FSPool: Learning Set Representations with Featurewise Sort Pooling

Yan Zhang, Jonathon Hare, Adam Prügel-Bennett

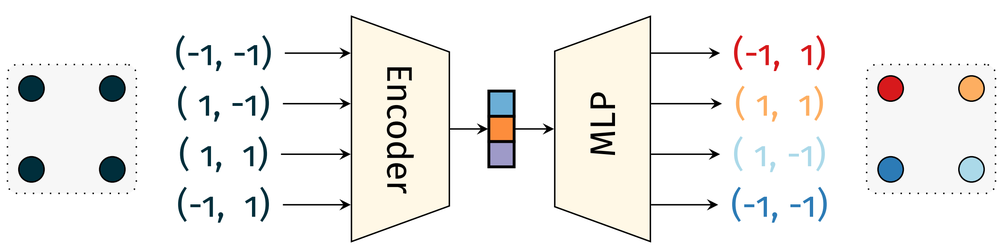
UNIVERSITY OF Southampton

Code available at
https://github.com/Cyanogenoid/fspool

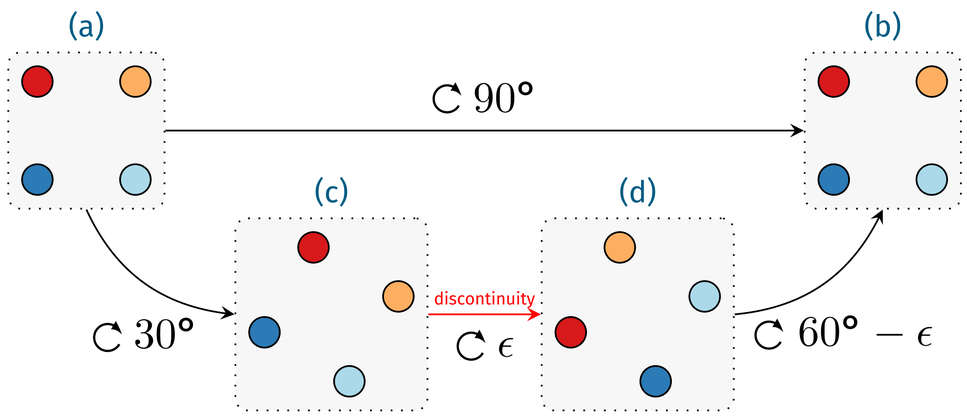
Predicting sets with MLPs or RNNs results in **discontinuities** due to the **responsibility problem**.

To pool a set into a vector, **sort** each feature **independently** to learn about their distribution.

- Sets are **unordered**, but MLP and RNN outputs are **ordered**.
  > **Discontinuities** from *responsibility problem*.
- In a normal set auto-encoder, each output is **responsible** for a set element.



- What happens when we auto-encode a rotating square?



The responsibility problem

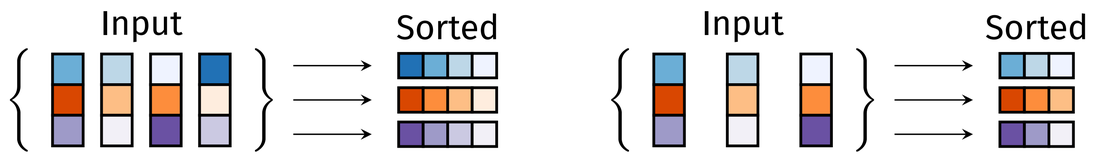
- (a) and (b) are the same set.
  > (a) and (b) encode to the same vector.
  > (a) and (b) have the same MLP output.
- (a) is turned into (b) by rotating 90°.
  > Rotation starts and ends with the same set.
  > MLP outputs can't just follow the 90° rotation!
  > There must be a discontinuity between (c) and (d)! All the outputs have to jump 90° anti-clockwise.

- *Smooth change of set requires discontinuous change of MLP outputs.*
- Neural networks don't like learning discontinuities.
- Present in every dataset where two points can be smoothly exchanged like this.

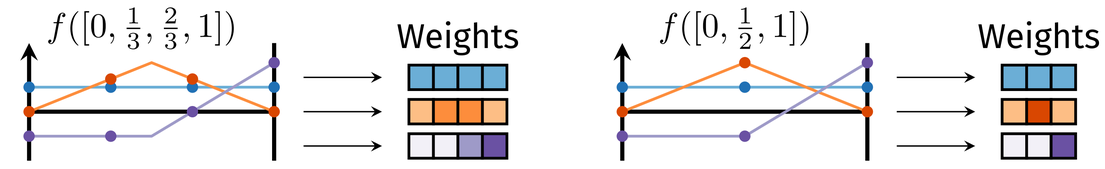
FSPool

- Pooling (set-to-vector) is used in lots of set tasks, but we want a **learnable**, **fast**, **permutation-invariant** alternative to sum and max.
- Numerical **sorting** is invariant, so let's use that! To handle variable-size sets, we can use "continuous" weights.

