

Nama : Adiyasa Nurfalah

NIM : 33221004

Tugas : Ke-1

## 1. Masalah Klasifikasi

### 1.1. Deskripsi Umum Experiment

Tugas kali ini adalah membangun model CNN untuk menyelesaikan masalah klasifikasi citra. Data yang digunakan dipilih dari Fashion MNIST (penjelasan mengenai data bisa dilihat di project notebook).

Eksperimen dalam tugas ini bertujuan untuk menjawab pertanyaan-pertanyaan berikut ini:

- Berapa banyaknya convolution layer yang optimal?
- Berapa ukuran filter yang optimal untuk setiap convolution layer?
- Berapa banyaknya filter yang optimal untuk setiap convolution layer?
- Berapa banyaknya hidden unit yang optimal pada bagian fully connected network?

Selain itu juga untuk mengeksplorasi mengenai fungsi optimizer, learning rate schedule, dan fungsi loss pada CNN yang disediakan secara built-in di Keras.

Hyper parameter yang di observasi yaitu:

- a. Epoch
- b. Learning rate
- c. Jumlah hidden unit di lapisan fully connected
- d. Ukuran filter di semua lapisan konvolusional
- e. Jumlah lapisan konvolusional dan jumlah filter di tiap lapisan konvolusional
- f. Fungsi optimizer
- g. Pengaruh penggunaan learning rate schedule
- h. Fungsi loss

Metode yang dilakukan adalah diawal kami menggunakan contoh notebook project Digit Detection yang dikeluarkan oleh Intro to Deep Learning milik MIT sebagai template, kemudian kami mulai observasi parameter-parameter yang telah disebutkan diatas berurutan. Tiap parameter yang diobservasi akan menggunakan parameter lain yang paling optimal hasilnya dari eksperimen sebelumnya, kecuali jika parameter tersebut belum diobservasi maka akan menggunakan nilai default dari template project Digit Detection.

Setiap parameter akan dilakukan percobaan menggunakan beberapa nilai diantara nilai default dari project Digit Detection untuk dilihat performanya. Setiap nilai parameter akan dijalankan 5 kali percobaan (running training dan test) karena karakteristik Neural Network yang selalu menghasilkan performa berbeda setiap kali di-run. Dari 5 kali percobaan kemudian dihitung rata-rata performanya. Ukuran performa yang digunakan adalah :

- akurasi training (0-1, dimana 1 = terbaik)
- loss training (angka kecil = terbaik)
- akurasi testing (0-1, dimana 1 = terbaik)

Lingkungan eksperimen yang digunakan adalah Google Colab dengan menggunakan GPU. Untuk waktu training tidak dicatat karena perbedaannya sangat kecil antar eksperimen yang dilakukan. Rata-rata waktu training yang dilakukan untuk setiap epoch adalah 6-9s.

### 1.2. Mencari Epoch Terbaik

Parameter Fix	
Jml. Convolutional layer	2
Ukuran filter di conv. Layer	3
Jml. Filter tiap conv. layer	24, 36
Hidden unit	128
Learning rate	0.001

No.	Epoch	Rerata Training Loss	Rerata Akurasi Training	Rerata Akurasi Testing
1	5	0.01422	0.995	0.90608
2	10	0.01198	0.9962	0.90688
3	15	0.0375	0.9864	0.90768
4	20	0.0117	0.9962	0.9049

#### Analisis:

Penambahan epoch tidak terlalu berpengaruh signifikan terhadap akurasi dan loss, namun tetap berpengaruh terhadap waktu training yang menjadi lebih lama. Setiap epoch menghabiskan waktu training sekitar 6s. Dari table diatas terlihat bahwa akurasi training tertinggi dan loss training terendah didapat ketika epoch = 20 namun akurasi testing tertinggi didapat ketika epoch = 15. Sementara itu ketika epoch = 5 performa model pun tetap masih baik, dengan perbedaan performa yang sangat kecil dibandingkan dengan performa terbaiknya (epoch 15 dan 20), sehingga kami memilih nilai 5 yang dijadikan nilai epoch patokan (bisa dikatakan terbaik, jika memperhitungkan waktu training) untuk eksperimen selanjutnya.

