

# Auxiliary Classifier Generative Adversarial Networks (ACGAN)

David Ishak Kosasih  
20195033

---

For ACGAN implementation in R, I used example from the following website :

[https://keras.rstudio.com/articles/examples/mnist\\_acgan.html](https://keras.rstudio.com/articles/examples/mnist_acgan.html).

When implementing this code, I only added several lines for saving the model. My ACGAN implementation, generated images and the trained model can be found on my GitHub :

<https://github.com/ishakdavidk/Machine-Learning/tree/master/Homework%203>.

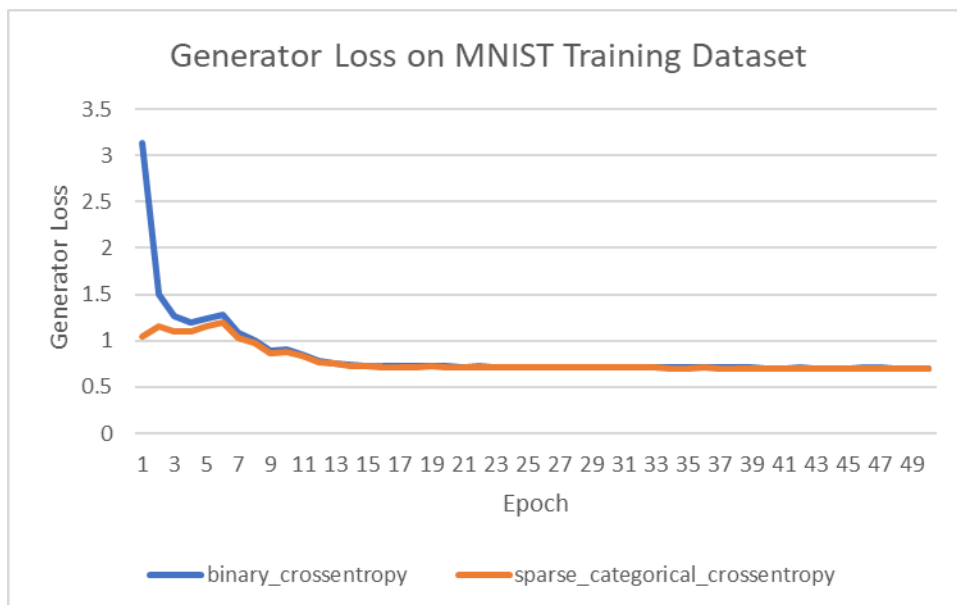
The trained models (generator and discriminator) were saved by using both `save()` and `saveRDS()` functions. For models that were saved by using `save()` function can be opened by calling `load()` function. For example :

