Cost aware Inference for IoT Devices

Pengkai Zhu
Boston University

Durmus Alp Emre Acar
Boston University

Feng Nan
Amazon Inc.

Prateek Jain
Microsoft Research

Venkatesh Saligrama
Boston University

Abstract

Networked embedded devices (IoTs) of limited CPU, memory and power resources are revolutionizing data gathering, remote monitoring and planning in many consumer and business applications. Nevertheless, resource limitations place a significant burden on their service life and operation, warranting cost-aware methods that are capable of distributively screening redundancies in device information and transmitting informative data. We propose to train a decentralized gated network that, given an observed instance at test-time, allows for activation of select devices to transmit information to a central node, which then performs inference. We analyze our proposed gradient descent algorithm for Gaussian features and establish convergence guarantees under good initialization. We conduct experiments on a number of real-world datasets arising in IoT applications and show that our model results in over 1.5X service life with negligible accuracy degradation relative to a performance achievable by a neural network.

1 INTRODUCTION

We introduce a novel distributed inference problem for energy-limited IoT devices, which offer exciting applications for machine learning. IoT devices augmented with sensors are increasingly deployed in various applications including consumer, business, infrastructure (Perera et al., 2015) and wearable technology[1] (Latré et al., 2011). These devices are capable of distributively gathering data and transmitting relevant information for system-wide monitoring and control. However, IoT devices are CPU, memory, power and bandwidth limited, which place a significant burden on their service life and operation. Hence, resource-aware distributed inference are critical for viability of such systems.

Several architectures have been proposed in this context (Viswanathan and Varshney, 1997). In the centralized architecture, sensing devices continuously gather data independently, and transmit it to a central node, where the data is aggregated and processed to perform inference. For low-powered IoT devices, transmit energy mostly dominates all other forms of battery usage (Halgamuge et al., 2009; Latré et al., 2011). So, while this architecture is simple, it can be wasteful and can significantly deplete power of each device leading to short service-life.

While many architectures such as serial, parallel and adhoc have been proposed (Viswanathan and Varshney, 1997), our focus is on decentralized architecture that transmits data only when necessary to the fusion center. Our communication efficient architecture exploits the following two observations: a) much of the gathered data either contains no information or is redundant, b) depending on the type of activity, some devices are more informative than others, and it is often sufficient to receive data from the most suitable devices rather than all the devices.

Consequently, it makes sense for a device to transmit information *only when* its data is useful for inference relative other devices in the network (Appadwedula et al., 2008). The fundamental challenge is that a device must not only determine whether it has useful information, but also deduce–without the benefit of communication—whether the data is redundant because there is another device with better information that should transmit.

A central challenge addressed in our paper is for each

[1]Sensors are embedded or implanted in various parts of the body for activity monitoring.

Proceedings of the 22[nd] International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89.

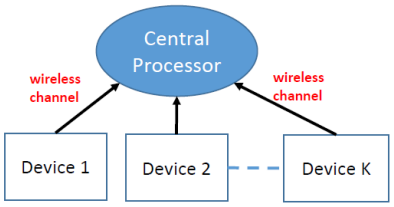


Figure 1: Cost-aware decentralized prediction with IoT devices: Gating functions at each device screen their local data to eliminate redundancies in information among different devices to conserve power while ensuring informativity for effective inference.

device to learn to autonomously evaluate, whether its data is critical for inference, and based on this evaluation, transmit information to a fusion center. The fusion center then aggregates the received information and makes a prediction. Naturally, our setup precludes data sharing among devices before transmission. In our framework, a device faces a dilemma, namely, to determine whether or not to transmit its data agnostic to another device's data.

This problem has been well-studied in the context of decentralized binary hypothesis testing for known probabilistic models (Tsitsiklis and Athans, 1984; Rago et al., 1996; Appadwedula et al., 2008). While the general problem is considered intractable, under restrictive assumptions of conditional independence (Rago et al., 1996; Appadwedula et al., 2008), several works have explicitly characterized the optimal decentralized protocols for different scenarios. Nevertheless, the knowledge of probabilistic models together with the assumption of conditional independence can be quite restrictive limiting their utility.

Motivated by these reasons we propose a novel learning framework to ensure energy and communication efficient inference. The benefit of a learning perspective in contrast to optimal decentralized decision theory is three fold. First, it allows for explicit parameterization of decision functions, that can be empirically optimized without the knowledge of probabilistic models. Second, we can account for a wide-variety of low to high-complexity models as well as a wide-variety of inference problems that reflect system requirements. Third, decentralization allows the model to consume less energy which makes significant improvement in lifetime of IoT devices. We learn a distributed prediction model for multi-class classification, which is agnostic to knowledge of probabilistic models. In our model, a data point x is specified by local observation x_k for device k. Each device is equipped with a gating function, $g_k(\cdot)$, which determines whether or not information from the device is to be transmitted. A device transmitting information, compresses its local data, x_k, and outputs $h_k(x_k)$. A central node aggregates received messages and outputs a prediction $f(\sum_{k=1}^{K} h_k(x_k)g_k(x_k))$. The communication savings occurs when edge devices cease transmission ($g_k(x_k) = 0$).

Our goal is to train prediction functions to minimize test-time prediction loss while limiting the number of participating devices averaged over the data points. We propose a non-convex objective that balances training error with device activation to jointly learn gating and compression functions by means of gradient descent. We analyze the effectiveness of our gradient descent scheme to optimize our non-convex objective. Motivated by our experimental results, we consider a simple Gaussian realizable setting, we show that the SGD algorithm is guaranteed to converge to the optimal set of parameters if we start with a sufficiently good initialization. While the Guassian setting is somewhat idealistic, the considered problems are quite challenging, and existing polynomial time convergence results in this domain use similar assumptions (Ge et al., 2017).

We conduct experiments on a number of real-world datasets. Our key finding is that our trained model is capable of adaptively gating instances that are either redundant or non-informative and significantly outperforms other methods.

