TOWARDS ROBUST INTERPRETABILITY WITH SELF-EXPLAINING NEURAL NETWORKS

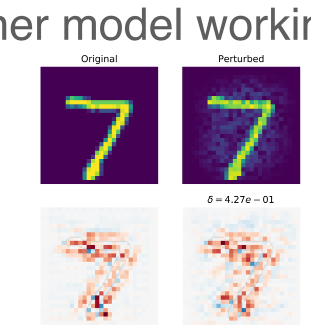
David Alvarez-Melis & Tommi S. Jaakkola | **MIT CSAIL**

Summary

- What makes linear models "interpretable"? Can we preserve it while increasing the complexity of the models?
- We identify basic desiderata for interpretability —explicitness, faithfulness and stability—and **enforce them during training**
- Leads to a class of rich complex models that produce **robust explanations** as intrinsic part of their operation

Motivation

- High modeling capacity often necessary for performance
- Recent work focused on producing **a-posteriori** explanations
- Explains locally w/ limited access to inner model workings:
  - gradients/error-back-propagation
  - black-box queries
- Challenges:
  - definition of locality
  - computational cost
  - explanations aren't robust (small Δ in input $\Rightarrow$ large Δ in expl)



- A-posteriori explanations are sometimes the only option (e.g. for already-trained models)
- Otherwise, can we make our models explain their predictions as **intrinsic** part of their operation?

Self-Explaining Neural Network

Def. $f(x) = g\big(\theta_1(x)h_1(x), \dots, \theta_k(x)h_k(x)\big)$ is a **self-explaining** model if:

1. g is monotone and completely additive
2. g is increasing on each $z_i := \theta_i(x)h_i(x)$
3. $\{h_i(x)\}_{i=1}^k$ is an interpretable representation of x
4. k is 'small'
5. θ Is locally-Lipschitz with respect to h

