Laporan Tugas 2 TF4063

Penggunaan Multi Layer Perceptron pada Klasifikasi Segitiga dan Sinyal Respon Step Sistem Orde Dua

Akhmad Aji Permadi
10118030

Program Studi Teknik Fisika
Fakultas Teknologi Industri
Institut Teknologi Bandung
2022

Daftar Isi

1 Pendahuluan

Belakangan ini, penggunaan machine learning mulai populer dalam proses pengambilan keputusan. Salah satu pemanfaatan machine learning adalah dalam proses klasifikasi. Klasifikasi adalah proses yang berkaitan dengan kategorisasi, di mana objek-objek dapat dibedakan dan dipahami.

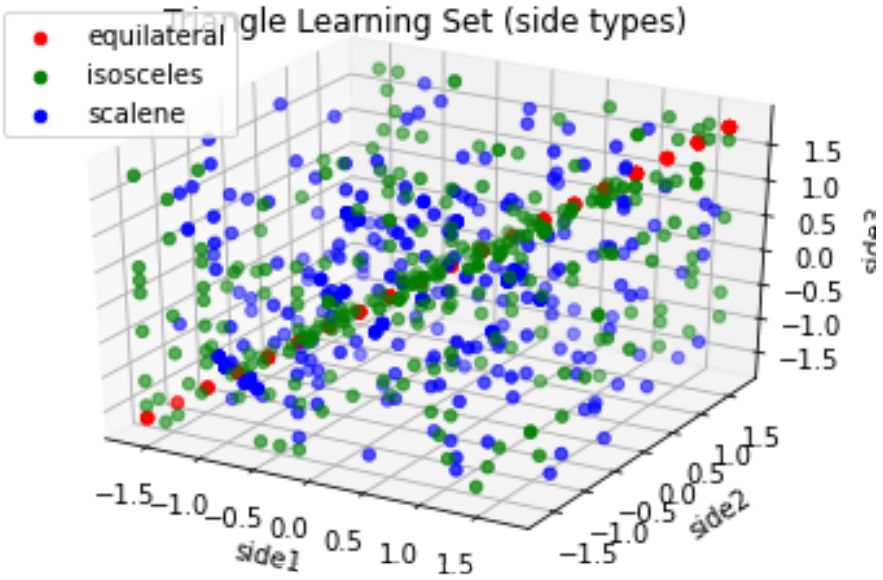
Scikit-learn memiliki MLPClassifier yang dapat dengan mudah membuat sebuah model untuk dapat menentukan klasifikasi sebuah objek. MLP sendiri merupakan singkatan dari Multi Layer Perceptron, merupakan jaringan saraf tiruan feedforward yang menghasilkan beberapa output dari input-input yang diberikan. Pada laporan ini, akan dibahas proses klasifikasi segitiga dan klasifikasi sinyal respon step sistem orde dua menggunakan MLPClassifier.

2 Klasifikasi Segitiga

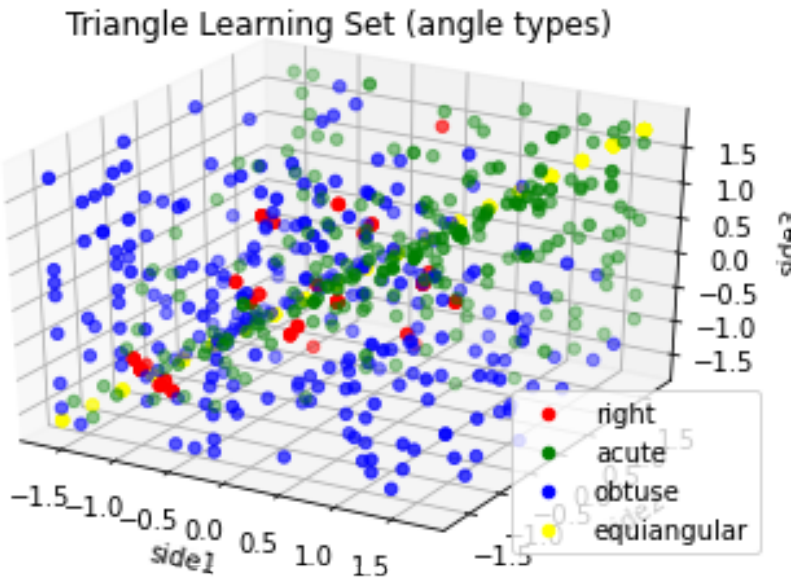
Sebuah segitiga dapat diklasifikasikan menjadi 3 jenis berdasarkan panjang sisinya dan menjadi 4 jenis berdasarkan besar sudut-sudutnya.Adapun untuk tiga jenis segitiga berdasarkan panjangnya meliputi: Equilateral Triangle (3 equal sides), Isosceles Triangle (2 equal sides), dan Scalene Triangle (no equal sides). Sedangkan untuk empat jenis segitiga berdasarkan besar sudutnya adalah: Right Triangle (1 angle = 90°), Acute Triangle (3 angles < 90°), Obtuse Triangle (1 angle > 90°), dan Equiangular Triangle (3 angles = 60°).

