

Modul_ANN

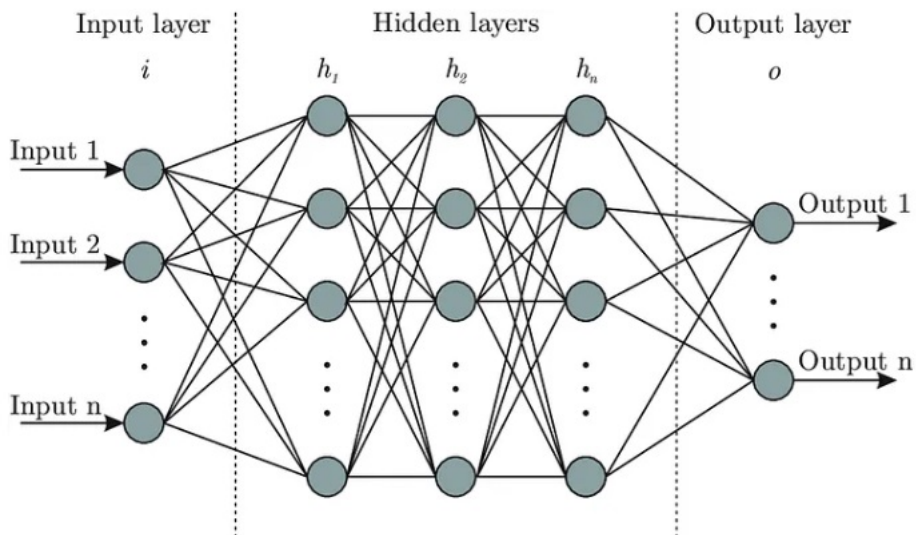
Roni Yunis

2023-11-22

Introduction to Neural Networks

Neural network atau jaringan saraf tiruan adalah bagian dari pembelajaran mesin yang terinspirasi oleh otak manusia. Jaringan ini meniru cara neuron biologis berkomunikasi satu sama lain untuk menghasilkan sebuah keputusan.

Jaringan saraf terdiri dari lapisan input, lapisan tersembunyi, dan lapisan output. Lapisan pertama menerima input mentah, diproses oleh beberapa lapisan tersembunyi, dan lapisan terakhir menghasilkan hasilnya.



Algoritme deep learning atau deep neural network terdiri dari beberapa lapisan tersembunyi dan node. “Deep” berarti kedalaman jaringan saraf. Algoritma ini umumnya digunakan untuk memecahkan masalah yang kompleks seperti klasifikasi gambar, pengenalan suara, dan pembuatan teks.

Jenis-jenis Jaringan Syaraf Tiruan

Beberapa jenis jaringan saraf digunakan untuk aplikasi pembelajaran mesin tingkat lanjut. Tidak ada satu model arsitektur yang cocok untuk semua. Jenis jaringan saraf tertua dikenal sebagai Perceptron, yang dibuat oleh Frank Rosenblatt pada tahun 1958.

Pada bagian ini, kita akan membahas 4 jenis jaringan saraf yang paling populer yang digunakan dalam industri teknologi dan bisnis .

Jaringan Syaraf Tiruan Feedforward

Jaringan saraf feedforward terdiri dari lapisan input, lapisan tersembunyi, dan lapisan output. Disebut feedforward karena data mengalir ke arah depan, dan tidak ada backpropagation. Jaringan ini banyak digunakan dalam Klasifikasi, Pengenalan suara, Pengenalan wajah, dan Pengenalan pola.

Perceptron Multi-Lapisan

Multi-Layer Perceptron (MLP) mengatasi kekurangan jaringan saraf feedforward yang tidak dapat belajar melalui backpropagation. Jaringan ini bersifat dua arah dan terdiri dari beberapa lapisan tersembunyi dan fungsi aktivasi. MLP menggunakan propagasi maju untuk input dan propagasi balik untuk memperbarui bobot. MLP adalah jaringan saraf dasar yang telah meletakkan dasar untuk visi komputer, teknologi bahasa, dan jaringan saraf lainnya.

Jaringan Syaraf Tiruan (CNN)

Convolution Neural Networks (CNN) umumnya digunakan dalam visi komputer, pengenalan gambar, dan pengenalan pola. Jaringan ini digunakan untuk mengekstraksi fitur-fitur penting dari gambar dengan menggunakan beberapa lapisan konvolusi. Lapisan konvolusi dalam CNN menggunakan matriks khusus (filter) untuk membelitkan gambar dan membuat peta.

Secara umum, Jaringan Syaraf Tiruan Konvolusi terdiri dari lapisan input, lapisan konvolusi, lapisan penyatuan, lapisan yang terhubung penuh, dan lapisan output.

Jaringan Syaraf Tiruan (RNN)

Recurrent Neural Networks (RNN) biasanya digunakan untuk data berurutan seperti teks, urutan gambar, dan deret waktu. Jaringan ini mirip dengan jaringan feed-forward, kecuali mereka mendapatkan input dari urutan sebelumnya menggunakan loop umpan balik. RNN digunakan dalam NLP, prediksi penjualan, dan prakiraan cuaca.

RNN hadir dengan masalah gradien yang menghilang, yang diselesaikan dengan versi lanjutan RNN yang disebut Jaringan Memori Jangka Pendek (LSTM) dan Jaringan Unit Berulang (GRU).

Secara sederhana, gradien adalah vektor yang menunjukkan arah dan tingkat pertumbuhan terbesar dari suatu fungsi pada suatu titik tertentu.

Contoh Implementasi ANN

Dataset “Garment Worker Productivity” adalah kumpulan data yang digunakan untuk memahami dan memprediksi produktivitas pekerja di industri garmen atau pakaian. Dataset ini dapat digunakan untuk melatih model prediksi produktivitas berdasarkan berbagai fitur atau faktor yang mungkin mempengaruhi kinerja pekerja.

Berikut adalah beberapa variabel ada dalam dataset ini berdasarkan deskripsi yang telah Anda berikan:

date: Tanggal pengamatan atau catatan.

quarter: Kuartal atau periode waktu tertentu di dalam tahun.

department: Departemen tempat pekerja bekerja, seperti “sweing” atau “finishing”.

day: Hari dalam seminggu di mana pengamatan dilakukan, seperti “Thursday”.

team: Nomor tim atau kelompok kerja pekerja.

targeted_productivity: Produktivitas yang ditargetkan atau standar produktivitas yang diharapkan.

smv (Standard Minute Value): Nilai menit standar yang menggambarkan waktu standar yang diperlukan untuk menyelesaikan satu unit pekerjaan.

wip (Work in Progress): Jumlah pekerjaan yang sedang berlangsung atau belum selesai.

over_time: Jumlah waktu kerja tambahan yang dihabiskan.

incentive: Insentif atau bonus yang mungkin diberikan kepada pekerja.

idle_time: Waktu ketidakaktifan atau waktu yang tidak digunakan untuk produksi.

idle_men: Jumlah pekerjaan yang tidak aktif atau pekerja yang tidak produktif.

no_of_style_change: Jumlah perubahan gaya atau desain.

no_of_workers: Jumlah pekerja yang terlibat dalam produksi.

actual_productivity: Produktivitas aktual yang diukur atau dicatat.

```
# Load Library
library(keras)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.3      v readr      2.1.4
## v forcats    1.0.0      v stringr   1.5.0
## v ggplot2    3.4.3      v tibble    3.2.1
## v lubridate  1.9.2      v tidyr     1.3.0
## v purrr      1.0.1
## -- 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
```

```
library(Metrics)
```

Load Dataset

```
dataset <- read.csv("data/garments_worker_productivity.csv")
head(dataset)
```

```
##      date quarter department      day team targeted_productivity  smv  wip
## 1 1/1/2015 Quarter1  sweing Thursday    8          0.80 26.16 1108
## 2 1/1/2015 Quarter1 finishing Thursday    1          0.75  3.94   NA
## 3 1/1/2015 Quarter1  sweing Thursday   11          0.80 11.41  968
## 4 1/1/2015 Quarter1  sweing Thursday   12          0.80 11.41  968
## 5 1/1/2015 Quarter1  sweing Thursday    6          0.80 25.90 1170
## 6 1/1/2015 Quarter1  sweing Thursday    7          0.80 25.90  984
##  over_time incentive idle_time idle_men no_of_style_change no_of_workers
## 1         7080         98         0         0              0             59.0
## 2          960          0         0         0              0              8.0
## 3         3660         50         0         0              0             30.5
## 4         3660         50         0         0              0             30.5
## 5          1920         50         0         0              0             56.0
## 6          6720         38         0         0              0             56.0
##  actual_productivity
## 1          0.9407254
## 2          0.8865000
## 3          0.8005705
## 4          0.8005705
## 5          0.8003819
## 6          0.8001250
```

```
glimpse(dataset)
```

```
## Rows: 1,197
## Columns: 15
## $ date          <chr> "1/1/2015", "1/1/2015", "1/1/2015", "1/1/2015", ~
## $ quarter       <chr> "Quarter1", "Quarter1", "Quarter1", "Quarter1", ~
## $ department    <chr> "sweing", "finishing ", "sweing", "sweing", "swe~
## $ day           <chr> "Thursday", "Thursday", "Thursday", "Thursday", ~
## $ team          <int> 8, 1, 11, 12, 6, 7, 2, 3, 2, 1, 9, 10, 5, 10, 8,~
## $ targeted_productivity <dbl> 0.80, 0.75, 0.80, 0.80, 0.80, 0.80, 0.75, 0.75, ~
## $ smv           <dbl> 26.16, 3.94, 11.41, 11.41, 25.90, 25.90, 3.94, 2~
## $ wip           <int> 1108, NA, 968, 968, 1170, 984, NA, 795, 733, 681~
## $ over_time     <int> 7080, 960, 3660, 3660, 1920, 6720, 960, 6900, 60~
## $ incentive     <int> 98, 0, 50, 50, 50, 38, 0, 45, 34, 45, 44, 45, 50~
## $ idle_time     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ idle_men      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ no_of_style_change <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ no_of_workers  <dbl> 59.0, 8.0, 30.5, 30.5, 56.0, 56.0, 8.0, 57.5, 55~
## $ actual_productivity <dbl> 0.9407254, 0.8865000, 0.8005705, 0.8005705, 0.80~
```

EDA

```
# Melihat data missing value
colSums(is.na(dataset))
```

```
##           date           quarter      department
##           0              0              0
##           day            team targeted_productivity
##           0              0              0
##           smv            wip      over_time
##           0              506              0
##           incentive      idle_time      idle_men
##           0              0              0
## no_of_style_change no_of_workers actual_productivity
##           0              0              0
```

```
# Menghapus data missing value
dataset <- na.omit(dataset)
colSums(is.na(dataset))
```

```
##           date           quarter      department
##           0              0              0
##           day            team targeted_productivity
##           0              0              0
##           smv            wip      over_time
##           0              0              0
##           incentive      idle_time      idle_men
##           0              0              0
## no_of_style_change no_of_workers actual_productivity
##           0              0              0
```

```
glimpse(dataset)
```

```
## Rows: 691
## Columns: 15
## $ date          <chr> "1/1/2015", "1/1/2015", "1/1/2015", "1/1/2015", ~
## $ quarter       <chr> "Quarter1", "Quarter1", "Quarter1", "Quarter1", ~
## $ department    <chr> "sweing", "sweing", "sweing", "sweing", "sweing"~
## $ day           <chr> "Thursday", "Thursday", "Thursday", "Thursday", ~
## $ team          <int> 8, 11, 12, 6, 7, 3, 2, 1, 9, 10, 5, 4, 1, 3, 8, ~
## $ targeted_productivity <dbl> 0.80, 0.80, 0.80, 0.80, 0.80, 0.75, 0.75, 0.75, ~
## $ smv           <dbl> 26.16, 11.41, 11.41, 25.90, 25.90, 28.08, 19.87, ~
## $ wip           <int> 1108, 968, 968, 1170, 984, 795, 733, 681, 872, 5~
## $ over_time     <int> 7080, 3660, 3660, 1920, 6720, 6900, 6000, 6900, ~
## $ incentive     <int> 98, 50, 50, 50, 38, 45, 34, 45, 44, 45, 50, 0, 5~
## $ idle_time     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ idle_men      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ no_of_style_change <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ no_of_workers <dbl> 59.0, 30.5, 30.5, 56.0, 56.0, 57.5, 55.0, 57.5, ~
## $ actual_productivity <dbl> 0.9407254, 0.8005705, 0.8005705, 0.8003819, 0.80~
```

Partisi Data

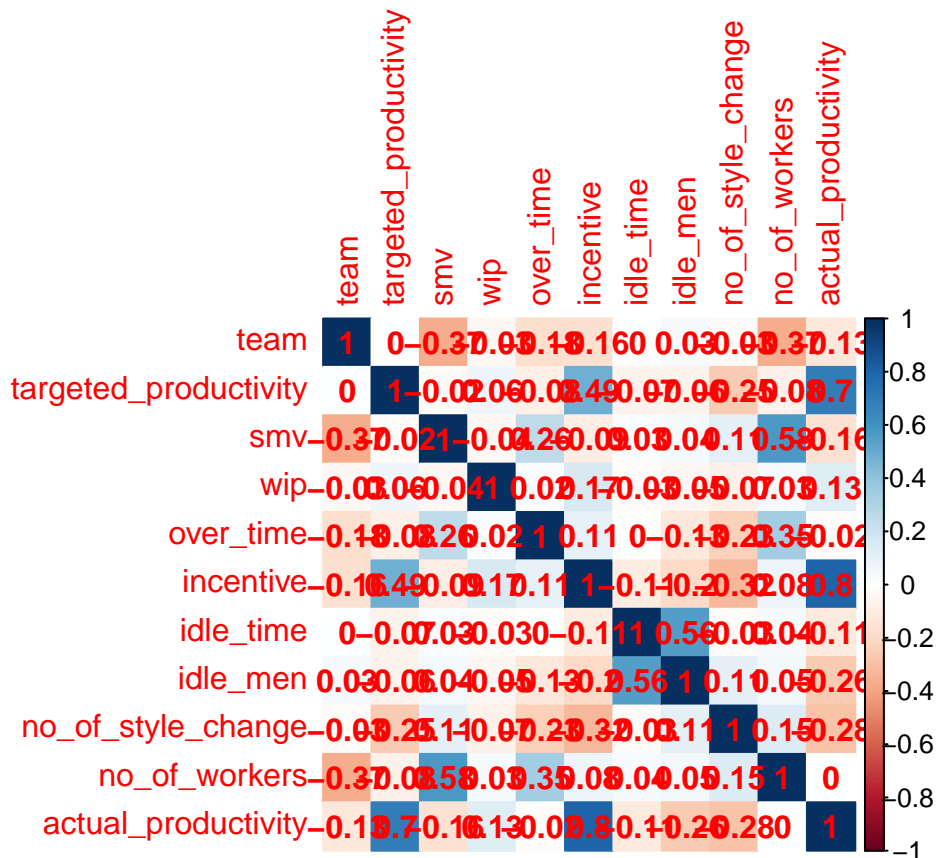
```
# partisi dataset menjadi data training dan testing
train_data <- dataset[1:400,]
test_data <- dataset[401:691,]
```

Feature Importance Analysis

```
# Uji Korelasi
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
dataset_cor <- cor(dataset[,5:15])
corrplot(dataset_cor, method = "color", addCoef.col = "red")
```



