Paper Review: The Deep Learning Vision for Heterogeneous Network Traffic Control: Proposal, Challenges, and Future Perspective

Nei Kato, Zubair Md. Fadlullah, Bomin Mao, Fengxiao Tang,
Osamu Akashi, Takeru Inoue, and Kimihiro Mizutani, Tohoku University / NTT
Accepted for Publication IEEE Wireless Communications, 2016

Eric Sturzinger

10 MAR 2017

UCDAVIS
UNIVERSITY OF CALIFORNIA

Outline

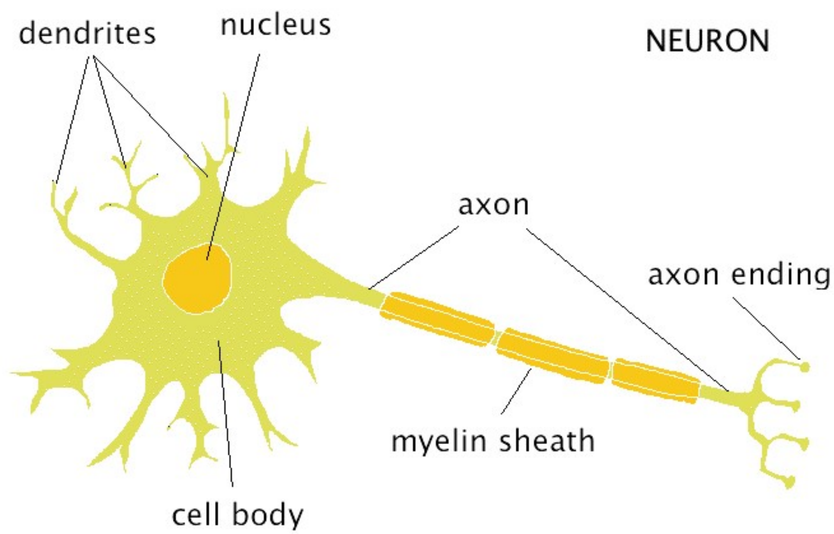
- Neural network components/description
- Mathematical example/back propagation
- Application to network problems (paper review)
- 3-phase approach
- Topology example/results

Basic Components

Nucleus – node
Axon – output link
Dendrites – input link

Neural network consists of 3 main components:

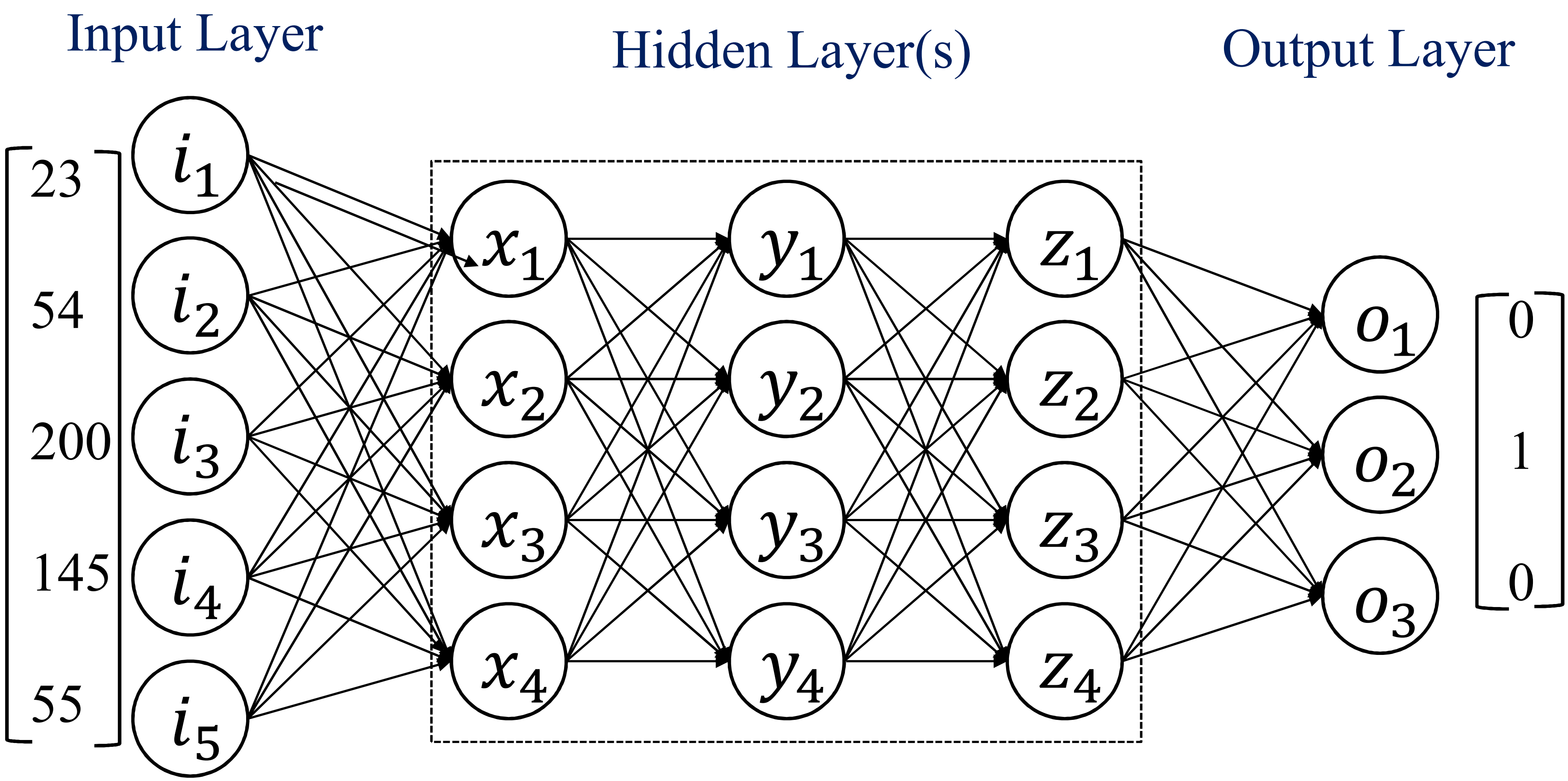
- Connections between neurons of different layers (topology)
- Link weight of each connection (continuously modified)
- Activation function at nucleus (input to output mapping)



