Geometric Deep Learning

Erik Schultheis, Kate Haitsiukevich, Çağlar Hızlı, Alison Pouplin, Vikas Garg

8.2.2024

Course overview

- Introductory lectures: Group Theory, Graph Neural Networks, GDL Blueprint, Manifolds
- Exercises (first set: next week; deadline 6.11.24)
- Presentations by You

Tentative Timeline

Week	Date	Session 1 (45')	Session 2 (45')
October - Period II			
Week 43	24.10	Introduction	Group Theory
Week 44	31.10	GNNs - Course I	GNNs - Course II
November - Period II			
Week 45	07.11	GNNs - Solutions	Sym. - Course
Week 46	14.11	Sym. - Solutions	Man. - Course I
Week 47	21.11	Man. - Course II	Paper
Week 48	28.11	Man. - Solutions	Paper
December- Period II			
Week 49	05.12	**Evaluation week**	
Week 50	12.12	**Evaluation week / Neurips**	
Week 51	19.12	Holidays	

Tentative Timeline

Week	Date	Session 1 (45’)	Session 2 (45’)
January - Period III			
Week 02	09.01	Paper	Paper
Week 03	16.01	Paper	Paper
Week 04	23.01	Paper	Paper
Week 05	30.01	Paper	Paper
February - Period III			
Week 06	6.02	Paper	Paper
Week 07	13.02	Paper	Paper
Week 08	20.02	Evaluation week	

Resources

- Geometric Deep Learning book
- Graph Represetation Learning book
- Equivariant and coordinate-independent CNNs book

Evaluation

- Participation in course
- Notebook exercises
- Paper presentation
- Writing assignments

Part I: Why Geometric Deep Learning

Learning on generic vector spaces

classification with real-valued data

Given training dataset $x_1, \ldots, x_n \in \mathbb{R}^d$ and labels $y_1, \ldots, y_n \in [C]$, generate a function so that

$$\mathbb{E}_{X,Y}[\ell(f(X), Y)] \to \min \ .$$

Learning on generic vector spaces

classification with real-valued data

Given training dataset $x_1, \ldots, x_n \in \mathbb{R}^d$ and labels $y_1, \ldots, y_n \in [C]$, generate a function so that

$$\mathbb{E}_{X,Y}[\ell(f(X), Y)] \to \min \ .$$

Requires *huge* amounts of data.

Real data usually contains structure

Handwritten digits, original

As a vector space, these contain the same information

Real data usually contains structure

Handwritten digits, permuted

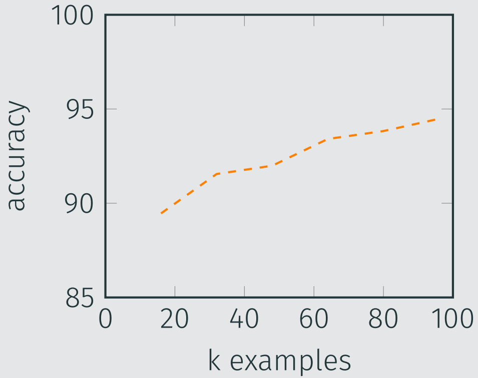
As a vector space, these contain the same information

Real data usually contains structure

Handwritten digits, permuted

As a vector space, these contain the same information

Structure enables sample efficiency



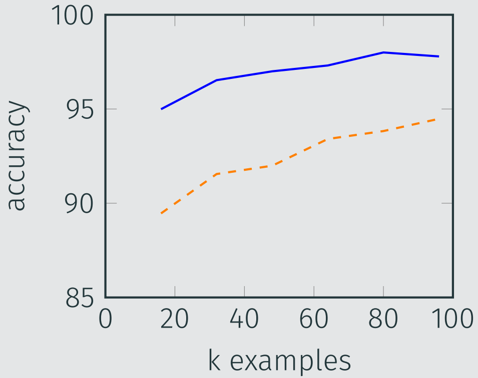
MLP (permutation invariant)

Real data usually contains structure

Handwritten digits, permuted

As a vector space, these contain the same information

Structure enables sample efficiency



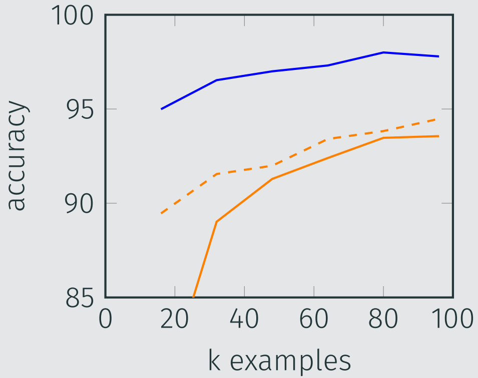
CNN (exploits structure)

Real data usually contains structure

Handwritten digits, permuted

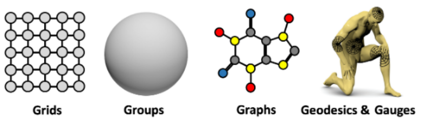
As a vector space, these contain the same information

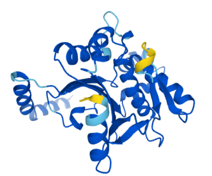
Structure enables sample efficiency



CNN (permuted images)

Examples of structured data





Architecture	Domain	Symmetry
CNN	Grid	Translation
Spherical CNN	Sphere / SO(3)	Rotation SO(3)
Mesh CNN	Manifold	Gauge Symmetry
GNN	Graph	Permutation
Deep Sets	Set	Permutation
Transformer	Complete graph	Permutation
LSTM	1D Grid	Time translation

`https://arxiv.org/pdf/2104.13478.pdf`; Jumper et al. (2021)

Part II: Statistical Learning Theory

Key Components of Statistical Learning

Data distribution

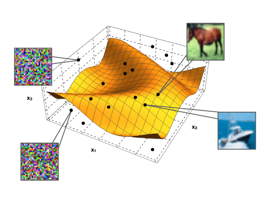
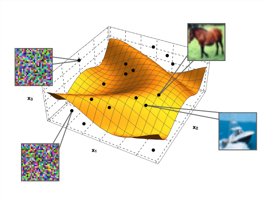


Image: Goldt et al. (2020)

Key Components of Statistical Learning

Data distribution



models

Approximation model

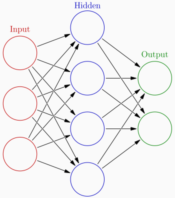
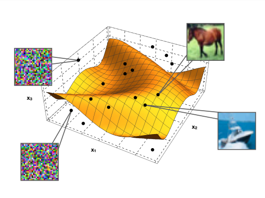


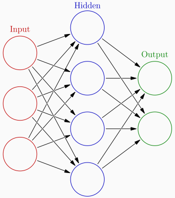
Image: Glosser.ca, CC BY-SA 3.0, via Wikimedia Commons

Key Components of Statistical Learning

Data distribution



Approximation model



models

compares

Error Metric

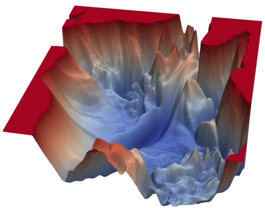


Image: Li et al. (2018)