### 1.3. Mencari Learning Rate Terbaik

Parameter Fix	
Jml. Convolutional layer	2
Ukuran filter di conv. Layer	3
Jml. Filter tiap conv. layer	24, 36
Hidden unit	128
Epoch	5

No.	Learning Rate	Rerata Training Loss	Rerata Akurasi Training	Rerata Akurasi Testing	Keterangan
1	0.1	2.31154	0.0995	0.1	
2	0.01	0.2241	0.9166	0.88124	

3	0.001	0.01422	0.995	0.90608	Dari eksperimen sebelumnya
4	0.0001	0.31418	0.8871	0.87462	

#### Analisis:

Kami menggunakan fungsi optimizer default template yaitu Adam, kami menetapkan beberapa nilai Learning Rate (LR) diantara nilai LR default dan didapatkan hasil bahwa LR = 0.1 paling buruk dan LR = 0.001 (default) adalah paling baik. Terlihat bahwa LR = 0.1 menghasilkan performa yang sangat buruk, penjelasan singkatnya adalah ketika LR terlalu besar maka proses training akan seperti bereksplorasi, proses pencarian optimum global (minimasi error) terlewat begitu saja karena langkah pencariannya terlalu besar sehingga error tetap besar, terjebak di optimum local yang paling dangkal (paling optimum local). Dari hasil eksperimen ini dapat disimpulkan bahwa parameter LR sangat berpengaruh terhadap performa.

#### 1.4. Mencari Jumlah Hidden Unit Terbaik

Parameter Fix	
Jml. Convolutional layer	2
Ukuran filter di conv. Layer	3
Jml. Filter tiap conv. layer	24, 36
Learning rate	0.001
Epoch	5

No.	Hidden Unit	Rerata Loss Training	Rerata Akurasi Training	Rerata Akurasi Testing	Keterangan
1	32	0.17724	0.9345	0.90212	
2	64	0.15274	0.9437	0.90436	
3	128	0.01422	0.995	0.90608	Dari eksperimen sebelumnya
4	256	0.10506	0.9606	0.90654	
5	512	0.0907	0.9662	0.90848	

#### Analisis:

Seperti sebelumnya, tidak ada patokan dalam tuning hyperparameter, dalam eksperimen ini kami menetapkan beberapa nilai Hidden Unit (HU) berdasarkan nilai yang dihasilkan dari 2<sup>^</sup>. Didapatkan kesimpulan bahwa parameter jumlah HU tidak terlalu berpengaruh signifikan terhadap performa namun tetap berpengaruh pada waktu training, karena dimensi matriks yang dihasilkan akan berbeda. Dari table diatas didapat bahwa jumlah HU yang optimal adalah = 128.

#### 1.5. Mencari Ukuran Filter Terbaik

Parameter Fix	
Jml. Convolutional layer	2
Learning rate	0.001

Jml. Filter tiap conv. layer	24, 36
Hidden unit	128
Epoch	5

No.	Filter Size	Rerata Loss Training	Rerata Akurasi Training	Rerata Akurasi Testing	Keterangan
1	3	0.01422	0.995	0.90608	Dari eksperimen sebelumnya
2	5	0.13442	0.9501	0.90562	
3	7	0.17378	0.9355	0.89498	

#### Analisis:

Pada eksperimen kali ini dilakukan pencarian ukuran filter terbaik untuk diterapkan di semua lapisan konvolusional. Nilai filter yang di observasi adalah 3x3, 5x5, dan 7x7. Didapatkan hasil bahwa ukuran filter terbaik adalah 3x3 dilihat dari rata-rata akurasi testingnya. Namun berdasarkan table diatas didapatkan bahwa ketiga ukuran filter menghasilkan akurasi yang hamper sama baiknya, ini menandakan ukuran filter tidak terlalu berpengaruh terhadap performa model.

