Generalized Regression Neural Network

Abdi Negara Guci

May 3, 2024

1 Pendahuluan

GRNN atau General Regression Neural Network adalah suatu arsitektur jaringan syaraf tiruan (neural network) yang dirancang khusus untuk melakukan tugas regresi. GRNN merupakan salah satu jenis jaringan radial basis function (RBF). GRNN cocok untuk masalah regresi dengan jumlah data pelatihan yang besar. Kelebihannya adalah konvergensi cepat dan tidak memerlukan iterasi seperti backpropagation. Namun, GRNN membutuhkan lebih banyak memori dan komputasi saat digunakan untuk data berdimensi tinggi.

General Regression Neural Network (GRNN) pertama kali dikembangkan oleh Specht pada tahun 1991 . GRNN merupakan feedforward neural network ,dimana prosesnya adalah untuk merespon lapisan input dengan memproses datainput dari satu lapisan ke lapisan selanjutnya tanpa adanya jalur balik dan juga merupakan salah satu dari jaringan radial basis yang digunakan pada pendekatan suatu fungsi. GRNN didasarkan pada teori regresi non linear (kernel) dengan estimasi nilai harapan dari output ditentukan dari himpunan input . Metode ini dapat digunakan untuk melakukan peramalan atau prediksi pada data historis dengan hasil dari variabel output didasarkan p ada minimal satu variabel input .

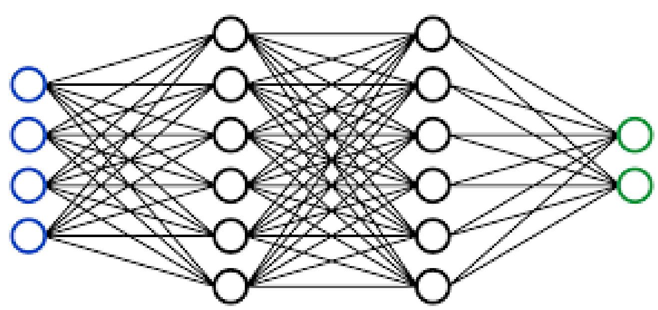

Figure 1: Gambar Neural Network.

2 Karakteristik Model

2.1 Fungsi Basis Radial (RBF)

GRNN menggunakan fungsi basis radial untuk mentransformasi data masukan ke dalam ruang fitur berdimensi lebih tinggi. Fungsi basis radial adalah fungsi matematika yang bergantung hanya pada jarak dari titik pusat, yang dikenal sebagai pusat atau prototipe. Fungsi basis radial yang paling umum digunakan dalam GRNN adalah fungsi Gaussian.

2.2 Pembelajaran Sekali Jalan

GRNN melakukan proses pembelajaran sekali jalan, yang berarti mereka belajar set data latih segera setelah disajikan. Hal ini membuatnya efisien terutama untuk tugas regresi dengan kumpulan data berukuran kecil hingga sedang.

2.3 Pelatihan Cepat

Pelatihan GRNN umumnya lebih cepat dibandingkan dengan jenis jaringan saraf lainnya karena tidak melibatkan proses optimisasi iteratif seperti gradien turun. Sebaliknya, ia menghitung bobot selama pelatihan berdasarkan data pelatihan dan menyimpannya untuk digunakan nanti selama prediksi.

2.4 Pendekatan Universal

Seperti banyak arsitektur jaringan saraf lainnya, GRNN adalah pendekatan fungsi universal, yang berarti mereka memiliki kapasitas untuk memperkirakan fungsi kontinu apa pun dengan akurasi sewenang-wenang dengan jumlah neuron tersembunyi yang cukup.

2.5 Tugas Regresi

Sementara beberapa jaringan saraf dirancang untuk tugas klasifikasi (di mana keluarannya adalah kategori atau label), GRNN secara khusus ditujukan untuk tugas regresi, di mana tujuannya adalah untuk memprediksi variabel keluaran kontinu berdasarkan fitur masukan.