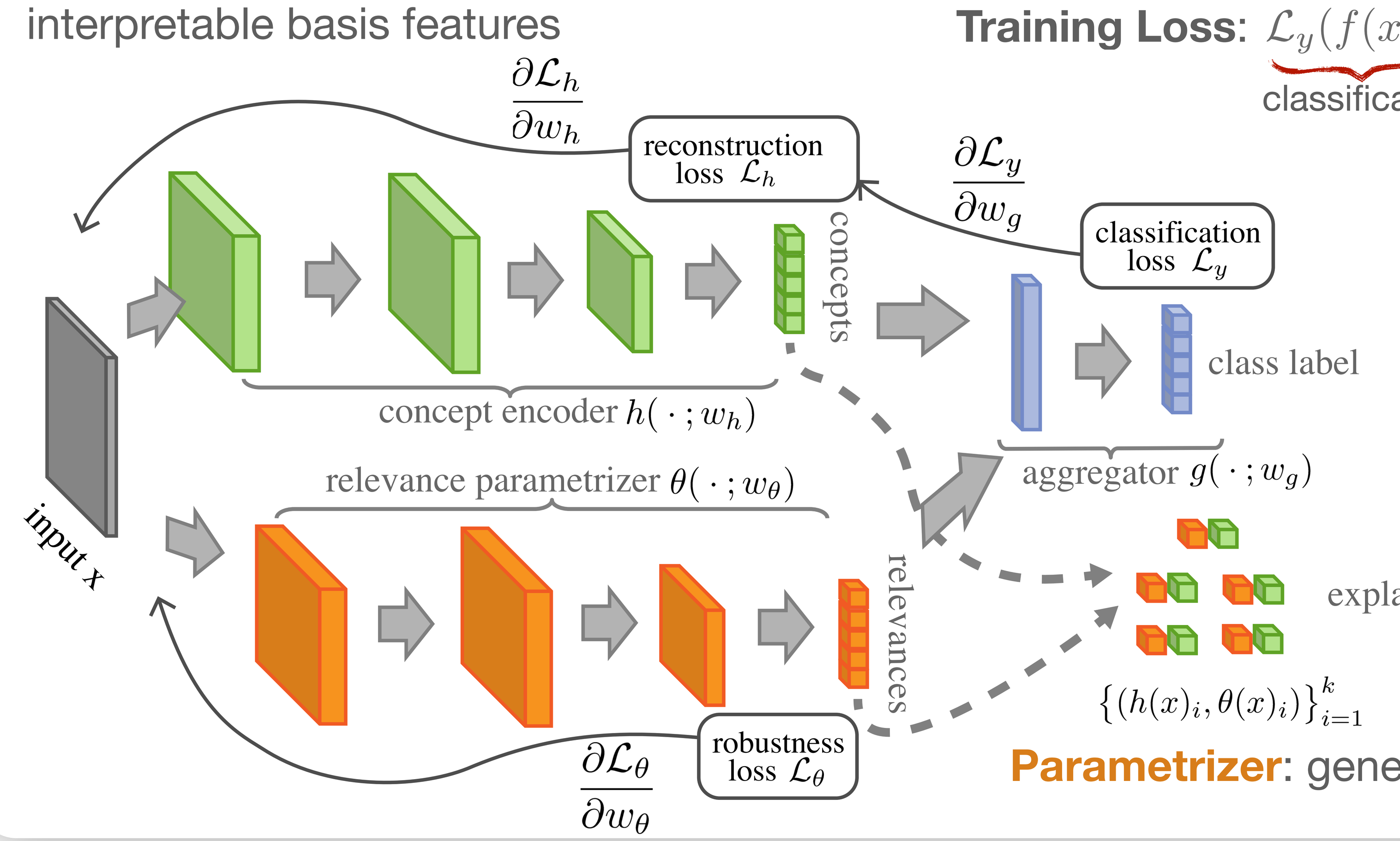
- e.g. sum, affine functions with positive coefficients
- application-dependent
- view f as function of $\xi := h(x)$. Want $\theta_i(x)$ to behave as (constant) coeffs of f w.r.t ξ , i.e. $\theta(x) \approx \nabla_\xi f$ use $\nabla_x f = \nabla_\xi f \cdot J_x^h$ to impose proxy condition:

$$\mathcal{L}_\theta(f(x)) := \|\nabla_x f(x) - \theta(x)^\top J_x^h(x)\| \approx 0$$

ensures f not only **looks** like a linear model but actually (locally) **behaves** like one!!!

Concept encoder: transforms input into interpretable basis features

Training Loss: $\mathcal{L}_y(f(x), y) + \lambda\mathcal{L}_\theta(f) + \xi\mathcal{L}_h(x, \hat{x})$ — classification, robustness, concept reconstruction



Aggregator: combines relevance scores and concepts to produce prediction

Parametrizer: generates concept relevance scores

Learning Interpretable Basis Concepts

- Explanation based on raw inputs suitable in low-dimension
- For high-dim inputs, raw features are not ideal for explanation
  - often lead to noisy explanations, sensitive to artifacts
  - hard to analyze coherently
  - lack of robustness is amplified
- Instead, operate on higher level features ("**concepts**"):
  - e.g. textures and shapes instead of raw pixels
- Ideally, concepts informed by in-domain expert knowledge
- If not available, concepts can be learnt with rest of the model
- Desiderata for concepts $h(x)$:

Desiderata	Proposed Approach
1. Fidelity: preserve relevant info	→ autoencoder loss
2. Diversity: few non-overlapping concepts	→ enforce sparsity
3. Grounding: be human-understandable	→ show prototypes

Interpretability Desiderata

(i) Explicitness/Intelligibility: Are the explanations immediate and understandable?

(ii) Faithfulness: Are relevance scores indicative of "true" relevance?

(iii) Stability: How consistent are the explanations for similar/neighboring inputs?