#### 1.6. Mencari Jumlah Convolutional Layer dan Jumlah Filter Terbaik

<b>Parameter Fix</b>	
Ukuran filter di conv. Layer	3
Learning rate	0.001
Hidden unit	128
Epoch	5

No.	Jml. Conv. Layer	Rerata Loss Training	Rerata Akurasi Training	Rerata Akurasi Testing	Keterangan
1	2 (12, 24)	0.16042	0.9409	0.90432	
2	2 (24, 36)	0.01422	0.995	0.90608	Dari eksperimen sebelumnya
3	2 (36, 48)				
4	3 (24, 36, 48)	0.274	0.8991	0.87518	
5	2 (16, 32)	0.13936	0.9483	0.90448	
6	2 (32, 64)	0.13668	0.9495	0.9053	
7	2 (64, 128)	0.12262	0.9538	0.90436	
8	3 (32, 64, 128)	0.15252	0.9432	0.89312	

#### Analisis:

Pada eksperimen kali ini kami mencari jumlah layer konvolusional dan jumlah filter tiap lapisannya tersebut. Tidak ada patokan nilai yang digunakan namun berdasarkan hasil percobaan jumlah lapisan konvolusional yang memungkinkan hanya hingga 3, selebihnya muncul error. Didapatkan jumlah

lapisan konvolusional yang paling optimal adalah 2 dengan masing-masing jumlah filter tiap lapisannya adalah 24 dan 36 filter. Dari table diatas terlihat bahwa jumlah lapisan konvolusional 2 atau 3 dengan berbagai variasi jumlah filternya tidak terlalu signifikan pengaruhnya terhadap performa model yang dihasilkan.

### 1.7. Mencari Fungsi Optimizer Terbaik

Parameter Fix	
Ukuran filter di conv. Layer	3
Learning rate	Default optimizer
Hidden unit	128
Epoch	5
Jml. Conv. Layer	2
Jml. Filter tiap Conv. Layer	24, 36

No.	Optimizer	Rerata Loss Training	Rerata Akurasi Training	Rerata Akurasi Testing
1	SGD	0.4153	0.849	0.82515
2	RMSprop	0.1957	0.9285	0.90135
3	Adam	0.2016	0.9259	0.9057
4	Adadelta	1.86195	0.5554	0.5551
5	Adagrad	0.6559	0.7627	0.7592
6	Adamax	0.28155	0.8985	0.8838
7	Nadam	0.19555	0.9277	0.89205
8	Ftrl	2.3026	0.0992	0.1

#### Analisis:

Eksperimen kali ini bertujuan untuk mencari fungsi optimizer terbaik yang dimiliki oleh Keras dengan parameter optimizer default. Didapatkan hasil bahwa Adam adalah fungsi optimizer terbaik yang menghasilkan performa yang tertinggi.

### 1.8. Pengaruh Penggunaan Learning Rate Schedule

Parameter Fix	
Ukuran filter di conv. Layer	3
Optimizer	Adam
Hidden unit	128
Epoch	5
Jml. Conv. Layer	2
Jml. Filter tiap Conv. Layer	24, 36

No.	LR Scheduler	Rerata Loss Training	Rerata Akurasi Training	Rerata Akurasi Testing
1	Exponential Decay	2.3115	0.0995	0.1
2	Piecewise Constant Decay	2.3729	0.1007	0.1

3	Polynomial Decay	2.3081	0.1	0.1
4	Inverse Time Decay	0.62645	0.7696	0.7642

#### Analisis:

Eksperimen kali ini bertujuan mencari tahu apakah penggunaan metode learning rate schedule akan meningkatkan performa. Parameter-parameter learning rate schedule di-set default sesuai jenisnya dan di dapatkan hasil bahwa penggunaan learning rate schedule tidak meningkatkan performa bahkan menjadikan performa memburuk.

#### 1.9. Mencari Fungsi Loss Terbaik

Parameter Fix	
Ukuran filter di conv. Layer	3
Optimizer	Adam
Learning rate	0.001 (default optimizer)
Hidden unit	128
Epoch	5
Jml. Conv. Layer	2
Jml. Filter tiap Conv. Layer	24, 36

No.	Loss Fn	Rerata Loss Training	Rerata Akurasi Training	Rerata Akurasi Testing	Keterangan
1	Binary Crossentropy	9.9929	0.0979	0.09905	
2	Categorical Crossentropy				Error: incompatible
3	Sparse Categorical Crossentropy	0.20235	0.9247	0.9067	
4	Poison	10.4616	0.0991	0.0999	
5	KL Divergence	20.72325	0.0992	0.0998	

#### Analisis:

Eksperimen kali ini bertujuan untuk mencari fungsi loss terbaik yang dimiliki oleh Keras. Dari 5 fungsi loss, hanya 4 yang kompatibel dengan studi kasus di tugas ini dan didapatkan bahwa Sparse categorical Crossentropy menghasilkan perform terbaik.

#### 1.10. Kesimpulan

Parameter terbaik yang didapatkan dari hasil eksperimen adalah:

Parameter	Value
Jml. Conv. Layer	2
Jml. Filter di Conv. Layer	24, 36

Parameter	Value
Ukuran. Filter di Conv. Layer	3x3, 3x3
Hidden Unit	128
Optimizer	Adam
Learning Rate	0.001 (default)
Loss Function	Sparse Categorical Crossentropy
Epoch	5

Kesimpulan lainnya adalah beberapa parameter sangat berpengaruh terhadap performa beberapa lainnya pengaruhnya tidak signifikan, berikut ringkasannya:

Parameter	Value
Jml. Conv. Layer	Tidak signifikan
Jml. Filter di Conv. Layer	Tidak signifikan
Ukuran. Filter di Conv. Layer	Tidak signifikan
Hidden Unit	Tidak signifikan
Optimizer	Signifikan
Learning Rate	Signifikan
Loss Function	Signifikan
Epoch	Tidak signifikan

## 2. Masalah Regresi

### 2.1. Deskripsi Umum Experiment

Tugas kali ini adalah membangun model Deep Neural Network (DNN) untuk menyelesaikan masalah regresi harga rumah. Data yang digunakan adalah Boston Housing Price (penjelasan mengenai data bisa dilihat di project notebook).

Eksperimen dalam tugas ini bertujuan untuk menjawab pertanyaan-pertanyaan berikut ini:

- Berapa banyaknya hidden layer yang optimal?
- Berapa banyaknya hidden unit yang optimal di setiap hidden layer?
- Apa activation function di setiap layer sehingga hasilnya optimal?

Selain itu juga untuk mengeksplorasi mengenai fungsi optimizer dan fungsi loss pada CNN yang disediakan secara built-in di Keras.

Hyper parameter yang di observasi yaitu:

- Epoch
- Learning rate
- Jumlah hidden layer
- Jumlah neuron di tiap hidden layer
- Fungsi aktivasi di tiap hidden layer
- Fungsi optimizer
- Fungsi loss

Metode yang dilakukan adalah diawal kami menggunakan contoh sourcecode [Boston Housing Proce Prediction](#) sebagai template, kemudian kami mulai observasi parameter-parameter yang telah disebutkan diatas berurutan. Tiap parameter yang diobservasi akan menggunakan parameter lain yang paling optimal hasilnya dari eksperimen sebelumnya, kecuali jika parameter tersebut belum diobservasi maka akan menggunakan nilai default dari template.

Setiap parameter akan dilakukan percobaan menggunakan beberapa nilai diantara nilai default dari template untuk dilihat performanya. Setiap nilai parameter akan dijalankan 2 kali percobaan (running training dan test) karena karakteristik Neural Network yang selalu menghasilkan performa berbeda setiap kali di-run. Dari 2 kali percobaan kemudian dihitung rata-rata performanya. Ukuran performa yang digunakan adalah :

- Mean Absolute Error (MAE) training (0-1, dimana 1 = terbaik)
- loss training (angka kecil = terbaik)
- MAE testing (0-1, dimana 1 = terbaik)

Lingkungan eksperimen yang digunakan adalah Google Colab dengan menggunakan GPU. Untuk waktu training tidak dicatat karena perbedaannya sangat kecil antar eksperimen yang dilakukan.

## 2.2. Mencari Epoch Terbaik

Parameter Fix	
Jml. Hidden layer	2
Jml. Neuron tiap Hidden layer	100
Learning rate	0.001
Activation tiap Hidden Layer	ReLU
Optimizer	RMSprop
Loss	MSE

No.	Epoch	Rerata Loss Training	Rerata MAE Training	Rerata MAE Testing
1	100	10.70995	2.262	3.2093
2	300	5.85765	1.6907	3.2468
3	500	3.8651	1.4007	2.60335
4	700	3.392	1.2939	2.97345
5	900	2.0701	0.9957	2.9193



### Analisis:

Penambahan epoch tidak terlalu berpengaruh signifikan terhadap akurasi dan loss, namun tetap berpengaruh terhadap waktu training yang menjadi lebih lama. Setiap epoch menghabiskan waktu training sekitar 9ms. Dari table diatas terlihat bahwa MAE testing terendah ketika 500 epoch, ketika epoch ditambah justru MAE testing meningkat. Walaupun loss training dan MAE training terbaik didapatkan ketika epoch = 900 namun pertimbangan MAE testing yang terkecil dan waktu training yang lebih singkat pada saat epoch = 500, maka kami memilih epoch terbaik = 500.