3 Pendekatan Matematika

3.1 Nilai Ekspektasi

Persamaan (1) menunjukkan persamaan nilai ekspektasi y terhadap $\mathbf{X}$ (atau) juga disebut regresi y pada $\mathbf{X}$.

$$E[y \mid X] = \frac{\int_{-\infty}^{\infty} y f(X, y)\, dy}{\int_{-\infty}^{\infty} f(X, y)\, dy} \tag{1}$$

dengan keterangan y merupakan nilai prediksi atau nilai output yang dihasilkan $\mathbf{X}$ merupakan vektor input $(x_1, x_2, ..., x_n)$ dengan n adalah jumlah dari variabel input atau variabel prediktor , $E[y \mid X]$ merupakan nilai ekpektasi dari output y berdasarkan vektor $\mathbf{X}$ dan $f(\mathbf{X}, y)$ merupakan fungsi kepadatan probabilitas probability density function /pdf) kontinu gabungan antara $\mathbf{X}$ dan y

Pembuktian:

3.2 Regresi Non-Parametrik Parzen

Ketika $f(\mathbf{X}, y)$ tidak diketahui, biasanya akan diestimasi dari sampel observasi $\mathbf{x}$ dan y. Untuk estimasi non-parametrik $f(\mathbf{X}, y)$, dapat digunakan estimasi yang dikembangkan oleh Parzen, yang ditunjukkan dengan persamaan (2).

$$\hat{f}(X, y) = \frac{1}{(2\pi)^{\frac{p+1}{2}} \sigma^{p+1}} \cdot \frac{1}{n} \sum_{i=1}^{n} \exp\left[-\frac{(X - X_i)^T (X - X_i)}{2\sigma^2}\right] \cdot \exp\left[-\frac{(y - y_i)^2}{2\sigma^2}\right] \tag{2}$$

dengan estimasi probabilitas $\hat{f}(X, y)$ berdasarkan pada nilai sampel X_i dan y_i dari nilai random X dan y. Kemudian, n adalah jumlah sampel oservasi, p adalah dimensi dari vektor variabel $\mathbf{x}$, dan σ adalah paramater smoothing.

Pembuktian:

- **Bentuk Dasar Estimasi Parzen:** Estimasi Parzen Window adalah salah satu metode non-parametrik yang digunakan untuk mengestimasi probabilitas keberadaan suatu titik dalam ruang fitur. Pada dasarnya, Parzen Window membagi ruang fitur menjadi jendela-jendela kecil dan menghitung jumlah titik data yang jatuh di dalam jendela tersebut.
- **Kernel Density Estimation (KDE):** Estimasi kerapatan kernel (KDE) merupakan metode yang digunakan dalam Parzen Window. Pada KDE, titik data disebarkan di sekitar titik pengamatan dengan menggunakan fungsi kernel. Fungsi kernel ini menunjukkan seberapa besar pengaruh setiap titik data pada titik pengamatan.

- **Fungsi Kernel Gaussian:** Pada estimasi Parzen Window, fungsi kernel yang umum digunakan adalah Gaussian. Fungsi kernel Gaussian didefinisikan sebagai berikut:

$$K(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right) \tag{3}$$

di mana σ adalah lebar kernel (bandwidth) yang menentukan seberapa besar jangkauan pengaruh setiap titik data.

dengan mempertimbangkan kasus dua dimensi, di mana kita memiliki variabel $\mathbf{X}$ dan y, sehingga persamaan (3) akan ditransformasikan menjadi dua dimensi, kita akan mengganti x dengan vektor $\mathbf{X}$ yang memiliki dua dimensi. Jadi, kita memiliki $\mathbf{X} = (x_1, x_2)$. Dengan menggunakan norma Euclidean untuk vektor, kita memiliki $\|\mathbf{X}\|^2 = x_1^2 + x_2^2$. Kita juga akan mengganti akar kuadrat tunggal dengan akar kuadrat dari penjumlahan kuadrat elemen vektor.

$$\begin{aligned}
K(\mathbf{X}) &= \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\|\mathbf{X}\|^2}{2\sigma^2}\right) \\
&= \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x_1^2 + x_2^2}{2\sigma^2}\right) \\
&= \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x_1^2}{2\sigma^2}\right) \cdot \exp\left(-\frac{x_2^2}{2\sigma^2}\right) \\
&= \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x_1^2}{2\sigma^2}\right) \cdot \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x_2^2}{2\sigma^2}\right) \\
&= \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \cdot \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp\left(-\frac{x_1^2}{2\sigma^2}\right) \exp\left(-\frac{x_2^2}{2\sigma^2}\right) \\
&= \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x_1^2}{2\sigma^2}\right) \exp\left(-\frac{x_2^2}{2\sigma^2}\right) \\
&= \frac{1}{2\pi\sigma^2} \exp\left(-\frac{\|\mathbf{X}\|^2}{2\sigma^2}\right)
\end{aligned} \tag{4}$$

Sekarang, kita akan menggunakan fungsi kernel ini dalam estimasi Parzen Window untuk menghitung probabilitas di titik (X, y). Estimasi Parzen Window adalah rata-rata dari fungsi kernel Gaussian yang diberikan oleh setiap titik data, dituliskan sebagai berikut:

$$\hat{f}(X, y) = \frac{1}{n} \sum_{i=1}^{n} K(X - X_i, y - y_i) \tag{5}$$

Dengan menggantikan $K(X - X_i, y - y_i)$ dengan fungsi kernel Gaussian yang telah diperluas ke dua dimensi, kita peroleh:

$$\begin{aligned}
\hat{f}(X, y) &= \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2\pi\sigma^2} \exp\left(-\frac{\|X - X_i\|^2 + (y - y_i)^2}{2\sigma^2}\right) \\
&= \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \cdot \frac{1}{n} \sum_{i=1}^{n} \exp\left(-\frac{\|X - X_i\|^2 + (y - y_i)^2}{2\sigma^2}\right) \\
&= \frac{1}{(2\pi)^{\frac{p+1}{2}} \sigma^{p+1}} \cdot \frac{1}{n} \sum_{i=1}^{n} \exp\left[-\frac{(X - X_i)^T (X - X_i)}{2\sigma^2}\right] \cdot \exp\left[-\frac{(y - y_i)^2}{2\sigma^2}\right]
\end{aligned} \tag{6}$$

Ini menunjukkan bahwa rumus yang diberikan sesuai dengan estimasi Parzen menggunakan kernel Gaussian.

3.3 Persamaan Umum Model GRNN

Dengan melakukan subsitusi pada persamaan (2) ke persamaan (1), sehingga diperoleh persamaan umum GRNN sebagai berikut.

$$\hat{y}(X) = \frac{\sum_{i=1}^{n} y_i \exp\left[-\frac{(D_i^2)}{2\sigma^2}\right]}{\sum_{i=1}^{n} \exp\left[-\frac{(D_i^2)}{2\sigma^2}\right]} \tag{7}$$

Pembuktian:

Dari persamaan (1) kita tau bahwa nilai ekspektasi y terhadap suatu vektor X didefinisikan sebagai berikut

$$E[y \mid X] = \frac{\int_{-\infty}^{\infty} y f(X,y)\, dy}{\int_{-\infty}^{\infty} f(X,y)\, dy}$$

dengan melakukan subtitusi persamaan (2) kedalam rumus diatas, diperleh persamaan sebagai berikut.

$$E[y \mid X] = \frac{\int_{-\infty}^{\infty} \frac{1}{(2\pi)^{\frac{p+1}{2}}\sigma^{p+1}} \frac{1}{n} \sum_{i=1}^{n} \exp\left[-\frac{(X-X_i)^T(X-X_i)}{2\sigma^2}\right] \exp\left[-\frac{(y-y_i)^2}{2\sigma^2}\right] y\, dy}{\int_{-\infty}^{\infty} \frac{1}{(2\pi)^{\frac{p+1}{2}}\sigma^{p+1}} \frac{1}{n} \sum_{i=1}^{n} \exp\left[-\frac{(X-X_i)^T(X-X_i)}{2\sigma^2}\right] \exp\left[-\frac{(y-y_i)^2}{2\sigma^2}\right] dy} \tag{8}$$

$$E[y \mid X] = \frac{\frac{1}{(2\pi)^{\frac{p+1}{2}}\sigma^{p+1}} \frac{1}{n} \sum_{i=1}^{n} \exp\left[-\frac{(X-X_i)^T(X-X_i)}{2\sigma^2}\right] \int_{-\infty}^{\infty} \exp\left[-\frac{(y-y_i)^2}{2\sigma^2}\right] y\, dy}{\frac{1}{(2\pi)^{\frac{p+1}{2}}\sigma^{p+1}} \frac{1}{n} \sum_{i=1}^{n} \exp\left[-\frac{(X-X_i)^T(X-X_i)}{2\sigma^2}\right] \int_{-\infty}^{\infty} \exp\left[-\frac{(y-y_i)^2}{2\sigma^2}\right] dy} \tag{9}$$