From Interpretable to Complex

- Starting point: **linear model** $f(x) = \theta^\top x = \sum_{i=1}^n \theta_i x_i + \theta_0$
- Interpretable because:
  1. inputs x_i **grounded** on meaningful observations
  2. θ_i have clear interpretation: $\pm$**contribution** of x_i to $f(x)$
  3. additive aggregation of $\theta_i x_i$ does't conflate feature-wise interpretation of impact

Step 1: Generalized coefficients. $f(x) = \boxed{\theta(x)}^\top x$

- Let coefficients depend on the input:
- Choose $\theta(\cdot)$ from a complex class (e.g. neural net)

Step 2: Beyond raw features. $f(x) = \theta(x)^\top \boxed{h(x)}$

- linear model explanation is only in terms of raw inputs
- allow more general features - ***interpretable basis concepts***

Step 3: Further generalization. $f(x) = \boxed{g}\big(\theta(x)_1 h(x)_1, \dots, \theta(x)_n h(x)_n\big)$

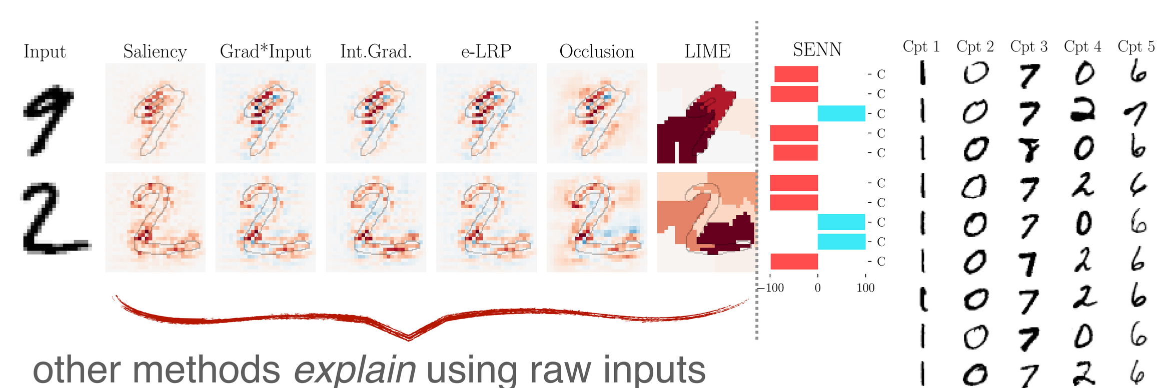
- Aggregation function more general than sum

Model is now nearly as powerful as any neural network but not really more interpretable (so far).

Need to **regularize** model to preserve the interpretability properties of the original linear model!!!

