Deep Residual Learning for Weakly-Supervised Relation Extraction

Yi Yao Huang
Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan
b02901042@ntu.edu.tw

William Yang Wang
Department of Computer Science
University of California, Santa Barbara
Santa Barbara, CA 93106 USA
william@cs.ucsb.edu

Abstract

Deep residual learning (ResNet) (He et al., 2016) is a new method for training very deep neural networks using identity mapping for shortcut connections. ResNet has won the ImageNet ILSVRC 2015 classification task, and achieved state-of-the-art performances in many computer vision tasks. However, the effect of residual learning on noisy natural language processing tasks is still not well understood. In this paper, we design a novel convolutional neural network (CNN) with residual learning, and investigate its impacts on the task of distantly supervised noisy relation extraction. In contradictory to popular beliefs that ResNet only works well for very deep networks, we found that even with 9 layers of CNNs, using identity mapping could significantly improve the performance for distantly-supervised relation extraction.

1 Introduction

Relation extraction is the task of predicting attributes and relations for entities in a sentence (Zelenko et al., 2003; Bunescu and Mooney, 2005; GuoDong et al., 2005). For example, given a sentence *"**Barack Obama** was born in **Honolulu**, Hawaii."*, a relation classifier aims at predicting the relation of *"**bornInCity**"*. Relation extraction is the key component for building relation knowledge graphs, and it is of crucial significance to natural language processing applications such as structured search, sentiment analysis, question answering, and summarization.

A major issue for relation extraction is the lack of labeled training data. In recent years, distant supervision (Mintz et al., 2009; Hoffmann et al., 2011; Surdeanu et al., 2012) emerges as the most popular method for relation extraction— it uses knowledge base facts to select a set of noisy instances from unlabeled data. Among all the machine learning approaches for distant supervision, the recently proposed Convolutional Neural Networks (CNNs) model (Zeng et al., 2014) achieved the state-of-the-art performance. Following their success, Zeng et al. (2015) proposed a piece-wise max-pooling strategy to improve the CNNs. Various attention strategies (Lin et al., 2016; Shen and Huang, 2016) for CNNs are also proposed, obtaining impressive results. However, most of these neural relation extraction models are relatively shallow CNNs—typically only one convolutional layer and one fully connected layer were involved, and it was not clear whether deeper models could have benefits on distilling signals from noisy inputs in this task.

In this paper, we investigate the effects of training deeper CNNs for distantly-supervised relation extraction. More specifically, we designed a convolutional neural network based on residual learning (He et al., 2016)—we show how one can incorporate word embeddings and position embeddings into a deep residual network, while feeding identity feedback to convolutional layers for this noisy relation prediction task. Empirically, we evaluate on the NYT-Freebase dataset (Riedel et al., 2010), and demonstrate the state-of-the-art performance using deep CNNs with identify mapping and shortcuts. In contrast to popular beliefs in vision that deep residual network only works for very deep CNNs, we show that even with a moderately deep CNNs, there are substantial improvements over vanilla CNNs for relation extraction. Our contributions are three-fold:

- We are the first to consider deeper convolutional neural networks for weakly-supervised relation extraction using residual learning;

- We show that our deep residual network model outperforms CNNs by a large margin empirically, obtaining state-of-the-art performances;
- Our identity mapping with shortcut feedback approach can be easily applicable to any variants of CNNs for relation extraction.

2 Deep Residual Networks for Relation Extraction

In this section, we describe a novel deep residual learning architecture for distantly supervised relation extraction. Figure 1 describes the architecture of our model.

2.1 Vector Representation

Let $\mathbf{x}_i$ be the *i*-th word in the sentence and *e1*, *e2* be the two corresponding entities. Each word will access two embedding look-up tables to get the word embedding $\mathbf{WF}_i$ and the position embedding $\mathbf{PF}_i$. Then, we concatenate the two embeddings and denote each word as a vector of $\mathbf{v}_i = [\mathbf{WF}_i, \mathbf{PF}_i]$.

2.1.1 Word Embeddings

Each representation $\mathbf{v}_i$ corresponding to $\mathbf{x}_i$ is a real-valued vector. All of the vectors are encoded in an embeddings matrix $\mathbf{V}_w \in \mathbb{R}^{d_w \times |V|}$ where V is a fixed-sized vocabulary.

