

Unsupervised Learning for C-RAN Power Control and Power Allocation

Rasoul Nikbakht, *Student Member, IEEE*, Anders Jonsson, Angel Lozano, *Fellow, IEEE*

Abstract—This paper applies a feedforward neural network trained in an unsupervised fashion to the problem of optimizing the transmit powers in centralized radio access networks operating on a cell-free basis. Both uplink and downlink are considered. Various objectives are entertained, some leading to convex formulations and some that do not. In all cases, the performance of the proposed procedure is very satisfactory and, in terms of computational cost, the scalability is manifestly superior to that of convex solvers. Moreover, the optimization relies on directly measurable channel gains, with no need for user location information.

Index Terms—Neural networks, unsupervised learning, cell-free networks, ultradense networks, power control, power allocation, C-RAN

I. INTRODUCTION

A preceding letter proposed leveraging the expressive power of neural networks (NNs) to solve large-dimensional parametric optimizations in an unsupervised fashion [?]. Then, in a follow-up letter, this approach was applied to power control for cellular systems [?]. In this final letter, it is further applied to the intertwined problems of power control and power allocation for centralized, possibly cloud-based, radio access networks (C-RAN). The more intricate optimizations that arise in this richer setting offer an excellent opportunity to further test the efficacy of the technique. In particular:

- In a C-RAN, the transmit power optimization is necessarily centralized and decidedly many-parametric; it runs into dimensionality issues faster than in cellular setups.
- The loss functions introduced in [?] continue to be applicable, but, in contrast with the cellular case, here they do not always admit convex forms. When they do not, the performance of our learning-based approach becomes itself a yardstick against which other solutions can be gauged.

C-RANs inherently operate on a cell-free basis: on each time-frequency resource unit, every access point (AP) communicates, rather than only with the user in its cell, with every user in the system [?]. This inherits ideas from network MIMO and takes to its natural limit the notion of cell cooperation [?]. C-RANs offer multiple advantages over their cellular counterparts, including large-scale diversity, interference handling,

flexibility, and elasticity, and it is reasonable to anticipate that, at least for ultradense deployments, C-RANs might become the norm [?], [?].

As in [?], we desire for the power control to respond only to the *large-scale* behavior of the channel gains, and not to the *small-scale* fluctuations that occur on scales of milliseconds (in time) and hundreds of kilohertz (in frequency). This guarantees relatively stable transmit powers, leaving the small-scale swings to the province of link adaptation [?].

II. C-RAN MODEL

On a time-frequency resource, a basic C-RAN consists of N APs and K users, all equipped with omnidirectional antennas. The positions of APs and users are uniformly random. Every AP can communicate with every user and, to render matched-filter beamforming effective, N is substantially larger than K .

A. Large-Scale Features

Signals are subject to distance-dependent power decay with exponent $\eta = 3.8$, for a large-scale channel gain $G_{n,k}$ between the k th user and the n th AP. Correspondingly, that link's signal-to-noise ratio is

$$\text{SNR}_{n,k} = \frac{G_{n,k}P}{\sigma^2} \quad (1)$$

with P the maximum transmit power and σ^2 the noise power.

Both APs and users are positioned uniformly at random, with shadowing implicitly embedded in that randomness [?].

For the sake of a cleaner notation, P/σ^2 is taken as equal for uplink and downlink, but asymmetries could be had by simply discriminating the respective SNR variables. We set P/σ^2 such that $\text{SNR}_{n,k} = 20$ dB at a distance d , where d would be the inter-AP spacing if the network were arranged as a hexagonal grid. Under reasonable values for P and σ^2 , this is compatible with a dense C-RAN.

In addition to $G_{n,k}$, the channel that connects the k th user with the n th AP includes a small-scale fading coefficient $h_{n,k} \sim \mathcal{N}_{\mathbb{C}}(0, 1)$, independent across users and APs. These fading coefficients are presumed known, with comments on the impact of channel estimation provided in Section ??.

