Keras를 활용한 ResNet 모델의 구현과 비교를 통한 이미지 분류

김설아
jena1218@naver.com

Abstract

최근 발전하고 있는 CNN 모델 중 ResNet은 파라미터가 많음에도, 인간의 인지 능력 오류 정확도보다 우수한 정확도를 달성한 모델이다. 본 논문에서는 Keras를 활용하여 ResNet 모델을 구현하고 분류 데이터셋에서 각 모델의 성능을 비교하였다. 실험 결과, ResNet-50 모델은 다른 모델들보다 우수한 성능을 보였으며, 이러한 결과를 통해 설명 가능한 인공지능 개발에 더 넓게 적용될 수 있음을 시사하였다. 추가적인 연구를 통해 다양한 데이터셋에서의 평가와 ResNeXt와 같은 모델의 적용을 고려할 필요가 있다.

1. Introduction

Convolutional Neural Network(이하 CNN) 네트워크 발전의 초창기에는 정확도를 개선하는 데에 초점이 맞추어져 있었으나 최근에는 파라미터 수와 연산량을 최소화 하고, 모바일 시스템에 적용하기 위해 개선되고 있다. 특히, 모델 크기, 속도, 에너지 효율성, 메모리 저장 용량 등의 한계를 뛰어넘기 위해 지속적이고 빠르게 발전하는 중이다[1].

CNN 구조는 1960 년대 후반부터 사용되어졌으나, 그 전까지는 저장 용량, 계산 자원의 부족으로 제한되어오다 컴퓨터의 발전과 함께 2010 년대에 들어서 높은 정확도를 가지는 아키텍처가 개발되기 시작했다.

본 논문에서는 알고리즘의 복잡성과 파라미터 수가 기하급수적으로 증가했지만, human error accuracy(약 5%)보다 우수한 error accuracy(3.57%)를 얻어낸 ResNet 모델[1]을 keras를 활용해서 구현하였다.

2. Related Works

CNN 아키텍처는 여러개의 convolution, activation, pooling 등의 레이어로 구성된 구조이다. 각 레이어는 이전의 레이어의 값에 따라 정해지며, 첫번째 레이어는 다음 공식으로 계산된다.

$$h^{(1)} = g^{(1)}(W^{(1)}x + b^{(1)}) \quad (1)$$

그 다음 레이어들은 다음 공식으로 계산된다.

$$h^{(i+1)} = g^{(i+1)}(W^{(i+1)}h^{(i)} + b^{(i+1)}) \quad (2)$$

이 때, x는 입력되는 훈련 데이터, w는 가중치를 의미한다[1].

2.1. AlexNet

AlexNet[2]은 ImageNet Large-Scale Visual Recognition Challenge(이하 ILSVRC) 2012에서 개발된 모델로 60M파라미터와 500,000개의 뉴런을 가진 CNN 모델이다. 파라미터 수로 인한 오버피팅을 줄이기 위해 dropout방법을 도입했으며 다음과 같은 구조를 가진다.

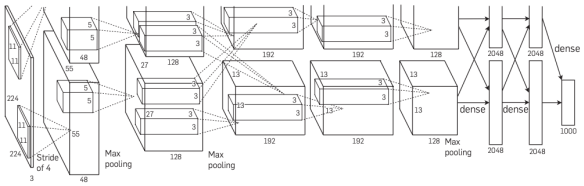


Figure 1. AlexNet[2]

5개의 convolutional layer, 3개의 max-pooling layer, 2개의 normalization layer, 3개의 fully connected layer로 구성되며 마지막으로 softmax가 활용되어 CONV 1 → MAXPOOL1 → NORM1 → CONV 2 → MAXPOOL2 → NORM2 → CONV 3 → CONV 4 → CONV 5 → MaxPOOL3 → FC6 → FC7 → FC8 로 구성되었다.
본 모델은 top-1, top-5 error rate가 각각 39.7%, 18.9%를 도출하며 해당 챌린지에서 우승하였다.

2.2. ZFNet

ZFNet[3]은 CNN의 수행 방법과 과정을 이해하기 위한 진단적 시각화 기술을 도입한 것으로 CNN 아키텍처 개선에 사용되었다. 이를 통해 AlexNet과 동일한 레이어 수를 가지고 error rate를 11.7%로 개선하여 ILSVRC 2013에서 우승하였다.

