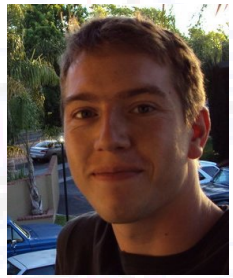
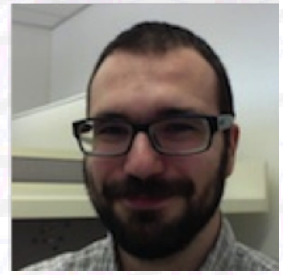
Learning Molecular Fingerprints from the Graph Up

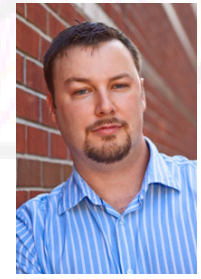



David Duvenaud, Dougal Maclaurin,

Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli,

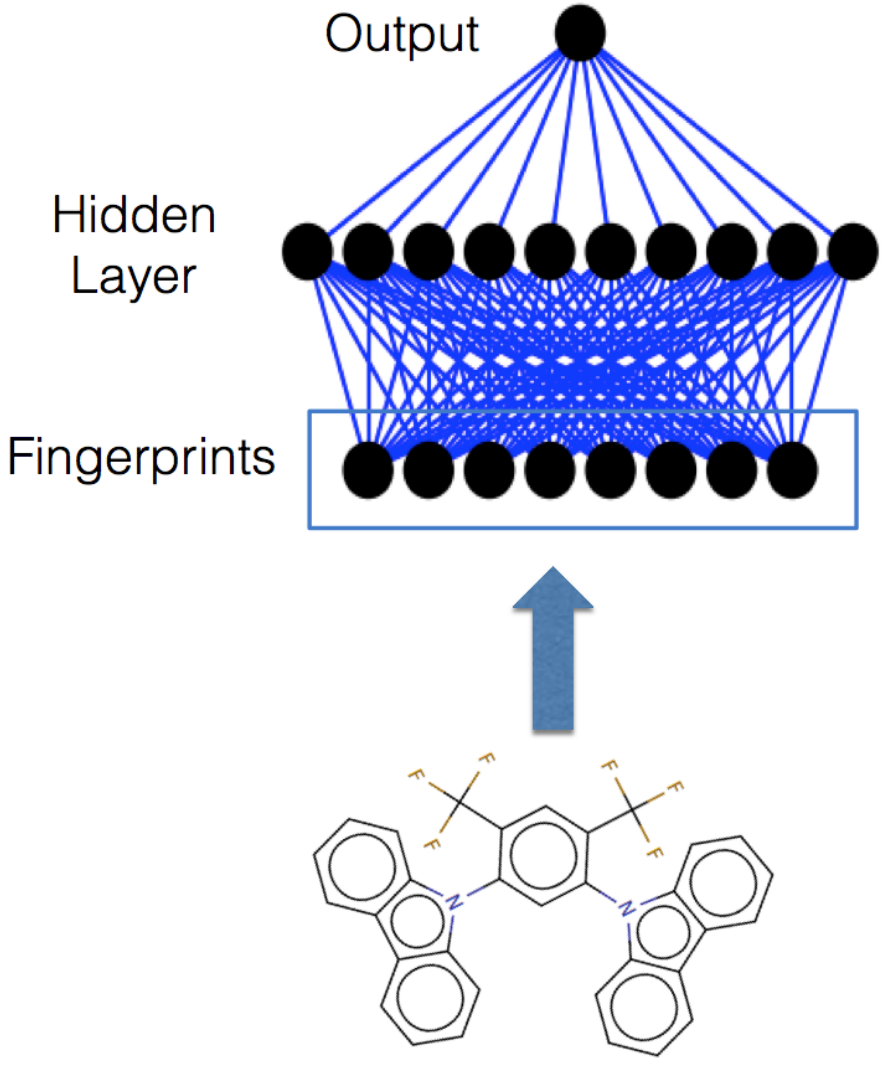




