Graph Neural Networks for crystal structures

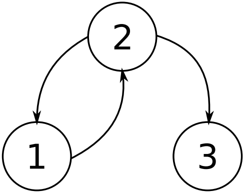
Aleksy Barcz

Warsaw University of Technology

May 8, 2014

Graph-oriented neural networks

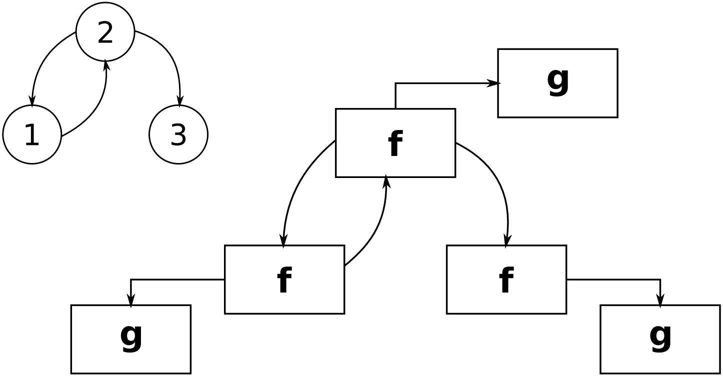
- Process information about graph structure directly
- Build meaningful representation
- Capable of classification and regression
- Based on feed-forward neural networks



Applications of graph-oriented neural networks

- Drug design
- Prediction of fuel compounds properties
- Object detection in 2D images
- Document mining

GNN - encoding network



- Building node representation: $x_n = f(\ldots)$
- Node classification: $o_n = g(x_n)$

Example - single substitution, tetragonal I cells

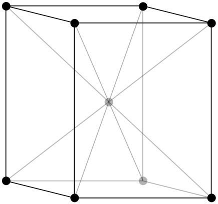


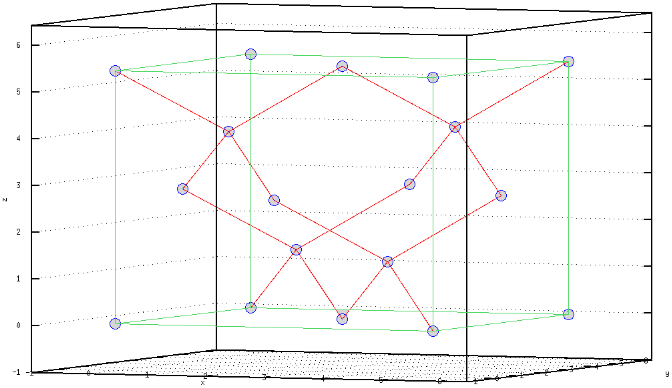
Table : Substitution existence

	accuracy	precision	recall
training set	96.00%	96.00%	96.00%
test set	94.92%	95.29%	94.51%

Table : Substitution location difference

	accuracy	precision	recall
training set	96.33%	94.26%	98.66%
test set	95.07%	92.05%	98.66%

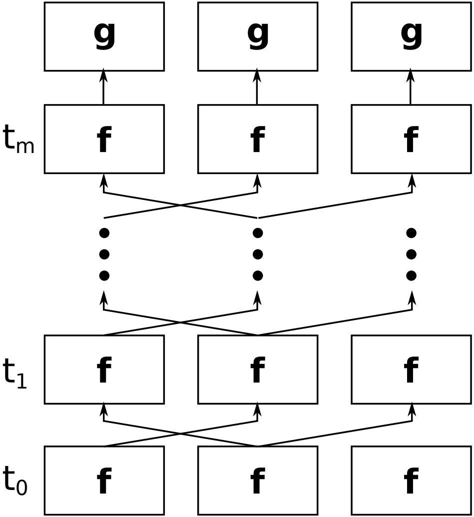
Application to Si defects - possibilities



GNN - learning algorithm

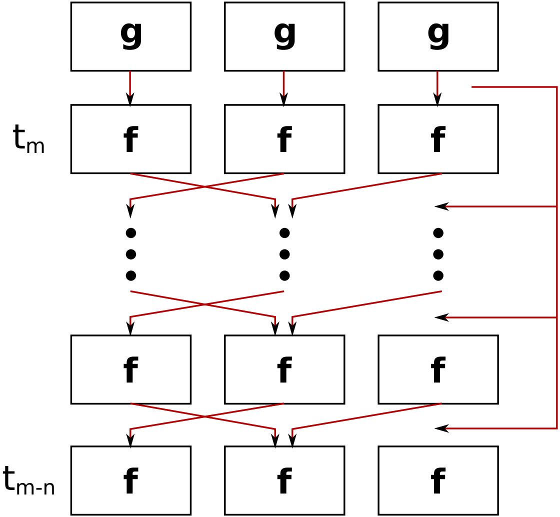
1. initialize randomly h_w and g_w weights
2. until stop criterion is satisfied:
   - random initialization of representation X
   - FORWARD : calculate $X = F_w(X)$ until fixed point is reached
   - BACKWARD : calculate $G_w(X)$ and backpropagate the error
   - update f_w and g_w weights

Forward - building representation



- unfolding

Backward - backpropagation

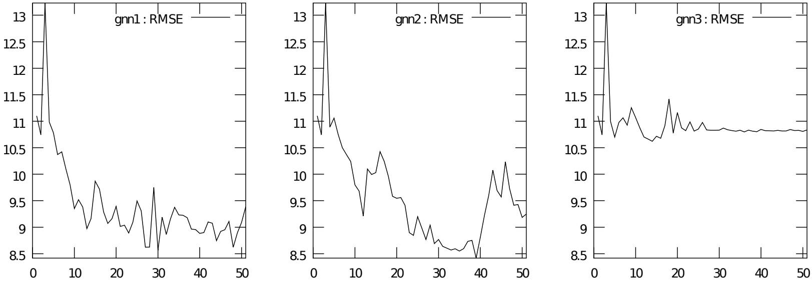


- BPTT
- Almeida-Pineda

How do we know F_w will reach fixed point?

- contraction map (Banach theorem)
- $||F_w(X_1) - F_w(X_2)|| \leq ||X_1 - X_2||$
- unique fixed point
- fixed point reached from every starting point
- very few iterations needed
- penalty imposed on F_w weights when the contraction is lost

How strong should the penalty be?



- *contractionConstant* $\in [1.2, 0.9, 0.6]$
- boundary value depends on dataset