ProtoNN: Compressed and Accurate kNN for Resource-scarce Devices


Chirag Gupta


Arun Sai Suggala*


Ankit Goyal


Harsha Vardhan Simhadri


Bhargavi Paranjape


Ashish Kumar


Saurabh Goyal


Raghavendra Udupa


Manik Varma


Prateek Jain

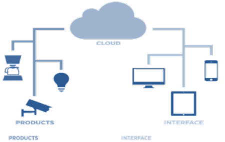
* presenting

August 8, 2017

ICML 2017, Sydney

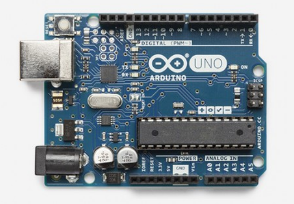
Machine Learning on Resource Scarce Devices

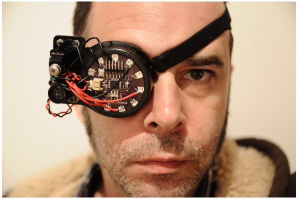
- Most Machine Learning (ML) models are designed for large machines.
  - Deep Learning on GPUs, TPUs.
- However, there are billions of devices with
  - small memory - few kB of RAM.
  - small compute power.



Case in point: Arduino Uno

The UNO is a simple device easily accessible to a large strata of developers and amateurs.



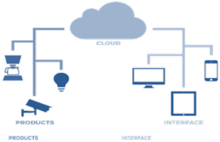


- 2kB RAM: often smaller than a single data-point!
- 32kB Read-Only Flash

Image Source: `http://blog.ocad.ca/wordpress/gdes3b16-fw201202-01/`

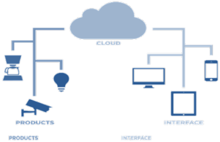
Machine Learning on Resource Scarce Devices

- Resource scarce devices such as Internet of Things (IoT) devices are dumb
  - used only for data gathering.
  - data transmitted to the cloud, where predictions are made.



Machine Learning on Resource Scarce Devices

- Resource scarce devices such as Internet of Things (IoT) devices are dumb
    - used only for data gathering.
    - data transmitted to the cloud, where predictions are made.
- Transmitting data to the cloud has issues
    - drains the battery of devices.
    - privacy, latency, and bandwidth concerns.



Machine Learning on Resource Scarce Devices

- Question: Can we make these devices Intelligent?
- Existing ML models either don't fit on these devices or perform poorly.
  - trivial compression leads to poor performance.

Prior Work on Memory Efficient ML

A huge line of work exists on designing memory efficient models:

- Compressing Neural Networks: [HMD16, IHM+16, YMD+15] .
- Pruning Random Forests: [NWS16, DJX16, KS12].
- Compressing k-Nearest Neighbours (kNN): [Ang05, KTWA14], [ZGK+17].

Prior Work on Memory Efficient ML

All these have one or more of the following problems:

- models don't fit into tiny devices with $\leq 2kB$ of RAM.
- perform poorly when compressed into tiny devices.
- Don't generalize to other supervised learning tasks such as multilabel classification, ranking.

Significance of the Work

Design an algorithm that can

- be deployed on tiny devices for prediction.
- provide near state-of-the-art performance.
- perform fast predictions without draining the battery.
- handle general supervised learning tasks
  - such as multilabel classification, ranking.

ProtoNN

- ProtoNN is a Prototype based kNN Approach.

ProtoNN

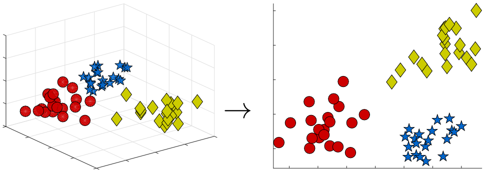
- ProtoNN is a Prototype based kNN Approach.
- Why kNN?
  - simple and interpretable models
  - generality - can model complex decision boundaries

ProtoNN

- ProtoNN is a Prototype based kNN Approach.
- Why kNN?
  - simple and interpretable models
  - generality - can model complex decision boundaries
- However, kNN has several issues:
  - large model size
  - large prediction time
  - poor accuracy - how to compute distance?