2 RELATED WORK

The problem we study is related to the distributed detection literature where the probability distributions are generally assumed known. In particular, (Rago et al., 1996) considered censoring sensors. The idea is that sensors transmit their observations to the fusion center only if they are deemed "informative". Under the assumption that the sensor observations are conditionally independent given the hypothesis and under a communication rate constraint, the authors showed that each sensor should transmit if and only if its local likelihood ratio falls outside a single interval. (Appadwedula et al., 2008) extend the censoring sensors framework and eliminate the need for joint optimization of the censoring regions. (Tsitsiklis and Athans, 1984) showed that finding the optimal censoring function in distributed detection problems is intractable without assuming conditional independence. While these studies inform us on the optimal strategies with known distributions, we study the learning setting where such distributions are unknown.

The learning problem of distributed prediction has been studied by (Yang et al., 2009, 2008) in the context of human action recognition using wearable motion sensors. The goal was to achieve high classification accuracy while reducing communication cost between the sensors and a fusion center. A sparse coding approach was used to represent each data point in terms of a sparse subset of samples belonging to each action class. Each sensor computes the sparse representation and makes a prediction based on how well the data point

is represented by any class. Finally, the fusion center predicts based on a majority vote.

Our model is also related to the sparsely-gated mixture-of-experts model (Shazeer et al., 2017), where a subset of experts is activated to predict any given data point. The goal there is to scale up the model capacity yet retain computational efficiency through a centralized gating function. In contrast, our model is motivated by distributed sensor network and utilizes distributed gating functions to improve communication efficiency.

Our model of gating function multiplying a local compression function leads to inherent non-convexity similar to that in the low-rank matrix completion model (Jain et al., 2013; Si et al., 2016) as well as the neural network model (Zhong et al., 2017). Inspired by successes in analyzing above mentioned problems (Zhong et al., 2017; Du et al., 2017), we provide local convergence guarantees for our model, which can be seen as a strict generalization of the low-rank matrix completion in the non-linear setting as well as one hidden layer neural network model. In particular, to prove local strong convexity of our optimization problem, we develop novel tools based on the Schur product theorem (Horn and Johnson, 1991).

More broadly, our work belongs to an active research area of resource-constrained machine learning. In particular, prediction costs in terms of feature acquisition (Xu et al., 2012; Nan et al., 2016; Nan and Saligrama, 2017; Trapeznikov and Saligrama, 2013; Wang et al., 2015), computation (Bolukbasi et al., 2017b; Wang et al., 2017, 2014; Bolukbasi et al., 2017a) and memory (Kumar et al., 2017; Gupta et al., 2017) have been studied. Our work focuses on reducing the communication cost in an IoT setting.

3 PROBLEM FORMULATION

We propose a learning based framework of the canonical decentralized detection problem (Rago et al., 1996), which unlike our setting, consider a binary hypothesis problem with known probabilistic models.

We consider an M-class classification problem with labels taking values in an index set $\mathcal{Y}$. A network of K edge devices sense their environment and make decisions as to whether to transmit the sensed data to a fusion center. The task of the fusion center is to aggregate information from the edge devices and output a prediction of the class label. We assume that the devices operate in a resource-limited environment and must trade-off utility of local sensed information with available resources.

We let X_k, $k = 1, 2, \ldots, K$ denote continuous random variables observed at each of the K devices. Following convention, lower case x_k denotes realization of random variable $X_k \in \mathcal{X}_k \subseteq \Re^D$. Let $X = [X_1, X_2, \ldots, X_K]^T \in \mathcal{X} \subseteq \Re^{D\times K}$ denote the random matrix of data from all devices and P the joint distribution on the product space $\mathcal{X} \times \mathcal{Y}$. Associated with each device is a gating function, $g_k : \mathcal{X}_k \to \Re^+$, which allows for censoring device k's data transmission. The cost of each transmission is a device-independent constant depending only on whether a transmission occurs[2]. So, in this paper, we assume that $g_k(x_k) > 0$ costs the same as $g_k(x_k) \neq 0$; therefore, our primary goal is to sparsify such activations. For an input, x_k, device k transmits when $g_k(x_k) > 0$ and it transmits a compressed statistic, using a compression function $h_k : \mathcal{X}_k \to \Re^d$. The fusion center fuses received observations, $f(x) \triangleq f((h_1(x_1), g_1(x_1)), \ldots, (h_K(x_K), g_K(x_K))) \in \Re^M$ and outputs a predicted label by choosing the component corresponding to the maximum score[3].

Our goal is to minimize the sum of expected loss and expected device activations. Namely, we consider a loss function $L(f(x), y)$ that penalizes the error between predicted output and the true label together with the average number of activations:

$$\min_{f,(h_1,g_1),\ldots,(h_K,g_K)} \mathbb{E}_P L(f((h_1, g_1), \ldots, (h_K, g_K)), Y) + \lambda \mathbb{E}_P \sum_{k=1}^{K} |g_k(X_k)|_0$$

where λ is a trade-off parameter that controls the activation budget and $|\cdot|_0$ is the ℓ_0 norm. As is standard practice, we choose a suitable parameter λ to meet desired average activation constraint, $\sum_{k=1}^{K} \mathbb{E}_P[\|g_k(X_k)\|_0] \leq B$.

Empirical Risk: In this paper we consider the case where full-training data is available for all devices along with ground-truth annotations. We are given n training data instances: $(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})$ with each data point comprised of features from different devices: $x^{(i)} = [x_1^{(i)}, \ldots, x_K^{(i)}]$, and label/response $y^{(i)}$. For simplicity we assume feature dimension, D, to be identical across all devices. Our empirical risk objective is to optimize the following empirical objective:

$$\min_{h_0,\ldots,h_K,g_1,\ldots,g_K} \sum_{i=1}^{n} L\left(f((h_1, g_1), \ldots, (h_K, g_K)), y^{(i)}\right) + \lambda \sum_{i=1}^{n} \sum_{k=1}^{K} |g_k(x_k)|_0 \quad (3.0.1)$$

[2] In low-power wireless devices, *transmit energy* is often dominated by activation and less so by the actual signal amplitude or feature output dimension particularly in low-rate scenarios.

