A novel reinforcement learning algorithm for virtual network embedding

A.Zamani

Supervised by: Dr. pourahmadi

Amirkabir University of Technology

Statistical Machine Learning, December 2018

Outline

Network modeling

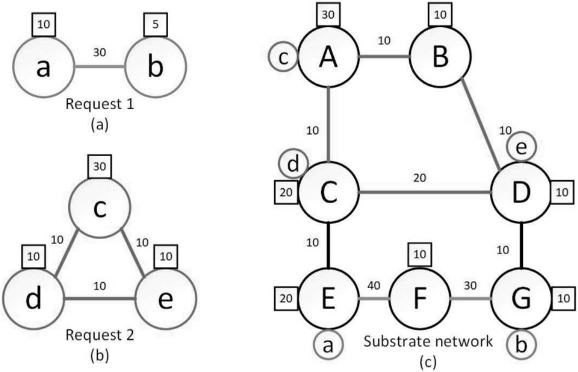
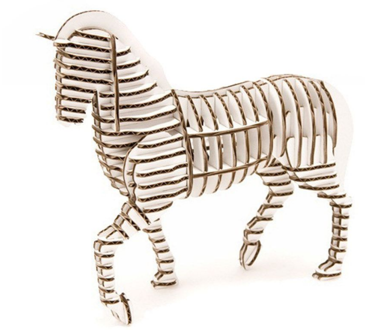


Figure: An example of virtual network embedding.

Network modeling

- Substrate network: $G^S = (N^S, L^S, A_N^S, A_L^S)$
- Request: $G^V = (N^V, L^V, C_N^V, C_L^V)$
- virtual network embedding process can be formulated as$\rightarrow$ mapping G^V to G^S : $G^S(N^V, L^V) \rightarrow G^S(N', P') \quad whereN' \subset N^S, P' \subset P^S$



Policy network

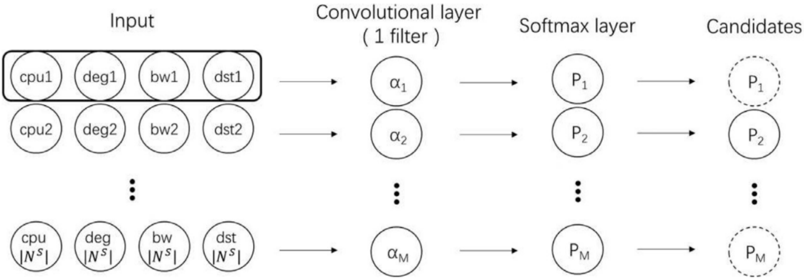


Figure: Policy network.

Feature extraction

- Computing capacity (CPU)
- Degree (DEG)
- Sum of bandwidth ($SUM^{(BW)}$)
  $\rightarrow \quad SUM^{(BW)}(n^S) = \sum_{l^s \in L(n^S)} BW(l^S)$
- Average distance to other host nodes AVG^{DST}
  $\rightarrow AVG^{(DST)}(n^S) = \frac{\sum_{\hat{n}^S \in \hat{N}^S} DST(n^S, \hat{n}^S)}{|\hat{N}^S|+1}$
- feature vector V_K $\rightarrow$
  $V_K = (CPU(n_k^S), DEG(n_k^S), SUM^{(BW)}(n_K^S), AVG^{(DST)}(n_K^S))^T$
- feature matrix M_f
  $\rightarrow M_f = (v_1, v_2, \ldots, v_{|N^S|})$

Feature extraction

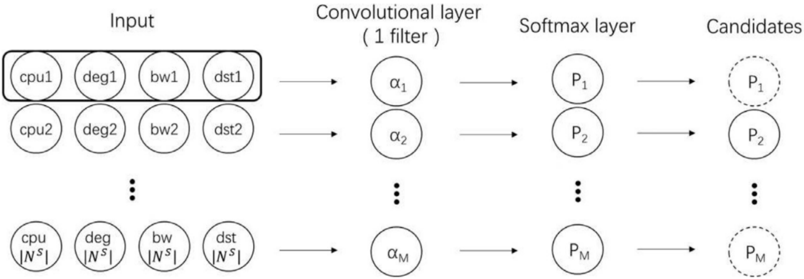
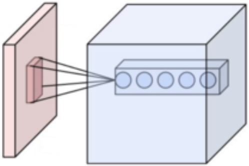


Figure: Policy network.

convolutional layer

- performs a convolution operation on th input
- produces a vector representing the available resources of each node

$h_K^c = w.v_K + b$



convolutional layer

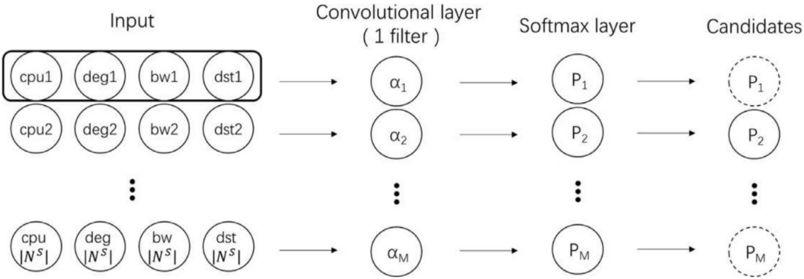
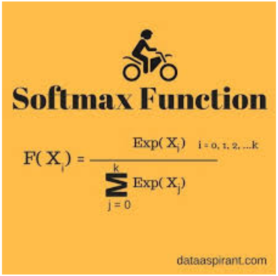


Figure: Policy network.

Softmax layer

- the n-dimensional vector into real values between 0 and 1 that add up to 1
- probability distribution over n different possible mappings

$$p_K = \frac{e^{h_k^c}}{\sum_i e^{h_K^c}}$$



Filter

- Some of the nodes are not able to host
- because they do not have enough computing resources
- add a filter to choose a set of candidate nodes with enough CPU capacities

Softmax & Filter

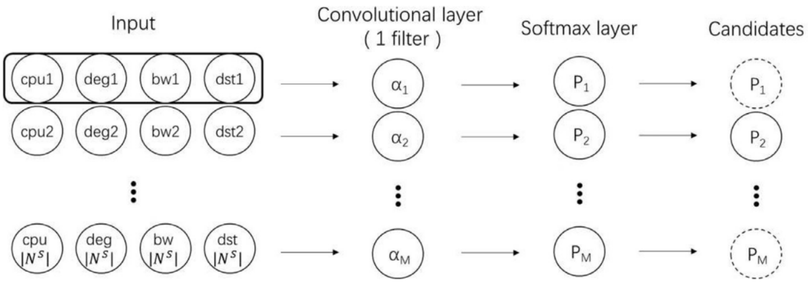


Figure: Policy network.

Training

- randomly initialize the parameters in the policy network
- cannot simply select the node with a maximal probability as the host
- exploration & exploitation
- sample from the set of available substrate nodes according to their probability
- select a node as the host

Training

- repeat this process until all the virtual nodes in a virtual request are assigned
- proceed to link mapping
- breadth-first search to find the shortest paths between each pair of nodes
- If no substrate node is available, the mapping fails
- in reinforcement learning, agent relies on reward signals to know if it is working properly

Training

- If we choose the ith node → vector y filled with zeros except the i th position which is one
- Cross-entropy loss → $L(y, P) = -\sum_i y_i log(P_i)$
- use backpropagation to compute the gradients of parameters
- stack the gradients g_f
- $g = \alpha.r.g_f$

Testing

- greedy strategy

Reward