ProtoNN

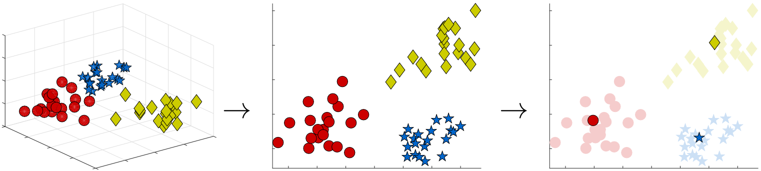
- ProtoNN jointly learns
  - a sparse low-d projection



(a) input (b) low-d projection

ProtoNN

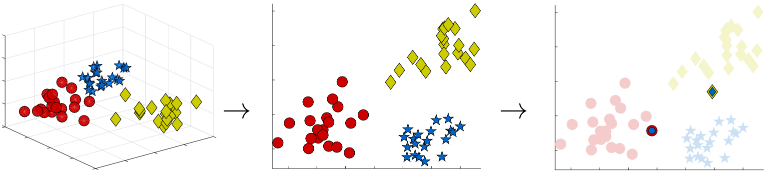
- ProtoNN jointly learns
  - a sparse low-d projection
  - a set of prototypes in the low dimensional space



(a) input (b) low-d projection (c) prototypes

ProtoNN

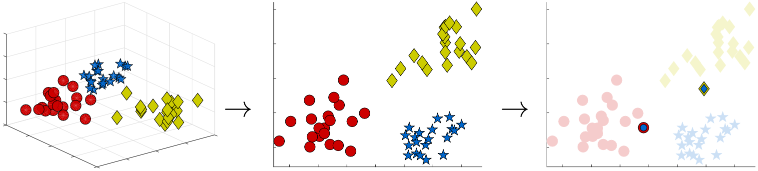
- ProtoNN jointly learns
  - a sparse low-d projection
  - a set of prototypes in the low dimensional space
  - and their labels.



(a) input (b) low-d projection (c) prototypes

ProtoNN

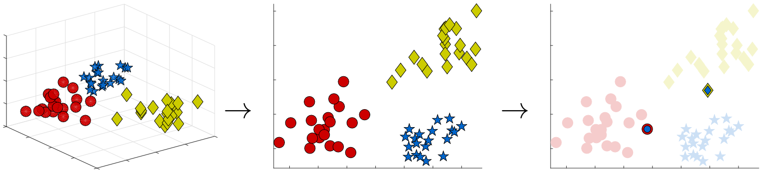
- Joint learning of low-d projection and prototypes:
  - reduces the model size.
  - lowers the prediction time.
  - improves the accuracy.



(a) input (b) low-d projection (c) prototypes

ProtoNN

- gives explicit control over the model size through hard sparsity (ℓ_0) constraints on the parameters.



(a) input $\rightarrow$ (b) low-d projection $\rightarrow$ (c) prototypes

ProtoNN (Formal)

Input:

- Feature vectors $\{\mathbf{x}_i\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^d$.
- Label vectors $\{\mathbf{y}_i\}_{i=1}^n$, where $\mathbf{y}_i \in \mathcal{Y}$.
- For classification with L classes, $\mathcal{Y} \in \{0, 1\}^L$.

Parameters:

- Low dimensional projection matrix: W
- Prototypes: $\{\mathbf{b}_1, \ldots, \mathbf{b}_m\}$
- Label vector of prototypes: $\{\mathbf{z}_1, \ldots, \mathbf{z}_m\}$

ProtoNN (Formal)

Decision Function for a point $\mathbf{x}$ is given by

$$s(\mathbf{x}) = \overbrace{\sum_{j=1}^{m}}^{\text{sum over all prototypes}} \left[\underbrace{K(\overbrace{W\mathbf{x}}^{\text{low-d projection}}, \mathbf{b}_j)}_{\text{similarity with } j^{th} \text{ prototype}} \mathbf{z}_j \right].$$

We choose K to be RBF kernel.

