Robert Tjarko Lange - Handwritten Notes 08/2019

MATH+/ BMS Summer School "Mathematics of Deep Learning"
2019 Schedule

Time	Monday 19.8.	Tuesday 20.8.	Wednesday 21.8.	Thursday 22.8.	Friday 23.8.
8.00	Registration				
8.45 - 9.00	Welcome				
9.00 - 10.30	Kuytniok 1	Fleuret 2	Vidal 3		L. Berlyand ~~Spokoiny~~ PDE ML
10.30 - 11.00	Coffee	Coffee	Coffee		Coffee
11.00 - 12.30	Fleuret 1	Vidal 2	Haber 2	basics of Neural ODE	TGIF Meeting
12.30 - 14.00	*Lunch*	*Lunch*	*Lunch*	*Lunch*	*Lunch*
14.00 - 15.30	Vidal 1	Haber 1	Fleuret 3	Haber 3	
15.30 - 16.00	Coffee	Coffee	Coffee	ODE/PDE ML Coffee	
16.00 - 17.30	Poster Session	Tensor Flow (Basics)	Tensor Flow (Basics)	Kutyniok 2 inverse Problems	

Time	Monday 26.8.	Tuesday 27.8.	Wednesday 28.8.	Thursday 29.8.	Friday 30.8.
9.00 - 10.30	Müller	Jenssen 2	Krause 1	Cohen 3	Leimkuhler 3
10.30 - 11.00	Coffee	Coffee	Coffee	Coffee	Coffee
11.00 - 12.30	Jenssen 1	Debate: New Horizons	Cohen 2	Schütte 2	Kutyniok 3
12.30 - 14.00	*Lunch*	*Lunch*	*Lunch*	*Lunch*	Closing
14.00 - 15.30	Leimkuhler 1	Schütte 1	Krause 2	Jenssen 3	
15.30 - 16.00	Coffee	Coffee	Coffee	Coffee	
16.00 - 17.30	Cohen 1	Practice Session	Leimkuhler 2	Practice Session	

to start on Thursday 22nd of August → before talk Rebrecht!

3rd Lecture Haber (UBC) – Using PDE to design stable architectures

- **DNN as ODEs**: $y_{j+1} = y_j + h K_j^{(2)} \sigma(K_j^{(1)} y_j + b_j)$ (h: stepsize) → timesteps ⇔ layers!

$$\dot{y} = K^{(2)}(t)\, \sigma(K^{(1)}(t)\, y + b(t))$$

→ Regularization & Early stopping ⇒ Badly-posed forward problem!

→ Given problem → Choose $K^{(1)}, K^{(2)}$ appropriately → Hamiltonian!

→ Optimize ⇔ Discretize ⇒ Which order?! DO vs. OD

↳ Classical: First discretize then optimize

↳ Why otherwise? Memory constraints!

- Multiple methods → Continuous process → More & more fine-grained! → better approx.
- Focus on DL coding frameworks → Parameters usually hidden!

↳ Does Conv2d have a bias? ; Are batch norm params trainable?

↳ Example: Runge Kutta ⇒ allows for larger step-size

↳ dt → stepsize! , # timesteps ⇔ "layers" → fixed number of evaluations

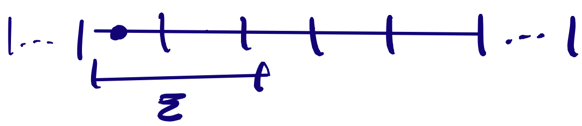
↳ Problem! No changes in dimensionality → Contrast 3rd example!

⇒ Understand how to switch dim / number of features in ODEs!

- Forget as much as possible! No clear best thing with practical convergence!
- DL like every inverse problem → look at intermediate steps
- Sanity check: Make sure that initial loss is close to random!
- Small batches initially → increase over time ⇒ Problem: Of parallelization of batch
- Monitor how gradient behaves: $g_t = \|\nabla\|$ → look at convergence!

- **CNNs as PDEs**: $K^{(1)}, K^{(2)}$: Conv kernels

→ Conv in 1D:



$K = (k_1, k_2, k_3)$

$$\begin{pmatrix} k_2 & k_3 & & 0 & & k_1 \\ 0 & k_1 & k_2 & k_3 & \ddots & 0 \\ k_3 & & 0 & & k_1 & k_2 \end{pmatrix}$$

→ degree of basis! ⇒ approx by deriv.

→ PDE → Space-time!

→ Parabolic PDEs : $y_t = -K(t)^T \sigma[K(t)y + b(t)]$

to Derived from: $y_{j+1} = y_j + h K_j^{(2)} \sigma(K_j^{(1)} y_j + b_j)$

with ansatz of $K_j^{(2)} = -(K_j^{(1)})^T$ ⇒ weight sharing!

to Non-linear heat equation relationship! ⇒ provides math. explanation

→ Weight decay ⇔ Tikhonov regularization

- Hyperbolic model/net: Leapfrog!

$$y_{tt} = -K(t)^T \sigma[K(t)y + b(t)]$$

$$y_{j+1} = 2y_j - y_{j-1} - h^2 K_j^T \sigma(K_j y + b_j)$$

→ Reversible: forward + backward in time!

→ less memory required for backprop ⇔ can reconstruct activations backwards from y_{j+1} and y_j → don't need to store all activations

- Explicit and semi-implicit methods! ⇒ Diffusion-Reaction process

- PDE design should be based on problem at hand!

- Batch size not related to (h) → controls what net can express!

→ ill-more problem vs. optimization!

- Neural ODE paper relationship ⇒ First opt. then discretize

to Assume that you can backward integrate ⇒ But can't!

to Fundamental problem in control theory!