2.3. VGGNet

VGGNet[4]은 AlexNet을 일반화 시킨 모델로 레이어의 깊이를 16~19 개로 증가시켰으며, convolution layer에 7x7 filter stack 대신 3 개의 3x3 filter stack을 사용해 다음과 같은 구조를 가진다.

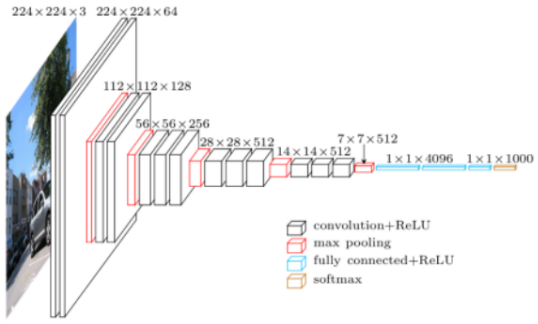


Figure 2. VGGNet[4]

[CONV 3 → CONV3 → POOL2]^2 → [CONV3 → CONV 3 → CONV3 → POOL2]^3 → FC → FC → FC로 구성되어, 이를 통해 top-5 error rate를 7.3%로 개선하여 ILSVRC 14 localization 부문에서 우승, classification 부문에서 준우승하였다.

2.4. GoogLeNet

GoogLeNet[5]은 병렬 필터 연산을 적용하여 컴퓨팅 자원을 효과적으로 활용하는 특징을 가진 모델로 concatenation을 통해 다양한 다중 스케일 처리를 수행한다. 새로운 topology를 도입하여 2010년 이후의 모든 아키텍처에 비해 네트워크의 깊이를 22개의 layer로 증가 시켜 ILSVRC 14에서 우승하였다.

2.5. NiN(Network In Network)

NiN[6]은 Mlpconv(multilayer perceptron(MLP) convolution) layer와 global averaging pooling layer로 구성되어 있다. Mlpconv는 receptive field내의 데이터를 추상화 하기 위해 더 복잡한 구조로 된 micro network를 포함되어 있다. 또한 네트워크 내부의 시각적 해석에 관한 연구가 수행되었다.

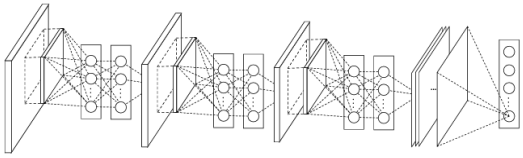
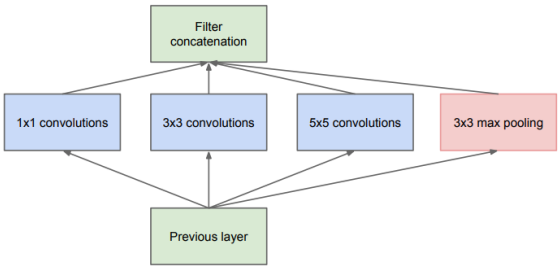
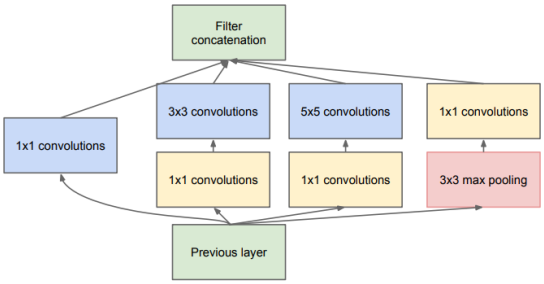


Figure 3. NiN[6]



(a) Inception module, naïve version



(b) Inception module with dimensionality reduction

Figure 4. GoogLeNet[5]

2.6. ResNet

ResNet[7]은 이미지 인식을 위한 잔차학습(Residual Learning)을 활용한 모델로 deep neural network에서 skip connection으로 vanishing gradient를 해결하기 위해 layer와 skip connection이 있는 residual block을 활용하였다

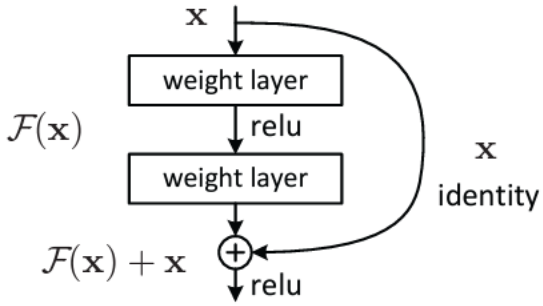


Figure 5. Residual Block[7]

이를 통해 152개의 레이어로까지 CNN ResNet을 학습할 수 있게 하였고, 8배로 깊게 확장되면서 복잡성을 감소 시켜 ILSVRC 2015에서 error rate 3.57%로 우승을 차지하였고, 인간의 인지 능력의 error rate를 뛰어넘는 수치를 달성하였다.

