Deep Learning II

8DM50 Machine Learning in Medical Imaging and Biology

Mitko Veta

Acknowledgements: slides by Jelmer Wolterink, UTwente

Learning goals for today

Have a good overview of neural networks architectures for (medical) image analysis

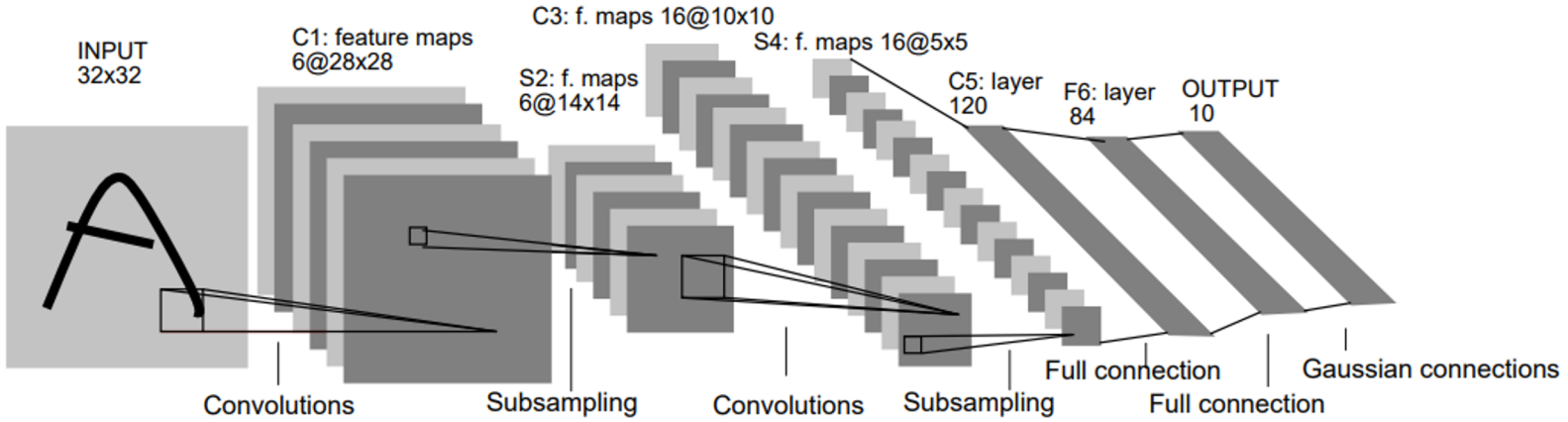
- Including architectures for sequential data

Obtain a general understanding of interpretability of deep learning models

Obtain a general understanding of generative adversarial networks

- Only one example of generative models, but arguably the most successful one

Recap: Convolutional neural networks



Demo

A standard convolu network consists of

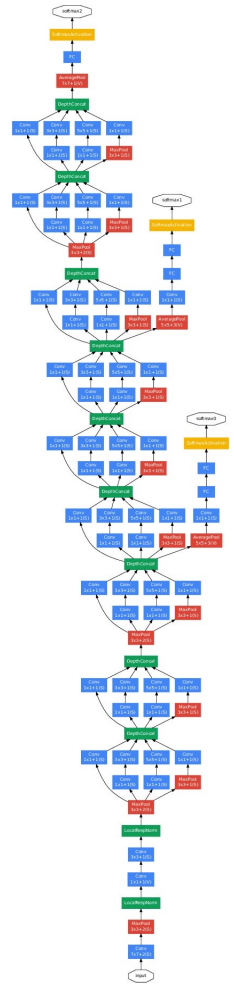
- **Convolutional** laye nsform input into feature maps
- **Subsampling** oper reduce size of feature maps, e.g. max pooling
- **Fully-connected** la erform classification (multi-layered perceptron)

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.

What happened between 1998 and 2012?

Compute

Data



Algorithms

Data: ImageNet challenge

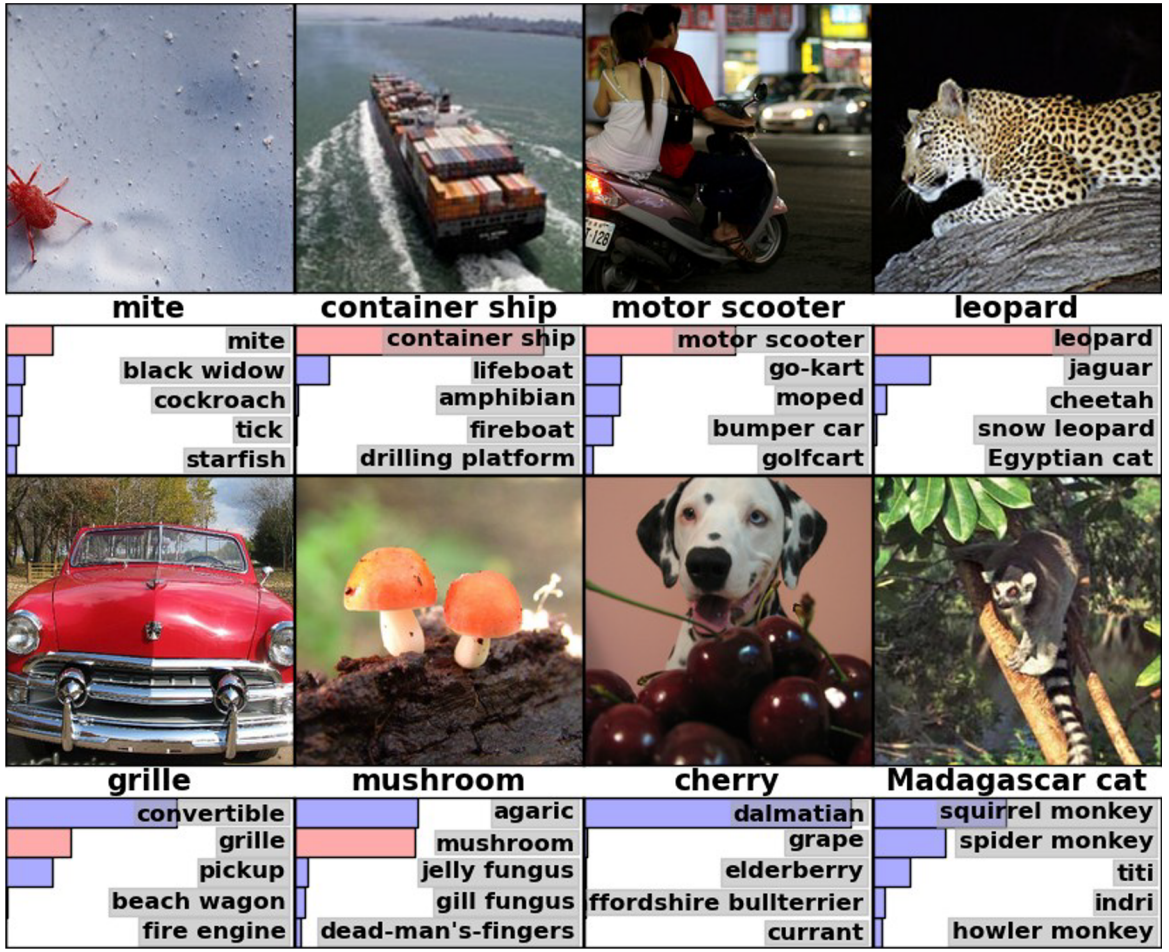
Benchmark for image classification/object detection

Data

- > 1,200,000 RGB images
- Images show one of **1000** classes

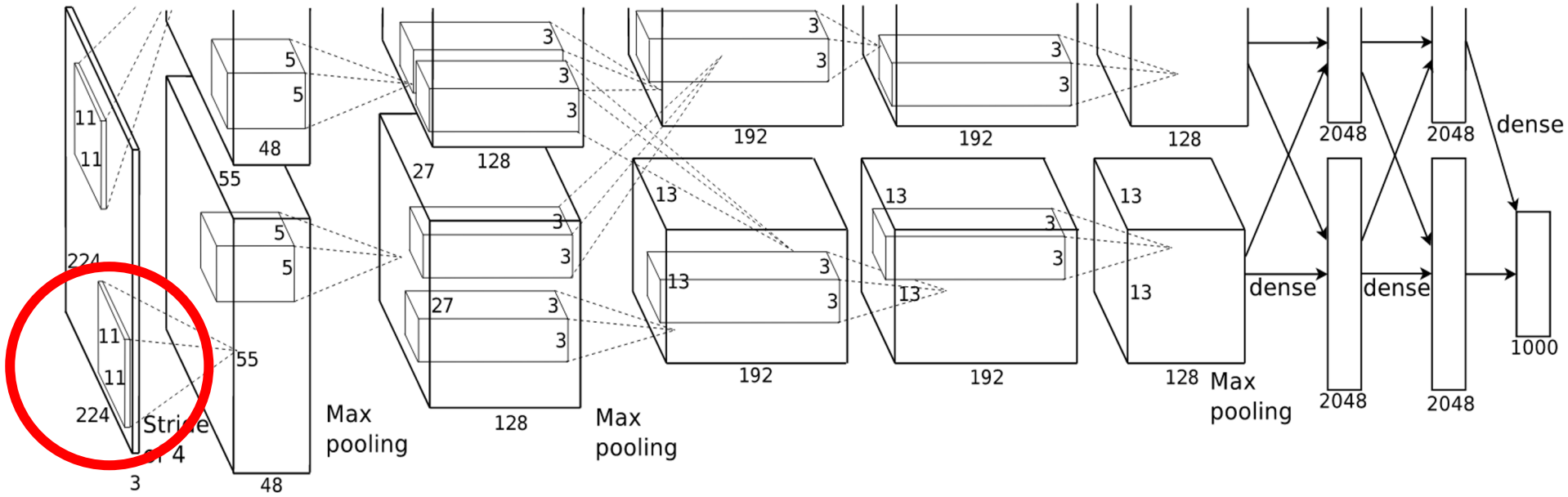
Task

- Detect label of image
- Top-1\top-5 accuracy



Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

AlexNet



- Substantially outperformed 'conventional' methods in 2012
- Convolutional + subsampling + fully connected layers
- Trained in parallel on two GPUs
- Training time
  - **2012**: 5 to 6 days (2 x GTX 580 3GB GPU)
  - **2017**: 24 minutes (supercomputer 32,000 cores)
- Large 11 x 11 convolution kernels

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

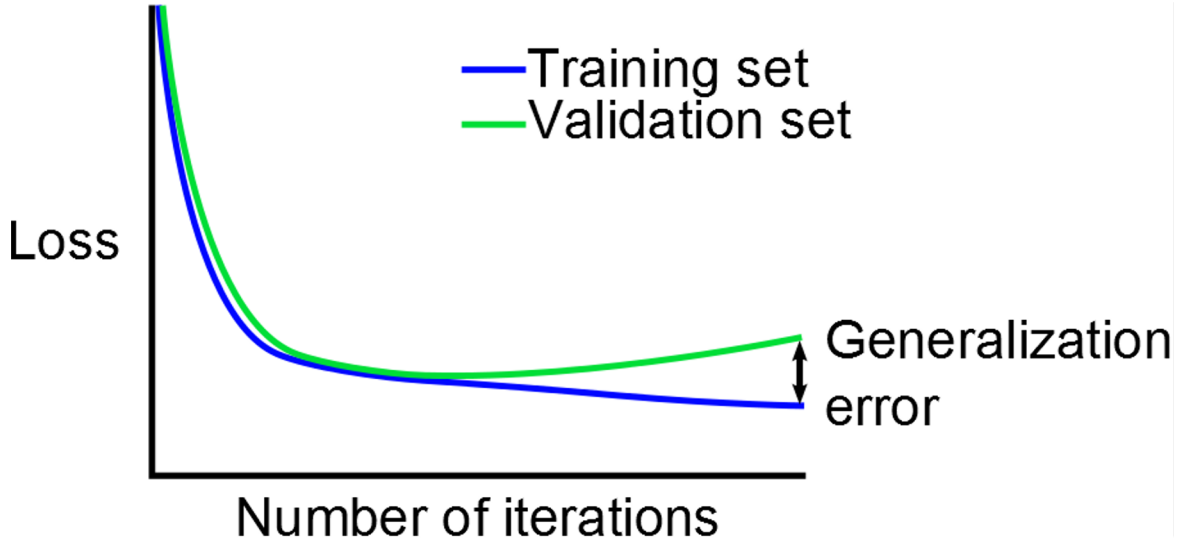
Overfitting

Reasons

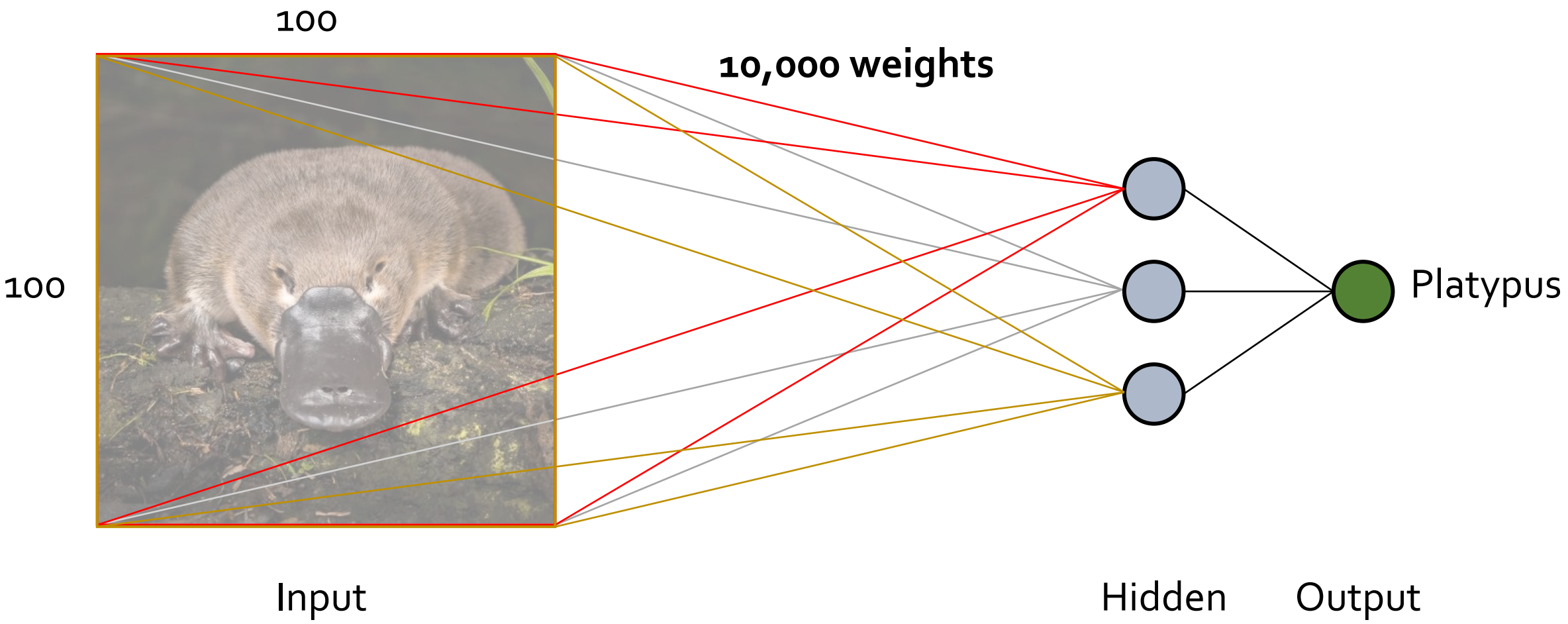
- Too many parameters
- Not enough data

One solution

- Reduce number of parameters

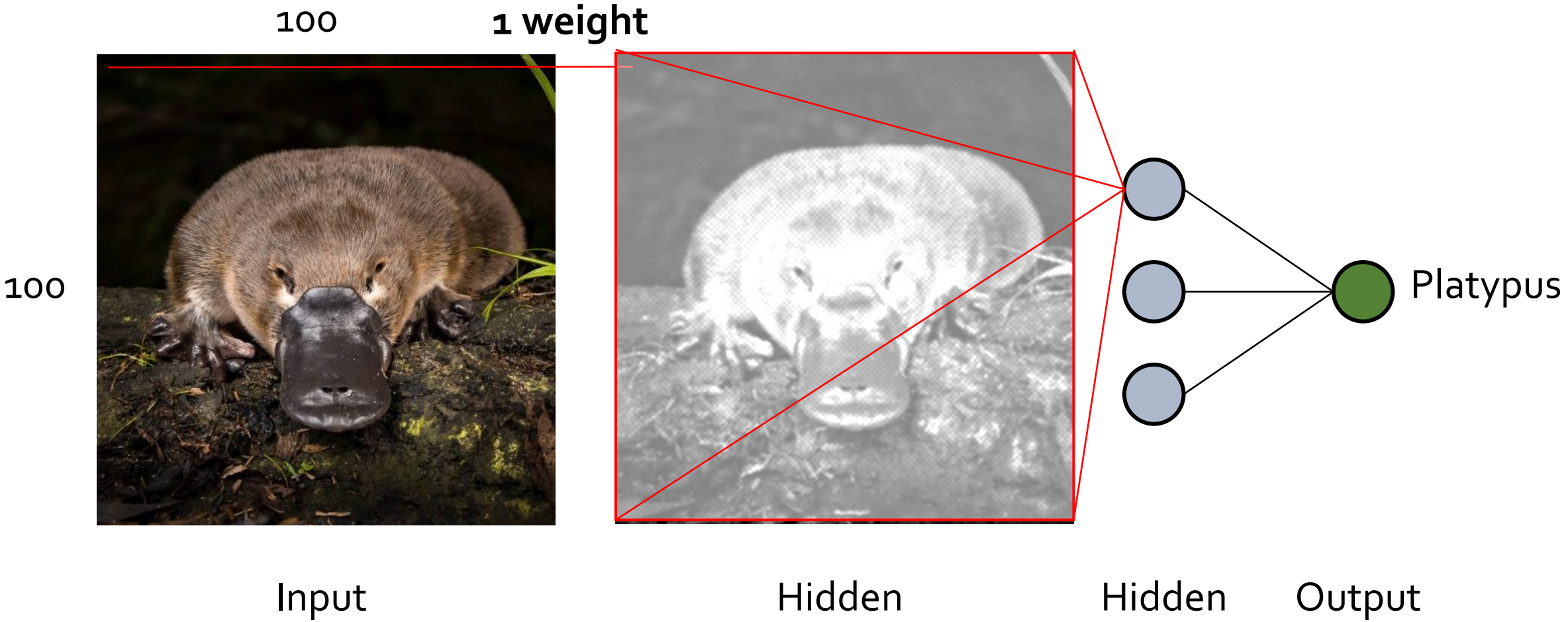


Kernel size



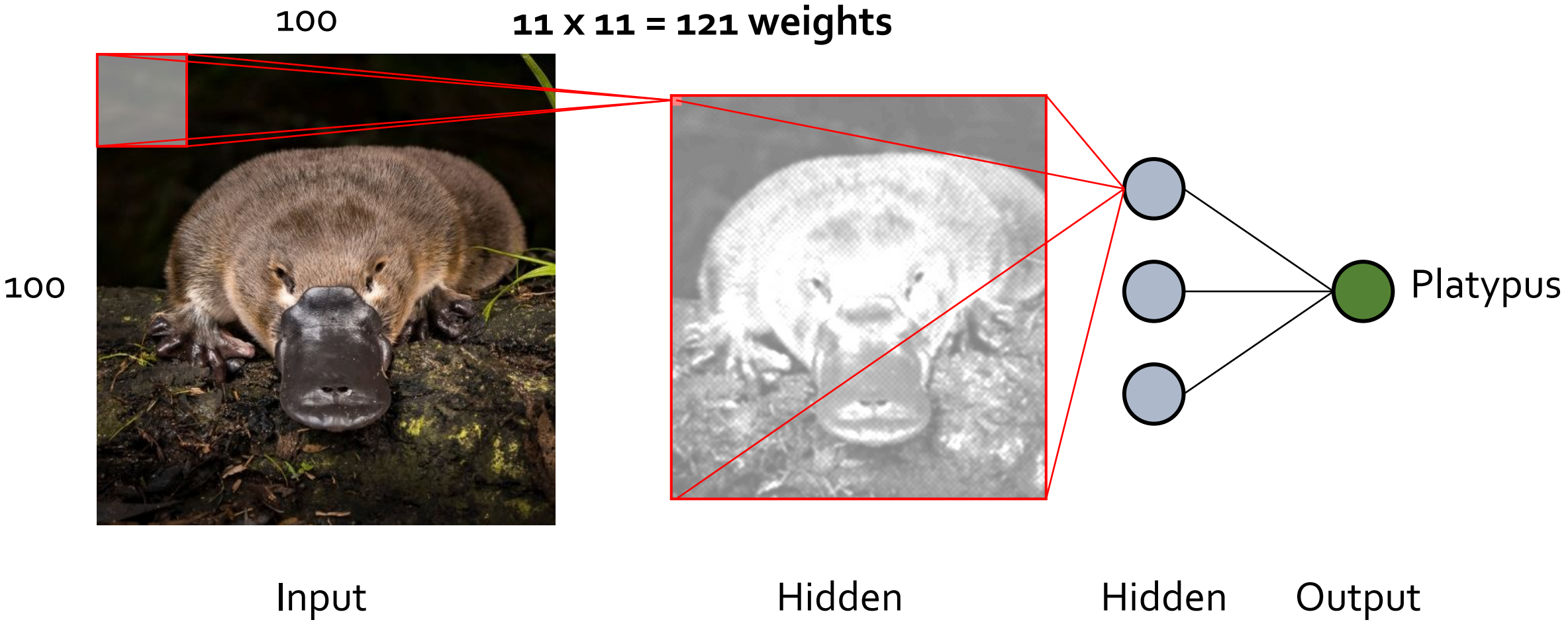
https://www.abc.net.au/news/2018-11-10/platypus-vulnerable-to-extinction-researchers-say/10477902

Kernel size



https://www.abc.net.au/news/2018-11-10/platypus-vulnerable-to-extinction-researchers-say/10477902

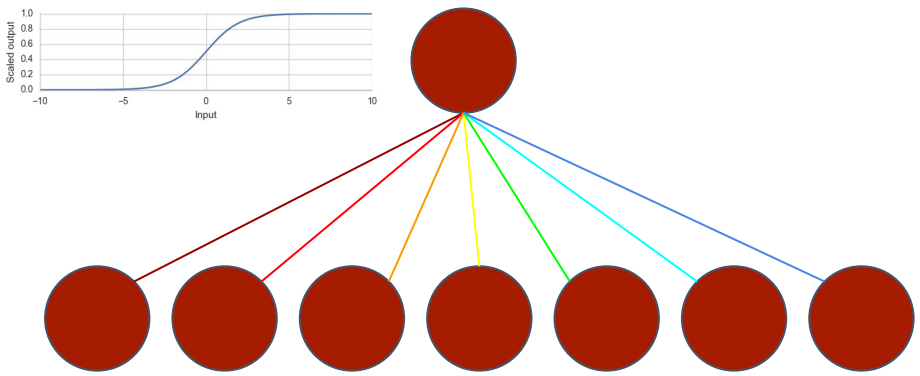
Kernel size

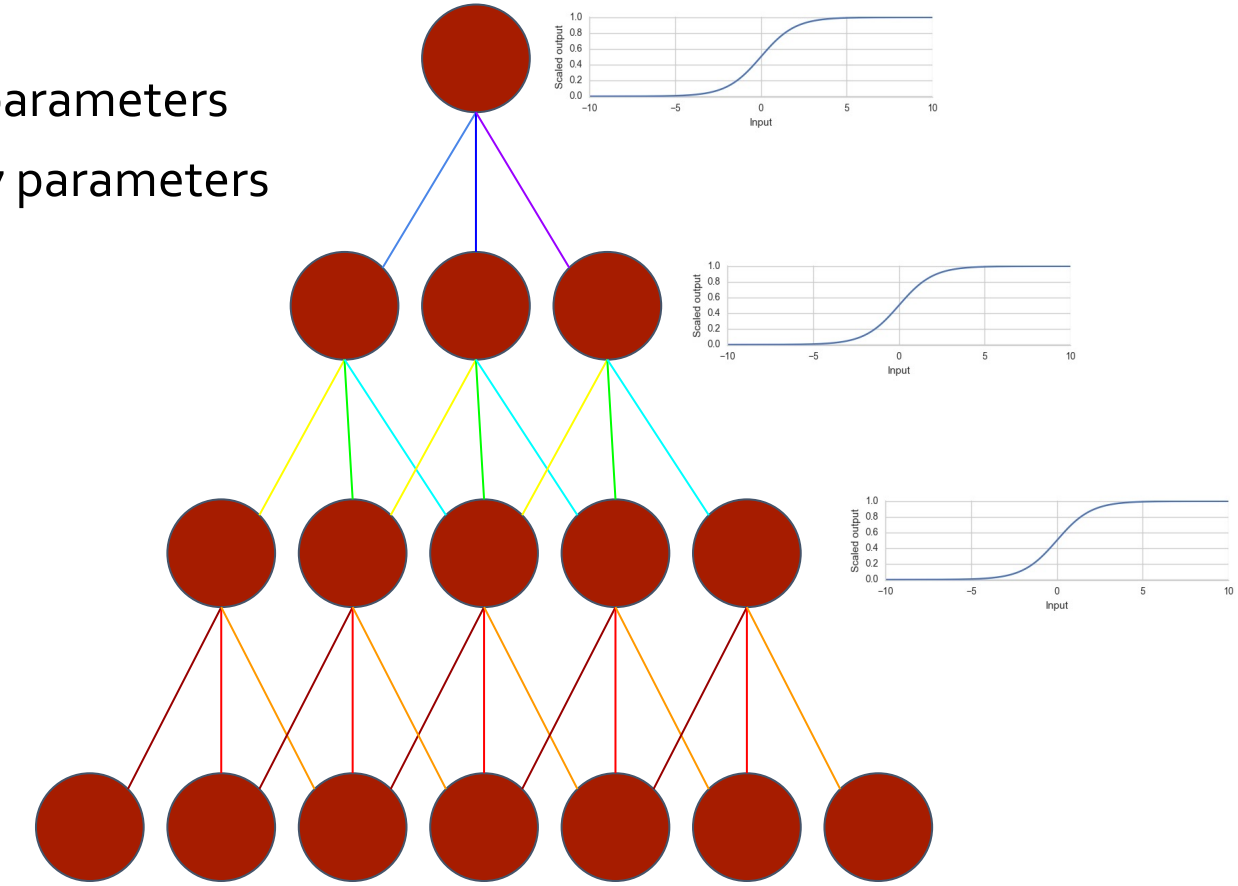


https://www.abc.net.au/news/2018-11-10/platypus-vulnerable-to-extinction-researchers-say/10477902

Using smaller kernels

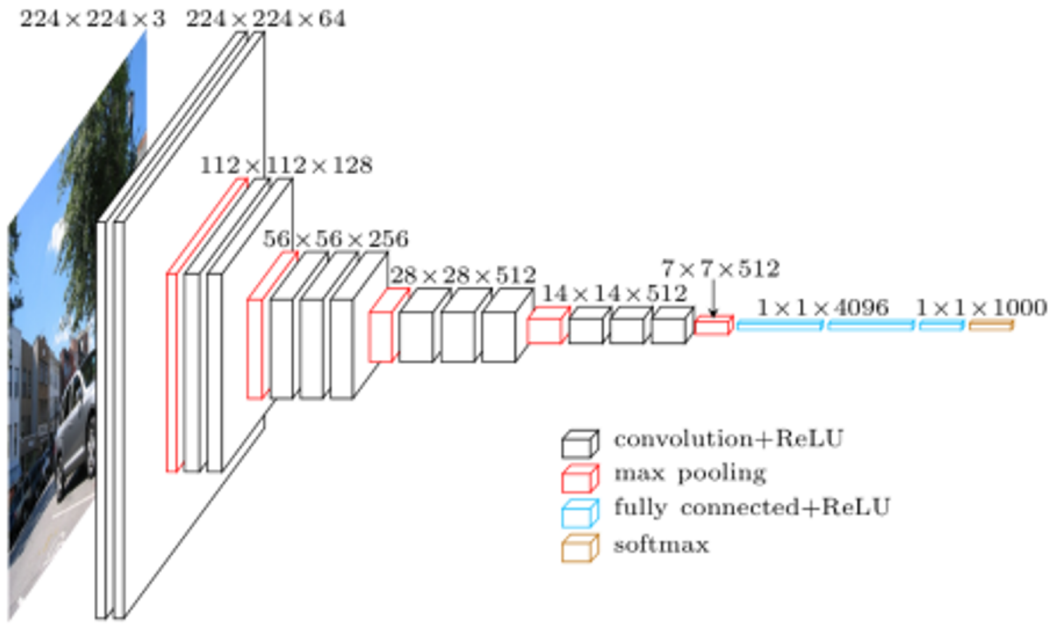
- Large kernels have many parameters: 7 x 7 = 49 parameters
- Smaller kernels reduce parameters: 3 x (3 x 3) = 27 parameters
- More nonlinearities means more abstraction





Simonyan et al. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

VGG-Net

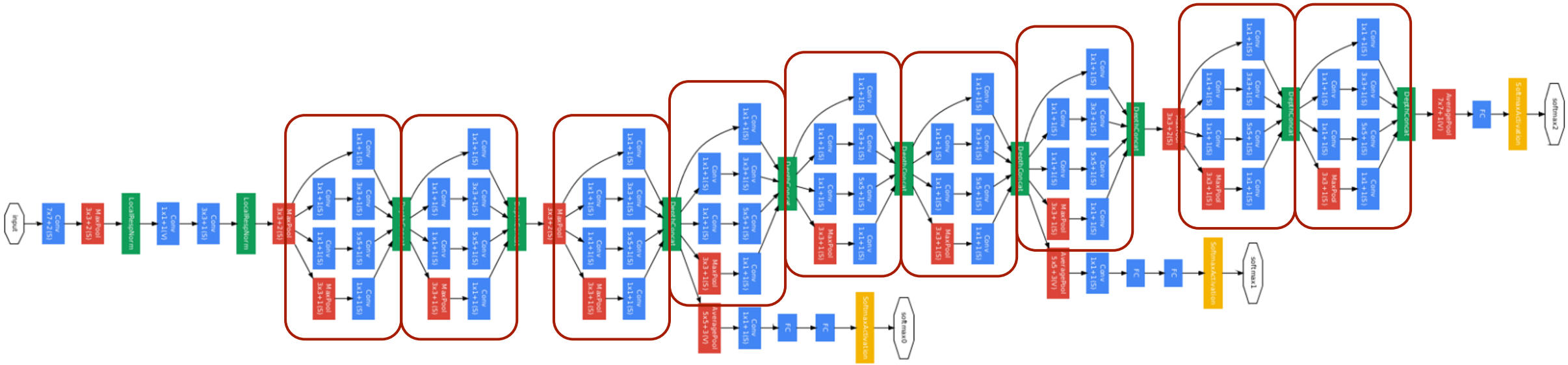


ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Simonyan et al. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

GoogLeNet (Inception v1)

- 22 layer-network
- Very deep compared to LeNet/AlexNet
- SOTA on ImageNet (when published)



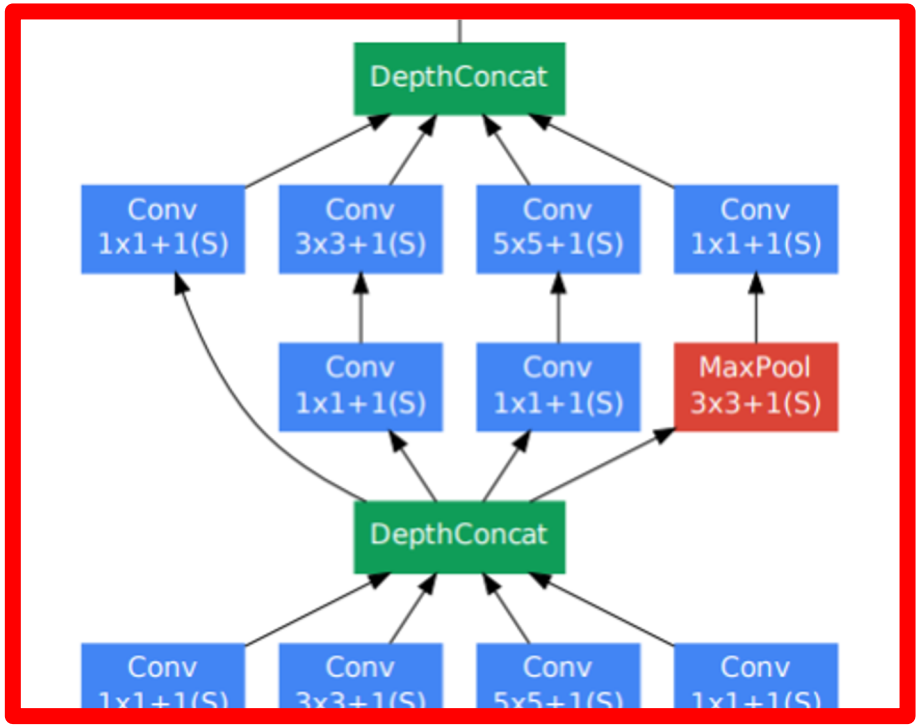
Szegedy et al. "Going deeper with convolutions." CVPR 2015.

Inception module

- Combine parallel multi-scale convolutions
- Let the model pick best filter size

1x1 3x3 5x5

- Bottleneck layers: 1 x 1 convolutions
  - Aggregate feature maps
  - Prevent explosion in number of parameters

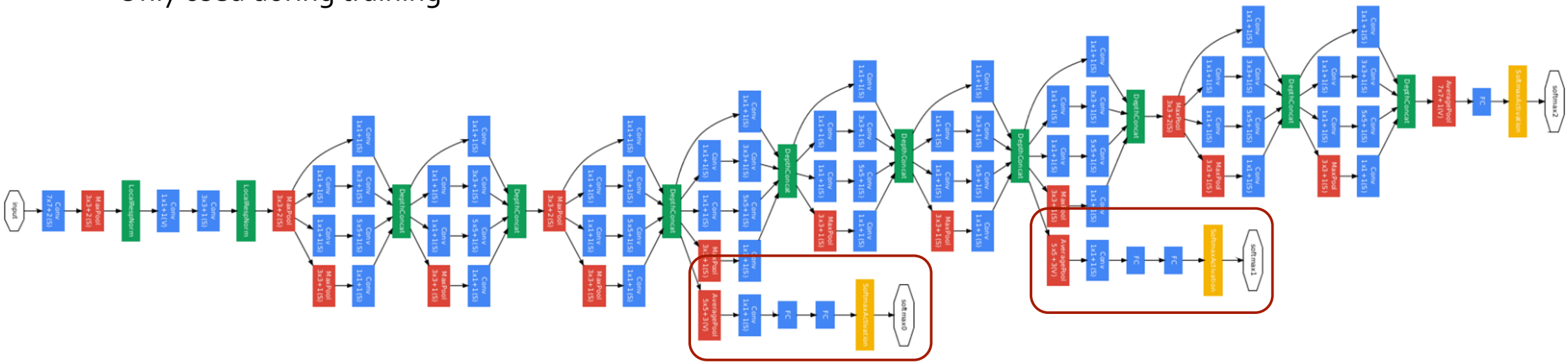


Szegedy et al. "Going deeper with convolutions." CVPR 2015.

Auxiliary classifiers

Auxiliary classifiers provide extra supervision

- Vanishing gradients
- Enforce useful features at intermediate layers
- Only used during training



Szegedy et al. "Going deeper with convolutions." CVPR 2015.

Residual connections

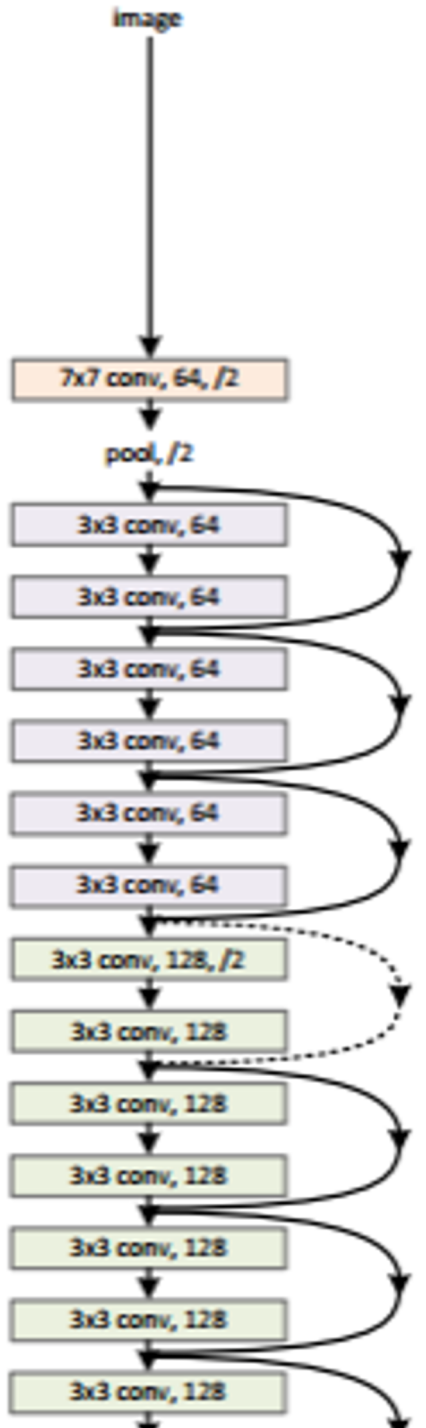
Very deep networks

- allow learning of better representations
- are difficult to optimize due to vanishing gradients

Residual connections can skip layers $H(x) = x + F(x)$

A deep network is at least as strong as it's shallower variant

- If adding layers doesn't help, just use the skip connection



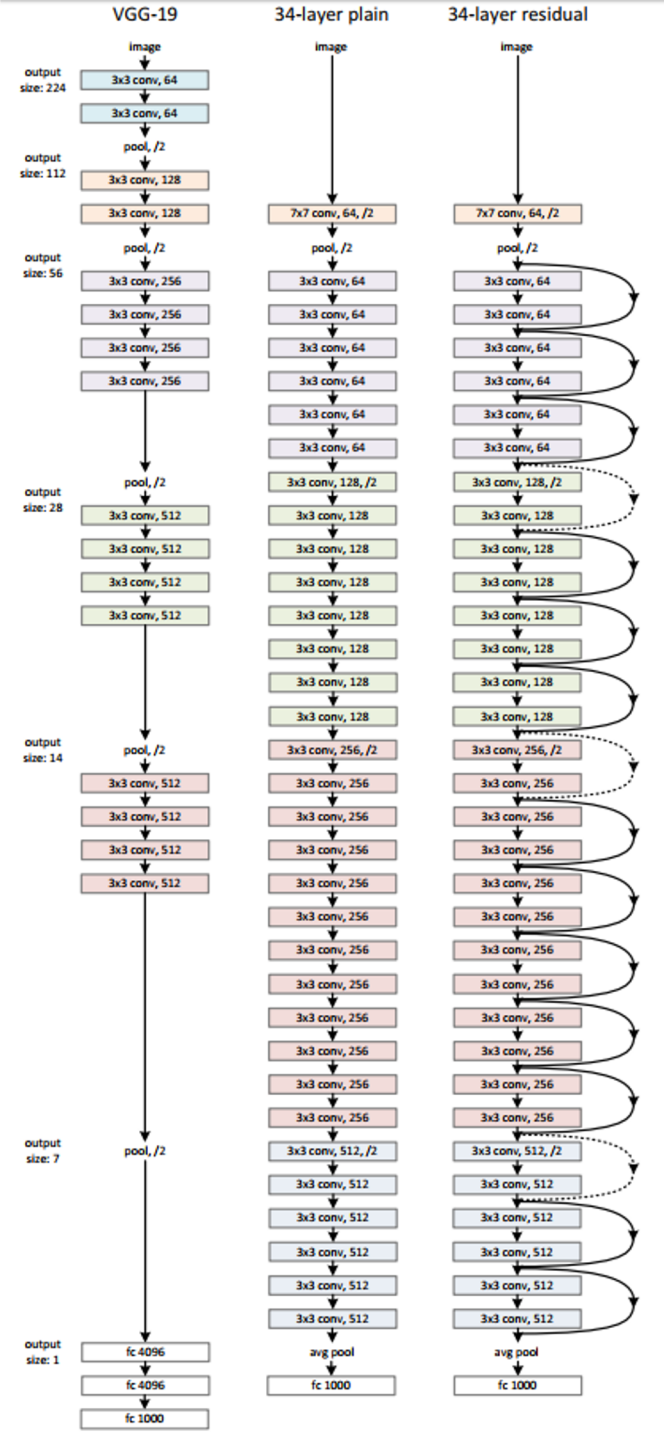
He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In CVPR (pp. 770-778).

Residual network (ResNet)

- Organize layers in blocks
- Use bottleneck layers
- Residual connections barely add computational complexity

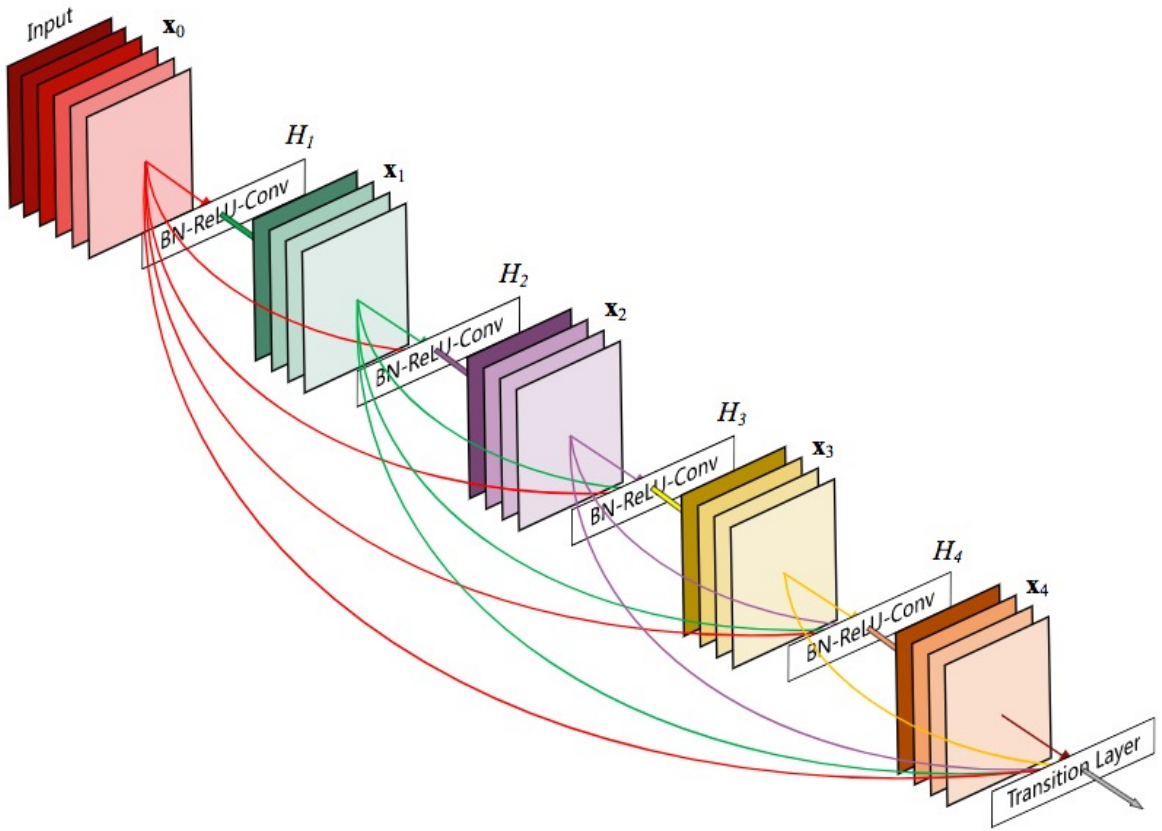
- SOTA on ImageNet (when published)

- Inspired
  - Wide residual nets (50-layer wide ResNet > 152-layer ResNet)
  - DenseNets: get identity mapping from all previous layers



He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In CVPR (pp. 770-778).

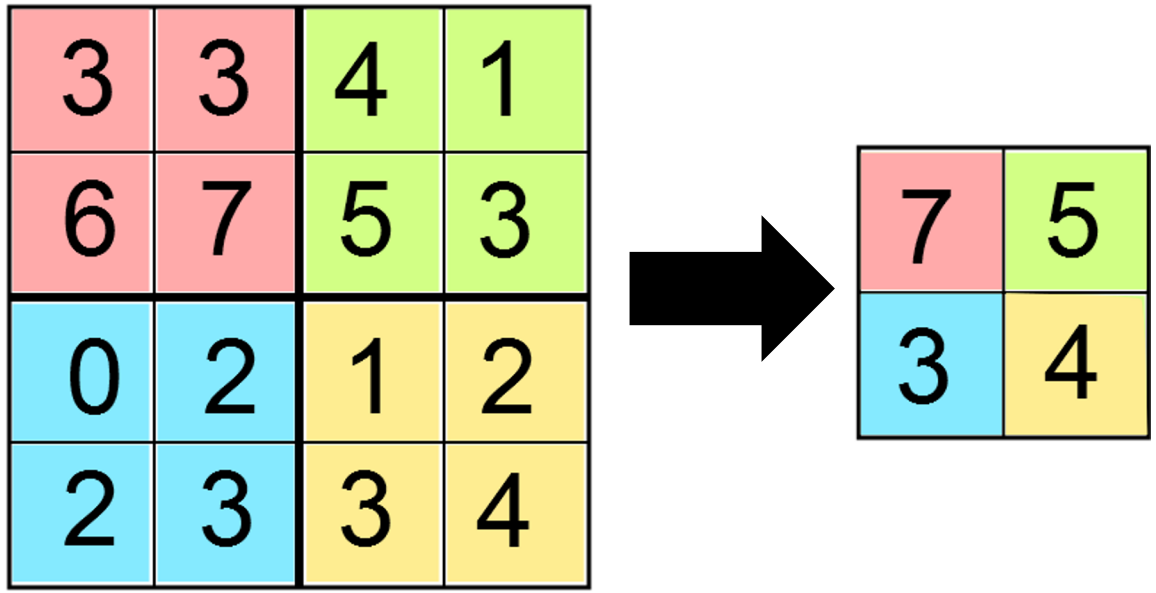
Dense networks



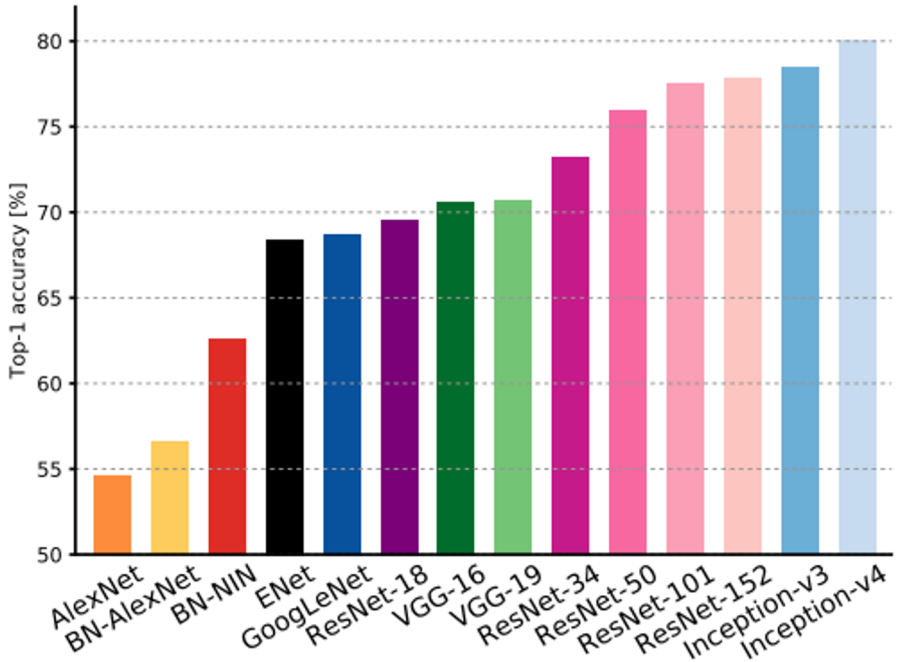
Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).

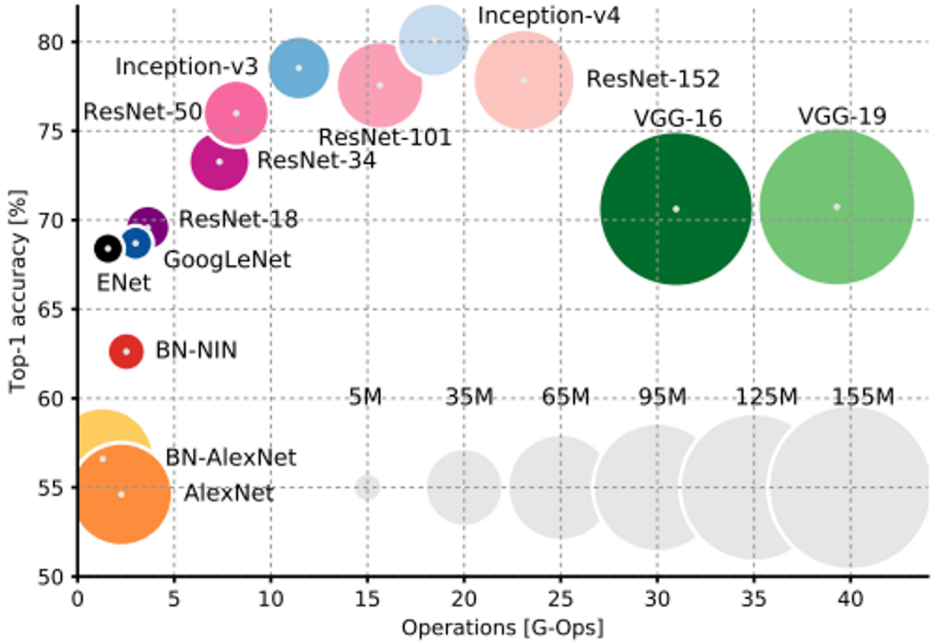
Downsampling

- We often want to go from a large image to a single prediction
- Use downsampling operations like pooling
- Pooling is not trainable



Complexity vs. accuracy

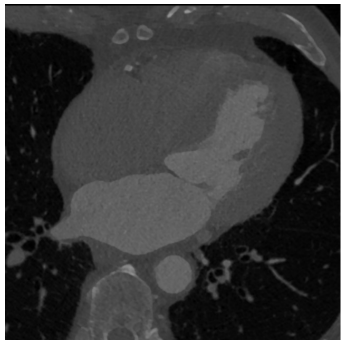




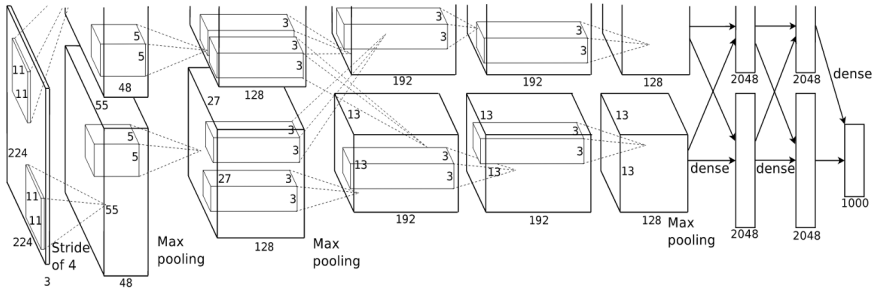
Canziani et al.. "An analysis of deep neural network models for practical applications." arXiv preprint arXiv:1605.07678 (2016).

Example: Organ localization in CT

2D image



AlexNet

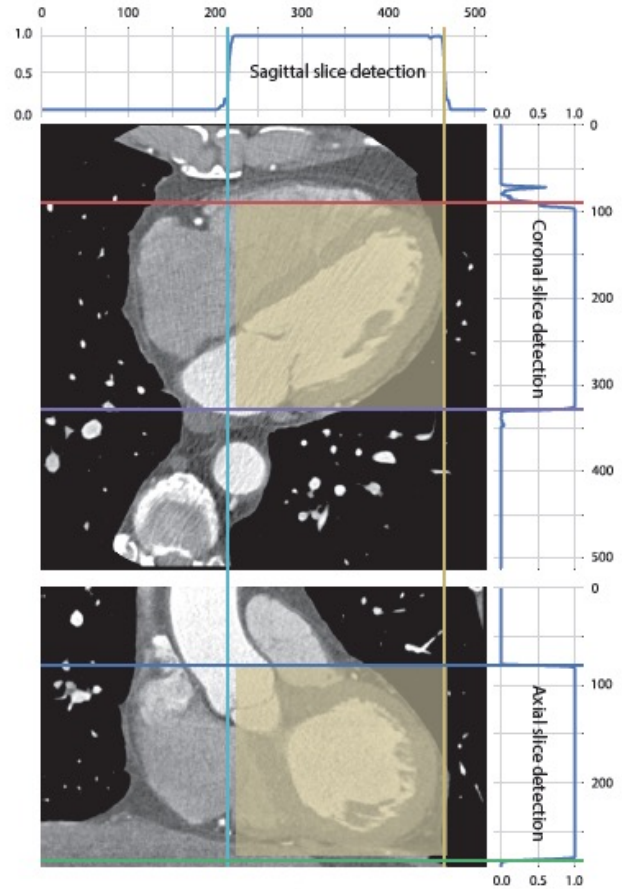


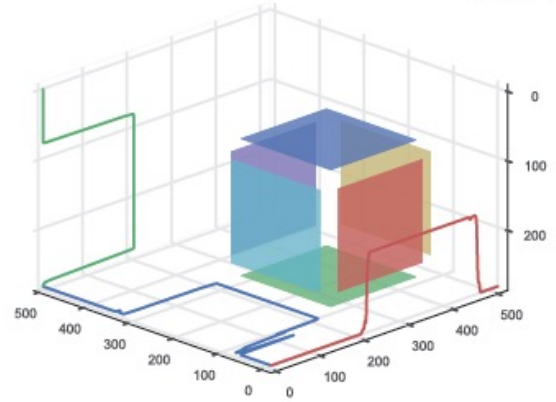
Ventricle?

Yes

No

For each image slice

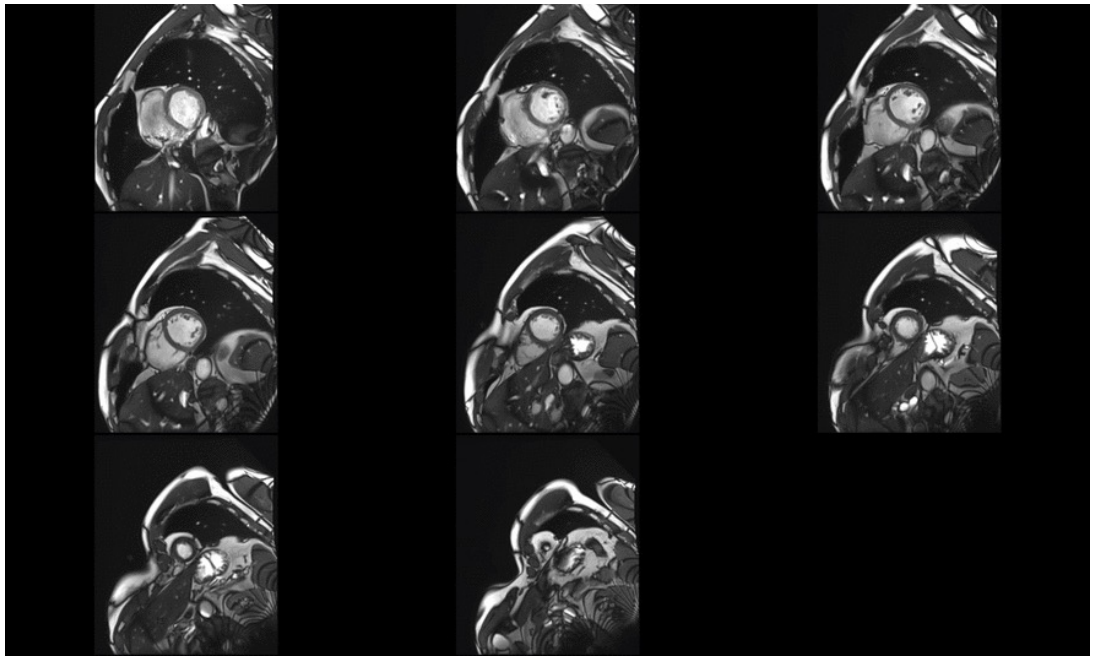




de Vos, B. D., Wolterink, J. M., de Jong, P. A., Leiner, T., Viergever, M. A., & Isgum, I. (2017). ConvNet-based localization of anatomical structures in 3-D medical images. IEEE TMI, 36(7), 1470-1481.

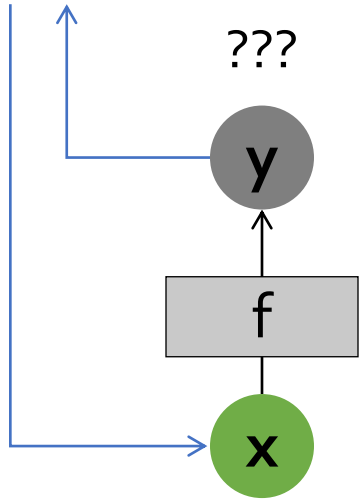
Sequences

- A lot of data is sequential
- E.g. videos, audio, text, ECG, medical images, …
- Can we use this in our neural network?



Recurrent neural networks (RNNs)

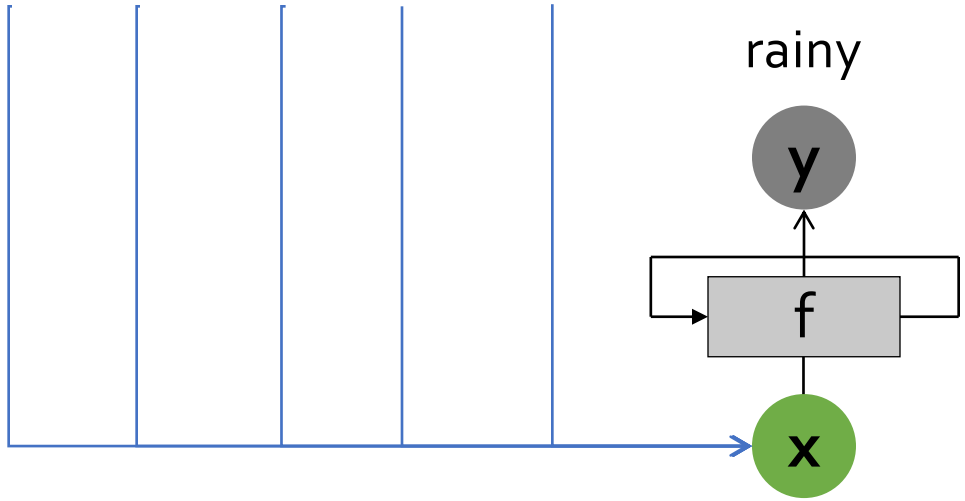
- It would be good to use information that came before
- A feedforward neural network has no 'memory'
- Consider training a neural network to predict the next word
    - "It's September, the weather is …"



http://colah.github.io/posts/2015-08-Understanding-LSTMs/

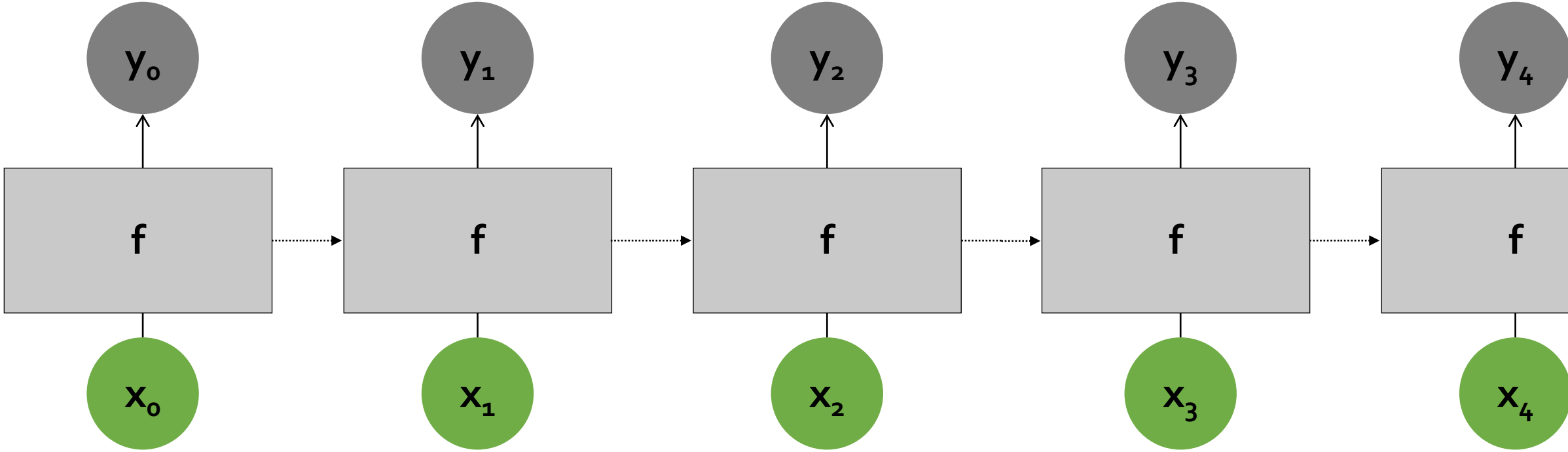
Recurrent neural networks (RNNs)

- It would be good to use information that came before
- A feedforward neural network has no 'memory'
- Consider training a neural network to predict the next word
    - "It's September, the weather is …"



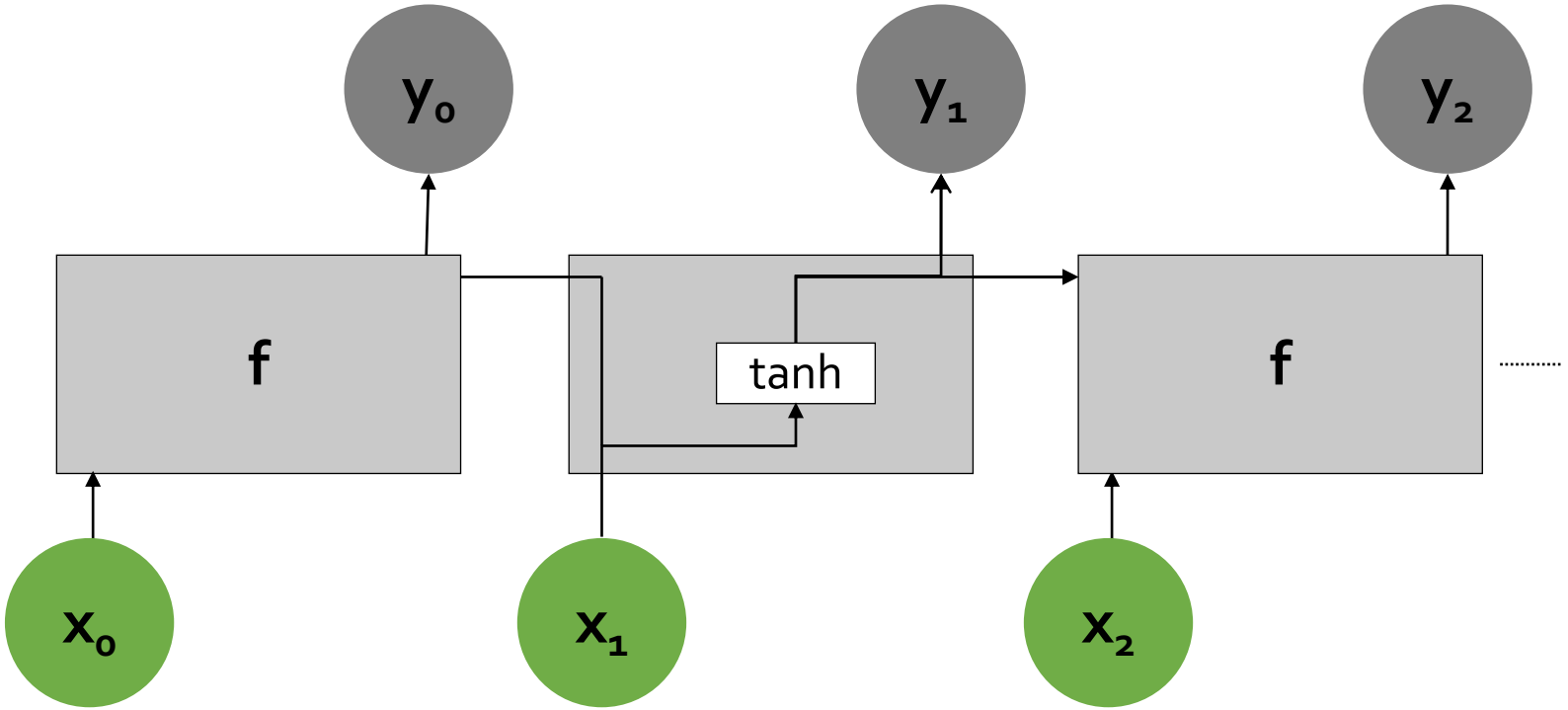
http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Unrolling a recurrent neural network



Unrolling a recurrent neural network

- Traditional RNNs have poor memory
- Previous outputs will get overwritten



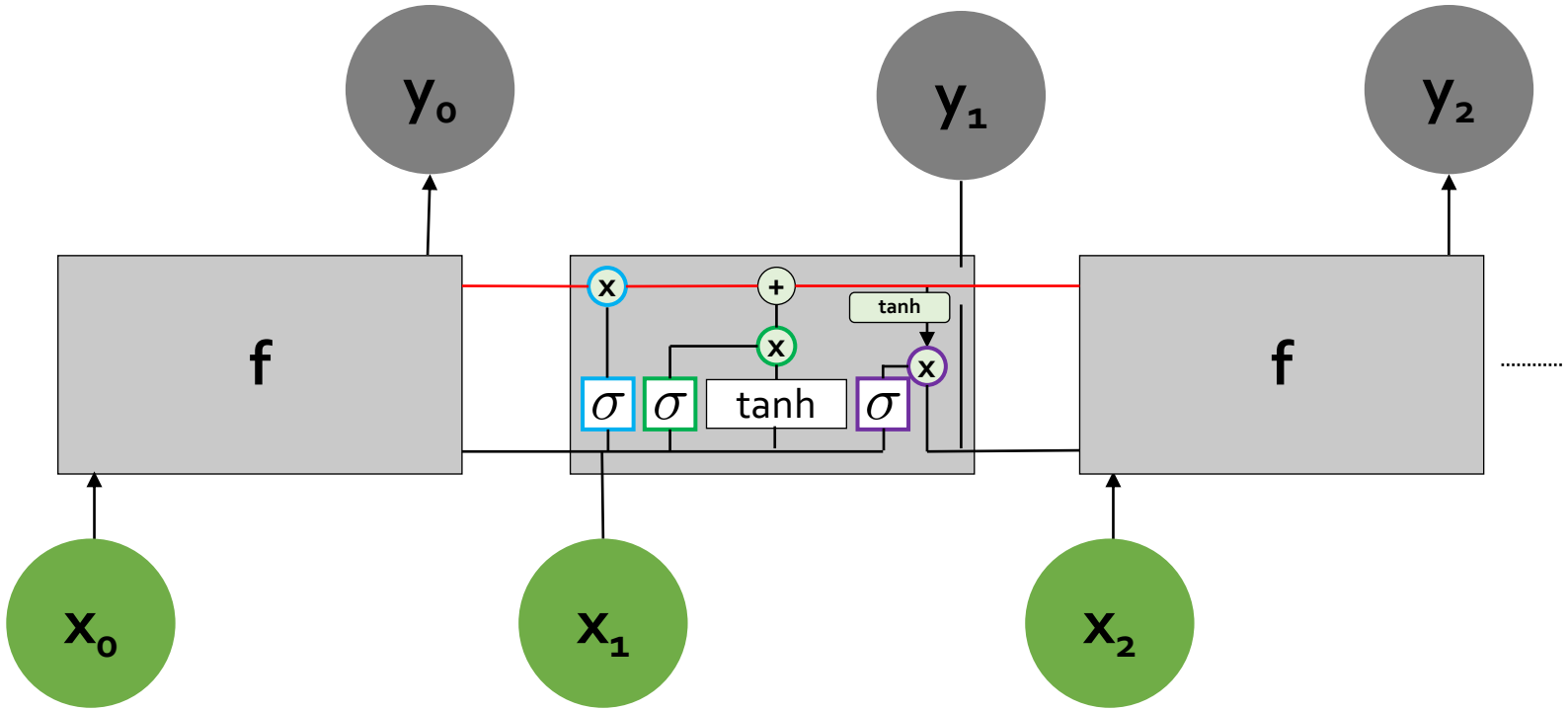
http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long short-term memory (LSTM)

1.Cell state

2.Gates

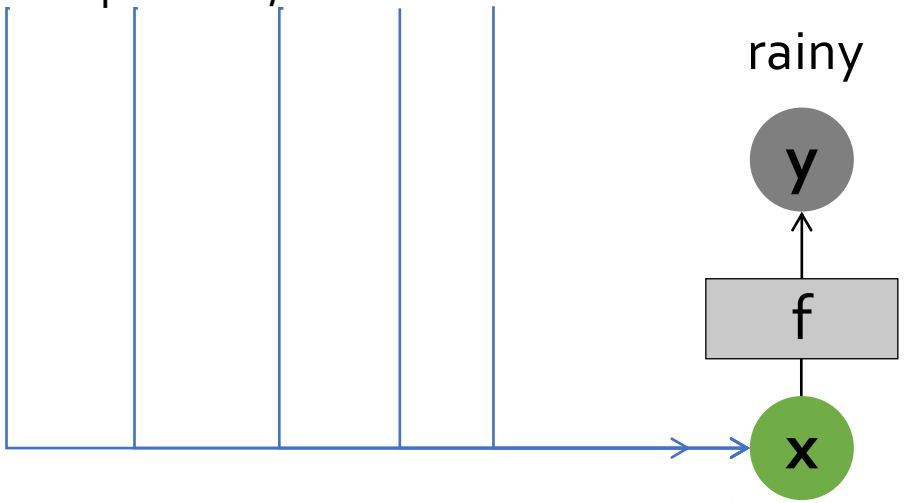
- Forget gate
- Input gate
- Output gate



Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.
http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Recurrent vs. feedforward

- Recurrent networks are intuitively appealing, but
  - feedforward networks are faster (parallel), simpler and they often very competitive
  - "It's September, the weather is …"



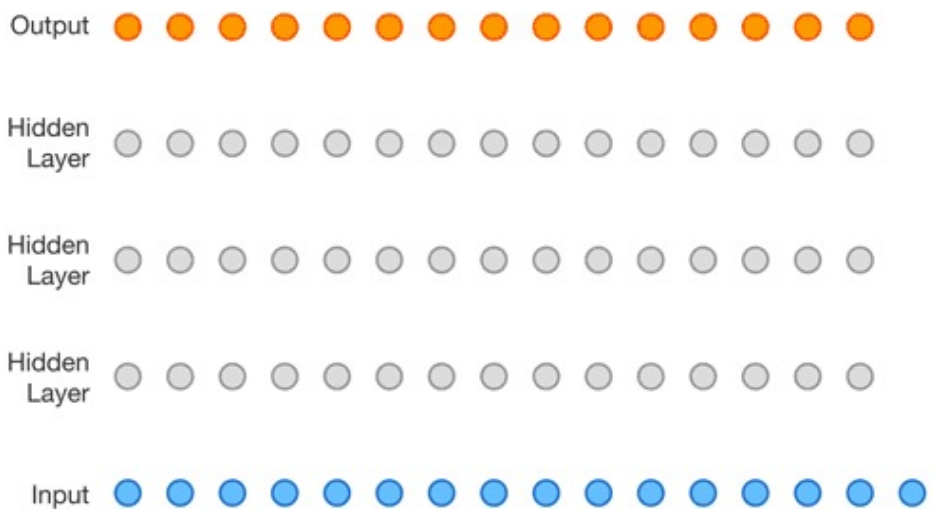
- One way to deal with large contexts in feedforward networks
  - dilated convolutions

Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271.

Dilated convolutions

In each layer, add more spacing between elements

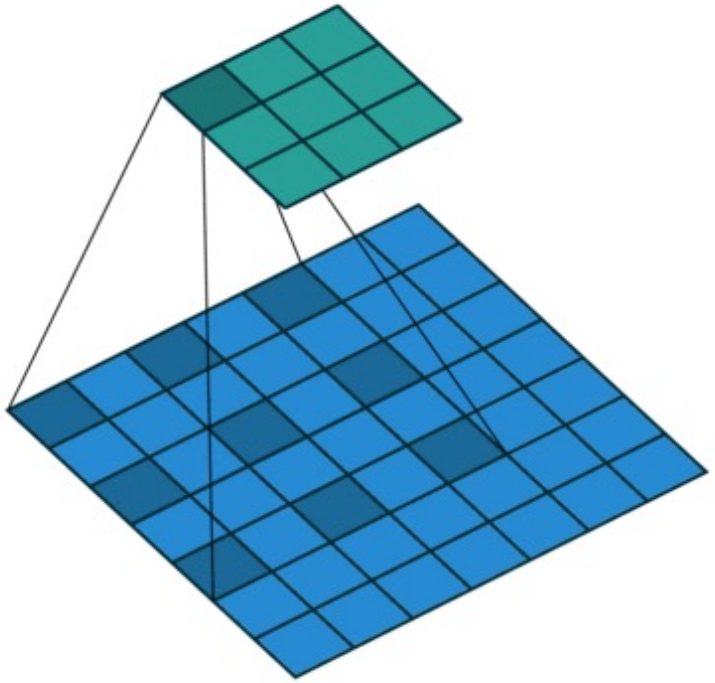
- increase receptive field
- prevent explosion in number of parameters



https://deepmind.com/blog/wavenet-generative-model-raw-audio/

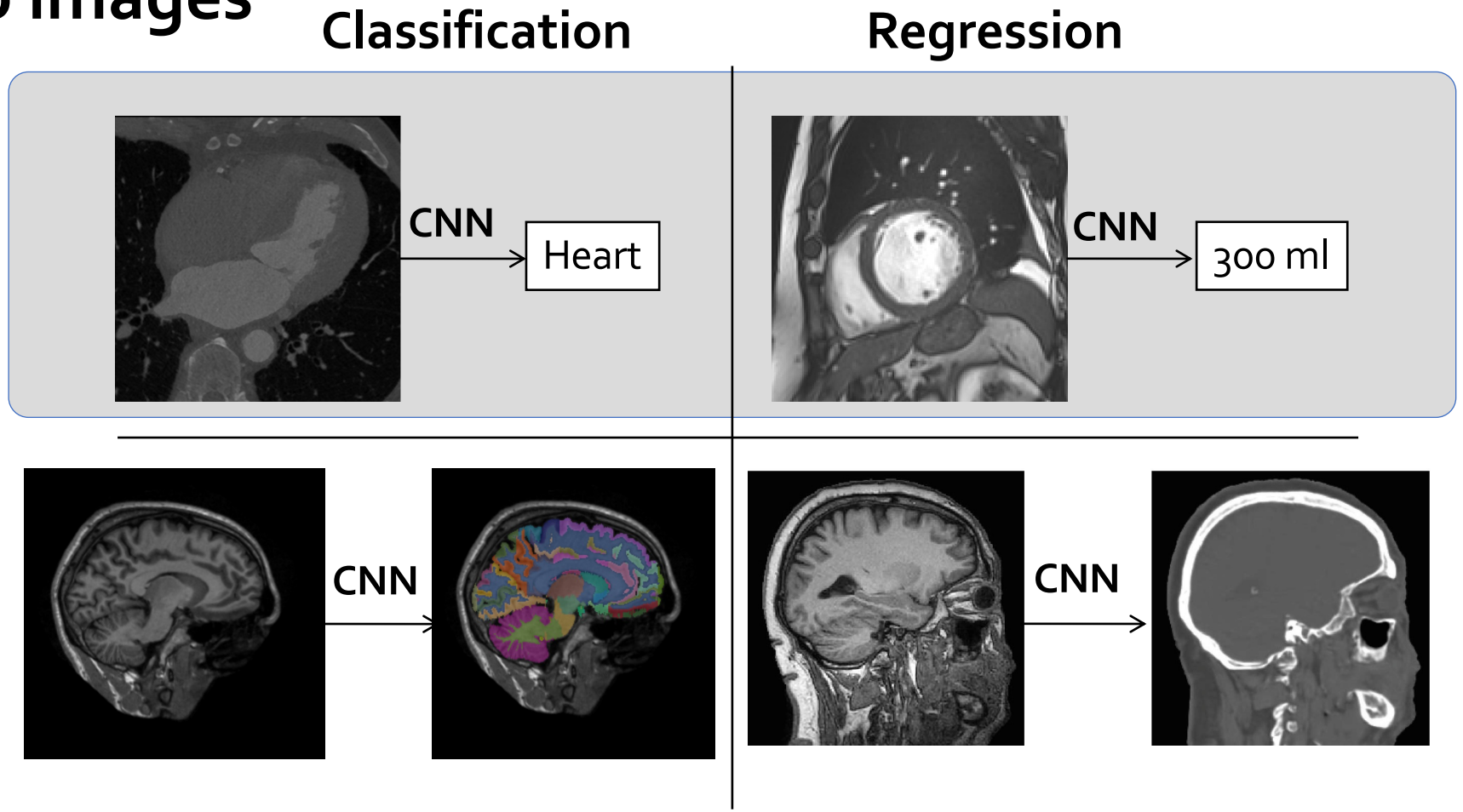
Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271.

Dilated convolutions



https://github.com/vdumoulin/conv_arithmetic/

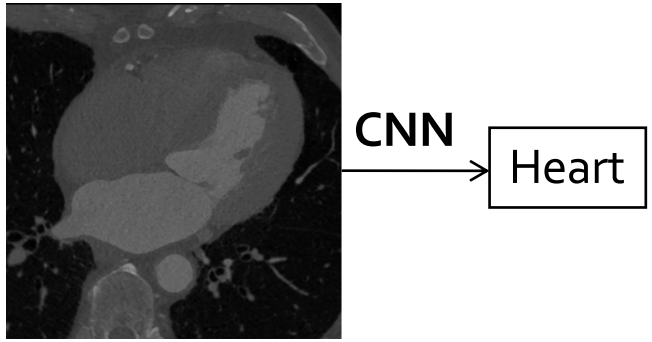
Back to images



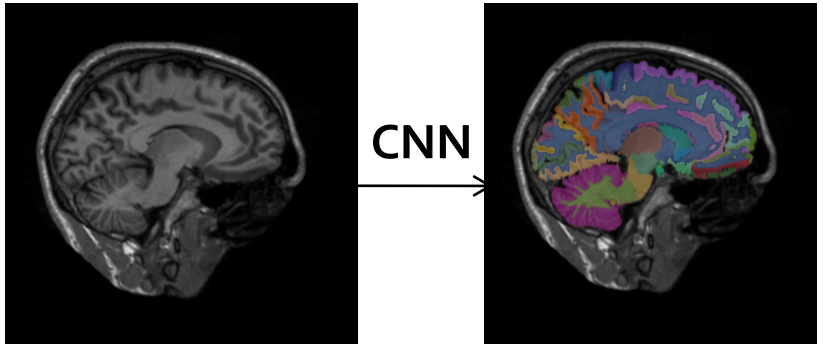
de Vos, Bob D., et al. "ConvNet-based localization of anatomical structures in 3-D medical images." IEEE Trans Med Imaging 36.7 (2017): 1470-1481.
Luo, Gongning, et al. "Multi-views fusion CNN for left ventricular volumes estimation on cardiac MR images." IEEE Transactions on Biomedical Engineering 65.9 (2018): 1924-1934.
Moeskops, Pim, et al. "Automatic segmentation of MR brain images with a convolutional neural network." IEEE transactions on medical imaging 35.5 (2016): 1252-1261.
Wolterink, Jelmer M., et al. "Deep MR to CT synthesis using unpaired data." International Workshop on Simulation and Synthesis in Medical Imaging. Springer, Cham, 2017.

CNNs for pixelwise prediction

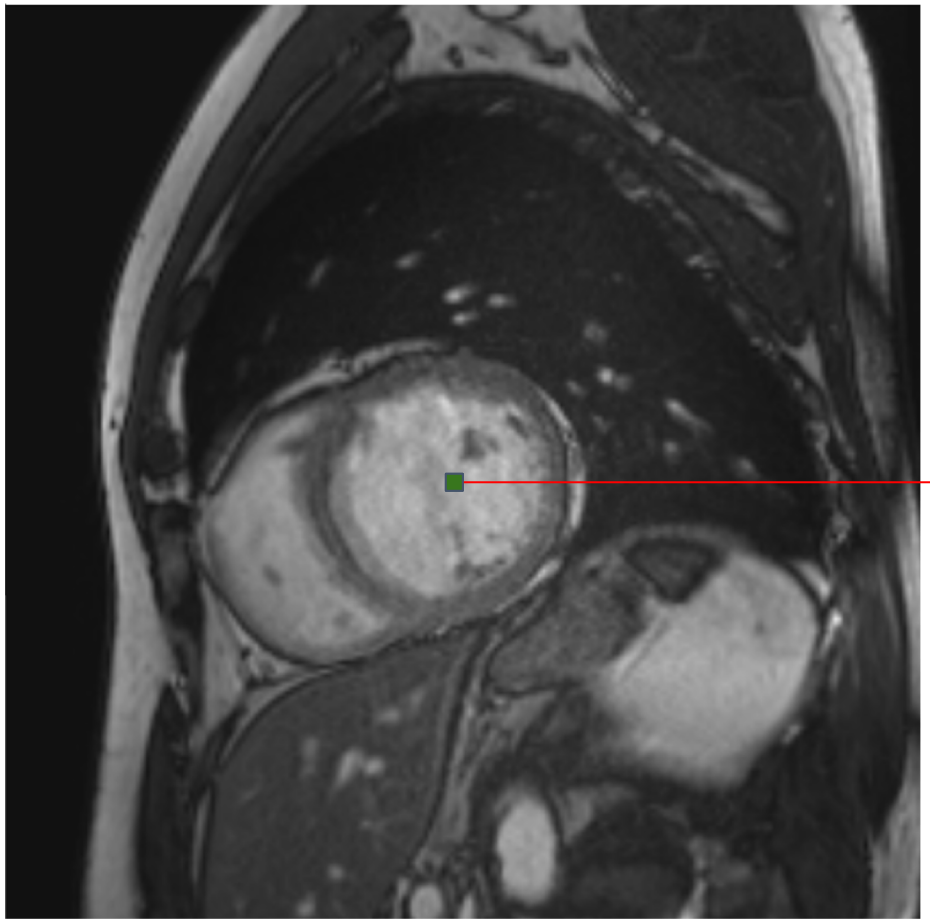
LeNet, AlexNet, VGG-Net, GoogLeNet all predict one value per **image**



Often, we want to predict one value per **pixel/voxel**



Sliding window



- **Image** = patch centered at voxel
- **Label** = class of center voxel

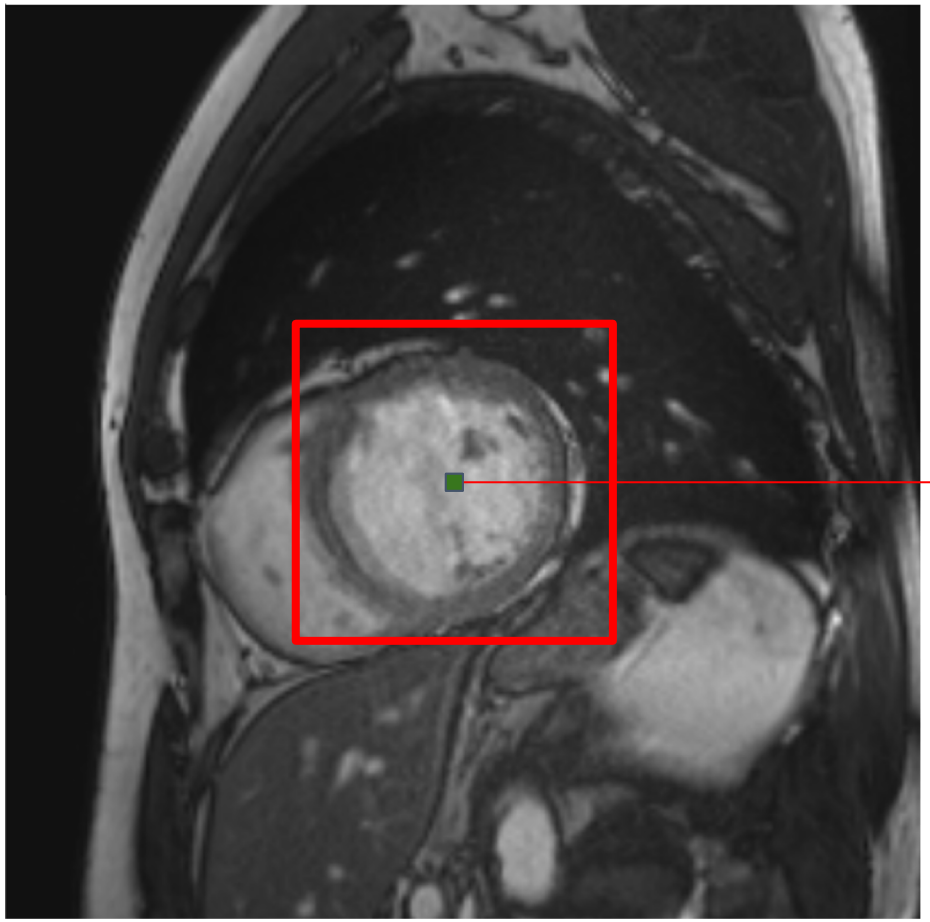
Background

Left ventricle

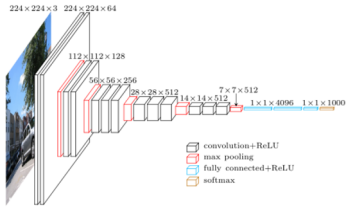
Myocardium

Right ventricle

Sliding window



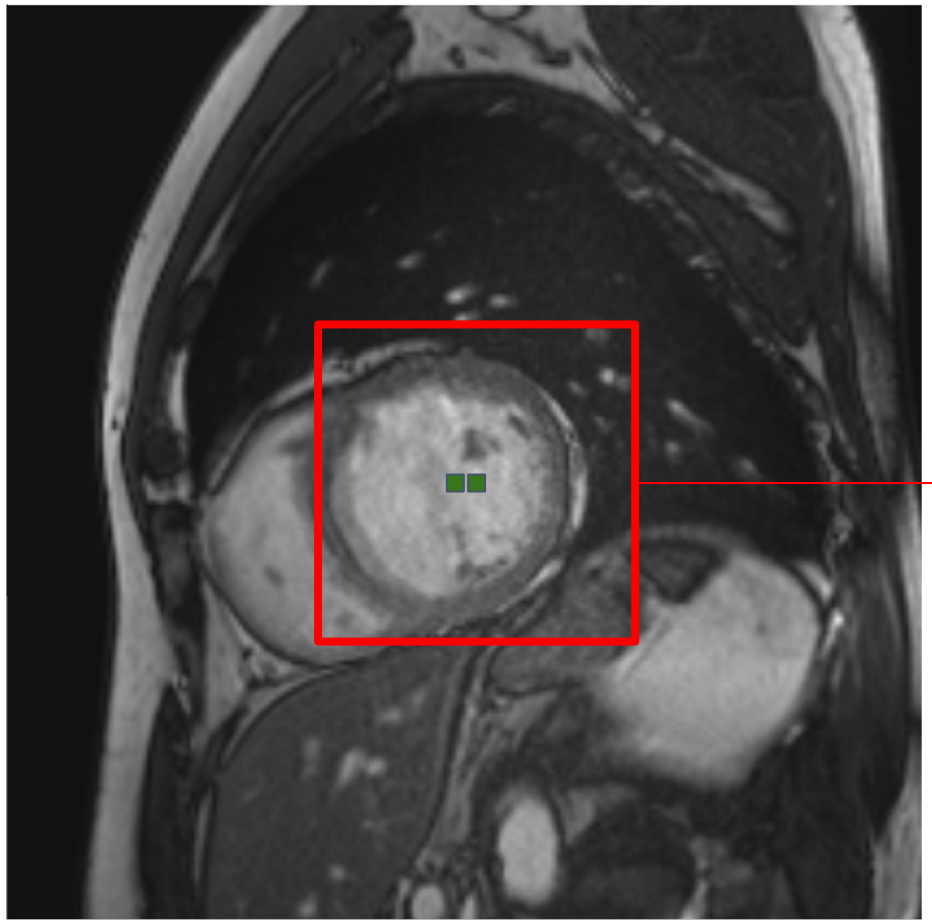
- **Image** = patch centered at voxel
- **Label** = class of center voxel



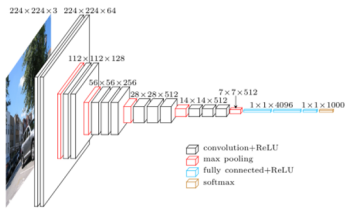
- Background
- Left ventricle
- Myocardium
- Right ventricle

Sliding window

Combination of thousands of image classification tasks



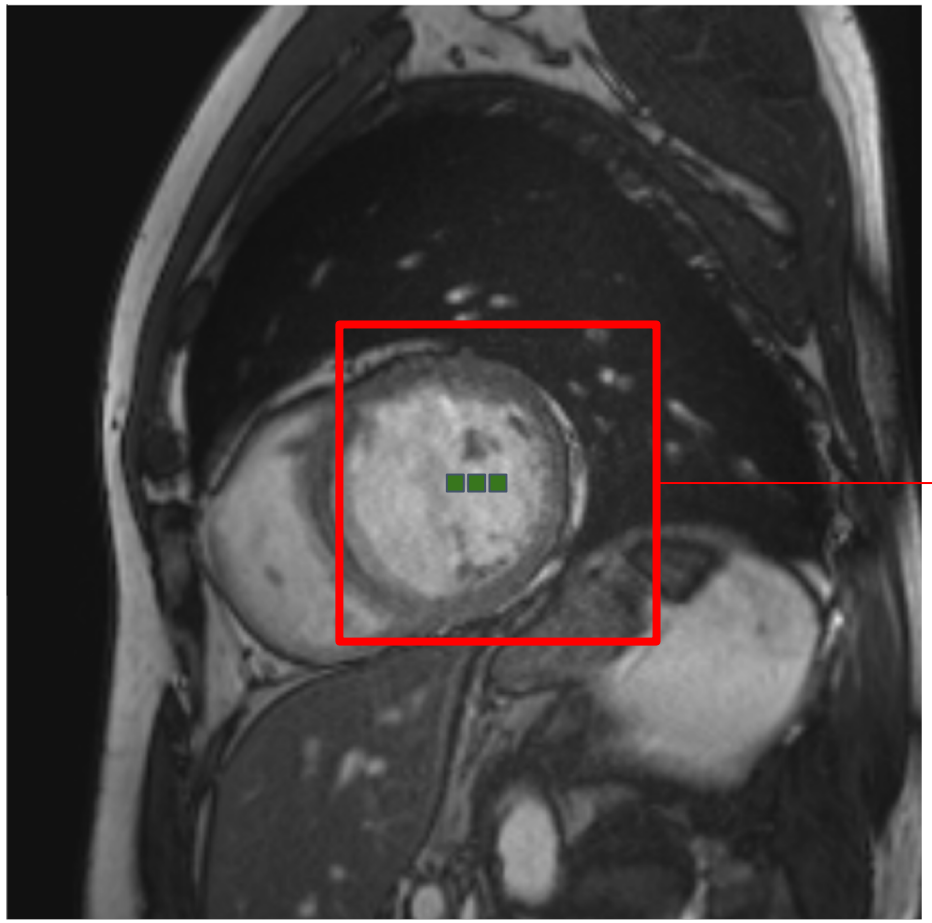
- **Image** = patch centered at voxel
- **Label** = class of center voxel



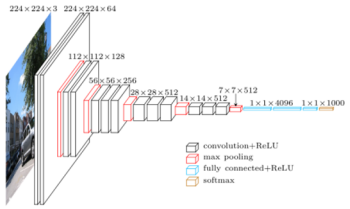
- Background
- Left ventricle
- Myocardium
- Right ventricle

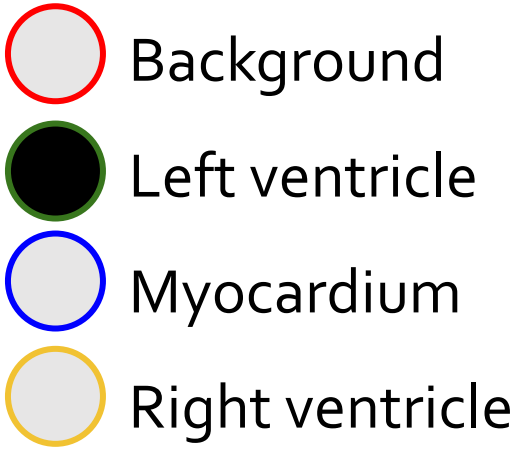
Sliding window

Combination of thousands of image classification tasks

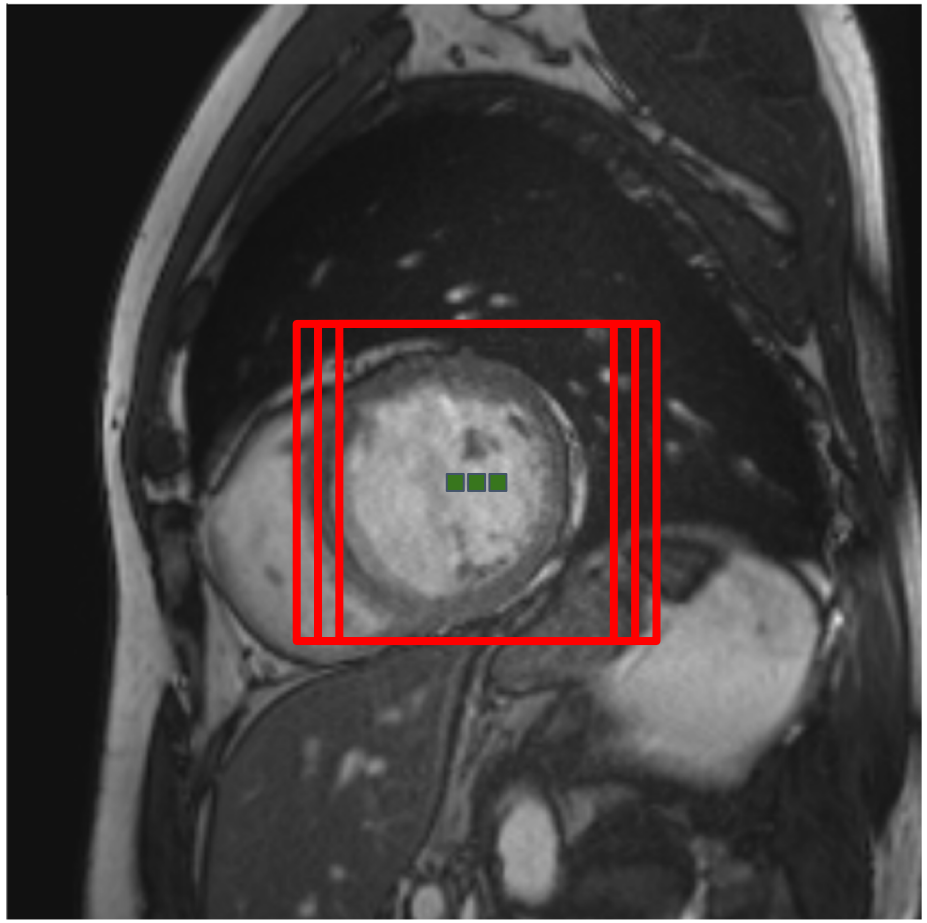


- **Image** = patch centered at voxel
- **Label** = class of center voxel





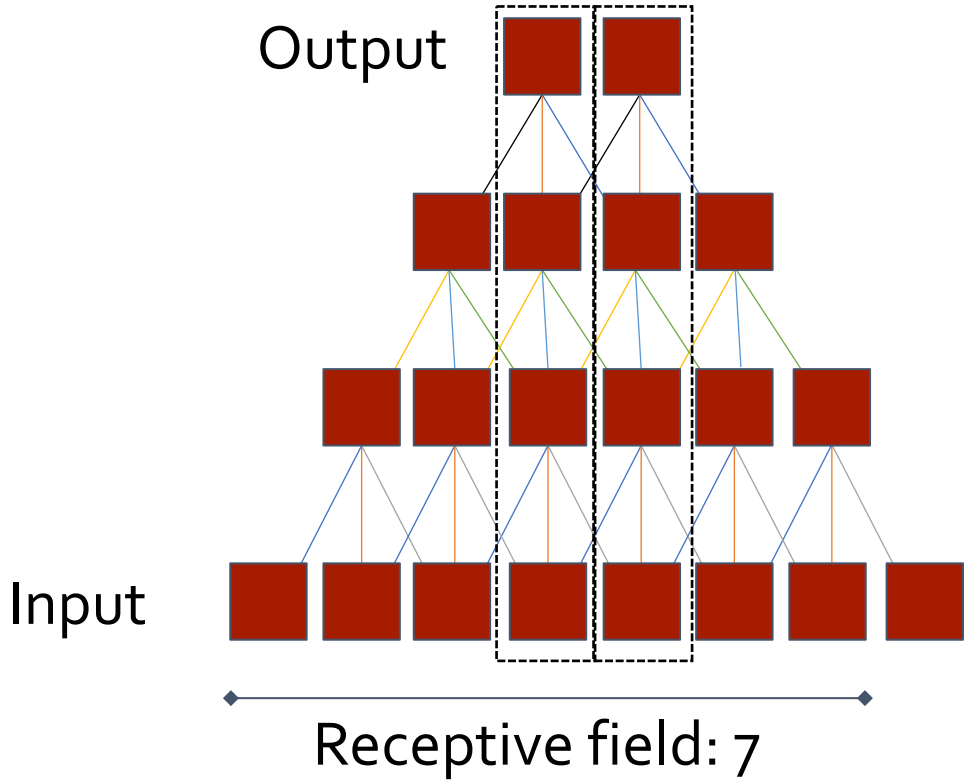
Sliding window



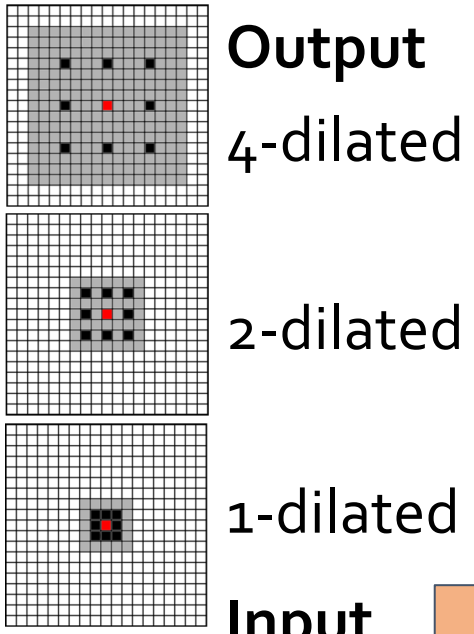
Sliding window approaches are inefficient

- Each patch is processed separately
- Lots of redundant operations
- We would like to re-use/share operations

All-convolutional network



Dilated convolutions



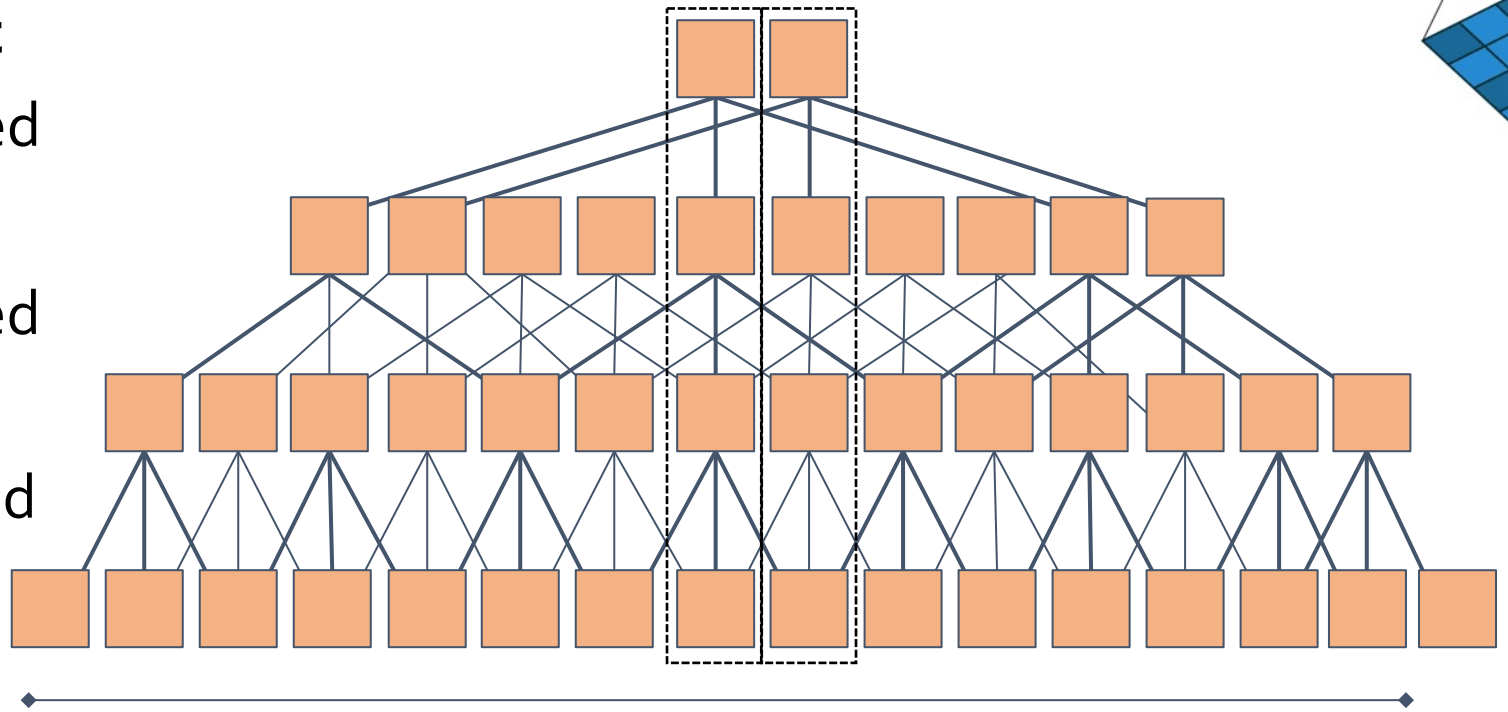
Output

4-dilated

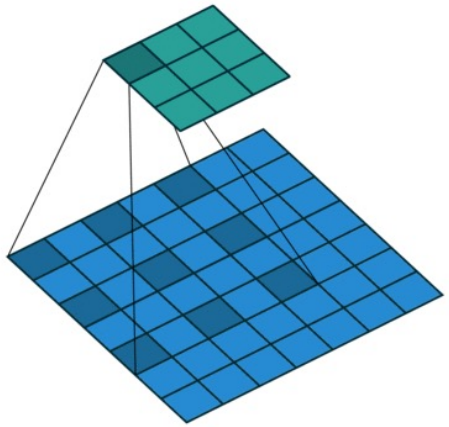
2-dilated

1-dilated

Input

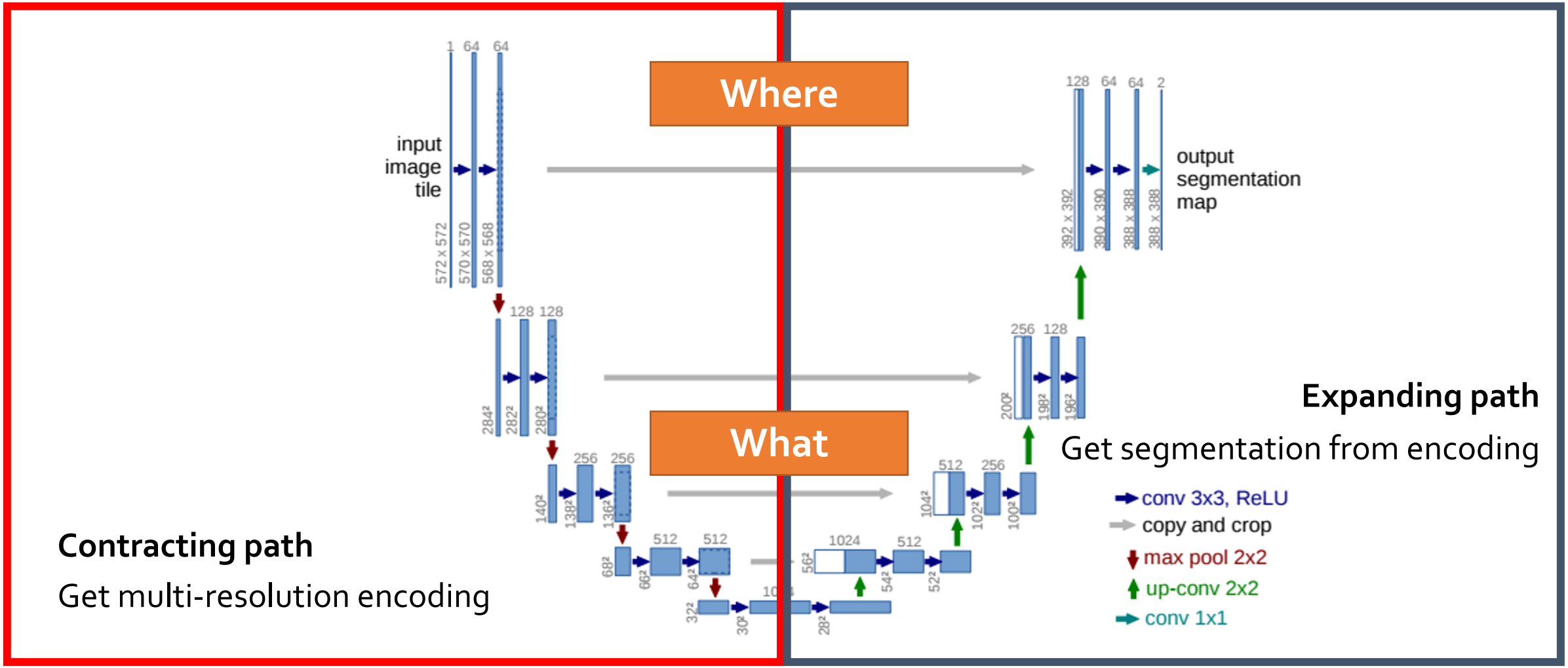


Receptive field: 15



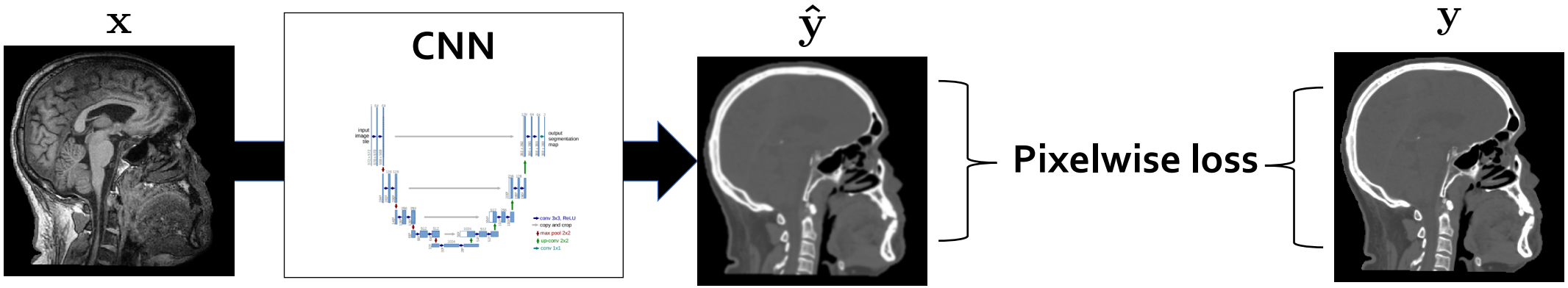
Yu, F., & Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122.

Encoder-decoder architecture: U-Net



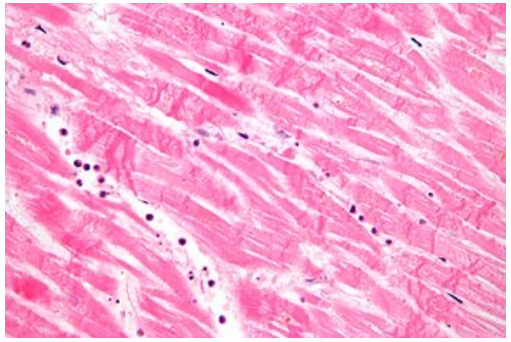
Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In MICCAI (pp. 234-241). Springer, Cham.

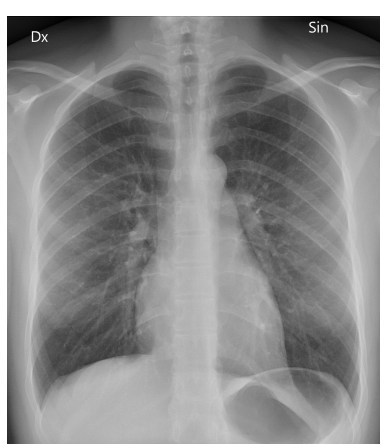
Training



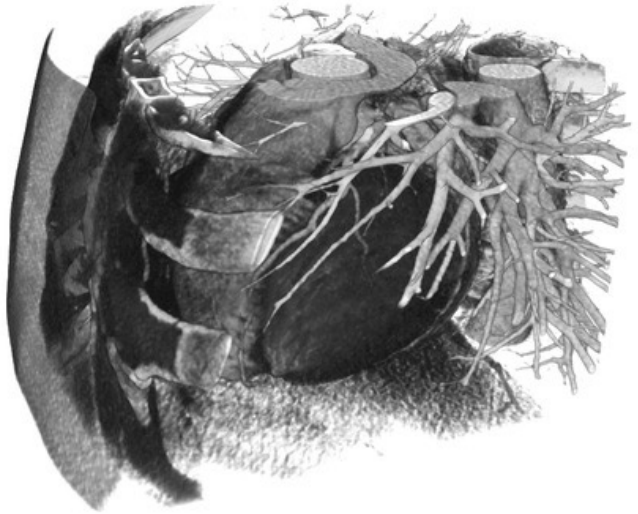
2D or 3D images

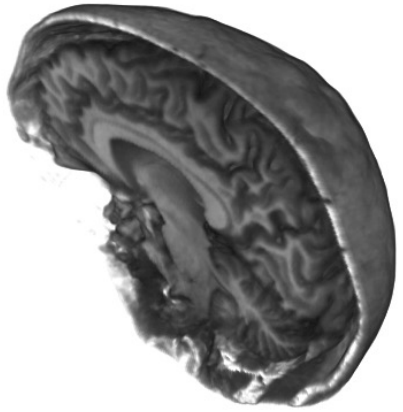
2D data





3D data





https://en.wikipedia.org/wiki/Histopathology
https://en.wikipedia.org/wiki/Fundus_photography
https://en.wikipedia.org/wiki/Chest_radiograph

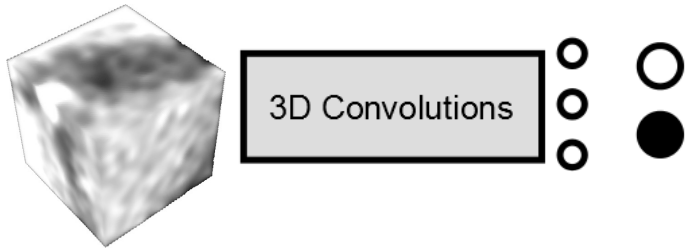
3D networks

Many medical images are 3D instead of 2D

- MR images
- CT images

Can we just use 3D layers instead of 2D layers? Sure!

- 3D convolution layers in Keras, TensorFlow, PyTorch, etc.
- 3D network architectures (e.g. U-Net, V-Net)



But

- Is your data really 3D (think about acquisition)? Isotropy?
- Increase in memory consumption + operations + parameters

Summary

Advanced architectures

- AlexNet, GoogleNet, VGG-Net, ResNet
- Deeper, larger, better + some tricks

Recurrent neural networks

- RNNs + LSTMs

Per image prediction != per voxel prediction

- All-convolutional networks
- Encoder-decoder architectures

2D/3D data

- Most 2D neural networks extend to 3D

Learning goals for today

Have a good overview of neural networks architectures for (medical) image analysis

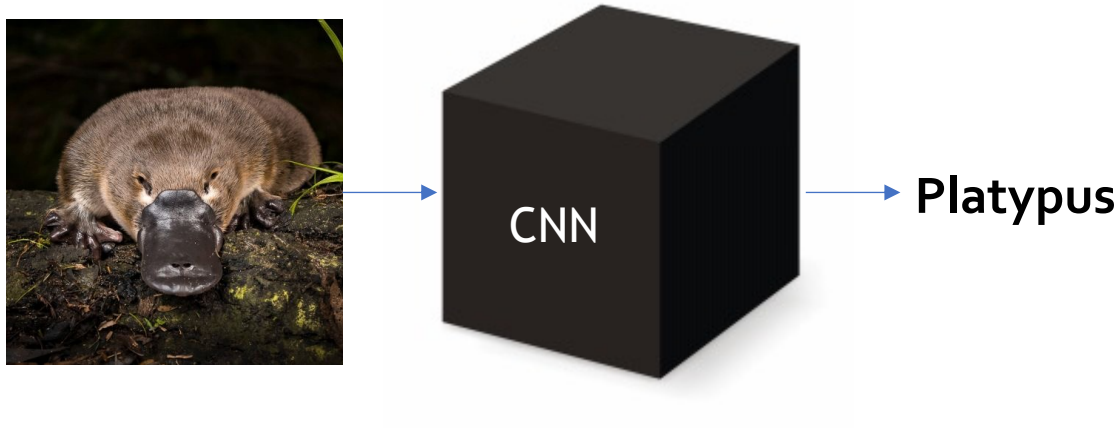
- Including architectures for sequential data

Obtain a general understanding of interpretability of deep learning models

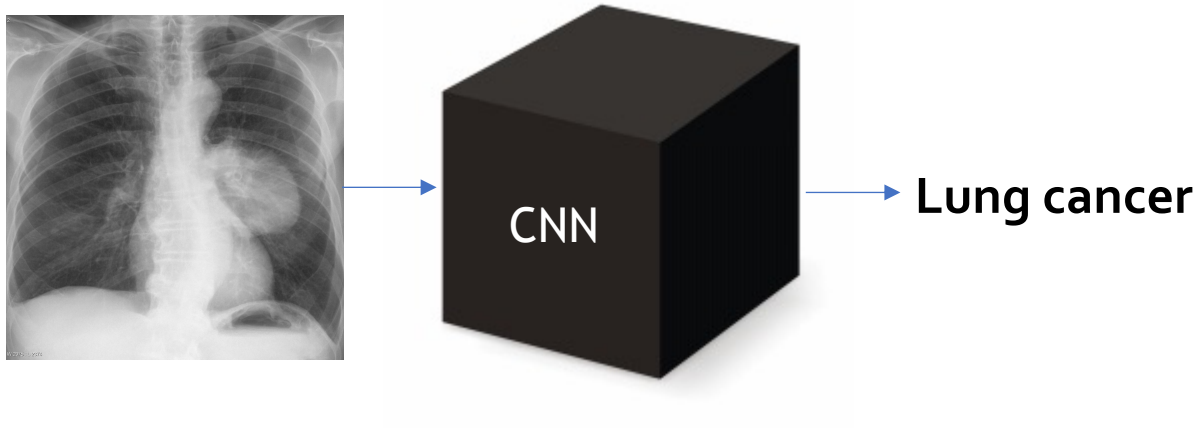
Obtain a general understanding of generative adversarial networks

- Only one example of generative models, but arguably the most successful one

Machine learning as a black box



Machine learning as a black box



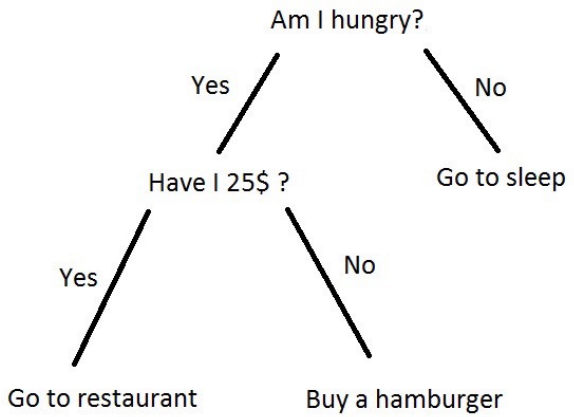
Interpretability: Can we predict what the model will do if the input changes?

Explainability: Do we know exactly what the model is doing?

Uncertainty: How sure is the network of its prediction?

Important questions for ethical use and liability

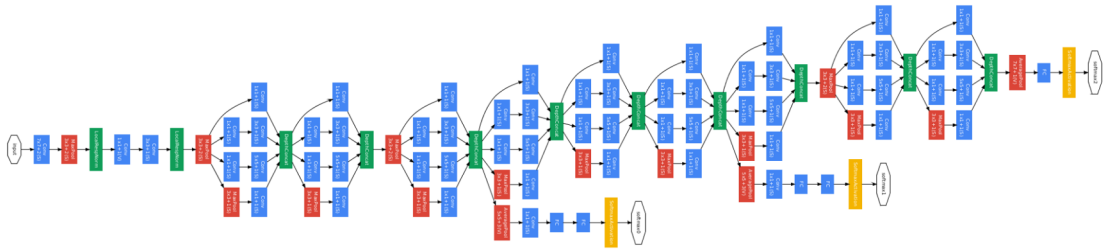
Complexity/performance trade-off



Decision tree

Explainable/interpretable

Performance



CNN

Explainable/interpretable

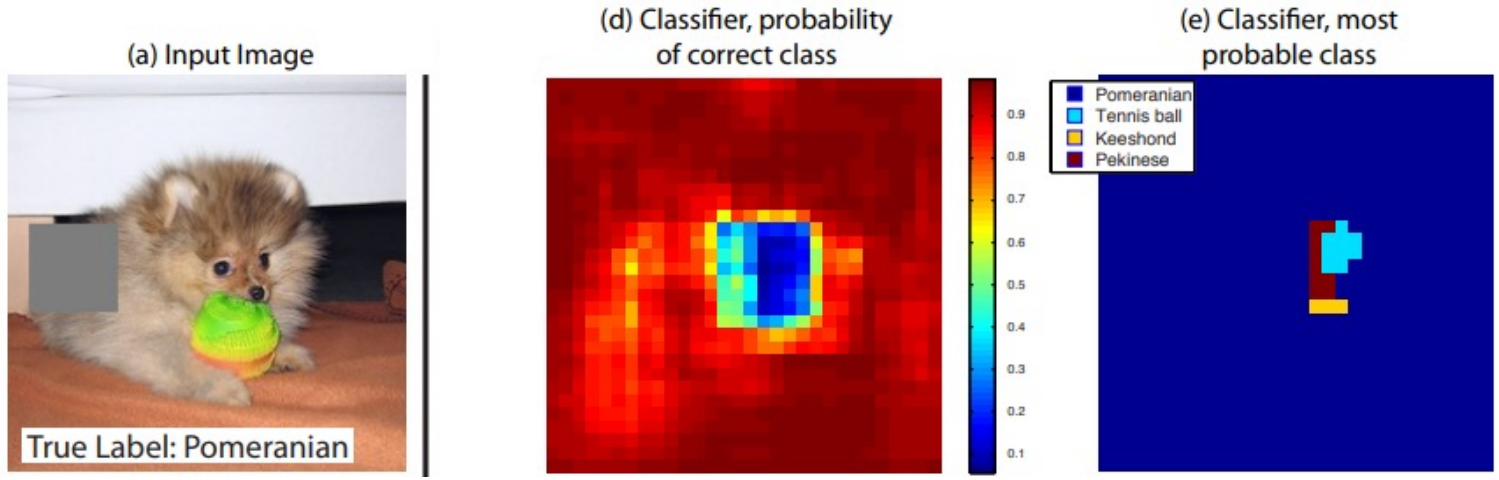
Performance

Interpreting CNNs: saliency maps

- Millions of parameters, impossible to know in detail what each parameter does
- Substitute: what part of the input is associated with a CNN feature or output?
- Popular approaches
  - Occlusion
  - Deconvolution
  - Class activation mapping

Occlusion

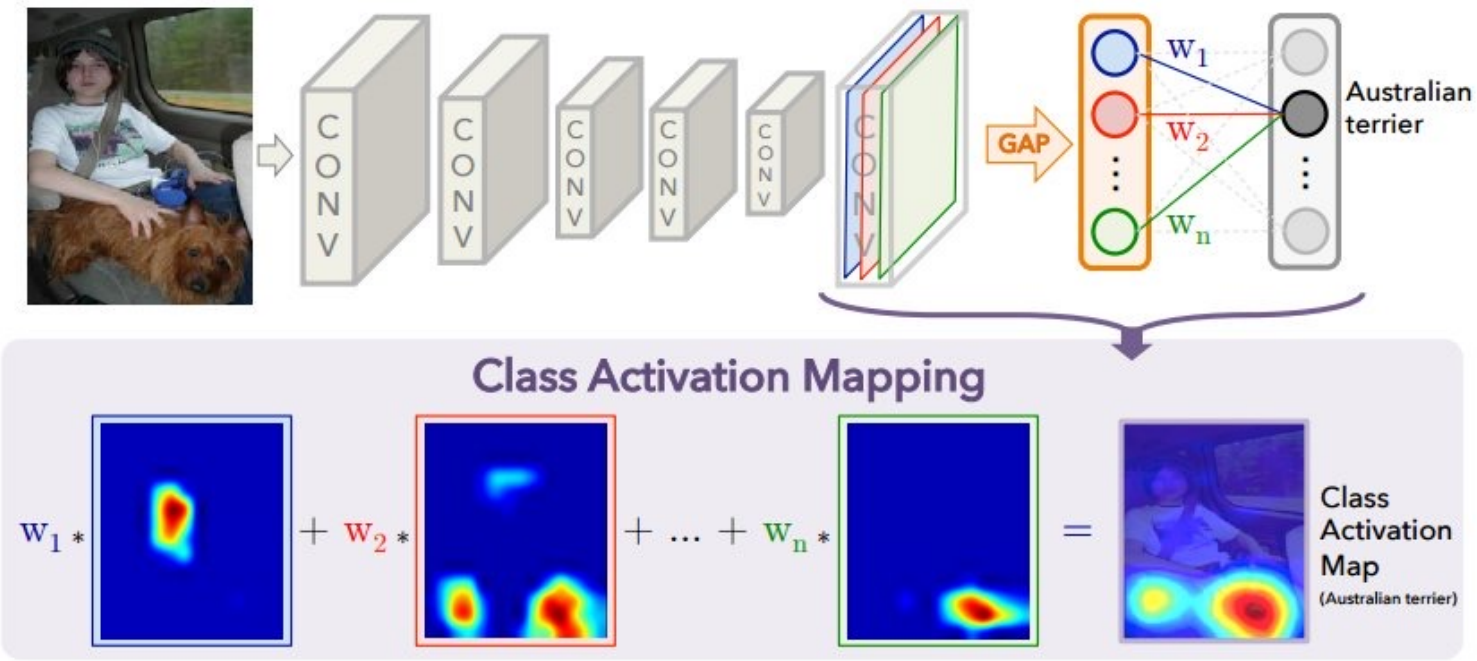
1. Move grey square over image (a)
2. See how classifier output for correct class changes (d)



Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In ECCV (pp. 818-833). Springer, Cham.

Class activation maps (CAM)

- Global average pooling: simply compute average of whole feature map
- Feature maps represent heat maps, weighting gives evidence

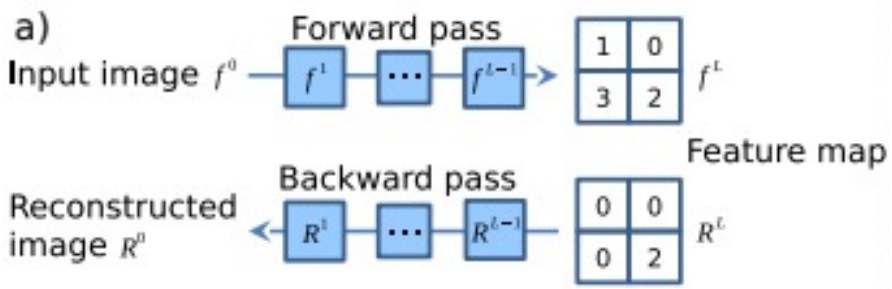


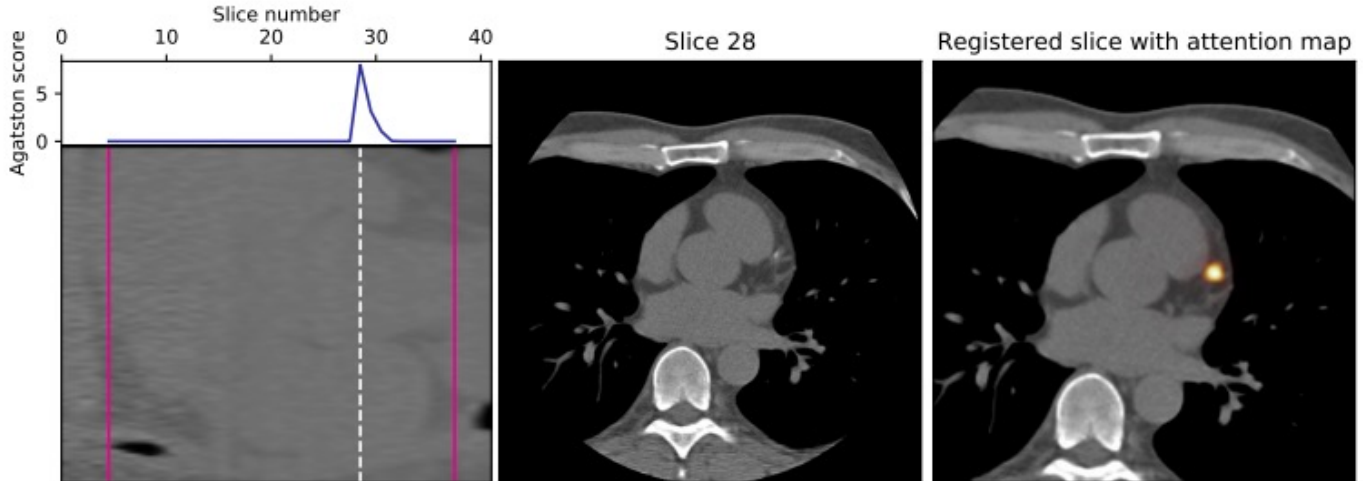
Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016). Learning deep features for discriminative localization. CVPR (pp. 2921-2929).

Deconvolution

Inversion: find input corresponding to a particular feature

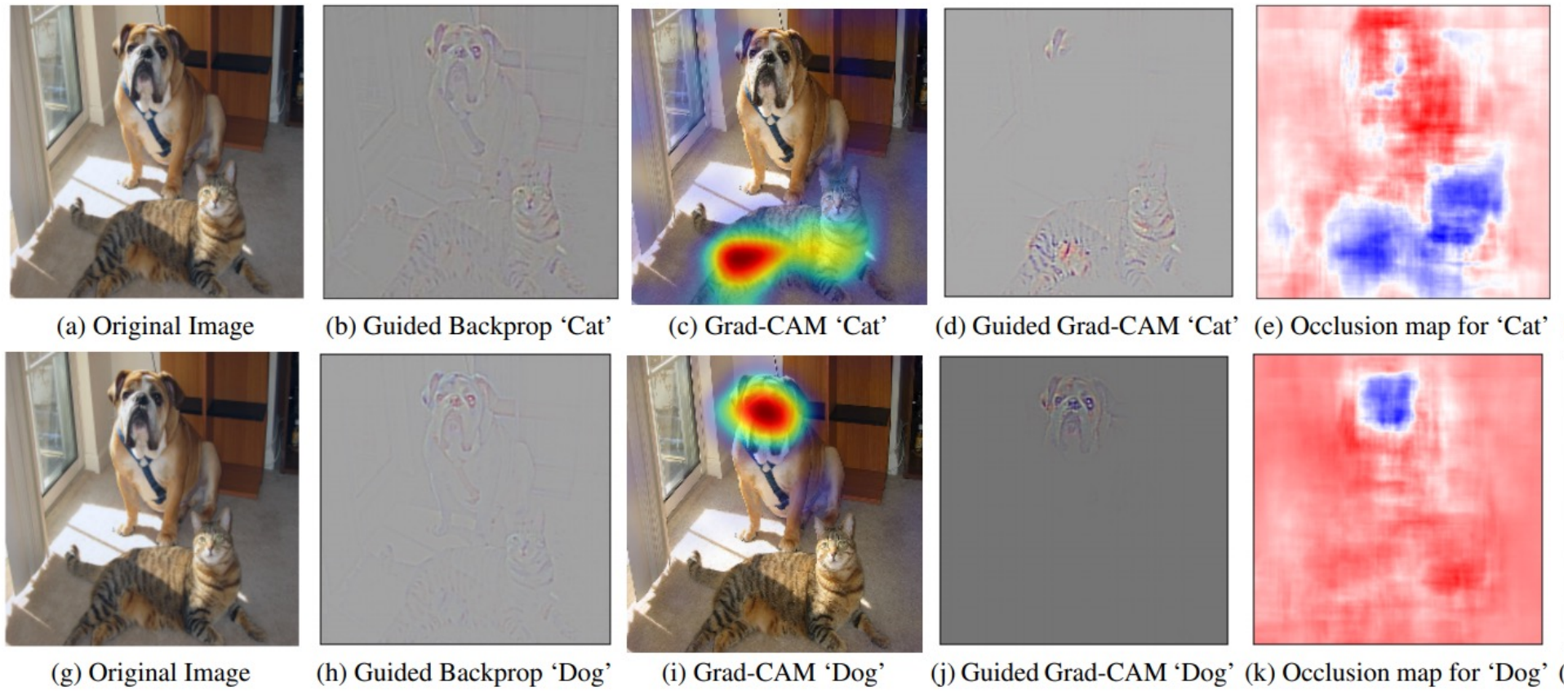
1. Push input through CNN
2. Fix value for one 'neuron', set all others to zero
3. Perform inverse of convolution, pooling, activation functions





Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In ECCV (pp. 818-833). Springer, Cham.
de Vos, B. D., Wolterink, J. M., Leiner, T., de Jong, P. A., Lessmann, N., & Išgum, I. (2019). Direct automatic coronary calcium scoring in cardiac and chest CT. IEEE transactions on medical imaging.
Springenberg, J. T., Dosovitskiy, A., Brox, T., & Riedmiller, M. (2014). Striving for simplicity: The all convolutional net. arXiv preprint arXiv:1412.6806.

Comparison



Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In ICCV (pp. 618-626).

Learning goals for today

Have a good overview of neural networks architectures for (medical) image analysis

- Including architectures for sequential data

Obtain a general understanding of interpretability of deep learning models

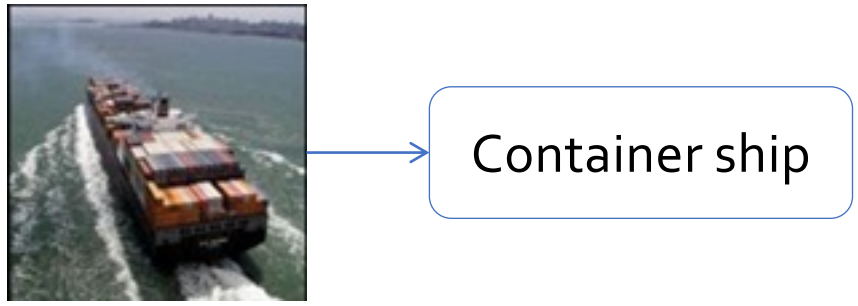
Obtain a general understanding of generative adversarial networks

- Only one example of generative models, but arguably the most successful one

Discriminative vs. generative models

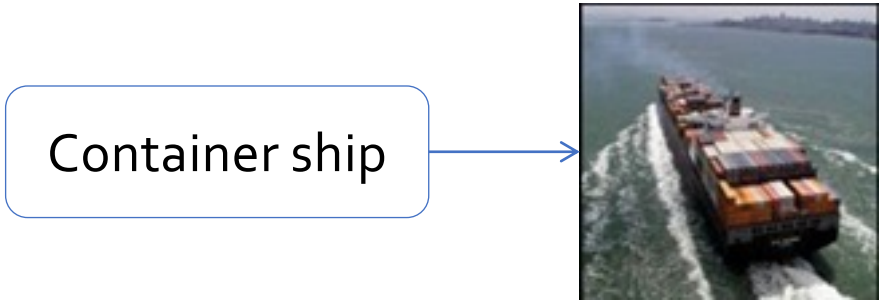
Discriminative models

- Go from high-dimensional sample to low-dimensional prediction
- E.g. from an image to the label of that image



Generative models

- Go from low-dimensional input to high-dimensional sample
- E.g. from an image label to an image sample



But why would we do this?

- **Unsupervised** learning: we don’t need labels for our training data
- Learning more about your data set
- Synthesizing new data for discriminative models
- Advanced image manipulation

Generative adversarial networks (GANs)

Instead of **one** neural network, we train **two** neural networks

1. A **generator** network generates samples of images
2. A **discriminator** network distinguishes generated samples from real samples

The **generator** network tries to fool the discriminator

Goodfellow et al. *Generative adversarial nets*, NIPS 2014

Generative adversarial networks (GANs)

1. A **generator** network generates samples of images
2. A **discriminator** network distinguishes generated samples from real samples

Generator
Counterfeiter

Discriminator
Detective

Goodfellow et al. *Generative adversarial nets*, NIPS 2014

Generative adversarial networks (GANs)

1. A **generator** network generates samples of images
2. A **discriminator** network distinguishes generated samples from real samples

Generator
Counterfeiter

Discriminator
Detective

Generative adversarial networks (GANs)

1. A **generator** network generates samples of images
2. A **discriminator** network distinguishes generated samples from real samples

Generator
Counterfeiter

Discriminator
Detective

Generative adversarial networks (GANs)

1. A **generator** network generates samples of images
2. A **discriminator** network distinguishes generated samples from real samples

Generator
Counterfeiter

Discriminator
Detective

Goodfellow et al. *Generative adversarial nets*, NIPS 2014

Generative adversarial networks (GANs)

A bit more formally…

We are given samples $x_1, \ldots, x_n$ from a data distribution p_{data}

We want to generate new samples similar to p_{data}

We can sample from p_{data} efficiently – examples are available

Goodfellow et al. *Generative adversarial nets*, NIPS 2014

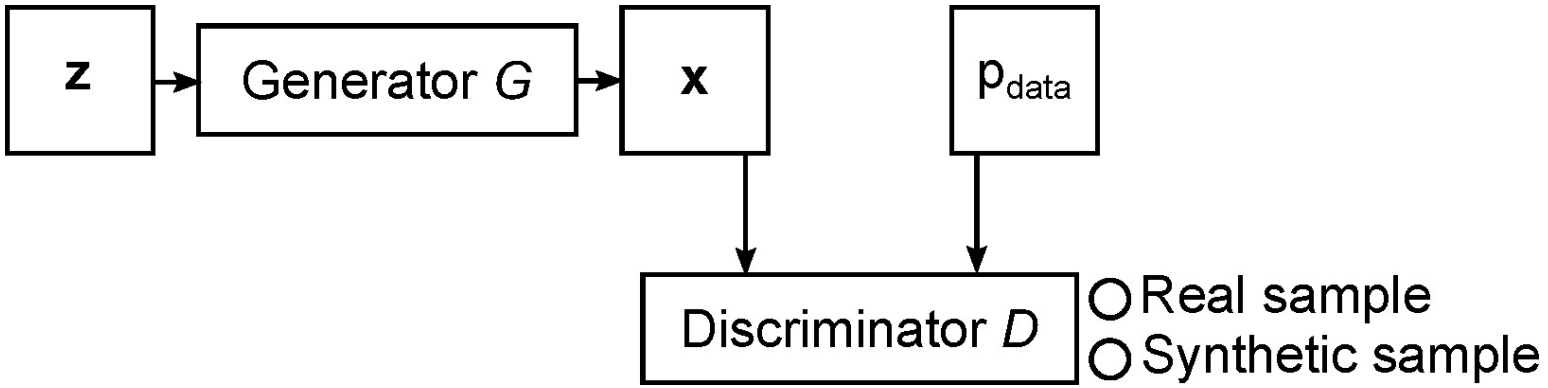
Generative adversarial networks (GANs)

Aim: Find a sample distribution p_{gen} that closely resembles p_{data}

Generator *G* maps points from a normal/uniform distribution *z* to samples *x* in p_{gen}

Discriminator *D* tells samples from p_{data} and p_{gen} apart

Generators *G* and *D* are networks with trainable parameters ϑ_G and ϑ_D



Goodfellow et al. *Generative adversarial nets*, NIPS 2014

Generative adversarial networks (GANs)

Discriminator *D* maximizes an objective function by assigning a high probability to real samples and a low probability to synthetic samples

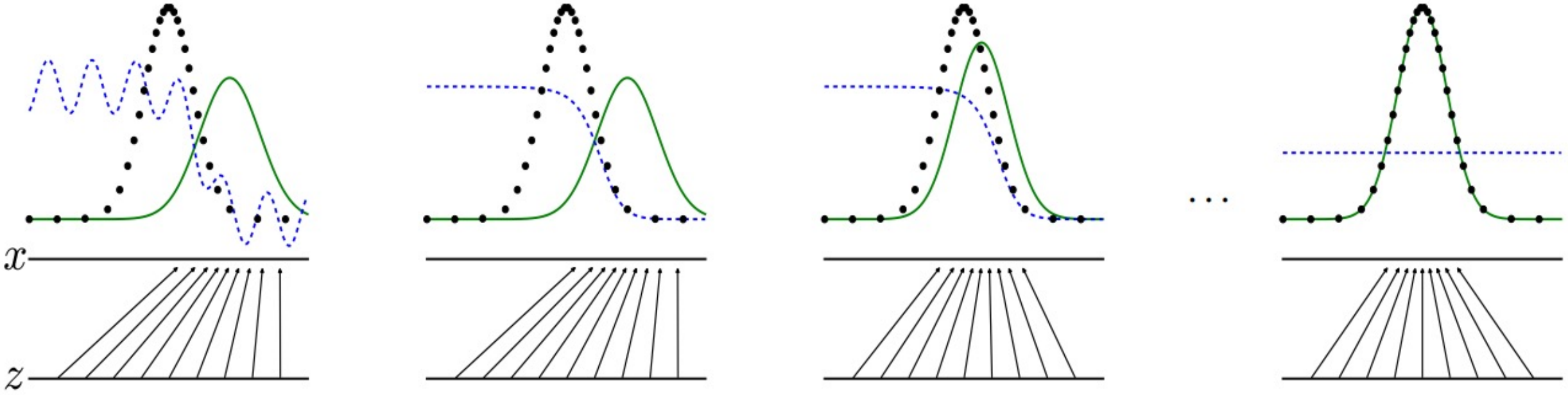
$$V^{(D)}(D,G) = \mathop{\mathbb{E}}_{x \sim p_{data}} [\log D(x)] + \mathop{\mathbb{E}}_{z \sim p_z} [\log (1 - D(G(z)))]$$

At the same time, *G* tries to minimize this objective function by generating samples that have a high probability of being real

$$V^{(G)}(D,G) = \mathop{\mathbb{E}}_{z \sim p_z} [\log (1 - D(G(z)))]$$

Goodfellow et al. *Generative adversarial nets*, NIPS 2014

GANs in theory



Start

Convergence

p_{real} p_{gen} $D(x)$

Demo: https://poloclub.github.io/ganlab/

Goodfellow et al. *Generative adversarial nets*, NIPS 2014

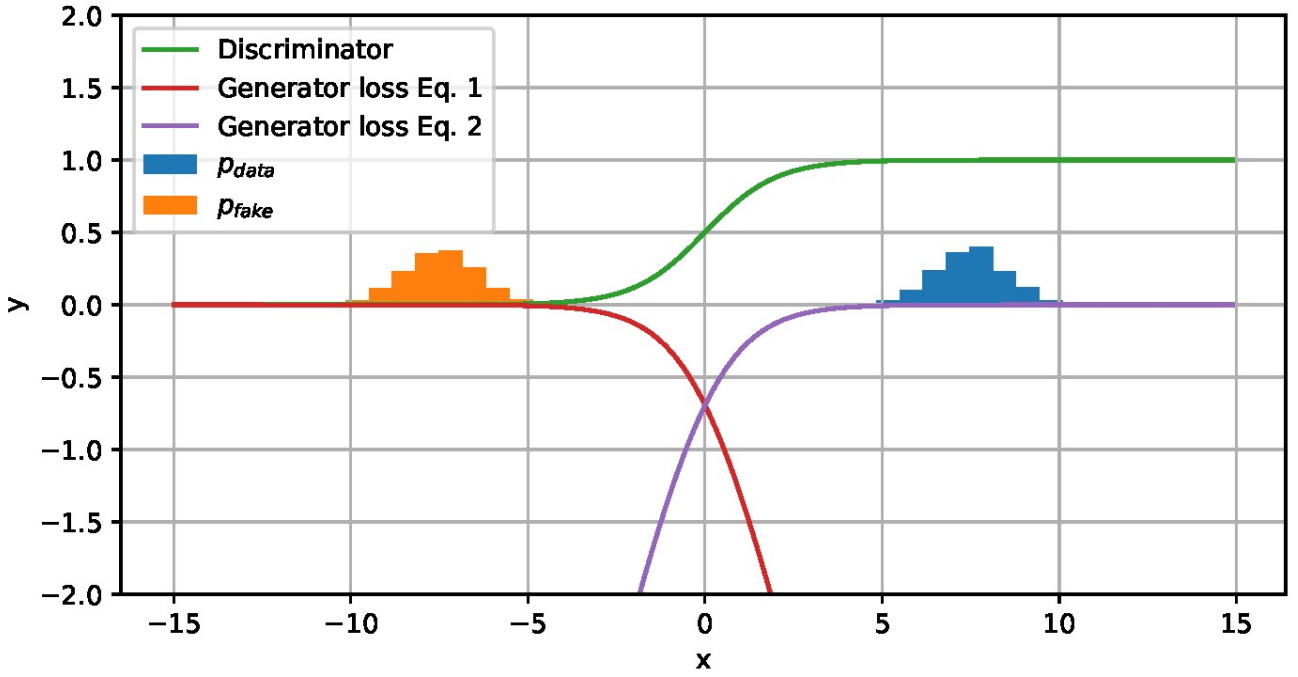
Challenges when training GANs

Poor gradients from discriminator *D* to generator *G*

Especially when the discriminator *D* can easily distinguish fake and real samples

1) $V^{(G)}(D,G) = \underset{z \sim p_z}{\mathbb{E}} [\log (1 - D(G(z)))]$

2) $V^{(G)}(D,G) = - \underset{z \sim p_z}{\mathbb{E}} [\log (D(G(z)))]$



Challenges when training GANs

1.Poor gradients from discriminator *D* to generator *G*

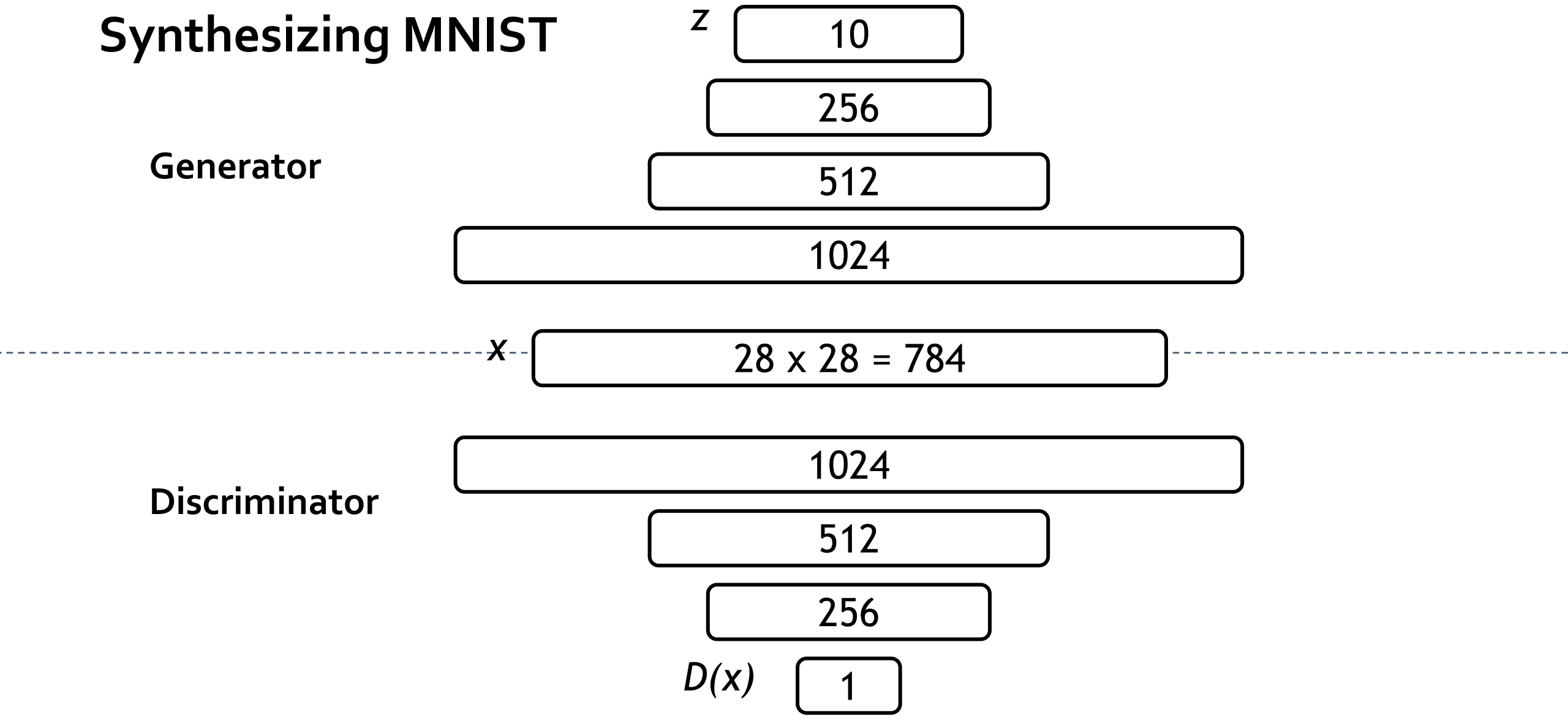
- Especially when the discriminator *D* can easily distinguish fake and real samples

2.Mode collapse

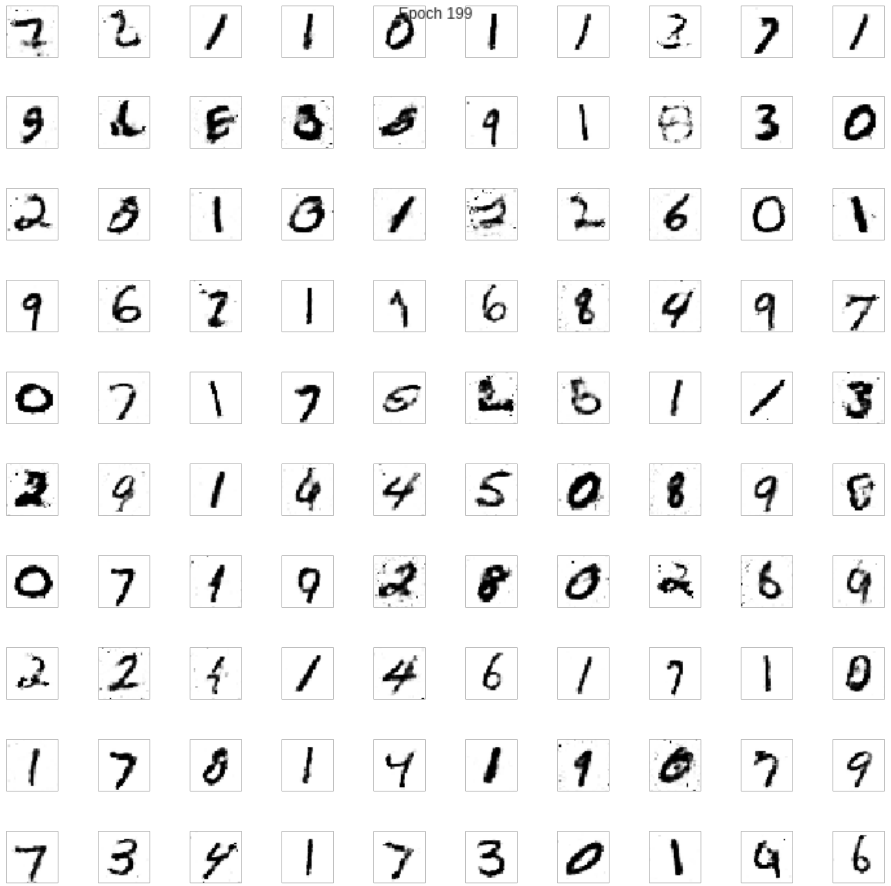
- Generator *G* maps all noise vectors to similar samples *G(z)*
- E.g. generator only synthesizes 1 in MNIST
- Generator and discriminator chase each other from mode to mode

3.Unclear when training is finished

Synthesizing MNIST

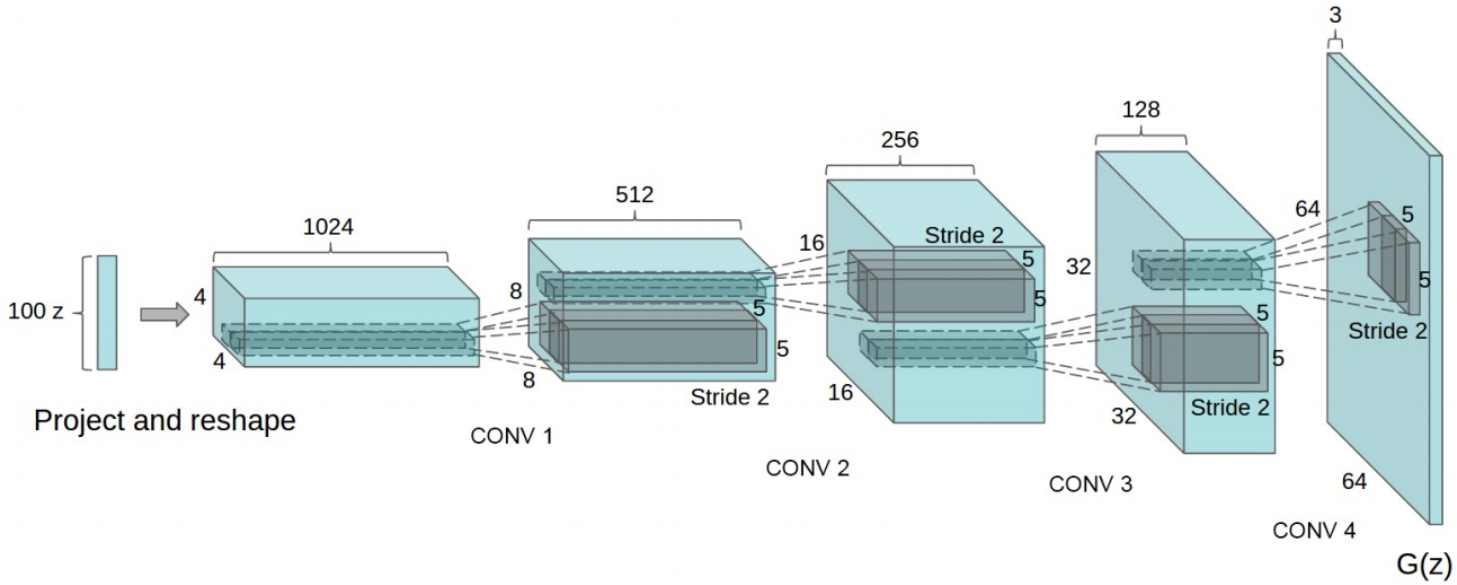


Synthesizing MNIST



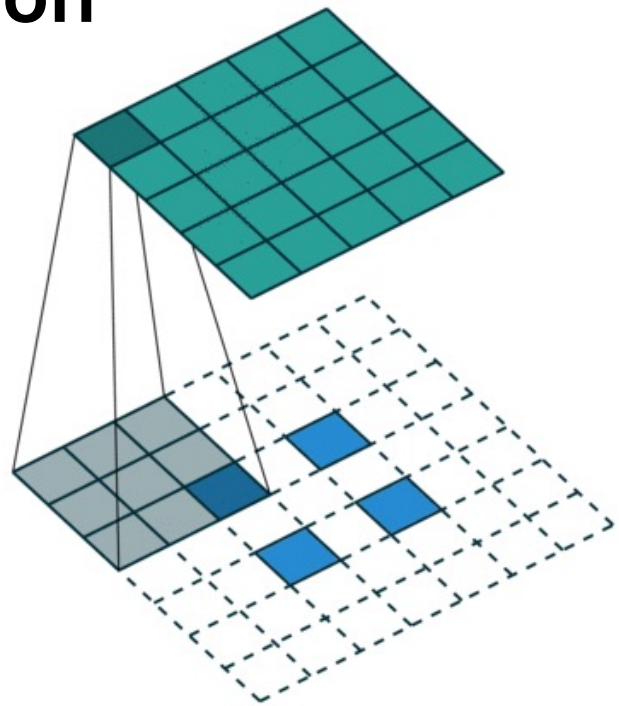
Synthesizing images

- Image synthesis requires convolutional generators and discriminators
- We know how to reduce the image size
- Now we need some way to increase the image size



Radford, A., Metz, L., & Chintala, S. (2015). arXiv preprint arXiv:1511.06434.

Transposed convolution



- Deconvolution
- Fractionally-strided convolution
- Transposed convolution

http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html

2014

2015

2016

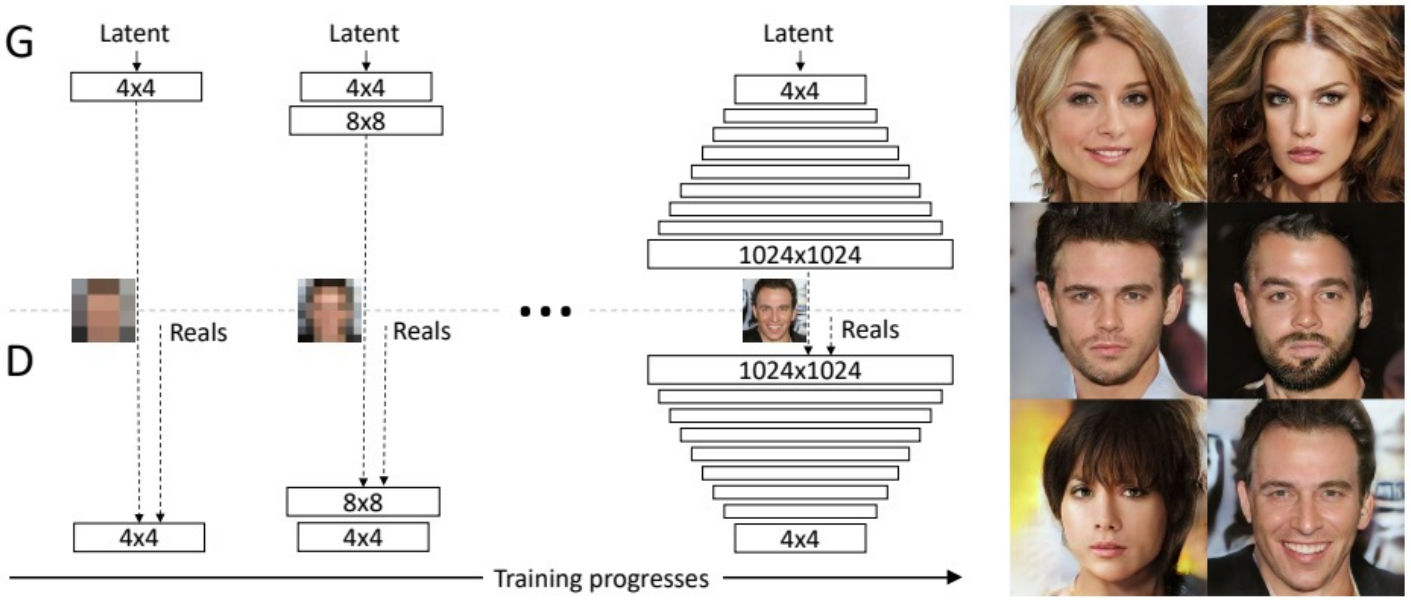
2017

2018

https://twitter.com/goodfellow_ian/status/1084973596236144640?s=19
https://arxiv.org/abs/1406.2661, https://arxiv.org/abs/1511.06434, https://arxiv.org/abs/1606.07536, https://arxiv.org/abs/1710.10196, https://arxiv.org/abs/1812.04948

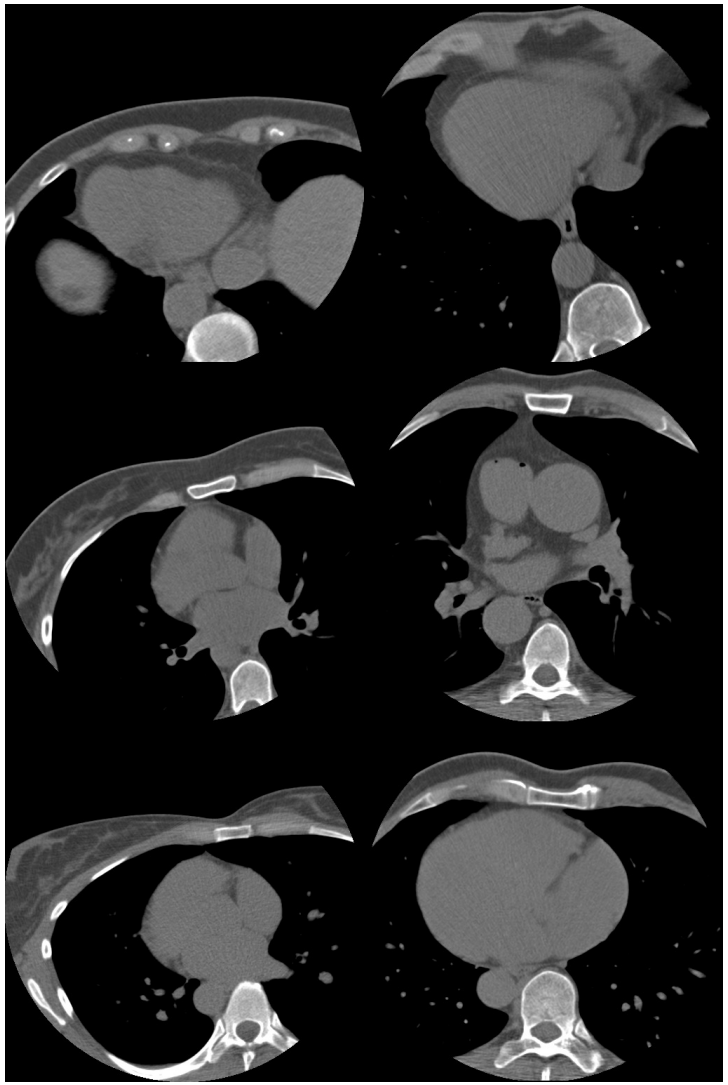
Progressive GAN

- Goal: high-resolution image synthesis (1024 x 1024 pixels)
- Problem: optimizing large and deep networks can lead to stability issues
- Trick: periodically increase resolution in *G* and *D*

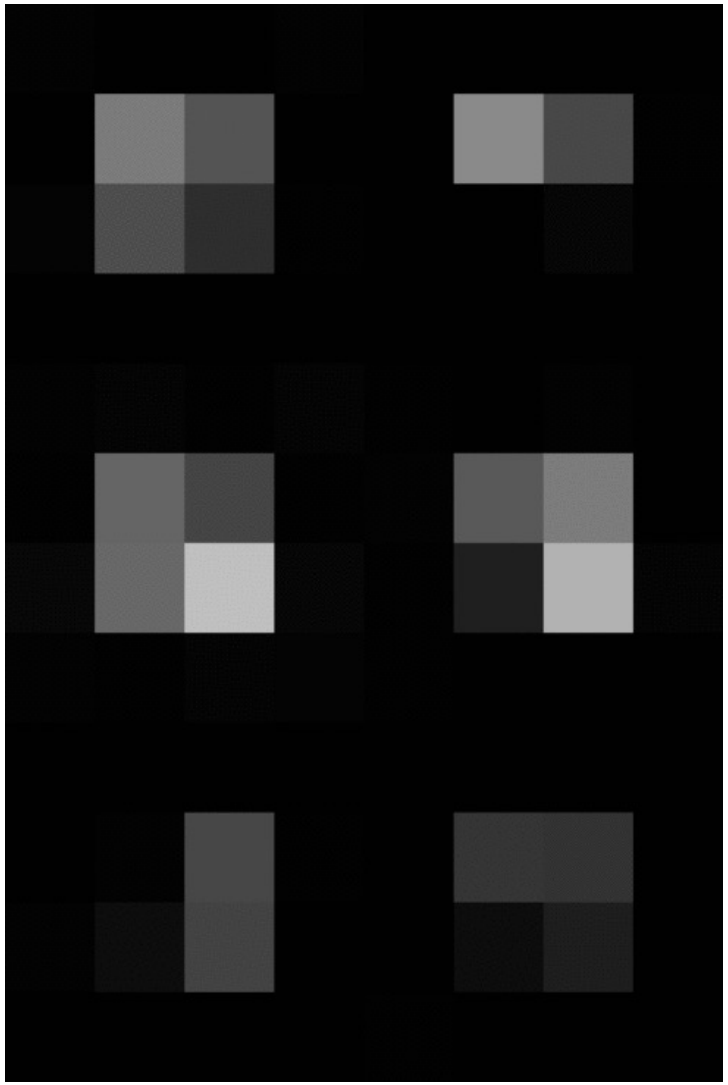


Karras et al. *Progressive Growing of GANs for Improved Quality, Stability, and Variation*, ICLR 2018

