Deep Learning for Case-Based Reasoning through Prototypes

Oscar Li , Hao Liu , Chaofan Chen , Cynthia Rudin

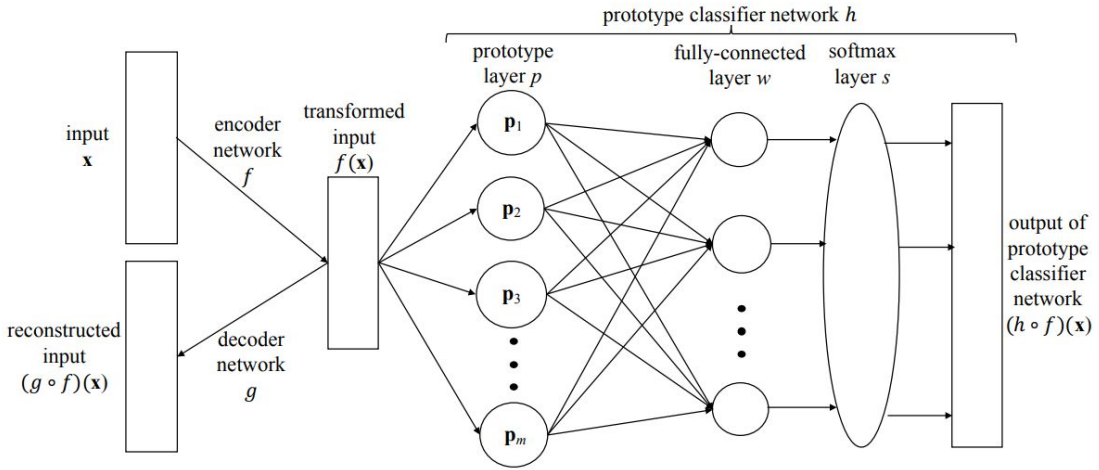
(Reimplementation)

Main idea

Add special prototype layer to learn prototypes.

Use prototypes to explain class prediction for test examples.

Prototypes similarity



Desired prototypes are:

$$R(g \circ f, D) = \frac{1}{n} \sum_{i=1}^{n} \|(g \circ f)(\mathbf{x}_i) - \mathbf{x}_i\|_2^2.$$

$$R_1(\mathbf{p}_1, ..., \mathbf{p}_m, D) = \frac{1}{m} \sum_{j=1}^{m} \min_{i \in [1,n]} \|\mathbf{p}_j - f(\mathbf{x}_i)\|_2^2,$$

$$R_2(\mathbf{p}_1, ..., \mathbf{p}_m, D) = \frac{1}{n} \sum_{i=1}^{n} \min_{j \in [1,m]} \|f(\mathbf{x}_i) - \mathbf{p}_j\|_2^2.$$

- well reconstructed (minimizing autoencoder error)
- as close as possible to at least one of the training examples in the latent space
- representative for all training examples (i.e. every encoded training example is as close as possible to one of the prototype vectors)

*We also use cross-entropy to penalize misclassification

Datasets

MNIST

FASHION-MNIST

CARS

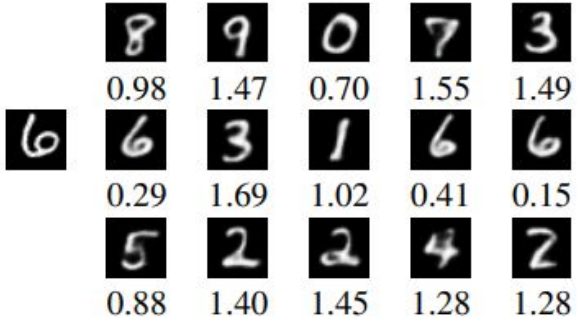
ROCK, PAPER, SCISSORS

AUTHORS

REIMPLEMENTATION

Learned prototypes

Distances between prototypes and test example

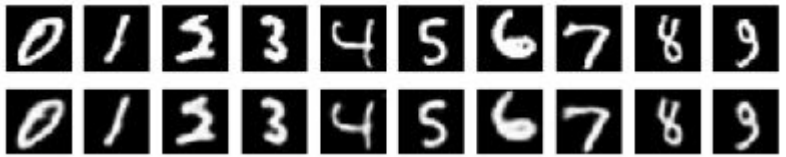


Accuracy scores

AUTHORS

	interpretable	without prototype l.	standard CNN
test set accuracy	99.22%	99.23%	99.24%

Autoencoder samples



REIMPLEMENTATION

	interpretable	without prototype l.	standard CNN
test set accuracy	99.12%	99.37%	99.09%

