Fast and Accurate Entity Recognition with Iterated Dilated Convolutions - Strubell et al.

M Saiful Bari

Research Assistant
School of Computer Science and Engineering
Nanyang Technological University

March 12, 2018

Outline

1. Task
   - Problem Description
   - Model
   - Problems
   - Intuitions
2. Proposed Method
   - Dilated Convolutions
   - Model Architecture
   - Results
3. Analysis

Problem Description

Task Name: Named-entity recognition.
Example: identify different kinds of noun in a sentence.

EU	rejects	German	call	to	boycott	British	lamb
B-ORG	O	I-MISC	O	O	O	I-MISC	O

Model: SOTA given by **lample** et al. in NAACL-2016. Later by **Strubell** et al in EMNLP 2017 (only speed gain).

- Lample et al : Bi-directional RNN at character and Word Level.
- Strubell et al : CNN at character(?) and ID-CNN at Word Level.

Model

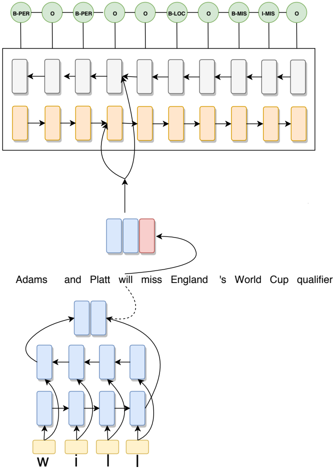


Figure: Model proposed by lample et al.

Probability Calculation

- **Softmax**:

$$tag[i] = \frac{e^{logit[i]}}{\sum_{j=1}^{NumberOfClass} e^{logit[j]}}$$

where $tag[i]$ is the **local probability**.

- **Linear-chain CRF:** Calculates a global score C.

$$C(y_1, \ldots, y_m) = b[y_1] \quad + \sum_{t=1}^{m} s_t[y_t] + \sum_{t=1}^{m-1} T[y_t, y_{t+1}] \quad + e[y_m]$$

$$= \text{begin} \quad + \text{scores} + \text{transitions} \quad + \text{end}$$

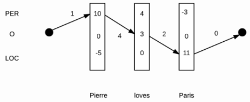
T is a trainable parameter.

Probability Calculation

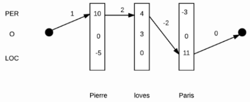
- **Linear-chain CRF:** Calculates a global score C.

$$C(y_1, \ldots, y_m) = b[y_1] \quad + \sum_{t=1}^{m} s_t[y_t] + \sum_{t=1}^{m-1} T[y_t, y_{t+1}] \quad + e[y_m]$$

$$= \text{begin} \qquad + \text{scores} + \text{transitions} \qquad + \text{end}$$



The path PER-O-LOC has a score of
$1 + 10 + 4 + 3 + 2 + 11 + 0 = 31$



The path PER-PER-LOC has a score of
$1 + 10 + 2 + 4 - 2 + 11 + 0 = 26$

Figure: Taking local probability won't help

Problems

- Training time is quite long (ConLL dataset(english) training time: 8-12 HOUR)
- Structurally RNN does not support better distributed computing (compared to CNN).
- Training with CRF layer is costly. $O(NumberOfClass^2 * seqLen)$ (Need Dynamic Programming)

$$\tilde{s}_t(y_t) = \text{argmax}_{y_t,\ldots,y_m} C(y_t,\ldots,y_m)$$
$$= \text{argmax}_{y_{t+1}} s_t[y_t] + T[y_t, y_{t+1}] + \tilde{s}_{t+1}(y^{t+1})$$

$$\mathbb{P}(y_1,\ldots,y_m) = \frac{e^{C(y_1,\ldots,y_m)}}{Z}$$

$$Z = \sum_{y_1,\ldots,y_m} e^{C(y_1,\ldots,y_m)}$$

- It takes lower time to train CRF layer in CPU than GPU

Intuitions

Original model trained by **lample** et al.: Number of parameter is not so high.

Average sentence length (According to PENN TreeBank) is around 23. (not so high)

Can we set an end-to-end training method with a CNN.

Outline

Dilated Convolutions

For sequence label training we can use **Dilated Convolutional** Neural Network (Fisher et al).

- Broader view of the input to capture more contextual information.
- Works better with less parameters.

Dilated Convolutions for image, according to original author

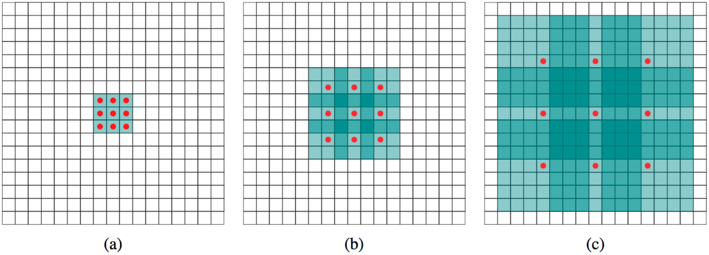


Figure 1: Systematic dilation supports exponential expansion of the receptive field without loss of resolution or coverage. (a) F_1 is produced from F_0 by a 1-dilated convolution; each element in F_1 has a receptive field of 3×3. (b) F_2 is produced from F_1 by a 2-dilated convolution; each element in F_2 has a receptive field of 7×7. (c) F_3 is produced from F_2 by a 4-dilated convolution; each element in F_3 has a receptive field of 15×15. The number of parameters associated with each layer is identical. The receptive field grows exponentially while the number of parameters grows linearly.