$$E[y \mid X] = \frac{\sum_{i=1}^{n} \exp\left[-\frac{(X-X_i)^T(X-X_i)}{2\sigma^2}\right] \int_{-\infty}^{\infty} \exp\left[-\frac{(y-y_i)^2}{2\sigma^2}\right] y\, dy}{\sum_{i=1}^{n} \exp\left[-\frac{(X-X_i)^T(X-X_i)}{2\sigma^2}\right] \int_{-\infty}^{\infty} \exp\left[-\frac{(y-y_i)^2}{2\sigma^2}\right] dy} \tag{10}$$

Lebih lanjut, pada persamaan (8), akan dilakukan pengintegralan pada nilai $\frac{\int_{-\infty}^{\infty} \exp\left[-\frac{(y-y_i)^2}{2\sigma^2}\right] y\, dy}{\int_{-\infty}^{\infty} \exp\left[-\frac{(y-y_i)^2}{2\sigma^2}\right] dy}$. Perhatikan bahwa integral tersebut melibatkan distribusi Gaussian atau distribusi normal. Integral dari fungsi Gaussian atau distribusi normal adalah standar dan dapat dinyatakan sebagai berikut.

$$\int_{-\infty}^{\infty} \exp\left[-\frac{x^2}{2\sigma^2}\right] dx = \sqrt{2\pi\sigma^2} \tag{11}$$

Sekarang mari kita terapkan informasi pada persamaan (11) kedalam integral yang diberikan. Mari kita sebutkan integral atas sebagai I_1 dan integral bawah sebagai I_2.

$$I_1 = \int_{-\infty}^{\infty} \exp\left[-\frac{(y-y_i)^2}{2\sigma^2}\right] y\, dy \tag{12}$$

$$I_2 = \int_{-\infty}^{\infty} \exp\left[-\frac{(y-y_i)^2}{2\sigma^2}\right] dy \tag{13}$$

Dengan menggunakan metode integral subtitusi, misalkan $u = \frac{y-y_i}{\sqrt{2}\sigma}$, sehingga $du = \frac{dy}{\sqrt{2}\sigma}$. Maka kita akan memperoleh:

$$\begin{aligned} I_1 &= \int_{-\infty}^{\infty} e^{-u^2} (\sqrt{2}\sigma u + y_i)\, du \\ &= \sqrt{2}\sigma \int_{-\infty}^{\infty} u e^{-u^2}\, du + y_i \int_{-\infty}^{\infty} e^{-u^2}\, du \\ &= 0 + y_i\sqrt{2\pi\sigma^2} \\ &= y_i\sqrt{2\pi}\sigma \end{aligned} \tag{14}$$

Dengan menggunakan fakta pada persamaan (14), sehingga diperoleh bahwa

$$\frac{y_i\sqrt{2\pi}\sigma}{\sqrt{2\pi}\sigma} = y_i$$

Jadi, hasil integral tersebut adalah y_i. Kemudian subtitusikan hasil ini kedalam persamaan (10), sehingga diperoleh persamaan sebagai berikut.

$$E[y \mid X] = \frac{\sum_{i=1}^{n} \exp\left[-\frac{(X-X_i)^T(X-X_i)}{2\sigma^2}\right](y_i)}{\sum_{i=1}^{n} \exp\left[-\frac{(X-X_i)^T(X-X_i)}{2\sigma^2}\right]} \tag{15}$$

dengan memisalkan $D_i^2 = (X - X_i)^T(X - X_i)$ pada persamaan (15), sehingga diperoleh rumus Umum GRNN yang sesuai dengan persamaan (7) yaitu.

$$E[y \mid X] = \frac{\sum_{i=1}^{n} y_i \exp\left[-\frac{(D_i^2)}{2\sigma^2}\right]}{\sum_{i=1}^{n} \exp\left[-\frac{(D_i^2)}{2\sigma^2}\right]}$$

$$\hat{y}(X) = \frac{\sum_{i=1}^{n} y_i \exp\left[-\frac{(D_i^2)}{2\sigma^2}\right]}{\sum_{i=1}^{n} \exp\left[-\frac{(D_i^2)}{2\sigma^2}\right]}$$

