More layers capture more invariances [1]
- More data to train deeper networks
- More computing (GPUs)
- Better regularization: Dropout
- New nonlinearities
  - Max pooling, Rectified linear units (ReLU) [2]
- Theoretical understanding of deep networks remains shallow

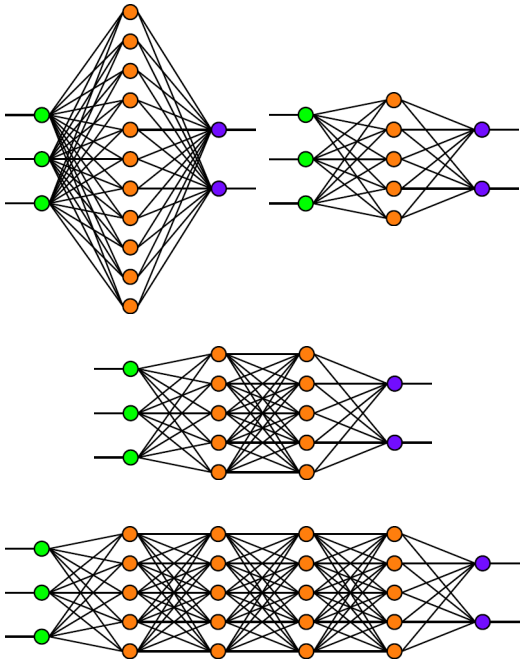




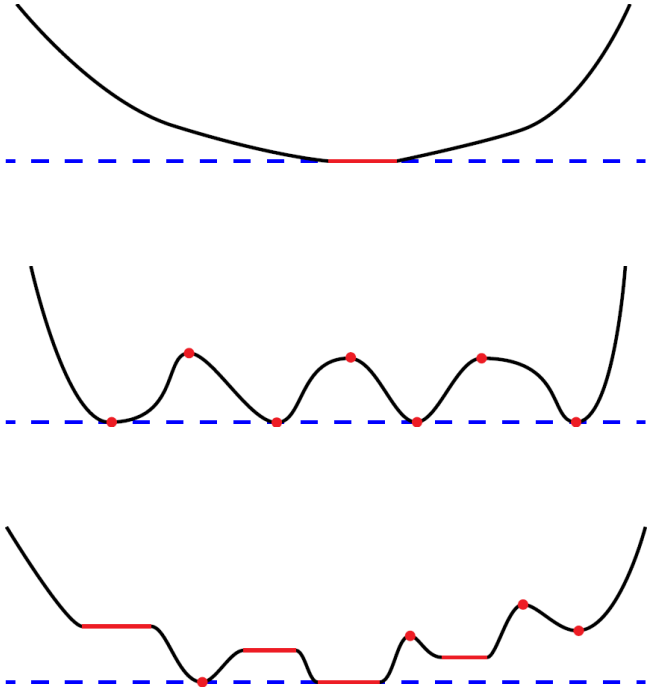
[1] Razavian, Azizpour, Sullivan, Carlsson, CNN Features off-the-shelf: an Astounding Baseline for Recognition. CVPRW'14.
[2] Hahnloser, Sarpeshkar, Mahowald, Douglas, Seung. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. Nature, 405(6789):947–951, 2000.

Key Theoretical Questions in Deep Learning

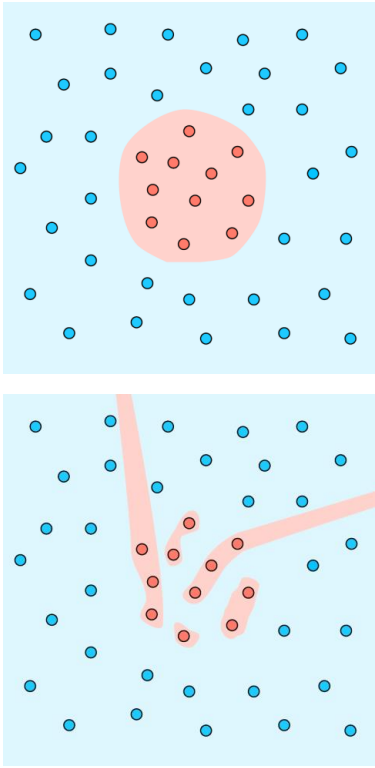
Architecture Design



Optimization



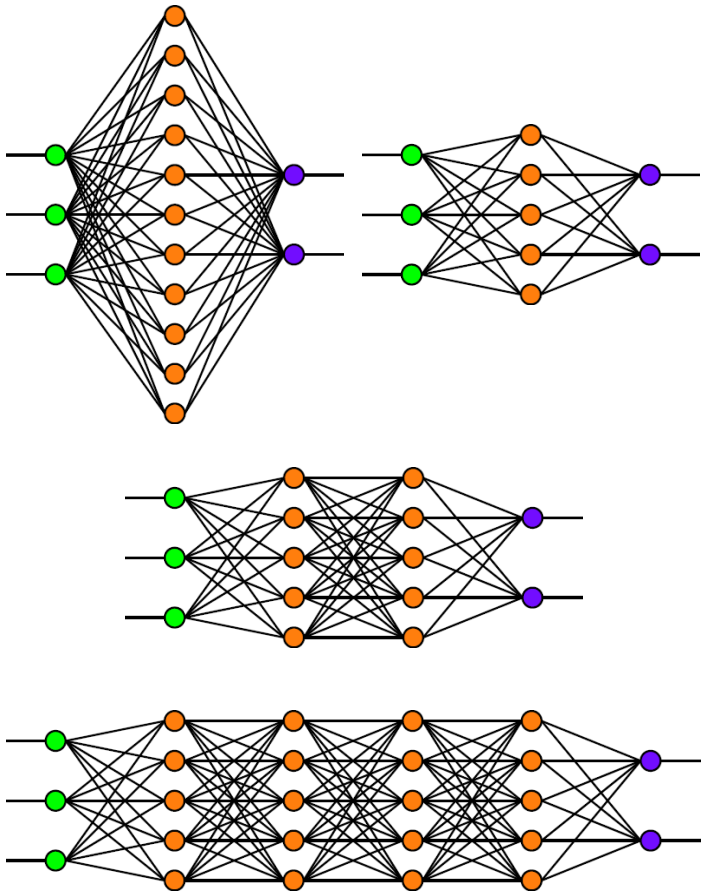
Generalization



Slide courtesy of Ben Haeffele

Key Theoretical Questions: Architecture

- **Are there principled ways to design networks?**
  - How many layers?
  - Size of layers?
  - Choice of layer types?
  - What classes of functions can be approximated by a feedforward neural network?
  - How does the architecture impact expressiveness? [1]



Slide courtesy of Ben Haeffele

[1] Cohen, et al., “On the expressive power of deep learning: A tensor analysis.” COLT. (2016)

Key Theoretical Questions: Architecture

- **Approximation, depth, width and invariance: earlier work**
  - Perceptrons and multilayer feedforward networks are universal approximators [Cybenko ’89, Hornik ’89, Hornik ’91, Barron ’93]

Key Theoretical Questions: Architecture

- **Approximation, depth, width and invariance: earlier work**
  - Perceptrons and multilayer feedforward networks are universal approximators [Cybenko '89, Hornik '89, Hornik '91, Barron '93]

Theorem [C'89, H'91] Let $\rho()$ be a bounded, non-constant continuous function. Let I_m denote the m-dimensional hypercube, and $C(I_m)$ denote the space of continuous functions on I_m. Given any $f \in C(I_m)$ and $\epsilon > 0$, there exists $N > 0$ and $v_i, w_i, b_i,\ i = 1 \ldots, N$ such that

$$F(x) = \sum_{i \leq N} v_i \rho(w_i^T x + b_i) \quad \text{satisfies}$$

$$\sup_{x \in I_m} |f(x) - F(x)| < \epsilon \ .$$

Key Theoretical Questions: Architecture

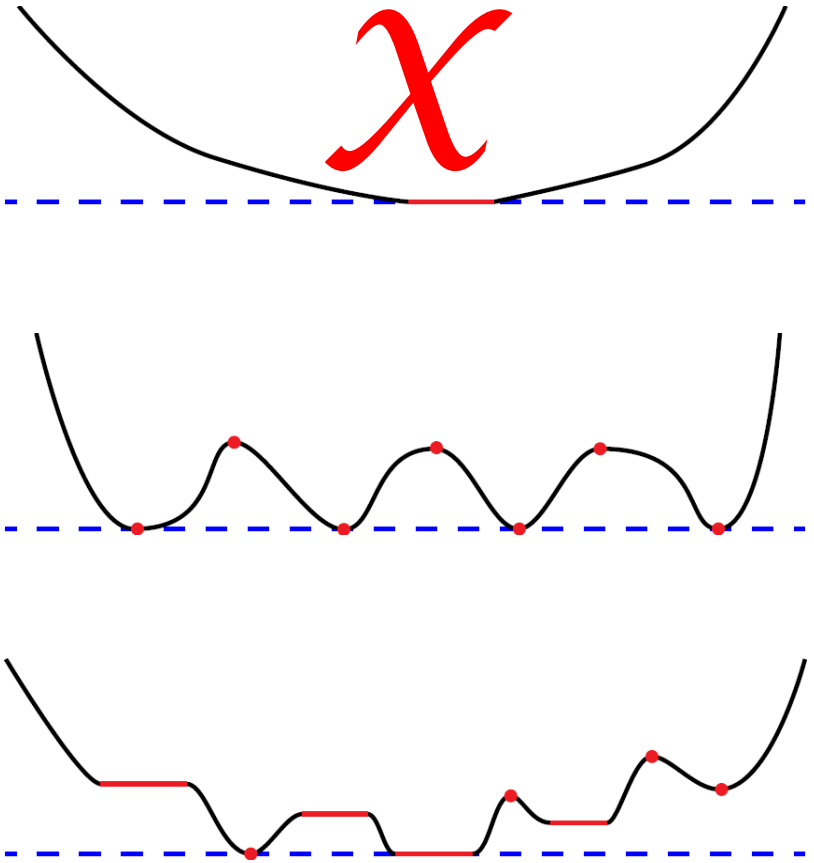
- **Approximation, depth, width and invariance: earlier work**
  - Perceptrons and multilayer feedforward networks are universal approximators [Cybenko ’89, Hornik ’89, Hornik ’91, Barron ’93]

- **Approximation, depth, width and invariance: recent work**
  - Gaps between deep and shallow networks [Montufar’14, Mhaskar’16]
  - Deep Boltzmann machines are universal approximators [Montufar’15]
  - Design of CNNs via hierarchical tensor decompositions [Cohen ’17]
  - Scattering networks are deformation stable for Lipschitz non-linearities [Bruna-Mallat ’13, Wiatowski ’15, Mallat ’16]
  - Exponential # of units needed to approximate deep net [Telgarsky’16]
  - Memory-optimal neural network approximation [Bölcskei ’17]

[1] Cybenko. Approximations by superpositions of sigmoidal functions, Mathematics of Control, Signals, and Systems, 2 (4), 303-314, 1989.
[2] Hornik, Stinchcombe and White. Multilayer feedforward networks are universal approximators, Neural Networks, 2(3), 359-366, 1989.
[3] Hornik. Approximation Capabilities of Multilayer Feedforward Networks, Neural Networks, 4(2), 251–257, 1991.
[4] Barron. Universal approximation bounds for superpositions of a sigmoidal function. IEEE Transactions on Information Theory, 39(3):930–945, 1993.
[5] Cohen et al. Analysis and Design of Convolutional Networks via Hierarchical Tensor Decompositions arXiv preprint arXiv:1705.02302
[6] Montúfar, Pascanu, Cho, Bengio, On the number of linear regions of deep neural networks, NIPS, 2014
[7] Mhaskar, Poggio. Deep vs. shallow networks: An approximation theory perspective. Analysis and Applications, 2016.
[8] Montúfar et al, Deep narrow Boltzmann machines are universal approximators, ICLR 2015, arXiv:1411.3784v3
[9] Bruna and Mallat. Invariant scattering convolution networks. Trans. PAMI, 35(8):1872–1886, 2013.
[10] Wiatowski, Bölcskei. A mathematical theory of deep convolutional neural networks for feature extraction. arXiv2015.
[11] Mallat. Understanding deep convolutional networks. Phil. Trans. R. Soc. A, 374(2065), 2016.
[12] Telgarsky, Benefits of depth in neural networks. COLT 2016.
[13] Bölcskei, Grohs, Kutyniok, Petersen. Memory-optimal neural network approximation. Wavelets and Sparsity 2017.

Key Theoretical Questions: Optimization

- **How to train neural networks?**
  - Problem is non-convex
  - What does the error surface look like?
  - How to guarantee optimality?
  - When does local descent succeed?



Slide courtesy of Ben Haeffele

Key Theoretical Questions: Optimization

- **Optimization theory: earlier work**
  - No spurious local minima for linear networks [Baldi-Hornik ’89]
  - Backprop fails to converge for nonlinear networks [Brady’89], converges for linearly separable data [Gori-Tesi’91-’92], or it gets stuck [Frasconi’97]
  - Local minima and plateaus in multilayer perceptrons [Fukumizu-Amari’00]

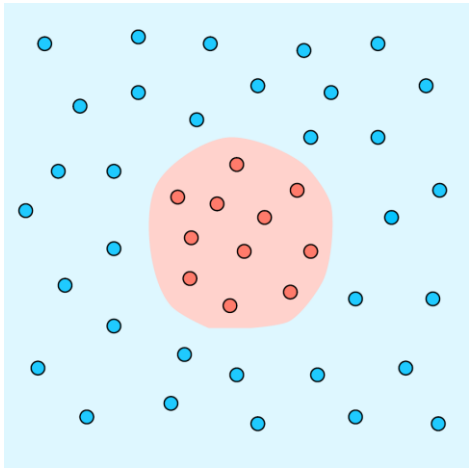
Key Theoretical Questions: Optimization

- **Optimization theory: earlier work**
  - No spurious local minima for linear networks [Baldi-Hornik '89]
  - Backprop fails to converge for nonlinear networks [Brady'89], converges for linearly separable data [Gori-Tesi'91-'92], or it gets stuck [Frasconi'97]
  - Local minima and plateaus in multilayer perceptrons [Fukumizu-Amari'00]
- **Optimization theory: recent work**
  - Convex neural networks in infinite number of variables [Bengio '05]
  - The loss surface of multilayer networks [Choromanska '15]
  - No spurious local minima for deep linear networks and square loss [Kawaguchi'16]
  - No spurious local minima for positively homogeneous networks [Haeffele-Vidal'15], but infinitely many local minima in general [Yun '18]
  - Attacking the saddle point problem [Dauphin '14]
  - Effect of gradient noise on the energy landscape [Chaudhari '15, Soudry '16]
  - Entropy-SGD is biased toward wide valleys [Chaudhari '17]
  - Deep relaxation: PDEs for optimizing deep nets [Chaudhari '17]
  - Guaranteed training of NNs using tensor methods [Janzamin '15]

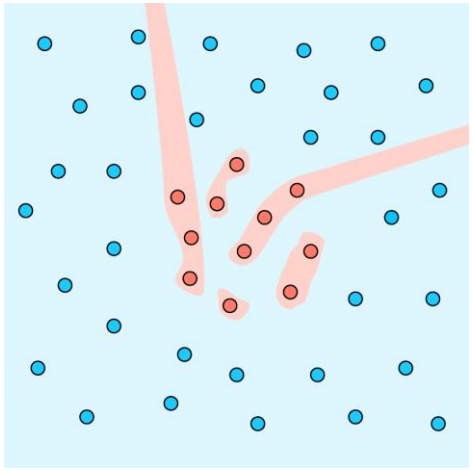
Key Theoretical Questions: Generalization

- **Classification performance guarantees?**
  - How well do deep networks generalize?
  - How should networks be regularized?
  - How to prevent under or over fitting?

✓ Simple



✗ Complex



Slide courtesy of Ben Haeffele

Key Theoretical Questions: Generalization

- **Generalization and regularization theory: earlier work**
  - # training examples grows polynomially with network size [1,2]

[1] Sontag. VC Dimension of Neural Networks. Neural Networks and Machine Learning, 1998.
[2] Bartlett, Maass. VC dimension of neural nets. The handbook of brain theory and neural networks, 2003.
[3] Caruana, Lawrence, Giles. Overfitting in neural nets: Backpropagation, conjugate gradient & early stopping. NIPS01.
[4] Srivastava. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. JMLR, 2014.
[5] Wan. Regularization of neural networks using dropconnect. ICML, 2013.
[6] Giryes, Sapiro, Bronstein. Deep Neural Networks with Random Gaussian Weights. arXiv:1504.08291.
[7] Sokolic. Margin Preservation of Deep Neural Networks, 2015
[8] Neyshabur. Path-SGD: Path-Normalized Optimization in Deep Neural Networks. NIPS 2015
[9] Behnam Neyshabur. Implicit Regularization in Deep Learning. PhD Thesis 2017
[10] Sokolic, Giryes, Sapiro, Rodrigues. Generalization error of invariant classifiers. In AISTATS, 2017.
[11] Sokolić, Giryes, Sapiro, Rodrigues. Robust Large Margin Deep Neural Networks. IEEE Transactions on Signal Processing, 2017.
[12] Shwartz-Ziv, Tishby. Opening the black box of deep neural networks via information. arXiv:1703.00810, 2017.
[13] Achille, Soatto. Information dropout: Learning optimal representations through noisy computation. *arXiv:* 2016.
[14] Liang, Poggio, Rakhlin, Stokes. Fisher-Rao Metric, Geometry and Complexity of Neural Networks. arXiv: 2017.
[15] Zhang, Bengio, Hardt, Recht, Vinyals. Understanding deep learning requires rethinking generalization. ICLR 2017.

Key Theoretical Questions: Generalization

- **Generalization and regularization theory: earlier work**
  - # training examples grows polynomially with network size [1,2]
- **Regularization methods: earlier and recent work**
  - Early stopping [3]
  - Dropout, Dropconnect, and extensions (adaptive, annealed) [4,5]

[1] Sontag. VC Dimension of Neural Networks. Neural Networks and Machine Learning, 1998.
[2] Bartlett, Maass. VC dimension of neural nets. The handbook of brain theory and neural networks, 2003.
[3] Caruana, Lawrence, Giles. Overfitting in neural nets: Backpropagation, conjugate gradient & early stopping. NIPS01.
[4] Srivastava. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. JMLR, 2014.
[5] Wan. Regularization of neural networks using dropconnect. ICML, 2013.
[6] Giryes, Sapiro, Bronstein. Deep Neural Networks with Random Gaussian Weights. arXiv:1504.08291.
[7] Sokolic. Margin Preservation of Deep Neural Networks, 2015
[8] Neyshabur. Path-SGD: Path-Normalized Optimization in Deep Neural Networks. NIPS 2015
[9] Behnam Neyshabur. Implicit Regularization in Deep Learning. PhD Thesis 2017
[10] Sokolic, Giryes, Sapiro, Rodrigues. Generalization error of invariant classifiers. In AISTATS, 2017.
[11] Sokolić, Giryes, Sapiro, Rodrigues. Robust Large Margin Deep Neural Networks. IEEE Transactions on Signal Processing, 2017.
[12] Shwartz-Ziv, Tishby. Opening the black box of deep neural networks via information. arXiv:1703.00810, 2017.
[13] Achille, Soatto. Information dropout: Learning optimal representations through noisy computation. *arXiv:* 2016.
[14] Liang, Poggio, Rakhlin, Stokes. Fisher-Rao Metric, Geometry and Complexity of Neural Networks. arXiv: 2017.
[15] Zhang, Bengio, Hardt, Recht, Vinyals. Understanding deep learning requires rethinking generalization. ICLR 2017.

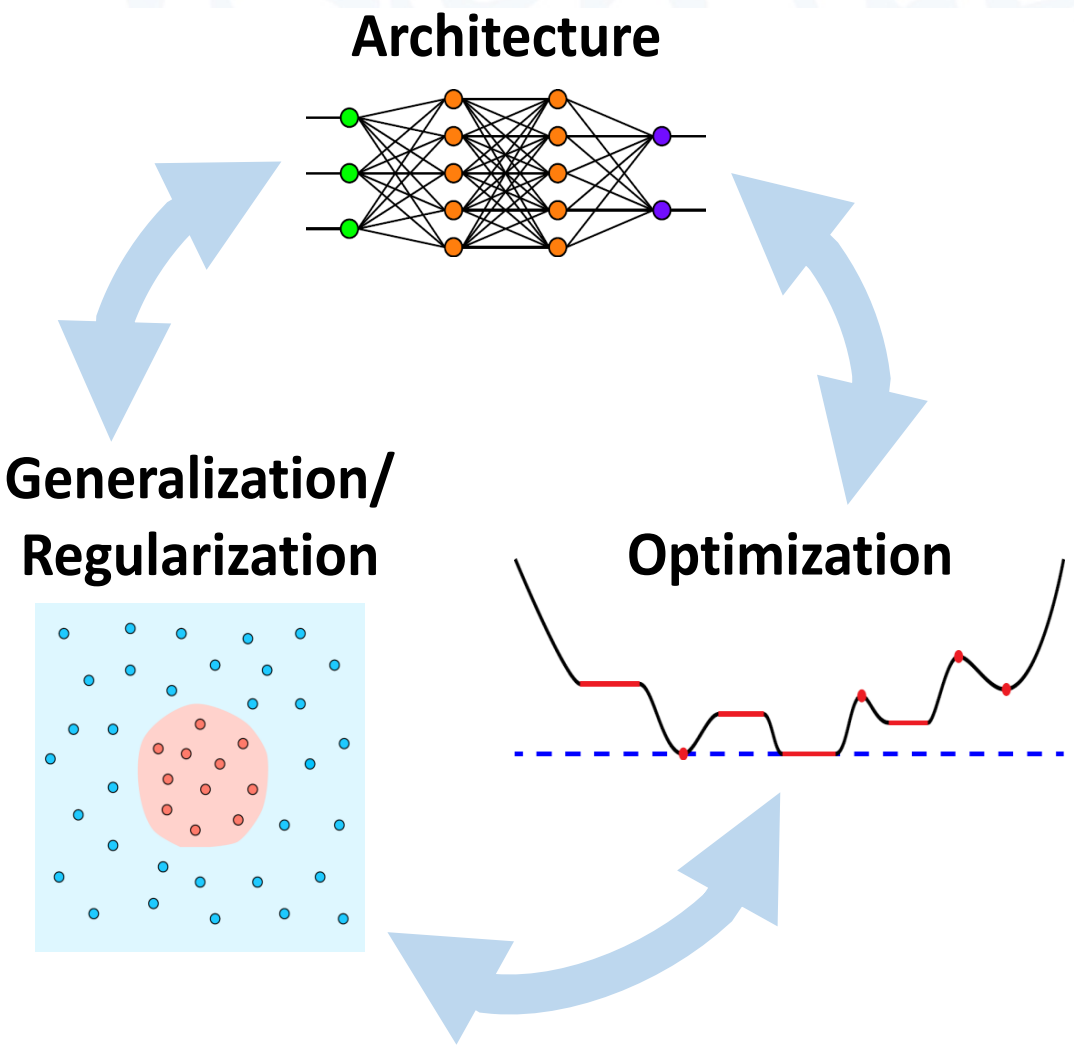
Key Theoretical Questions: Generalization

- **Generalization and regularization theory: earlier work**
  - # training examples grows polynomially with network size [1,2]
- **Regularization methods: earlier and recent work**
  - Early stopping [3]
  - Dropout, Dropconnect, and extensions (adaptive, annealed) [4,5]
- **Generalization and regularization theory: recent work**
  - Distance and margin-preserving embeddings [6,7]
  - Path SGD/implicit regularization & generalization bounds [8,9]
  - Product of norms regularization & generalization bounds [10,11]
  - Information theory: info bottleneck, info dropout, Fisher-Rao [12,13,14]
  - Rethinking generalization: [15]

[1] Sontag. VC Dimension of Neural Networks. Neural Networks and Machine Learning, 1998.
[2] Bartlett, Maass. VC dimension of neural nets. The handbook of brain theory and neural networks, 2003.
[3] Caruana, Lawrence, Giles. Overfitting in neural nets: Backpropagation, conjugate gradient & early stopping. NIPS01.
[4] Srivastava. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. JMLR, 2014.
[5] Wan. Regularization of neural networks using dropconnect. ICML, 2013.
[6] Giryes, Sapiro, Bronstein. Deep Neural Networks with Random Gaussian Weights. arXiv:1504.08291.
[7] Sokolic. Margin Preservation of Deep Neural Networks, 2015
[8] Neyshabur. Path-SGD: Path-Normalized Optimization in Deep Neural Networks. NIPS 2015
[9] Behnam Neyshabur. Implicit Regularization in Deep Learning. PhD Thesis 2017
[10] Sokolic, Giryes, Sapiro, Rodrigues. Generalization error of invariant classifiers. In AISTATS, 2017.
[11] Sokolić, Giryes, Sapiro, Rodrigues. Robust Large Margin Deep Neural Networks. IEEE Transactions on Signal Processing, 2017.
[12] Shwartz-Ziv, Tishby. Opening the black box of deep neural networks via information. arXiv:1703.00810, 2017.
[13] Achille, Soatto. Information dropout: Learning optimal representations through noisy computation. *arXiv:* 2016.
[14] Liang, Poggio, Rakhlin, Stokes. Fisher-Rao Metric, Geometry and Complexity of Neural Networks. arXiv: 2017.
[15] Zhang, Bengio, Hardt, Recht, Vinyals. Understanding deep learning requires rethinking generalization. ICLR 2017.

