

ESTIMASI KUAT TEKAN BETON 28 HARI MENGUNAKAN ALGORITMA NEURAL NETWORK

Chandra Hilmawan¹

¹Universitas Adhirajasa Reswara Sanjaya
Jl. Sekolah Internasional No1-2, Antapani, Bandung
e-mail: chandrahilmawan98@gmail.com

Abstrak

Kuat tekan beton berperan sangat penting dalam pembangunan infrastruktur, beton sendiri merupakan bahan yang diperoleh dari campuran semen hidrolik, agregat kasar, agregat halus, air dan bahan tambahan lainnya. Akan tetapi tidak dapat ditentukan secara pasti ketahanan beton tersebut, maka dari itu salah satu cara untuk mengetahuinya menggunakan aplikasi *rapid miner* dengan perhitungan estimasi menggunakan algoritma *neural network*. Akurasi yang dihasilkan dari algoritma *neural network* terhadap dataset *concrete compressive strength* menggunakan *split validation* paling baik dihasilkan pada rasio 90% *data training* dan 10% *data testing* dengan *root mean square error* sebesar 6.782.

Kata Kunci: *Estimasi, Kuat Tekan Beton, Rapid Miner, Neural Network.*

Abstract

The compressive strength of concrete plays a very important role in infrastructure development, concrete itself is a material obtained from a mixture of hydraulic cement, coarse aggregate, fine aggregate, water and other additives. However, it cannot be determined with certainty the resistance of the concrete, therefore one way to find out is using the rapid miner application by calculating estimates using the neural network algorithm. The accuracy generated from the neural network algorithm on the concrete compressive strength dataset using split validation is best generated in ratio of 90% training data and 10% testing data with a root mean square error of 6,782.

Keywords: *Estimation, Concrete Compressive Strength, Rapid Miner, Neural Network.*

Keywords: FLO, Acceptance Analysis, Technology Acceptance Model

1. Pendahuluan

Pembangunan infrastruktur merupakan salah satu aspek penting dan vital untuk mempercepat proses pembangunan nasional. Infrastruktur juga memegang peranan penting sebagai salah satu roda penggerak pertumbuhan ekonomi (Direktorat Riset dan Pengabdian Masyarakat UI, 2017). Gerak laju dan pertumbuhan ekonomi suatu negara tidak dapat pisahkan dari ketersediaan infrastruktur seperti transportasi, lalu lintas dan jalan.

Menurut Undang-Undang Republik Indonesia No. 22 Tahun 2009 tentang Lalu Lintas dan Angkutan jalan, pemerintah wajib menyediakan fasilitas pendukung penyelenggaraan lalu lintas dan jalan (Undang-Undang Republik Indonesia, 2009). Akan tetapi, ketersediaan jalan raya yang masih sangat terbatas yaitu hanya 1,7

km per 1000 penduduk. Kementerian Pekerjaan Umum dan Perumahan Rakyat mencanangkan pemeliharaan dan pembuatan jalan sepanjang 46.812 km dan 495.698 meter untuk jembatan (Direktorat Riset dan Pengabdian Masyarakat UI, 2017).

Guna memenuhi kebutuhan tersebut, maka diperlukan bahan baku jalan dan jembatan, diantaranya adalah beton. Beton banyak dipilih karena kekuatan beton sangat tinggi, murah serta sangat mudah dicetak, dibentuk dan dipindahkan (Tjokrodimulyo, 2017). Akan tetapi beton tetap memiliki kekurangan diantaranya, kualitas dan kuat tekannya sangat bergantung pada proporsi pencampuran bahan-bahan pembentuk beton dengan bahan agregatnya (Mulyono, 2015).

Beton adalah bahan yang diperoleh dengan mencampurkan semen hidrolik, agregat kasar, agregat halus, air dan bahan tambahan (Mulyono, 2015). Kekuatan beton diukur dengan besarnya beban per satuan luas, yang menyebabkan beton hancur bila dibebani gaya tekan tertentu atau biasa disebut dengan kuat tekan beton (Tjokrodimulyo, 2017). Kuat tekan beton sangat bergantung pada karakteristik dan komposisi bahan-bahan pembentuk beton (Setiyorini & Wahono, Penerapan Metode Bagging untuk Mengurangi Data Noise pada Neural Network untuk Estimasi Kuat Tekan Beton, 2015).

