Spatial Deep Learning for Wireless Scheduling

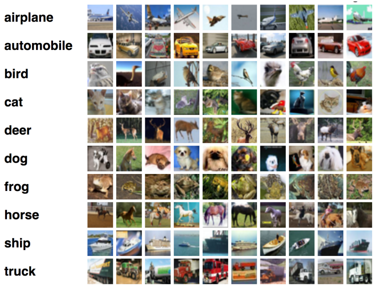
Wei Yu

Electrical and Computer Engineering Department
University of Toronto

2019

Machine Learning

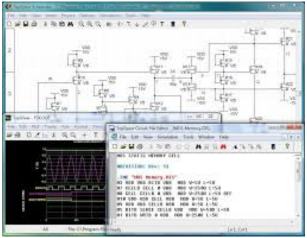
- Machine learning is having tremendous successes in many tasks:
  - Image classification; speech recognition/translation; face recognition...
  - Superhuman performance is now commonplace.



(images from Internet subject to copyright)

Machine vs. Human

- In many tasks, machines have *always* been better than human:
  - Examples: numerical calculation; circuit simulation; coding/decoding
  - Simple low-dimensional input – complex calculation
  - Accurate computational models are available.



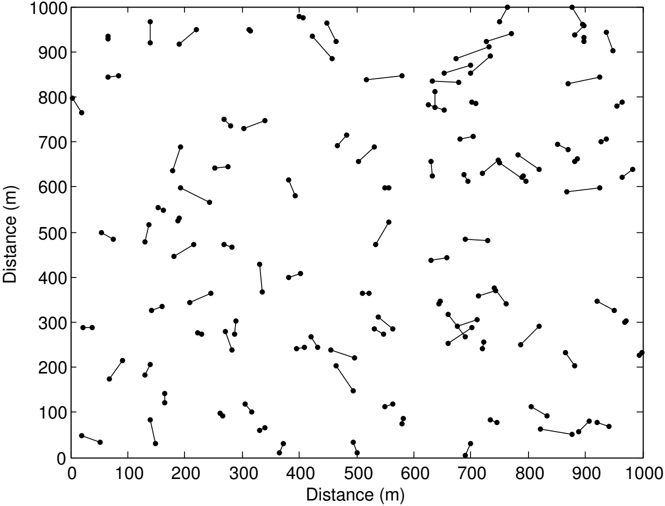
(images from Internet subject to copyright)

- Surprise is that machines are now better in *uniquely human* tasks:
  - Complex high-dimensional input – making judgements
  - Accurate computational models are NOT available.

Machine Learning for Wireless Communications

- In communication engineering, channel models are cherished:
  - Link level: Additive white Gaussian noise (AWGN) model is justified.
  - Network level: Opportunities abound when there is human element.
  - System level: Can machine learn to perform complex optimization?
- Main point of this talk: The role of machine learning is when
  - Models are expensive to obtain.
  - Optimization is complex and difficult to perform.

Link Scheduling in Device-to-Device Networks



Scheduling for Dynamic Interference Control

- The data rate of link i is limited by interference:

$$R_i(\mathbf{x}) = \log\left(1 + \frac{|h_{ii}|^2 p_i x_i}{\sum_{j\in\mathcal{L}, j\neq i} |h_{ij}|^2 p_j x_j + \sigma^2}\right).$$

- Coordinated scheduling and power control:
  - Scheduling: Select a subset of links to activate, i.e., $x_i \in \{0, 1\}$.
  - For this talk, we assume fixed power p_i.
- This is an NP-hard discrete optimization problem.
- Its relaxation is non-convex: $\mathbf{x}$ in both numerator & denominator.

Traditional Optimization Based Approach

- Formulate a weighted sum-rate maximization problem:

$$\begin{aligned} \text{maximize} \quad & \sum_{i \in \mathcal{L}} w_i \log \left(1 + \frac{|h_{ii}|^2 p_i x_i}{\sum_{j \in \mathcal{L}, j \neq i} |h_{ij}|^2 p_j x_j + \sigma^2} \right). \\ \text{subject to} \quad & x_i \in \{0, 1\} \end{aligned}$$

- Traditional approach: Two-step process
  - Obtain channel state information (CSI): $\theta = \{h_{ij}\}$.
  - Solve the optimization problem to obtain $\mathbf{x}^*$ given θ.
- This traditional approach faces two challenges:
  - Solving the optimization problem is hard due to non-convexity.
  - Obtaining CSI is expensive due to the limited coherence time.