4 Model Arsitektur GRNN

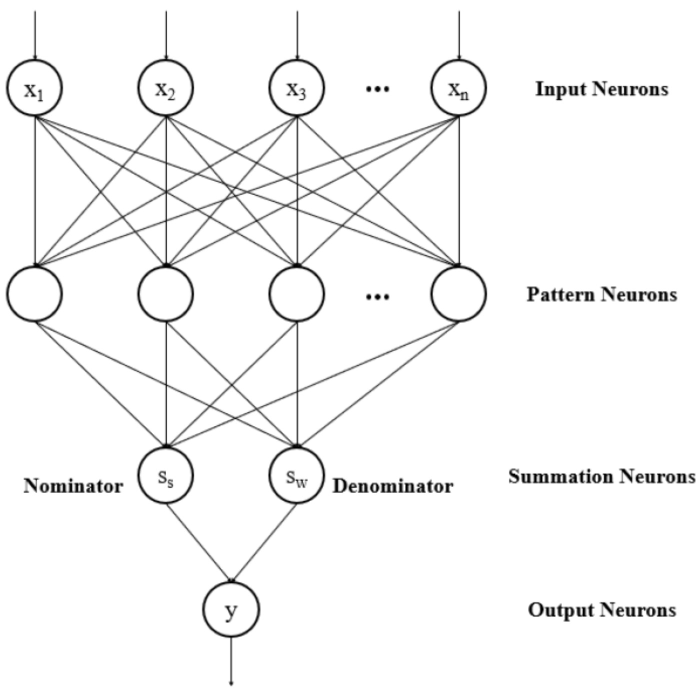

Figure 2: Gambar Arsitektur GRNN.

- **Input Layer:**

  Input Neuron digunakan untuk menerima informasi. Pada lapisan ini data input dari masing masing variabel prediktor akan diletakkan dan tidak ada proses apapun di neuron ini. Kemudian data dari neuron input

- **Pattern Layer:**

  Pada neuron pola, dilakukan proses menggabungkan dan memproses data secara sistematik. Jumlah neuron pada lapisan ini adalah sejumlah data yang ada pada data training . Terdapat fungsi aktivasi untuk melakukan transfer sinyal jaringan pada lapisan ini, dimana fungsi aktivasi tersebut dapat dihitung hasilnya dengan menggunakan persamaan (16) dibawah ini.

$$\theta = \exp\left[-\frac{(X - X_i)^T(X - X_i)}{2\sigma^2}\right] \tag{16}$$

  dengan X adalah vektor input pada variabel prediktor, X_i adalah vektor training yang diwakili oleh neuron pola i, dan θ adalah parameter smoothing akan ditransfer menuju neuron selanjutnya.

- **Summation Layer:**

  Setelah dari neuron pola, hasilnya akan diteruskan ke neuroan penjumlahan. Pada neuron ini, terdapat dua jenis penjumlahan, yaitu penjumlahan aritmatika sederhana dan penjumlahan terboboti. Dua jenis penjumlahan ini di lakukan secara terpisah.

$$S_w = \sum_{i=1}^{n} \theta_i(y_i) \tag{17}$$

$$S_s = \sum_{i=1}^{n} \theta_i \tag{18}$$

  ,

- **Output layer:**

  Neuron terakhir adalah neuron output, dimana pada neuron ini, terjadi pembagian antar penjumlahan dari neuron penjumlahan. Dari pembagian ini a kan dihasilkan output y sebagai hasil GRNN dan ditunjukkan dengan persamaan berikut.

$$y = \frac{S_w}{S_s} \tag{19}$$

5 Contoh Soal Manual

Diberikan tabel berikut yang memuat data tentang berat badan (kg), tinggi (cm), dan umur (tahun) dari beberapa individu:

No.	Berat (kg)	Tinggi (cm)	Umur (tahun)
1	55	165	30
2	70	180	45
3	62	170	35
4	75	175	40
5	60	160	28

Dalam konteks metode GRNN (General Regression Neural Network), kita diminta untuk membagi data baris 1 hingga 4 menjadi set pelatihan (train set), sementara data baris 5 akan dijadikan set pengujian (test set).

Langkah-langkah:

- **Menentukan Training set dan Test set:** Berdasarkan soal yang diberika, kita tau bahwa baris kelima merupakan test set, oleh karena itu baris ke 1-4 merupakan training set.

No.	Berat (kg)	Tinggi (cm)	Umur (tahun) (y)	Keterangan
1	55	165	30	Training Set
2	70	180	45	Training Set
3	62	170	35	Training Set
4	75	175	40	Training Set
5	60	160	28	Test Set

- **Input Layer:** Berdasarkan perolehan data pada poin sebelumnya, selas bahwa hasil inputannya adalah sebagai berikut.

$$X = \begin{pmatrix} 55 & 165 \\ 70 & 180 \\ 62 & 170 \\ 75 & 175 \end{pmatrix}, X_i = \begin{pmatrix} 60 & 160 \end{pmatrix}, y = \begin{pmatrix} 30 \\ 45 \\ 35 \\ 40 \end{pmatrix}, y_i = 28 \tag{20}$$

- **Pattern Layer:** Berdasarkan nilai Input layer, maka diperoleh:

$$\theta^{(5)} = \exp\left[-\frac{(X - X_i)^T (X - X_i)}{2\sigma^2}\right] \tag{21}$$

$$\theta^{(5)} = \exp\left[-\frac{\left(\begin{pmatrix} 55 & 165 \\ 70 & 180 \\ 62 & 170 \\ 75 & 175 \end{pmatrix} - \begin{pmatrix} 60 & 160 \end{pmatrix}\right)^T \left(\begin{pmatrix} 55 & 165 \\ 70 & 180 \\ 62 & 170 \\ 75 & 175 \end{pmatrix} - \begin{pmatrix} 60 & 160 \end{pmatrix}\right)}{2\sigma^2}\right] \tag{22}$$

$$\theta^{(5)} = \exp\left[-\frac{\begin{pmatrix} -5 & 5 \\ 10 & 20 \\ 2 & 10 \\ 15 & 15 \end{pmatrix}^T \begin{pmatrix} -5 & 5 \\ 10 & 20 \\ 2 & 10 \\ 15 & 15 \end{pmatrix}}{2\sigma^2}\right] \tag{23}$$

$$\theta^{(5)} = \exp\left[-\frac{\begin{pmatrix} 5 & 20 & 10 & 15 \\ -5 & 10 & 2 & 15 \end{pmatrix} \begin{pmatrix} 5 & -5 \\ 20 & 10 \\ 10 & 2 \\ 15 & 15 \end{pmatrix}}{2\sigma^2}\right] \tag{24}$$

$$\theta^{(5)} = \exp\left[-\frac{\begin{pmatrix} 50 & 500 & 104 & 450 \end{pmatrix}}{2\sigma^2}\right] \tag{25}$$

pilih nilai θ dalam rentang (0-1) dalam kasus ini, untuk mempermudah perhitungan maka akan dipilih 1. Sehingga diperoleh nilai berikut.

$$\theta^{(5)} = \begin{pmatrix} \exp(-25) & \exp(-250) & \exp(-52) & \exp(-225) \end{pmatrix} \tag{26}$$

- **Summation Layer:** Masukan nilai yang telah diperoleh dari pattern layer. Hasil kalkulasi nilai S_s adalah sebagai berikut.

$$S_s = \sum_{i=1}^{n} \theta_i \tag{27}$$

$$S_s = (\exp(-25) + \exp(-250) + \exp(-52) + \exp(-225)) \tag{28}$$

$$S_s = 1.39 \times 10^{-12} \tag{29}$$

$$\tag{30}$$

Sedangkan hasil kalkulasi nilai S_w adalah sebagai berikut.

$$
\begin{aligned}
S_w &= \sum_{i=1}^{n} \theta_i(y_{i)}) \\
S_w &= \sum_{i=1}^{n} \left(\begin{pmatrix}\exp(-25) & \exp(-250) & \exp(-52) & \exp(-225)\end{pmatrix}\right)\begin{pmatrix}30 & 45 & 35 & 40\end{pmatrix} \\
S_w &= \sum_{i=1}^{n} \left(\begin{pmatrix}30\exp(-25) & 45\exp(-250) & 35\exp(-52) & 40\exp(-225)\end{pmatrix}\right) \\
S_w &= (30\exp(-25) + 45\exp(-250) + 35\exp(-52) + 40\exp(-225)) \\
S_w &= 4.1 \times 10^{-12}
\end{aligned}
$$