Timothy Hirzel, Alán Aspuru-Guzik, Ryan P. Adams

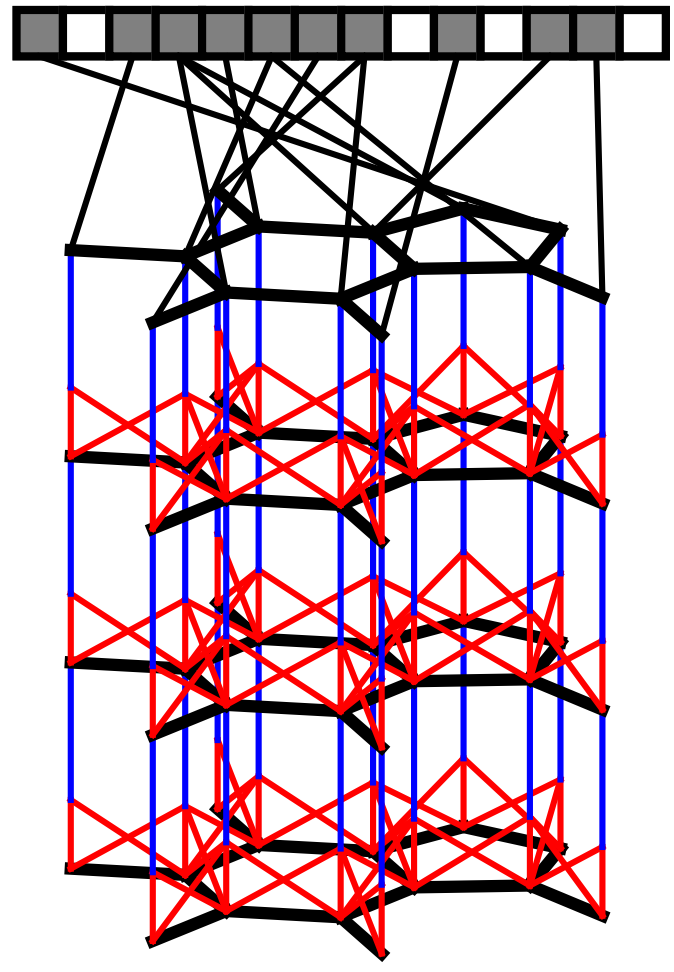
Motivation

- Want to do regression on molecules
- For virtual screening of drugs, materials, etc.
- Problem: Molecules can be any size and shape
- Only know how to learn from fixed-size examples.
- How to take a molecule in and produce a fixed-size vector?



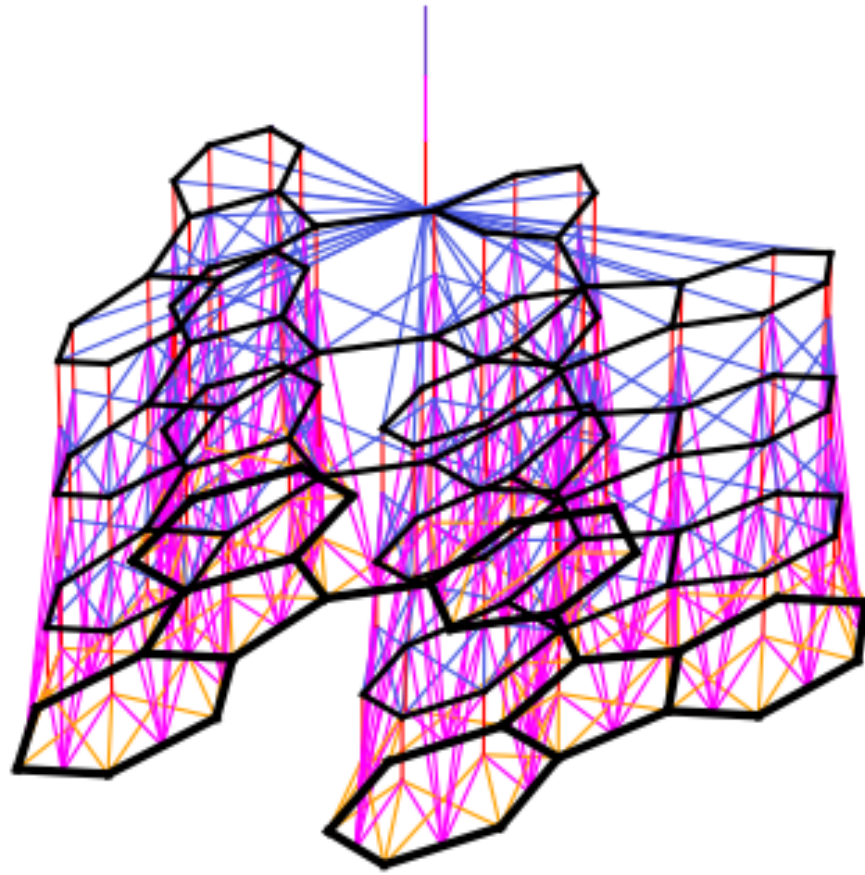
Circular Fingeprints

- Standard method lists all substructures below a certain size
- Can do this by combining hashes of each atom with and bonded neighbors
- Hash value indexes into a fixed-sized vector
- Problem: can’t optimize with gradients



What would Ryan do?

- Maybe we can build a message-passing network
- same function is applied to each node (atom) and its neighbors
- Like a convolutional net
- At the top, add all node's vectors together
- If we use a softmax, this generalizes circular fingerprints



Continuous-izing Circular Fingerprints

Circular fingerprints

Input: molecule, radius R, fingerprint length S
Initialize: fingerprint vector $\mathbf{f} \leftarrow \mathbf{0}_S$
for each atom a in molecule **do**
  $\mathbf{r}_a \leftarrow g(a)$ ▷ lookup atom features
for $L = 1$ to R **do** ▷ for each layer
  for each atom a in molecule **do**
    $\mathbf{r}_1 \dots \mathbf{r}_N = \text{neighbors}(a)$
    $\mathbf{v} \leftarrow [\mathbf{r}_a, \mathbf{r}_1, \dots, \mathbf{r}_N]$ ▷ concatenate
    $\mathbf{r}_a \leftarrow \text{hash}(\mathbf{v})$ ▷ hash function
    $i \leftarrow \text{mod}(r_a, S)$ ▷ convert to index
    $\mathbf{f}_i \leftarrow 1$ ▷ Write 1 at index
Return: binary vector $\mathbf{f}$

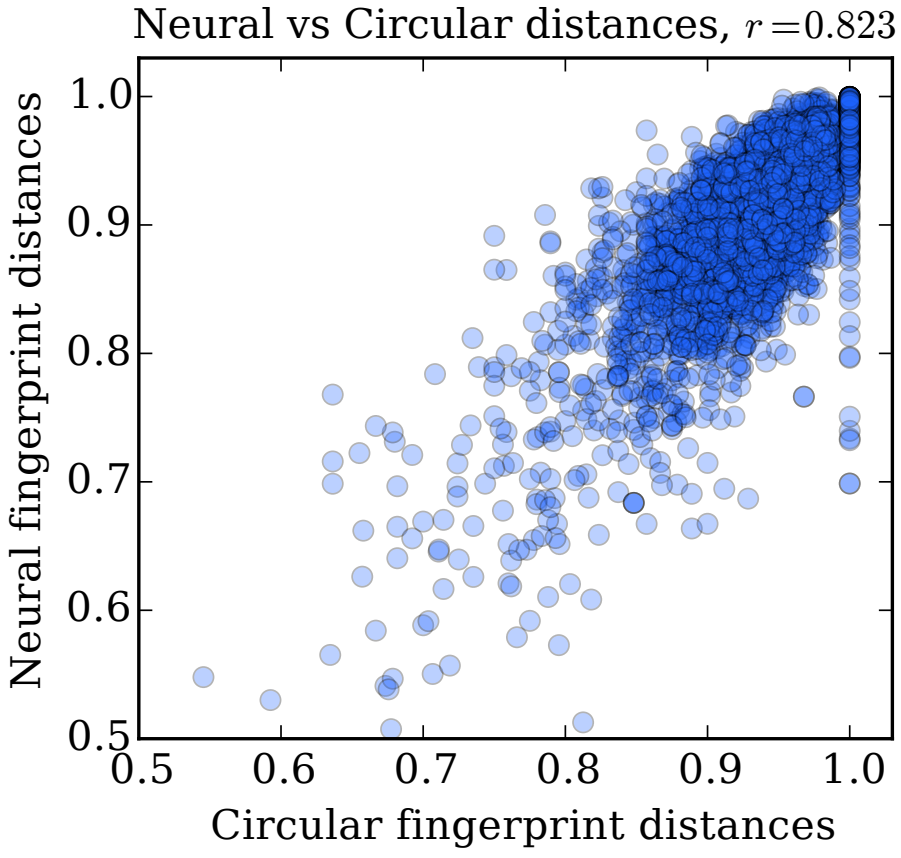
Neural graph fingerprints

Input: molecule, radius R, weights $H_1^1 \dots H_R^5$, output weights $W_1 \dots W_R$
Initialize: fingerprint vector $\mathbf{f} \leftarrow \mathbf{0}_S$
for each atom a in molecule **do**
  $\mathbf{r}_a \leftarrow g(a)$ ▷ lookup atom features
for $L = 1$ to R **do** ▷ for each layer
  for each atom a in molecule **do**
    $\mathbf{r}_1 \dots \mathbf{r}_N = \text{neighbors}(a)$
    $\mathbf{v} \leftarrow \mathbf{r}_a + \sum_{i=1}^{N} \mathbf{r}_i$ ▷ sum
    $\mathbf{r}_a \leftarrow \sigma(\mathbf{v} H_L^N)$ ▷ smooth function
    $\mathbf{i} \leftarrow \text{softmax}(\mathbf{r}_a W_L)$ ▷ sparsify
    $\mathbf{f} \leftarrow \mathbf{f} + \mathbf{i}$ ▷ add to fingerprint
Return: real-valued vector $\mathbf{f}$

Every non-differentiable operation is replaced with a differentiable analog.

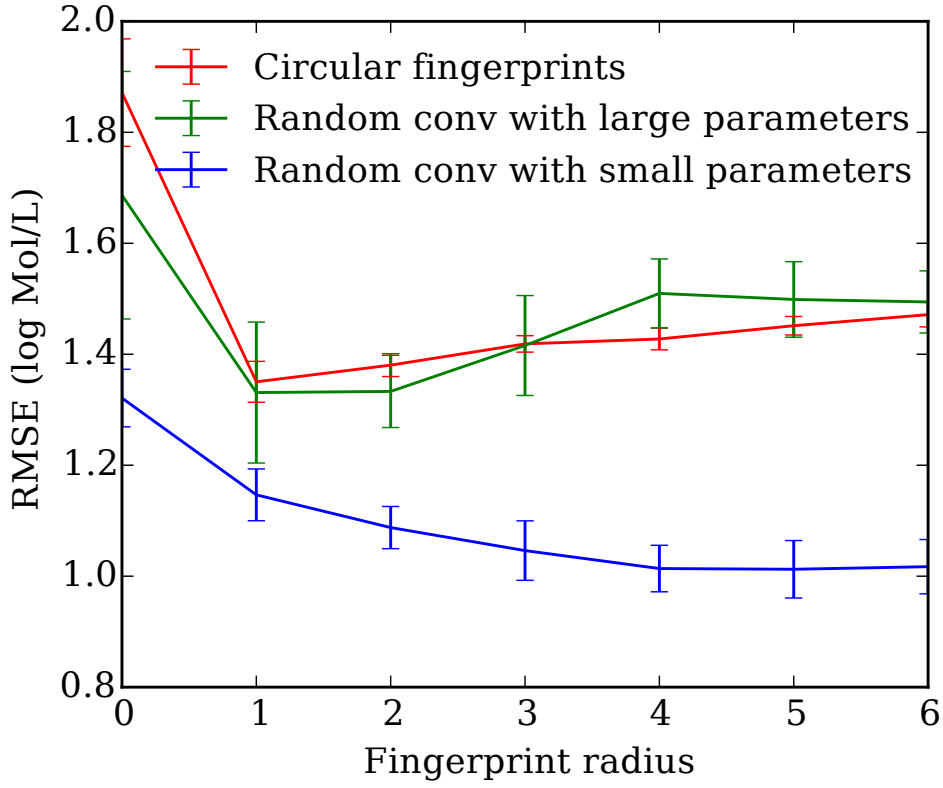
Generalizing Circular Fingerprints

- If we generalize existing fingerprints, we can't not win (unless we overfit)
- large random weights makes neural nets act like hash functions
- Looked at similarities between pairwise distances.



Generalizing Circular Fingerprints

- If we generalize existing fingerprints, we can't not win (unless we overfit)
- large random weights makes neural nets act like hash functions
- Looked at performance of random weights.



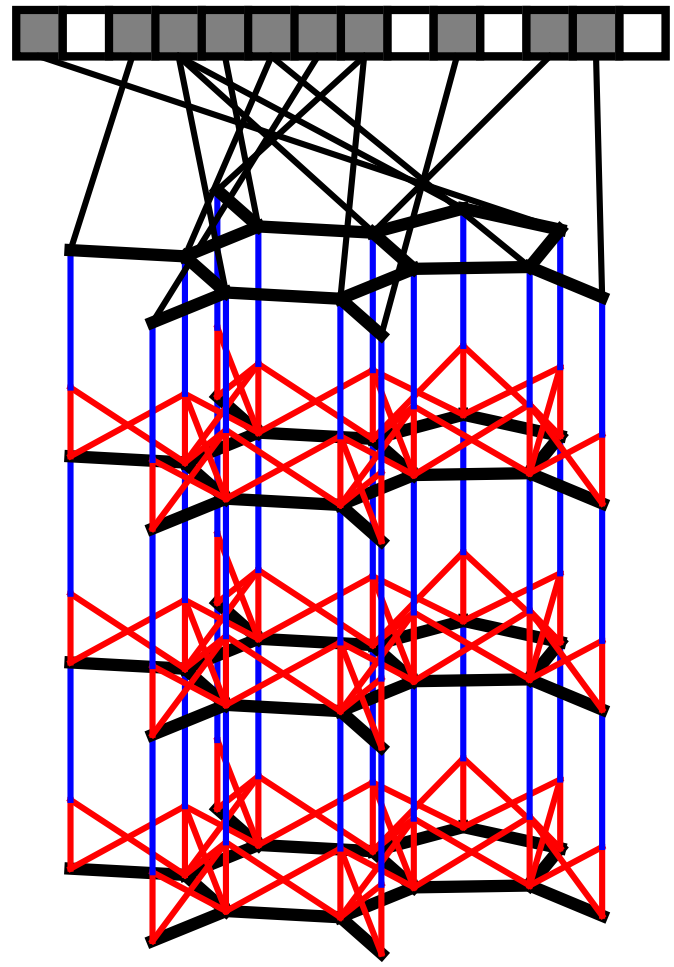
Performance

Dataset Units	Solubility log Mol/L	Drug efficacy EC_{50} in nM	Photovoltaic efficiency percent
Predict mean	4.29 ± 0.40	1.47 ± 0.07	6.40 ± 0.09
Circular FPs + linear layer	1.84 ± 0.08	$\mathbf{1.13 \pm 0.03}$	2.62 ± 0.07
Circular FPs + neural net	1.40 ± 0.15	1.24 ± 0.03	2.04 ± 0.07
Neural FPs + linear layer	0.74 ± 0.09	$\mathbf{1.16 \pm 0.03}$	2.71 ± 0.13
Neural FPs + neural net	$\mathbf{0.53 \pm 0.07}$	$\mathbf{1.17 \pm 0.03}$	$\mathbf{1.44 \pm 0.11}$

- Could also try varying depth of neural net on top (used one hidden layer here)

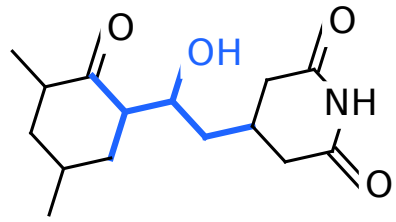
Interpretability

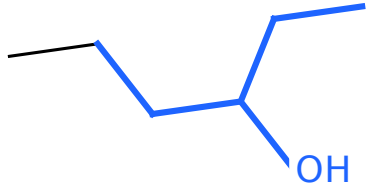
- Circular fingerprints activate for a single substructure
- No generalization
- No notion of similarity
- Let's put a linear layer on top of neural fingerprints and examine which fragments activate most predictive features.

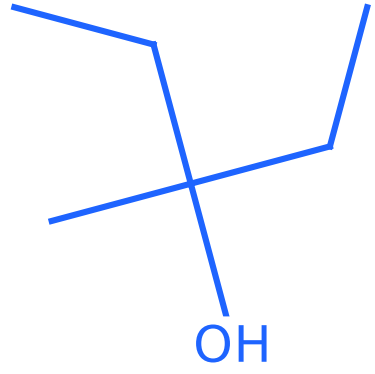


Interpretability: Solubility

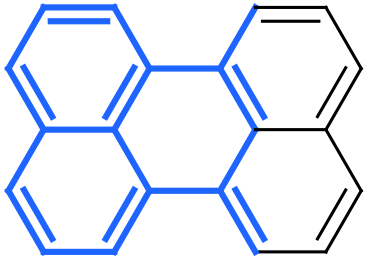
Fragments activating feature most predictive of solubility:

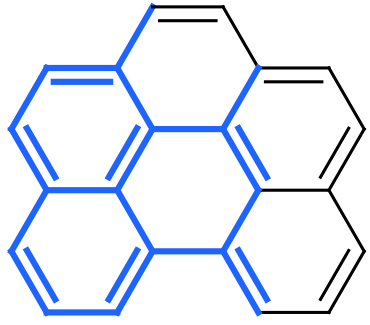


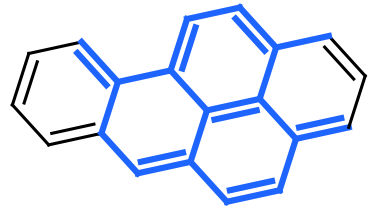




most predictive of insolubility:

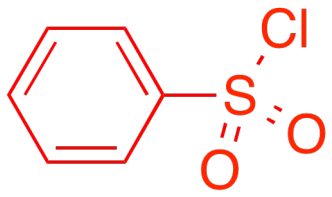


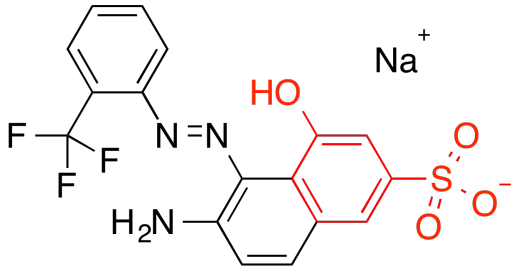


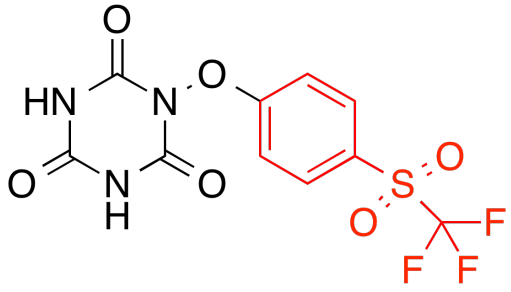


Interpretability: Toxicity

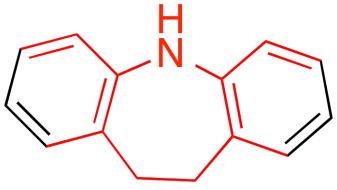
Fragments most activated by toxicity feature on SR-MMP dataset:

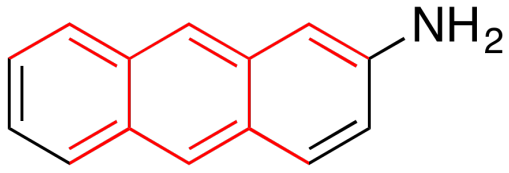


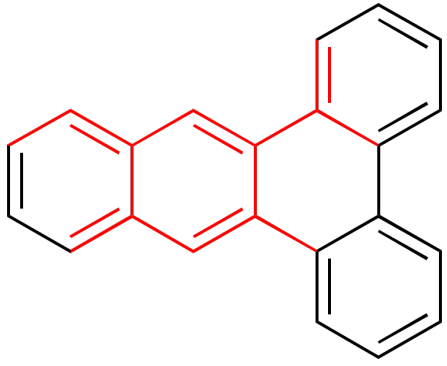




Fragments most activated by toxicity feature on NR-AHR dataset:

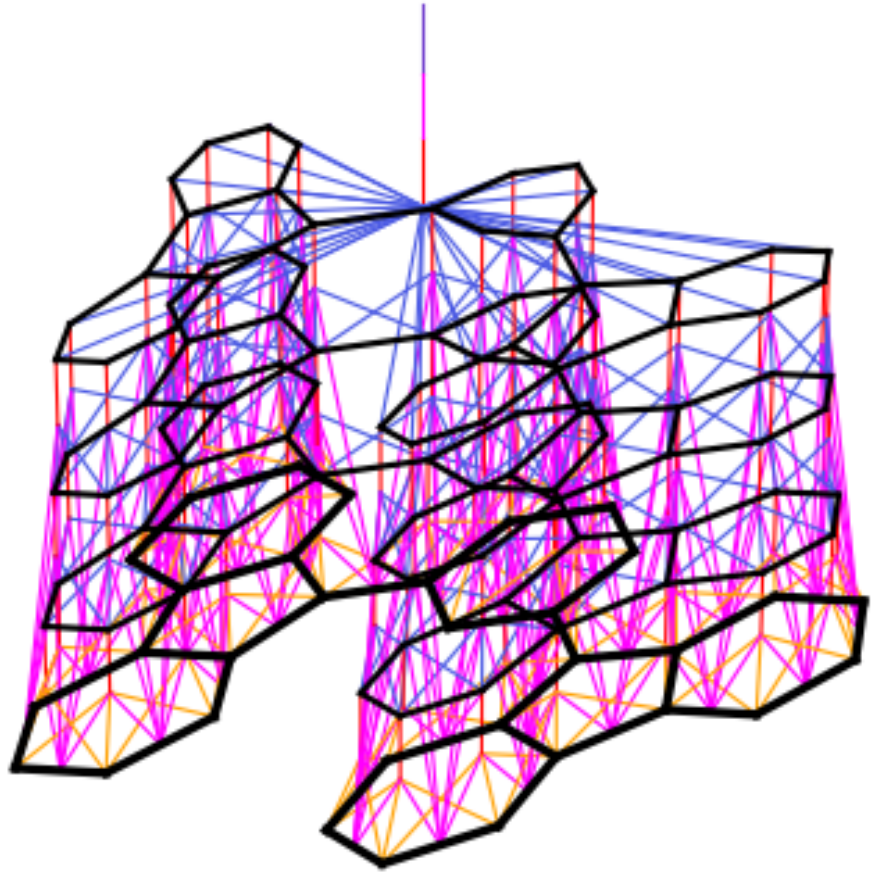






Future Work

- Limitation: Slow because of so many weight transforms
- Could use low-rank weight matrices
- Limitation: All features are local
- Could learn to “parse” molecules
- But how to take gradients?



Delaney, John S. ESOL: Estimating aqueous solubility directly from molecular structure. *Journal of Chemical Information and Computer Sciences*, 44(3):1000–1005, 2004.

Gamo, Francisco-Javier, Sanz, Laura M, Vidal, Jaume, de Cozar, Cristina, Alvarez, Emilio, Lavandera, Jose-Luis, Vanderwall, Dana E, Green, Darren VS, Kumar, Vinod, Hasan, Samiul, et al. Thousands of chemical starting points for antimalarial lead identification. *Nature*, 465(7296):305–310, 2010.

Hachmann, Johannes, Olivares-Amaya, Roberto, Atahan-Evrenk, Sule, Amador-Bedolla, Carlos, Sánchez-Carrera, Roel S, Gold-Parker, Aryeh, Vogt, Leslie, Brockway, Anna M, and Aspuru-Guzik, Alán. The Harvard clean energy project: large-scale computational screening and design of organic photovoltaics on the world community grid. *The Journal of Physical Chemistry Letters*, 2(17):2241–2251, 2011.

Tox21 Challenge. National center for advancing translational sciences. http://tripod.nih.gov/tox21/challenge, 2014. [Online; accessed 2-June-2015].