Understanding, Interpreting and Design Neural Network Models Through Tensor Representation

Furong Huang

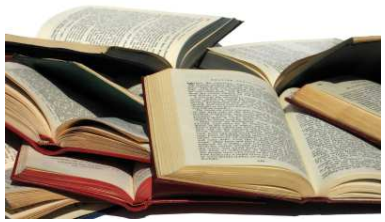
University of Maryland

TMWS4: Efficient Tensor Representations for Learning and Computational Complexity
May 18, 2021

Neural Network - Nonlinear Function Approximation

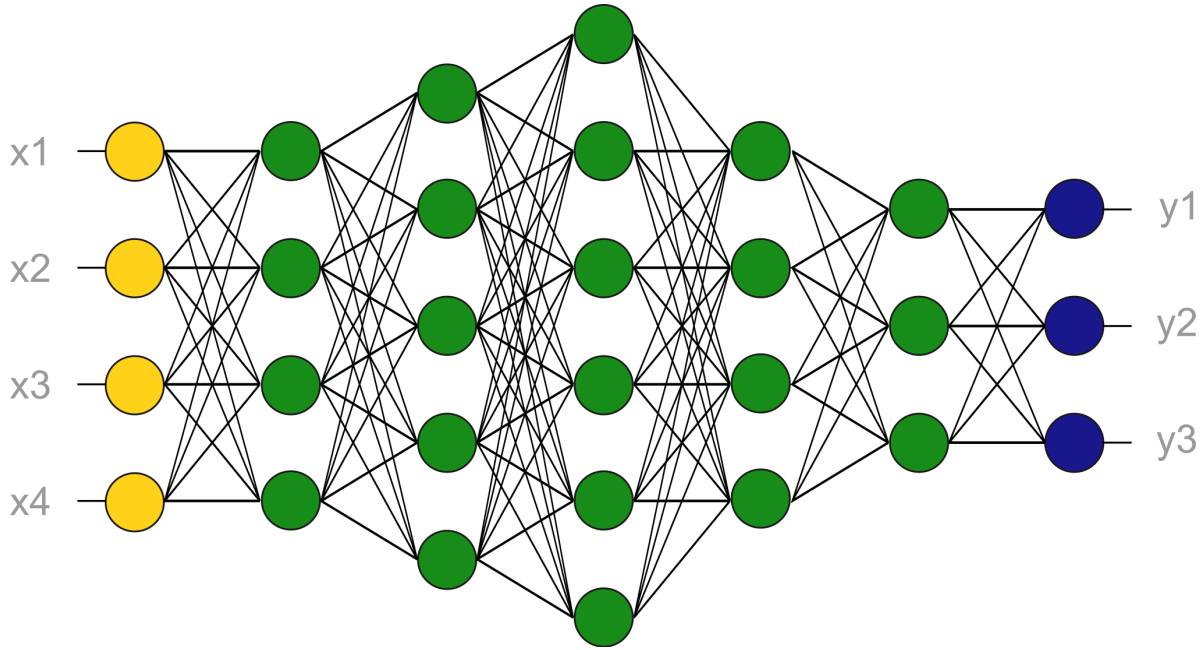
Image classification

Speech recognition



Text processing

Success of Deep Neural Networks



- computation power growth
- enormous labeled data

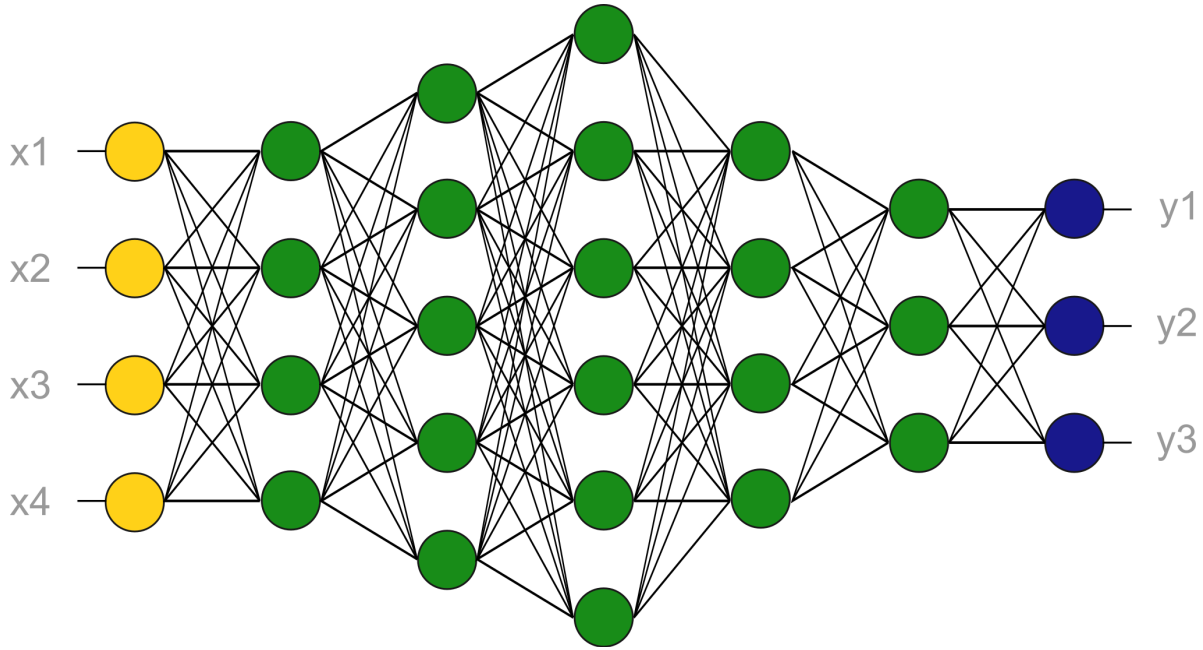
Neural Network - Nonlinear Function Approximation

Image classification

Speech recognition

Text processing

Success of Deep Neural Networks



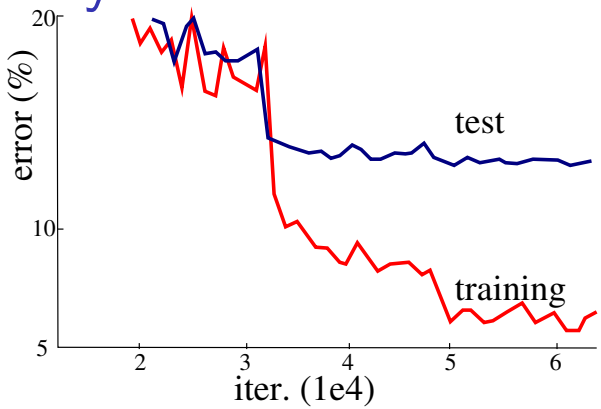
- computation power growth
- enormous labeled data

Express Power

- linear composition vs nonlinear composition
- shallow network vs deep structure

Challenge In Understanding Generalization of Deep Neural Network

Mystery of Over Parameterization: # of samples $\ll$ # of parameters

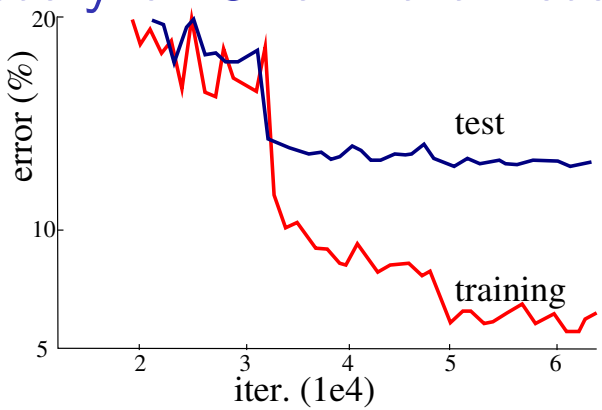


ResNet-32:6×10^4 imgs, 4.6×10^5 params

- Small (Test Error - Training Error)

Challenge In Understanding Generalization of Deep Neural Network

Mystery of Over Parameterization: # of samples $\ll$ # of parameters



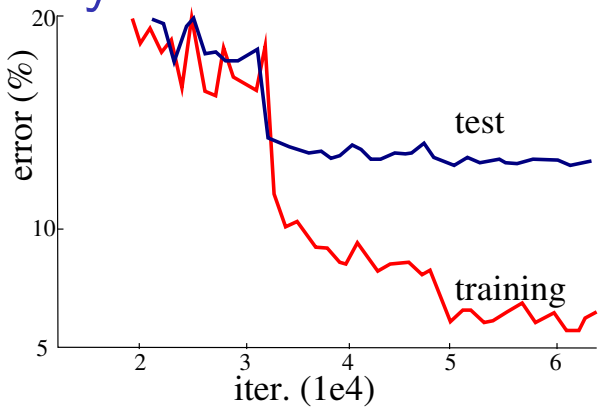
ResNet-32:6×10^4 imgs, 4.6×10^5 params

$$\underbrace{\mathbb{E}[g(x,y)]}_{\text{Test Error}} - \underbrace{\frac{1}{m}\sum_{i=1}^{m} g(x_i,y_i)}_{\text{Training Error}} \leq \underbrace{2\mathfrak{R}_m(G)}_{\text{Rademacher complexity}} + \sqrt{\frac{\log\frac{1}{\delta}}{2m}}$$

- Small (Test Error - Training Error)
- Rademacher complexity of deep nets is large

Challenge In Understanding Generalization of Deep Neural Network

Mystery of Over Parameterization: # of samples $\ll$ # of parameters



ResNet-32:6×10^4 imgs, 4.6×10^5 params

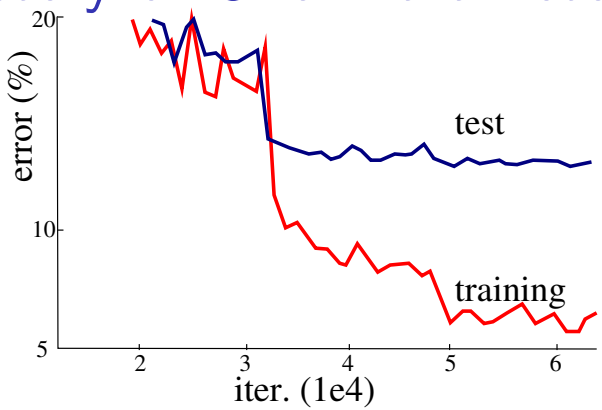
$$\underbrace{\mathbb{E}[g(x,y)]}_{\text{Test Error}} - \underbrace{\frac{1}{m}\sum_{i=1}^{m} g(x_i,y_i)}_{\text{Training Error}} \leq \underbrace{2\mathfrak{R}_m(G)}_{\text{Rademacher complexity}} + \sqrt{\frac{\log\frac{1}{\delta}}{2m}}$$

- Small (Test Error - Training Error)
- Rademacher complexity of deep nets is large

Mismatch between Empirical Observation and Learning Theory

Challenge In Understanding Generalization of Deep Neural Network

Mystery of Over Parameterization: # of samples $\ll$ # of parameters



ResNet-32:6×10^4 imgs, 4.6×10^5 params

$$\underbrace{\mathbb{E}[g(x,y)]}_{\text{Test Error}} - \underbrace{\frac{1}{m}\sum_{i=1}^{m} g(x_i,y_i)}_{\text{Training Error}} \leq \underbrace{2\mathfrak{R}_m(G)}_{\text{Rademacher complexity}} + \sqrt{\frac{\log\frac{1}{\delta}}{2m}}$$

- Small (Test Error - Training Error)
- Rademacher complexity of deep nets is large

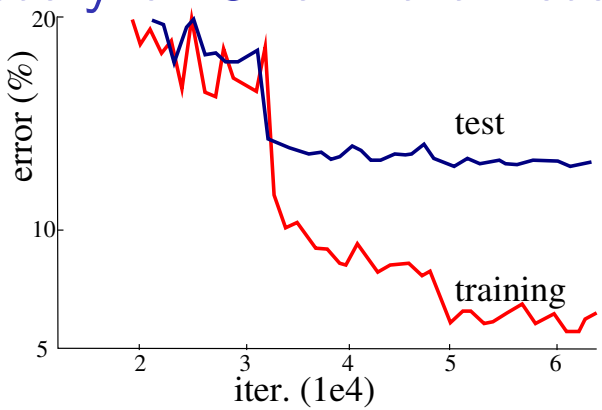
Mismatch between Empirical Observation and Learning Theory

