Siamese Neural Networks for One-shot Image Recognition

Gregory Koch & Richard Zemel & Ruslan Salakhutdinov

Department of Computer Science, University of Toronto. Toronto, Ontario, Canada

동국대학교 컴퓨터공학과
최수정

목차

- 배경
- One-shot learning
- Approach
- Deep Siamese Networks for Image Verification
- Model
- Learning
- The Omniglot Dataset
- Verification
- 20-way-one-shot classification
- MNIST One-shot Trial

배경

- Machine learning은 web search, spam detection, caption generation , and speech and image recognition 등에서 매우 성공적으로 사용되었다.
- **그러나**, supervised information이 적은 data에 대해서는 잘 예측하지 못하였다.
- 새로운 데이터에 대해 retraining 해야 하거나 또는 limited data로 인해 retraining가 불가능하다는 이러한 불편함에서 연구를 시작하게 되었다.

One-shot learning

- **one-shot learning** : 클래스에 대해 1개의 example만 제공한다는 제약조건에서 classification 하는 것이다.
- 도매인의 특정한 성질(domain-specific features)을 성립하거나 or 목표 과제에 따라 매우 다른 추론 절차를 통해 다루어 질 수 있다. → 이러한 방식은 다른 타입의 문제에 대해서는 문제가 될 수 있다.
- ▶ **input의 구조에 관련된 가정을 제한하고, few examples로 부터 성공적으로 모델을 생산할 수 있는 성질(features)를 자동으로 획득하는 one-shot learning 방법을 제안**한다.
- 이러한 방식은 학습 알고리즘 자체의 cost가 상당할 수 있지만, priors를 부과하지 않아 매우 강력하다고 할 수 있다.

Approach

- character recognition에 초점
- 1) Unknown class에 대해 예측을 할 수 있도록 **일반적인 이미지의 특성(features)을 학습**할 수 있고,
- 2) 기본적인 최적화 기술을 이용해 **데이터 쌍을 쉽게 학습**할 수 있는,
- **3) 도매인의 특정한 성질(domain-specific features)에 의존하지 않는** 경쟁적 접근을 제공하는
- 1)~3) Large Siamese convolutional neural networks를 사용한다.
- One-shot image classification 모델을 구축하지 위해 먼저, image 쌍의 class-identity를 차별(구별)할 수 있는 신경망을 학습시켰다.

Deep Siamese Networks for Image Verification

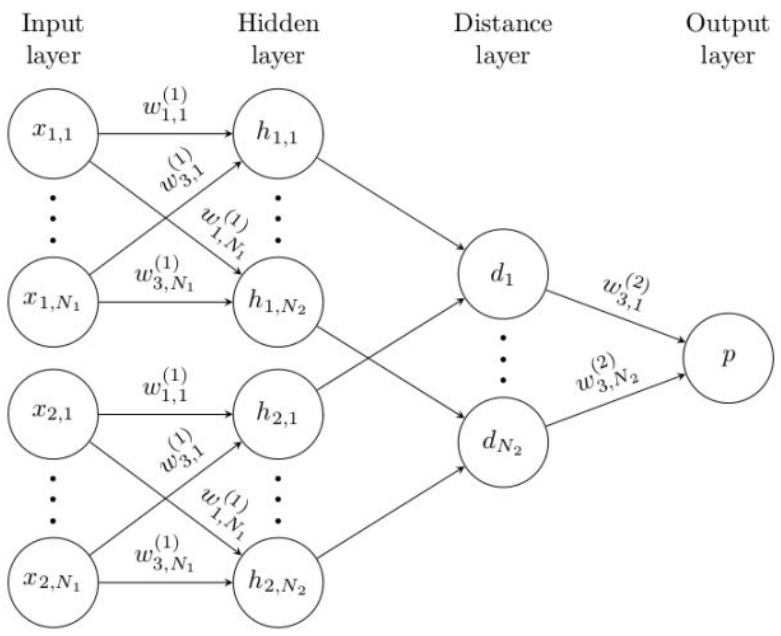

Figure 3. A simple 2 hidden layer siamese network for binary classification with logistic prediction p. The structure of the network is replicated across the top and bottom sections to form twin networks, with shared weight matrices at each layer.

