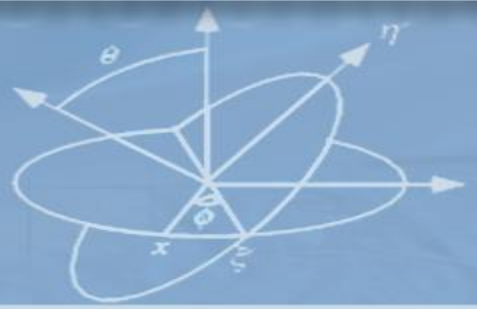
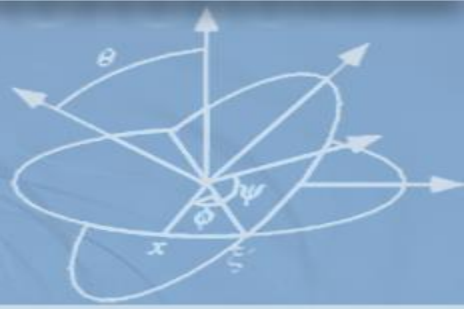
JHU vision lab

Global Optimality in Neural Network Training

Benjamin D. Haeffele and René Vidal

Johns Hopkins University, Center for Imaging Science. Baltimore, USA

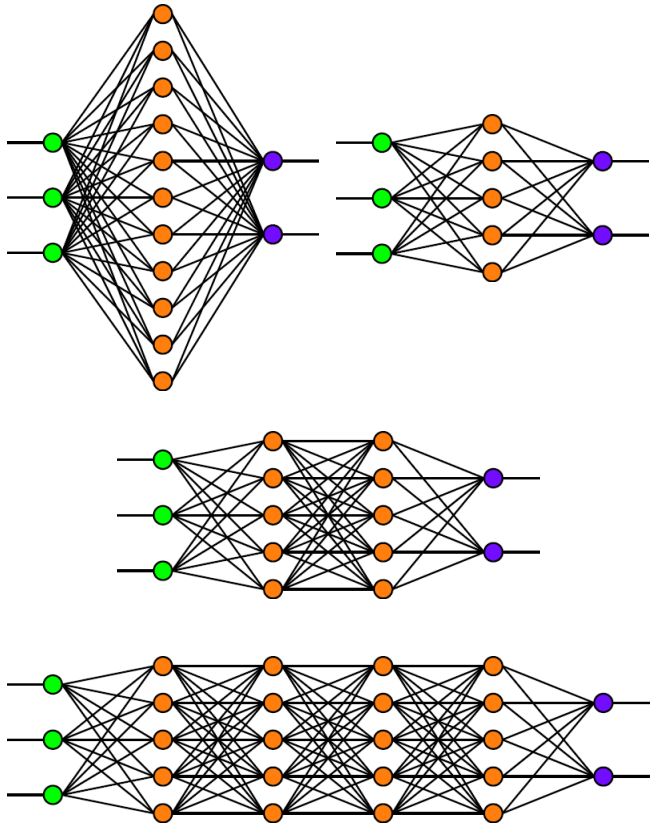




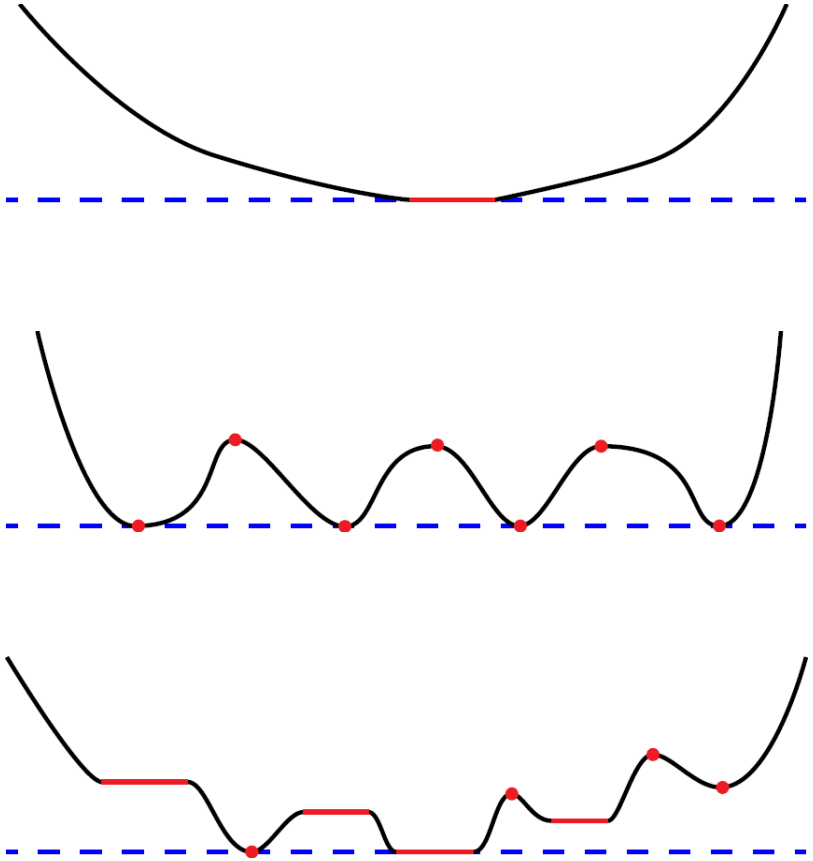
Center for IMAGING SCIENCE

Questions in Deep Learning

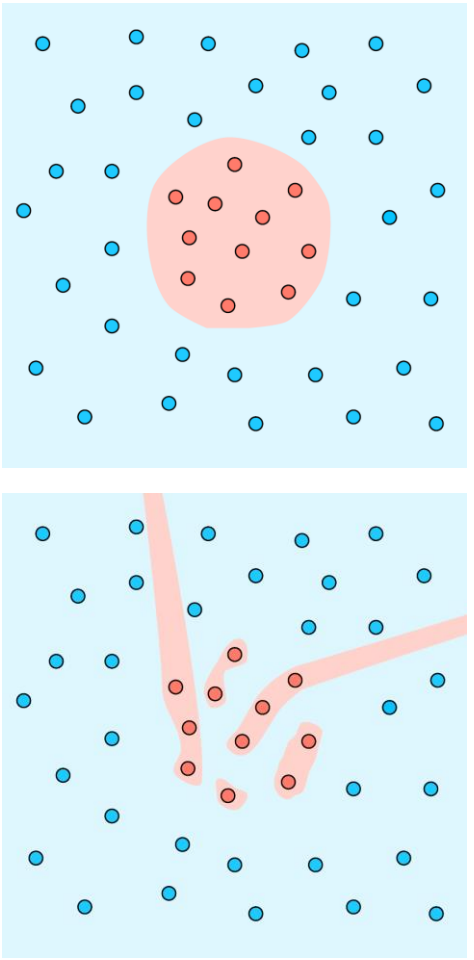
Architecture Design



Optimization

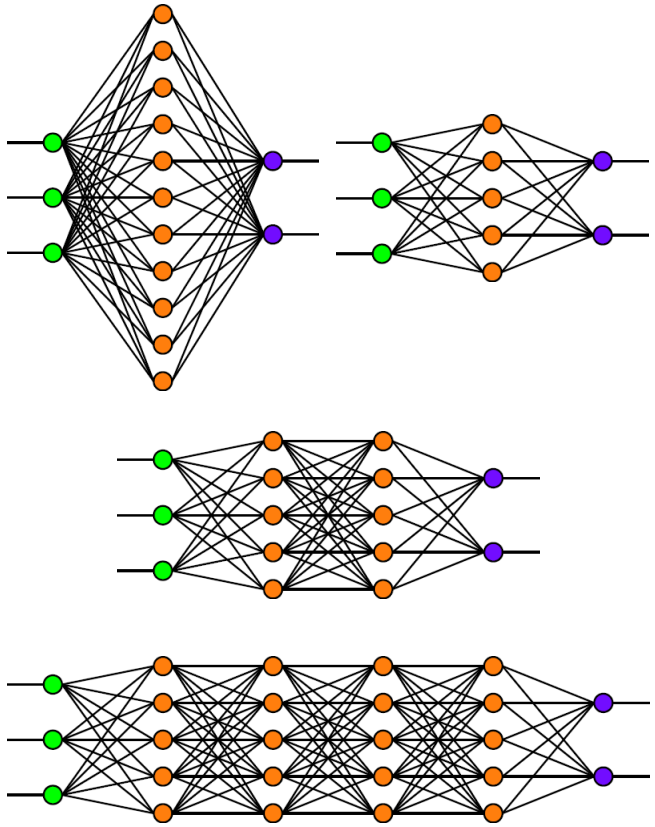


Generalization



Questions in Deep Learning

Are there principled ways to design networks?



- How many layers?
- Size of layers?
- Choice of layer types?
- How does architecture impact expressiveness? [1]

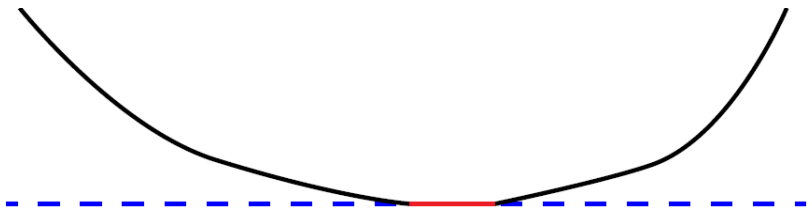
[1] Cohen, et al., “On the expressive power of deep learning: A tensor analysis.” COLT. (2016)

Questions in Deep Learning

How to train neural networks?

Questions in Deep Learning

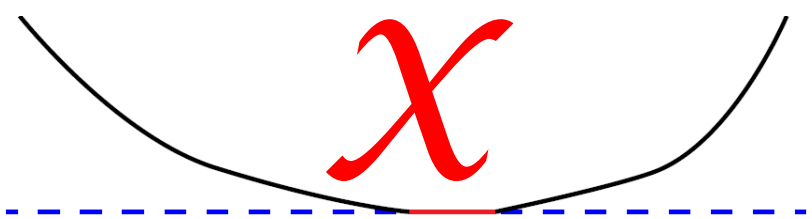
How to train neural networks?



- Problem is non-convex.

Questions in Deep Learning

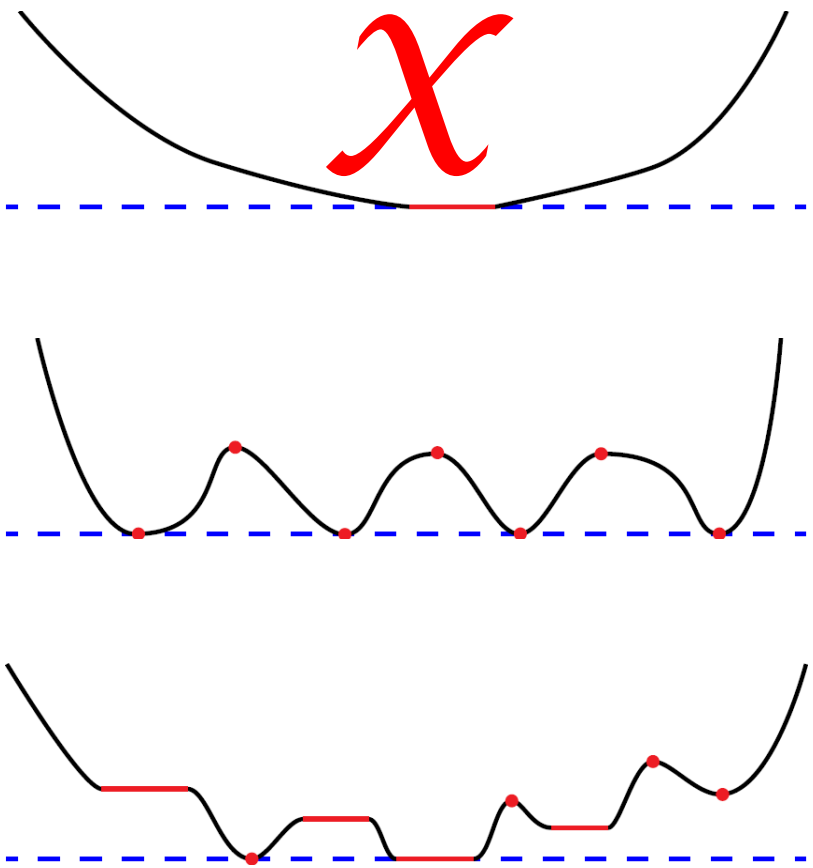
How to train neural networks?



- Problem is non-convex.

Questions in Deep Learning

How to train neural networks?



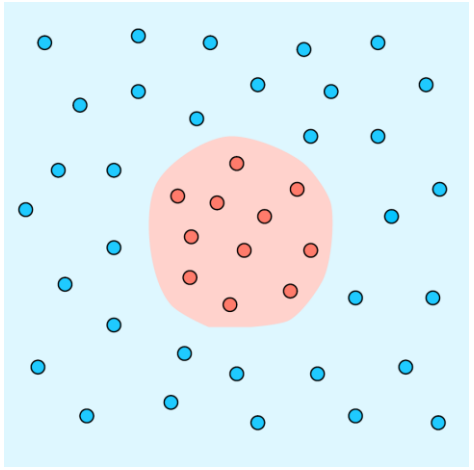
- Problem is non-convex.
- What does the loss surface look like? [1]
- Any guarantees for network training? [2]
- How to guarantee optimality?
- When will local descent succeed?

[1] Choromanska, et al., "The loss surfaces of multilayer networks." Artificial Intelligence and Statistics. (2015)
[2] Janzamin, et al., "Beating the perils of non-convexity: Guaranteed training of neural networks using tensor methods." arXiv (2015).

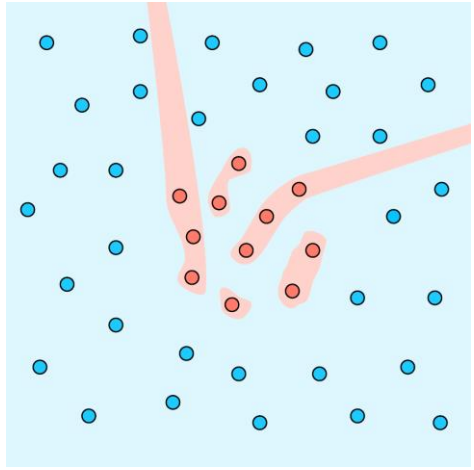
Questions in Deep Learning

Performance Guarantees?

✓**Simple**

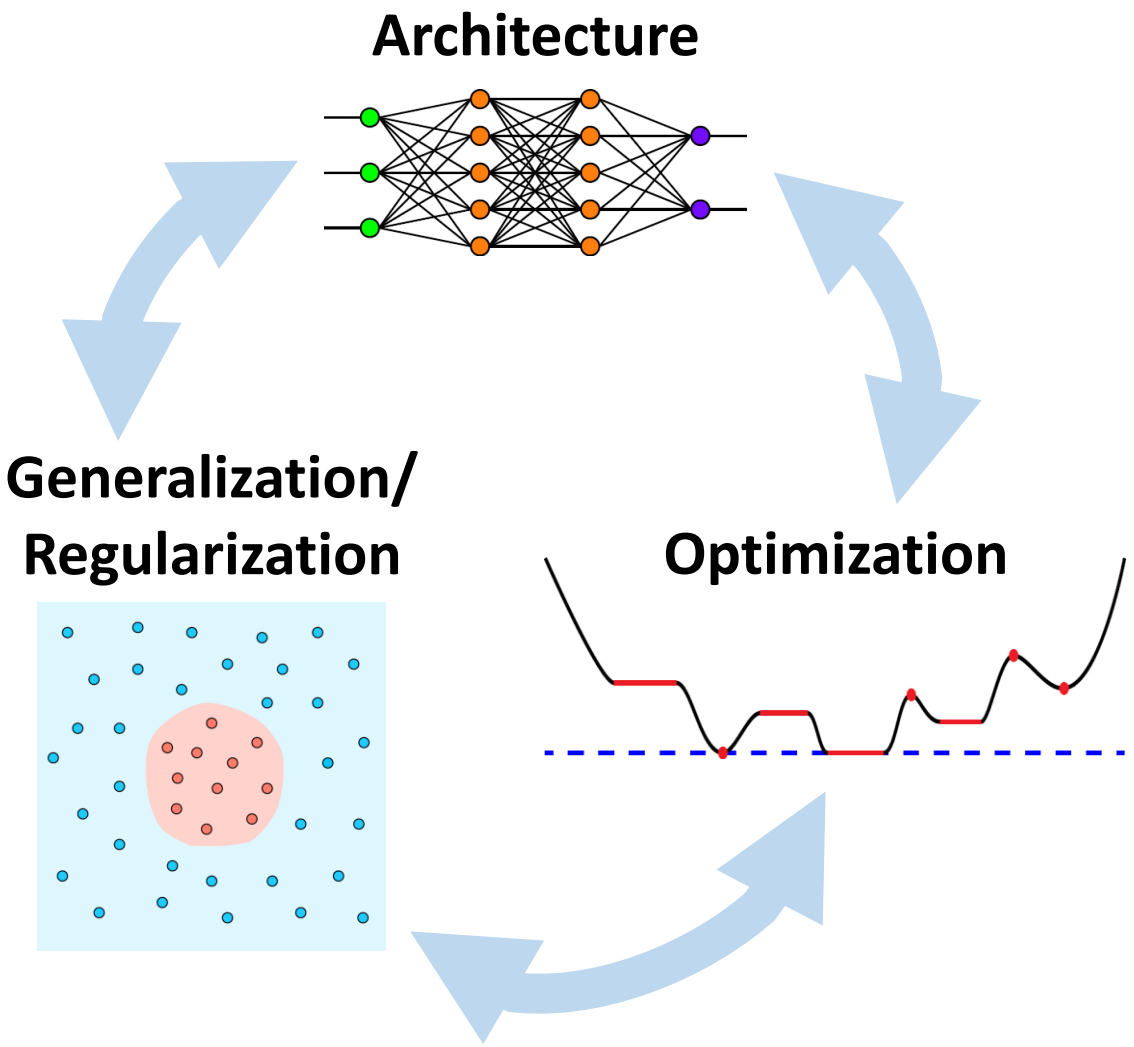


χ **Complex**



- How do networks generalize?
- How should networks be regularized?
- How to prevent overfitting?

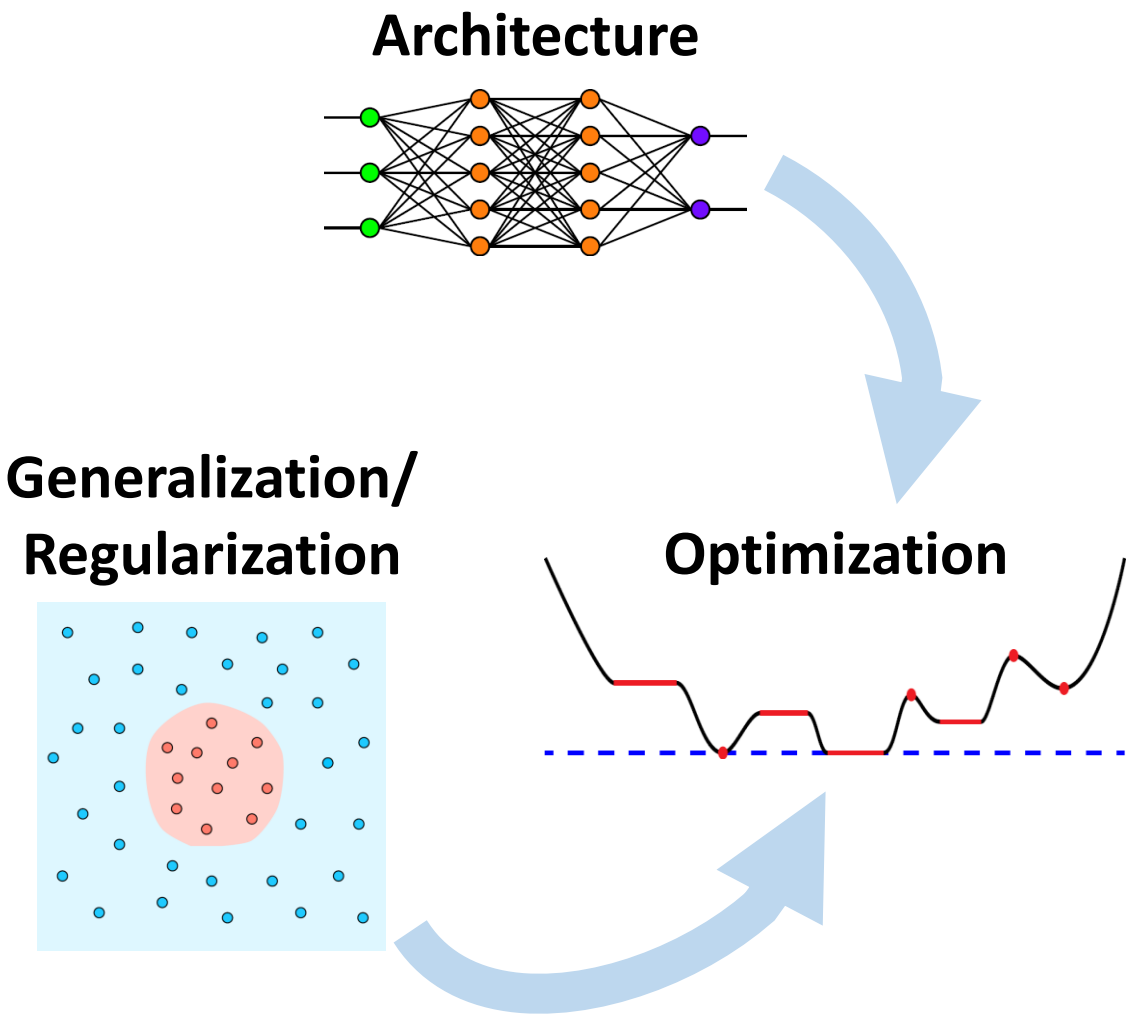
Interrelated Problems



- Optimization can impact generalization. [1]
- Architecture has a strong effect on the generalization of networks. [2]
- Some architectures could be easier to optimize than others.