Source: http://webspace.ship.edu/cgboer/theneuron.html

UCDAVIS
UNIVERSITY OF CALIFORNIA

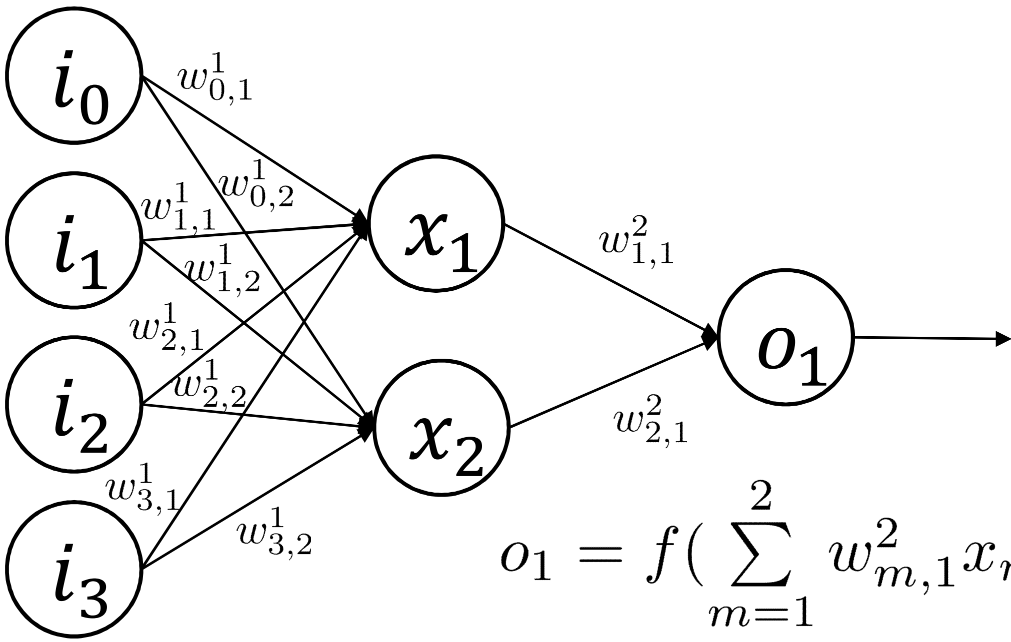
Network Layers

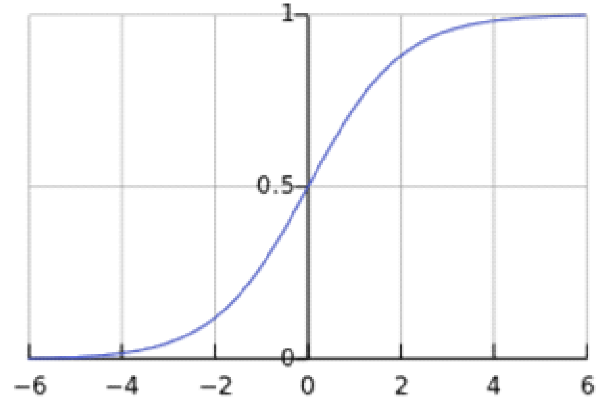


UC DAVIS
UNIVERSITY OF CALIFORNIA

Mathematical Example

Layer 1 Layer 2 Layer 3





$$f(x) = \frac{1}{1+e^{-x}}$$

Sigmoid function

$$o_1 = f(\sum_{m=1}^{2} w^2_{m,1} x_m)$$

Activation function: sigmoid, tanh, etc.