2.1.2 Position Embeddings

In relation classification, we focus on finding a relation for entity pairs. Following (Zeng et al., 2014), a PF is the combination of the relative distances of the current word to the first entity e_1 and the second entity e_2. For instance, in the sentence "*Steve_Jobs is the founder of Apple.*", the relative distances from *founder* to e_1 (*Steve_Job*) and e_2 are 3 and -2, respectively. We then transform the relative distances into real valued vectors by looking up one randomly initialized position embedding matrices $\mathbf{V}_p \in \mathbb{R}^{d_p \times \|P\|}$ where P is fixed-sized distance set. It should be noted that if a word is too far from entities, it may be not related to the relation. Therefore, we choose maximum value e_{max} and minimum value e_{min} for the relative distance.

In the example shown in Figure 1, it is assumed that d_w is 4 and d_p is 1. There are two position embeddings: one for e_1, the other for e_2. Finally, we concatenate the word embeddings and position embeddings of all words and denote a sentence of length *n* (padded where necessary) as a vector

$$\mathbf{v} = \mathbf{v}_1 \oplus \mathbf{v}_2 \oplus ... \oplus \mathbf{v}_n$$

where $\oplus$ is the concatenation operator and $\mathbf{v}_i \in \mathbb{R}^d$ ($d = d_w + d_p \times 2$).

2.2 Convolution

Let $\mathbf{v}_{i:i+j}$ refer to the concatenation of words $\mathbf{v}_i, \mathbf{v}_{i+1}, ..., \mathbf{v}_{i+j}$. A convolution operation involves a *filter* $\mathbf{w} \in \mathbb{R}^{hd}$, which is applied to a window of *h* words to produce a new feature. A feature c_i is generated from a window of word $\mathbf{v}_{i:i+h-1}$ by

$$c_i = f(\mathbf{w} \cdot \mathbf{x}_{i:i+h-1} + b)$$

Here $b \in \mathbb{R}$ is a bias term and f is a non-linear function. This filter is applied to each possible window of words from $\mathbf{v}_1$ to $\mathbf{v}_n$ to produce *feature* $\mathbf{c} = [c_1, c_2, ..., c_{n-h+1}]$ with $\mathbf{c} \in \mathbb{R}^s (s = n - h + 1)$.

2.3 Residual Convolution Block

Residual learning connects low-level to high-level representations directly, and tackles the vanishing gradient problem in deep networks. In our model, we design the residual convolution block by applying shortcut connections. Each residual convolutional block is a sequence of two convolutional layers, each one followed by an ReLU activation. The kernel size of all convolutions is h, with padding such that the new feature will have the same size as the original one. Here are two convolutional *filter* $\mathbf{w}_1$, $\mathbf{w}_2 \in \mathbb{R}^{h \times 1}$. For the first convolutional layer:

$$\tilde{c}_i = f(\mathbf{w}_1 \cdot c_{i:i+h-1} + b_1)$$

For the second convolutional layer:

$$\acute{c}_i = f(\mathbf{w}_2 \cdot \tilde{c}_{i:i+h-1} + b_2)$$

Here b_1, b_2 are bias terms. For the residual learning operation:

$$\mathbf{c} = \mathbf{c} + \acute{\mathbf{c}}$$

Conveniently, the notation of $\mathbf{c}$ on the left is changed to define as the output vectors of the block. This operation is performed by a shortcut connection and element-wise addition. This block will be multiply concatenated in our architecture.

2.4 Max Pooling, Softmax Output

We then apply a max-pooling operation over the *feature* and take the maximum value $\hat{c} = max\{\mathbf{c}\}$.

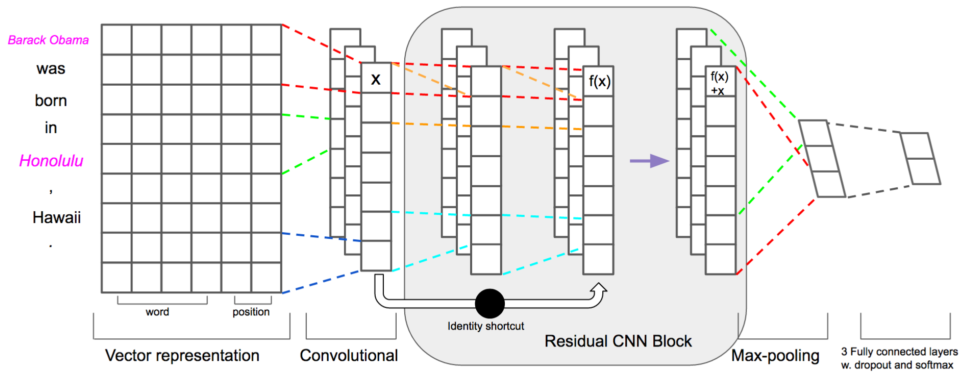

Figure 1: The architecture of ResCNN used for relation extraction.