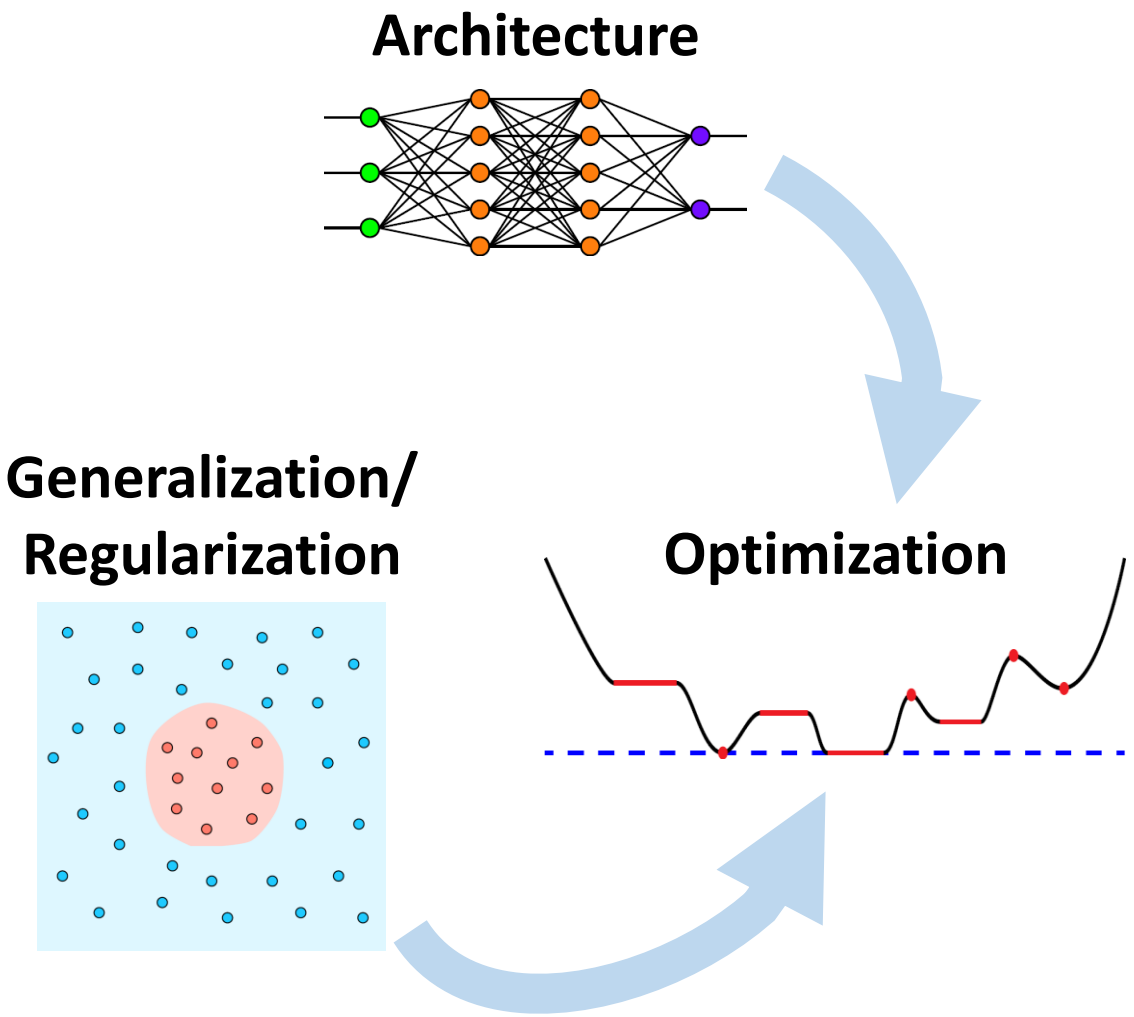
[1] Neyshabur, et al., “In Search of the Real Inductive Bias: On the Role of Implicit Regularization in Deep Learning.” ICLR workshop. (2015).
[2] Zhang, et al., “Understanding deep learning requires rethinking generalization.” ICLR. (2017).

Today's Talk: The Questions



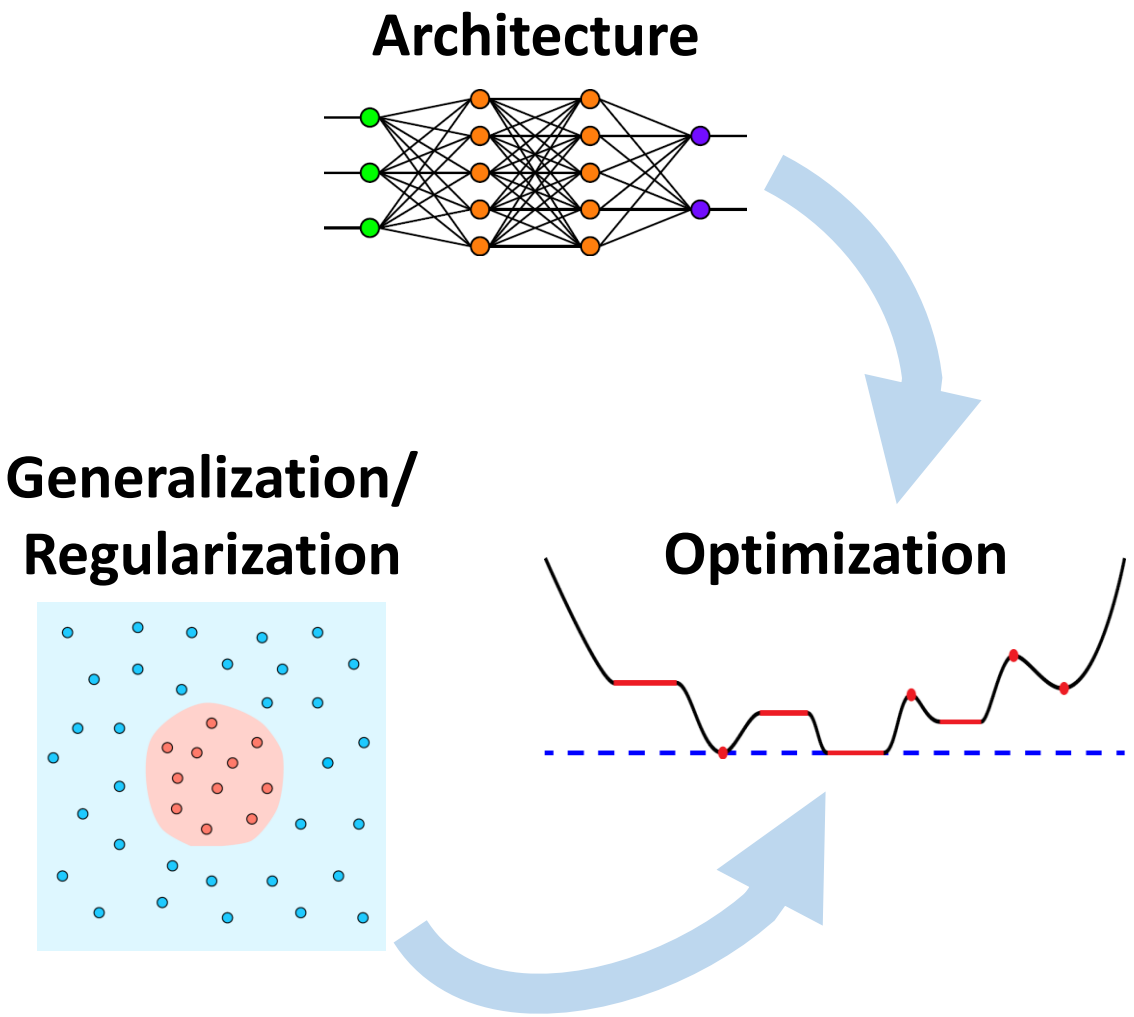
- Are there properties of the network architecture that allow efficient optimization?

Today's Talk: The Questions



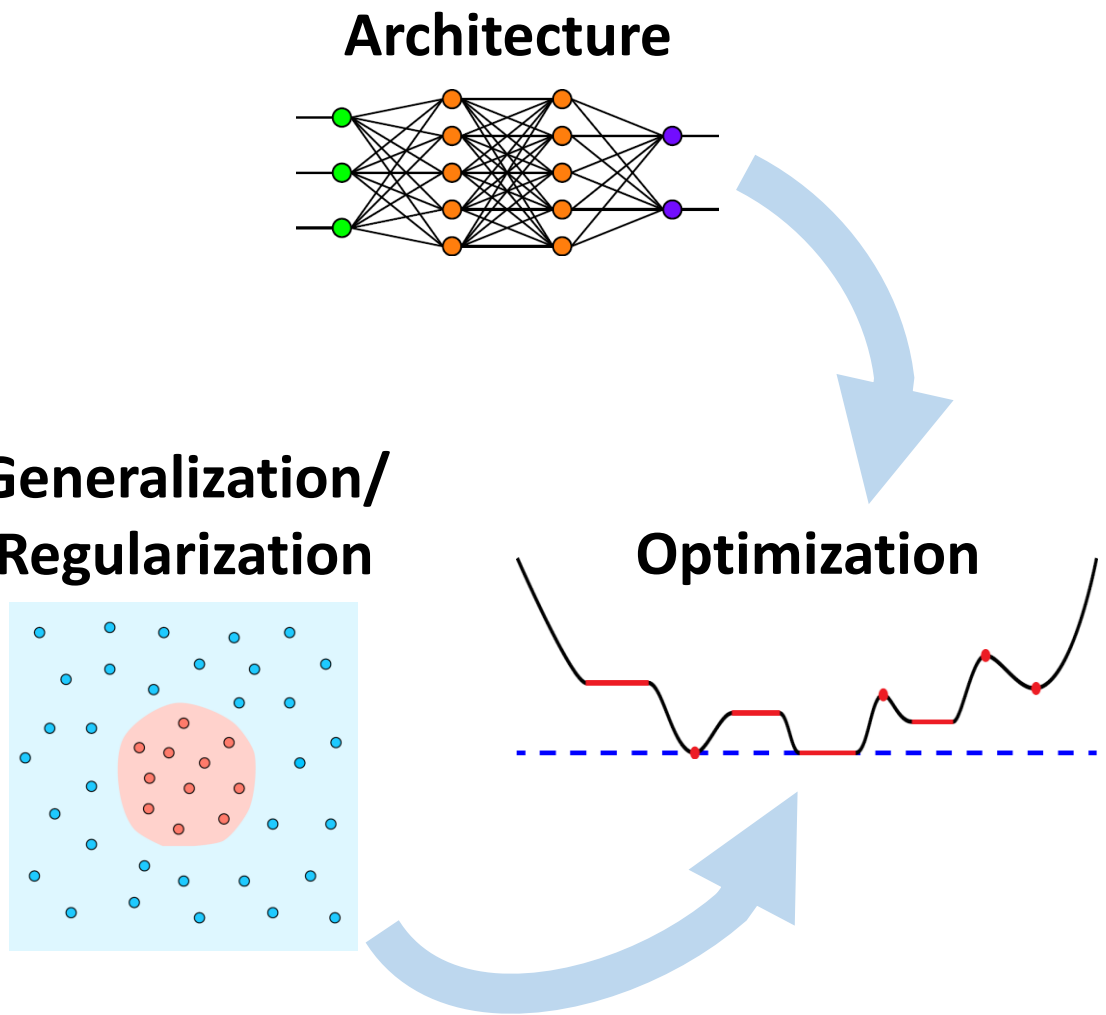
- Are there properties of the network architecture that allow efficient optimization?
  - Positive Homogeneity
  - Parallel Subnetwork Structure

Today's Talk: The Questions



- Are there properties of the network architecture that allow efficient optimization?
  - Positive Homogeneity
  - Parallel Subnetwork Structure
- Are there properties of the regularization that allow efficient optimization?

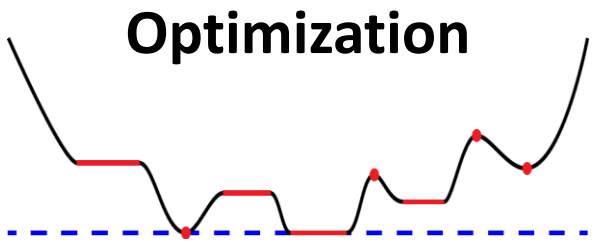
Today's Talk: The Questions



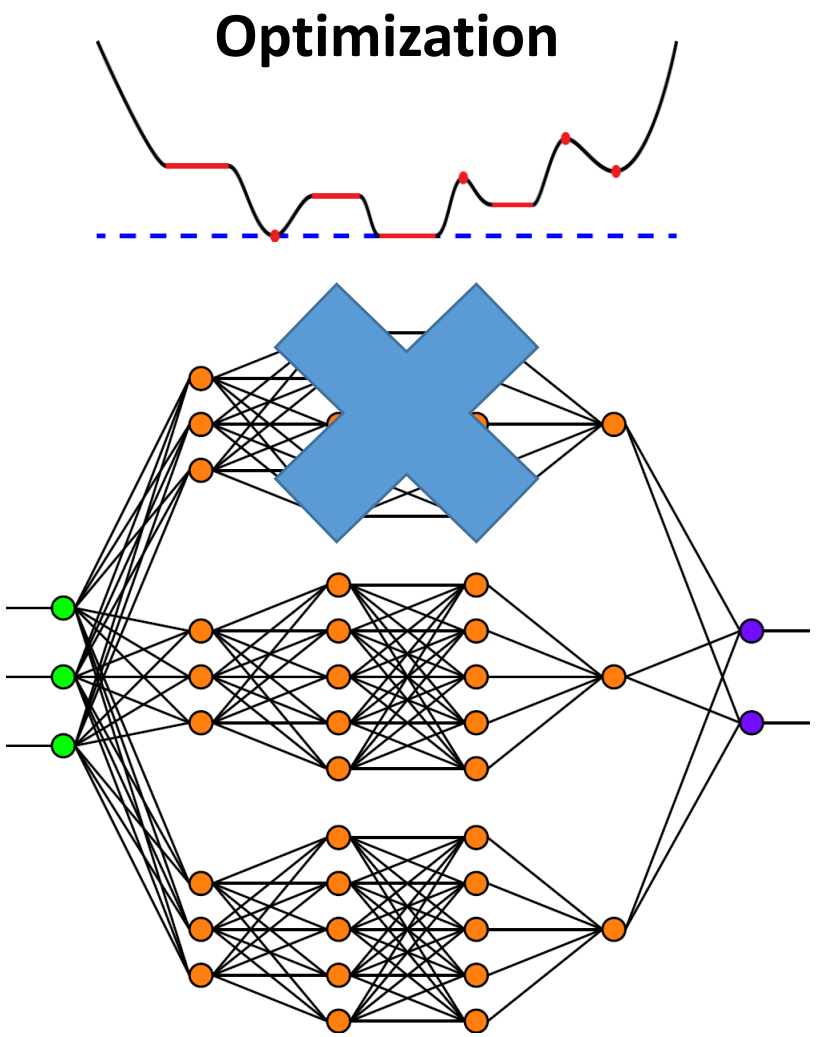
- Are there properties of the network architecture that allow efficient optimization?
  - Positive Homogeneity
  - Parallel Subnetwork Structure
- Are there properties of the regularization that allow efficient optimization?
  - Positive Homogeneity
  - Adapt network architecture to data [1]

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

Today's Talk: The Results

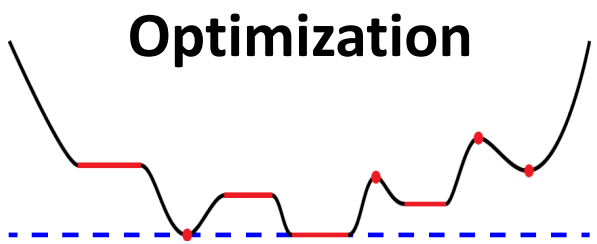


Today's Talk: The Results

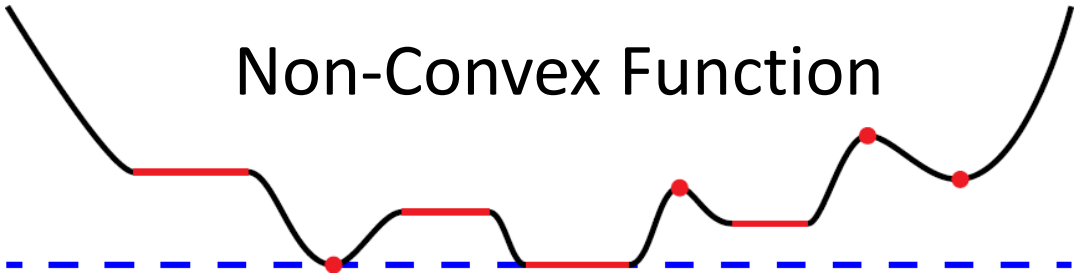


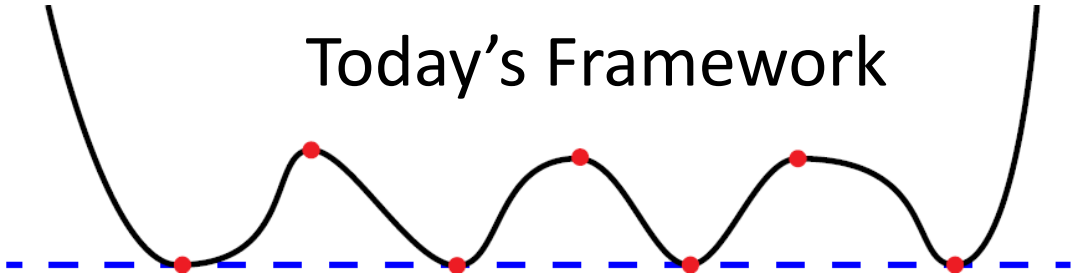
- A local minimum such that one subnetwork is all zero is a global minimum.

Today's Talk: The Results

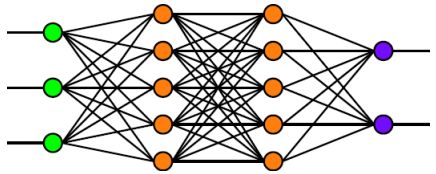


- Once the size of the network becomes large enough…
- Local descent can reach a global minimum from any initialization.



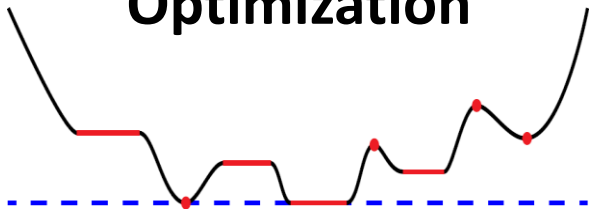


Outline

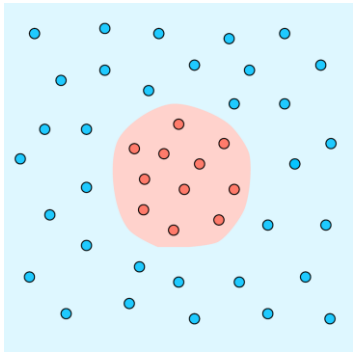


Architecture

Optimization



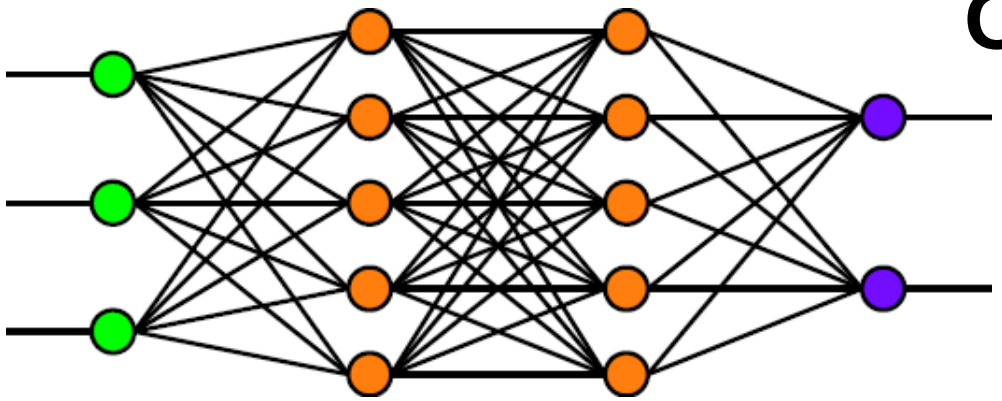
Generalization/ Regularization



1. **Network properties that allow efficient optimization**
   - **Positive Homogeneity**
   - Parallel Subnetwork Structure
2. Network size from regularization
3. Theoretical guarantees
   - Sufficient conditions for global optimality
   - Local descent can reach global minimizers

Key Property 1: Positive Homogeneity

- Start with a network.



