



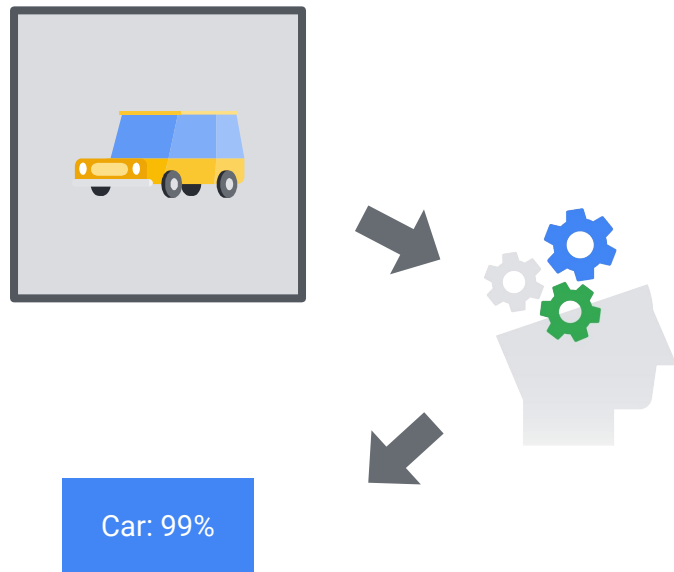
Recursive Sketches for Modular Deep Learning

Badih Ghazi, Rina Panigrahy, **Joshua R. Wang** (Google Research)

ICML 2019: Long Beach, CA

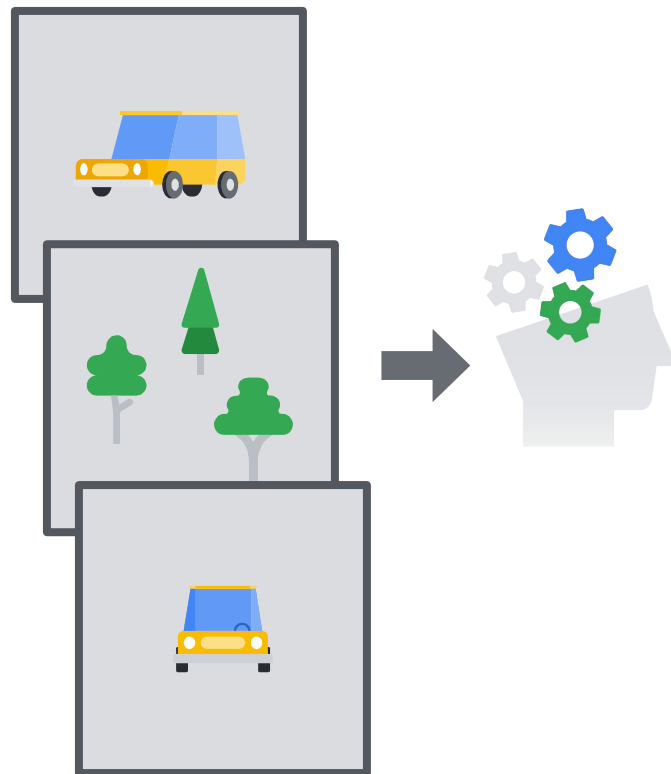
Object Recognition

- Rich literature around ML techniques for object recognition.
- Typical problem format.
 - Input: Picture
 - Output: Its object(s)



Object Memory

- This talk: twist on typical task.
 - Input: Picture
 - Output: Succinct representation of its object(s)
- **Theorem.** Can utilize model that solves the previous task as a primitive to solve this task.



Modular Networks 101

- Module: independent neural network component.
- Modules communicate via one's output serving as another's input.
- **Intuition.** Convolutional Neural Nets first find low-level objects (edge) and build up to high-level objects (cat).