[3] As such there is no loss of generality between the gating/prediction decomposition and a generalized prediction function that combines the two operations.

3.1 Model Parameterization

Our proposed architecture is a prediction system composed of a compression function h_k and gating function g_k. Both h_k and g_k are linear functions with g_k passed through a non-linear activation unit; the centralized prediction function is then the sum of the compression functions modulated by the gating functions from all of the sensors[4]. Mathematically, for each instance, x:

$$f(x_1, x_2, \ldots, x_k) = \psi(\sum_{k=1}^{K} h_k(x_k) g_k(x_k) + b), \quad (3.1.1)$$

where $h_k(x_k) = v_k^T x_k + a_k \in \Re^d$, and $g_k(x_k) = \phi(w_k^T x_k + b_k) \in \Re$ is the gating output of the kth sensor. The function $\phi(\cdot)$ is chosen as a ReLU unit or a sigmoid for the purpose of optimization. The function $\psi(\cdot)$ predicts a label based on the received gated outputs. Since we do not impose resource constraints, models of varying complexity are allowed.

The communication savings in this setup occurs when $g_k(x_k^{(i)}) = 0$ and the device does not need to transmit. We relax the ℓ_0 loss in Eq. 3.0.1 by an ℓ_1 function. The ℓ_1 objective reduces to a summation of activation functions due to non-negativity of gating output.

Dimensionality of Compression Output: We examine different choices. For scenarios where feature dimension is a significant factor of energy budget, we require a low-dimensional output and choose $d = M$, which is the number of class labels. This situation can be viewed as each device outputting local predictions, which are then fused at the fusion center. In other scenarios where activation primarily contributes to the energy budget we map different device feature vectors to independent parts of a sufficiently large d-dimensional feature space, so that the fusion center sees concatenated sequence of ungated device feature vectors. We will explore these different choices in our experiments.

Our goal is to minimize prediction error while limiting the number of "participating" devices averaged over the data points. Mathematically, we would like to solve the following optimization problem:

$$\min_{h_0, h_1, \ldots, h_K, g_1, \ldots, g_K} \quad \sum_{i=1}^{n} L\left(\sum_k h_k(x_k^i) g_k(x_k^i), y^{(i)}\right) + \lambda \sum_{i=1}^{n} \sum_{k=1}^{K} g_k(x_k) \quad \text{(OPT)}$$

where L is the loss function and λ is a multiplier to tradeoff the accuracy with communication budget. For classification, we typically use a softmax function and score accuracy loss by means of cross-entropy, although other types of losses could also be considered. We solve (OPT) by means of stochastic gradient descent (SGD).

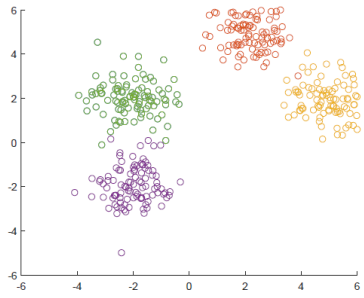


Figure 2: Demonstration with two devices, with measurements along the different axis, and various forms of gating for binary classification on a four Gaussian mixture. Example A: positive: yellow/purple; Negative: red/green. No linear classifier can separate data. A gated classifier $f(x_1, x_2) = x_1 1_{x_1 \geq 0} - x_2$ achieves 100% accuracy with 25% reduced activation. Example B: positive- yellow/green; Negative- red/purple. Vector gating with $f(x_1, x_2) = x_1 1_{x_1 \geq 0} - 2x_2 1_{x_2 \geq 0} + x_2$ is now required.

As described earlier we tune λ to obtain the activation functions that meet a desired budget. For a prespecified device usage, we sweep potential values using cross validation to determine the suitable λ. This is a standard technique in many algorithms such as Lasso (Tibshirani, 1996).

Vector-Valued Gating: We generalize our model Eq. 3.1.1 to express complex compression functions (see Fig. 2). To do this we consider a family of piecewise linear compression functions, $h_{ij}(x_i), j = 1, 2, \ldots, J$ and construct gating vectors $g_i(x_i) = [g_{i1}(x_i), \ldots, g_{iJ}(x_i)]$ for device i. The compression function $h_i(x_i) = \sum_{j \in \mathcal{J}} h_{ij}(x_i) g_{ij}(x_i)$ is now a superposition of linear compression functions weighted by the gated output. Activation is enforced through ℓ_1/ℓ_∞ penalty.

3.2 Exactness of Model with Naive Bayes

To build intuition we will show that Eq. 3.1.1 has the form of the optimal classifier under the Naive-Bayes assumption, namely, when the device features are independent when conditioned on class label: $P(x_1, x_2, \ldots, x_K \mid y) = \prod_{i=1}^{K} P(x_i \mid y)$. This is a simplifying assumption that has been conventionally adopted in the context of decentralized detection theory as a compromise between accuracy and mathematical tractability[5]. In this setting it is sufficient to consider local-likelihoods for the device-wise feature mappings. The optimal gating is a function of

[4] While linear model can be kernelized to incorporate highly complex decision boundaries and activation functions, the kernel methods suffer from substantial computational scaling.

[5] For instance, this assumption is satisfied for a M-component Gaussian mixture. Each component corresponds to a class-label and is described by a mean-vector and covariance equal to identity.

just the local-likelihood at each device. In particular, we let $\ell_{i,j}(x_i) = \log P(x_i \mid y = j)$, $j \in [M]$ and $h_i(x_i) = [\ell_{i,1}(x_i), \ell_{i2}(x_i), \ldots, \ell_{i,M}(x_i)]^T$.