Learned prototypes

S S P P R

Autoencoder results

Explainable vs standard classification

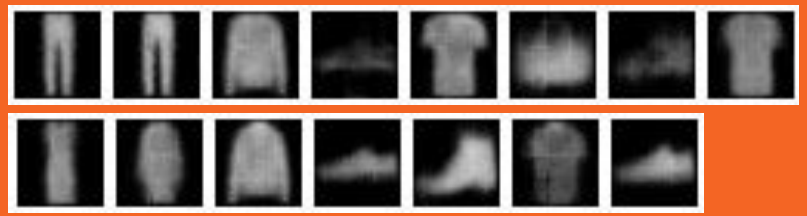
Explainable accuracy: **79.57%**

Standard accuracy: **78.76%**

AUTHORS

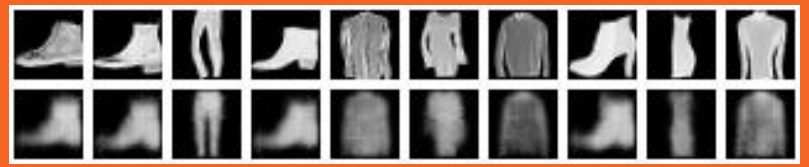
REIMPLEMENTATION

Learned prototypes



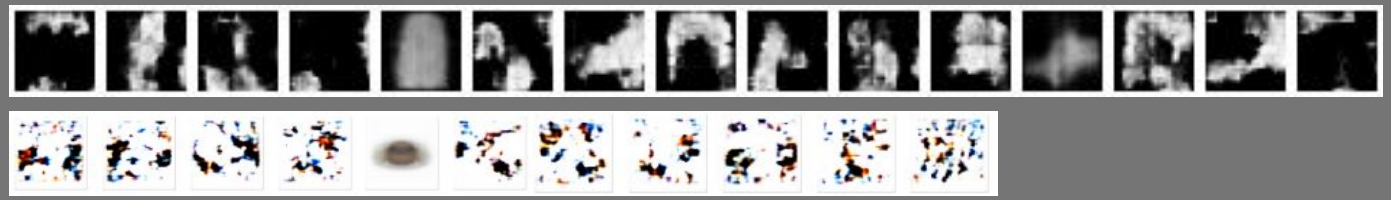
Accuracy & Autoencoder samples

	test set accuracy
authors	89.95%
reimplementation	90.23%

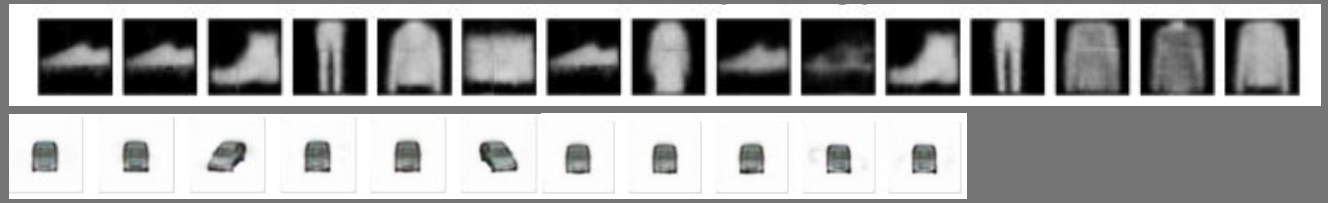


Ablation study

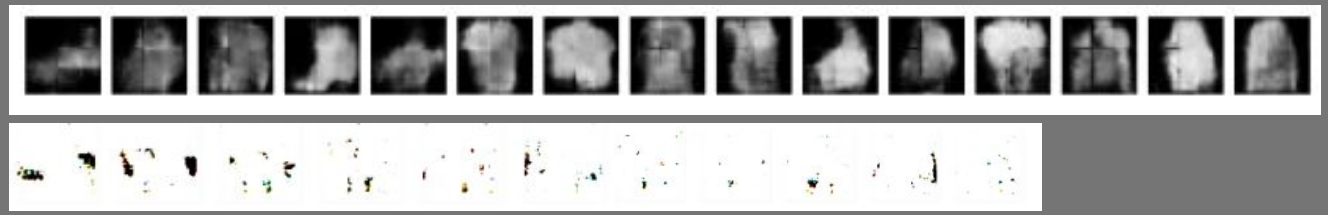
- without R1



- without R2



- without R1 and R2



Any questions?

Thanks for your attention!