Real



Fake



Karras et al. *Progressive Growing of GANs for Improved Quality, Stability, and Variation*, ICLR 2018

StyleGAN



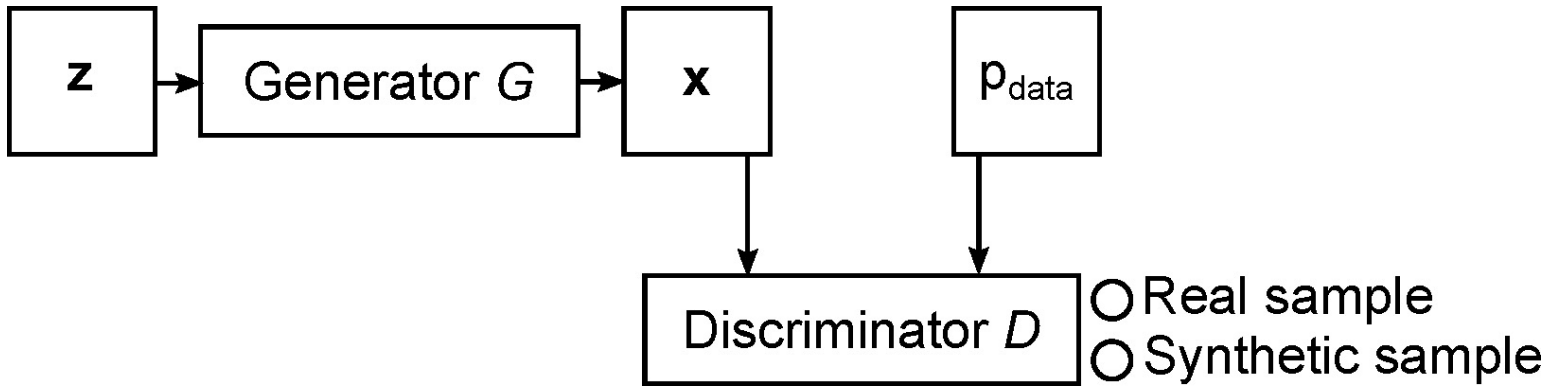


Karras, T., Laine, S., & Aila, T. (2018). A style-based generator architecture for generative adversarial networks. arXiv preprint arXiv:1812.04948.
https://thispersondoesnotexist.com/

Spaces

- GANs can help us learn more about our data
- Different latent space points correspond to different sample space points
- The latent space is structured
- Interpolation in the latent space leads to smooth transitions

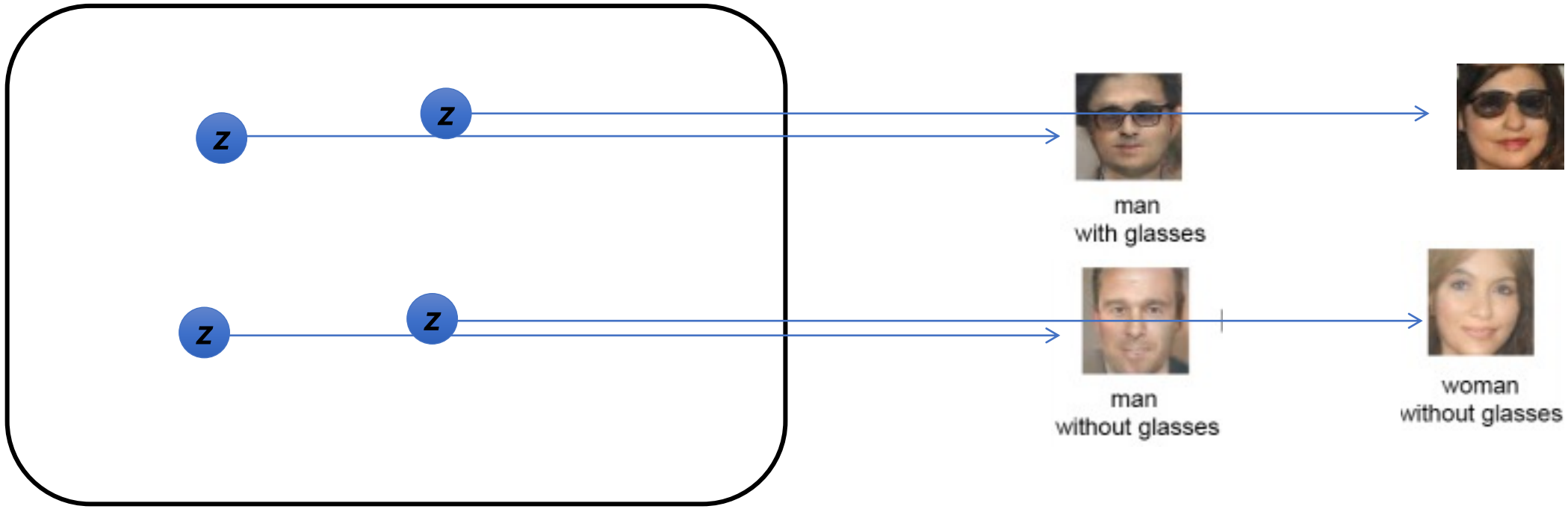
The latent space **The sample space**



Spaces

The latent space

The sample space

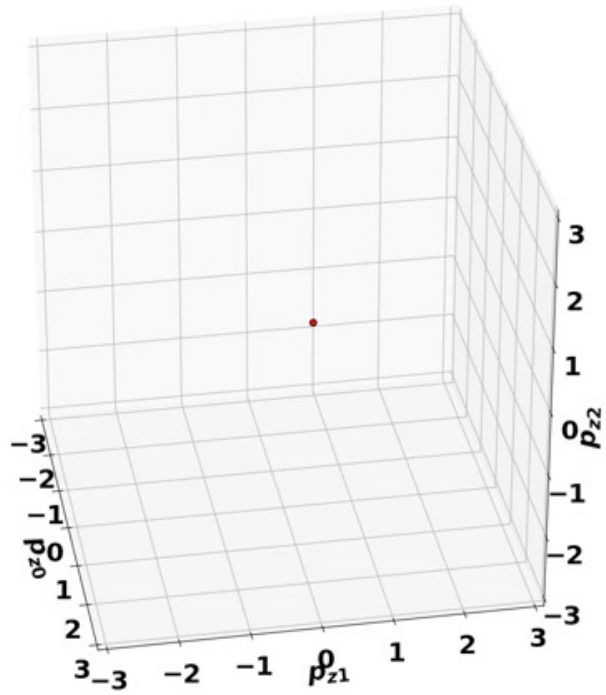


Radford, A., Metz, L., & Chintala, S. (2015). arXiv preprint arXiv:1511.06434.

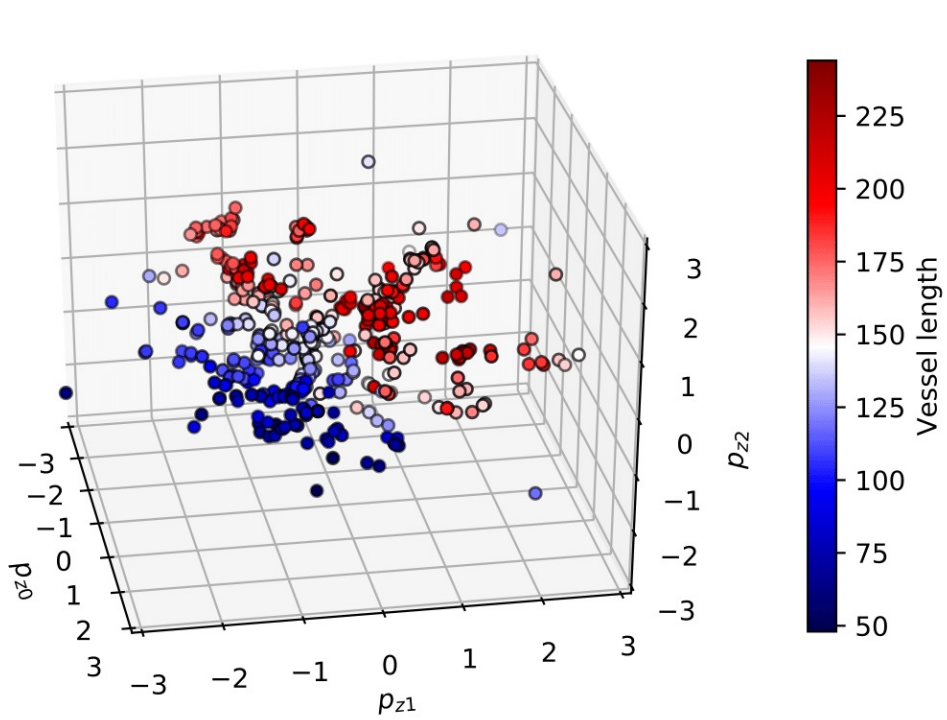
Example: coronary artery synthesis

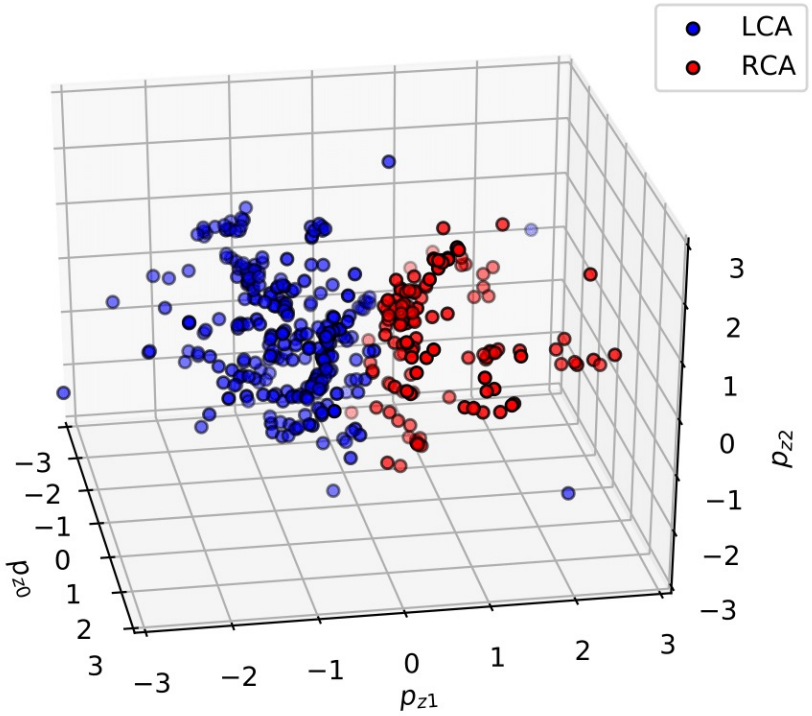
The sample space

The latent space



Example: coronary artery synthesis

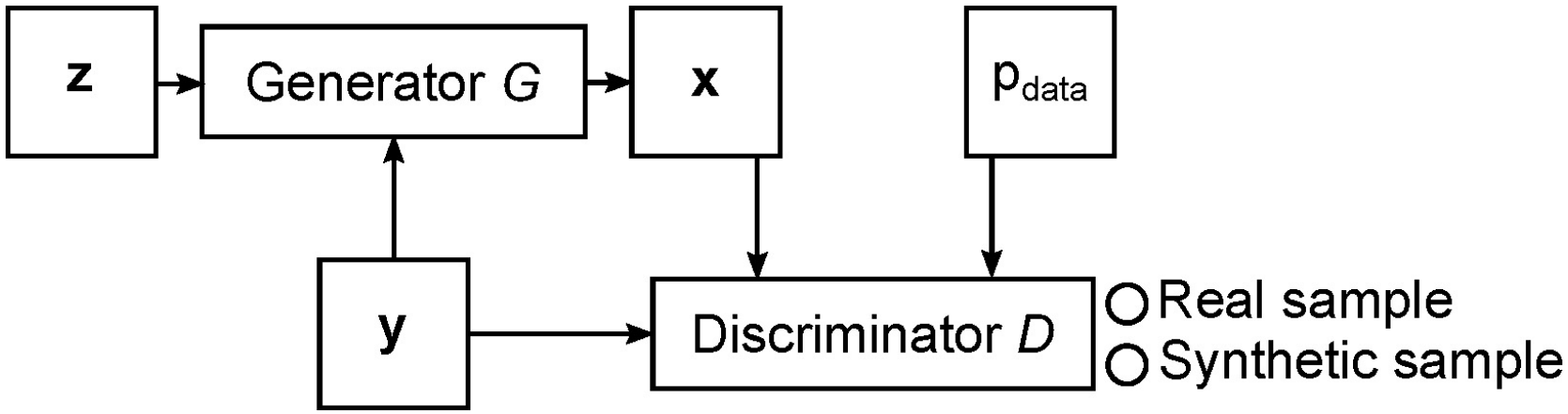




Wolterink, J. M., Leiner, T., & Isgum, I. (2018). Blood Vessel Geometry Synthesis using Generative Adversarial Networks. arXiv preprint arXiv:1804.04381.

Conditional GAN (cGAN)

- GAN: difficult to control what is generated
- Condition GAN on extra input *y*

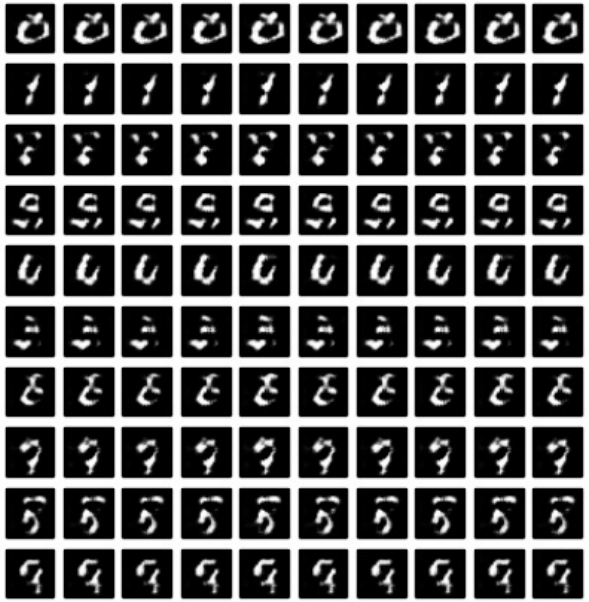


Mirza et al. *Conditional generative adversarial nets*, 2014

Conditional GAN (cGAN)

The generator and discriminator optimize an objective function based on the predictions for real samples and synthetic samples, given extra information

$$\min_G \max_D V(D,G) = \mathop{\mathbb{E}}_{x \sim p_{data}} [\log D(x|y)] + \mathop{\mathbb{E}}_{z \sim p_z} [\log (1 - D(G(z|y)|y))]$$



Epoch 1

Mirza et al. *Conditional generative adversarial nets*, 2014

Remaining slides: self study

BigGANs

CycleGANs

GANs for medical image analysis

+ watch the video lecture from last year given by Jelmer Wolterink

BigGAN

- State of the art in conditional GAN training
- Trained on ImageNet (~million images)
- Get any class you want

Brock, A., Donahue, J., & Simonyan, K. (2018). Large scale gan training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096.

In medical imaging

We typically have some image to condition on, ‘image-to-image’ applications

- Segmentation
- Synthesis
- Image quality enhancement

It’s not always easy to determine a loss function that captures what we want

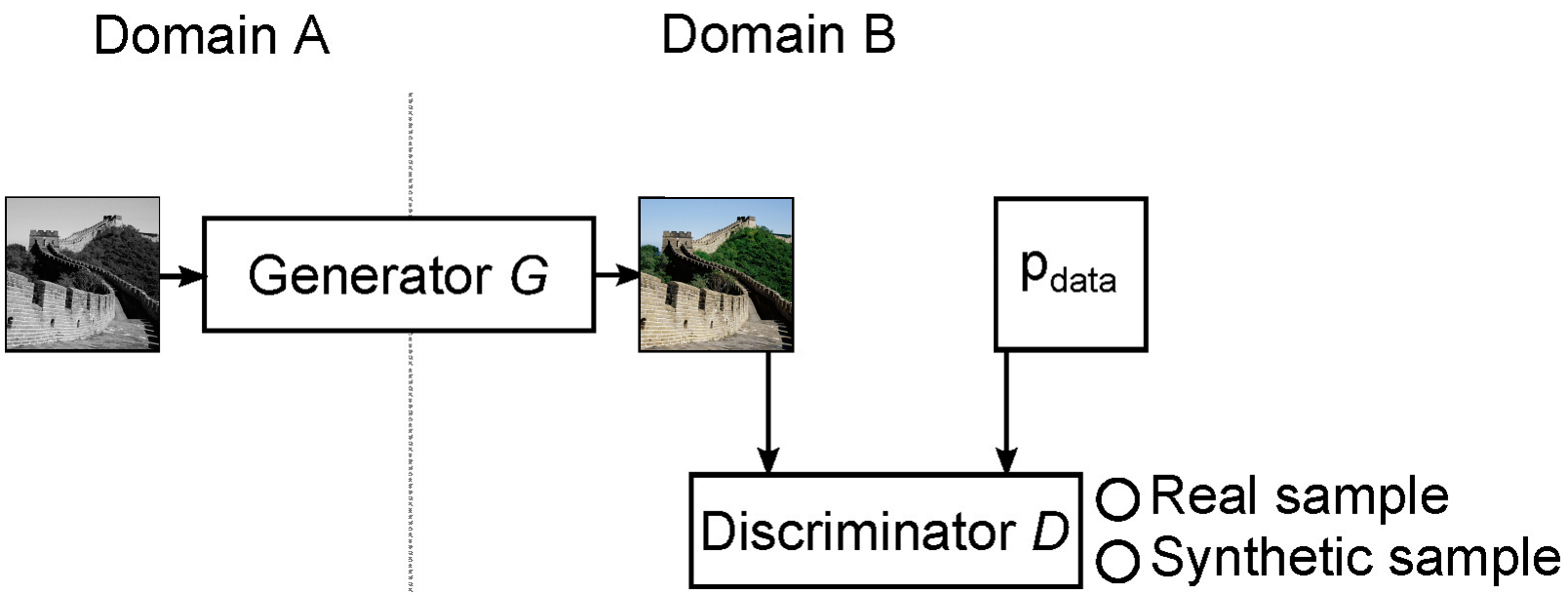
- Noise-free segmentations?
- Sharp images?

Use a conditional GAN where *y* is the input image

Image-to-image translation

Domain A: input image *y*

Domain B: output image *x*

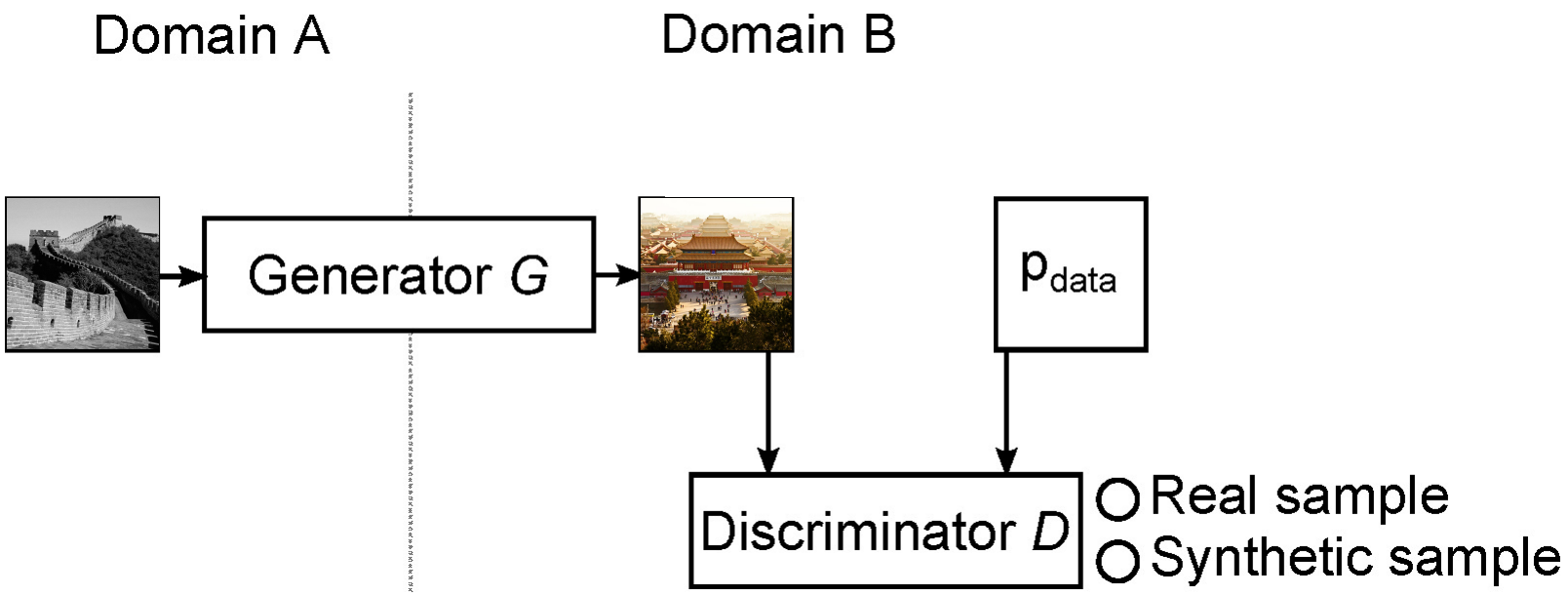


Isola et al. *Image-to-Image Translation with Conditional Adversarial Nets*, CVPR 2017

Image-to-image translation

The output image *x* could be very realistic but 'wrong'

Regularize to make sure that it corresponds to the input image *y*



Isola et al. *Image-to-Image Translation with Conditional Adversarial Nets*, CVPR 2017

Paired vs. unpaired training data

Paired

Domain A

Domain B

Unpaired

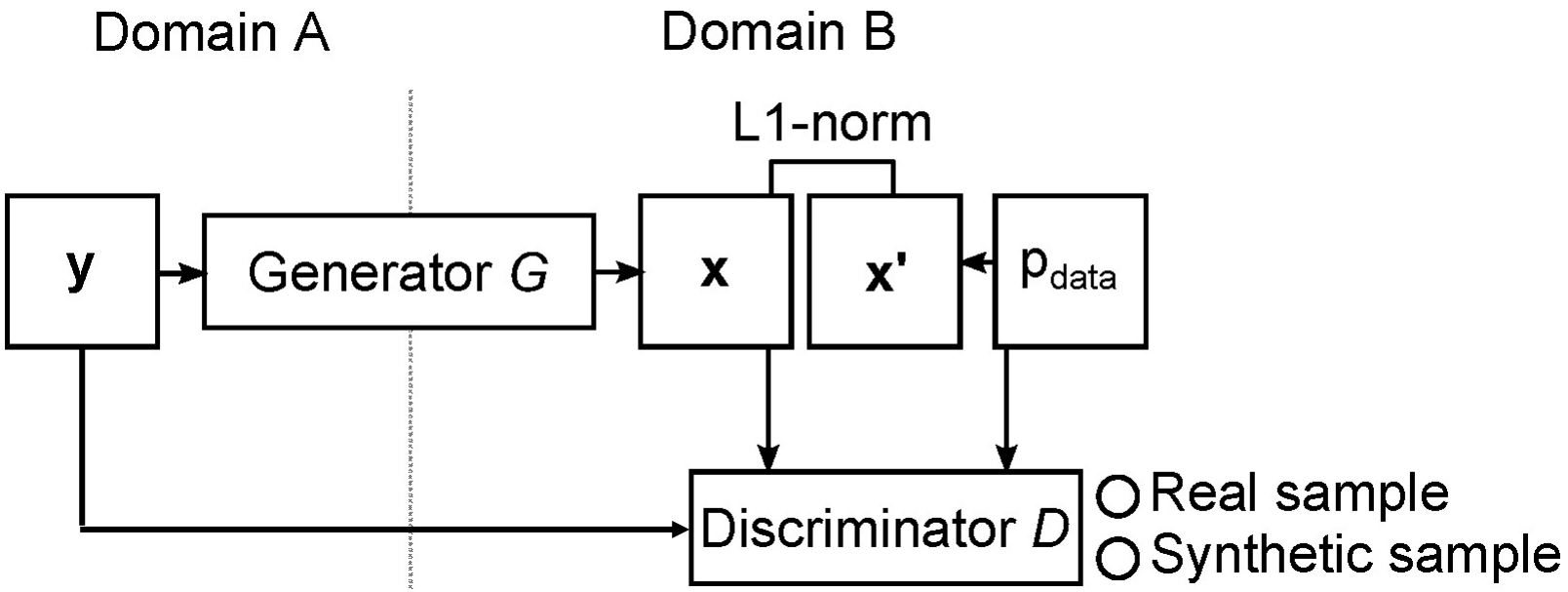
Domain A

Domain B

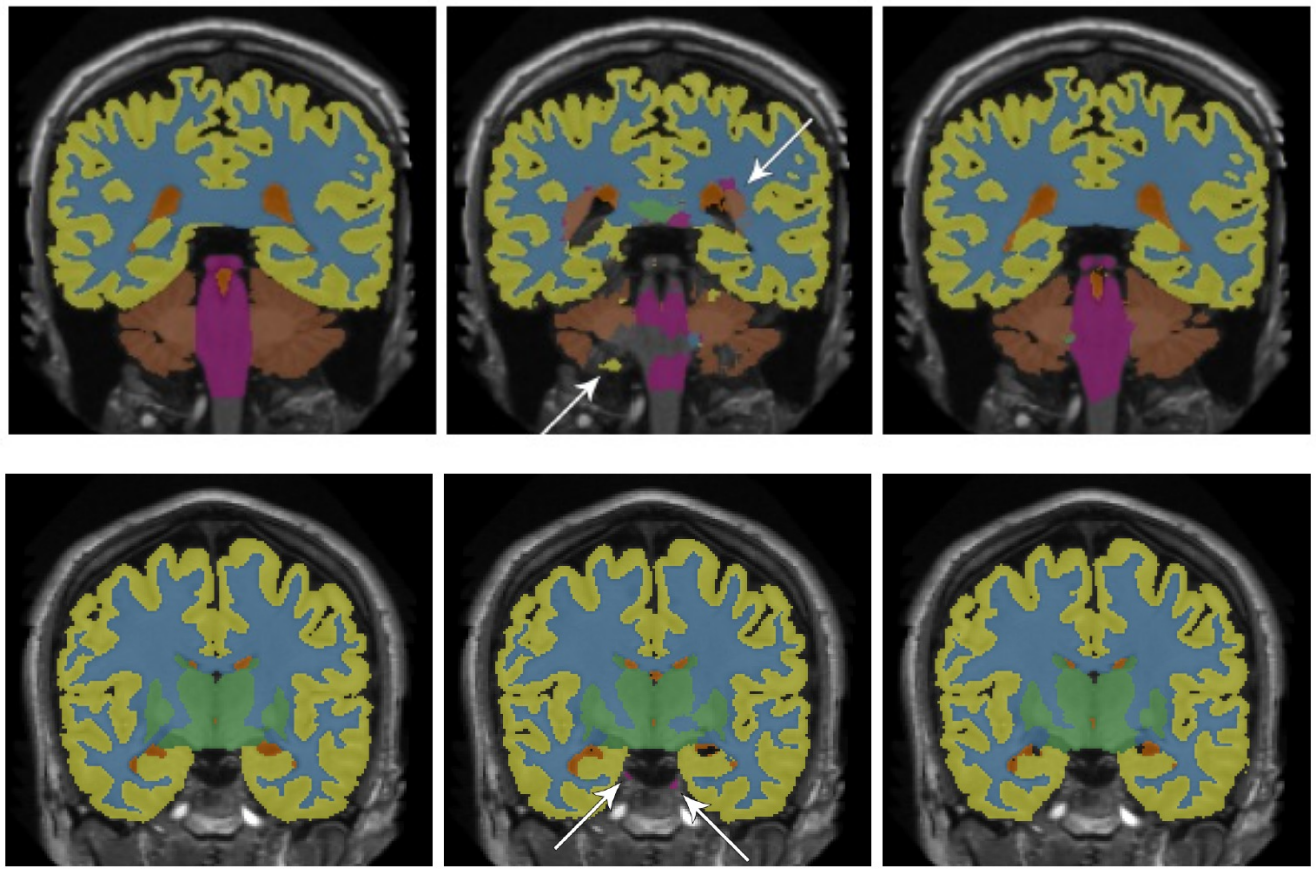
Paired image-to-image translation

The discriminator and generator optimize an objective function based on predictions for real and synthetic images in domain B, given input images in domain A while minimizing the difference between real and synthetic images in domain B

$$\min_G \max_D V(D,G) = \mathop{\mathbb{E}}_{x,y\sim p_{data_{B,A}}}[\log D(x,y)] + \mathop{\mathbb{E}}_{y\sim p_{data_A}}[\log(1 - D(G(y),y)))] + \lambda \mathop{\mathbb{E}}_{x,y\sim p_{data_{B,A}}}[||x - G(y)||]_1$$