Lemma 3.1. *The optimal classifier under the Naive Bayes assumption has the form:* $f(x_1, x_2, \ldots x_K) = \sum_{i=1}^{K} h_i(x_i) g_i(h_i(x_i)) + b$.

For Gaussian random variables the local likelihood ratios are linear functions of features leading to optimal classifier taking the form: $f(x) = \sum_{k=1}^{K} (\nu_k^T x_k + b_k) g_k(\nu_k^T x_k + b_k) + d$ for some fixed constant, d, vectors b_k and matrices $\nu_k \in \Re^{M \times D}$. Mathematically, Eq. 3.1.1 is a direct extension with arbitrary weightings matrices. (Appadwedula et al., 2008) show that for specific communication constraints, the gating function is a two-sided threshold on the likelihood ratio but is intractable in general even under conditional independence. Nevertheless, as seen from Fig. 2, the proposed parameterizations are capable of learning rules even when independence assumptions are not satisfied. This motivates our discriminative learning based approach.

4 ANALYSIS

We use SGD to optimize a non-convex objective function given by (OPT), and hence in general, the algorithm can converge to a significantly suboptimal stationary point. However, in our simulations the algorithm recovers nearly optimal solution in a few iterations, suggesting that the problem is "easy" for "nice" data. In this section, we formalize this intuition by studying problem (OPT) in a simple realizable setting with Gaussian data. That is, suppose there exists an optimal set of parameters $\Theta^* = (V^*, W^*, a^*, b^*)$ that leads to the response variable; $V^* = [v_1^*, \ldots, v_K^*] \in \Re^{M \times K}$, $W^* = [w_1^*, \ldots, w_K^*] \in \Re^{M \times K}$, $a^* = [a_1, \ldots, a_K] \in \Re^K$, and $b^* = [b_1^*, \ldots, b_K^*] \in \Re^K$. That is,

$$y(x) = \sum_{k=1}^{K} (x_k^T v_k^* + a_k^*) \phi(x_k^T w_k^* + b_k^*), \qquad (4.0.1)$$

where $\phi : \Re \to \Re$ is a Sigmoid, ReLU activation etc.

Also, let each data point $x^{(i)} \sim \mathcal{D}$ where $\mathcal{D}$ is a Gaussian distribution. Without loss of generality (Wlog), we can assume that $\mathcal{D}$ is 0-mean spherical normal distribution. Considering squared loss function, the population and the empirical risk minimization problems is:

$$L(V, W, a, b) = E_{x \sim \mathcal{D}} (y - \sum_{k=1}^{K} (x_k^T v_k + a_k) \phi(x_k^T w_k + b_k))^2, \qquad (4.0.2)$$

$$\widehat{L}(V, W, a, b) = \frac{1}{n} \sum_{i=1}^{n} (y^{(i)} - \sum_{k=1}^{K} (v_k^T x_k^{(i)} + a_k) \phi(w_k^T x_k^{(i)} + b_k))^2.$$

To understand how gradient descent style algorithms perform for this problem, we first show that while the parameterization of the objective function is non-convex, it is a *strongly convex* objective function in a closed set containing the global optima.

Theorem 4.1. *Let $\mathcal{D} = N(0, I)$ and let data points x^s, $1 \le s \le n$ be generated i.i.d. from $\mathcal{D}$ and the response y^s is given by (4.0.1) with optimal parameters $(V^*, W^*, a^* = 1, b^* = 1)$ s.t. $\|w_i^*\| = 1$, $\|v_i^*\| = 1$. Also, let the activation function ϕ be such that $|\phi(z)| \le B|z|^q$, $|\phi(z) - \phi(z')| \le L_0|z - z'|$, $\phi'(z) \le L_1|z|^p$, $\phi''(z) \le L_2$ for some global constants B, L_0, L_1, L_2, p, q. Furthermore, let $n \ge CK^2 d^2 \log^C d$.*

Then, for any fixed $\Theta = (V, W, a = 1, b = 1)$ such that $\|\Theta - \Theta^*\| \le \frac{C_0}{\sqrt{K}}$, the following holds (w.p. $\ge 1 - \frac{10K^2}{n^{100}}$):

$$C_1 I \preceq \widehat{H}(\Theta) \preceq C_2 I$$

where $\widehat{H}(\Theta)$ is the Hessian of empirical loss (4.0.2) evaluated at Θ. C_0, C, C_1, C_2 are global constants.

Following convention we bound the error as a polynomial factor n^{-100} for concreteness.

Remark: Note that our result requires new analysis and new techniques. For instance, the most closely related setting (Zhong et al., 2017) (see Sec. 4) considers the model $y = \sum_i a_i \phi(w_i x)$. Only w_i is the optimization parameter; a_i assumed constant in their algorithm analysis, which greatly simplifies their proof. Furthermore, x is shared across all nodes. In contrast our model is $y = \sum_i (v_i x_i + a_i) \phi(w_i x_i + b_i)$ where both v_i and w_i are optimization parameters. This objective is difficult even when $\phi(\cdot)$ is linear, since it would be a bilinear expression. Technically, we need to deal with the fact that our Hessian has cross-product terms due to multiplication of v_i and w_i. In contrast (Zhang et al., 2017) does not, which they leverage to simplify analysis. We develop a new method of analysis based on on Schur-Product theorem in our proof of Theorem 4.1, which can itself be of independent interest.

We now leverage Theorem 4.1 and standard arguments to show that the standard gradient descent algorithm converges to the global optimal in small number of iterations. To do this in Theorem 4.2 we show strong convexity for a fixed parameter Θ^t along the line joining Θ^t and Θ^* so as to leverage the analysis of standard SGD analysis, which only requires this fact.