Key Theoretical Questions are Interrelated

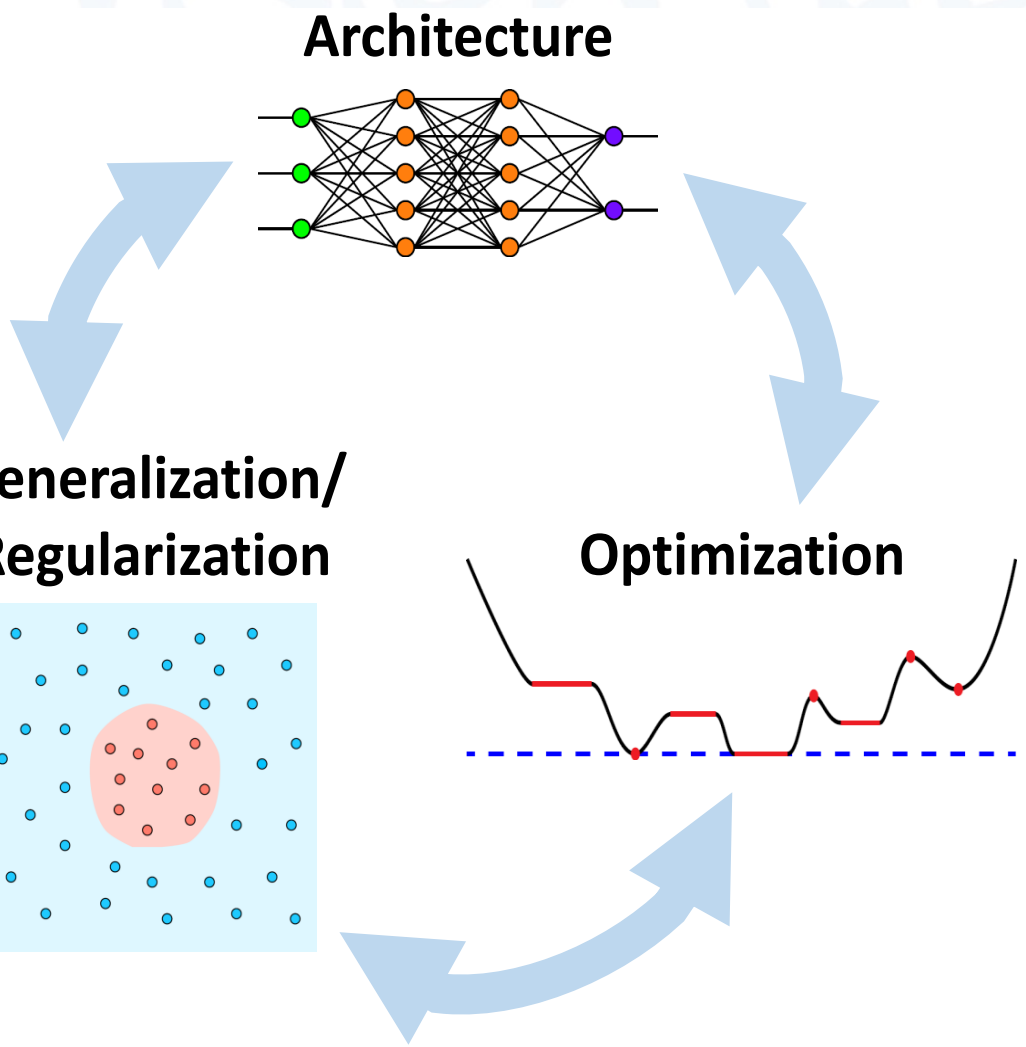
- Optimization can impact generalization [1,2]
- Architecture has strong effect on generalization [3]
- Some architectures could be easier to optimize than others [4]



[1] Neyshabur et. al. In Search of the Real Inductive Bias: On the Role of Implicit Regularization in Deep Learning." ICLR workshop. (2015).
[2] P. Zhou, J. Feng. The Landscape of Deep Learning Algorithms. 1705.07038, 2017
[3] Zhang, et al., "Understanding deep learning requires rethinking generalization." ICLR. (2017).
[4] Haeffele, Vidal. Global optimality in neural network training. CVPR 2017.

Toward a Unified Theory?

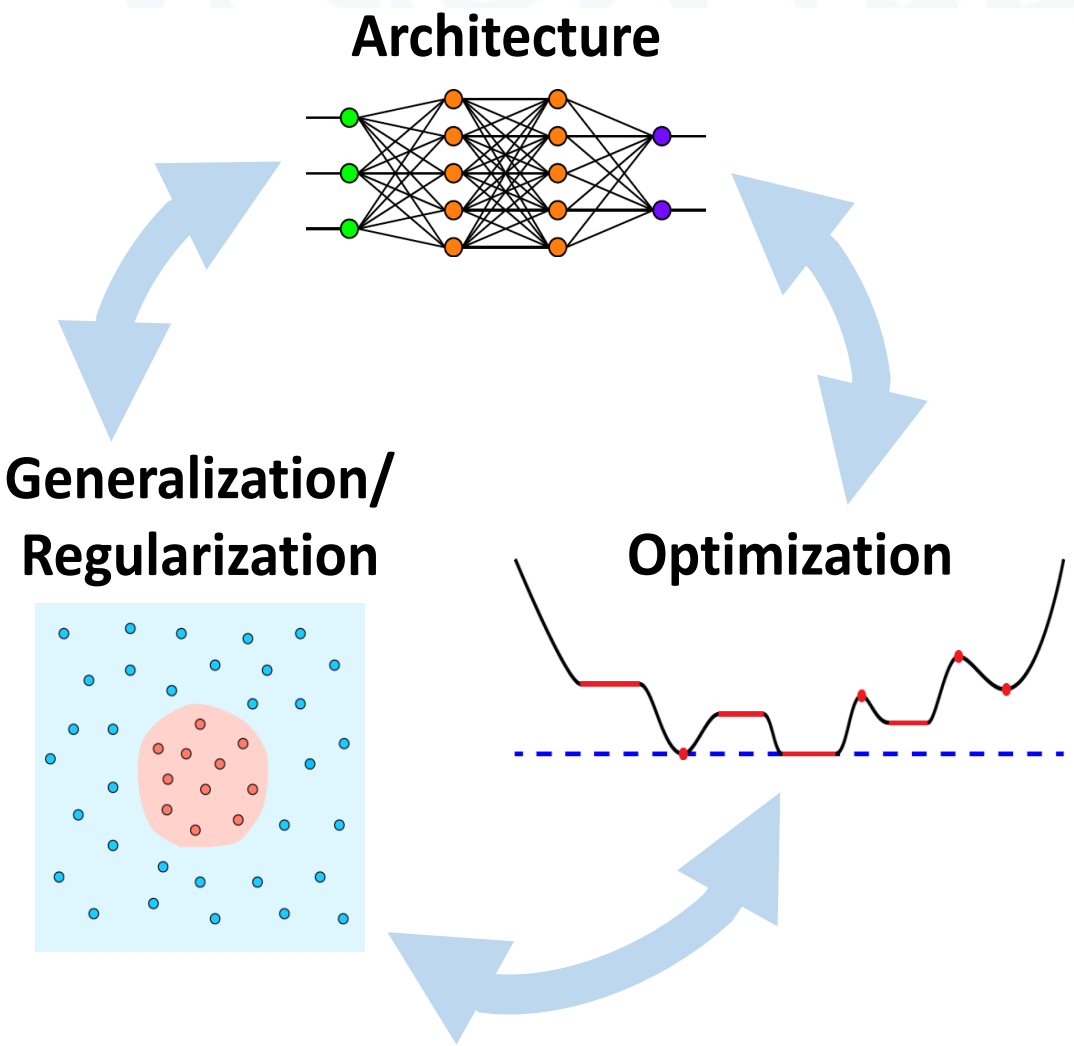
- Dropout regularization is equivalent to regularization with products of weights [1,2]
- Regularization with product of weights generalizes well [3,4]
- No spurious local minima for product of weight regularizers [5]



[1] Cavazza, Lane, Moreiro, Haeffele, Murino, Vidal. An Analysis of Dropout for Matrix Factorization, AISTATS 2018.
[2] Poorya Mianjy, Raman Arora, Rene Vidal. On the Implicit Bias of Dropout. ICML 2018.
[3] Neyshabur, Salakhutdinov, Srebro. Path-SGD: Path-Normalized Optimization in Deep Neural Networks. NIPS 2015
[4] Sokolic, Giryes, Sapiro, Rodrigues. Generalization error of Invariant Classifiers. AISTATS, 2017.
[5] Haeffele, Vidal. Global optimality in neural network training. CVPR 2017.

Part I: Analysis of Optimization

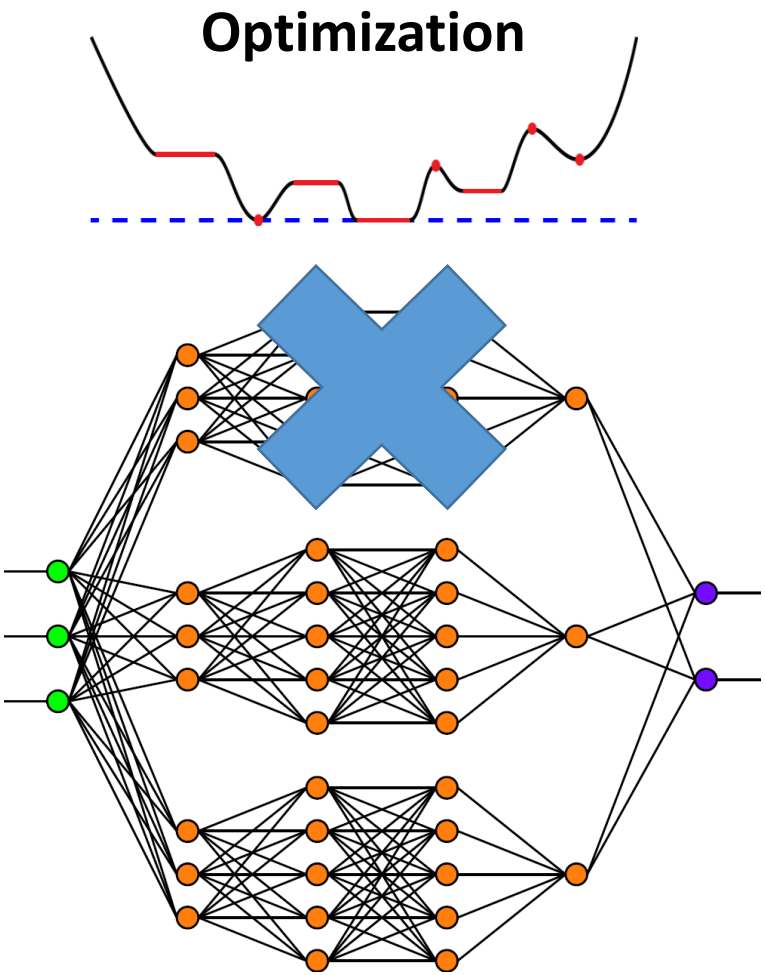
- What properties of the network architecture facilitate optimization?
  - Positive homogeneity
  - Parallel subnetwork structure
- What properties of the regularization function facilitate optimization?
  - Positive homogeneity
  - Adapt network structure to the data [1]



Picture courtesy of Ben Haeffele

[1] Bengio, et al., “Convex neural networks.” NIPS. (2005)

Main Results

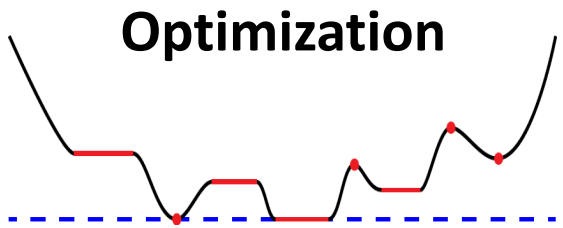


Theorem 1:
A local minimum such that all the weights from one subnetwork are zero is a global minimum

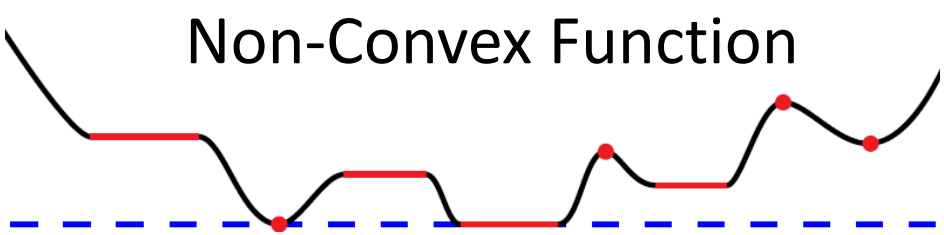
[1] Haeffele, Young, Vidal. Structured Low-Rank Matrix Factorization: Optimality, Algorithm, and Applications to Image Processing, ICML '14
[2] Haeffele, Vidal. Global Optimality in Tensor Factorization, Deep Learning and Beyond, arXiv, '15
[3] Haeffele, Vidal. Global optimality in neural network training. CVPR 2017.

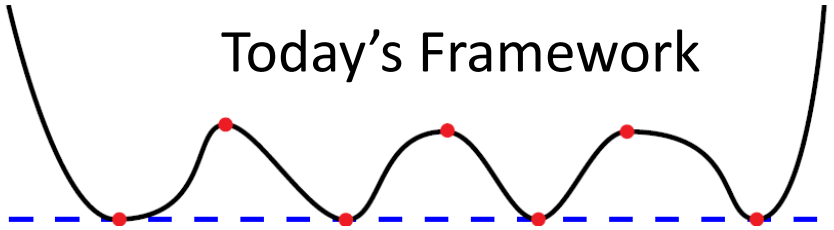
JOHNS HOPKINS
MATHEMATICAL INSTITUTE
for DATA SCIENCE

Main Results



Theorem 2:
If the size of the network is large enough, local descent can reach a global minimizer from any initialization





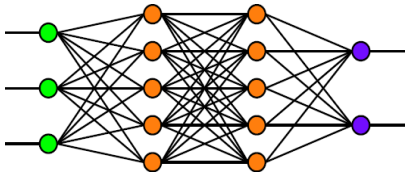
[1] Haeffele, Young, Vidal. Structured Low-Rank Matrix Factorization: Optimality, Algorithm, and Applications to Image Processing, ICML '14
[2] Haeffele, Vidal. Global Optimality in Tensor Factorization, Deep Learning and Beyond, arXiv, '15
[3] Haeffele, Vidal. Global optimality in neural network training. CVPR 2017.

JOHNS HOPKINS
MATHEMATICAL INSTITUTE
for DATA SCIENCE

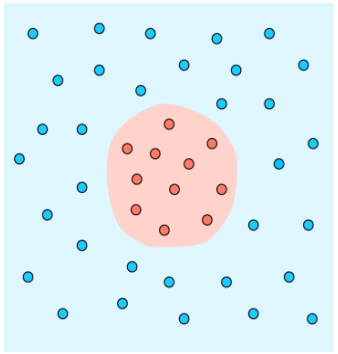
Part II: Analysis of Dropout for Linear Nets

- What objective function is being minimized by dropout?
- What type of regularization is induced by dropout?
- What are the properties of the optimal weights?

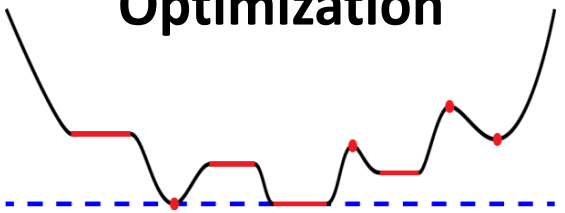
Architecture



Generalization/ Regularization

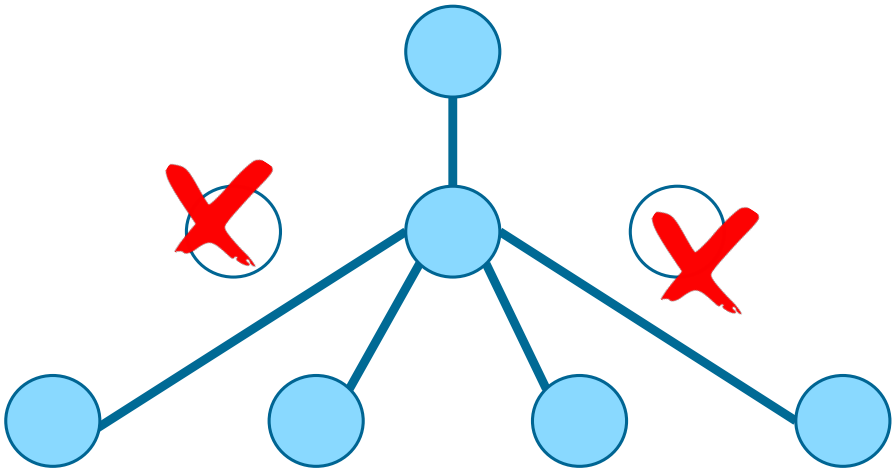


Optimization



Picture courtesy of Ben Haeffele

Main Results for Linear Nets



Theorem 3:
Dropout is SGD applied to a stochastic objective.

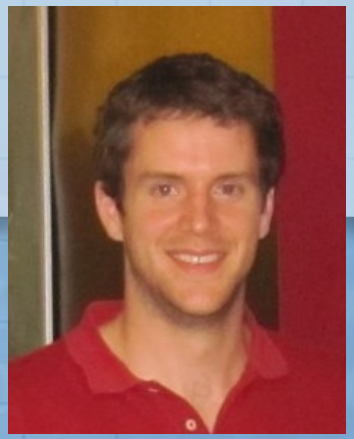
Theorem 4:
Dropout induces explicit low-rank regularization (nuclear norm squared).

Theorem 5:
Dropout induces balanced weights.

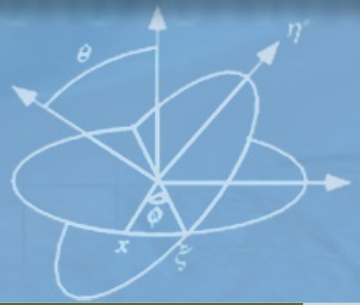
Jacopo Cavazza, Connor Lane, Benjamin D. Haeffele, Vittorio Murino, René Vidal. An Analysis of Dropout for Matrix Factorization. AISTATS 2018

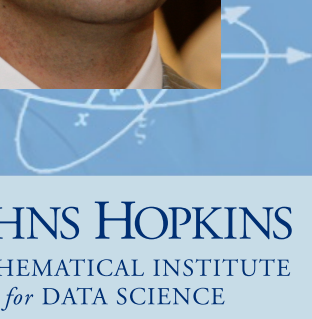
JHU vision lab

Global Optimality in Matrix and Tensor Factorization, Deep Learning & Beyond



Ben Haeffele and René Vidal
Center for Imaging Science
Mathematical Institute for Data Science
Johns Hopkins University



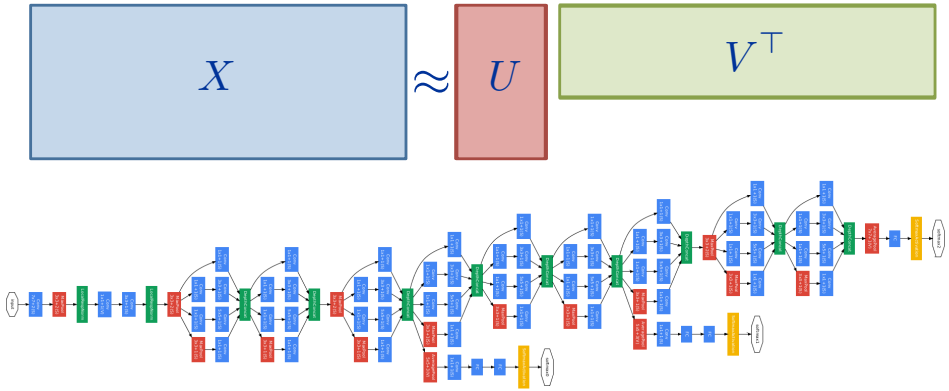


THE DEPARTMENT OF BIOMEDICAL ENGINEERING
The Whitaker Institute at Johns Hopkins

JOHNS HOPKINS
MATHEMATICAL INSTITUTE
for DATA SCIENCE

Outline

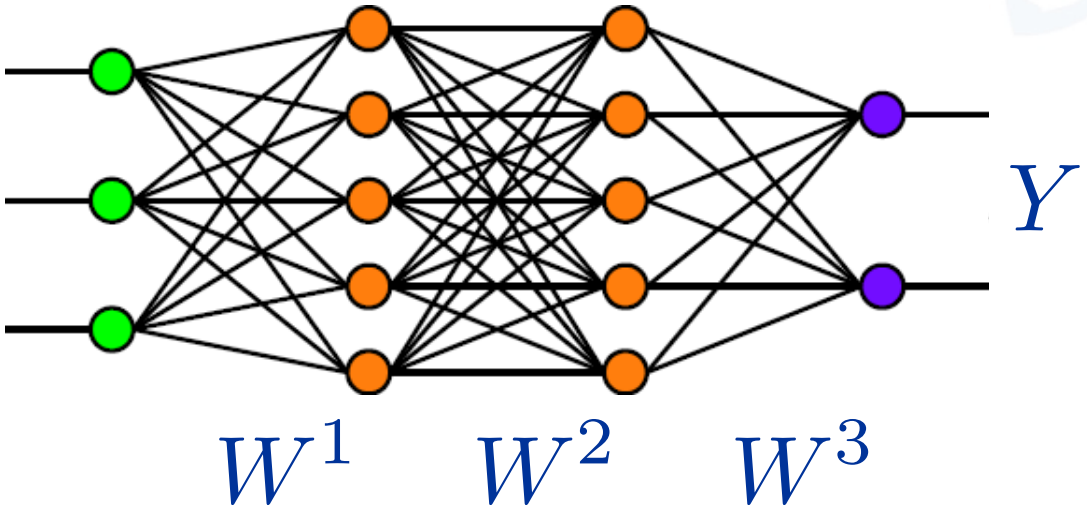
- **Architecture properties that facilitate optimization**
  - Positive homogeneity
  - Parallel subnetwork structure
- **Regularization properties that facilitate optimization**
  - Positive homogeneity
  - Adapt network structure to the data
- **Theoretical guarantees**
  - Sufficient conditions for global optimality
  - Local descent can reach global minimizers



[1] Haeffele, Young, Vidal. Structured Low-Rank Matrix Factorization: Optimality, Algorithm, and Applications to Image Processing, ICML '14
[2] Haeffele, Vidal. Global Optimality in Tensor Factorization, Deep Learning and Beyond, arXiv, '15
[3] Haeffele, Vidal. Global optimality in neural network training. CVPR 2017.