We have described the process by which *one* feature is extracted from *one* filter. Take all features into one high level extracted feature $\mathbf{z} = [\hat{c}_1, \hat{c}_2, ..., \hat{c}_m]$(note that here we have m filters). Then, these features are passed to a fully connected softmax layer whose output is the probability distribution over relations. Instead of using $y = \mathbf{w} \cdot \mathbf{z} + b$ for output unit y in forward propagation, dropout uses $y = \mathbf{w} \cdot (\mathbf{z} \circ \mathbf{r}) + b$ where $\circ$ is the element-wise multiplication operation and $\mathbf{r} \in \mathbb{R}^m$ is a 'masking' vector of Bernoulli random variables with probability p of being 1. In the test procedure, the learned weight vectors are scaled by p such that $\hat{\mathbf{w}} = p\mathbf{w}$ and used (without dropout) to score unseen instances.

3 Experiments

3.1 Experimental Settings

In this paper, we use the word embeddings released by (Lin et al., 2016) which are trained on the NYT-Freebase corpus (Riedel et al., 2010). We fine tune our model using validation on the training data. The word embedding is of size 50. The input text is padded to a fixed size of 100. Training is performed with tensorflow adam optimizer, using a mini-batch of size 64, an initial learning rate of 0.001. We initialize our convolutional layers following (Glorot and Bengio, 2010). The implementation is done using Tensorflow 0.11. All experiments are performed on a single NVidia Titan X (Pascal) GPU. In Table 1 we show all parameters used in the experiments.
We experiment with several state-of-the-art baselines and variants of our model.

Window size h	3
Word dimension d_w	50
Position dimension d_p	5
Position maximum distance e_{max}	30
Position minimum distance e_{min}	-30
Number of filters m	128
Batch size B	64
Learning rate λ	0.001
Dropout probability p	0.5

Table 1: Parameter settings

- **CNN-B**: Our implementation of the CNN baseline (Zeng et al., 2014) which contains one convolutional layer, and one fully connected layer.
- **CNN+ATT**: CNN-B with attention over instance learning (Lin et al., 2016).
- **PCNN+ATT**: Piecewise CNN-B with attention over instance learning (Lin et al., 2016).
- **CNN**: Our CNN model which includes one convolutional layer and three fully connected layers.
- **CNN-x**: Deeper CNN model which has x convolutional layers. For example, CNN-9 is a model constructed with 9 convolutional layers (1 + 4 residual cnn block without identity shortcut) and three fully connected layers.
- **ResCNN-x**: Our proposed CNN-x model with residual identity shortcuts.

We evaluate our models on the widely used NYT freebase larger dataset (Riedel et al., 2010). Note that ImageNet dataset used by the original ResNet paper (He et al., 2016) has 1.28 million training instances. NYT freebase dataset includes 522K training sentences, which is the largest dataset in relation extraction, and it is the only suitable dataset to train deeper CNNs.

3.2 NYT-Freebase Dataset Performance

The advantage of this dataset is that there are 522,611 sentences in training data and 172,448 sentences in testing data and this size can support

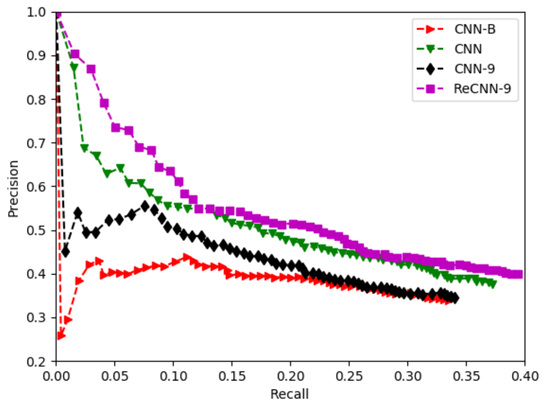

Figure 2: Comparing ResCNN to different CNNs.

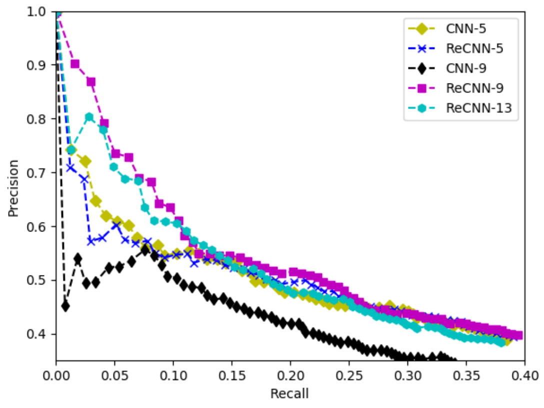

Figure 3: Varying the depths of our models.

us to train a deep network. Similar to previous work (Zeng et al., 2015; Lin et al., 2016), we evaluate our model using the held-out evaluation. We report both the aggregate curves precision/recall curves and Precision@N (P@N).