2.1 Pembuatan Dataset

Langkah pertama yang dilakukan sebelum membuat model yang dapat mengklasifikasikan jenis segitiga adalah mempersiapkan dataset. Pada bagian ini, dataset yang digunakan adalah dataset yang dibuat dengan menggunakan kode milik Pak Mursito dengan sedikit modifikasi. Data yang dibuat berjumlah 1000 segitiga dengan informasi panjang ketiga sisi sebagai X dan label $[t_{side}, t_{angle}]$ sebagai y yang merupakan jenis segitiga berdasarkan sisi dan sudutnya. Kemudian dataset akan dibagi dengan proporsi $0.75 : 0.25$ untuk keperluan training dan test dataset. Bagian sisi-sisi (fitur X) sebelum masuk ke tahap proses training dilakukan standardisasi. Pada gambar 1 dan 2 ditunjukkan persebaran tiap jenis segitiga berdasarkan sisi dan sudutnya.



Gambar 1: Triangle learning set berdasarkan jenis sisinya



Gambar 2: Triangle learning set berdasarkan jenis sudutnya

2.2 Training Model

Pada bagian ini, output yang diharapkan dari model yang akan dibuat adalah model dapat memberikan hasil klasifikasi sebuah segitiga berdasarkan jenis sisi dan sudutnya. Dengan menggunakan **MLPClassifier** dari library **Scikit-learn**, hal ini tidak cukup sulit ketika akan dilakukan klasifikasi sebuah segitiga berdasarkan jenis sisi dan sudutnya secara terpisah. Namun, ketika akan dilakukan proses training langsung di mana output berjumlah dua sekaligus berupa $[t_{side}, t_{angle}]$, akan menyebabkan error seperti screenshot pada gambar 3. Hal ini menunjukkan bahwa MLPClassifier dari Scikit-Learn hanya menerima multi-output berupa data biner (0 atau 1), sedangkan data yang sedang digunakan sekarang termasuk ke dalam multi-class multi-output (kelas yang ada tidak dapat direpresentasikan hanya dengan 0 dan 1).

```
/usr/local/lib/python3.7/dist-packages/sklearn/preprocessing/_label.py in fit(self, y)
    293         if "multioutput" in self.y_type_:
    294             raise ValueError(
--> 295                 "Multioutput target data is not supported with label binarization"
    296             )
    297         if _num_samples(y) == 0:

ValueError: Multioutput target data is not supported with label binarization
```

Gambar 3: Error MLPClassifier untuk Multi-class Multi-Ouput

Setelah melakukan research dengan membaca beberapa dokumentasi di Scikit-learn dan artikel terkait, namun masih belum menemukan titik terang penggunaan MLPClassifier dalam multi-class multi-output, akhirnya Saya terpikirkan tentang dua alternatif untuk mengatasi permasalahan ini. Alternatif yang pertama adalah melakukan training secara terpisah, yang berarti akan ada dua model berupa klasifikasi segitiga berdasarkan sisi dan klasifikasi segitiga berdasarkan sudut. Kemudian untuk alternatif kedua adalah pembuatan label baru untuk merepresentasikan kombinasi dari $[t_{side}, t_{angle}]$ sehingga dapat

dilakukan training model menggunakan MLPClassifier hanya dengan 1 output. Sebagai contoh untuk alternatif dua, misalnya label 1 merepresentasikan $[Isosceles, Right]$, 2 merepresentasikan $[Isosceles, Acute]$, dan seterusnya. Untuk dapat menentukan alternatif mana yang lebih baik, berikutnya dibuat sebuah tabel 1 yang berisi kelebihan dan kelemahan dari masing-masing alternatif.

