The Edge of Machine Learning

Multiple Instance Learning for Fast, Stable and Early RNN Predictions

Don Dennis,
Microsoft Research India,
Joint work with Chirag P., Harsha and Prateek
Accepted to NIPS ’18
Algorithms for the IDE - EdgeML

- A library of machine learning algorithms
- Trained on the cloud
- Ability to run on tiniest of IoT devices
Previous Work: EdgeML Classifiers

ProtoNN
Gupta et al., ICML ’17

Bonsai
Kumar et al., ICML ’17

Fast(G)RNN
Kusupati et al., NIPS ’18

Code: https://github.com/Microsoft/EdgeML
Previous Work: EdgeML Applications

GesturePod

Patil et al.,
(to be submitted)

Wake Word

(work in progress)

Code: En route
Problem
Problem

• Given time series data point, classify it as a certain class.
• GesturePod:
  – Data: Accelerometer and gyroscope information
  – Task: Detect if gesture was performed
Problem

RNN

Feature vector
Problem

RNN

Feature vector

Classifier
Problem

![Diagram showing a process involving RNN, Feature vector, Classifier, ProtoNN, and Bonsai.]

- RNN
  - Feature vector
  - Classifier
  - ProtoNN and Bonsai
Problem

- Expensive!
- Prohibitive on IoT Devices

Flowchart:
- RNN
- Feature vector
- Classifier

Additional notes:
- ProtoNN and Bonsai
RNNs are Expensive

• For time series data: \( X = [x_1, x_2, x_3, \ldots, x_T] \quad x \in \mathbb{R}^d \)

• \( T \) RNN updates are performed:

\[
  h_t = \sigma\left( w x_t + u h_{t-1} + b \right)
\]

• \( T \) is determined by the data labelling process. Example GesturePod – 2 seconds.
RNNs are Expensive

• For time series data: \( X = [x_1, x_2, x_3, \ldots, x_T] \quad x \in \mathbb{R}^d \)

• \( T \) RNN updates are performed:

\[
h_t = \sigma(wx_t + uh_{t-1} + b)
\]

• \( T \) is determined by the data labelling process. Example GesturePod – 2 seconds.
RNNs are Expensive

Observe how $k << T$.

- RNN runs over longer data point – *unnecessarily large* $T$ and prediction time.
- Predictors must recognize signatures with different offsets - *requires larger* predictors.
- Sequential compute.
- Also lag.
RNNs are Expensive

Solution?

Approach 1 of 2: Exploit the fact that $k \ll T$ and learn a smaller classifier.

How?
STEP 1: Divide $X$ into smaller instances.
• STEP 1: Divide $X$ into smaller instances.
• STEP 1: Divide $X$ into smaller instances.

• STEP 2: Identify positive instances. Discard negative (noise) instances.
• **STEP 1:** Divide $X$ into smaller instances.

• **STEP 2:** Identify positive instances. Discard negative (noise) instances.
How?

- **STEP 1:** Divide $X$ into smaller instances.

- **STEP 2:** Identify positive instances. Discard negative (noise) instances.

- **STEP 3:** Use these instances to train a smaller classifier.
• **STEP 1:** Divide $X$ into smaller instances.

• **STEP 2:** Identify positive instances. Discard negative (noise) instances.

• **STEP 3:** Use these instances to train a smaller classifier.

**Note!** Most of the instances are just *noise.*
How?

• STEP 1: Divide X into smaller instances.

• STEP 2: Identify positive instances. Discard negative (noise) instances.

• STEP 3: Use these instances to train a smaller classifier.
How?

- **STEP 1:** Divide $X$ into smaller instances.
- **STEP 2:** Identify positive instances. Discard negative (noise) instances.
- **STEP 3:** Use these instances to train a smaller classifier.
How?

- **STEP 1**: Divide $X$ into smaller instances.
- **STEP 2**: Identify positive instances. Discard negative (noise) instances.
- **STEP 3**: Use these instances to train a smaller classifier.