Training Objective:

$$\arg\min_{W,\{\mathbf{b}_j,\mathbf{z}_j\}_{j\in[m]}} \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{y}_i - s(\mathbf{x}_i)\|_2^2$$

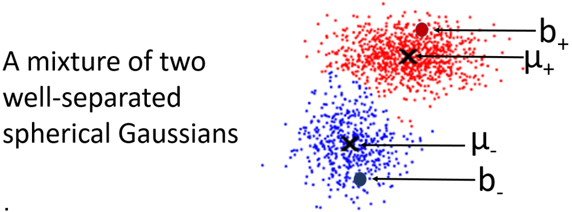
$$\text{s.t.} \quad \|W\|_0 \le s_W, \|B\|_0 \le s_B, \|Z\|_0 \le s_Z.$$

ProtoNN - Optimization

- We use Alternating Minimization to optimize the training objective.
  - we alternate over Z, B, W while fixing the other two parameters.
  - for each sub-problem we use Projected SGD with Nesterov's acceleration.

Analysis of ProtoNN

- One of the first analysis of an algorithm for resource scarce devices.



- Notation:
  - $\boldsymbol{\mu}_+$, $\boldsymbol{\mu}_-$ - centers of +ve and -ve classes.
  - Fix $W = I$, $Z = [\boldsymbol{e}_1,\ \boldsymbol{e}_2]$. Let $\boldsymbol{b}_+, \boldsymbol{b}_-$ be the prototypes.

Image Source: `http://recognize-speech.com/basics/introduction-to-gaussian-mixture-models`

Analysis of ProtoNN

Theorem (simplified)

Let $\bar{\boldsymbol{\mu}} := \boldsymbol{\mu}_+ - \boldsymbol{\mu}_-$, $n \to \infty$. *Suppose:*

Analysis of ProtoNN

Theorem (simplified)

Let $\bar{\boldsymbol{\mu}} := \boldsymbol{\mu}_+ - \boldsymbol{\mu}_-$, $n \to \infty$. *Suppose:*

- ***(Separation)*** $\|\boldsymbol{b}_+ - \boldsymbol{\mu}_+\| \geq 8\|\bar{\boldsymbol{\mu}}\| \exp\left\{-\frac{\|\bar{\boldsymbol{\mu}}\|^2}{4}\right\}$
- ***(mild regularity)*** $d \geq 4\|\bar{\boldsymbol{\mu}}\|^2$

Analysis of ProtoNN

Theorem (simplified)

Let $\bar{\boldsymbol{\mu}} := \boldsymbol{\mu}_+ - \boldsymbol{\mu}_-$, $n \to \infty$. *Suppose:*

- ***(Separation)*** $\|\boldsymbol{b}_+ - \boldsymbol{\mu}_+\| \geq 8\|\bar{\boldsymbol{\mu}}\| \exp\left\{-\frac{\|\bar{\boldsymbol{\mu}}\|^2}{4}\right\}$
- ***(mild regularity)*** $d \geq 4\|\bar{\boldsymbol{\mu}}\|^2$

Then, gradient descent update $\boldsymbol{b}'_+ = \boldsymbol{b}_+ - \eta\nabla_{\boldsymbol{b}_+}\mathcal{R}_{emp}$, *with appropriate* $\eta \geq 0$ *satisfies the following with constant probability:*

$$\underbrace{\|\boldsymbol{b}'_+ - \boldsymbol{\mu}_+\|^2 \leq \|\boldsymbol{b}_+ - \boldsymbol{\mu}_+\|^2 \left(1 - 0.01 \exp\left\{-\frac{\|\bar{\boldsymbol{\mu}}\|^2}{4}\right\}\right)}_{\textit{Geometric Convergence}}$$

Results

Variety of experiments on several benchmark datasets:

- ProtoNN vs. Compressed, Uncompressed baselines.
- ProtoNN for binary, multiclass, multilabel classification.
- Energy consumption, Prediction time of ProtoNN on resource scarce device (Arduino Uno microcontroller).