Fitting Random Labeling of the Training Data

- Bad generalization happens: DNN easily fits a random data
- Regularization techniques does not help with generalization

Challenge In Understanding Generalization of Deep Neural Network

Mystery of Over Parameterization: # of samples $\ll$ # of parameters



ResNet-32:6×10^4 imgs, 4.6×10^5 params

$$\underbrace{\mathbb{E}[g(x,y)]}_{\text{Test Error}} - \underbrace{\frac{1}{m}\sum_{i=1}^{m} g(x_i,y_i)}_{\text{Training Error}} \leq \underbrace{2\mathfrak{R}_m(G)}_{\text{Rademacher complexity}} + \sqrt{\frac{\log\frac{1}{\delta}}{2m}}$$

- Small (Test Error - Training Error)
- Rademacher complexity of deep nets is large

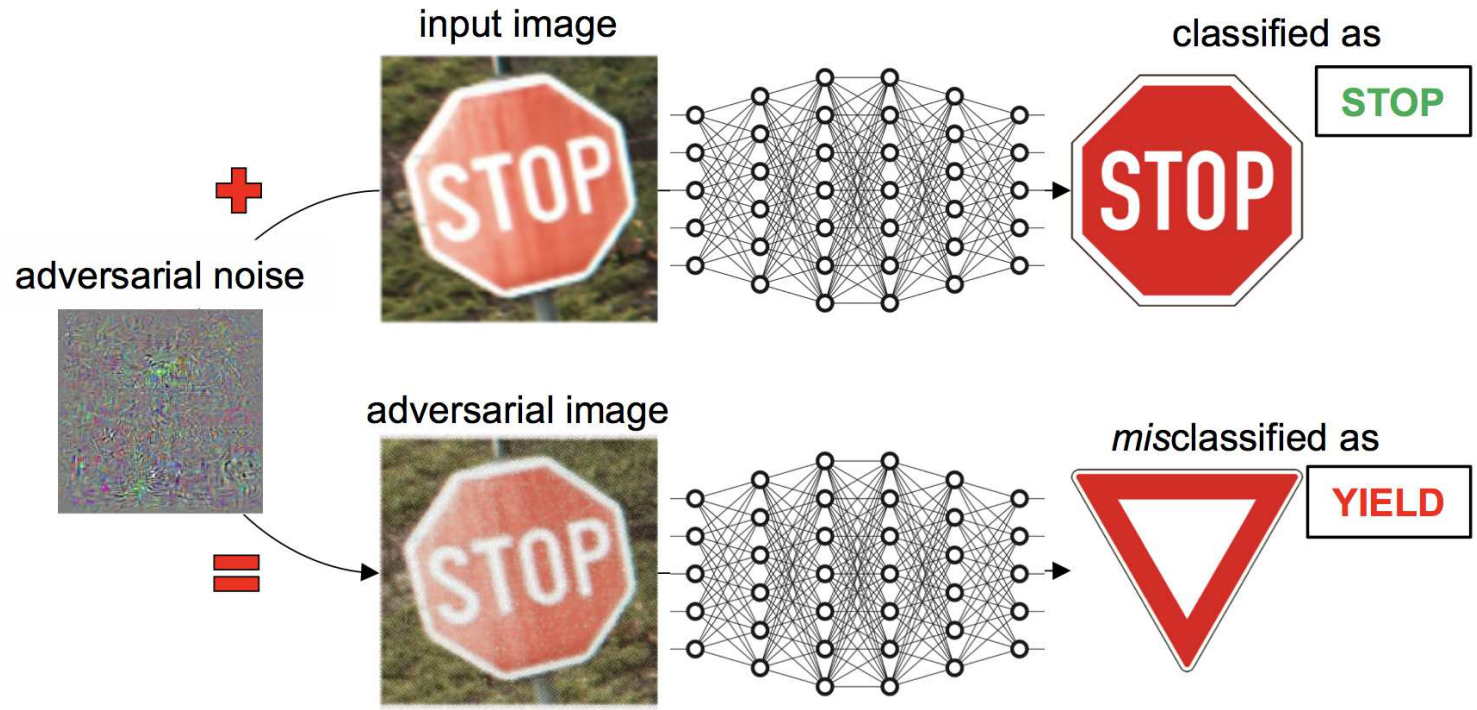
Mismatch between Empirical Observation and Learning Theory

Fitting Random Labeling of the Training Data

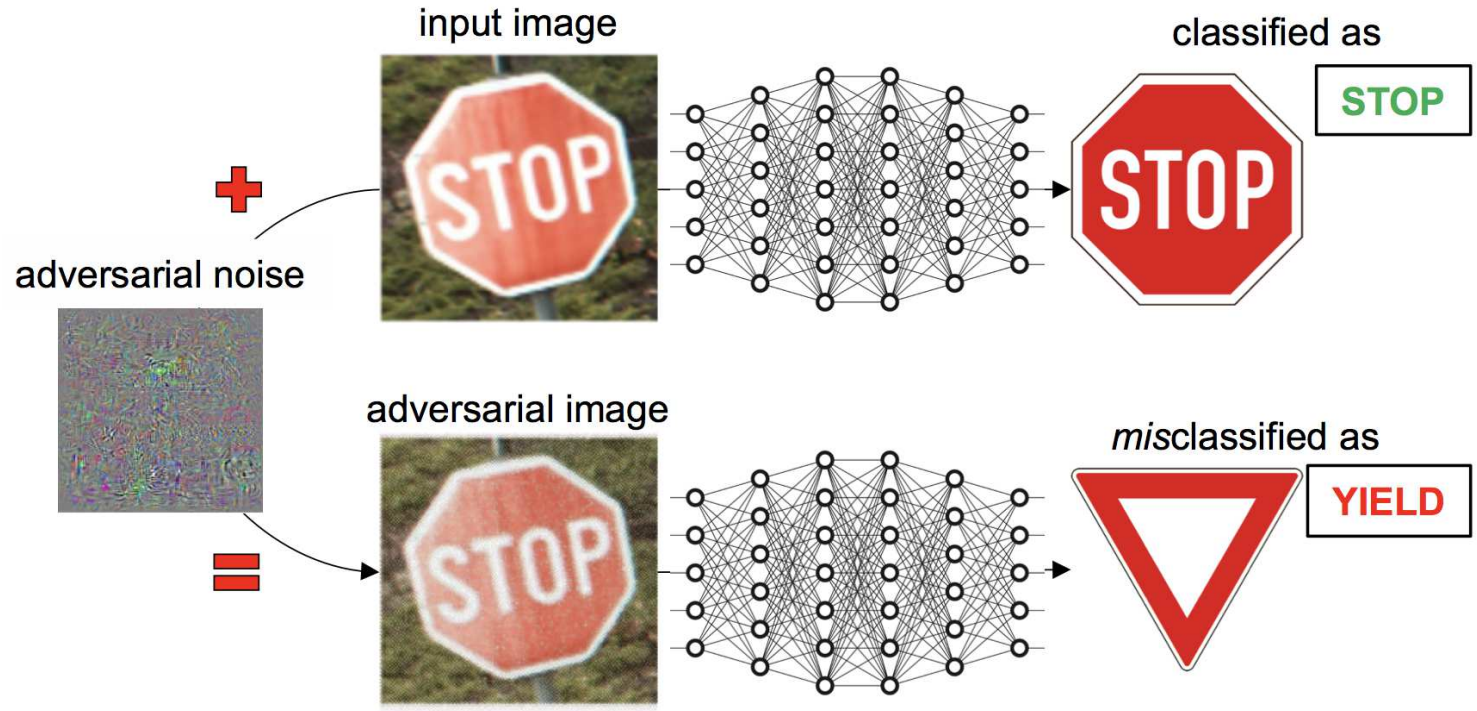
- Bad generalization happens: DNN easily fits a random data
- Regularization techniques does not help with generalization

Conventional Wisdom Fails

Challenge In Robustness of Deep Neural Network



Challenge In Robustness of Deep Neural Network



Reliable AI Not Guaranteed Yet

Outline

Outline

Challenge In Understanding Generalization of Deep Neural Networks

Mismatch between Empirical Observation and Learning Theory

- The deep neural network models are in principle rich enough to memorize the training data (# parameters $\gg$ # of examples)
- However in practice, they do **not** overfit

Challenge In Understanding Generalization of Deep Neural Networks

Mismatch between Empirical Observation and Learning Theory

- The deep neural network models are in principle rich enough to memorize the training data (# parameters $\gg$ # of examples)
- However in practice, they do **not** overfit

Conceptual Challenge to Statistical Learning Theory

- Traditional measures of model complexity **struggle** to explain the generalization ability of large networks
- We have yet to discover a **measure** under which these enormous models are **simple**

Computational Challenge In Large Neural Networks

Test

- Requires large amount of computation and memory storage.
  - Ill-suited for smart phones or IoT device.
- Repeated cost.

Computational Challenge In Large Neural Networks

Test

- Requires large amount of computation and memory storage.
  - Ill-suited for smart phones or IoT device.
- Repeated cost.

Questions

- How to explain generalization ability? Effective capacity?
- Can we compress the neural network for computational efficiency?
- Can compression be used to understand generalization?
- How to compress the neural network without much performance loss?

Goal

Better Understanding of Generalization

Goal

Better Understanding of Generalization

Derive **efficient** and **provably correct** algorithms that reduce the **number of parameters** in the nets, yielding generalization bounds that:

Goal

Better Understanding of Generalization

Derive **efficient** and **provably correct** algorithms that reduce the **number of parameters** in the nets, yielding generalization bounds that:

- are better than naive parameter counting
- depend on simple, intuitive and measurable properties of the network
- empirically correlate with generalization

Goal

Better Understanding of Generalization

Derive **efficient** and **provably correct** algorithms that reduce the **number of parameters** in the nets, yielding generalization bounds that:

- are better than naive parameter counting
- depend on simple, intuitive and measurable properties of the network
- empirically correlate with generalization

More Efficient Model, Faster Prediction

- Fewer number of parameters
- Preserved high expressive power
- No fine-tuning

A simple compression framework for proving generalization bounds

Characterize Generalization via Compression-based Framework

Compressible Nets

- A sample S with m examples
- A deep net f with $\gg m$ parameters
- A compressed net g with q parameters (at most r discrete values)

$$|f(x)[i] - g(x)[i]| \leq \gamma \qquad \forall (x, y) \in S \qquad \forall i$$

Characterize Generalization via Compression-based Framework

Compressible Nets

- A sample S with m examples
- A deep net f with $\gg m$ parameters
- A compressed net g with q parameters (at most r discrete values)

$$|f(x)[i] - g(x)[i]| \leq \gamma \qquad \forall (x, y) \in S \qquad \forall i$$

Generalization of the Compressed Net

$$L_0(g) \leq \widehat{L}_\gamma(f) + O\left(\sqrt{\frac{q \log r}{m}}\right)$$

$$\mathbb{P}_{(x,y)\sim\mathcal{D}}\left[g(x)[y] \leq \max_{i \neq y} g(x)[i]\right] \leq \mathbb{P}_{(x,y)\in S}\left[f(x)[y] \leq \gamma + \max_{i \neq y} f(x)[i]\right] + O\left(\sqrt{\frac{q \log r}{m}}\right)$$

Can we design compression techniques that reduce the actual number of parameter and speed up prediction?

Can we design compression techniques that reduce the actual number of parameter and speed up prediction?

Can we still provide theoretical guarantees for such compression?

Can we design compression techniques that reduce the actual number of parameter and speed up prediction?

Can we still provide theoretical guarantees for such compression?

Can the compression depend on simple, intuitive and measurable properties of the network?

