DL hw3. 姓名:許庭恩 學號:111321514

cVAE

---

cVAE Cifar10:

USE PyTorch

Use Early-Stop to avoid overfitting and keep the best model. (patience=20)

- **Dataset analysis:**

  Cifar10 has 10 classes, which size are 32*32 RGB channel images.

  And these images' categories are 'airplane', 'automobile', 'bird', 'cat', 'deer', 'dog', 'frog', 'horse', 'ship', 'truck'. The training data contain 60000 images and test data contain 10000 images.

- **Dataset preprocess & Metric calculate:**

  In the VAE models, we do not need split train data into train and valid. So, our training data are origin size 60000.

  To calculate the FID、IS performance metric, we need to use the InceoptionV3 model pre-train weight and normalize the same range with std and mean.

  The InceoptionV3 input size original is 299*299. So, when we input the images to calculate our FID and IS score need to resize into 299*299. But if we use this size to evaluate per epoch in our data, we need more GPU VRAM and Times. Thus we only calculate the performance metrics in the per different architecture final model.

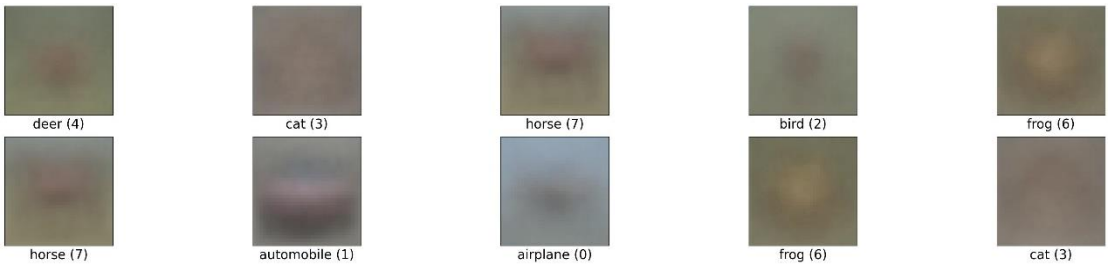
  The FID and IS score is not like accuracy、precision, and recall…intuitive. So, we save the reconstructed images per train epoch to let us know whether the VAE is work perfectly.

Experiment

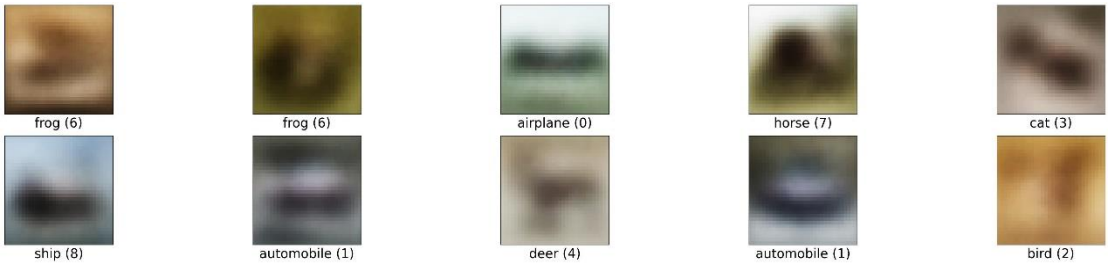
The FID and IS score was calculated by models which generate 1000 images per class.

- **LOSS different**

  The loss function is the core of the whole neural network. In the VAE the loss needs to "sum", rather like some classification、segmentation tasks, they usually use "mean" loss. Why did we say that? Because when we use "mean" the generated images fail always. We can see it is very different. That shows some pictures.



Loss use “mean”



Loss use “sum”

Learning rate: 0.0001
Optimizer: Adam
Batch size: 512

- **cVAE with MLP**

  In conditional VAE based on MLP, we use concatenating the label to the MLP tail. We make the label into a one-hot code. That means if the label is 8, the looks like [0,0,0,0,0,0,0,0,1,0].