2nd Lecture Kutyniok (TU Berlin) - DL meets Inverse Problems

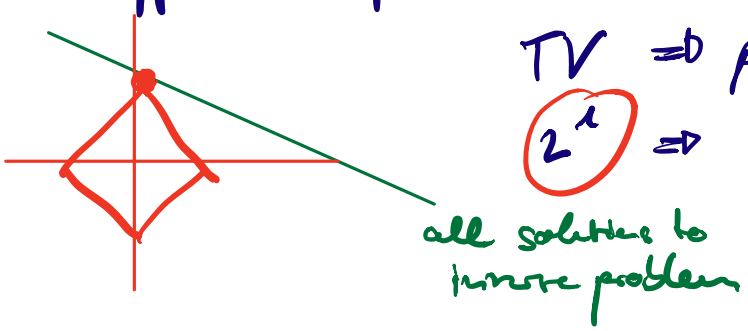
- Talk outline: Between model-based and data-driven approaches
- **Inverse problem** = Recovered original data from transformed version
  - → Examples: Fill in missing data, Feature extraction ⇒ Regularization
  - → Formulation:

$$K: X \to Y \text{ with } Kx = y \quad \forall x, y \in X, Y$$

  - → Hadamard conditions ⇒ Hadamard
    - ① Existence, ② Uniqueness: $x \in X$ is unique, ③ Stability (continuity!)
  - → Regularization to ensure uniqueness! → PRIOR!
- **Tikhonov Regularization**: to approx solution $x^\alpha \in X$, $\alpha > 0$

$$\min_x \mathcal{J}_\alpha(x) := \underbrace{\|Kx - y\|^2}_{\text{data fidelity term}} + \alpha \underbrace{\|x\|^2}_{\text{reg. term}}, \quad x \in X$$

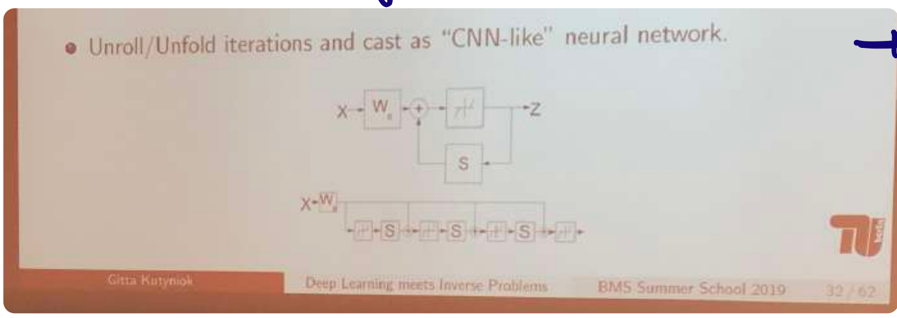
  - → Generalization ⇒ why needed? → $\alpha \cdot \mathcal{R}(x)$
    - to ensure continuous dependence on data
    - to Different examples: L^2 ⇒ smallest energy
      TV ⇒ piecewise constant features
      L^1 ⇒ Sparsity inducing



- Sparsity: Sparse solutions vs. compressible solutions
  - to make non-zero coefficients ⇒ exact
  - to rapid decay in size of coefficients ⇒ precise
- **Compressed sensing**: $y = Ax \to d \gg n$ ⇒ Sparse solution after trafo
  - to Compression + sensing in single step ⇒ solve via regularization!

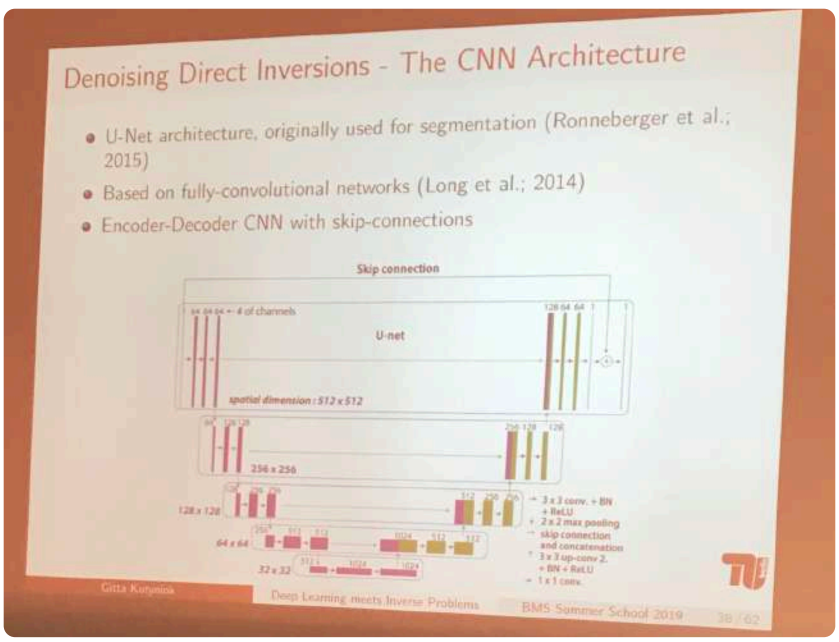
- Frame ⟺ generalization of orthonormal basis
  → $(\psi_\lambda)_\lambda$; $\exists\, 0 < A \le B < \infty$ with $A\|x\|^2 \le \sum_\lambda |\langle x, \psi_\lambda \rangle|^2 \le B\|x\|^2 \;\forall x$ (learn!)
  → Redundancy: Robust to loss of dimensions/basis! ⇒ Can be transmitted / used for noise reduction!
  → Find such a system ⇒ dictionary (representation syst.) learning algorithms, ↳ k-SVD!
  wavelets, shearlets

- ISTA algorithm: $x_{n+1} = S_{\tau/L}\left[x_n - 1/L \cdot W_e^T (W_e x_n - y)\right]$
  ↳ soft-threshold
  for solving $\operatorname{argmin}_x \|W_e x - y\|_2^2 + \tau \|x\|_1$



→ Formulate in network structure!
→ LeCun et al (2010)

