Deep Learning

10.1 Inside DNNs

Dr. Konda Reddy Mopuri
kmopuri@iittp.ac.in
Dept. of CSE, IIT Tirupati

1. Understanding how the DNNs learn or what happens inside the deep architecture is not simple

1. Understanding how the DNNs learn or what happens inside the deep architecture is not simple
2. Not many tools can help us analyse these complex architectures

1. In case of CNNs, we can look at
    - the parameters (e.g. filters as images)

1. In case of CNNs, we can look at
   - the parameters (e.g. filters as images)
   - activation maps for the given input

1. In case of CNNs, we can look at
    - the parameters (e.g. filters as images)
    - activation maps for the given input
    - distribution of activations for a population of samples

1. In case of CNNs, we can look at
   - the parameters (e.g. filters as images)
   - activation maps for the given input
   - distribution of activations for a population of samples
   - derivative of selected activations wrt a given input

1. In case of CNNs, we can look at
   - the parameters (e.g. filters as images)
   - activation maps for the given input
   - distribution of activations for a population of samples
   - derivative of selected activations wrt a given input
   - synthesize samples that can cause maximal activations at selected neurons

Visualize in case of simpler models

1. Visualize the weights via plotting

Visualize in case of simpler models

1. Visualize the weights via plotting
2. e.g. input for an MLP is 2D vector

Visualize in case of simpler models

1. Visualize the weights via plotting
2. e.g. input for an MLP is 2D vector
3. weights in the first hidden layer are lines, we can plot them

Visualize in case of simpler models

1. Visualize the weights via plotting
2. e.g. input for an MLP is 2D vector
3. weights in the first hidden layer are lines, we can plot them
4. observe these lines during the course of training

CNN filters

1. Similar visualization gets challenging on practical DNNs because of the high dimensional inputs

CNN filters

1. Similar visualization gets challenging on practical DNNs because of the high dimensional inputs
2. View the filters learned by CNNs as images

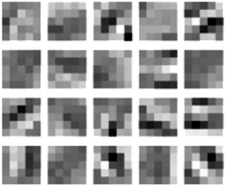
CNN filters

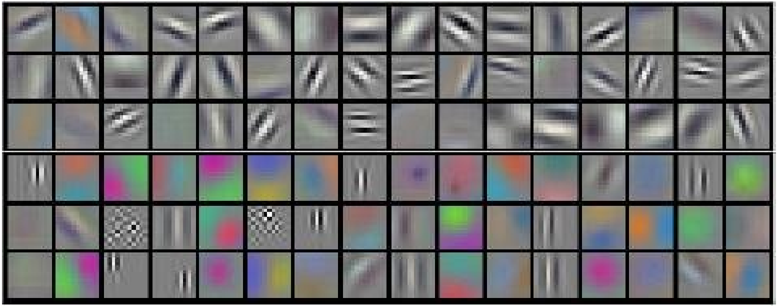
1. Similar visualization gets challenging on practical DNNs because of the high dimensional inputs
2. View the filters learned by CNNs as images
3. Appropriate at the first layer (since the filters will have less number of channels like actual images)

CNN filters

1. Similar visualization gets challenging on practical DNNs because of the high dimensional inputs
2. View the filters learned by CNNs as images
3. Appropriate at the first layer (since the filters will have less number of channels like actual images)
4. Later layers will have multiple channels and makes it difficult to visualize

CNN Filters (LeNet and AlexNet)





Visualize feature maps

1. Since CNNs maintain the 2D structure, we can visualize the feature/activation maps at any layer

Visualize feature maps

1. Since CNNs maintain the 2D structure, we can visualize the feature/activation maps at any layer
2. Since there may be a large number of channels, we may have to pick a subset of them

Visualize feature maps

1. Since CNNs maintain the 2D structure, we can visualize the feature/activation maps at any layer
2. Since there may be a large number of channels, we may have to pick a subset of them
3. Visualizing individual maps may not be very helpful