ProtoNN vs. Compressed Baselines

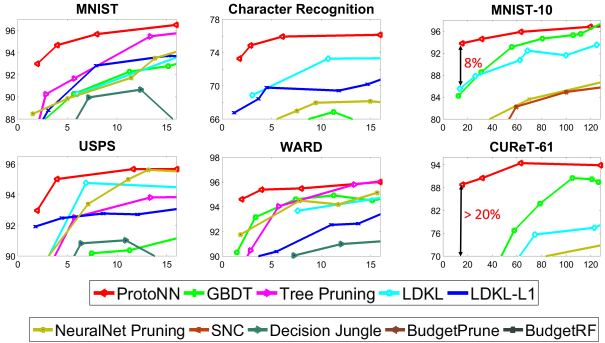


Figure: Model size (kB, X-axis) vs Accuracy (%, Y-axis). Left two columns are for binary datasets and the right most column is for multiclass datasets.

ProtoNN vs. Uncompressed Baselines on binary datasets

Dataset		ProtoNN (16kB)	kNN	SNC	BNC	GBDT	1-hidden NeuralNet	RBF-SVM
character recognition	model size	15.94 (400x)	6870.3	441.2	70.88	625	314.06	6061.71
	accuracy	**76.14**	67.28	74.87	70.68	72.38	72.53	75.6
eye	model size	10.32	14592	3305	1311.4	234.37	6401.56	7937.45
	accuracy	90.82	76.02	87.76	80.61	83.16	90.31	**93.88**
mnist	model size	15.96	183750	4153.6	221.35	1171.87	3070	35159.4
	accuracy	96.5	96.9	95.74	98.16	**98.36**	98.33	98.08
usps	model size	11.625	7291	568.8	52.49	234.37	504	1659.9
	accuracy	95.67	96.7	**97.16**	95.47	95.91	95.86	96.86
ward	model size	15.94	17589.8	688	167.04	1171.87	3914.06	7221.75
	accuracy	96.01	94.98	96.01	93.84	**97.77**	92.75	96.42
cifar	model size	15.94	78125	3360	144.06	1562.5	314.06	63934.2
	accuracy	76.35	73.7	76.96	73.74	77.19	75.9	**81.68**

The model size of ProtoNN is restricted to 16kB. No model size restrictions are imposed on the baselines.

ProtoNN vs. Uncompressed Baselines on multiclass datasets

Dataset		ProtoNN (64kB)	kNN	SNC	BNC	GBDT	1-hidden NeuralNet	RBF SVM
letter-26	model size	63.4	1237.8	145.08	31.95	20312	164.06	568.14
	accuracy	97.10	95.26	96.36	92.5	97.16	96.38	**97.64**
mnist-10	model size	63.4	183984.4	4172	220.46	5859.37	4652.34	39083.7
	accuracy	95.88	94.34	93.6	96.68	97.9	**98.44**	97.3
usps-10	model size	63.83	7291.4	568.8	51.87	390.62	519.53	1559.6
	accuracy	94.92	94.07	94.77	91.23	94.32	94.32	**95.4**
curet-61	model size	63.14 (140x)	10037.5	513.3	146.70	2382.81	1310	8940.8
	accuracy	94.44	89.81	95.87	91.87	90.81	95.51	**97.43**

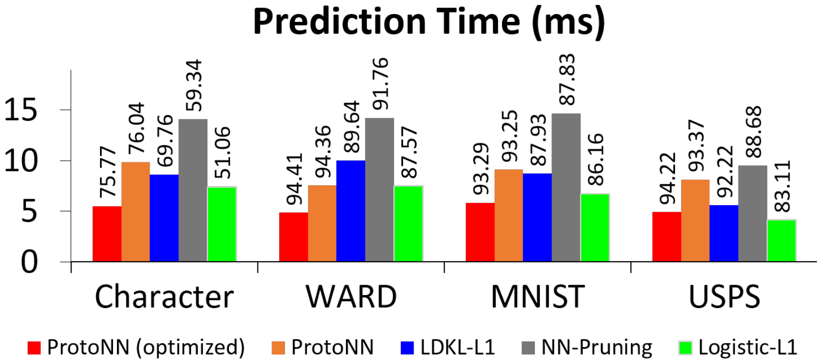
The model size of ProtoNN is restricted to 64kB. No model size restrictions are imposed on the baselines.

Multilabel Classification

Dataset		ProtoNN	FastXML	DiSMEC	SLEEC
mediamill	model size	54.8K	7.64M	**48.48K**	57.95M
train samples = 30993	Precision@1	85.19	83.65	**87.25**	86.12
feature dim. = 120	Precision@3	69.01	66.92	69.3	**70.31**
Label dim. = 101	Precision@5	54.39	52.51	54.19	**56.33**
delicious	model size	925.04K (40x)	**36.87M**	1.97M	7.34M
train samples = 12920	Precision@1	**68.92**	**69.41**	66.14	67.77
feature dim. = 500	Precision@3	**63.04**	**64.2**	61.26	61.27
Label dim. = 983	Precision@5	**58.32**	**59.83**	56.30	56.62
eurlex	model size	**5.03M**	410.8M	79.86M	61.74M
train samples = 15539	Precision@1	77.74	71.36	**82.40**	79.34
feature dim. = 5000	Precision@3	65.01	59.85	**68.50**	64.25
Label dim. = 3993	Precision@5	53.98	50.51	**57.70**	52.29

Performance of ProtoNN on multilabel classification.

Experiments on an Arduino Uno



Energy (mJ) = (0.2455 J/s) * Prediction time (ms)

Conclusion and Future Work

Conclusion

- ML for resource scarce devices is a high-impact research area.
- ProtoNN:
  - can be deployed on tiny devices (with $\leq 2kB$ of RAM).
  - can provide fast predictions.
  - has state-of-the-art performance.
  - can handle general supervised learning tasks.

Conclusion and Future Work

Conclusion

- ML for resource scarce devices is a high-impact research area.
- ProtoNN:
  - can be deployed on tiny devices (with $\leq 2kB$ of RAM).
  - can provide fast predictions.
  - has state-of-the-art performance.
  - can handle general supervised learning tasks.

Future Work

- Multilabel Classification with millions of labels.
- Unsupervised learning tasks such as Nonlinear Matrix Factorization and Anomaly Detection.

Questions?

Poster: Wednesday session (poster #16)
Website: http://harsha-simhadri.org/EdgeML/

References I

[Ang05] Fabrizio Angiulli, *Fast condensed nearest neighbor rule*, ICML, 2005.

[DJX16] O. Dekel, C. Jacobbs, and L. Xiao, *Pruning decision forests*, Personal Communications, 2016.

[HMD16] S. Han, H. Mao, and W. J. Dally, *Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding*, ICLR, 2016.

[IHM+16] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer, *Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size*, arXiv preprint arXiv:1602.07360 (2016).

[KS12] Vrushali Y Kulkarni and Pradeep K Sinha, *Pruning of random forest classifiers: A survey and future directions*, Data Science & Engineering (ICDSE), 2012 International Conference on, IEEE, 2012, pp. 64–68.

[KTWA14] Matt J. Kusner, Stephen Tyree, Kilian Weinberger, and Kunal Agrawal, *Stochastic neighbor compression*, ICML, 2014.

[NWS16] F. Nan, J. Wang, and V. Saligrama, *Pruning random forests for prediction on a budget*, 2016.

[YMD+15] Zichao Yang, Marcin Moczulski, Misha Denil, Nando de Freitas, Alex Smola, Le Song, and Ziyu Wang, *Deep fried convnets*, Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1476–1483.

References II

[ZGK+17] Kai Zhong, Ruiqi Guo, Sanjiv Kumar, Bowei Yan, David Simcha, and Inderjit Dhillon, *Fast Classification with Binary Prototypes*, Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (Fort Lauderdale, FL, USA) (Aarti Singh and Jerry Zhu, eds.), Proceedings of Machine Learning Research, vol. 54, PMLR, 20–22 Apr 2017, pp. 1255–1263.