Key Components of Statistical Learning

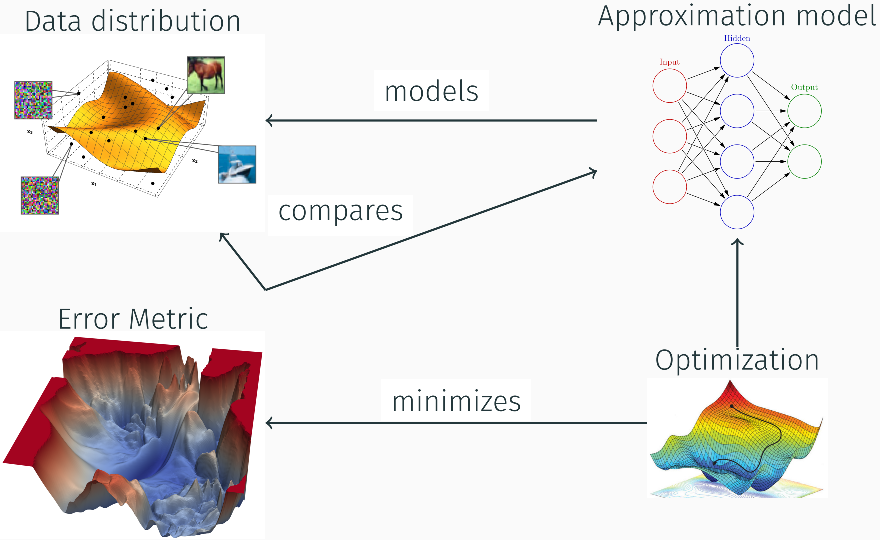


Image: Amini et al. (2019)

Data distribution

Definitions

- data point/instance $x_i \in \mathcal{X} = \mathbb{R}^d$
- label $y_i \in \mathbb{R}$ (regression), $y_i \in [C]$ (classification)
- data set: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$
- data distribution: $x_i \sim_{\text{i.i.d.}} \mathbb{P}$
- ground truth: $y_i = f^*(x_i)$

Example

- $x_i =$  , $y_i = 2$
- $\mathcal{D}$: 50000 image-label pairs
- $\mathbb{P}$ distribution of hand-drawn images; $\mathbb{P}\,($$) > \mathbb{P}\,($$)$
- $f^*\,($$) = 2$

Error Measure

Definitions

- loss $\ell : \mathbb{R} \times \mathbb{R} \longrightarrow \mathbb{R}_{\geq 0}$
- risk $\mathcal{R}[f] = \mathbb{E}[\ell(f(X), f^*(X))]$
- empirical risk $\hat{\mathcal{R}}[f] = n^{-1} \sum_i [\ell(f(x_i), y_i]$

Examples

- Squared error, absolute error (regression)
- cross-entropy (classification)

Error Measure

Definitions

- loss $\ell : \mathbb{R} \times \mathbb{R} \longrightarrow \mathbb{R}_{\geq 0}$
- risk $\mathcal{R}[f] = \mathbb{E}[\ell(f(X), f^*(X))]$
- empirical risk $\hat{\mathcal{R}}[f] = n^{-1} \sum_i [\ell(f(x_i), y_i]$

Examples

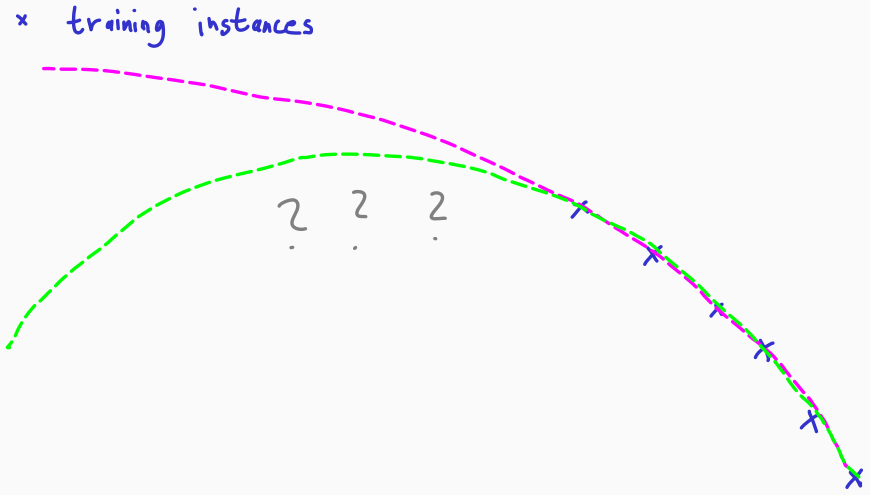
- Squared error, absolute error (regression)
- cross-entropy (classification)

Task: minimize $\mathcal{R}[f]$ with access only to $\hat{\mathcal{R}}[f]$

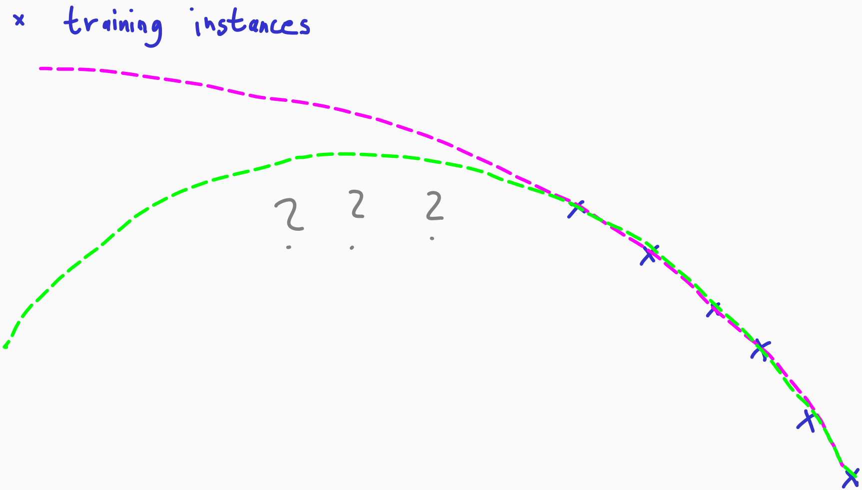
We need to make some concessions

× training instances

We need to make some concessions



We need to make some concessions



We might be unlucky with the i.i.d. sample we draw

Most of the time, we should be mostly correct

Probably Approximately Correct (PAC) Learning

An algorithm A is a PCA-learner if there exists a function $m(\epsilon, \delta)$, such that for every $\epsilon, \delta \in (0, 1)$ and every distribution $\mathbb{P}$, when running the learning algorithm on a i.i.d. sample S of size $m(\epsilon, \delta)$, with probability at least δ we have

$$\mathcal{R}[A(S)] \leq \epsilon\,.$$

Shalev-Shwartz and Ben-David. (2014). *Understanding Machine Learning - From Theory to Algorithms.*

Most of the time, we should be mostly correct

Probably Approximately Correct (PAC) Learning

An algorithm A is a PCA-learner if there exists a function $m(\epsilon, \delta)$, such that for every $\epsilon, \delta \in (0,1)$ and every distribution $\mathbb{P}$, when running the learning algorithm on a i.i.d. sample S of size $m(\epsilon, \delta)$, with probability at least δ we have

$$\mathcal{R}[A(S)] \leq \epsilon\,.$$

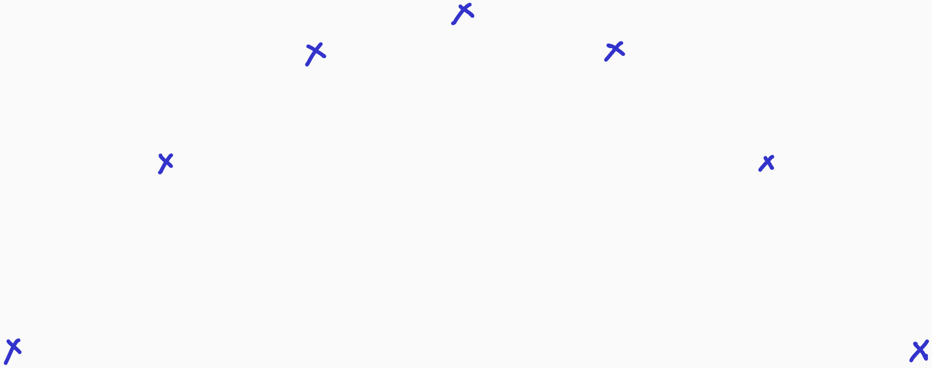
Probably (with probability δ) we are *approximately* (with tolerance ϵ) correct.