### 2.3. Mencari Learning Rate Terbaik

Parameter Fix	
Jml. Hidden layer	2
Jml. Neuron tiap Hidden layer	100
Epoch	500
Activation tiap Hidden Layer	ReLU
Optimizer	RMSprop
Loss	MSE

No.	Learning Rate	Rerata Loss Training	Rerata MAE Training	Rerata MAE Testing	Keterangan
1	0.1	9.66125	2.3307	3.16865	
2	0.01	3.4625	1.2777	2.35905	
3	0.001	3.8651	1.4007	2.60335	Dari eksperimen sebelumnya
4	0.0001	16.3517	2.7447	3.2324	

### Analisis:

Kami menggunakan fungsi optimizer default template yaitu RMSprop, kami menetapkan beberapa nilai Learning Rate (LR) diantara nilai LR default dan didapatkan hasil bahwa LR = 0.01 adalah paling baik. Sedangkan LR = 0.0001 menghasilkan loss training yang besar yaitu 16.3517. Dari hasil eksperimen ini dapat disimpulkan bahwa parameter LR sangat berpengaruh terhadap performa.

### 2.4. Mencari Jumlah Lapisan Hidden

Parameter Fix	
Learning Rate	0.01
Jml. Neuron tiap Hidden layer	100
Epoch	500
Activation tiap Hidden Layer	ReLU
Optimizer	RMSprop
Loss	MSE

No.	Jml. Hidden Layer	Rerata Loss Training	Rerata MAE Training	Rerata MAE Testing	Keterangan
1	1	3.81995	1.3994	2.82275	
3	2	3.4625	1.2777	2.35905	Dari eksperimen sebelumnya
5	3	3.24055	1.3082	2.7201	

#### Analisis:

Pada eksperimen kali ini kami mencari jumlah layer hidden yang paling optimal. Tidak ada patokan nilai yang digunakan namun kami batasi hingga 3 layer karena performa yang dihasilkan berbeda jauh. Didapatkan jumlah laer hidden yang paling optimal adalah 2. Dari table diatas terlihat bahwa jumlah layer hidden tidak terlalu signifikan pengaruhnya terhadap performa model yang dihasilkan.

#### 2.5. Mencari Jumlah Hidden Unit di Tiap Lapisan

Parameter Fix	
Learning Rate	0.01
Jml. Hidden layer	2
Epoch	500
Activation tiap Hidden Layer	ReLU
Optimizer	RMSprop
Loss	MSE

No.	Hidden Unit tiap Layer	Rerata Loss Training	Rerata MAE Training	Rerata MAE Testing	Keterangan
1	50	3.37545	1.3467	2.6053	
2	100	3.4625	1.2777	2.35905	Dari eksperimen sebelumnya
3	150	3.8689	1.448	2.57535	

#### Analisis:

Seperti sebelumnya, tidak ada patokan dalam tuning hyperparameter, dalam eksperimen ini kami menetapkan beberapa jumlah Hidden Unit (HU) berdasarkan kelipatan 50. Didapatkan kesimpulan bahwa parameter jumlah HU tidak terlalu berpengaruh signifikan terhadap performa namun tetap berpengaruh pada waktu training, karena dimensi matriks yang dihasilkan akan berbeda. Dari table diatas didapat bahwa jumlah HU yang optimal adalah = 100.