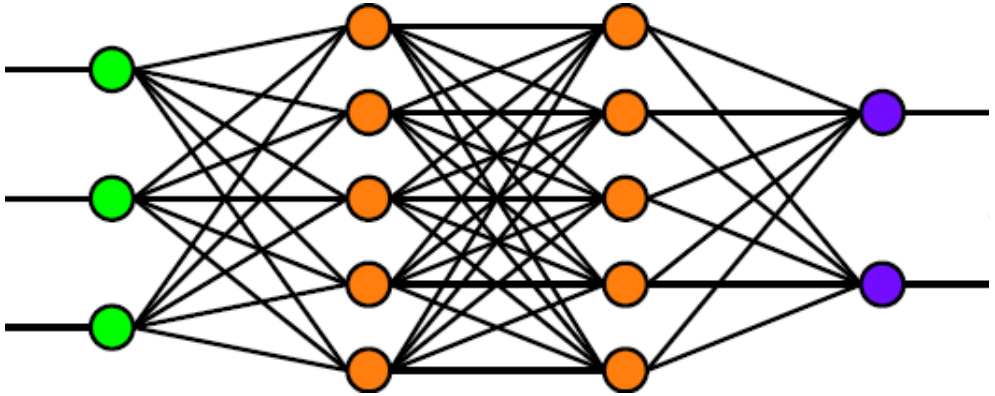
Key Property #1: Positive Homogeneity

- Start with a network



Key Property #1: Positive Homogeneity

- Start with a network



$$W^1 \quad W^2 \quad W^3$$

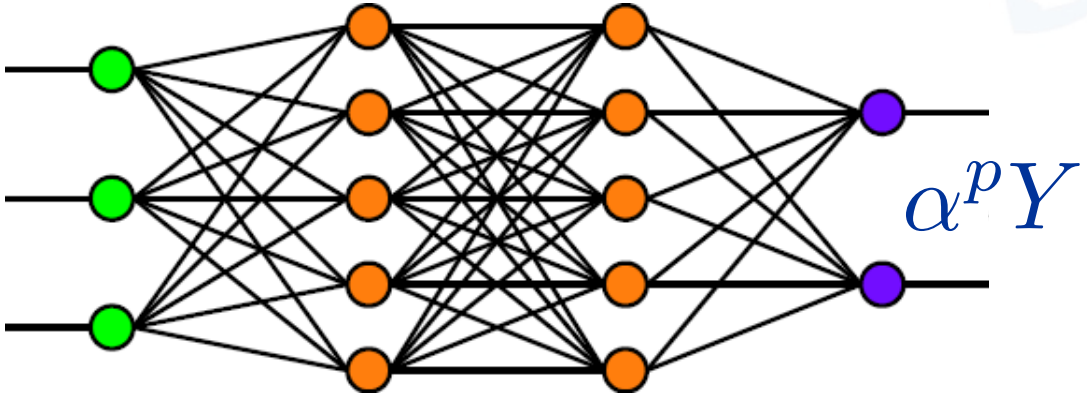
- Scale the weights by

$$\alpha \geq 0$$

$$\alpha W^1 \quad \alpha W^2 \quad \alpha W^3$$

Key Property #1: Positive Homogeneity

- Start with a network



- Scale the weights by

$$\alpha \geq 0$$

$$\alpha W^1 \;\; \alpha W^2 \;\; \alpha W^3$$

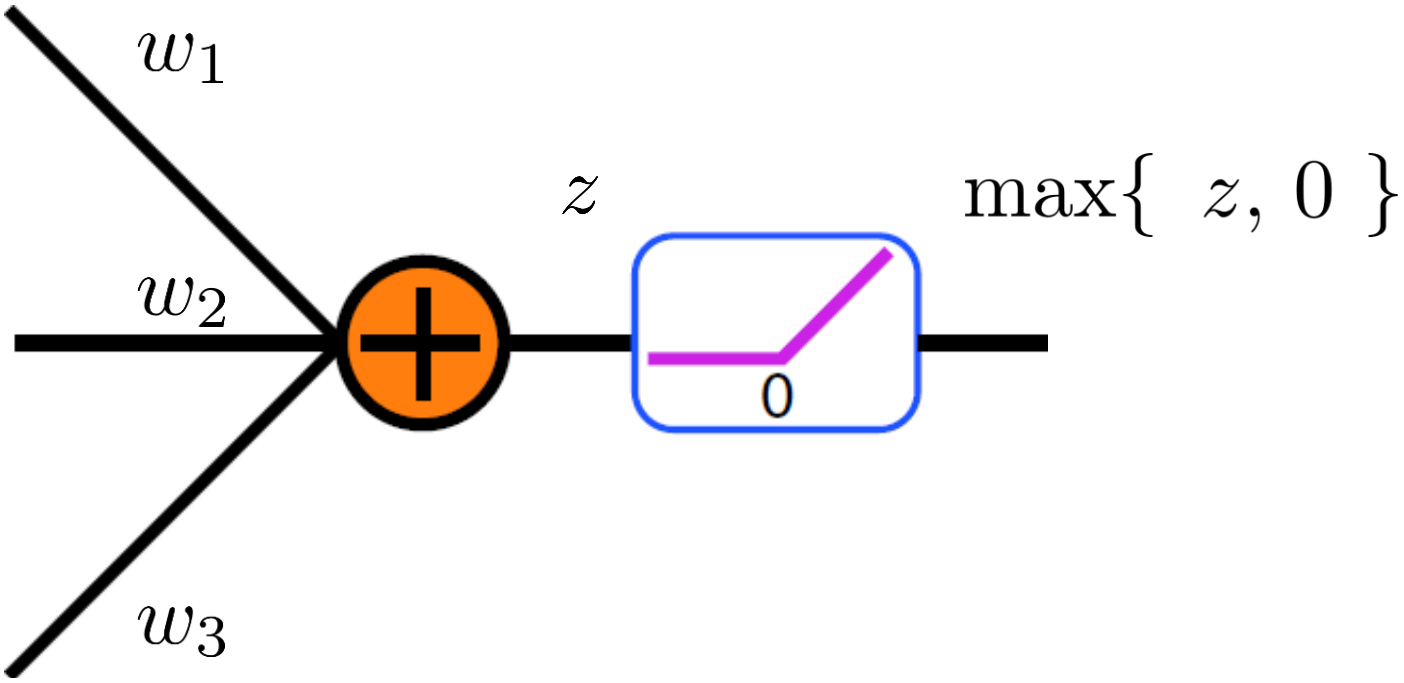
- Output is scaled by α^p, where p = degree of homogeneity

$$\Phi(W^1, W^2, W^3) = Y$$

$$\Phi(\alpha W^1, \alpha W^2, \alpha W^3) = \alpha^p Y$$

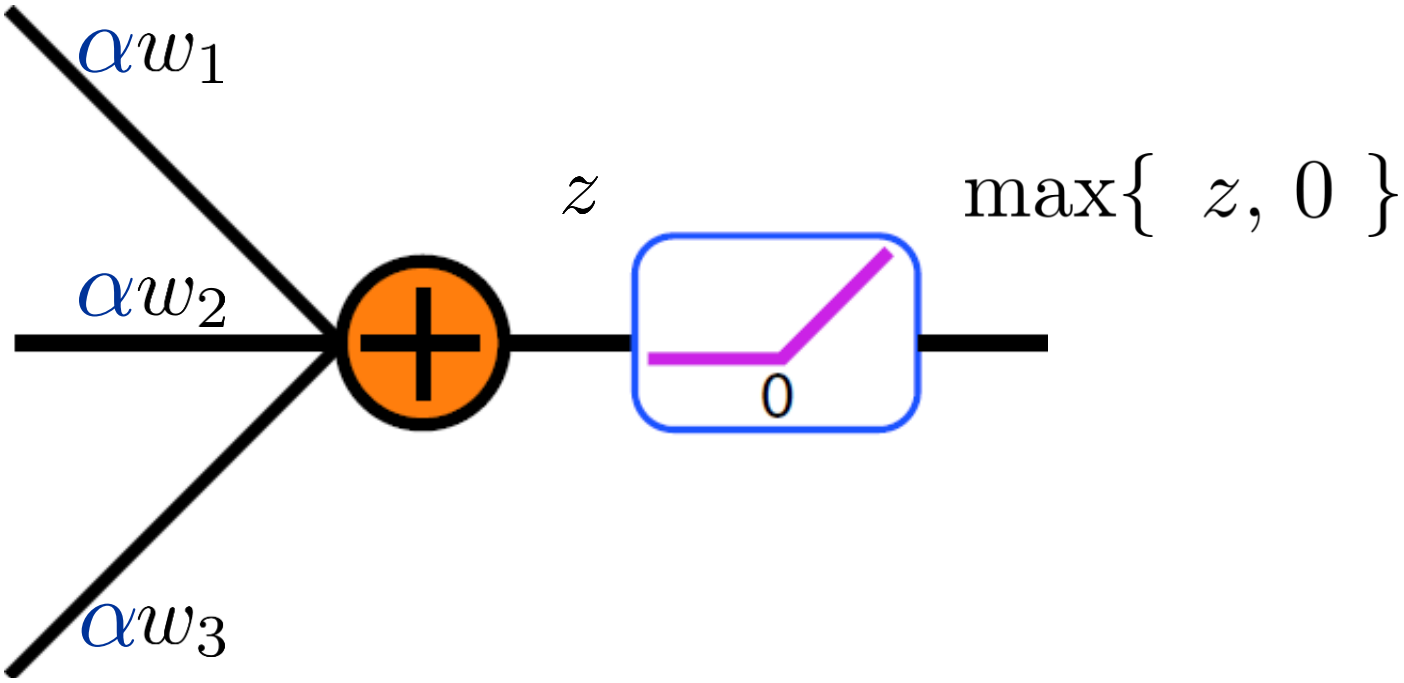
Examples of Positively Homogeneous Maps

- **Example 1**: Rectified Linear Units (ReLU)



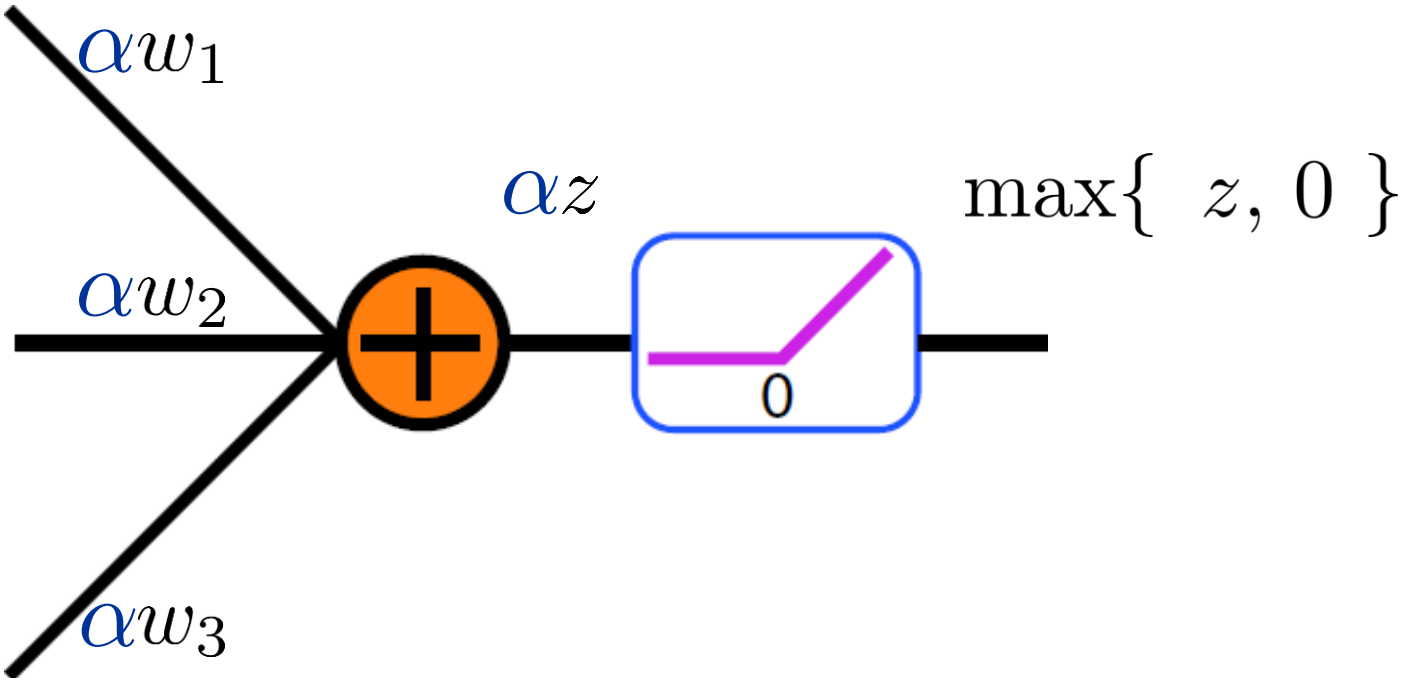
Examples of Positively Homogeneous Maps

- **Example 1**: Rectified Linear Units (ReLU)



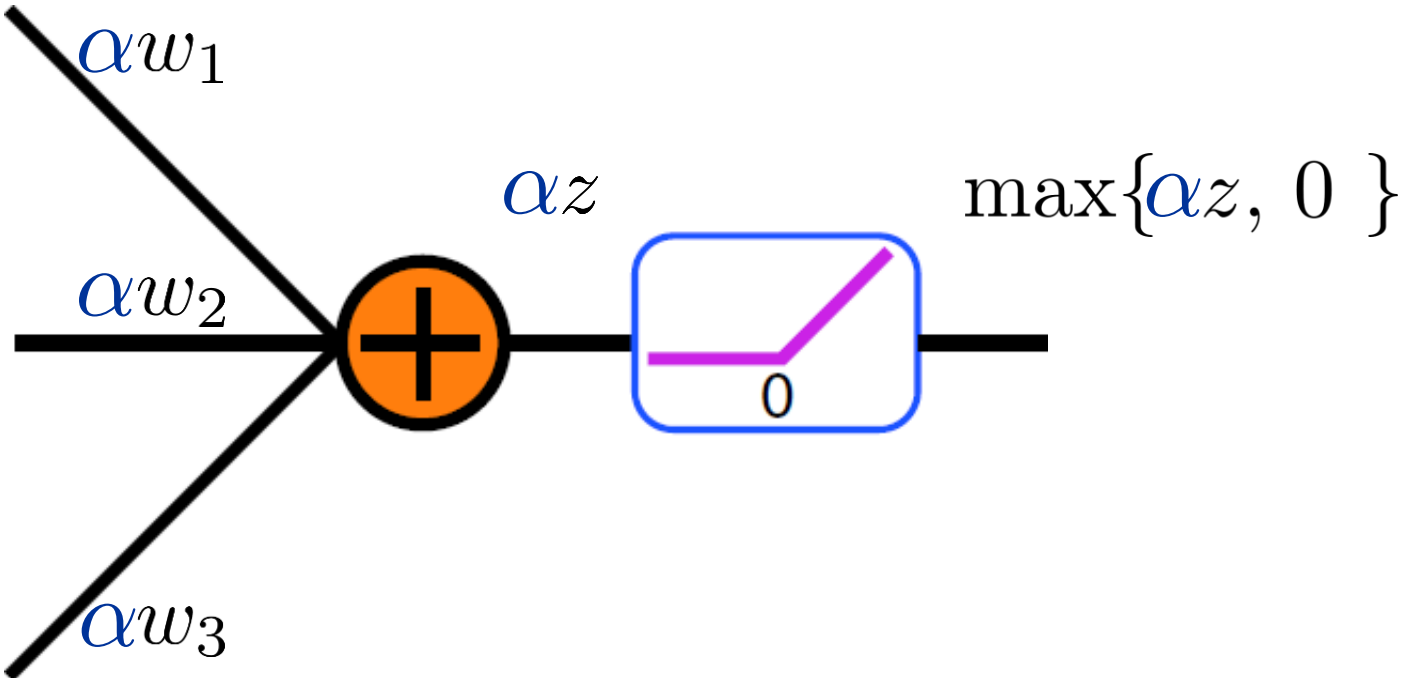
Examples of Positively Homogeneous Maps

- **Example 1**: Rectified Linear Units (ReLU)



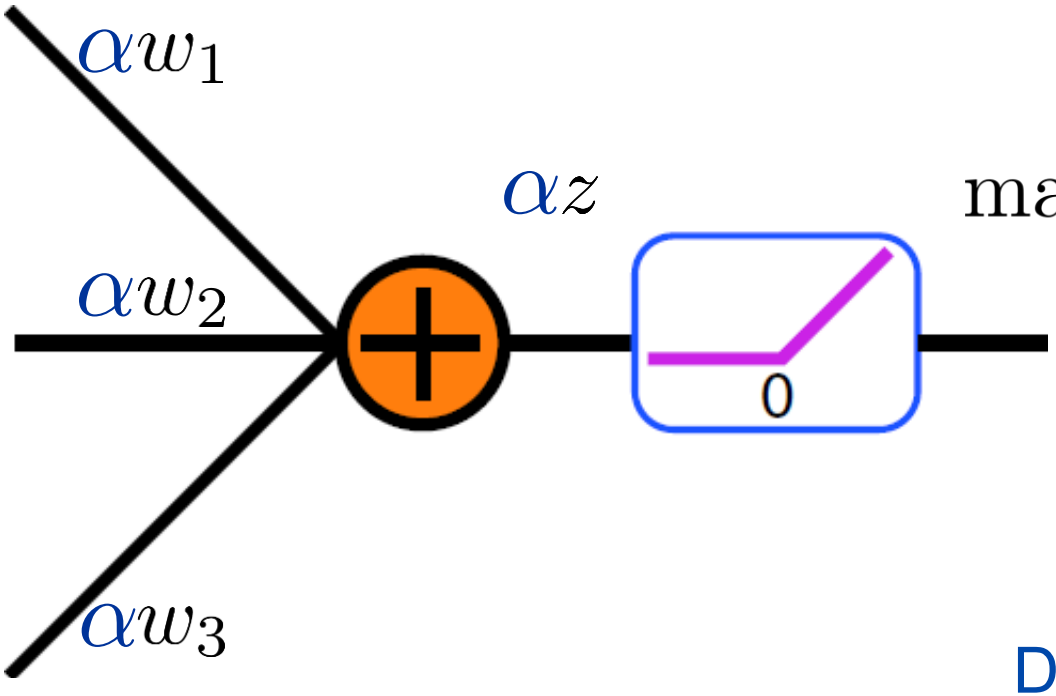
Examples of Positively Homogeneous Maps

- **Example 1**: Rectified Linear Units (ReLU)



Examples of Positively Homogeneous Maps

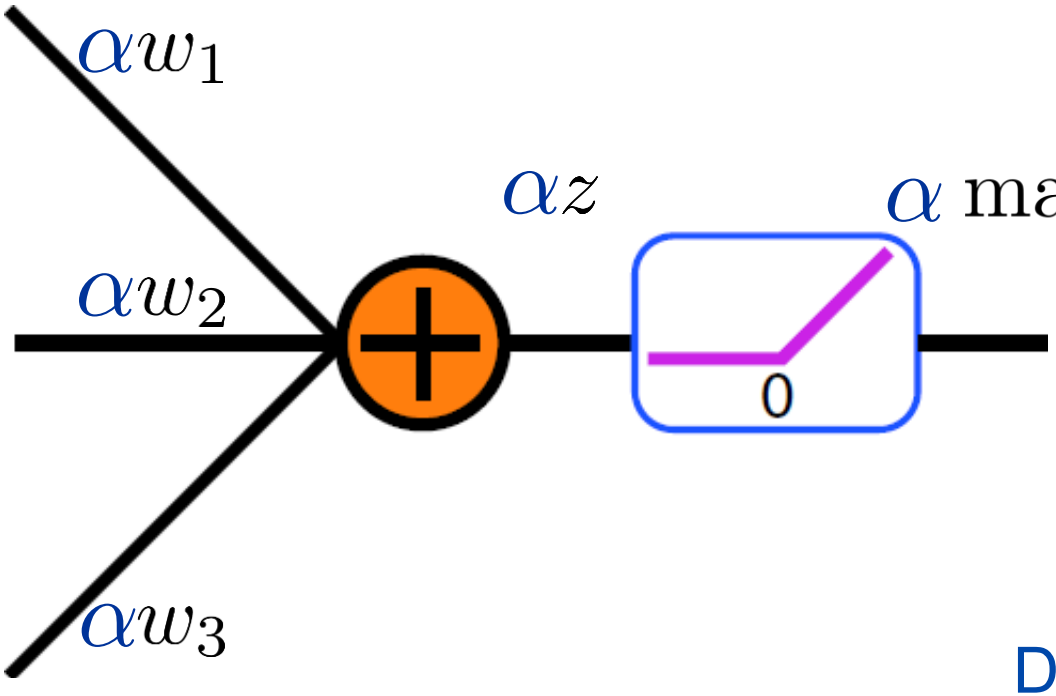
- **Example 1**: Rectified Linear Units (ReLU)