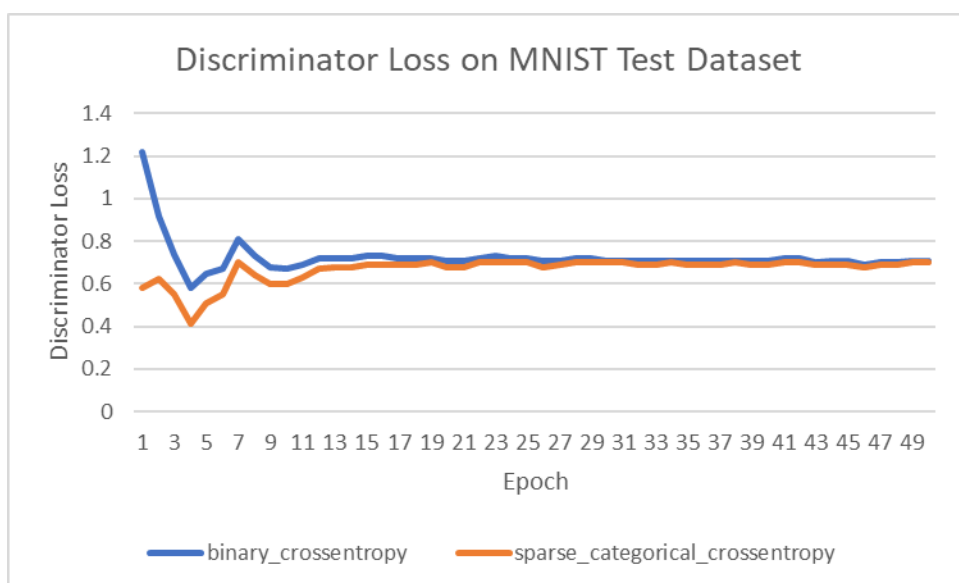
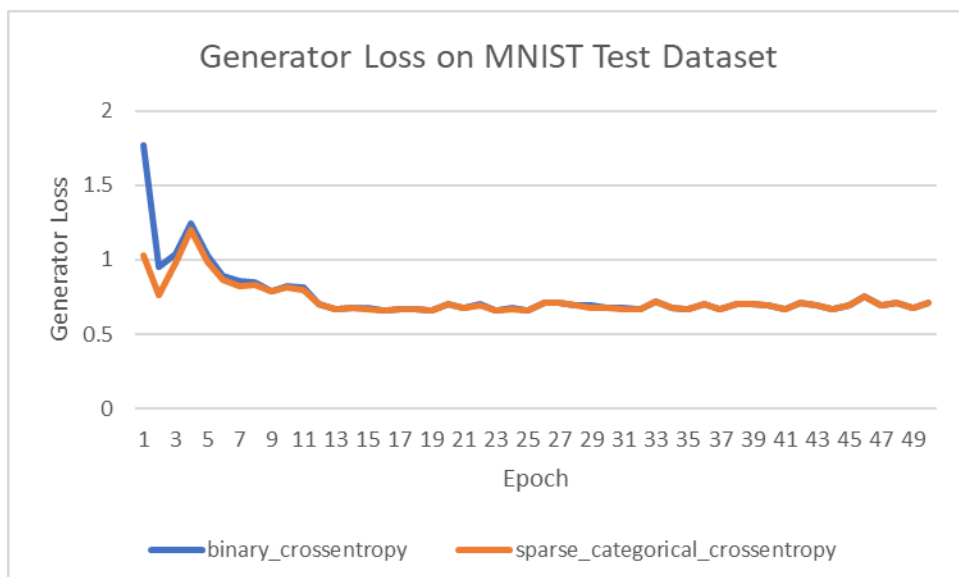
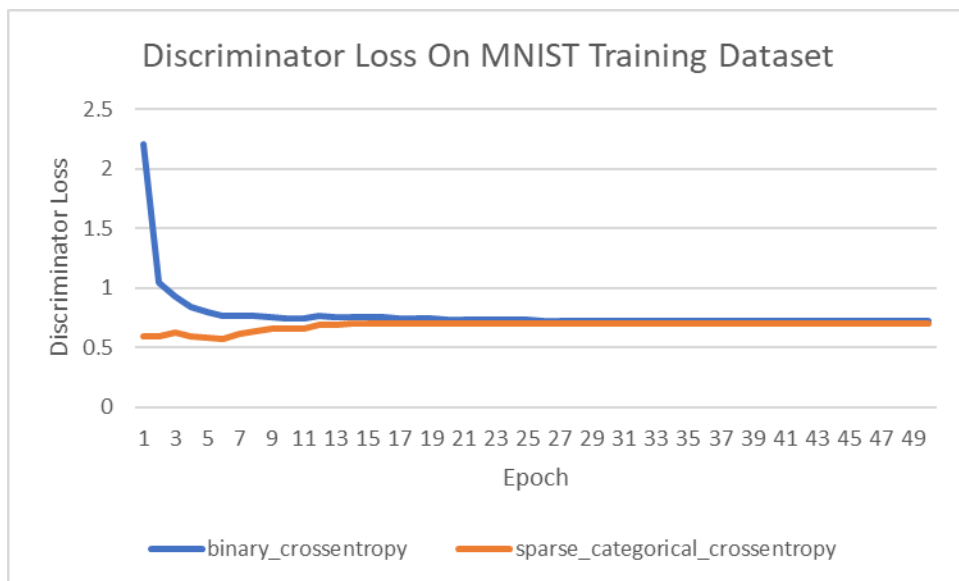
```
model <- load(file = ACGAN_MNIST_model_discriminator.rda)
```

For models that were saved by using `saveRDS` function, can be opened by calling `readRDS()` function. For example :

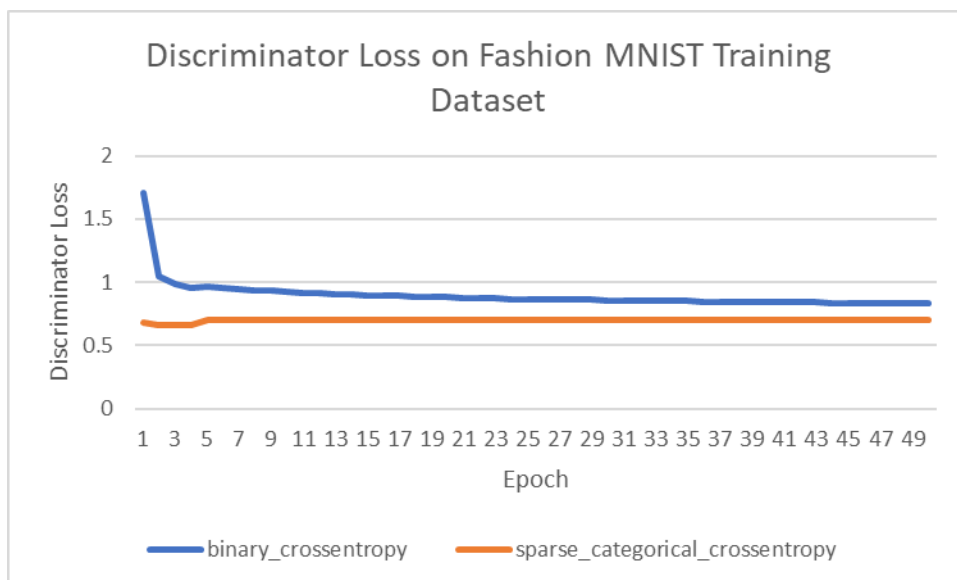
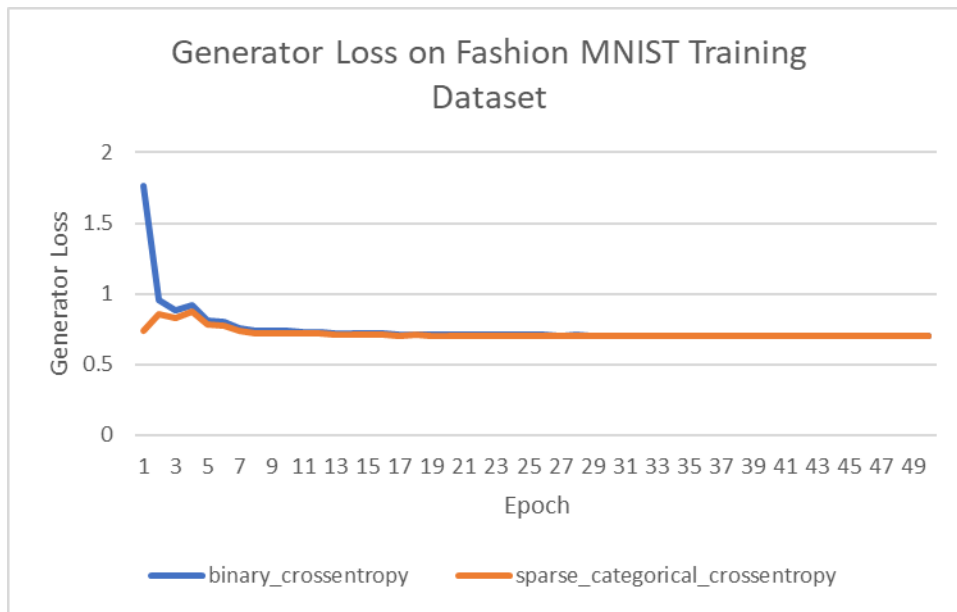
```
model <- readRDS("ACGAN_FashionMNIST_model_generator (saveRDS).rda")
```

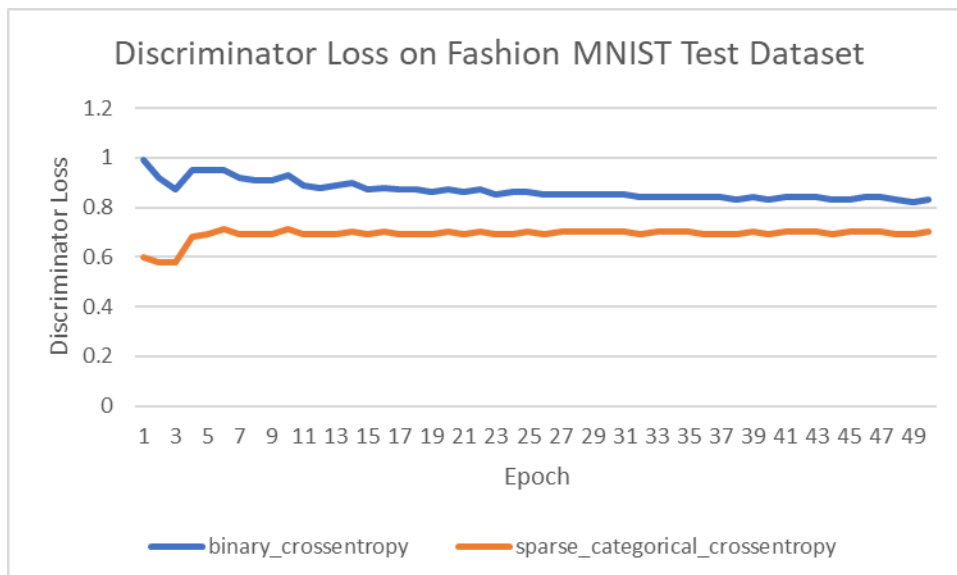
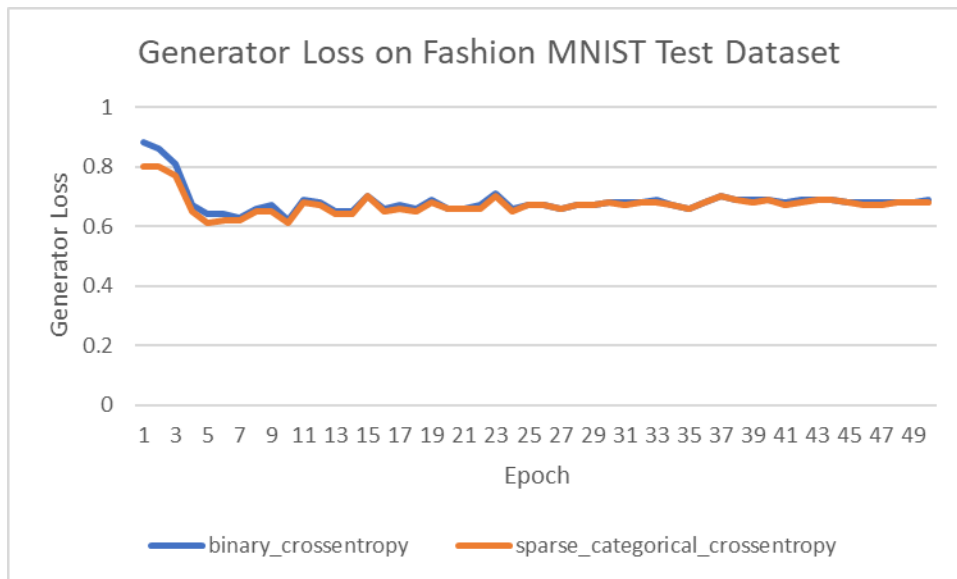
## Loss Function Plot for MNIST





## Loss Function Plot for Fashion MNIST





## Analysis

In the loss function plots above, there are two lines : blue and orange. The blue line describes binary cross entropy loss while the orange line depicts sparse categorical cross entropy loss. We have two loss functions because in ACGAN, not only does the discriminator guess whether the given input images fake or real, but it also needs to predict the class label (discriminator is trained to maximize LS (the log-likelihood of the correct source) + LC (the log-likelihood of the correct class) while Generator is trained to maximize LC – LS). Binary cross entropy was used for images real-ness prediction while sparse categorical cross-entropy loss was used for the class label prediction.

For MNIST dataset, after approximately 13 epochs, the two loss function approached similar value in both generator and discriminator. In fact, if we check the generated images, we can start to see a fairly good result after this epoch.

For fashion MNIST dataset, in my opinion, it didn't really show big improvement even after 50 epochs; however, after 15 epochs, it did generate pretty reasonable results.