Outline of This Talk

1. Part I: Fractional Programming
2. Part II: Learn to Optimize

Part I: Scheduling via Fractional Programming

Single-Ratio Fractional Programming (FP)

- Given functions $A(\mathbf{x}) \geq 0$ and $B(\mathbf{x}) > 0$, a *single-ratio* FP problem is

$$\begin{aligned} \underset{\mathbf{x}}{\text{maximize}} \quad & \frac{A(\mathbf{x})}{B(\mathbf{x})} \\ \text{subject to} \quad & \mathbf{x} \in \mathcal{X}. \end{aligned}$$

- A classic approach is to decouple the ratio by Dinkelbach's transform:

$$\begin{aligned} \underset{\mathbf{x}}{\text{maximize}} \quad & A(\mathbf{x}) - yB(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{x} \in \mathcal{X}. \end{aligned}$$

  Then update $y = A(\mathbf{x})/B(\mathbf{x})$.

- When $A(\mathbf{x})$ is concave and $B(\mathbf{x})$ is convex, this leads to global optimal $\mathbf{x}$.

However, this cannot be extended to the multiple-ratio case.

A New Quadratic Transform

- We propose a novel quadratic transform that reformulates

$$\begin{aligned} \underset{\mathbf{x}}{\text{maximize}} \quad & \frac{A(\mathbf{x})}{B(\mathbf{x})} \\ \text{subject to} \quad & \mathbf{x} \in \mathcal{X}. \end{aligned}$$

as the following problem:

$$\begin{aligned} \underset{\mathbf{x},y}{\text{maximize}} \quad & 2y\sqrt{A(\mathbf{x})} - y^2 B(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{x} \in \mathcal{X}. \end{aligned}$$

- Both the optimal variable $\mathbf{x}^*$ and optimal objective value are the same.
- Proof: For fixed $\mathbf{x}$, the optimal $y = \frac{\sqrt{A(\mathbf{x})}}{B(\mathbf{x})}$.

This transform can be readily extended to a multiple-ratio case.

Multiple-Ratio FP

- Given K pairs of $A_i(\mathbf{x}) \geq 0$ and $B_i(\mathbf{x}) > 0$, a multiple-ratio problem is

$$\begin{aligned} \underset{\mathbf{x}}{\text{maximize}} \quad & \sum_{i=1}^{K} \frac{A_i(\mathbf{x})}{B_i(\mathbf{x})} \\ \text{subject to} \quad & \mathbf{x} \in \mathcal{X}. \end{aligned}$$

- By the quadratic transform, the problem is reformulated as

$$\begin{aligned} \underset{\mathbf{x},\mathbf{y}}{\text{maximize}} \quad & \sum_{i=1}^{K} \left(2y_i \sqrt{A_i(\mathbf{x})} - y_i^2 B_i(\mathbf{x}) \right) \\ \text{subject to} \quad & \mathbf{x} \in \mathcal{X}. \end{aligned}$$

D2D Scheduling Problem

- The data rate of link i is

$$R_i(\mathbf{x}) = \log\left(1 + \frac{|h_{ii}|^2 p_i x_i}{\sum_{j\in\mathcal{L}, j\neq i} |h_{ij}|^2 p_j x_j + \sigma^2}\right).$$

- Formulate maximum weighted sum rate problem as a multi-ratio problem:

$$\begin{aligned} \underset{\mathbf{x}}{\text{maximize}} \quad & \sum_{i\in\mathcal{L}} w_i R_i(\mathbf{x}) \\ \text{subject to} \quad & x_i \in \{0,1\},\ \forall i. \end{aligned}$$