Experiments

Explicitness/Intelligibility

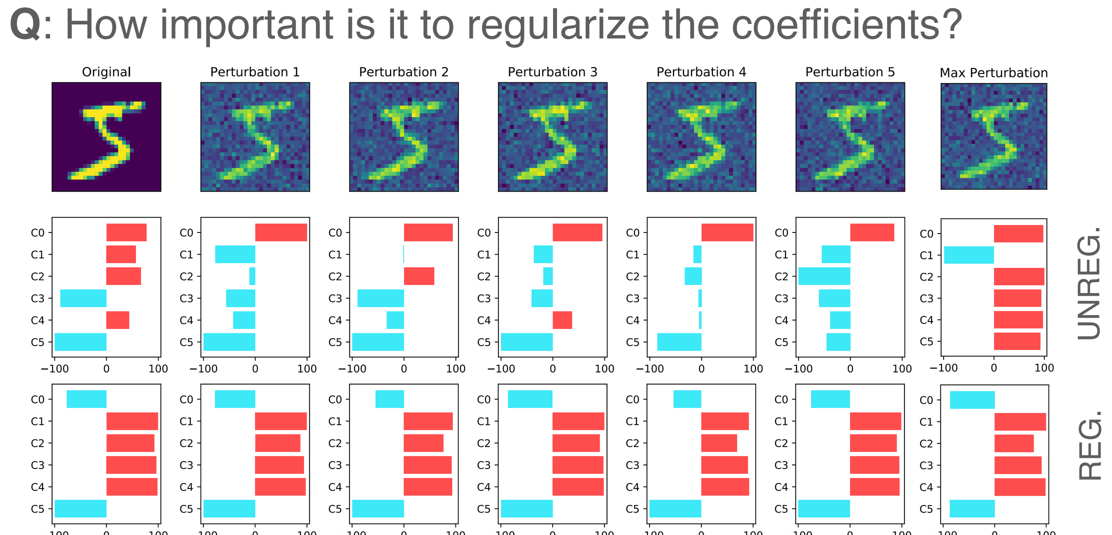


other methods *explain* using raw inputs

SENN *explains* using concepts

Ablation Results

Q: How important is it to regularize the coefficients?



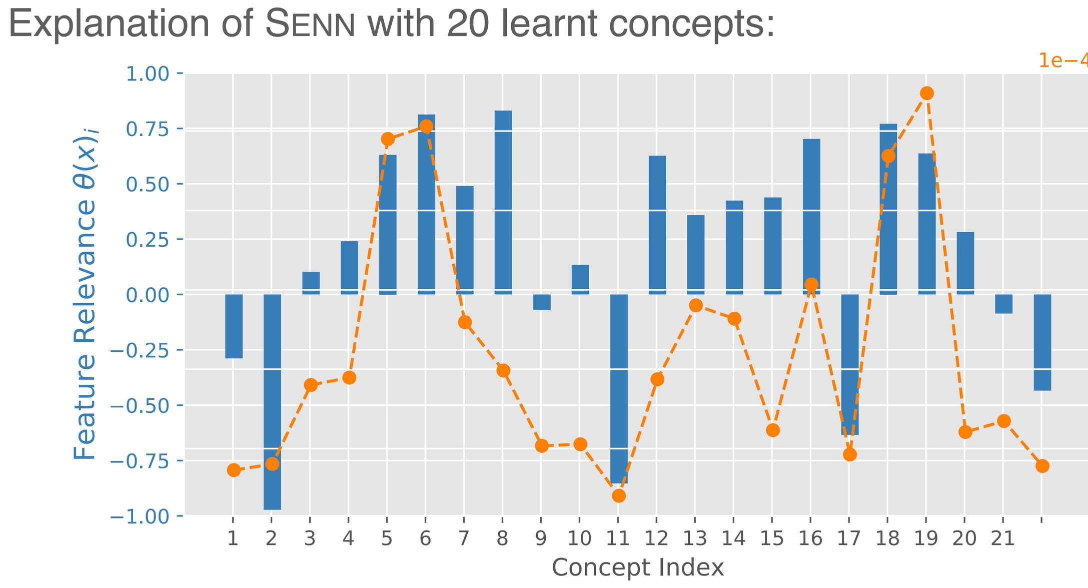
A: **Very important!**

Faithfulness

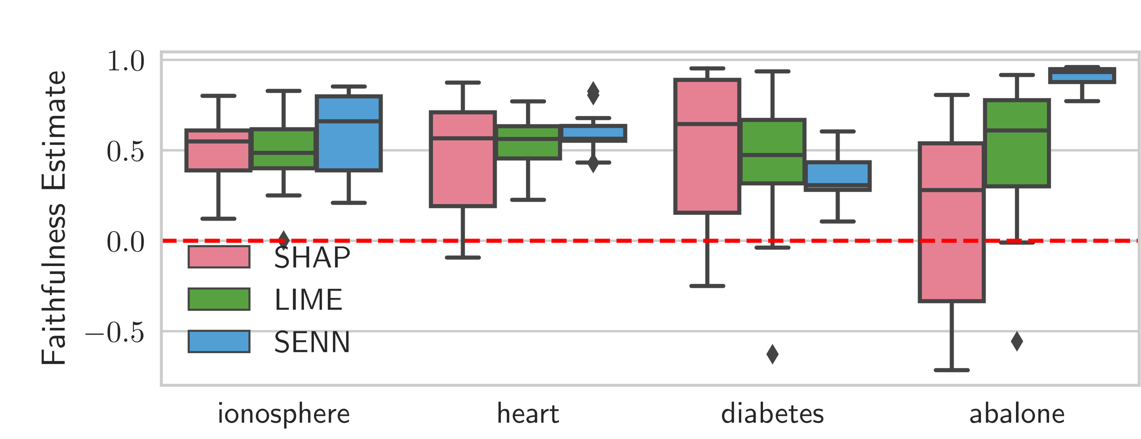
compare θ_i vs change in prediction from removing x_i :

