AI IN NETWORKING - FINAL REPORT

TEAM MEMBERS

V V SUSHEEL(20171164)
A S D HARSHA(20171136)
K ABHIRAM(20171011)

IIIT Hyderabad

$\mathscr{PL}$

1 Introduction

Evolution to the 5th generation cellular technology (5G) and beyond networks will see an increase in network complexity. This increased complexity calls for a fundamental change in the design and operation of communication networks. AI and its sub-categories like machine learning and deep learning have been evolving as a discipline, to the point that nowadays this mechanism allows fifth-generation (5G) wireless networks to be predictive and proactive, which is essential in making the 5G vision conceivable.

Let us understand this with a practical scenario. Say an MNC want to set up a company wide video conference for all their locations around the world. It's important that everyone can view this call with high-quality, low-latency video, and that they can send high-quality video into the call too, when it's time for the Q and A.

Often there are subtle problems that are difficult to detect or predict prior to the event, even in a testing scenario. So during the event, if any issues arises it will be impossible to identify and fix the problem in time. In fact, in such a scenario it is not possible to know how the event is going for all users, without them submitting real-time feedback. What if the users are not able to submit real-time feedback due to the issues in network?

This is where AI centric networks fills in. Future Networks must no longer be built to transport user data only but rather designed to support exchange of data and the ability to handle algorithms capable of real time decision making. Recent advances in AI promises to address many complex problems in Networking.

In this project we will provide an overview on integration of AI functionalities Communication Networks. We will start with a brief introduction about the key factors for successful AI deployment in future networks. We will then proceed to talk about the use cases of various AI learning techniques like Supervised,Unsupervised and Reinforcement learning in various network related problems. At the end we have also explained a Neural Network for Optimization of Routing in Communication Networks .

2 Key Factors for Successful AI deployment

2.1 Distribution of Network Intelligence

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 differentiate between autonomous node-level AI, localized AI, and global AI.

1. **Autonomous node-level AI**
   Autonomous node-level AI 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

2. **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. Localized AI can also refer to scenarios where data is geographically localized.

3. **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.

Each of the above has their own benefits and Challenges. The following table summarises the challenges and benefits with each of the above 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.

2.2 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.

2.3 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

3 Supervised Learning

In supervised learning, each training example has to be fed along with their respective label. The notion is that training a learning model on a sample of the problem instances with known optima, and then use the model to recognize optimal solutions to new instances. A typical task on supervised learning is to predict a target numeric value, given a set of features, called predictors. This description of the task is called regression. We now look into few applications of this supervised learning in communication networks.

3.1 Channel Learning

Wireless communication networks rely heavily on channel state information (CSI) to make informed decision for signal processing and network operations. 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 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.

3.1.1 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.

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($\mathbf{m}$).

3.1.2 Application of Channel Learning Framework - Cell Selection in Multi-tier Networks

We will now briefly talk about an application of Channel learning framework. We will define a few terms which will be used extensively in this example

1. **Small cells**
   In context of communications, Small cells are the low cost cells that can be deployed in residential areas and can be connected to core network via broadband. Small cells are increasingly being deployed in 5G networks to cope with the high traffic demands.

2. **Channel Sounding** Channel Sounding is a technique that evaluates the various characteristics of an RF Channel that is being used for wireless communication.

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.

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.

3.1.3 A Neural Network based algorithm

We now briefly talk about a NN based algorithm for approximating $f(.)$ for the cell selection problem. We first obtain a training set which contains some observations of h_o and the corresponding $\mathbf{m}$. Then we use these training data to choose a candidate function that best approximates function $f(.)$. We will now list down various steps of the algorithm.

1. **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.

2. **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, as most of the Neural networks use sigmoid as the activation function which saturates at extremely large or small input values. 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.

3. **Output Encoding**
   We employ the probabilistic output coding scheme. Eg: Suppose we need to choose from K small cells, then the output is encoded as a K length vector, with each entry stands for the probability that the corresponding small cell is the best cell.

4. **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. This is because, Cross-entropy is a measure of the difference between two probability distributions for a given random variable. We can use various optimisation algorithms to minimise the cost function and to approximate $f(.)$.

5. **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.