B. Uplink

In the uplink, the n th AP observes

$$y_n = \sum_{k=0}^{K-1} \sqrt{G_{n,k}} h_{n,k} \sqrt{p_k} s_k + v_n, \quad (2)$$

R. Nikbakht, A. Jonsson, and A. Lozano are with Univ. Pompeu Fabra, 08018 Barcelona (e-mail: angel.lozano@upf.edu). Work supported by the European Research Council under the H2020 Framework Programme/ERC grant 694974, by the Maria de Maeztu Units of Excellence Programme (MDM-2015-0502), and by the ICREA Academia program. Parts of this paper were presented at the 2019 IEEE Int'l Symp. Personal, Indoor & Mobile Radio Communications [?] and at the 2020 IEEE Int'l Conf. Communications [?].

where s_k is the unit-power symbol emitted by user k whereas $p_k \in [0, 1]$ is its power control coefficient and $v_n \sim \mathcal{N}_{\mathbb{C}}(0, \sigma^2)$ is the noise. With the observations of the N APs centrally combined so as to effect matched-filter beamforming for each user, the SINR of user k equals [?]

$$\text{sinr}_k^{\text{UL}} = \frac{p_k \left(\sum_{n=0}^{N-1} \text{SNR}_{n,k} |h_{n,k}|^2 \right)^2}{\text{den}_k} \quad (3)$$

with

$$\begin{aligned} \text{den}_k &= \sum_{\ell \neq k} p_{\ell} \left| \sum_{n=0}^{N-1} \sqrt{\text{SNR}_{n,k} \text{SNR}_{n,\ell}} h_{n,k}^* h_{n,\ell} \right|^2 \\ &\quad + \sum_{n=0}^{N-1} \text{SNR}_{n,k} |h_{n,k}|^2. \end{aligned} \quad (4)$$

A subsequent expectation over the small-scale fading yields the operationally meaningful local-average SINR [?], i.e.,

$$\mathbb{E}[\text{sinr}_k^{\text{UL}}] = \mathbb{E} \left[\frac{p_k \left(\sum_{n=0}^{N-1} \text{SNR}_{n,k} |h_{n,k}|^2 \right)^2}{\text{den}_k} \right]. \quad (5)$$

To formulate loss functions that, as intended, drive the NN based only on large-scale quantities, we replace $|h_{n,k}|^2$ and $|h_{n,\ell}|^2$ by their expected value (unity), and $h_{n,k}^* h_{n,\ell}$ for $\ell \neq k$ also by their expected value (zero), obtaining the proxy

$$\text{SINR}_k^{\text{UL}} = \frac{p_k \left(\sum_{n=0}^{N-1} \text{SNR}_{n,k} \right)^2}{\sum_{\ell \neq k} p_{\ell} \sum_{n=0}^{N-1} \text{SNR}_{n,k} \text{SNR}_{n,\ell} + \sum_{n=0}^{N-1} \text{SNR}_{n,k}}. \quad (6)$$

Other approximations to (??) could also be legitimate proxies to drive the NN.

C. Downlink

Turning now to the downlink, with conjugate beamforming the n th AP transmits $\sum_{k=0}^{K-1} \sqrt{p_{n,k} P} h_{n,k} s_k$, where s_k is the unit-power symbol intended for user k while $p_{n,k}$ is the share of power that the n th AP devotes to such user, subject to $\sum_{k=0}^{K-1} p_{n,k} \in [0, 1]$ in order for the AP's total power not to exceed P . User k observes

$$y_k = \sum_{n=0}^{N-1} \sqrt{G_{n,k} p_{n,k} P} h_{n,k}^* s_k \quad (7)$$

$$+ \sum_{n=0}^{N-1} \sqrt{G_{n,k} P} h_{n,k}^* \sum_{\ell \neq k} \sqrt{p_{n,\ell}} h_{n,\ell} s_{\ell} + v_k, \quad (8)$$

from which its local-average SINR can be seen to be

$$\mathbb{E}[\text{sinr}_k^{\text{DL}}] = \mathbb{E} \left[\frac{\left(\sum_{n=0}^{N-1} \sqrt{\text{SNR}_{n,k} p_{n,k}} |h_{n,k}|^2 \right)^2}{\sum_{\ell \neq k} \left| \sum_{n=0}^{N-1} \sqrt{\text{SNR}_{n,k} p_{n,\ell}} h_{n,k}^* h_{n,\ell} \right|^2 + 1} \right]. \quad (9)$$