Alternatif	Pros	Cons
Alternatif 1	Tingkat akurasi mungkin lebih tinggi (*) Tidak memerlukan proses pelabelan baru	Ada dua model yang perlu diperhatikan Memerlukan biaya waktu dan tenaga lebih untuk proses maintenance dua model
Alternatif 2	Lebih efisien Bisa fokus hanya pada satu model	Membutuhkan data lebih banyak untuk dapat memperoleh akurasi lebih tinggi dari alternatif 1 (**) Memerlukan proses pelabelan baru

Tabel 1: Kelebihan dan Kelemahan masing-masing alternatif

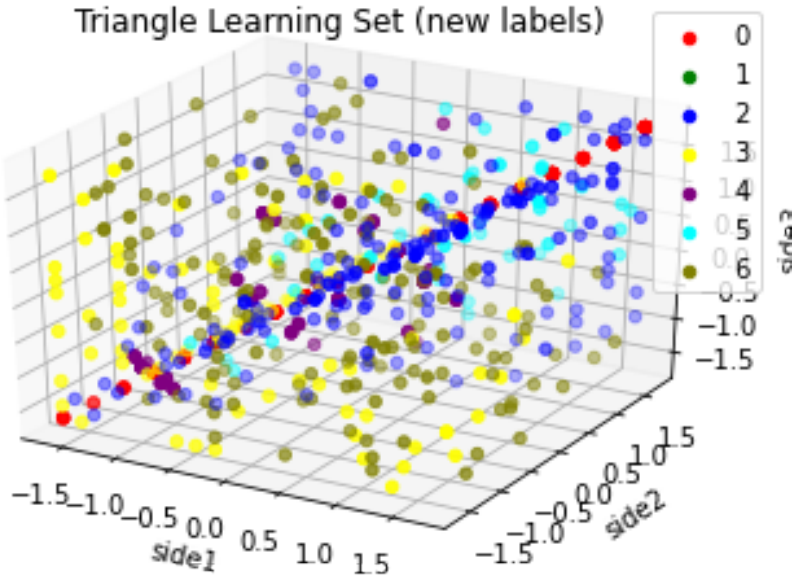
Pada alternatif 1 kemungkinan lebih cepat didapatkan akurasi yang baik (*) dikarenakan setiap model hanya difokuskan pada paling banyak ke 4 kelas (untuk klasifikasi segitiga berdasarkan sudutnya). Sedangkan untuk Alternatif 2, memerlukan data lebih dalam proses trainingnya (**) dikarenakan banyaknya label yang merepresentasikan setiap kombinasi $[t_{side}, t_{angle}]$ membuat banyaknya data setiap label cenderung lebih sedikit dibandingkan ketika menggunakan alternatif 1.

Setelah melihat beberapa pros-cons yang beberapa di antaranya masih praduga, maka diputuskan untuk menggunakan alternatif 2. Alasan mendasar dipilihnya alternatif 2 dibandingkan dengan alternatif 1 adalah jika dilihat dari sisi efisiensi, alternatif 2 cenderung lebih mudah dalam proses maintenance model tanpa harus membagi fokus kepada dua model yang berbeda. Oleh karena itu, selanjutnya dilakukan proses pelabelan baru untuk setiap kombinasi $[t_{side}, t_{angle}]$. Adapun pelabelan baru akan ditunjukkan pada tabel 2 berikut.

Label	Kombinasi $[t_{side}, t_{angle}]$
0	[Equilateral, Equiangular]
1	[Isosceles, Right]
2	[Isosceles, Acute]
3	[Isosceles, Obtuse]
4	[Scalene, Right]
5	[Scalene, Acute]
6	[Scalene, Obtuse]

Tabel 2: Pelabelan baru untuk kombinasi $[t_{side}, t_{angle}]$

Setelah melakukan proses pelabelan baru pada dataset training dan test, diperlihatkan persebaran kombinasi $[t_{side}, t_{angle}]$ pada gambar 4.