#### 2.6. Mencari Fungsi Aktivasi Terbaik di Tiap Layer

Parameter Fix	
Learning Rate	0.01
Jml. Hidden layer	2
Epoch	500
Jml. Neuron tiap Hidden layer	100

Optimizer	RMSprop
Loss	MSE

No.	Activation	Rerata Loss Training	Rerata MAE Training	Rerata MAE Testing	Keterangan
1	relu	3.4625	1.2777	2.35905	Dari eksperimen sebelumnya
2	sigmoid	3.58235	1.3573	2.91805	
3	softmax	84.193	6.5967	6.51425	
4	softplus	4.5325	1.5832	2.8081	
5	softsign	2.81145	1.1679	2.7362	
6	tanh	1.8372	0.9859	3.42925	
7	selu	3.44105	1.388	3.1276	
8	elu	3.4918	1.3745	2.8774	
9	exponential				Error: incompatible

#### Analisis:

Eksperimen kali ini bertujuan untuk mencari fungsi aktivasi terbaik di tiap hidden layer yang dimiliki oleh Keras. Dari 9 fungsi loss, hanya 8 yang kompatibel dengan studi kasus di tugas ini dan didapatkan bahwa ReLU menghasilkan performa terbaik.

#### 2.7. Mencari Fungsi Optimizer Terbaik

Parameter Fix	
Learning Rate	0.01
Jml. Hidden layer	2
Epoch	500
Jml. Neuron di tiap Hidden Layer	100
Loss	MSE
Activation tiap Hidden Layer	ReLU

No.	Optimizer	Rerata Loss Training	Rerata MAE Training	Rerata MAE Testing	Keterangan
1	SGD	89.93505	6.7158	6.90685	
2	RMSprop	3.4625	1.2777	2.35905	Dari eksperimen sebelumnya
3	Adam (LR=default (0.001))	4.4426	1.4571	2.49525	
4	Adadelata	305.3833	14.143	14.9672	
5	Adagrad	45.55295	4.6507	5.15405	
6	Adamax	9.11705	2.105	3.27445	
7	Nadam	4.5289	1.4944	2.745	

8	Ftrl	56.44705	5.2742	5.70395	
9	Adam (LR=0.01)	3.5326	1.3636	2.85505	

#### Analisis:

Eksperimen kali ini bertujuan untuk mencari fungsi optimizer terbaik yang dimiliki oleh Keras dengan parameter optimizer default. Setelah dilakukan didapatkan hasil bahwa perform RMSprop dan Adam (default) hamper seimbang, sehingga kami mencoba tuning learning rate default Adam namun tetap tidak bisa mengalahkan performa RMSprop.

### 2.8. Mencari Fungsi Loss Terbaik

Parameter Fix	
Learning Rate	0.01
Jml. Hidden layer	2
Epoch	500
Jml. Neuron di tiap Hidden Layer	100
Optimizer	RMSprop
Activation tiap Hidden Layer	ReLU

No.	Loss Fn	Rerata Loss Training	Rerata MAE Training	Rerata MAE Testing	Keterangan
1	Binary Crossentropy	-321.47565	18.783	19.8105	
2	Categorical Crossentropy	2.6323E-06	21.902	22.9084	
3	Sparse Categorical Crossentropy				Error: incompatible
4	Poison	-47.9596	1.1966	2.7286	
5	KL Divergence	16.1181	22.418	23.4075	
6	MSE	3.4625	1.2777	2.35905	Dari eksperimen sebelumnya

#### Analisis:

Eksperimen kali ini bertujuan untuk mencari fungsi loss terbaik yang dimiliki oleh Keras. Dari 6 fungsi loss, hanya 5 yang kompatibel dengan studi kasus di tugas ini dan didapatkan bahwa MSE menghasilkan performa terbaik.

### 2.9. Kesimpulan

Parameter terbaik yang didapatkan dari hasil eksperimen adalah:

Parameter	Value
Jml. Hidden Layer	2

Parameter	Value
Jml. Neuron di Tiap Hidden Layer	100
Fungsi Aktivasi di Tiap Hidden Layer	ReLU
Optimizer	RMSprop
Learning Rate	0.01
Loss Function	MSE
Epoch	500

Kesimpulan lainnya adalah beberapa parameter sangat berpengaruh terhadap performa beberapa lainnya pengaruhnya tidak signifikan, berikut ringkasannya:

Parameter	Value
Jml. Hidden Layer	Tidak signifikan
Jml. Neuron di Tiap Hidden Layer	Tidak signifikan
Fungsi Aktivasi di Tiap Hidden Layer	Signifikan
Optimizer	Signifikan
Learning Rate	Signifikan
Loss Function	Signifikan
Epoch	Tidak signifikan