- Denoising w. additive Gaussian Noise → Solved (Elad et al., '16)
- Denoising Direct Inversion (Ye et al; 2016): First reconstruct and then denoise reconstructed image
  transport priors (strong) here!
  ↳ Classical architecture: Unets

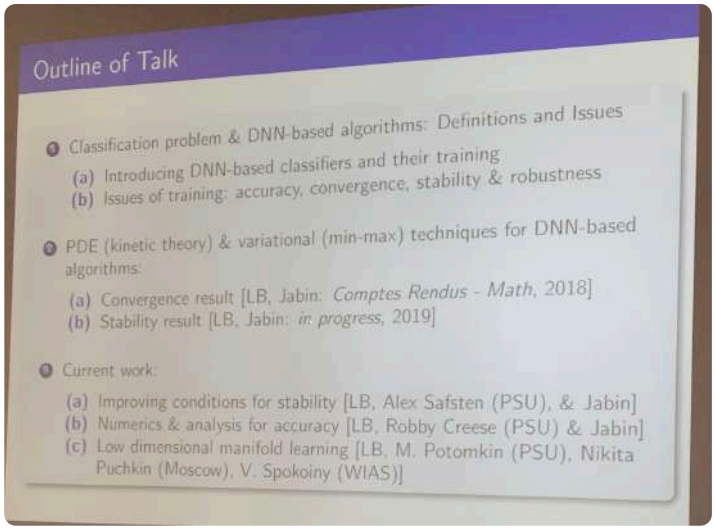


- Procedure to solve Tikhonov:
  Douglas-Rachford
  ↳ also related to denoising ⇒ soft-threshold

- Often times nowadays pre-trained denoising CNN is taken and plugged into denoising parts of application!

- Alternative approach via generative models ⇒ GANs // Applications to CT scans

Lecture – Leonid Berlyand (Penn State) – 'PDE techniques in DL'



- Stability → Discipline ⇒ Numerics, Computation from ANALYSIS PoV to disentangle contributions

$\phi : \mathbb{R}^n \to [0,1]^K$ ⇒ Non-exact $\phi(s,\alpha)$ to approximate Classifier ⇒ Opt. α ↓ proms!

- $\langle L(\alpha,s) \rangle_T$

- Rate of convergence : $\|\alpha^{n-1} - \alpha^n\| < f(n)$
- Stability ↔ dynamics of classification accuracy
- Universal approx. Th.: Single Hidden Layer NN w. inf. width can approx any smooth func. ⇒ Only EXISTENCE!

$$\bar{L}(\alpha) = -\sum_{SET} \nu(s) \log p_{class}(\alpha, s) = \sum_{SET} \nu(s) L(\alpha, s)$$

↓ true weight of class ⇒ $[0, 0, 1, \dots, 0]$

- Accuracy ≠ Loss! → Min-max problem
- PDE perspective via SGD
  → Stochasticity does not allow you to take continuous time limit
  → Requires kinetic PDE approach → entropy – entropy dissipation

$\frac{dE}{dt} = -D(t)$ → E : entropy → $E(t) := \bar{L}(\bar{\alpha}(t))$
→ D : dissipation → $D(t) := |\nabla_\alpha \bar{L}(\bar{\alpha})|^2$

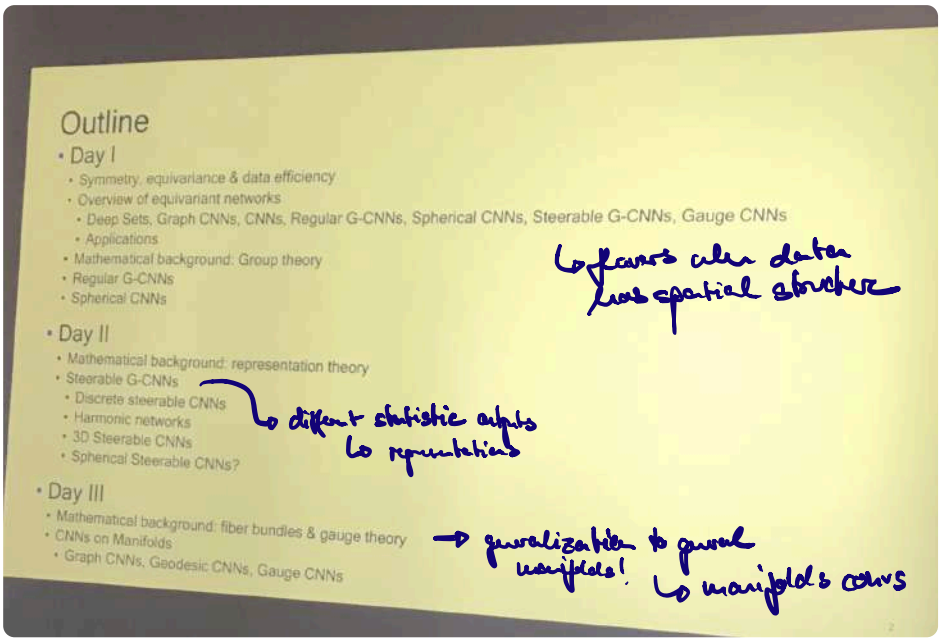
to Gronwall bound : $E(t) = \bar{L}(t) \le C/t \to 0$ as $t \to \infty$

→ Fokker–Planck Equation : Allows to formulate class. formula with. pdf

$$\text{func}(\alpha) = \int_\rho \text{fn}(\alpha)$$

1st Lecture – Taco Cohen – Equivariant Networks

- Prior knowledge of structure of symmetry of problem ⇒ data efficiency



↳ learns when data has spatial structure

↳ different statistic adapts to representations

→ generalizable to general manifolds!

↳ manifolds convs

- Symmetry = Trafo that does not change object

↳ ML: * Prob. distr. → Sampling in multi-modal distr.

* Label fcts. $X \mapsto Y$

* Parameter space → permutation of neurons and their weights

* causality = invariance to shifts in time.

