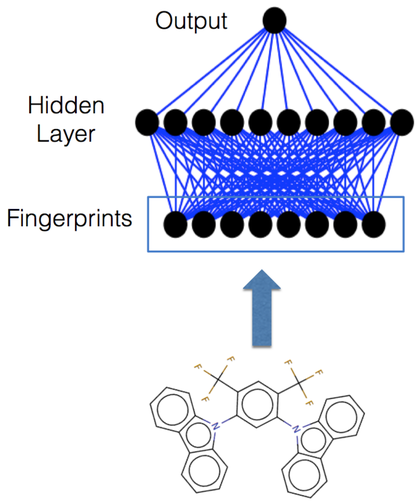
Convolutional Networks on Graphs for Learning Molecular Fingerprints

David Duvenaud*, Dougal Maclaurin*, Jorge Aguilera-Iparraguirre

UNIVERSITY OF CAMBRIDGE

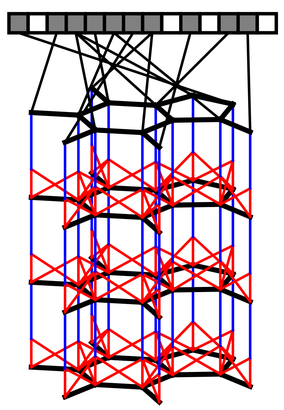
How to do regression on graphs?

- Input can be any size or shape
- Hard to turn into fixed-length vector
- In our case, graphs represent molecules
- Applications to photovoltaics, organic LEDS, flow batteries and pharmaceuticals



Circular fingerprints

- Maps variable-sized molecular graph to fixed-length binary vector
- Binary features indicate presence of substructures



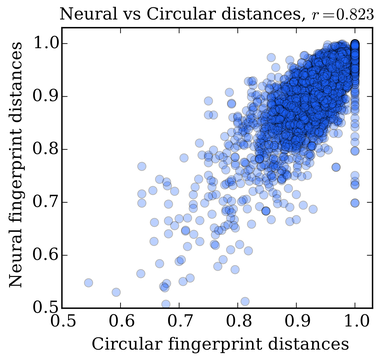
Can be efficiently computed using local operations:

- At each layer, hash the features of each atom and its neighbors/bonds
- More layers correspond to increasing radius of substructures
- Interpret each hash as integer and set that entry to one

Was state-of-the-art for large-scale regression and classification.

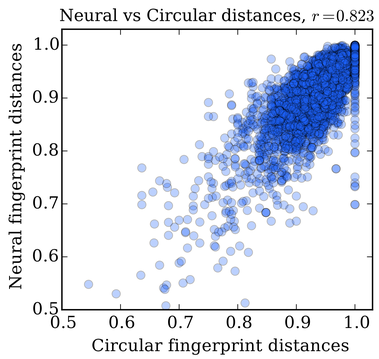
Learning Curves - Ia vs Ib

Comparison of learning



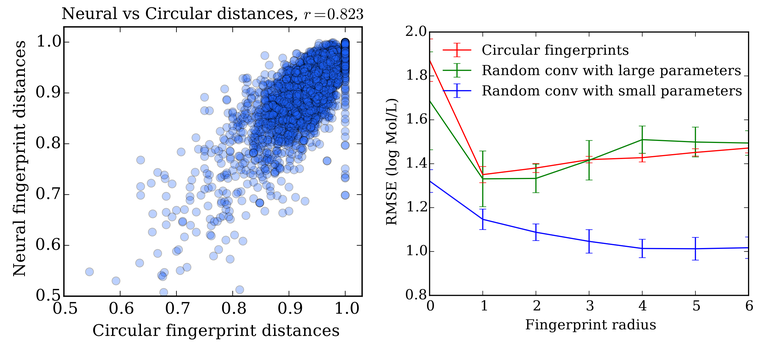
Visualisation of learned manifolds

Comparison of learning



Full VB

Large random weights give similar behavior to circular fingeprints:

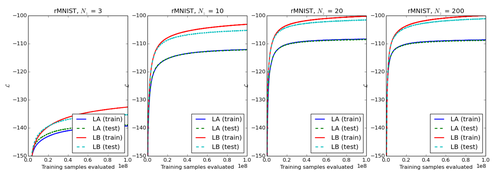


Small random weights already much better than circular fingerprints!
Can do even better by optimizing for given task.

Architecture experiments

We tested various set-up of the autoencoder

- Increasing the depth of the encoder
- Different activations functions
- Different activations functions



Conclusion

- Can learn graph features end-to-end!
- Works on other types of graphs too
- Code at `github.com/HIPS/neural-fingerprint`
- Autodiff package that works on standard Numpy code: `github.com/HIPS/autograd`