Can we achieve that, at least?

Shalev-Shwartz and Ben-David. (2014). *Understanding Machine Learning - From Theory to Algorithms.*

Minimizing empiral risk might not minimize population risk

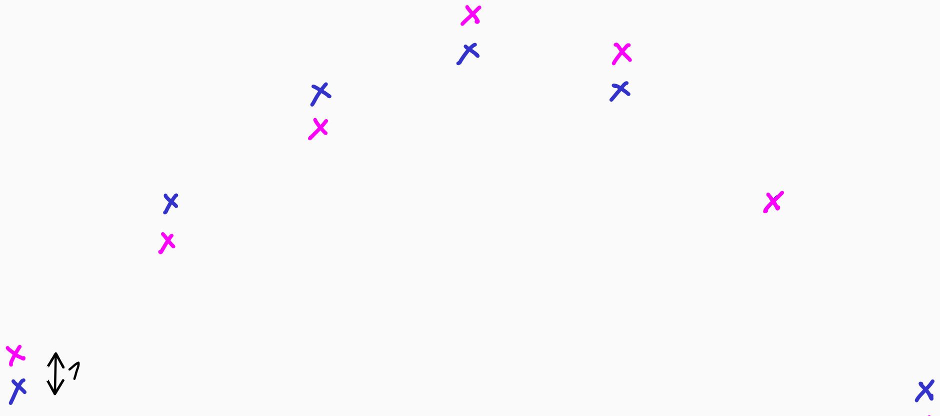
× training instances



Minimizing empiral risk might not minimize population risk

× training instances

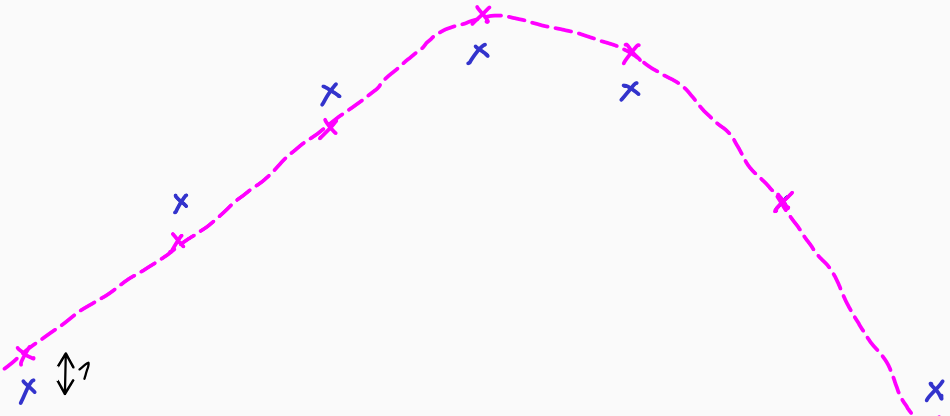
×: $\hat{R} = 6/7$



Minimizing empiral risk might not minimize population risk

× training instances

×: $\hat{R} = 6/7 \quad R \approx 1$

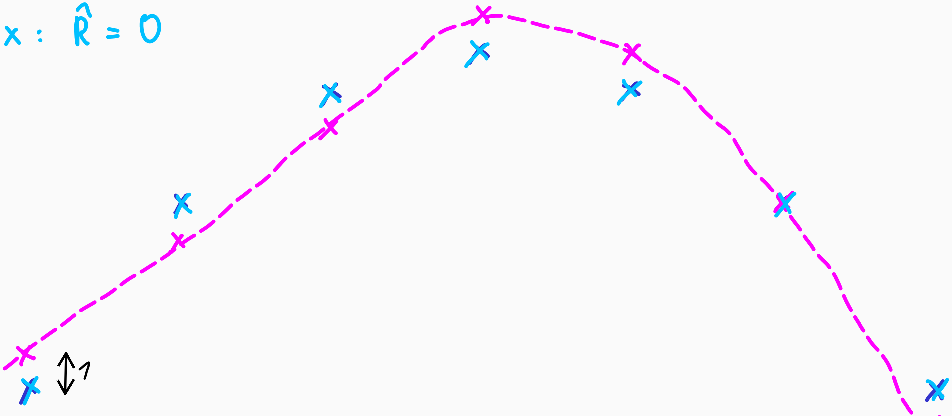


Minimizing empiral risk might not minimize population risk

× training instances

×: $\hat{R} = 6/7 \quad R \approx 1$

×: $\hat{R} = 0$

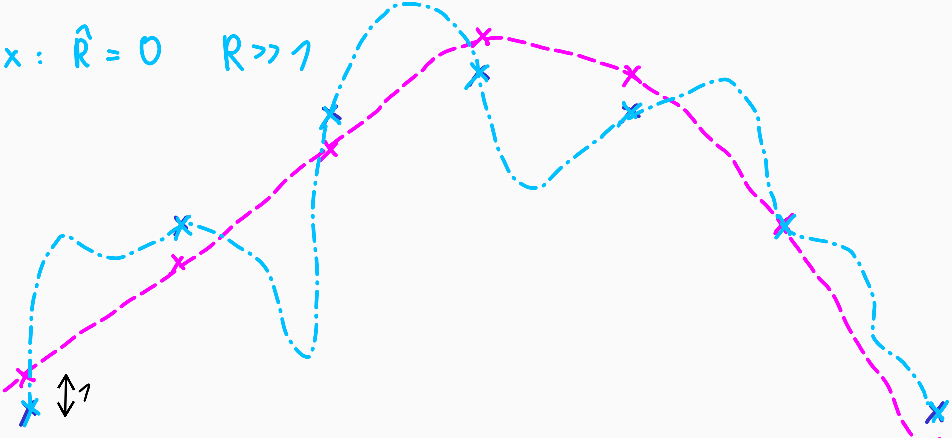


Minimizing empiral risk might not minimize population risk

× training instances

×: $\hat{R} = 6/7$ $R \approx 1$

×: $\hat{R} = 0$ $R \gg 1$



We need to make some concessions

No free lunch theorem

Let A be any learning algorithm for the task of binary classification with 0-1 loss over a domain $\mathcal{X}$. Let m be any number smaller than $|\mathcal{X}|/2$, representing a training set size. Then, there exists a distribution $\mathbb{P}$ over $\mathcal{X} \times \{0, 1\}$ such that