**Robust Learning**

Standard techniques don’t apply.  
- Too much noise.  
- Ignores temporal structure of the data.
How?

• STEP 1: Divide $X$ into smaller instances.

• STEP 2: Identify positive instances. Discard negative (noise) instances.

• STEP 3: Use these instances to train a smaller classifier.

Robust Learning

Standard techniques don’t apply.
• Too much noise.
• Ignores temporal structure of the data

Traditional Multi Instance Learning (MIL)
How?

- STEP 1: Divide X into smaller instances.
- STEP 2: Identify positive instances. Discard negative (noise) instances.
- STEP 3: Use these instances to train a smaller classifier.

Robust Learning

Standard techniques don’t apply.
- Too much noise.
- Ignores temporal structure of the data.

Traditional Multi-Instance Learning (MIL)

Standard techniques don’t apply.
- Heterogenous.
- Ignores temporal structure of the data.
Exploit temporal locality with MIL/Robust learning techniques

Property 1: Positive instances are clustered together.
Property 2: Number of positive instances can be estimated.
Algorithm: MI-RNN

Two phase algorithm – alternates between identifying positive instances and training on the positive instances.
Algorithm: MI-RNN

• Step 1:
  Assign labels
  Instance = source data
Algorithm: MI-RNN

• **Step 1:**
  Assign labels
  Instance = source data
Algorithm: MI-RNN

- **Step 1:**
  Assign labels
  
  Instance = source data
Algorithm: MI-RNN

- **Step 2:**
  Train classifier on this data
Algorithm: MI-RNN

• Step 2:
  Train classifier on this data
Algorithm: MI-RNN

- Step 2:
  Train classifier on this data
Algorithm: MI-RNN

• Step 2: Train classifier on this data
Algorithm: MI-RNN

- **Step 2:**
  Train classifier on this data

Classifier will be confused.
Low prediction confidence.
Step 3:
Wherever possible, use classifier’s prediction score to pick top-κ

Should satisfy property 1 and property 2
Algorithm: MI-RNN

• Step 3: Wherever possible, use classifier’s prediction score to pick top-κ

Should satisfy property 1 and property 2
Algorithm: MI-RNN

- **Step 4:**
  Repeat with new labels
MI-RNN: Does It Work?
MI-RNN: Does It Work?

• Of course!
**MI-RNN: Does It Work?**

- Of course!
- Theoretical analysis:
  
  Convergence to global optima in linear time for *nice* data
MI-RNN: Does It Work?

• Of course!

• Theoretical analysis:
  Convergence to global optima in linear time for *nice* data

• Experiments:
  Significantly improve accuracy while saving computation
  – Various tasks: activity recognition, audio keyword detection, gesture recognition
## MI-RNN: Does It Work?

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Hidden Dim</th>
<th>LSTM</th>
<th>MI-RNN</th>
<th>Savings %</th>
</tr>
</thead>
<tbody>
<tr>
<td>HAR-6 <em>(Activity detection)</em></td>
<td>8</td>
<td>89.54</td>
<td>91.92</td>
<td>62.5</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>92.90</td>
<td>93.89</td>
<td></td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>93.04</td>
<td>91.78</td>
<td></td>
</tr>
<tr>
<td>Google-13 <em>(Audio)</em></td>
<td>16</td>
<td>86.99</td>
<td>89.78</td>
<td>50.5</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>89.84</td>
<td>92.61</td>
<td></td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>91.13</td>
<td>93.16</td>
<td></td>
</tr>
<tr>
<td>WakeWord-2 <em>(Audio)</em></td>
<td>8</td>
<td>98.07</td>
<td>98.08</td>
<td>50.0</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>98.78</td>
<td>99.07</td>
<td></td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>99.01</td>
<td>98.96</td>
<td></td>
</tr>
</tbody>
</table>
MI-RNN: Does It Work?