Model

```
# Menentukan variabel respons dan prediktor
response_variable <- "actual_productivity"
predictor_variables <- c("targeted_productivity", "smv", "wip", "incentive", "idle_men")

# Membagi dataset menjadi set pelatihan dan set pengujian (sudah terpisah)
set.seed(123) # Untuk reproduktibilitas
training_data <- train_data
testing_data <- test_data

# Membuat model ANN
model <- keras_model_sequential() %>%
  layer_dense(units = 64, activation = "relu", input_shape = length(predictor_variables)) %>%
  layer_dense(units = 32, activation = "relu") %>%
  layer_dense(units = 1,) # Output layer tanpa aktivasi untuk regresi

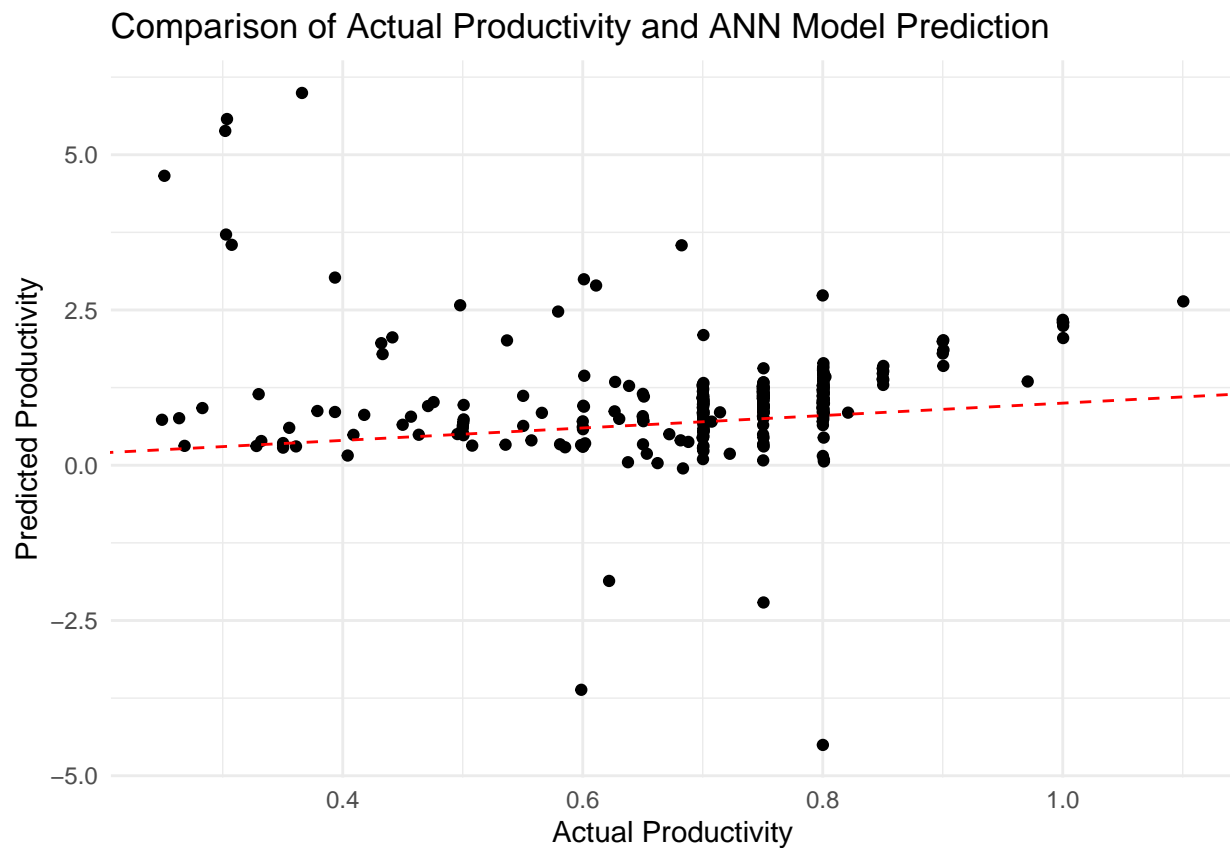
# Menentukan fungsi kerugian (loss function) dan optimizer
model %>% compile(
  loss = "mean_squared_error",
  optimizer = optimizer_adam(),
  metrics = c("mean_squared_error")
)
```

```
# Melatih model
history <- model %>% fit(
  x = as.matrix(training_data[, predictor_variables]),
  y = training_data[[response_variable]],
  epochs = 100, # Jumlah iterasi pelatihan
  validation_split = 0.2, # Pembagian data validasi
  verbose = 1
)
```

```
# Melakukan prediksi pada set pengujian
predictions_ann <- predict(model, as.matrix(testing_data[, predictor_variables]))
```