Compression Using Invariant Structure In Deep Neural Networks

Common Compression Techniques

- Pruning, quantization, encoding and knowledge distillation

Compression Using Invariant Structure In Deep Neural Networks

Common Compression Techniques

- Pruning, quantization, encoding and knowledge distillation

Low Rank Approximation

- Complementary to other techniques
- Reduce the number of parameters by a factor **polynomial** in the dimension
  - Caveat: only when the weight matrices (convolutional kernels) are **low rank**

Compression Using Invariant Structure In Deep Neural Networks

Common Compression Techniques

- Pruning, quantization, encoding and knowledge distillation

Low Rank Approximation

- Complementary to other techniques
- Reduce the number of parameters by a factor **polynomial** in the dimension
  - Caveat: only when the weight matrices (convolutional kernels) are **low rank**

Exploiting other invariant structure via low rank approximation?

Periodicity, modulation and low rank?

CP Layer

New Architecture

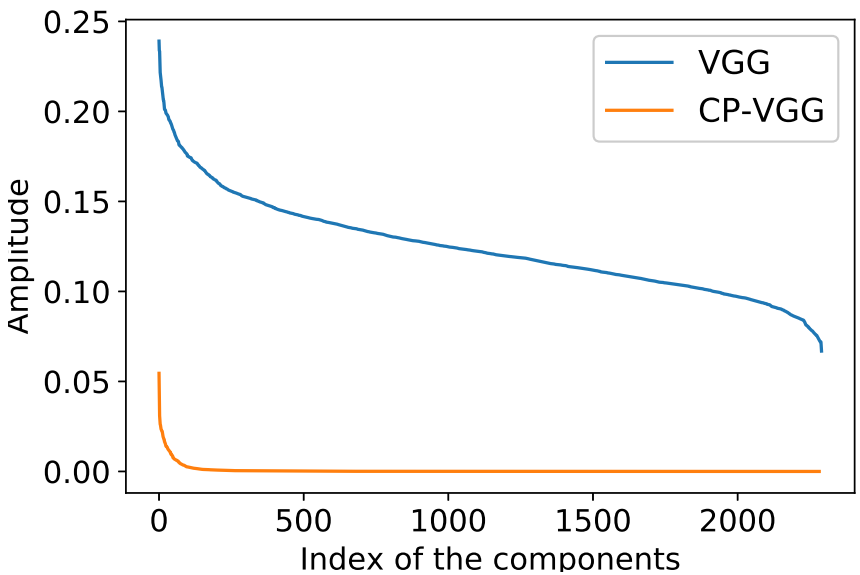
A *CP Layer* represents a given N-order weight tensor $\boldsymbol{\mathcal{K}}$ with R sets of components denoted by $\left(\lambda^{(r)}, \{\boldsymbol{v}_j^{(r)}\}_{j=1}^{N}\right)_{r=1}^{R}$, such that $\boldsymbol{\mathcal{K}} = \sum_{r=1}^{R} \lambda^{(r)} \boldsymbol{v}_1^{(r)} \otimes \cdots \otimes \boldsymbol{v}_N^{(r)}$.

CP Layer

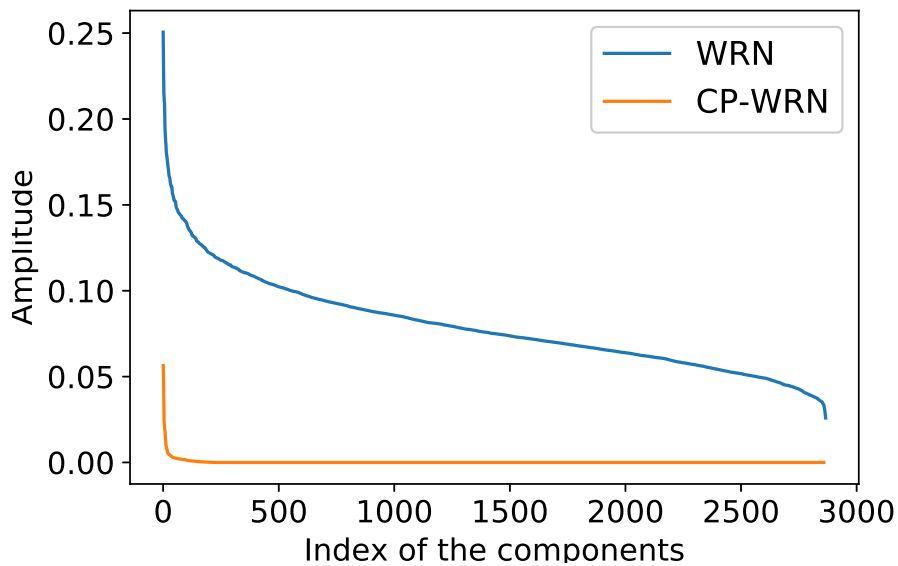
New Architecture

A *CP Layer* represents a given N-order weight tensor $\mathcal{K}$ with R sets of components denoted by $\left(\lambda^{(r)}, \{\boldsymbol{v}_j^{(r)}\}_{j=1}^{N}\right)_{r=1}^{R}$, such that $\mathcal{K} = \sum_{r=1}^{R} \lambda^{(r)} \boldsymbol{v}_1^{(r)} \otimes \cdots \otimes \boldsymbol{v}_N^{(r)}$.

CP Layer exhibits "Low Rankness"



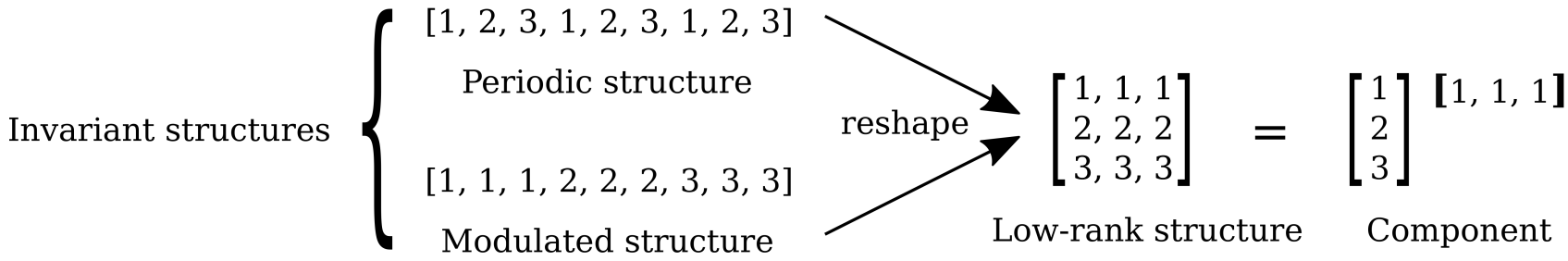
VGG conv13



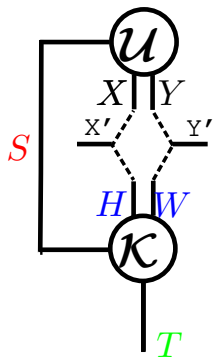
WRN conv27

Reshaped Tensor Decomposition

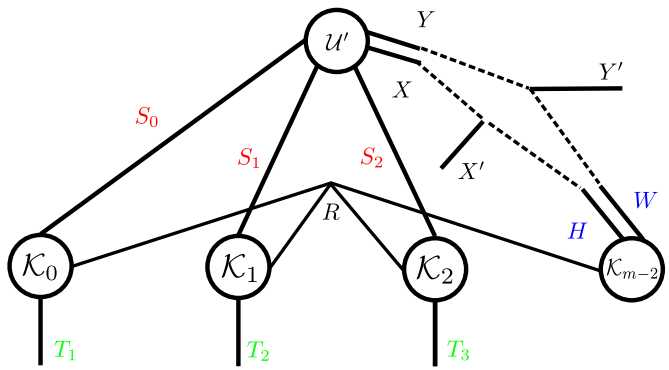
Reshaped Tensor Decomposed Form



More compact representations



Uncompressed



Compressed via r-CP

CP Decomposition on the Reshaped Kernel

- Param. $\#: HWST \rightarrow (m(ST)^{\frac{1}{m}}R + HW)R$

Main Theorem: Generalization Error Bound

To achieve γ compression on sample S

$\tilde{O}\left(\sum_{k=1}^{n} \hat{R}^{(k)}\right)$ number of parameters is required to achieve γ compression on sample S

$$L_0(g) \leq \widehat{L}_\gamma(f) + \tilde{O}\left(\sqrt{\frac{\sum_{k=1}^{n} \hat{R}^{(k)}}{m}}\right)$$

- Rank for layer-k:

$$\hat{R}^{(k)} = \min\left\{j \in [R^{(k)}] | \sqrt{n2^n (\xi_j^{(k)})^2 \theta_j^{(k)} \Pi_{i=k+1}^{n} (t^{(i)})^2 \eta^{(i)}} \leq C\right\}$$

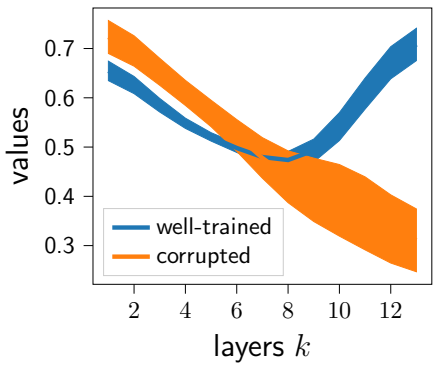
Measurable network properties determine $\hat{R}^{(k)}$

that achieve γ compression on sample S

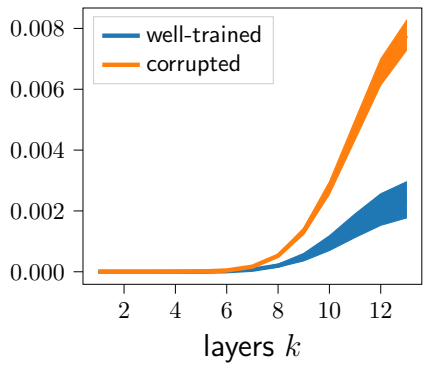
Measurable Network Properties

Minimal sized network to achieve γ compression on sample S

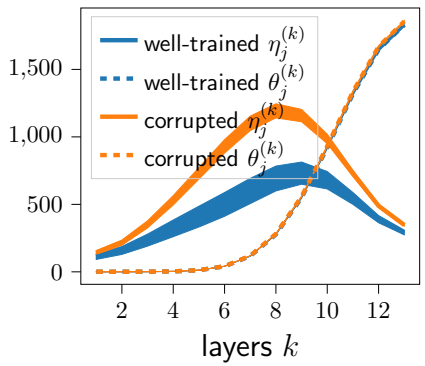
Tensorization Factor $t_j^{(k)}$



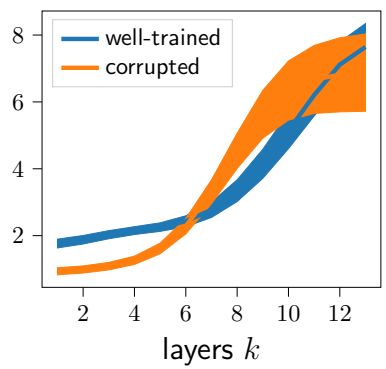
Tensor Noise Bound $\xi_j^{(k)}$



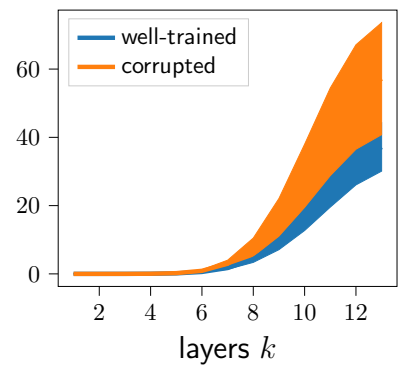
Fourier Factors $\theta_j^{(k)}, \eta_j^{(k)}$



Layer Cushion $\zeta^{(k)}$



Rank $\propto t_j^{(k)} \xi_j^{(k)} \theta_j^{(k)} \eta_j^{(k)} / \zeta^{(k)}$



Remark: Small $t^{(k)}$, $\xi^{(k)}$, $\theta^{(k)}$, $\eta^{(k)}$ $\rightarrow$ highly compressible networks $\rightarrow$ tighter generalization bounds