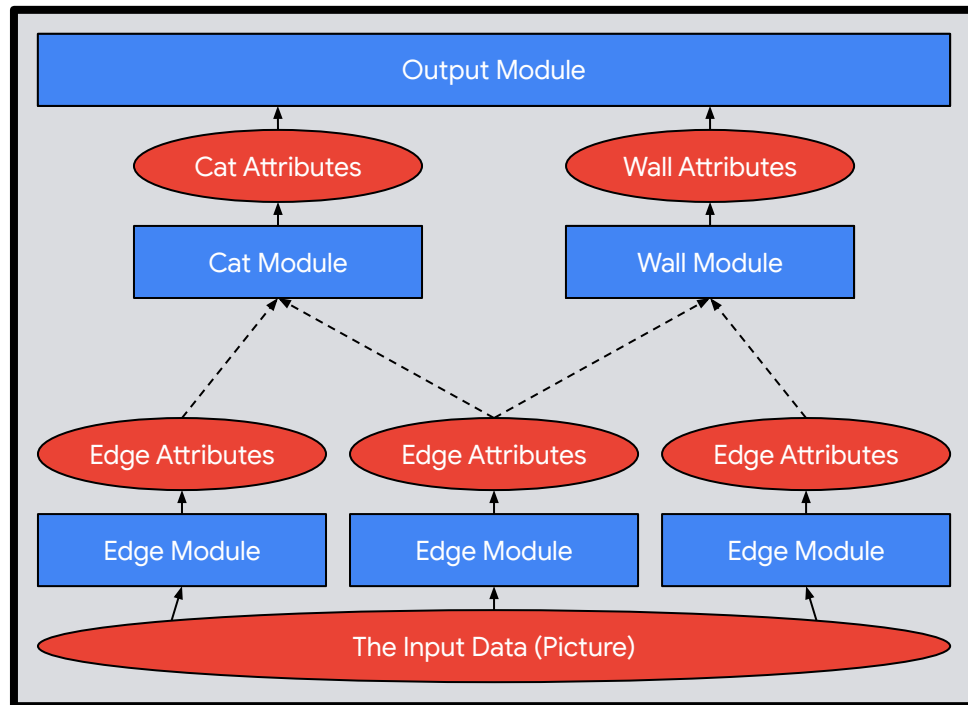
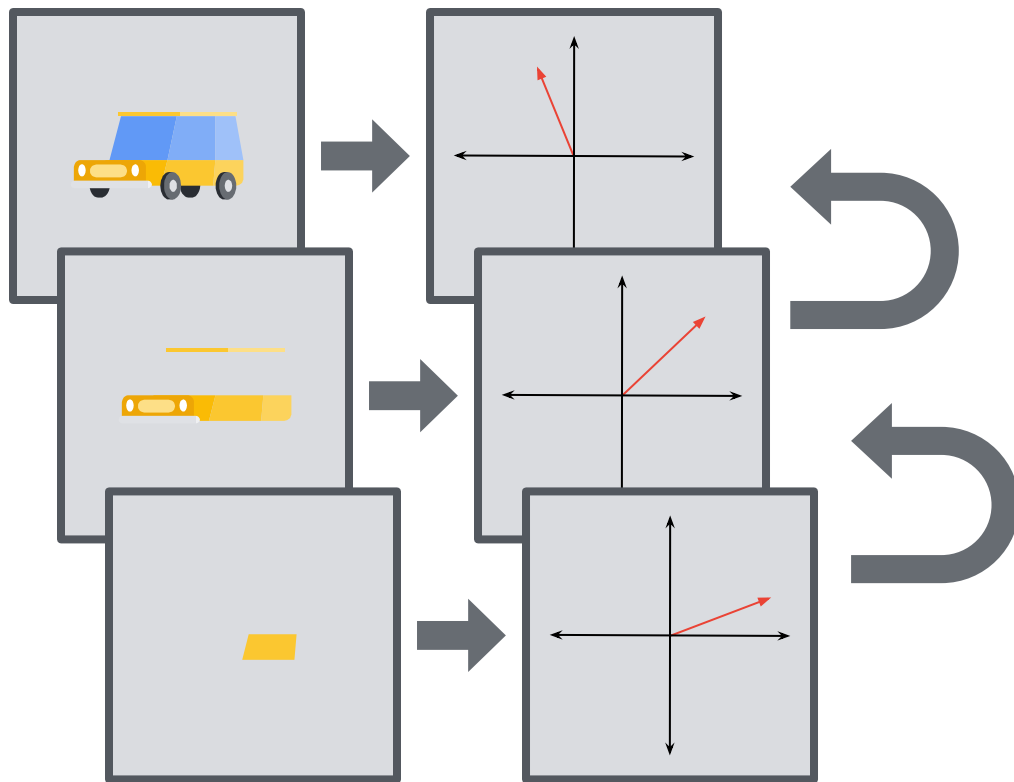