↳ Symmetry as inductive bias → constraints define laws of physics

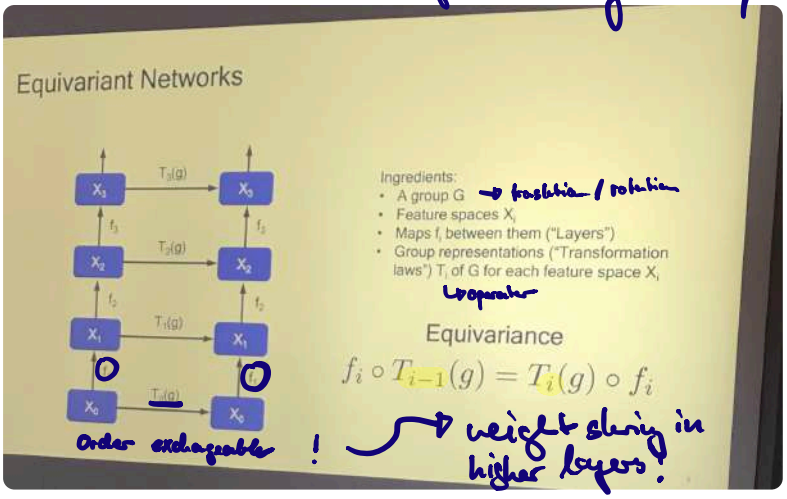
↳ Statistics: De Finetti ⇒ Exchangeability ⇔ Conditional independence

- Trafo Group : Set of trafo → identity, composed associativity, invertable

↳ Rotations (filters, feature maps), Euclidean motions, Projections, Diffeomorphisms (Continuous, discrete) (Translation, rotation)

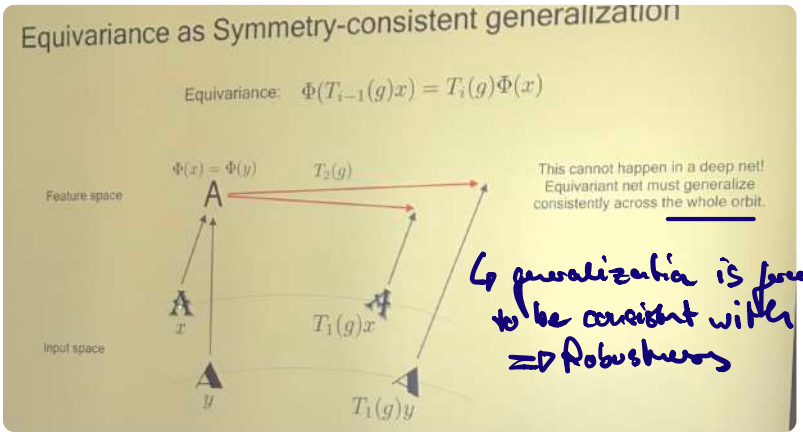
- Picasso Problem : Invariance not enough → Individual parts are not enough to tell you the identity of an object

→ CNN success: weight sharing + depth + Translation Equivariance



→ $f_2 \cdot f_1$ equiv. ⇒ $f_2 \circ f_1$ equiv.

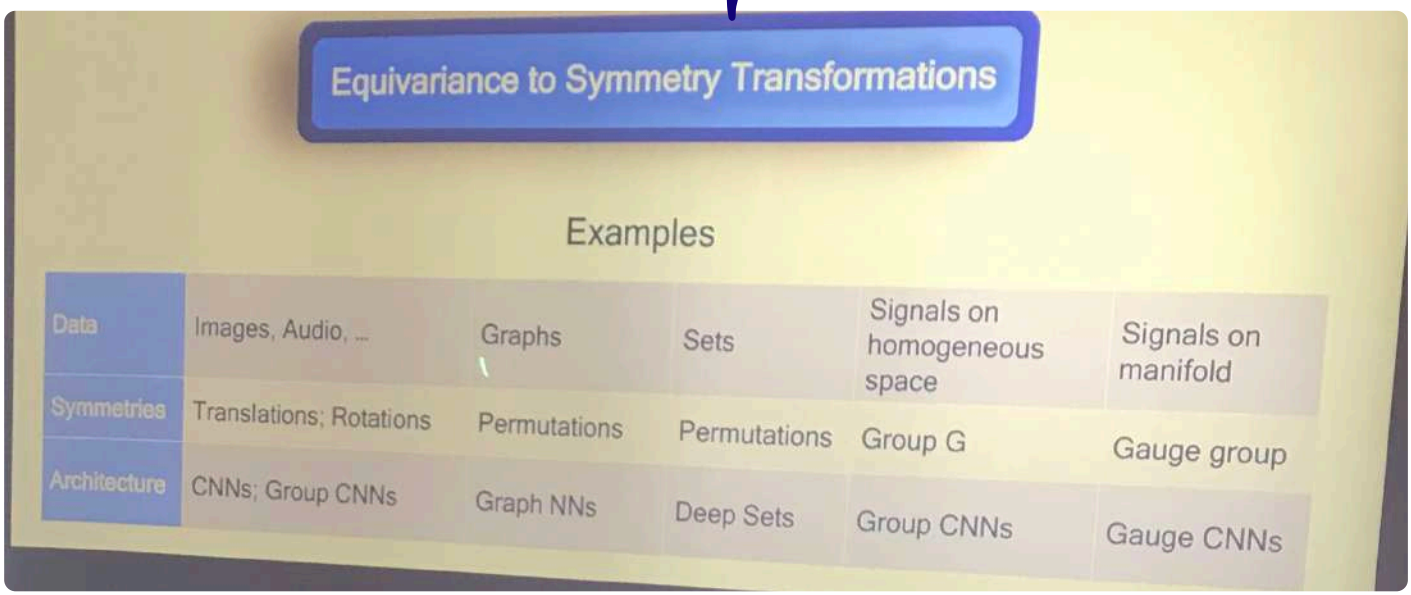
↳ To build an Equiv NetW: Make sure that all layers respect symmetries / are equivariant



G generalization is forced to be consistent with orbit!
⇒ Robustness

- Data Augmentation
  → does not guarantee equivariance!!
  → Combinatorial explosion in number of symmetries

- Residual connections ⇒ Equivariance also work as well!



