GANS

look up:

adam – method for stochastic optimization

pseudo-label approach

VAE

CatGAN

infoGAN

iGAN

GANs for big imgs

=goal of the generator network is to map a random vector

to a realistic image (SIMGAN)

kinda deal with loss func prob of CNN (because kak, blurry imgs obv fake)

best on a limited domain (e.g. just faces) (SSLGAN)

not good at high res imgs (SSLGAN,GPGAN)

not good at big images (SSLGAN)

= known to be unstable and tends to introduce artifacts (SIMGAN)

= training GANs is difficult and

a perfect D is useless to a training G cos gradient of V(D,G)=0 (BIGAN)

often leads to oscillatory behavior, divergence, or modeling

only part of the data distribution. (PerceptSimilairty)

Analysis of learned representations is an important but largely unsolved problem (i.e. inverting them) (PerceptSimilairty)

**prob: how to eval Gans? (RGAN)**

**Nearest neighbors [euclid = kak for imgs]? Get humans to be D? Use them as classifiers? How about use 2 GANs and pit their Ds against their Gs and vice versa on the test data (Gen. Adversarial Metric). Visual Turing test/amazon mechanical turk**

Orig

**Generative model estimation via adversarial process (2player minimax game): train generative model G that captures the data dist, and discriminative model D that *estimates P(sample came from the training data rather than G).* G must max P(D makes a mistake). 1 solution = G recovers the training data and D = ½ everywhere. don’t need inference/markov chains. Only use backprop to get gradients**

G makes samples by passing rand noise through an MLP (D also = MLP)

most work had been = making a parametric specification of a prob dist func, then train by max’ing log likelihood (DBM), but likelihood funcs intractable. Thus try gen samples from the desired dist but without representing this likelihood

GANs can’t model discrete data (!)

D(x) = scalar = p(x came from data rather than G). Train g to min(log(1-D(G(z)) (or max(log(D(G(z) ) if early on because then G is so bad it’s obvious).

**alternate k steps optimizing D and then 1 of optimizing G to stop overfitting and to stop G collapsing too many vals of z to the same val of X (helvetica scenario)**

**G uses ReLu and sigmoid, D uses maxout activations. Only use noise as input to bottom layer**

estimate probability of the test set data under p g by fitting a Gaussian Parzen window to the

samples generated with G and reporting the log-likelihood under this distribution. The σ parameter

of the Gaussians was obtained by cross validation on the validation set

future: CGANs, semi-supervised GANs

Deep Conv GANs (DCGANN)

**use large amount of unlabled data to learn intermediate reps, then can use those on supervised tasks like img class. Do first step using GANs, then use their D and G for supervised tasks.**

**core features:**

**1. all-convolutional net, no deterministic spatial pooling (e.g. maxpool), but rather strided convolutions for D, fractional-strided convs for Gs.**

**2. no fully-connected layers on top.** Global avg pool => higher stability but slower convergence

**3. batch norm (not G output layer and not D input layer)** helps gradient flow, stops Helvetica

**4. relu all G layers except output tanh, leaky ReLu for all D layers, not maxout**

Other unsupervised representation learning techs = clustering (e.g. kmeans), auto-encoders (stacked/convolutionally), deep belief nets and ladder structures

generating natural images = parametric and non-parametric

original GAN images noisy and incomprehensible. Recurrent and Deconvolution network GANs made better images but didnt use their G and Ds for classification

LAPGAN = iteratively upscale lowres imgs

**future? Use for video**

Recurrent GAN(GRAN)

**Like LAPGAN, gen img in sequence, but not coarse-to-fine, let net learn struct *by itself***

**G = recurrent feedback loop that takes seq of noise and draws output at multiple t’s.**

Conditional GANS (CONDGANS)

**Direct D and G’s training with condition *y* (extra input layer to D and G)** like class labels/some part of the data for inpainting/data from a diff modality

hard to scale supervised Nns to be able to deal with klomp output categories, also often is a 1-to-1 mapping ,where instead should be probabilistic (cos some things can have many labels)

Stacked GAN (SGAN)

**Use top-down stack of GANs (learn to gen a lower lvl rep conditioned on a higher one)to make better imgs. Go from abstract representations to specific representations. i.e. does multi-lvl representations**

prob: Deep gen. Models bad when large variation in data dist

top-most GAN receives labels, bottom-most gens imgs

prob: challenging for GANs to generate diverse images with sufficient detail (hence LAPGAN)

tries to transfer knowledge in a bottom-up DNN to a gen. Model (and not another DNN)

matches intermediate representations with *adversarial* loss, not l2 loss

Wasserstein GAN (WGAN) DONT understand

GANs offer much more flexibility in the definition of the objective function than VAE

Training GANs is well known for being del-icate and unstable,

Simul/Synth GAN(simulGAN)

**Improve synthetic img quality using ‘refiner’ network (uses unlabeled real data) trained with a GAN**

**inputs =imgs, NOT random noise vectors**

**use a combo of adversarial and self-regularization loss to add realism**

**uses local discrimination rather than global. Uses discriminator network that classifies all local image patches separately**

***discriminator updated using history of refined imgs. This memory ensures it doesn’t forget what makes a fake img fake***

training a deep neural network on synthetic data leads to improved performance

instead of using real img datasets (hard to annotate), use synthetic imgs that have automatic annotations. But what about gap btwn simulated imgs and real imgs? V hard and expensive to up their realism quality.

C-RNN-GAN (CRGAN)

**can we use a GAN for continuous sequential data?**

**Basically, have D and G are RNNs (use music). D = bi-directional LSTM**

Semi-Supervised learning w/ Context-Condition GAN (SSLGAN) [img inpainting]

**train a GAN’s G to fill in missing patches of imgs. D checks if filled-in img legit. Thus by doing this we get a “regularizer for standard supervised training of the discriminator”. This is cos D learns features that can be generalized to classifying objs**

A common approach to semi-supervised learning is to combine a supervised and unsupervised ob-

jective during training. As a result unlabeled data can be leveraged to aid the supervised task.

GAN discriminator learns a hierarchical image representation that is useful for object classification.

**Rather than determining if a full image is real or fake, context conditional GANs address a different task: determining if a part of an image is real or fake giventhe surrounding context.**

Img Blending Gaussian Poisson GAN (GPGAN)

**use GP and GANs to get high res img blending (CtrlC ctrlV img blending look less bad)**

uses gradient-based image blending methods (GP) to blend the colours better in composited imgs

“We find that a network with only convolutional layers could not learn to blend composited images

for the lack of global information across the whole imagewhich is essential for image blending task. **This suggests that standard fully connected layers are necessary for conditional image generation.**

Super Resolution GAN (SRGAN)

**perceptual loss func = adversarial and content loss (based on perceptual similarity and not pixel similarity)**

**Use ResNet skip-connection and diverge from MSE as the sole optimization target.**

super res = hard. Most work done on min(MSreconstructionE). But this isn’t enough, still don’t have hi freq details, although min(MSRE) does min(peak signal to noise ratio) (PSNR used to eval)

but PSNR and MSRE can’t do perceptually-relevant details cos at the pixel level

Perceptual Similarity Img Gen (PerceptSimilarity)

img gen models usally use dist-based loss functions -> over-smoothed results

Euclid squared dist btwn imgs = > blurry imgss

rather, **compute dists btwn img features extracted by DNNs. DeepSim = feature loss + pixel space loss + adversarial loss**

CNNs good, cos invariant to small smooth deformations but sensitive to perceptually important image properties, for example sharp edges and textures.

The most popular perceptual image similarity metric is the structural similar-

ity metric (SSIM) (Wang et al., 2004), which compares the local statistics of image patches

Face aging with CGANS (faceaging)

**use condit GAN with additional identity preserving feature**

auto-encoders optimize l2 reconstruction loss -> blurry imgs

Contrary to autoencoders, cGANs do not have an explicit mechanism for inverse mapping of an input image x with attributes y to a latent vector z which is necessary for image econstruction: x = G(z, y). ->train encoder that’s a neural network which approximates the inverse mapping

Perceptual GANs for small obj detection (smallObjDetection)

G= transforms orig, poor features of small objs to highly discriminative ones by intro’ing fine-grained deets from lower layers (thus intermediate reps are in “super res”)

D = not just usual task but also must justify increase in detection accuracy

**hard to detect small objs due to low resolution and noisy representation**

**narrow representation diff of small objs from big ones**

CGANs for img translation (image-to-image)

**a patchgan – D tries to determine if each NxN patch in a img is fake**

image translation = translating one possible rep of a scene into another, given enough training data

CGANs learn a ‘structured loss’ (each pixel not indep of all others)

restrict the GAN discriminator to only model high-frequency structure, relying on an L1 term to

force low-frequency correctness

Bidirectional GANs (adversarialfeaturelearning)

**Use encoder E which maps data x to latent representations z (gives a label ‘for free’)**

**GANs have no way of doing inverse mapping – of projecting data back into the latent space**

Guarantee that at the global optimum, G and E are each other’s inverse

BiGANs not domain sensitive, needn’t suffer from domain shift between

the pre-training task and the transfer task

InfoGAN (infoGAN)

**encourages GAN to learn interpretable and meaningful representations by maximizing the mutual information between a fixed small subset of the GAN’s noise variables and the observations**

**replaces noise vector with 2 parts:**

**1. z, an incompressible noise vector**

**2. the ‘latent code’ (set of variables) e.g. c1 = MNIST digit type, big c2 = wide no., small c2 = thin no.**

**Must be high mutual information between latent codes c and generator distribution G(z, c)**

representation learning goal = use unlabelled data to learn a representation that exposes important semantic features as easily decodable factors (and useful for later tasks like classification)

**an unsupervised learning algorithm must in effect correctly guess the likely set of downstream classification tasks without being directly exposed to them**

BUT cos don’t know later tasks when training, a *disentangled representation*, one which explicitly represents the salient attributes of a data instance, should be helpful for the relevant but unknown tasks (e.g. if training on faces, should have separate sets of dimensions for eye colour and hairstyle)

can scale to big datasets and trains in same amount of time

**GANs dont restrict how G uses the noise, thus G could use it in ‘entangled’ way**

**future? learning hierarchical latent representations, improving semi-supervised learning with better codes , and using InfoGAN as a high-dimensional data discovery tool.**

iGAN (iGAN)

**Use GANs to learn manifold of natural imgs, then use manifold to constrain img manipulations by ppl**

**why GANs? Cos they make good imgs and “the** **Euclidean distance in the latent space often corresponds to a perceptually meaningful visual similarity”**

approach:

1. project normal photo onto approximation of img manifold by finding the closest latent feature vector z of the GAN to the original image

2. update latent vector z in real time so that img good for user and *stays on manifold*

super hard to edit imgs unless you’re a pro (i.e. hard to change them without making them look fake)

classic visual manipulation paradigm does not prevent the user from “falling off” the manifold of natural images

GANs make imgs by sampling latent vector space at random thus not able to create and/or manipulate visual content in a user-controlled fashion

**future? Not just shape , colour but also texture and other structure changes**

CatGAN (CATGAN)

**Get a D that maximize mutual information between the inputs x and the labels y (as predicted through the conditional distribution p(y|x, D)) for a number of K unknown categories. G tries to get D to accept bogus inputs**

assumes uniform dist over K

normal GAN can’t do the job cos D just says “okay, it’s from the data”

**Instead of asking D to predict the probability of x belonging to X we can require D to assign all examples to one of K categories (or classes), while staying uncertain of class assignments for samples from the generative model G**

**D: . must (i) be certain of class assignment for samples from D, (ii) be uncertain of assignment for generated samples, and (iii) use all classes equally 3**

**G : must generate samples with highly certain class assignments, and (ii) equally distribute samples across all K classes**.

how to learn classifiers from partially/unlabeled data?

Clustering = generative (kmeans) [try to model the dist p(x)] / discriminative [try to put data into categories without modelling p(x)]