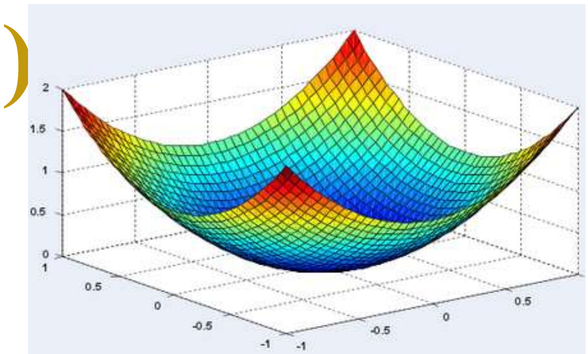
$$x_1 = \sum_{m=0}^{3} w^1_{m,1} i_m$$

Forward Propagation →

UCDAVIS
UNIVERSITY OF CALIFORNIA

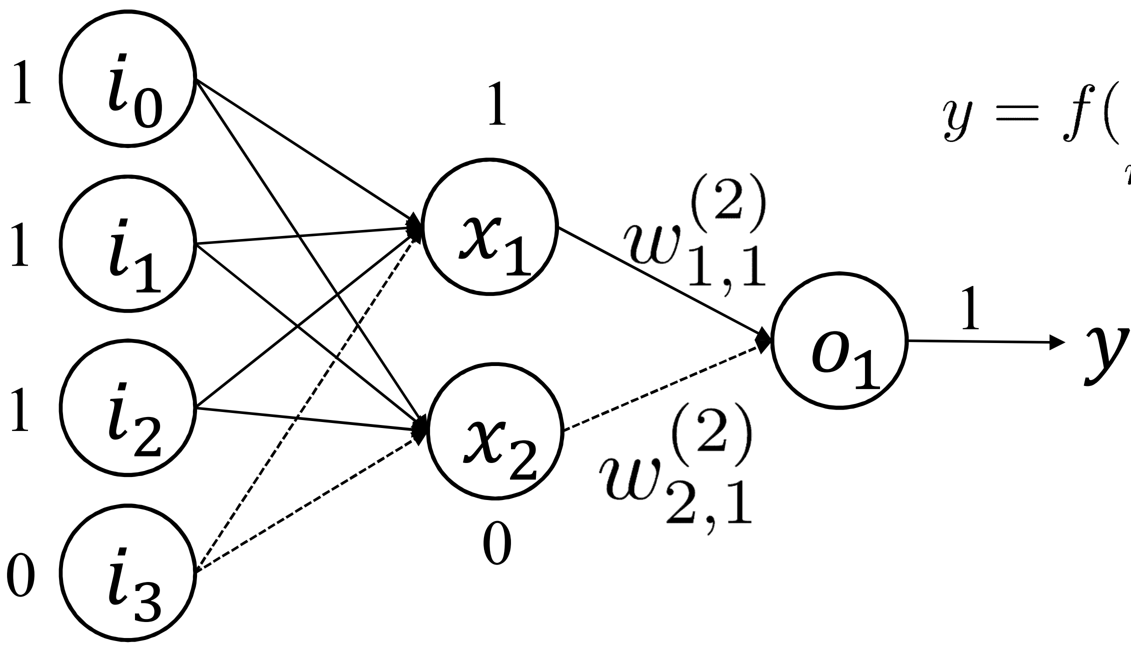
Mathematical Example (cont.)

Network can act as a series of logic gates:
binary or continuous: [0,1]



Quora

Layer 1 Layer 2 Layer 3



$$y = f\Big(\sum_{m=1}^{2} w^{(2)}_{m,1} f\Big(\sum_{m=0}^{3} w^{(1)}_{n,m} i_n\Big)\Big)$$