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Hidden Dim</th>
<th>LSTM</th>
<th>MI-RNN</th>
<th>Savings %</th>
</tr>
</thead>
<tbody>
<tr>
<td>HAR-6 (Activity detection)</td>
<td>8</td>
<td>89.54</td>
<td>91.92</td>
<td>62.5</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>92.90</td>
<td>93.89</td>
<td></td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>93.04</td>
<td>91.78</td>
<td></td>
</tr>
<tr>
<td>Google-13 (Audio)</td>
<td>16</td>
<td>86.99</td>
<td>89.78</td>
<td>50.5</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>89.84</td>
<td>92.61</td>
<td></td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>91.13</td>
<td>93.16</td>
<td></td>
</tr>
<tr>
<td>WakeWord-2 (Audio)</td>
<td>8</td>
<td>98.07</td>
<td>98.08</td>
<td>50.0</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>98.78</td>
<td>99.07</td>
<td></td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>99.01</td>
<td>98.96</td>
<td></td>
</tr>
</tbody>
</table>

MI-RNN better than LSTM almost always
## MI-RNN: Does It Work?

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Hidden Dim</th>
<th>LSTM</th>
<th>MI-RNN</th>
<th>Savings %</th>
</tr>
</thead>
<tbody>
<tr>
<td>GesturePod-6</td>
<td>8</td>
<td>-</td>
<td>98.00</td>
<td>50</td>
</tr>
<tr>
<td>(Gesture detection)</td>
<td>32</td>
<td>94.04</td>
<td>99.13</td>
<td></td>
</tr>
<tr>
<td></td>
<td>48</td>
<td>97.13</td>
<td>98.43</td>
<td></td>
</tr>
<tr>
<td>DSA-19</td>
<td>32</td>
<td>84.56</td>
<td>87.01</td>
<td>28</td>
</tr>
<tr>
<td>(Activity detection)</td>
<td>48</td>
<td>85.35</td>
<td>89.60</td>
<td></td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>85.17</td>
<td>88.11</td>
<td></td>
</tr>
</tbody>
</table>

MI-RNN better than LSTM almost always
## MI-RNN: Savings?

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Hidden Dim</th>
<th>LSTM</th>
<th>Hidden Dim</th>
<th>MI-RNN</th>
<th>Savings</th>
<th>Savings at ~1% drop</th>
</tr>
</thead>
<tbody>
<tr>
<td>HAR-6</td>
<td>32</td>
<td>93.04</td>
<td>16</td>
<td>93.89</td>
<td>10.5x</td>
<td>42x</td>
</tr>
<tr>
<td>Google-13</td>
<td>64</td>
<td>91.13</td>
<td>32</td>
<td>92.61</td>
<td>8x</td>
<td>32x</td>
</tr>
<tr>
<td>WakeWord-2</td>
<td>32</td>
<td>99.01</td>
<td>16</td>
<td>99.07</td>
<td>8x</td>
<td>32x</td>
</tr>
<tr>
<td>GesturePod-6</td>
<td>48</td>
<td>97.13</td>
<td>8</td>
<td>98.00</td>
<td>72x</td>
<td>-</td>
</tr>
<tr>
<td>DSA-19</td>
<td>64</td>
<td>85.17</td>
<td>32</td>
<td>87.01</td>
<td>5.5x</td>
<td>-</td>
</tr>
</tbody>
</table>
**MI-RNN: Savings?**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Hidden Dim</th>
<th>LSTM</th>
<th>Hidden Dim</th>
<th>MI-RNN</th>
<th>Savings</th>
<th>Savings at ~1% drop</th>
</tr>
</thead>
<tbody>
<tr>
<td>HAR-6</td>
<td>32</td>
<td>93.04</td>
<td>16</td>
<td>93.89</td>
<td>10.5x</td>
<td>42x</td>
</tr>
<tr>
<td>Google-13</td>
<td>64</td>
<td>91.13</td>
<td>32</td>
<td>92.61</td>
<td>8x</td>
<td>32x</td>
</tr>
<tr>
<td>WakeWord-2</td>
<td>32</td>
<td>99.01</td>
<td>16</td>
<td>99.07</td>
<td>8x</td>
<td>32x</td>
</tr>
<tr>
<td>GesturePod-6</td>
<td>48</td>
<td>97.13</td>
<td>8</td>
<td>98.00</td>
<td>72x</td>
<td>-</td>
</tr>
<tr>
<td>DSA-19</td>
<td>64</td>
<td>85.17</td>
<td>32</td>
<td>87.01</td>
<td>5.5x</td>
<td>-</td>
</tr>
</tbody>
</table>