3.2 Path loss Prediction Using Support Vector Regression

To ensure an acceptable level of quality of service for users in a wireless data network, network designers rely on signal propagation path loss models. Radio wave propagation models are a series of mathematical calculation developed to predict path characteristics and losses in a given environment. For example, propagation models have traditionally focused on predicting the average received signal strength at a given distance from the transmitter, plus the variability in the signal intensity near a particular location area. Thus, propagation models are mathematical tools used by engineers and scientists to plan and optimize wireless network systems.
Popular empirical models for pathloss are the Okumura-Hata Model and Ericsson 9999 Model.The Okumura model is one of the most used models for signal prediction in the urban areas. It applies to frequencies in the range of 150 MHz to 1920 MHz and distances from 1 to 100 km . The Okumura model is entirely based on measured data without any analytical explanation. Nevertheless, it is very practical and has become a standard for planning land mobile radio systems in Japan. The Okumura model is a very good model in urban and suburban areas, but not so good in rural areas due to its slow response to rapid changes in the terrain.
One can use Neural Networks for this pathloss prediction considering it as a regression model.Here we are doing the regression using Support vectors which has the benefit of lower training time compared to ANNs.

3.2.1 Support Vector Regression

Initially developed for solving classification problems, SVM techniques can be successfully applied in regression, i.e., for function approximation problems.
In regression problems, we estimate the functional dependence of the output variable on an n -dimensional input variable. In other words, we deal with real valued functions and we model an $\mathbb{R}^n$ to $\mathbb{R}$ mapping.

The general regression learning problem is as follows. The ML algorithm is given a set of training data from which it tries to learn the input-output relationship. So, consider a training data set $D = \{(\mathbf{x}_i, y_i) \in \mathbb{R}^n \times \mathbb{R}, i = 1, 2, \ldots, \ell\}$ with ℓ pairs $(\mathrm{x}_1, y_1), (\mathrm{x}_2, y_2), \ldots, (\mathrm{x}_\ell, y_\ell)$, where the inputs are n -dimensional vectors $\mathbf{x}_i \in \mathbb{R}^n$, the outputs $y_i \in \mathbb{R}$ are continuous values and ℓ is the number of samples in the training data set.

Starting from the linear regression problem, assume that $h(\mathbf{x}_i, \mathbf{w})$ is a linear regression hyperplane given by

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

where $\mathbf{w} \in \mathbb{R}^n$ is the normal vector to this hyperplane, the scalar $b \in \mathbb{R}$ is called a bias, $\langle \cdot, \cdot \rangle$ is the inner product operator. In the case of SVR, we measure the error of approximation instead of the margin used in classification. With this in mind, we use a function named Vapnik's linear loss function with ε -insensitivity zone defined as [16]

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

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

The solution of the linear regression learning problem concerns to find the linear function that approximates the training pairs $(\mathbf{x}_i, y_i)$ with an accuracy ε. In other words, we need to find a vector $\mathbf{w}$ that minimizes the error, which implies to solve the optimization problem given by

$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2$$

restricted to $|e_i| \le \varepsilon$ To obtain sparse solutions and penalize the large residuals, a penalty term is included

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

The optimization problem can be relaxed using slack variables ξ and $\hat{\xi}$, we can rewrite the optimization problem as

$\min_{\mathbf{w}, b} \left[\frac{1}{2} \|\mathbf{w}\|^2 + C \left(\sum_{i=1}^{\ell} (\xi_i + \hat{\xi}_i) \right) \right]$ under the restrictions

$$\begin{cases} |e_i| = \varepsilon + \xi \\ |e_i| = \varepsilon + \hat{\xi} \\ \xi, \hat{\xi} \ge 0 \end{cases}$$

which can be solved using Lagrange multipliers. After calculating the Lagrange multiplier vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^*$, the best regression hyperplane obtained is given by

$$h(\mathbf{x}_i, \mathbf{w}) = \sum_{i=1}^{\ell} \left(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*\right) \left\langle \mathbf{x}_i^T, \mathbf{x}_i \right\rangle + b$$

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

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. At last, the theoretical path loss of the Okumura-Hata model was also used as an input of the SVR training algorithm.K-fold cross validation is used as a performance measure.

4 Unsupervised Learning

In 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. If needed we can select the number of clusters we want the algorithm to create. Popular Unsupervised learning models are the K-means clustering ,Hierarchical clustering. We discuss few applications of these in communication scenarios.