Experiments - Expressive Power of CP Layers

Performance Comparison: Traditional NN vs Our CP-Layer NN

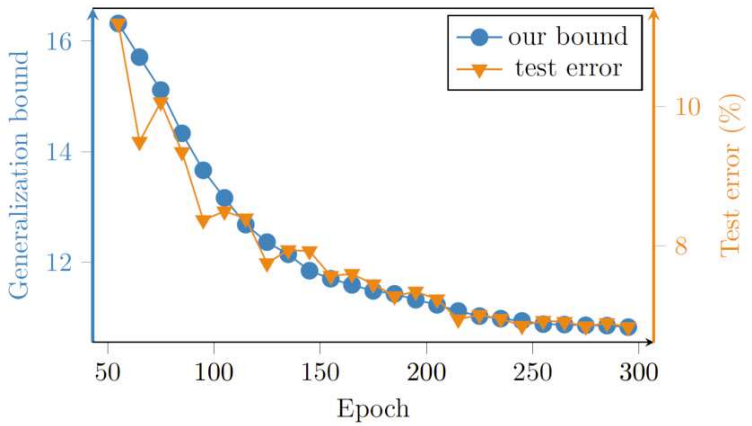
Acc / Archi / Set	VGG-16		WRN-28-10	
	with CPL	without CPL	with CPL	without CPL
Traning	100%			
Test	93.68%	92.64%	94.18%	95.83%

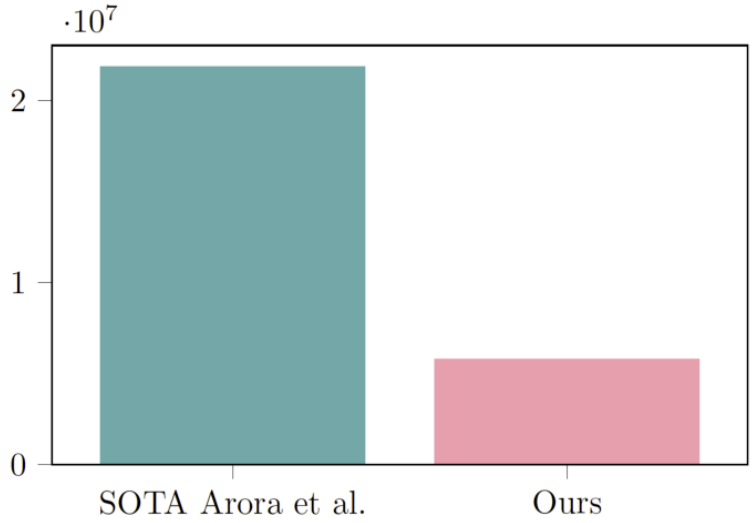
Training and test performance of the traditional VGG-16, WRN-28-10 **vs** our CP-VGG-16, CP-WRN-28-10 on CIFAR10 dataset.

Our CP-Layer neural networks achieve comparable expressive power

Experiments - Bound Evaluation

Generalization bound vs Test Error for VGG-16 with CPL





Generalization bound

- matches with trend of the generalization error
- predicts the generalization error especially well at the beginning/end of training
- is 10 times tighter than the bound in (Arora et. al. 2018)

Use the measurable bound to improve generalization

Experiments - Generalization Improvements under Label Noise

Label Noise Setting

Assign random labels to a proportion of the training examples and train the neural network until convergence

- Memorization effect is directly linked to the deteriorated generalization performance
- Memorization effect could be studied through label noise setting

Network / Corruption Ratio	0.2	0.4	0.6	0.8
VGG	68.76	44.26	24.89	13.21
VGG CP-Layer	**71.09**	**51.76**	**35.60**	**20.06**

CP-Layer achieves better generalization under various label corruption ratios

Experiments - Compression

Successful Compression of CIFAR10 Resnet-32 Network (Su, Li, Bhattacharjee & H., 2018)

	Compression rate					Compression rate			
	5%	10%	20%	40%		2%	5%	10%	20%
SVD	83.09	87.27	89.58	90.85	r-TR†	-	80.80	-	90.60
CP	84.02	86.93	88.75	88.75	r-CP	85.7	89.86	**91.28**	-
TK	83.57	86.00	88.03	89.35	r-TK	61.06	71.34	81.59	87.11
TT	77.44	82.92	84.13	86.64	r-TT	78.95	84.26	87.89	-

- Testing accuracies of tensor methods under compression rates.
- The uncompressed network achieves 93.2% accuracy.
- CIFAR10 Resnet-32 has 0.46M parameters that have to be trained and retained during testing.

Experiments - Large Scale Compression

Successful Compression of ImageNet Resnet-50 Network (Su, Li, Bhattacharjee & H., 2018)

# samples	Uncompressed # params.: 25M	TT (E2E) # params.: 2.5M	r-TT (Seq) # params.: 2.5M
0.24M	4.22	2.78	44.35
0.36M	6.23	3.99	46.98
0.60M	9.01	7.48	49.92
1.20M	17.3	12.80	52.59
2.40M	30.8	18.17	**54.00**

- Testing accuracy of tensor methods compared to the uncompressed ImageNet Resnet-50.

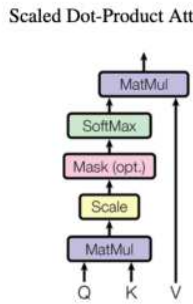
Outline

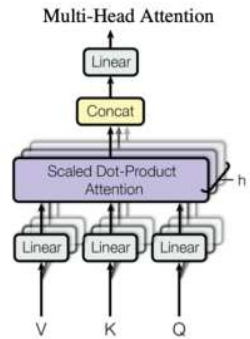
Model Design of Self-attention Units in Transformers

Multihead Self-Attention Units

$$head_i = \text{softmax}(\frac{QK^T}{\sqrt{D}})V = \text{softmax}(\frac{XW_i^Q(XW_i^X)^T}{\sqrt{D}})XW_i^V$$

$$\text{Multi-Head Self-Attention} = \text{Concat}(head_1, \ldots, head_H)W^O$$





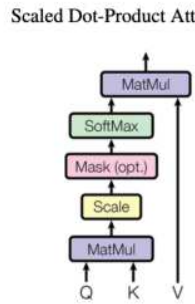
- Tremendous impact in a variety of applications: language, audio, image, graph and etc.

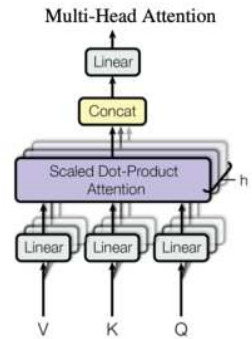
Model Design of Self-attention Units in Transformers

Multihead Self-Attention Units

$$head_i = \text{softmax}(\frac{QK^T}{\sqrt{D}})V = \text{softmax}(\frac{XW_i^Q(XW_i^X)^T}{\sqrt{D}})XW_i^V$$

$$\text{Multi-Head Self-Attention} = \text{Concat}(head_1, \ldots, head_H)W^O$$





- Tremendous impact in a variety of applications: language, audio, image, graph and etc.

However

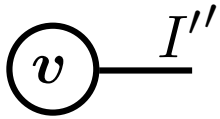
- Lack rigorous visual interpretation.
- Role of multi-head self-attention unclear.
  - Introduced as a parallelism. Is that all it does?
- Scalability is a concern. More efficient model design?
  - Existing works (Reformer, Linformer, Linear Transformers, etc) focus on the efficiency w.r.t. sequence length. What about model parameters?

A Rigorous Visual Interpretation of Self-attention

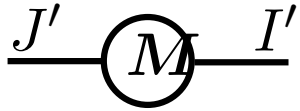
Tensor Diagram: Rigorous Graphical Representation of Operations between Multi-dimensional Arrays/High-order Tensors

- *Arrays* denoted as *nodes with legs*, which are orientation invariant.

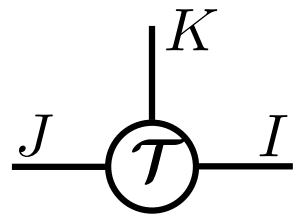
Scalar $a \in \mathbb{R}$



Vector $\boldsymbol{v} \in \mathbb{R}^{I''}$



Matrix $\boldsymbol{M} \in \mathbb{R}^{I' \times J'}$



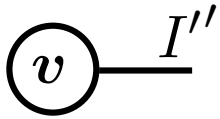
Tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{I \times J \times K}$

A Rigorous Visual Interpretation of Self-attention

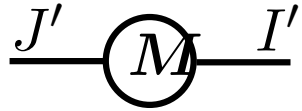
Tensor Diagram: Rigorous Graphical Representation of Operations between Multi-dimensional Arrays/High-order Tensors

- *Arrays* denoted as *nodes with legs*, which are orientation invariant.

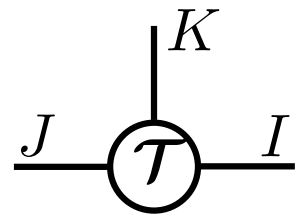
Scalar $a \in \mathbb{R}$



Vector $\boldsymbol{v} \in \mathbb{R}^{I''}$



Matrix $\boldsymbol{M} \in \mathbb{R}^{I' \times J'}$

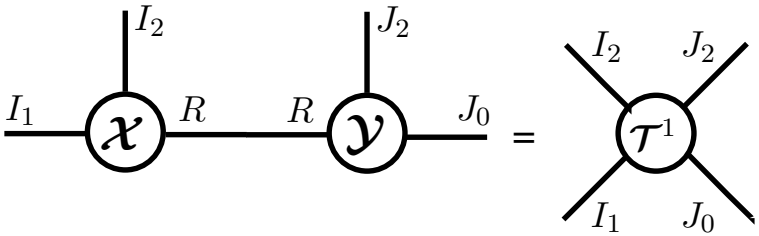


Tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{I \times J \times K}$

- *Operations* denoted as *edges* connecting the *node legs*.

Mode-(R,R) contraction

$\boldsymbol{\mathcal{X}} \times_R^R \boldsymbol{\mathcal{Y}} \to \boldsymbol{\mathcal{T}}^1$



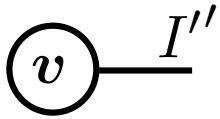
$$\boldsymbol{\mathcal{T}}^1_{i_1,i_2,j_0,j_2} = \sum_r \boldsymbol{\mathcal{X}}_{r,i_1,i_2} \boldsymbol{\mathcal{Y}}_{j_0,r,j_2}$$

A Rigorous Visual Interpretation of Self-attention

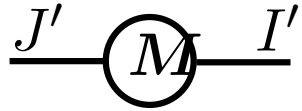
Tensor Diagram: Rigorous Graphical Representation of Operations between Multi-dimensional Arrays/High-order Tensors

- *Arrays* denoted as *nodes with legs*, which are orientation invariant.

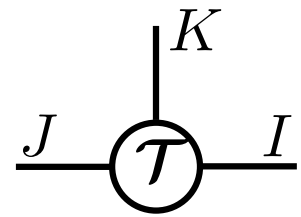
Scalar $a \in \mathbb{R}$



Vector $\boldsymbol{v} \in \mathbb{R}^{I''}$



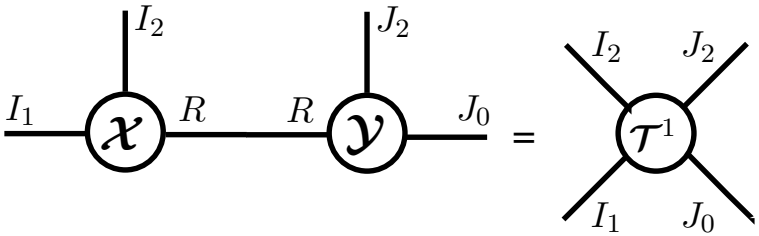
Matrix $\boldsymbol{M} \in \mathbb{R}^{I' \times J'}$



Tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{I \times J \times K}$

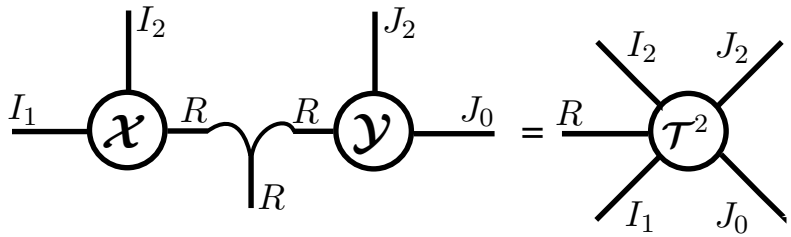
- *Operations* denoted as *edges* connecting the *node legs*.