MI-RNN achieves same or better accuracy with $\frac{1}{2}$ or $\frac{1}{4}$ of LSTM hidden dim.
MI-RNN in Action

Synthetic MNIST:
   Detecting the presence of Zero.
MI-RNN in Action
RNNs are Expensive

Solution?

Approach 2 of 2: Early Prediction

How?
Can we do even better?

- For a lot of cases, looking only at a small prefix is enough to classify/reject.

Early Prediction
Can we do even better?

• Existing work:
  – Assumes pretrained classifier and uses secondary classifiers
  – Template matching approaches
  – Separate policy for early classification

• Not feasible!
Early Prediction

Our Approach

Inference: Predict at each step – stop as soon as prediction confidence is high.

Training: Incentivize early prediction by rewarding correct and early detections.
Algorithm: E-RNN

Regular Loss: \[ L(X, y) = (W^\top h_T - y)^2 \]

Early Loss: \[ L_e(X, y) = \sum_{t=1}^{T} (W^\top h_t - y)^2 \]
Algorithm: E-RNN

Regular Loss: \[ L(X, y) = (W^\top h_T - y)^2 \]

Early Loss: \[ L_e(X, y) = \sum_{t=1}^{T} (W^\top h_t - y)^2 \]

Incentivizes early and consistent prediction.
E-RNN: How well does it work?
E-RNN: How well does it work?

• Abysmally bad 😞
E-RNN: How well does it work?

- Abysmally bad 😞
- In GesturePod-6, we lose 10-12% accuracy attempting to predict early.
E-RNN: How well does it work?

- Abysmally bad 😞
- In GesturePod-6, we lose 10-12% accuracy attempting to predict early.
- Gets confused easily due to common prefixes!
E-RNN: How well does it work?

- MI-RNN can help!
- Instances are very tight around signatures.
E-RNN: How well does it work?

- MI-RNN can help!
- Instances are very tight around signatures.
E-RNN: How well does it work?

- MI-RNN can help!
- Instances are very tight around signatures.
- Low confusion - common prefixes are small.
Algorithm: EMI-RNN

• Combine the MI-RNN training routine with E-RNN loss function and train jointly.
• Not only do you predict on smaller windows, but you predict early very often!
EMI-RNN: Results
EMI-RNN: Results

For HAR-6, we are 8x faster at 8 hidden size with better accuracy.
EMI-RNN: Results

Comparing across hidden sizes, savings amplify by 4-16x
## Raspberry Pi0

<table>
<thead>
<tr>
<th>Device</th>
<th>Hidden Dim.</th>
<th>LSTM (ms)</th>
<th>MI-RNN (ms)</th>
<th>EMI-RNN (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPi0</td>
<td>16</td>
<td>28.14</td>
<td>14.06</td>
<td>5.62</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>74.46</td>
<td>37.41</td>
<td>14.96</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>226.1</td>
<td>112.6</td>
<td>45.03</td>
</tr>
<tr>
<td>RPi3</td>
<td>16</td>
<td>12.76</td>
<td>6.48</td>
<td>2.59</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>33.10</td>
<td>16.47</td>
<td>6.58</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>92.09</td>
<td>46.28</td>
<td>18.51</td>
</tr>
</tbody>
</table>

1GHz, Single-core CPU - 512MB RAM
Conclusions and Future Work

• 8x – 72x savings with MI-RNN. Additional savings from early prediction.
• Better or match LSTM performance.
• 10x performance gain away from Arduino class devices:
  • EMI-FastGRNN
  • Rolling LSTM
Thank You!
Support Recovery for Orthogonal Matching Pursuit: Upper and Lower Bounds

Somani et al., NIPS ’18