Lagrangian Reformulation

- Consider the problem $\max_x \log\left(1+\frac{A}{B}\right)$.
- Rewrite as $\max_x \log(1+\gamma)$ subject to $\gamma = \frac{A}{B}$.
- Introduce the Lagrangian $L = \log(1+\gamma) - \lambda\left(\gamma - \frac{A}{B}\right)$.
- ...
- Reformulate $\max_x \log(1+\frac{A}{B}) \iff \max_{(x,\gamma)} f_r(x,\gamma)$, where

$$f_r(x,\gamma) = \log(1+\gamma) - \gamma + \frac{(1+\gamma)A}{A+B},$$

Reformulations for D2D System

- Applying this technique to D2D gives (recall $x_i \in \{0, 1\}$ is for schedule)

$$f_r(\mathbf{x}, \boldsymbol{\gamma}) = \sum_{i \in \mathcal{L}} w_i \log(1 + \gamma_i) - \sum_{i \in \mathcal{L}} w_i \gamma_i + \underbrace{\sum_{i \in \mathcal{L}} \frac{w_i(1 + \gamma_i)|h_{ii}|^2 p_i x_i}{\sum_{j \in \mathcal{L}} |h_{ij}|^2 p_j x_j + \sigma^2}}_{\text{multiple-ratio term}}.$$

- Thus, $\max_{\mathbf{x}} \sum w_i \log(1 + \mathsf{SINR}_i) \Longleftrightarrow \max_{\mathbf{x}, \boldsymbol{\gamma}} f_r(\mathbf{x}, \boldsymbol{\gamma})$.

- For fixed $\mathbf{x}$, the optimal $\boldsymbol{\gamma}$ is (by solving $\partial f_r / \partial \gamma_i = 0$)

$$\gamma_i^* = \frac{|h_{ii}|^2 p_i x_i}{\sum_{j \in \mathcal{L}, j \neq i} |h_{ij}|^2 p_j x_j + \sigma^2}.$$

Reformulations for D2D System (cont.)

- Further applying the quadratic transform to the last term of f_r gives

$$f_q(\mathbf{x},\boldsymbol{\gamma},\mathbf{y}) = \sum_{i\in\mathcal{L}} 2y_i\sqrt{w_i(1+\gamma_i)|h_{ii}|^2 p_i x_i} - \sum_{i\in\mathcal{L}} y_i^2\left(\sum_{j\in\mathcal{L}}|h_{ij}|^2 p_j x_j + \sigma^2\right) + \text{const}(\boldsymbol{\gamma}).$$

- Thus, $\max_{\mathbf{x}}\sum w_i\log(1+\mathsf{SINR}_i) \iff \max_{\mathbf{x},\boldsymbol{\gamma}} f_r \iff \max_{\mathbf{x},\boldsymbol{\gamma},\mathbf{y}} f_q$.

- For fixed $\mathbf{x}$ and $\boldsymbol{\gamma}$, the optimal $\mathbf{y}$ is (by solving $\partial f_q/\partial y_i = 0$)

$$y_i^* = \frac{\sqrt{w_i(1+\gamma_i)|h_{ii}|^2 p_i x_i}}{\sum_{j\in\mathcal{L}}|h_{ij}|^2 p_j x_j + \sigma^2}.$$

Optimization of Scheduling Variable

- Note that f_q can be decoupled on a per-link basis with respect to $\mathbf{x}$:

$$f_q(\mathbf{x}, \boldsymbol{\gamma}, \mathbf{y}) = \text{const}(\boldsymbol{\gamma}, \mathbf{y}) + \sum_{i \in \mathcal{L}} Q_i(x_i, \boldsymbol{\gamma}, \mathbf{y})$$

  where the per-link function Q_i is defined to be

$$Q_i(x_i, \boldsymbol{\gamma}, \mathbf{y}) = 2y_i \sqrt{w_i(1+\gamma_i)|h_{ii}|^2 p_i x_i} - \sum_{j \in \mathcal{L}} y_j^2 |h_{ji}|^2 p_i x_i.$$

- The optimal solution for $\mathbf{x}$ now becomes straightforward:

$$x_i^* = \arg\max_{x_i} Q_i(x_i, \boldsymbol{\gamma}, \mathbf{y}).$$

Proposed FPLinQ for D2D Scheduling

Algorithm 1 FPLinQ for scheduling D2D links

0) Initialize all the variables to feasible values.
repeat
  1) Update $\boldsymbol{\gamma}$;
  2) Update $\mathbf{y}$;
  3) Update $\tilde{\mathbf{x}}$ (relax $\mathbf{x}$ to real number);
until Convergence
4) Recover the integer $\mathbf{x}$.

