Comparing Three Different Pooling Methods Used in Convolutional Neural Networks

Sixuan Wu
University of Toronto
sixuan.wu@mail.utoronto.ca

Qinchen Wang
University of Toronto
quinn.wang@mail.utoronto.ca

Guangwei Xia
University of Toronto
guangwei.xia@mail.utoronto.ca

Abstract

Convolutional neural networks (CNN)[1] has been very effective in image related deep learning problems. It has been shown that VGG networks yield high classification accuracies on benchmark datasets such as the ImageNet[2]. However, with an increasing number of layers in the network and the popular use of max pooling, CNN may suffer from information loss. In this paper, we compare the performance of three different pooling methods - Spectral pooling [3], Gaussian pooling [4] and max pooling. Their respective performance will be evaluated on two different datasets - image dataset comprised of 101 object categories, Caltech101 [5], and a collection of different categories of audio recordings, ESC50 [6]. The original paper of those pooling layers only be tested on image image classification but not on audio classification. We show that Spectral pooling and Gaussian pooling achieves comparative results with max pooling on some datasets, but can yield appreciably better results on other datasets.

1 Introduction

Numerous tricks has been introduced to further improve VGG performance, generalizability and optimize for computation cost. One of such tricks is the introduction of pooling. Max pooling is used to reduce the number of parameters while dropping excess information, but it does it in a crude way - choosing the highest activation. The highest activation may not always correspond to the activation with the most information, which leads to information loss when we drop all other than the max activation [7]. On the other hand, some recently proposed alternative pooling methods such as Spectral pooling [3] and Gaussian pooling [4] aims to reduce parameter size while preserving much more information. In the next section, we will introduce how these two pooling layers work, and how they may preserve more information. In this paper, we want to test these two alternative pooling layers and compare them with the popularly used max pooling layer. For a fair comparison, we used VGG16, without pretraining, for all model training, changing only the pooling layers to all be one of max pooling, Spectral pooling or Gaussian pooling. Our complete code and be found under this Github repository: `https://github.com/wsxwsx543/CSC413-Project`.

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

2 Related Works

2.1 Fourier Transformation

Discrete Fourier transformation(DFT) [8] is a useful transformation in mathematics and engineering fields which can associate the frequency domain and time/spatial domain. Since images can be viewed as discrete 2-dimensional signals, then the discrete Fourier transformation can be applied to the image. The Fourier transformation can be represented as following:

$$A_{kl} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} a_{mn} \exp\{-2\pi i(\frac{mk}{M} + \frac{nl}{N})\}$$

A_{kl} indicates the k^{th} row and l^{th} column of the output frequency image and the a_{mn} represents the m^{th} row and n^{th} column of the input image whose shape is $M \times N$. And the inverse Fourier Transformation can be represented as:

$$a_{kl} = \frac{1}{M} \sum_{m=0}^{M-1} \frac{1}{N} \sum_{n=0}^{N-1} \omega_M^{km} \omega_N^{ln} A_{mn}$$

where a is the output space domain image and A is the input frequency domain image. The image is with shape $M \times N$ and $\omega_M = e^{\frac{2\pi i}{M}}$ and $\omega_N = e^{\frac{2\pi i}{N}}$. Notice that the Euler's formula[9] is:

$$e^{i\theta} = \cos\theta + i\sin\theta$$

On the implementation side, we use the internal library of PyTorch for FFT.

2.2 VGG Convolution Neural Network

VGG convolutional network is a network structure using deep convolutional layers, with all convolutional filters fixed in a small size to be 3×3. By the results from [10], we know classification accuracy is enhanced with higher representation depth, while on the large-scale image recognition, it performs well.
On the implementation side, we use the internal library of PyTorch for VGG16. An illustration of the network struction can be found in figure 5 in the appendix section.

2.3 Spectral Pooling

Spectral pooling is useful to extract features from feature maps and can prevent the loss of information. We first transform the image from spatial domain to frequency domain by DFT. Then we crop the representation image in the frequency domain to keep the low frequency part where store the main information of the images. Finally we transform the cropped image from frequency domain to spatial domain back and feed it to the next layer. The pseudocode of implementation can be found in [3]

2.4 Gaussian-based pooling

Based on the Gaussian probabilistic model [4], Gaussian pooling is efficient at downsizing the feature map, which greatly improves the performance of CNNs. In this pooling method, the input neuron activations are assumed to follow Gaussian distribution, and aggregated into mean and standard deviation. Then inverse softplus-Gaussian distribution is used to formulate the trainable local pooling. The inverse softplus-Gaussian distribution is proposed as:

$$\eta \sim \mathcal{N}_{isp}(\mu_0, \sigma_0) \Leftrightarrow \eta = softplus(\hat{\epsilon}) = log\{1 + exp(\hat{\epsilon})\}, where\ \hat{\epsilon} \sim \mathcal{N}(\mu_0, \sigma_0)$$