- There exists a function $f : \mathcal{X} \longrightarrow \{0, 1\}$ with $\mathcal{R}[f] = 0$
- With probability of at least 1/7 over the choice of training set $\mathcal{S} \sim \mathbb{P}^m$ we have $\mathcal{R}[A(S)] \geq 1/8$

Shalev-Shwartz and Ben-David. (2014). *Understanding Machine Learning - From Theory to Algorithms.*

We need to make some concessions

No free lunch theorem

Let A be any learning algorithm for the task of binary classification with 0-1 loss over a domain $\mathcal{X}$. Let m be any number smaller than $|\mathcal{X}|/2$, representing a training set size. Then, there exists a distribution $\mathbb{P}$ over $\mathcal{X} \times \{0,1\}$ such that

- There exists a function $f : \mathcal{X} \longrightarrow \{0,1\}$ with $\mathcal{R}[f] = 0$
- With probability of at least 1/7 over the choice of training set $\mathcal{S} \sim \mathbb{P}^m$ we have $\mathcal{R}[A(S)] \geq 1/8$

$\Longrightarrow$ **We need to restrict the set of admissible functions**

Shalev-Shwartz and Ben-David. (2014). *Understanding Machine Learning - From Theory to Algorithms.*

We need to make some concessions

No free lunch theorem

Let A be any learning algorithm for the task of binary classification with 0-1 loss over a domain $\mathcal{X}$. Let m be any number smaller than $|\mathcal{X}|/2$, representing a training set size. Then, there exists a distribution $\mathbb{P}$ over $\mathcal{X} \times \{0,1\}$ such that

- There exists a function $f : \mathcal{X} \longrightarrow \{0,1\}$ with $\mathcal{R}[f] = 0$
- With probability of at least 1/7 over the choice of training set $\mathcal{S} \sim \mathbb{P}^m$ we have $\mathcal{R}[A(S)] \geq 1/8$

$\Longrightarrow$ **We need to restrict the set of admissible functions**

$\Longrightarrow$ **approximation model**

Shalev-Shwartz and Ben-David. (2014). *Understanding Machine Learning - From Theory to Algorithms.*

Approximation model and complexity measure

Hypothesis class

The model (or hypothesis) class is a subset $\mathcal{F} \subset \{f : \mathcal{X} \longrightarrow \mathbb{R}\}$

Examples

- Polynomials up to degree k
- Neural networks of a given architecture

Approximation model and complexity measure

Hypothesis class

The model (or hypothesis) class is a subset $\mathcal{F} \subset \{f : \mathcal{X} \longrightarrow \mathbb{R}\}$

Examples

- Polynomials up to degree *k*
- Neural networks of a given architecture

Complexity measure

A non-negative mapping $\gamma : \mathcal{F} \longrightarrow \mathbb{R}_{\geq 0}$ that captures how "complex" a hypothesis is.

Examples

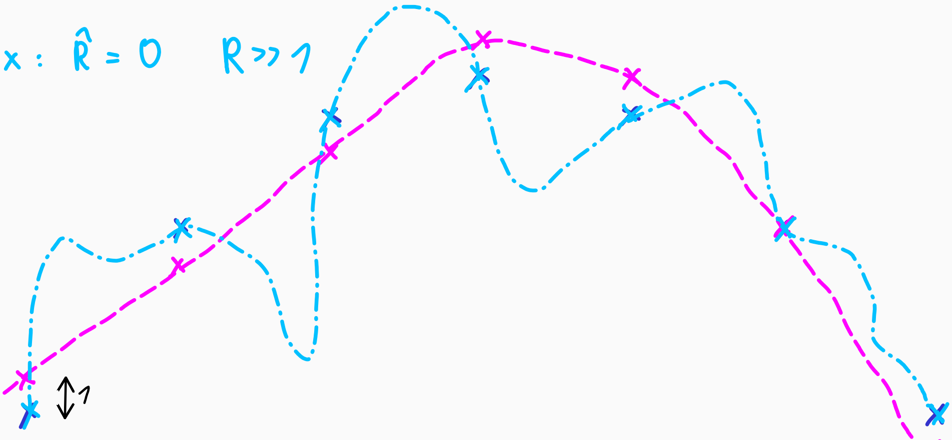
- Degree of polynomial
- Number of neurons in network

Less complex functions less likely to overfit

× training instances

×: $\hat{R} = 6/7 \quad R \approx 1$

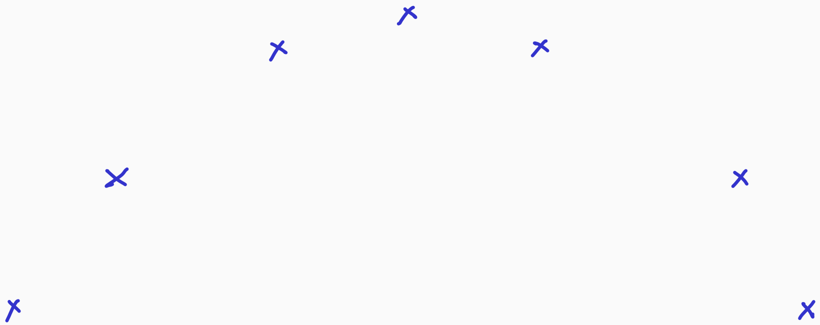
×: $\hat{R} = 0 \quad R \gg 1$



Now we get to make a trade-off

Restrict to simple hypothesis $\mathcal{F}_\beta := \{f \in \mathcal{F} : \gamma(f) \leq \beta\}$:

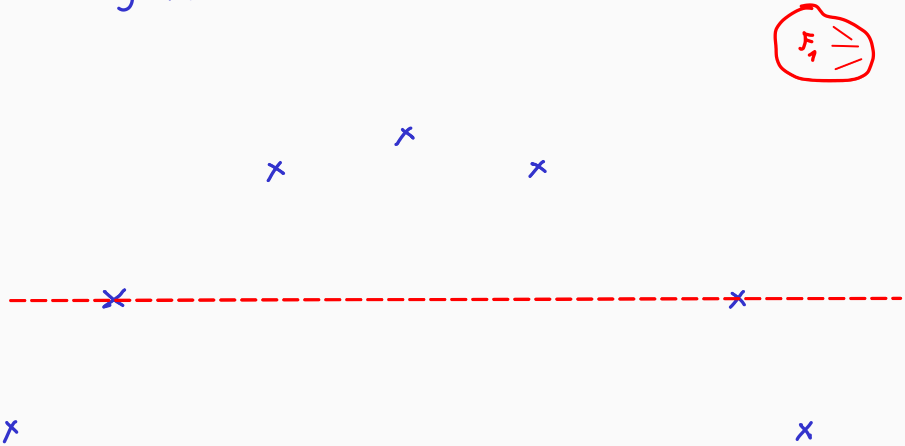
× training instances



Now we get to make a trade-off

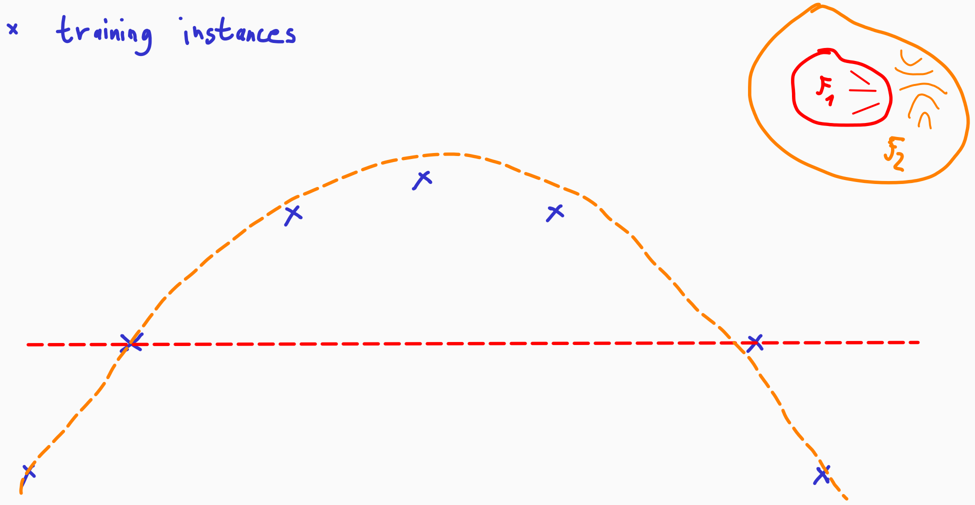
Restrict to simple hypothesis $\mathcal{F}_\beta := \{f \in \mathcal{F} : \gamma(f) \leq \beta\}$:

× training instances



Now we get to make a trade-off

Restrict to simple hypothesis $\mathcal{F}_\beta := \{f \in \mathcal{F} : \gamma(f) \leq \beta\}$:



We can generalize this example to a generic error decomposition

$$
\begin{aligned}
\mathcal{R}[\hat{f}] - \inf_{f\in\mathcal{F}} \mathcal{R}[f] &= \left(\mathcal{R}[\hat{f}] - \inf_{f\in\mathcal{F}_\delta} \mathcal{R}[f]\right) + \left(\inf_{f\in\mathcal{F}_\delta} \mathcal{R}[f] - \inf_{f\in\mathcal{F}} \mathcal{R}[f]\right) \\
&= \left(\hat{\mathcal{R}}[\hat{f}] - \inf_{f\in\mathcal{F}_\delta} \hat{\mathcal{R}}[f]\right) + \left(\inf_{f\in\mathcal{F}_\delta} \hat{\mathcal{R}}[f] - \inf_{f\in\mathcal{F}_\delta} \mathcal{R}[f]\right) + \left(\mathcal{R}[\hat{f}] - \hat{\mathcal{R}}[\hat{f}]\right) + \epsilon_{\text{approx}} \\
&\leq \epsilon_{\text{opt}} + 2 \sup_{f\in\mathcal{F}_\delta} \left|\hat{\mathcal{R}}[f] - \mathcal{R}[f]\right| + \epsilon_{\text{approx}} \\
&= \epsilon_{\text{opt}} + \epsilon_{\text{stat}} + \epsilon_{\text{approx}}
\end{aligned}
$$

Controlling sources of error simulataneously is challenging

- In practice, SGD seems to be successfull at minimizing ϵ_{opt}
- Small hypothesis class $\mathcal{F}_\delta$: $\downarrow \epsilon_{\text{stat}}$, $\uparrow \epsilon_{\text{approx}}$
- Large hypothesis class $\mathcal{F}_\delta$: $\uparrow \epsilon_{\text{stat}}$, $\downarrow \epsilon_{\text{approx}}$

Lipschitzness is too weak – Staticical curse of dimensionality

Lipschitz

A function f is L-Lipschitz if $\forall x_1, x_2 \in \mathcal{X}$:

$$\|f(x_1) - f(x_2)\| \leq L \cdot \|x_1 - x_2\| .$$

$\implies$ limits rate of variation of the function

Curse of dimensionality

How many training points do we need to learn a 1-Lipschitz function on a d-dimensional hypercube?

Barron functions too strong – Approximation curse of dimensionality

A function f with Fourier-transform $\hat{f}$ is in the Barron-class, if

$$\int \hat{f}(\omega)\|\omega\|_2^2 \mathrm{d}\omega < \infty\,.$$

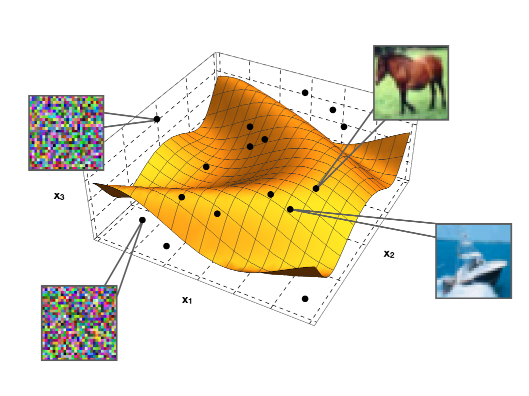
Barron functions too strong – Approximation curse of dimensionality

A function f with Fourier-transform $\hat{f}$ is in the Barron-class, if

$$\int \hat{f}(\omega)\|\omega\|_2^2 \mathrm{d}\omega < \infty\,.$$

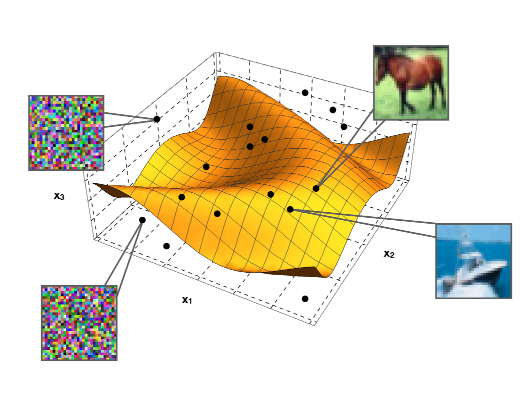
$\implies$ High frequency components need to decay faster than $\|\omega\|^{-3}$; functions need to be very smooth.

The input data distribution is assumed to have low-dimensional structure *embedded* in a high-dimensional space



Goldt et al. (2020). "Modeling the Influence of Data Structure on Learning in Neural Networks: The Hidden Manifold Model"

The input data distribution is assumed to have low-dimensional structure *embedded* in a high-dimensional space



Can we exploit that?

Goldt et al. (2020). “Modeling the Influence of Data Structure on Learning in Neural Networks: The Hidden Manifold Model”

Part III: Signals over geometric domains

Geometric Domains

MNIST image: $x \in \mathbb{R}^{28\times 28}$ as a vector. Defined on a grid of 28×28 *pixels*.
$\implies$ Mapping from pixel to intensity.

Geometric Domains

MNIST image: $x \in \mathbb{R}^{28\times 28}$ as a vector. Defined on a grid of 28×28 *pixels*.

$\implies$ Mapping from pixel to intensity.

Signal

Given some domain Ω, a signal is a mapping

$$x\colon \Omega \longrightarrow \mathcal{C}$$

from domain locations to c-dimensional vectors (*channels*) in vector space $\mathcal{C}$.

The space of all signals is $\mathcal{X}(\Omega, \mathcal{C})$.

Geometric Domains

MNIST image: $x \in \mathbb{R}^{28\times 28}$ as a vector. Defined on a grid of 28×28 *pixels*.

$\implies$ Mapping from pixel to intensity.

Signal

Given some domain Ω, a signal is a mapping

$$x\colon \Omega \longrightarrow \mathcal{C}$$

from domain locations to c-dimensional vectors (*channels*) in vector space $\mathcal{C}$.

The space of all signals is $\mathcal{X}(\Omega, \mathcal{C})$.