W^1 W^2 W^3

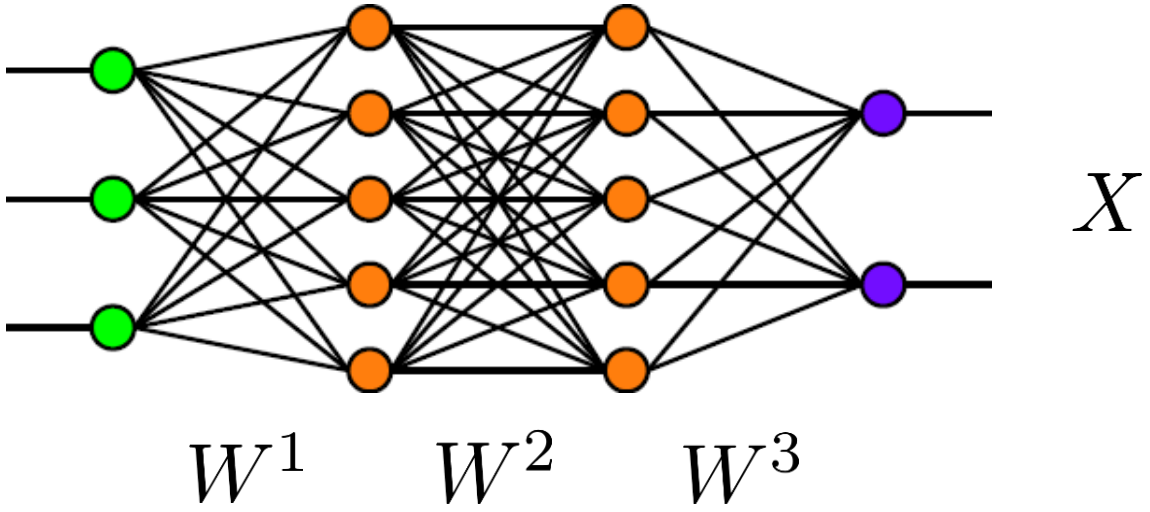
Network Weights

Network Outputs

X

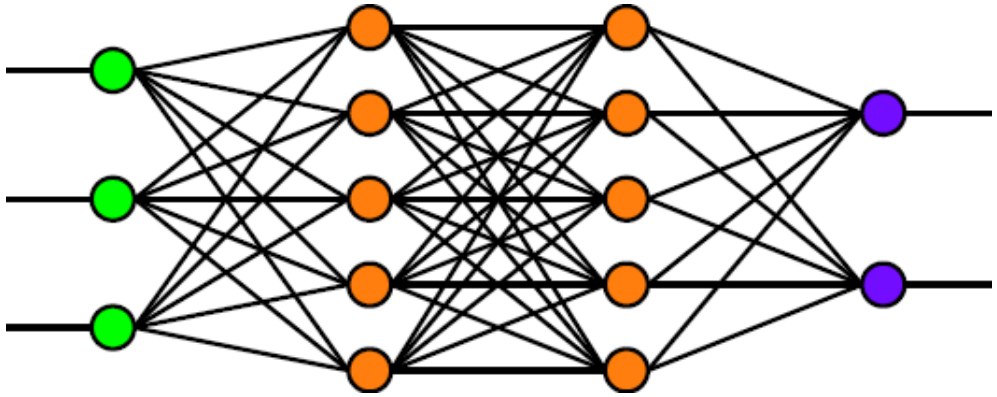
Key Property 1: Positive Homogeneity

- Scale the weights by a non-negative constant.



Key Property 1: Positive Homogeneity

- Scale the weights by a non-negative constant.



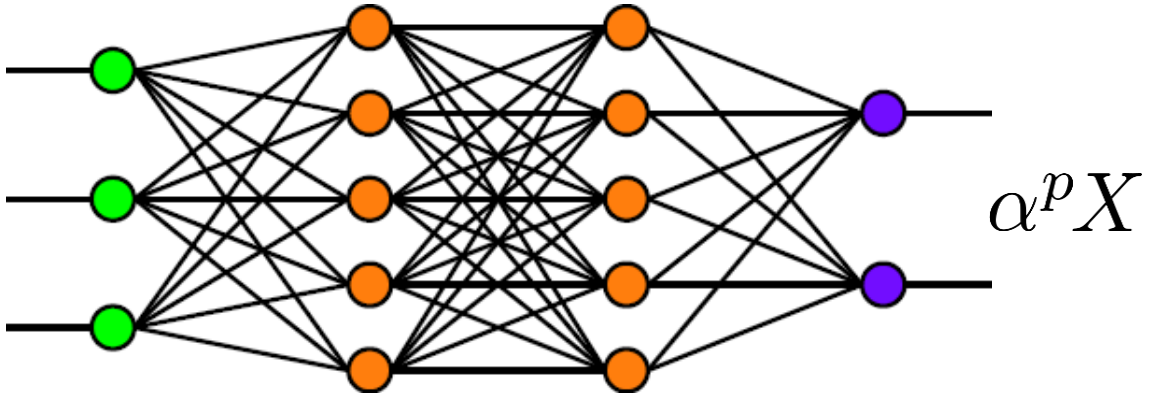
X

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

$\alpha \geq 0$

Key Property 1: Positive Homogeneity

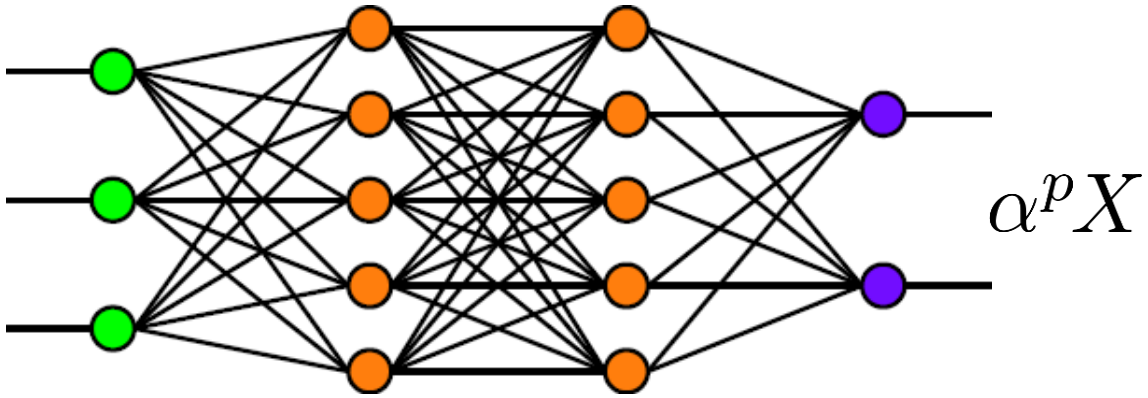
- The network output scales by the constant to some power.



$$\alpha \geq 0$$

Key Property 1: Positive Homogeneity

- The network output scales by the constant to some power.



$$\alpha \geq 0$$

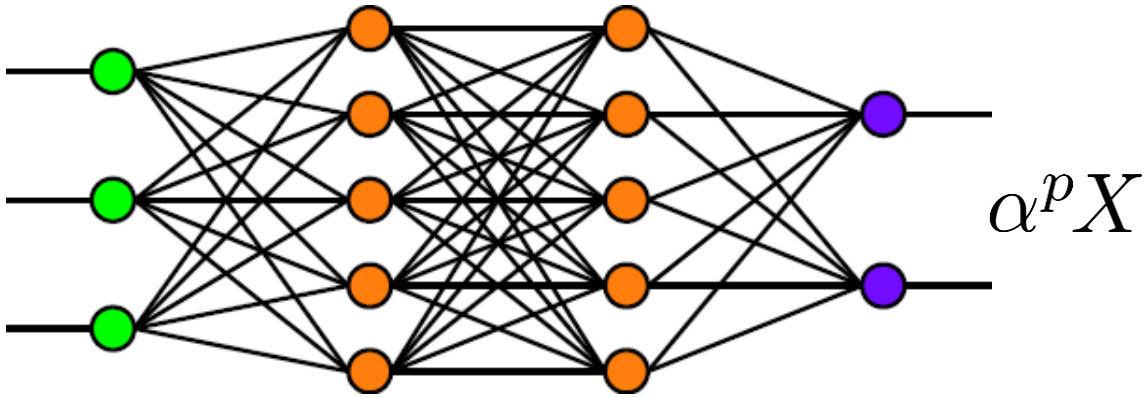
Network Mapping

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

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

Key Property 1: Positive Homogeneity

- The network output scales by the constant to some power.



$$\alpha \geq 0$$

Network Mapping

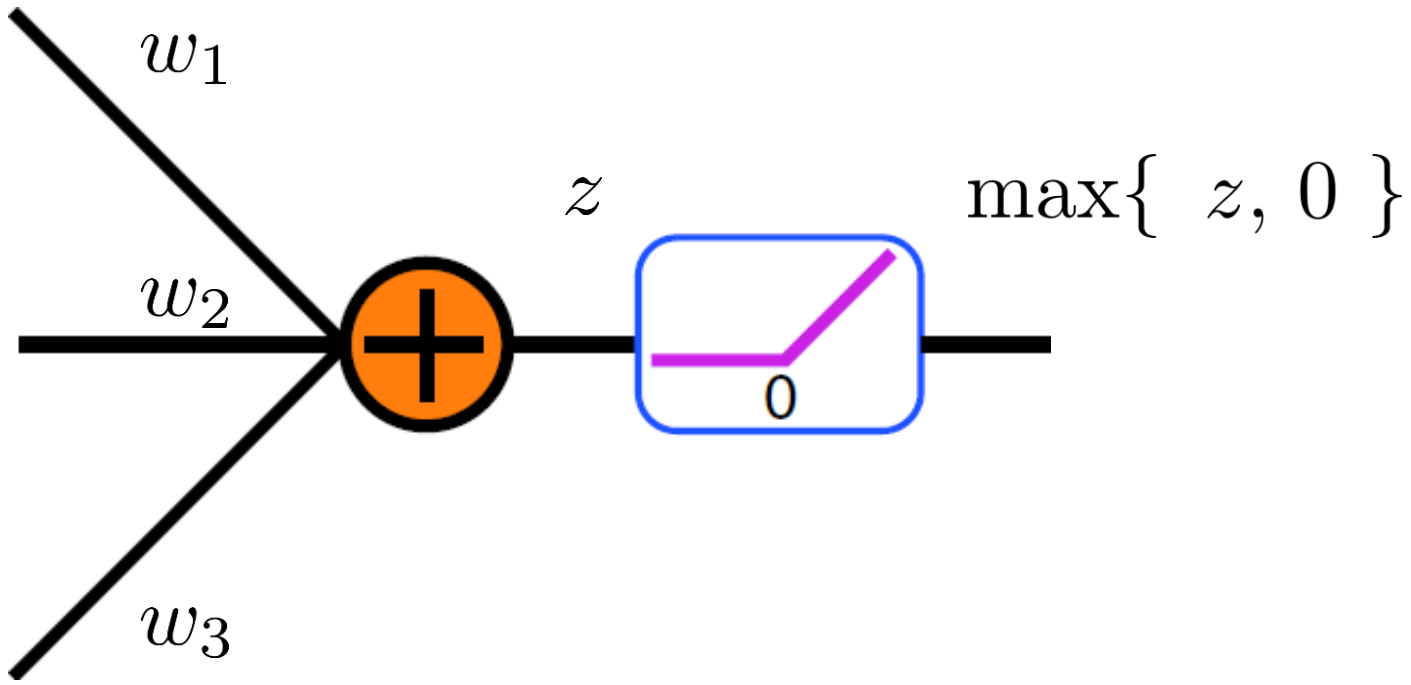
$$\Phi(W^1, W^2, W^3) = X$$

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

p - Degree of positive homogeneity

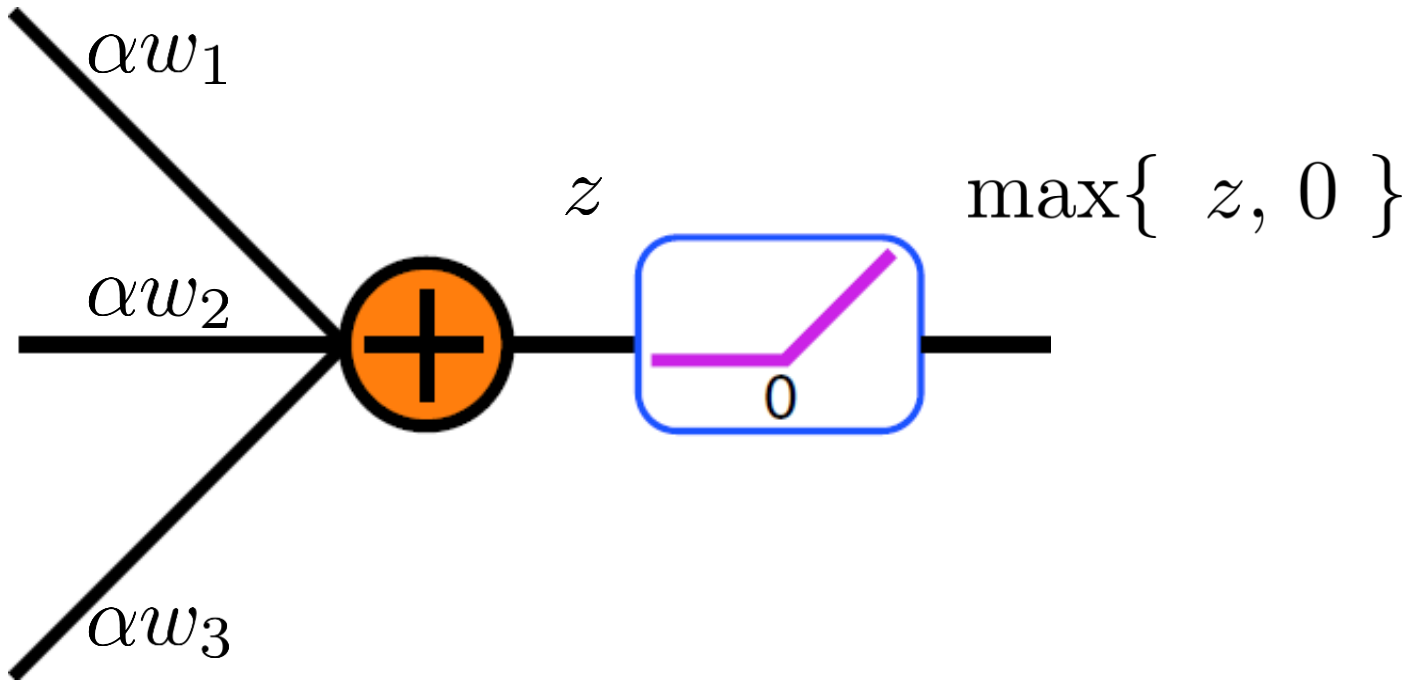
Most Modern Networks Are Positively Homogeneous

- **Example:** Rectified Linear Units (ReLUs)



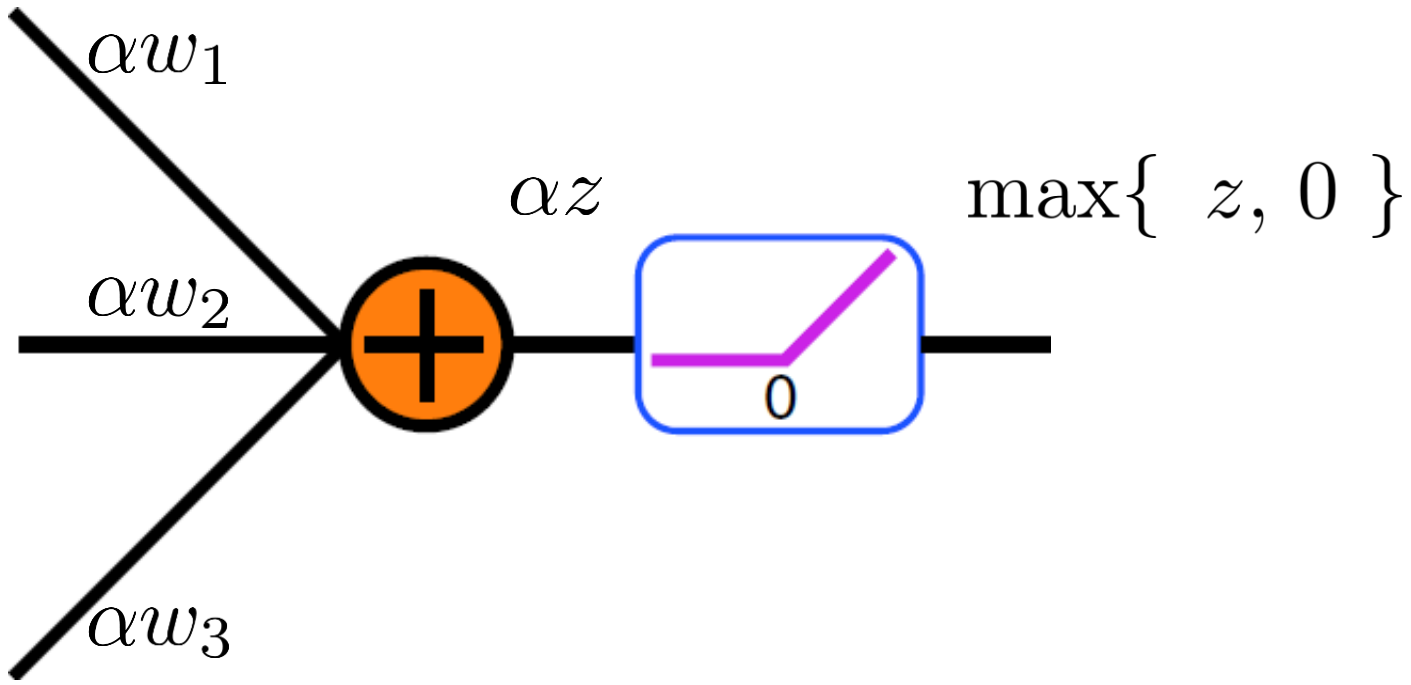
Most Modern Networks Are Positively Homogeneous

- **Example:** Rectified Linear Units (ReLUs)



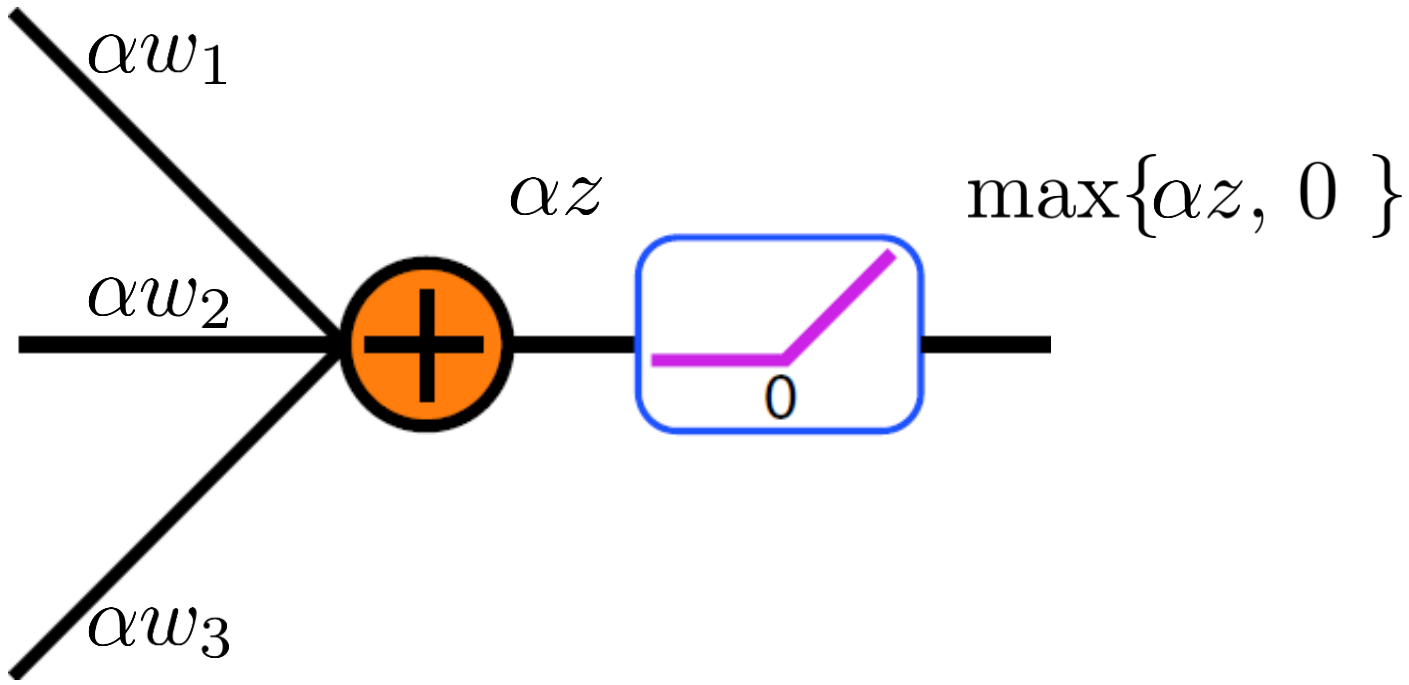
Most Modern Networks Are Positively Homogeneous

- **Example:** Rectified Linear Units (ReLUs)



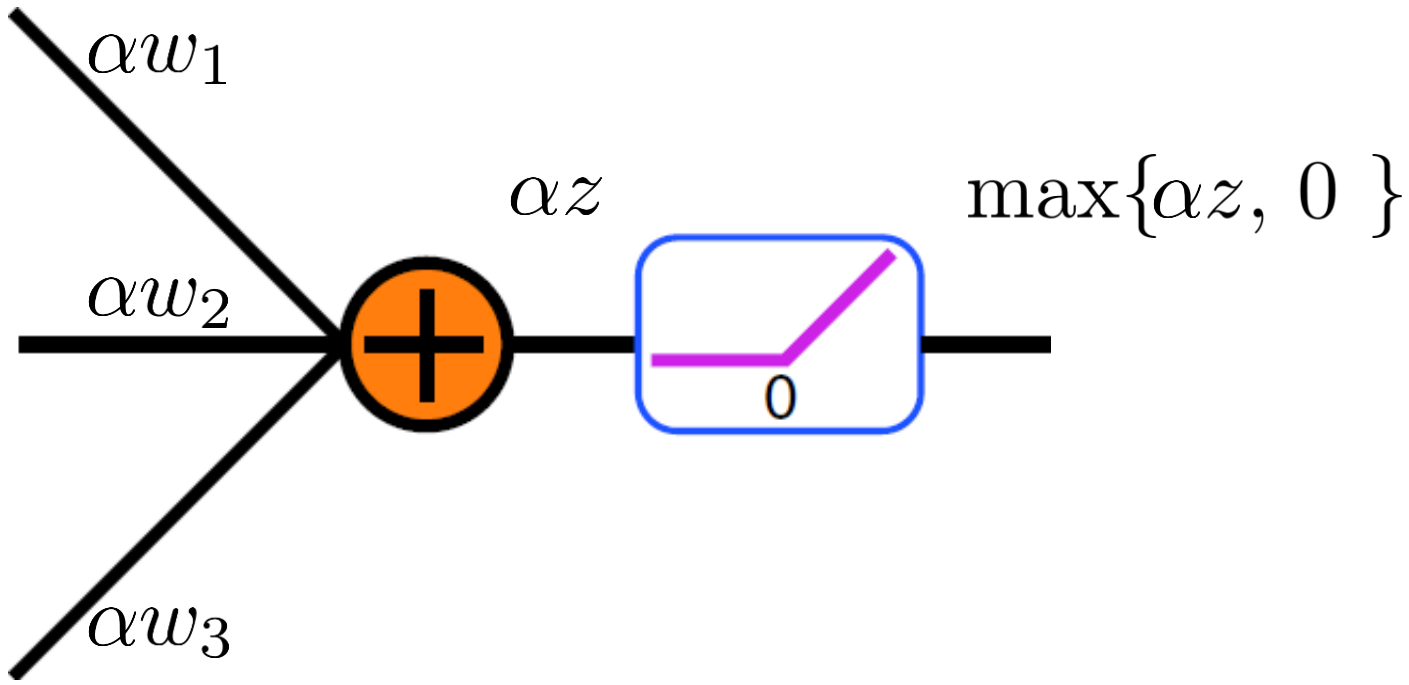
Most Modern Networks Are Positively Homogeneous

