AI in Networking

Team 12

IIIT Hyderabad

November 25, 2020

Outline

Introduction to AI in Networking
Supervised Learning-Channel Learning, Path Loss prediction.
Unsupervised Learning- K-means (relay node selection),
Hierarchical clustering.
Reinforcement Learning- Routing optimization.
Hopfield Neural Networks for Routing Optimization.

Key factors for successful AI deployment

AI centric networks will integrate intelligent functions across the wireless infrastructure, cloud, and end-user devices with the lower-layer learning agents targeting local optimization functions while higher-level cognitive agents pursuing global objectives and system-wide awareness. We will talk about various kinds of network intelligence. We will also highlight the important factors required for successful AI deployment in future networks.

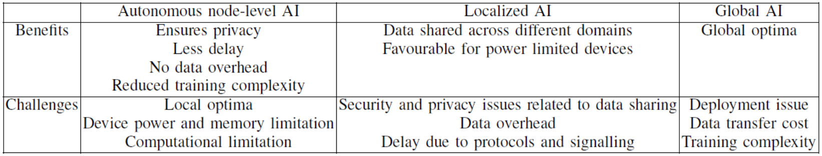
Distribution of Network Intelligence

Autonomous Node Level AI
It is used to solve self-contained problems at individual network components or devices where no data is required to be passed through the network. For eg: at switches and routers in a network

Localized AI
Localized AI is where AI is applied to one network domain. Localized AI requires data to be passed in the network, however, is constrained to a single network domain.

Global AI
Global AI is where a centralized entity requires knowledge of the whole network and needs to collect data and knowledge from different network domains

Challenges and Benefits of Network Intelligence



	Autonomous node-level AI	Localized AI	Global AI
Benefits	Ensures privacy Less delay No data overhead Reduced training complexity	Data shared across different domains Favourable for power limited devices	Global optima
Challenges	Local optima Device power and memory limitation Computational limitation	Security and privacy issues related to data sharing Data overhead Delay due to protocols and signalling	Deployment issue Data transfer cost Training complexity

An important aspect in design is to coordinate such intelligence across different network domains in order to optimize the end-to-end network performance.

Data acquisition

Acquiring and labelling data is fundamental. Data should be tailor made so as to be represented accurately in feature space. We need to use appropriate transforms on data so as to train our Neural Nets. The success of integrating AI in future wireless networks will not only depend on the capability of the technology but also on the security provided to the data and models. It is crucial to guarantee obtaining accurate data sets and AI models by avoiding data from false base stations or compromised network devices.

Efficient Training process

For AI based techniques in wireless networks to be efficient, it is crucial to design algorithms with an accurate training process. Human knowledge and theoretical reasoning are important for limiting the space that ML solutions need to explore thus improving performance and speeding up the training process. AI agents should interact with the user so as to take into consideration the users goals and intentions during the learning phase.

Now we will discuss the applications of AI in networking based on various learning models

Supervised Learning

Supervised Learning: Each training example has to be fed along with their respective label. We go through some applications of Supervised Learning in networks.

Supervised Learning - Channel Learning

Channel state information (CSI) refers to known channel properties of a communication link. This information describes how a signal propagates from the transmitter to the receiver and represents the combined effect of properties of a channel.

The most prevalent method to obtain CSI is "pilot aided training", in which a known sequence of signals is sent between the transmitter and the receiver so that the channel responses can be estimated. However this method is quite energy-inefficient.

This issue can be tackled by introducing a new method known as **Channel Learning**. This method uses Supervised learning algorithms to infer unobservable CSI from the observable one.

Channel Learning Framework

Assume a user is equipped with a single-antenna device to communicate with Base-station(BS) antennas in set A. However, only the channel to the antennas which is in the observable subset $O \subset A$ can be obtained, whereas the channel to the rest of the antennas, which constitutes the unobservable subset $U \subset A$, is unknown to either the user or the network. We denote the channel to the observable antennas as h_o and the channel to the unobservable antennas as h_u.

Channel Learning Framework

The user wants to know some metrics of the unobservable channel according to the metric function $\mathbf{m} = m(h_u)$. We can formulate channel learning as the task to approximate a function $f(.)$ from the observable channel to the unobservable metrics($\mathbf{m}$). **We can use supervised learning algorithms to approximate $f(.)$.**

In the training phase, the unobservable metric($\mathbf{m}$) can be easily calculated from the unobservable channel samples, which can be collected through traditional pilot-aided channel estimation. Once training is finished, only the observable channel responses are needed to predict the unobservable metric($\mathbf{m}$).

Applications of Channel learning - Cell Selection problem

We will now briefly talk about an application of Channel learning framework.

The network needs to know the user's channel to a large number of small cells in order to select the best small cell. This usually requires the network to scan for potential small cells, which is extremely energy and time-consuming.

How can Channel Learning be used here?

Channel learning framework can be applied here to solve these problems. A possible configuration is to place large antenna arrays at large cells and use them as the "observable antennas" to infer the channel to the overlapping small cells(unobservable antennas). The users intending to communication send sounding signals to the observable array at the large cell. Based on the array response h_o, the network infers the best small cell for each user. Here the unobservable channel h_u is the channel between the user and small cells.

A Neural Net based algorithm

Extracting Angular domain information

The first step of our algorithm is to extract angular-domain information from the raw observable channel response h_o. We do this because it is hard for machine learning algorithms to automatically decode the complex relationship between spatial-domain channel response and user locations.

Forming Feature space

The second step is to translate the information from the previous step into a feature space. NN cannot handle input features that have large dynamic range. However, the dynamic range of the channel responses may span several orders of magnitude since users may lie in a wide geographical area. To tackle the above problem we take the logarithm of the angular-domain magnitude and then quantise it.

Output encoding
We employ the probabilistic output coding scheme.

Cost Function and Training
As we have used probabilistic coding scheme for output, we can use cross-entropy between the NN outputs and the desired training outputs as the loss function which needs to be minimised

Prediction
After the NN has been trained, we can use it to select the best small cell. We take the array response from a user and pre process the array response as in steps 1 and 2 to form the quantized input feature. Then we feed the input feature into the trained NN. The NN will output a selection probability for each of the output vector entry. We select the small cell with the largest probability.

Pathloss Prediction

Pathloss Prediction using Support Vector Regression: Network designers rely on signal propagation path loss models to ensure quality of service in wireless networks. To provide adaptability, the use of machine learning techniques has been considered to predict characteristics of the wireless channel.

Pathloss Prediction using Support Vector Regression

Support Vector Regression:

- In simple regression we try to minimise the error rate. While in SVR we try to fit the error within a certain threshold.
- We can also use Neural Networks for this pathloss prediction considering it as a regression model. But here we are doing the regression using Support vectors which has the benefit of lower training time compared to ANNs.

Pathloss Prediction using Support Vector Regression

Let the predicted hyperplane be

$$h(\mathbf{x}_i, \mathbf{w}) = \left\langle \mathbf{w}^T, \mathbf{x}_i \right\rangle + b$$

Error

$$e_i = y_i - h(\mathbf{x}_i, \mathbf{w}).$$

Vapnik's Loss function:

$$E(e_i) = |e_i|_\varepsilon = \begin{cases} 0, \text{ if } |e_i| \leq \varepsilon \\ |e_i| - \varepsilon, \text{ otherwise} \end{cases}$$

Thus, the loss is equal to zero if the difference between the predicted $h(\mathbf{x}_i, \mathbf{w})$ and the measured value y_i is less than ε

Pathloss Prediction using Support Vector Regression

Optimization problem:

$$\min_{\mathbf{w},b}\left[\frac{1}{2}\|\mathbf{w}\|^2 + C\left(\sum_{i=1}^{\ell} E\left(e_i\right)\right)\right]$$

We can introduce slack variables and relax the optimization problem.
In case of non-linear regression we can map the vectors to higher dimensions and make it a linear regression problem using popular kernels.

Pathloss Prediction using Support Vector Regression

Various field data of each measured point were collected to compose the feature vector of the SVR process. Such field data were antenna-separation distance, terrain elevation, horizontal angle, vertical angle, latitude, longitude, horizontal and vertical attenuation of the antenna,the theoretical path loss of the Okumura-Hata model was also used as an input of the SVR training algorithm.
K-fold cross validation can be used for tuning parameters.