Figure. Abstract view of modular network processing image of a room.

Recursive Sketches

- Our mechanism creates a sketch for each object detected by the modular network.
- Recursive, because sketch of an object incorporates the sketch of sub-objects.
- Sketching tricks: (i) apply random matrix and (ii) take a weighted sum.
- Input represented by **top-level sketch**.



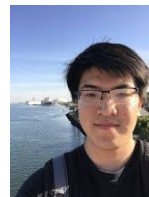
Provable Sketch Properties

- **Attribute Recovery.** Object attributes can be approximately recovered from top-level sketch.
- **Sketch-to-Sketch Similarity.** Two completely unrelated sketches have small inner product; two sketches with similar objects have large inner product.
- **Summary Statistics.** If there are multiple objects produced by same module, can approximately recover their summary statistics like count/mean.
- **Graceful Erasure.** Erasing all but sketch prefix, we still get above properties (but increase recovery error).


Recursable Dictionary Learning

- Previous slide properties required knowing random matrices chosen by the sketch.
- **Recursable Dictionary Learning.** Given enough sketches, can approximately recover the random matrices (and object attribute vectors).
- Dictionary learning “unwinds” one level of sketching recursion.
- **Trickier than Classical Dictionary Learning.** The noisy output becomes noisy input for the next stage, so the error guarantee and error tolerance must be of the same form.

Recap: Recursive Sketches



- Takeaway Message.** Can utilize model that solves the object recognition as a primitive to generate useful and efficient sketches of inputs.
- Computing our Sketches.** Built out of (i) apply random matrix and (ii) take a weighted sum.
- Let's chat!** Poster #73 @ Pacific Ballroom.



Google Research
Equal Contribution

Recursive Sketches for Modular Deep Learning

Badih Ghazi* Rina Panigrahy* Joshua R. Wang*

Problem Setup

- Object recognition is a well studied ML problem
- Typical problem format (object recognition)
 - Input: Picture
 - Output: Its Object(s)
- Humans capable of remembering images and relating them to each other...
- Modified problem format (object memory)
 - Input: Picture
 - Output: Succinct representation of its object(s)
- Takeaway Message.** With access to a pre-trained (modular) network that solves the object recognition problem, we can solve the object memory problem.

Sketching Mechanism Properties

- Attribute Recovery.** Object attribute vectors can be approximately recovered from the top-level sketch.
- Sketch-to-Sketch Similarity.** Two completely unrelated sketches have small inner product. Two sketches that have similar objects will have large inner product.
- Summary Statistics.** If there are multiple objects of the same class, can approximately recover summary statistics like count/mean.
- Grayscale Erasure.** Erasing sketch suffices maintains these properties (but will increase error).
- Above properties require knowing the random matrices chosen by the sketching mechanism...
- Recursive Dictionary Learning.** Given enough (input, sketch) pairs, the random matrices can be approximately recovered.

Random Matrix Distribution

Sketch Matrix entries are i.i.d. Gaussian $\mathcal{N}(0, \frac{1}{n})$. n = Number of Rows.

Proofs using following properties of distribution.