Does not change rectification

Examples of Positively Homogeneous Maps

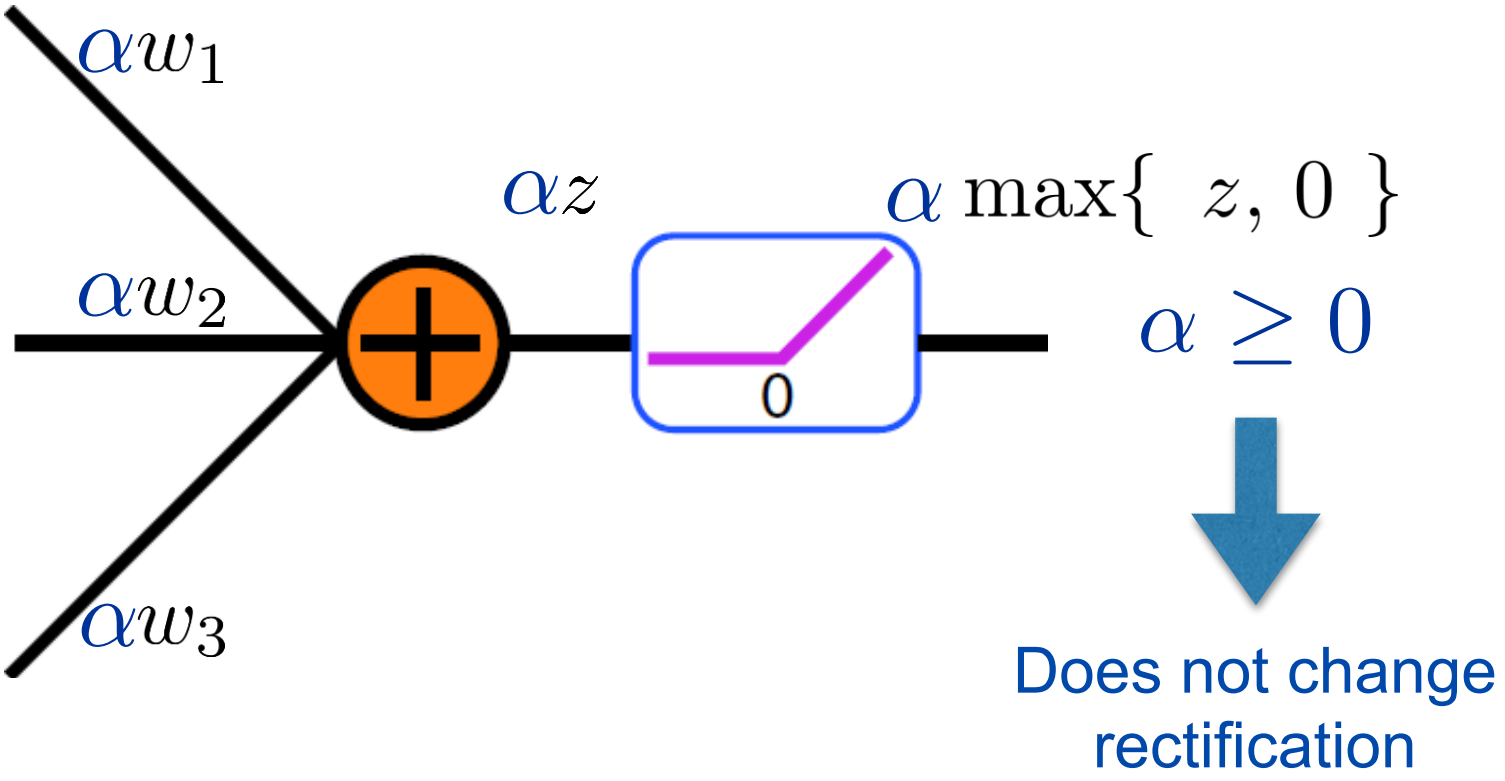
- **Example 1**: Rectified Linear Units (ReLU)



Does not change rectification

Examples of Positively Homogeneous Maps

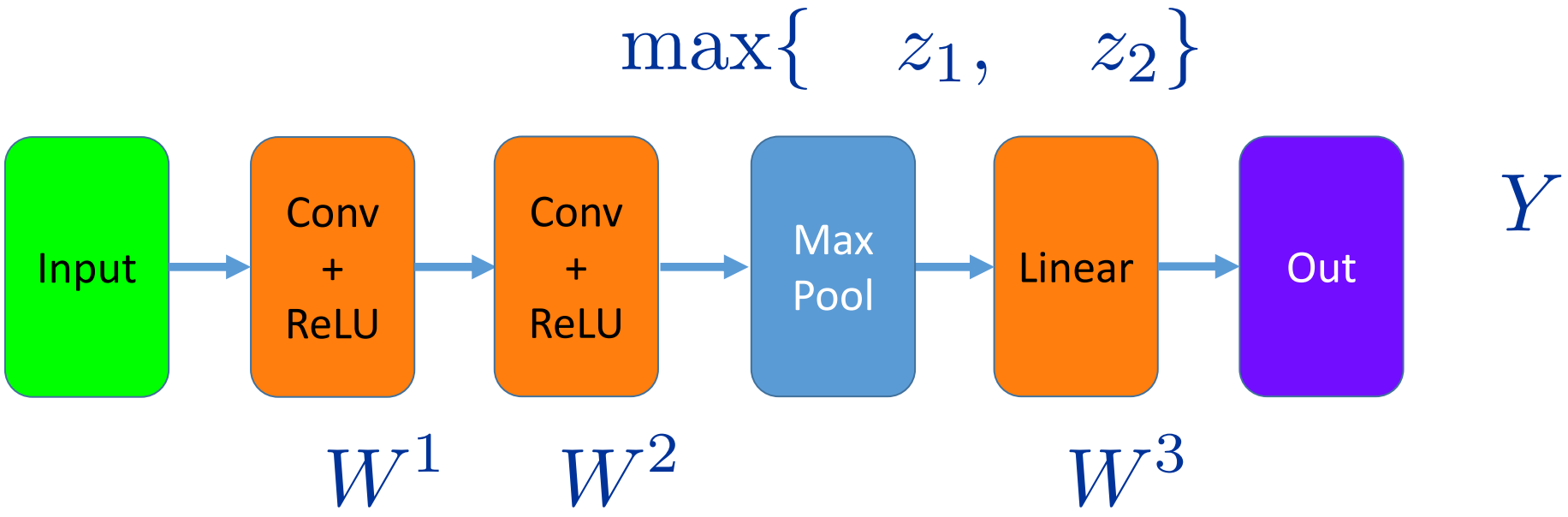
- **Example 1**: Rectified Linear Units (ReLU)



- Linear + ReLU layer is positively homogeneous of degree 1

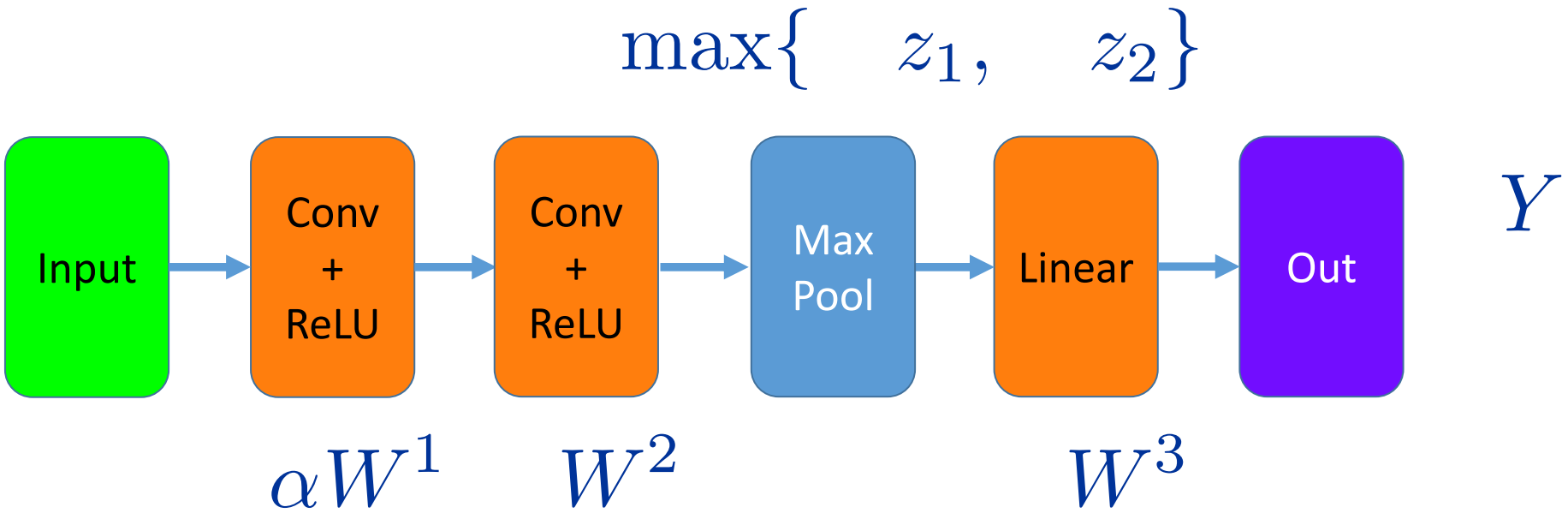
Examples of Positively Homogeneous Maps

- **Example 2**: Simple networks with convolutional layers, ReLU, max pooling and fully connected layers



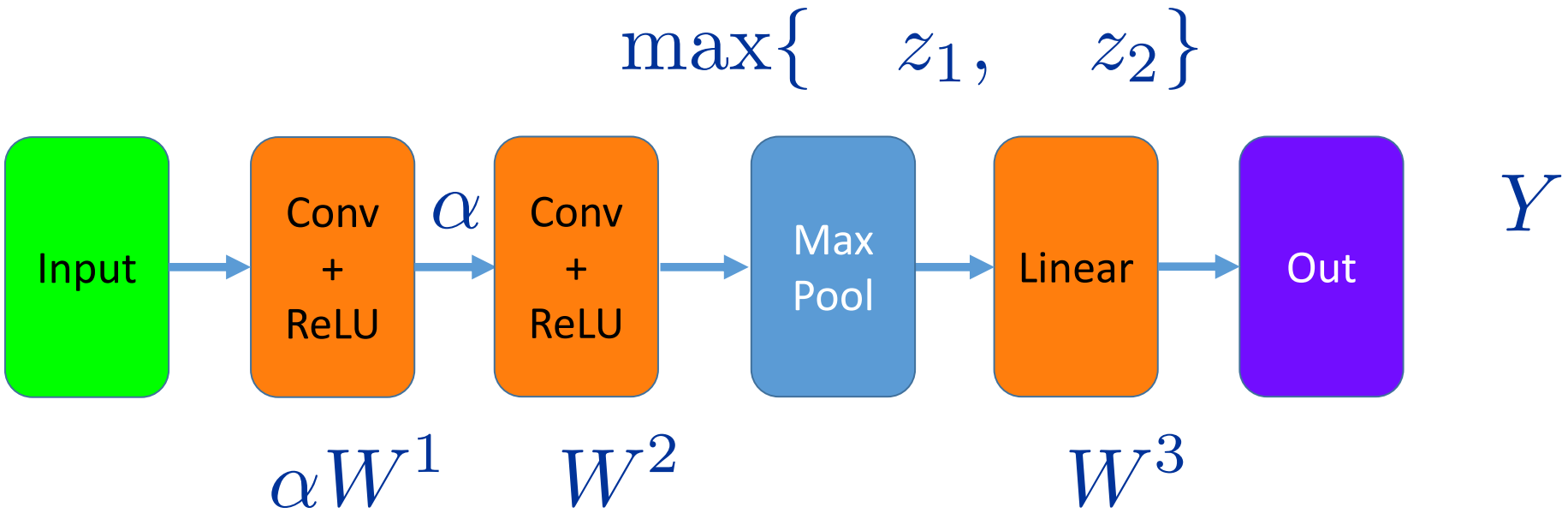
Examples of Positively Homogeneous Maps

- **Example 2**: Simple networks with convolutional layers, ReLU, max pooling and fully connected layers



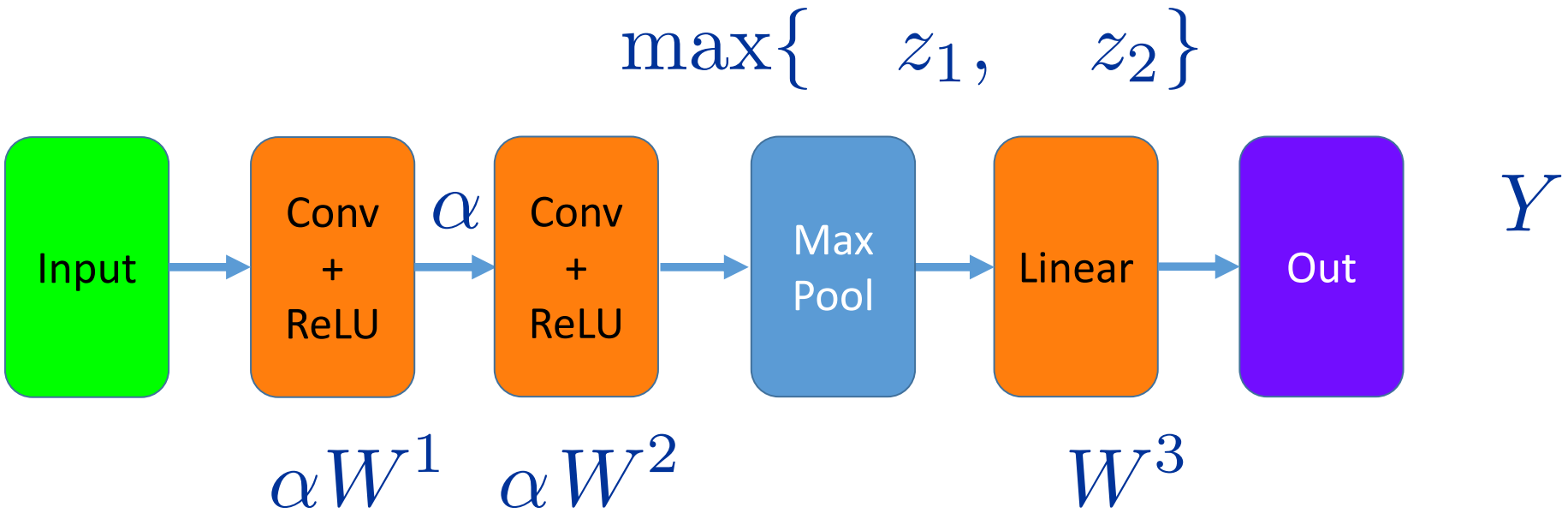
Examples of Positively Homogeneous Maps

- **Example 2**: Simple networks with convolutional layers, ReLU, max pooling and fully connected layers



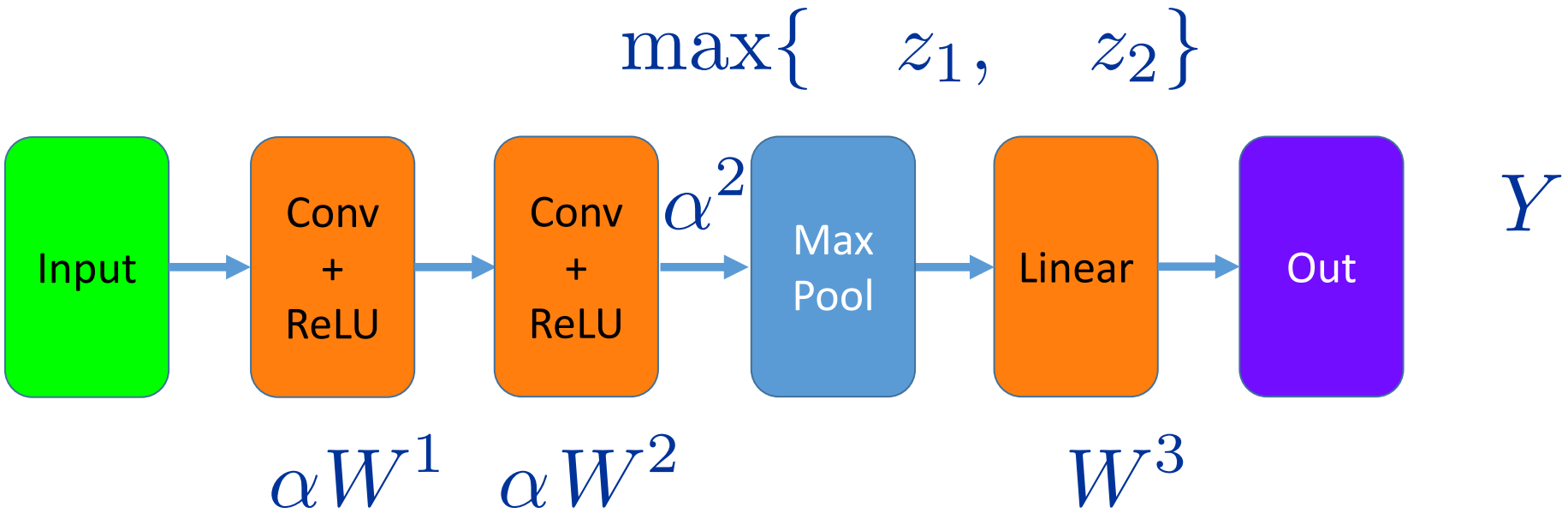
Examples of Positively Homogeneous Maps

- **Example 2**: Simple networks with convolutional layers, ReLU, max pooling and fully connected layers



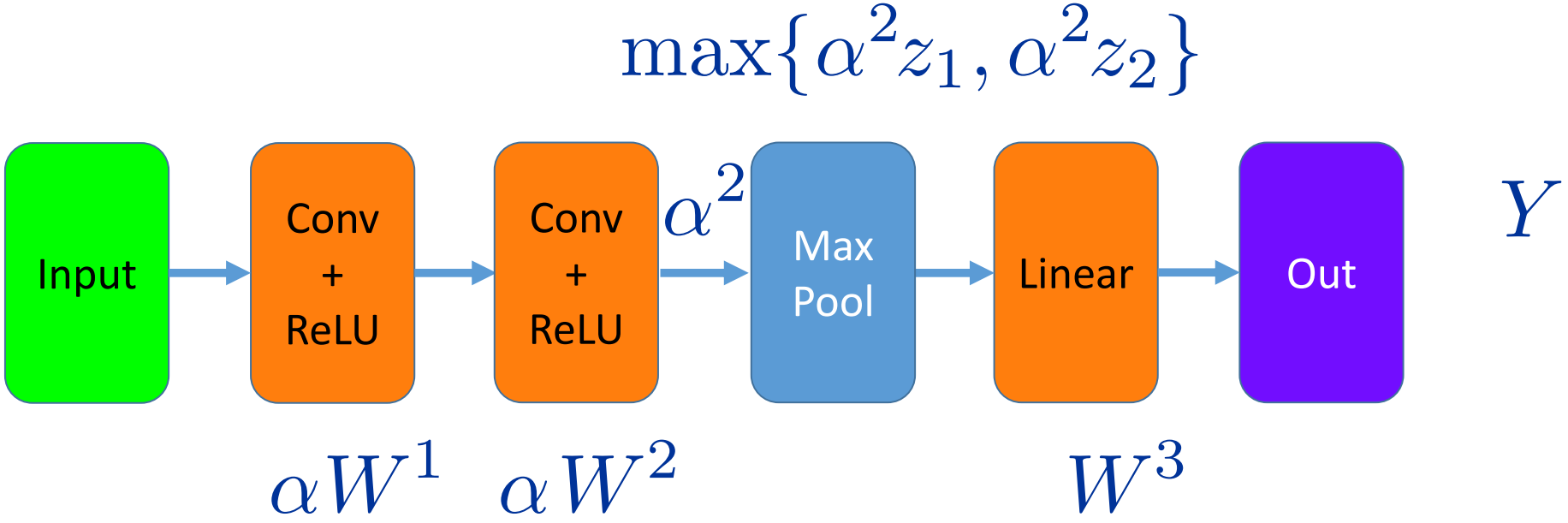
Examples of Positively Homogeneous Maps

- **Example 2**: Simple networks with convolutional layers, ReLU, max pooling and fully connected layers



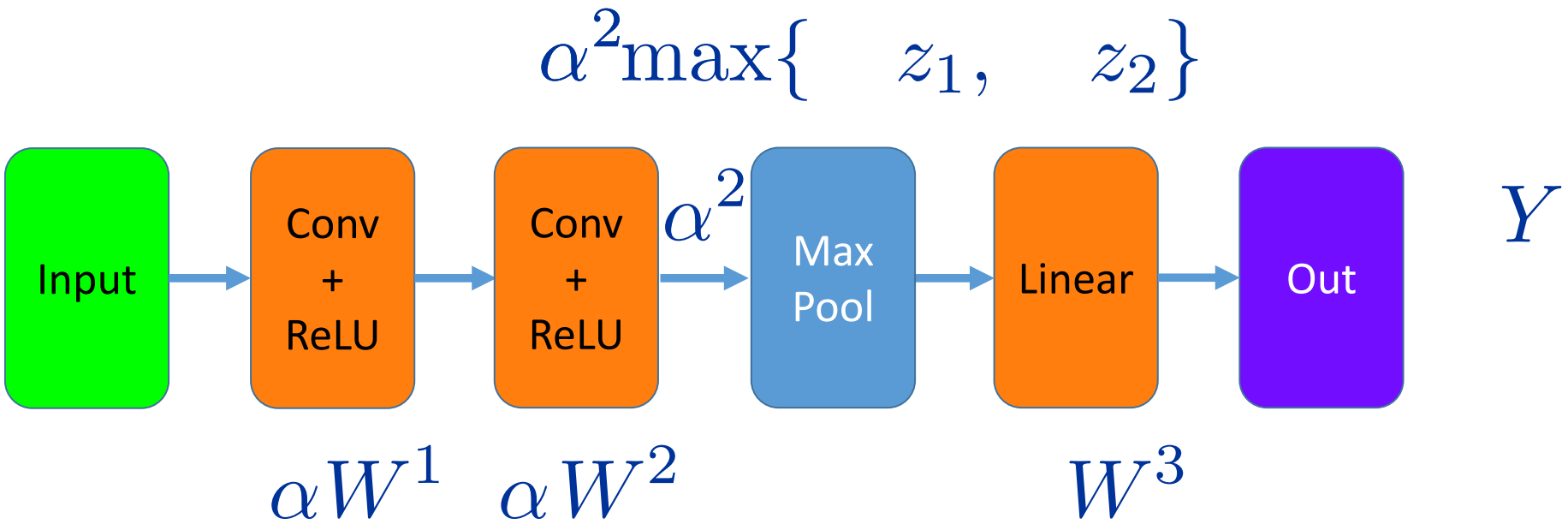
Examples of Positively Homogeneous Maps

- **Example 2**: Simple networks with convolutional layers, ReLU, max pooling and fully connected layers