Unsupervised Learning

Unsupervised Learning:The training data is unlabeled, and the system attempts to learn without any guidance. This technique is particularly useful when we want to detect groups of similar characteristics.

Anomaly detection

Anomaly detection in cellular networks:Anomaly detection for large scale cellular networks can be used by network operators to optimize network performance and enhance mobile user experience. Network anomaly detection methods in the literature can be classified into 3 main categories: statistics-based, classification-based,and clustering-based

Big Data driven Anomaly detection

- Statistics-based anomaly detection : Threshold based methods.(Needs prior knowledge)
- Classification based detection : Use of Supervised learning methods such as SVM,Decision Trees (Needs Training data)
- Clustering based : Hierarchical Clustering ,K-means.

Multi-hop Wireless Broadcast Relay Selection in Urban Vehicular Networks: Many services rely on the performance of broadcast communication to disseminate data packets. For example, In an urban vehicular network, an efficient way of broadcasting data packets is choosing some vehicles as relay nodes. The base station only needs to multicast the data packets to these relay vehicles and the packets would spread to the whole network through D2D(Device to Device) communication among vehicles. In this situation, the strategy of relay selection becomes a key factor of the broadcast efficiency.

Selection of Relay Nodes

K-means for Relay nodes selection: 1 Select n vehicles as a relay set randomly.
2 Initialize relay clusters with those n vehicles.
3 Partition vehicles to their closest relay cluster.
4 Get the mean values of the objects' location in each cluster.
5 Find the closest vehicles to these values and update the relay set.
6 Repeat step 3 to step 5 until it gets a stable relay set.
7 Multicast through D2D communication

Reinforcement Learning

1. The idea of routing is to create a routing table which is pre-computed using statistics(shortest path algorithms).
2. However, the network parameters and connections dynamically change over time.
3. So, we would like to have an algorithm where the routing tables are dynamically updated and optimized for the real-time scenario.
4. The update process by experimenting and updating.

Problem Formulation:

1. The main problem to solve here is the node to which the packet must be routed among the adjacent nodes.
2. Let us say that the packet(P) is at node x and the destination node is d.
3. Let us say that we were to estimate the delay of packet going from x to d through node X's neighbor Y. Let the delay be denoted by $Q_x(d, z)$

Problem Formulation: Continued

1. The minimum time taken by the packet to reach d through y would be min $Q_y(d, z)$, where z represents all the neighbor nodes of y.
$$t = minQ_y(d, z) \tag{1}$$
2. The new time taken by packet to move from x to d through y is equal to $q + s + t$, where s is the transmission time from x to y and q is queuing delay at x.
3. we now update the expected delay time and update the minimum delay and routing table.

Algorithm: Update Equation for Q-routing Algorithm

We would like to update $Q_x(y, d)$. The update should be proportional to difference between new and old delays.

$$\Delta Q_x(d, y) = \eta(q + s + t - Q_x(d, y)) \tag{2}$$

This is similar to Q-learning algorithm popular in reinforcement learning literature. The algorithm is also similar to Bellman-Ford algorithm.

Results

The algorithm is shown to sustain higher loads relatively better than the static pre-computed routes. The figure below shows the simulation of the networks for low and high loads.

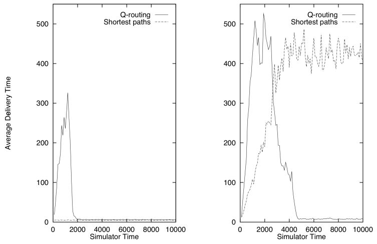


Figure 1: Performance on Low and High Loads

Observations

We can see that at lower load levels the pre-computed paths perform well because queuing and transmission delays are very low. But with increasing load, queuing delay increases which is not taken into account by the pre-computed statistic.