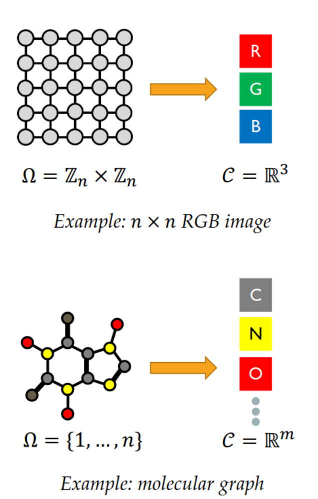


Image: `https://geometricdeeplearning.com/lectures/`

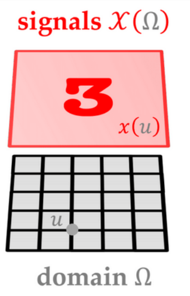
We can imbue signals with a Hilbert-space structure

Addition and scalar multiplication

For two signals x and y on Ω, $\alpha, \beta \in \mathbb{R}$, define $\alpha x + \beta y$ through

$$(\alpha x + \beta y)(\omega) := \alpha x(\omega) + \beta y(\omega) \quad \forall \omega \in \Omega\,.$$

$\implies$ vector space



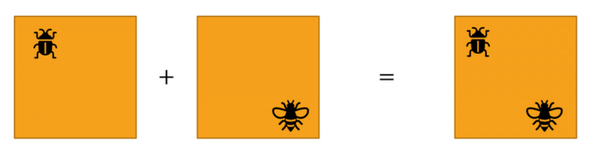
We can imbue signals with a Hilbert-space structure

Addition and scalar multiplication

For two signals x and y on Ω, $\alpha, \beta \in \mathbb{R}$, define $\alpha x + \beta y$ through

$$(\alpha x + \beta y)(\omega) := \alpha x(\omega) + \beta y(\omega) \quad \forall \omega \in \Omega\,.$$

$\implies$ vector space



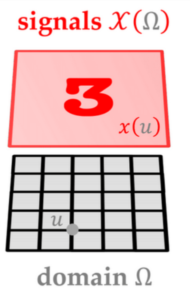


Image: `https://geometricdeeplearning.com/lectures/`

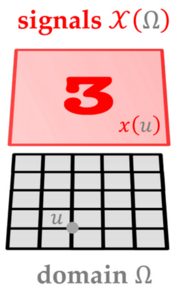
We can imbue signals with a Hilbert-space structure

Inner product

With an inner product $\langle \cdot, \cdot \rangle_{\mathcal{C}}$ on $\mathcal{C}$, and a measure μ on Ω, define inner product on $\mathcal{X}(\Omega, \mathcal{C})$:

$$\langle x, y \rangle := \int_{\Omega} \langle x(\omega), y(\omega) \rangle_{\mathcal{C}} \, \mathrm{d}\mu(\omega) \,.$$

$\implies$ Hilbert space



We can imbue signals with a Hilbert-space structure

Inner product

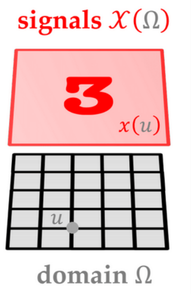
With an inner product $\langle \cdot, \cdot \rangle_{\mathcal{C}}$ on $\mathcal{C}$, and a measure μ on Ω, define inner product on $\mathcal{X}(\Omega, \mathcal{C})$:

$$\langle x, y \rangle := \int_{\Omega} \langle x(\omega), y(\omega) \rangle_{\mathcal{C}} \, \mathrm{d}\mu(\omega) \, .$$

$\implies$ Hilbert space

Example: MNIST - single channel, counting measure

$$\langle x, y \rangle = \sum_{i=1}^{28} \sum_{j=1}^{28} x[i,j] \cdot y[i,j]$$



Symmetries

A transformation of an object that leaves the object unchanged.

Image: Antonio Kless, CC BY-SA 4.0, via Wikimedia Commons

Symmetries

A transformation of an object that leaves the object unchanged.

Symmetry of the label function

Recall: $f^* : \mathcal{X} \longrightarrow \mathcal{Y}$ ground-truth label function.

A transformation $g : \mathcal{X} \longrightarrow \mathcal{X}$ is a symmetry of the label function, if $f^* \equiv f^* \circ g$.

Image: Antonio Kless, CC BY-SA 4.0, via Wikimedia Commons

Symmetries

A transformation of an object that leaves the object unchanged.

Symmetry of the label function

Recall: $f^* : \mathcal{X} \longrightarrow \mathcal{Y}$ ground-truth label function.

A transformation $g : \mathcal{X} \longrightarrow \mathcal{X}$ is a symmetry of the label function, if $f^* \equiv f^* \circ g$.

Example: Horizontal flip

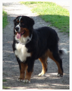
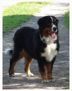
$f^*\Big($  $\Big) = f^*\Big($  $\Big) =$ "dog" .

Image: Antonio Kless, CC BY-SA 4.0, via Wikimedia Commons

Symmetries of the label function

- Any bijective function that respects class boundaries is a symmetry of the label function
- $\implies$ if we knew all symmetries, a *single* instance per class would be enough to learn.
- If we know some symmetries, less training data is needed.

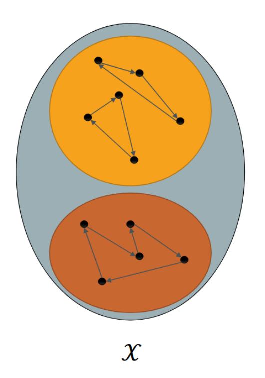


Image: `https://geometricdeeplearning.com/lectures/`

Symmetries of the label function

- Any bijective function that respects class boundaries is a symmetry of the label function
- $\implies$ if we knew all symmetries, a *single* instance per class would be enough to learn.
- If we know some symmetries, less training data is needed.
- Can exploit symmetries of the underlying domain.

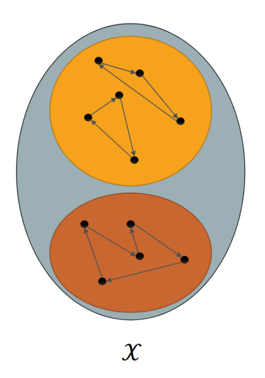


Image: `https://geometricdeeplearning.com/lectures/`

Symmetries of geometric domains

- A transformation $g : \Omega \longrightarrow \Omega$ is a symmetry of the domain Ω if it preserves its structure.

