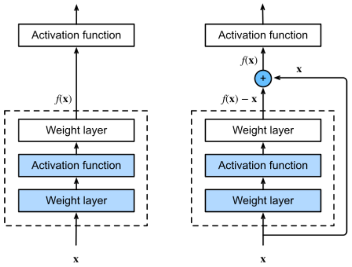
Deep Learning

Modern Architectures - II



Learning goals

- GoogleNet
- ResNet
- DenseNet
- U-Net

GoogLeNet

INCEPTION MODULES

- The Inception block is equivalent to a subnetwork with four paths.
- It extracts information in parallel through convolutional layers of different window shapes and max-pooling layers.
- 1×1 convolutions reduce channel dimensionality on a per-pixel level. Max-pooling reduces the resolution.

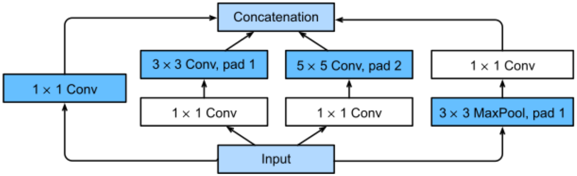
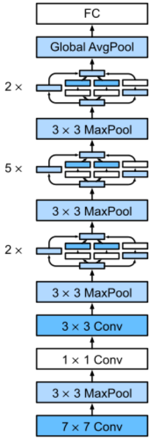


Figure: Inception Block.

GOOGLENET ARCHITECTURE

- GoogLeNet connects multiple well-designed Inception blocks with other layers in series.
- The ratio of the number of channels assigned in the Inception block is obtained through a large number of experiments on the ImageNet dataset.
- GoogLeNet, as well as its succeeding versions, was one of the most efficient models on ImageNet, providing similar test accuracy with lower computational complexity.

GOOGLENET ARCHITECTURE



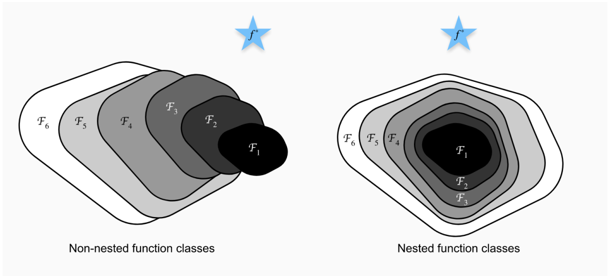
credit : D2DL

Figure: The GoogLeNet architecture.

Residual Networks (ResNet)

RESIDUAL BLOCK (SKIP CONNECTIONS)

Problem setting: theoretically, we could build infinitely deep architectures as the net should learn to pick the beneficial layers and skip those that do not improve the performance automatically.



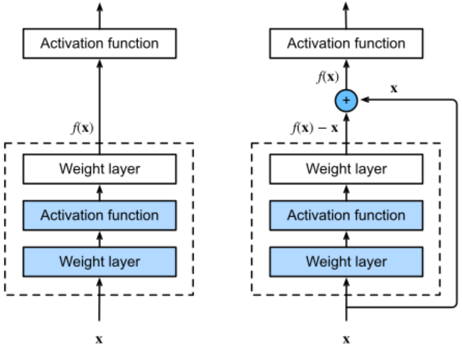
credit : D2DL

Figure: For non-nested function classes, a larger function class does not guarantee to get closer to the “truth” function ($\mathcal{F}*$). This does not happen in nested function classes.

RESIDUAL BLOCK (SKIP CONNECTIONS)

- But: this skipping would imply learning an identity mapping $\mathbf{x} = \mathcal{F}(\mathbf{x})$. It is very hard for a neural net to learn such a 1:1 mapping through the many non-linear activations in the architecture.
- Solution: offer the model explicitly the opportunity to skip certain layers if they are not useful.
- Introduced in *He et. al , 2015* and motivated by the observation that stacking evermore layers increases the test- as well as the train-error ($\neq$ overfitting).

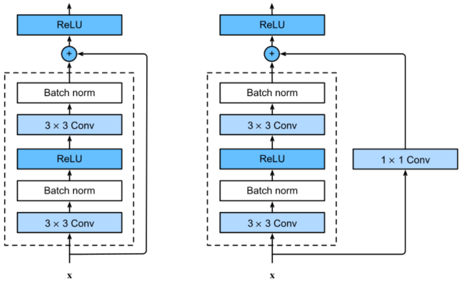
RESIDUAL BLOCK (SKIP CONNECTIONS)



credit : D2DL

Figure: A regular block (left) and a residual block (right).

RESIDUAL BLOCK (SKIP CONNECTIONS)



credit : D2DL

Figure: ResNet block with and without 1×1 convolution.The information flows through two layers and the identity function. Both streams of information are then element-wise summed and jointly activated.

RESIDUAL BLOCK (SKIP CONNECTIONS)

- Let $\mathcal{H}(\mathbf{x})$ be the optimal underlying mapping that should be learned by (parts of) the net.
- **x** is the input in layer *l* (can be raw data input or the output of a previous layer).
- $\mathcal{H}(\mathbf{x})$ is the output from layer *l*.
- Instead of fitting $\mathcal{H}(\mathbf{x})$, the net is ought to learn the residual mapping $\mathcal{F}(\mathbf{x}) := \mathcal{H}(\mathbf{x}) - \mathbf{x}$ whilst **x** is added via the identity mapping.
- Thus, $\mathcal{H}(\mathbf{x}) = \mathcal{F}(\mathbf{x}) + \mathbf{x}$, as formulated on the previous slide.
- The model should only learn the **residual mapping** $\mathcal{F}(\mathbf{x})$
- Thus, the procedure is also referred to as **Residual Learning**.

RESIDUAL BLOCK (SKIP CONNECTIONS)

- The element-wise addition of the learned residuals $\mathcal{F}(\mathbf{x})$ and the identity-mapped data $\mathbf{x}$ requires both to have the same dimensions.
- To allow for downsampling within $\mathcal{F}(\mathbf{x})$ (via pooling or valid-padded convolutions), the authors introduce a linear projection layer W_s .
- W_s ensures that $\mathbf{x}$ is brought to the same dimensionality as $\mathcal{F}(\mathbf{x})$ such that:

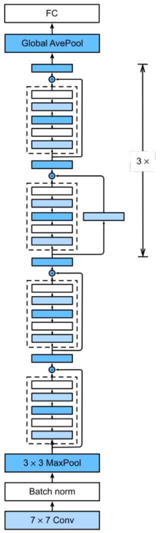
$$y = \mathcal{F}(\mathbf{x}) + W_s\mathbf{x},$$

- y is the output of the skip module and W_s represents the weight matrix of the linear projection (# rows of W_s = dimensionality of $\mathcal{F}(\mathbf{x})$).
- This idea applies to fully connected layers as well as to convolutional layers.

RESNET ARCHITECTURE

- The residual mapping can learn the identity function more easily, such as pushing parameters in the weight layer to zero.
- We can train an effective deep neural network by having residual blocks.
- Inputs can forward propagate faster through the residual connections across layers.
- ResNet had a major influence on the design of subsequent deep neural networks, both for convolutional and sequential nature.

RESNET ARCHITECTURE



credit : D2DL

Figure: The ResNet-18 architecture.

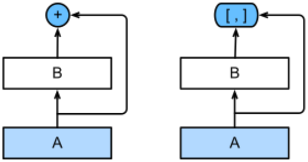
Densely Connected Networks (DenseNet)

FROM RESNET TO DENSENET

- ResNet significantly changed the view of how to parametrize the functions in deep networks.
- DenseNet (dense convolutional network) is to some extent the logical extension of this [Huang et al., 2017].
- Dense blocks where each layer is connected to every other layer in feedforward fashion.
- Alleviates vanishing gradient, strengthens feature propagation, encourages feature reuse.
- To understand how to arrive at it, let us take a small detour to mathematics:
  - Recall the Taylor expansion for functions. For the point $x = 0$ it can be written as: $f(x) = f(0) + f'(0)x + \frac{f''(0)}{2!}x^2 + \frac{f'''(0)}{3!}x^3 + \dots$.

FROM RESNET TO DENSENET

- The key point is that it decomposes a function into increasingly higher order terms. In a similar vein, ResNet decomposes functions into : $f(\mathbf{x}) = \mathbf{x} + g(\mathbf{x})$.
- That is, ResNet decomposes f into a simple linear term and a more complex nonlinear one. What if we want to capture (not necessarily add) information beyond two terms? One solution was DenseNet [Huang et al., 2017].



credit : D2DL

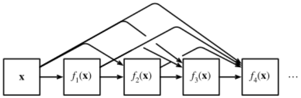
Figure: DensNet Block.