Here, $\boldsymbol{\gamma}$ and $\mathbf{y}$ are intermediate variables that coordinate the scheduling of the links and slow down the convergence, thus outperforming greedy.

Learn to Optimize?

- FPLinQ is highly effective, but it still requires CSI.
- Obtaining CSI is expensive due to limited coherence time.
- Can a machine learn the optimal solution directly?

Part II: Learn to Schedule without CSI

Learn to Optimize

- If optimization is hard, can we learn the optimal solution directly?

$$\theta \longrightarrow \boxed{\max_{\mathbf{x}} f(\mathbf{x};\theta)} \longrightarrow \mathbf{x}^*$$

Instead of optimizing $\mathbf{x}$ for $\theta = \{h_{ij}\}$, we learn the mapping $\theta \to \mathbf{x}^*$

- Train a deep neural network to learn the functional mapping
  - Supervised learning: Using many examples of $(\theta, \mathbf{x}^*)$ from FPLinQ.
  - Unsupervised learning: Directly maximizing $f(\mathbf{x};\theta)$. Better strategy.

- Ask the deep neural network to produce the optimal $\mathbf{x}^*$ for a new θ.

Learn from Geographic Information

- For wireless scheduling, coherence time is limited, feedback is costly:
- Obtaining and feeding back $\theta = \{h_{ij}\}$ become the bottleneck:

$$|\{h_{ij}\}| = O(N^2)$$

- Instead of the channels, we use geographic location information as θ. Location information scales as $O(N)$.

$$\theta \longrightarrow \boxed{\max_{\mathbf{x}} f(\mathbf{x};\theta)} \longrightarrow \mathbf{x}^*$$

Neural network learns to map the geographic information to $\mathbf{x}^*$.

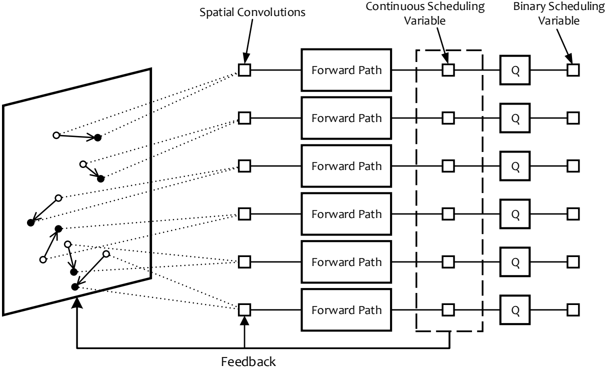
Spatial Deep Learning Approach

- Randomly generate D2D network (e.g 50 links over 500mx500m area)
- The network is represented Geographic Location Information (GLI): a set of vectors $\{(\mathbf{l}_i^{\text{tx}}, \mathbf{l}_i^{\text{rx}})\}_i$, where $\mathbf{l}_i^{\text{tx}} \in \mathbb{R}^2$ and $\mathbf{l}_i^{\text{rx}} \in \mathbb{R}^2$ are the transmitter and the receiver locations of the ith link, respectively.
- Convolutional neural network with geographic information as input.
  - Back propagation over convolutional filter and connection weights.
  - Unsupervised learning with sum-rate as the objective function.
  - Testing/Validation on new D2D networks.

Related work using fully connected network with CSI as input: H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for wireless resource management," *IEEE Trans. Signal Processing*, 2018.

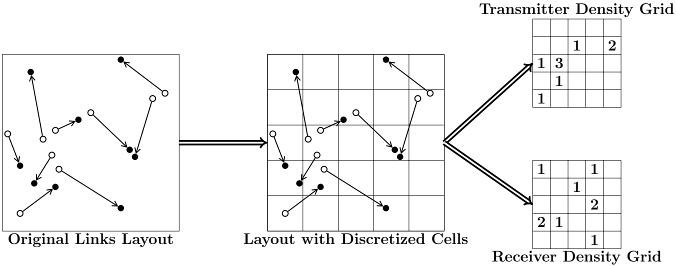
Novel Deep Neural Network

- The overall network structure consists of multiple feedback stages:



- Each iteration has a convolution and a fully connected forward path.