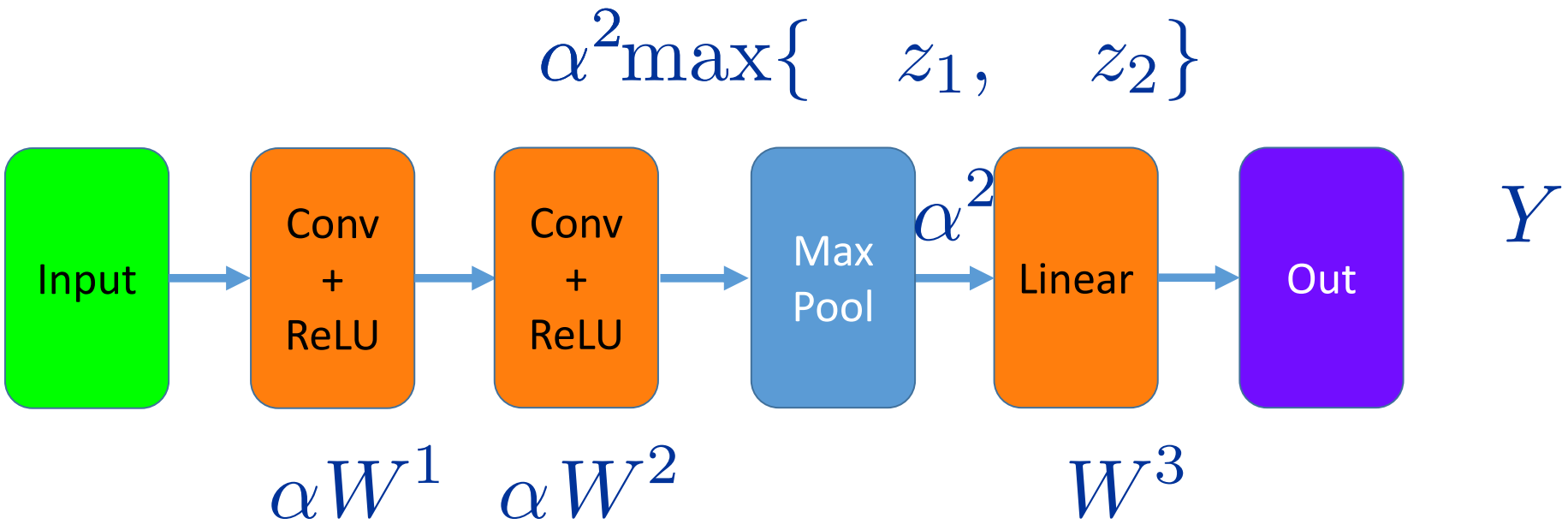
Examples of Positively Homogeneous Maps

- **Example 2**: Simple networks with convolutional layers, ReLU, max pooling and fully connected layers



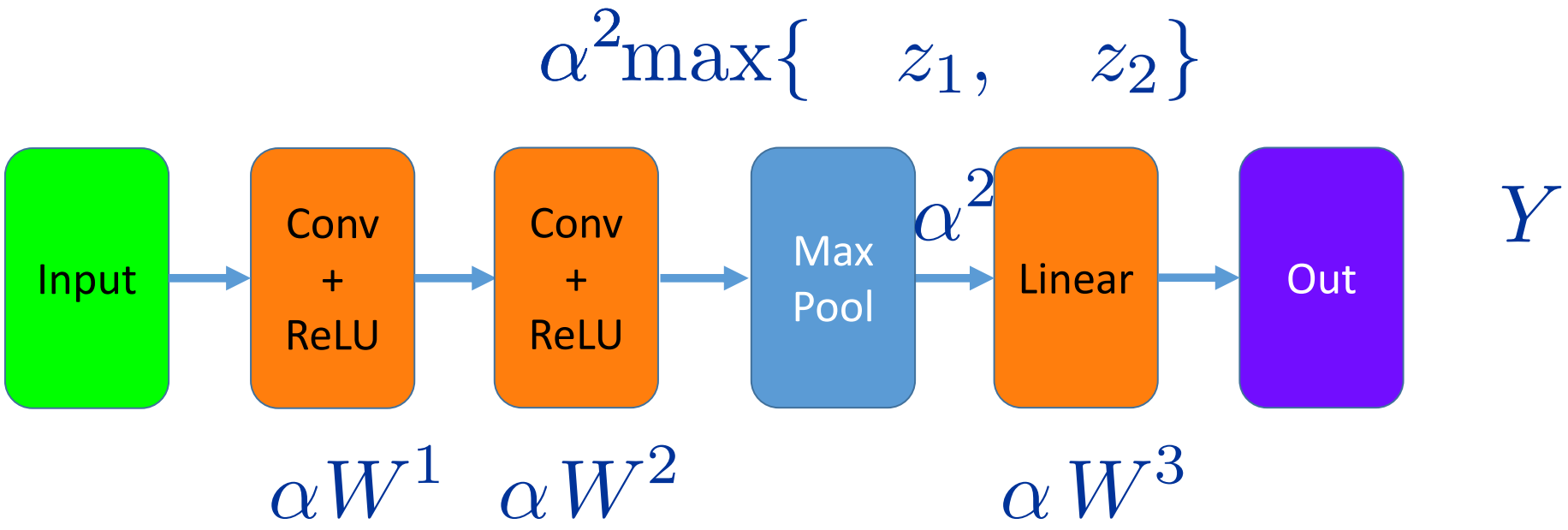
Examples of Positively Homogeneous Maps

- **Example 2**: Simple networks with convolutional layers, ReLU, max pooling and fully connected layers



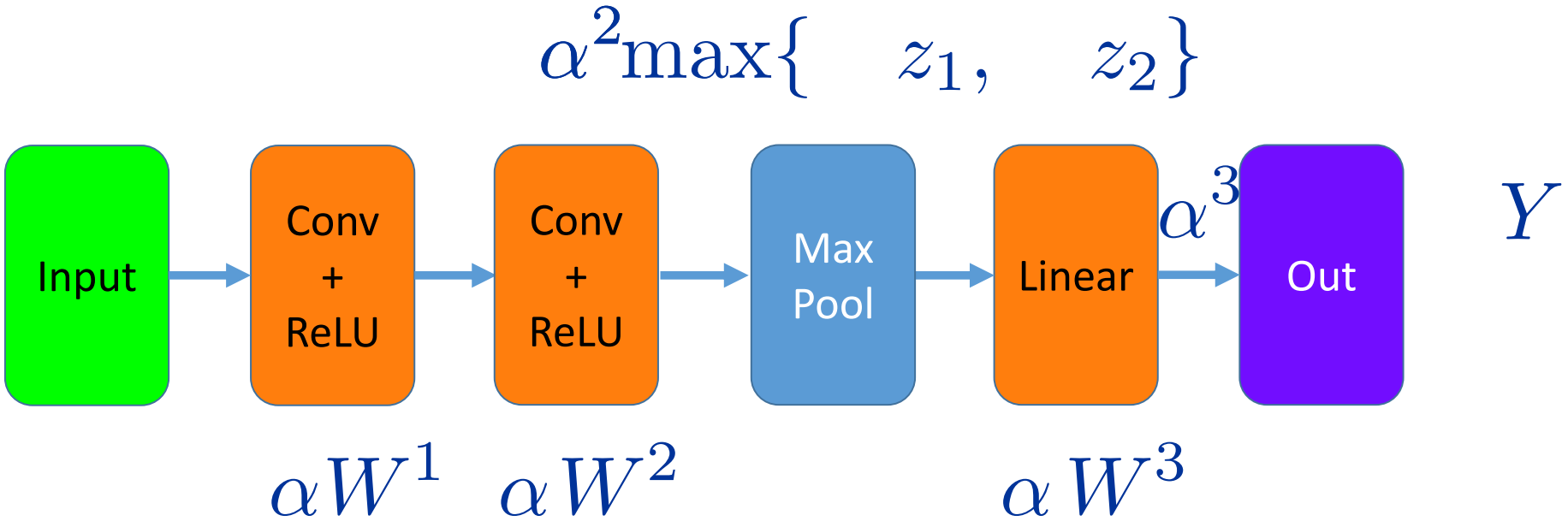
Examples of Positively Homogeneous Maps

- **Example 2**: Simple networks with convolutional layers, ReLU, max pooling and fully connected layers



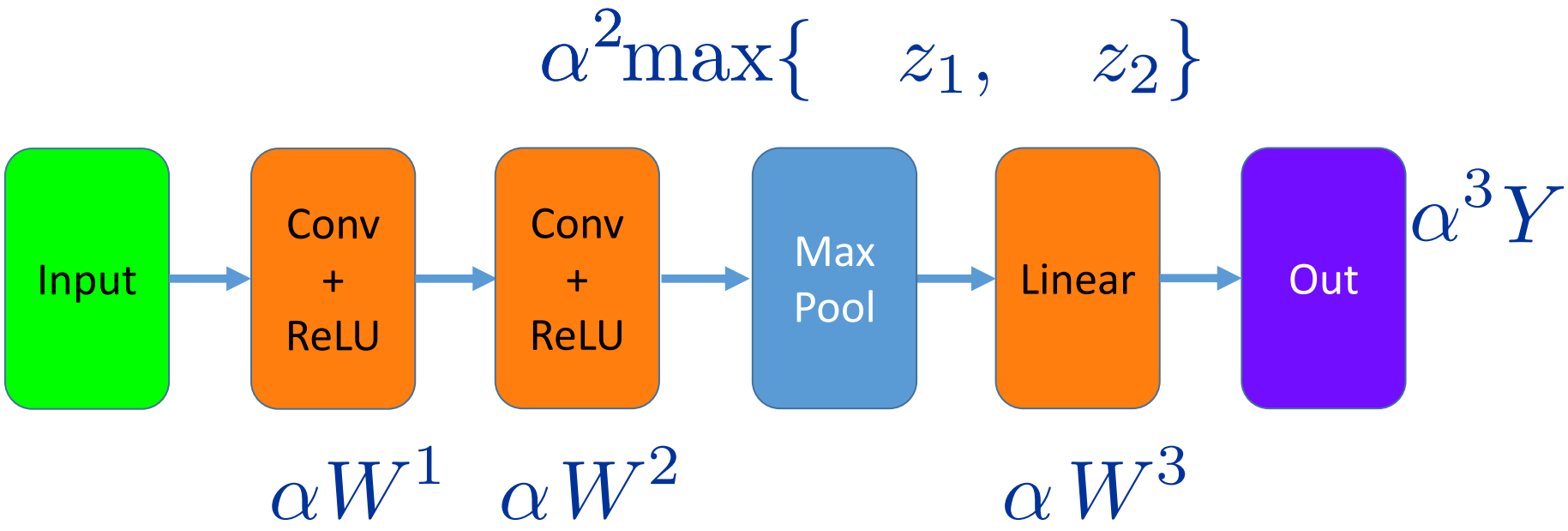
Examples of Positively Homogeneous Maps

- **Example 2**: Simple networks with convolutional layers, ReLU, max pooling and fully connected layers



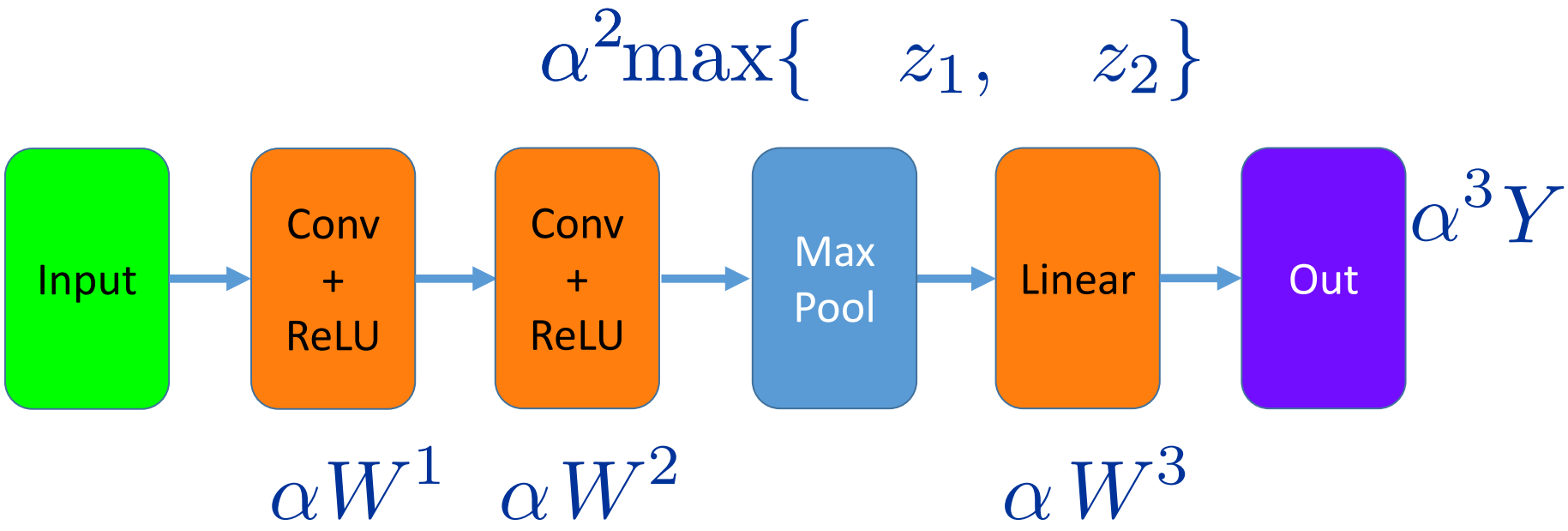
Examples of Positively Homogeneous Maps

- **Example 2**: Simple networks with convolutional layers, ReLU, max pooling and fully connected layers



Examples of Positively Homogeneous Maps

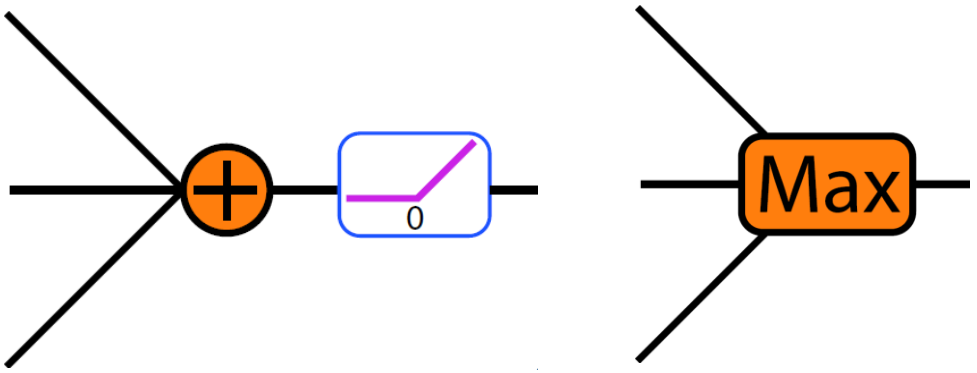
- **Example 2**: Simple networks with convolutional layers, ReLU, max pooling and fully connected layers



- Typically each weight layer increases degree of homogeneity by 1

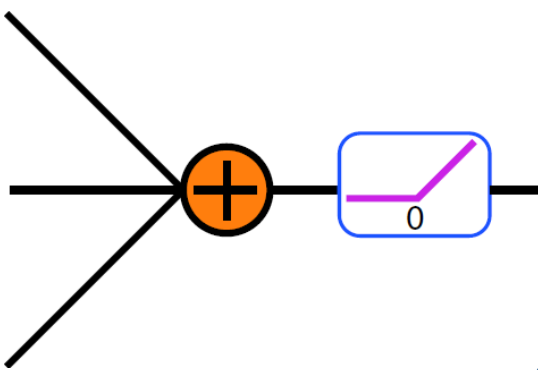
Examples of Positively Homogeneous Maps

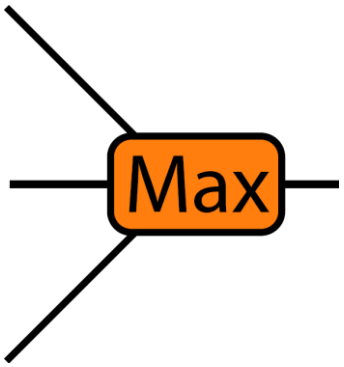
- Some Common Positively Homogeneous Layers
  - Fully Connected + ReLU
  - Convolution + ReLU
  - Max Pooling
  - Linear Layers
  - Mean Pooling
  - Max Out
  - Many possibilities...

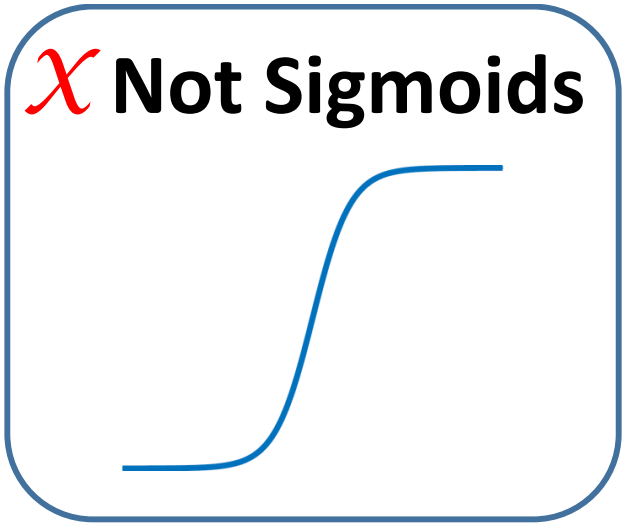


Examples of Positively Homogeneous Maps

- Some Common Positively Homogeneous Layers
  - Fully Connected + ReLU
  - Convolution + ReLU
  - Max Pooling
  - Linear Layers
  - Mean Pooling
  - Max Out
  - Many possibilities...

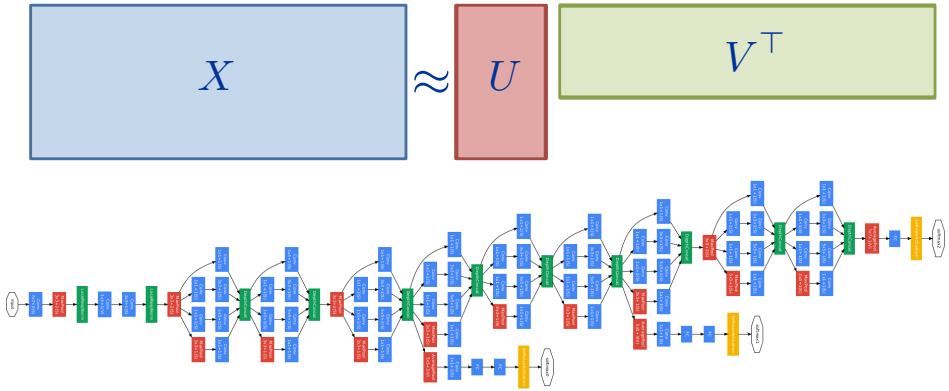






Outline

- **Architecture properties that facilitate optimization**
  - Positive homogeneity
  - Parallel subnetwork structure
- **Regularization properties that facilitate optimization**
  - Positive homogeneity
  - Adapt network structure to the data
- **Theoretical guarantees**
  - Sufficient conditions for global optimality
  - Local descent can reach global minimizers



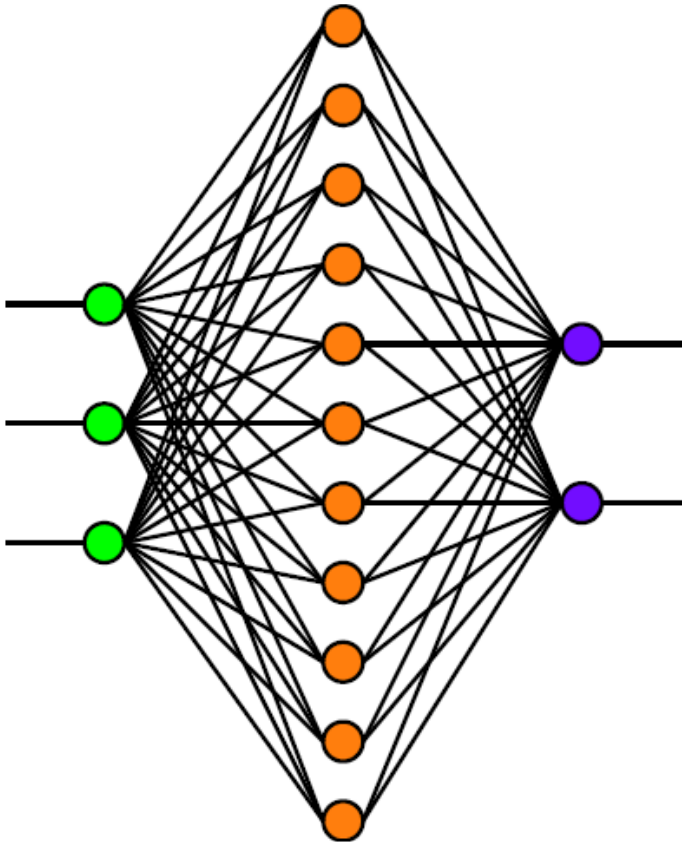
[1] Haeffele, Young, Vidal. Structured Low-Rank Matrix Factorization: Optimality, Algorithm, and Applications to Image Processing, ICML '14
[2] Haeffele, Vidal. Global Optimality in Tensor Factorization, Deep Learning and Beyond, arXiv, '15
[3] Haeffele, Vidal. Global optimality in neural network training. CVPR 2017.