- **Example:** Rectified Linear Units (ReLUs)



Most Modern Networks Are Positively Homogeneous

- **Example:** Rectified Linear Units (ReLUs)

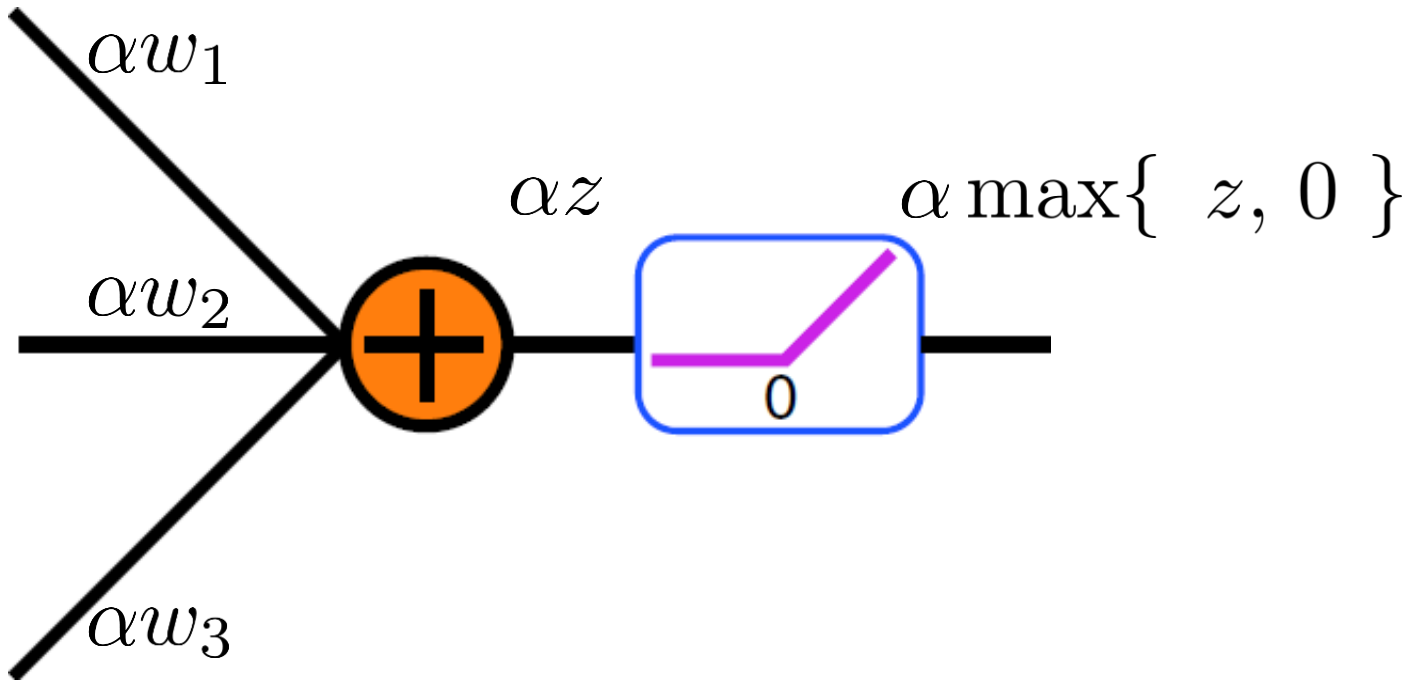


$$\alpha \geq 0$$

Doesn't change rectification

Most Modern Networks Are Positively Homogeneous

- **Example:** Rectified Linear Units (ReLUs)

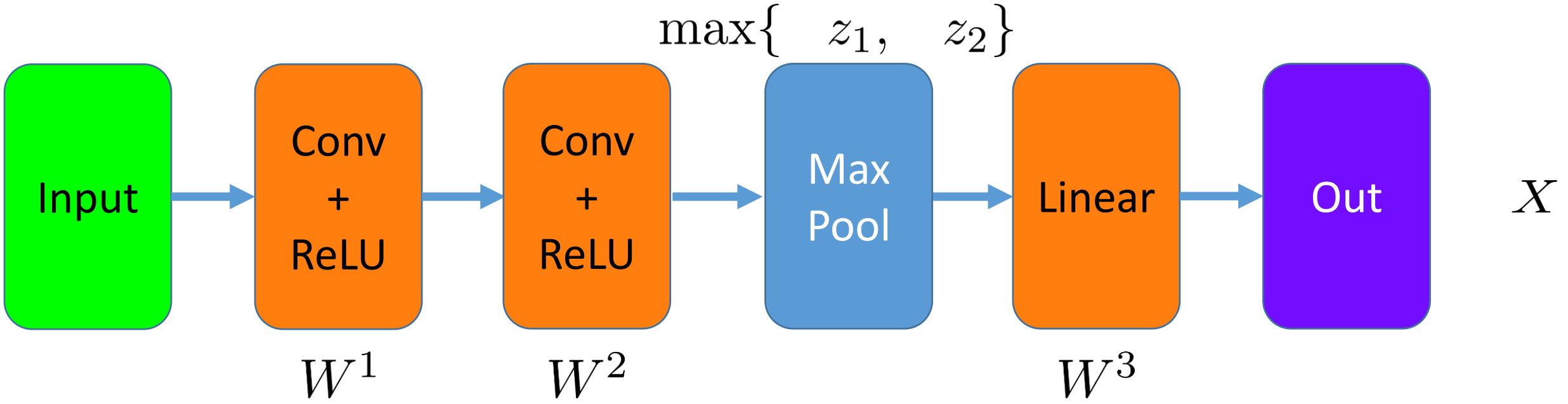


$$\alpha \geq 0$$

Doesn't change rectification

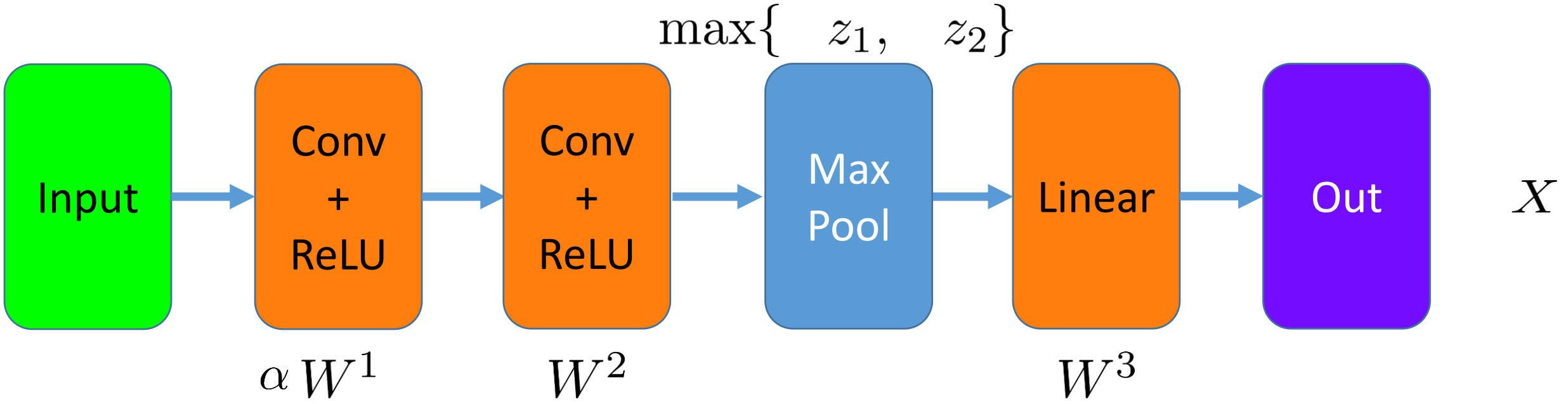
Most Modern Networks Are Positively Homogeneous

- **Simple Network**



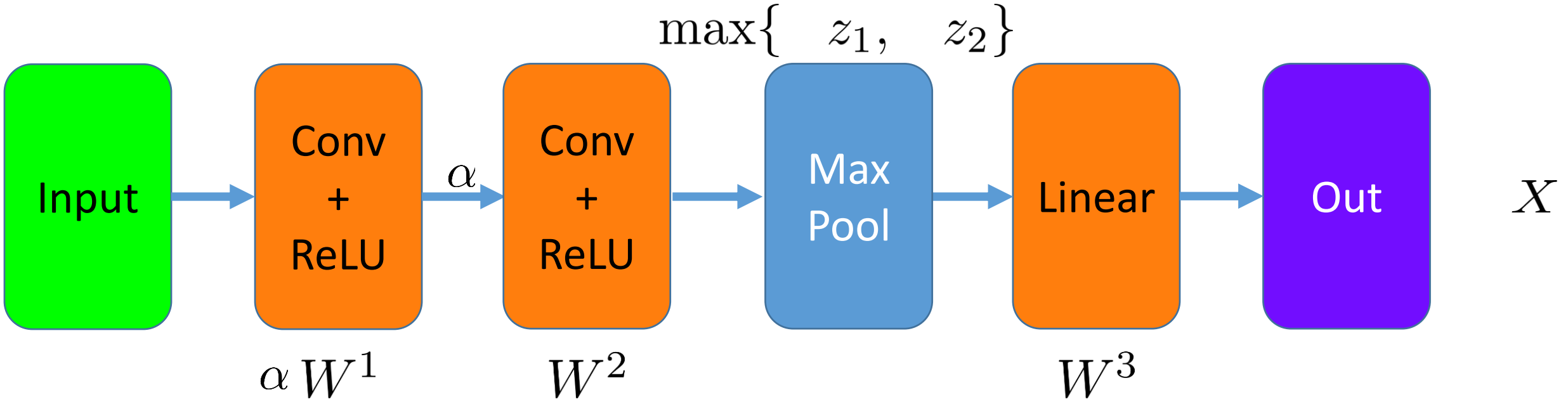
Most Modern Networks Are Positively Homogeneous

- **Simple Network**



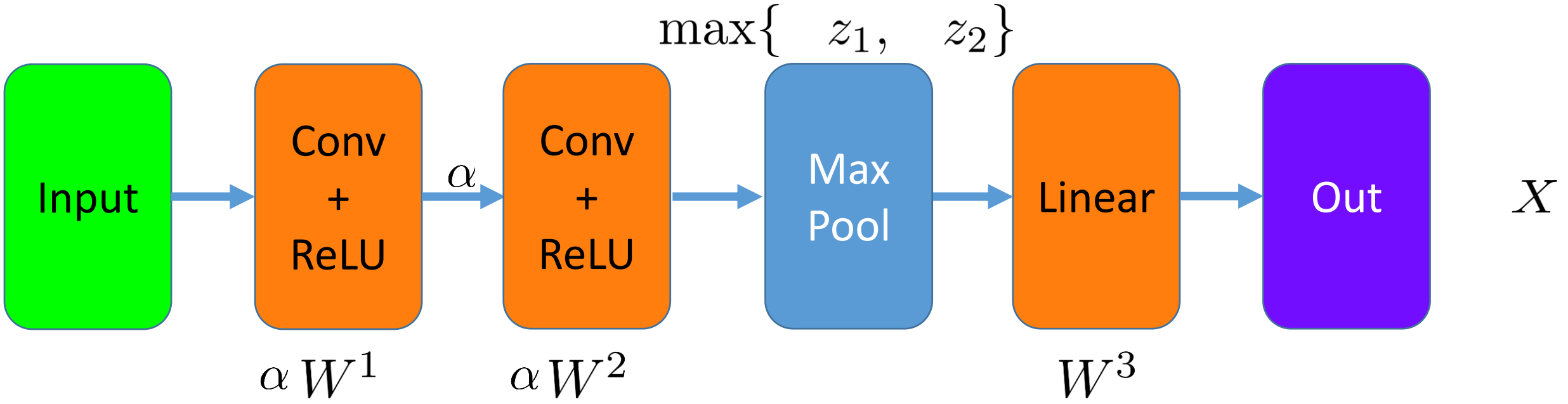
Most Modern Networks Are Positively Homogeneous

- **Simple Network**



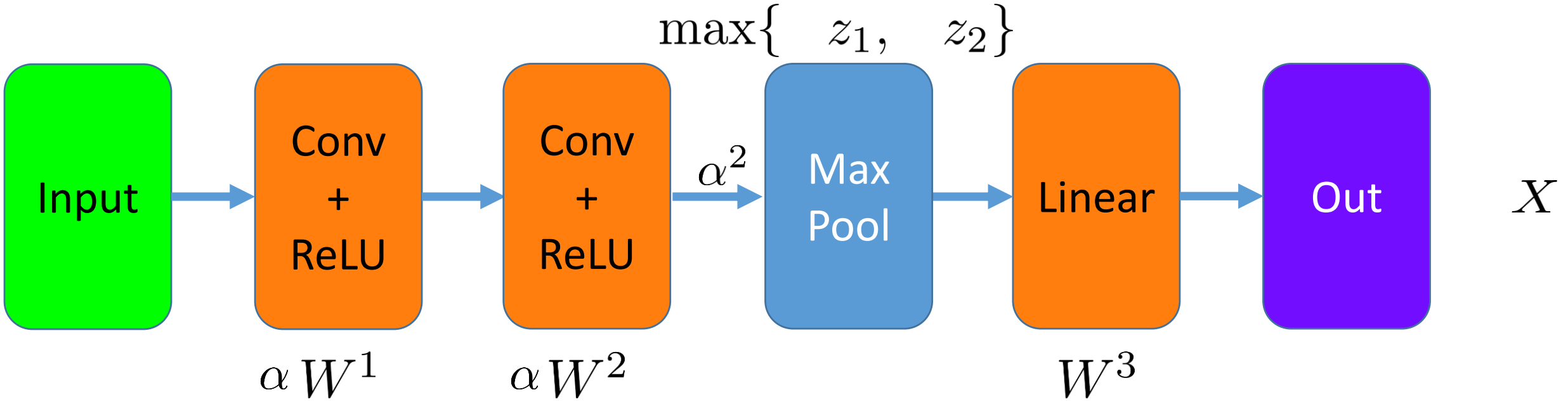
Most Modern Networks Are Positively Homogeneous

- **Simple Network**



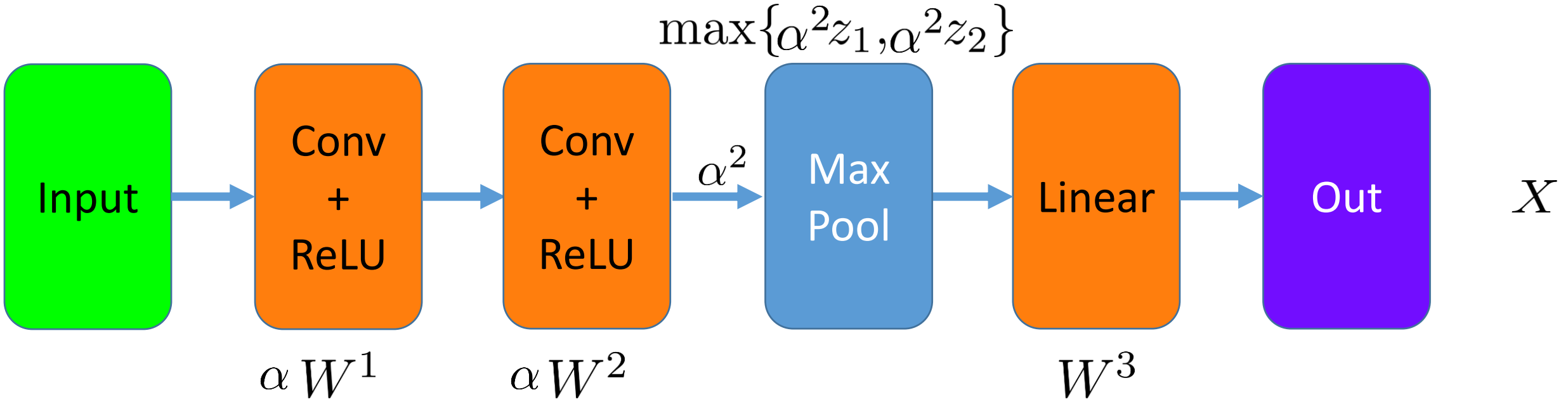
Most Modern Networks Are Positively Homogeneous

- **Simple Network**



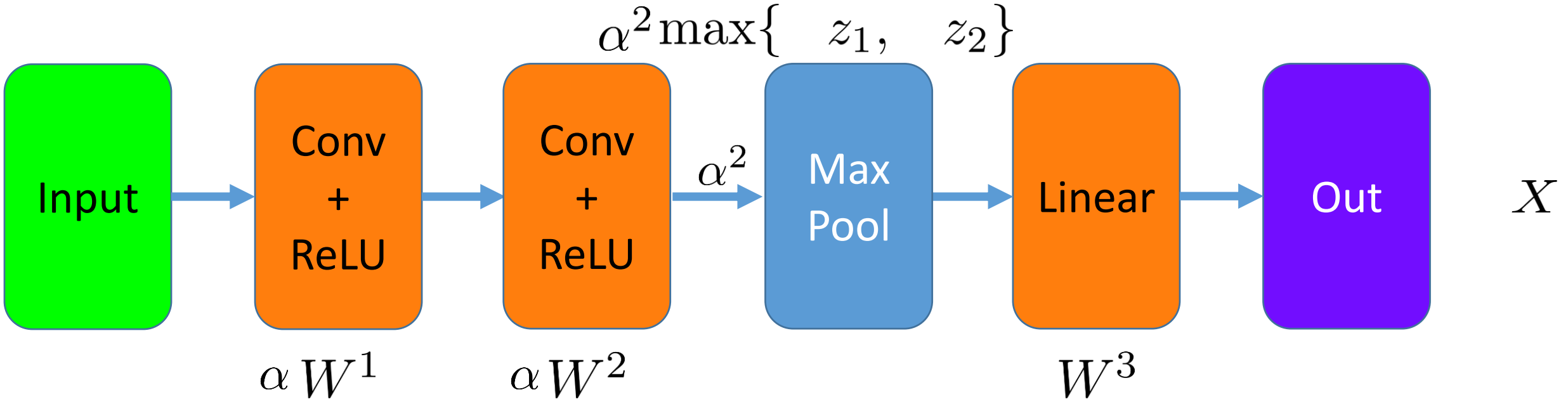
Most Modern Networks Are Positively Homogeneous

- **Simple Network**



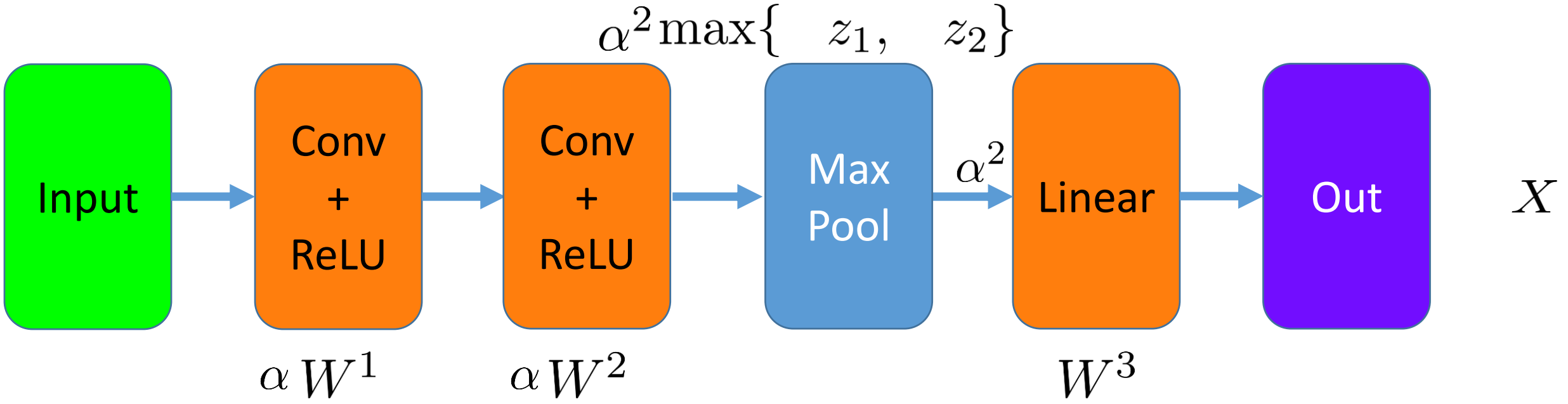
Most Modern Networks Are Positively Homogeneous

- **Simple Network**



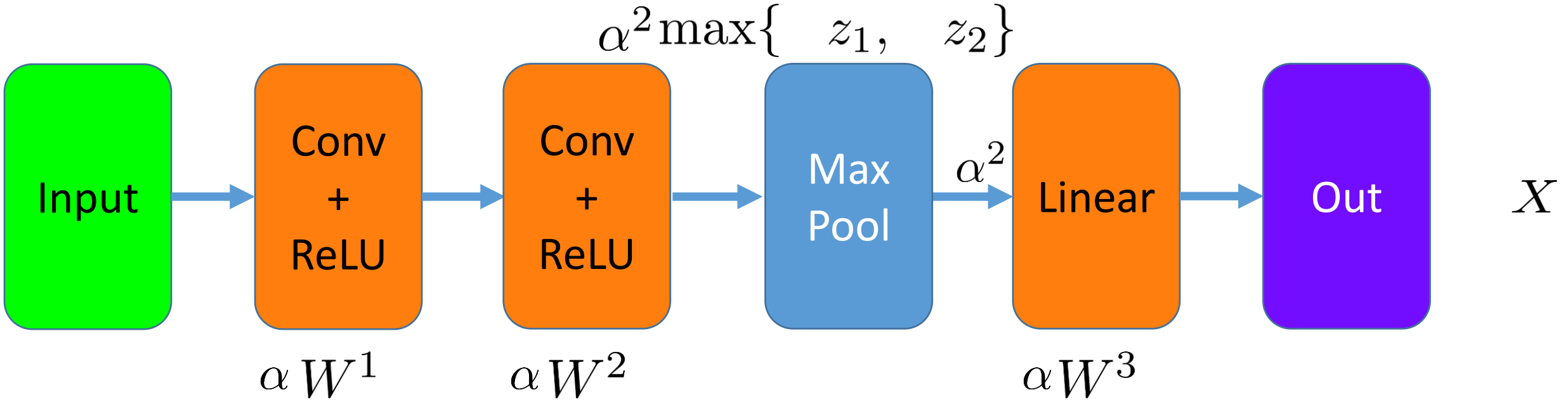
Most Modern Networks Are Positively Homogeneous

