Generative Models for Aircrafts Icing Image Prediction

12/28/23

李云艾

Content

- **Background**
- **Variational Autoencoder Model and its Modification**

---model

---results

- **Generative Adversial Neural Network and Wasserstein GAN Model**

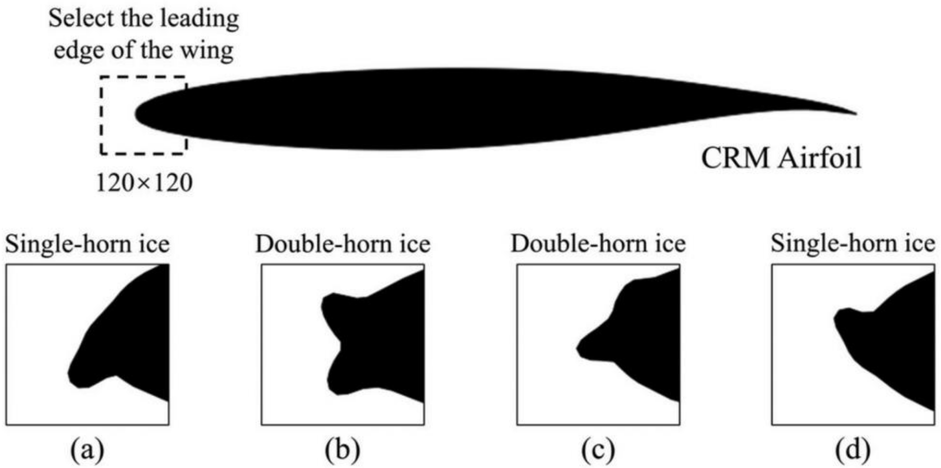
---model

---results

Background

- icing image:

wing (fixed shape) + ice (to predict under certain icing conditions)



Icing condition
MVD (medium volume diameter)(μm)
LWC (liquid water content)/(g/m^3)
Temperature (°C)
Icing time (s)
Velocity (m/s)
AOA (angle of attack) (deg)
Chord length (inch)

low dim input (7) --→ high dim out put (142*142) (categories unknown)

A Common Way to Process the Input Data

- Inspired by Conditional GAN

icing condition (dimension incresed by NN)

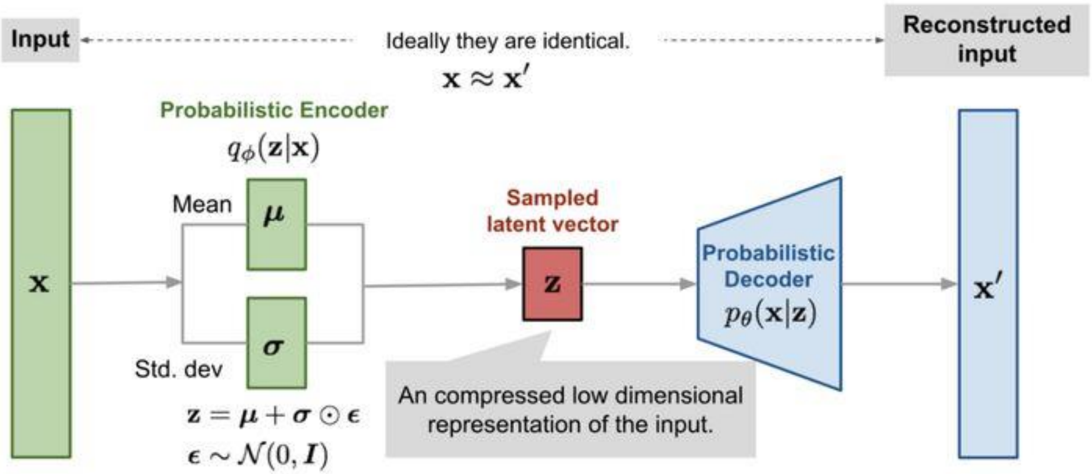
+

original wing image without ice (dimension decresed)

to (40*40)

Original VAE Model

- **basic model stucture:**



parameters:

input&output: 142*142 1-channel image

latent dimension: 1024

- **loss**=KL-divergence +WeightedFocal_loss
- **KL-divergence** (check next slide)
- **WeightedFocal_loss**= (reduces the relative loss for well-classified examples (p>0.5), putting more focus on hard, misclassified examples)

- problem: vanishing KL loss (Posterior Collapse)

Modified Model : VAE + BN

- Problems : the KL between can be computed as:

When the decoder is autoregressive, it can recover the data independent of the latent z (only by noises). The optimization will encourage the approximate posterior to approach the prior which results to the zero value of the K.