Key Property #2: Parallel Subnetworks

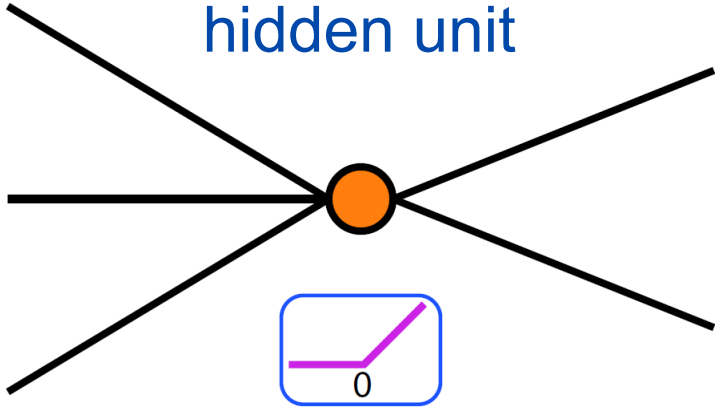
- Subnetworks with identical structure connected in parallel

Key Property #2: Parallel Subnetworks

- Subnetworks with identical structure connected in parallel
- **Simple example**: network with a single hidden layer

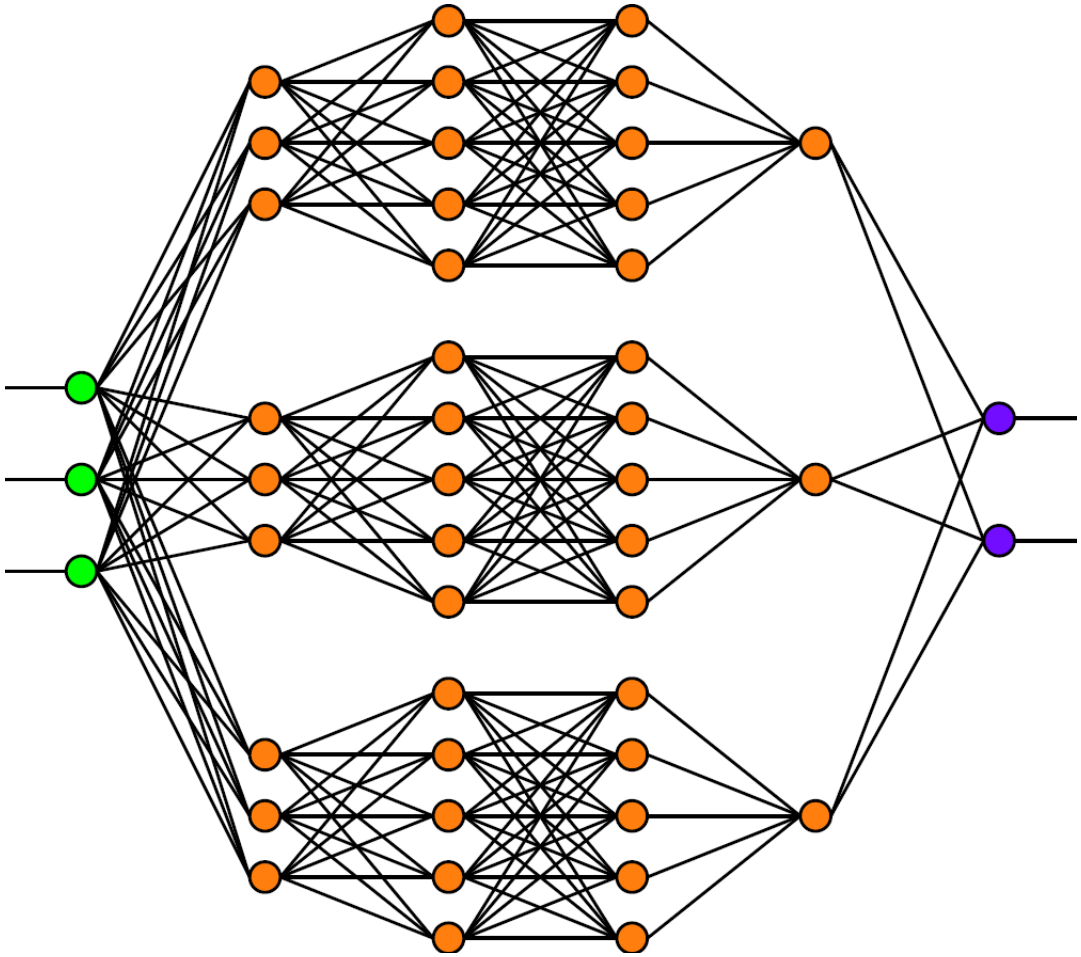


Subnetwork:
one ReLU
hidden unit

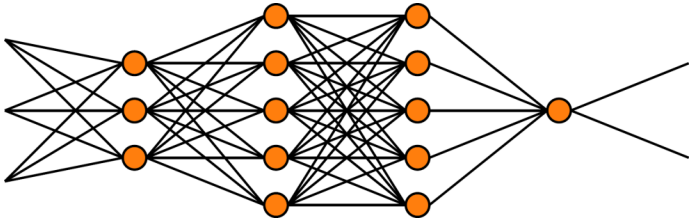


Key Property #2: Parallel Subnetworks

- Any positively homogeneous sub-network can be used

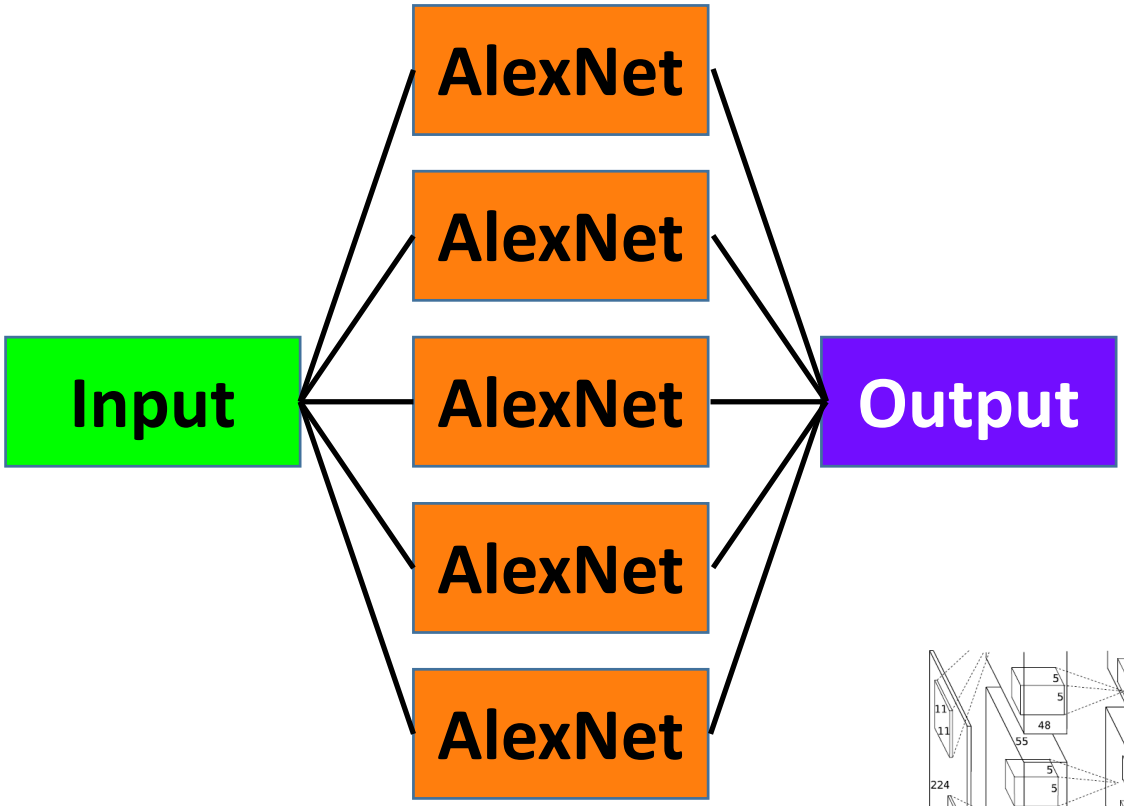


Subnetwork: multiple ReLU layers

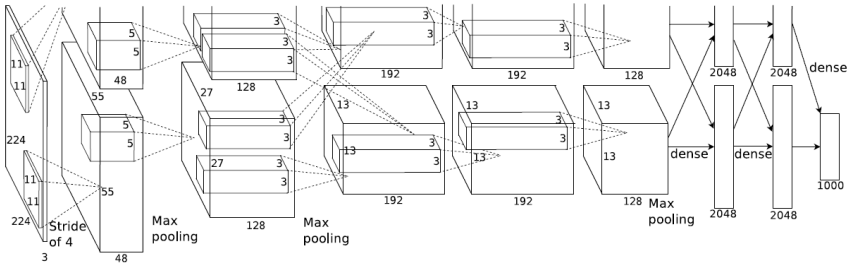


Key Property #2: Parallel Subnetworks

- **Example**: Parallel AlexNets [1]



Subnetwork:
AlexNet



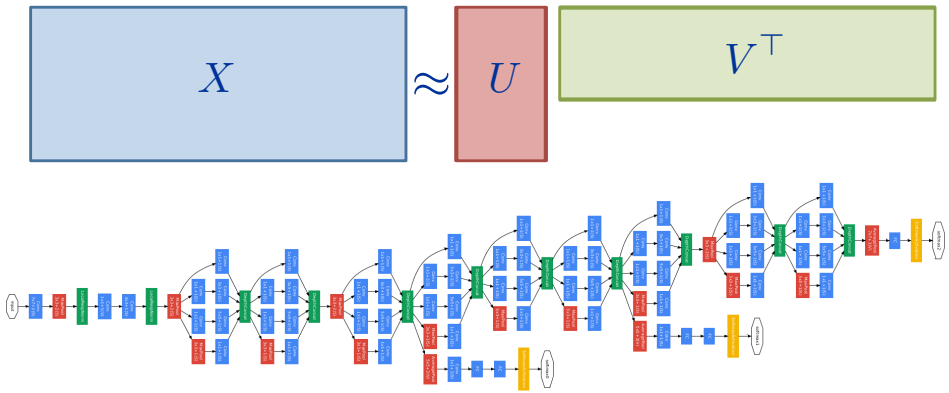
[1] Krizhevsky, Sutskever, and Hinton. "Imagenet classification with deep convolutional neural networks." NIPS, 2012

Outline

- **Architecture properties that facilitate optimization**
  - Positive homogeneity
  - Parallel subnetwork structure

- **Regularization properties that facilitate optimization**
  - Positive homogeneity
  - Adapt network structure to the data

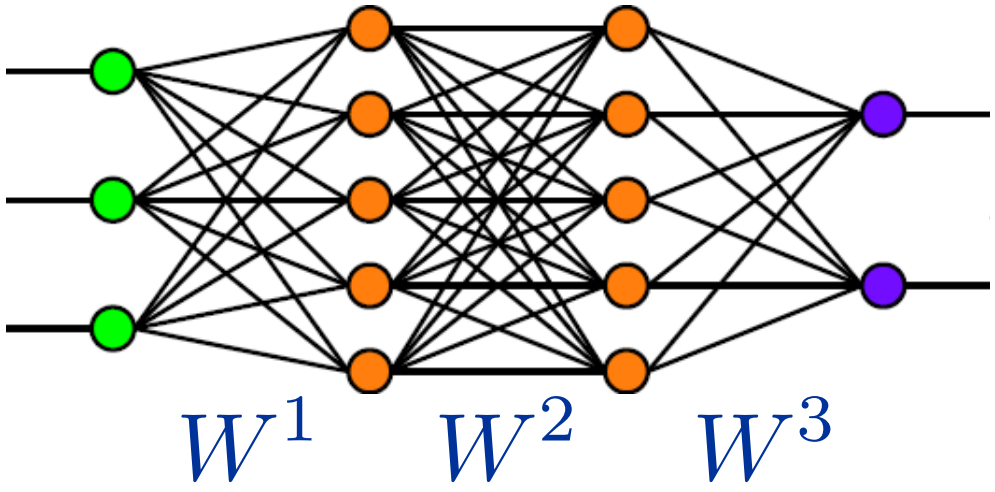
- **Theoretical guarantees**
  - Sufficient conditions for global optimality
  - Local descent can reach global minimizers



[1] Haeffele, Young, Vidal. Structured Low-Rank Matrix Factorization: Optimality, Algorithm, and Applications to Image Processing, ICML '14
[2] Haeffele, Vidal. Global Optimality in Tensor Factorization, Deep Learning and Beyond, arXiv, '15
[3] Haeffele, Vidal. Global optimality in neural network training. CVPR 2017.

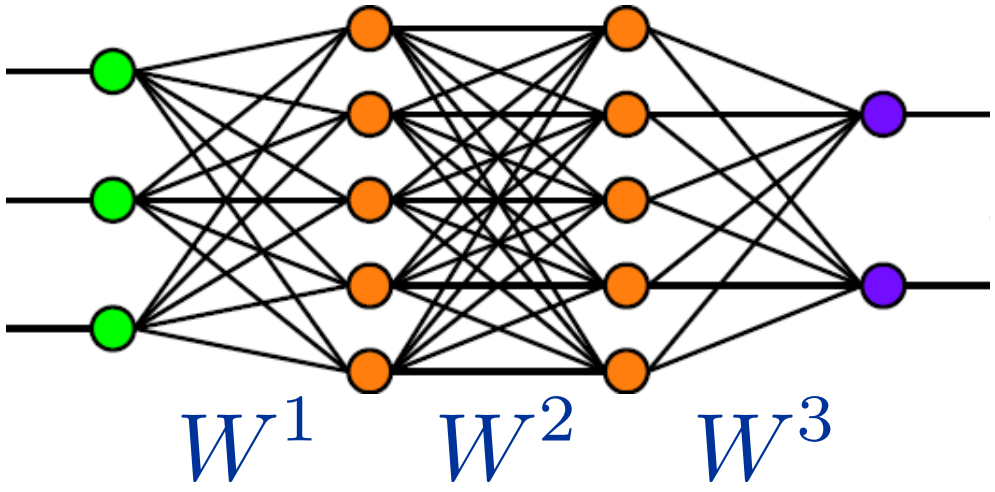
Basic Regularization: Weight Decay

$$\Theta(W^1, W^2, W^3) = \|W^1\|_F^2 + \|W^2\|_F^2 + \|W^3\|_F^2$$



Basic Regularization: Weight Decay

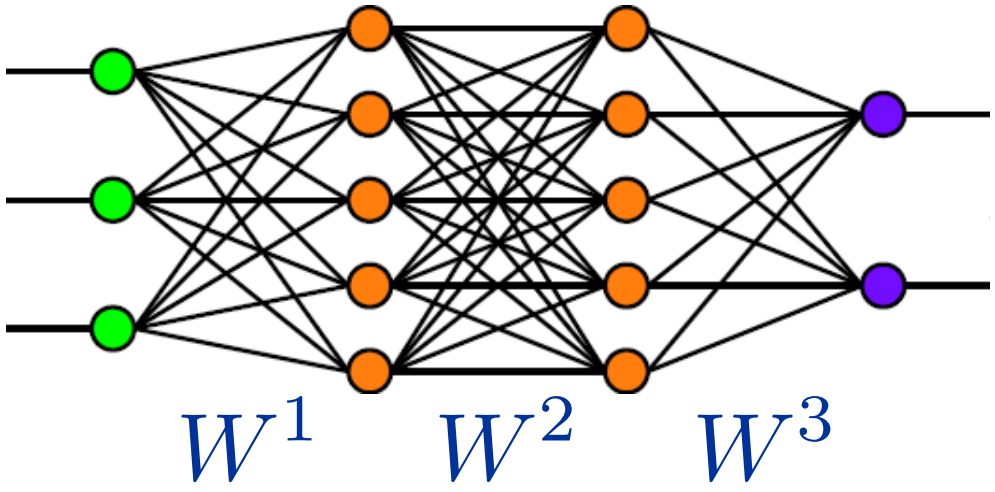
$$\Theta(W^1, W^2, W^3) = \|W^1\|_F^2 + \|W^2\|_F^2 + \|W^3\|_F^2$$



$$\Theta(\alpha W^1, \alpha W^2, \alpha W^3) = \alpha^2 \Theta(W^1, W^2, W^3)$$

Basic Regularization: Weight Decay

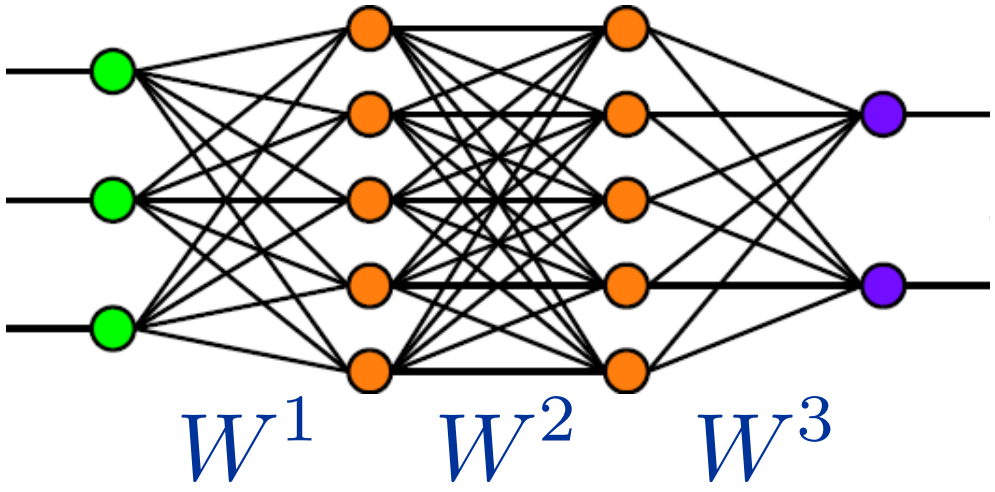
$$\Theta(W^1, W^2, W^3) = \|W^1\|_F^2 + \|W^2\|_F^2 + \|W^3\|_F^2$$



$$\Theta(\alpha W^1, \alpha W^2, \alpha W^3) = \alpha^2 \Theta(W^1, W^2, W^3)$$
$$\Phi(\alpha W^1, \alpha W^2, \alpha W^3) = \alpha^3 \Phi(W^1, W^2, W^3)$$

Basic Regularization: Weight Decay

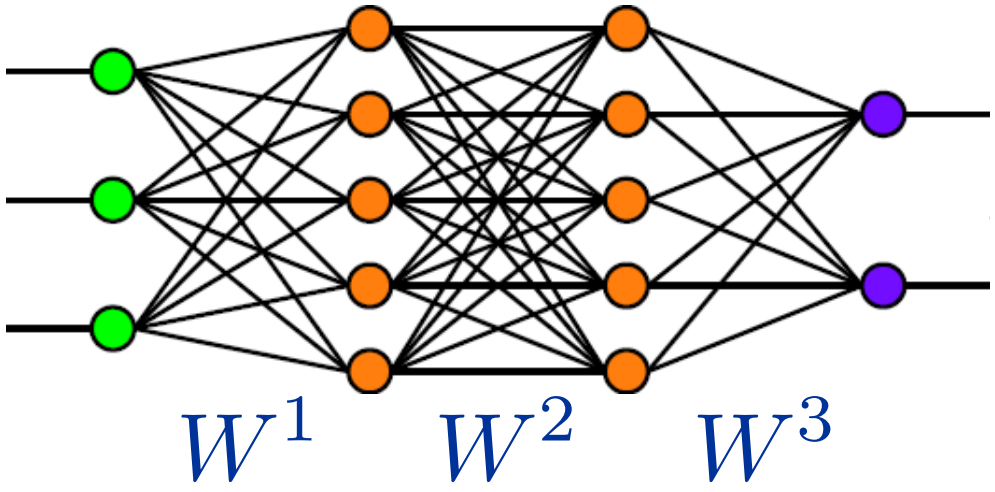
$$\Theta(W^1, W^2, W^3) = \|W^1\|_F^2 + \|W^2\|_F^2 + \|W^3\|_F^2$$



$$\Theta(\alpha W^1, \alpha W^2, \alpha W^3) = \alpha^2 \Theta(W^1, W^2, W^3)$$
$$\Phi(\alpha W^1, \alpha W^2, \alpha W^3) = \alpha^3 \Phi(W^1, W^2, W^3)$$

Basic Regularization: Weight Decay

$$\Theta(W^1, W^2, W^3) = \|W^1\|_F^2 + \|W^2\|_F^2 + \|W^3\|_F^2$$

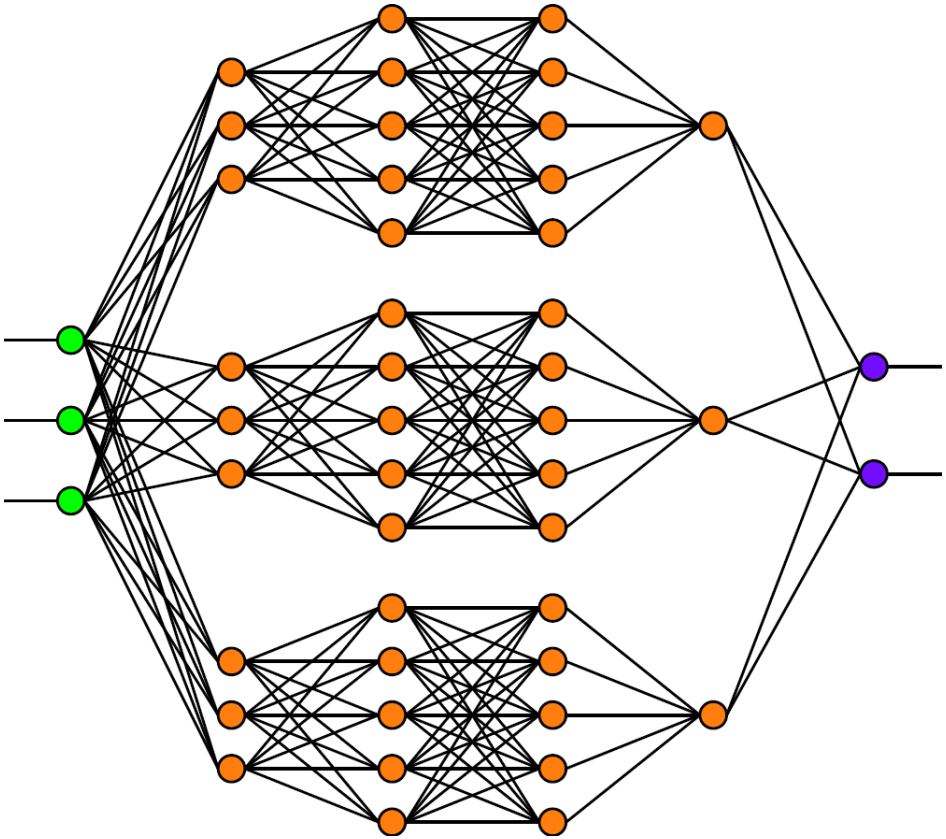


$$\Theta(\alpha W^1, \alpha W^2, \alpha W^3) = \alpha^2 \Theta(W^1, W^2, W^3)$$
$$\Phi(\alpha W^1, \alpha W^2, \alpha W^3) = \alpha^3 \Phi(W^1, W^2, W^3)$$