이후 ResNet을 개선한 ResNeXt, Hybrid CNN 아키텍처(MRF-CNN, RIFD-CNN, TI-POOLING, CNN

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv 1	112 x 112	7 x 7, 64, stride 2				
conv2_x	56 x 56	3 x 3 max pool, stride 2				
		$\begin{bmatrix}3\times3, & 64\\ 3\times3, & 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, & 64\\ 3\times3, & 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, & 64\\ 3\times3, & 64\\ 1\times1, & 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, & 64\\ 3\times3, & 64\\ 1\times1, & 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, & 64\\ 3\times3, & 64\\ 1\times1, & 256\end{bmatrix}\times3$
con3_x	28 x 28	$\begin{bmatrix}3\times3, & 128\\ 3\times3, & 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, & 128\\ 3\times3, & 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, & 128\\ 3\times3, & 128\\ 1\times1, & 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, & 128\\ 3\times3, & 128\\ 1\times1, & 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, & 128\\ 3\times3, & 128\\ 1\times1, & 512\end{bmatrix}\times4$
conv4_x	14 x 14	$\begin{bmatrix}3\times3, & 256\\ 3\times3, & 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, & 256\\ 3\times3, & 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, & 256\\ 3\times3, & 256\\ 1\times1, & 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, & 256\\ 3\times3, & 256\\ 1\times1, & 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, & 256\\ 3\times3, & 256\\ 1\times1, & 1024\end{bmatrix}\times6$
conv5_x	7 x 7	$\begin{bmatrix}3\times3, & 512\\ 3\times3, & 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, & 512\\ 3\times3, & 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, & 512\\ 3\times3, & 512\\ 1\times1, & 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, & 512\\ 3\times3, & 512\\ 1\times1, & 2048\end{bmatrix}\times$	$\begin{bmatrix}1\times1, & 512\\ 3\times3, & 512\\ 1\times1, & 2048\end{bmatrix}\times$
	1 x 1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Table 1. ResNet Layers

+ DPCL, LF-CNN 등), Squeeze, embedded을 활용해 개선된 CNN 아키텍처, mobile 환경에서의 CNN 아키텍처 등 다양한 방향으로의 발전이 이루어지고 있다[1].

3. Method

Keras 활용하여 ResNet blocks(34-layer, 50-layer, plain-34-layer, plain-50-layer)를 ResNet Model을 구현해 classification dataset으로 모델 성능을 비교하였다.

3.1. Keras

인간의 인지능력 error rate를 뛰어넘은 ResNet을 구현하기 위해 python으로 작성된 오픈 소스 신경망 라이브러리인 Keras[8]를 사용하였다. keras는 사용성과 직관성이 좋아 빠르고 간단하게 딥러닝 모델을 구축하고 훈련할 수 있다,본 논문에서는 Python 3.8, Tensroflow 2.9 를 사용하였다.

3.2. ResNet blocks

결과의 비교를 위해 잔차 연결이 있는 34-layer, 50-layer, 잔차 연결이 없는 34-layer, 50-layer를 구성하였다. Table 1 과 같이 34-layer는 3x3 커널이 사용되었고 64, 128, 256, 512 필터로 구성되어 2 개의 convolution layer를 각각 3, 4, 6, 3 번 반복하였다. 50-layer는 1x1,3x3 필터를 번갈아 사용하였으며 3 개의 convolution layer를 각각 3, 4, 6, 3 번 반복하였다. 각 convolutional layer에는 batch nomalization이 적용되었으며 activation function은 ReLu를 사용하였다.

잔차 연결(skip connection)은 Fig. 5 에서와 같이 2 개의 convolution layer를 건너뛰며 수행되었다.

