Curriculum Learning for Handwritten Text Line Recognition

Jérôme Louradour
and Christopher Kermorvant

j1@a2ia.com
Outline

1. RNN for Handwritten Text Line Recognition
   - Offline Handwritten Text Recognition
   - Recurrent Neural Networks (RNN)

2. Curriculum Learning

3. Experiments
Outline

1. RNN for Handwritten Text Line Recognition
   - Offline Handwritten Text Recognition
   - Recurrent Neural Networks (RNN)

2. Curriculum Learning

3. Experiments
Dear Charles,

You are cordially invited to the grand opening of my new art gallery entitled "The New Era of Debates and Paintings" on July 11th, 2018.

P.S.: Your presence is obligatory due to your great help in launching my career.
Dear Charlize,

You are cordially invited to the grand opening of my new art gallery intitled "The new era of Media Music and paintings" on July 17th 2012. 

P.S: Your presence is obligatory due to your great help of launching my career.

Locating and transcribing paragraph

- Least labeling effort
- But not enough: We don’t know yet how to train a 2D recognizer!
Locating and transcribing paragraph

- Least labeling effort
- But not enough: We don’t know yet how to train a 2D recognizer!

Locating and transcribing lines

- can be done manually
- can be obtained robustly from the paragraph’s locations and content, with automatic line segmentation (and a rough text recognizer if the line breaks have not been labeled)
Locating and transcribing paragraph

- Least labeling effort
- But not enough: We don’t know yet how to train a 2D recognizer!

Locating and transcribing words

- is awkward to do manually (lot of work, ambiguities. . . )
- Automatic word segmentation cannot be robust (without text recognition and language modeling)

Even worse with characters...
Dear Charlize,

You are cordially invited to the grand opening of my new art gallery intitled "The new era of Media Music and paintings" on July 17th 2012.

P.S: Your presence is obligatory due to your great help of launching my career.

P.S: Your presence is obligatory due to your great help of launching my career.

Locating and transcribing

- Least labeling effort
- But not enough: We don’t know yet how to train a 2D recognizer!

Locating and transcribing words lines
Recurrent Neural Networks (RNN)
State-of-the-art in Handwritten Text Recognition

Task: Image (2D sequence) $\mapsto$ 1D sequence of characters

RNN Network Architecture (Graves & Schmidhuber, 2008)

- Multi-Directional layers of LSTM unit
  - “Long-Short Term Memory” – 2D recurrence in 4 possible directions
- Convolutions: parameterized subsampling layers
- Collapse layer: from 2D to 1D (output $\sim \log P$)
Recurrent Neural Networks (RNN)
State-of-the-art in Handwritten Text Recognition

Task: Image (2D sequence) \(\mapsto\) 1D sequence of characters

### RNN Network Architecture (Graves & Schmidhuber, 2008)
- Multi-Directional layers of LSTM unit
  - “Long-Short Term Memory” – 2D recurrence in 4 possible directions
- Convolutions: parameterized subsampling layers
- Collapse layer: from 2D to 1D (output \(\sim\) log \(P\))

### CTC Training ("Connectionist Temporal Classification")
- The network can output all possible symbols and also a blank output
- Minimization of the Negative Log-Likelihood \(-\log(P(Y|X))\) (NLL)
Connectionist Temporal Classification (CTC)
Deal with several possible alignments between two 1D sequences

\[ \sim - \log P(Y|X) \]

- \( U = 3 \): Number of target symbols
- \( T \): Number of RNN outputs \( \propto \) image width
- Basic decoding strategy (without lexicon neither language model):

\[ [\emptyset \ldots ]T \ldots [\emptyset \ldots ]E \ldots [\emptyset \ldots ]A \ldots [\emptyset \ldots ] \mapsto \text{"TEA"} \]
Connectionist Temporal Classification (CTC)

Deal with several possible alignments between two 1D sequences

$\sum \log P(Y|X)$

- $U = 3$: Number of target symbols
- $T$: Number of RNN outputs $\propto$ image width
- Basic decoding strategy (without lexicon neither language model):

  $[\emptyset \ldots ]T \ldots [\emptyset \ldots ]E \ldots \emptyset \ldots E \ldots [\emptyset \ldots ] \mapsto \text{"TEE"}$
Connectionist Temporal Classification (CTC)
Deal with several possible alignments between two 1D sequences

Number of possible alignments exponential with respect to $T$ and $U$!