Karakteristik bahan-bahan pembentuk beton yang harus disesuaikan dengan perilaku struktur yang akan dibuat. Dalam ilmu sipil, memprediksi sifat mekanik bahan konstruksi adalah tugas penelitian yang penting (Chou & Pham, 2013). Sifat dan karakteristik bahan pembentuk beton akan mempengaruhi mutu beton (Mulyono, 2015).

Mutu beton yang baik adalah jika beton tersebut memiliki kuat tekan tinggi (antara 20–50 Mpa, pada umur 28 hari). Dengan kata lain dapat diasumsikan bahwa mutu beton ditinjau dari kuat tekannya saja (Tjokrodimulyo, 2017). Kuat tekan beton akan berubah sesuai dengan bertambahnya umur beton tersebut dengan umur 3, 7, 14, 21, 28, 90, dan 365 hari. Usia beton paling optimal untuk mulai digunakan adalah beton yang berumur 28 hari. Secara umum, para ahli laboratorium melakukan mix design dengan aturan standar tertentu yang dilakukan secara manual dengan melihat tabel dan grafik referensi dan keadaan lapangan, tetapi cara tersebut sangat tidak efisien dan tidak menjamin akurasi. Untuk menjamin tingkat akurasi dalam mengestimasi kuat tekan beton sampai saat ini telah banyak penelitian yang dilakukan dengan berbagai macam metode komputasi dengan berbagai jenis dataset kuat tekan beton, dalam cabang ilmu komputer yang disebut data mining.

2. Metode Penelitian

Menurut (Dawson, 2009) ada empat metode penelitian yang paling umum digunakan yaitu: action research, experiment, case study, dan survey. Pada penelitian ini menggunakan metode experiment, yaitu penelitian yang melibatkan penyelidikan kepada beberapa variable menggunakan tes tertentu yang

dikendalikan sendiri oleh peneliti. Pada penelitian ini dilakukan beberapa tahapan penelitian sebagai berikut:

1. Pengumpulan Data (Data Gathering)
2. Pengolahan Data Awal (Data Pre-processing)
3. Metode yang Diusulkan (Proposed Method)
4. Eksperimen dan Pengujian Model (Model Test and Experiment)
5. Evaluasi dan Validasi Hasil (Result Evaluation and Validation)

3.1. Pengumpulan Data (Data Gathering)

Data yang digunakan pada penelitian ini merupakan data sekunder. Data sekunder adalah data yang tidak diperoleh langsung dari obyek penelitian, melainkan telah dikumpulkan oleh pihak lain. Data sekunder yang digunakan pada penelitian ini merupakan kumpulan data dari University of California Irvine Machine Learning Repository, data tersebut merupakan data 1030 eksperimen komposisi beton dengan berbagai macam hasil kuat tekan beton. Dataset ini diambil dari UCI Machine Learning Repository yang diunduh melalui <https://archive.ics.uci.edu/ml/datasets/Concrete+Compressive+Strength>.

Dataset ini berisi informasi setiap percobaan komposisi beton yang diwakili dalam data yang ditetapkan oleh 8 atribut yang merupakan komposisi pembentuk beton yang seluruhnya merupakan atribut numerik dengan 1 atribut kelas/output yang juga merupakan atribut numerik yang mewakili kuat tekan beton dalam satuan MPa (mega pascals).