- **Output Layer:** Berdasarkan hasil yang telah diperoleh sebelumnya pada summation layer, sehingga diperoleh

$$
\begin{aligned}
\hat{y}^{(5)} &= \frac{S_w}{S_s} \\
\hat{y}^{(5)} &= \frac{4.1 \times 10^{-12}}{} \\
\hat{y}^{(5)} &= 1.39 \times 10^{-12} \\
\hat{y}^{(5)} &= 2.949 \approx 30
\end{aligned}
$$

- **Menghitung Mean Square Error:**Setelah diperoleh Nilai Prediksi, selanjutnya melakukan evaluasi model dengan menghitung MSE.

$$
\begin{aligned}
MSE &= \frac{(\hat{y} - y)}{y_{Size}} \\
MSE &= \frac{(30 - 28)}{2} \\
MSE &= 0.5
\end{aligned}
$$

Dengan mengetahui bahwa nilai MSE berada $0 < x < 1$ ini berarti model GRNN baik dalam memprediksi data $y^{(5)}$.

6 Implementasi Python

```
import numpy as np
import math
from sklearn.model_selection import train_test_split
from tqdm import tqdm

class GRNN:

    def __init__(self, x_train, y_train, x_test, y_test):
        self.x_train = x_train
        self.y_train = y_train
        self.x_test = x_test
        self.y_test = y_test
        self.std = np.ones((1, self.y_train.size))  # Standard deviations

    def activation_func(self, distances):  # Gaussian kernel
        return np.exp(-(distances**2) / (2 * (self.std**2)))

    def output(self, i):
```

```
        distances = np.sqrt(np.sum((self.x_test[i] - self.x_train)**2, axis=1))  # Euclidean distance
        return self.activation_func(distances)

    def denominator(self, i):
        return np.sum(self.output(i))

    def numerator(self, i):
        return np.sum(self.output(i) * self.y_train)

    def predict(self):
        predict_array = np.array([])

        for i in tqdm(range(self.y_test.size), desc="Predicting"):
            predict = np.array([self.numerator(i) / self.denominator(i)])
            predict_array = np.append(predict_array, predict)

        return predict_array

    def mean_squared_error(self):
        return (self.predict() - self.y_test)**2 / self.y_test.size

    def root_mean_squared_error(self):
        return np.sqrt(self.mean_squared_error())
```

Penjelasan Kodingan Python:

1. `__init__`:
   - Ini adalah metode konstruktor untuk kelas. Menerima empat parameter: `x_train`, `y_train`, `x_test`, dan `y_test`.
   - `x_train`, `y_train`, `x_test`, dan `y_test` adalah data pelatihan dan pengujian.
   - `std` diinisialisasi sebagai matriks dengan ukuran 1xN, di mana N adalah jumlah sampel pelatihan. Ini adalah standar deviasi untuk kernel Gauss yang digunakan dalam algoritma.
2. `activation_func`:
   - Metode ini menerima parameter `distances`, yang merupakan jarak antara titik pengujian tertentu dan semua titik pelatihan.
   - Mengembalikan nilai dari fungsi aktivasi, yang merupakan kernel Gauss, diterapkan pada setiap jarak.
3. `output`:
   - Metode ini menghitung keluaran (aktivasi) untuk satu titik pengujian.
   - Menghitung jarak Euclidean antara titik pengujian dan semua titik pelatihan, kemudian menerapkan fungsi aktivasi pada jarak tersebut.
4. `denominator`:
   - Metode ini menghitung penyebut dalam rumus prediksi.
   - Ini adalah jumlah dari semua aktivasi untuk titik pengujian tertentu.
5. `numerator`:
   - Metode ini menghitung pembilang dalam rumus prediksi.
   - Ini adalah jumlah dari hasil perkalian antara aktivasi dan nilai target untuk titik pelatihan tertentu.
6. `predict`:

- Metode ini digunakan untuk melakukan prediksi pada semua titik pengujian.
- Menggunakan rumus prediksi untuk setiap titik pengujian dan menyimpan hasilnya dalam sebuah array.

7. `mean_squared_error`:
    - Metode ini menghitung rata-rata kesalahan kuadrat antara nilai prediksi dan nilai target pada set data pengujian.
8. `root_mean_squared_error`:
    - Metode ini menghitung akar rata-rata kesalahan kuadrat antara nilai prediksi dan nilai target pada set data pengujian.