Symmetries of geometric domains

- A transformation $g : \Omega \longrightarrow \Omega$ is a symmetry of the domain Ω if it preserves its structure.
- Permutation of elements in a set preserves set membership
- Euclidian isometries (rotation, translation, reflection) preserve angles and distances in Euclidian spaces ($\mathbb{R}^d$)
- Diffeomorphism preserves manifold structure

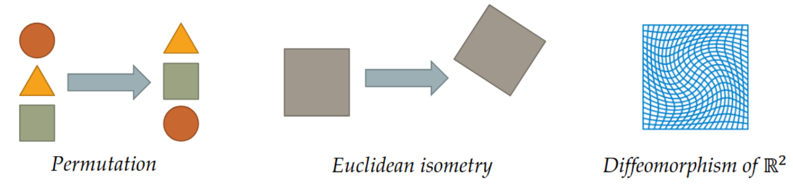


Image: `https://geometricdeeplearning.com/lectures/`

We can lift symmetries on the domain to symmetries on signals

Given a symmetry $g : \Omega \longrightarrow \Omega$, we can define a symmetry $\tilde{g}$ on $\mathcal{X}(\Omega, \mathcal{C})$ through:

$$\tilde{g}(x)(\omega) = x(g^{-1}(\omega))\,.$$

We can lift symmetries on the domain to symmetries on signals

Given a symmetry $g : \Omega \longrightarrow \Omega$, we can define a symmetry $\tilde{g}$ on $\mathcal{X}(\Omega, \mathcal{C})$ through:

$$\tilde{g}(x)(\omega) = x(g^{-1}(\omega)) \, .$$

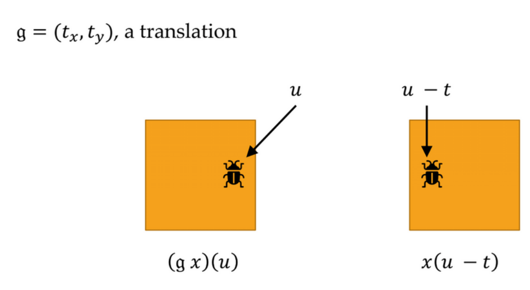


Image: `https://geometricdeeplearning.com/lectures/`

The mathematical theory is symmetries is **group theory**

Part IV: Group Theory

What is a group?

Definition

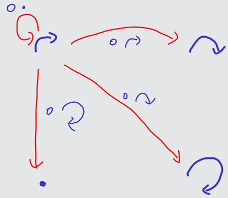
A collection of *abstract* transformations $G = (\{u, v, w, \ldots\}, \circ)$ that can be combined with a "multiplication" $\circ$:

Closedness: $u \circ v \in G$

Associativity: $u \circ (v \circ w) = (u \circ v) \circ w$

Neutral Element: $\exists e \in G : eu = ue = u.$

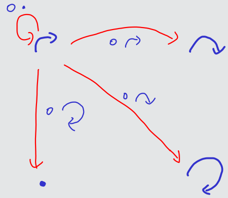
Inverse: $\forall u \in G : \exists u^{-1} : u^{-1}u = e$



What is a group?

Definition

A collection of *abstract* transformations $G = (\{u, v, w, \dots\}, \circ)$ that can be combined with a "multiplication" $\circ$:



Closedness: $u \circ v \in G$

Associativity: $u \circ (v \circ w) = (u \circ v) \circ w$

Neutral Element: $\exists e \in G : eu = ue = u.$

Inverse: $\forall u \in G : \exists u^{-1} : u^{-1}u = e$

Examples

Natural numbers $\mathbb{N} = (\{1, 2, \dots\}, +)$

What is a group?

Definition

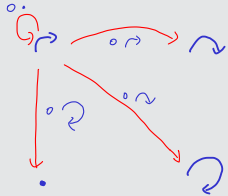
A collection of *abstract* transformations $G = (\{u, v, w, \ldots\}, \circ)$ that can be combined with a “multiplication” $\circ$:

Closedness: $u \circ v \in G$

Associativity: $u \circ (v \circ w) = (u \circ v) \circ w$

Neutral Element: $\exists e \in G : eu = ue = u.$

Inverse: $\forall u \in G : \exists u^{-1} : u^{-1}u = e$



Examples

Natural numbers $\mathbb{N} = (\{1, 2, \ldots\}, +)$ No

What is a group?

Definition

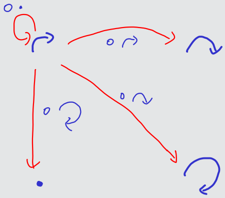
A collection of *abstract* transformations $G = (\{u, v, w, \ldots\}, \circ)$ that can be combined with a "multiplication" $\circ$:

Closedness: $u \circ v \in G$

Associativity: $u \circ (v \circ w) = (u \circ v) \circ w$

Neutral Element: $\exists e \in G : eu = ue = u.$

Inverse: $\forall u \in G : \exists u^{-1} : u^{-1}u = e$



Examples

Natural numbers $\mathbb{N} = (\{1, 2, \ldots\}, +)$ No

Integers $\mathbb{Z} = (\{\ldots, -1, 0, 1, \ldots\}, +)$

What is a group?

Definition

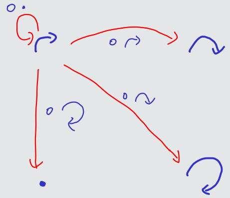
A collection of *abstract* transformations $G = (\{u, v, w, \ldots\}, \circ)$ that can be combined with a “multiplication” $\circ$:

Closedness: $u \circ v \in G$

Associativity: $u \circ (v \circ w) = (u \circ v) \circ w$

Neutral Element: $\exists e \in G : eu = ue = u.$

Inverse: $\forall u \in G : \exists u^{-1} : u^{-1}u = e$



Examples

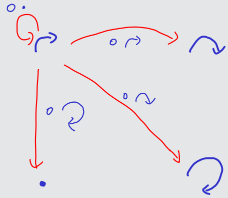
Natural numbers $\mathbb{N} = (\{1, 2, \ldots\}, +)$ No

Integers $\mathbb{Z} = (\{\ldots, -1, 0, 1, \ldots\}, +)$ Yes

What is a group?

Definition

A collection of *abstract* transformations $G = (\{u, v, w, \ldots\}, \circ)$ that can be combined with a "multiplication" $\circ$:



Closedness: $u \circ v \in G$

Associativity: $u \circ (v \circ w) = (u \circ v) \circ w$

Neutral Element: $\exists e \in G : eu = ue = u.$

Inverse: $\forall u \in G : \exists u^{-1} : u^{-1}u = e$

Examples

Natural numbers $\mathbb{N} = (\{1, 2, \ldots\}, +)$ No

Integers $\mathbb{Z} = (\{\ldots, -1, 0, 1, \ldots\}, +)$ Yes

Nonzero reals $\mathbb{R}^* = (\mathbb{R} \setminus \{0\}, \cdot)$

What is a group?

Definition

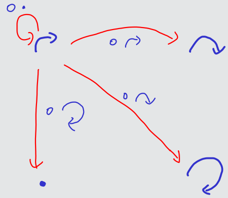
A collection of *abstract* transformations $G = (\{u, v, w, \dots\}, \circ)$ that can be combined with a "multiplication" $\circ$:

Closedness: $u \circ v \in G$

Associativity: $u \circ (v \circ w) = (u \circ v) \circ w$

Neutral Element: $\exists e \in G : eu = ue = u.$

Inverse: $\forall u \in G : \exists u^{-1} : u^{-1}u = e$



Examples

Natural numbers $\mathbb{N} = (\{1, 2, \dots\}, +)$ No

Integers $\mathbb{Z} = (\{\dots, -1, 0, 1, \dots\}, +)$ Yes

Nonzero reals $\mathbb{R}^* = (\mathbb{R} \setminus \{0\}, \cdot)$ Yes

What is a group?