Mode-(R,R) contraction $\boldsymbol{\mathcal{X}} \times_R^R \boldsymbol{\mathcal{Y}} \to \boldsymbol{\mathcal{T}}^1$



$$\boldsymbol{\mathcal{T}}^1_{i_1,i_2,j_0,j_2} = \sum_r \boldsymbol{\mathcal{X}}_{r,i_1,i_2} \boldsymbol{\mathcal{Y}}_{j_0,r,j_2}$$

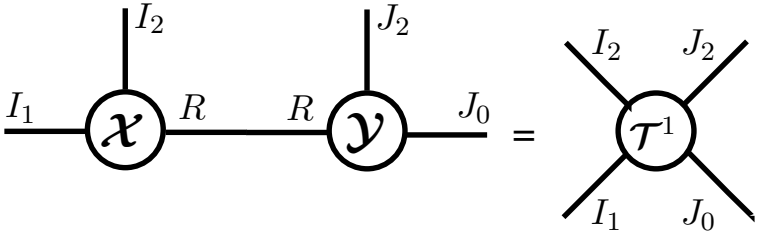
Mode-(R,R) batch multiplication $\boldsymbol{\mathcal{X}} \otimes_R^R \boldsymbol{\mathcal{Y}} \to \boldsymbol{\mathcal{T}}^2$



$$\boldsymbol{\mathcal{T}}^2_{r,i_1,i_2,j_0,j_2} = \boldsymbol{\mathcal{X}}_{r,i_1,i_2} \boldsymbol{\mathcal{Y}}_{j_0,r,j_2}$$

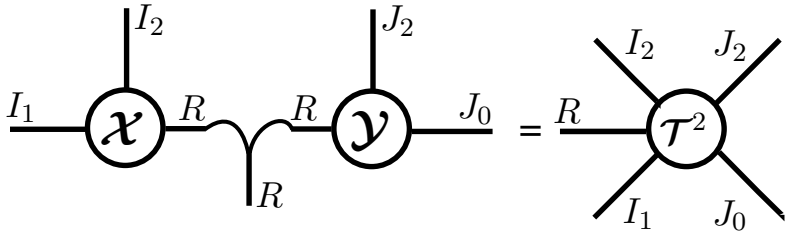
Benefits of Tensor Diagram

Mode-(R,R) contraction
$\mathcal{X} \times_R^R \mathcal{Y} \to \mathcal{T}^1$



$\mathcal{T}^1_{i_1,i_2,j_0,j_2} = \sum_r \mathcal{X}_{r,i_1,i_2} \mathcal{Y}_{j_0,r,j_2}$

Mode-(R,R) batch multiplication
$\mathcal{X} \otimes_R^R \mathcal{Y} \to \mathcal{T}^2$



$\mathcal{T}^2_{r,i_1,i_2,j_0,j_2} = \mathcal{X}_{r,i_1,i_2} \mathcal{Y}_{j_0,r,j_2}$

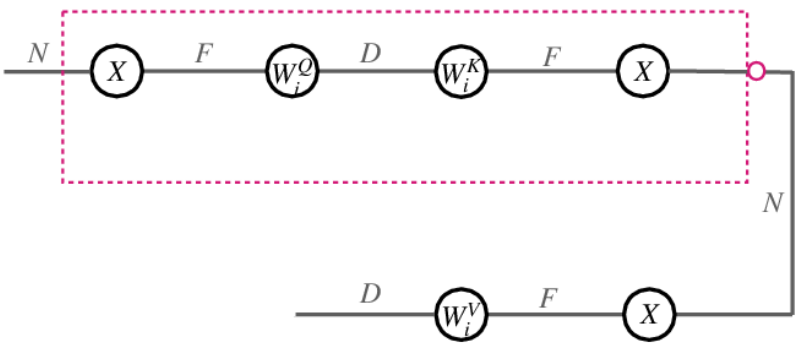
- **Orientation Invariant and Explicit Mark of Dimensionality.** Unlike mathematical formulas, no need to carefully order of the "modes" of the data objects. E.g., $AB, BA, AB^\top, A^\top B$ and etc.
- **Easy Notation of Multi-linear Operations.** Different legs/modes "operate on their own".
- **Resultant via node merging.** Merge connected nodes, output dimension is represented as the dangling legs.
- **Evaluation of a series of operations denoted as multi-step merging of nodes.**

Rigorous Visual Representation of Single-head Self-Attention

Scaled Dot-Product Attention

$$head_i = \text{softmax}(\frac{QK^T}{\sqrt{D}})V = \text{softmax}(\frac{XW_i^Q(XW_i^X)^T}{\sqrt{D}})XW_i^V$$

Rigorous Visual Representation



Advantages

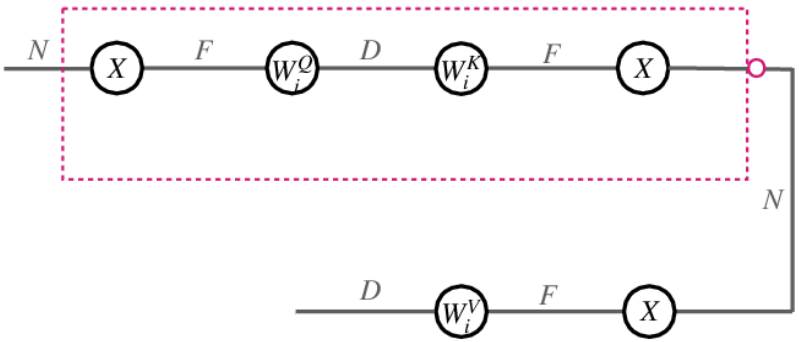
- Invariant to arrangement of the input data matrix X (word per row or per column).
- Model size is explicitly shown.
- Softmax constraints the evaluation ordering of the graph.

Rigorous Visual Representation of Multi-head Self-Attention

$$head_i = \text{softmax}(\frac{QK^T}{\sqrt{D}})V = \text{softmax}(\frac{XW_i^Q(XW_i^X)^T}{\sqrt{D}})XW_i^V$$

$$\text{Multi-Head Self-Attention} = \text{Concat}(head_1, \ldots, head_H)W^O$$

Single-head **Multi-head**

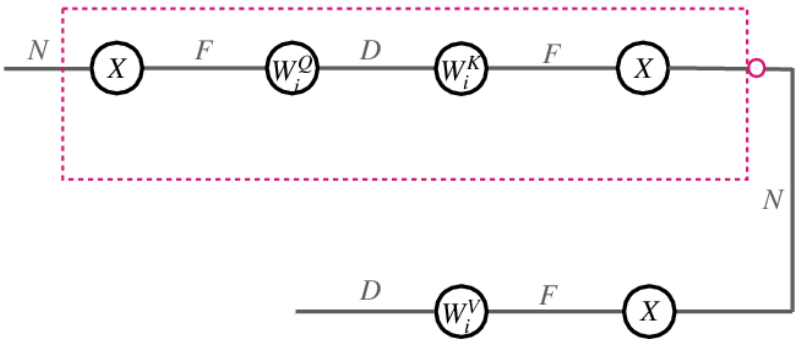


Rigorous Visual Representation of Multi-head Self-Attention

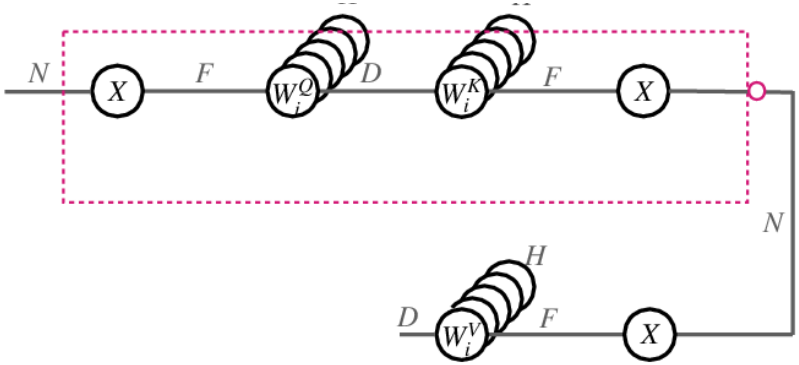
$$head_i = \text{softmax}(\frac{QK^T}{\sqrt{D}})V = \text{softmax}(\frac{XW_i^Q(XW_i^X)^T}{\sqrt{D}})XW_i^V$$

$$\text{Multi-Head Self-Attention} = \text{Concat}(head_1, \ldots, head_H)W^O$$

Single-head



Multi-head



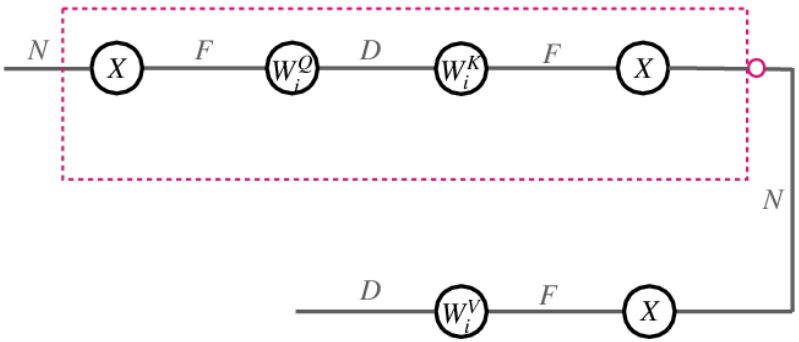
- The modes along which data X contracts with the weights are unchanged.

Rigorous Visual Representation of Multi-head Self-Attention

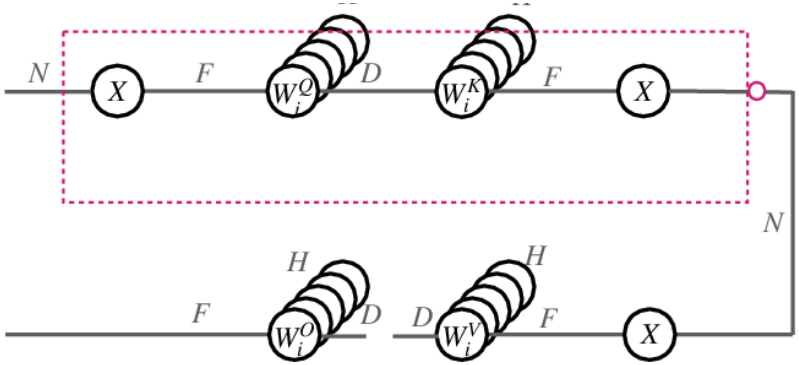
$$head_i = \text{softmax}(\frac{QK^T}{\sqrt{D}})V = \text{softmax}(\frac{XW_i^Q(XW_i^X)^T}{\sqrt{D}})XW_i^V$$

$$\text{Multi-Head Self-Attention} = \text{Concat}(head_1, \ldots, head_H)W^O$$

Single-head



Multi-head



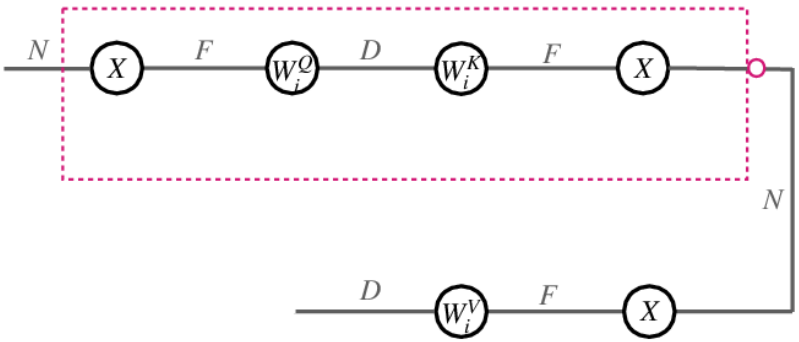
- The modes along which data X contracts with the weights are unchanged.
- Stack weight matrices W^Q, W^K, W^V and W^O from multi-heads as third order tensors (i.e., three-legged nodes).