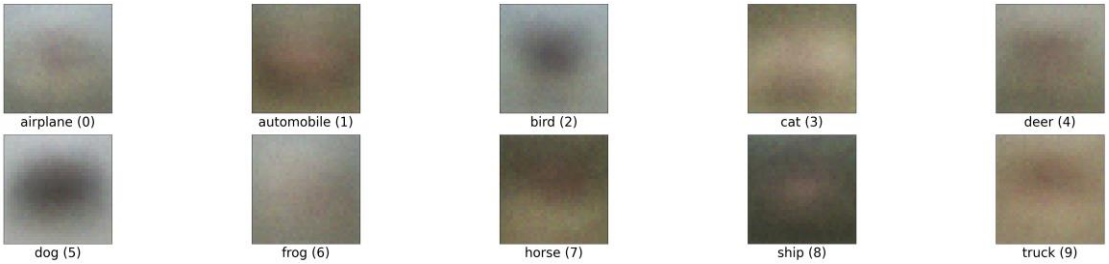
  EX:

  MLP shape=[3072], label=8 => MLP + label = [3082]

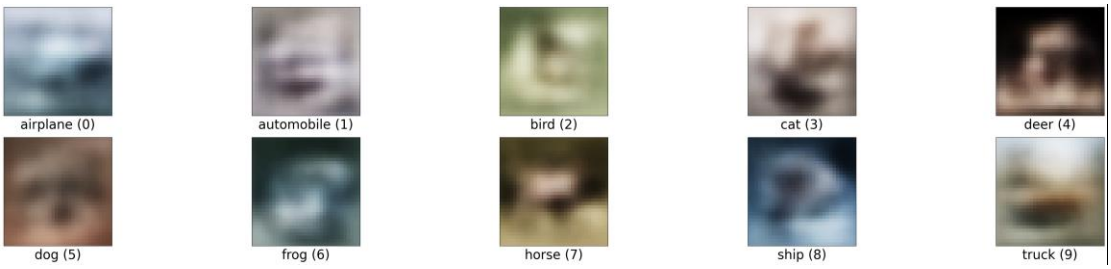
MLP architecture

	M1	M2	M3	M4
MLP1	512	1024		
MLP2	256	512		
MLP3	128	256		
MLP4	x	128		
Latent z	64	128	196	256
params	3,469,440	7,720,192	7,772,552	7,818,752
FID score	362.6826	255.4329	254.7927	252.6307
IS score	1.2501	1.5703	1.557	1.5569

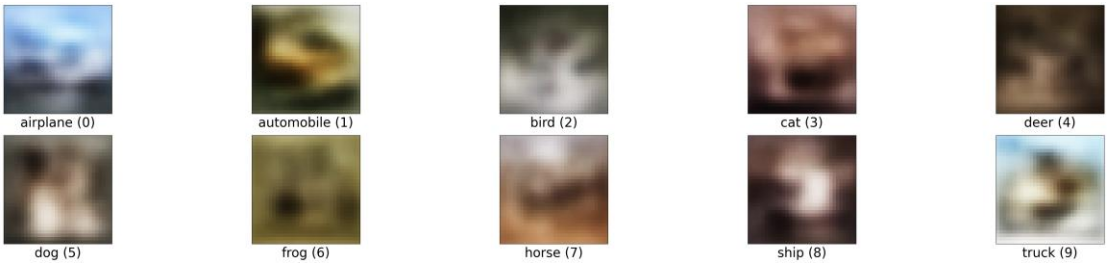
These images were generated by Gaussian sample lanten_z



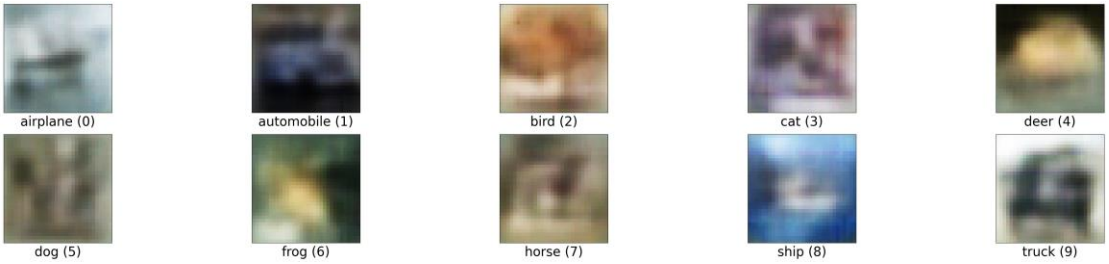
M1



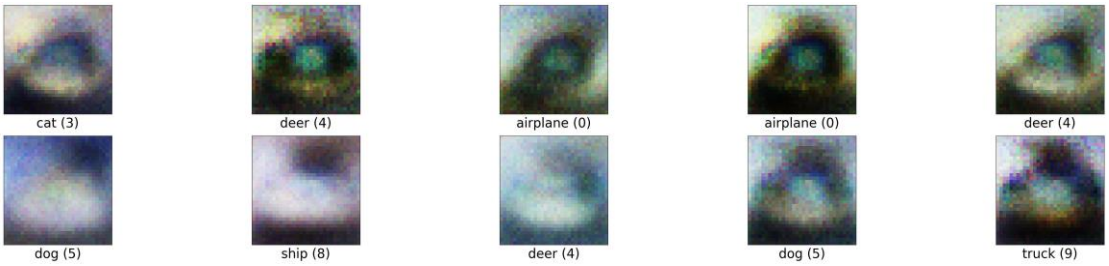
M2



M3



M4



Train too much may lead the model to collapse

- **cVAE with CNN**

  In conditional VAE based on CNN, we use concatenating the label to a channel. We make this channel all pixel values like label number. That means if the label is 8, the whole channel pixel values are 8.

  EX:

  Images shape=[C,H,W] = [3,32,32], label=8 => reshape label=[1,32,32] and all value=8, than images + label [4,32,32], so our encoder input is 4 channel.
  Conv :[in_channel, out_channel, kernel]

  Because there are too many images, so in CNN we only show the best architecture and weight generate images.

CNN architecture 1

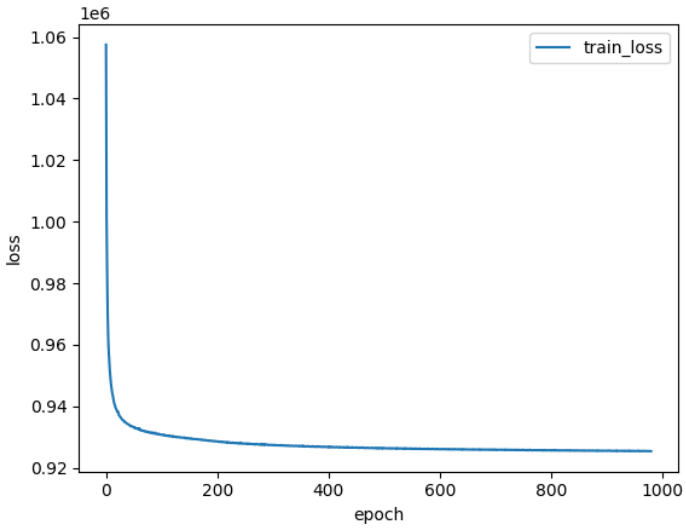
	M1	M2	M3	M4
1Conv	[4,16,5]			
2Conv	[16,32,5]			
MLP1	256			
Latent z	64	128	196	256
params	491,366	540,646	593,006	639,206
FID score	185.046	193.0256	187.7394	183.1079
IS score	1.5905	1.5923	1.5970	1.6085

CNN architecture 2

	M1	M2	M3	M4
1Conv	[4,32,3]			
2Conv	[32,64,3]			
3Conv	[64,128,3]			
MLP1	512			
Latent z	64	128	196	256
params	5,014,582	5,113,194	5,217,778	5,310,058
FID	184.7101	223.3839	186.8628	203.9175
IS	1.5997	1.5489	1.672	1.5636