Again replacing $|h_{n,k}|^2$ and $h_{n,k}^* h_{n,\ell}$ for $\ell \neq k$ by their expected values (respectively one and zero), we obtain the proxy

$$\text{SINR}_k^{\text{DL}} = \frac{\left(\sum_{n=0}^{N-1} \sqrt{\text{SNR}_{n,k} p_{n,k}} \right)^2}{1 + \sum_{n=0}^{N-1} \text{SNR}_{n,k} \sum_{\ell \neq k} p_{n,\ell}}. \quad (10)$$

In contrast with the C-RAN uplink, and with the cellular uplink and downlink, the optimization of the transmit powers in the C-RAN downlink actually subsumes two intertwined problems:

- Power control, which amounts to adjusting the total transmit power at every AP, i.e., $\sum_{k=0}^{K-1} p_{n,k} \forall n$.
- Power allocation, which entails dividing the power of each AP among the users, i.e., setting $p_{n,k}$ for every k , subject to the total power established for the n th AP.

This enhanced problem richness is reflected by the existence of NK downlink power coefficients (as opposed to K or N) and by the aggregate (rather than individual) nature of the constraint at each AP.

III. LOSS FUNCTIONS

Owing to the different form of the constraints in uplink and downlink, the formulation of the loss functions must be individualized for either case.

A. Soft Max-Min

For the uplink, our first loss function adopts the form

$$L_{\text{MM}}^{\text{UL}} = \frac{1}{K} \left[\underbrace{\sum_{k=0}^{K-1} \exp \left(\frac{\alpha_k}{(\text{SINR}_k^{\text{UL}} + 0.01)^{0.4}} \right)}_{\text{Objective}} + \underbrace{\sum_{k=0}^{K-1} [p_k - 1]^+}_{\text{Constraints}} \right] \quad (11)$$

where $\{\alpha_k\}$ are regulating factors and $[z]^+ = \max(0, z)$. As $\{\alpha_k\}$ grow large, the loss becomes dominated by the smallest SINR and a hard max-min policy emerges. Conversely, for decreasing $\{\alpha_k\}$, this max-min behavior softens as SINRs other than the smallest one enter the optimization.

The offset 0.01 added to SINR_k prevents the loss from being dragged down by users below -20 dB and avoids numerical problems in the learning stage. The exponent 0.4 compresses the dynamic range, improving the high-SNR behavior and making the learning more stable.¹

For the downlink,

$$\begin{aligned} L_{\text{MM}}^{\text{DL}} &= \frac{1}{K} \left[\underbrace{\sum_{k=0}^{K-1} \exp \left(\frac{\alpha_k}{(\text{SINR}_k^{\text{DL}} + 0.01)^{0.4}} \right)}_{\text{Objective}} \right. \\ &\quad \left. + \underbrace{\sum_{n=0}^{N-1} \left[\sum_{k=0}^{K-1} p_{n,k} - 1 \right]^+}_{\text{Constraints}} \right]. \end{aligned} \quad (12)$$

In both uplink and downlink, the multiplier applied to the constraints, obtained from cross-validation, is unity. The ensuing probability that the solution falls outside the feasible set, and the impact thereof, are quantified in Section ??.

¹A variation of (??) that excludes this compression is considered in [?].

TABLE I
NN SETTINGS.