Rigorous Visual Representation of Multi-head Self-Attention

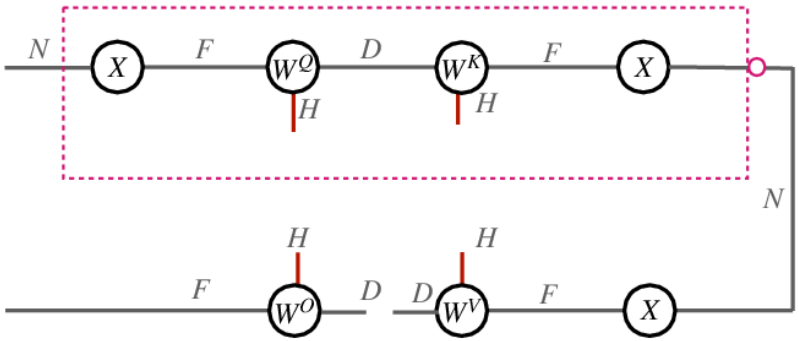
$$head_i = \text{softmax}(\frac{QK^T}{\sqrt{D}})V = \text{softmax}(\frac{XW_i^Q(XW_i^X)^T}{\sqrt{D}})XW_i^V$$

$$\text{Multi-Head Self-Attention} = \text{Concat}(head_1, \ldots, head_H)W^O$$

Single-head



Multi-head



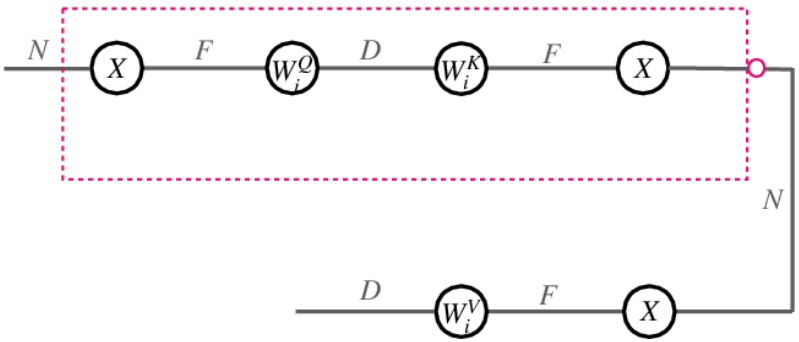
- The modes along which data X contracts with the weights are unchanged.
- Stack weight matrices W^Q, W^K, W^V and W^O from multi-heads as third order tensors (i.e., three-legged nodes).

Rigorous Visual Representation of Multi-head Self-Attention

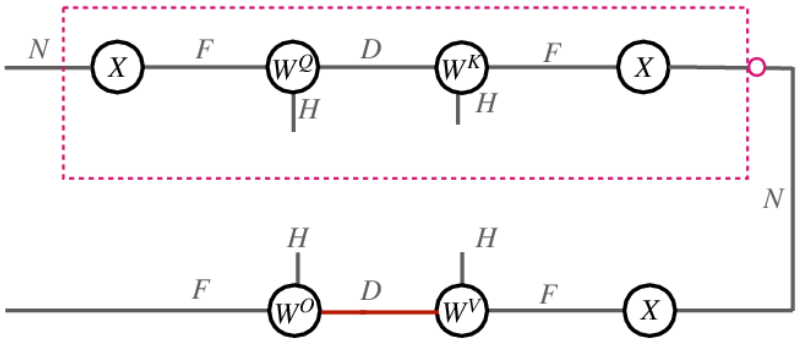
$$head_i = \text{softmax}(\frac{QK^T}{\sqrt{D}})V = \text{softmax}(\frac{XW_i^Q(XW_i^X)^T}{\sqrt{D}})XW_i^V$$

$$\text{Multi-Head Self-Attention} = \text{Concat}(head_1, \ldots, head_H)W^O$$

Single-head



Multi-head



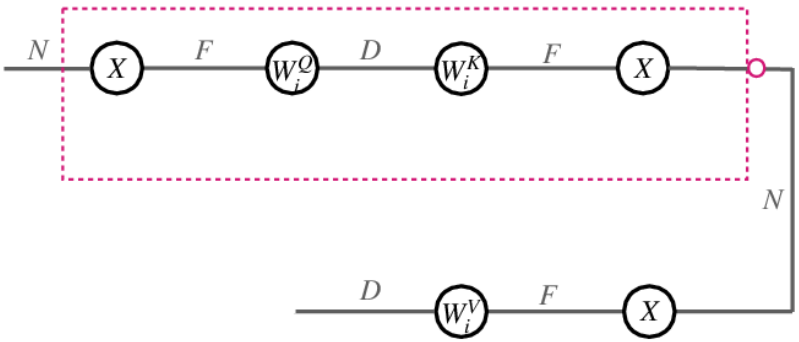
- The modes along which data X contracts with the weights are unchanged.
- Stack weight matrices W^Q, W^K, W^V and W^O from multi-heads as third order tensors (i.e., three-legged nodes).
- *Mode-D* of W^O and W^V are contracted.

Rigorous Visual Representation of Multi-head Self-Attention

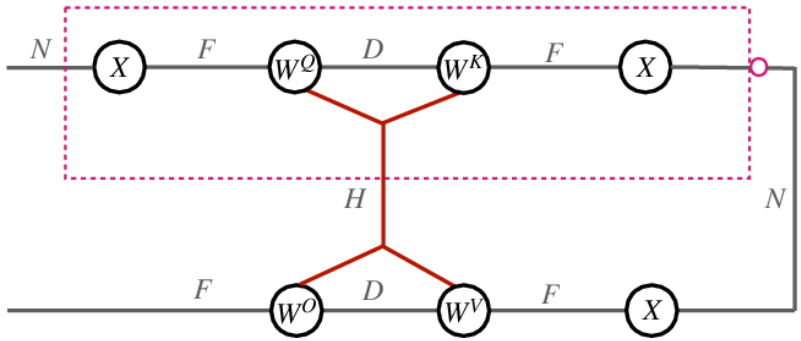
$$head_i = \text{softmax}(\frac{QK^T}{\sqrt{D}})V = \text{softmax}(\frac{XW_i^Q(XW_i^X)^T}{\sqrt{D}})XW_i^V$$

$$\text{Multi-Head Self-Attention} = \text{Concat}(head_1, \ldots, head_H)W^O$$

Single-head



Multi-head



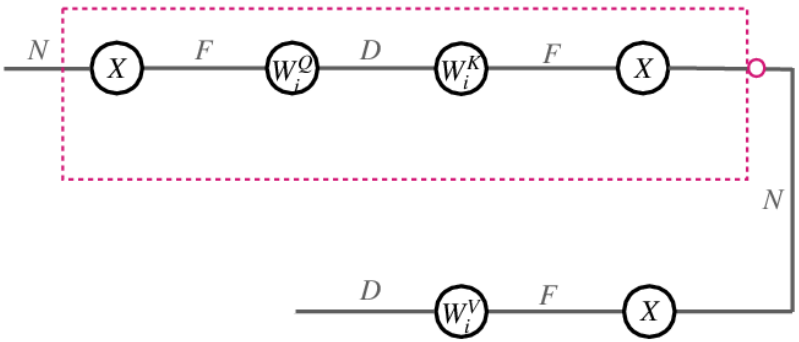
- The modes along which data X contracts with the weights are unchanged.
- Stack weight matrices W^Q, W^K, W^V and W^O from multi-heads as third order tensors (i.e., three-legged nodes).
- *Mode-D* of W^O and W^V are contracted.
- Concatenation of the H feature maps followed by contraction with *mode-H* of W^O is equivalent to contraction along mode-H of all weight tensors.

Rigorous Visual Representation of Multi-head Self-Attention

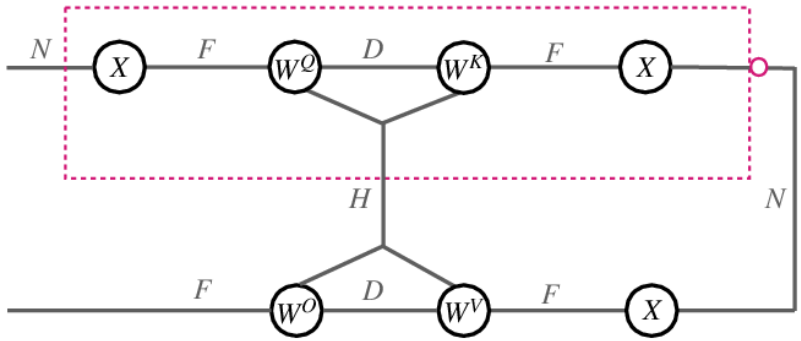
$$head_i = \text{softmax}(\frac{QK^T}{\sqrt{D}})V = \text{softmax}(\frac{XW_i^Q(XW_i^X)^T}{\sqrt{D}})XW_i^V$$

$$\text{Multi-Head Self-Attention} = \text{Concat}(head_1, \ldots, head_H)W^O$$

Single-head



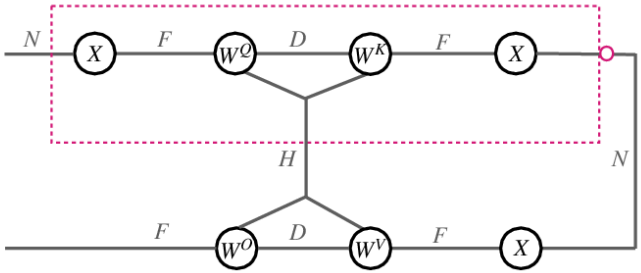
Multi-head



- The modes along which data X contracts with the weights are unchanged.
- Stack weight matrices W^Q, W^K, W^V and W^O from multi-heads as third order tensors (i.e., three-legged nodes).
- *Mode-D* of W^O and W^V are contracted.
- Concatenation of the H feature maps followed by contraction with *mode-H* of W^O is equivalent to contraction along mode-H of all weight tensors.

Improved Model Design

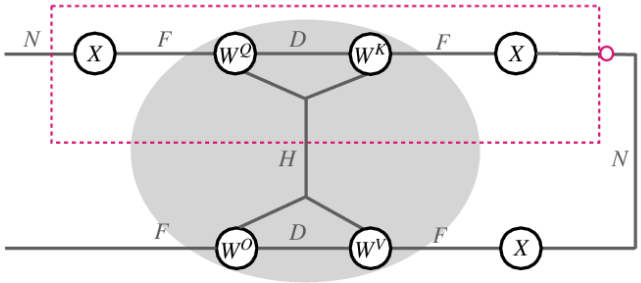
Original Multi-head



- The role of multi-head H?
- The role of feature dimension D_1?
- Both necessary? Which one is more crucial?

Improved Model Design

Original Multi-head

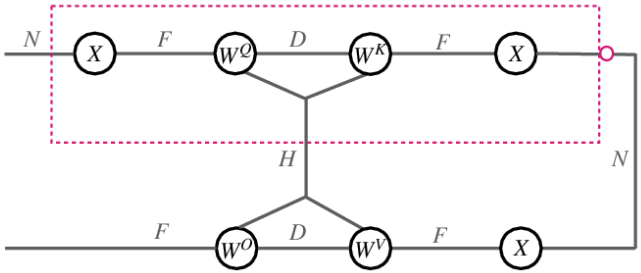


- The role of multi-head H?
- The role of feature dimension D_1?
- Both necessary? Which one is more crucial?

Intuition: holistic consideration of the weights W^Q, W^K, W^V and W^O

Improved Model Design

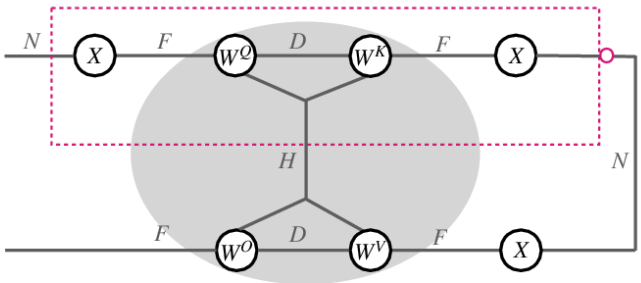
Original Multi-head



- The role of multi-head H?
- The role of feature dimension D_1?
- Both necessary? Which one is more crucial?