Atribut	Deskripsi	Tipe Data
Cement	Jumlah semen dalam satuan kg/m ³	Numerik
Blast Furnace Slag	Jumlah bahan pembakaran dalam satuan kg/m ³	Numerik
Fly Ash	Jumlah abu terbang dalam satuan kg/m ³	Numerik
Water	Jumlah air dalam satuan kg/m ³	Numerik
Superplasticizer	Jumlah superplasticizer dalam satuan kg/m ³	Numerik
Coarse Aggregate	Jumlah penggunaan agregat kasar dalam satuan kg/m ³	Numerik
Fine Aggregate	Jumlah penggunaan agregat halus dalam satuan kg/m ³	Numerik
Age	Usia beton dalam satuan hari	Numerik
Concrete Compressive Strength	Kuat tekan beton dalam satuan MPa (<i>megapascals</i>)	Numerik

Tabel 3.1. Atribut Dataset

Atribut	Minimum	Maksimum	Rata-rata	Standar Deviasi
Cement	102	540	265,443	104,671
Blast Furnace Slag	0	359,4	86,285	87,8264
Fly Ash	0	200,1	62,7947	66,2275
Water	121,75	247	183,059	19,3294
Superplasticizer	0	32,2	6,9946	5,39208
Coarse Aggregate	801	1145	956,059	83,8015
Fine Aggregate	594	992,6	764,377	73,1201
Age	1	365	45,6621359	63,1699116
Concrete Compressive Strength	2,33180783	82,5992248	35,8178358	16,7056792

Tabel 3.2. Statistik Dataset

3.2. Pengolahan Data Awal (Data Pre-processing)

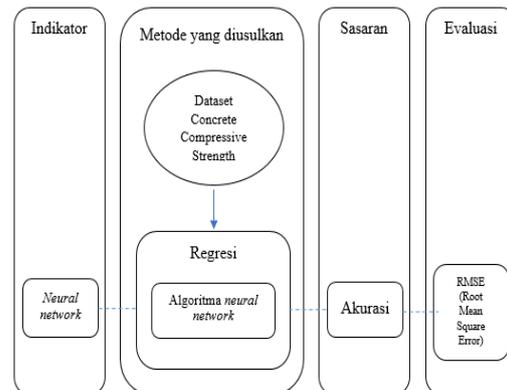
Pengolahan data awal merupakan tindak lanjut dari pengumpulan data. Pada penelitian ini, pengolahan data awal dengan mengelompokkan dataset berdasarkan atribut *age* atau usia beton. Data yang dipilih untuk diolah pada penelitian ini hanya data komposisi beton yang berusia 28 hari. Oleh karena itu pada pengolahan data awal ini, akan disaring dari 1030 data menjadi 425 data yang memiliki atribut *age* 28 hari.

Setelah dilakukan filter data berdasarkan usia beton 28 hari, selanjutnya akan dilakukan normalisasi data menggunakan *Z-transform* dengan tujuan penyekalaan data ke interval yang lebih kecil dan seragam. Setelah tahapan normalisasi data, dilakukan pembagian dataset menjadi *data training* dan *data testing*.

Untuk menguji model yang dikembangkan, data akan dibagi menjadi dua bagian, yaitu *data training* dan *data testing*. *Data training* digunakan untuk pengembangan model, sedangkan *data testing* digunakan untuk pengujian model. Pembagian dataset ini menggunakan *split validation* dengan rasio bervariasi mulai 90:10, 80:20, 70:30, 60:40 dan 50:50. Serta menggunakan *cross validation* untuk dapat mengambil model algoritma *neural network* yang nantinya akan diimplementasikan pada data baru. Ini dilakukan dengan tujuan agar dapat mengetahui rata-rata performa yang dihasilkan dari algoritma *neural network*.

3.3. Metode yang Diusulkan (Proposed Method)

Metode yang diusulkan yaitu menerapkan algoritma neural network sebagai algoritma classifier. Hasil klasifikasi kemudian di evaluasi akurasi menggunakan root mean square error (RMSE). Gambaran metode yang diusulkan dapat dilihat pada gambar 3.1.



Gambar 3.1. Metode yang diusulkan

Pengolahan data awal membagi data menjadi data training dan data testing menggunakan *split validation* dan *cross validation*. Setelah itu data training akan dilatih dengan algoritma neural network. Setelah didapatkan model. Model tersebut akan diterapkan untuk menguji model terhadap data testing.