	Input layer	Hidden layer	Output layer
Neurons (Uplink)	200	50	$K = 12$
Neurons (Downlink)	1000	1000	$NK = 360$
Activation function	RLU	RLU	Linear
Regularization	L2 norm $\lambda = .001$	L2 norm $\lambda = .001$	L2 norm $\lambda = .001$

B. Max-Product

The maximization of $\Pi_{k=0}^{K-1} \text{SINR}_k$, or equivalently of its logarithm, amounts to the minimization of

$$L_{\text{MP}}^{\text{UL}} = \frac{1}{K} \left[\underbrace{\sum_{k=0}^{K-1} \beta_k \log_e \left(0.01 + \frac{1}{\text{SINR}_k^{\text{UL}} + 0.01} \right)}_{\text{Objective}} + \underbrace{0.1 \sum_{k=0}^{K-1} [p_k - 1]^+}_{\text{Constraints}} \right] \quad (13)$$

in the uplink, and of

$$L_{\text{MP}}^{\text{DL}} = \frac{1}{K} \left[\underbrace{\sum_{k=0}^{K-1} \beta_k \log_e \left(0.01 + \frac{1}{\text{SINR}_k^{\text{DL}} + 0.01} \right)}_{\text{Objective}} + \underbrace{0.1 \sum_{n=0}^{N-1} \left[\sum_{k=0}^{K-1} p_{n,k} - 1 \right]^+}_{\text{Constraints}} \right] \quad (14)$$

in the downlink. Here, the constraint multiplier is 0.1. When the factors $\{\beta_k\}$ are equal, the combination $\{\text{SINR}_k\}$ minimizing the above loss function offers a satisfying tradeoff between average performance and fairness. And, by regulating $\{\beta_k\}$, specific users could be afforded higher priorities.

Again, the offset shifting SINR_k by 0.01 avoids it being pulled down by users below -20 dB while a second offset added to $\frac{1}{\text{SINR}_k + 0.01}$ lessens the pull of users above 20 dB.

IV. POWER CONTROL AND POWER ALLOCATION

Connecting back with the description in [?], for the uplink the input parameter θ contains the ingredients that appear in (??), namely $\sum_{n=0}^{N-1} \text{SINR}_{n,k} \forall k$ and $\sum_{n=0}^{N-1} \text{SINR}_{n,k} \text{SINR}_{n,\ell} \forall k, \ell$, while the optimization vector x is made of the power coefficients $\{p_k\}$.² For the downlink, in turn, $\theta = \{\text{SINR}_{n,k}\}$ and $x = \{p_{n,k}\}$. The loss function is given by L_{MM} or L_{MP} , as appropriate.

²The uplink learning could be expedited by restricting the input parameter to $\sum_{n=0}^{N-1} \text{SINR}_{n,k} \forall k$, i.e., the ingredients in the numerator of (??), disregarding those in the denominator. This alternative, explored in [?], yields an only slightly diminished performance. Relevant to this alternative, as an additional baseline, is the fractional power control for C-RANs, which relies on the same inputs [?].

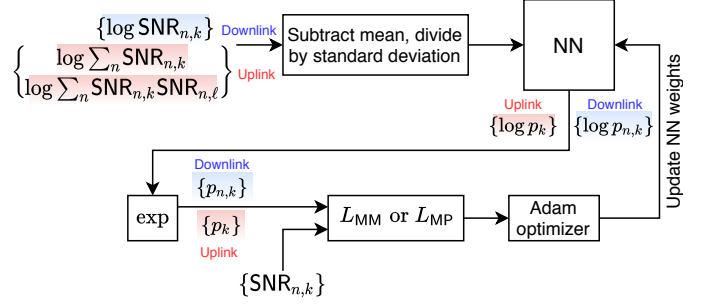


Fig. 1. Learning pipeline.

A. Learning Stage

The learning pipelines for both uplink and downlink are combined into Fig. ?? while the NN parameters are summarized in Table ?. The input parameters are first converted to log scale and subsequently rendered zero-mean and unit variance. The processing then properly starts with feature extraction by an input layer equipped with rectified linear unit (RLU) activation functions. Afterwards, a hidden layer processes the data also via RLUs, and an output layer with linear activation functions generates power coefficients in log scale; this guarantees positive outputs and averts numerical problems. From the SNRs and the corresponding NN outputs, the loss function of choice is quantified and an Adam optimizer is applied to minimize it. To prevent oscillations around local optima, the learning rate is reduced gradually from 0.001 down to 0.0001. And, to avoid overfitting, L2-norm regularization is employed in conjunction with the Adam optimizer: a portion $\lambda = 0.001$ of the L2 norm of the weights is added to the loss.