Gambar 4: Triangle learning set dengan label baru

Dataset dengan label baru dilakukan proses training dan mendapatkan hasil metric seperti pada gambar 5. Pada metric tersebut tidak terlihat adanya label 1, hal ini terjadi dikarenakan pada data training yang dimiliki (bahkan pada data test juga) tidak ditemukan data segitiga dengan label 1 (Isosceles, Right). Sebuah segitiga dikatakan sebagai Isosceles dan Right apabila memiliki satu sudut sebesar 90° dan kedua sisinya sama panjang. Hal tersebut hanya terjadi ketika sisi miring dari segitiga tersebut memiliki panjang $x\sqrt{2}$ untuk x adalah panjang kedua sisi lainnya. Berhubung data yang dibuat hanya berdasarkan bilangan bulat (integer), maka hal tersebut tidak terjadi. Maka, pada kasus ini, dapat diasumsikan bahwa label 1 yang merupakan tipe segitiga Isosceles dan Right tidak dapat ditemukan (tidak ada). Model yang dihasilkan pada proses training ini akan menjadi base model untuk model-model berikutnya yang sudah dilakukan hyperparameter tuning.

	precision	recall	f1-score	support
0	0.95	1.00	0.98	63
2	0.69	0.68	0.69	53
3	0.63	0.90	0.74	30
4	0.91	1.00	0.95	42
5	0.22	0.18	0.20	11
6	0.85	0.57	0.68	51
accuracy			0.80	250
macro avg	0.71	0.72	0.71	250
weighted avg	0.80	0.80	0.79	250

Gambar 5: Metric base model klasifikasi segitiga

2.3 Hyperparameter Tuning

Dalam proses modelling, perlu dilakukan proses hyperparameter tuning untuk dapat dilakukan komparasi model mana yang lebih baik. Adapun pada bagian ini, akan dilakukan proses tuning pada beberapa hyperparameter seperti: jumlah layer, activation, solver, dan learning rate. Berikut disajikan tabel 3 yang akan memperlihatkan nilai akurasi yang didapat dari masing-masing model dangan hyperparameter yang berbeda.

Model	hidden layer size	activation	solver	lr	acc
base model	(30,30,30)	logistic	lbfgs	adaptive	0.80
model 1	(30,60,30)	logistic	lbfgs	adaptive	0.86
model 2	(30,60,15)	logistic	lbfgs	adaptive	0.75
model 3	(30,30)	logistic	lbfgs	adaptive	0.92
model 4	(30,90)	logistic	lbfgs	adaptive	0.93
model 5	(60)	logistic	lbfgs	adaptive	0.95
model 6	(30,30,30)	tanh	lbfgs	adaptive	0.91
model 7	(30,30,30,30)	tanh	lbfgs	adaptive	0.94
model 8	(30,30,30,60)	tanh	lbfgs	adaptive	0.95
model 9	(30,30,30,30,30)	tanh	lbfgs	adaptive	0.97
model 10	(30,30,30,30,30)	tanh	sgd	adaptive	0.93
model 11	(30,30,30,30)	tanh	sgd	adaptive	0.97
model 12	(30,30,30,60)	tanh	sgd	adaptive	0.95
model 13	(30,30,60,90)	tanh	sgd	adaptive	0.96
model 14	(30,30,30)	tanh	sgd	adaptive	0.95
model 15	(30,30,30)	tanh	adam	adaptive	0.96
model 16	(30,30,30,30)	tanh	adam	adaptive	0.97
model 17	(30,30,30,30,30)	tanh	adam	adaptive	0.96
model 18	(30,30,30)	relu	adam	adaptive	0.97
model 19	(30,60,90,30)	relu	adam	adaptive	0.96
model 20	(30,30,30)	relu	adam	constant	0.97
model 21	(30,30,30,30)	relu	adam	constant	0.95
model 22	(30,30,30,30,30)	relu	adam	constant	0.93

model 23	(30,30,30)	relu	adam	invscaling	0.97
model 24	(30,30,60)	relu	lbfgs	invscaling	0.96
model 25	(30,30,30,60,90)	relu	lbfgs	invscaling	0.98

Tabel 3: Hyperparameter Tuning untuk model Triangle Classification

2.4 Pembahasan

Dari beberapa kali percobaan sebelumnya, terlihat bahwa model 25 memiliki akurasi paling baik, dengan detail nilai tiap parameternya seperti yang tertulis pada tabel 3. Perbedaan nilai parameter pada proses training mempengaruhi akurasi. Seperti yang terlihat pada model 2, model 3, model 4, dan model 5, dengan nilai-nilai activation, solver, dan lr yang sama (tertulis pada tabel), perbedaan jumlah ukuran hidden layer cukup mempengaruhi besar akurasi. Pada model 2 hingga model 5, semakin sedikit jumlah ukuran hidden layer, maka semakin besar nilai akurasinya. Namun hal ini tidak berlaku pada model 24 dan model 25, yang mana dengan activation, solver, dan lr yang sama (namun berbeda dengan model 2-5), memberikan hasil akurasi yang lebih baik untuk jumlah ukuran hidden layer yang lebih banyak.

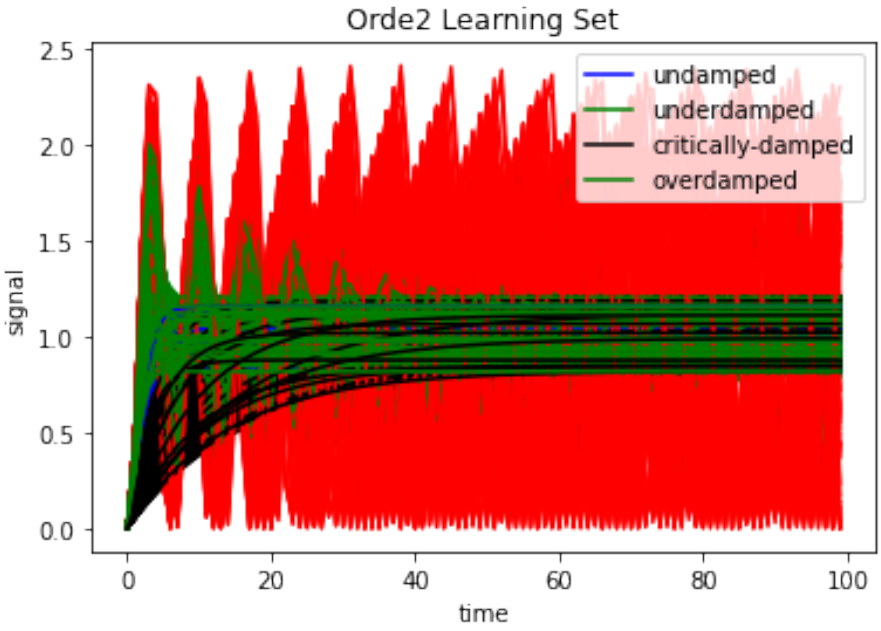
Jumlah ukuran hidden layer bukan satu-satunya yang mempengaruhi besar kecilnya akurasi, melainkan ada faktor lain yang ikut terlibat dalam menentukannya. Pada data yang digunakan, hyperparameter yang cukup baik adalah seperti yang tertulis pada model 25 dengan nilai akurasi sebesar 0.98.