Setelah menguji model pada data testing, dilakukan evaluasi terhadap model dengan cara menghitung akurasi yang dihasilkan dengan root mean square error (RMSE). Proses ini akan berulang seterusnya sebanyak lima kali dengan rasio data training dan data testing yang bervariasi. Setelah itu dihitung rata-rata hasil RMSE pada setiap rasio data training dan testing untuk mengetahui performa dari algoritma neural network dalam mengklasifikasi terapan yang paling optimal untuk estimasi kuat tekan beton.

3.4. Eksperimen dan Pengujian Model (Model Test and Experiment)

Penelitian yang dilakukan dalam eksperimen ini menggunakan komputer untuk melakukan proses perhitungan terhadap model yang diusulkan. Tahapan eksperimen pada penelitian ini adalah:

1. Menyiapkan 2 dataset untuk eksperimen, dataset training dan dataset testing.
2. Mendesain arsitektur neural network.

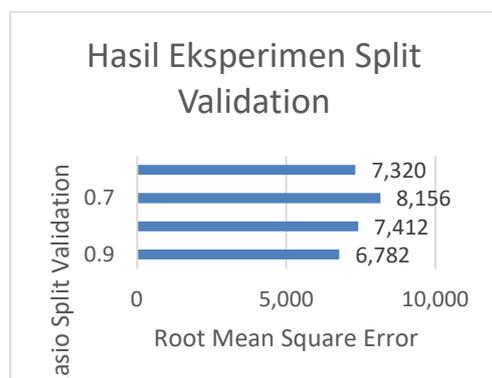
3. Melakukan training dan testing terhadap model neural network dan mencatat hasil RMSE nya.
4. Melakukan komparasi hasil akurasi pada kedua lima kali eksperimen bergantung pada rasio.
5. Mengembangkan aplikasi berdasarkan algoritma neural network.

3.5. Evaluasi dan Validasi Hasil (Result Evaluation and Validation)

Pada tahap ini akan dilakukan proses pengujian model yang dihasilkan oleh tool Rapidminer dengan mengevaluasi perbandingan hasil akurasi seluruh eksperimen yaitu eksperimen neural network pada dataset dengan lima rasio yang berbeda menggunakan split validation dan cross validation. Sementara itu, evaluasi yang digunakan adalah root mean square error.

3. Hasil dan Pembahasan

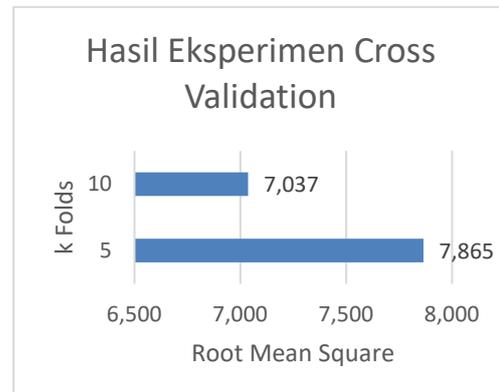
Hasil dalam penelitian dilakukan dalam dua eksperimen yaitu eksperimen estimasi kuat tekan beton menggunakan algoritma neural network dengan split validation dan eksperimen kedua estimasi kuat tekan beton menggunakan algoritma neural network dengan cross validation. Hasil akurasi paling buruk diperoleh pada rasio 70% data training dan 30% data testing dengan root mean square error sebesar 8.156. Sementara rata-rata RMSE yang dihasilkan pada eksperimen ini sebesar 7.417. Perbandingan akurasi pada eksperimen ini dapat dilihat pada Gambar 4.1.



Gambar 4.1. Grafik Perbandingan Hasil Eksperimen Split Validation

Hasil akurasi paling buruk diperoleh k folds 5 dengan root mean square error sebesar 7.865. Perbandingan akurasi pada