$$E = \frac{1}{2}\sum_{m=1}^{M}(t-y)^2$$

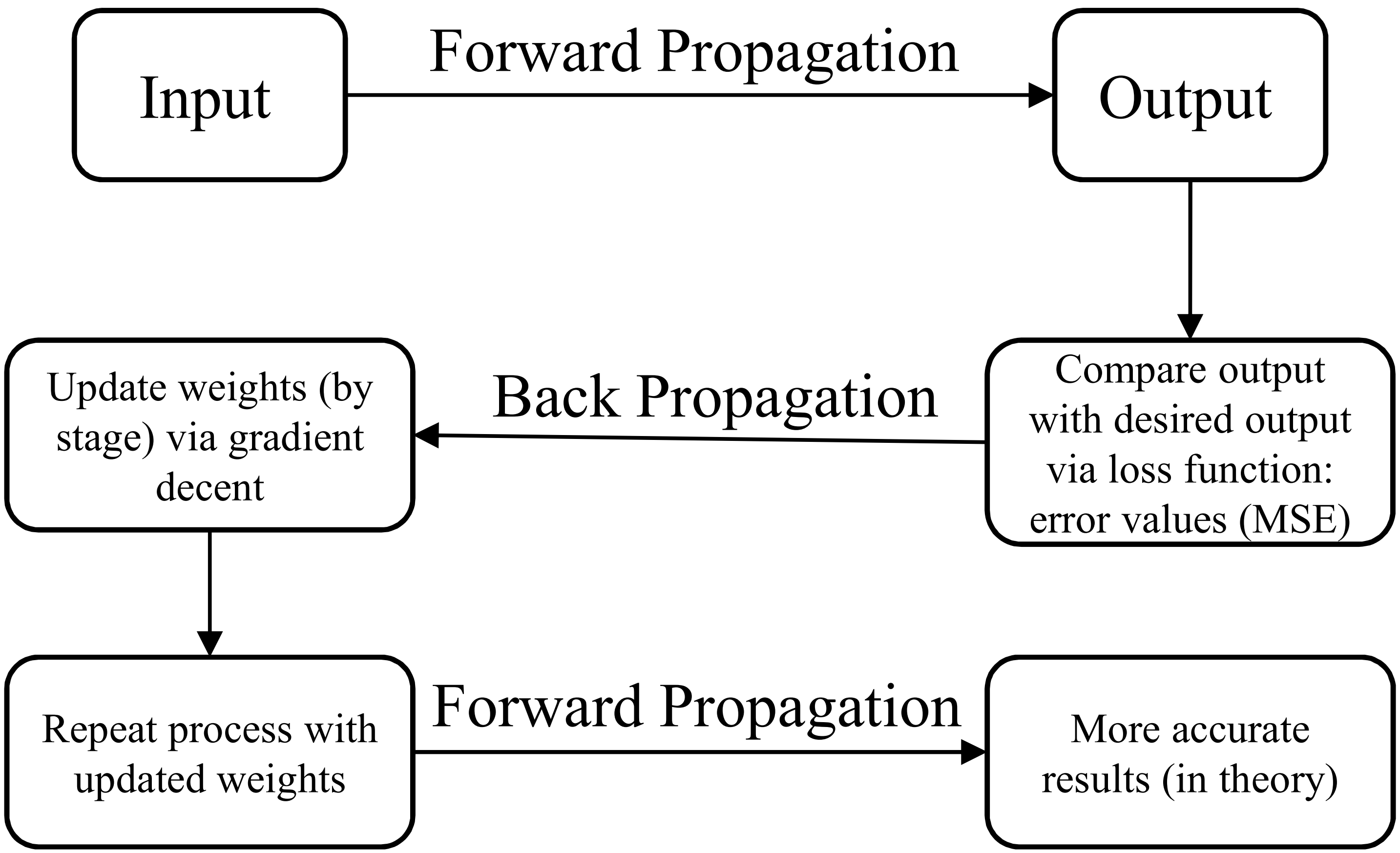
$$\frac{\partial E}{\partial w^2_{1,1}} = \frac{\partial E}{y}\frac{\partial y}{o_1}\frac{o_1}{w^2_{1,1}}$$

$$\frac{\partial y}{o_1} = f(x)(1-f(x))$$

$$w^{(2)}_{1,1} := w^{(2)}_{1,1} + \alpha\frac{\partial E}{\partial w^2_{1,1}}$$

$$\frac{\partial o_1}{w^2_{1,1}} = x_1$$

Back Propagation



UCDAVIS
UNIVERSITY OF CALIFORNIA

Application to Network Problem

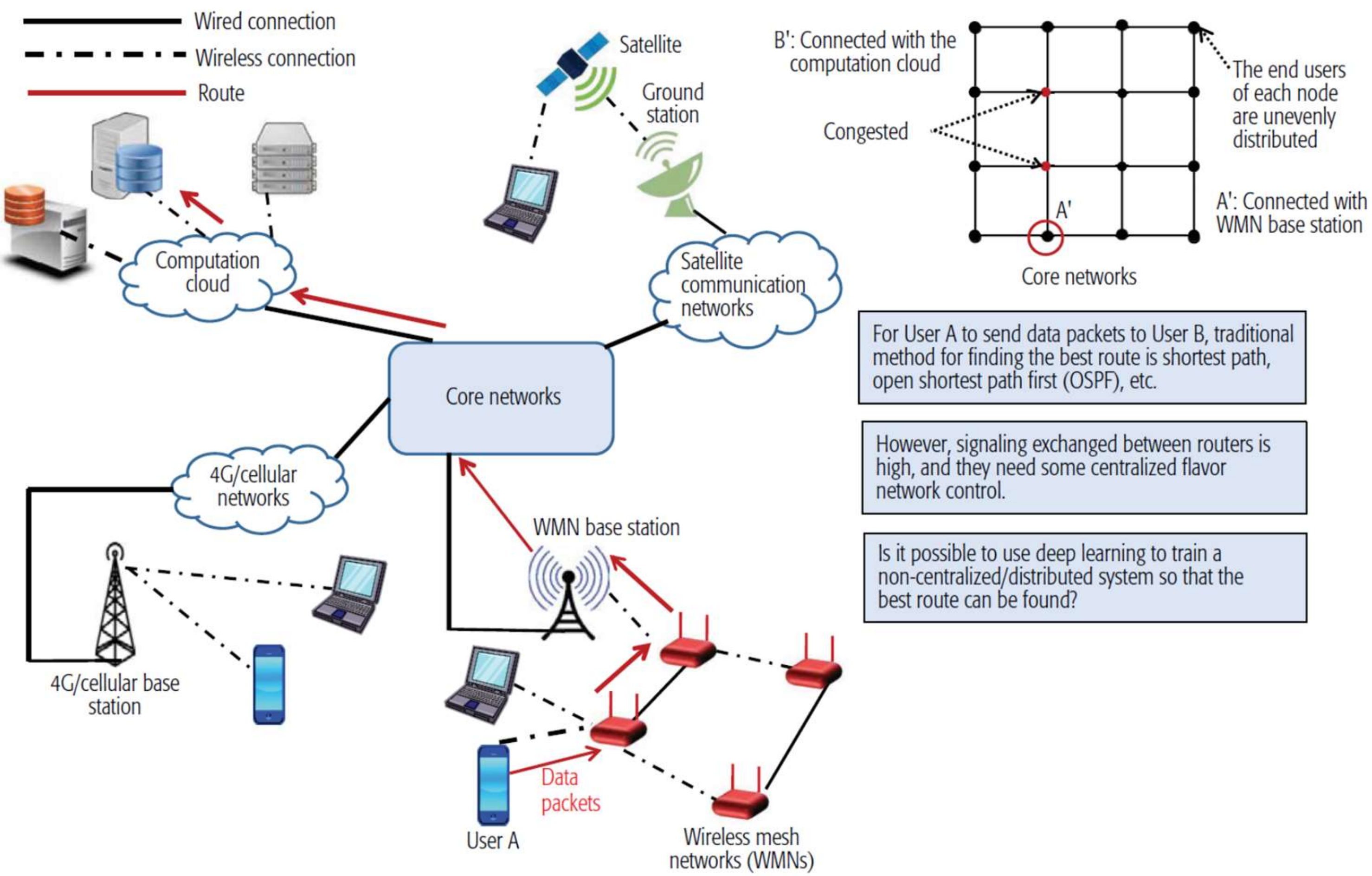
- Problem Statement: Apply a deep neural network model to optimize highly dynamic traffic flow via routing solutions in heterogeneous networks (wired/wireless).

