Summer Review 3

Modularity Matters: Learning Invariant Relational Reasoning Tasks

Jason Jo[1], Vikas Verma[2], Yoshua Bengio[1]
1: MILA, Universit de Montral
2: Department of Computer Science Aalto University, Finland

Reviewed by : Arshdeep Sekhon

[1]Department of Computer Science, University of Virginia
https://qdata.github.io/deep2Read/

Relational Reasoning

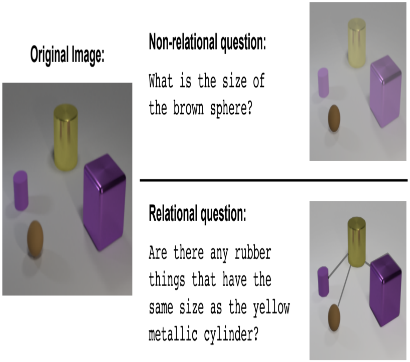


Figure: An image containing four objects is shown alongside non-relational and relational questions.[]

Invariant Relational Reasoning

Figure: OCR: humans can accommodate other variables of presentation, including specularity and fluctuating contrast. In contrast, computer programs that can cope with the variability of the presentations of the characters suffer from false alarms (false detections) between or overlapping the real characters, or in the

The Task: Invariant Relational Reasoning

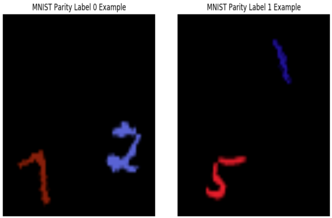


Figure 1: **(Left)** Label 0 example: MNIST digit pair $\{2, 7\}$ with different parity (one odd digit, one even digit) and **(Right)** Label 1 example: MNIST digit pair $\{1, 5\}$ of the same parity (both odd digits). Digits are subject to random translations, scalings, rotations and coloring. Best viewed in color.

Figure: The MNIST parity task

The Task: Invariant Relational Reasoning

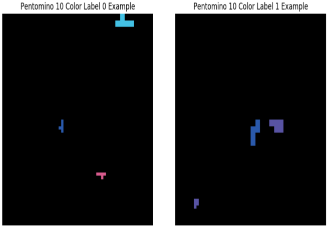


Figure 2: **(Left)** Label 0 example (all the shapes are of the same sprite type) and **(Right)** Label 1 example (there exists a sprite of a *different type* than the other sprites). Sprites are subject to random translations, scalings, rotations and coloring. Best viewed in color.

Figure: The colorized Pentomimo task

Invariant *Relational Reasoning*

Labels encode a semantic rule between the objects in the image

- MNIST parity digits encode *AND* on the parity of the digits
- Pentomino Task: *XOR* operation.
- For example, if there are 3 sprites in each figure and two types of sprites in the data, (AAA) or (BBB), label $=0$ else label$=1$

Invariant Relational Reasoning

- Invariant: the images can undergo random translation, scaling, rotation and coloring transforms.
- But the label doesn't change

Deep CNNs and Importance of Invariant representations

- DeepCNNs state-of-the-art in visual tasks.
- work well in *i.i.d* settings.
- But Adversarial Noise breaks most models.
- very sensitive to *out of distribution* settings
- Need for discriminative and highly invariant representations.

DeepCNNs: Distributed Representations

- , Most can be interpreted as learning deep hierarchies of fully distributed features
- for features f_l^1, f_l^2 at level l of the hierarchy, these features get applied to the same input y_{l-1}.
- Many to many relationships between two representations.
- Each concept is represented by many neurons.
- Each neuron participates in the representation of many concepts.[1]

[1] Geoffrey Hinton's lecture

Current Deep CNNs and IRR tasks

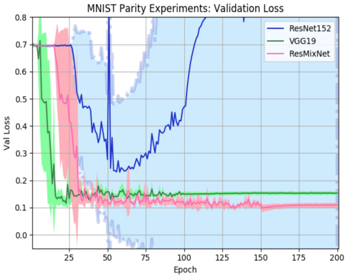


Figure:

Fully Distributed to *Modularized* Prior: Motivation

- modularized: A series of functional magnetic resonance imaging (fMRI) studies revealing substantial evidence for a distinct cortical region in humans that responds selectively to images of the human body, as compared with a wide range of control stimuli.[]
- Modularity leads to invariance: invariant hypothesis that computational goal of the ventral stream is to compute an invariant-to-transformations and discriminative signature for recognition leads to modularity.[]

The Tasks

- require a machine learning model to learn higher order invariances
- because the objects undergo random translation, scaling, rotation and coloring transformations
- additional challenge introduced by restricting the training set size
- **high invariance, low sample regime**

Differences between the two tasks

- Object distribution:
  - MNIST: curvilinear digit strokes
  - Pentomino: rigid polygonal shapes
- Relational Rule: XOR vs AND
- Pentomino has more sparsity, and more freeedom for translation
- Curved edges in MNIST vs Straight edges in Pentomino : curved are more important for discriminative purposes.