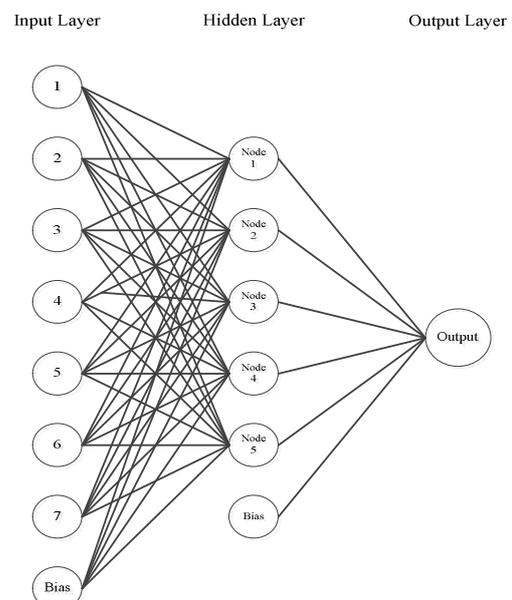
eksperimen ini dapat dilihat pada Gambar 4.2.



Gambar 4.2. Grafik Perbandingan Hasil Eksperimen Cross Validation

Model Algoritma Neural Network

Berdasarkan hasil eksperimen, dapat ditarik kesimpulan nilai RMSE terendah diperoleh pada hasil training menggunakan cross validation dengan k folds 10. Dengan RMSE yang dihasilkan sebesar 7.037, sementara k folds 5 menghasilkan RMSE yang lebih besar yakni 7.865 serta rata-rata yang dihasilkan pada split validation sebesar 7.417. Sehingga model algoritma yang digunakan pada penelitian ini adalah hasil dari training menggunakan cross validation.



Gambar 4.3. Model Algoritma Neural Network

Model algoritma tersebut dihasilkan oleh RapidMiner yang terdiri dari input layer,

hidden layer dan output layer. Input layer menunjukkan jumlah atribut yang mendeskripsikan bahan baku pembentuk beton, angka 1 hingga 7 merepresentasikan atribut cement, slag, fly ash, water, superplasticizer, coarse aggregate, fine aggregate.

Model algoritma menghasilkan 5 simpul pada hidden layer yang diberi nama simpul 1 hingga 5 serta menghasilkan 1 layer output. Model tersebut digunakan untuk menghitung perkiraan kuat tekan beton. Setiap node pada hidden layer memiliki bobot tertentu terhadap masing-masing input layer. Bobot input layer terhadap hidden layer dapat dilihat pada tabel 4.3.

Input Layer	Bobot				
	Node 1	Node 2	Node 3	Node 4	Node 5
Cement	1,816	-2,698	-0,645	2,728	-2,566
Slag	1,853	-0,769	0,548	0,188	-2,696
Fly ash	0,729	-0,007	-0,549	-2,074	-1,759
Water	-1,162	2,103	3,059	-1,504	-1,764
SP	0,848	-3,460	-1,304	3,301	2,671
Coarse Aggr.	-0,872	0,386	-0,170	2,152	-1,946
Fine Aggr.	0,531	-1,174	-0,489	0,846	-1,128
Bias	-1,589	-3,652	-3,928	-2,943	-3,036

Tabel 4.3. Tabel Bobot Input Layer terhadap Hidden Layer

Hidden layer pun memiliki bobot tertentu terhadap masing-masing output layer, dimana output layer yang dihasilkan pada model algoritma penelitian hanya satu. Bobot hidden layer terhadap output layer dapat dilihat pada Tabel 4.4.

1. Normalisasi Data

Melakukan normalisasi terhadap data yang akan dihitung menggunakan Z-transform. Normalisasi merupakan proses penyekalaan data ke interval yang lebih kecil sehingga mempermudah perhitungan. Z-transform dapat dilakukan dengan rumus sebagai berikut:

$$f(y) = \frac{y - \text{mean}}{\text{standard deviation}}$$

Dimana:

Node	<u>Bobot</u>
Node 1	1,391
Node 2	-1,160
Node 3	0,788
Node 4	0,617
Node 5	-1,474
<u>Threshold</u>	0,508

Tabel 4.4. Tabel Bobot Hidden Layer terhadap Input Layer

Untuk menguji model algoritma dapat melakukan estimasi kuat tekan beton, maka diujicobakan sample data uji menggunakan model algoritma neural network yang telah dihasilkan oleh RapidMiner. Pengujian model algoritma menggunakan sampel data sebagai berikut:

No	Cement	Slag	Fly ash	Water	SP	Coarse Aggr.	Fine Aggr.
Mean	265,4433882	86,285	62,7947	183,06	6,995	956,06	764,3766353
Stddev	104,6705266	87,826	66,2275	19,329	5,392	83,802	73,12006037
Data	168,9	42,2	124,3	158,3	10,8	1080,8	796,2
Hasil	31,12						

Tabel 4.5. Sampel Data Pengujian

Tahapan melakukan estimasi menggunakan model algoritma neural network pada sampel data tersebut yaitu:

y = nilai atribut

mean = nilai rata-rata dari atribut pada keseluruhan dataset

$\text{standard deviation}$ = nilai standar deviasi dari atribut pada keseluruhan dataset

$$f(y_1) = \frac{168,9 - 265,44}{104,7} = -0,923$$

(Cement)

$$f(y_2) = \frac{42,2 - 86,285}{87,83} = -0,502$$

(Slag)

$$f(y_3) = \frac{124,3-62,795}{66,23} = \mathbf{0,928}$$

(Fly Ash)

$$f(y_4) = \frac{158,3-183,06}{19,33} = \mathbf{-1,279}$$

(Water)

$$f(y_5) = \frac{10,8-6,994}{5,392} = \mathbf{0,711}$$

(Superplastic)

$$f(y_6) = \frac{1080,8-956,06}{83,8} = \mathbf{1,489}$$

(Coarse Aggr.)

$$f(y_7) = \frac{796,2-764,38}{73,12} = \mathbf{0,435}$$

(Fine Aggr.)

2. Menghitung setiap simpul pada hidden layer

Setiap simpul yang dihasilkan pada *hidden layer* kemudian dihitung menggunakan rumus sebagai berikut:

$$Node = \sum (weight \cdot f) + bias$$

Dimana:

weight = bobot dari setiap input layer terhadap hidden layer nya

f = nilai atribut yang telah di normalisasi

bias = nilai bias pada masing-masing node

Bobot pada rumus diatas merupakan bobot yang dihasilkan oleh RapidMiner bersamaan dengan model algoritmanya. Berikut ini merupakan tabel bobot untuk setiap *input layer* terhadap setiap simpul *hidden layer*.

Input Layer	Bobot				
	Node 1	Node 2	Node 3	Node 4	Node 5
Cement	1,816	-2,698	-0,645	2,728	-2,566
Slag	1,853	-0,769	0,548	0,188	-2,696
Fly ash	0,729	-0,007	-0,549	-2,074	-1,759
Water	-1,162	2,103	3,059	-1,504	-1,764
SP	0,848	-3,460	-1,304	3,301	2,671
Coarse Aggr.	-0,872	0,386	-0,170	2,152	-1,946
Fine Aggr.	0,531	-1,174	-0,489	0,846	-1,128
Bias	-1,589	-3,652	-3,928	-2,943	-3,036

Tabel 4.6. Tabel Bobot Input Layer terhadap Hidden Layer

$$\begin{aligned} Node1 &= (-0,923 \cdot 1,816) \\ &+ (-0,502 \cdot 1,853) \\ &+ (0,928 \cdot 0,729) \\ &+ (-1,279 \cdot -1,162) \\ &+ (0,711 \cdot 0,848) \\ &+ (1,489 \cdot -0,872) \\ &+ (0,435 \cdot 0,531) \\ &+ (-1,589) = \mathbf{-2,495} \end{aligned}$$

$$\begin{aligned} Node2 &= (-0,923 \cdot -2,698) \\ &+ (-0,502 \cdot -0,769) \\ &+ (0,928 \cdot -0,007) \\ &+ (-1,279 \cdot 2,103) \\ &+ (0,711 \cdot -3,46) \\ &+ (1,489 \cdot 0,386) \\ &+ (0,435 \cdot -1,174) \\ &+ (-3,652) = \mathbf{-5,871} \end{aligned}$$

$$\begin{aligned} Node3 &= (-0,923 \cdot -0,645) \\ &+ (-0,502 \cdot 0,548) \\ &+ (0,928 \cdot -0,549) \\ &+ (-1,279 \cdot 3,059) \\ &+ (0,711 \cdot -1,304) \\ &+ (1,489 \cdot -0,17) \\ &+ (0,435 \cdot -0,489) \\ &+ (-3,928) = \mathbf{-9,424} \end{aligned}$$

$$\begin{aligned} Node4 &= (-0,923 \cdot 2,728) \\ &+ (-0,502 \cdot 0,188) \\ &+ (0,928 \cdot -2,074) \\ &+ (-1,279 \cdot -1,504) \\ &+ (0,711 \cdot 3,301) \\ &+ (1,489 \cdot 2,152) \\ &+ (0,435 \cdot 0,846) \\ &+ (-2,943) = \mathbf{0,364} \end{aligned}$$

$$\begin{aligned} Node5 &= (-0,923 \cdot -2,566) \\ &+ (-0,502 \cdot -2,696) \\ &+ (0,928 \cdot -1,759) \\ &+ (-1,279 \cdot -1,764) \\ &+ (0,711 \cdot 2,671) \\ &+ (1,489 \cdot -1,946) \\ &+ (0,435 \cdot -1,128) \\ &+ (-3,036) = \mathbf{-0,179} \end{aligned}$$

3. Aktifkan fungsi sigmoid pada setiap simpul yang ada dalam hidden layer

Fungsi *sigmoid* merupakan operasi yang ditambahkan pada akhir perhitungan neural network, baik pada *hidden layer* maupun pada *output layer*. Mengaktifkan fungsi *sigmoid* dapat dilakukan dengan rumus sebagai berikut:

$$\begin{aligned} Sigmoid(node) \\ &= \frac{1}{1 + exponential^{-(node)}} \end{aligned}$$

Dimana:

sigmoid = nilai sigmoid pada setiap node

node = hasil perhitungan pada setiap simpul

$$\begin{aligned} \text{Sigmoid}(\text{node1}) &= \frac{1}{1 + e^{-(-2,495)}} \\ &= \mathbf{0,076} \end{aligned}$$

$$\begin{aligned} \text{Sigmoid}(\text{node2}) &= \frac{1}{1 + e^{-(-5,871)}} \\ &= \mathbf{0,003} \end{aligned}$$

$$\begin{aligned} \text{Sigmoid}(\text{node3}) &= \frac{1}{1 + e^{-(-9,424)}} \\ &= \mathbf{0,000} \end{aligned}$$

$$\begin{aligned} \text{Sigmoid}(\text{node4}) &= \frac{1}{1 + e^{-(-0,364)}} \\ &= \mathbf{0,590} \end{aligned}$$

$$\begin{aligned} \text{Sigmoid}(\text{node5}) &= \frac{1}{1 + e^{-(-0,179)}} \\ &= \mathbf{0,455} \end{aligned}$$

4. Menghitung simpul pada output layer

Setelah dilakukan aktivasi fungsi sigmoid pada setiap simpul yang ada pada hidden layer, selanjutnya adalah menghitung simpul output layer berdasarkan setiap simpul yang ada pada hidden layer ditambahkan dengan nilai threshold yang didapatkan dari model algoritma neural network. Menghitung simpul output layer dapat menggunakan rumus sebagai berikut:

$$\begin{aligned} \text{Output} &= \sum (\text{weight} \cdot S(\text{node})) \\ &+ \text{bias} \end{aligned}$$

Dimana:

weight = bobot hidden layer terhadap output layer

S(node) = nilai *sigmoid* untuk setiap node

bias = nilai threshold dari model algoritma

Bobot pada rumus diatas merupakan bobot yang dihasilkan oleh RapidMiner bersamaan dengan model algoritmanya. Berikut ini merupakan tabel bobot untuk setiap hidden layer terhadap setiap simpul output layer:

Node	<u>Bobot</u>
Node 1	1,391
Node 2	-1,160
Node 3	0,788
Node 4	0,617
Node 5	-1,474
<u>Threshold</u>	0,508

Tabel 4.7. Tabel Bobot Hidden Layer terhadap Input Layer

$$\begin{aligned} \text{Output} &= (0,076 \cdot 1,391) + (0,003 \cdot -1,16) \\ &+ (0,000 \cdot 0,788) \\ &+ (0,590 \cdot 0,617) \\ &+ (0,455 \cdot -1,474) \\ &+ (0,508) = \mathbf{0,304} \end{aligned}$$

5. Aktifkan fungsi linear pada simpul output layer

Fungsi linear sama seperti fungsi sigmoid yang merupakan operasi yang ditambahkan pada akhir perhitungan neural network, akan tetapi hanya dapat diimplementasikan pada output layer neural network untuk perhitungan numerik. Mengaktifkan fungsi linear dapat dilakukan dengan rumus sebagai berikut:

$$\text{Linear}(\text{Output}) = \text{Output}$$

$$\text{Linear}(\text{Output}) = \mathbf{0,304}$$

6. Denormalisasi output untuk mendapatkan nilai estimasi

Setelah mengaktifkan fungsi linear output layer perlu dilakukan denormalisasi dikarenakan sebelumnya data asli mengalami proses normalisasi sehingga datanya diskalakan. Oleh karena itu nilai fungsi linear output layer harus dikembalikan melalui proses denormalisasi menggunakan Z-transform menggunakan rumus sebagai berikut:

$$\begin{aligned} f'(\text{output}) &= \text{standard deviation} \cdot \\ &\text{Linear}(\text{output}) + \text{mean} \end{aligned}$$

$$\begin{aligned} f'(\text{output}) &= 14,71 \cdot 0,304 + 36,748 \\ &= \mathbf{41,214} \end{aligned}$$

4. Kesimpulan

Berdasarkan hasil pengujian menggunakan model algoritma neural network didapatkan hasil estimasi kuat tekan beton pada data sampel sebesar 41,214. Jika dibandingkan dengan data real yang didapatkan pada dataset sebesar 31,12. Sehingga dapat disimpulkan gap error atau perbedaan hasil estimasi dengan hasil sesungguhnya sebesar 10,094. Jika dibandingkan dengan hasil eksperimen menggunakan cross validation didapatkan nilai RMSE sebesar 7,037 maka dapat disimpulkan bahwa perbedaan hasil estimasi dengan hasil sesungguhnya tidak terpaut jauh dari RMSE yang sudah dihitung oleh RapidMiner.

Referensi :

- Direktorat Riset dan Pengabdian Masyarakat UI. (2017). *Laporan Pertanggung Jawaban Kegiatan Seminar Infrastruktur 2017*. Depok: Direktorat Riset dan Pengabdian Masyarakat Universitas Indonesia.
- Undang-Undang Republik Indonesia. (2009). *Undang-Undang Nomor 22 Tahun 2009 Tentang Lalu Lintas dan Jalan*. Jakarta: Undang-Undang Republik Indonesia.
- Tjokrodimulyo, K. (2017). *Teknologi Beton*. Yogyakarta: KMTS FT UGM.
- Mulyono, T. (2015). *Teknologi Beton*. Yogyakarta: Andi Offset.
- Setiyorini, T., & Wahono, R. S. (2015). Penerapan Metode Bagging untuk Mengurangi Data Noise pada Neural Network untuk Estimasi Kuat Tekan Beton. *Journal of Intelligent Systems*, 37-42.
- Chou, J., & Pham, A. (2013). Enhanced Artificial Intelligence for Ensemble Approach to predicting High Performance Concrete Compressive Strength. *Construction and Building Materials*, 554-563.
- Chen, H., Zhang, J., Xu, Y., Chen, B., & Zhang, K. (2013). Performance Comparison of Artificial Neural Network and Logistic Regression Model for Differentiating lung nodules on CT Scans. *Expert System with Application*, 11503-11509.
- Nazari, A., & Pacheco, F. T. (2013). Predicting Compressive Strength of Different Geopolymers by Artificial Neural Network. *Ceramics International*, 2247-2257.
- Setiyorini, T., & Asmono, R. T. (2018). Komparasi Metode Neural Network, Support Vector Machine dan Linear Regression pada Estimasi Kuat Tekan Beton. *Jurnal Techno Nusa Mandiri*, 51-56.