To streamline the learning, rather than a single large database, 50 (for the uplink) and 500 (for the downlink) databases of 12800 system realizations are generated and, over each, 200 updates of the NN weights take place; each update relies on a randomly selected batch of 128 realizations. The initialization is also random.

While the uplink learning curve is similar to its cellular counterpart [?] and not shown for the sake of brevity, the downlink learning curve is different (see Fig. ??). Besides settling more slowly due to the larger number of power coefficients, it exhibits two stages that are not distinguishable in the uplink and that can be roughly mapped to the constraints and to the objective, the constituent parts of the loss function.

B. Performance Evaluation

For the C-RAN uplink, the power control problem can be cast in convex form under both of the considered loss functions. Therefore, performance benchmarks can be produced with an off-the-shelf convex solver.

For the C-RAN downlink, conversely, the feasible sets are convex but the loss functions L_{MM} and L_{MP} cannot be cast as convex in $\{p_{n,k}\}$. This is, in essence, because any increase in a user's transmit power simultaneously increases the interference to other users and detracts from their transmit powers. No convex-solver benchmarks are thus available in general. Only in the hard max-min limit (L_{MM} with $\alpha_k \rightarrow \infty$) does the

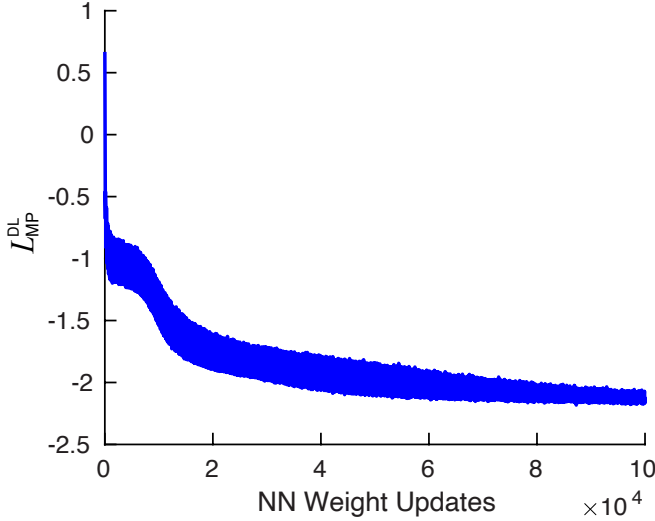


Fig. 2. Downlink max-product learning curve with $\beta_k = 1$.

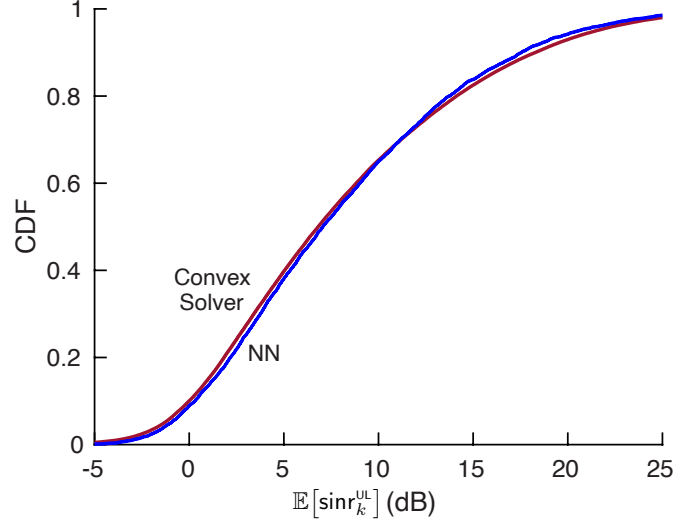


Fig. 4. CDF of $\mathbb{E}[\text{sinr}_k^{\text{UL}}]$ for max-product ($\beta_k = 1$): NN vs convex solver.