Density Grid

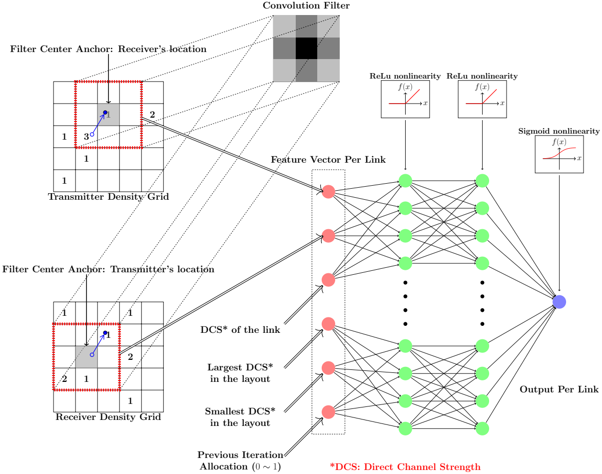


- Two density grid matrices are constructed with 5m×5m cells to represent the density of the active transmitters and receivers:

$$T(s,t) = \sum_{\{i|(s_i^{\text{tx}},t_i^{\text{tx}})=(s,t)\}} x_i$$

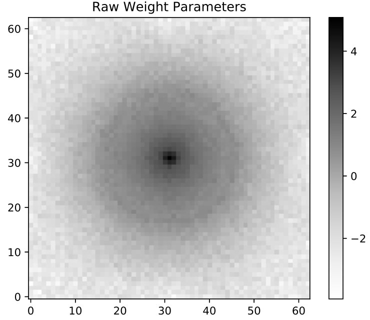
$$R(s,t) = \sum_{\{i|(s_i^{\text{rx}},t_i^{\text{rx}})=(s,t)\}} x_i.$$

Forward Path of Each Link



Convolution Filter: Summarizing the Interference

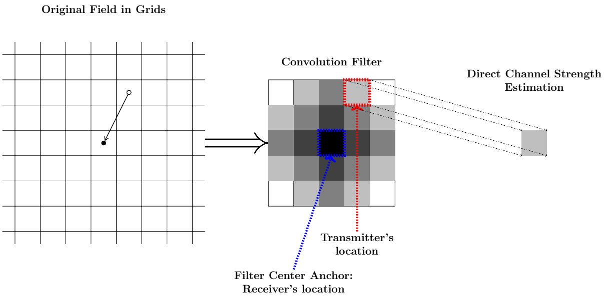
- The convolution filter range is up to 315 meters $\times$ 315 meters.
- Each link's own transmitter/receiver is subtracted from convolution.



- Radial pattern of the filter indicates delaying interference intensity.

Direct Link Strength

- Channel strength is estimated by extracting weight of trained filter.

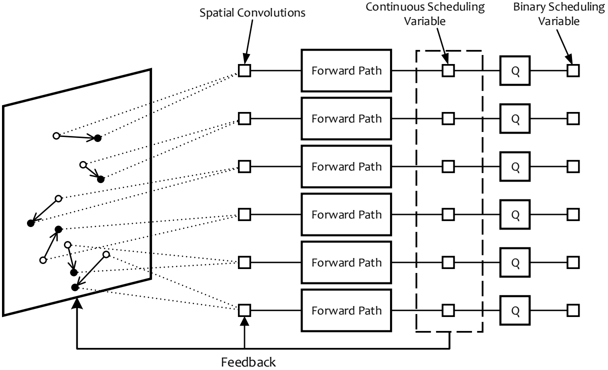


Fully Connected Stage

- After convolution stage, we form the feature vector for each link:
  - The total interference the transmitter causes to other links
  - The total interference the receiver is subject to by other links
  - The link strength and its range over the layout
  - The allocation status of the link from previous iteration via feedback
- This feature vector serves as input for the fully connected stage, with standard fully connected hidden layers with ReLU nonlinearities.
- The last layer uses sigmoid nonlinearity to squash the outputs into $[0, 1]$, indicating the power allocation of the link at end of iteration.

Feedback Connection

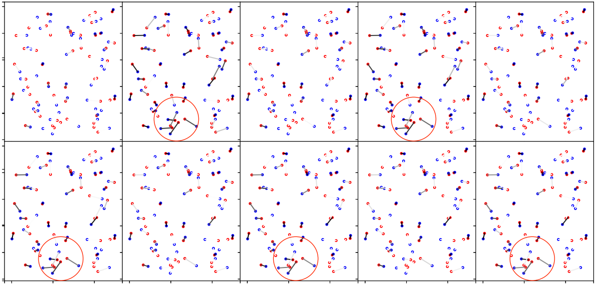
- The overall network structure consists of multiple feedback stages:



- We use x_i from the previous iteration to update the density grids, as well as forming the feature vector of the current iteration.