- **Simple Network**



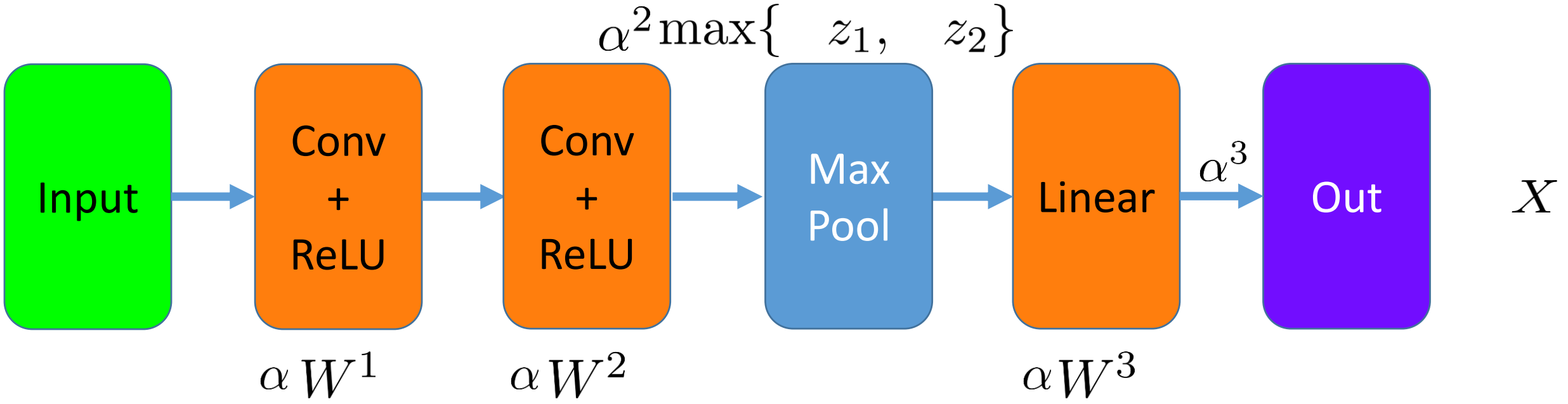
Most Modern Networks Are Positively Homogeneous

- **Simple Network**



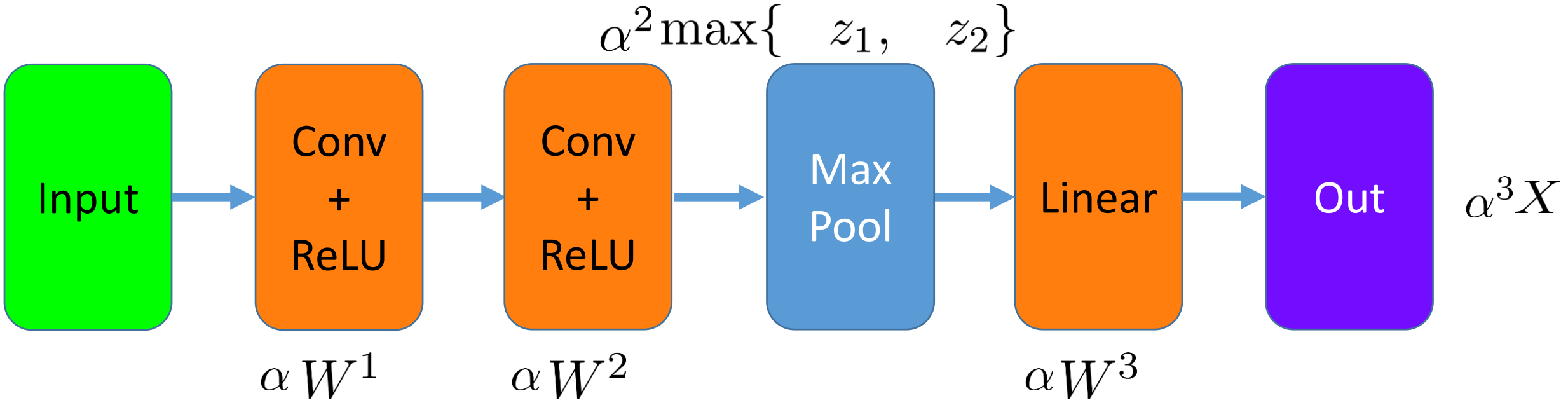
Most Modern Networks Are Positively Homogeneous

- **Simple Network**



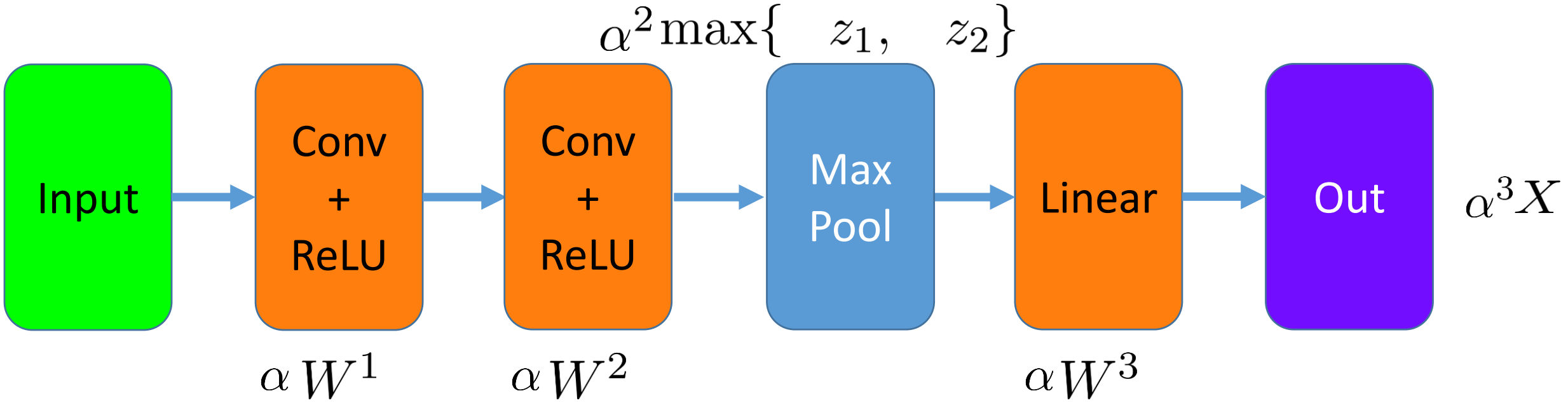
Most Modern Networks Are Positively Homogeneous

- **Simple Network**



Most Modern Networks Are Positively Homogeneous

- **Simple Network**

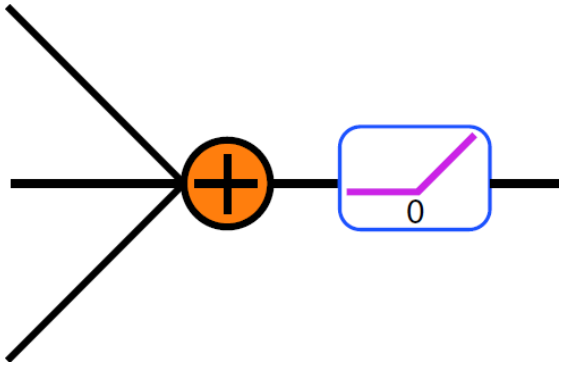


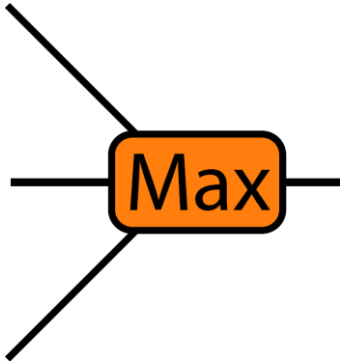
- Typically each weight layer increases degree of homogeneity by 1.

Most Modern Networks Are Positively Homogeneous

Some Common Positively Homogeneous Layers

- ✓Fully Connected + ReLU
- ✓Convolution + ReLU
- ✓Max Pooling
- ✓Linear Layers
- ✓Mean Pooling
- ✓Max Out
- ✓Many possibilities…

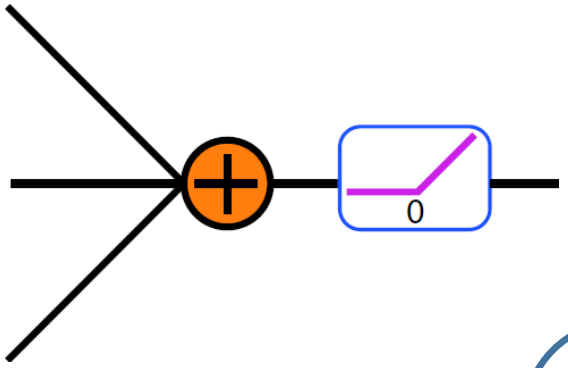


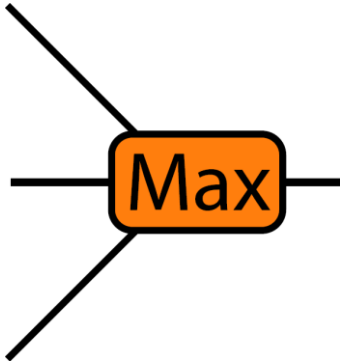


Most Modern Networks Are Positively Homogeneous

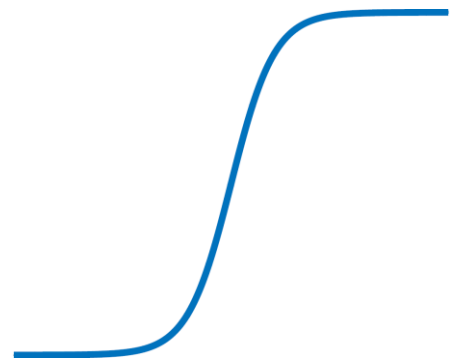
Some Common Positively Homogeneous Layers

- ✓Fully Connected + ReLU
- ✓Convolution + ReLU
- ✓Max Pooling
- ✓Linear Layers
- ✓Mean Pooling
- ✓Max Out
- ✓Many possibilities…

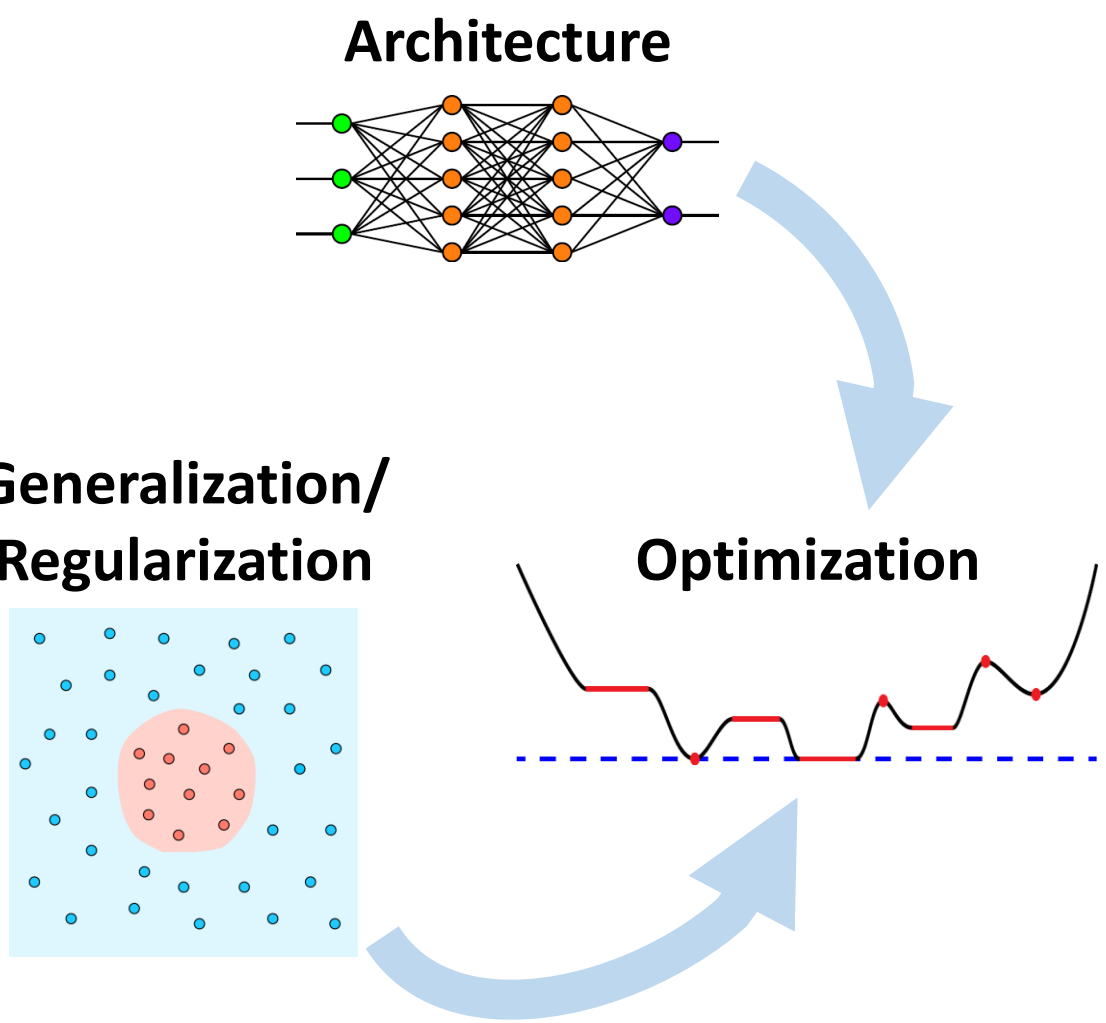




χ **Not Sigmoids**



Outline



1. **Network properties that allow efficient optimization**
   - Positive Homogeneity
   - **Parallel Subnetwork Structure**
2. Network regularization
3. Theoretical guarantees
   - Sufficient conditions for global optimality
   - Local descent can reach global minimizers

Key Property 2: Parallel Subnetworks

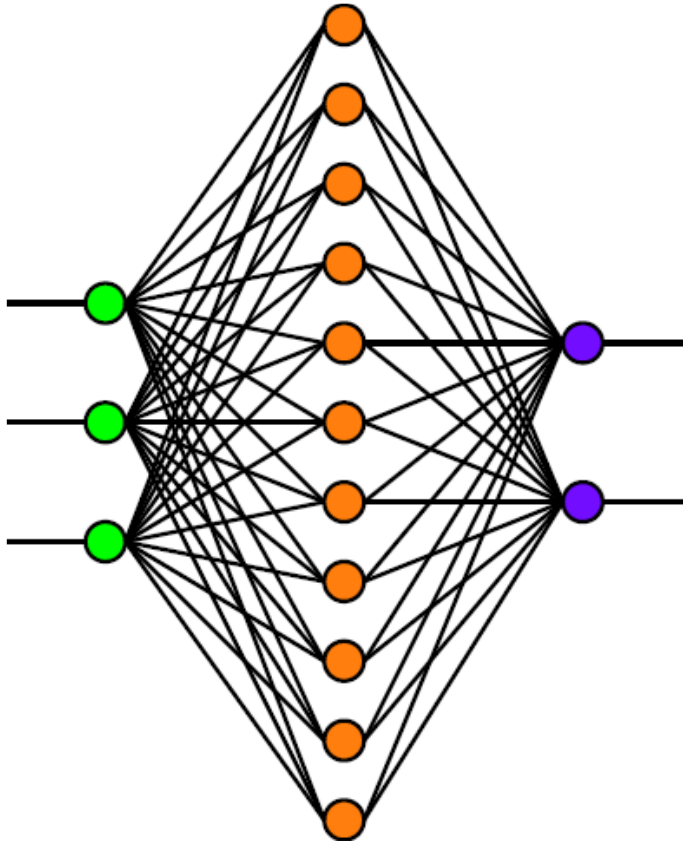
- **Subnetworks with identical architecture connected in parallel.**

Key Property 2: Parallel Subnetworks

- **Subnetworks with identical architecture connected in parallel.**
- **Simple Example:** Single hidden layer network

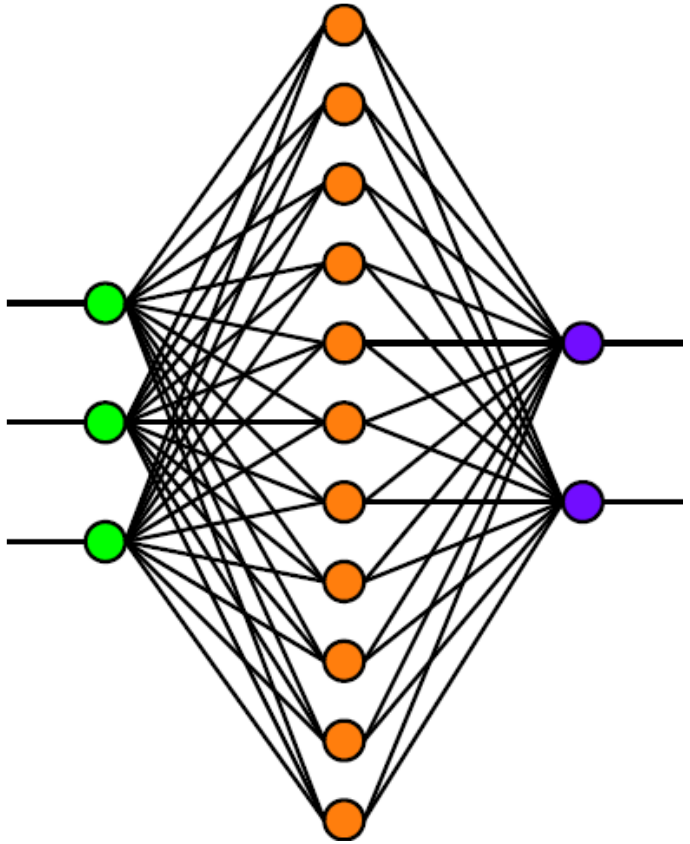
Key Property 2: Parallel Subnetworks

- **Subnetworks with identical architecture connected in parallel.**
- **Simple Example:** Single hidden layer network

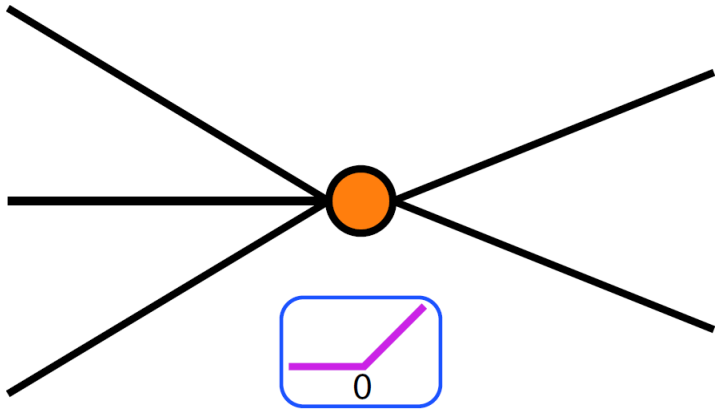


Key Property 2: Parallel Subnetworks

- **Subnetworks with identical architecture connected in parallel.**
- **Simple Example:** Single hidden layer network