- How should input and output layers be characterized?

    Offered traffic at each node, current average system delay on each link from congestion, link quality in unstable (mobile, WSN) links; output: path, also apply to processing/storage node?

- Algorithm time interval: how often should solutions change?

Heterogeneous Networks



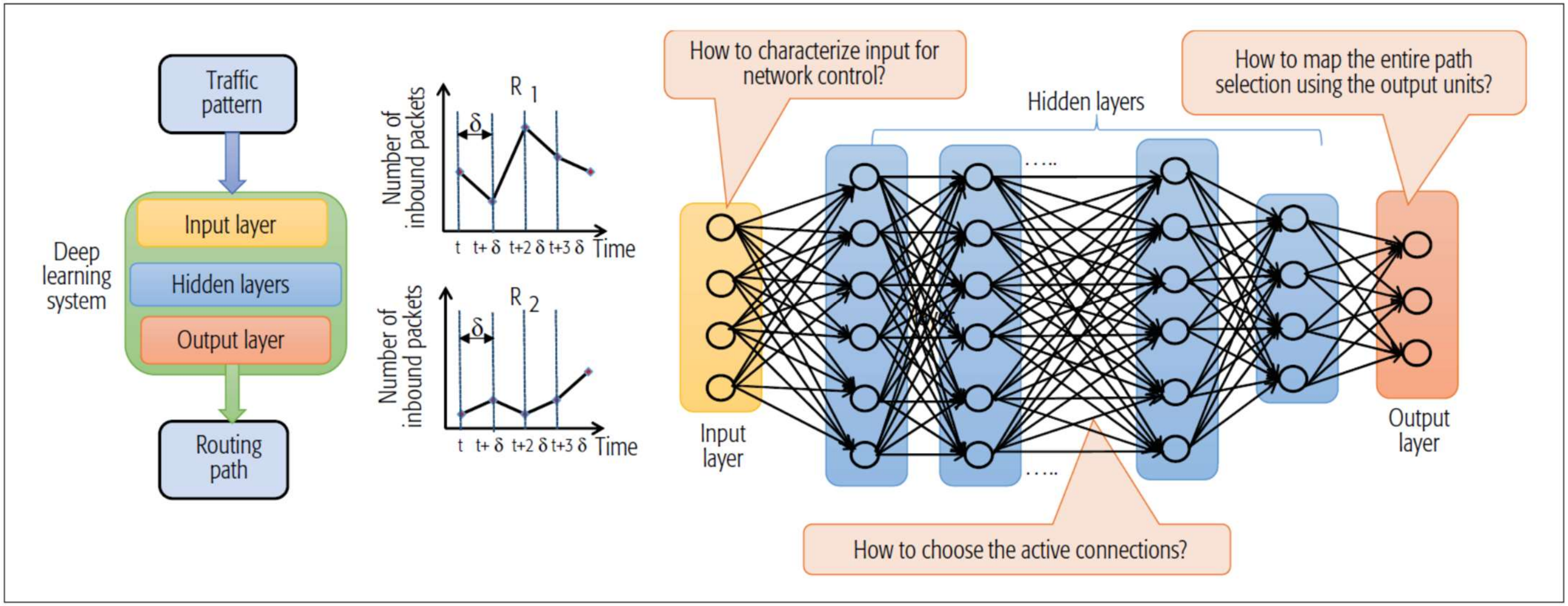
For User A to send data packets to User B, traditional method for finding the best route is shortest path, open shortest path first (OSPF), etc.