Stochastic Updates

- Oscillating ON/OFF behavior may occur



- Randomization is performed to break these oscillations.

Simulation Model

- Full frequency reuse with 5MHz bandwidth at 2.4GHz carrier frequency; 1.5m antenna height and 2.5dB antenna gain.
- Additive white Gaussian noise at -169dBm/Hz
- SNR gap at 6dB
- Max transmit power is set to be constant across each link at 40dBm
- Short-range outdoor model ITU-1411 distance-dependent pathloss

Training Configurations

- Each layout consists of 50 randomly placed D2D links over 500 meters $\times$ 500 meters region.
- Neural network is trained with datasets of 800,000 layouts with either
  - Tx-to-Rx distance uniformly distributed in 2 meters $\sim$ 65 meters, or
  - Tx-to-Rx distance uniformly distributed in 30 meters $\sim$ 70 meters.
- In the training stage, the feedback runs for 5 iterations.
- In the testing stage, the feedback runs for 30 iterations.

Sum Rate Maximization: Same Training/Testing Setting

Table: Sum rate of 50 links over 500m×500m area, with $30\text{m} \sim 70\text{m}$ Tx-to-Rx distance distribution, over 5000 testing layouts

Sum Rate (%)	CSI	No Fading	With Fading
Spatial Deep Learning	–	92.2	71.8
Greedy	✓	84.8	95.9
Strongest Links	✓	59.7	65.4
Random Selection	–	35.3	31.7
All Active	–	26.7	25.3
FP	✓	100	100

If scheduling using FP without fast fading, FP achieves 77.7%.

Generalizability

- We vary the network layouts to test the robustness of our approach:
  - Larger area with more links while maintaining link density
  - Same area but with more links and higher link densities
  - Layouts with varying distributions for Tx-to-Rx distances

Sum Rate Maximization: Generalizability

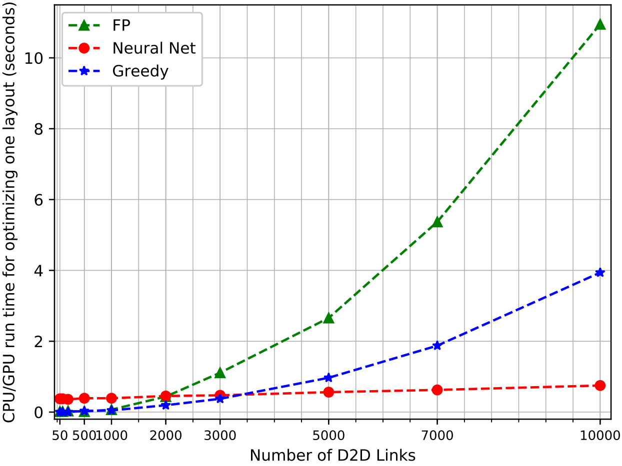
Table: $30\text{m} \sim 70\text{m}$ Tx-to-Rx distance distribution

Sum Rate (%)	FP	Neural Network	Greedy
200 Links; 1000×1000 m^2	100	94.58	104.68
450 Links; 1500×1500 m^2	100	95.46	106.03

Sum Rate (%)	FP	Neural Network	Greedy
200 Links; 500×500 m^2	100	92.17	89.73
500 Links; 500×500 m^2	100	91.35	92.41

Sum Rate (%)	FP	Neural Network	Greedy
$10\text{m} \sim 50\text{m}$	100	98.44	94.00
$30\text{m} \sim 100\text{m}$	100	88.12	85.51
30m fixed	100	96.64	84.56

Computational Complexity



Complexity of neural network is $O(N)$. Greedy and FP are both $O(N^2)$.