Definition

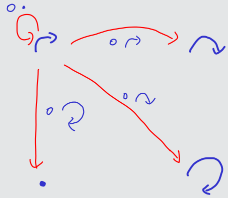
A collection of *abstract* transformations $G = (\{u, v, w, \ldots\}, \circ)$ that can be combined with a "multiplication" $\circ$:

Closedness: $u \circ v \in G$

Associativity: $u \circ (v \circ w) = (u \circ v) \circ w$

Neutral Element: $\exists e \in G : eu = ue = u.$

Inverse: $\forall u \in G : \exists u^{-1} : u^{-1}u = e$



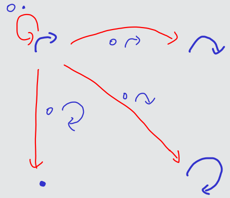
Examples

Natural numbers $\mathbb{N} = (\{1, 2, \ldots\}, +)$	No	Nonzero reals $\mathbb{R}^* = (\mathbb{R} \setminus \{0\}, \cdot)$	Yes
Integers $\mathbb{Z} = (\{\ldots, -1, 0, 1, \ldots\}, +)$	Yes	Nonzero reals $\mathbb{R}^* = (\mathbb{R} \setminus \{0\}, +)$	

What is a group?

Definition

A collection of *abstract* transformations $G = (\{u, v, w, \dots\}, \circ)$ that can be combined with a "multiplication" $\circ$:



Closedness: $u \circ v \in G$

Associativity: $u \circ (v \circ w) = (u \circ v) \circ w$

Neutral Element: $\exists e \in G : eu = ue = u.$

Inverse: $\forall u \in G : \exists u^{-1} : u^{-1}u = e$

Examples

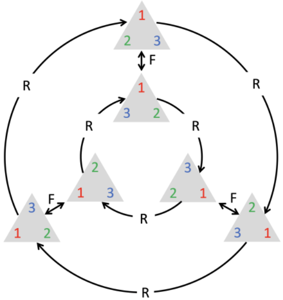
Natural numbers $\mathbb{N} = (\{1, 2, \dots\}, +)$	No	Nonzero reals $\mathbb{R}^* = (\mathbb{R} \setminus \{0\}, \cdot)$	Yes
Integers $\mathbb{Z} = (\{\dots, -1, 0, 1, \dots\}, +)$	Yes	Nonzero reals $\mathbb{R}^* = (\mathbb{R} \setminus \{0\}, +)$	No

The group captures the *structure* of how transformations combine

Structurally, these
are the same group:
$(\{0\}, +) \equiv (\{1\}, \cdot)$

The group captures the *structure* of how transformations combine

Structurally, these are the same group: $(\{0\}, +) \equiv (\{1\}, \cdot)$



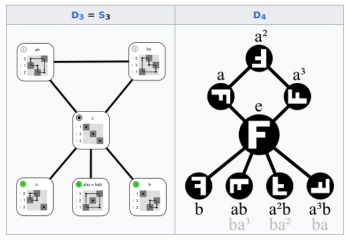
	id	R	R^2	F	FR	FR^2
id	id	R	R^2	F	FR	FR^2
R	R	R^2	id	RF	RFR	RFR^2
R^2	R^2	id	R	R^2F	R^2FR	R^2FR^2
F	F	FR	FR^2	id	R	FR
FR	FR	FR^2	F	FRF	FRFR	$FRFR^2$
FR^2	FR^2	F	FR	FR^2F	FR^2FR	FR^2FR^2

Figure 4: Left: an equilateral triangle with corners labelled by $1, 2, 3$, and all possible rotations and reflections of the triangle. The group D_3 of rotation/reflection symmetries of the triangle is generated by only two elements (rotation by 60° R and reflection F) and is the same as the group Σ_3 of permutations of three elements. Right: the multiplication table of the group D_3. The element in the row $\mathfrak{g}$ and column $\mathfrak{h}$ corresponds to the element $\mathfrak{gh}$.

https://arxiv.org/pdf/2104.13478.pdf

Some important groups

$C(n)$	Cyclic group	$x \mapsto x + 1 \bmod n$
$D(n)$	Dihedral	Vertices of a regular n-gon under reflection and rotation
$S(n)$	Permutation	Arbitrary permutation
$\mathrm{SO}(n)$	Special ortho.	Rotations in $\mathbb{R}^n$
$\mathrm{O}(n)$	Orthogonal	Rotations and reflections in $\mathbb{R}^n$
$\mathrm{GL}(n)$	General linear	Invertible linear transforms of I



$$C(n) \subset D(n) \subset S(n)$$

$$\mathrm{SO}(n) \subset \mathrm{O}(n) \subset \mathrm{GL}(n)$$

Group Actions: Making group elements actually transform something

Definition

A group action $\mathcal{A}$ of a group G *on* some set Ω is a mapping $G \times \Omega \longrightarrow \Omega$, $(g, \omega) \mapsto g.\omega$, that is *compatible* with the group structure, that is,

$$(g \circ h).\omega = g.(h.\omega)$$

$$e.\omega = \omega$$

Group Actions: Making group elements actually transform something

Definition

A group action $\mathcal{A}$ of a group G *on* some set Ω is a mapping $G \times \Omega \longrightarrow \Omega$, $(g, \omega) \mapsto g.\omega$, that is *compatible* with the group structure, that is,

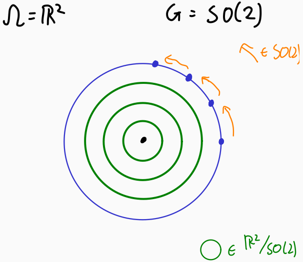
$$(g \circ h).\omega = g.(h.\omega)$$
$$e.\omega = \omega$$

Examples: The same group can act differently

- The group $(\mathbb{R}, +)$ acting on the set $\mathbb{R}$: $u.v = u + v$
- The group $(\mathbb{R}, +)$ acting on the vector space $\mathbb{R}^2$ as a horizontal shift: $u.(x, y) = (x + u, y)$
- The group $(\mathbb{R}, +)$ acting on the vector space $\mathbb{R}^2$ as a vertical shift: $u.(x, y) = (x, y + u)$

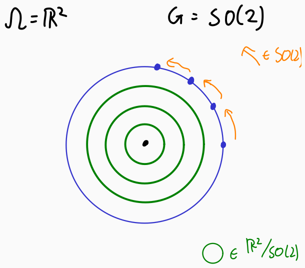
Orbits

- The trajectory that can be reached by applying any group element to a fixed starting point
- $G.\omega = \{g.\omega \mid g \in G\}$
- Partitions the space into equivalence classes $[\omega] \in \Omega/G$



Orbits

- The trajectory that can be reached by applying any group element to a fixed starting point
- $G.\omega = \{g.\omega \mid g \in G\}$
- Partitions the space into equivalence classes $[\omega] \in \Omega/G$



One sample per orbit enough to learn label function

Summary and Outlook

Today

- Statistical learning theory: Need assumptions to enable successfull learning in high-dimensional spaces
- Inputs as signals on a geometric space: Allows us to exploit symmetries of the underlying structure
- Group theory: Mathematical framework for symmetries

Next Week

- Sets and Graphs: Permutation group
- Graph neural networks as a first instantiation of the *geometric deep learning blueprint*

References

[1] Alexander Amini et al. *Spatial Uncertainty Sampling for End-to-End Control.* 2019. arXiv: `1805.04829`.

[2] Sebastian Goldt et al. **"Modeling the Influence of Data Structure on Learning in Neural Networks: The Hidden Manifold Model".** In: *Phys. Rev. X* 10 (4 Dec. 2020), p. 041044.

[3] John Jumper et al. **"Highly accurate protein structure prediction with AlphaFold".** In: *nature* 596.7873 (2021), pp. 583–589.

[4] Hao Li et al. **"Visualizing the Loss Landscape of Neural Nets".** In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018.

[5] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning -*