Advanced Multimodal Machine Learning

Lecture 10.1: Optimization, VAE, GANs

Louis-Philippe Morency
Lecturer: Amir Zadeh

We will have a moment of silence at the beginning of class in memory of those in our community who died on Saturday:

Joyce Fienberg
Richard Gottfried
Rose Mallinger
Jerry Rabinowitz
Cecil Rosenthal
David Rosenthal
Bernice Simon
Sylvan Simon
Daniel Stein
Melvin Wax
Irving Younger

What you can do:

- Support one another, recognizing especially the impact on our Jewish community
- Speak up for respect and tolerance of diverse ideas, lifestyles, religions
- Give blood - www.vitalant.org (Central Blood Bank)
- GoFundMe page - https://www.gofundme.com/tree-of-life-synagogue-shooting and/or https://jewishpgh.org/our-victims-of-terror-fund/

If you need someone to talk to: https://www.cmu.edu/counseling/

Lecture Objectives

- Practical Deep Model Optimization
  - Adaptive Optimization Methods
  - Regularization
  - Co-adaptation
  - Multimodal Optimization
- VAE
- GAN

Lecture Objectives

- Practical Deep Model Optimization
  - Adaptive Optimization Methods
  - Regularization
  - Co-adaptation
  - Multimodal Optimization
- VAE
- GAN

Adaptive Learning Rate

General Idea: Let neurons who just started learning have huge learning rate.

Adaptive Learning Rate is an active area of research:

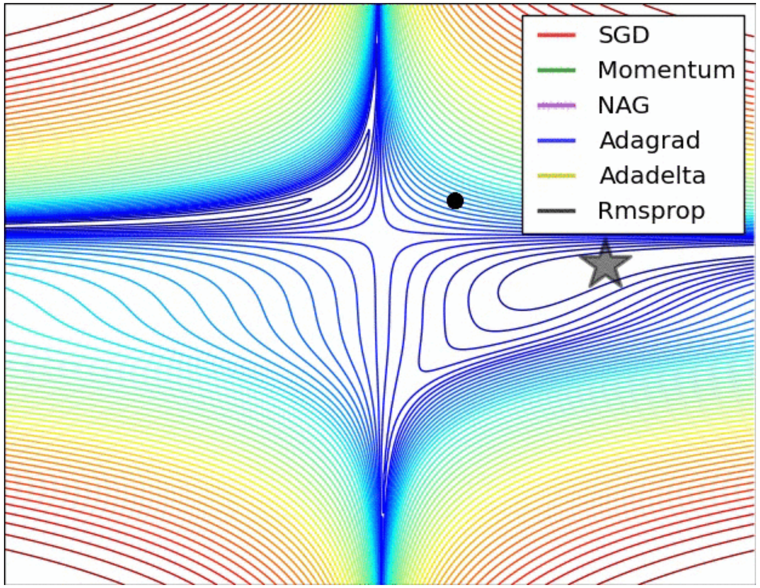
- Adadelta
- RMSProp

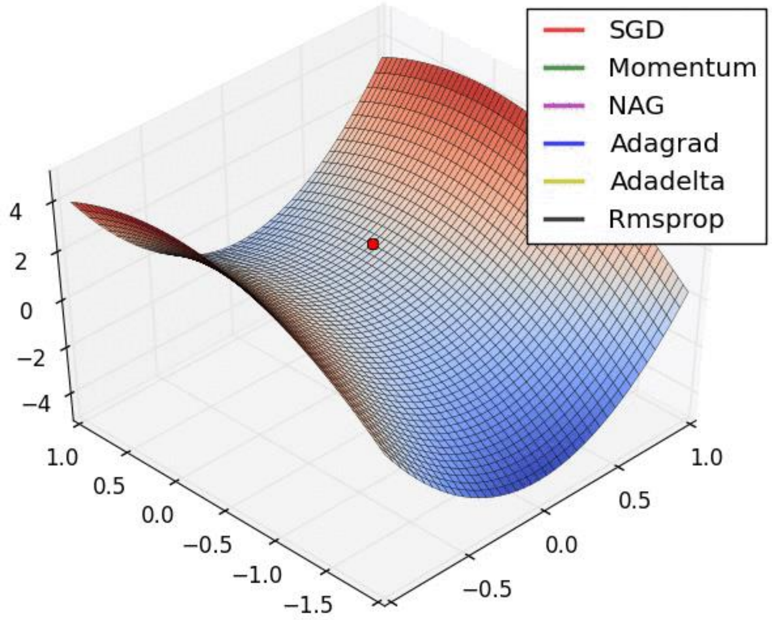
```
cache = decay_rate * cache + (1 - decay_rate) * dx**2
x += - learning_rate * dx / (np.sqrt(cache) + eps)
```

- Adam

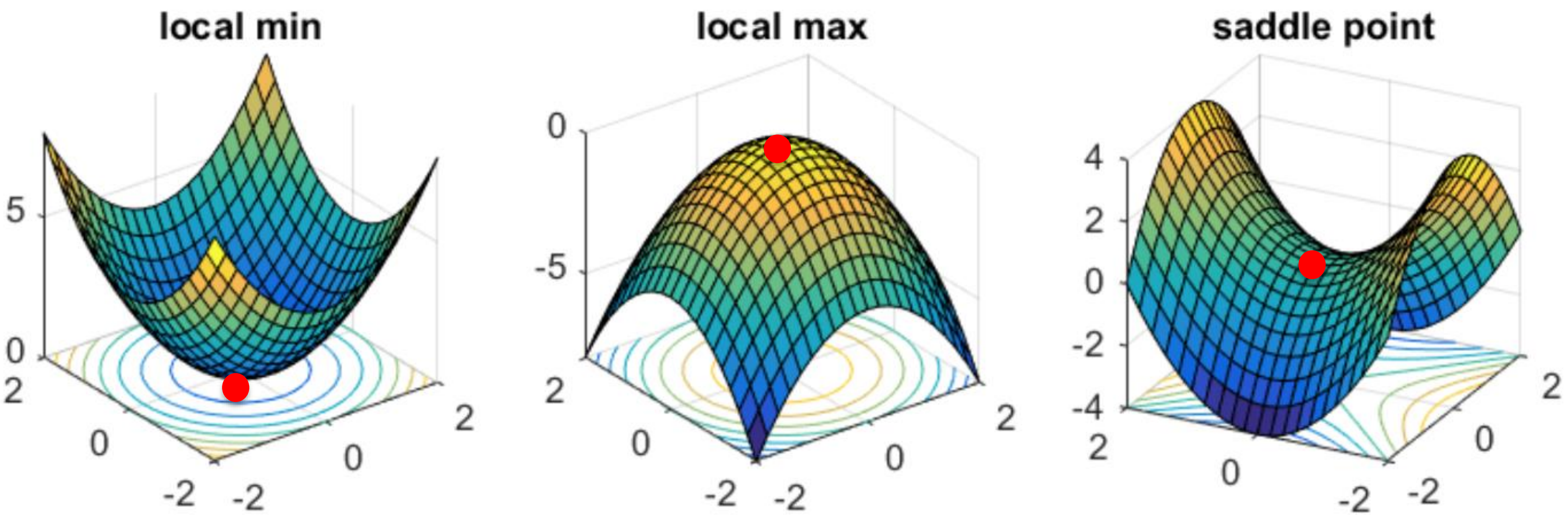
```
m = beta1*m + (1-beta1)*dx
v = beta2*v + (1-beta2)*(dx**2)
x += - learning_rate * m / (np.sqrt(v) + eps)
```

Comparison


SGD
Momentum
NAG
Adagrad
Adadelta
Rmsprop


SGD
Momentum
NAG
Adagrad
Adadelta
Rmsprop
4
2
0
−2
−4
1.0
0.5
0.0
−0.5
−1.0
−1.5
1.0
0.5
0.0
−0.5

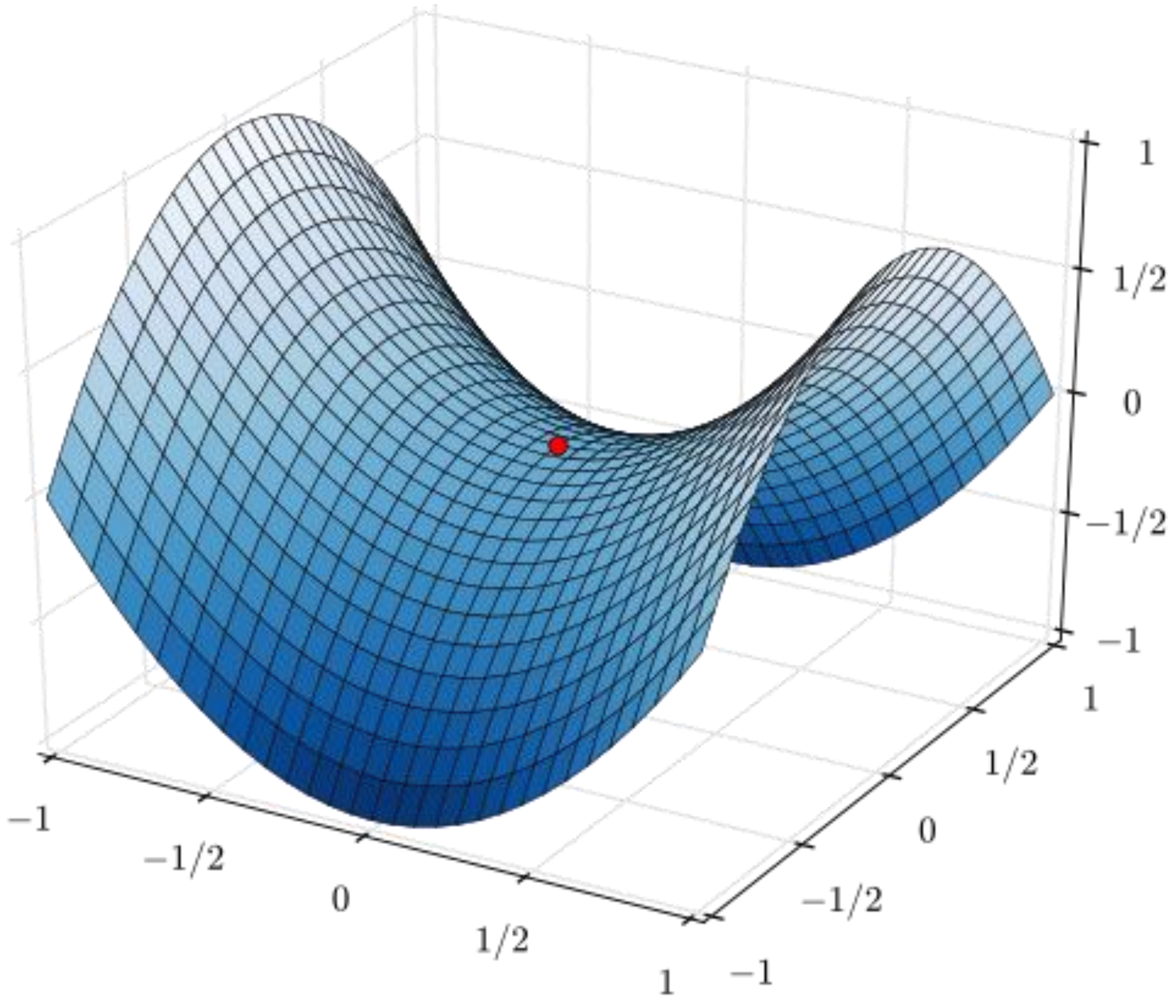
Critical Points



Saddle Points

- Deep Learning Optimization:
  - Deep Learning problems in general have many local minimas
  - Many (not all) of them are actually almost as good as global minima due to parameter permutation
  - However it is NP-hard to even find a local minima
- Lots and lots of saddles in many deep learning problems.

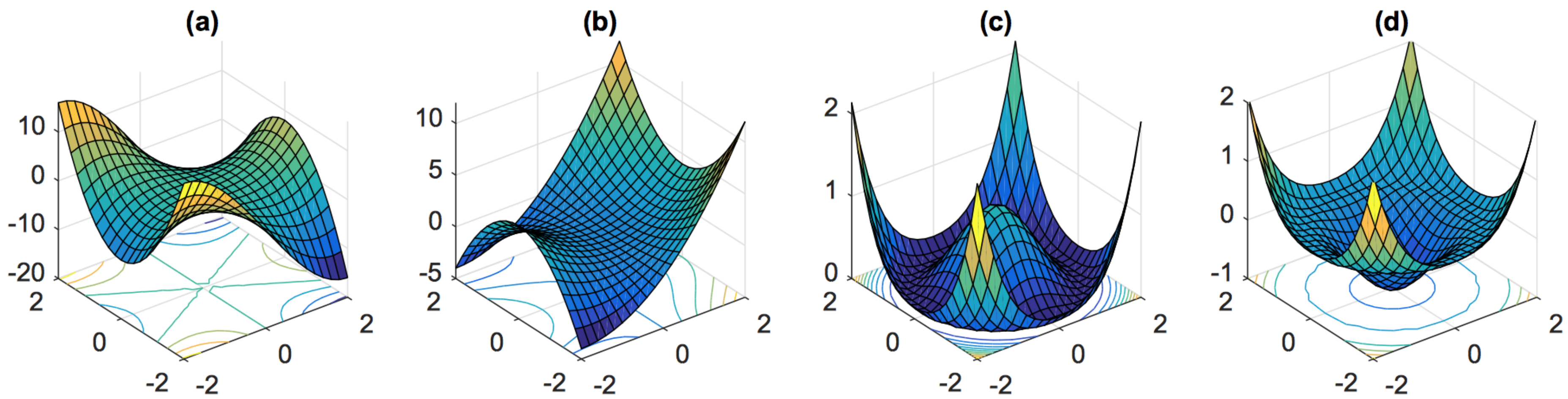
Why Saddles are Bad



Detecting Saddles

- One way to detect saddles:
  - Calculate Hessian at point x
  - If Hessian is indefinite you have a saddle for sure.
  - If Hessian is not indefinite you really can't tell.
- My loss isn't changing:
  - You are definitely close to a critical point
    - You may be in a saddle point
    - You may be in the local minima/maxima
  - One trick: quickly check the sorrounding
    - Best practical trick if Hessian is not indefinite.

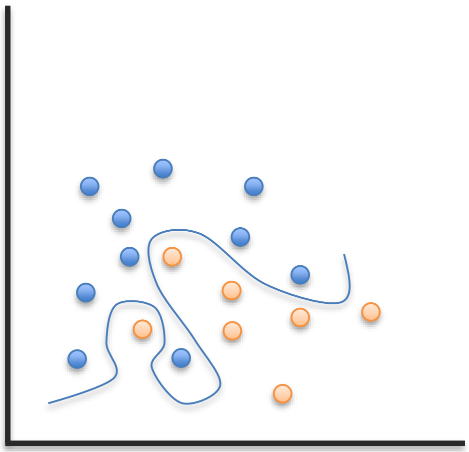
Bad Saddle Points



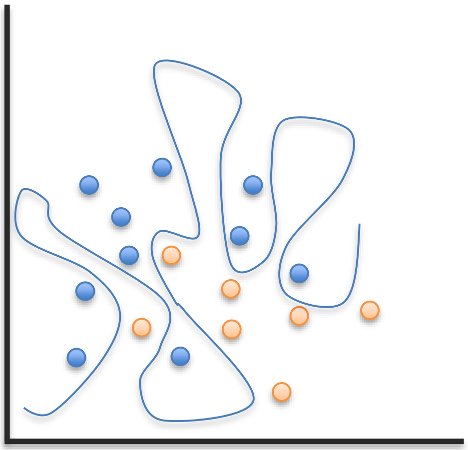
https://arxiv.org/pdf/1602.05908.pdf

Example

Real

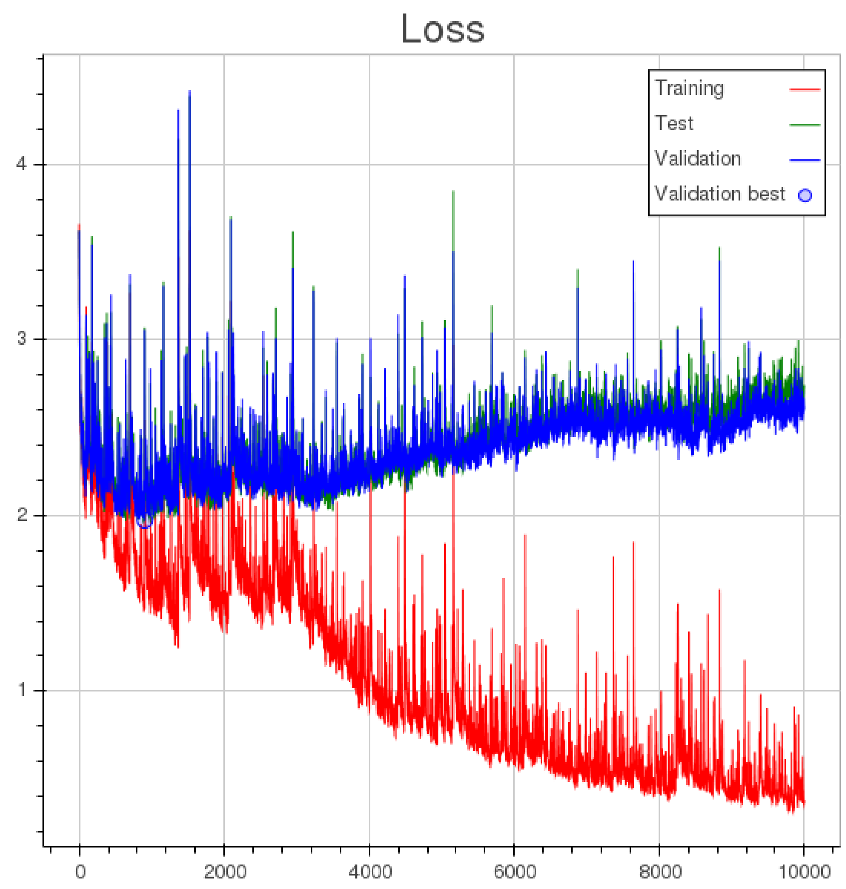


Our
Model



Not the fault of learning rate or momentum

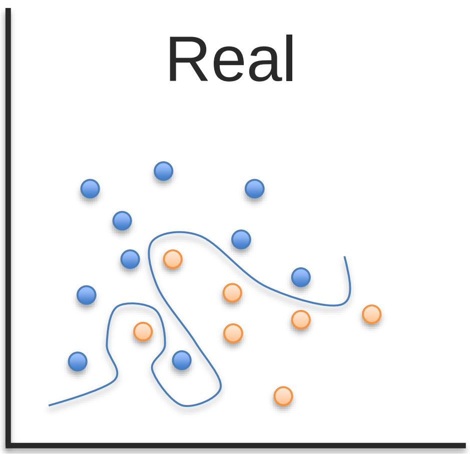
Example

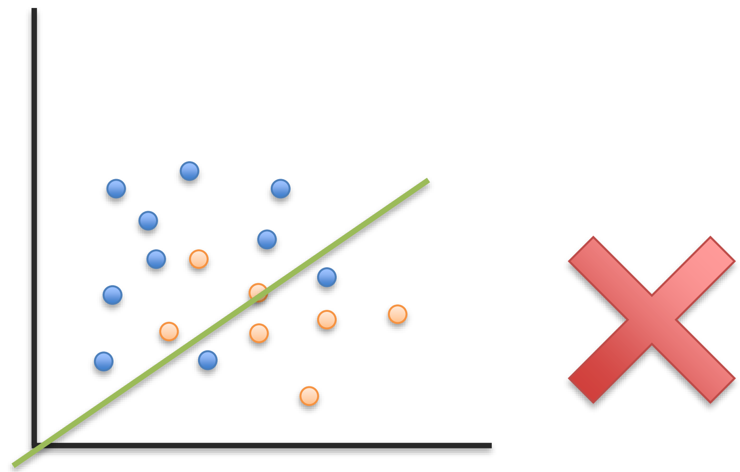

Loss
Training
Test
Validation
Validation best
4
3
2
1
0
2000
4000
6000
8000
10000

Bias-Variance

- Problem of bias and variance
  - Simple models are unlikely to find the solution to a hard problem, thus probability of finding the right model is low.

No longer SOT!

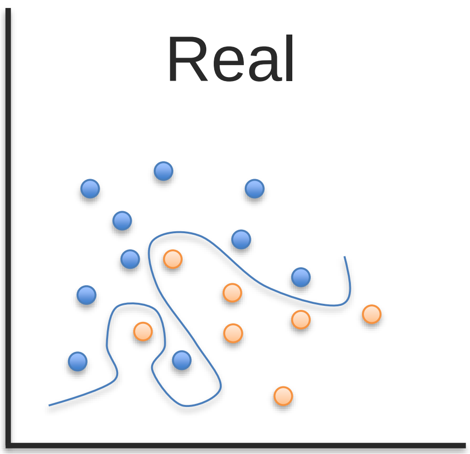


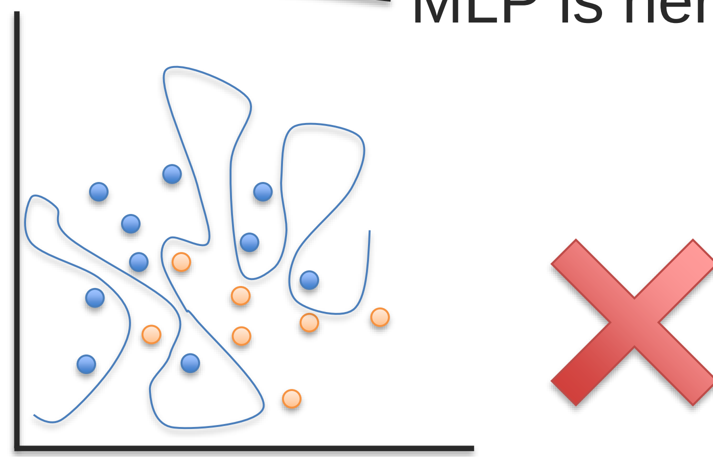


Bias-Variance

- Problem of bias and variance
  - Simple models are unlikely to find the solution to a hard problem, thus probability of finding the right model is low.
  - Complex models find many solutions to a problem, thus probability of finding the right model is again low.

MLP is here!



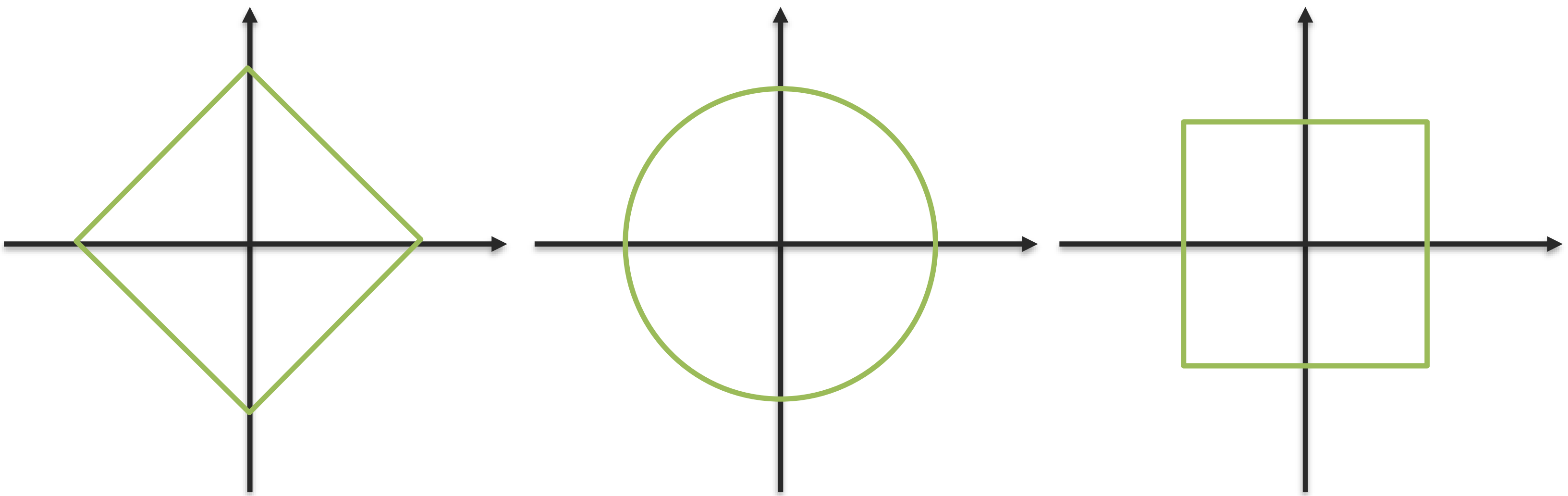


Lecture Objectives

- Practical Deep Model Optimization
  - Adaptive Optimization Methods
  - Regularization
  - Co-adaptation
  - Multimodal Optimization
- VAE
- GAN

Regularization

- Parameter Regularization:
  - Adding prior to the network parameters
  - L^p Norms



L^1 L^2 L^∞

Minimize: $Loss(x;\theta) + \propto||\theta||$

Parameter Regularization

- Parameter Regularization:
  - L^1(Lasso) and L^2 (Ridge) are the most famous norms used. Sometimes combined (Elastic)
  - Other norms are computationally ineffective.
- Maximum a posteriori (MAP) estimation:
  - Having priors one the model parameters
  - L^2 can be seen as a Gaussian prior on model parameters θ
  - A generalization of L^2 is called Tikhonov Regularization with Multivariate Gaussian prior on model parameters.
    - Assuming Correlation between parameters one can build a Mahalanobis variation of Tikhonov Regularization.

Structural Regularization

- Lots of models can learn everything.
- Go for simpler ones.
- Use task specific models:
  - CNNs
  - RecNNs
  - LSTMs
  - GRUs

Occam's razor

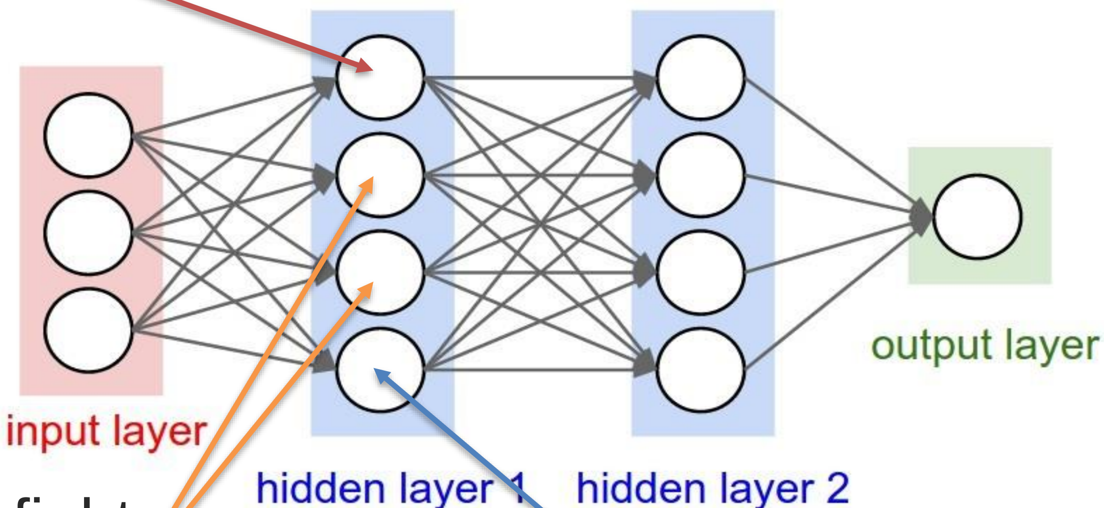
Lecture Objectives

- Practical Deep Model Optimization
  - Adaptive Optimization Methods
  - Regularization
  - Co-adaptation
  - Multimodal Optimization
- VAE
- GAN

Example

- A neuron learns something that is not useful:
    1. Learn something useful
    2. Other neurons learn to mitigate it.

Useless neuron

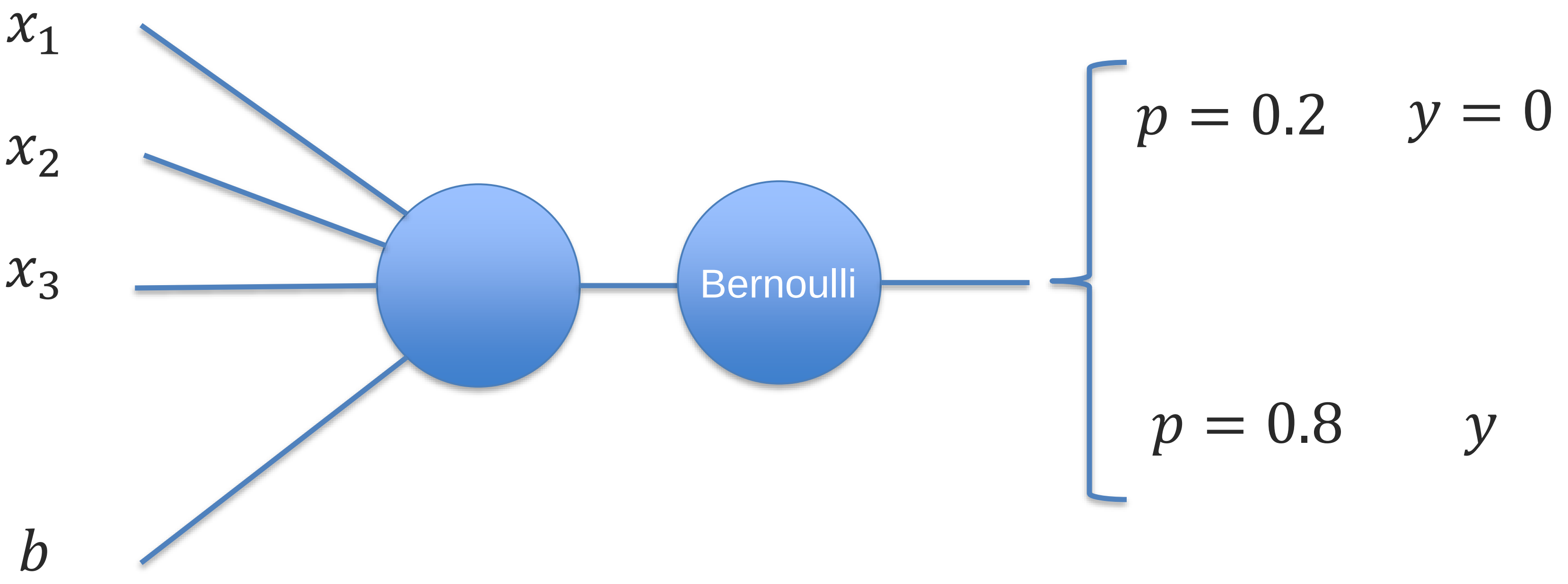


Learning to fight useless neuron

Actually learning something

Dropout

- Simply multiply the output of a hidden layer with a mask of 0s and 1s (Bernoulli)



Dropout

Forward step: multiply with a Bernoulli distribution per epoch, batch or sample point. Question: which one works better?

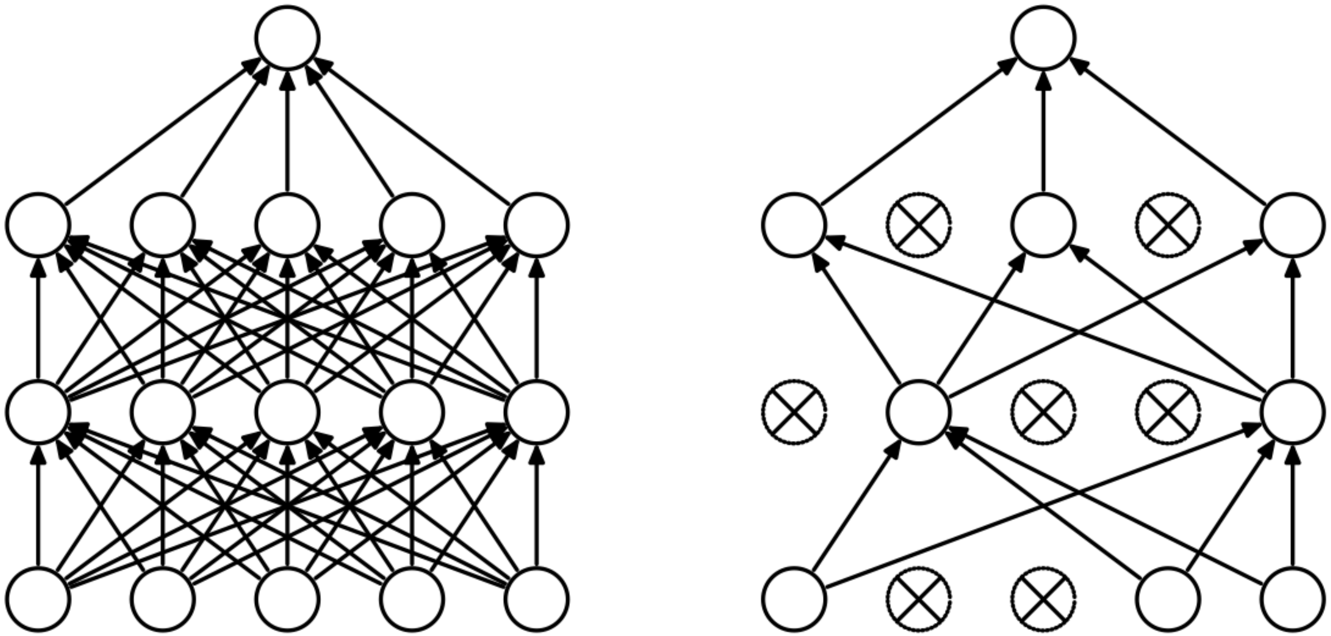
Backward step: just calculate the gradients same as before. Question: some neurons are out of the network, so how does this work?

All good? Nope

Multiply the weights by $1 - p_i$

Dropout

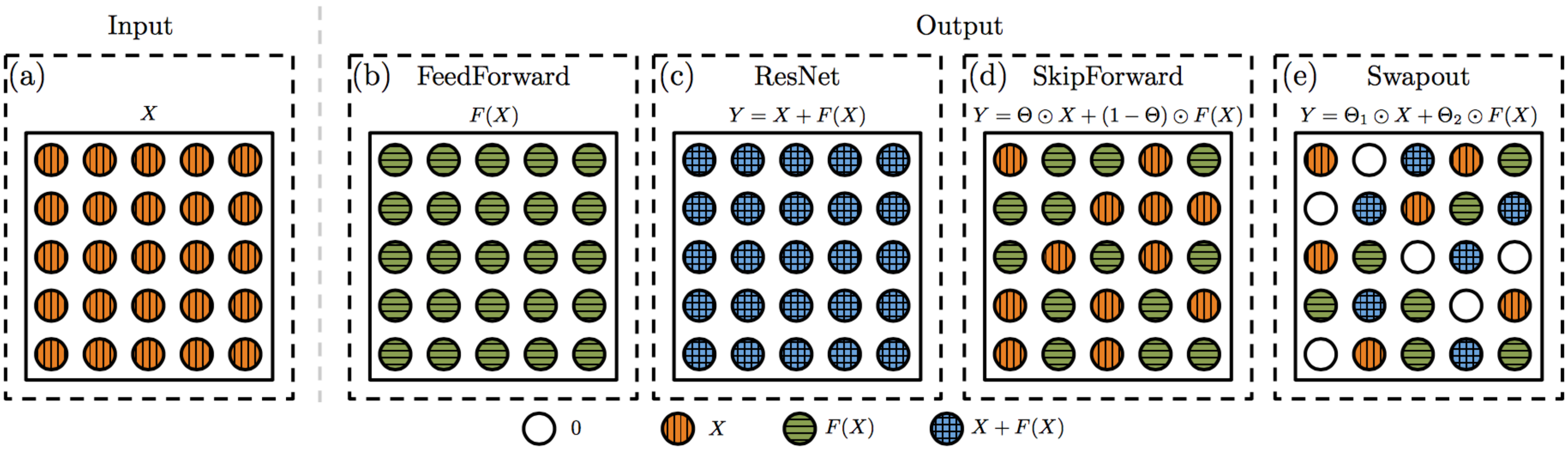
Stop co-adaptation + learn ensemble



Other variations

- Gaussian dropout: instead of multiplying with a Bernoulli random variable, multiply with a Gaussian with mean 1.

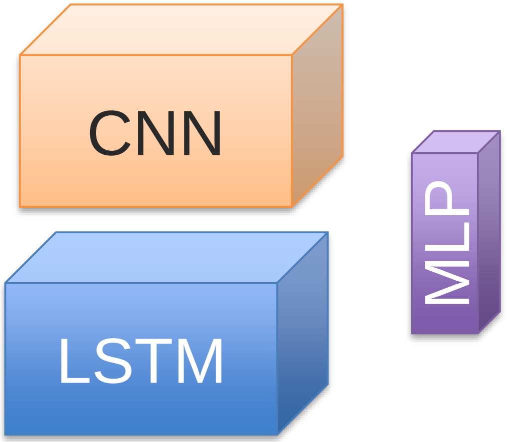
- Swapout: Allow skip-connections to happen



Lecture Objectives

- Practical Deep Model Optimization
  - Adaptive Optimization Methods
  - Regularization
  - Co-adaptation
  - Multimodal Optimization
- VAE
- GAN

Multimodal Optimization

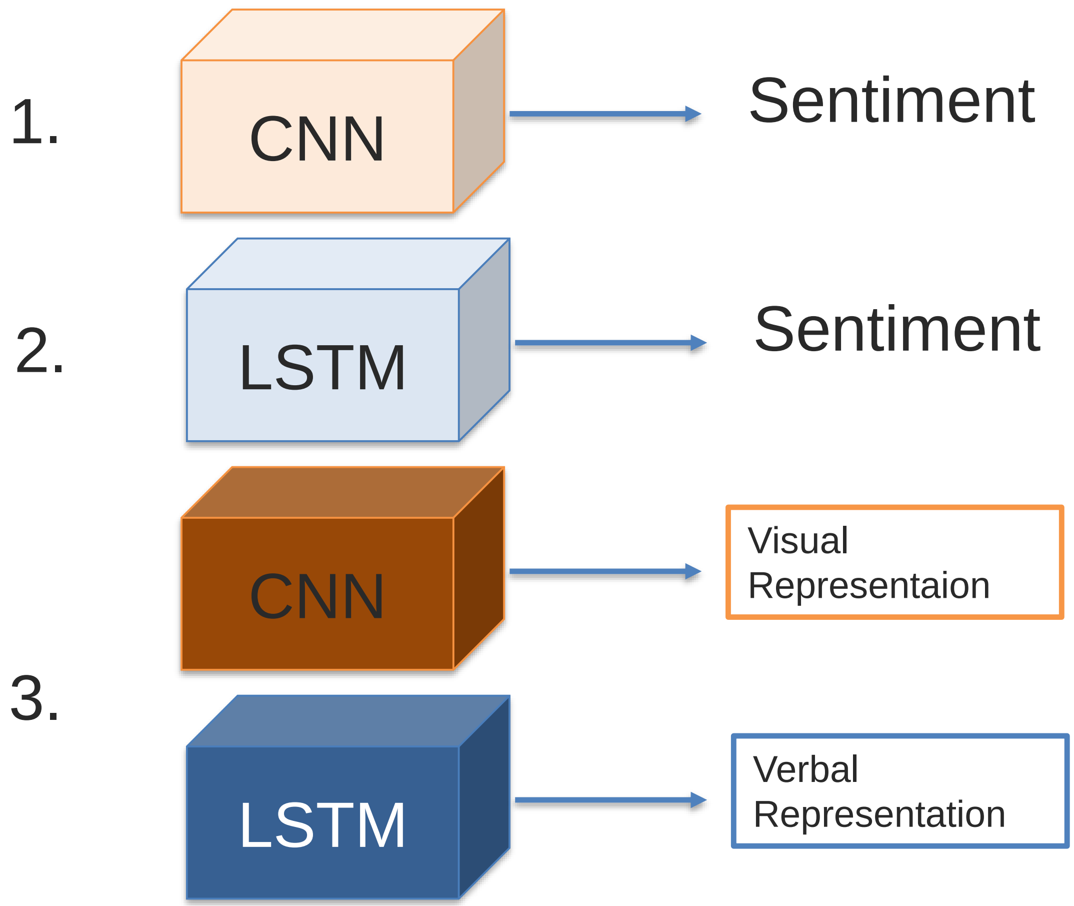


- Biggest Challenge:
  - Data from different sources
  - Different networks
- Example:
  - Question Answering: LSTM(s) connected to a CNN
  - Multimodal Sentiment: LSTM(s) fused with MLPs and 3D-CNNs
- CNNs work well with high decaying learning rate
- LSTMs work well with adaptive methods and normal SGD
- MLPs are very good with adaptive methods

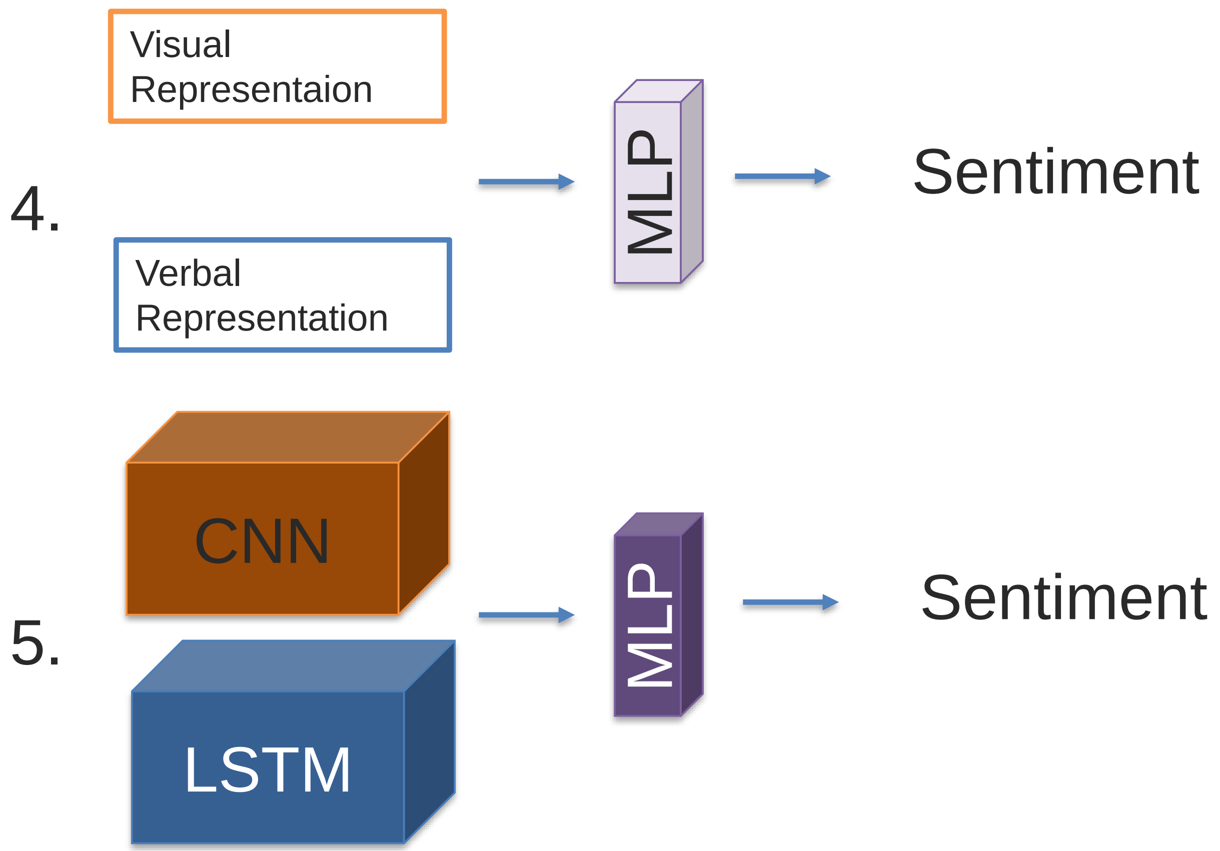
Multimodal Optimization

- How to work with all of them?
- Pre-training is the most straight forward way:
  - Train each individual component of the model separately
  - Put together and fine tune
- Example: Multimodal Sentiment Analysis

Pre-training



Pre-training



Pre-training Tricks

- In the final stage (5), it is better to not use adaptive methods such as Adam.
  - Adam starts with huge momentum on all the networks parameters and can destroy the effects of pretraining.
  - Simple SGD mostly helpful.
- Initialization from other pre-trained models:
  - VGG for CNNs
  - Language models for RNNs
  - Layer by layer training for MLPs

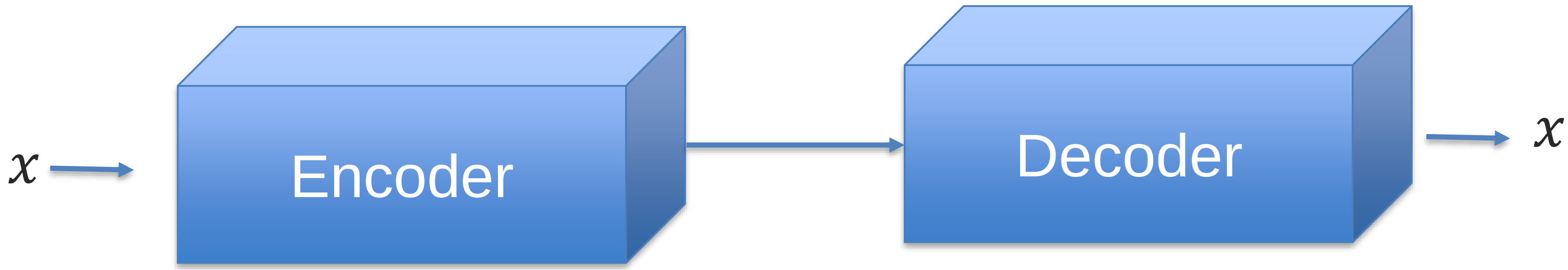
Lecture Objectives

- Practical Deep Model Optimization
  - Adaptive Optimization Methods
  - Regularization
  - Co-adaptation
  - Multimodal Optimization
- VAE
- GAN

Questions?

Auto-encoder

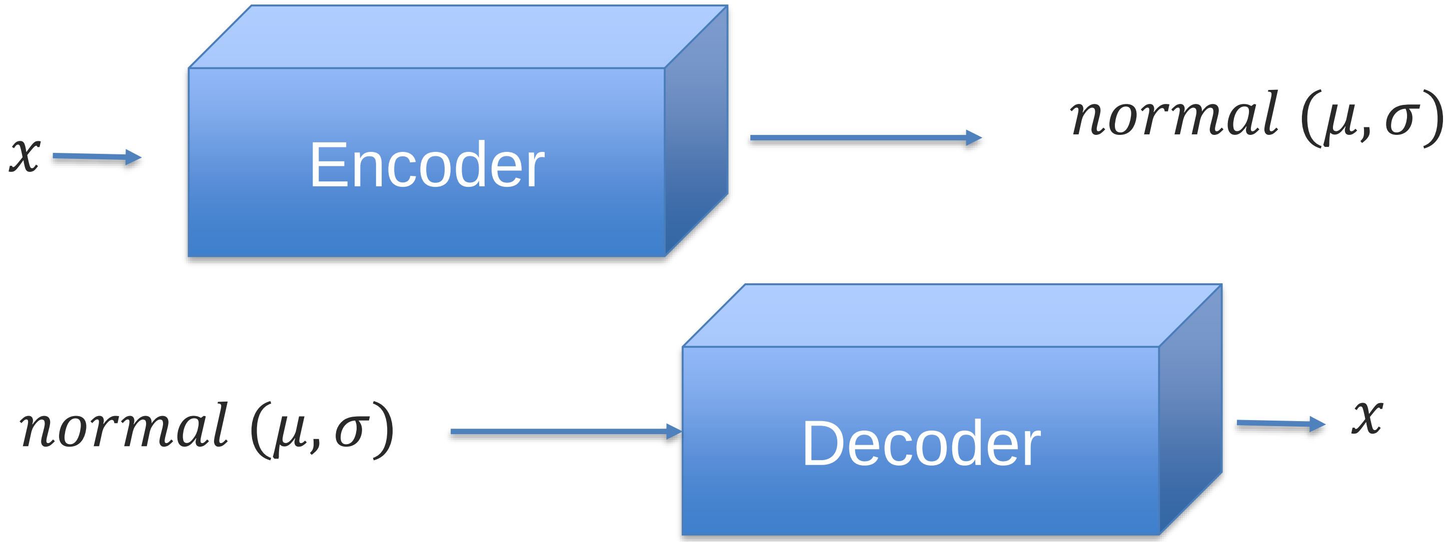
- A combination of an Encoder and a Decoder encoding x and decoding x



- The loss reconstruction error of x.

Variational Auto-encoder

- We assume exact inference is not possible but approximation is possible.



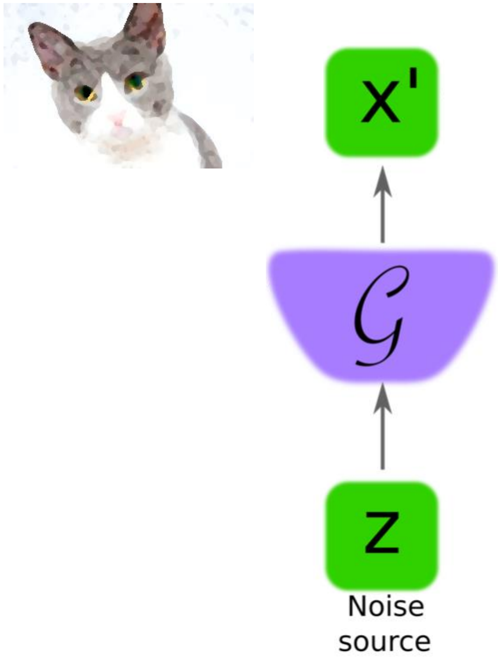
Variational Auto-encoder

- A probability controls the encoder space
  - More meaningful representations
- Space is split in euclidean-meaningful representations.
- The normal distributions have nice properties.

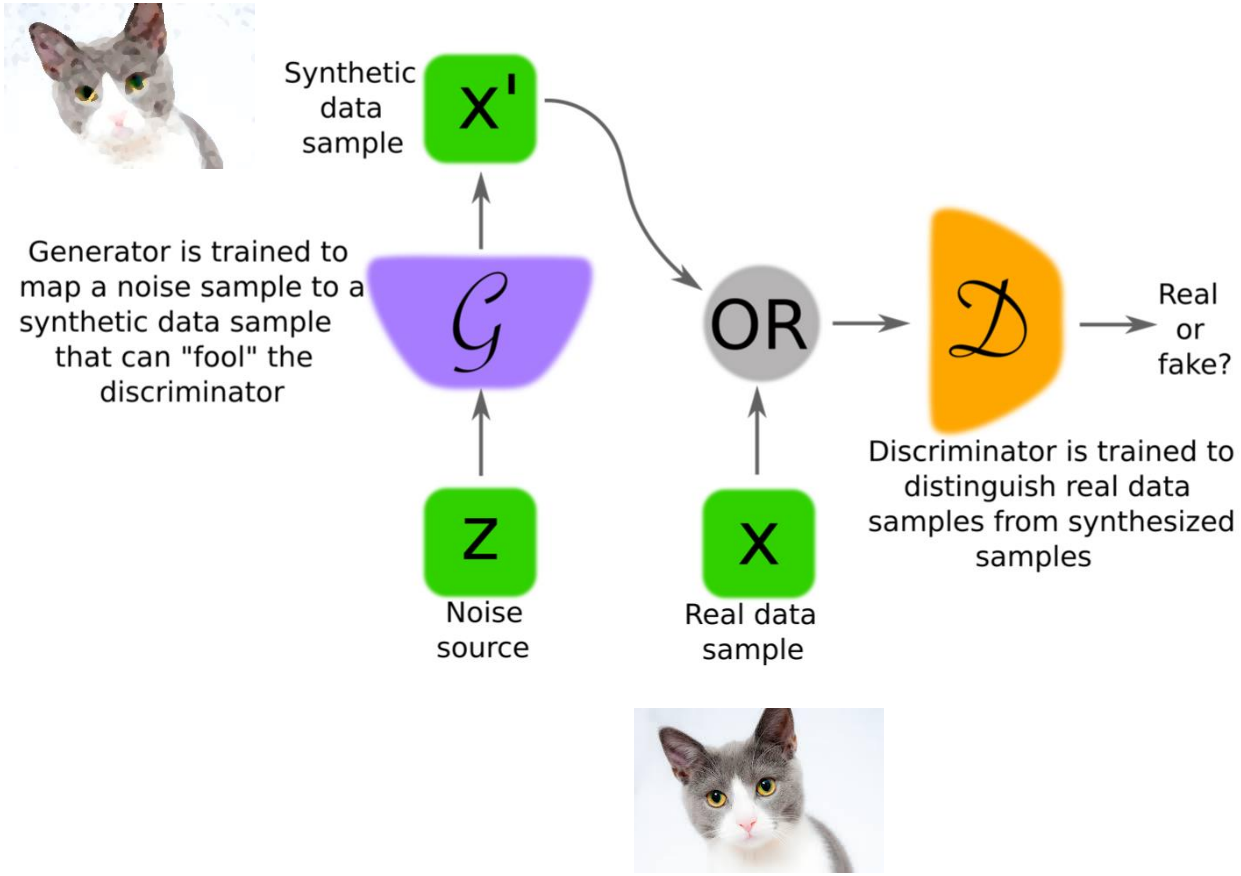
Lecture Objectives

- Practical Deep Model Optimization
  - Adaptive Optimization Methods
  - Regularization
  - Co-adaptation
  - Multimodal Optimization
- VAE
- GAN

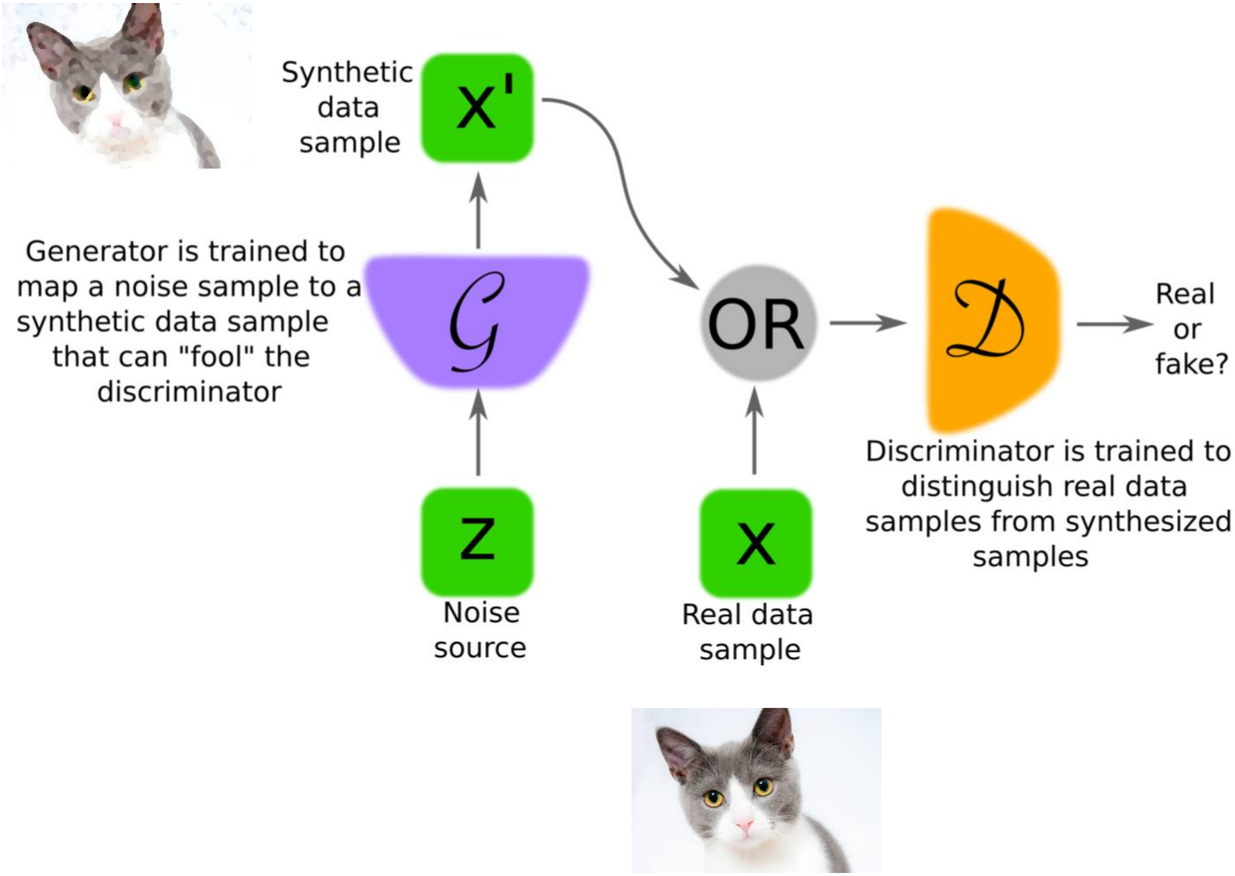
Generative Adversarial Networks



Generative Adversarial Networks

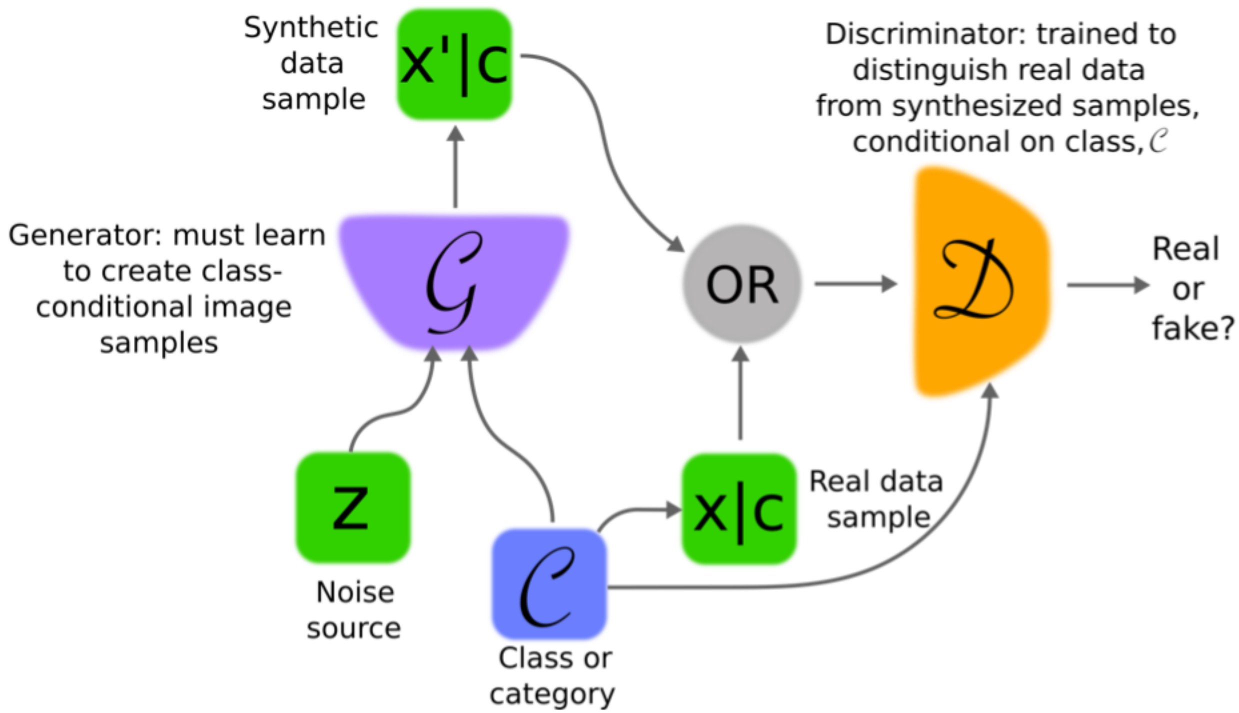


Generative Adversarial Networks

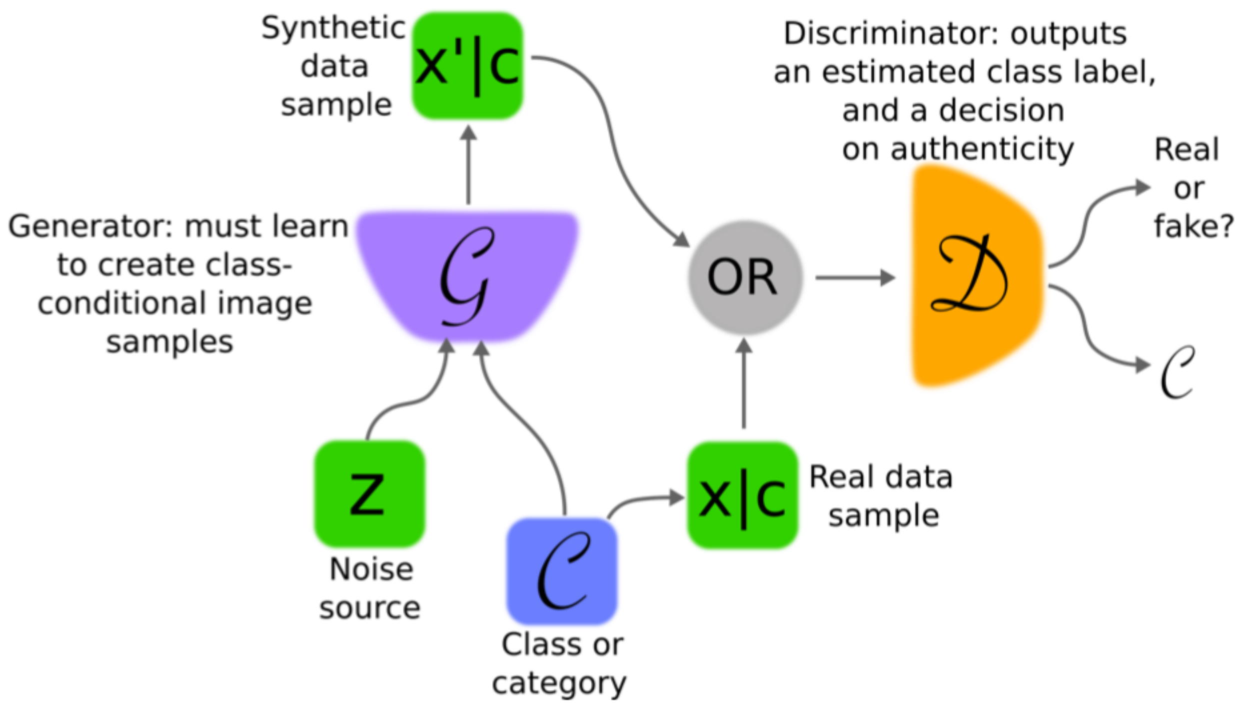


$$\max_{\mathcal{D}} \min_{\mathcal{G}} V(\mathcal{G}, \mathcal{D}) \qquad V(\mathcal{G}, \mathcal{D}) = \mathbb{E}_{p_{data}(\mathbf{x})} \log \mathcal{D}(\mathbf{x}) + \mathbb{E}_{p_g(\mathbf{x})} \log(1 - \mathcal{D}(\mathbf{x}))$$

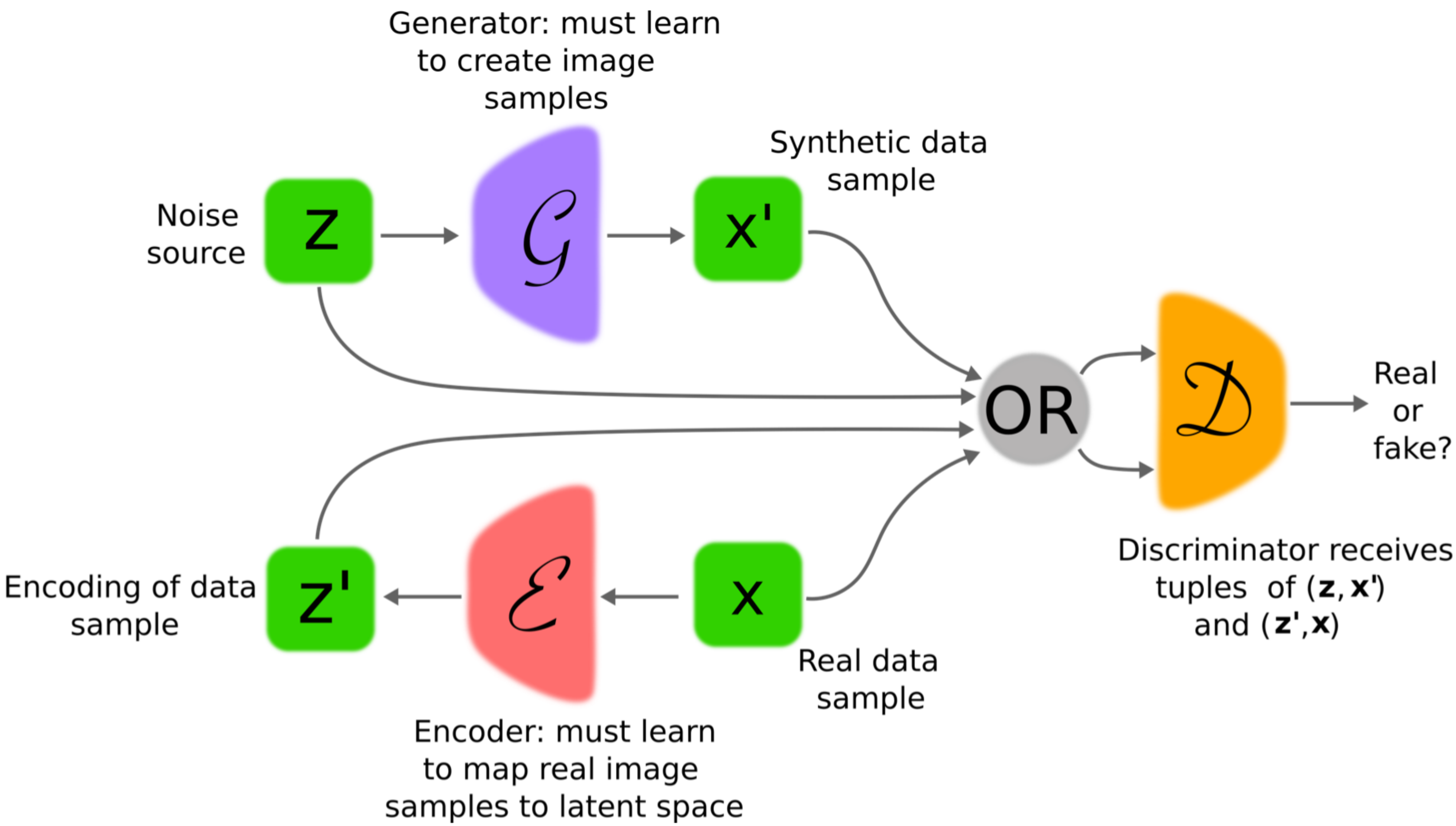
Conditional GAN



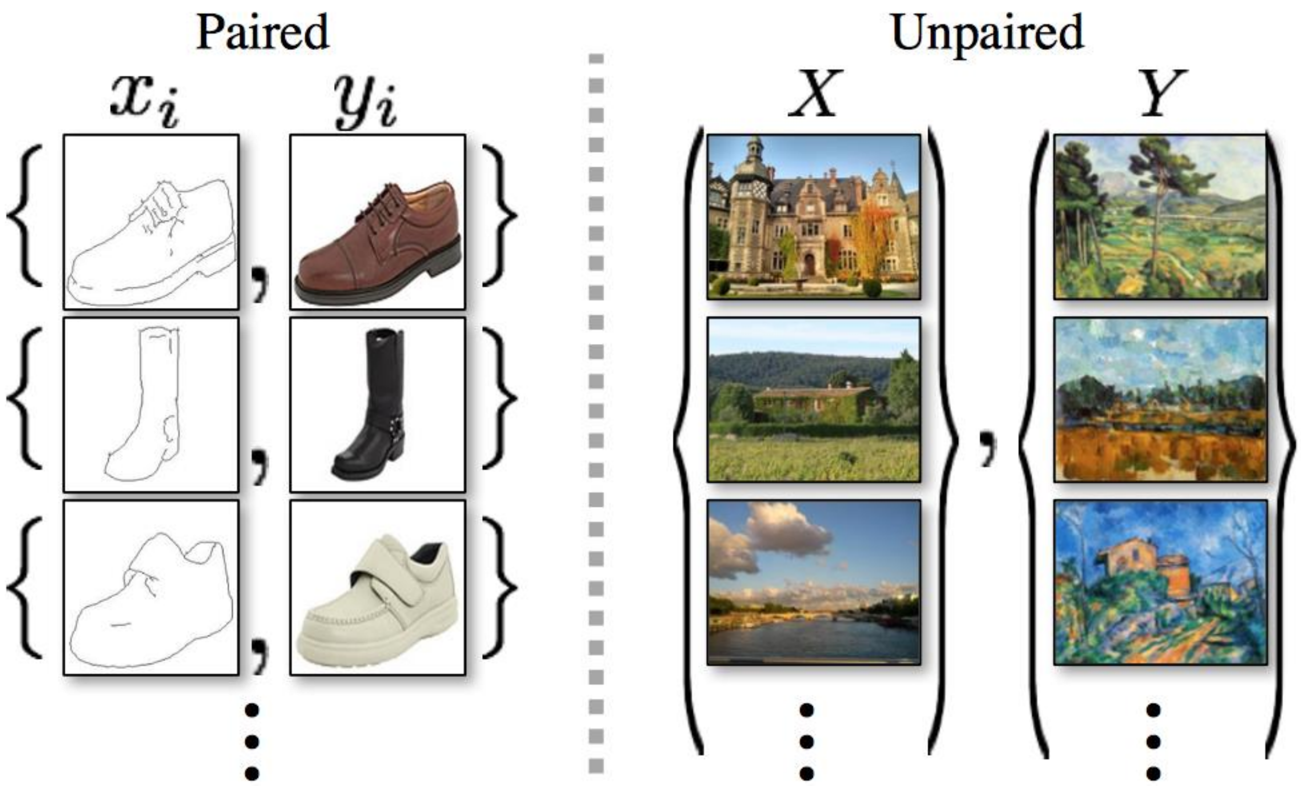
Info GAN


Synthetic data sample
x'|c
Discriminator: outputs an estimated class label, and a decision on authenticity
Real or fake?
Generator: must learn to create class-conditional image samples
$\mathcal{G}$
OR
$\mathcal{D}$
$\mathcal{C}$
z
Noise source
$\mathcal{C}$
Class or category
x|c
Real data sample