However, signaling exchanged between routers is high, and they need some centralized flavor network control.

Is it possible to use deep learning to train a non-centralized/distributed system so that the best route can be found?

UCDAVIS
UNIVERSITY OF CALIFORNIA

General Process

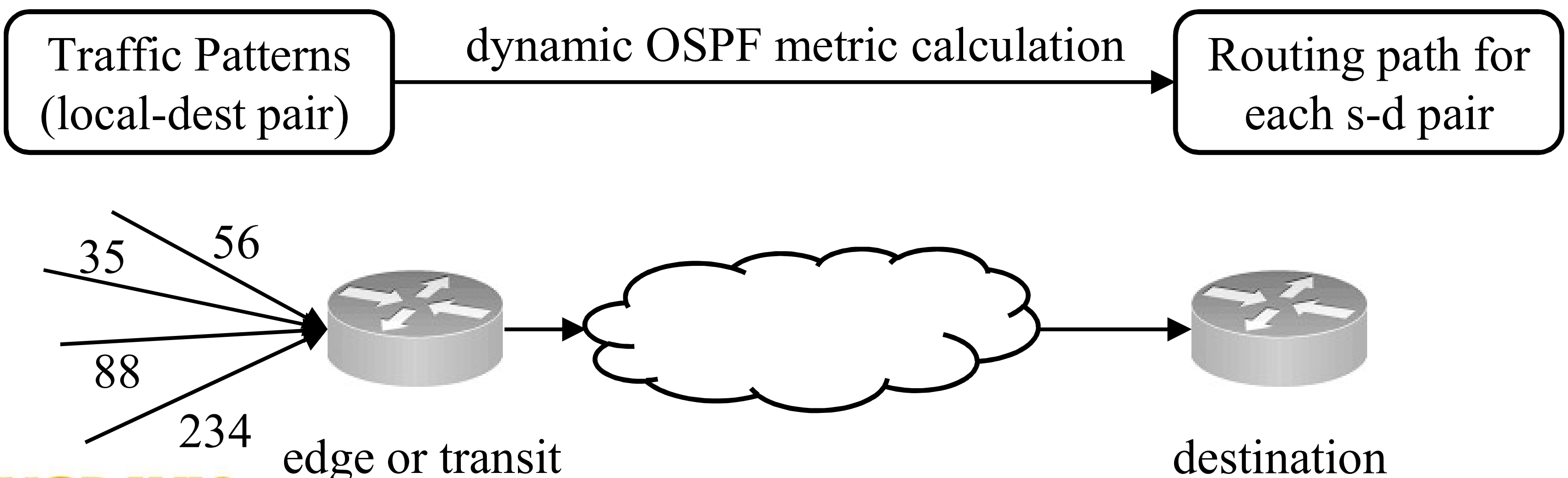


Traffic pattern: (# packets) of previous time interval at local node

UCDAVIS
UNIVERSITY OF CALIFORNIA

Phases - Initial

- Simulate OSPF operation and map various inputs (traffic patterns) with associated routing path calculations (outputs).
- - Practicality? Routes do not change dynamically
- OSPFv3: Dynamic Interface Cost Support: Max data rate, current data rate, resources, latency, and relative link quality
- Data set produced:

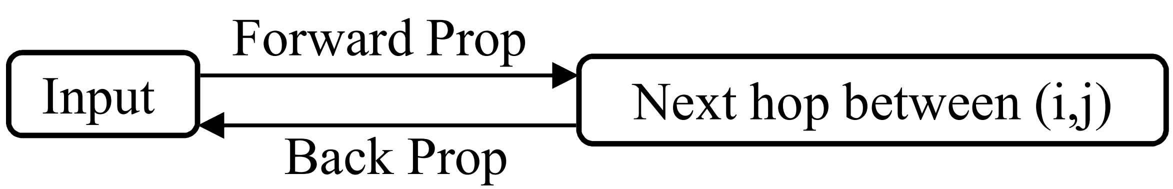


UCDAVIS
UNIVERSITY OF CALIFORNIA

Phases (cont.) - Training

- **Training**: For each edge or transit router, i, and destination j:
  Input = traffic pattern for last 3 time intervals, (N x δ)
  $DL_{i,j}$: (Each edge router contains N - T - 1 DLs, transit: N - T

$$TP = \begin{bmatrix} T_{1,1} & T_{1,2} & T_{\delta} \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ T_{N,1} & T_{N,2} & T_{N,\delta} \end{bmatrix}$$



- Generates weight matrix: $WM_{i,j}$
- Initially 4 hidden layers, increased until performance decreases, due to overfitting