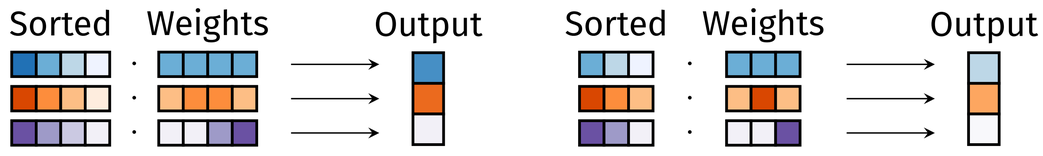
Step 1:
Sort each feature across the elements of the set.



Step 2:
Define continuous weights with piecewise linear functions. Evaluate at four positions for set of size four, three positions for set of size three, and so on.



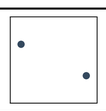
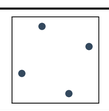
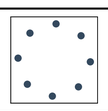
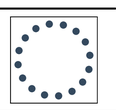
Step 3:
Dot product to pool into single feature vector.



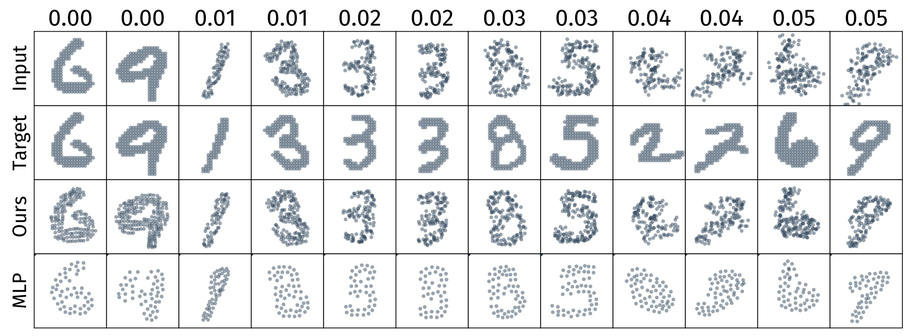
FSUnpool

- To avoid the responsibility problem, we can make the auto-encoder **permutation-equivariant**. Rotating the square should rotate the outputs too.
- Store the permutation of the sort in FSPool, then **invert** permutation to restore the same order as the input $\implies$ permutation-equivariant.
- FSUnpool (vector-to-set): "Unpool" by doing Step 3 in reverse, then "unsort" by doing Step 1 in reverse (invert permutation).

Auto-encoding sets

Rotating polygons Eval: Chamfer loss				
FSPool & FSUnpool	**0.001**	**0.001**	**0.001**	**0.000**
MLP + Chamfer loss	1.189	1.771	0.274	1.272
MLP + Hungarian loss	1.517	0.400	0.251	1.266
Random	72.848	19.866	5.112	1.271

Denoising auto-encoder for different noise levels



Encoding sets

MNIST set classification	1 epoch of training			10 epochs of training		
Weights from auto-encoder:	Frozen	Unfrozen	Random init	Frozen	Unfrozen	Random init
FSPool	**82.2%**$_{\pm 2.1}$	**86.9%**$_{\pm 1.3}$	**84.7%**$_{\pm 1.9}$	**84.3%**$_{\pm 1.8}$	**91.5%**$_{\pm 0.5}$	**91.9%**$_{\pm 0.5}$
Sum	76.6%$_{\pm 1.3}$	68.7%$_{\pm 3.5}$	30.3%$_{\pm 5.6}$	79.0%$_{\pm 1.0}$	77.7%$_{\pm 2.3}$	72.7%$_{\pm 3.4}$
Mean	25.7%$_{\pm 3.6}$	32.2%$_{\pm 10.5}$	30.1%$_{\pm 1.6}$	36.8%$_{\pm 5.0}$	75.0%$_{\pm 2.7}$	73.0%$_{\pm 1.7}$
Max	73.6%$_{\pm 1.3}$	73.0%$_{\pm 3.5}$	56.1%$_{\pm 5.6}$	77.3%$_{\pm 0.9}$	80.4%$_{\pm 1.8}$	76.9%$_{\pm 1.3}$

CLEVR		Epochs to reach accuracy			Time for
	Accuracy	98.00%	98.50%	99.00%	350 epochs
FSPool	**99.27%**$_{\pm 0.18}$	**141**$_{\pm 5}$	**166**$_{\pm 16}$	**209**$_{\pm 33}$	8.8 h
RN	98.98%$_{\pm 0.25}$	144$_{\pm 6}$	189$_{\pm 29}$	*268$_{\pm 46}$	15.5 h
Janossy	97.00%$_{\pm 0.54}$	–	–	–	11.5 h
Sum	99.05%$_{\pm 0.17}$	146$_{\pm 13}$	191$_{\pm 40}$	281$_{\pm 56}$	8.0 h
Mean	98.96%$_{\pm 0.27}$	169$_{\pm 6}$	225$_{\pm 31}$	273$_{\pm 33}$	8.0 h
Max	96.99%$_{\pm 0.26}$	–	–	–	8.0 h

- Replace sum or max with **FSPool** for consistent improvements.
- Also improves Relation Networks and a top graph classifier.
- **FSUnpool** avoids the responsibility problem, but only in auto-encoders.
  > See *Deep Set Prediction Networks* poster at this workshop for general solution.