- Solutions: 1. Avoid posterior collapse by keeping the expectation
- of the KL' s distribution positive. The batch normalization to the parameters ensure a positivelower bound of this expectation.
- 2. Cost anneling : Gradually increase weight on the Kl divergence term

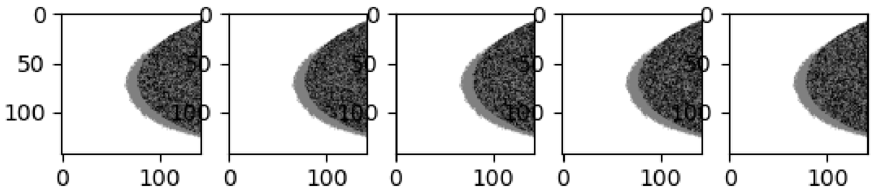
Qile Zhu, Jianlin Su *A Batch Normalized Inference Network Keeps the KL Vanishing Away*

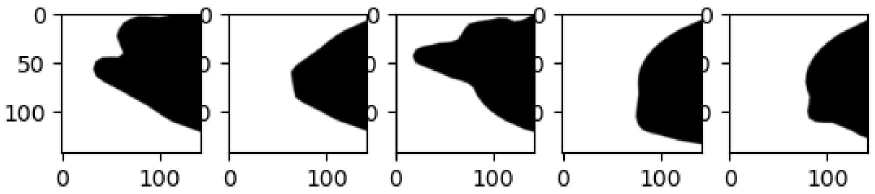
Bowman et al. *Generating sentences from a continuous space.* CONLL 2016.

Results

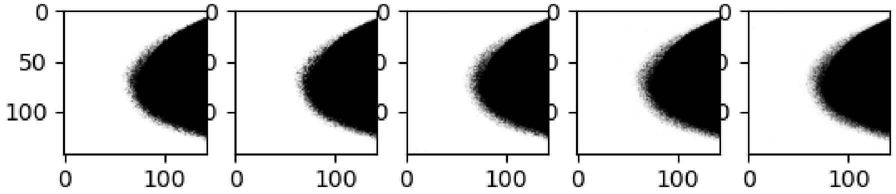
×No clear differences between generated images

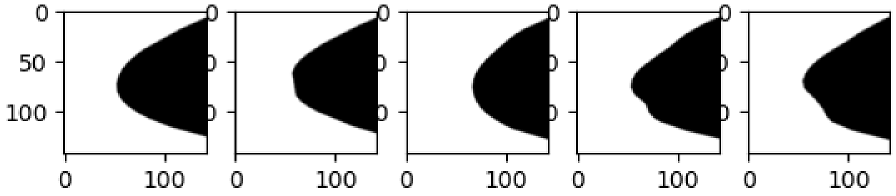
(The model still doesn't work better after putting a weighted mask on the non-wing area （as below））





×Poor performance on predicting irregular icing images

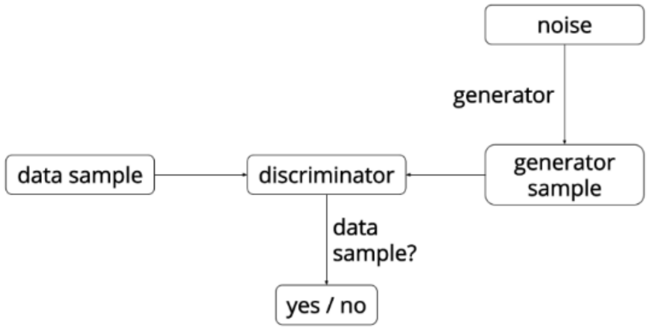




PSNR	MSE
14.49	0.035
15.30	0.029
18.34	0.014
15.42	0.028
15.46	0.028

Average PSNR ：
14
（batch_size=64）

Generative Adversial Neural Network



- Basic model stucture:
- loss function:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