4.1 Big Data Driven Anomaly Detection for Cellular Networks

Anomaly detection is a method of searching for data items that do not match an expected behavior or pattern in a given data set.Network anomaly detection methods in the literature can be classified into three main categories: statistics-based, classification-based, and clustering-based.
Statistics-based anomaly detection methods are the ones which apply statistical models for anomaly detection, mainly include threshold-based methods and principal component analysis (PCA) approach. Threshold-based methods first select features, and then estimate features distribution with suitable statistical model. Finally, a threshold is set to distinguish normal traffic activity from the abnormal one.. The main drawback of the threshold-based technique is that it needs prior knowledge about anomalous behaviors to set the appropriate threshold. Although the PCA method requires no priori knowledge and can report abrupt changes after a long monitoring time, the effectiveness of PCA anomaly detection technique is sensitive to the degree of cohesion of the traffic data.

Classification-based methods are supervised learning methods such as the rule-based method, decision tree, Bayesian network, neural network, and support vector machine (SVM) .Classification-based methods utilize labeled training data to learn a classifier, and then employ the learned classifier to classify a test instance as normal or anomalous. Although classification-based anomaly detection methods are popular because of their considerable flexibility and high detection probability, they require labeled training data.

Clustering techniques are widely used in anomaly detection . Clustering does not require pre-labeled data to group similar data instances. We can have a framework that uses hierarchical clustering method to cluster mobile call profile into different categories and can identify abnormal traffic activities by comparing with normal one.The authors compared hierarchical clustering with the k-means clustering and found that k-means clustering approach typically has lower time complexity.

By using the some feature extraction procedure we can get the traffic pattern. Now a city is segmented into several distinct grouped areas. In each grouped area, k-means algorithm employs phone activity value in some time slots of users as objects to detect the anomalies.

4.2 Multi-hop Wireless Broadcast Relay Selection in Urban Vehicular Networks

Another application for K-means clustering algorithm and its classification capabilities is in selecting an efficient relay selection among urban vehicular networks.In [4] The authors investigated the methods for multi-hop wireless broadcast and how K-means is a key factor in the decision-making and learning steps of the base stations, that learn from the distribution of the devices and chooses automatically which are the most suitable devices to use as a relay node.
Many services rely on the performance of broadcast communication to disseminate data packets. 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. The following is the algorithm in [4].

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

4.3 Unsupervised Machine Learning in 5G Networks for Low Latency Communications

4.3.1 Fog Networking

Stringent latency requirements of 5G applications have driven a paradigm shift in the state-of-art 4G networks based on the idea of making cloud closer to the end devices known as fog networking. **Fog Networking** is an architecture that uses edge devices to carry out a substantial amount of computation,storage and communication locally which is then routed over to internet backbone. In fog networking, some nodes are specialised as fog nodes to control and provide services to end devices.

Fog networking can be effectively incorporated in heterogeneous networks(HetNets) that are composed of a high power node (HPN) and many low power nodes (LPNs), where some LPNs are upgraded as fog nodes.This procedure can enhance the performance of HetNets by not only reducing the interference among LPN's, but also provide service to the end devices relying on the storage capability of fog nodes.

LPNs that are upgraded to fog nodes can be determined based on an **unsupervised soft clustering** machine learning algorithm similar to K-Means. We design an algorithm to cluster LPN's so that leader of each cluster is upgraded to fog node. We assume that number of fog nodes and location of all LPN's with a cell are known. The metric in designing the above algorithm is to reduce latency.

4.4 Clustering Algorithm

Assume that the number of fog nodes and the LPN are known and there is a data set composed of the geographical locations of the LPNs such as $X = x_1, x_2......x_N \in \mathrm{R}^2$ where N is the total number of LPNs and K of them will be upgraded to fog nodes.

To determine a clustering that reduces the latency within the network, LPN's are clustered according to their channel strength, which means that each LPN associates with fog nodes that have channels above a certain quality. The steps of the algorithm are as below

1. Set the number of fog nodes that will be upgraded from LPNs and the number of LPNs that are given as a priori information.
2. Specify the quality of channels among all LPNs.
3. Find the fog nodes according to the channel quality.
4. Determine the probability of connection between fog nodes and LPNs according to the channel quality.