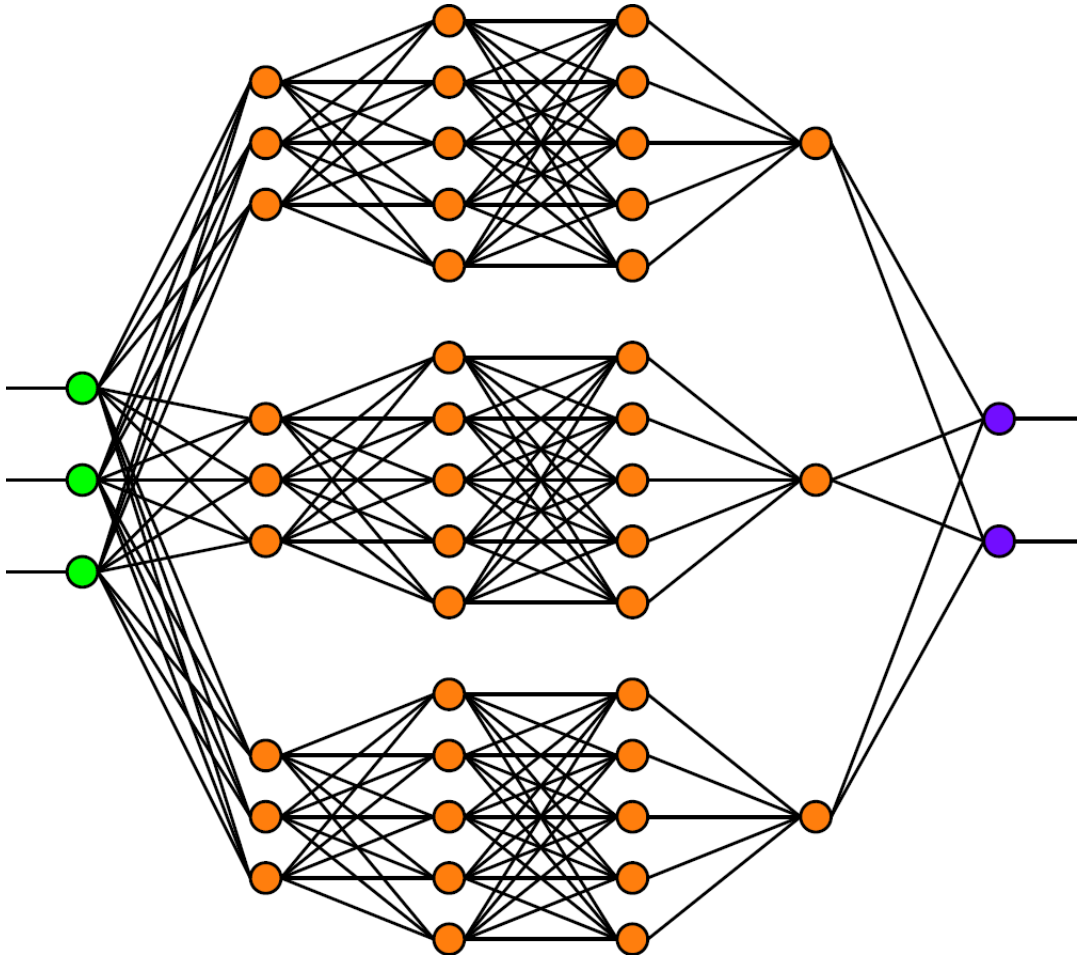


- **Subnetwork:** One ReLU hidden unit

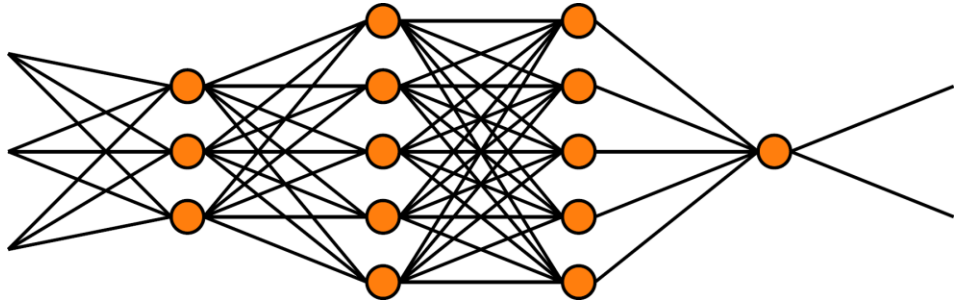


Key Property 2: Parallel Subnetworks

- Any positively homogeneous subnetwork can be used

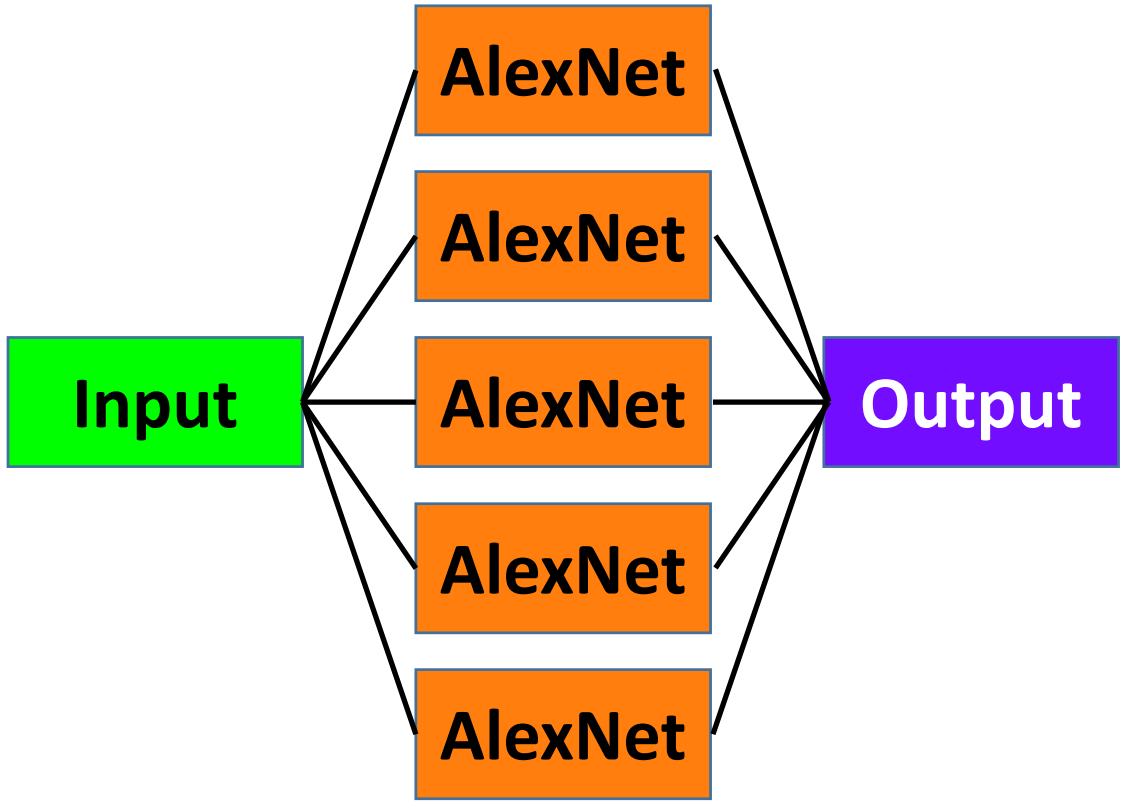


- **Subnetwork:** Multiple ReLU layers

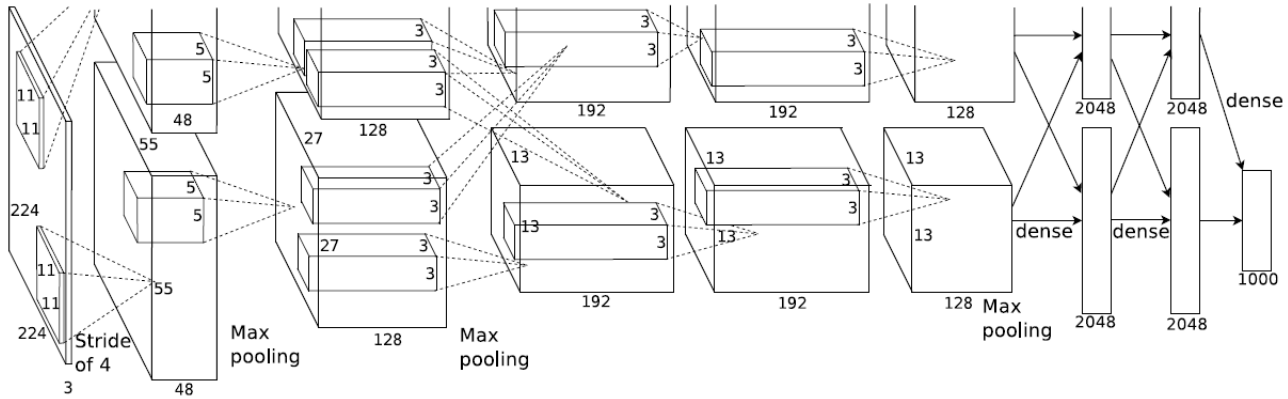


Key Property 2: Parallel Subnetworks

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



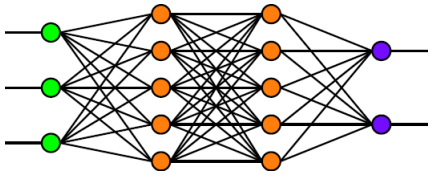
- **Subnetwork:** AlexNet



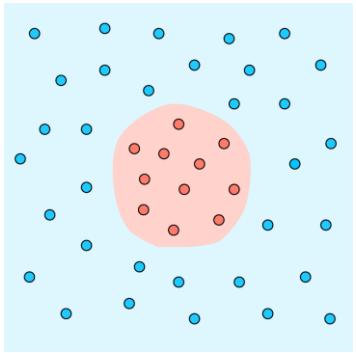
[1] Krizhevsky, Sutskever, and Hinton. "Imagenet classification with deep convolutional neural networks." NIPS, 2012.

Outline

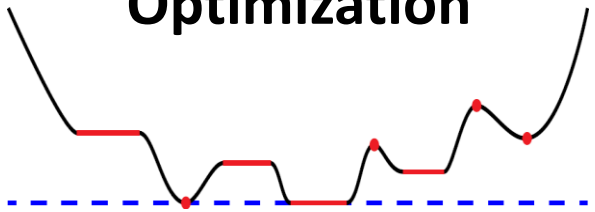
Architecture



Generalization/ Regularization



Optimization

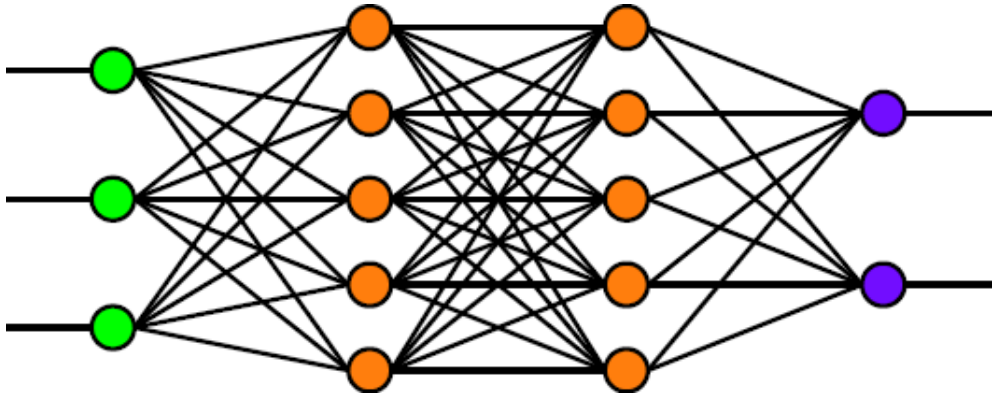


1. Network properties that allow efficient optimization
   - Positive Homogeneity
   - Parallel Subnetwork Structure

2. **Network regularization**

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

Basic Regularization: Weight Decay

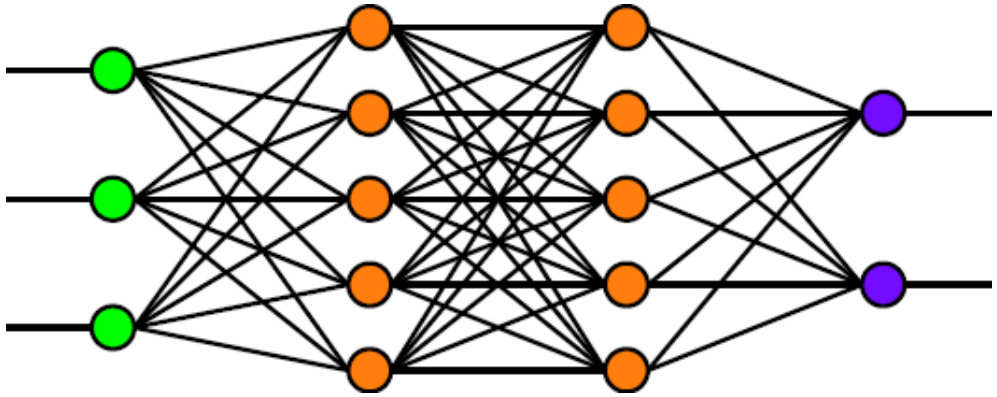


W^1 W^2 W^3

Network Weights

Basic Regularization: Weight Decay

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

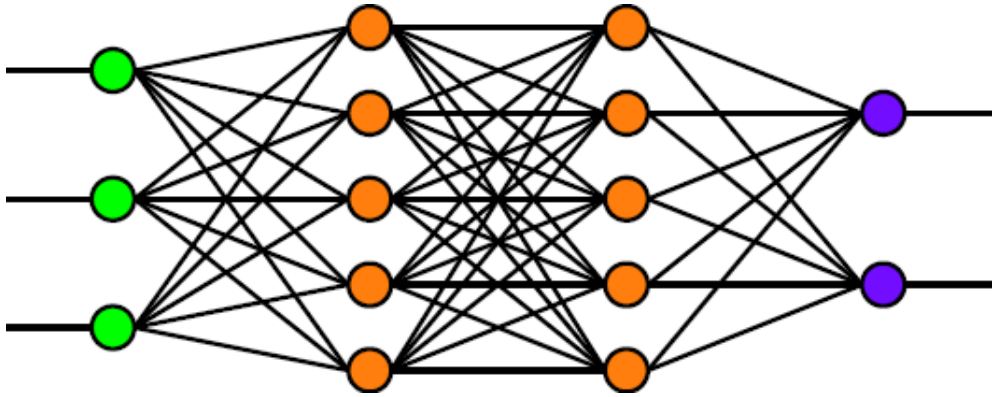


W^1 W^2 W^3

Network Weights

Basic Regularization: Weight Decay

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



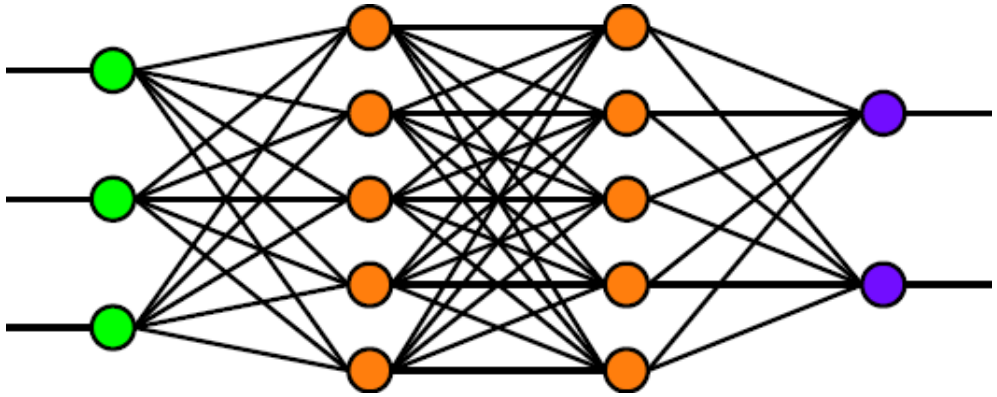
W^1 W^2 W^3

Network Weights

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

Basic Regularization: Weight Decay

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



W^1 W^2 W^3

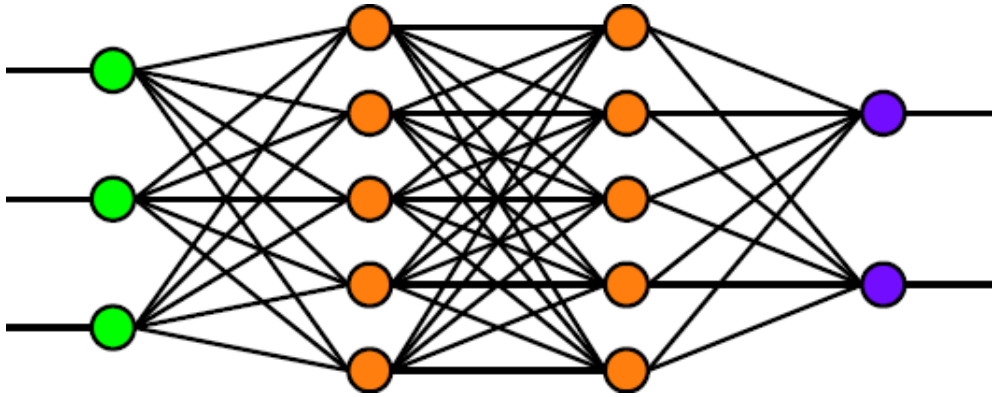
Network Weights

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

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

Basic Regularization: Weight Decay

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



W^1 W^2 W^3

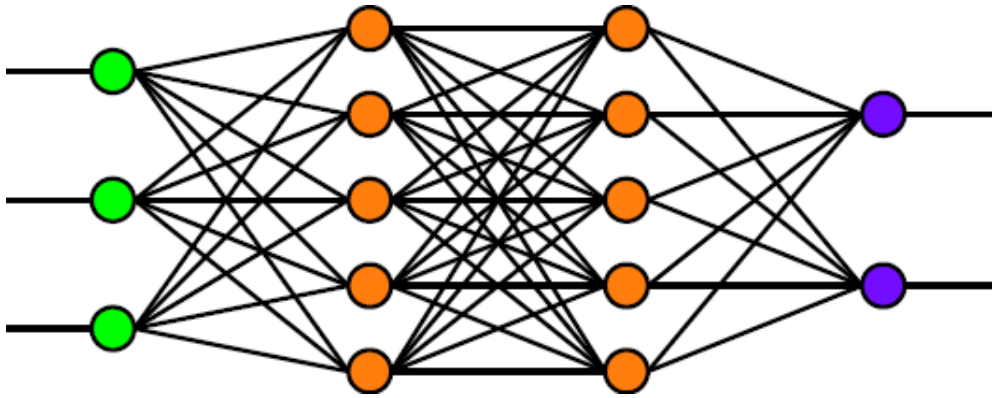
Network Weights

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

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

Basic Regularization: Weight Decay

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



W^1 W^2 W^3

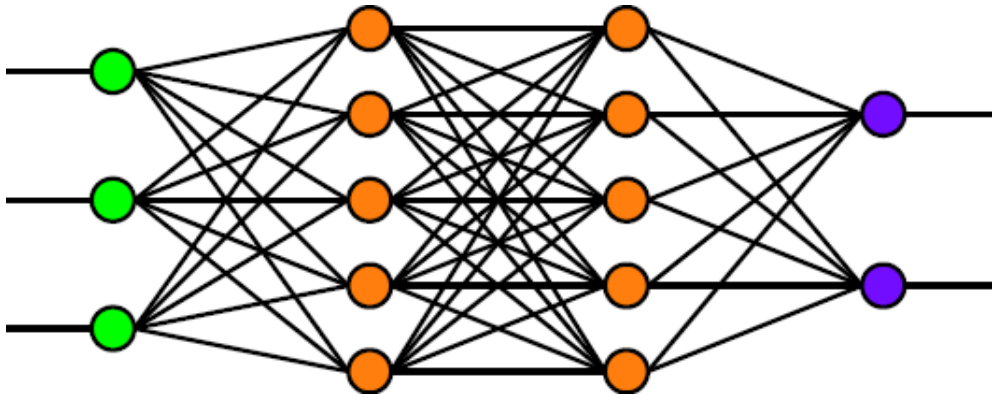
Network Weights

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

Degrees of positive homogeneity don't match = Bad things happen.

Basic Regularization: Weight Decay

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



W^1 W^2 W^3

Network Weights

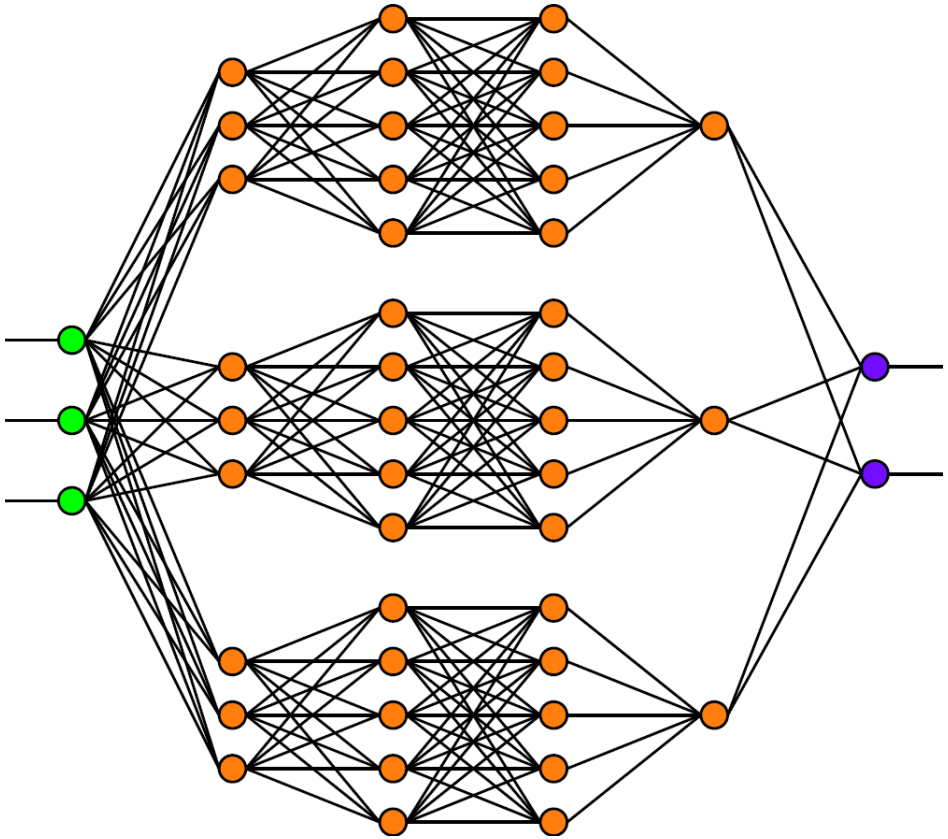
$$\Theta(\alpha W^1, \alpha W^2, \alpha W^3) = \alpha^2 \Theta(W^1, W^2, W^3)$$
$$\Phi(\alpha W^1, \alpha W^2, \alpha W^3) = \alpha^3 \Phi(W^1, W^2, W^3)$$

Degrees of positive homogeneity don't match = Bad things happen.

Proposition: There will always exist non-optimal local minima.