In Figure 2, we compare the proposed ResCNN model with various CNNs. First, CNNs with multiple fully-connected layers obtained very good results, which is a novel finding. Second, the results also suggest that deeper CNNs with residual learning help extracting signals from noisy distant supervision data. We observe that overfitting happened when we try to add more layers and the performance of CNN-9 is much worse than CNN. We find that ResNet can solve this problem and ResCNN-9 obtains better performance as compared to CNN-B and CNN and dominates the precision/recall curve overall.

We show the effect of depth in residual networks in Figure 3. We observe that ResCNN-5 is worse than CNN-5 because the ResNet does not work well for shallow CNNs, and this is consistent with the original ResNet paper. As we increase the network depth, we see that CNN-9 does overfit to the training data. With residual learning, both ResCNN-9 and ResCNN-13 provide significant improvements over CNN-5 and ResCNN-5 models. In contradictory to popular beliefs that ResNet only works well for very deep networks, we found that even with 9 layers of CNNs, using identity mapping could significantly improve the performance learning in a noisy input setting.

P@N(%)	100	200	300	Mean
CNN+ATT	76.2	68.6	59.8	68.2
PCNN+ATT	**76.2**	**73.1**	**67.4**	**72.2**
CNN-B	41.0	40.0	41.0	40.7
CNN	64.0	61.0	55.3	60.1
CNN-5	64.0	58.5	54.3	58.9
ResCNN-5	57.0	57.0	54.3	56.1
CNN-9	56.0	54.0	49.7	53.2
ResCNN-9	**79.0**	**69.0**	**61.0**	**69.7**
ResCNN-13	76.0	65.0	60.3	67.1

Table 2: P@N for relation extraction with different models. Top: models that select training data. Bottom: models without selective attention.

The intuition of ResNet help this task in two aspect. First, if the lower, middle, and higher levels learn hidden lexical, syntactic, and semantic representations respectively, sometimes it helps to bypass the syntax to connect lexical and semantic space directly. Second, ResNet tackles the vanishing gradient problem which will decrease the effect of noise in distant supervision data.

In Table 2, we compare the performance of our models to state-of-the-art baselines. We show that our ResCNN-9 outperforms all models that do not select training instances. And even without piecewise max-pooling and instance-based attention, our model is on par with the PCNN+ATT model.

For the more practical evaluation, we compare the results for precision@N where N is small (1, 5, 10, 20, 50) in Table 3. We observe that our ResCNN-9 model dominates the performance when we predict the relation in the range of higher probability. ResNet helps CNNs to focus on the highly possible candidate and mitigate the noise effect of distant supervision. We believe that residual connections actually can be seen as a form of renormalizing the gradients, which prevents the model from overfitting to the noisy distant supervision data.

In our distant-supervised relation extraction experience, we have two important observations: (1) We get significant improvements with CNNs adding multiple fully-connected layers. (2) Resid-

P@N(%)	1	5	10	20	50
PCNN+ATT	**1**	0.8	**0.9**	0.75	0.7
ResCNN-9	**1**	**1**	**0.9**	**0.9**	**0.88**

Table 3: P@N for relation extraction with different models where N is small. We get the result of PCNN+ATT using their public source code.

ual learning could significantly improve the performance for deeper CNNs.

4 Conclusion

In this paper, we introduce a deep residual learning method for distantly-supervised relation extraction. We show that deeper convolutional models help distill signals from noisy inputs. With shortcut connections and identify mapping, the performances are significantly improved. These results aligned with a recent study (Conneau et al., 2017), suggesting that deeper CNNs do have positive effects on noisy NLP problems.

References

Razvan Bunescu and Raymond J Mooney. 2005. Subsequence kernels for relation extraction. In *NIPS*, pages 171–178.

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2017. Very deep convolutional networks for natural language processing. *EACL 2017*.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 249–256.

Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 427–434. Association for Computational Linguistics.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 541–550. Association for Computational Linguistics.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *ACL 2016*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer.

Yatian Shen and Xuanjing Huang. 2016. Attention-based convolutional neural network for semantic relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 455–465. Association for Computational Linguistics.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *Journal of machine learning research*, 3(Feb):1083–1106.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP), Lisbon, Portugal*, pages 17–21.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, Jun Zhao, et al. 2014. Relation classification via convolutional deep neural network. In *COLING*, pages 2335–2344.