- Sparsity.** Limits collisions and allows recovery of inputs.
- Column signatures.** Allow recovery of column indices.
- Module signatures.** Allow clustering of columns into modules.
- Random sign flips.** Decorrelate coordinates in different blocks.
- D-Desynchronization Property.**
 - For any unit vectors u, v , with high probability over R :

$$|u^T R v| \leq \frac{1}{\sqrt{D}} + \epsilon$$
- S-sammetry Property.**
 - For any unit vector u , with high probability over R :

$$|u^T R^T u| \leq \frac{1}{\sqrt{S}} + \epsilon$$
- Intuition.** Applying random matrix R "scrambles" a vector. It can be decoded with R^T , but has low dot product w/ vectors independent of R .

Modular Nets 101

- Several independent neural networks (modules) communicating via one's output serving as another's input.
- Have inspired several practical architectures:
 - Neural Modular Networks (NMGNet, NMG-T)
 - Capsule Neural Networks (HGTOO, HKW11, SH17)
 - PathNet (FB17)
- For object recognition, a module identifies a class of objects and outputs the attributes of the object that it recognized.

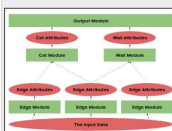


Figure: Abstract view of how a modular network processing the image of a room. Modules are drawn as rectangles, inputs/outputs are ovals.

The Road to a Sketch

- Initial Simplifying Assumptions:
 - Each module produces at most one object.
 - Network has depth one.
- Prototype A (Assumptions 1,2)**

$$y_{i,j} = \sum_{k=1}^n w_{i,j,k} x_k$$

$$y_{i,j} = \sum_{k=1}^n w_{i,j,k} x_k$$
- Prototype B (Last Assumption 2)**

$$y_{i,j} = \sum_{k=1}^n w_{i,j,k} x_k$$

$$y_{i,j} = \sum_{k=1}^n w_{i,j,k} x_k$$
- Sans Assumption 1:** multiplying by a transpose yields sum of attribute vectors instead of single attribute vector.
- Summary Statistics?** Incorporating μ , allows us to recover the count (and thus mean too).
- Attribute Recovery?** Use "transparent" matrix $(R^T)Y$ to also distinguish each object.
- Final Sketching Mechanism (No Assumptions)**

$$y_{i,j} = \sum_{k=1}^n w_{i,j,k} x_k$$

$$y_{i,j} = \sum_{k=1}^n w_{i,j,k} x_k$$
- Sans Assumption 2:** our sketches need to be recursive. Use tuple technique to combine object with inputs.

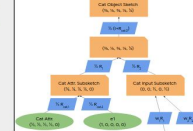


Figure: The object sketch of a cat (final mechanism). (Sub)-sketches are cat rectangles, random matrices are parallelograms. Arrows run from input to output vector.

References

[AR13] Jacob Andreas, Marcus Rohlfach, Trevor Darrell, and Dan Klein. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 39–48, 2013.

[FB17] Chiyuan Francisco, Dylan Branson, Charles Simabali, Viet Doan, David Wu, Austin A. Evans, Alexander Pfister, and Drew Bernstein. Pathnet: Evolution chooses good-to-good descent in rapid neural architecture search. *arXiv preprint arXiv:1701.08560*, 2017.

[HB*17] Honglak Lee, Jacob Andreas, Marcus Rohlfach, Trevor Darrell, and Ronit Rubinfeld. Learning to parse: End-to-end neural parsing. In *Recent Advances in Computer Vision*, pages 94–107, 2017.

[SH17] Geoffrey E Hinton, Zoubin Ghahramani, and Terence J. Sejnowski. Learning to parse: An attention-based recursive neural network. *arXiv preprint arXiv:1701.08560*, 2017.

[SH17] Geoffrey E Hinton, Zoubin Ghahramani, and Terence J. Sejnowski. Learning to parse: An attention-based recursive neural network. *arXiv preprint arXiv:1701.08560*, 2017.

[SH17] Saeed Saleem, Nicholas French, and Geoffrey E Hinton. Dynamic neural network capsules. In *Advances in Neural Information Processing Systems*, pages 3006–3016, 2017.