- Deep Sets ⇒ Zaheer et al 17'
  → ordering does not matter ⇒ Permutation
  → Summation is trivially independent to ordering

APPLICATIONS

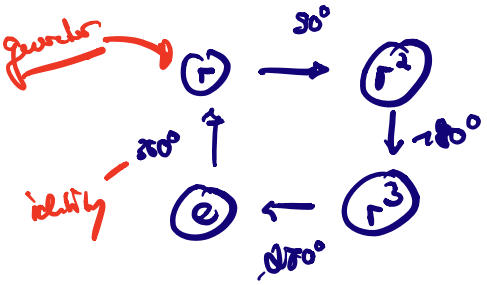
- Graph Convolution via Message Passing

- CNNs ⇒ Filters only work ↑ & ↔
  → Non-rotation equivariant
  → Group-Convolution ⇒ 4-90°-channels

- 3D G-CNNs ⇒ 3d Cubes → + lot more symmetries important!
- Scale Equivariance! ⇒ generally non-invertible → semi-group conv.

- CNNs on Manifolds
  - → Local nhoods via tangent space
  - → ambiguity in choice of filters ⇒ ~~gauge~~ symmetry → Mark Pool

- Group Theory Basics
  - → Cayley Diagrams ⇒ Group representation as a graph



* For each element g in G add node
* For each node g and generator h: add a edge!

  - → Cosets: For $g \in G, h \in H$: $g \sim g' \Leftrightarrow \exists h \in H : g' = gh$
    - ↳ reflexive, symmetric & transitive
    - ↳ Equivalence classes = Cosets ⇒ Visualize in Cayley! ↳ Quotient
  - → Group Action: Map $(g, x) \mapsto gx$
  - → Homogenous space ≅ Quotient space w/o privileged origin/cosets

CNN	G	H	G/H
Conventional CNN	Z^2 (planar translation)	{e} (trivial)	Z^2 (2d pixel grid)
P4-CNN	P4 (discrete roto-translation)	C4 (4-fold rotation)	Z^2 (2d pixel grid)
SE(2)-CNN	SE(2) (continuous roto-translation)	SO(2) (continuous rotation)	R^2 (continuous plane)
Spherical CNN	SO(3) (continuous 3D rotation)	SO(2) (continuous 2D rotation)	S^2 (sphere)
DNA CNN	$Z \times Z_2$ (translation + reverse-complement)	Z_2 (reverse-complement)	Z (1d grid)

- Regular G-CNN
  - → Group Conv: Output feature map at $g \in G$: Transform filter by g and compute inner product with input feature map
  - → feature map: $f: X \to \mathbb{R}^c$ ⇒ think of as filter/channels stacked for one particular location!
  - → Filter bank / Kernel: $\psi: X \to \mathbb{R}^{k \times c}$ ⇒ Support small: 3x3 e.g.

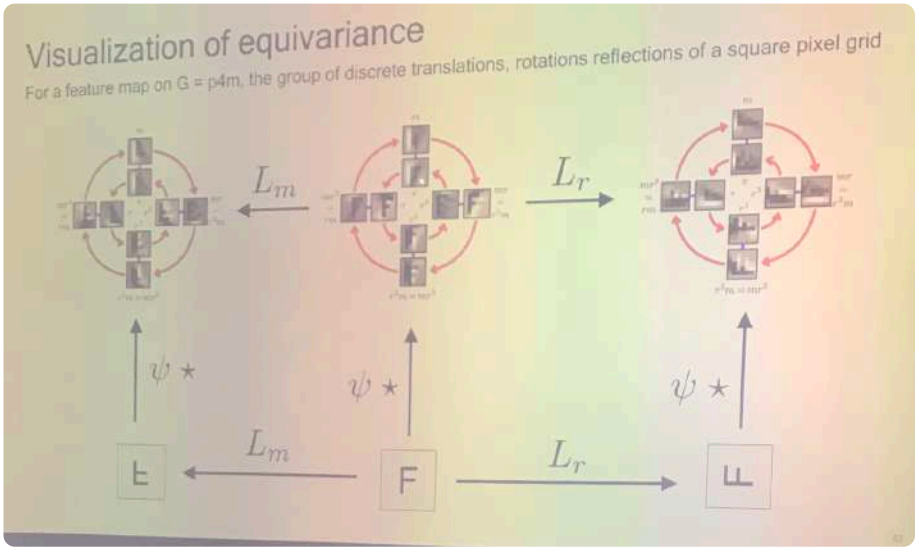
→ Transf. of feature maps & filters : Group actions on feature maps & filters

$$[L_g f](x) = f(g^{-1}x)$$

→ Group Convolution: $[\psi * f](g) = \langle L_g \psi, f \rangle$

($[\psi * f](g)$: feat. on group; $L_g \psi$: transf. filter; f: input feat.)

$$= \sum_{x \in X} \sum_{c=1}^{C} \psi_c(g^{-1}x) f_c(x)$$

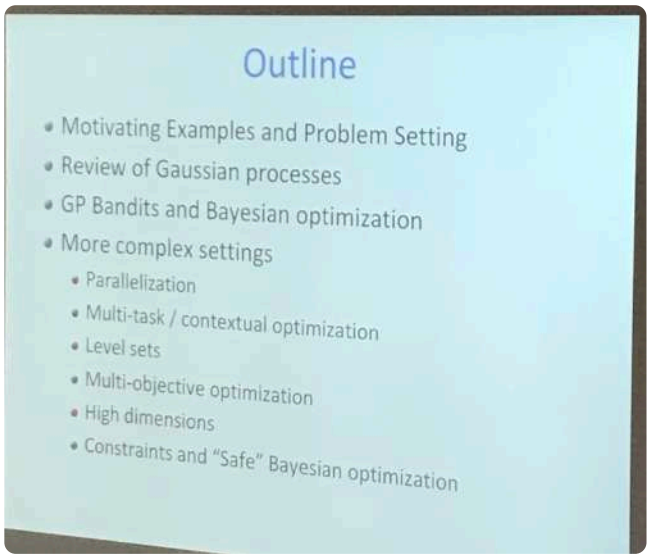


- Implementation
  - → Filter bank just bigger!
  - to index for rotation
  - to filter Transf. Group!
  - 2×2 Filter ⇒ Rotate!