- **Proposition** non-matching degrees => spurious local minima

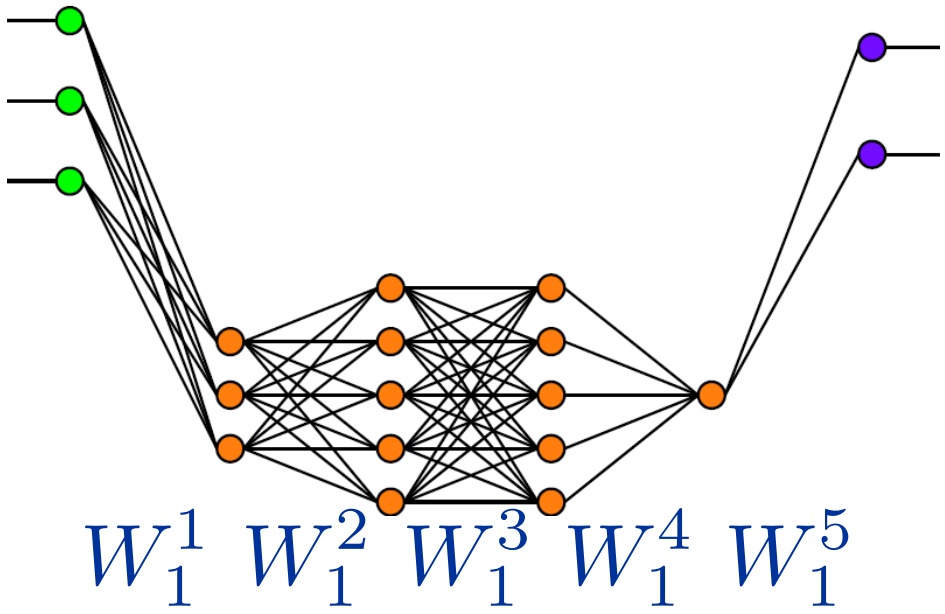
Regularizer Adapted to Network Size

- Start with a positively homogeneous network with parallel structure



Regularizer Adapted to Network Size

- Take the weights of one subnetwork and define a regularizer as $\theta(W_1^1, W_1^2, W_1^3, W_1^4, W_1^5)$ with the properties:

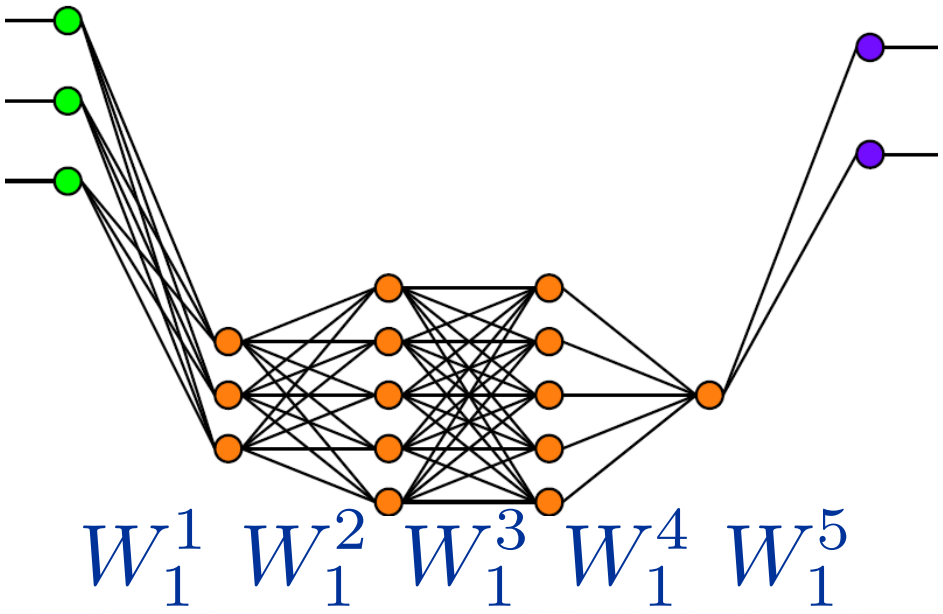


Regularizer Adapted to Network Size

- Take the weights of one subnetwork and define a regularizer as $\theta(W_1^1, W_1^2, W_1^3, W_1^4, W_1^5)$ with the properties:
  - Positive semi-definite
  - Positively homogeneous with the same degree as network

$$\Phi(\alpha W) = \alpha^p \Phi(W)$$

$$\theta(\alpha W) = \alpha^p \theta(W)$$

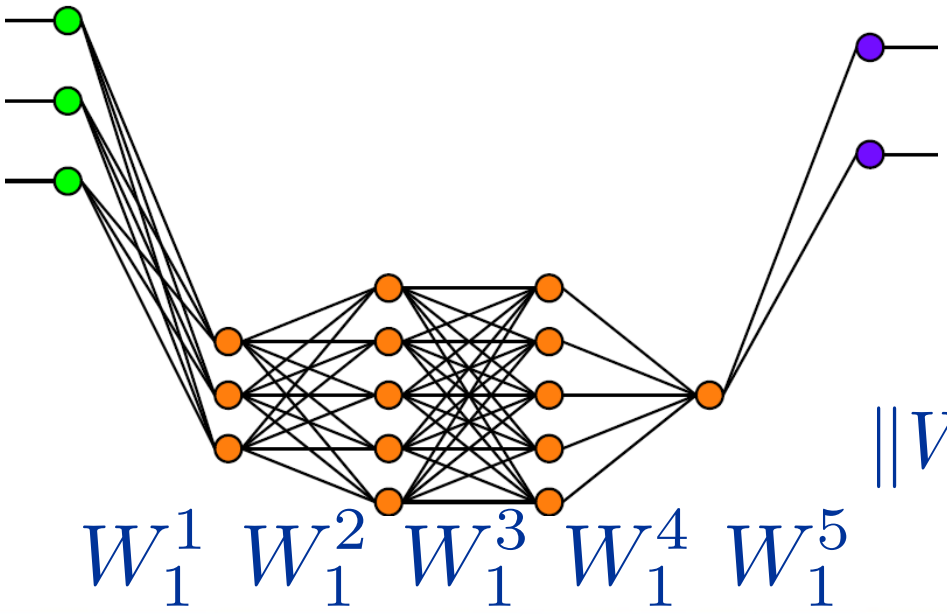


Regularizer Adapted to Network Size

- Take the weights of one subnetwork and define a regularizer as $\theta(W_1^1, W_1^2, W_1^3, W_1^4, W_1^5)$ with the properties:
  - Positive semi-definite
  - Positively homogeneous with the same degree as network

$$\Phi(\alpha W) = \alpha^p \Phi(W)$$

$$\theta(\alpha W) = \alpha^p \theta(W)$$

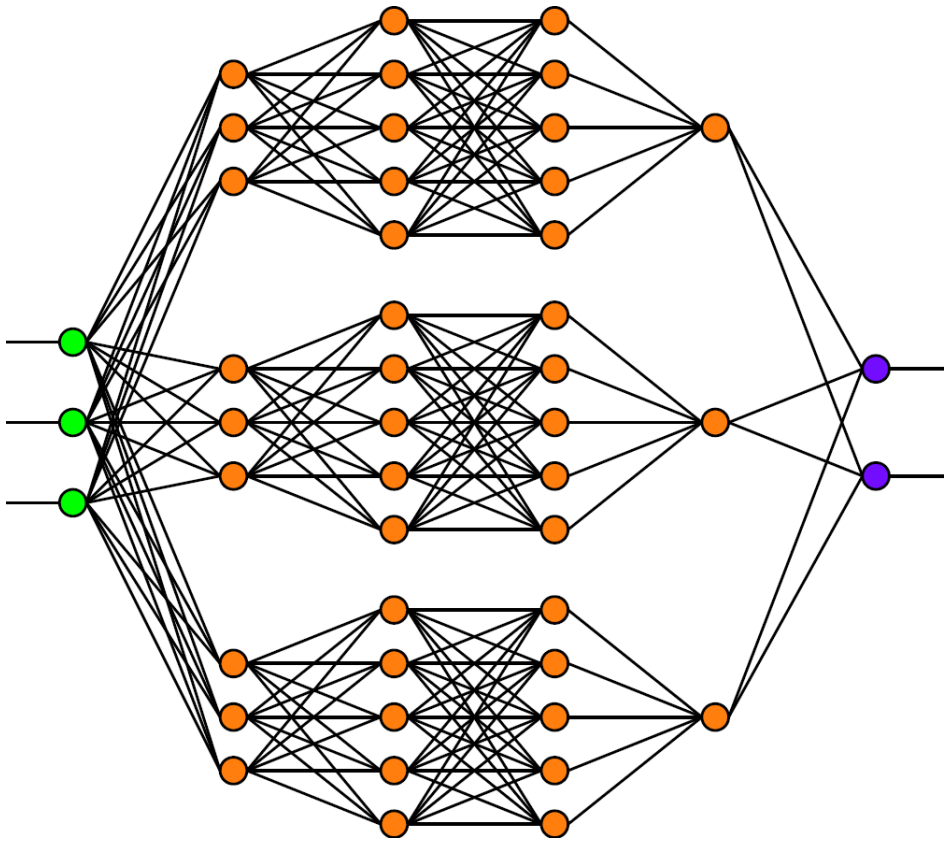


- **Example:** product of norms

$$\|W_1^1\| \|W_1^2\| \|W_1^3\| \|W_1^4\| \|W_1^5\|$$

Regularizer Adapted to Network Size

- Sum over all subnetworks

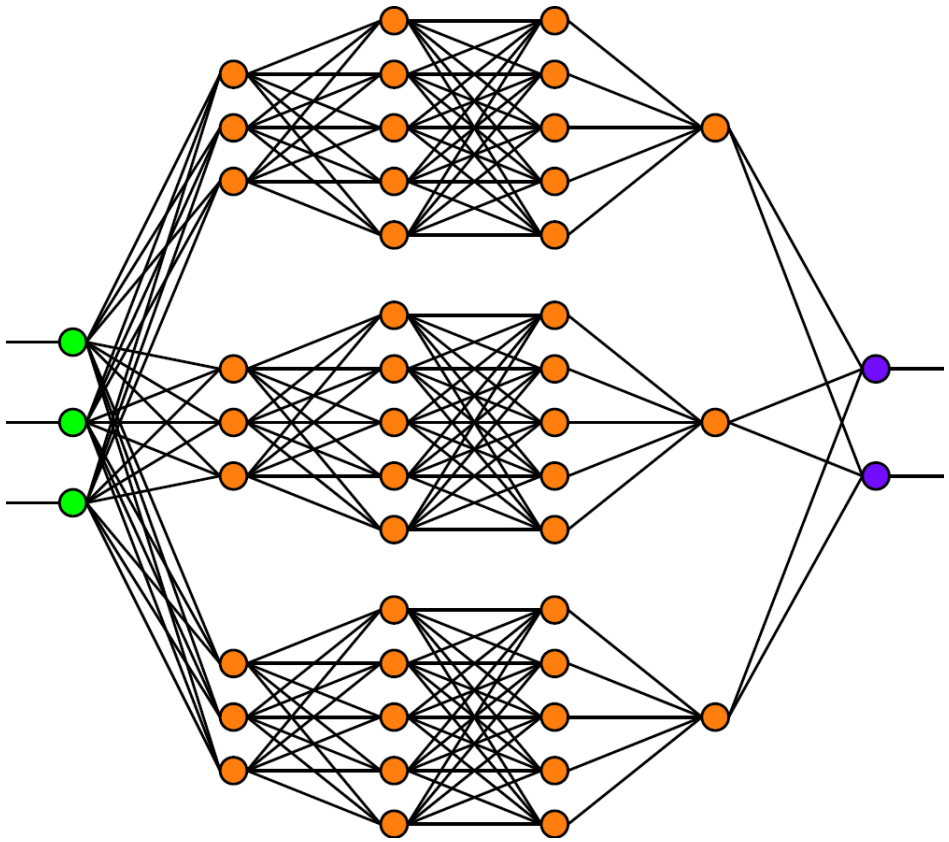


$$\Theta(W) = \sum_{i=1}^{r} \theta(W^i)$$

$$r = \#\ \text{subnets}$$

Regularizer Adapted to Network Size

- Sum over all subnetworks



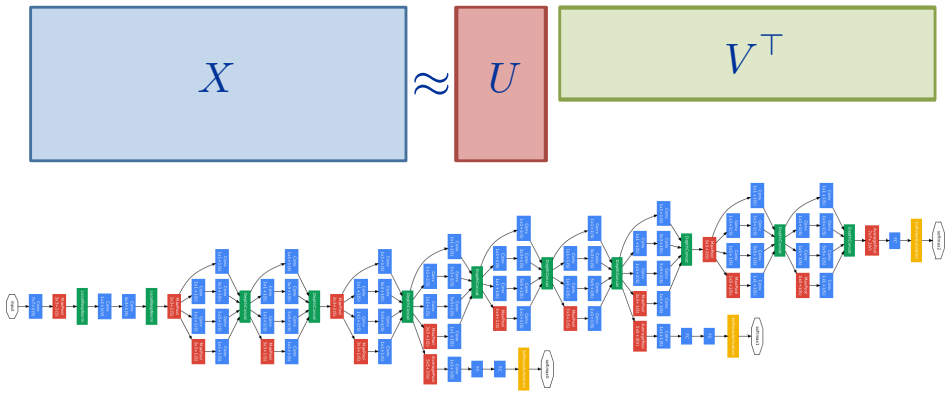
$$\Theta(W) = \sum_{i=1}^{r} \theta(W^i)$$

$$r = \#\ \text{subnets}$$

- Allow r to vary
- Adding a subnetwork is penalized by an additional term in the sum
- Regularizer constraints number of subnetworks

Outline

- **Architecture properties that facilitate optimization**
  - Positive homogeneity
  - Parallel subnetwork structure
- **Regularization properties that facilitate optimization**
  - Positive homogeneity
  - Adapt network structure to the data
- **Theoretical guarantees**
  - Sufficient conditions for global optimality
  - Local descent can reach global minimizers



[1] Haeffele, Young, Vidal. Structured Low-Rank Matrix Factorization: Optimality, Algorithm, and Applications to Image Processing, ICML '14
[2] Haeffele, Vidal. Global Optimality in Tensor Factorization, Deep Learning and Beyond, arXiv, '15
[3] Haeffele, Vidal. Global optimality in neural network training. CVPR 2017.

Main Results: Matrix Factorization

- **Convex formulations**:

$$\min_{X} \ell(Y, X) + \lambda \|X\|_*$$

Factorized formulations

$$\min_{U,V} \ell(Y, UV^\top) + \lambda \Theta(U, V)$$

[1] Burer, Monteiro. Local minima and convergence in low- rank semidefinite programming. Math. Prog., 2005.
[2] Cabral, De la Torre, Costeira, Bernardino, “Unifying nuclear norm and bilinear factorization approaches for low-rank matrix decomposition,” CVPR, 2013, pp. 2488–2495.
[3] Bach, Mairal, Ponce, Convex sparse matrix factorizations, arXiv 2008.
[4] Bach. Convex relaxations of structured matrix factorizations, arXiv 2013.

Main Results: Matrix Factorization

- **Convex formulations**:

$$\min_X \ell(Y, X) + \lambda \|X\|_*$$

Factorized formulations

$$\min_{U,V} \ell(Y, UV^\top) + \lambda \Theta(U, V)$$

- Variational form of the nuclear norm [1,2]

$$\|X\|_* = \min_{U,V} \sum_{i=1}^{r} \|U_i\|_2 \|V_i\|_2 \quad \text{s.t.} \quad UV^\top = X$$

$$\|X\|_* = \sum \sigma_i(X)$$

[1] Burer, Monteiro. Local minima and convergence in low- rank semidefinite programming. Math. Prog., 2005.
[2] Cabral, De la Torre, Costeira, Bernardino, "Unifying nuclear norm and bilinear factorization approaches for low-rank matrix decomposition," CVPR, 2013, pp. 2488–2495.
[3] Bach, Mairal, Ponce, Convex sparse matrix factorizations, arXiv 2008.
[4] Bach. Convex relaxations of structured matrix factorizations, arXiv 2013.

Main Results: Matrix Factorization

- **Convex formulations**:

$$\min_X \ell(Y, X) + \lambda \|X\|_*$$

Factorized formulations

$$\min_{U,V} \ell(Y, UV^\top) + \lambda \Theta(U, V)$$

- Variational form of the nuclear norm [1,2]

$$\|X\|_* = \min_{U,V} \sum_{i=1}^{r} \|U_i\|_2 \|V_i\|_2 \quad \text{s.t.} \quad UV^\top = X$$

$$\|X\|_* = \sum \sigma_i(X)$$

[1] Burer, Monteiro. Local minima and convergence in low- rank semidefinite programming. Math. Prog., 2005.
[2] Cabral, De la Torre, Costeira, Bernardino, "Unifying nuclear norm and bilinear factorization approaches for low-rank matrix decomposition," CVPR, 2013, pp. 2488–2495.
[3] Bach, Mairal, Ponce, Convex sparse matrix factorizations, arXiv 2008.
[4] Bach. Convex relaxations of structured matrix factorizations, arXiv 2013.

Main Results: Matrix Factorization

- **Convex formulations**:

$$\min_X \ell(Y, X) + \lambda \|X\|_*$$

Factorized formulations

$$\min_{U,V} \ell(Y, UV^\top) + \lambda \Theta(U, V)$$

- Variational form of the nuclear norm [1,2]

$$\|X\|_* = \min_{U,V} \sum_{i=1}^{r} \|U_i\|_2 \|V_i\|_2 \quad \text{s.t.} \quad UV^\top = X$$

- A natural generalization is the projective tensor norm [3,4]

$$\|X\|_{u,v} = \min_{U,V} \sum_{i=1}^{r} \|U_i\|_u \|V_i\|_v \quad \text{s.t.} \quad UV^\top = X$$

[1] Burer, Monteiro. Local minima and convergence in low- rank semidefinite programming. Math. Prog., 2005.
[2] Cabral, De la Torre, Costeira, Bernardino, "Unifying nuclear norm and bilinear factorization approaches for low-rank matrix decomposition," CVPR, 2013, pp. 2488–2495.
[3] Bach, Mairal, Ponce, Convex sparse matrix factorizations, arXiv 2008.
[4] Bach. Convex relaxations of structured matrix factorizations, arXiv 2013.

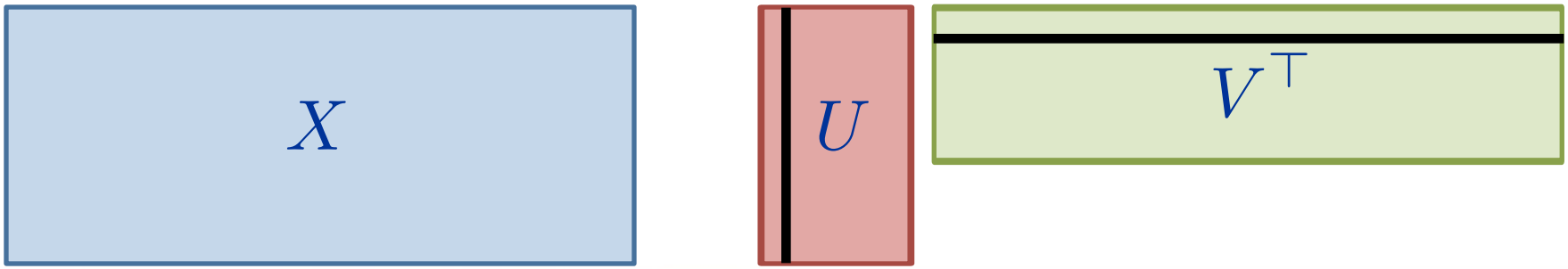
Main Results: Matrix Factorization

- **Theorem 1**: Assume ℓ is convex and once differentiable in X. A local minimizer (U, V) of the non-convex factorized problem

$$\min_{U,V} \ell(Y, UV^\top) + \lambda \sum_{i=1}^{r} \|U_i\|_u \|V_i\|_v$$

such that for some i $U_i = V_i = 0$, is a global minimizer. Moreover, $UV^\top$ is a global minimizer of the convex problem

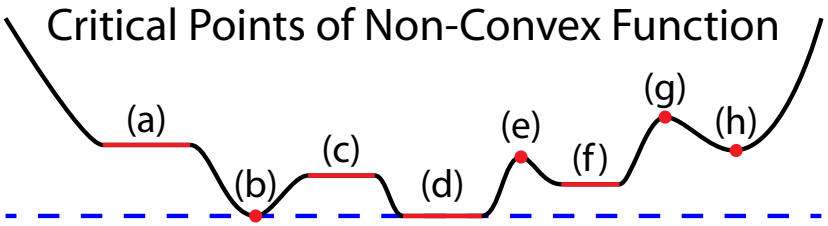
$$\min_{X} \ell(Y, X) + \lambda \|X\|_{u,v}$$

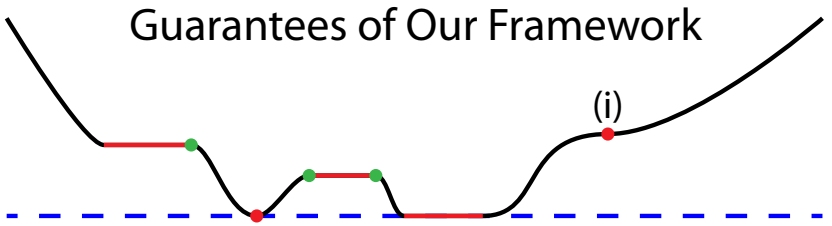


[1] Haeffele, Young, Vidal. Structured Low-Rank Matrix Factorization: Optimality, Algorithm, and Applications to Image Processing, ICML '14
[2] Haeffele, Vidal. Global Optimality in Tensor Factorization, Deep Learning and Beyond, arXiv '15

Main Results: Matrix Factorization

- **Theorem 2:** If the number of columns is large enough, local descent can reach a global minimizer from any initialization



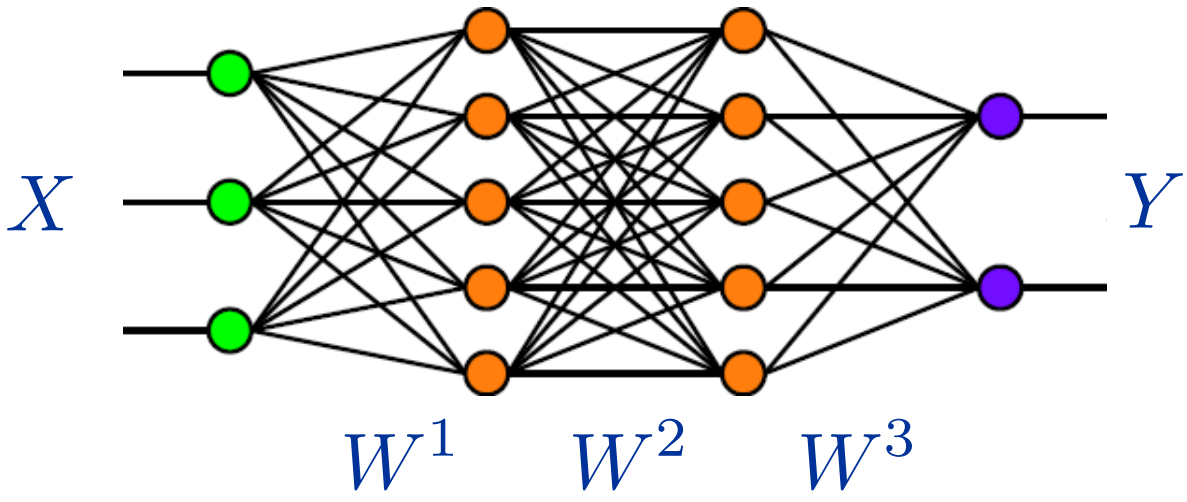


- **Meta-Algorithm:**
  - If not at a local minima, perform local descent
  - At local minima, test if Theorem 1 is satisfied. If yes => global minima
  - If not, increase size of factorization and find descent direction (u,v)