FROM RESNET TO DENSENET

As shown in previous Figure, the key difference between ResNet and DenseNet is that in the latter case outputs are concatenated (denoted by $[,]$) rather than added. As a result, we perform a mapping from x to its values after applying an increasingly complex sequence of functions:

$$\mathbf{x} \to [\mathbf{x}, f_1(\mathbf{x}), f_2([\mathbf{x}, f_1(\mathbf{x})]), f_3([\mathbf{x}, f_1(\mathbf{x}), f_2([\mathbf{x}, f_1(\mathbf{x})])]), \ldots] .$$

In the end, all these functions are combined in MLP to reduce the number of features again. In terms of implementation this is quite simple: rather than adding terms, we concatenate them.
The name DenseNet arises from the fact that the dependency graph between variables becomes quite dense. The last layer of such a chain is densely connected to all previous layers.



credit : D2DL

Figure: The DensNet architecture.

U-Net

U-NET

- U-Net is a fully convolutional net that makes use of upsampling (via transposed convolutions, for example) as well as skip connections.
- Input images are getting convolved and down-sampled in the first half of the architecture.
- Then, they are getting upsampled and convolved again in the second half to get back to the input dimension.
- Skip connections throughout the net combine feature maps from earlier layers with those from later layers by concatenating both sets of maps along the depth/channel axis.
- Only convolutional and no dense layers are used.

U-NET

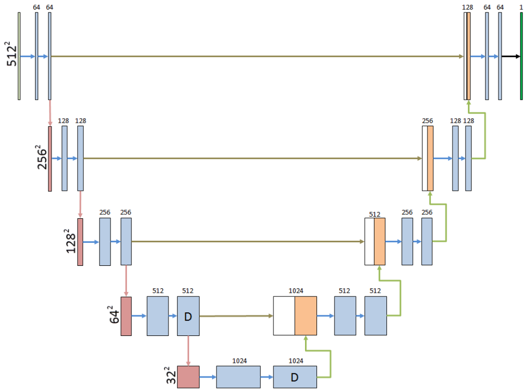


Figure: Illustration of the architecture. Blue arrows are convolutions, red arrows max-pooling operations, green arrows upsampling steps and the brown arrows merge layers with skip connections. The height and width of the feature blocks are shown on the vertical and the depth on the horizontal. D are dropout layers.

U-NET

- Example problem setting: train a neural net to pixelwise segment roads in satellite imagery.
- Answer the question: **Where is the road map?**

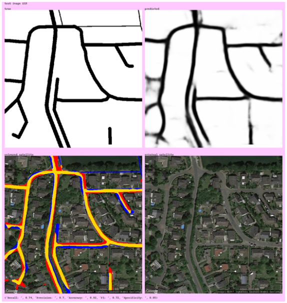
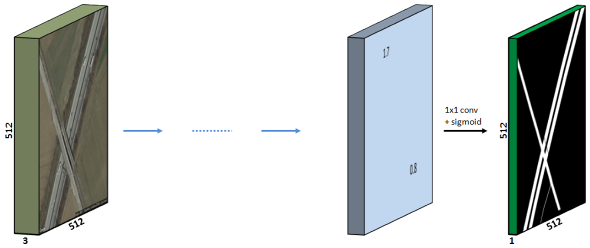

test image 119
true
predicted
coloured satellite
original satellite
('Recall: ', 0.74, 'Precision: ', 0.7, 'Accuracy: ', 0.92, 'F1: ', 0.72, 'Specificity: ', 0.95)

Figure: Model prediction on a test satellite image. Yellow are correctly identified pixels, blue false negatives and red false positives.

U-NET

- The net takes an RGB image [512, 512, 3] and outputs a binary (road / no road) probability mask [512, 512, 1] for each pixel.
- The model is trained via a binary cross entropy loss which was combined over each pixel.



$$Output_{i,j} = \frac{1}{1+exp(-1.7 \cdot \mathbf{w})} \in (0,1)$$

Figure: Scheme for the input/ output of the net architecture.

REFERENCES

B. Zhou, Khosla, A., Labedriza, A., Oliva, A. and A. Torralba (2016)
Deconvolution and Checkerboard Artifacts
http://cnnlocalization.csail.mit.edu/Zhou_Learning_Deep_Features_CVPR_2016_paper.pdf

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke and Andrew Rabinovich (2014)
Going deeper with convolutions
https://arxiv.org/abs/1409.4842

Kaiming He, Zhang, Xiangyu, Ren, Shaoqing, and Jian Sun (2015)
Deep Residual Learning for Image Recognition
https://arxiv.org/abs/1512.03385

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva and Antonio Torralba (2016)
Learning Deep Features for Discriminative Localization
http://cnnlocalization.csail.mit.edu/Zhou_Learning_Deep_Features_CVPR_2016_paper.pdf

REFERENCES

Olaf Ronneberger, Philipp Fischer, Thomas Brox (2015)
U-Net: Convolutional Networks for Biomedical Image Segmentation
http://arxiv.org/abs/1505.04597