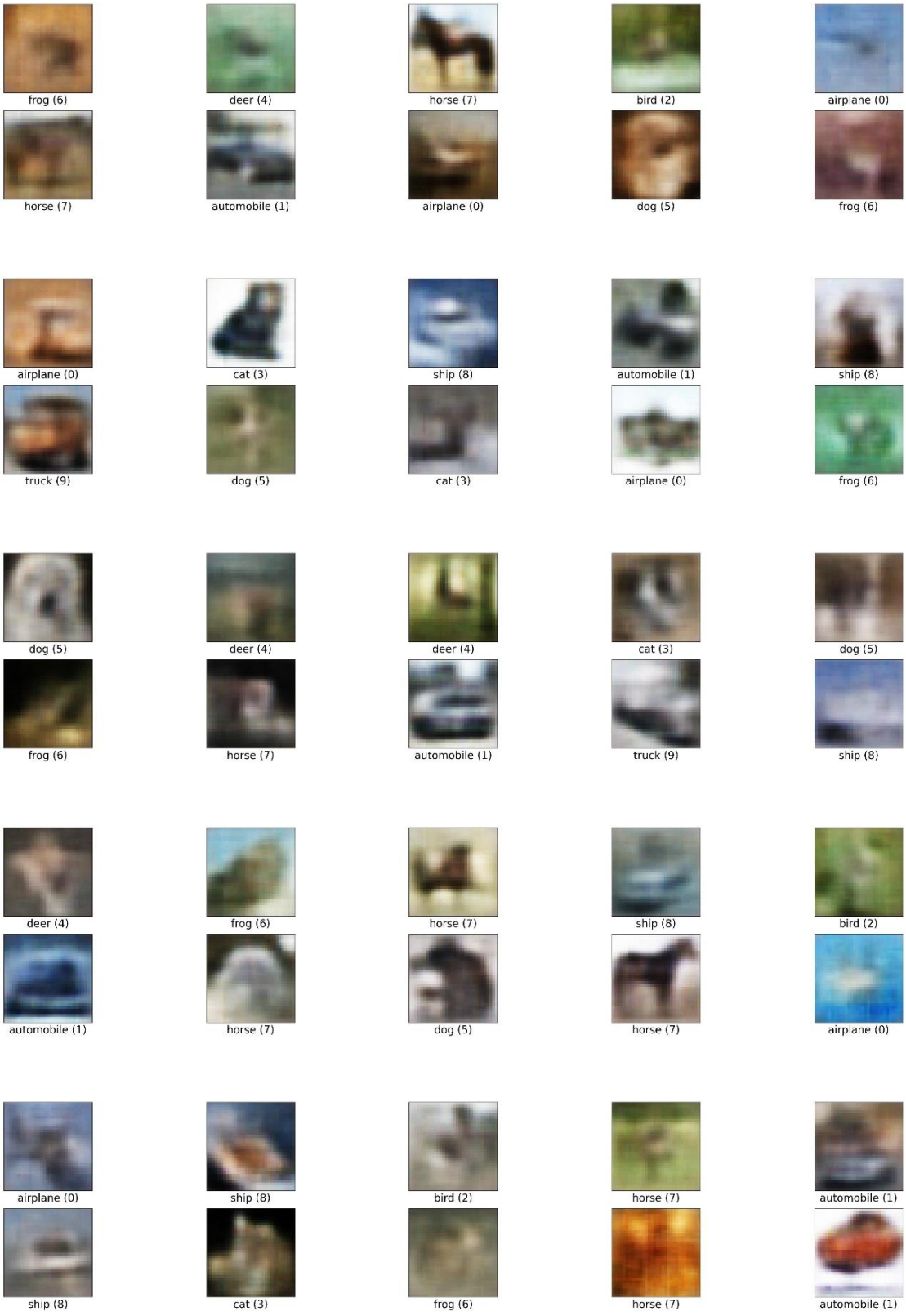
CNN architecture 3

	M1	M2	M3	M4
1Conv	[4,64,3]			
2Conv	[64,128,3]			
3Conv	[128,256,3]			
MLP1	512			
Latent z	64	128	196	256
params	10,293,258	10,391,690	10,496,274	10,496,274
FID score	220.0153	186.2041	182.441	186.8723
IS score	1.5762	1.7423	1.7436	1.7335