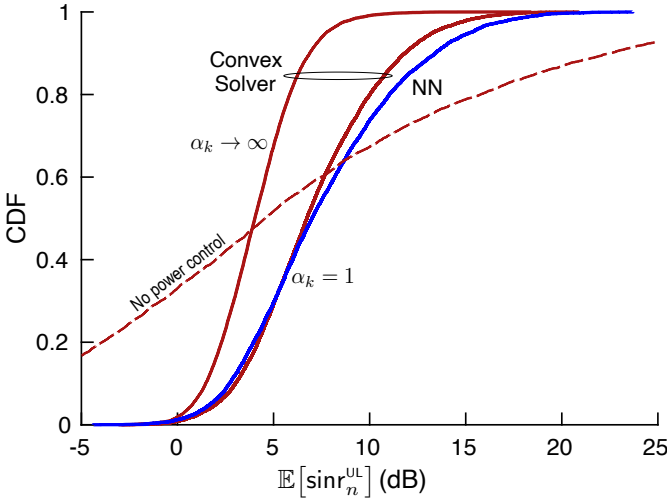


Fig. 3. CDF of $\mathbb{E}[\text{sinr}_k^{\text{UL}}]$ for soft max-min: NN vs convex solver ($\alpha_k = 1$). Also shown is the baseline performance without power control, and the hard max-min performance ($\alpha_k \rightarrow \infty$).

downlink power optimization become quasi-convex in $\{p_{n,k}\}$ and can be solved through a tedious bisection search whose steps entail a sequence of convex feasibility optimizations [?].

We consider $N = 30$ APs and $K = 12$ users. The performance is evaluated via the cumulative distribution function (CDF) of local-average SINRs over the ensemble of user and AP positions. The NN and the convex-solver benchmarks (when available) are driven by the proxies in (??) and (??), while the performance with the obtained powers are asserted by means of the actual local-average SINRs in (??) and (??).

The uplink performance for the soft max-min loss function, shown in Fig. ??, prompts the following remarks:

- Power control is instrumental to avoid major disparities among users.
- For given $\{\alpha_n\}$ (unity in this case), the agreement with the convex solver is excellent. The NN slightly favors high-SNR users, falling shy of the convex benchmark in

terms of the soft max-min behavior.

- Hardening the loss function is ill-advised, as that uniformly worsens the performance; in the limit, the entire system is conditioned by its worst situated user.

For the max-product loss function, Fig. ?? again reveals an excellent match between the NN and the convex solver.

In the downlink, where convex solvers do not in general provide optimality guarantees, we invoke as baseline

$$p_{n,k} = \frac{\text{SNR}_{n,k}}{\sum_{k=0}^{K-1} \text{SNR}_{n,k}} \quad n = 0, \dots, N-1 \quad (15)$$

whereby every APs transmits its complete power and the share that the n th AP allocates to user k is proportional to the strength of the corresponding link. This baseline, rather standard as a complement to matched-filter beamforming, is exhibited in Fig. ?? alongside the NN-based performance under both soft max-min and max-product loss functions. In both cases, the NN outperforms markedly the baseline, with more emphasis on fairness—a steeper CDF—in the soft max-min case and less emphasis in the max-product case.

With very small probability, the objective part of the loss function could push the solution to violate one or several constraints. As shown by Fig. ??, the probability that the transmit power of an AP exceeds its maximum value is below 2% in the soft max-min case with $\alpha_k = 1$, and the impact of trimming down these power spillovers is minute (see Fig. ??).

C. Computational Cost

As in [?] and [?], we invoke, as a measure of the computational cost, the running time on a CPU-based platform.

Each NN inference is in essence a matrix multiplication, over two (in the uplink) and three (in the downlink) orders of magnitude faster than a convex solver for our 30-AP system.