- Non-linearities $[C_\nu f](g) = [\nu \circ f](g) = \nu(f(g))$
- Properties / Obstructions → loss of equivariance!
- Trainability → Gradient can propagate through all rotated filter banks!

1st lecture - Andreas Krause - Bayesian Optimization

- Protein Fitness Example → Collaboration w. chemistry nobel prize winner Arnold
  - ↳ SEQUENTIAL EXPERIMENTATION! W. NOISY FEEDBACK
  - ↳ E-E Dilemma → NEED TO USE REGULARITY



- k-arm stochastic bandits
  - → allocation of T tokens
  - → payout arm $i \Rightarrow$ mean f_i
  - ↳ ∞-arm generalization!
    - $\Rightarrow$ Can't try all options once
    - $\Rightarrow$ Need to generalize/predict
- structured bandits: Linear, Lipschitz, combinatorial, etc.
- Original BO paper → Kuščus 75'
  - ↳ less theoretical background compared to bandits even though problem is very similar!

Cumulative Regret: $R_T = \sum_{t=1}^{T} (\max_x f(x) - f(x_t))$ ← instantaneous

↳ sublinear if $R_T/T \to 0$

Simple Regret: $S_T = \min_{t \in \{1,\dots,T\}} (\max_x f(x) - f(x_t))$

↳ Min vs. Avg.: $S_T \leq R_T/T$

→ Q: Does the max always have to be known?

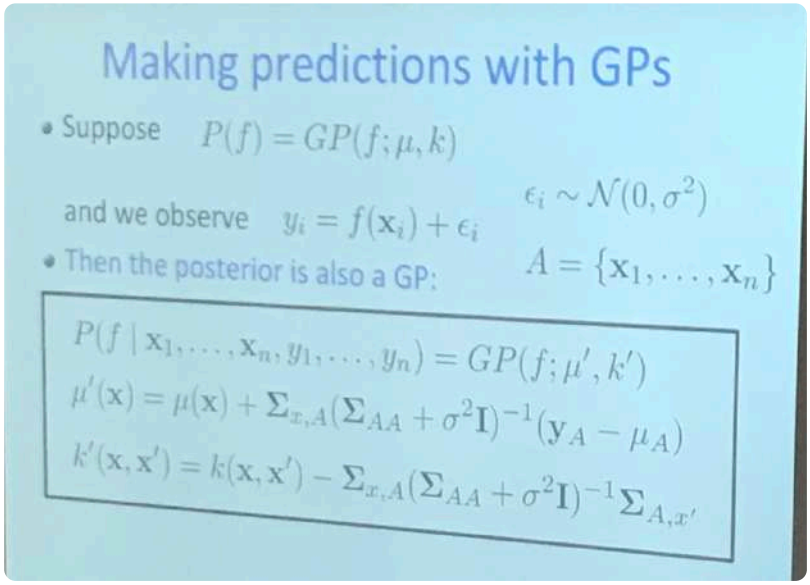
- Gaussian Process ⇒ ∞-dim multiv. Gaussian ⇒ over fct. space
  ↳ finite marginals are multiv. Gaussians

$\mu : X \to \mathbb{R}$, $K : X \times X \to \mathbb{R}$ ⇒ PARAMETRIZ.

↳ $N(\mu_*, \Sigma_{**})$ ⇒ K: kernel, μ: mean fct.

↳ cov./kernel characterizes smoothness assumptions!]

- positive definiteness requirement of kernels — CORRELATIONS
- $K(x, x') = \exp\left(-\frac{(x-x')^2}{l^2}\right)$ — squared exp. → light scale ⇒ wigglines!
- Linear kernel: $K(x, x') = x^T x'$ ⇒ Bayesian Linear Regression
  ↳ $\Phi(x)^T \Phi(x')$ → learn!



- Kernel Characterization
  - Prior knowledge
  - Max ML. ⇒ Empirical Bayes
    ① Integrate out hyperp.
    - Online adaptation!
    ② Form of Bayesian inference ⇒ have to use non-conjugate prior!

↳ Bandits ⇒ Surrogate model per fct. of → How can we construct the fct./GP when we don't know what the max. is?!

- Uncertainty quantifiable?!
  → Entropy: $H(x) = E[-\log p(x)]$ ⇒ Max info gain/entropy reduction

↳ $F(S) = H(f) - H(f \mid y_S) = \frac{1}{2} \log |I + \sigma^{-2} \Sigma_{SS}|$

entropy before eval of y_S — entropy afterward of y_S — of selected point!