5 Reinforcement Learning

Reinforcement Learning scheme is based on a learning system often called agent, that reacts to the environment. The agent performs actions and gets rewards or penalties (negative rewards) in return for its actions. That means that the agent has to learn by itself creating a policy that defines the action that the agent should choose in a certain situation. The aim of the reinforcement-learning task is to maximize the aforementioned reward over time.
However, this learning process, even though proved to converge, takes a lot of time to reach the best policy as it has to explore and gain knowledge of an entire system, making it unsuitable and inapplicable to large-scale networks. Consequently, applications of reinforcement learning are very limited in practice. Recently deep learning has been introduced as a new breakthrough technique. It can overcome the limitations of reinforcement learning, and thus open a new era for the development of reinforcement learning, namely Deep Reinforcement Learning (DRL). The DRL embraces the advantage of Deep Neural Networks (DNNs) to train the learning process, thereby improving the learning speed and the performance of reinforcement learning algorithms. As a result, DRL has been adopted in a numerous applications of reinforcement learning in practice such as robotics, computer vision, speech recognition, and natural language processing.

5.1 Advantages of DRL

1. DRL can obtain the solution of sophisticated network optimizations. Thus, it enables network controllers, e.g., base stations, in modern networks to solve non-convex and complex problems, e.g., joint user association, computation, and transmission schedule, to achieve the optimal solutions without complete and accurate network information.

2. DRL allows network entities to learn and build knowledge about the communication and networking environment. Thus, by using DRL, the network entities, e.g., a mobile user, can learn optimal policies, e.g., base station selection, channel selection, handover decision, caching and offloading decisions, without knowing channel model and mobility pattern.

3. DRL provides autonomous decision-making. With the DRL approaches, network entities can make observation and obtain the best policy locally with minimum or without information exchange among each other. This not only reduces communication overheads but also improves security and robustness of the networks.

4. DRL improves significantly the learning speed, especially in the problems with large state and action spaces. Thus, in large-scale networks, e.g., IoT systems with thousands of devices, DRL allows network controller or IoT gateways to control dynamically user association, spectrum access, and transmit power for a massive number of IoT devices and mobile users.

5. Several other problems in communications and networking such as cyber-physical attacks, interference management, and data offloading can be modeled as games, e.g., the non-cooperative game. DRL has been recently used as an efficient tool to solve the games, e.g., finding the Nash equilibrium, without the complete information.

6 Neural Network for Optimization of Routing in Communication Networks

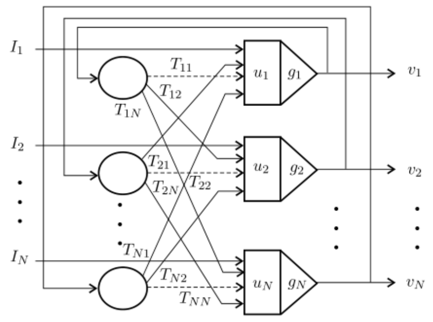
In modern communication networks, particularly in packet switched networks, routing is an important process that has a significant impact on the network's performance Ideal routing algorithm comprises finding the "optimal" path(s) between source and destination router, enabling high-speed data transmission and avoiding a packet loss. Since modern communication traffic is characterized by huge amount of source-destination pairs, high variability, non-linearity and unpredictability, the routing policy is a very hard task. Under some assumptions the optimal routing may be considered as the shortest path computations that have to be carried out in real time. This makes neural networks very good candidates for solving the problem, due to their high computational speed and the possibility of working with uncertain data.The suggested routing algorithm is designed to find the optimal path i.e the shortest path (if possible), taking into account the traffic conditions: the incoming traffic flow, routers occupancy, and link capacities, avoiding the packet loss due to the input buffer overflow.

Hop-field neural networks (HNN) is introduced by Hop-field and Tank (1985) for solving Traveling Salesman Problem(TSP). Later it was used in communication networks.Travelling Salesman Problem: Given a set of cities and distance between every pair of cities, the problem is to find the shortest possible route that visits every city exactly once and returns to the starting point.The problem is a famous NP hard problem. There is no polynomial time know solution for this problem.
Clearly for n cities case there are $n!$ possible loops. Among all solutions there are $2n$ loops with the same lengths, since each loop has the degeneracy of n -th order, depending on the starting point in the loop, and the second-order degeneracy depending on the loop direction. Consequently, there are $n!/2n$ different loops among we need the shortest one. The TSP problem is computationally very hard if the number of cities increases. For instance, for $n = 5$ there is only 12 different loops, for $n = 10$ the number is 181,440 while for $n = 30$ it amounts 4.4×10^{30}.