UCDAVIS
UNIVERSITY OF CALIFORNIA

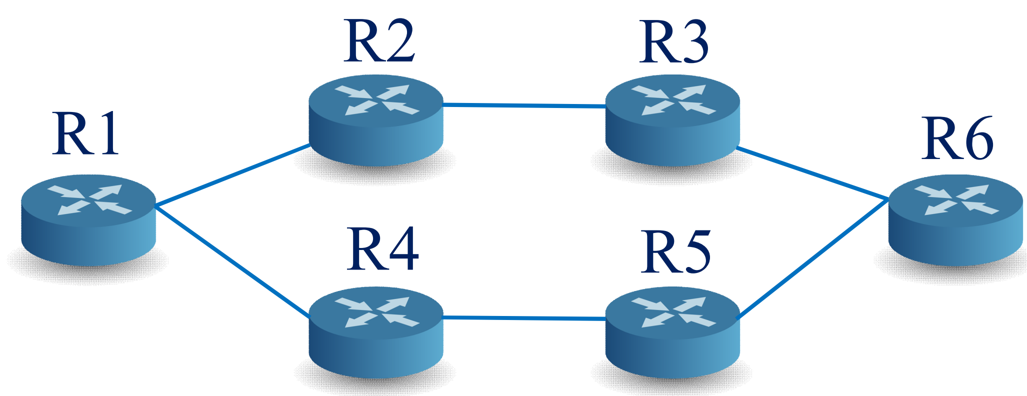
Phases (cont.) - Running

- Each edge router must execute all DL systems to generate a complete path: all routers send respective traffic patterns and WMs to all edge routers via multicast at every time interval

- It then runs each DL system (every possible node and destination) to determine next hop combining to generate complete path

$$DL_{1,6} = 2$$
$$DL_{2,6} = 3$$
$$DL_{3,6} = 6$$

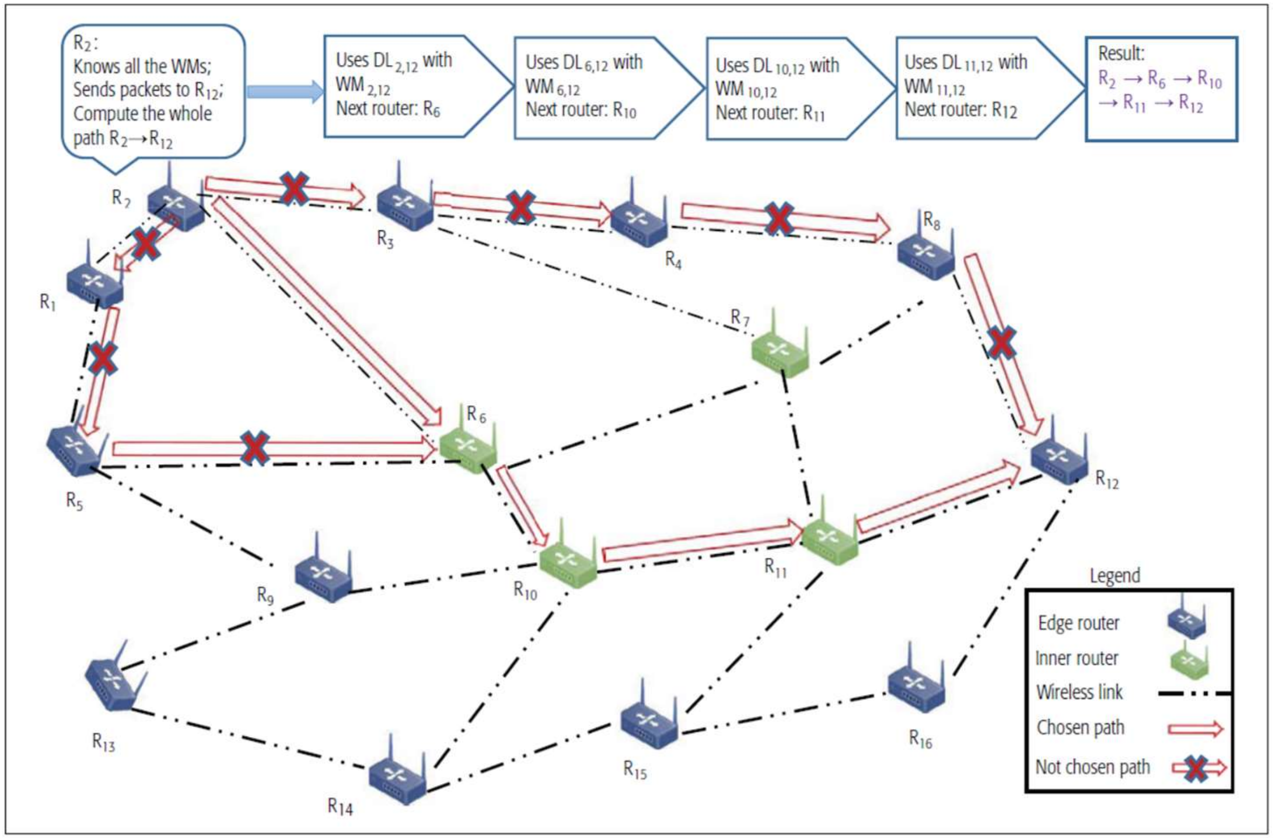
Output = [2,3,6] = $\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$