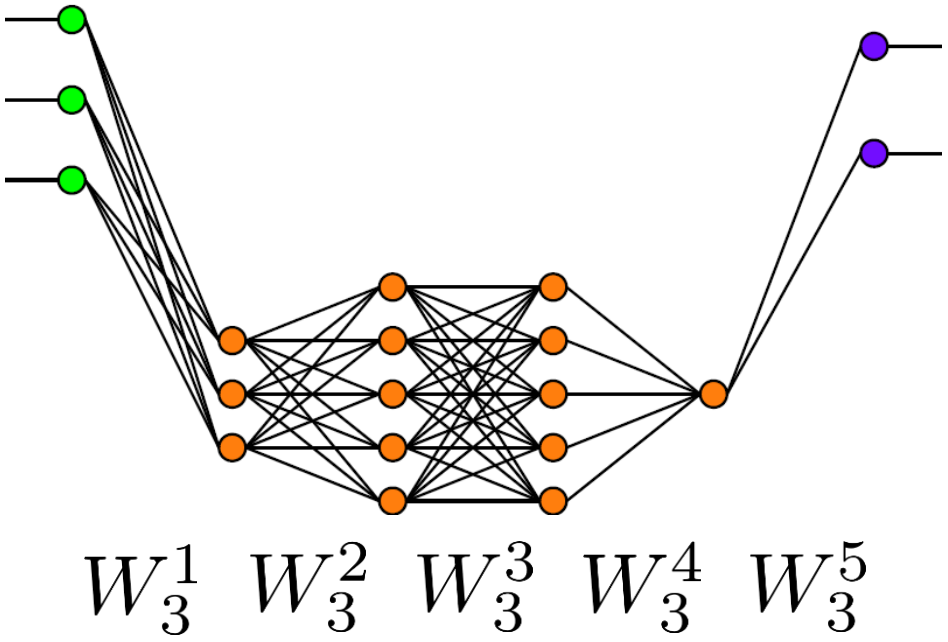
Adapting the size of the network via regularization

- Start with a positively homogeneous network with parallel structure



Adapting the size of the network via regularization

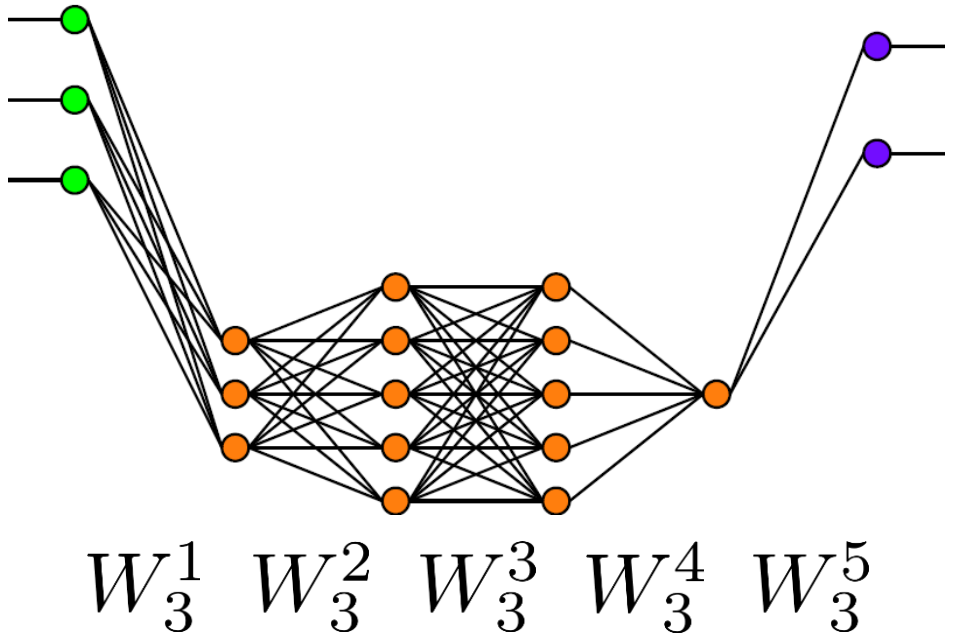
- Take the weights of one subnetwork.



Adapting the size of the network via regularization

- Define a regularization function on the weights.

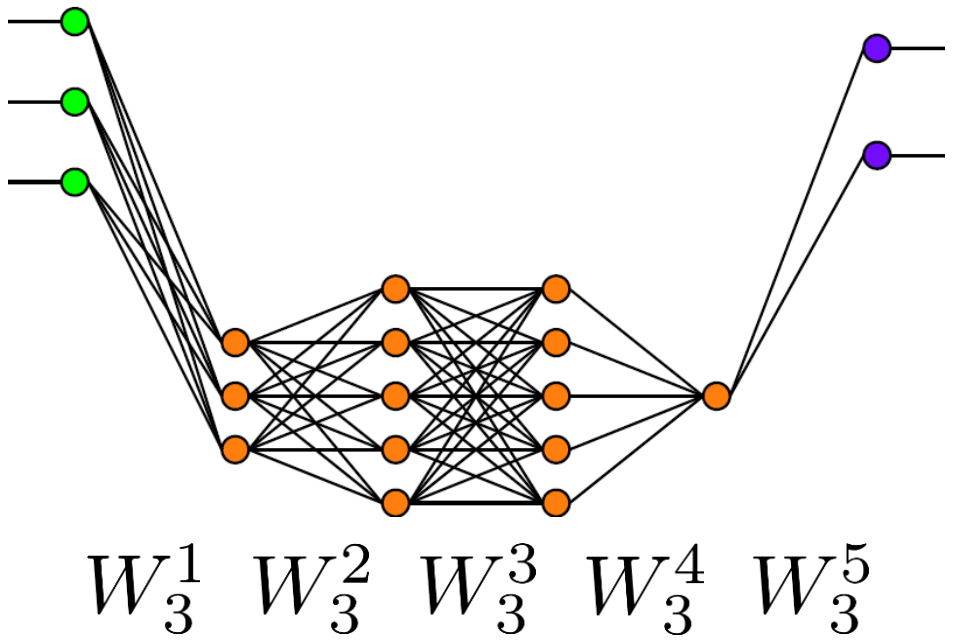
$$\theta(W_3^1, W_3^2, W_3^3, W_3^4, W_3^5)$$



Adapting the size of the network via regularization

- Define a regularization function on the weights.

$$\theta(W_3^1, W_3^2, W_3^3, W_3^4, W_3^5)$$

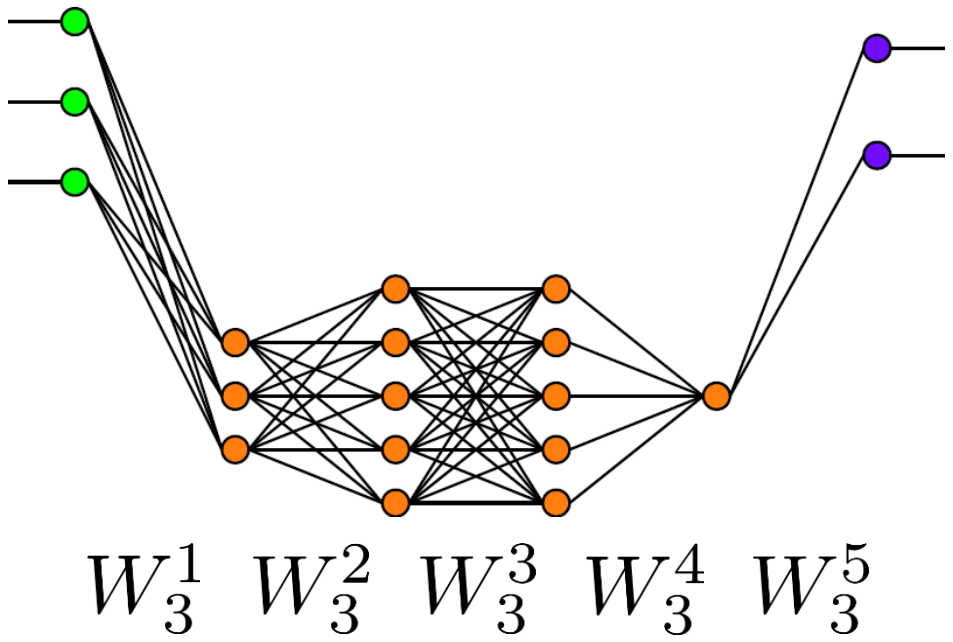


- Non-negative.
- Positively homogeneous with same degree as network mapping.

Adapting the size of the network via regularization

- Define a regularization function on the weights.

$$\theta(W_3^1, W_3^2, W_3^3, W_3^4, W_3^5)$$



- Non-negative.
- Positively homogeneous with same degree as network mapping.

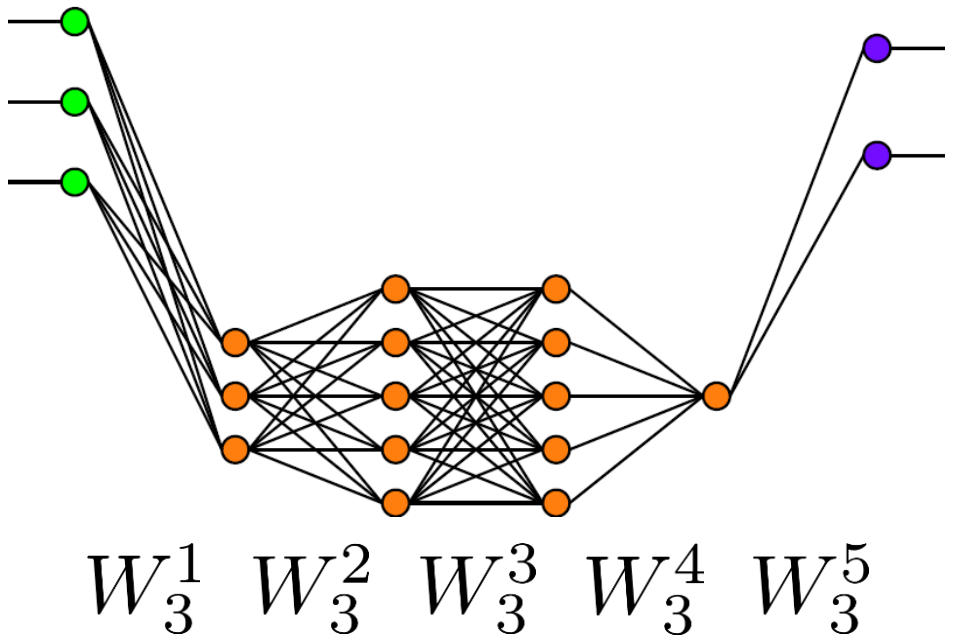
$$\Phi(\alpha W) = \alpha^p \Phi(W)$$

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

Adapting the size of the network via regularization

- Define a regularization function on the weights.

$$\theta(W_3^1, W_3^2, W_3^3, W_3^4, W_3^5)$$



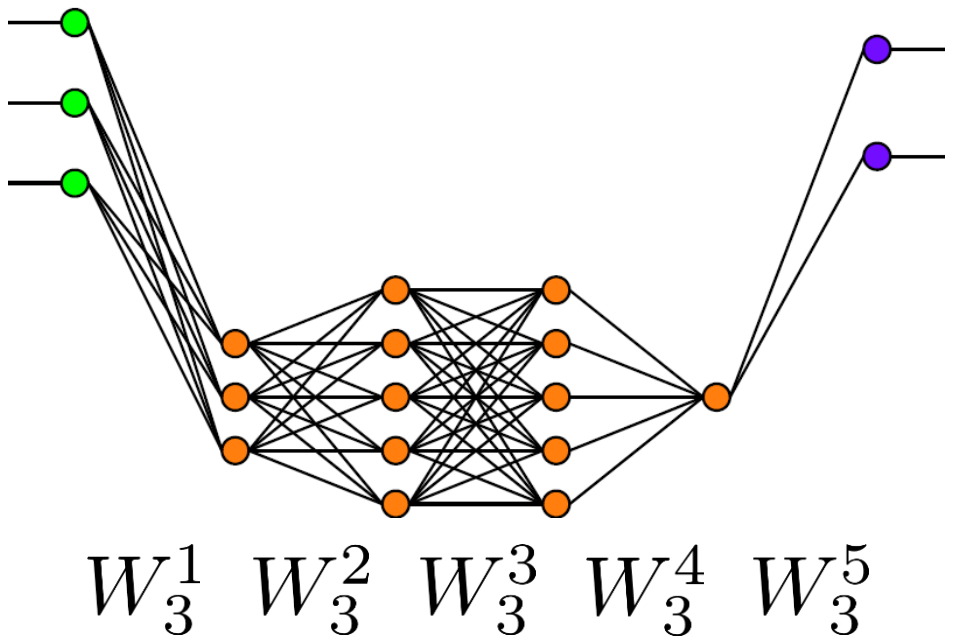
- Non-negative.
- Positively homogeneous with same degree as network mapping.

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

Adapting the size of the network via regularization

- Define a regularization function on the weights.

$$\theta(W_3^1, W_3^2, W_3^3, W_3^4, W_3^5)$$



- Non-negative.
- Positively homogeneous with same degree as network mapping.

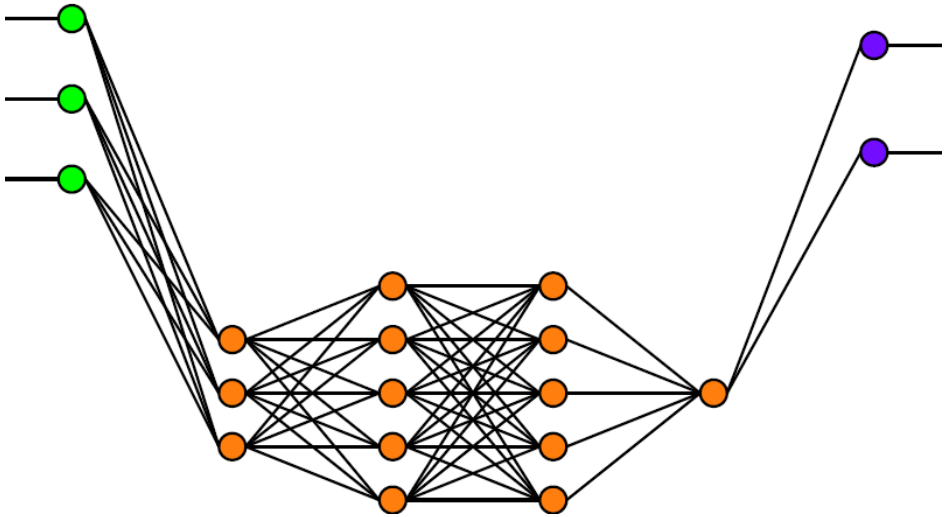
Example: Product of norms

$$\|W_3^1\|\|W_3^2\|\|W_3^3\|\|W_3^4\|\|W_3^5\|$$

Adapting the size of the network via regularization

- Sum over all the subnetworks.

$$\theta(\text{Subnetwork } 3)$$



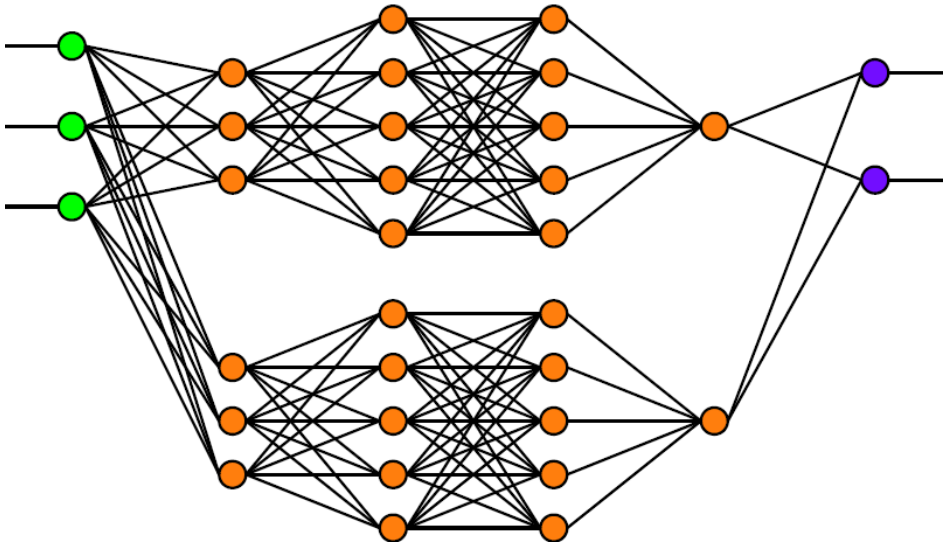
Subnetwork 3

Adapting the size of the network via regularization

- Sum over all the subnetworks.

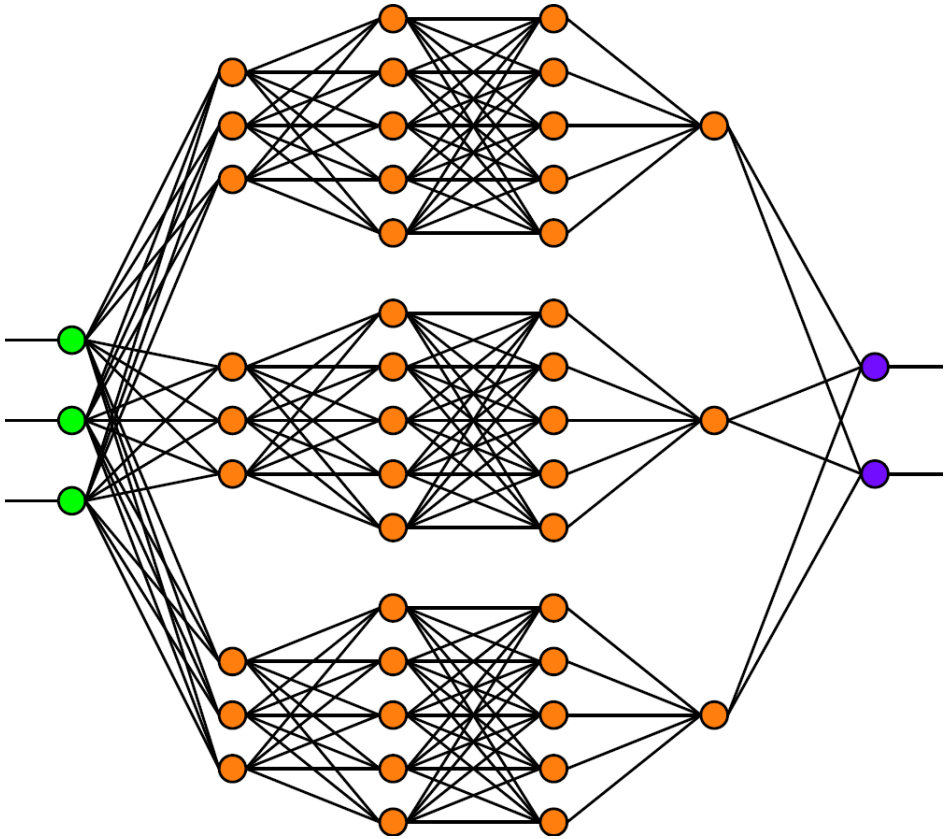
$$\theta(\text{Subnetwork } 3)+$$
$$\theta(\text{Subnetwork } 2)$$

Subnetwork 2



Adapting the size of the network via regularization

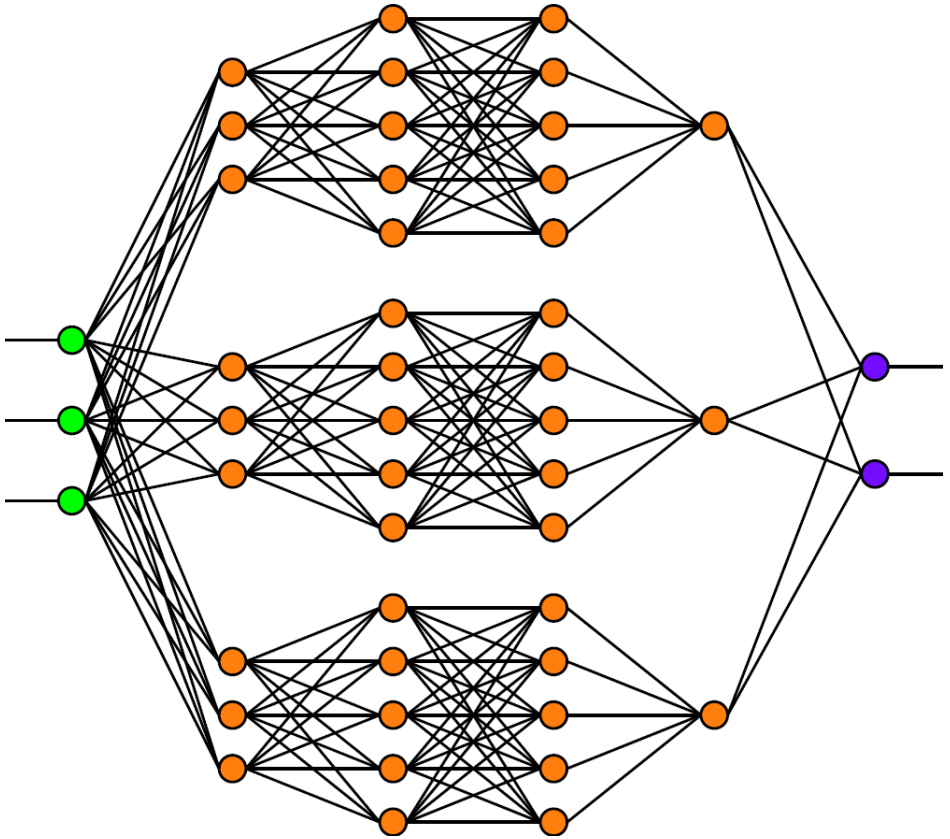
- Sum over all the subnetworks.



$$\theta(\text{Subnetwork } 3)+$$
$$\theta(\text{Subnetwork } 2)+$$
$$\theta(\text{Subnetwork } 1)$$

Adapting the size of the network via regularization

- Sum over all the subnetworks.



