Auto Encoders

Hamza Gouaref

2024-10-28

```r
library(reticulate)

use_python("C:/Users/TWTEC/anaconda3/envs/r-keras/python.exe", required = TRUE)
```

```r
library(keras)
install_keras(method = "conda", envname = "r-keras")
```

```
##
## Installation complete.
```

```r
use_python("C:/Users/TWTEC/anaconda3/envs/r-keras/python.exe", required = TRUE)
```

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.0     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(tidymodels)
```

```
## -- Attaching packages -------------------------------------- tidymodels 1.2.0 --
## v broom        1.0.5      v rsample      1.2.1
## v dials        1.2.1      v tune         1.2.0
## v infer        1.0.7      v workflows    1.1.4
## v modeldata    1.3.0      v workflowsets 1.1.0
## v parsnip      1.2.1      v yardstick    1.3.1
## v recipes      1.0.10
## -- Conflicts ----------------------------------------- tidymodels_conflicts() --
## x scales::discard()        masks purrr::discard()
## x dplyr::filter()          masks stats::filter()
## x recipes::fixed()         masks stringr::fixed()
## x yardstick::get_weights() masks keras::get_weights()
```

```
## x dplyr::lag()             masks stats::lag()
## x yardstick::spec()        masks readr::spec()
## x recipes::step()          masks stats::step()
## * Use tidymodels_prefer() to resolve common conflicts.
```

```r
library(ggplot2)
library(reshape2)
```

```
##
## Attachement du package : 'reshape2'
##
## L'objet suivant est masqué depuis 'package:tidyr':
##
##     smiths
```

Load and Preprocess the MNIST Dataset

```r
mnist <- dataset_mnist()

x_train <- mnist$train$x
x_test <- mnist$test$x

# Rescale images to values between 0 and 1
x_train <- x_train / 255
x_test <- x_test / 255

# Reshape to add channel dimension
x_train <- array_reshape(x_train, c(nrow(x_train), 28, 28, 1))

x_test <- array_reshape(x_test, c(nrow(x_test), 28, 28, 1))

# Check the shape to confirm
dim(x_train)
```

```
## [1] 60000    28    28     1
```

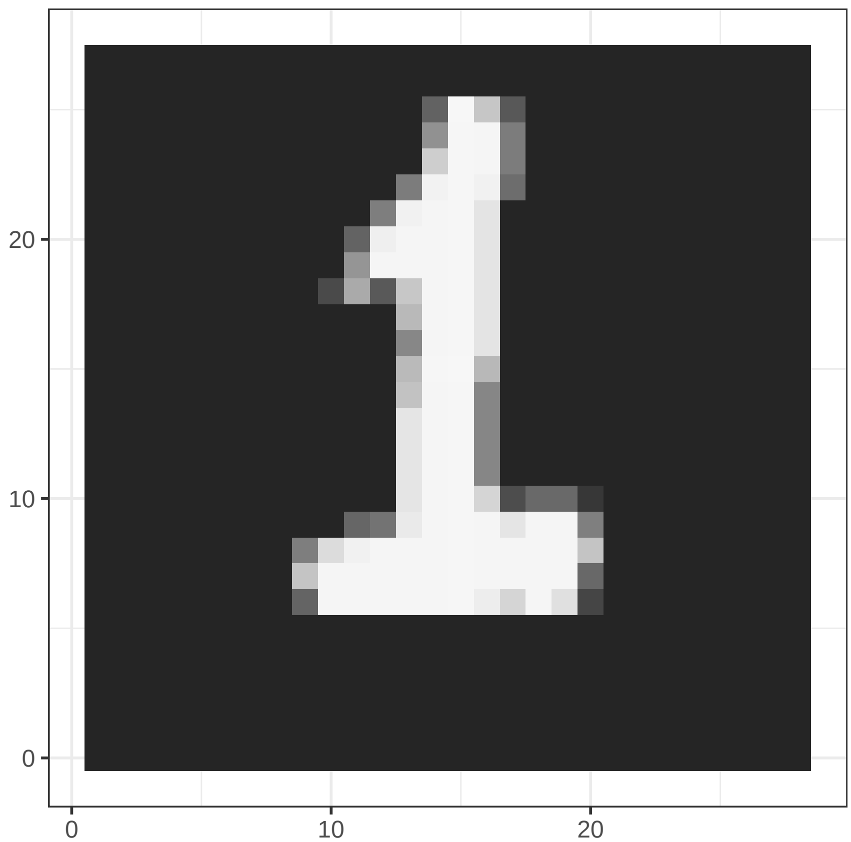
Exemple

```r
index_image <- sample(1:60000,1)


x_train[index_image, 1:28, 1:28,1] %>%
  as_data_frame() %>%
  rownames_to_column(var = "y") %>%
  pivot_longer(cols = V1:V28, names_to = "x", values_to = "val") %>%
  mutate(x = str_replace(x, "V" ,""),
         x = as.numeric(x),
         y = as.numeric(y),
         y = 28-y) %>%
```

```r
  ggplot(aes(x = x, y = y)) +
  geom_tile(aes(fill = val)) +
  scale_fill_distiller(type = "seq", direction = -1,palette = "Greys") +
  coord_fixed() +
  theme_bw() +
  theme(legend.position = "none",
        axis.title = element_blank())
```

```
## Warning: `as_data_frame()` was deprecated in tibble 2.0.0.
## i Please use `as_tibble()` (with slightly different semantics) to convert to a
##   tibble, or `as.data.frame()` to convert to a data frame.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: The `x` argument of `as_tibble.matrix()` must have unique column names if
## `.name_repair` is omitted as of tibble 2.0.0.
## i Using compatibility `.name_repair`.
## i The deprecated feature was likely used in the tibble package.
##   Please report the issue at <https://github.com/tidyverse/tibble/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Build the Variational Autoencoder (VAE)

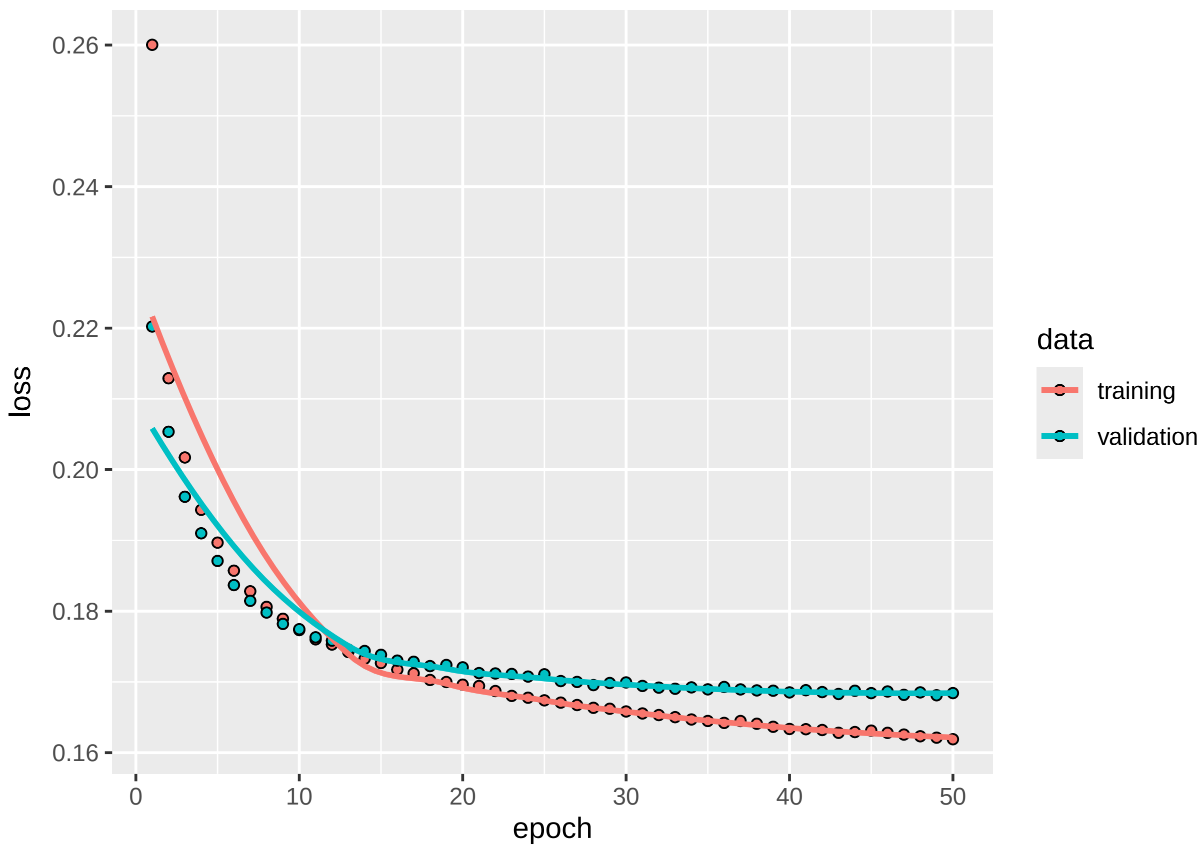
```r
x_train <- array_reshape(x_train, c(nrow(x_train), 784))  # Flatten 28x28 to 784 features

input_dim <- 784       # Flattened 28x28 images
latent_dim <- 2        # Latent space dimension for generating new digits

# Encoder
encoder_input <- layer_input(shape = input_dim)
encoder_output <- encoder_input %>%
  layer_dense(units = 512, activation = "relu") %>%
  layer_dense(units = 256, activation = "relu") %>%
  layer_dense(units = latent_dim)  # Compressed latent representation

encoder <- keras_model(encoder_input, encoder_output)

# Decoder
decoder_input <- layer_input(shape = latent_dim)
decoder_output <- decoder_input %>%
  layer_dense(units = 256, activation = "relu") %>%
  layer_dense(units = 512, activation = "relu") %>%
  layer_dense(units = input_dim, activation = "sigmoid")  # Reconstructed image

decoder <- keras_model(decoder_input, decoder_output)

# VAE Model - Encoder + Decoder
vae_input <- layer_input(shape = input_dim)
latent_space <- encoder(vae_input)
vae_output <- decoder(latent_space)
vae <- keras_model(vae_input, vae_output)

# Compile the VAE model
vae %>% compile(
  optimizer = "adam",
  loss = "binary_crossentropy"
)

# Train the VAE model
history <- vae %>% fit(
  x_train, x_train,
  epochs = 50,
  batch_size = 256,
  validation_split = 0.2
)
```

```
## Epoch 1/50
## 188/188 - 6s - loss: 0.2600 - val_loss: 0.2202 - 6s/epoch - 34ms/step
## Epoch 2/50
## 188/188 - 4s - loss: 0.2129 - val_loss: 0.2054 - 4s/epoch - 20ms/step
## Epoch 3/50
## 188/188 - 4s - loss: 0.2017 - val_loss: 0.1962 - 4s/epoch - 20ms/step
## Epoch 4/50
## 188/188 - 4s - loss: 0.1943 - val_loss: 0.1910 - 4s/epoch - 19ms/step
```

```
## Epoch 5/50
## 188/188 - 4s - loss: 0.1897 - val_loss: 0.1871 - 4s/epoch - 20ms/step
## Epoch 6/50
## 188/188 - 4s - loss: 0.1857 - val_loss: 0.1837 - 4s/epoch - 19ms/step
## Epoch 7/50
## 188/188 - 4s - loss: 0.1828 - val_loss: 0.1815 - 4s/epoch - 20ms/step
## Epoch 8/50
## 188/188 - 4s - loss: 0.1806 - val_loss: 0.1798 - 4s/epoch - 19ms/step
## Epoch 9/50
## 188/188 - 4s - loss: 0.1789 - val_loss: 0.1782 - 4s/epoch - 20ms/step
## Epoch 10/50
## 188/188 - 4s - loss: 0.1773 - val_loss: 0.1774 - 4s/epoch - 19ms/step
## Epoch 11/50
## 188/188 - 4s - loss: 0.1760 - val_loss: 0.1763 - 4s/epoch - 19ms/step
## Epoch 12/50
## 188/188 - 4s - loss: 0.1753 - val_loss: 0.1758 - 4s/epoch - 19ms/step
## Epoch 13/50
## 188/188 - 4s - loss: 0.1742 - val_loss: 0.1746 - 4s/epoch - 20ms/step
## Epoch 14/50
## 188/188 - 4s - loss: 0.1733 - val_loss: 0.1744 - 4s/epoch - 19ms/step
## Epoch 15/50
## 188/188 - 4s - loss: 0.1727 - val_loss: 0.1738 - 4s/epoch - 19ms/step
## Epoch 16/50
## 188/188 - 4s - loss: 0.1717 - val_loss: 0.1730 - 4s/epoch - 19ms/step
## Epoch 17/50
## 188/188 - 4s - loss: 0.1712 - val_loss: 0.1728 - 4s/epoch - 20ms/step
## Epoch 18/50
## 188/188 - 4s - loss: 0.1703 - val_loss: 0.1722 - 4s/epoch - 20ms/step
## Epoch 19/50
## 188/188 - 4s - loss: 0.1700 - val_loss: 0.1724 - 4s/epoch - 19ms/step
## Epoch 20/50
## 188/188 - 4s - loss: 0.1696 - val_loss: 0.1721 - 4s/epoch - 20ms/step
## Epoch 21/50
## 188/188 - 4s - loss: 0.1694 - val_loss: 0.1712 - 4s/epoch - 20ms/step
## Epoch 22/50
## 188/188 - 4s - loss: 0.1687 - val_loss: 0.1712 - 4s/epoch - 19ms/step
## Epoch 23/50
## 188/188 - 4s - loss: 0.1680 - val_loss: 0.1711 - 4s/epoch - 20ms/step
## Epoch 24/50
## 188/188 - 4s - loss: 0.1678 - val_loss: 0.1708 - 4s/epoch - 19ms/step
## Epoch 25/50
## 188/188 - 4s - loss: 0.1674 - val_loss: 0.1711 - 4s/epoch - 19ms/step
## Epoch 26/50
## 188/188 - 4s - loss: 0.1671 - val_loss: 0.1701 - 4s/epoch - 19ms/step
## Epoch 27/50
## 188/188 - 4s - loss: 0.1667 - val_loss: 0.1700 - 4s/epoch - 19ms/step
## Epoch 28/50
## 188/188 - 4s - loss: 0.1663 - val_loss: 0.1695 - 4s/epoch - 20ms/step
## Epoch 29/50
## 188/188 - 4s - loss: 0.1662 - val_loss: 0.1698 - 4s/epoch - 20ms/step
## Epoch 30/50
## 188/188 - 4s - loss: 0.1658 - val_loss: 0.1699 - 4s/epoch - 20ms/step
## Epoch 31/50
## 188/188 - 4s - loss: 0.1655 - val_loss: 0.1694 - 4s/epoch - 19ms/step
```

```
## Epoch 32/50
## 188/188 - 4s - loss: 0.1653 - val_loss: 0.1692 - 4s/epoch - 19ms/step
## Epoch 33/50
## 188/188 - 4s - loss: 0.1650 - val_loss: 0.1690 - 4s/epoch - 19ms/step
## Epoch 34/50
## 188/188 - 4s - loss: 0.1647 - val_loss: 0.1692 - 4s/epoch - 19ms/step
## Epoch 35/50
## 188/188 - 4s - loss: 0.1645 - val_loss: 0.1689 - 4s/epoch - 21ms/step
## Epoch 36/50
## 188/188 - 4s - loss: 0.1642 - val_loss: 0.1693 - 4s/epoch - 19ms/step
## Epoch 37/50
## 188/188 - 4s - loss: 0.1645 - val_loss: 0.1689 - 4s/epoch - 20ms/step
## Epoch 38/50
## 188/188 - 4s - loss: 0.1641 - val_loss: 0.1688 - 4s/epoch - 20ms/step
## Epoch 39/50
## 188/188 - 4s - loss: 0.1637 - val_loss: 0.1688 - 4s/epoch - 20ms/step
## Epoch 40/50
## 188/188 - 4s - loss: 0.1633 - val_loss: 0.1685 - 4s/epoch - 20ms/step
## Epoch 41/50
## 188/188 - 4s - loss: 0.1633 - val_loss: 0.1688 - 4s/epoch - 19ms/step
## Epoch 42/50
## 188/188 - 4s - loss: 0.1632 - val_loss: 0.1686 - 4s/epoch - 20ms/step
## Epoch 43/50
## 188/188 - 4s - loss: 0.1628 - val_loss: 0.1683 - 4s/epoch - 20ms/step
## Epoch 44/50
## 188/188 - 4s - loss: 0.1629 - val_loss: 0.1688 - 4s/epoch - 20ms/step
## Epoch 45/50
## 188/188 - 4s - loss: 0.1631 - val_loss: 0.1684 - 4s/epoch - 20ms/step
## Epoch 46/50
## 188/188 - 4s - loss: 0.1628 - val_loss: 0.1686 - 4s/epoch - 19ms/step
## Epoch 47/50
## 188/188 - 4s - loss: 0.1626 - val_loss: 0.1682 - 4s/epoch - 19ms/step
## Epoch 48/50
## 188/188 - 4s - loss: 0.1623 - val_loss: 0.1685 - 4s/epoch - 19ms/step
## Epoch 49/50
## 188/188 - 4s - loss: 0.1621 - val_loss: 0.1681 - 4s/epoch - 20ms/step
## Epoch 50/50
## 188/188 - 4s - loss: 0.1619 - val_loss: 0.1684 - 4s/epoch - 19ms/step
```

```r
# Plot training history
plot(history)
```



Generate new digit by sampling from latent space

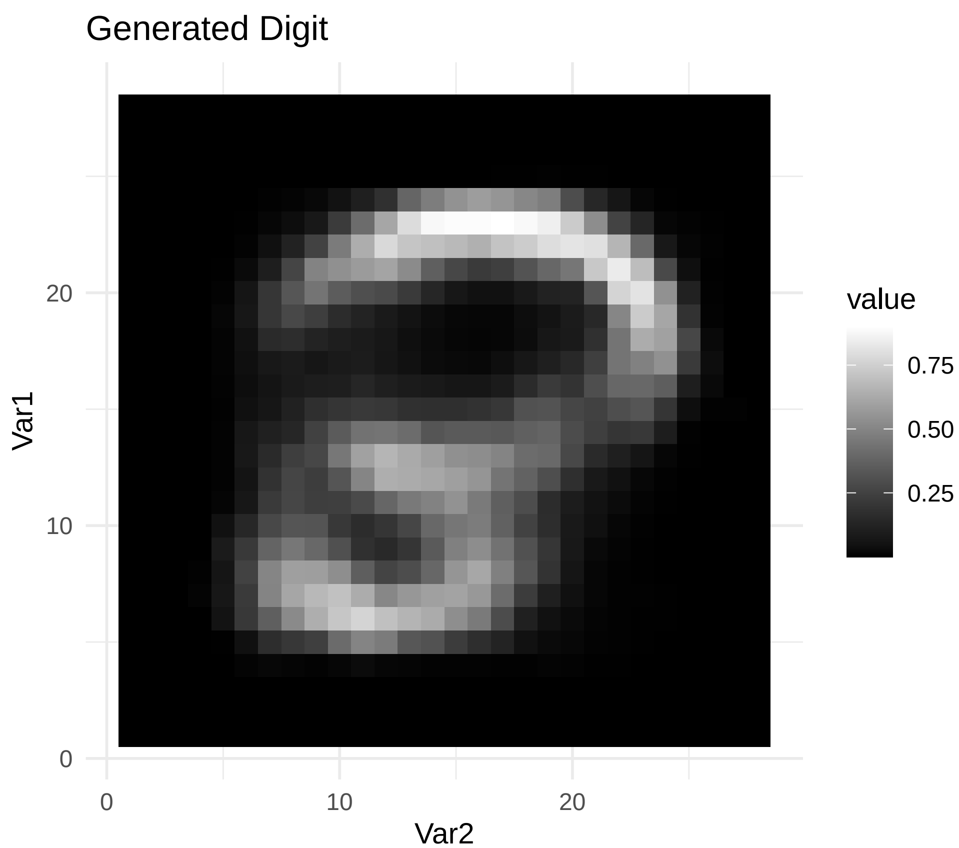
```r
sampled_latent <- matrix(runif(2, min = -2, max = 2), nrow = 1)
sampled_latent# Random point in latent space
```

```
##            [,1]      [,2]
## [1,] -0.6656602 0.4332547
```

```r
generated_digit <- decoder %>% predict(sampled_latent)
```

```
## 1/1 - 0s - 166ms/epoch - 166ms/step
```

```r
generated_digit <- array_reshape(generated_digit, c(28, 28))

# Plot the generated digit

ggplot(melt(generated_digit), aes(Var2, Var1, fill = value)) +
  geom_tile() +
  scale_fill_gradient(low = "black", high = "white") +
  theme_minimal() +
  coord_fixed() +
  ggtitle("Generated Digit")
```


Generated Digit
value
0.75
0.50
0.25
Var1
20
10
0
0
10
20
Var2