UCDAVIS
UNIVERSITY OF CALIFORNIA

Topology

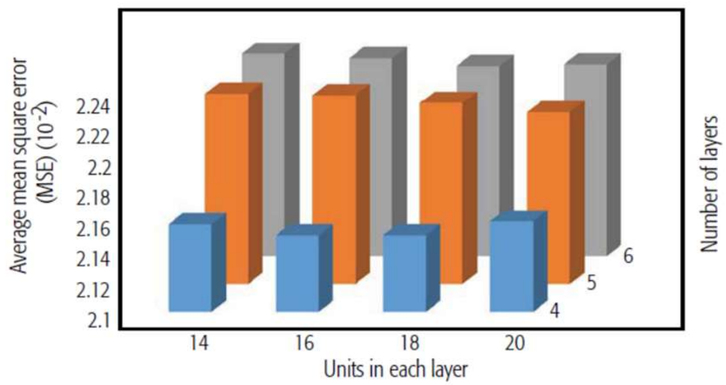
- N total nodes
- Edge: traffic sources and destinations - (N – T)
- Transit: only forward traffic (T) - green
- Each edge node has N – T – 1 Deep Learning systems
- Each transit node has N DL systems (per destination)

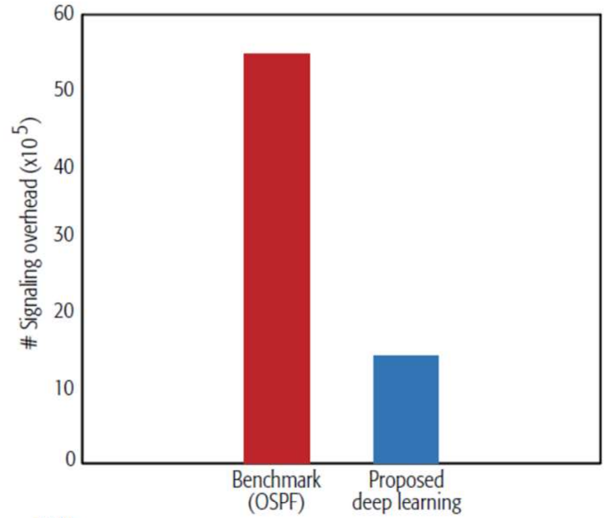


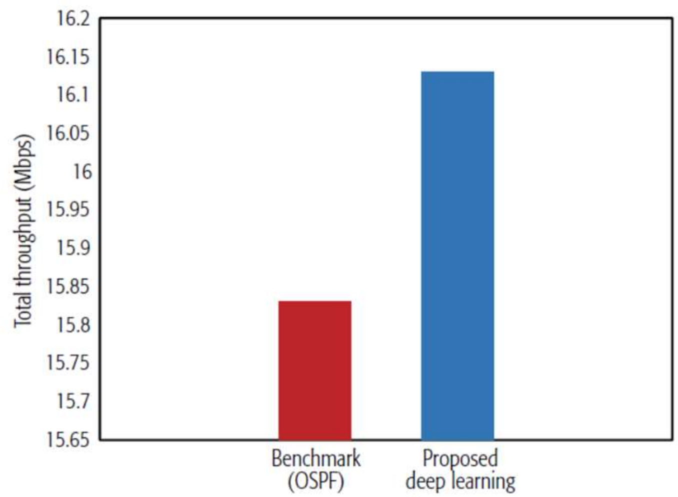
UCDAVIS
UNIVERSITY OF CALIFORNIA

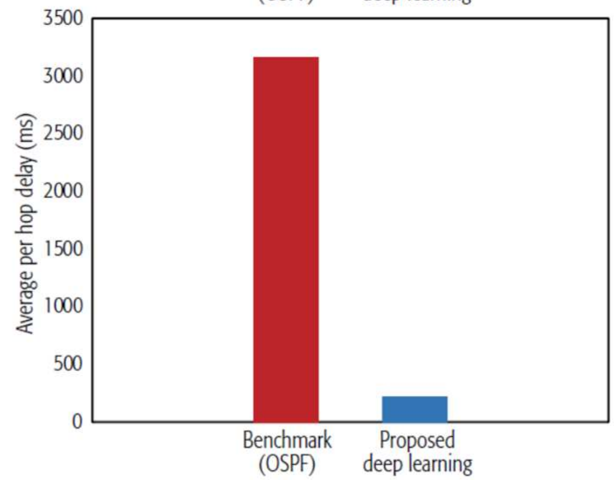
Results

- 4 hidden layers is ideal, 5 and 6 increase MSE
- Units in each layer also affects MSE
- Slightly increased throughput (less packet loss)
- Less signaling overhead than OSPF
- Less per hop delay (unclear)









UCDAVIS
UNIVERSITY OF CALIFORNIA

Analysis & Conclusion

- Neural networks can be used to model and optimize complex systems with many non-linear effects
- This problem contains aspects of both classification and regression methods: input traffic changes in a relatively continuous manner and output is a binary solution (next hop)
- Potentially applicable to networks with unreliable links (WSNs) or heterogeneous networks (as in this example) spanning WSN, 4G/5G, and satellite
- Need better method for choosing target solution to train against