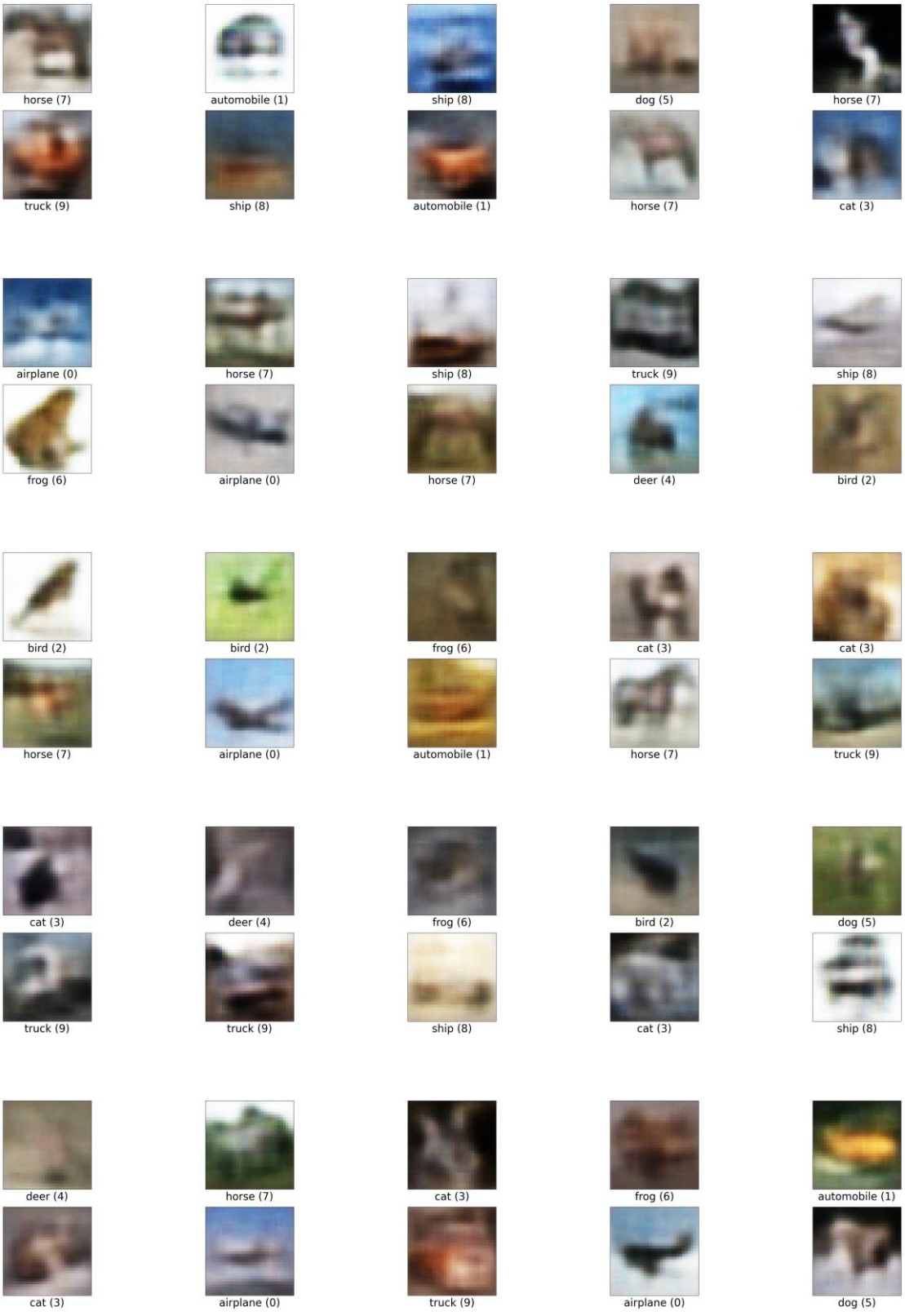


CNN architecture 3 train loss

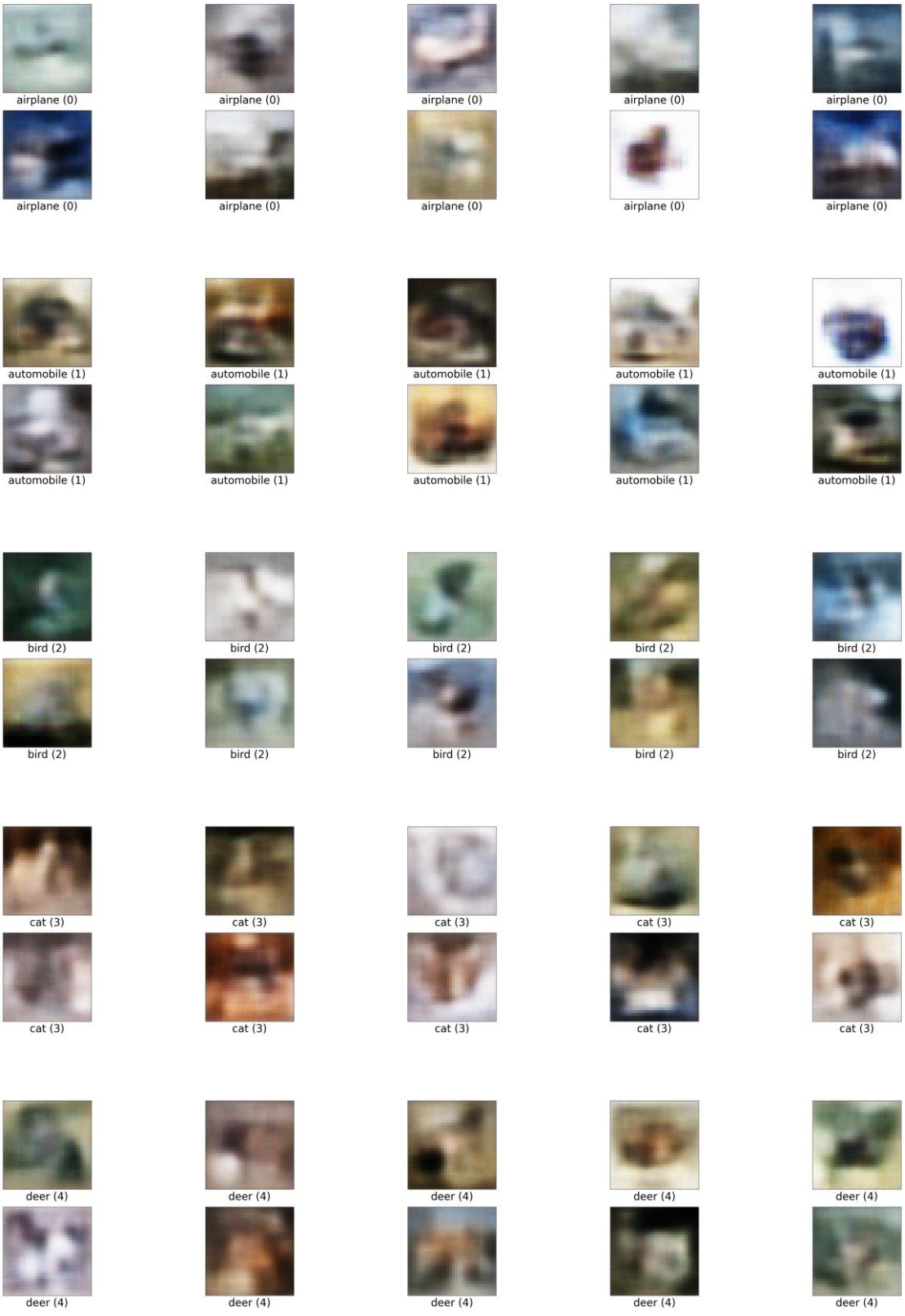
Let us show the CNN architecture 3-M3 model reconstructing images from the Cifar10 train.