6.1 Hopfield Neural Network

HNN are recurrent or fully interconnected neural networks.The processing elements (neurons) are fully connected: output, v_i, of each ith cell is connected to inputs of all other neurons via synaptic weights, T_{ij}, producing the modification of cell inputs,u_i, and thus changing the network state. If the system is stable successive iterations lead to smaller

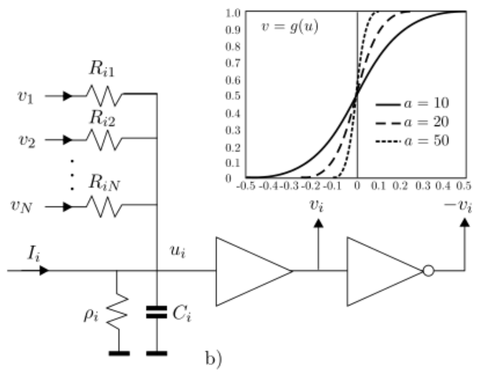
and smaller output changes and network reaches some minima of the system energy. Each cell is externally excited by input bias I_i.
The Hopfield neural computational circuit is shown in figure below.



The activation function is the sigmoid function g(u_i)

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

The coefficient a_i determines the steepness of the sigmoidal function.In hardware implementation, processing cells are realized as summing amplifiers with nonlinear transfer function.Possible hardware realization of ith neuron and its activation function are shown below.



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.

Hopfield network with N neurons may have $M = 2^N$ distinct states associated with an

N -dimensional hypercube with sides $v_i \in \{0,1\}$. Stable states are determined by the network weights and the current inputs. It is well known that recurrent networks are stable if the matrix of weights is symmetrical with zeros on its main diagonal. In Hopfield realization that means, $T_{ij} = T_{ji}$, and $T_{ii} = 0$
The general energy function of the neural network is

$$E = -\frac{1}{2}\sum_{i=1}^{N}\sum_{\substack{j=1\\ i\neq j}}^{N} T_{ij}v_j v_i - \sum_{i=1}^{N} I_i v_i$$

The change in energy, due to the change in the state of neuron i, is

$$\frac{\partial E}{\partial v_i} = -\sum_{j=1}^{N} T_{ij}v_j - \sum_{i=1}^{N} I_i$$

implies

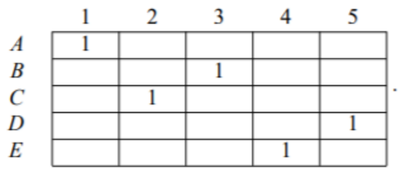
$$C_i\frac{du_i}{dt} = -\frac{u_i}{R_i} - \frac{\partial E}{\partial v_i}$$

or, in stable state, when signals are unchangeable: $du_i/dt = 0$, one can obtain

$$\partial E = -\frac{u_i}{R_i}\partial v_i$$

6.2 HNN for shortest path

TSP can be described by a square matrix of dimension $n \times n$ whose rows correspond to the particular city, X, while columns denote the sequence, i, of passing through cities. Such a matrix has only one non-zero element, of the value of unity, at each row and at each column. As an example, for $n = 5$ cities, labeled A, B, C, D and E, with shortest path loop $A - C - B - E - D - A$, this matrix has the following form:



	1	2	3	4	5
A	1				
B			1		
C		1			
D					1
E				1	

Hopfield used a neural net with N = n×n neurons and by minimizing its energy function it gets the shortest path - the minimum of the energy function (the most stable state) corresponding to the shortest path. The routers belonging to this path have the final states equal to unity, as in matrix. Energy function must favor strongly stable states of the form as described by a matrix, and of the n! such solutions it must favor those representing shortest paths. Possible energy function satisfying these requirements is of the form

$$E = \frac{A}{2}\sum_X \sum_i \sum_{j \neq i} v_{Xi} v_{Xj} + \frac{B}{2}\sum_i \sum_X \sum_{X \neq Y} v_{Xi} v_{Yi}$$
$$+ \frac{C}{2}\left(\sum_X \sum_i v_{Xi} - n\right)^2 + \frac{D}{2}\sum_X \sum_{Y \neq X} \sum_i d_{XY} v_{Xi} \left(v_{Y,i+1} + v_{Y,i-1}\right)$$