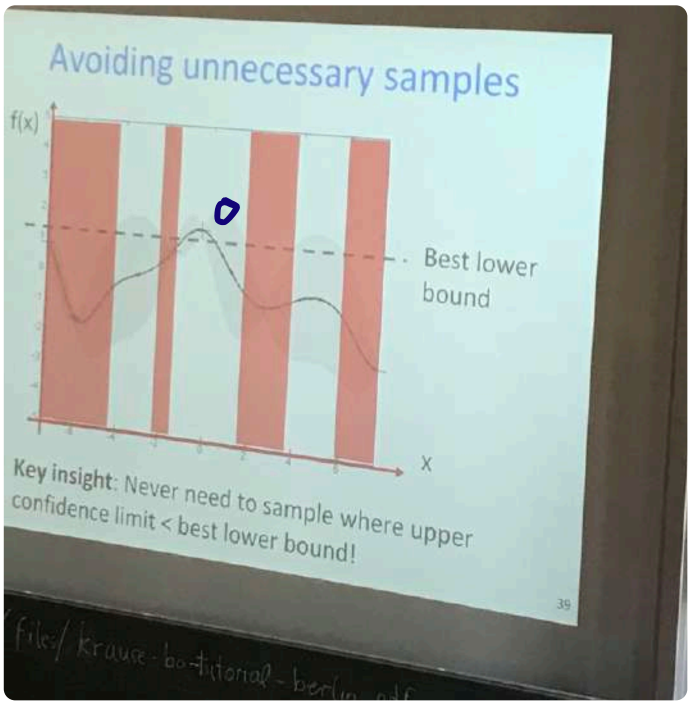
↳ NP hard to optimize

→ Greedy: $F(S)$ is monotone submodular $\Rightarrow S = \max_S \Sigma_{SS}$

(lets recap)

$\forall x \in D, \forall A \subseteq B \subseteq D: F(A \cup \{x\}) - F(A) \geq F(B \cup \{x\}) - F(B)$

↳ Only explore → learn about function but waste a lot of samples in part of space where function is exp. to be small → OPTIMIZATION → E vs. E



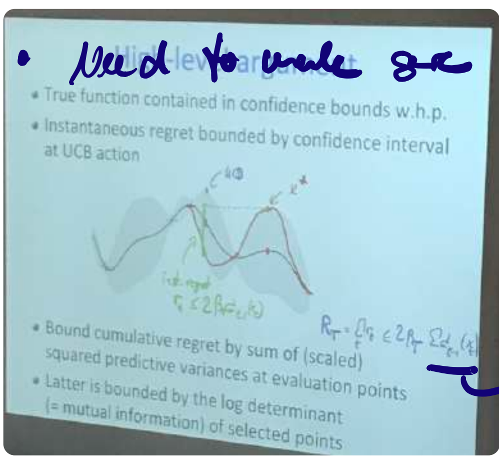
- UCB: Take best mean value at the current state μ^{max}

↳ $x_t = \arg\max_{x \in D} \mu_{t-1}(x) + \beta_t \sigma_{t-1}(x)$

increase over time — time depending since variance decreases over time

↳ Max vs. Min! ⇒ Here objective is to max function!

↳ Can derive convergence!



- Need to make sure that true function lies within confidence bounds!

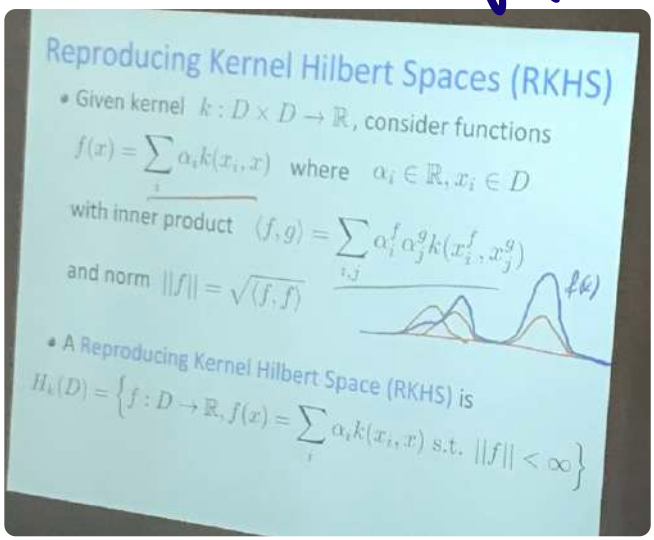
variance contracts over time!