Invariant and Selective Representations

$$I \sim I' \implies \mu(I) = \mu(I') \tag{1}$$

$$\mu(I) = \mu(I') \implies I \sim I' \tag{2}$$

Together with invariance, selectivity asserts that two points have the same representation if and only if they are one a transformation of the other.

Why modularize?

- Distributed Representations are a problem when :
  - large number of invariances
  - small data
- interference problem in such cases: for example in an MLP, one neuron may receive conflicting gradient updates dependent on output units
  - but, using separate neurons for each output unit is wastage of resources
- When a dataset has a large number of invariances, a machine learning model must learn to associate a large number of seemingly unrelated patterns with one another, worsens interference problem
- for example the MNIST Parity task: model must learn to associate the digit pairing 1, 4 with 2, 7 : the same label of 0, but the digit pairings have different geometric properties.

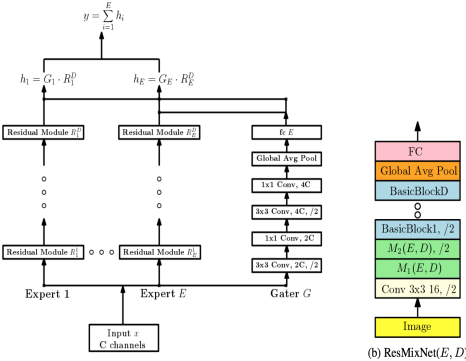
Modularizing

- specialized sub-modules in the architecture.

Hypothesis

"In the case of invariant relational learning, we hypothesize that modularity allows for the development of specialized neural circuitry that can learn to associate many seemingly unrelated patterns."

G is an E-length probability vector



(a) $M(E, D)$: A mixture of E experts, where each expert is a D stack of residual modules and a gater network G which weights all the experts and forms an additive mixture.

(b) ResMixNet(E, D) consisting of D stacks of residual modules and E experts. Best viewed in color.

Figure: M(E,D)

$$y = \sum^{E} G[i]E[i] \quad (3)$$

Modular

ResMixNet

- The first layer: fully distributed convolutional layer
- appropriate prior for the MNIST Parity and colorized Pentomino tasks because each of those datasets share low level features, e.g. curvilinear digit strokes for MNIST Parity and straight edges for the colorized Pentomino.
- each expert receives the same input as any other expert, but then each expert learns its own specialized representation through its D stack of residual modules.
- stack two expert modules M1 and M2 together and then have the third block be a D stack of Basic Block residual modules.

Results: MNIST Parity

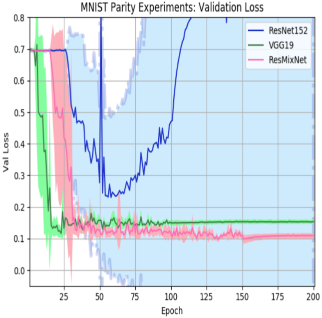


Figure 4: Average Validation Loss of the best performing models: ResNet152, VGG19-BN and ResMixNet(2,2).

Figure: Pentomino Results

Results: Pentomino

Table 2: Pentomino 10 Color Generalization Results

Model	Parameter Count	Test Error
ResNet26	370K	$34.61 \pm 4.67\%$
ResNet50	758K	$30.31 \pm 1.74\%$
ResNet152-Bottleneck	3.66M	$31.02 \pm 3.06\%$
VGG19-BN	20M	$34.01 \pm 4.72\%$
ResMixNet(4,1)	**193K**	$\mathbf{0.88 \pm 0.12\%}$

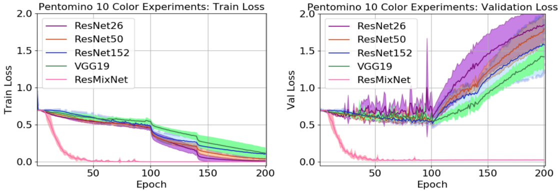


Figure 5: **(Left)** Train Loss and **(Right)** Validation Loss performance on the Pentomino 10 color dataset. Best viewed in color.

Figure: Pentomino Results

Table 3: CIFAR-10, CIFAR-100 and SVHN Results

Model	Dataset	Num. Params	Test Accuracy
ResNet50	CIFAR-10	758K	93.48%
ResMixNet(5,3)	CIFAR-10	748K	92.74%
ResNet50	CIFAR-100	764K	71.81%
ResMixNet(5,3)	CIFAR-100	754K	66.35%
ResNet50	SVHN	758K	95.45%
ResMixNet(5,3)	SVHN	748K	95.58%

Figure: CIFAR10 Results

- ResMixNet may not be a good prior for CIFAR-100

Links to relevant papers

- On invariance and selectivity in representation learning