Denoising with Generative Models

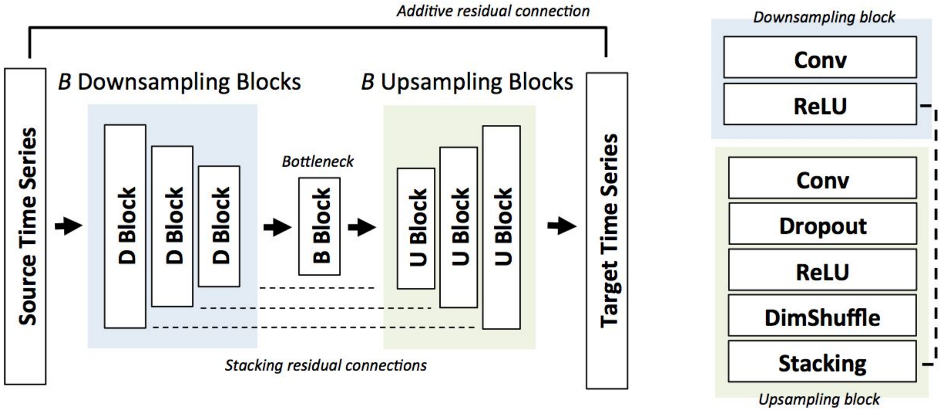
Loïs Bilat

MA3 IN - Fall 2019 - Semester Project - 12 ECTS

Supervisor : Brian Sifriger

Reminder

- Low quality => high quality audio files
- Super-resolution, denoising and dereverberation
- Before midterm : Bottleneck architecture (*Kuleshov*, *Enam*, *Ermon* : **Audio Super Resolution with Neural Networks**, 2017)



What’s new ?

- Scheduler
- Improved Architecture
  - Discriminator network
  - Auto-encoder network
- Collaborative GAN

Scheduler

- When loss reaches a plateau, decrease learning rate
- Many parameters
  - What is a “plateau” ?
  - By how much do we decrease ?
  - ...

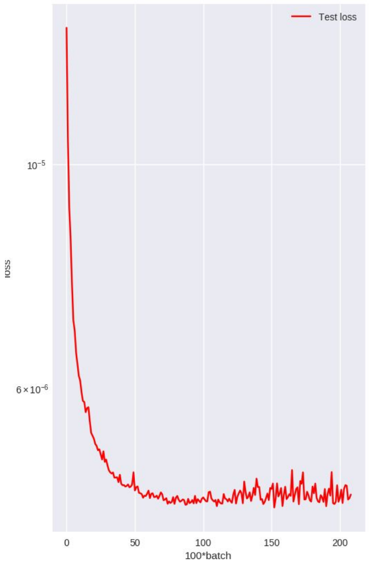
Scheduler

Metric : LSD (Log-spectral distance), lower = better

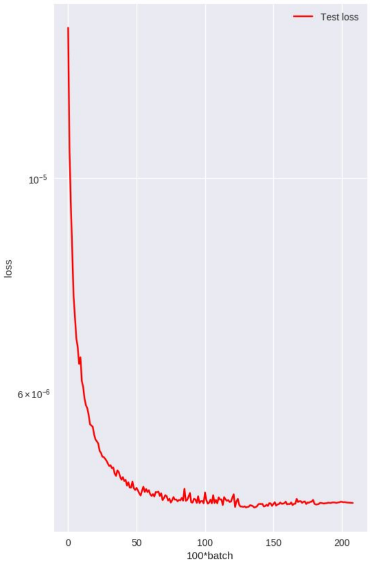
(Difference in the frequency space)

$$\mathrm{LSD}\left(X, X_{ref}\right) = \frac{1}{W}\sum_{w=1}^{W}\sqrt{\frac{1}{K}\sum_{k=1}^{N}\left(\log_{10}\frac{|X(w,k)|^2}{|X_{ref}(w,k)|^2}\right)^2}$$

$\mathrm{LSD}(X_{low},X_{high})$	$\mathrm{LSD}(X_{improved},X_{high})$	$\mathrm{LSD}(X_{improved+scheduler},X_{high})$
2.2235	**1.6079**	1.6779



no scheduler



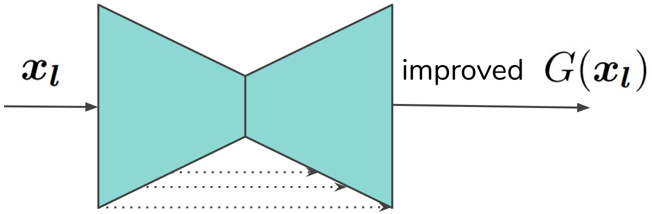
scheduler

Original Network

Original high quality $\boldsymbol{x_h}$

low quality $\boldsymbol{x_l}$

Generator



$$\mathcal{L}_{L2} = \frac{1}{W} \sum_{i=1}^{W} \| \boldsymbol{x_{h,i}} - G(\boldsymbol{x_l})_i \|_2^2$$

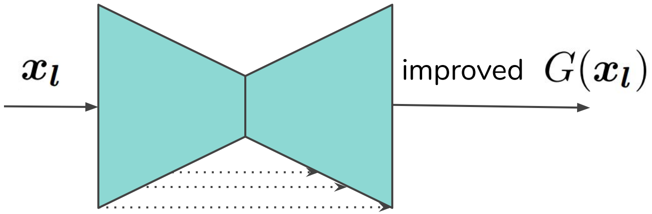
$$\mathcal{L}_G = \mathcal{L}_{L2}$$

Discriminator Network

Original high quality $\boldsymbol{x_h}$

low quality $\boldsymbol{x_l}$

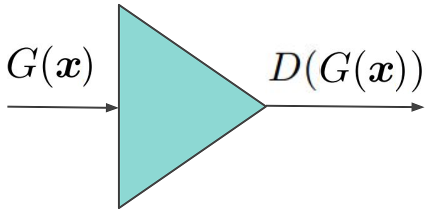
Generator



$$\mathcal{L}_{L2} = \frac{1}{W} \sum_{i=1}^{W} \|\boldsymbol{x}_{\boldsymbol{h},i} - G(\boldsymbol{x_l})_i\|_2^2$$

$$\mathcal{L}_G = \mathcal{L}_{L2} + \lambda_{adv}\mathcal{L}_{adv}$$

Discriminator



real or *fake* high quality audio ?

Train using $\boldsymbol{x_h}$ -> real

$G(\boldsymbol{x_l})$ -> fake

Binary Cross-entropy loss

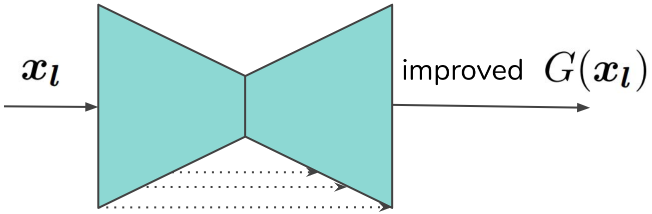
$$\mathcal{L}_{adv} = -\log D(G(\boldsymbol{x_l}))$$

Auto-encoder network

Original high quality $\boldsymbol{x_h}$

low quality $\boldsymbol{x_l}$

Generator

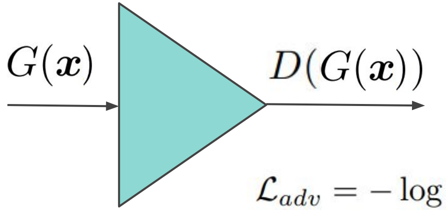


$$\mathcal{L}_{L2} = \frac{1}{W}\sum_{i=1}^{W} \|\boldsymbol{x_{h,i}} - G(\boldsymbol{x_l})_i\|_2^2$$

$$\mathcal{L}_{adv} = -\log D(G(\boldsymbol{x_l}))$$

$$\mathcal{L}_G = \mathcal{L}_{L2} + \lambda_f \mathcal{L}_f + \lambda_{adv}\mathcal{L}_{adv}$$

Discriminator

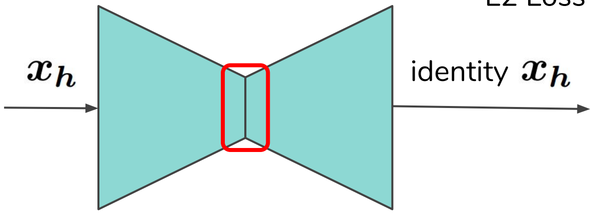


real or *fake* high quality audio ?