And the probability density function of $\mathcal{N}_{isp}$ is :

$$\mathcal{N}_{isp}(x; \mu_0, \sigma_0) = \frac{1}{\sqrt{2\pi}\sigma_0} \frac{exp(x)}{exp(x) - 1} exp\{-\frac{1}{2\sigma_0^2}(log[exp(x) - 1] - \mu_0)^2\}$$

Thus, at the inference layer, the pooling form is defined as:

$$Y = \mu_x + softplus(\mu_0)\sigma_x$$

3 Method

3.1 Caltech101

This dataset is chosen because it is an image dataset with a good number of categories, but not too much that makes training from scratch too hard. First we import image data from the Caltech101 dataset. Images in the Caltech101 dataset are not all square images of the same size. Their respective widths and lengths are roughly 200 to 300 pixels. An illustration of images in the dataset is shown in figure 6 under the appendix section. Due to limited computational power and time, we performs resizing and center cropping each image to same size of 50x50 pixels.

We then import the VGG16 model, without pretraining. The reason we do not use training model is the pretrained model is trained on max-pooling layers, while we want to observe the difference between max-pooling, spectral pooling and Gaussian pooling. For experiments with alternative pooling methods, we replace all the max pooling layers with a Spectral pooling layer or a Gaussian pooling layer that crops the input width to a half it's original size. We also adjust the final fully connected layer to match the number of classes in the dataset - which is 102. In addition to the two alternative pooling layers, we also train a VGG16 model from scratch on this dataset for fair comparison. The loss function we use is cross-entropy and the optimizer we used is Adam [11].

We then train this model using the splitted training set for a total of 50 epochs. At the end of each epoch, we record the average training loss for this epoch, as well as the validation accuracy by evaluating the result of current model's prediction on the validation set. In the end, we plot curves of the average training loss and validation accuracies for the different pooling methods on the same graph, as shown in section 4.

3.2 ESC50

The ESC-50 dataset is a labeled collection of 2000 environmental audio recordings suitable for benchmarking methods of environmental sound classification. And for each wave file, it contains 5 seconds audio and each class contains 40 examples.

This classification task can be done by CNN because we can transform each audio file to mel-spectrogram, then we use CNN with different pooling layers to do the classification task on the transformed images. We implement this transformation method with librosa library. The reason why we use this dataset is the papers discussed spectral pooling and Gaussian-based pooling [3, 4] only show the results on image dataset but not on audio dataset.

We download the VGG16 model without pretraining from the torchvision library and train the model with Adam Optimizer [11] and the cross-entropy as the loss function until we observe the convergence on validation accuracy and training loss. We replace the max pooling layers with spectral pooling and Gaussian pooling layers respectively before each experiments on the pooling layers with ESC50 dataset. The result can be found in section 4.

4 Result and Discussions

For training and loss accuracy, evolution of average training loss is illustrated in figures 1 and 2. Evolution of validation accuracy is shown in figures 3 and 4. Some statistical results are shown in tables 1 and 2.

Pooling layer	Spectral pooling	Gaussian pooling	max pooling
Minimum average training loss	0.0338526	0.0755908	0.0463839
Maximum validation accuracy	0.595	0.541429	0.514286

Table 1: Minimum average training loss and maximum validation accuracy comparison amongst Spectral pooling, Gaussian pooling and max pooling on Caltech101

Based on our experiments, we can observe an improvement with spectral pooling and Gaussian pooling on both image object classification and audio classification tasks. On object classification task

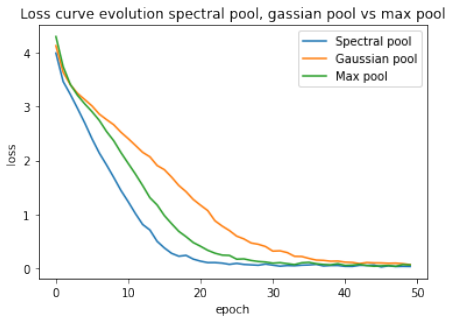

Figure 1: Evolution of average training loss using different pooling layers on Caltech101

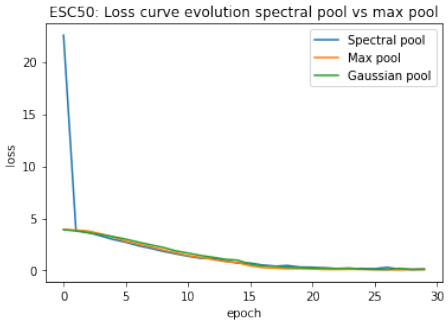

Figure 2: Evolution of average training loss using different pooling layers on ESC50