Advantages

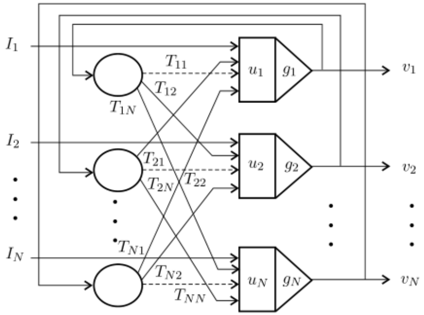
1. Topological changes like new connections or disconnected links are shown to be effecting the proposed algorithm less. The algorithm quickly adapted to the changes.
2. Periodic changes in network traffic was also quickly adapted to by the Q-routing algorithm.

Design of Neural Network for Routing in Communication Networks :
The optimal routing policy can be modelled as a shortest path problem where instead of the distance between routers some cost metric can be used describing the transmission conditions such as the link capacity, traffic flow, transit time, etc

Hopfield Neural Networks

One type of ANN that used on solving Shortest Path problem is Hopfield neural network (HNN). Firstly, HNN is introduced by Hopfield and Tank for solving Traveling Salesman Problem (TSP). HNN is a recurrent neural network (fully connected) which has a energy function which the system tries to decrease and reach a stable state.

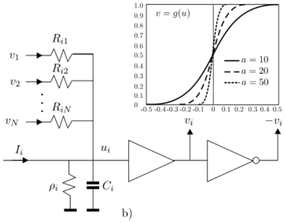
HNN



The activation function is the sigmoid function g(u_i)

$$g_i\left(u_i\right) = \frac{1}{1+e^{-a_i \cdot u_i}}$$

HNN



State equations of the circuit is described by

$$C_i \frac{du_i}{dt} = \sum_{\substack{j=1 \\ j \neq i}}^{N} T_{ij} v_j - \frac{u_i}{R_i} + I_i, \quad i = 1, 2, \ldots, N$$

where Ri is an equivalent resistance connected to the cell's capacitor Ci.

HNN for Shortest Path

$$\begin{aligned} \mathrm{E} &= \frac{\mu_1}{2}\sum_{X}\sum_{\substack{i\neq X\\(X,0\neq(d,s)}} C_{Xi}v_{Xi} + \frac{\mu_2}{2}\sum_{X}\sum_{\substack{i\neq X\\(X,n\neq|d,s)}} \rho_{Xi}v_{Xi} + \frac{\mu_5}{2}(1-v_{ds}) \\ &+ \frac{\mu_3}{2}\sum_{X}\left(\sum_{i\neq X}v_{Xi} - \sum_{i\neq X}v_{iX}\right)^2 + \frac{\mu_4}{2}\sum_{i}\sum_{X\neq i}v_{Xi}(1-v_{Xi}) \end{aligned}$$

Coefficients C_{Xi} are the link costs from router X to router i and the terms ρ_{Xi} describe the connection between routers: the value is 1 if routers are not connected, and 0 for connected routers.

HNN for Shortest Path

$$\mathrm{E} = \frac{\mu_1}{2}\sum_{X}\sum_{\substack{i\neq X \\ (X,0\neq(d,s)}} C_{Xi}v_{Xi} + \frac{\mu_2}{2}\sum_{X}\sum_{\substack{i\neq X \\ (X,n\neq|d,s)}} \rho_{Xi}v_{Xi} + \frac{\mu_5}{2}(1-v_{ds})$$
$$+\frac{\mu_3}{2}\sum_{X}\left(\sum_{i\neq X} v_{Xi} - \sum_{i\neq X} v_{iX}\right)^2 + \frac{\mu_4}{2}\sum_{i}\sum_{X\neq i} v_{Xi}(1-v_{Xi})$$

In the energy function, the term μ_1 minimizes the total cost; μ_2 prevents nonexistent links from being included in the chosen path; μ_3 is zero for every router in the valid path ; μ_4 forces the state of the neural network to converge to one of the stable states. The state vi is close to 1 for router belonging to the valid path, otherwise the state is close to 0. The term μ_5 is zero when the output v_{ds} is equal to 1.

HNN for SP

From this energy function and HNN constraint equation we can get the External Bias I_i and the Coefficient matrix containing T_{ij}

Thank You