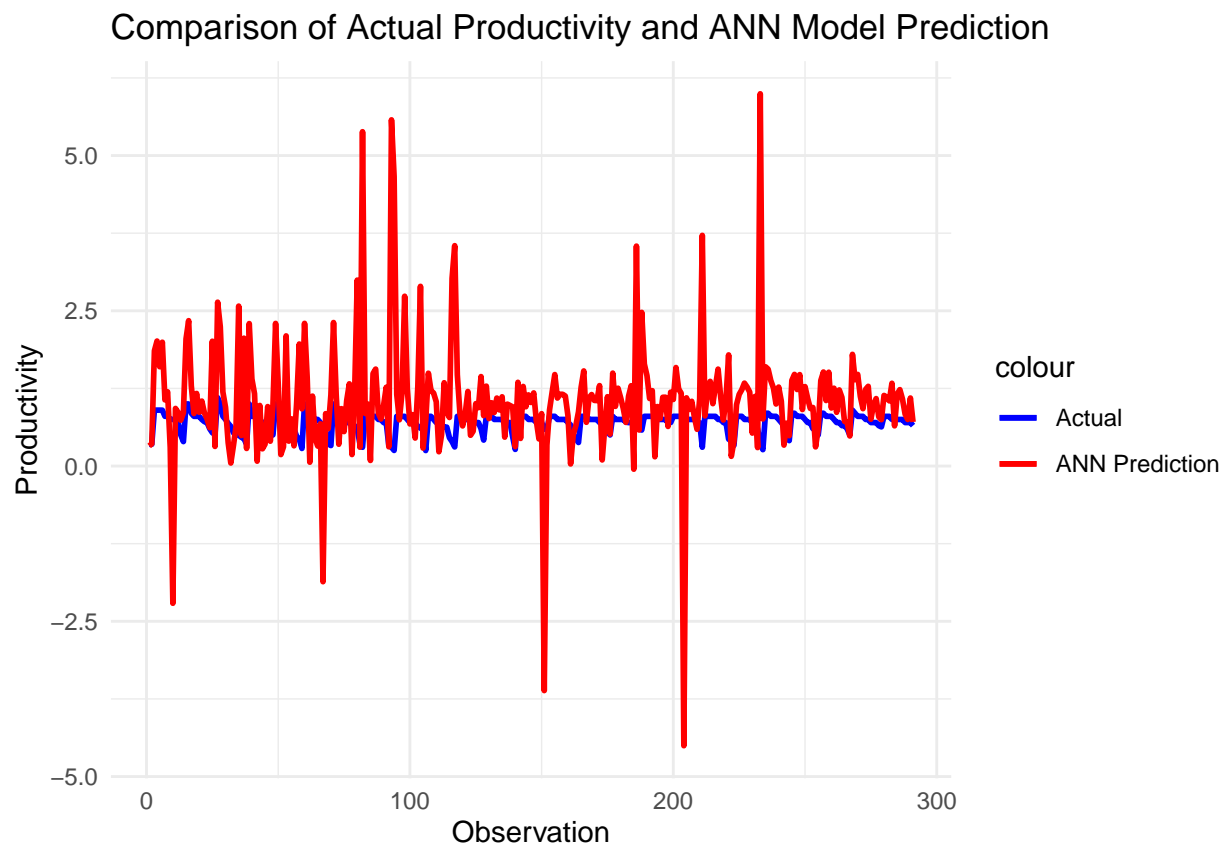
```
# Visualisasi hasil prediksi
result_data_ann <- data.frame(Productivity = testing_data$actual_productivity, Predictions = predictions_ann)

ggplot(data = result_data_ann, aes(x = Productivity, y = Predictions)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1, color = "red", linetype = "dashed") +
  labs(x = "Actual Productivity", y = "Predicted Productivity") +
  scale_x_continuous(labels = scales::comma) +
  scale_y_continuous(labels = scales::comma) +
  ggtitle("Comparison of Actual Productivity and ANN Model Prediction") +
  theme_minimal()
```



```
# Buat grafik garis
ggplot(data = result_data_ann, aes(x = 1:length(testing_data$actual_productivity))) +
  geom_line(aes(y = testing_data$actual_productivity, color = "Actual"), size = 1) +
  geom_line(aes(y = predictions_ann, color = "ANN Prediction"), size = 1) +
  labs(x = "Observation", y = "Productivity") +
  scale_color_manual(values = c("Actual" = "blue", "ANN Prediction" = "red")) +
  ggtitle("Comparison of Actual Productivity and ANN Model Prediction") +
  scale_y_continuous(labels = scales::comma) +
  theme_minimal()
```

```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```



```
head(result_data_ann, 10)
```

```
##      Productivity Predictions
## 1      0.3503017    0.3238916
## 2      0.3503017    0.3572130
## 3      0.9006324    1.8563204
## 4      0.9004708    2.0133677
## 5      0.9004708    1.5992403
```



```
## 6      0.8999841    1.9941988
## 7      0.8008000    1.0655298
## 8      0.8004020    1.1955729
## 9      0.7507970    0.3024377
## 10     0.7506481   -2.2100511
```

Evaluasi Model

```
# Evaluasi ANN Model

# Hitung MAE
mae_value_ann <- mae(testing_data$actual_productivity, predictions_ann)

# Hitung MSE
mse_value_ann <- mse(testing_data$actual_productivity, predictions_ann)

# Hitung RMSE
rmse_value_ann <- rmse(testing_data$actual_productivity, predictions_ann)

# Hitung MAPE
mape_value_ann <- mape(testing_data$actual_productivity, predictions_ann)

# Tampilkan hasil evaluasi
cat(paste("MAE: ", mae_value_ann, "\n"))
```

```
## MAE:  0.574012043393865
```

```
cat(paste("MSE: ", mse_value_ann, "\n"))
```

```
## MSE:  1.01341394741761
```

```
cat(paste("RMSE: ", rmse_value_ann, "\n"))
```

```
## RMSE:  1.00668463155926
```

```
cat(paste("MAPE: ", mape_value_ann, "%\n"))
```

```
## MAPE:  1.01804983652251 %
```