- $\log P(Y|X)$ computed efficiently (forward-backward algo)
- But still, the task for which is trained the RNN is twofold:
  1. (classification) recognizing all the characters
  2. (localization) aligning the characters with the image stream
Connectionist Temporal Classification (CTC)

Start learning (on long sequences) is hard

Three stages when learning a RNN directly from lines of text:

1. Slow beginning: spend time without recognizing any character. Plateau which sometimes never seems to end (unlearnable?).
2. Steep acceleration: efficient learning.
3. Plateau: long time spent to finetune, to learn the details.
Connectionist Temporal Classification (CTC)

Start learning (on long sequences) is hard

\[ \text{Initialization} \]

RNN outputs

CTC gradients

"It was a splendid interpretation"
Connectionist Temporal Classification (CTC)

Start learning (on long sequences) is hard

---

Early training (plateau)

RNN outputs

CTC gradients

“It was a splendid interpretation”
Connectionist Temporal Classification (CTC)

Start learning (on long sequences) is hard

Middle of training (efficient learning)

RNN outputs

CTC gradients

“It was a splendid interpretation”
Connectionist Temporal Classification (CTC)

Start learning (on long sequences) is hard

**Finetuning**

RNN outputs

CTC gradients

"It was a splendid interpretation"
Connectionist Temporal Classification (CTC)

Start learning (on long sequences) is hard

End of training (convergence achieved)

RNN outputs

CTC gradients

"It was a splendid interpretation"
Outline

1 RNN for Handwritten Text Line Recognition
   - Offline Handwritten Text Recognition
   - Recurrent Neural Networks (RNN)

2 Curriculum Learning

3 Experiments
Curriculum Learning

General Principle

- **Key idea:** Carefully choosing training samples with a view to *start simple* and *gradually increase the complexity*.
- **Goal:** Converge faster, and hopefully generalize better in the end.
- (Elman, 1993) First application in language modeling.
- (Bengio et al., 2009) Well suited for *Stochastic Gradient Descent*.

```python
for (input, target) in Oracle():
    output = RNN.Forward(input)
    outGrad = CTC_NLL.Gradient(output, target)
    paramGrad = RNN.BackwardGradient(input, ..., outGrad)
    RNN.Update(paramGrad)
end for
```
Curriculum Learning

General Principle

- **Key idea:** Carefully choosing training samples with a view to start simple and gradually increase the complexity.
- **Goal:** Converge faster, and hopefully generalize better in the end.
- (Elman, 1993) First application in language modeling.
Curriculum Learning

Proposal

- What must be defined to apply a curriculum:
  - How to estimate if a training samples is \textit{a priori} easy?
- For text recognition and sequence transduction in general:
  - Shortest sequences are usually easier
- \textit{Example:} Training on words before training on lines.
Curriculum Learning

Proposal

• What must be defined to apply a curriculum:
  How to estimate if a training samples is \emph{a priori} easy?

• For text recognition and sequence transduction in general:
  Shortest sequences are usually easier

• \textbf{Example}: Training on words before training on lines.

RNN outputs

\begin{align*}
\text{RNN outputs} & \uparrow \\
\text{“It was a splendid interpretation”} & \uparrow
\end{align*}

CTC gradients

\begin{align*}
\text{CTC gradients} & \uparrow \\
\text{“It\_was\_a\_splendid\_interpretation”} & \uparrow
\end{align*}
Curriculum Learning

Proposal

- What must be defined to apply a curriculum:
  - How to estimate if a training samples is a priori easy?
- For text recognition and sequence transduction in general:
  - Shortest sequences are usually easier
- **Example:** Training on words before training on lines.

![RNN outputs](image1)

![CTC gradients](image2)

\[ \text{RNN outputs} \]

\[ \text{CTC gradients} \]

\[ \uparrow \]

"It"
Curriculum Learning

Proposal

- What must be defined to apply a curriculum:
  How to estimate if a training samples is *a priori* easy?

- For text recognition and sequence transduction in general:
  Shortest sequences are usually easier

- *Example:* Training on words before training on lines.
  Nice approach, but requires labeling effort.
Curriculum Learning
Proposal

- What must be defined to apply a curriculum:
  How to estimate if a training samples is \textit{a priori} easy?

- For text recognition and sequence transduction in general:
  Shortest sequences are usually easier

- \textit{Example}: Training on words before training on lines.
  Nice approach, but requires labeling effort.