3.3. ResNet Models

Table 1.에서와 같이 첫번째 convolution layer는 7x7 커널, 64 필터로, 두번째 convolution layer는 3x3 max pooling으로 시작되어 모델별로 설정한 블록으로 구성되고 최종적으로 average pooling을 거쳐 softmax를 활용한 fully connected convolution layer로 전체 모델이 구성되었다.
각 모델의 epoch는 30 로, batch size는 32 로 설정하였다.

4. Result

고양이와 강아지 2 개의 클래스로 분류되어 총 25000 의 이미지 데이터 셋[9]를 활용해 각 모델의 성능을 비교해 보았다. loss function은 sparse categorical crossentropy, Adam 옵티마이저를 사용하였고 모델 성능은 정확도를 기준으로 평가하였다.

Fig 6.에서와 같이 resnet-34-layer는 plain-34-layer, plain-50-layer와 유의미한 차이가 없었으나 resnet-50-layer는 처음부터 비교적 좋은 성능으로 시작하여 학습이 진행될수록 타 모델들과 유의미한 차이를 볼 수 있었다.

5. Conclusion

본 논문에서 사용한 모델의 결과에 대한 일반화를 위해서는 더 다양한 데이터 셋에서의 평가가 필요하다[10]. 또한, 학습시간 오래 걸리는 문제를 해결하기 위해서는 제안된 ResNeXt와 같은 모델을 사용할 수 있을 것이다.

최근 AI 분야의 발전 속도와 방향에 맞추어 다양한 분야와의 결합을 고려해 더 넓게 적용하도록 해야하며 특히, 설명 가능한

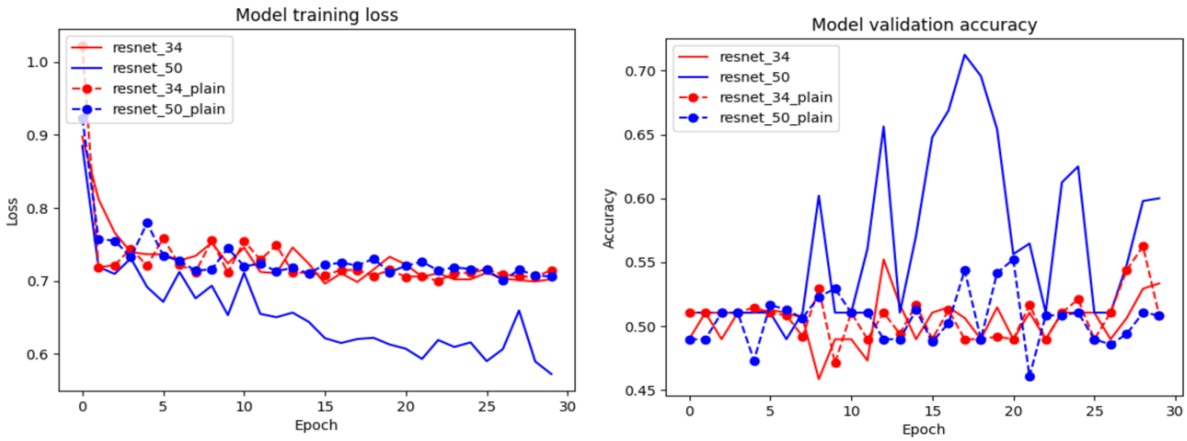


Figure 6. Result

인공지능(eXplainable Artificial Intelligence, XAI)으로의 모델 개발이 필요하다.

References

[1] Elhassouny, Azeddine, and Florentin Smarandache. "Trends in deep convolutional neural Networks architectures: A review." 2019 International conference of computer science and renewable energies (ICCSRE). IEEE, 2019.

[2] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems 25 (2012).

[3] Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13. Springer International Publishing, 2014.

[4] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014)

[5] Szegedy, Christian, et al. "Going deeper with convolutions." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[6] Lin, Min, Qiang Chen, and Shuicheng Yan. "Network in network." arXiv preprint arXiv:1312.4400 (2013).

[7] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[8] https://github.com/keras-team/keras

[9] https://paperswithcode.com/dataset/cats-vs-dogs

[10] Recht, Benjamin, et al. "Do cifar-10 classifiers generalize to cifar-10?." arXiv preprint arXiv:1806.00451 (2018).