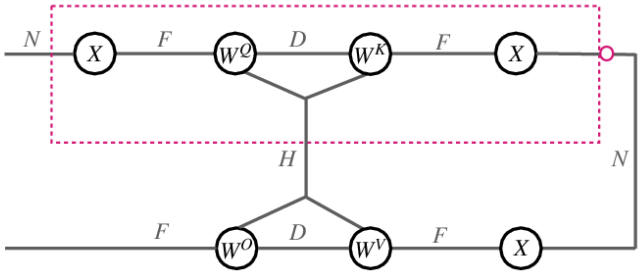
Intuition: holistic consideration of the weights W^Q, W^K, W^V and W^O

Ours: Lossless Multi-head



Improved Model Design

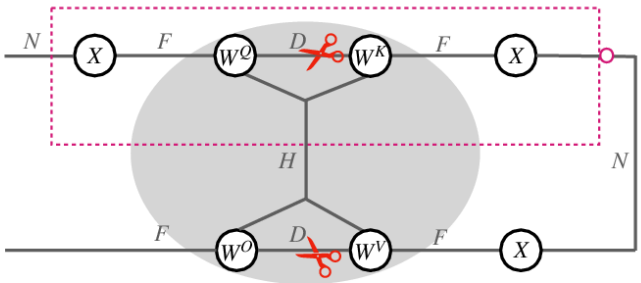
Original Multi-head



- The role of multi-head H?
- The role of feature dimension D_1?
- Both necessary? Which one is more crucial?

Intuition: holistic consideration of the weights W^Q, W^K, W^V and W^O

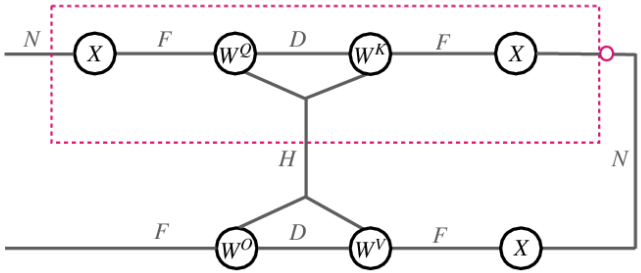
Ours: Lossless Multi-head



- Remove the lossy and thus redundant dot-product/contraction in attention units

Improved Model Design

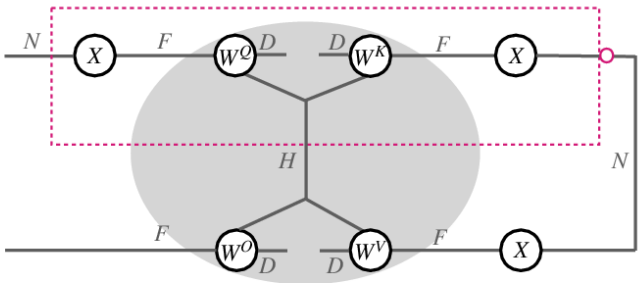
Original Multi-head



- The role of multi-head H?
- The role of feature dimension D_1?
- Both necessary? Which one is more crucial?

Intuition: holistic consideration of the weights W^Q, W^K, W^V and W^O

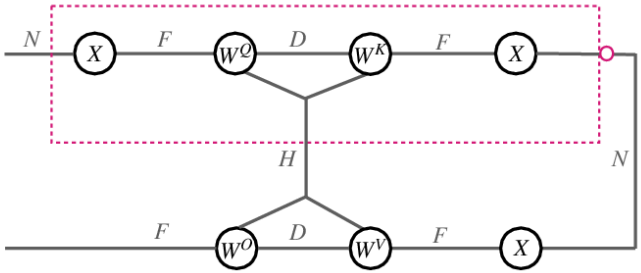
Ours: Lossless Multi-head



- Remove the lossy and thus redundant dot-product/contraction in attention units
- Maintain the role of multi-heads

Improved Model Design

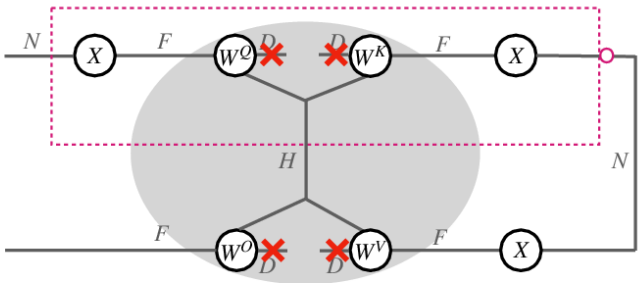
Original Multi-head



- The role of multi-head H?
- The role of feature dimension D_1?
- Both necessary? Which one is more crucial?

Intuition: holistic consideration of the weights W^Q, W^K, W^V and W^O

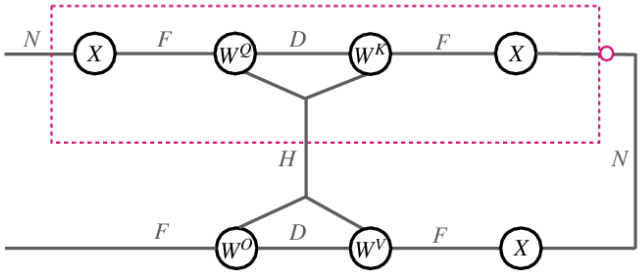
Ours: Lossless Multi-head



- Remove the lossy and thus redundant dot-product/contraction in attention units
- Maintain the role of multi-heads

Improved Model Design

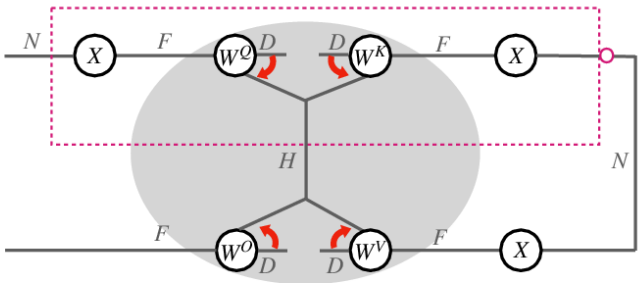
Original Multi-head



- The role of multi-head H?
- The role of feature dimension D_1?
- Both necessary? Which one is more crucial?

Intuition: holistic consideration of the weights W^Q, W^K, W^V and W^O

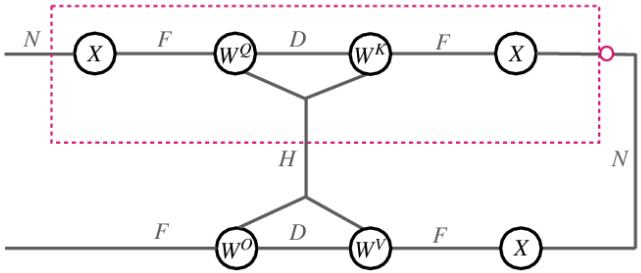
Ours: Lossless Multi-head



- Remove the lossy and thus redundant dot-product/contraction in attention units
- Maintain the role of multi-heads

Improved Model Design

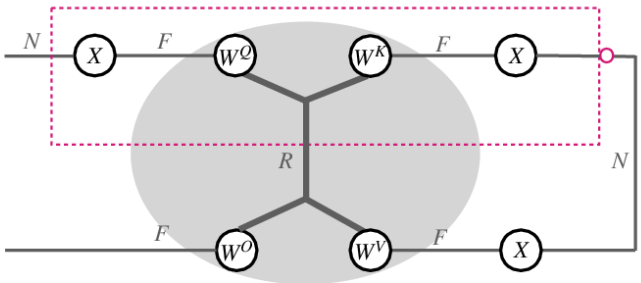
Original Multi-head



- The role of multi-head H?
- The role of feature dimension D_1?
- Both necessary? Which one is more crucial?

Intuition: holistic consideration of the weights W^Q, W^K, W^V and W^O

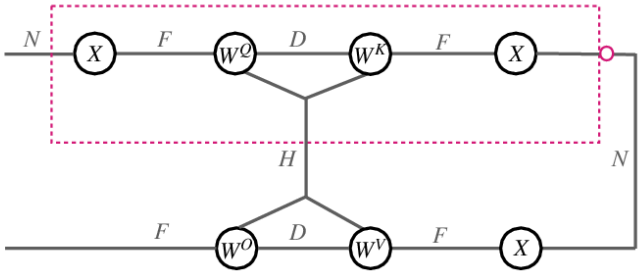
Ours: Lossless Multi-head



- Remove the lossy and thus redundant dot-product/contraction in attention units
- Maintain the role of multi-heads
- A CP form

Improved Model Design

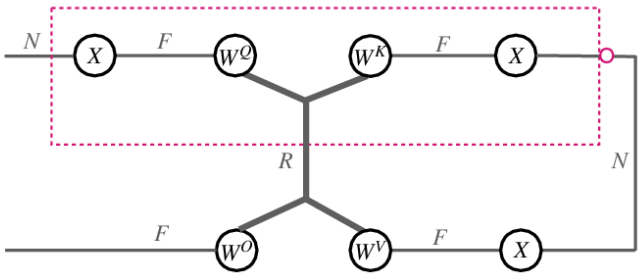
Original Multi-head



- The role of multi-head H?
- The role of feature dimension D_1?
- Both necessary? Which one is more crucial?

Intuition: holistic consideration of the weights W^Q, W^K, W^V and W^O

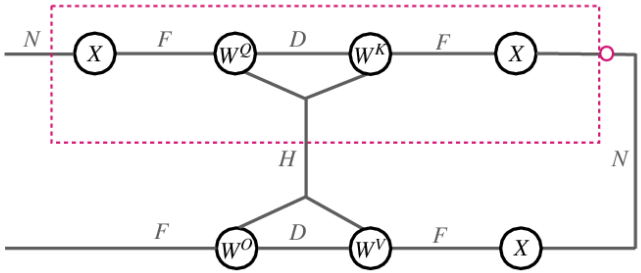
Ours: Lossless Multi-head



- Remove the lossy and thus redundant dot-product/contraction in attention units
- Maintain the role of multi-heads
- A CP form

Improved Model Design

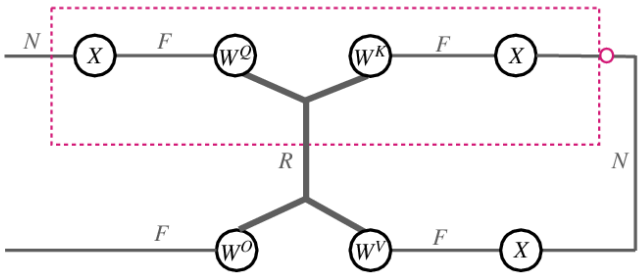
Original Multi-head



- The role of multi-head H?
- The role of feature dimension D_1?
- Both necessary? Which one is more crucial?

Intuition: holistic consideration of the weights W^Q, W^K, W^V and W^O

Ours: Lossless Multi-head



- Remove the lossy and thus redundant dot-product/contraction in attention units
- Maintain the role of multi-heads
- A CP form

- Improved expressive power (under same number of parameters $R = DH$): we prove that the expressive power of our *Lossless Multi-head* is **larger** than that of the original multi-head self-attention units.
- Maintained expressive power (with $\frac{1}{D}$ parameters $R = H$): we prove that the expressive power of our *Lossless Multi-head* is **equivalent** to the original multi-head self-attention units.

Larger Expressive Power Under Same Number of Parameters

Perplexity scores on Language Modeling: PTB dataset

Model	Test Perplexity Score (lower the better)
Original Multi-head	52.7
Lossless Multi-head	**51.9**

BLEU scores on Neural Machine Translation: WMT2016

Model	Test BLEU Score (higher the better)
Original Multi-head	27.1
Lossless Multi-head	**28.4**

BPD (bits per dimension) scores on Image Generation: CIFAR10

Model	Test BPD Score (lower the better)
Original Multi-head	3.47
Lossless Multi-head	**3.09**

Maintained Expressive Power Under Compression

Perplexity scores on Language Modeling: PTB dataset

Model	Compression Rate	Test Perplexity Score
Original Multi-head	100%	52.7
Lossless Multi-head	2%	55.4

Maintained 95% of the performance using 2% number of parameters

BLEU scores on Neural Machine Translation: WMT2016

Model	Compression Rate	Test BLEU Score
Original Multi-head	100%	27.1
Lossless Multi-head	2%	26.3

Maintained 97% of the performance using 2% number of parameters

BPD scores on Image Generation: CIFAR10

Model	Compression Rate	Test BPD Score (lower the better)
Original Multi-head	100%	3.47
Lossless Multi-head	2%	3.71

Maintained 93% of the performance using 2% number of parameters

Outline

Adversarial Examples

A Game Between Attacker and Defender

$$\min_{\boldsymbol{\theta}} \max_{\tilde{\boldsymbol{x}} \in B_\epsilon^p(\boldsymbol{x})} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[L(f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}), y) \right]$$

- Game: pick a ϵ that limits the power of the adversary
- Attack: pick $\tilde{\boldsymbol{x}}$ within ℓ_p norm ϵ ball of $\boldsymbol{x}$, $B_\epsilon^p(\boldsymbol{x})$
- Defense: adjust the model $\boldsymbol{\theta}$ to minimize loss on adversarial examples $(\tilde{\boldsymbol{x}}, y)$

Provable Defense

A defense $\boldsymbol{\theta}$ against the worst possible $\tilde{\boldsymbol{x}}$

Guaranteed Adversarial Robustness by Model Design?