Visualize feature maps

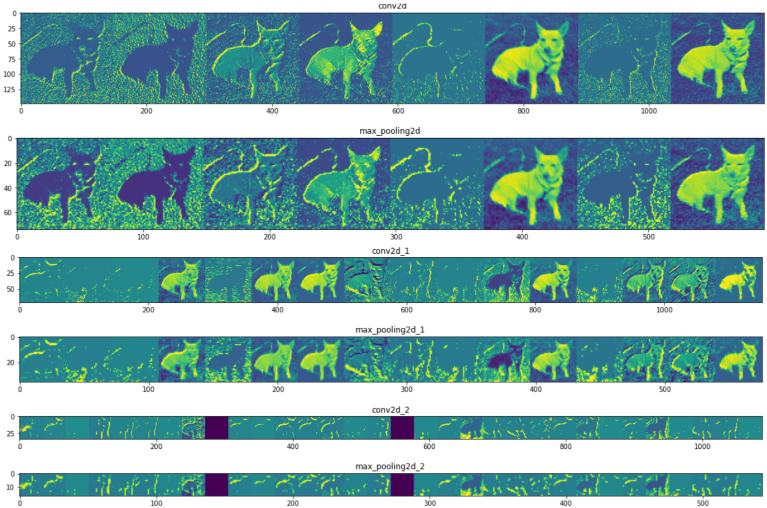


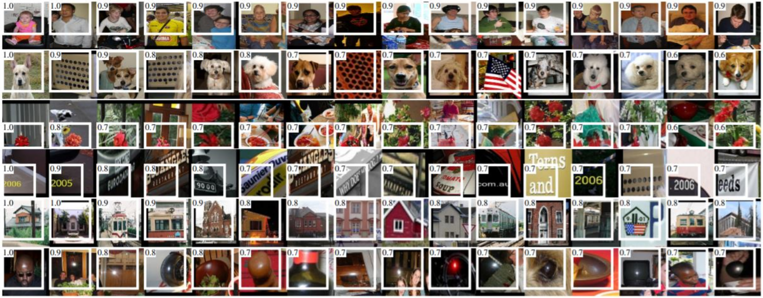
Figure credits: Shivang Shrivastav

Visualize feature maps

1. Although it is quite hard to precisely identify the role each filter, one may identify edge and some blob detectors.
2. As we go deeper in the network, the more difficult it gets to understand the role of filters

Visualize feature maps

1. Allows in particular to find units with a clear semantic/concept

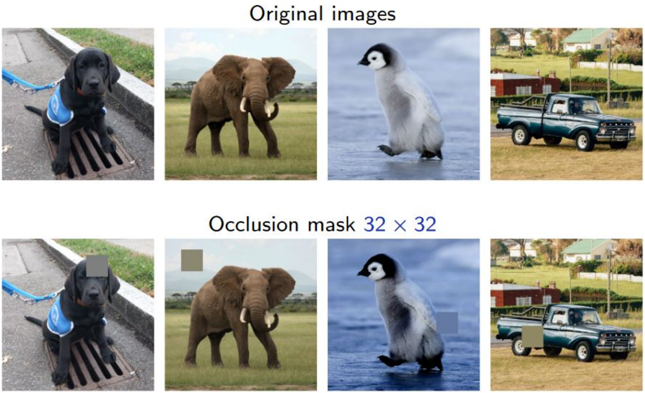


Ross Girshick et al. 2014

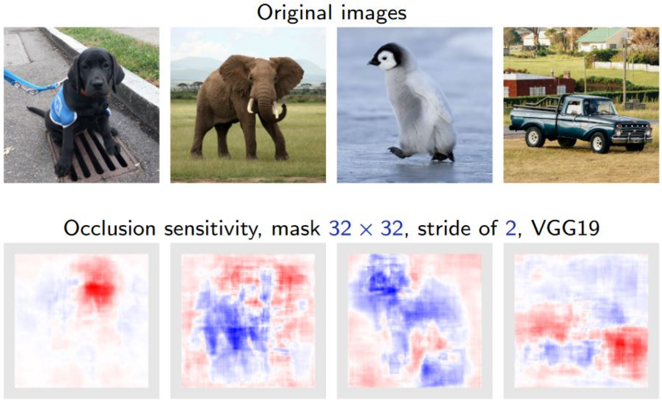
Analyze the model behavior in the neighborhood of an input