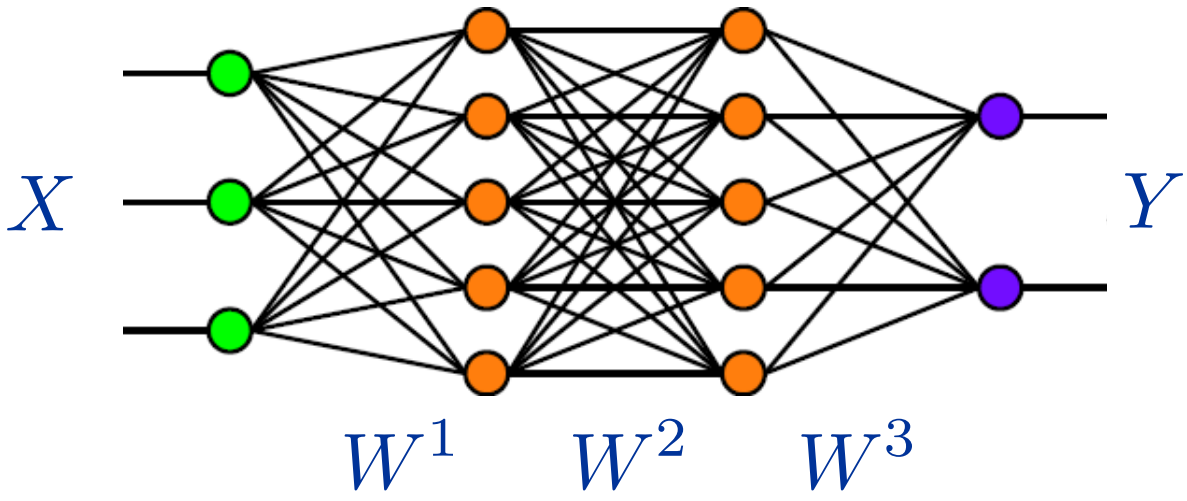
$$r \leftarrow r+1 \quad U \leftarrow \begin{bmatrix} U & u \end{bmatrix} \quad V \leftarrow \begin{bmatrix} V & v \end{bmatrix}$$

[1] Haeffele, Vidal. Global Optimality in Tensor Factorization, Deep Learning and Beyond, arXiv '15

From Matrix Factorization to Deep Learning



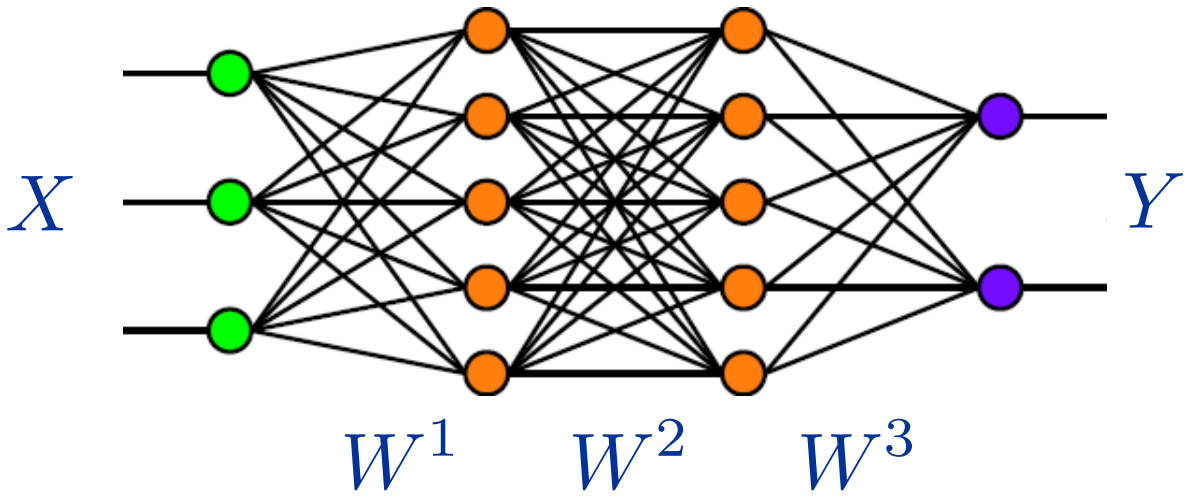
From Matrix Factorization to Deep Learning



X

input

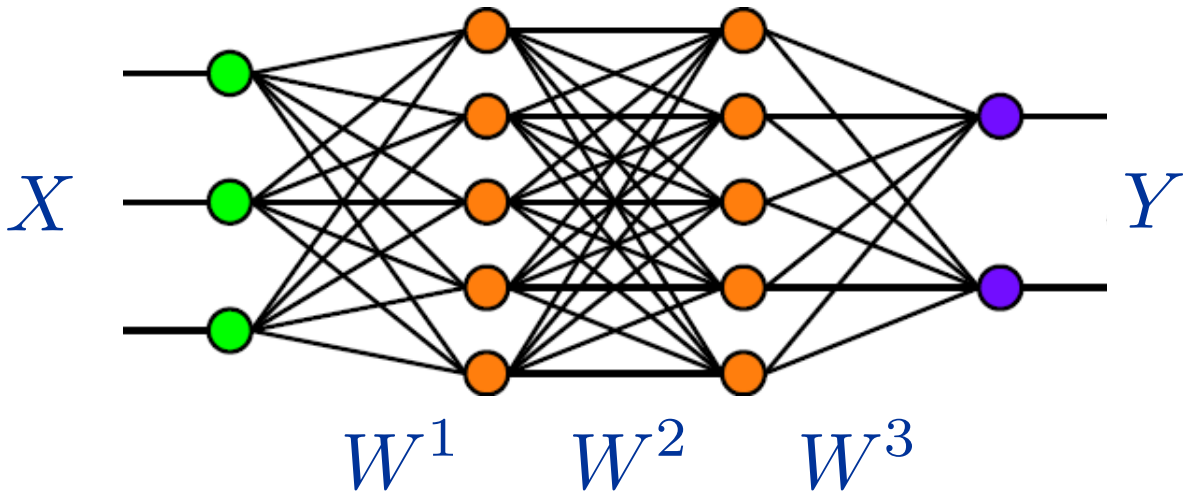
From Matrix Factorization to Deep Learning



$$XW^1$$

input weights

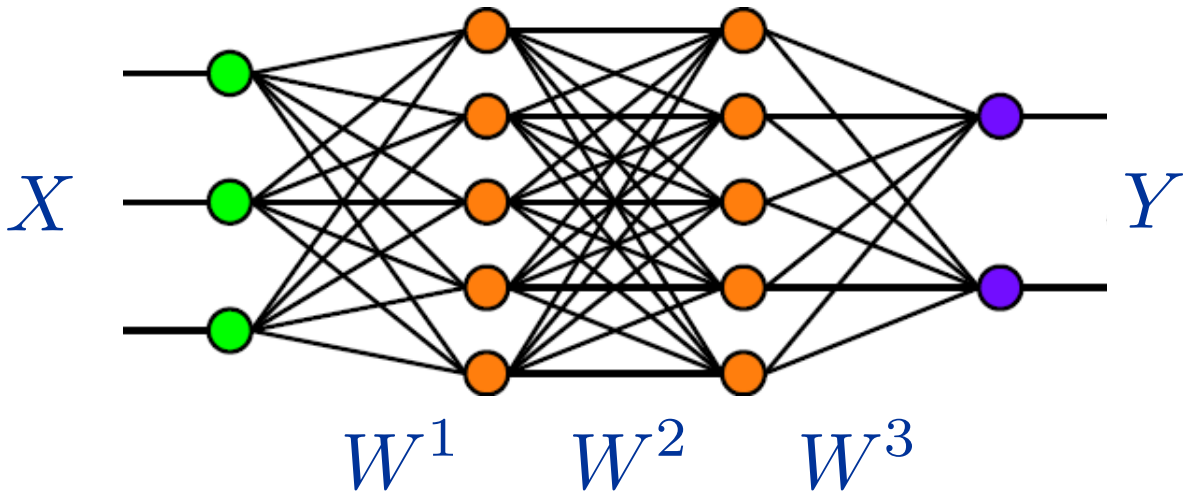
From Matrix Factorization to Deep Learning



$$\psi_1(XW^1)$$

activation input weights

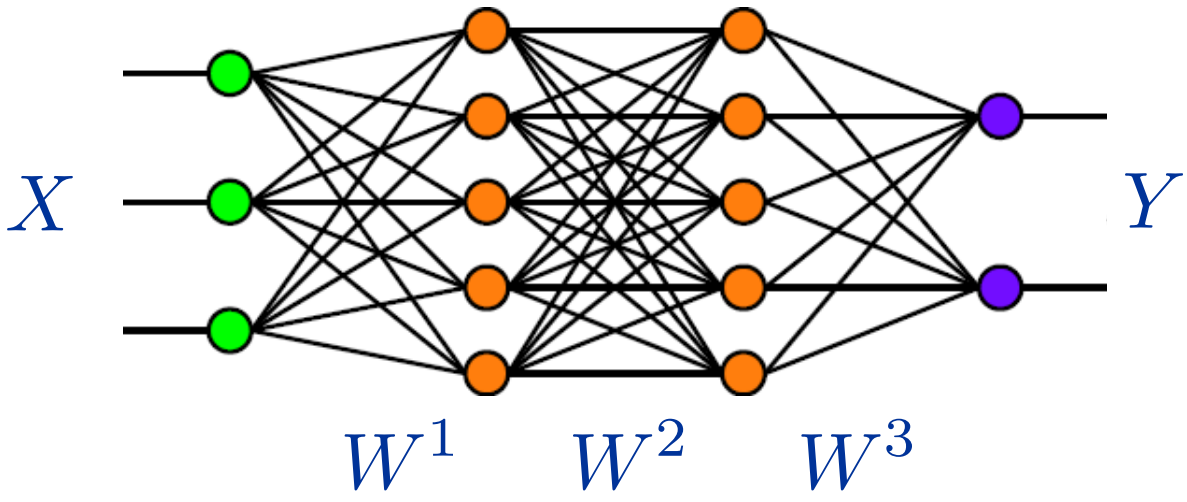
From Matrix Factorization to Deep Learning



$$\psi_1(XW^1)W^2$$

activation input weights

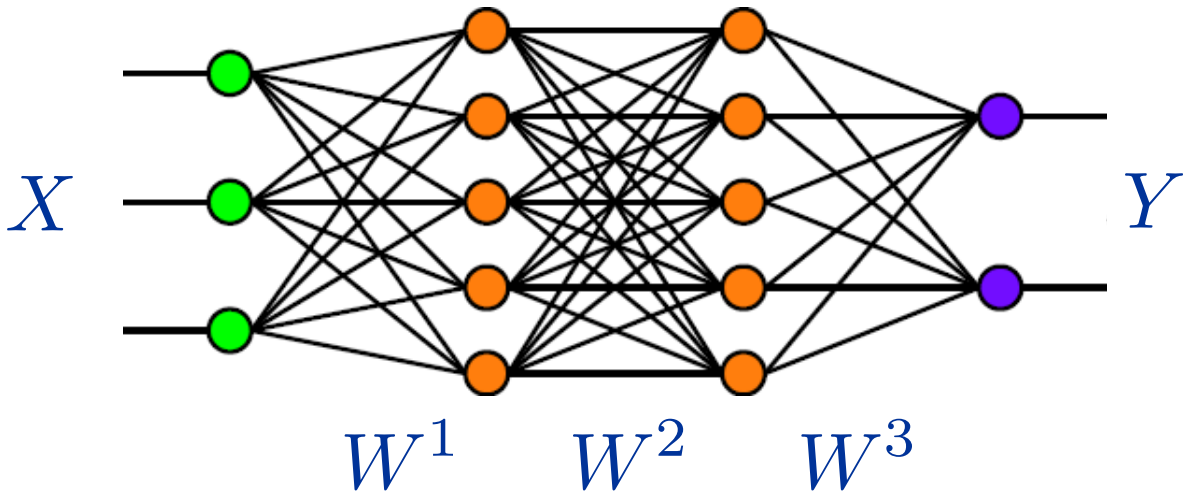
From Matrix Factorization to Deep Learning



$$\psi_2(\,\psi_1(XW^1)W^2\,)$$

activation input weights

From Matrix Factorization to Deep Learning



$$\Phi(X, W^1, \ldots, W^K) = \psi_K(\cdots\psi_2(\psi_1(XW^1)W^2)\cdots W^K)$$

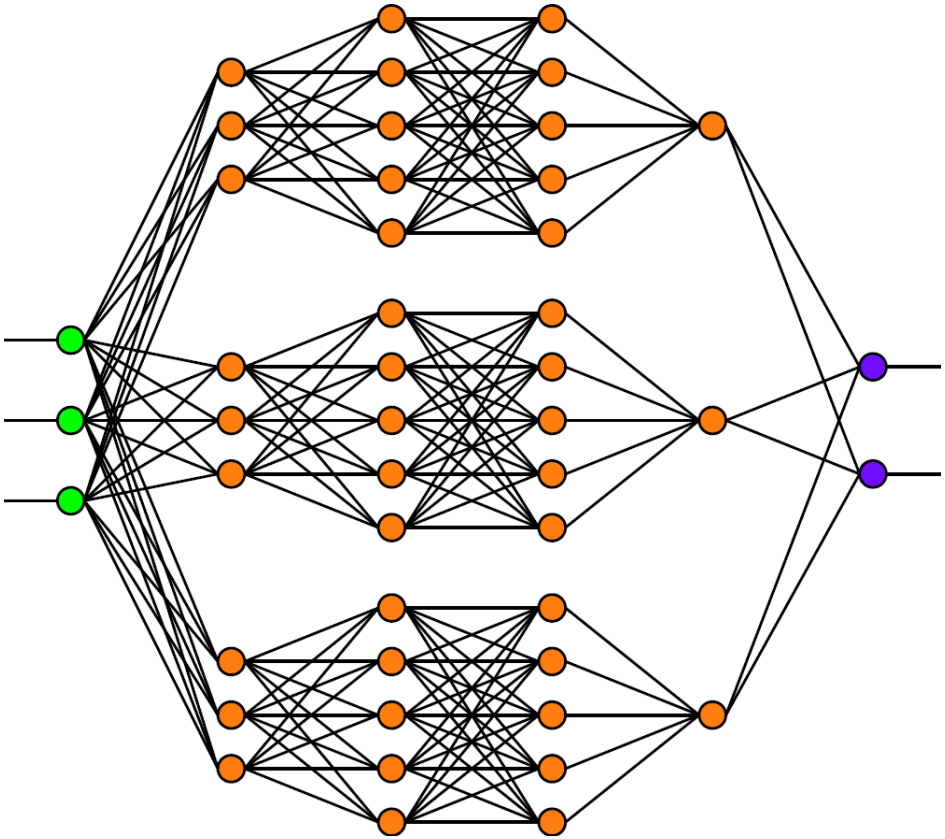
output

activation

input

weights

From Matrix Factorization to Deep Learning



- In matrix factorization we had

$$\Phi(U,V) = \sum_{i=1}^{r} U_i V_i^\top$$

- In positively homogeneous networks with parallel structure we have

$$\Phi(W^1,\ldots,W^K) = \sum_{i=1}^{r} \phi(W_i^1,\ldots,W_i^K)$$

From Matrix Factorization to Deep Learning

- In matrix factorization we had "generalized nuclear norm"

$$\|Z\|_{u,v} = \min_{U,V,r} \sum_{i=1}^{r} \|U_i\|_u \|V_i\|_v \quad \text{s.t.} \quad UV^\top = Z$$

From Matrix Factorization to Deep Learning

- In matrix factorization we had "generalized nuclear norm"

$$\|Z\|_{u,v} = \min_{U,V,r} \sum_{i=1}^{r} \|U_i\|_u \|V_i\|_v \quad \text{s.t.} \quad UV^\top = Z$$

- By analogy we define "nuclear deep net regularizer"

$$\Omega_{\phi,\theta}(Z) = \min_{\{W^k\},r} \sum_{i=1}^{r} \theta(W_i^1, \ldots, W_i^K) \text{ s.t. } \Phi(W_i^1, \ldots, W_i^K) = Z$$

where θ is positively homogeneous of the same degree as ϕ

From Matrix Factorization to Deep Learning

- In matrix factorization we had "generalized nuclear norm"

$$\|Z\|_{u,v} = \min_{U,V,r} \sum_{i=1}^{r} \|U_i\|_u \|V_i\|_v \quad \text{s.t.} \quad UV^\top = Z$$

- By analogy we define "nuclear deep net regularizer"

$$\Omega_{\phi,\theta}(Z) = \min_{\{W^k\},r} \sum_{i=1}^{r} \theta(W_i^1, \ldots, W_i^K) \text{ s.t. } \Phi(W_i^1, \ldots, W_i^K) = Z$$

where θ is positively homogeneous of the same degree as ϕ

- **Proposition:** $\Omega_{\phi,\theta}$ is convex

From Matrix Factorization to Deep Learning

- In matrix factorization we had "generalized nuclear norm"

$$\|Z\|_{u,v} = \min_{U,V,r} \sum_{i=1}^{r} \|U_i\|_u \|V_i\|_v \quad \text{s.t.} \quad UV^\top = Z$$

- By analogy we define "nuclear deep net regularizer"

$$\Omega_{\phi,\theta}(Z) = \min_{\{W^k\},r} \sum_{i=1}^{r} \theta(W_i^1, \ldots, W_i^K) \text{ s.t. } \Phi(W_i^1, \ldots, W_i^K) = Z$$

where θ is positively homogeneous of the same degree as ϕ

- **Proposition:** $\Omega_{\phi,\theta}$ is convex

- **Intuition:** regularizer Θ "comes from a convex function"

Main Results: Tensor Fact. & Deep Learning

$$\min_{\{W^k\}_{k=1}^K} \ell(Y, \Phi(X, W^1, \ldots, W^K)) + \lambda \Theta(W^1, \ldots, W^K)$$

[1] Haeffele, Young, Vidal. Structured Low-Rank Matrix Factorization: Optimality, Algorithm, and Applications to Image Processing, ICML '14
[2] Haeffele, Vidal. Global Optimality in Tensor Factorization, Deep Learning and Beyond, arXiv, '15
[3] Haeffele, Vidal. Global optimality in neural network training. CVPR 2017.

JOHNS HOPKINS
MATHEMATICAL INSTITUTE
for DATA SCIENCE

Main Results: Tensor Fact. & Deep Learning

$$\min_{\{W^k\}_{k=1}^K} \ell(Y, \Phi(X, W^1, \ldots, W^K)) + \lambda \Theta(W^1, \ldots, W^K)$$

- **Assumptions:**
  - $\ell(Y, Z)$: convex and once differentiable in Z
  - Φ and Θ: sums of positively homogeneous functions of same degree

$$\phi(\alpha W_i^1, \ldots, \alpha W_i^K) = \alpha^p \phi(W_i^1, \ldots, W_i^K) \quad \forall \alpha \geq 0$$

[1] Haeffele, Young, Vidal. Structured Low-Rank Matrix Factorization: Optimality, Algorithm, and Applications to Image Processing, ICML '14
[2] Haeffele, Vidal. Global Optimality in Tensor Factorization, Deep Learning and Beyond, arXiv, '15
[3] Haeffele, Vidal. Global optimality in neural network training. CVPR 2017.

Main Results: Tensor Fact. & Deep Learning

$$\min_{\{W^k\}_{k=1}^K} \ell(Y, \Phi(X, W^1, \ldots, W^K)) + \lambda \Theta(W^1, \ldots, W^K)$$

- **Assumptions:**
  - $\ell(Y, Z)$: convex and once differentiable in Z
  - Φ and Θ: sums of positively homogeneous functions of same degree

$$\phi(\alpha W_i^1, \ldots, \alpha W_i^K) = \alpha^p \phi(W_i^1, \ldots, W_i^K) \quad \forall \alpha \geq 0$$

- **Theorem 1:** A local minimizer such that for some *i* and all k $W_i^k = 0$ is a global minimizer

[1] Haeffele, Young, Vidal. Structured Low-Rank Matrix Factorization: Optimality, Algorithm, and Applications to Image Processing, ICML '14
[2] Haeffele, Vidal. Global Optimality in Tensor Factorization, Deep Learning and Beyond, arXiv, '15
[3] Haeffele, Vidal. Global optimality in neural network training. CVPR 2017.

Main Results: Tensor Fact. & Deep Learning

$$\min_{\{W^k\}_{k=1}^K} \ell(Y, \Phi(X, W^1, \ldots, W^K)) + \lambda \Theta(W^1, \ldots, W^K)$$

- **Assumptions:**
  - $\ell(Y, Z)$: convex and once differentiable in Z
  - Φ and Θ: sums of positively homogeneous functions of same degree

$$\phi(\alpha W_i^1, \ldots, \alpha W_i^K) = \alpha^p \phi(W_i^1, \ldots, W_i^K) \quad \forall \alpha \geq 0$$

- **Theorem 1:** A local minimizer such that for some *i* and all k $W_i^k = 0$ is a global minimizer

- **Theorem 2:** If the size of the network is large enough, local descent can reach a global minimizer from any initialization

[1] Haeffele, Young, Vidal. Structured Low-Rank Matrix Factorization: Optimality, Algorithm, and Applications to Image Processing, ICML '14
[2] Haeffele, Vidal. Global Optimality in Tensor Factorization, Deep Learning and Beyond, arXiv, '15
[3] Haeffele, Vidal. Global optimality in neural network training. CVPR 2017.

JOHNS HOPKINS
MATHEMATICAL INSTITUTE
for DATA SCIENCE

Summary so Far

- **Size matters**
  - Optimize not only the network weights, but also the network size
  - Today: size = number of neurons or number of parallel networks
  - Tomorrow: size = number of layers + number of neurons per layer
- **Regularization matters**
  - Use “positively homogeneous regularizer” of same degree as network
  - How to build a regularizer that controls number of layers + number of neurons per layer
- **Not done yet**
  - Checking if we are at a local minimum or finding a descent direction can be NP hard
  - Need “computationally tractable” regularizers

More Information,

Vision Lab @ JHU

http://www.vision.jhu.edu

Center for Imaging Science @ JHU

http://www.cis.jhu.edu

Mathematical Institute for Data Science @ JHU

http://www.minds.jhu.edu

Thank You!