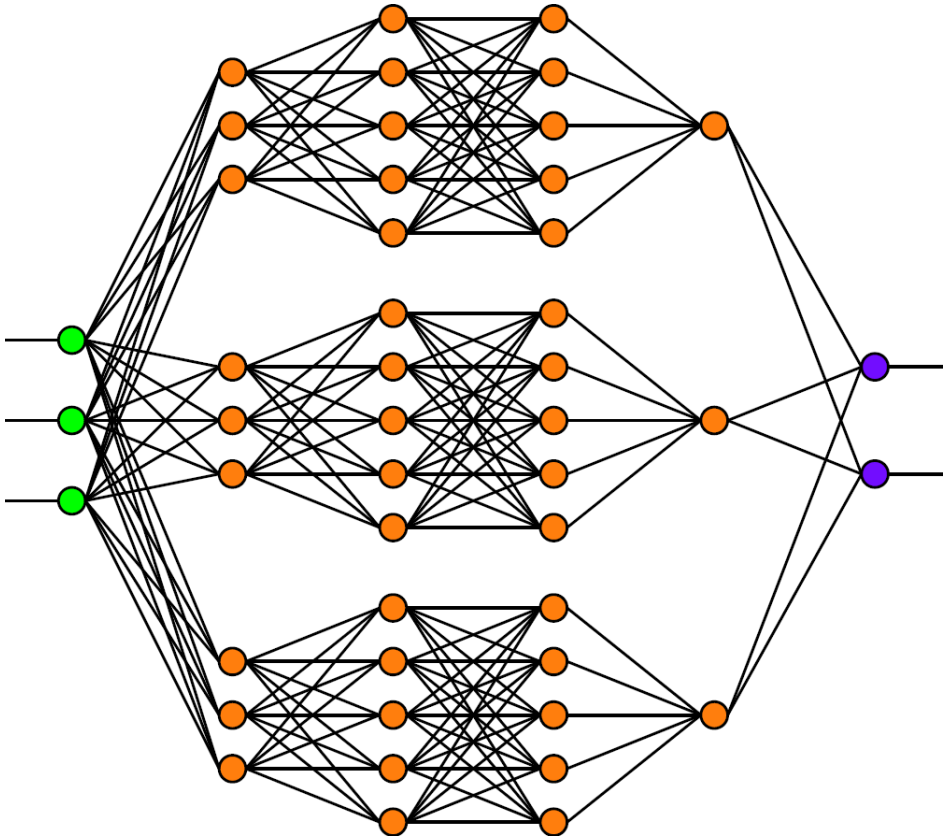
$$\theta(\text{Subnetwork } 3)+$$
$$\theta(\text{Subnetwork } 2)+$$
$$\theta(\text{Subnetwork } 1)$$

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

r = # of Subnetworks

Adapting the size of the network via regularization

- Allow the number of subnetworks to vary.

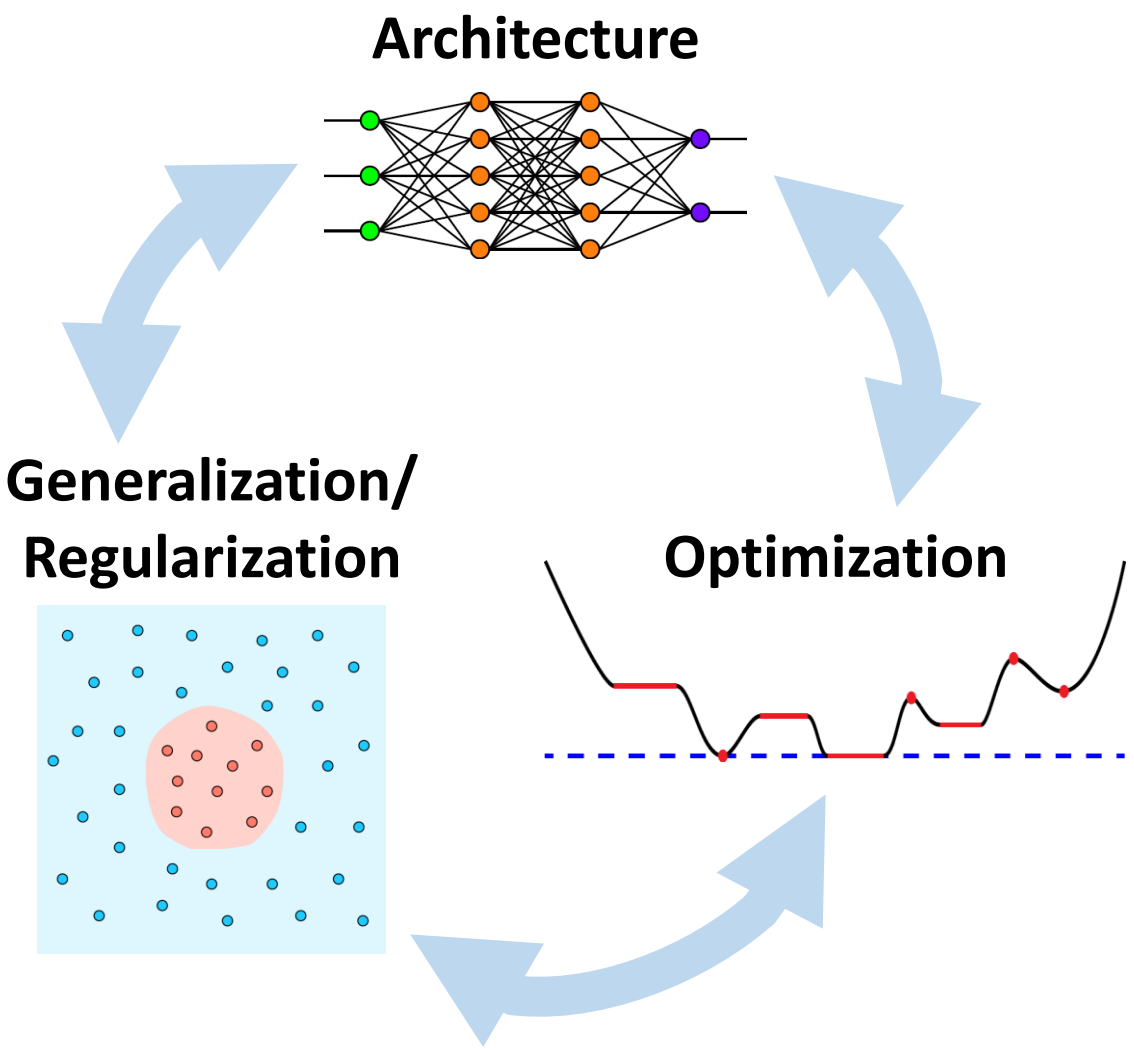


$$\Theta(W) = \sum_{i=1}^{r} \theta(\text{Subnetwork } i)$$

r = # of Subnetworks

- Adding a subnetwork is penalized by an additional term in the sum.
- Acts to constrain the number of subnetworks.

Outline



1. Network properties that allow efficient optimization
   - Positive Homogeneity
   - Parallel Subnetwork Structure

2. Network regularization

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

Our problem

Our problem

- The non-convex problem we're interested in

$$f(W) \equiv \ell(Y, \Phi(W)) + \lambda\Theta(W)$$

Our problem

- The non-convex problem we're interested in

$$f(W) \equiv \ell(Y, \Phi(W)) + \lambda \Theta(W)$$

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

Our problem

- The non-convex problem we're interested in

$$f(W) \equiv \ell(Y, \Phi(W)) + \lambda\Theta(W)$$

Labels

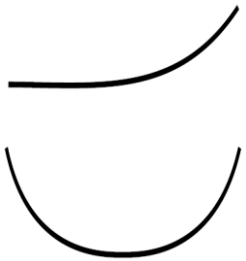
$$\Theta(W) = \sum_{i=1}^{r} \theta(\text{Subnetwork } i)$$

Loss Function: $\ell(Y, X)$

Assume convex and once differentiable in X

Examples:

- Cross-entropy
- Least-squares



Why do all this?

Why do all this?

- Induces a convex function on the network outputs.

$$F(X) \equiv \ell(Y, X) + \lambda\Omega(X)$$

Why do all this?

- Induces a convex function on the network outputs.

$$F(X) \equiv \ell(Y, X) + \lambda\Omega(X)$$

Induced Function: $\Omega(X)$

Comes from the regularization

Why do all this?

- Induces a convex function on the network outputs.

$$F(X) \equiv \ell(Y, X) + \lambda\Omega(X)$$

Induced Function: $\Omega(X)$

Comes from the regularization

$$\Omega(X) = \min_{W} \Theta(W) \quad \text{subject to} \quad X = \Phi(W)$$

Why do all this?

- Induces a convex function on the network outputs.

$$F(X) \equiv \ell(Y, X) + \lambda\Omega(X)$$

Induced Function: $\Omega(X)$

Comes from the regularization

$$\Omega(X) = \min_{W} \Theta(W) \quad \text{subject to} \quad X = \Phi(W)$$

$$\Omega(X) \leq \Theta(W) \qquad \forall X = \Phi(W)$$

Why do all this?

- Induces a convex function on the network outputs.

$$F(X) \equiv \ell(Y, X) + \lambda\Omega(X)$$

Induced Function: $\Omega(X)$

Comes from the regularization

$$\Omega(X) = \min_{W} \Theta(W) \quad \text{subject to} \quad X = \Phi(W)$$

$$\Omega(X) \leq \Theta(W) \qquad \forall X = \Phi(W)$$

Proposition: $\Omega(X)$ is a convex function.

Why do all this?

- Induces a convex function on the network outputs.

$$F(X) \equiv \ell(Y, X) + \lambda\Omega(X)$$

- The convex problem provides an achievable lower bound for the non-convex network training problem.

$$f(W) \equiv \ell(Y, \Phi(W)) + \lambda\Theta(W)$$

$$F(X) \leq f(W) \qquad \forall X = \Phi(W)$$

Why do all this?

- Induces a convex function on the network outputs.

$$F(X) \equiv \ell(Y, X) + \lambda\Omega(X)$$

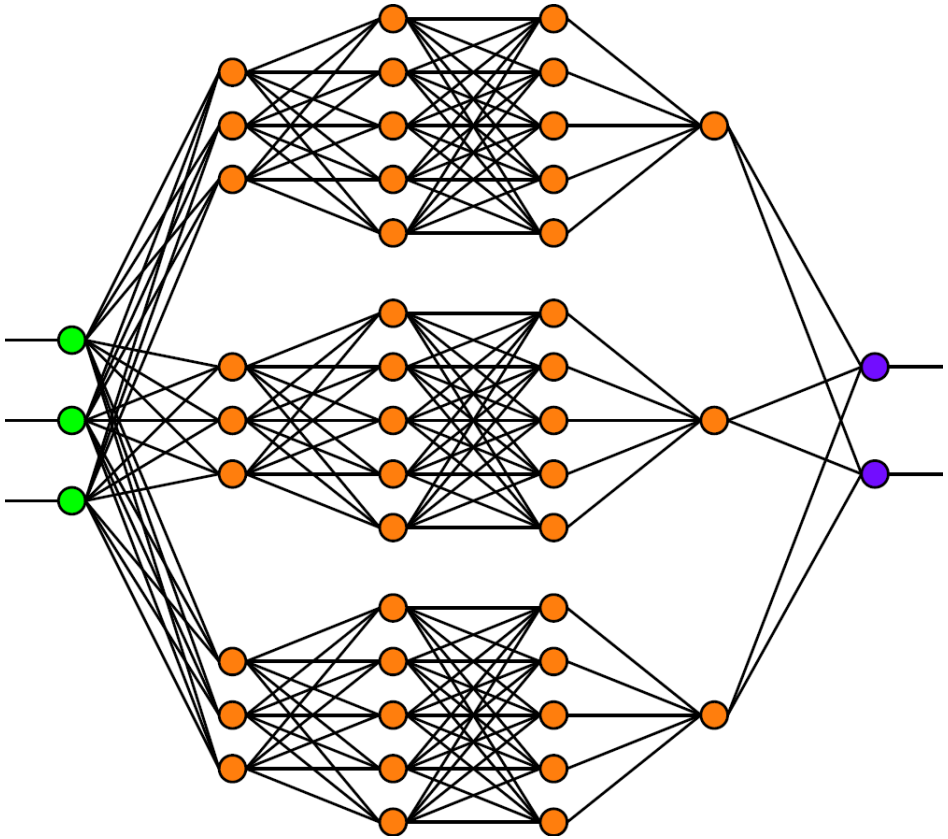
- The convex problem provides an achievable lower bound for the non-convex network training problem.

$$f(W) \equiv \ell(Y, \Phi(W)) + \lambda\Theta(W)$$

$$F(X) \leq f(W) \qquad \forall X = \Phi(W)$$

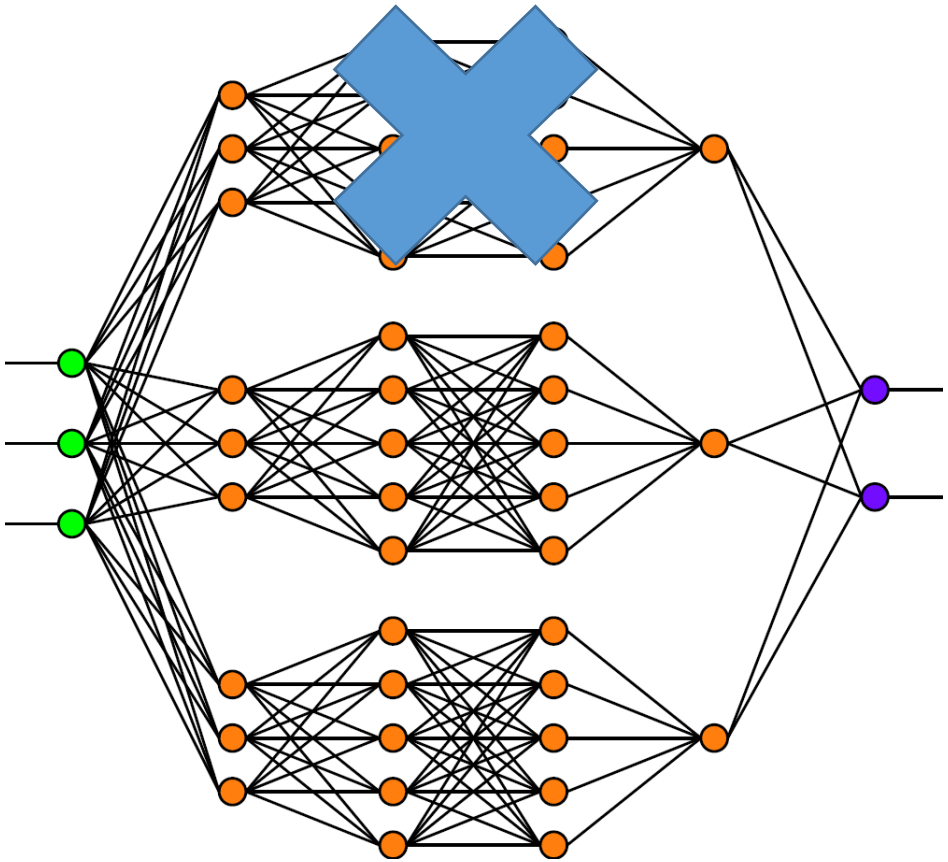
- Use the convex function as an analysis tool to study the non-convex network training problem.

Sufficient Conditions for Global Optimality



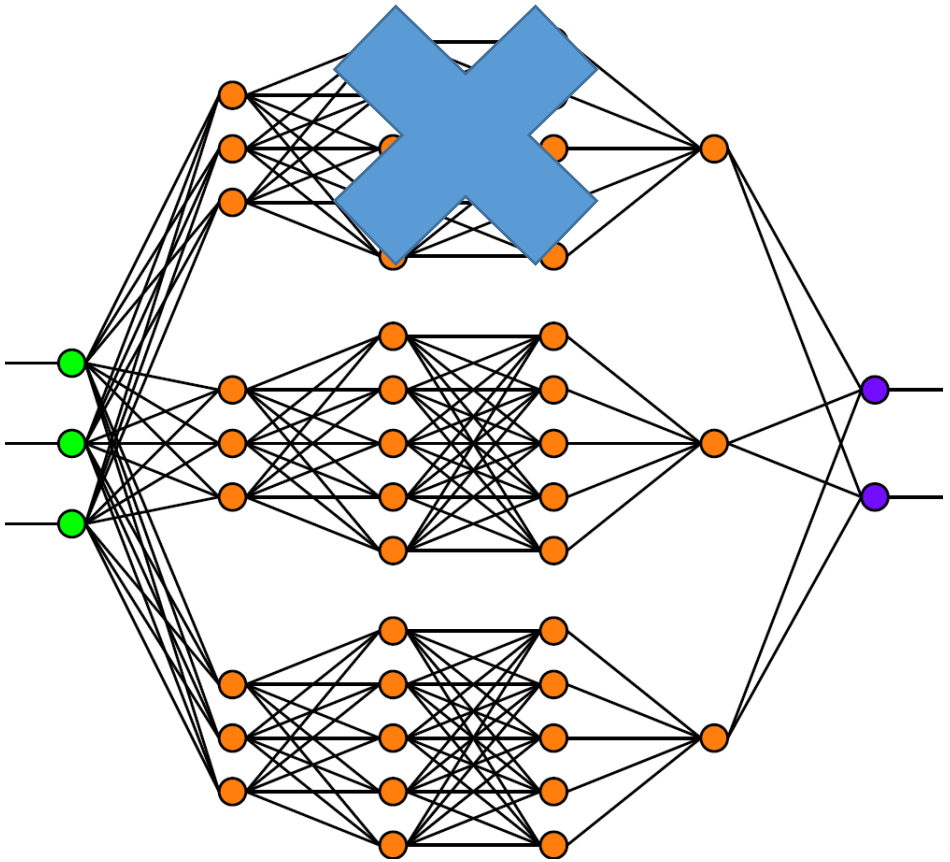
- **Theorem:** A local minimum such that one subnetwork is all zero is a global minimum.

Sufficient Conditions for Global Optimality



- **Theorem:** A local minimum such that one subnetwork is all zero is a global minimum.

Sufficient Conditions for Global Optimality



- **Theorem:** A local minimum such that one subnetwork is all zero is a global minimum.

- **Intuition:** The local minimum satisfies the optimality conditions for the convex problem.

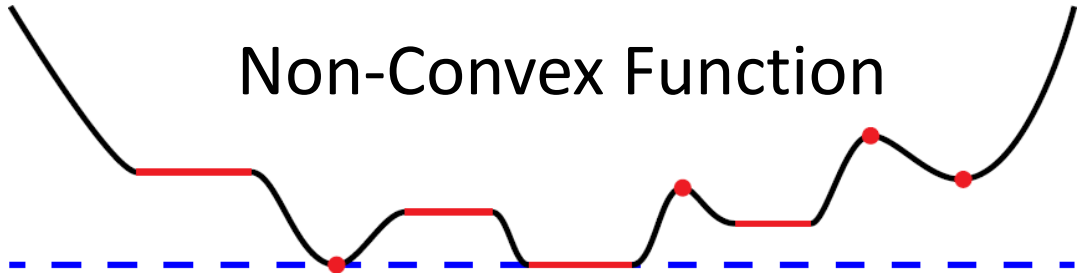
$$F(X) \leq f(W)$$

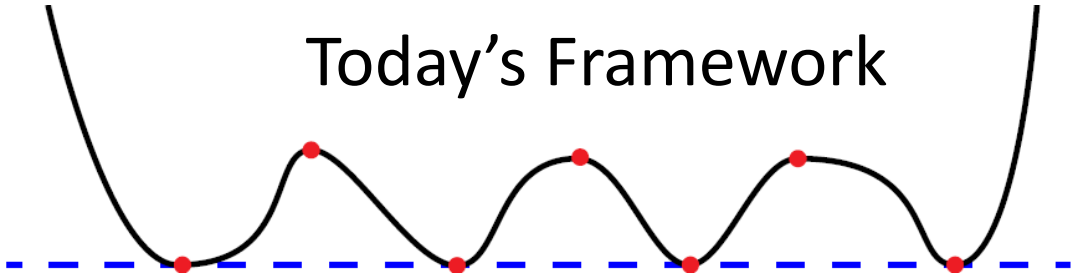
Global Minima from Local Descent

- **Theorem:** If the size of the network is large enough (has enough subnetworks), then a global minimum can always be reached by local descent from any initialization.

Global Minima from Local Descent

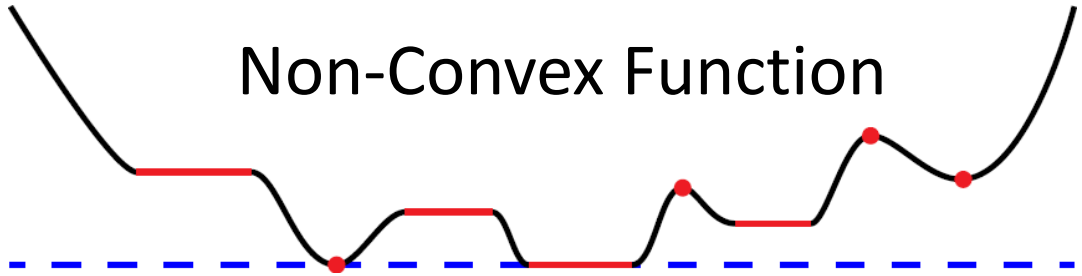
- **Theorem:** If the size of the network is large enough (has enough subnetworks), then a global minimum can always be reached by local descent from any initialization.

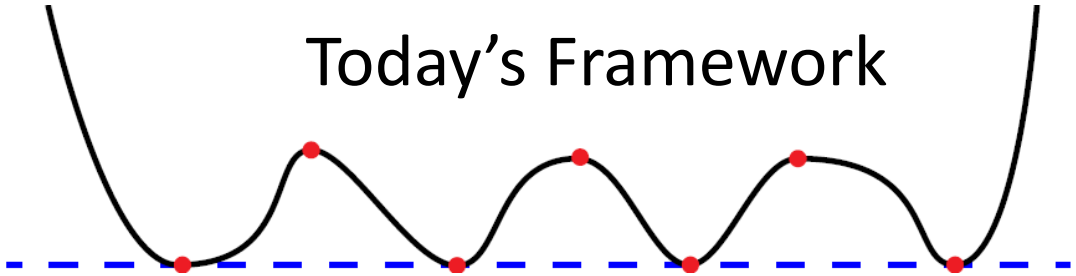




Global Minima from Local Descent

- **Theorem:** If the size of the network is large enough (has enough subnetworks), then a global minimum can always be reached by local descent from any initialization.





- Meta-Algorithm:
  - If not at a local minima, perform local descent
  - At local minima, test if first Theorem is satisfied
  - If not, add a subnetwork in parallel and continue
  - Maximum number of subnetworks guaranteed to be bounded by the dimensions of the network output

Conclusions

- Network size matters
  - Optimize network weights AND network size
  - Current: Size = Number of parallel subnetworks
  - Future: Size = Number of layers, neurons per layer, etc…
- Regularization design matters
  - Match the degrees of positive homogeneity between network and regularization
  - Regularization can control the size of the network
- Not done yet
  - Several practical and theoretical limitations

Thank You

Vision Lab @ Johns Hopkins University
http://www.vision.jhu.edu

Center for Imaging Science @ Johns Hopkins University
http://www.cis.jhu.edu

Work supported by NSF grants 1447822, 1618485 and 1618637