3 Klasifikasi Sinyal Respon Step Sistem Orde 2

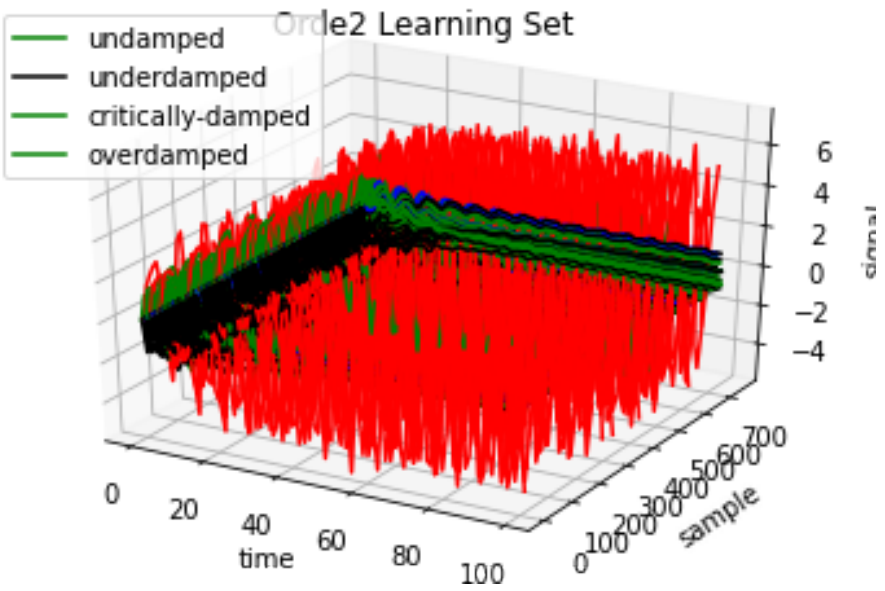
Sinyal respon step sistem orde 2 dapat diklasifikasikan menjadi 4 jenis, yang mana untuk setiap jenisnya memiliki karakteristik sinyal yang berbeda-beda. Adapaun keempat jenis tersebut meliputi: undamped, underdamped, critically-damped, overdamped.

3.1 Pembuatan Dataset

Dataset yang digunakan pada bagian ini dibuat menggunakan kode milik Pak Mursito. Dataset terdiri dari 1000 data sinyal beserta tipenya sebagai label. Data sinyal dibentuk dengan 100 step waktu. Data yang sudah dibuat kemudian dibagi menjadi dua data dengan proporsi 0.75 : 0.25 untuk keperluan training dan test dataset. Pada gambar 6 dan 7 menunjukkan plot untuk Orde 2 learning set dalam 2D dan 3D.



Gambar 6: Orde 2 learning set 2D



Gambar 7: Orde 2 learning set 3D

Data pada plot memang tidak terlihat terlalu jelas dikarenakan bentuk data yang panjang dan banyak data yang tidak sedikit. Untuk banyaknya data pada setiap kelas di data training dan data test akan disajikan pada tabel 4. Jumlah setiap kelasnya terlihat cukup timpang, di mana undamped dan critically-damped relatif cukup sedikit jika dibandingkan dua kelas lainnya, underdamped dan overdamped. Namun tidak masalah, untuk saat ini data yang ada akan tetap dilakukan proses training model klasifikasi sinyal respon.

Kelas	Data Training	Data Test
undamped	37	16
underdamped	330	105
critically-damped	76	33
overdamped	307	96

Tabel 4: Jumlah data di setiap kelas

3.2 Training Model

Pada bagian ini akan dilakukan proses training dataset yang sudah dibuat sebelumnya menggunakan **MLPClassifier** dari library **Scikit-learn**. Hyperparameter yang digunakan untuk base model ini tertulis pada tabel 5.

Hyperparameter	Value
hidden layer size	(30)
activation	logistic
solver	lbfgs
learning rate	adaptive
max iter	1000
momentum	0.9

Tabel 5: Hyperparameter base model sinyal respon

Proses training model dengan dataset yang sudah dibuat dan hyperparameter seperti pada tabel 5 memberikan nilai akurasi yang cukup tinggi, yaitu sebesar 0.96. Untuk hasil metric base model ini dapat dilihat pada gambar 8.

	precision	recall	f1-score	support
0	0.92	0.69	0.79	16
1	0.95	0.97	0.96	105
2	0.92	1.00	0.96	33
3	1.00	0.99	0.99	96
accuracy			0.96	250
macro avg	0.95	0.91	0.92	250
weighted avg	0.96	0.96	0.96	250

Gambar 8: Metric base model klasifikasi sinyal respon

3.3 Hyperparameter Tuning

Meskipun pada sub-section sebelumnya telah didapatkan base model dengan akurasi yang terbilang cukup tinggi, namun akan tetap dilakukan hyperparameter tuning untuk mendapatkan akurasi yang lebih tinggi. Adapun hyperparameter yang akan dilakukan tuning meliputi: jumlah layer, activation, solver, dan learning rate. Pada tabel 6 telah disajikan model beserta hyperparameternya dengan akurasi yang diperoleh.