$$\text{faithfulness}(\theta_i) = \text{corr}\big(\theta_i, f(x_1, \dots, x_n) - f(x_1, \dots, \not{x_i}, \dots, x_n)\big)$$

Explanation of SENN with 20 learnt concepts:



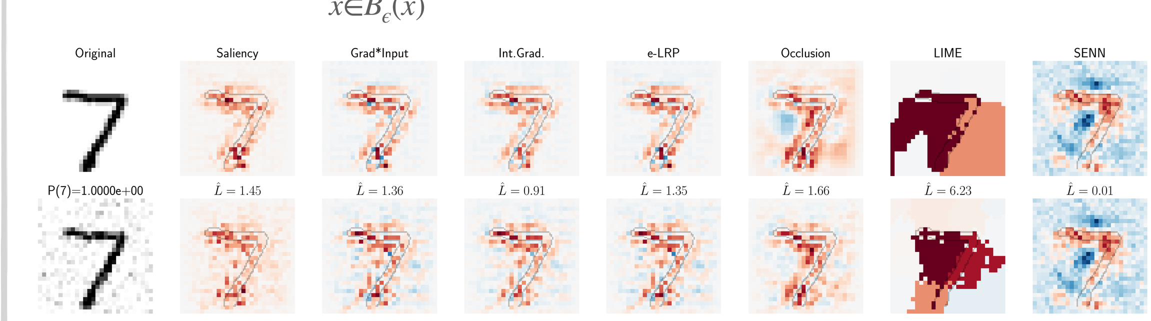
Results aggregated over full dataset:



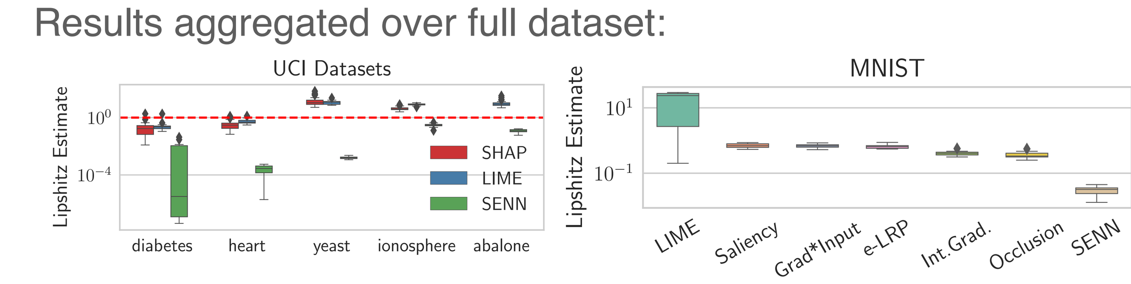
Stability

relative change in explanation vs explanation units:

$$\hat{L}(x) = \arg\max_{\hat{x} \in B_\epsilon(x)} \|f_{expl}(x) - f_{expl}(\hat{x})\|_2 / \|h(\hat{x}) - h(x)\|$$



Results aggregated over full dataset:



Effect of regularization on SENN's stability: