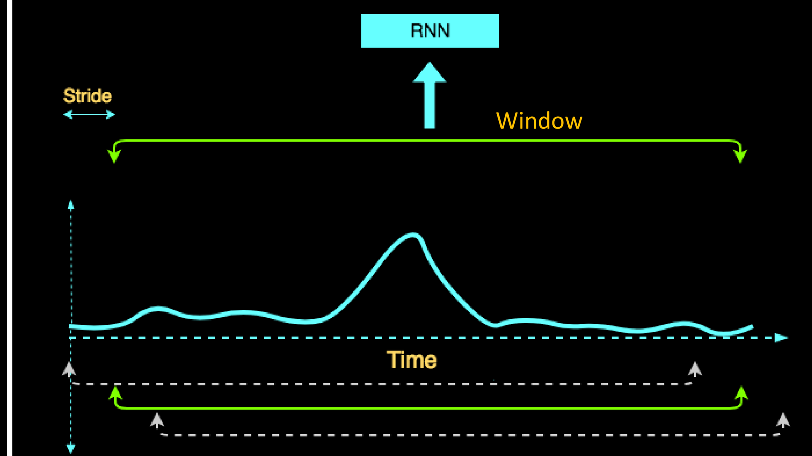
Microsoft Research

EMI-RNN: Multiple Instance Learning for Efficient Sequential Data Classification on Resource Constrained Devices

Don Kurian Dennis Chirag Pabbaraju Harsha Vardhan Simhadri Prateek Jain

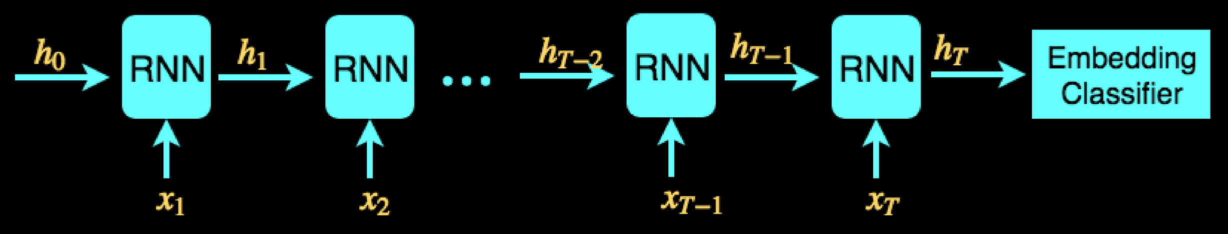
github.com/Microsoft/EdgeML

Recurrent Neural Networks (RNN)



- State-of-the art for time series.
- Data is divided into overlapping windows and an RNN is run over each.

- Each RNN run is a sequence of updates to its internal state.
- The state update rule: complicated, non-linear and expensive.



$$\mathbf{h_t} = \sigma(\mathbf{Wx}_t + \mathbf{Uh}_{t-1} + \mathbf{b})$$

- Prohibitively expensive for edge devices
- Key-word spotting: Feature computation + prediction every 30ms for real-time response! Vanilla LSTM takes 64ms!

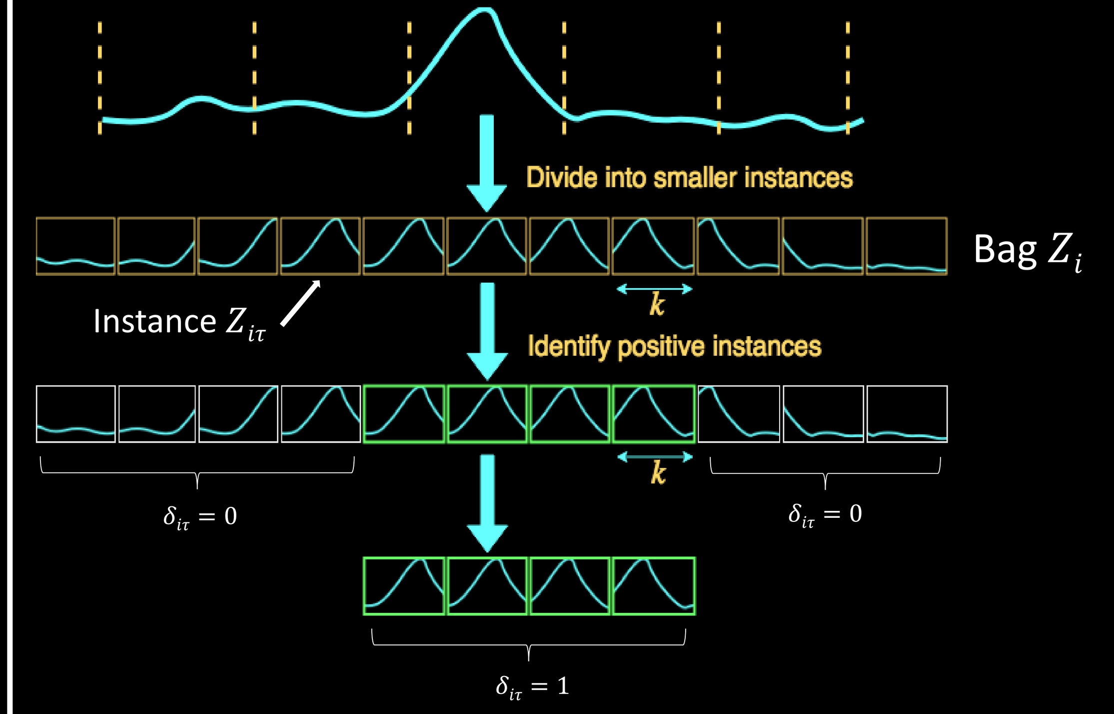
Typical class signature length k << T

- Example: Keyword spotting --- keyword 'Up' usually 100-200ms (k) while the RNN window T ~ 1 second.
- Learning on these: k (<< T) step RNN.
- Common prefixes are small : predict early.
- Class signature can lie anywhere!

Contributions

- EMI-RNN: exploits temporal structure and above observations
- USP: a) higher accuracy than baseline RNN architectures
  b) reduce inference time by as much as 72x
  c) Allows deployment on tiny devices like Raspberry Pi0, M4 MCU
- Techniques: multi-instance learning (MIL) + early prediction
- Analysis: recovers provably optimal solution in non-homogeneous MIL settings --- first such result for non-homogeneous MIL

MI-RNN: Multiple-Instance RNN



- Divide into *bag* of overlapping *k* length windows (*instances*).
- Isolate the instances with signature. Relabel these instances and train.
- NP-Hard in general!

Exploit temporal locality and approximate signature length with *MIL/Robust learning techniques* in the optimization problem.

Formulation learns model f as well as starting index s_i of the class signature in each data point

$$\min_{f,\, s_i, 1\le i\le n} \frac{1}{n}\sum_{i,\tau} \delta_{i\tau}\cdot \ell\left(f(Z_{i\tau}), y_i\right), \qquad s.t., \delta_{i\tau} = \begin{cases} 1, & \tau \in [s_i, s_i+\kappa] \\ 0, & \tau \notin [s_i, s_i+\kappa] \end{cases}$$

Algorithm

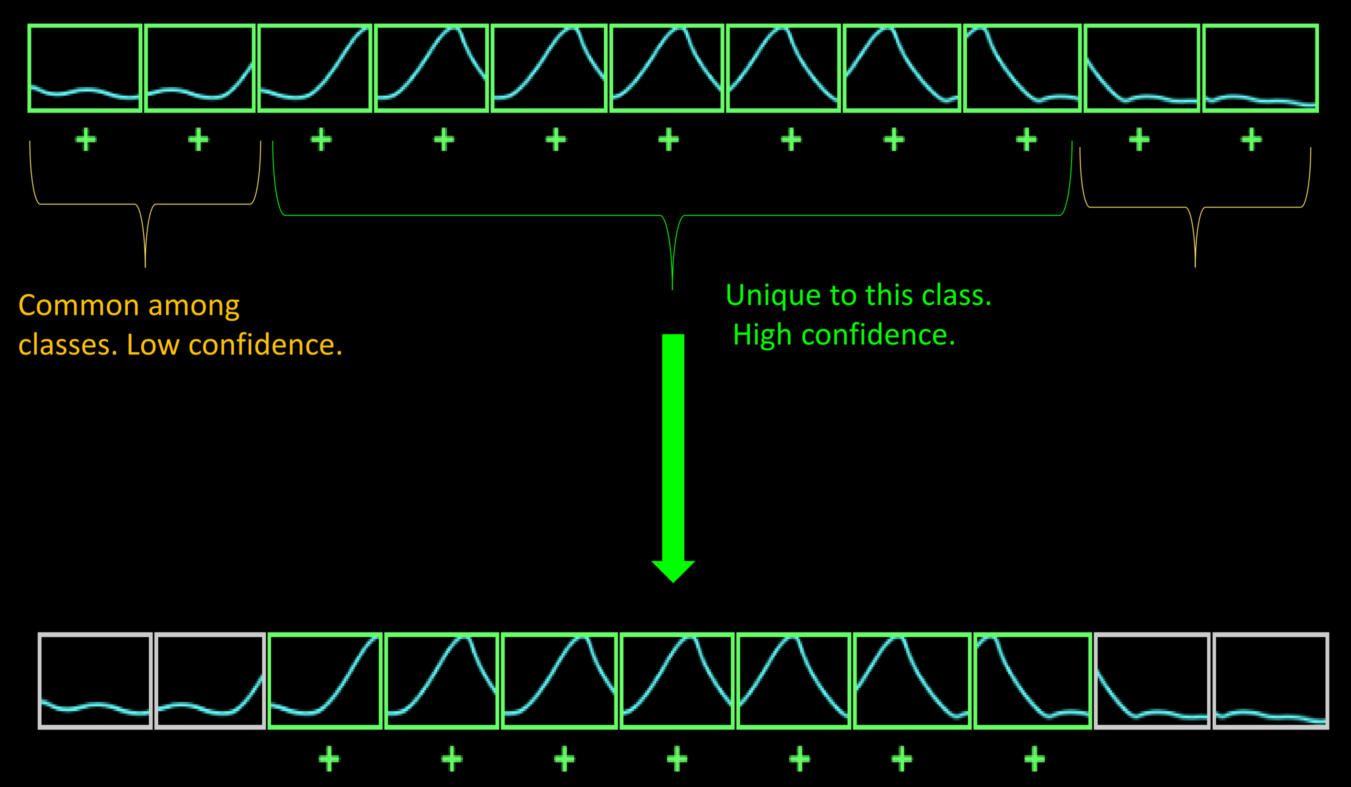
Step 1: Assign labels $(Z_{i\tau}, y_{i\tau}), s.t.\, y_{i\tau} = y_i, \forall\, \tau$

Step 2: Train classifier f_t on this miss-labelled data

Step 3: $Score(s_i) = \sum_{j=s_i}^{s_i+\kappa} f_t(Z_{ij})$ and pick $argmax_{s_i}\ Score(s_i)$

Step 4: Update labels. Repeat with new labels

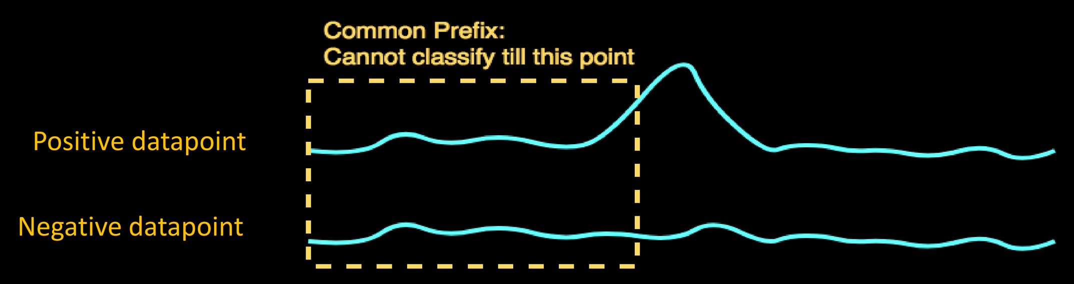
Theorem: In $O(\log n)$ iterations, the true positive set will be recovered **exactly**, with high prob.



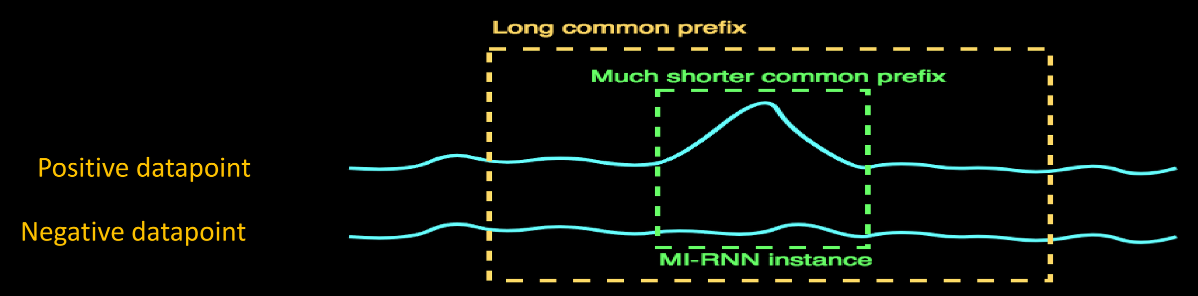
Setting:

- Two classes: $Z_{i,\tau}^{N}$ --- negative class instances sampled from a Gaussian with mean μ^-
- $Z_{i,\tau}^{P}$ --- positive class instances, lie in a small ball around μ^+
- $||\mu^+ - \mu^-|| \geq C \log T$
- Let $n \geq \frac{dT||\mu^+ - \mu^-||^2|}{k^2}$

EMI-RNN: Early Multi-Instance RNN



Naïve early prediction difficult due to common prefixes

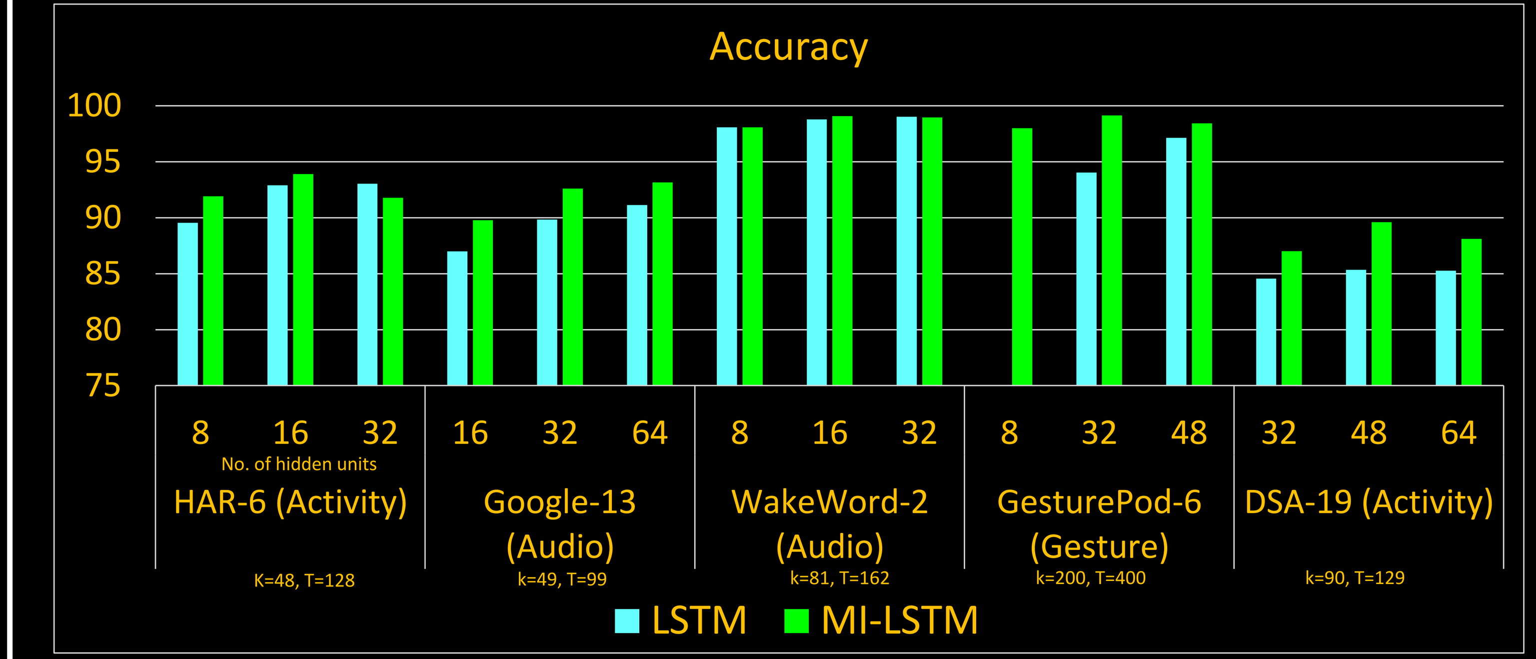


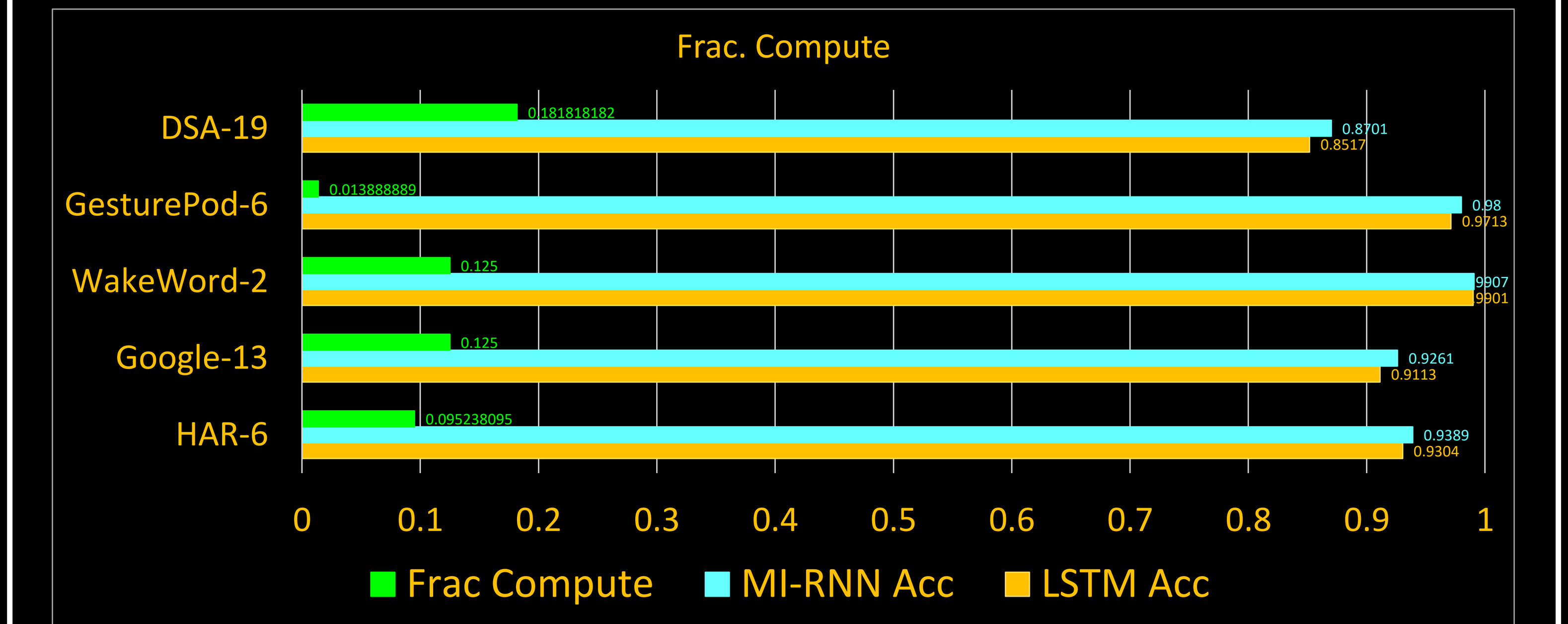
MI-RNN removes common prefixes making early prediction effective

Incentivize early prediction during training EMI-RNN: Jointly train MI-RNN with early loss

$$\mathrm{L_e}(\mathbf{X}, \mathbf{y}) = \sum_{t=1}^{T} (\mathbf{v}^{\top}\mathbf{h_t} - \mathbf{y})^2$$

Results

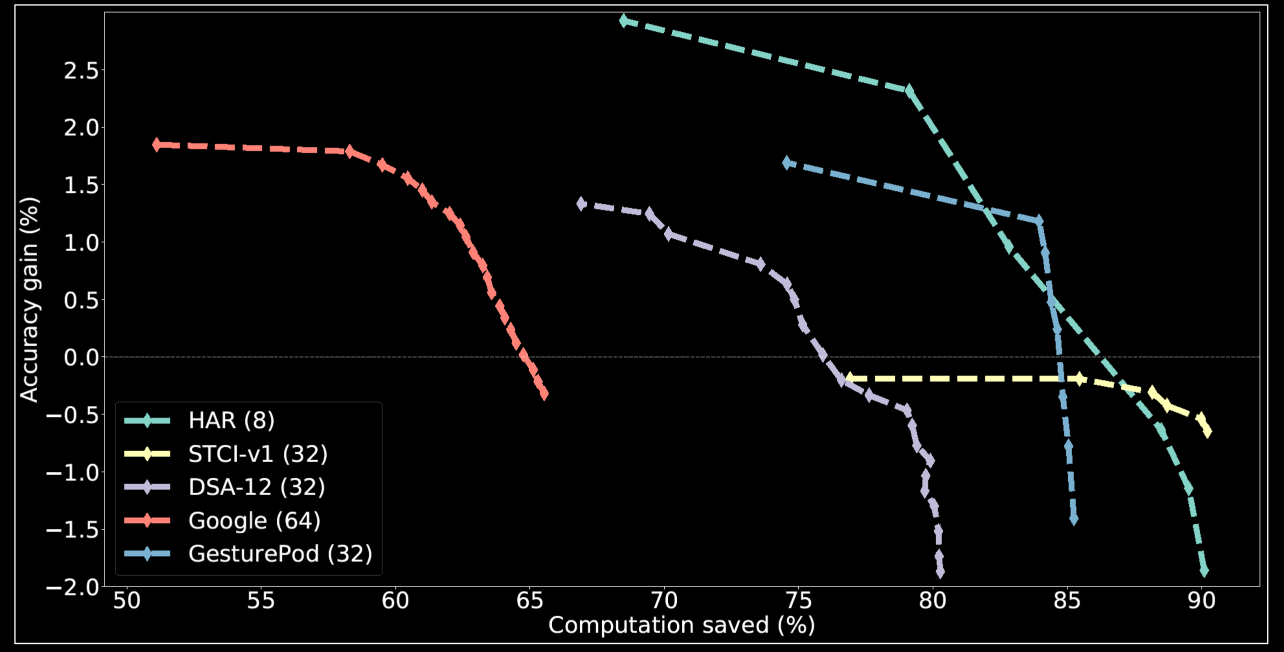




Fraction compute compared to beat LSTM accuracy. (w/o early prediction)

Device	Hidden Dim.	LSTM (ms)	Accuracy	MI-RNN (ms)	Accuracy	EMI-RNN (ms)
RPi0 (22.5 ms)	16	28.14	86.99	14.06	89.78	5.62
	32	74.46	89.84	37.41	92.61	14.96
	64	226.1	91.13	112.6	93.16	45.03
RPi3 (26.39 ms)	16	12.76	86.99	6.48	89.78	2.59
	32	33.1	89.84	16.47	92.61	6.58
	64	92.09	91.13	46.28	93.16	18.51

Execution time on Raspberry Pi0 and Pi3. Real-time constraint in parenthesis.



EMI-RNN: Computation Savings vs Accuracy Gain