Model	hidden layer size	activation	solver	lr	acc
base model	(30)	logistic	lbfgs	adaptive	0.96
model 1	(30, 30)	logistic	lbfgs	adaptive	0.97
model 2	(30, 30, 30)	logistic	lbfgs	adaptive	0.97
model 3	(30, 30, 30)	identity	lbfgs	adaptive	0.98
model 4	(30, 30, 30)	tanh	lbfgs	adaptive	0.98
model 5	(30, 30, 30)	relu	lbfgs	adaptive	0.97
model 6	(30, 30, 30)	relu	sgd	adaptive	0.96

model 7	(30, 30, 30)	relu	adam	adaptive	0.94
model 8	(30, 30, 30)	relu	adam	constant	0.94
model 9	(30, 30, 30)	relu	adam	invscaling	0.94
model 10	(30, 30)	relu	adam	invscaling	0.95
model 11	(30)	relu	adam	invscaling	0.97
model 12	(30)	tanh	adam	constant	0.95
model 13	(30, 30)	tanh	adam	constant	0.96
model 14	(30, 30, 30)	tanh	adam	constant	0.95
model 15	(30)	identity	adam	invscaling	0.98

Tabel 6: Hyperparameter Tuning untuk model Sinyal Respon Classification

3.4 Pembahasan

Hasil yang diperoleh pada base model relatif lebih bagus jika dibandingkan dengan model-model lain yang telah dilakukan hyperparameter tuning, meskipun masih di atas 0.90. Jika dilihat pada base model, model 1, dan model 2, dengan activation, solver, dan lr yang sama seperti yang tertulis pada tabel, jumlah ukuran hidden layer mempengaruhi nilai akurasi, di mana semakin banyak ukuran hidden layer, maka nilai akurasi pun akan semakin tinggi. Namun, dengan nilai activation, solver, dan lr sama pada model 9, model 10, dan model 11 (namun berbeda dari base model, model 1, dan model 2), semakin sedikit ukuran hidden layer justru membuat nilai akurasi semakin besar. Selain itu, dengan jumlah ukuran hidden layer, activation, dan layer yang sama (pada model 5, model 6, dan model 7), solver lbfgs mampu memberikan nilai akurasi yang lebih tinggi dibandingkan dua solver lainnya, sgd dan adam.

Dari beberapa analisis sebelumnya, dapat diketahui bahwa pemilihan kombinasi nilai hyperparameter sangat menentukan nilai akurasi yang didapatkan. Sehingga, tidak dapat disimpulkan secara pasti apakah menambahkan ukuran hidden layer akan menambah nilai akurasi atau sebaliknya. Parameter-parameter lain seperti activation, solver, dan lr juga turut berperan dalam membuat akurasi model menjadi baik atau bahkan semakin menurun.

4 Kesimpulan dan Saran

4.1 Kesimpulan

1. dataset yang berbeda memerlukan penyesuaian hyperparameter yang berbeda pula untuk memperoleh akurasi yang baik
2. menaikkan nilai akurasi dapat dilakukan dengan menambah/mengurangi jumlah ukuran hidden size, mengganti activation, solver, maupun learning rate yang digunakan, namun tidak ada ketentuan pasti keputusan mana yang harus dilakukan untuk dapat menaikkan nilai akurasi
3. Diperoleh akurasi paling tinggi pada model Klasifikasi Segitiga sebesar 0.98 dan akurasi paling tinggi pada model Klasifikasi Sinyal Respon Step Sistem Orde dua sebesar 0.98

4.2 Saran

Untuk mendapatkan akurasi yang lebih tinggi dari yang sudah diperoleh, dapat dilakukan penambahan data ataupun melakukan kombinasi nilai parameter yang belum dicoba

5 Daftar Pustaka

Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

6 Lampiran

Notebook .ipynb yang digunakan dalam pembuatan laporan ini dapat dilihat pada link github berikut: Triangle-and-Signal-Response-Classification