- We would like a fully automated methods, to start training on short
  sequences, before switching to all sequences including the long ones.

```python
for (input, target) in Oracle():
    output = RNN.Forward(input)
    outGrad = CTC_NLL.Gradient(output, target)
    paramGrad = RNN.BackwardGradient(input, ..., outGrad)
    RNN.Update(paramGrad)
end for
```
Our “oracle” is defined by a probability to draw a training sample \{ image \mathbf{X}_t, target sequence \mathbf{Y}_t \}:

\[
P_\lambda (\text{train on } (\mathbf{X}_t, \mathbf{Y}_t)) = \frac{1}{N_\lambda} \left( \text{shortness}(\mathbf{Y}_t) \right)^\lambda
\]

where

- \text{shortness}(\mathbf{Y}_t) = \frac{1}{\max(\text{minLength}, |\mathbf{Y}_t|)} \in [0, 1]
  - \text{minLength} \geq 1 (avoid focusing on extremely short sequences)
  
  \[
  \lambda \geq 0: \text{hyper-parameter to tune how much short lines are favoured.}
  \]
  - Start with \( \lambda = 3 \)
  - Progressively move towards uniform distribution
    \( \lambda = 0 \) reached after the first 5 epochs.

- \( N_\lambda = \sum_t (\text{shortness}(\mathbf{Y}_t))^\lambda \): normalization constant
Outline

1 RNN for Handwritten Text Line Recognition
   - Offline Handwritten Text Recognition
   - Recurrent Neural Networks (RNN)

2 Curriculum Learning

3 Experiments
Experiments

Databases and performance assessment

<table>
<thead>
<tr>
<th>Database</th>
<th>Language</th>
<th># different characters</th>
<th>Training subset</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAM</td>
<td>English</td>
<td>78</td>
<td>9,462</td>
</tr>
<tr>
<td>Rimes</td>
<td>French</td>
<td>114</td>
<td>11,065</td>
</tr>
<tr>
<td>OpenHaRT</td>
<td>Arabic</td>
<td>154</td>
<td>91,811</td>
</tr>
</tbody>
</table>

Convergence curves (on validation dataset):

- **X-axis unit:** # of target characters browsed
- **Optimized cost:** Negative Log-Likelihood (NLL)

\[
\text{normNLL} = \frac{\sum_t \text{NLL} (Y_t | X_t)}{\sum_t |Y_t|}
\]

- **Character Error Rate:**

\[
\text{CER} = \frac{\sum_t \text{EditDistance}(Y_t, \hat{f}(X_t))}{\sum_t |Y_t|}
\]
### Results

<table>
<thead>
<tr>
<th></th>
<th>IAM</th>
<th>Rimes</th>
<th>OpenHaRT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline:</td>
<td>normNLL</td>
<td>0.810</td>
<td>0.367</td>
</tr>
<tr>
<td></td>
<td>CER(%)</td>
<td>21.1%</td>
<td>9.45%</td>
</tr>
<tr>
<td>Curriculum:</td>
<td>normNLL</td>
<td>0.734</td>
<td>0.341</td>
</tr>
<tr>
<td></td>
<td>CER(%)</td>
<td>18.8%</td>
<td>9.00%</td>
</tr>
</tbody>
</table>
## Experiments

### Results

<table>
<thead>
<tr>
<th></th>
<th>IAM</th>
<th>Rimes</th>
<th>OpenHaRT</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Baseline:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>norm NLL</td>
<td>0.810</td>
<td>0.367</td>
<td>0.5001</td>
</tr>
<tr>
<td>CER(%)</td>
<td>21.1%</td>
<td>9.45%</td>
<td>13.8%</td>
</tr>
<tr>
<td><strong>Curriculum:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>norm NLL</td>
<td>0.734</td>
<td>0.341</td>
<td>0.494</td>
</tr>
<tr>
<td>CER(%)</td>
<td>18.8%</td>
<td>9.00%</td>
<td>14.0%</td>
</tr>
</tbody>
</table>

Curriculum Learning for Handwritten Text Line Recognition

Jérôme Louradour and Christopher Kermorvant
Conclusions and future work

- Curriculum Learning can halve training times of RNN for text recognition of lines.
- By removing the plateau in the beginning, it also makes training supervision easier.
- It can also improve accuracy, if the training set is rather small.
- Remaining problem: how to speed up training in the last stage? (learning plateau in the end)
  - Discriminative training (switching from CTC to a more discriminative criterion)
  - Selecting samples to end training, as in active learning (select the most complicated samples)
Thank you for your attention!

Questions and comments are welcome.

jl@a2ia.com