Bachelor Thesis: Attention! - Learning Link State Routing Protocols

PATRICK KALMBACH, Technical University of Munich

Topic description bachelor thesis.

1 SCENARIO

The thesis is part of a larger project with the goal, to use deep learning to learn a routing protocol in spirit of link-state routing protocols\(^1\). This protocol should be tailored to a specific network and learn specific forwarding policies through supervised-learning.

The learned protocol has two goals:

- Learn the link state advertisements, called Hidden Link State Advertisements (HLSAs)\(^2\)
- Learn for each router, which routers in the network actually need its HLSA to make forwarding decisions.

To achieve those goals, the used neural-network architecture makes heavy use of (self-) attention, multi-head attention [3] and graph-attention [4].

The remainder of this document is as follows: Sec. 2 introduces the representation of nodes in the graph. Sec. 3 describes the learning tasks. Sec. 4 introduces the different neural network architectures that should be implemented and evaluated during the thesis. Sec. 5 introduces the concrete research questions that should be answered in your thesis.

2 GRAPH REPRESENTATION

Each node in the graph has a representation in a binary space that encodes the location of the node in the graph. A natural representation that comes to mind are IP-addresses. Using IP-addresses, nodes can be represented with a vector in \(\{0, 1\}^{32}\) or \(\{0, 1\}^{128}\) in case

\(^1\)Check out how link-state protocols work if you do not know this. Specifically, check high-level how OSPF and ISIS work.

\(^2\)In reference to the hidden state of neural networks, since the HLSAs are essentially hidden variables in the neural network architecture.

Author’s address: Patrick Kalmbach, Technical University of Munich, Patrick.Kalmbach@tum.de.
if IPv6. Depending on the domain, IP-addresses are very efficient in encoding the location of a node in the graph. An example is the IP-addressing scheme in fat-tree topologies [1]. In wide-area networks the addressing schemes are not well known, but are expected to also exhibit some form of pattern. Just because of how networks are operated and intra-domain routing algorithms such as OSPF, ISIS and MPLS work.

In addition to the representation through IP-addresses, we use binary encodings that are specifically learned to encode neighborhood information [2].

Encodings that contain information regarding the location of a node in the graph relative to other nodes is important to have a learnable pattern for the neural network. If no such pattern exists, then no correlation between destination address and forwarding decision can be learned.

One task of the thesis is to evaluate how well a neural network can learn forwarding decisions using random IP-addresses, learned encodings and designed IP-addresses.

3 LEARNING TASKS

Two different learning tasks exist: Learning shortest path based forwarding and learning network state dependent forwarding.

In case of shortest path based forwarding, the neural network should learn forwarding decisions for each node using the encoding of the destination node and the encoding of its neighbors to determine the output port. This is a supervised-learning task. The forwarding decisions are obtained by performing shortest paths through the network.

Network state based forwarding takes additionally the current state of the network into account. Here, HLSAs are important that inform a specific node about the state in distant parts of the network. Intuitively, depending on traffic patterns and network topology, only HLSAs of certain nodes might be required to make a forwarding decision. This learning task is also a supervised learning task. Necessary training data is generated with a flow- or packet-level simulator.

---

3 The method developed from Misra and Bhatia in [2] is already implemented.
4 Data generation will be my part. I will either directly provide the information to you or give you a ready script with which you can then generate the data yourself. Implementing the simulation is not part of the thesis.
4 NEURAL NETWORK ARCHITECTURES

This section introduces the different neural architectures that have to be implemented as part of the thesis.

4.1 Shortest Path Forwarding

Fig. 1 shows the basic neural structure to learn shortest path based forwarding. The architecture has two inputs: The encoding of the destination and the encoding of the neighbors of the current node. The encoding of the destination is passed as query into a Multi-head attention module. The encoding of the neighbors is passed as value and key into the Multi-head attention module. The output of the Multi-head attention module is passed into a sequence of fully connected layers that finally produce the output. The output layer represents the individual port.

Note that the encoding of the destination is a vector, while the encodings of the neighbors is a matrix. The result of the multi-head attention module will thus also be a single vector.
4.2 Network state based forwarding

Fig. 2 shows the neural network architecture that is used to make forwarding decisions based on the current state in the network. Fig. 3 shows the neural structure that is used to compute the Hidden Link State Advertisements (HLSA) that are used as input to the structure in Fig. 2. The two structures are combined in one neural network architecture. Thus, learning of HLSAs, localization based representations and how to combine them to produce a final output can be learned end-to-end.

The neural network that makes the forwarding decision has two path-ways with two inputs each. The first pathway is the same one as the structure in Fig. 1. This structure should provide localization based information, i.e., generate some representation of direction in the graph. The second pathway takes as input HLSAs of all nodes in the network and flow-level information of the current flow. This information can be the embedding of source, destination and potential other information that are available. The flow information is passed as query into a multi-head attention module. The HLSAs are passed as
key and value into the multi-head attention module. Again, flow info is a vector, so the output of the multi-head attention module is again a vector.

The outputs of both multi-head attention modules are passed through a sequence of dense layers. The output of those are concatenated and passed through a final sequence of fully connected layers that produce the forwarding decision.

The HLSAs are computed with the structure in Fig. 3. On each node a self-attention mechanism is used within a multi-head attention module. The self-attention mechanism takes as input a matrix of vectors. Each vector represents one incident link to the node. The vector contains the encoding of the neighbor and additional link level information such as utilization, queue size, availability, etc. The output of the multi-head attention module is then passed through a sequence of fully connected layers. The output is pooled, which produces the final HLSA of that node.

5 RESEARCH QUESTION

Your thesis should investigate the following research questions:
• What is the impact of node encodings on the ability of the neural network to learn shortest path based forwarding rules?
• Is inductive learning possible?  
• How do the attention weights look like for the attention mechanism that combines HLSAs with flow level information? Are the weights sparse or can sparse weights be learned?
• What is the required size of the HLSAs to learn stateful forwarding decisions?
• What is the impact of traffic and topology on the ability of the neural network to learn forwarding decisions and how does this reflect in the attention weights?

6 SCOPE OF THE THESIS

The scope of the thesis is the implementation of the neural network architectures, hyperparameter search and the evaluation of the research questions in Sec. 5. The neural network architecture must be implemented in PyTorch with Python using docker containers. That is, the PyTorch code runs inside a docker container that holds all required dependencies. Necessary evaluations should also be done using Python.

The thesis does not include performing and implementing flow- or packet level simulations. Providing the required data is the task of the supervisor and not the student. Similarly, it is the task of the supervisor to provide the learned node encodings to the student. Depending on the difficulty to obtain this data, the supervisor might provide a framework that can generate the required information with minimal interactions from the side of the student (e.g., like providing a input topology).

REFERENCES


5Can nodes that have not been included in the training process be included after the network has been trained, i.e., can the architecture make forwarding decisions for nodes that were not present during training
Bachelor Thesis: Attention! - Learning Link State Routing Protocols