$$\mathcal{L}_{adv} = -\log D(G(\boldsymbol{x_l}))$$

Auto-encoder

Train using $\boldsymbol{x_h}$ -> $\boldsymbol{x_h}$
L2 Loss

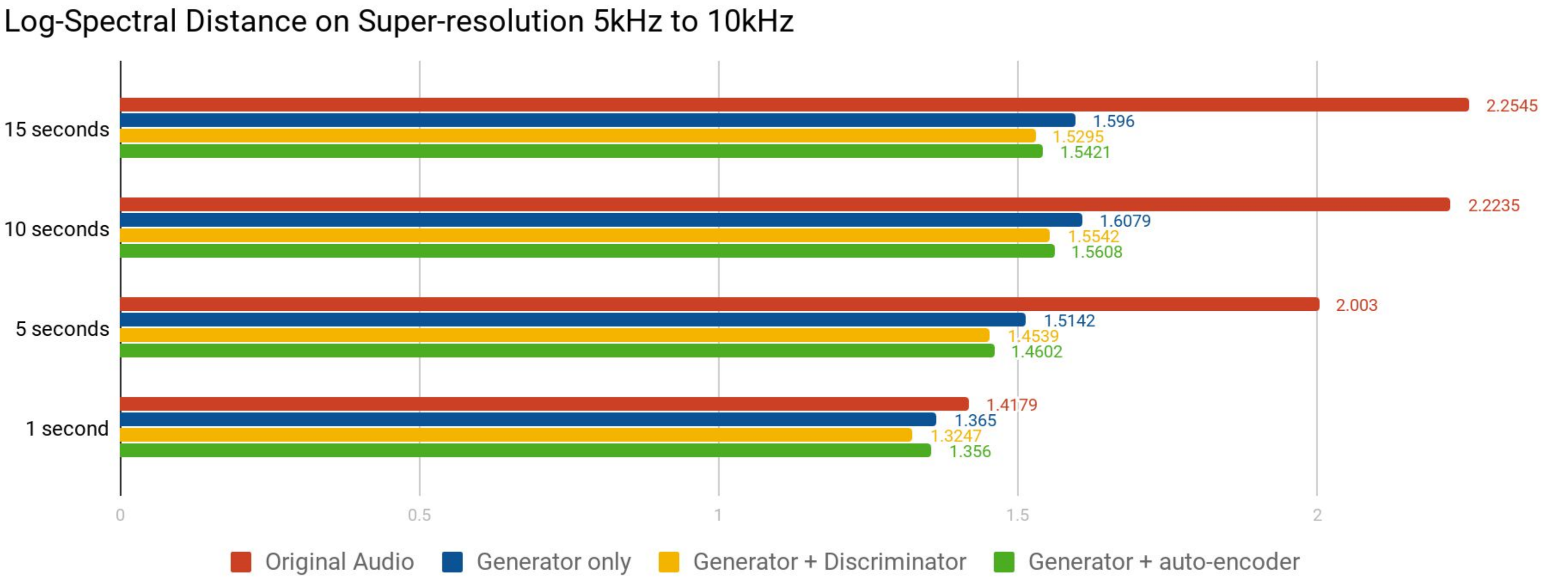


$$\mathcal{L}_f = \frac{1}{C_f W_f}\sum_{c=1}^{C_f}\sum_{i=1}^{W_f} \|\phi(\boldsymbol{x_h})_{i,c} - \phi(G(\boldsymbol{x_l}))_{i,c}\|_2^2$$

Improved Architecture

- Fully modular :
  - Discriminator, auto-encoder, both or none
- Lots of parameters to tune
  - Tune the lambdas $\mathcal{L}_G = \mathcal{L}_{L2} + \lambda_f \mathcal{L}_f + \lambda_{adv} \mathcal{L}_{adv}$
  - When do we use composite loss ? (only start when discriminator/autoencoder are good enough)
  - 3x hyperparameters (learning rate, momentum, decay, dropout, ...)
  - => hard to find the best values

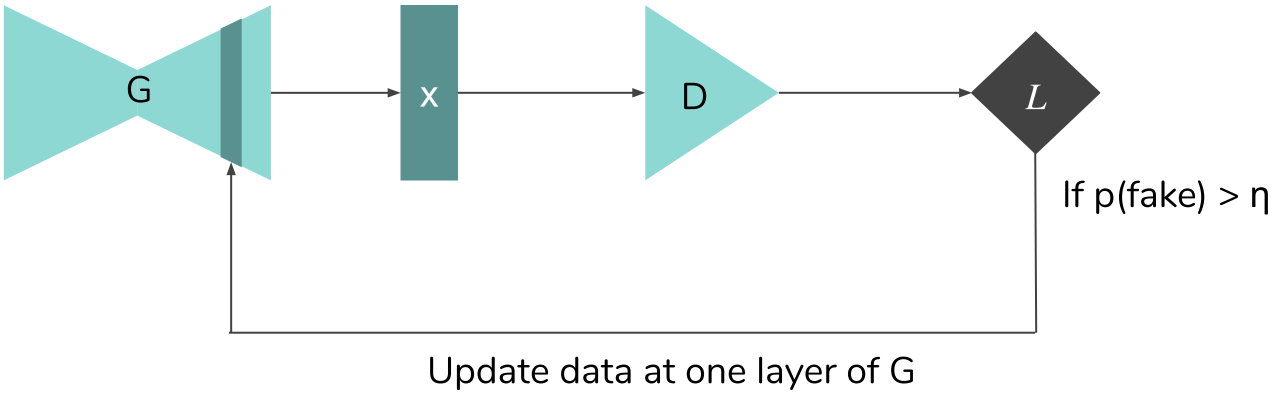
Results



- Discriminator and auto-encoder useful
- Can try Auto-Encoder + Discriminator together

Collaborative Gan

- Training as usual
- At the end, “refine” the samples
- Repeat while not good enough



Not working yet ...

Demo

Original

Super-resolution

Denoising

Conclusion

- Complex architecture with many features
- Metric shows improvement, not obvious when listening
- Hyperparameter tuning required
- Running on the workstation complicated
    - meanwhile, report started : vita.bilat.xyz