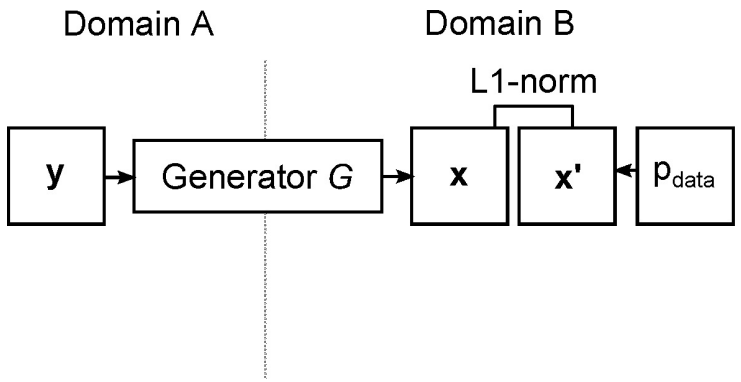
Brain MR segmentation



Ground truth

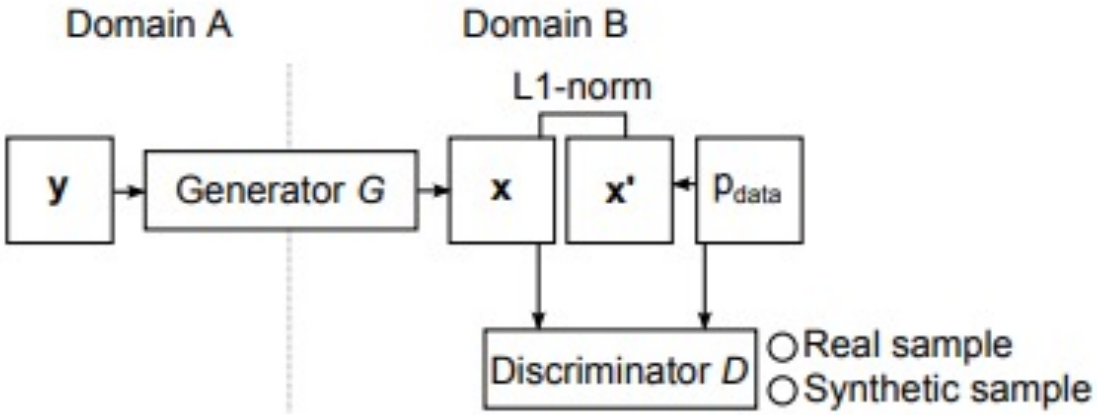
No adversarial training

Adversarial training



Moeskops et al. *Adversarial training and dilated convolutions for brain MRI segmentation*, MICCAI DLMIA Workshop, 2017

Unpaired training: self-regularization



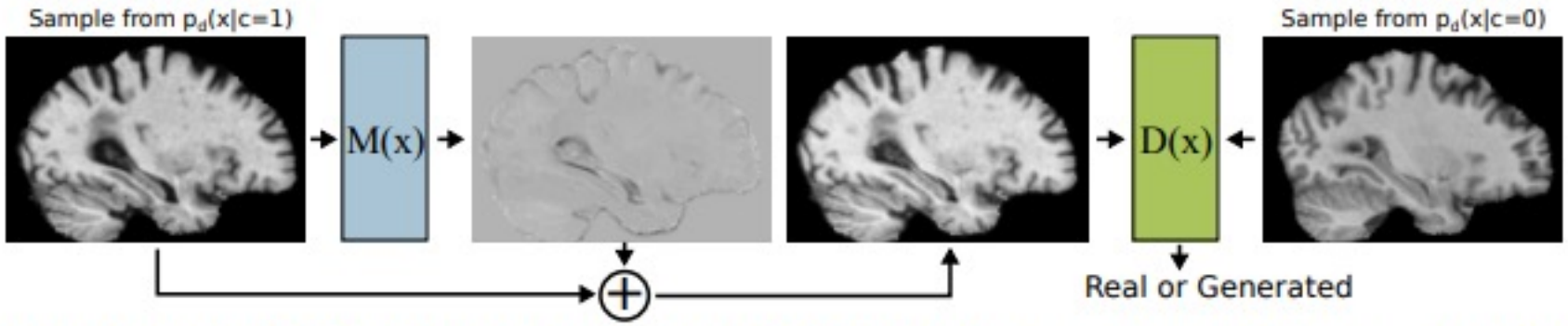
Unpaired training: self-regularization

Domain A

Brain MR images of patients with Alzheimer's

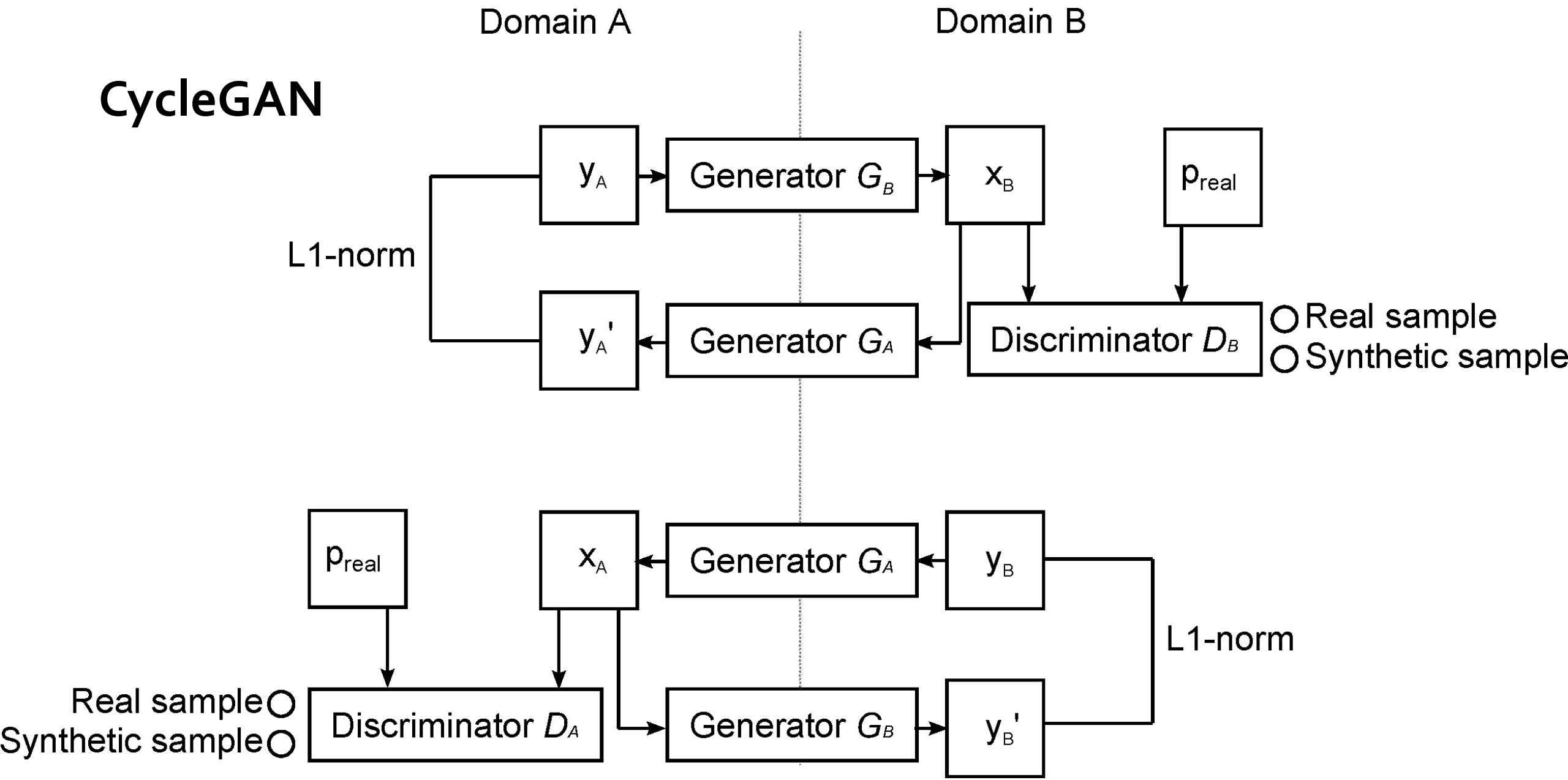
Domain B

Brain MR images of healthy patients



Baumgartner, C. F., Koch, L. M., Can Tezcan, K., Xi Ang, J., & Konukoglu, E. (2018). Visual feature attribution using Wasserstein GANs. In CVPR (pp. 8309-8319).

CycleGAN



Zhu et al. *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*, ICCV 2017

CycleGAN

GAN objective functions in domain B and domain A and a cycle consistency loss

$$V_B(D_B, G_B) = \mathop{\mathbb{E}}_{x \sim p_{data_B}} [\log D_B(x)] + \mathop{\mathbb{E}}_{y \sim p_{data_A}} [\log (1 - D_B(G_B(y)))]$$

$$V_A(D_A, G_A) = \mathop{\mathbb{E}}_{y \sim p_{data_A}} [\log D_A(y)] + \mathop{\mathbb{E}}_{x \sim p_{data_B}} [\log (1 - D_A(G_A(x)))]$$

$$V_{Cycle}(G_A, G_B) = \mathop{\mathbb{E}}_{y \sim p_{data_A}} [||G_A(G_B(y)) - y||_1] + \mathop{\mathbb{E}}_{x \sim p_{data_B}} [||G_B(G_A(x)) - x||_1]$$

$$\min_{G_A, G_B} \max_{D_A, D_B} V(G_A, G_B, D_A, D_B) = V_B(D_B, G_B) + V_A(D_A, G_A) + \lambda V_{Cycle}(G_A, G_B)$$

Zhu et al. *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*, ICCV 2017

CycleGAN

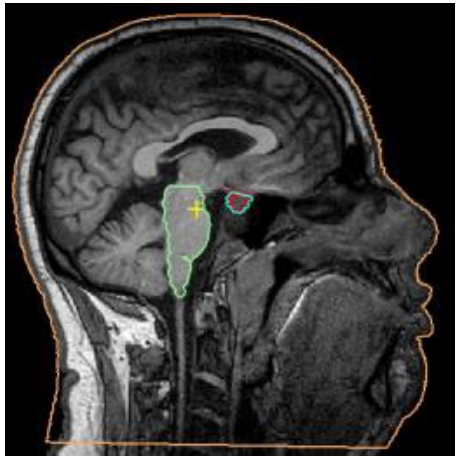
Domain A | **Domain B**

https://github.com/junyanz/CycleGAN

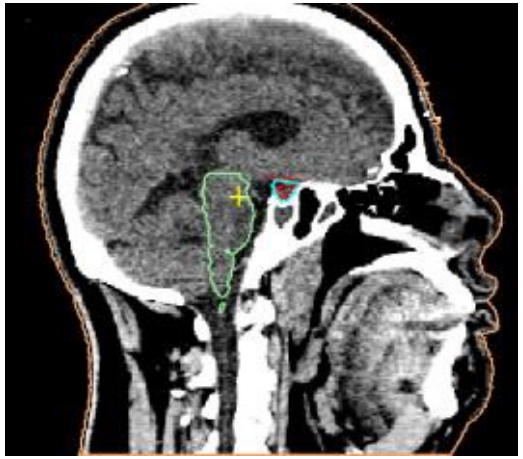
Example: MR to CT synthesis

- Radiotherapy treatment planning requires
  - MR volume
    - Soft tissue contrast
    - Tissue delineation
  - CT volume
    - Electron density
    - Dose calculation

- Acquisition of both volumes leads to
  - Increase in time and money
  - Decrease in patient comfort



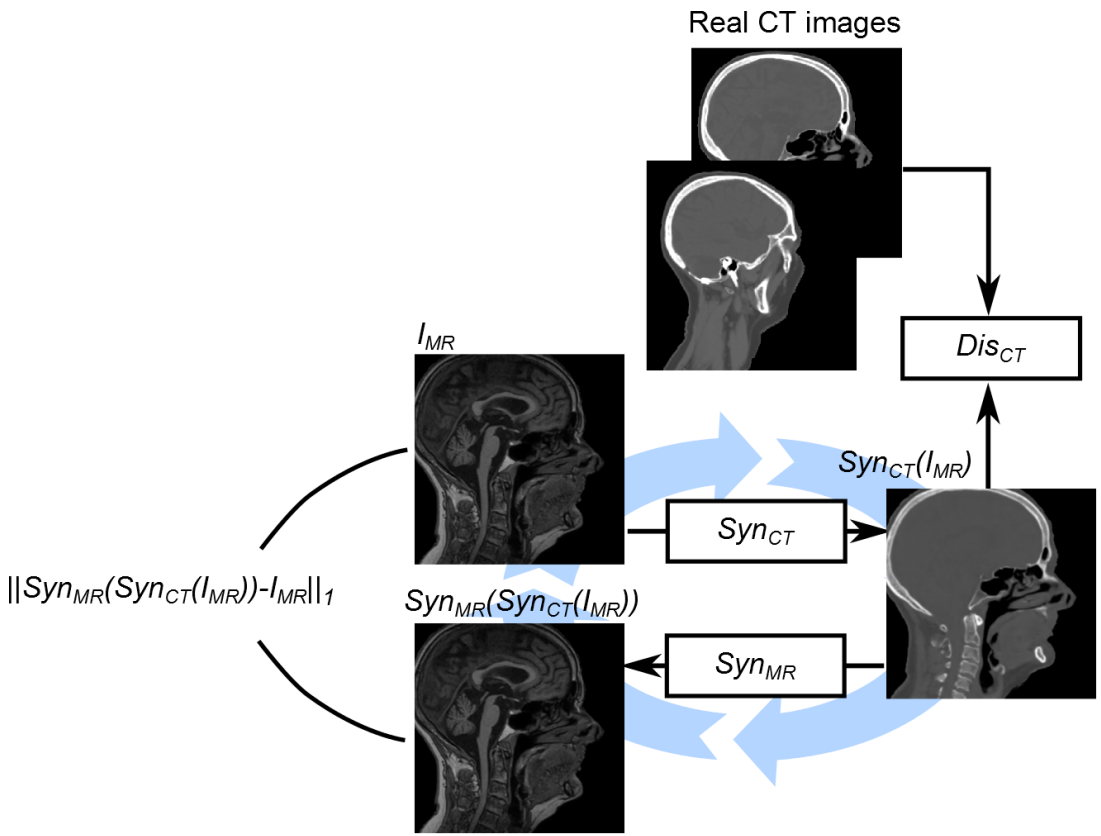
MR



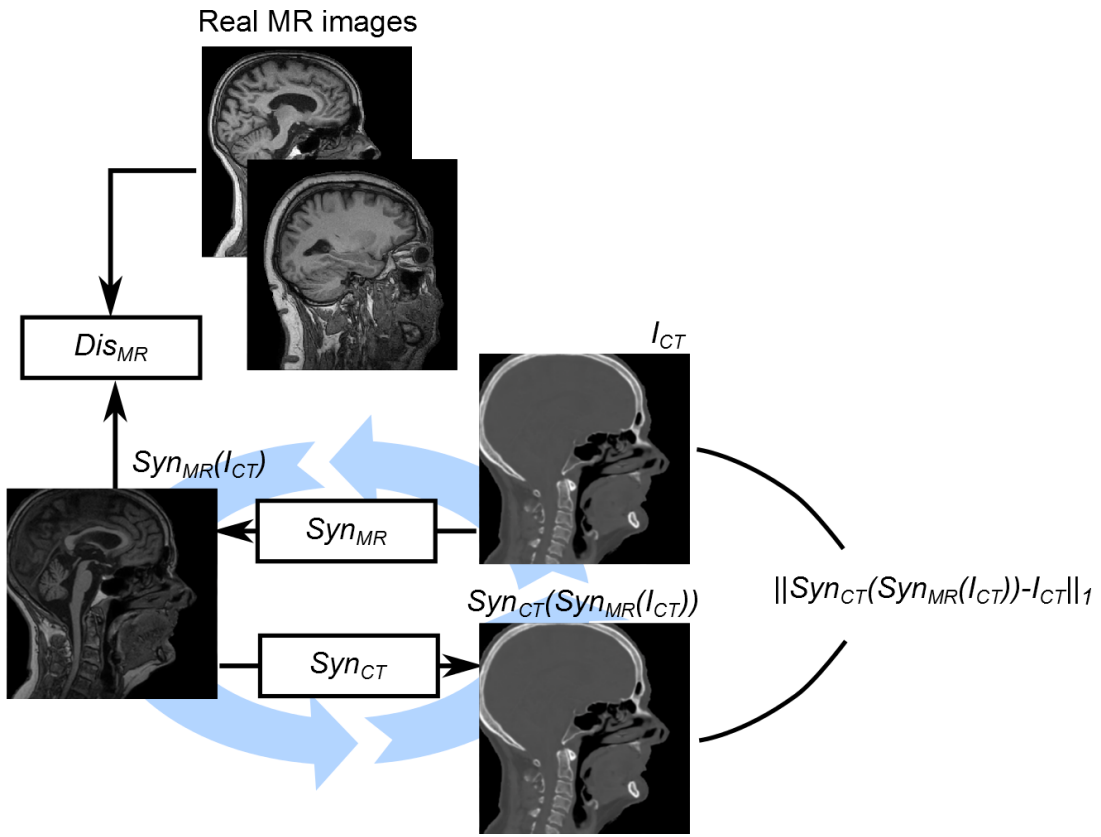
CT

Wolterink et al. *Deep MR to CT synthesis using unpaired data*, MICCAI SASHIMI Workshop 2017

Example: MR to CT synthesis



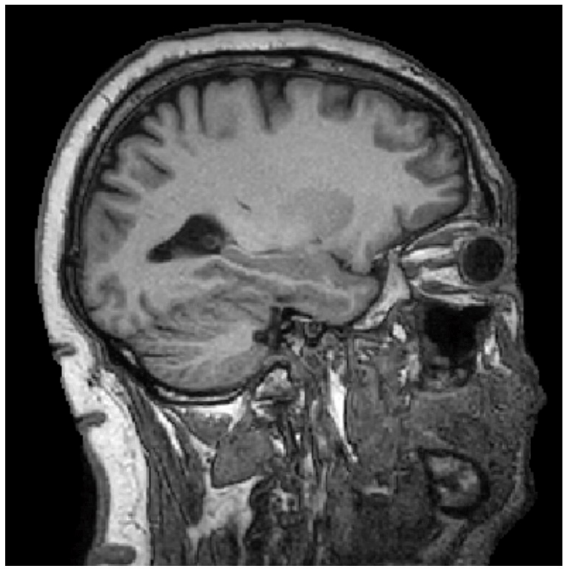
Forward cycle



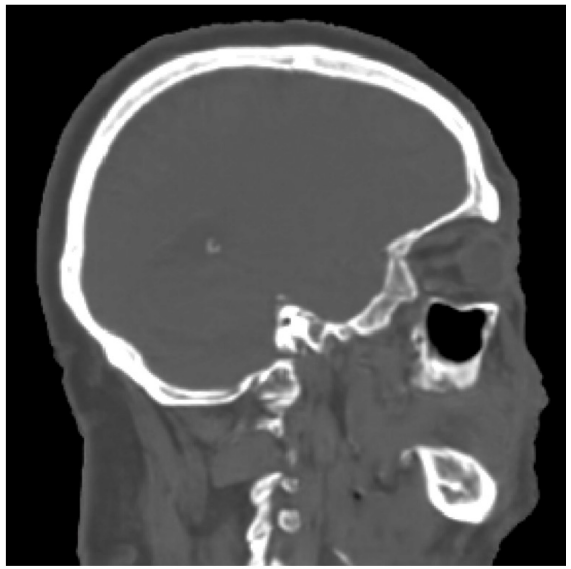
Backward cycle

Zhu et al. *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*, ICCV 2017

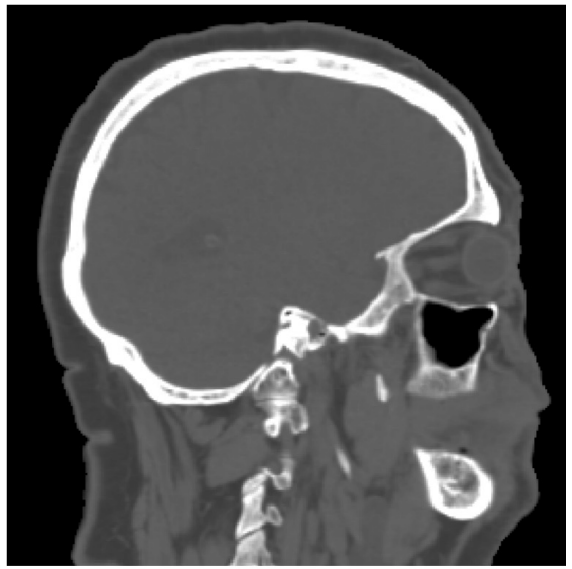
Example: MR to CT synthesis



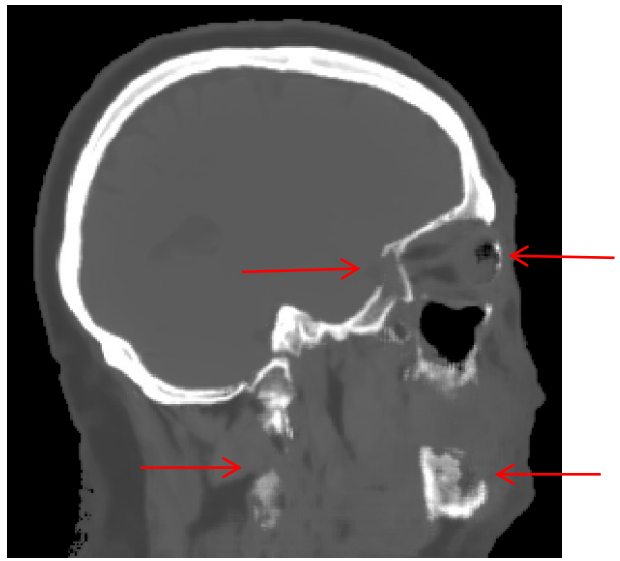
I_{MR}



$Syn_{CT}(I_{MR})$



I_{CT}



Regression

Zhu et al. *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*, ICCV 2017

Example: MR to CT synthesis

I_{MR} $Syn_{CT}(I_{MR})$ I_{CT} $|Syn_{CT}(I_{MR})-I_{CT}|$

Zhu et al. *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*, ICCV 2017

CycleGAN

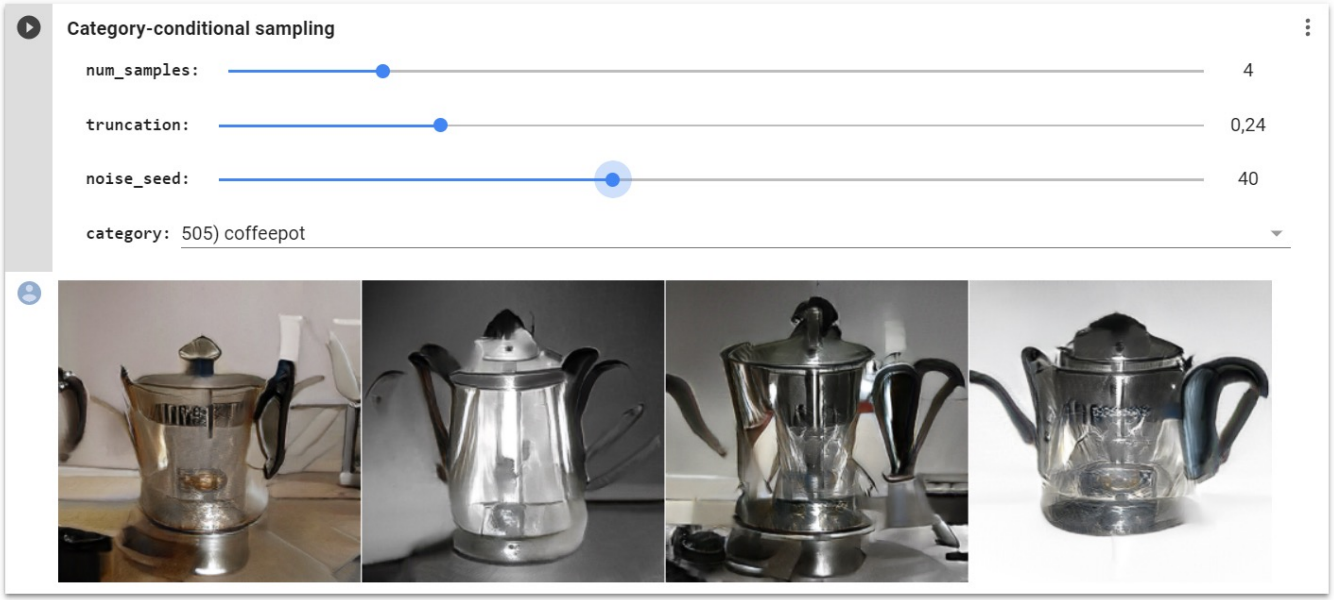
Domain A

Domain B

https://github.com/junyanz/CycleGAN

Summary

- Interpretability + explainability important for practical ML use
- Generative adversarial networks are SOTA for image synthesis
- Conditional GANs synthesize samples with particular characteristics
- Adversarial networks can define loss functions that we cannot
- Many applications in medical image analysis



Practical assignment

Train your own GAN! https://tinyurl.com/capitaselectacolab

- Match a normal distribution
- MNIST digits (unconditional + conditional)
- Histopathology images: https://github.com/basveeling/pcam
- Run with free GPU: in Playground (no saving) or make copy in Google Drive (log in)
- Experiments with BigGAN https://tinyurl.com/y8yuqyqv