The first three terms, if A, B and C are sufficiently large and positive, enforce valid tours, while the last term assures that the valid tour represents the shortest path. Subscripts are of the form 'modulo n ', that means the nth city is adjacent in the tour to both city $(n-1)$ and city $1: v_{Y,n+j} = v_{Y,j}$. In all terms double indices of the form Xi are used where the first (row) subscript corresponds to the city name and the second (column) subscript has the interpretation of the position of that city in a tour.

From the above relations the connection(conductance) matrix for the TSP problem solving, has the terms

$$T_{Xi,Yj} = -A\delta_{XY}\left(1-\delta_{ij}\right) - B\delta_{ij}(1-\delta_{XY}) - C - Dd_{XY}\left(\delta_{j,i+1} + \delta_{j,i-1}\right)$$

where $\delta_{ii} = 1, \delta_{ij} = 0$, for $i \neq j$ and the bias currents are $I_{xi} = Cn$.

6.3 Design of Neural Network for Routing in Packet-Switched Network

The optimal routing in communication network may be considered as the shortest path problem. For this purpose instead of distances, d_{ij}, describing the path between routers i and j, some other attributes called the link costs, C_{ij}, are associated to links, describing the transmission conditions between routers . Then, the goal is to minimize the total cost, i.e., to find the 'path' between source (s)and destination router $(d): C_{si} + C_{ij} + C_{jk} + \cdots + C_{rd}$, that has the minimum length.There are many types of neural networks for this purpose but some of them need prior knowledge of number of routers in the path and some need to change the conductances based on the source destination pair.

Significant improvements in neural network algorithm are derived in paper offered by Ali and Kamoun . For n routers problem their computational network uses $n(n-1)$ neurons - the diagonal elements in the connection matrix $n \times n$ are removed - and find the shortest path from final stable neuron states. A suitable energy function is of the form

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

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. 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 (the number of incoming links is equal to the number of outgoing links); μ_4 forces the state of the neural network to converge to one of the stable states - corners of the hypercube defined by $v_i \in \{0,1\}$. 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. This term is introduced to ensure the source and the destination routers belong to the solution (the shortest path). The main contribution in Ali-Kamoun's paper [2] is that synaptic conductances are constant, given by

$$T_{Xi,Yj} = \mu_4 \delta_{XY} \delta_{ij} - \mu_3 \left(\delta_{XY} + \delta_{ij} - \delta_{jX} - \delta_{[Y} \right)$$

while the link costs and the information about the connection between routers are associated to the bias currents

$$\begin{aligned} I_{Xi} &= -\frac{\mu_1}{2} C_{Xi} (1 - \delta_{Xd} \delta_{is}) - \frac{\mu_2}{2} \rho_{Xi} (1 - \delta_{Xd} \delta_{is}) - \frac{\mu_4}{2} + \frac{\mu_5}{2} \delta_{Xd} \delta_{is} \\ &= \begin{cases} \frac{\mu_5}{2} - \frac{\mu_4}{2}, & \text{for } (X,i) = (d,s) \\ -\frac{\mu_1}{2} C_{Xi} - \frac{\mu_2}{2} \rho_{Xi} - \frac{\mu_4}{2}, & \text{otherwise} \end{cases} \end{aligned}$$

In this way the neural network SP algorithm becomes very attractive for real time processes. Ali and Kamoun applied this algorithm to optimize delay in computer networks under different network topologies and link costs. We can also address the packet loss reduction. Namely, each link is characterized by its capacity, K_{ij} (in data units per second). Note that the link capacity from router i to router j and in opposite direction should not be the same, in general. On the other hand, each link has its traffic density, G_{ij} (also in data units per second), describing the actual traffic flow in particular direction. Certainly, the traffic density may not exceed the link capacity; i.e., it must be satisfied $K_{ij} \geq G_{ij}$. If, in some circumstances, incoming packets overflow the input buffer, these packets would be lost. In order to minimize this effect we introduce a new term observing the free space in a link capacity. This term is the difference $\left(K_{ij} - G_{ij}\right)$ and the routing have to find a path with enough free space in link capacity. Since the neural network is designed to minimize the energy function, actual term in energy function have to be of the form . Following this reason we introduced new term, μ_6, in biases