Theorem 4.2. *Consider the setting of Theorem 4.1. Then gradient descent method which samples new set of points $(x^s, y^s), 1 \le s \le n$ at each step and is initialized with Θ_0 s.t. $\|\Theta^* - \Theta_0\| \le \frac{C_0}{\sqrt{K}}$, converges to a parameter Θ after T-iterations such that: $\|\Theta - \Theta^*\| \le C_4^{-C_5 T}$, where $0 \le C_4 \le 1, C_5 > 0$ are global constants.*

Note that similar results have been shown to hold for

one hidden layer neural networks (Zhong et al., 2017). However, our prediction function is significantly more complicated than a single hidden layer network and hence, the existing results do not apply. In particular, our Hessian structure is more complex and has several more interdependent blocks and requires novel analysis that is based on the Schur product theorem (Horn and Johnson, 1991).

5 EMPIRICAL RESULTS

5.1 Synthetic dataset

We first illustrate using synthetic data that SGD can recover the ground truth parameters from a neighborhood around them. The purpose is to empirically validate Theorem 4.2 and so no budget is enforced.

Data generation We generate $n = 2000$ sample data points $x^{(i)}, i = 1, \ldots, n$, each consists of features from $K = 2$ devices and each device has 10 dimensional features: $x_k^{(i)} \in \Re^{10}, k = 1, 2$. The features are generated from an i.i.d. standard normal distribution. We then generate the ground truth parameters V^*, W^*, b^* with each element drawn from i.i.d. standard normal distribution. Finally, we generate the regression targets based on: $y^{(i)} = \sum_{k=1}^{K} v_k^* x_k^{(i)} \phi(w_k^* x_k^{(i)} + b_k^*)$, where we have a choice of using ReLU/Sigmoid function for ϕ.

Algorithm We perform alternating minimization with SGD of the following objective.

$$\min_{v,w,b} \quad \sum_{i=1}^{n} \left(\sum_{k=1}^{K} v_k x_k^{(i)} \phi(w_k x_k^{(i)} + b_k) - y^{(i)} \right)^2 .$$

We first fix W, b, perform 50 steps of gradient descend for V; then fix V and perform the same number of gradient descend steps for W, b; repeat for 500 times. **Initialization** We iteratively minimize the objective with different initialization of V, W, b by simply adding a scaled version of Gaussian noise to the ground truth parameters: $V_0 = V^* + \text{NoiseLevel} \times N_V$, where NoiseLevel is a scalar value controlling the magnitude of the noise and N_V is drawn from standard normal distribution with the same dimension as V^*. Likewise we initialize W_0 and b_0.

We compare the rate at which the parameters converge to the ground truth values with ReLU and sigmoid activations under different initialization noise levels. We repeat each experiment for 10 times and report the mean. We report both the loss as well as the cosine distance of W and V from their ground truth values. The NoiseLevel in (a) and (b) of Figure 3 are set to 1 while it is set to 4 in (c) and (d). We observe that the when the initial parameter is close to the ground truth they converge very quickly; this is consistent with our analysis in Theorem 4.2. We also notice that ReLU activation leads to faster convergence than the sigmoid.

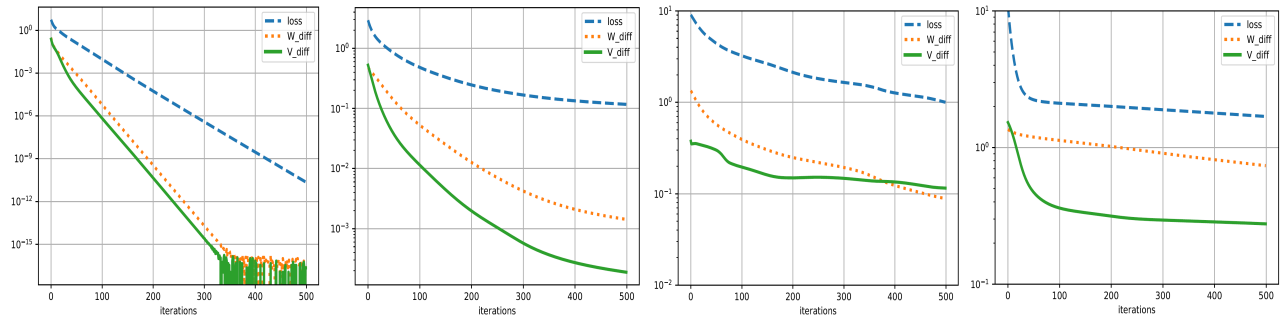
Table 1: Summary of Datasets: number of data points n for training and testing, sensor types, number of sensor units K, feature dimension D measured by each sensor and number of classes M. (*: acm = accelerometer, mgm = magnetometer, gyr = gyroscope, geo = geomagnetic)

Dataset	n(train/test)	sensor type*	K	D	M
DailySports	6080/3040	acm, gyr, mgm	5	1125	19
WARD	44735/20000	acm, gyr	5	40	13
AReM	28160/14079	received signal strength	3	2	7
DukeReID	2828/707	camera	8	2048	707
WFRobot	3638/1818	ultrasound	4	6	4
HAR	6866/3433	acm, gyr	5	119	6
GLEAM	429/211	linear acceleration gravity, gyr, acm rotation vector, geo	6	18	7
GAS	9274/4636	chemical	16	8	6

5.2 Real world datasets

We test our method on eight real world datasets that appear in diverse IoT contexts including multi-sensor data fusion—AReM (Palumbo et al., 2016), activity monitoring and recognition with wearable sensors—WARD (Yang et al., 2009), DailySports (Altun et al., 2010), smart phone based monitoring—HAR (Anguita et al., 2013), Google Glass—GLEAM (Rahman et al., 2015), multi-camera surveillance—DukeREID (Gou et al., 2017), robotic navigation—WFRobot (Dheeru and Karra Taniskidou, 2017), and gas sensor array for chemical detection—GAS (Vergara et al., 2012)g. Apart from GLEAM, HAR, WFROBOT datasets, all of the other datasets have non-collocated sensors and are equipped with individual transmitters. Our results for these are based on equipping each sensor with a transmitter. We summarize the datasets in Table 1.