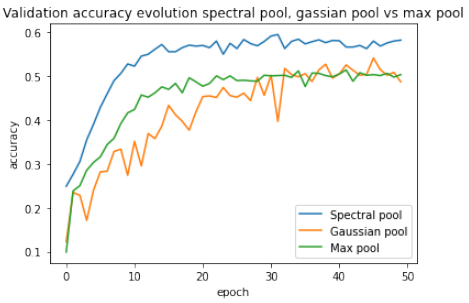

Figure 3: Validation accuracy during training steps on Caltech101

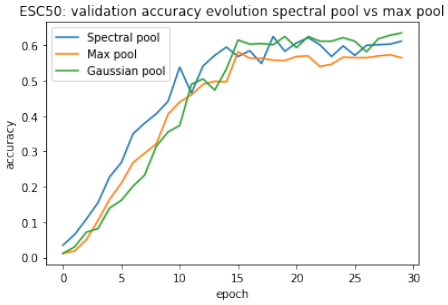

Figure 4: Validation accuracy during training steps on ESC50

Pooling layer	Spectral pooling	Gaussian pooling	max pooling
Minimum average training loss	0.113626	0.1000576	0.06593107
Maximum validation accuracy	0.625	0.635	0.582

Table 2: Minimum average training loss and maximum validation accuracy comparison amongst Spectral pooling, Gaussian pooling and max pooling on ESC50

with Caltech101 dataset, validation accuracy of spectral pooling is much higher than the Gaussian pooling and max pooling. This is because the spectral pooling can decrease the information loss of the image while the assumption of Gaussian pooling that the pooling neurons weights and image pixels follow the normal distribution is inaccurate. In addition, the spectral pooling seems to have the fastest convergence speed while the Gaussian converges slowest.

On audio classification task(ESC50), Gaussian pooling achieves the highest validation accuracy while the spectral pooling takes the second place because the mel-spectrogram of different audios can be different at detailed area, which results in the functionality of information loss prevention is not as good as in object classification task.

5 Summary

We can conclude with different types of dataset we need to select different pooling methods to achieve a better performance. Even if trainable pooling layers spectral and Gaussian pooling achieves a better result, the limitation of spectral pooling is it is hard to extract detailed features which are always embedded by high frequency part of the images while the limitation of Gaussian pooling is the pooling neurons weights and image pixels need to follow the Gaussian distribution.

References

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. 2012.

[2] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *arXiv:1409.0575*, 01 2015.

[3] Oren Rippel, Jasper Snoek, and Ryan P. Adams. Spectral representations for convolutional neural networks. *arXiv:1506.03767v1*, 06 2015.

[4] Takumi Kobayashi. Gaussian-based pooling for convolutional neural networks. 32, 2019.

[5] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. *IEEE. CVPR 2004, Workshop on Generative-Model Based Vision*, 2004.

[6] Karol J. Piczak. ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd Annual ACM Conference on Multimedia*, pages 1015–1018. ACM Press.

[7] Daniel C. Elton. Geoffrey hinton on what's wrong with cnns. 09 2017.

[8] Fourier transformation. `https://en.wikipedia.org/wiki/Fourier_transform`.

[9] Euler formula. `https://en.wikipedia.org/wiki/Euler%27s_formula`.

[10] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations (ICLR)*, 2015.

[11] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980v9*, 09 2017.

6 Appendix

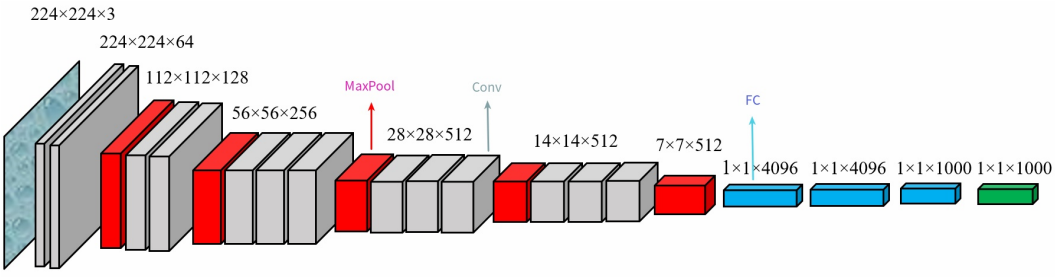

Figure 5: VGG16's structure sample graph

Figure 6: Examples of images in the Caltech101 dataset [5]

Contributions: Sixuan Wu wrote the backbone of the code, and is responsible for experimenting on the ESC50 dataset. Qinchen Wang is responsible for improvements on the code for Spectral pooling, and experimenting on the Caltech101 dataset. Guangwei Xia is responsible for formalizing the mathematical background in the related works section, and discussing the results and summarizing our findings.