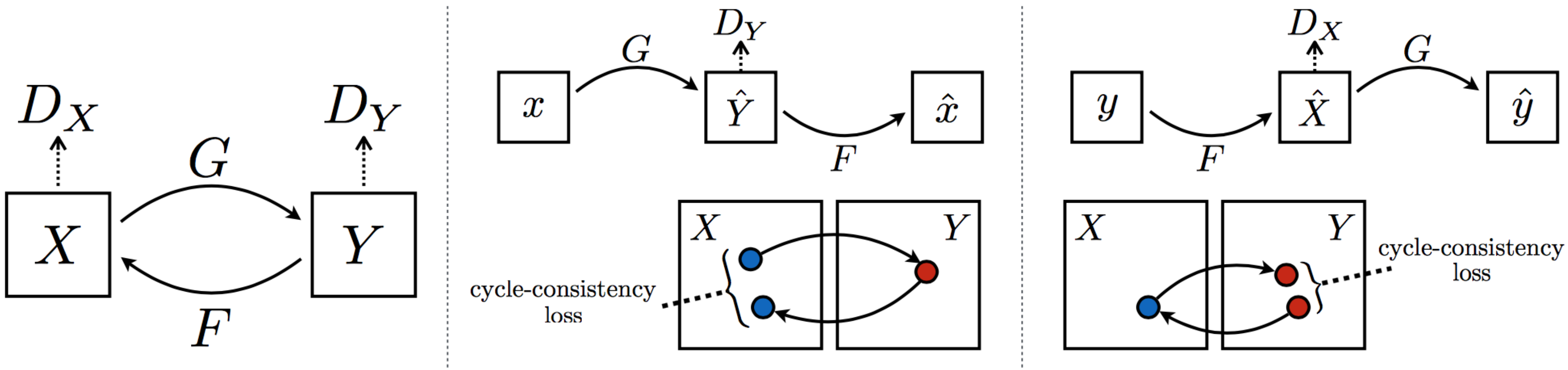
BiGAN


Generator: must learn
to create image
samples
Synthetic data
sample
Noise
source
z
$\mathcal{G}$
x'
OR
$\mathcal{D}$
Real
or
fake?
Encoding of data
sample
z'
$\mathcal{E}$
x
Real data
sample
Discriminator receives
tuples of (**z**, **x'**)
and (**z'**,**x**)
Encoder: must learn
to map real image
samples to latent space

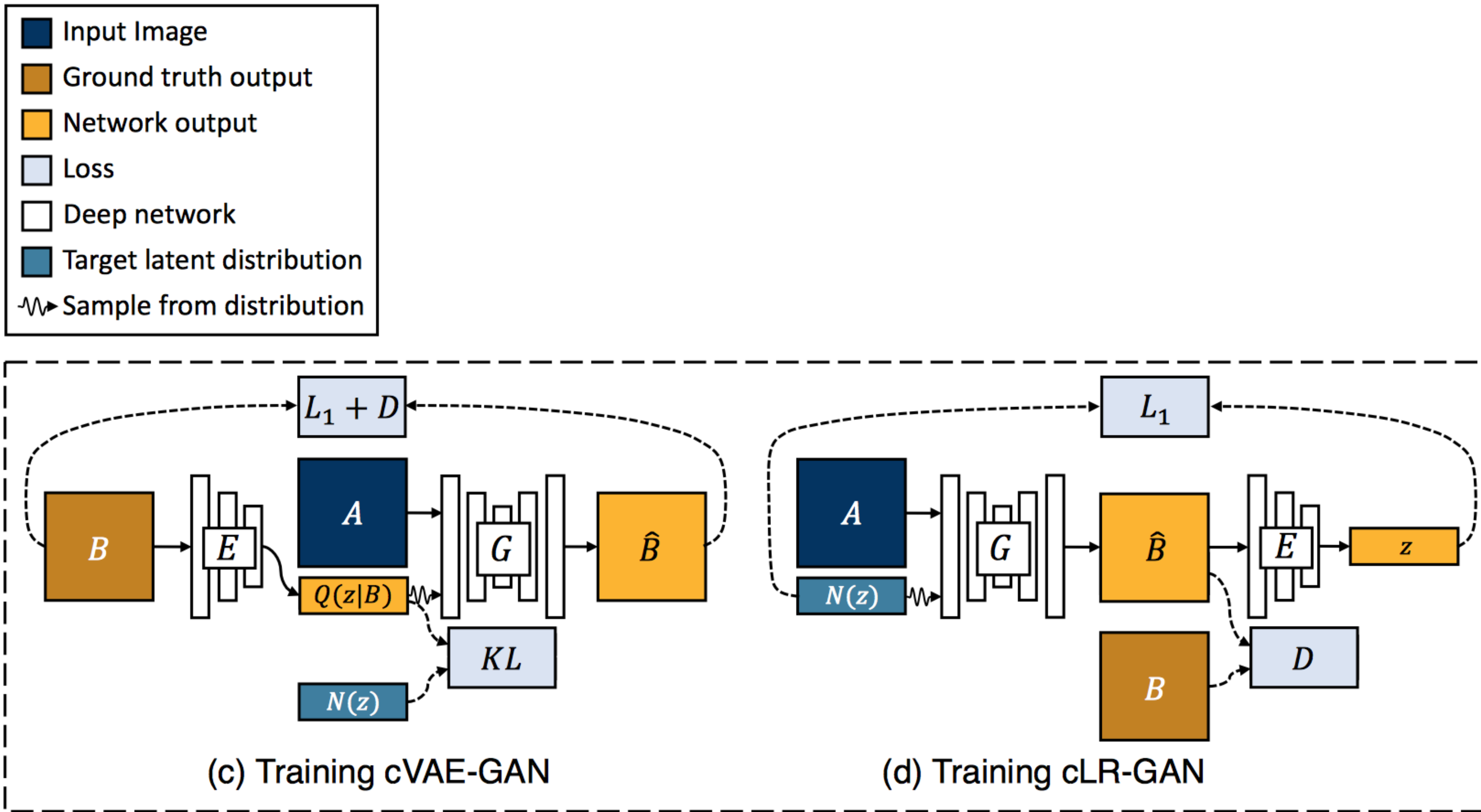
Cycle GAN



Cycle GAN

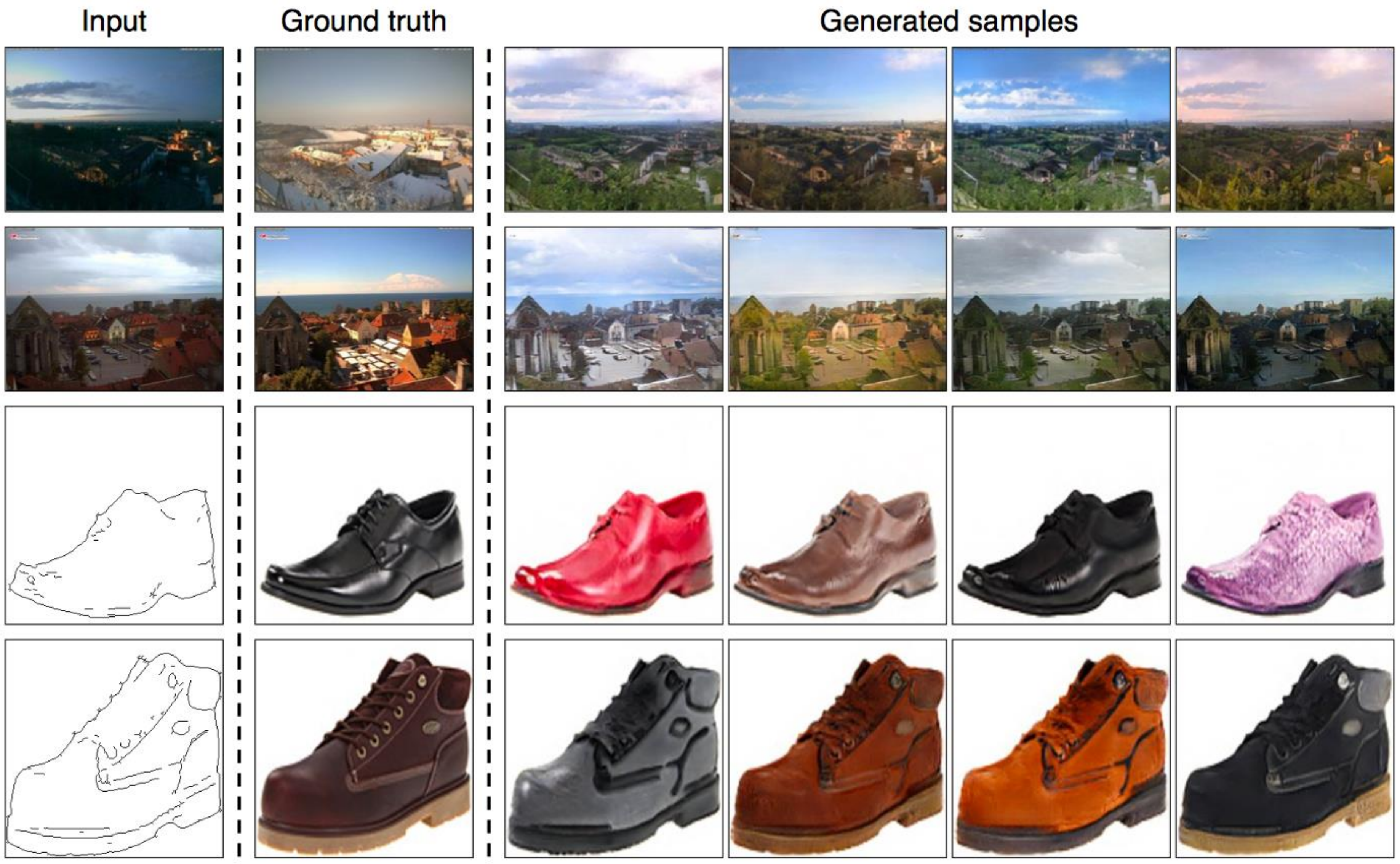

D_X
D_Y
G
F
X
Y
x
$\hat{Y}$
$\hat{x}$
y
$\hat{X}$
$\hat{y}$
cycle-consistency loss
cycle-consistency loss

BiCycle GAN



(c) Training cVAE-GAN

(d) Training cLR-GAN


Input
Ground truth
Generated samples

Questions?