Baseline Non-Adaptive Methods: We benchmark energy gains against best performance obtainable with two strong *non-adaptive* models.
Unconstrained Baseline Model (BaseM): In this baseline our objective is to maximize accuracy without the constraint on battery usage. This means that each device can send its feature vector independently to the fusion center. The fusion center takes the concatenated set of device feature vectors and outputs a prediction. We train a fully connected neural network with several hidden layers at the fusion center to maximize accuracy. Our experiments indicate that for the datasets under consideration, a 3 hidden layer NN achieve top-accuracy, and additional layers tend to overtrain. Performance on test data is then tabulated.
Limited Battery Baseline Model (LimBaseM): In this case we combinatorially search over the best k out of K available devices that achieve the highest accuracy for each k with the NN architecture of BaseM. Feature vectors from these k-devices are then concatenated at the fusion center and the NN is trained. This approach



(a) ReLU, NoiseLevel 1 (b) Sigmoid, NoiseLevel 1 (c) ReLU, NoiseLevel 4 (d) Sigmoid, NoiseLevel 4

Figure 3: Parameter recovery under ReLU and Sigmoid, under different initialization noise levels.

is non-adaptive in the sense that for a fixed cost level, or k, the same subset of sensors are active for all data points. In contrast, our method can activate different sensors for different data points adaptively thanks to the gating function. Note that because k sensors must be selected, there is a possibility that accuracy can drop due to noisy features as in DukeReiD dataset.

Baseline Adaptive Method: *Greedy Entropy Model (GreedyEntM):* A naive adaptive scheme that often works well is based on computing a local prediction for a device followed by gating instances that have entropy/margin of the predicted labels larger than a threshold (Bolukbasi et al., 2017b). In this method, the prediction probability from a device is transmitted if its entropy is smaller than a threshold. The fusion center sums the transmitted local prediction probabilities and feeds it into the network to make final prediction. The threshold is chosen based on available energy budget.

Performance Metrics: We consider two metrics: accuracy achievable and total battery usage on average across all devices to achieve that accuracy. Battery usage for non-adaptive schemes (no gating) is computed based on number of active devices. For adaptive methods, battery usage is computed based on the average number of device activations averaged over test examples in the test data. A battery gain of 2X in the table implies that the battery life is twice that of the unlimited baseline model (BaseM).

Adaptive Schemes: For our adaptive scheme we implemented the model described in Eq. 3.1.1 with different compression functions listed below.
Low-Dimensional Compression Model (LoDiM): For this we chose the output dimension to be the number of target classes ($d = M$). Our objective here is to benchmark scenarios where output feature dimension must also be factored into battery usage. In this scenario device outputs are viewed as "local" predictions which are then summed into a single M dimensional vector that is fed into the fusion center network.
Concatenated Compression Model (ConCoM): Again $d = M$, but the transmitted device features are concatenated and input to the fusion center. As described in Sec. 3 this setup can also be viewed as a individual predicted outputs that are mapped into independent blocks in a KM dimensional space.
Uncompressed Concatenated Model (UnCoM): Here $d = D$, i.e., the entire feature vector that is not gated is transmitted to fusion center.

Implementation: In all our adaptive schemes we trained an end-to-end model with (i) Gating function using ReLU activation; (ii) Linear transformation for the compression function and 3-layer fully connected NN architecture as in the baseline model. We implemented a conventional SGD scheme to jointly train gating, compression function and the fusion networks. For all the datasets, we set the number of nodes in the hidden layers of the NNs to be (64, 16), except for DukeReID, where we set it to (1024, 1024) due to the high dimensional features. The hyperparameters (e.g. learning rate, regularizer, etc.) for the baseline model is tuned to obtain the highest accuracy. The NNs of proposed models are trained from scratch and share the same hyperparameters to have a fair comparison. The only hyperparameter we fine-tuned is the training epoch, where the number of epochs is chosen through cross validation on the training set, and then the model is trained on the entire training data by the epochs. A local output vector is assumed to be transmitted if the corresponding ReLU gating function is nonzero during test time. In the DukeREID multi-camera dataset, each instance is a person, who is viewed by 8 cameras with possibly non-overlapping views. Our goal is to recognize the person with minimal number of camera views. The number of classes (people) consists of a list of 707 people. The problem is quite challenging since the training data is limited with a large number of classes. For training and testing we used a pre-trained Inception-v3 network, trained on Market-1501 dataset (Zheng et al., 2015), and extracted the final layer features as inputs to each device (camera).

Baselines: Adaptive vs Non-adaptive: From the tabulated results, it is evident that the naive adaptive scheme (GreedyEntM) is not even competitive to the non-adaptive baseline. Under the same level of battery