- for fixed generator, ,to learn to classify fake images D(G(z))=0 and real images (D(x)=0)
- for fixed discriminator, to train generator model that makes D's output 0
- Problem: discriminator_loss ↓ (and vanishes) generator_loss ↑ (fail to generate images that could be mistaken for real data)

Solutions:

--- add Gauss noise to better train the discriminator (failed)

--- use wgan

Introduction to Wasserstein GAN

- classification(discriminator) ➡ regression("critic")
- **Modifications Details**

---**optimizer** momentum method × Adam ➡ RMSprop (unstable loss gradient)

---no sigmoid before output (no longer a 0-1 classifier)

---a **clip** on parameters to ensure Lipschitz Continuous (for the convergence)

--- **no log term** in the loss function : (JS divergence ➡ EM/Wasserstein distance)

$$W(P_r, P_g) = \inf_{\gamma \sim \Pi(P_r, P_g)} \mathbb{E}_{(x,y)\sim\gamma}[||x - y||]$$

$$K \cdot W(P_r, P_g) \approx \max_{w: |f_w|_L \leq K} \mathbb{E}_{x\sim P_r}[f_w(x)] - \mathbb{E}_{x\sim P_g}[f_w(x)]$$

(ICML 2017) Martin Arjovsky , *Towards Principled Methods for Training Generative Adversarial Networks*

Principle behind Wasserstein GAN

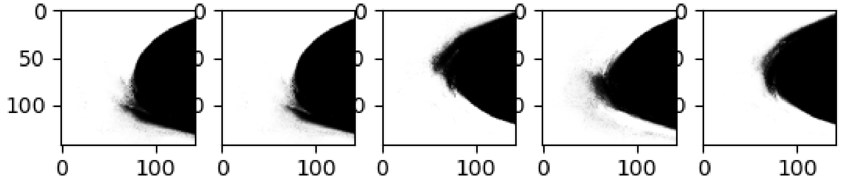
- Original Loss: $-P_r(x)\log D(x) - P_g(x)\log[1 - D(x)]$
- Let its derivative with respect to D be 0: $D^*(x) = \frac{P_r(x)}{P_r(x)+P_g(x)}$

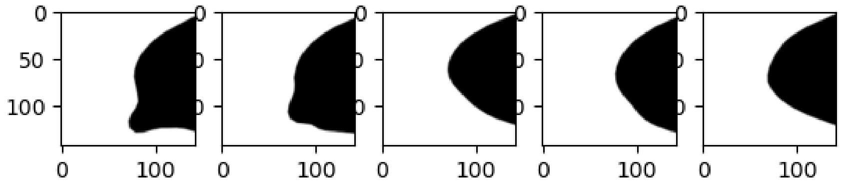
$$\mathbb{E}_{x\sim P_r} \log \frac{P_r(x)}{\frac{1}{2}[P_r(x)+P_g(x)]} + \mathbb{E}_{x\sim P_g} \log \frac{P_g(x)}{\frac{1}{2}[P_r(x)+P_g(x)]} - 2\log 2$$

- The generator loss defined by the original GAN is equivalent to minimizing the JS divergence between the true distribution and the generated By simple calculation it is easy to see If the two distributions have no overlapping parts at all, or their overlapping parts are negligible, the loss is a const.
- When the support set is a low-dimensional manifold in a high-dimensional space, the
- the probability that the overlapping part of the measure is 0 is 1
- This eventually leads to a (approximate) gradient of 0 for the generator and the gradient vanishes.

Results

Still, it's hard to train a balanced Adversial Model





√ Some details (horns) is predicted

× unsmooth , blur on the boundary

× low average PSNR

PSNR	MSE
15.97	0.025
20.56	0.008
14.26	0.037
12.47	0.056
17.74	0.017

average of a batch ： 15

Some other possible Modifications

- Classify the dataset

- Icing condition data pre-processing

- Other methods to put weight on the icing area