$$\begin{aligned} I_{Xi} &= -\frac{\mu_1}{2} C_{Xi} (1 - \delta_{Xd} \delta_{is}) - \frac{\mu_2}{2} \rho_{Xi} (1 - \delta_{Xd} \delta_{is}) \\ &- \frac{\mu_4}{2} + \frac{\mu_5}{2} \delta_{Xd} \delta_{is} + \frac{\mu_6}{2} [1 - (K_{Xi} - G_{Xi})](1 - \delta_{Xd} \delta_{is}) \\ &= \begin{cases} \frac{\mu_5}{2} - \frac{\mu_4}{2}, & \text{for } (X,i) = (d,s) \\ -\frac{\mu_1}{2} C_{Xi} - \frac{\mu_2}{2} \rho_{Xi} - \frac{\mu_4}{2} + \frac{\mu_6}{2} [1 - (K_{Xi} - G_{Xi})], & \text{otherwise} \end{cases} \end{aligned}$$

The term μ_6 minimizes the possibility of packet loss under the constraint $K_{ij} \geq G_{ij}$. If G_{ij} exceeds the link capacity, K_{ij}, this router is removed from the algorithm by setting an appropriate term ρ_{ij} into the matrix $\boldsymbol{\rho}$ defining the existence of links between routers (the routers connectivity) to the value of 1 By substituting the Bias currents and Conductance matrix in the state equation, we get

$$\begin{aligned} C_i \frac{du_i}{dt} &= -\frac{u_i}{R_i} - \frac{\mu_1}{2} C_{Xi} (1 - \delta_{Xd} \delta_{is}) - \frac{\mu_2}{2} \rho_{Xi} (1 - \delta_{Xd} \delta_{is}) \\ &- \mu_3 \sum_{Y \neq X} (v_{XY} - v_{YX}) + \mu_3 \sum_{Y \neq X} (v_{iY} - v_{Yi}) \\ &- \frac{\mu_4}{2} (1 - 2 v_{Xi}) + \frac{\mu_5}{2} \delta_{Xd} \delta_{is} + \frac{\mu_6}{2} [1 - (K_{Xi} - G_{Xi})](1 - \delta_{Xd} \delta_{is}) \end{aligned}$$

7 References

[1] J. Liu, R. Deng, S. Zhou, and Z. Niu, “Seeing the unobservable: Channel learning for wireless communication networks,” 2015 IEEE Global Communications Conference.
[2] R. D. a. Timoteo, D. C. Cunha, and G. D. C. Cavalcanti, “A Proposal for Path Loss Prediction in Urban Environments using Support Vector Re- gression,” Advanced International Conference on Telecommunications
[3] M. S. Parwez, D. B. Rawat, and M. Garuba, “Big data analytics for user-activity analysis and user-anomaly detection in mobile wireless network,” IEEE Transactions on Industrial Informatics
[4] W. Song, F. Zeng, J. Hu, Z. Wang, and X. Mao, “An UnsupervisedLearning-Based Method for Multi-Hop Wireless Broadcast Relay Selection in Urban Vehicular Networks,” IEEE Vehicular Technology Conference.
[5] Artificial Intelligence in 5G Technology: A Survey by Manuel Eugenio Morocho Cayamcela , Wansu Lim
[6] Neural Network for Optimization of Routing in Communication Networks by Nenad Kojic, Irini Reljin, and Branimir Reljin
[7] When Machine learning meets Wireless Networks : Deployment,Challenges and Applications - Ursula Challita, Henrik Ryden, Hugo Tullberg
[8] Applications of Deep Reinforcement Learning in Communications and Networking: A Survey by Nguyen Cong Luong, Dinh Thai Hoang, Shimin Gong, Dusit Niyato, Ping Wang, Ying-Chang Liang, Dong In Kim.
[9] E. Balevi and R. D. Gitlin, “Unsupervised machine learning in 5G networks for low latency communications,” 2017 IEEE 36th International Performance Computing and Communications Conference.