Dataset	BaseM/LimBaseM		GreedyEntM		UnCoM		ConCoM		LoDiM	
	Acc(%)	Saving	Acc(%)	Saving	Acc(%)	Saving	Acc(%)	Saving	Acc(%)	Saving
DailySports	95.2	1x	90.1	1x	94.9	2.02x	94.9	2.09x	**95.2**	1.19x
	83.6	2.50x	70.4	2.50x	92.7	2.98x	94.1	2.42x	92.6	1.61x
WARD	99.4	1x	96.5	1.x	**99.1**	1.42x	99.0	1.30x	98.1	1.33x
	98.9	1.25x	94.2	1.25x	97.5	1.48x	97.3	1.55x	97.3	1.42x
AReM	75.6	1x	70.1	1x	74.9	1x	**77.1**	1x	69.6	1x
	71.1	1.50x	66.7	1.50x	66.3	1.28x	71.5	1.58x	69.6	1.07x
Duke4ReID	87.6	1x	57.1	1.x	**86.3**	2.82x	85.4	2.62x	79.3	1.80x
	57.5	2.67x	34.2	2.48x	85.7	2.94x	79.7	2.99x	70.5	2.41x
WFRobot	92.2	1x	69.6	1.x	93.3	1.22x	**93.6**	1.44x	87.0	1.21x
	88.5	1.33x	74.7	1.33x	86.3	1.93x	90.4	1.93x	78.2	1.91x
HAR	98.8	1x	97.7	1.x	**98.4**	2.05x	98.0	1.79x	97.9	1.83x
	98.4	1.67x	96.9	1.67x	97.8	2.81x	97.4	2.46x	96.6	2.55x
GLEAM	80.1	1x	78.7	1.x	**80.5**	1.01x	80.5	1x	79.7	1.02x
	77.7	1.20x	79.2	1.20x	72.5	1.19x	75.3	1.25x	78.7	1.18x
GAS*	88.1	1x	77.7	1.x	**90.3**	1.51x	90.2	1.50x	89.1	1.52x
	84.4	2x	72.2	2.x	84.7	2.13x	85.1	2.05x	85.6	1.97x

Table 2: Tabulation of Empirical Test Results on 8 real datasets for baseline models (BaseM, LimBaseM, GreedyEntM) and adaptive models (low-dimensional compression (LoDiM), Concatenated Compression (ConCoM), Uncompressed Model (UncoM). A battery gain 2X implies that the battery life is twice that of the unlimited baseline model (BaseM). For each dataset, the highest achievable accuracy is in **bold**. (*We select best k sensors by random sampling since Gas has 16 sensors.)

saving, the accuracy for GreedyEntM is lower than BaseM. Therefore, we focus on comparing with non-adaptive baselines in the following.

Our Adaptive vs (Non-adaptive) Baselines: Our objective is to match accuracy of BaseM with adaptive schemes with high battery gain. Our second objective is to benchmark highest accuracy for a similar battery usage as the adaptive scheme. We do this by exploring different k-device combinations (LimBaseM), with k chosen to be comparable to the battery usage of the top adaptive scheme, and identify device combinations that achieves maximum accuracy with k devices[6].

First, we see that we match accuracy of the baseline with significant battery gains. In particular, as much as 2.81X gain with negligible accuracy degradation. We see that in all cases, our method of activating sensors in an input dependent manner out-performs the non-adaptive best-k method even though the best-k method benefits from concatenated device features. So the performance improvement in our method can be attributed to *adaptivity* realized through instance-dependent gating. Interestingly, we also see improvement in top performance for GAS/AReM datasets even over the unconstrained baseline (BaseM) as well. This is because a gated classifier in the large budget region functions as an ensemble of local predictions from the devices leading to boosted performance.

Comparisons of Adaptive Schemes: We highlight a few key observations that we can draw from the table. First, we observe that UnCoM typically dominates both ConCom and LoDim, achieving both high accuracy and battery gains. This is not altogether surprising because, except for gating, we transmit the full device features. LoDiM linearly aggregates all of the features into a single M-dimensional vector and performs poorly on AReM and DukeReID. AReM dataset is a low-dimensional dataset with more classes than features, and so ConCom dominates UnCoM on AReM. DukeReID has large number of classes and since each class is sparsely viewed, combining features linearly leads to poor discriminability. On the other hand, on GLEAM dataset, the best performance is achieved by LoDiM. This is due to the fact that the training data is rather limited and more complex models suffer from overfitting. Nevertheless, on other datasets, LoDiM usually achieves comparable accuracies but with reduced battery gain. This implies that the gated prediction performance is quite sensitive to output dimension of compression functions. Nevertheless, if activation energy also scales linearly with feature dimension, we would expect a different accuracy/battery-usage tradeoff, which will favor LowDim considerably.

[6]Note that since k is an integer we are unable to match the battery usage exactly but are typically quite close.

6 CONCLUSION

We proposed a novel learning framework for distributed inference for IoT applications based on training a decentralized gated network that, given an observed instance at test-time, allows for activation of select devices to transmit information to a central node, which then performs inference. This approach allows IoT devices to save energy due to reduced transmissions. We demonstrated significant energy gains on several real-world datasets with negligible accuracy degradation. We presented theoretical analysis and showed that under reasonable initialization, our SGD algorithm is guaranteed to converge for our non-convex objective to the global minima in the realizable case.

References

Altun, K., Barshan, B., and Tunçel, O. (2010). Comparative study on classifying human activities with miniature inertial and magnetic sensors. *Pattern Recognition*, 43(10):3605 – 3620.

Anguita, D., Ghio, A., Oneto, L., Parra, X., and Reyes-Ortiz, J. L. (2013). A public domain dataset for human activity recognition using smartphones. In *ESANN*.

Appadwedula, S., Veeravalli, V. V., and Jones, D. L. (2008). Decentralized detection with censoring sensors. *IEEE Transactions on Signal Processing*, 56(4):1362–1373.

Bolukbasi, T., Chang, K.-W., Wang, J., and Saligrama, V. (2017a). Resource constrained structured prediction. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Bolukbasi, T., Wang, J., Dekel, O., and Saligrama, V. (2017b). Adaptive neural networks for efficient inference. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 527–536, International Convention Centre, Sydney, Australia. PMLR.

Dheeru, D. and Karra Taniskidou, E. (2017). UCI machine learning repository.

Du, S. S., Lee, J. D., Tian, Y., Poczos, B., and Singh, A. (2017). Gradient descent learns one-hidden-layer cnn: Don't be afraid of spurious local minima. *arXiv preprint arXiv:1712.00779*.

Ge, R., Lee, J. D., and Ma, T. (2017). Learning one-hidden-layer neural networks with landscape design. *arXiv preprint arXiv:1711.00501*.

Gou, M., Karanam, S., Liu, W., Camps, O. I., and Radke, R. J. (2017). Dukemtmc4reid: A large-scale multi-camera person re-identification dataset. *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1425–1434.