Dilated Convolutions for language model

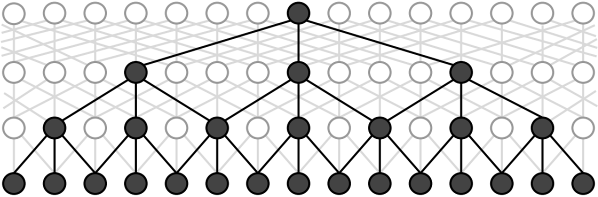


Figure: A dilated CNN block with maximum dilation width 4 and filter width 3.

$r = l(w-1)+1$
where , r number of node covered at layer l with filter width w.
total number of node at layer l is, $2^{l+1}-1$ (**exponential growth**).

Dilated Convolutions for language model

$r = l(w-1)+1$

where , r number of node covered at layer l with filter width w.

total number of node at layer l is, $2^{l+1} - 1$(**exponential growth**).

Just four stacked dilated convolutions of width 3 produces token representations with a n effective input width of 31 tokens.

greater than the average sentence length of PENN TreeBank(23)

Regular convolution,

$$c_t = W_c \bigoplus_{k=0}^{r} x_{t \pm k}$$

Dilated Convolution,

$$c_t = W_c \bigoplus_{k=0}^{r} x_{t \pm k\delta}$$

Output Resolution

- To avoid this scaling, one could pool representations across the sequence, but this is not appropriate for sequence labeling
- it reduces the output resolution of the representation.

	f1	f2	f3	f4
max_pool ->				
max_pool ->				
max_pool ->				
max_pool ->				

Model Architecture

We denote the j^{th} dilated convolutional layer of dilation width δ as $D_\delta(j)$
The first layer in the network is a dilation-1 convolution $D_1^(0))$.

$$i_t = D_1^{(0)} x_t$$

Next, L_c layers of dilated convolutions of exponentially increasing dilation width are applied to i_t.

$$c_t^{(0)} = i_t$$

$$c_t^{(j)} = relu(D_{2^{L_C - 1}}^{(j-1)} c_t^{(j-1)})$$

add a final dilation-1 layer to the stack

$$c_t^{L_c+1} = relu(D_1^{(L_C)} c_t^{(L_C)})$$

Model Architecture

We refer to this stack of dilated convolutions as a block $B()$.
To avoid overfitting we stack the block on top of another so that the number of parameter does not increase.
We iteratively apply $B()$ L_b times

$$b_t^{(1)} = B(i_t)$$

$$b_t^k = B(b_t^{(k-1)})$$

$$h_t^{(L_b)} = W_0 b_t^{(L_b)}$$

loss calculation

Cross-entropy loss,

$$\frac{1}{T}\sum_{t=1}^{T}\log P(y_t|h_t^{(L_b)})$$

Iterative-dialated loss,

$$\frac{1}{L_b}\sum_{k=1}^{L_b}\frac{1}{T}\sum_{t=1}^{T}\log P(y_t|h_t^{(k)})$$

By rewarding accurate predictions after each application of the block, we learn a model where later blocks are used to refine initial predictions.

Results

Model	F1
Ratinov and Roth (2009)	86.82
Collobert et al. (2011)	86.96
Lample et al. (2016)	90.33
Bi-LSTM	89.34 ± 0.28
4-layer CNN	89.97 ± 0.20
5-layer CNN	90.23 ± 0.16
ID-CNN	90.32 ± 0.26
Collobert et al. (2011)	88.67
Passos et al. (2014)	90.05
Lample et al. (2016)	90.20
Bi-LSTM-CRF (re-impl)	90.43 ± 0.12
ID-CNN-CRF	$\mathbf{90.54} \pm \mathbf{0.18}$

Figure: F1 Score comparison with existing model. (Without character level model)

Results

Model	Speed
Bi-LSTM-CRF	$1\times$
Bi-LSTM	$9.92\times$
ID-CNN-CRF	$1.28\times$
5-layer CNN	$12.38\times$
ID-CNN	$14.10\times$

Figure: Time comparison to get results.

Adding document-level context improves every model.

Outline

1. Task
   - Problem Description
   - Model
   - Problems
   - Intuitions
2. Proposed Method
   - Dilated Convolutions
   - Model Architecture
   - Results
3. Analysis

Analysis

- Sequence information can be learned by ID-CNN.
- Experiment setup is not clear in the paper.
- They didn't specify the improvement of iterative training in the result section.
- They didn't specify what is document-level context (may be char level model).
- Follow up : Training RNNs as Fast as CNNs Lei et al.
- Link to the code.