- <u>Twin network</u>로 이루어져 있다.
- <u>Symmetric</u>하기 때문에 Input 의 순서를 (B,A)에서 (A,B)로 바꿔도 output값은 동일하다.
- Input 2개가 매우 유사하면 feature space에서 매우 가까이 위치한다.
- h1과 h2 feature 벡터 사이의 L1 거리를 계산하고, activation function으로 sigmoid를 사용해 [0,1] 범위의 output을 반환한다.

Model

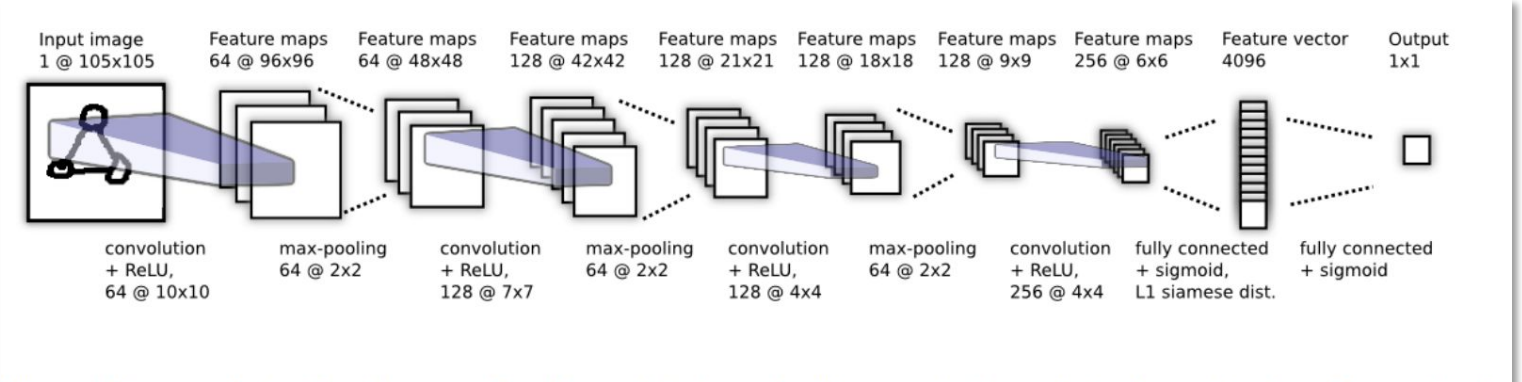

Figure 4. Best convolutional architecture selected for verification task. Siamese twin is not depicted, but joins immediately after the 4096 unit fully-connected layer where the L1 component-wise distance between vectors is computed.

- 본 연구에서 사용한 모델은 L layers를 가지는 **Siamese CNN**이다.
- fully-connected layers를 하기 전에 multiple CNN을 사용하여 **공간적 특성(spatial features)**를 얻는다.
- 첫번째 L-2 layers에서 output에 대해 ReLU를 사용하고, 나머지는 sigmoidal units를 사용한다.
- Convolution layer는 다양한 크기이면서 stride는 1로 고정된 Filters, ReLU activate function으로 구성되고 optimal하게 max-pooling을 사용한다.

Model

$$a_{1,m}^{(k)} = \text{max-pool}(\max(0, \mathbf{W}_{l-1,l}^{(k)} \star \mathbf{h}_{1,(l-1)} + \mathbf{b}_l), 2)$$
$$a_{2,m}^{(k)} = \text{max-pool}(\max(0, \mathbf{W}_{l-1,l}^{(k)} \star \mathbf{h}_{2,(l-1)} + \mathbf{b}_l), 2)$$

- 위의 식은 <u>각 layer에서 k번째 filter map</u>을 나타낸 것이다.
- $W_{l-1,l}$: layer l에 대한 <u>feature maps를 3차원으로 나타낸 tensor</u>
- ★ : 각 convolutional filter와 input feature maps 사이의 출력 단위만 반환하는 <u>valid convolutional operation</u>.
- 마지막 convolutional layer에서 units은 single vector로 flat되고. 그후에 fully-connected layer로 연결된다.

$$\boldsymbol{p} = \sigma(\sum_j \alpha_j |h_{1,L-1}^{(j)} h_{2,L-1}^{(j)}|)$$

- 마지막 layer에서 위의 식을 통해 두 개의 거리를 계산하고, [0,1] 범위의 값을 반환한다.
- 즉, 두 feature vector 의 similarity를 계산한다.