Orthogonal Convolution Layers

Convolution layer $\boldsymbol{h}$ ($\boldsymbol{y} = \boldsymbol{h} * \boldsymbol{x}$, i.e., $y_t[i] = \sum_s \sum_n h_{ts}[n] x_s[i-n]$) is orthogonal if

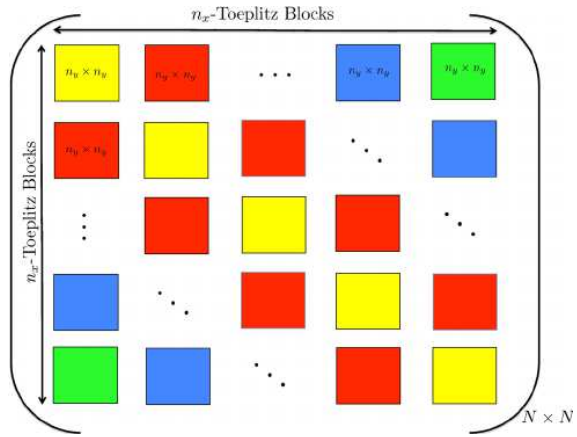
$$\|\boldsymbol{y}\| = \|\boldsymbol{x}\|, \ \forall \boldsymbol{x}$$

Motivation of Orthogonal Convolution Neural Networks

- **Easier optimization**: forward norm and gradient norm preserved.
- **Robustness** against adversarial perturbations: Lipschitz constant strictly less than 1, adversarial perturbations can not be amplified to flip the prediction. Uncontrollable Lipschitz constant causes instability.
- **Better generalization**: works on showing the generalization error positively related to the standard deviation of each linear layer's singular values.

Challenges in Implementing Orthogonal Convolutional Layers

Convolution $\equiv$ Multiplication of a block-Toeplitz Matrix



Each colored block is also a Toeplitz matrix

Existing works

- Incorrectly compute the spectrum of the naively flattened kernel
- Inefficiently compute the singular values for all frequency components
- Unable to deal with variants to the traditional convolutional layer, including dilated, stride, group variants, and combinations.

Enforcing Universal Unitary Property in the Frequency Domain

Standard Convolution Theorem

For a standard convolution layer $\boldsymbol{h}$: $\boldsymbol{y} = \boldsymbol{h} * \boldsymbol{x}$, convolution in the spatial domain $\equiv$ multiplication in the frequency domain

$$\text{i.e.,} \qquad \boldsymbol{Y}(z) = \boldsymbol{H}(z)\boldsymbol{X}(z), \ \forall z \in \mathbb{C}$$

where $\boldsymbol{X}(z) = \sum_{n=0}^{N-1} \boldsymbol{x}[n]z^{-n}$, $\boldsymbol{Y}(z) = \sum_{n=0}^{N-1} \boldsymbol{y}[n]z^{-n}$,$\boldsymbol{H}(z) = \sum_{n=-L}^{R} \boldsymbol{h}[n]z^{-n}$.

Enforcing Universal Unitary Property in the Frequency Domain

Standard Convolution Theorem

For a standard convolution layer $\boldsymbol{h}$: $\boldsymbol{y} = \boldsymbol{h} * \boldsymbol{x}$, convolution in the spatial domain $\equiv$ multiplication in the frequency domain

$$\text{i.e.,} \qquad \boldsymbol{Y}(z) = \boldsymbol{H}(z)\boldsymbol{X}(z), \ \forall z \in \mathbb{C}$$

where $\boldsymbol{X}(z) = \sum_{n=0}^{N-1} \boldsymbol{x}[n] z^{-n}$, $\boldsymbol{Y}(z) = \sum_{n=0}^{N-1} \boldsymbol{y}[n] z^{-n}$, $\boldsymbol{H}(z) = \sum_{n=-L}^{R} \boldsymbol{h}[n] z^{-n}$.

Sufficient and Necessary Condition for Orthogonal Convolution

Orthogonal standard convolution layer

$\Updownarrow$

Unitary $H(z)$ for **all** frequency $z = e^{jw}$, known as **A Paraunitary System**

A Framework for Orthogonal Layers

A Complete Characterization of Paraunitary System via Factorization

Any Paraunitary system can be factorized as multiplications of factors

$$\boldsymbol{H}(z) = \boldsymbol{V}(z;\boldsymbol{U}^{(-L)})\dots\boldsymbol{V}(z;\boldsymbol{U}^{(-1)})\,\boldsymbol{Q}\,\boldsymbol{V}(z^{-1};\boldsymbol{U}^{(1)})\dots\boldsymbol{V}(z^{-1};\boldsymbol{U}^{(R)}),$$

where factors $\boldsymbol{V}(z;\boldsymbol{U}^{(i)}) = \boldsymbol{I} - \boldsymbol{U}^{(i)}\boldsymbol{U}^{(i)^\top} + \boldsymbol{U}^{(i)}\boldsymbol{U}^{(i)^\top} z$ are parameterized by column-orthogonal matrices $\{\boldsymbol{U}^{(i)}\}_{i=-L}^{R}$ and orthogonal matrix $\boldsymbol{Q}$.

A Framework for Orthogonal Layers

A Complete Characterization of Paraunitary System via Factorization

Any Paraunitary system can be factorized as multiplications of factors

$$\boldsymbol{H}(z) = \boldsymbol{V}(z;\boldsymbol{U}^{(-L)})\dots\boldsymbol{V}(z;\boldsymbol{U}^{(-1)})\,\boldsymbol{Q}\,\boldsymbol{V}(z^{-1};\boldsymbol{U}^{(1)})\dots\boldsymbol{V}(z^{-1};\boldsymbol{U}^{(R)}),$$

where factors $\boldsymbol{V}(z;\boldsymbol{U}^{(i)}) = \boldsymbol{I} - \boldsymbol{U}^{(i)}\boldsymbol{U}^{(i)\top} + \boldsymbol{U}^{(i)}\boldsymbol{U}^{(i)\top}z$ are parameterized by column-orthogonal matrices $\{\boldsymbol{U}^{(i)}\}_{i=-L}^{R}$ and orthogonal matrix $\boldsymbol{Q}$.

Reparameterization Motivated by the Complete Factorization

We propose to re-parameterize a convolution layer with filter size $L+R$, using **learnable** column-orthogonal matrices $\{\boldsymbol{U}^{(i)}\}_{i=-L}^{R}$ and $\boldsymbol{Q}$, as

- convolution of R filters, $\left\{\boldsymbol{v}^{(i)} = [\boldsymbol{I} - \boldsymbol{U}^{(i)}\boldsymbol{U}^{(i)\top}, \boldsymbol{U}^{(i)}\boldsymbol{U}^{(i)\top}]\right\}_{i=1}^{R}$
- followed by convolution of an orthogonal matrix $\boldsymbol{Q}$, and then
- convolution of L filters $\left\{\boldsymbol{v}^{(-i)} = [\boldsymbol{U}^{(i)}\boldsymbol{U}^{(i)\top}, \boldsymbol{I} - \boldsymbol{U}^{(i)}\boldsymbol{U}^{(i)\top}]\right\}_{i=1}^{L}$.

Summary of Contributions

- Network is strictly orthogonal by design, no SVD, no additional computation cost.
- Establish equivalence between orthogonal convolution layer in the spatial domain and a Paraunitary system in the spectral domain
- A complete characterization of all Paraunitary matrices guarantees expressive power

Summary of Contributions

- Network is strictly orthogonal by design, no SVD, no additional computation cost.
- Establish equivalence between orthogonal convolution layer in the spatial domain and a Paraunitary system in the spectral domain
- A complete characterization of all Paraunitary matrices guarantees expressive power
- Develop customized polyphase transformation to spectral domain for variants of traditional convolutions (dilated, stride and group convolutions)
  - such that designing orthogonal layers for those variants of convolutions is equivalent to implementing a Paraunitary system in the spectral domain

Summary of Contributions

- Network is strictly orthogonal by design, no SVD, no additional computation cost.
- Establish equivalence between orthogonal convolution layer in the spatial domain and a Paraunitary system in the spectral domain
- A complete characterization of all Paraunitary matrices guarantees expressive power
- Develop customized polyphase transformation to spectral domain for variants of traditional convolutions (dilated, stride and group convolutions)
  - such that designing orthogonal layers for those variants of convolutions is equivalent to implementing a Paraunitary system in the spectral domain
- Scale to state-of-the-art deep architectures with skip connections, including deep ResNet, WideResNet, etc.
- Proper initialization for deep Lipschitz networks is proposed.

Evaluation of Orthogonality

Evaluating $\frac{\|y\|}{\|x\|}$ for Convolution Layers

Conv.	mean	std
Ours	$1 + 3.84 \times 10^{-8}$	7.32×10^{-8}
Cayley	$1 + 2.88 \times 10^{-4}$	1.90×10^{-4}
BCOP	$1 + 2.59 \times 10^{-3}$	6.14×10^{-3}
SVCM	0.571	3.31×10^{-3}
RKO	0.334	1.74×10^{-3}
OSSN	0.578	3.44×10^{-3}

Ours is more concentrated to 1 with orders of magnitude smaller variance.

Comparison with SOTA Orthogonal Convolution

		ResNet9				WideResNet10-10				WideResNet22-best	
ϵ	Test Acc.	Ours	Cayley	BCOP	RKO	Ours	Cayley	BCOP	RKO	Ours	Cayley
0	Clean	**82.19**	81.70	80.72	80.06	**84.09**	82.99	81.39	81.50	**88.23**	86.23
$\frac{36}{255}$	PGD	71.21	73.77	73.27	73.37	74.29	76.02	74.56	74.72	75.23	75.60

- For shallow networks: improved clean accuracies and comparable robust accuracies
- One of two existing methods that scale up to deep networks
- The only method that transforms SOTA architectures without modifications.

Comparison with SOTA Orthogonal Convolution

		ResNet9				WideResNet10-10				WideResNet22-best	
ϵ	Test Acc.	Ours	Cayley	BCOP	RKO	Ours	Cayley	BCOP	RKO	Ours	Cayley
0	Clean	**82.19**	81.70	80.72	80.06	**84.09**	82.99	81.39	81.50	**88.23**	86.23
$\frac{36}{255}$	PGD	71.21	73.77	73.27	73.37	74.29	76.02	74.56	74.72	75.23	75.60

- For shallow networks: improved clean accuracies and comparable robust accuracies
- One of two existing methods that scale up to deep networks
- The only method that transforms SOTA architectures without modifications.

Summary and Outlook

Summary

- **Generalization**: compression and data aware tighter generalization bound
- **Interpretation**: vigorous visual representation and improvement of transformers
- **Robustness**: provable defense, generating the worst possible attack

Outlook

- **Generalization**: combining other compression techniques
- **Interpretation**: extension to graph data
- **Robustness**: more realistic threat models

Summary and Outlook

Summary

- **Generalization**: compression and data aware tighter generalization bound
- **Interpretation**: vigorous visual representation and improvement of transformers
- **Robustness**: provable defense, generating the worst possible attack

Outlook

- **Generalization**: combining other compression techniques
- **Interpretation**: extension to graph data
- **Robustness**: more realistic threat models

Thanks

furongh@umd.edu