Proportional Fairness Scheduling

- Consider long-term average rate over exponentially weighted window:

$$\bar{R}_i^t = (1-\alpha)\bar{R}_i^t + \alpha R_i^t \tag{8}$$

- Proportional fairness scheduling aims to optimize:

$$\sum_{i=1}^{N} \log(\bar{R}_i). \tag{9}$$

- Equivalently, we optimize weighted sum rate:

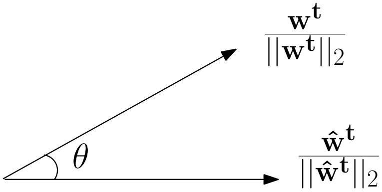
$$\sum_{i=1}^{N} w_i R_i^t \tag{10}$$

where

$$w_i = \left.\frac{\partial U(\bar{R}_i^t)}{\partial R}\right|_{\bar{R}_i^t} = \left.\frac{\partial \log(\bar{R}_i^t)}{\partial R}\right|_{\bar{R}_i^t} = \frac{1}{\bar{R}_i^t}. \tag{11}$$

Weighted Sum Rate Maximization via Sum Rate Max

- New Idea: The weighted sum rate optimization at each time slot could be approximated by sum rate optimization on a subset of links.
- We aim to use binary weights to approximate real weights:
  - Let $\mathbf{w^t}$ denote the original proportional fairness weight vector.
  - Find a binary vector $\hat{\mathbf{w}}^{\mathbf{t}}$ to minimize the angle in between.



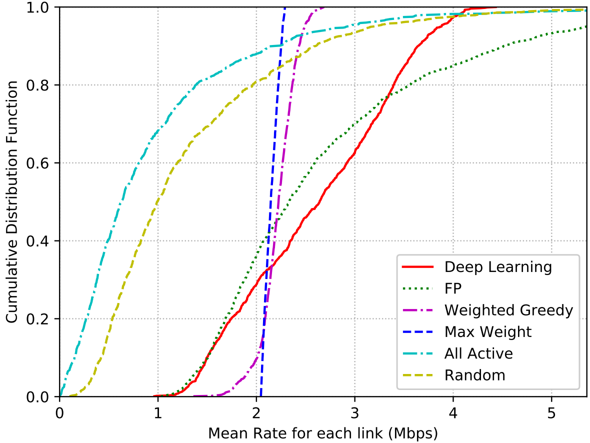
  - The "best" binary approximation to the original real weight vector.

Proportionally Fair Scheduling

Table: Sum rate of 50 links over 500m×500m area, with $30\text{m} \sim 70\text{m}$ Tx-to-Rx distance distribution, for 10 testing layouts over 500 scheduling slots

Methods	CSI	Sum Log Utility	5-Percentile Rate
Spatial Deep Learning	–	45.35	1.40 Mbps
Greedy	✓	39.63	1.88 Mbps
Random Selection	–	0.90	0.33 Mbps
All Active	–	-27.59	0.07 Mbps
FP	✓	45.24	1.35 Mbps

Cumulative Distribution of Average Rates over the Links



Summary

- We propose a new fractional programming method for link scheduling.
- We propose a novel neural network for scheduling interfering D2D links, bypassing channel estimation, while achieving good performance
- Key features:
  - Unsupervised learning using the sum rate as the optimization objective
  - Spatial convolution of geographic information to estimate interference
  - Per-link fully connected neural network to make scheduling decision
  - Overall feedback structure with stochastic update

Concluding Remarks

- Traditional communication system design: Model then optimize.
- Machine learning approach:
  - Use a universal and highly expressive model, e.g. deep neural network
  - Rely on large amount of training data.
- Machine learning is most useful when:
  - Models are difficult or expensive to obtain.
  - Inputs are high-dimensional or heterogeneous.
  - Computational complexity of producing optimized output is high.
- Matching neural network architecture to problem structure is key.

Further Information

- Wei Cui, Kaiming Shen, and Wei Yu,
  "Spatial Deep Learning for Wireless Scheduling",
  IEEE Journal on Selected Areas in Communications, vol. 37, no. 6, pp. 1248-1261, June 2019.
- Kaiming Shen and Wei Yu,
  "Fractional Programming for Communication Systems – Part I: Power Control and Beamforming",
  IEEE Transactions on Signal Processing, vol. 66, no. 10, pp. 2647-2630, May 15, 2018.
- Kaiming Shen, and Wei Yu,
  "Fractional Programming for Communication Systems – Part II: Uplink Scheduling via Matching",
  IEEE Transactions on Signal Processing, vol. 66, no. 10, pp. 2631-2644, May 15, 2018.