Learning

Loss Function : regularized cross-entropy

- M : minibatch size
- i : indexs
- $y({x_1}^{(i)}, {x_2}^{(i)})$: minibatch에 대한 labels를 포함한 length-M vector
- $y\left({x_1}^{(i)}, {x_2}^{(i)}\right) = 1$: same class
- $y\left({x_1}^{(i)}, {x_2}^{(i)}\right) = 0$: different class

$$\mathcal{L}(x_1^{(i)}, x_2^{(i)}) = \mathbf{y}(x_1^{(i)}, x_2^{(i)}) \log \mathbf{p}(x_1^{(i)}, x_2^{(i)}) + (1 - \mathbf{y}(x_1^{(i)}, x_2^{(i)})) \log (1 - \mathbf{p}(x_1^{(i)}, x_2^{(i)})) + \boldsymbol{\lambda}^T |\mathbf{w}|^2$$

Learning

Optimization

- minibatch size : 128
- Learning rate : η_j
- Momentum : μ_j
- L2 regularization weights : λ_j

$$\mathbf{w}_{kj}^{(T)}(x_1^{(i)}, x_2^{(i)}) = \mathbf{w}_{kj}^{(T)} + \Delta\mathbf{w}_{kj}^{(T)}(x_1^{(i)}, x_2^{(i)}) + 2\lambda_j|\mathbf{w}_{kj}|$$

$$\Delta\mathbf{w}_{kj}^{(T)}(x_1^{(i)}, x_2^{(i)}) = -\eta_j \nabla w_{kj}^{(T)} + \mu_j \Delta\mathbf{w}_{kj}^{(T-1)}$$

Weight initialization : normal distribution with zero-mean and std

- Convolutional layer : weight = (0,0.01) , Biases = (0.5,0.01)
- Fully-connected layer : weight = (0,0.2) , Biases = (0.5, 0.01)

Learning

Learning schedule

- Learning rate : 1 epoch 당 1%씩 감소 (식 = $\eta_j{}^{(T)} = 0.99\eta_j{}^{(T-1)}$)
- Momentum : 모든 layer에서 0.5로 시작하고 μ_j 보다 작을 때까지 매 epoch마다 선형적으로 증가.
- 각 네트워크마다 최대 200epochs 학습하고, 320개의 one-shot learning task set에 대해 error를 모니터한다.
- 20epochs동안 error가 감소하지 않는다면 학습을 멈추고, 해당 파라미터를 사용한다.
- Error가 계속 감소한다면 마지막 파라미터를 사용한다.

Learning

Hyperparameter optimization

- Hyperparameter selection : a Bayesian optimization framework를 사용
- Learning rate : $\eta_j \in [10^{-4}, 10^{-1}]$
- Momentum : $\mu_j \in [0,1]$
- L2 regularization weights : $\lambda_j \in [0,0.1]$
- Convolutional filters size $\in [3 \times 3, 20 \times 20]$
- Number of convolutional filters $\in [16, 256]$ (multiples of 16)
- Fully-connected layers $\in [128, 4096]$ (multiples of 16)

Learning

Affine distortions

- Training set을 구성할 때, 조금의 왜곡을 주었다. (Figure 5 참고)