As far as the learning, which needs to take place upon changes in the C-RAN or the environment, it is of interest to compare the learning time of our unsupervised NN with that of a supervised NN of the same dimensions [?], [?],

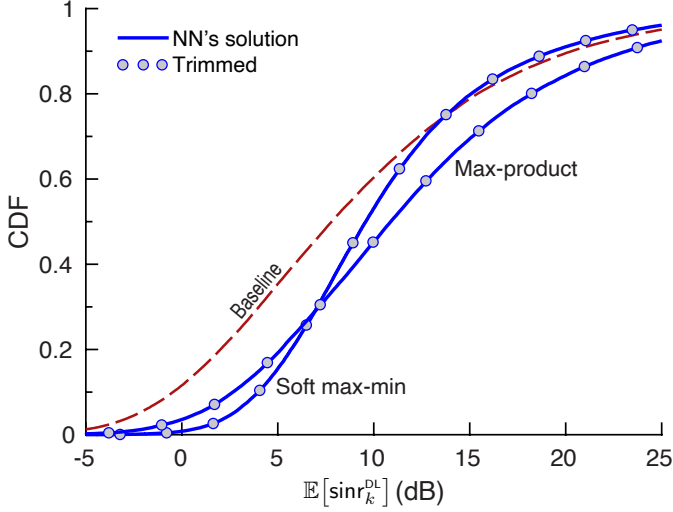


Fig. 5. CDF of $\mathbb{E}[\text{sinr}_k^{\text{DL}}]$: NN (for soft max-min with $\alpha_k = 1$ and for max-product with $\beta_k = 1$) vs the baseline in (??). Also shown, in circles, is the modified performance when the powers are trimmed such that no constraint is violated.

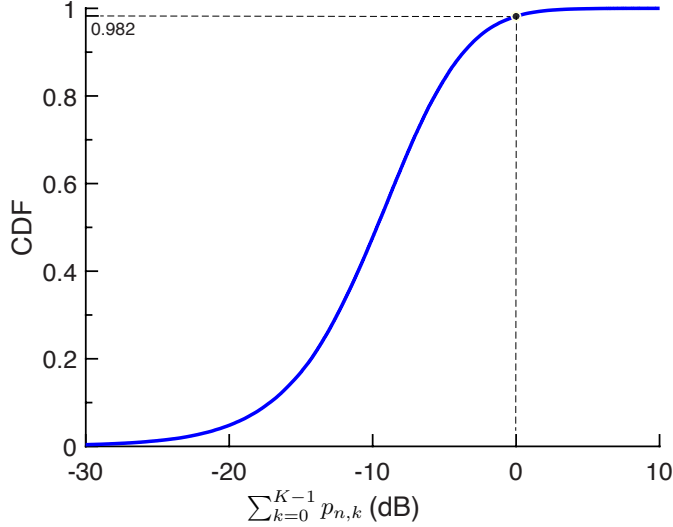


Fig. 6. CDF of $\sum_{k=0}^{K-1} p_{n,k}$ as produced by the NN for the soft max-min loss function ($\alpha_n = 1$).

For the uplink, where the considered objectives lead to convex formulations, the NN performance matches that of convex solvers while incurring orders of magnitude less computational cost. For the downlink, where those objectives cannot be tackled with certainty by convex solvers, the NN itself becomes a source of benchmark results. On that note, it would be of interest to tackle other nonconvex loss functions, for instance those related to energy efficiency with minimum performance guarantees at the users [?], [?].

The NN-based power optimizations continue to be effective if the assumption of known fading coefficients is overcome, and the fading coefficients are explicitly estimated on the basis of pilot transmission. Some results under that proviso are presented in [?], [?] for uplink and downlink, respectively.

The Python code developed to produce the results in this letter is openly available [?].

[?], [?]. While the training effort per parameter sample is essentially the same, a supervised NN requires producing those samples by solving the corresponding optimizations in the first place, again resulting in the two-to-three-order-of-magnitude advantage for the example at hand. (Allowing the NN to train on a more apt GPU-based platform, the superiority of unsupervised training increases by yet another order of magnitude.) And, for nonconvex objectives, it is challenging to generate those training samples in the first place.

V. SUMMARY

The unsupervised learning approach to parametric optimization described in [?] has been shown to be very effective for the problems of power control and power allocation in cell-free C-RANs—even with the NN trained with a proxy to the actual objective, the local-average SINR.