Data Augmentation using VAE for enhancing Classification with various Dataset

TaeHo Yoon
Advisor:Prof.ChongSun Park

SungKyunkwan University

2th June 2023

Overview

Logistic model

- n=100000, traindata=80000, testdata=20000
- $x_1, x_2, x, \cdots, x_{10} \sim U(0,1)$
- $\beta = (1, 1.5, 2, 2.5, 0, -1, -1.5, -2, -2.5, 0)$
- $p(y|x) = \frac{e^{x_1+1.5x_2+2x_3+2.5x_4+0x_5-1x_6-1.5x_7-2x_8-2.5x_9+0x_{10}}}{1+e^{x_1+1.5x_2+2x_3+2.5x_4+0x_5-1x_6-1.5x_7-2x_8-2.5x_9+0x_{10}}}$
- Fit the train data to the logistic regression model and measure the accruacy of the test data.

	False	**True**
0	7356	2536
1	2720	7388

73.09062%

Logistic model

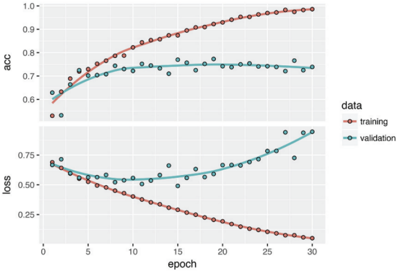
- Downsampling data with category 1 at a rate of 0.05%.
- Oversampling by the original number of data with category 1.
- Fit the Oversampling data to the logistic regression model and measure the accuracy of the test data.

	False	**True**
0	7900	1992
1	4257	5851

57.88484%

- It is confirmed that the performance of the model has decreased as the information of the data with category 1 is reduced.

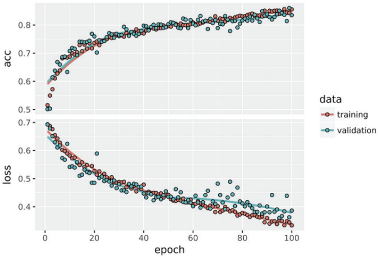
CNN model



- training accuracy is higher than validation accuracy training loss is lower than validation loss.
- An overfitting problem has occurred.

Data Augmentation

sparse data → augmentation data



- The validation accuracy is improved.
- The problem of overfitting has improved to a certain extent.

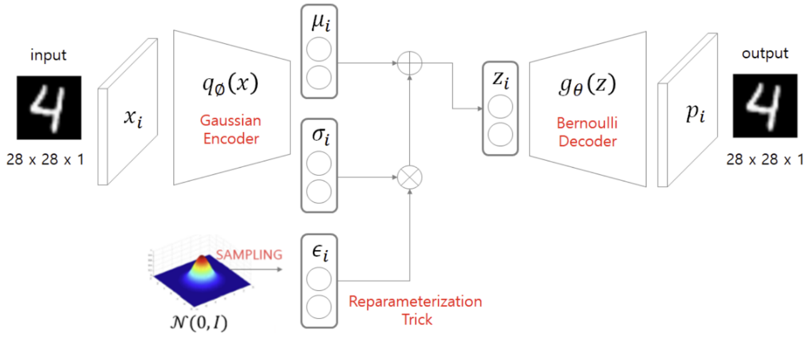
[Deep Learning with R (Francis Chollet with J.J. Allaire)]

Data Augmentation

- Cherry Khosla (2020) suggested that image data augmentation can improve the performance of the CNN model.
- Zubayer Islam, Mohamed Abdel-Aty (2021) suggested that the performance of SVM and logistic regression improves when augmenting collision data with VAE.

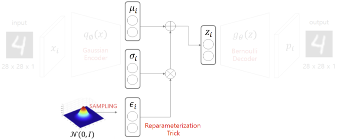
  - **VAE with predicted loss function**: Development of a VAE model that increases predictive performance by adding a predictive loss function in addition to two existing VAE loss functions.

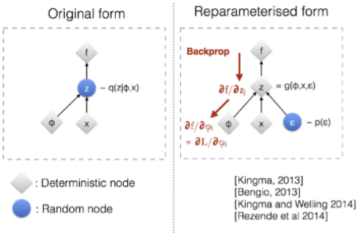
VAE



- input:$x_i \to q_\phi(x) \to \mu_i, \sigma_i$
- $\mu_i, \sigma_i, \epsilon_i \to z_i$
- $z_i \to g_\theta(z) \to p_i$: output

Reparameterization trick

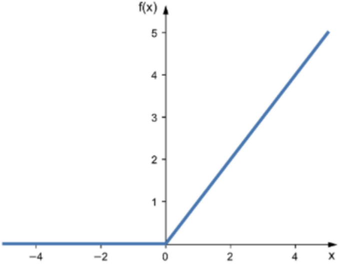




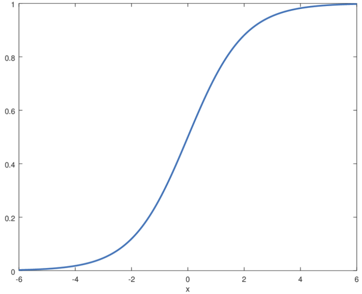
- $z^{i,l} \sim N(\mu_i, {\sigma_i}^2 I) \Rightarrow z^{i,l} = \mu_i + {\sigma_i}^2 \odot \epsilon$
- $\epsilon \sim N(0, I)$
  same distribution! but it makes backpropagation possible

Activation Function

- RELU: $R(x) = \max(0, x)$
- sigmoid: $\sigma(x) = \frac{1}{1+e^{-x}}$



Relu



Sigmoid

Loss Function

- Data likelihood: $p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$ (intractable)
- Posterior density also intractable: $p_\theta(z|x) = \frac{p_\theta(x|z)p_\theta(z)}{p_\theta(x)}$

- $\log p_\theta(x^{(i)}) = E_{z \sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right]$

$$= E_Z\left[\log \frac{p_\theta(x^{(i)}|z)p_\theta(z)}{p_\theta(z|x^{(i)})}\right]$$

$$= E_z\left[\log \frac{p_\theta(x^{(i)}|z)p_\theta(z)}{p_\theta(z|x^{(i)})} \frac{q_\phi(z|x^{(i)})}{q_\phi(z|x^{(i)})}\right]$$

$$= E_z\left[\log p_\theta(x^{(i)}|z)\right] - E_z\left[\log \frac{q_\phi(z|x^{(i)})}{p_\theta(z)}\right] + E_z\left[\log \frac{q_\phi(z|x^{(i)})}{p_\theta(z|x^{(i)})}\right]$$

$$= E_z\left[\log p_\theta(x^{(i)}|z)\right] - D_{KL}\left(q_\phi(z|x^{(i)})||p_\theta(z)\right) + D_{KL}\left(q_\phi(z|x^{(i)})\right.$$

ELBO

- $E_z \left[\log p_\theta(x^{(i)}|z)\right] - D_{KL}\left(q_\phi(z|x^{(i)})||p_\theta(z)\right) = \pounds(x^i, \theta, \phi)$
- $D_{KL}(q_\phi(z|x^{(i)})||p_\theta(z|x^{(i)}) \geq 0$

$logp_\theta(x^i|z) >= \pounds(x^i, \theta, \phi)$
variational lower bound("ELBO")
$\Rightarrow$
$\theta^*, \phi^* = \underset{\theta,\phi}{argmax} \sum_{i=1}^{N} \pounds(x^{(i)}, \theta, \phi)$
Training: Maximize lower bound

Loss Function

- $\underset{\theta,\phi}{argmin} \sum_i -E_{q_\phi(z|x_i)}\left[log(p(x_i|g_\theta(z)))\right] + KL(q_\phi(z|x_i)||p(z))$

reference

$D_{KL}(N_0||N_1) =$
$\frac{1}{2}\left(tr({\sum}_1{}^{-1}{\sum}_0) + (\mu_1 - \mu_0) - k + ln\left(\frac{det \sum_1}{det \sum_0}\right)\right)$

$\mathbf{KL}(\mathbf{q}_\phi(z|x_i)||p(z))$

$\frac{1}{2}\left\{tr({\sigma_i}^2) + \mu_i^T\mu_i - J + ln\frac{1}{\prod_{j=1}^J \sigma_{i,j}{}^2)}\right\} =$
$\frac{1}{2}\left\{\sum_{j=1}^J {\sigma_{i,j}}^2 + \sum_{j=1}^J \mu_{i,j}^2 - J - \sum_{j=1}^J ln({\sigma_{i,j}}^2)\right\} =$
$\frac{1}{2}\sum_{j=1}^J({\mu_{i,j}}^2 + {\sigma_{i,j}}^2 - ln({\sigma_{i,j}}^2) - 1)$

Reconstruction error

$$E_{q_\phi(z|x_i)}\left[\log(p_\theta(x_i|z))q_\phi(z|x_i)dz\right] = \int \log(p_\theta(x_i|z))q_\phi(z|x_i)dz \rightarrow$$

$$\text{monte-carlo technique} \approx \frac{1}{L}\sum_{z^{i,l}} \log(p_\theta \rightarrow (x_i|z^{i,l})) \rightarrow$$

$$\log(p_\theta(x_i|z^i))$$

- L is the number of samples for latent vector.
- Usually, L is set to 1 for convenience.

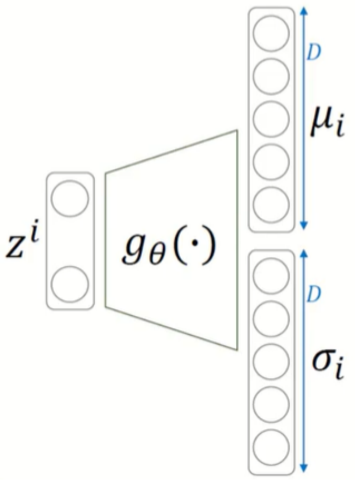


Figure: Decoder

[Decoder,likelihood](gaussian distribution

$\log(\mathrm{p}_\theta(x_i|z^i)) = log(N(x_i;\mu_i,{\sigma_i}^2)) = -\sum_{j=1}^{D}\frac{1}{2}log({\sigma_{i,j}}^2) + \frac{(x_{i,j}-\mu_{i,j})^2}{2{\sigma_{i,j}}^2}$

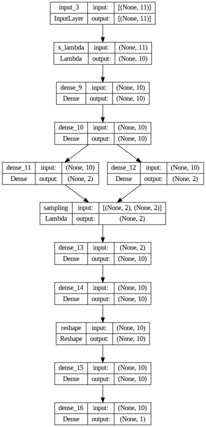
For gaussian distribution with identity covariance

$\log\big(p_\theta(x_i|z^i)\big) \propto \sum_{j=1}^{D}(x_{i,j} - \mu_{i,j})^2 (SquaredError)$

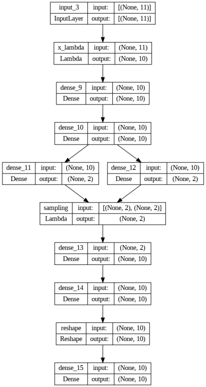
Loss Function

- Reconstruction error: $-\sum_{j=1}^{D}(x_{i,j}-\mu_{i,j})^2$
- Regularization: $\frac{1}{2}\sum_{j=1}^{J}(\mu_{i,j}^2+\sigma_{i,j}^2-\ln(\sigma_{i,j})-1)$
- Binary cross entropy:
  $-\frac{1}{N}\sum_{i=0}^{N} y_i \cdot \log(\hat{y}) + (1-y_i)\cdot \log(1-\hat{y}_i)$

Variational Autoencoder



(VAE)Prediction



(VAE1)Image Generation

VAE

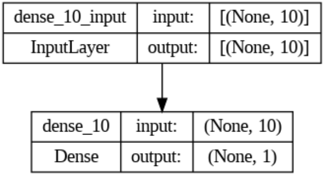
Layer (type)	Output Shape	Param #	Activation function
input_1 (InputLayer)	[(None, 11)]	0	
lambda	(None, 10)	0	
dense	(None, 10)	110	relu
dense_1	(None, 10)	110	relu
dense_2	(None, 2)	22	
dense_3	(None, 2)	22	
concatenate	(None, 4)	0	
lambda_2	(None, 2)	0	
dense_4	(None, 10)	30	relu
dense_5	(None, 10)	110	relu
reshape	(None, 10)	0	
dense_6	(None, 10)	110	sigmoid
dense_7	(None, 1)	11	sigmoid

Data Processing

- train data=80000, test data=20000
- $x_1, x_2, x, \cdots, x_{10} \sim U(0,1)$
- $y = \frac{e^{x_1+1.5x_2+2x_3+2.5x_4+0x_5-1x_6-1.5x_7-2x_8-2.5x_9+0x_{10}}}{1+e^{x_1+1.5x_2+2x_3+2.5x_4+0x_5-1x_6-1.5x_7-2x_8-2.5x_9+0x_{10}}}$

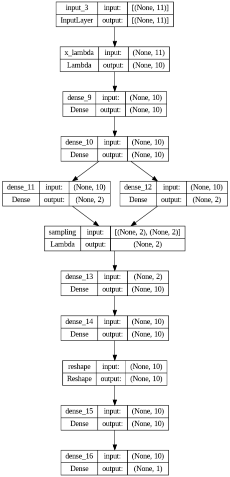
- data→downsampling data with category 1 at 0.01 ratio→oversampling data with category 1

Feed forward neural network



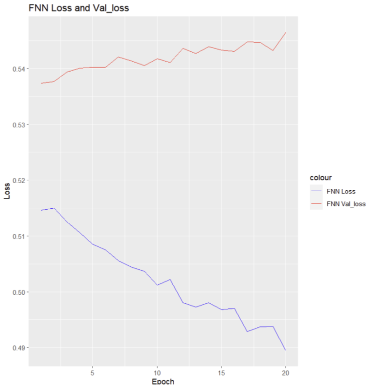
- set.seed(1)
- data: Oversampling data
- batch size:128, epochs=5
- optimizer:rmsprop, loss='binary crossentropy
- test data [loss:0.5572421] [accuracy: 0.71055]

Model fitting processing



1. Fit the oversampling data to the VAE model.(epochs=1,batch size=160)
2. 60 data with category 1 + 20 data generated by VAE1 out of 60 data + 80 data with category 0 $\rightarrow$ 107200 total data
3. Fitting the entire data to the FFN.(epochs=10,batch size=160)
4. Repeat the above process for the number of epochs

- loss function weight: (xent loss:1),(kl loss:-0.5),(p loss:0.1)



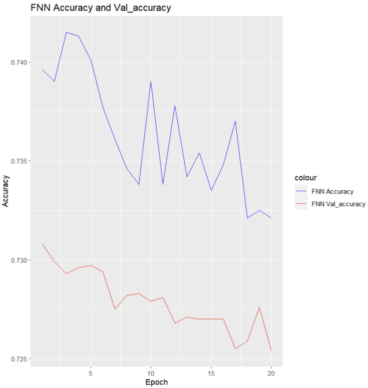
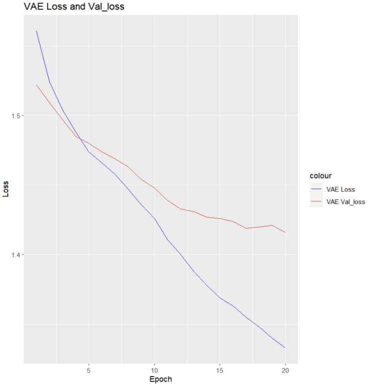


Figure: FNN loss and accuracy

- loss function weight: (xent loss:1),(kl loss:-0.5),(p loss:0.1)



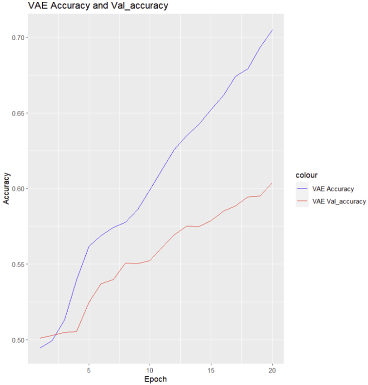


Figure: VAE loss and accuracy

- loss function weight: (xent loss:1),(kl loss:-0.5),(p loss:0.0)

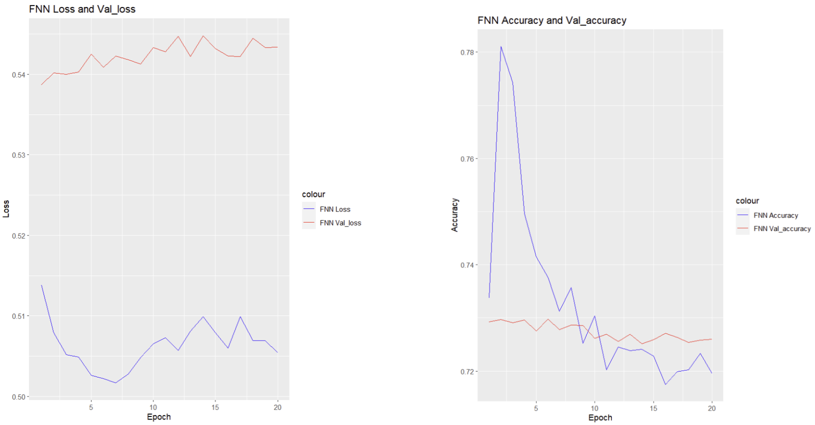


Figure: FNN loss and accuracy

- loss function weight: (xent loss:1),(kl loss:-0.5),(p loss:0.0)

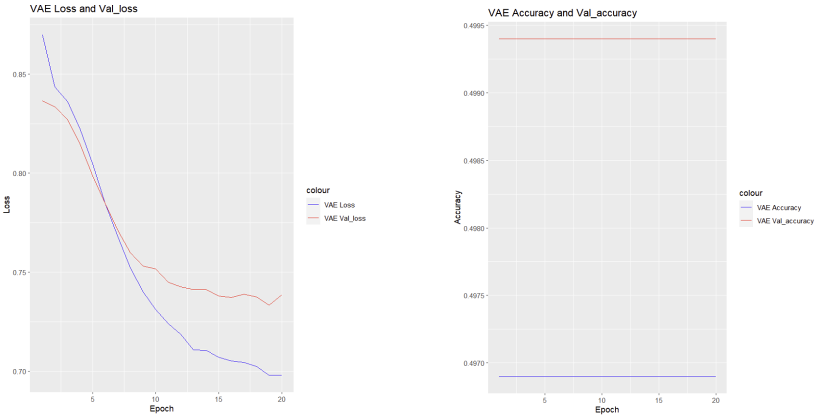
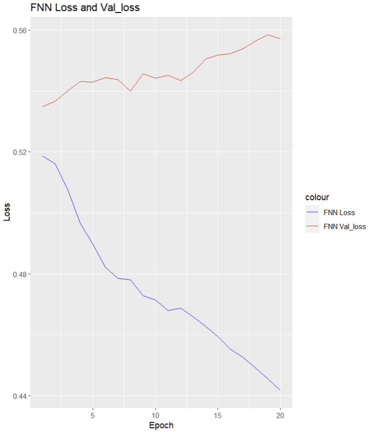


Figure: VAE loss and accuracy

- loss function weight: (xent loss:0),(kl loss:0.),(p loss:1.0)



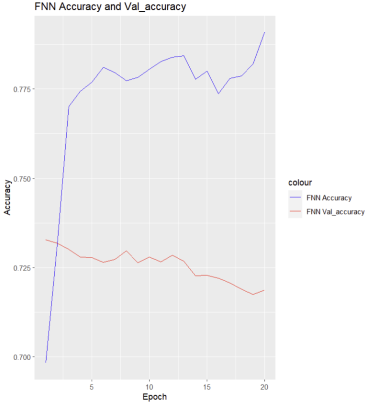


Figure: FNN loss and accuracy

- loss function weight: (xent loss:0),(kl loss:0.0),(p loss:1.0)

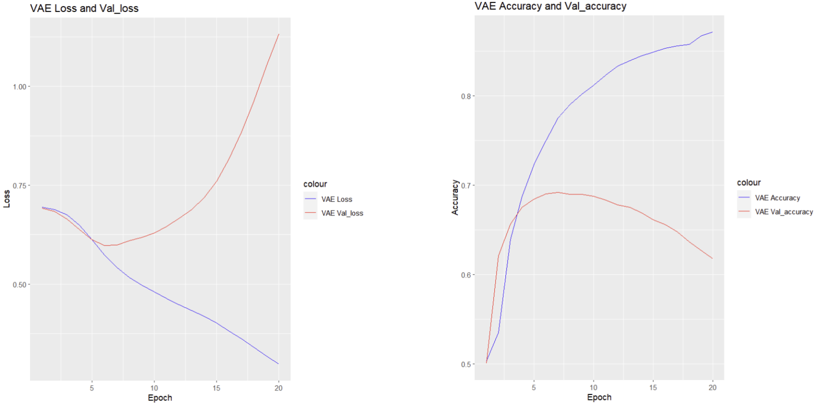
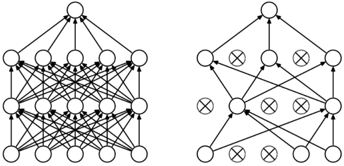


Figure: VAE loss and accuracy

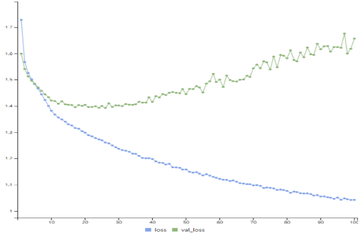
Dropout

- Nitish Srivastava(2014) suggested that randomly drop units (along with their connections) from the neural reduces overfitting and gives major improvements over other regularization methods. network during training.



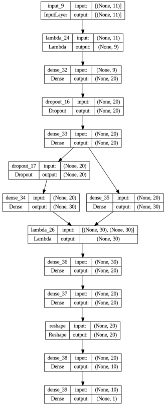
(a) Standard Neural Net (b) After applying dropout.

Dropout



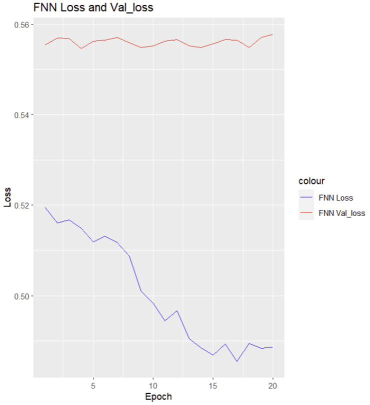
VAE

Model fitting processing



1. Fit the oversampling data to the VAE model(Models with drop out added)(epochs=2,batch size=160).
2. Category 1 data + Category 0 data + Data generation from VAE1: (Data with VAE predictive probability close to 0.5) + Data with category 0 matched categories of 1.
3. Fit the entire data to the FFN.(epochs=7,batch size=160)
4. Repeat the above process for the number of epochs.

- loss function weight: (xent loss:1),(kl loss:-0.5),(p loss:1.0)



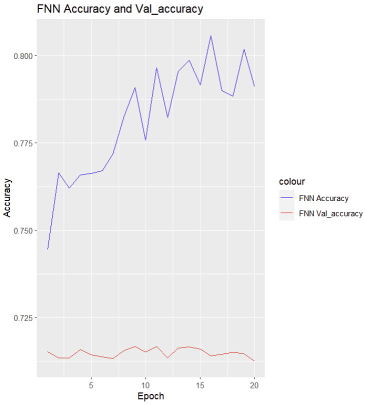
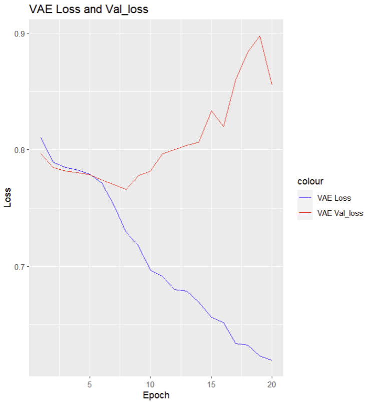


Figure: FNN loss and accuracy

- loss function weight: (xent loss:1),(kl loss:-0.5),(p loss:1.0)



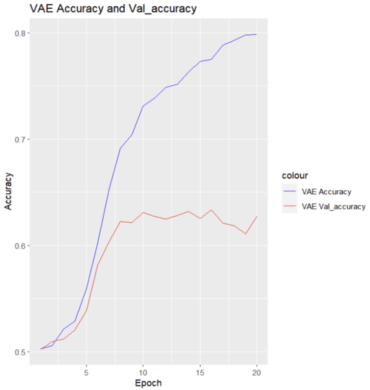


Figure: VAE loss and accuracy

- loss function weight: (xent loss:1),(kl loss:-0.5),(p loss:0.0)

Figure: FNN loss and accuracy

- loss function weight: (xent loss:1),(kl loss:-0.5),(p loss:0.0)

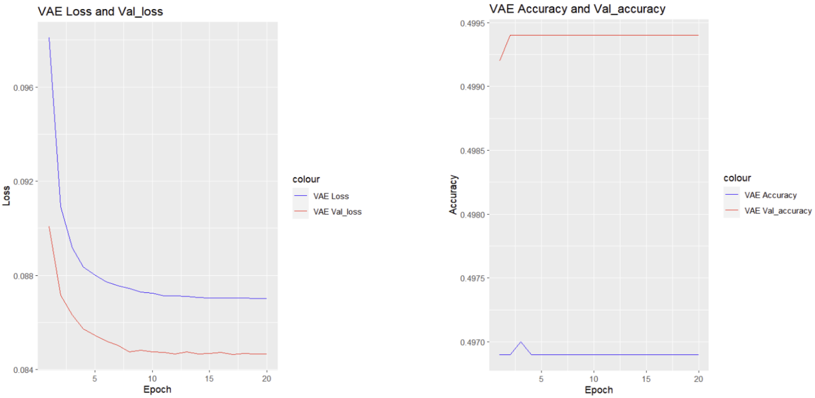


Figure: VAE loss and accuracy

- loss function weight: (xent loss:0),(kl loss:0.0),(p loss:1.0)

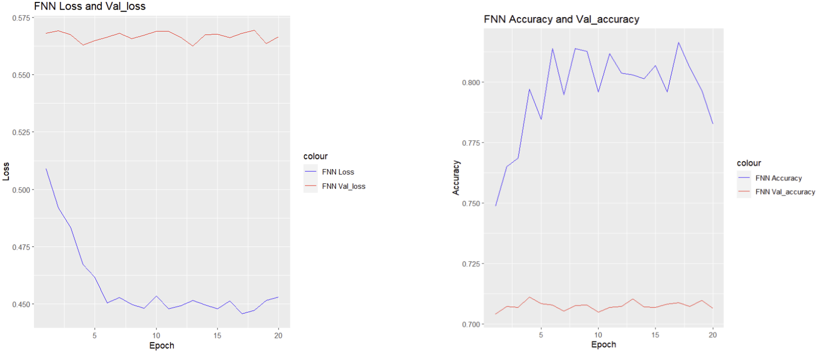
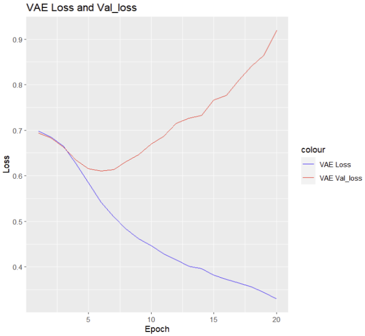


Figure: FNN loss and accuracy

- loss function weight: (xent loss:0),(kl loss:0.),(p loss:1.0)



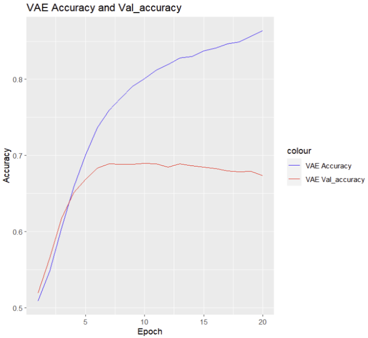


Figure: VAE loss and accuracy

Future study

- Finding a way to improve the model by adjusting various factors in the vae model.
- When the performance of the model improves, the model is applied to bankrupcy data or credit card data.
- applying the model not only to two-dimensional data but also to three-dimensional data.
- Checking performance by fitting models that determine multiple categories instead of two categories

References

- DiederikP.Kingma, MaxWelling(2022) Auto-Encoding VariationalBayes
- Zubayer Islam *, Mohamed Abdel-Aty, Qing Cai, Jinghui Yuan(2021) Crash data augmentation using variational autoencoder
- Francis chollet with JJ.Allaire(2017) deep learning with r
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. (2014) Dropout: A simple way to prevent neural networks from overfitting
- Cherry Khosla,Baljit Singh Saini(2020) Enhancing Performance of Deep Learning Models with different Data Augmentation Techniques: A Survey