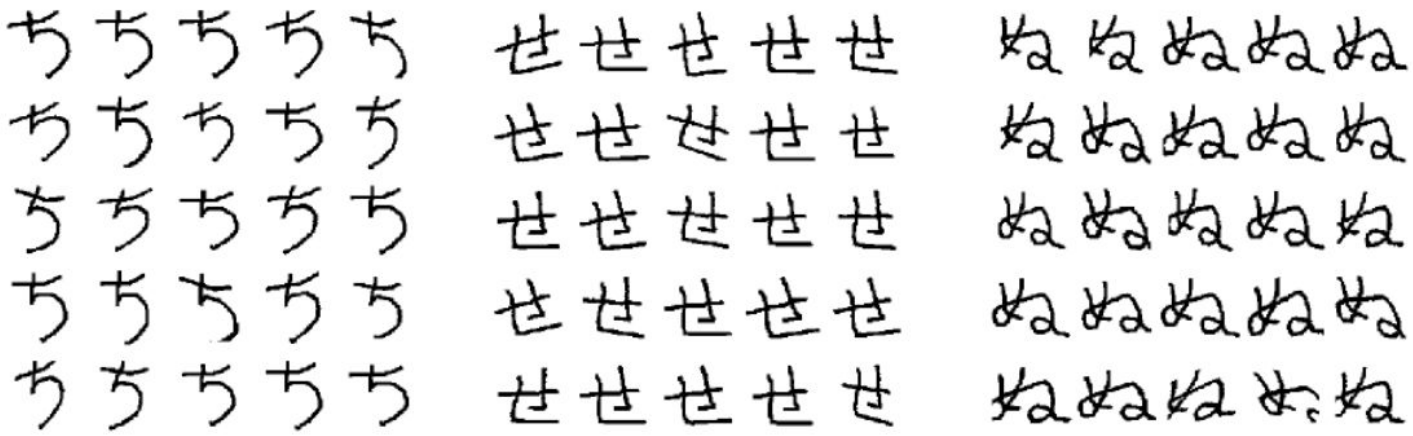

Figure 5. A sample of random affine distortions generated for a single character in the Omniglot data set.

The Omniglot Dataset

- Handwritten character recognition domain에서 사용한다.
- 50개의 알파벳으로 구성되고, 각 알파벳의 글자수는 15~40자까지 다양하다.
- 40개의 배경 set : 모델을 구축할 때 사용된다.
- 10개의 평가 set : one-shot classification 성능을 측정하기 위해 사용된다.

Figure 6. The Omniglot dataset contains a variety of different images from alphabets across the world.

Verification

- Training data : 60% (alphabets : 30/50 , drawers : 12/20)
- 랜덤 sampling을 통해 same/different pairs가 30k, 90k, 150k개 인 training data set를 만든다.
- 8 종류의 왜곡을 통해 이미지 데이터를 증폭한다.
- 즉 270k, 810k, 1350k 개의 이미지쌍이 생성된다.

Table 1. Accuracy on Omniglot verification task (siamese convolutional neural net)

Method	Test
30k training	
no distortions	90.61
affine distortions x8	91.90
90k training	
no distortions	91.54
affine distortions x8	93.15
150k training	
no distortions	91.63
affine distortions x8	**93.42**

20-way-one-shot classification

- Verification task를 master한 Siamese network를 얻었다.
- 20-way-one-shot classification task로 평가한다.

$$C^* = \mathrm{argmax}_c \mathbf{p}^{(c)}$$

- Test 이미지 x 와 , 20개의 class의 이미지 x_c 를 이용해 가장 유사도가 높은 class를 예측한다.
- 성능을 비교하기 위해 Lake, Brenden M, Salakhutdinov, Ruslan R, and Tenenbaum, Josh. One-shot learning by inverting a compositional causal process. 논문의 결과를 가져왔다.
- 정확도는 HBPL과 Humans를 제외하고 가장 높은 **92.0%** 이다.

Table 2. Comparing best one-shot accuracy from each type of network against baselines.

Method	Test
Humans	95.5
Hierarchical Bayesian Program Learning	95.2
Affine model	81.8
Hierarchical Deep	65.2
Deep Boltzmann Machine	62.0
Simple Stroke	35.2
1-Nearest Neighbor	21.7
Siamese Neural Net	58.3
Convolutional Siamese Net	92.0

MNIST One-shot Trial

- Omniglot에 대해 학습시킨 모델이 MNIST에 대해 잘 일반화되는지 test해보고 싶다.
- 10-way-one-shot classification task를 진행하였다 .
- Omniglot과 유사하게 진행하였고, 위에서 봤던 92%보다는 낮지만 70.3%라는 합리적인 결과가 나왔다.

Table 3. Results from MNIST 10-versus-1 one-shot classification task.

Method	Test
1-Nearest Neighbor	26.5
Convolutional Siamese Net	70.3

감사합니다 ☺