Common Architecture Elements
Classification, Segmentation, Detection

ImageNet classification performance
(for up-to-date top-performers see leaderboards of datasets like ImageNet or COCO)

Some notable architecture elements shared by many successful architectures:

- Residual Blocks
- Dense Blocks

**Grouped Convolutions**

**Dilated Convolutions**

**Skip Connections** (UNet)

**Attention** (Spatial and over Channels)
Dilated (Atrous) Convolutions

Problem: increasing the receptive field costs a lots of parameters.
Idea: spread out the samples used in each convolution.

Yu and Koltun, Multi-scale Context Aggregation by Dilated Convolutions, ICLR 2016
Dilated (Atrous) Convolutions

Problem: increasing the receptive field costs a lots of parameters.
Idea: spread out the samples used for a convolution.

Dumoulin and Visin, A guide to convolution arithmetic for deep learning, arXiv 2016
**Grouped Convolutions (Inception Modules)**

Problem: conv. parameters grow quadratically in the number of channels

Idea: split channels into groups, remove connections between different groups

Image from: Xie et al., *Aggregated Residual Transformations for Deep Neural Networks*, CVPR 2017
Example: Sketch Simplification

Learning to Simplify: Fully Convolutional Networks for Rough Sketch Cleanup, Simo-Serra et al.
Example: Sketch Simplification

- Loss for thin edges saturates easily
- Authors take extra steps to align input and ground truth edges

Pencil: input
Red: ground truth

Learning to Simplify: Fully Convolutional Networks for Rough Sketch Cleanup, Simo-Serra et al.
Image Decomposition

• A selection of methods:
  • *Direct Instrinsics*, Narihira et al., 2015
  • *Learning Data-driven Reflectance Priors for Intrinsic Image Decomposition*, Zhou et al., 2015
  • *Decomposing Single Images for Layered Photo Retouching*, Innamorati et al. 2017
Image Decomposition: Decomposing
Single Images for Layered Photo Retouching
Example Application: Denoising
Deep Features
Autoencoders

- Features learned by deep networks are useful for a large range of tasks.
- An autoencoder is a simple way to obtain these features.
- Does not require additional supervision.

Manash Kumar Mandal, Implementing PCA, Feedforward and Convolutional Autoencoders and using it for Image Reconstruction, Retrieval & Compression, https://blog.manash.me/
Shared Feature Space: Interactive Garments

Wang et al., Learning a Shared Shape Space for Multimodal Garment Design, Siggraph Asia 2018
Transfer Learning

Features extracted by well-trained CNNs often generalize beyond the task they were trained on.

input image → encoder → decoder → useful features (latent vectors)

original task (normals) vs. new task (edges)

Images from: Zamir et al., Taskonomy: Disentangling Task Transfer Learning, CVPR 2018
Taxonomy of Tasks: Taskonomy

Images from: Zamir et al., Taskonomy: Disentangling Task Transfer Learning, CVPR 2018

http://taskonomy.stanford.edu/api/
Taxonomy of Tasks: Taskonomy

Images from: Zamir et al., Taskonomy: Disentangling Task Transfer Learning, CVPR 2018
Few-shot, One-shot Learning

• With a good feature space, tasks become easier
• In classification, for example, nearest neighbors might already be good enough
• Often trained with a Siamese network feature space

Feature training:
- lots of examples from class subset A

One-shot:
- train regressor with one example of each class in class subset B

regressor (e.g. NN)

feature computation

Style Transfer

• Combine content from image A with style from image B

What is Style and Content?

Remember that features in a CNN often generalize well.

Define style and content using the layers of a CNN (VGG19 for example):

shallow layers describe style

deep layers describe content
Optimize for Style A and Content B

same pre-trained networks, fix weights

A

B

optimize to have same style/content features
Style Transfer: Follow-Ups

more control over the result

Images from: Gatys, et al., Controlling Perceptual Factors in Neural Style Transfer, CVPR 2017
Johnson et al., Perceptual Losses for Real-Time Style Transfer and Super-Resolution, ECCV 2016
Style Transfer for Videos

Artistic style transfer for videos

Manuel Ruder
Alexey Dosovitskiy
Thomas Brox

University of Freiburg
Chair of Pattern Recognition and Image Processing

Ruder et al., *Artistic Style Transfer for Videos*, German Conference on Pattern Recognition 2016
Adversarial Image Generation
Generative Adversarial Networks

Player 1: generator
Scores if discriminator can’t distinguish output from real image

Player 2: discriminator
Scores if it can distinguish between real and fake from dataset
GANs to CGANs (Conditional GANs)

increasingly determined by the condition

Karras et al., Progressive Growing of GANs for Improved Quality, Stability, and Variation, ICLR 2018
Kelly and Guerrero et al., FrankenGAN: Guided Detail Synthesis for Building Mass Models using Style-Synchronized GANs, Siggraph Asia 2018
Isola et al., Image-to-Image Translation with Conditional Adversarial Nets, CVPR 2017
Image Credit: Zhu et al., Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, ICCV 2017
Image-to-image Translation

- \( \approx \) learn a mapping between images from example pairs
- Approximate sampling from a conditional distribution \( p_{\text{data}}(x \mid c) \)

Image Credit: Image-to-Image Translation with Conditional Adversarial Nets, Isola et al.
Adversarial Loss vs. Manual Loss

Problem: A good loss function is often hard to find

Idea: Train a network to discriminate between network output and ground truth

Adversarial Loss vs. Manual Loss

CycleGANs

• Less supervision than CGANs: mapping between unpaired datasets
• Two GANs + cycle consistency
CycleGAN: Two GANs ...

- Not conditional, so this alone does not constrain generator input and output to match

\[ D_{\psi_1}(\hat{x}) \]
\[ C_{\psi_2}(\hat{y}) \]

Image Credit: Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, Zhu et al.
CycleGAN: ... and Cycle Consistency

L1 Loss function: 
\[ \| y - \tilde{y} \|_1 \]

L1 Loss function: 
\[ \| x - \tilde{x} \|_1 \]

Image Credit: Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, Zhu et al.
The Conditional Distribution in CGANs

\[ A \quad \xrightarrow{} \quad p(B|A) \]

(a) Input night image

Image from: Zhu et al., Toward Multimodal Image-to-Image Translation, NIPS 2017
The Conditional Distribution in CGANs

\[ p(B|A) \]

Zhu et al., Toward Multimodal Image-to-Image Translation, NIPS 2017
BicycleGAN

KL-divergence loss

\( \sim p(B|A) \)

L2 loss

encoder

generator

\( z \)
BicycleGAN

\[ N(0, 1) \]

KL-divergence loss

L2 loss

cycle 1

encoder

A

generator

B

encoder

\sim p(B|A)

adversarial loss

discriminator

L2 loss

cycle 2
FrankenGAN

input: façade shape

1\textsuperscript{st} step: window/door layout

2\textsuperscript{nd} step: texture

3\textsuperscript{rd} step: sem. labels

separate training sets:
Progressive GAN

- Resolution is increased progressively during training
- Also other tricks like using minibatch statistics and normalizing feature vectors

![Diagram showing resolution progression in training](image-url)
**StackGAN**

Condition does not have to be an image

- **condition**
  - This bird is grey with white on its chest and has a very short beak
  - This bird is grey with white on its chest and has a very short beak

- **low-res generator**
  - high-res generator

- **low-res disc.**
  - high-res disc.

This flower has white petals with a yellow tip and a yellow pistil

A large bird has large thighs and large wings that have white wingbars

Zhang et al., StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks, ICCV 2017
Disentanglement

Entangled: different properties may be mixed up over all dimensions

Disentangled: different properties are in different dimensions

specified property: number

specified property: character

other properties

other properties

Mathieu et al., *Disentangling factors of variation in deep representations using adversarial training*, NIPS 2016
Attention and Gray Box Learning
Attention in Deep Learning

Why is this hard for the network?

1) Locality of convolutions
2) Driven only by data from shallower layers (no semantics)
Attention in Deep Learning

Problem: architecture constrains information flow. For example, in a typical CNN, at a given image location (red), information about other image locations (grey) is available in a resolution that depends on the spatial distance.
Attention Based on Semantics

Idea: use higher-level semantics to select relevant information

Spatial Transformer Networks

Residual Attention Network for Image Classification

Jaderberg et al., *Spatial Transformer Networks*, NIPS 2015

Wang et al., *Residual Attention Network for Image Classification*, CVPR 2017
Attention to Distant Details

Idea: gather information from distant details based on their features

Non-local Neural Networks

Wang et al., Non-local Neural Networks, CVPR 2018

Attention GAN

Zhang et al., Self-Attention Generative Adversarial Networks, CVPR 2018
Attention to Distant Details

Idea: gather information from distant details based on their features

Zhang et al., Self-Attention Generative Adversarial Networks, CVPR 2018
Squeeze and Excitation: Attention over Channels

Idea: weigh (emphasize and suppress) channels based on global information

Hu et al., Squeeze-and-Excitation Networks, CVPR 2018
Gray Box Learning

Problem: Most networks are black boxes.
Idea: Regress parameters for a small set of well-known operations.

Hu et al., *Exposure: A White-Box Photo Post-Processing Framework*, Siggraph 2018
Summary

• Common Architecture Elements
  (Dilated Convolution, Grouped Convolutions)

• Deep Features
  (Autoencoders, Transfer Learning, One-shot Learning, Style Transfer)

• Adversarial Image Generation
  (GANs, CGANs)

• Interesting Trends
  (Attention, “Gray Box” Learning)