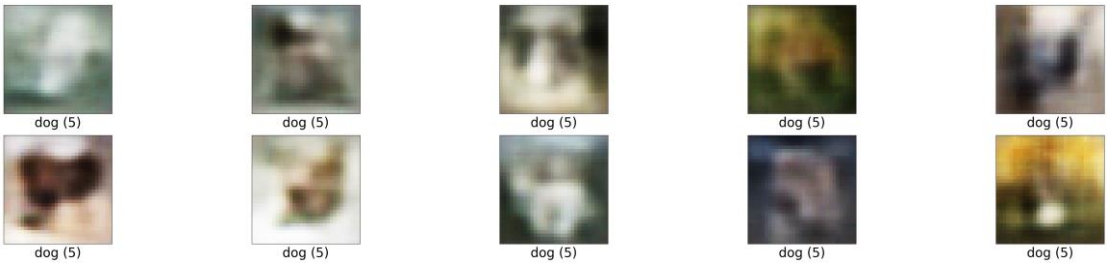


Let us show the CNN architecture 3-M3 model reconstructing images from the Cifar10 test.



Let us show the CNN architecture 3-M3 model reconstructing images from the Gaussian sample lanten_z.




dog (5)
dog (5)
dog (5)
dog (5)
dog (5)
dog (5)
dog (5)
dog (5)
dog (5)
dog (5)
frog (6)
frog (6)
frog (6)
frog (6)
frog (6)
frog (6)
frog (6)
frog (6)
frog (6)
frog (6)
horse (7)
horse (7)
horse (7)
horse (7)
horse (7)
horse (7)
horse (7)
horse (7)
horse (7)
horse (7)
ship (8)
ship (8)
ship (8)
ship (8)
ship (8)
ship (8)
ship (8)
ship (8)
ship (8)
ship (8)
truck (9)
truck (9)
truck (9)
truck (9)
truck (9)
truck (9)
truck (9)
truck (9)
truck (9)
truck (9)

- **Conclusion**

In the experiment, we found the BCE+KL divergence loss needs to sum, otherwise, the reconstructing image will be very blurry, and how every adjust training epoch will not be improved.

And if the learning rate is up to 0.001 the model will easy to collapse when training more than 100 epochs. We try a lot of learning rates, including schedule strategy. We finally define the learning rate at 0.0001. And it trains more stable and good performance.

The MLP & CNN architecture per model we train averages 800 epochs, and the FID and IS score shows that CNN architecture's performance is better. In the CNN, most of the models can arrive at the FID value of around 180, but the MLP model is around 250.

Because the Cifar10 images are very small, the convolution needs more careful, if the kernel size is too big, the image cannot convolution many times. And if the padding is too much, they are having too many fake values. So, based on our experiment, CNN's best architecture is using 3 convolutions, which kernel sizes are 3, filter numbers 64, 128, and 256, and Liner MLP is 512.

Finally, in image reconstruction visualization, we can see at the training step, the horse, automobile, and truck are very good. But the deer, bird, cat, and dog have some difficulties to recognize. Although in the testing step, some of the automobiles are not reconstructed well, they are some birds seem good.

If we only use the decoder which is a part of VAE, we need to input the Gaussian distribution and sample from it. However, the images do look not good as above. But, we still can see the contours.

So, our CNN model can arrive FID score is 182.441 and IS score is 1.7436. Total using 10,496,274 parameters.

We think this homework arrives at our expected result because VAE's ability is not very good like Stable Diffusion.

CVAE is an interesting homework. Although it takes a lot of time to do (because we use PyTorch instead of Keras which is too simple), it makes us understand Neural Network architecture more. And define the loss function by self is a huge progress for using those packages and understanding the meaning.

Very much appreciate the teacher giving us this homework and teaching the mathematics concept. In this class, we learn a lot of knowledge about AI and coding more smoothly. Finally, this Deep Learning is the most necessary course in any case!!

Experiments equipment: GPU: RTX3060 12G VRAM