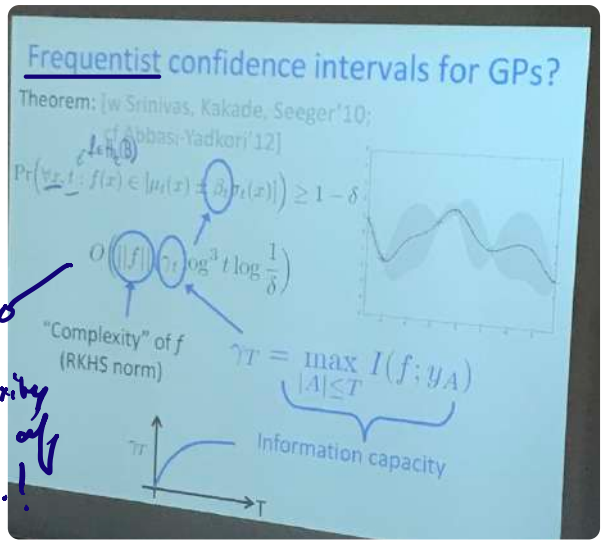
→ γ_T ⇒ estimate via simulation

where $\gamma_T = \max F_T$

↳ bound on avg. regret $O(\gamma_T / T)$

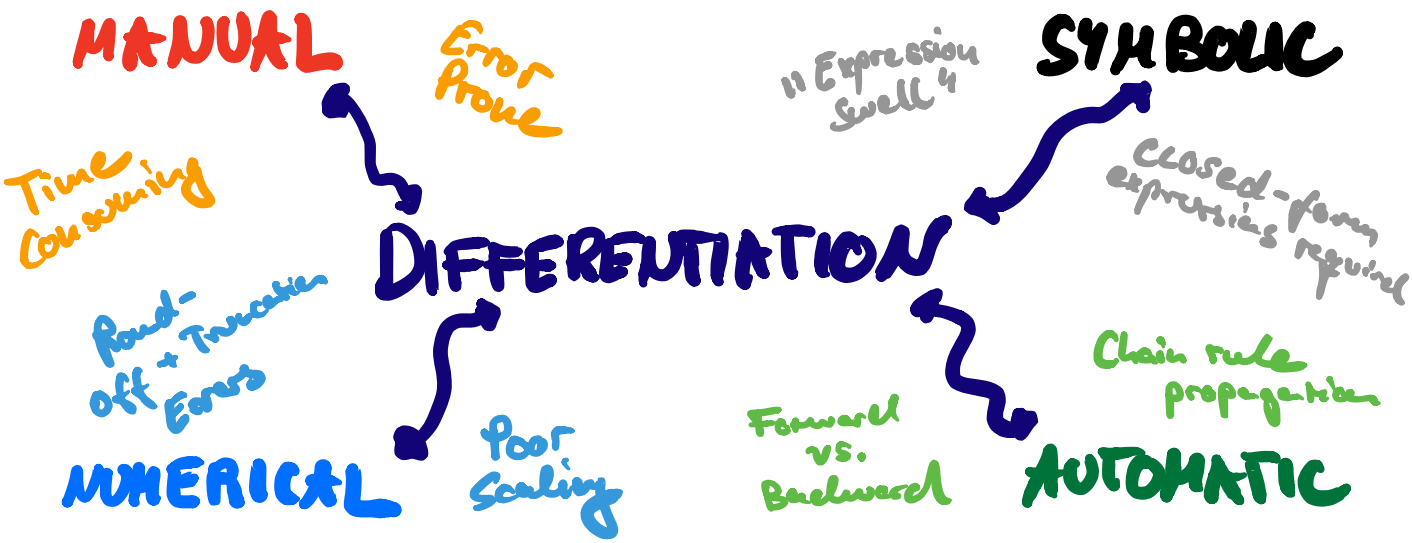
- Need 2 crucial things: [1] Fine prediction with bands [2] max $f(x)$ bound
- Robustness: Family of functions → Want $f(x) \in [\mu_0(x) \pm \beta_T \sigma_0(x)]$





let β_T grow w. complexity of family of fcts.!

→ Generally speaking: Non-convex opt. of acq. fct.!
to Lipschitz-optimization to Gradient ascent!



2nd Lecture - Andreas Krause - Extensions

Andreas Krause - BO/GP Bandits - Lecture 2

- Parallelization = Delayed feedback → Batch Selection
  → internal noise → also value in doing same exp. multiple times

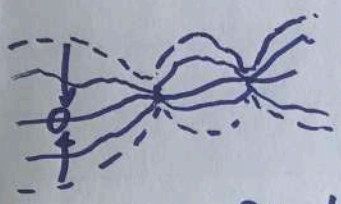
$$x_t = \arg\max_{x \in D} \left[\underbrace{\mu_{t-1}(x)}_{\text{fixed}} + \beta_t^{(\text{mid})} \underbrace{\sigma_{t-1}(x)}_{\text{"hallucinate"}} \right]$$

→ anticipate variance reduction!

↳ idea: inflate confidence bands such that no matter what happens during experiments → they are still calibrated!

↳ Then hallucinate to variance reduction
at 1st point in batch, afterwards
select 2nd point, hallucinate var. red.,
...

↳ Google Vizier AutoML



→ Others: Schonlau 97! : Multi-Point EI
Azimi et al '10: Simulation Matching

- Multi-Task / Transfer BO → Contextual GP bandits

→ context var. $z_t \in \mathcal{Z}$ → $y_t = f(x_t, z_t) + \epsilon_t$

↳ $r_t = \sup_x f(x, z_t) - f(x_t, z_t)$

↳ Idea: map contexts to actions → Between bandits and RL!
→ not model transitions / sequence of contexts → arbitrary

$$x_t = \arg\max_{x \in D} \mu_{t-1}(x, z_t) + \beta_t \sigma_{t-1}(x, z_t)$$

↳ guarantees along context-action pairs!

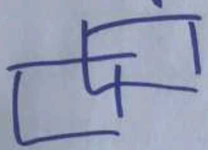
$$k((x,z),(x',z')) = k(x,x') \cdot k(z,z')$$

→ or additive

→ Sublinear contextual regret translates!

other bounds?

- Multiple Objectives → Pareto Optimality Considerations
  - → Predict how pareto function looks like from few evals
  - ↳ GPs for different objectives! → Different GPs!
  - ↳ Allows to rule out several evaluation points!
  - → allow for study in pareto classification



→ sample from regions most informative!

PARETO!

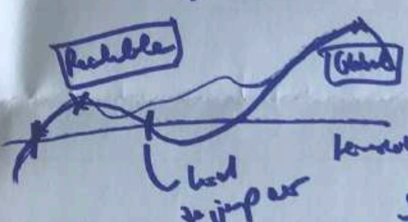
- Discrete Vars.
  - Graph kernels
  - Learn representations

- Safe Exploration + Constraints ⇒ DBS / Stimulation
  - → Sim2Real

$$\max_{\theta_t} f(\theta_t) \quad s.t. \; g(\theta_t) \geq 0$$

↳ safety threshold

→ Max. Unknown fct. under unknown constraints
↳ Need regularity ass.!



threshold

→ Focus exploration where lower bound > thresh.

→ Starting point: Max. Acq. Fct. over certified safe domain!

SafeUCB ↳ risk: getting stuck in local opt.!

→ Need to provide incentive to explore boundary
↳ Pick actions that max g → increase range of actions

→ Need to transform time / physical costs into bits!

$$\hat{f}(x) = f(x) + \boxed{\sigma(x)}$$

bias induced by sim and not real!

↳ Not like contextual bandits since we are always (regardless of context) interested in max f and not $\hat{f}$.

→ Heteroscedasticity. Time / History / Action dependent noise: closer to RL problem! Informativeness vs. Uncertainty
↳ closer to RL problem!
Nikolov, Kirschner, Berkenkamp ICLR 2019

- High-dims ⇒ Many vars → structure / graphics → subspace selection → Mutny et al 19! ICML