Gupta, C., Suggala, A. S., Goyal, A., Simhadri, H. V., Paranjape, B., Kumar, A., Goyal, S., Udupa, R., Varma, M., and Jain, P. (2017). ProtoNN: Compressed and accurate kNN for resource-scarce devices. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1331–1340, International Convention Centre, Sydney, Australia. PMLR.

Halgamuge, M. N., Zukerman, M., Ramamohanarao, K., and Vu, H. L. (2009). An estimation of sensor energy consumption. *Progress In Electromagnetics Research B*.

Horn, R. A. and Johnson, C. R. (1991). *Topics in Matrix Analysis*. Cambridge University Press.

Jain, P., Netrapalli, P., and Sanghavi, S. (2013). Low-rank matrix completion using alternating minimization. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 665–674, New York, NY, USA. ACM.

Kumar, A., Goyal, S., and Varma, M. (2017). Resource-efficient machine learning in 2 KB RAM for the internet of things. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1935–1944, International Convention Centre, Sydney, Australia. PMLR.

Latré, B., Braem, B., Moerman, I., Blondia, C., and Demeester, P. (2011). A survey on wireless body area networks. *Wireless Networks*.

Nan, F. and Saligrama, V. (2017). Adaptive classification for prediction under a budget. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 4727–4737. Curran Associates, Inc.

Nan, F., Wang, J., and Saligrama, V. (2016). Pruning random forests for prediction on a budget. In *Advances in Neural Information Processing Systems 29*, pages 2334–2342. Curran Associates, Inc.

Palumbo, F., Gallicchio, C., Pucci, R., and Micheli, A. (2016). Human activity recognition using multisensor data fusion based on reservoir computing. 8:87–.

Perera, C., Liu, C. H., and Jayawardena, S. (2015). The emerging internet of things marketplace from an industrial perspective: A survey. *IEEE Transactions on Emerging Topics in Computing*.

Rago, C., Willett, P., and Bar-Shalom, Y. (1996). Censoring sensors: a low-communication-rate scheme for distributed detection. *IEEE Transactions on Aerospace and Electronic Systems*, 32(2):554–568.

Rahman, S. A., Merck, C., Huang, Y., and Kleinberg, S. (2015). Unintrusive eating recognition using google glass. In *Proceedings of the 9th International Conference on Pervasive Computing Technologies for Healthcare*, PervasiveHealth '15, pages 108–111, ICST, Brussels, Belgium, Belgium. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., and Dean, J. (2017). Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.

Si, S., Chiang, K.-Y., Hsieh, C.-J., Rao, N., and Dhillon, I. S. (2016). Goal-directed inductive matrix completion. In *Proceedings of the 22Nd ACM SIGKDD*

International Conference on Knowledge Discovery and Data Mining, KDD '16, pages 1165–1174, New York, NY, USA. ACM.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B)*, 58:267–288.

Trapeznikov, K. and Saligrama, V. (2013). Supervised sequential classification under budget constraints. In *Artificial Intelligence and Statistics*, pages 581–589.

Tsitsiklis, J. N. and Athans, M. (1984). On the complexity of decentralized decision making and detection problems. In *The 23rd IEEE Conference on Decision and Control*, pages 1638–1641.

Vergara, A., Vembu, S., Ayhan, T., A. Ryan, M., Homer, M., and Huerta, R. (2012). Chemical gas sensor drift compensation using classifier ensembles. s 166–167:320–329.

Viswanathan, R. and Varshney, P. K. (1997). Distributed detection with multiple sensors part i. fundamentals. *Proceedings of the IEEE*, 85(1):54–63.

Wang, J., Bolukbasi, T., Trapeznikov, K., and Saligrama, V. (2014). Model selection by linear programming. In *European Conference on Computer Vision*, pages 647–662. Springer.

Wang, J., Trapeznikov, K., and Saligrama, V. (2015). Efficient learning by directed acyclic graph for resource constrained prediction. In *Advances in Neural Information Processing Systems*, pages 2152–2160.

Wang, X., Yu, F., Dou, Z.-Y., and Gonzalez, J. E. (2017). Skipnet: Learning dynamic routing in convolutional networks. *arXiv preprint arXiv:1711.09485*.

Xu, Z. E., Weinberger, K. Q., and Chapelle, O. (2012). The greedy miser: Learning under test-time budgets. In *Proceedings of the 29th International Conference on Machine Learning, ICML*.

Yang, A. Y., Iyengar, S., Kuryloski, P., and Jafari, R. (2008). Distributed segmentation and classification of human actions using a wearable motion sensor network. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8.

Yang, A. Y., Jafari, R., Sastry, S. S., and Bajcsy, R. (2009). Distributed recognition of human actions using wearable motion sensor networks. *Journal of Ambient Intelligence and Smart Environments*, 1(2):103–115.

Zhang, Y., Wainwright, M. J., Jordan, M. I., et al. (2017). Optimal prediction for sparse linear models? lower bounds for coordinate-separable m-estimators. *Electronic Journal of Statistics*, 11(1):752–799.

Zheng, L., Shen, L., Tian, L., Wang, S., Wang, J., and Tian, Q. (2015). Scalable person re-identification: A benchmark. In *Computer Vision, IEEE International Conference on*.

Zhong, K., Song, Z., Jain, P., Bartlett, P. L., and Dhillon, I. S. (2017). Recovery guarantees for one-hidden-layer neural networks. In *International Conference on Machine Learning*, pages 4140–4149.

Acknowledgment

The authors would like to thank the Area Chair and the reviewers for their constructive comments. This work was supported by the Office of Naval Research Grant N0014-18-1-2257, NGA-NURI HM1582-09-1-0037 and the U.S. Department of Homeland Security, Science and Technology Directorate, Office of University Programs, under Grant 2013-ST-061-ED0001.