1. We may estimate the importance of a portion of input via occlusion

Occlusion sensitivity



Occlusion sensitivity



Saliency maps

1. Highlight the parts of input that are influential for the output

Saliency maps

1. Highlight the parts of input that are influential for the output
2. Compute the derivative of the output wrt the input over a trained DNN model (Simonyan *et al.* 2013)

$$\nabla_{|x} f_c(x; w)$$

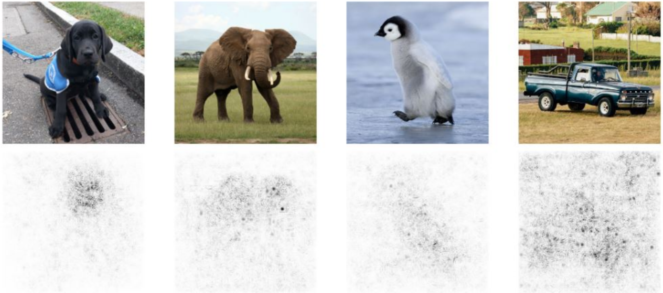
Saliency maps

1. Highlight the parts of input that are influential for the output
2. Compute the derivative of the output wrt the input over a trained DNN model (Simonyan *et al.* 2013)

   $$\nabla_{|x} f_c(x; w)$$

3.
```
input.requires_grad_()
output = model(input)
grad_input, = torch.autograd.grad(output[0, c], input)
```

Gradient as saliency



Smooth Grad

1. (Pixel level) Gradient is noisy

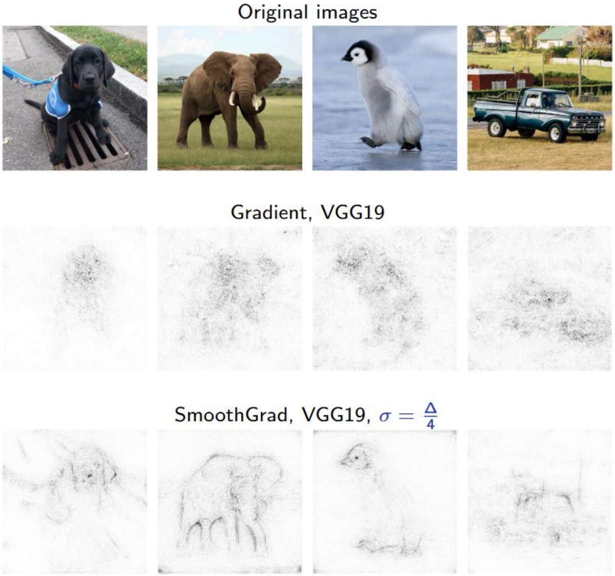
Smooth Grad

1. (Pixel level) Gradient is noisy
2. We can smooth the gradient by taking the mean gradient over (slightly) perturbed inputs (Smilkov *et al.* 2017)

$$\tilde{\nabla}_{|x} f_y(x; w) = \frac{1}{N} \sum_{n=1}^{N} \nabla_{|x} f_y(x + \epsilon_n; w)$$

where $\epsilon_n \sim \mathcal{N}(0, \sigma^2 I)$

Smooth Grad


Original images
Gradient, VGG19
SmoothGrad, VGG19, $\sigma = \frac{\Delta}{4}$

Improvements to gradient based visualization

1. Deconvolution by Zeiler and Fergus
2. Guided Backpropagation by Springenberg et al. (2014)
3. Class Activation Maps (Zhou et al. 2016)
4. Gradient-weighted Class Activation Mapping (Grad-CAM) by Selvaraju et al. (2016)
5. ...

References

1. Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik; Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 580-587
2. K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. CoRR, abs/1312.6034, 2013
3. D Smilkov, N. Thorat, B. Kim, F. Viegas, and M. Wattenberg. Smoothgrad: removing noise by adding noise. CoRR, abs/1706.03825, 2017.
4. M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In European Conference on Computer Vision (ECCV), 2014
5. J. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. CoRR, abs/1412.6806, 2014
6. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. CoRR, abs/1610.02391, 2016
7. B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning Deep Features for Discriminative Localization. CVPR'16 (arXiv:1512.04150, 2015).