

Tugas Individu :

Eksplorasi Hyperparameter CNN dan Neural Network

Persoalan Klasifikasi

Dataset yang digunakan Kaggle digit-recognizer (MNIST) dan Model yang digunakan Model sequential.

Berapa banyaknya convolution layer yang optimal?

Dua convolution layer dan 3 convolution layer dicobakan dalam percobaan ini, hasil 3 convolutional layer dalam hal akurasi lebih tinggi daripada yang menggunakan 2 convolutional layer.

```
model.evaluate(x_train, y_train, verbose=2)
```

```
1313/1313 - 13s - loss: 0.0173 - accuracy: 0.9945 - 13s/epoch - 10ms/step  
[0.017250070348381996, 0.9944999814033508]
```

Gambar 1. Hasil akurasi dengan 2 convolutional layer

```
model.evaluate(x_train, y_train, verbose=2)
```

```
1313/1313 - 14s - loss: 0.0163 - accuracy: 0.9945 - 14s/epoch - 10ms/step  
[0.016264379024505615, 0.9944761991500854]
```

Gambar 2. Hasil akurasi dengan 3 convolutional layer

Berapa ukuran filter yang optimal untuk setiap convolution layer?

Pada convolution layer pertama menggunakan 5x5 kernel size dan pada convolution layer kedua dan ketiga menggunakan 3x3 kernel size

Berapa banyaknya filter yang optimal untuk setiap convolution layer?

Pada convolution layer pertama menggunakan 8 features dan pada convolution layer kedua dan ketiga menggunakan 16 features, setiap filter Tergantung dari kebutuhannya.

Berapa banyaknya hidden unit yang optimal pada bagian fully connected network?

Dua dense layer

Dari semua pilihan yang disediakan oleh Keras Optimizer, mana yang menghasilkan kinerja paling baik (pada nilai parameter default)?

Pada permasalahan ini adam optimizer menghasilkan kinerja yang lebih baik

Dari Keras Optimizer yang optimal (pada nilai parameter default), lakukan eksplorasi lebih lanjut apakah ada learning rate schedule yang menghasilkan kinerja yang lebih baik lagi. Dari semua pilihan yang disediakan oleh Keras (Probabilistic) Losses, mana yang menghasilkan kinerja paling baik?

Learning rate yang digunakan adalah 0.001, $\beta_1=0.9$, $\beta_2=0.999$, $\epsilon=1e-07$. Categorical_crossentropy, Menghitung kerugian crossentropy antara label dan prediksi, Gunakan fungsi kerugian crossentropy ini ketika ada dua atau lebih kelas label.

Persoalan Regresi

Dataset yang digunakan adalah Boston Housing Price untuk persoalan regresi ini.

Berapa banyaknya hidden layer yang optimal?

Mencobakan 1 hidden layer dan 2 hidden layer, untuk persoalan ini hasil dari 1 hidden layer lebih unggul nilai MAE (Mean Absolute Error) lebih kecil daripada yang memiliki 2 hidden layer

```
model_1.evaluate(X_test,y_test)
```

```
4/4 [=====] - 0s 5ms/step - loss: 2.9905 - mae: 2.9905  
[2.9905169010162354, 2.9905169010162354]
```

Gambar 3. Hasil MAE dari 1 hidden layer

```
model_1.evaluate(X_test,y_test)
```

```
4/4 [=====] - 0s 3ms/step - loss: 3.3385 - mae: 3.3385  
[3.3384552001953125, 3.3384552001953125]
```

Gambar 4. Hasil MAE dari 2 hidden layer

Berapa banyaknya hidden unit yang optimal di setiap hidden layer?

Disetiap hidden layer terdapat 10 features

Apa activation function di setiap layer sehingga hasilnya optimal?

Dari hasil percobaan menggunakan sigmoid, relu, softmax, didapatkan hasil dengan menggunakan activation function relu dan linear lebih baik pada persoalan ini.

Dari semua pilihan optimizer, apa optimizer yang hasilnya optimal?

Adam optimizer menghasilkan hasil yang lebih baik dari pada optimizer yang lainnya

Dari semua pilihan loss function, apa yang hasilnya optimal?

Loss function yang digunakan pada persoalan ini adalah MAE (Mean Absolute Error)

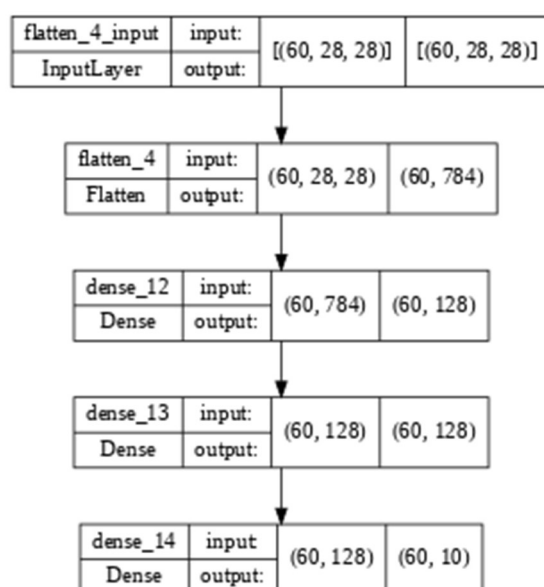
Gambar ilustrasi dari arsitektur CNN dan Fully Connected NN yang optimal.

Pada Persoalan Klasifikasi

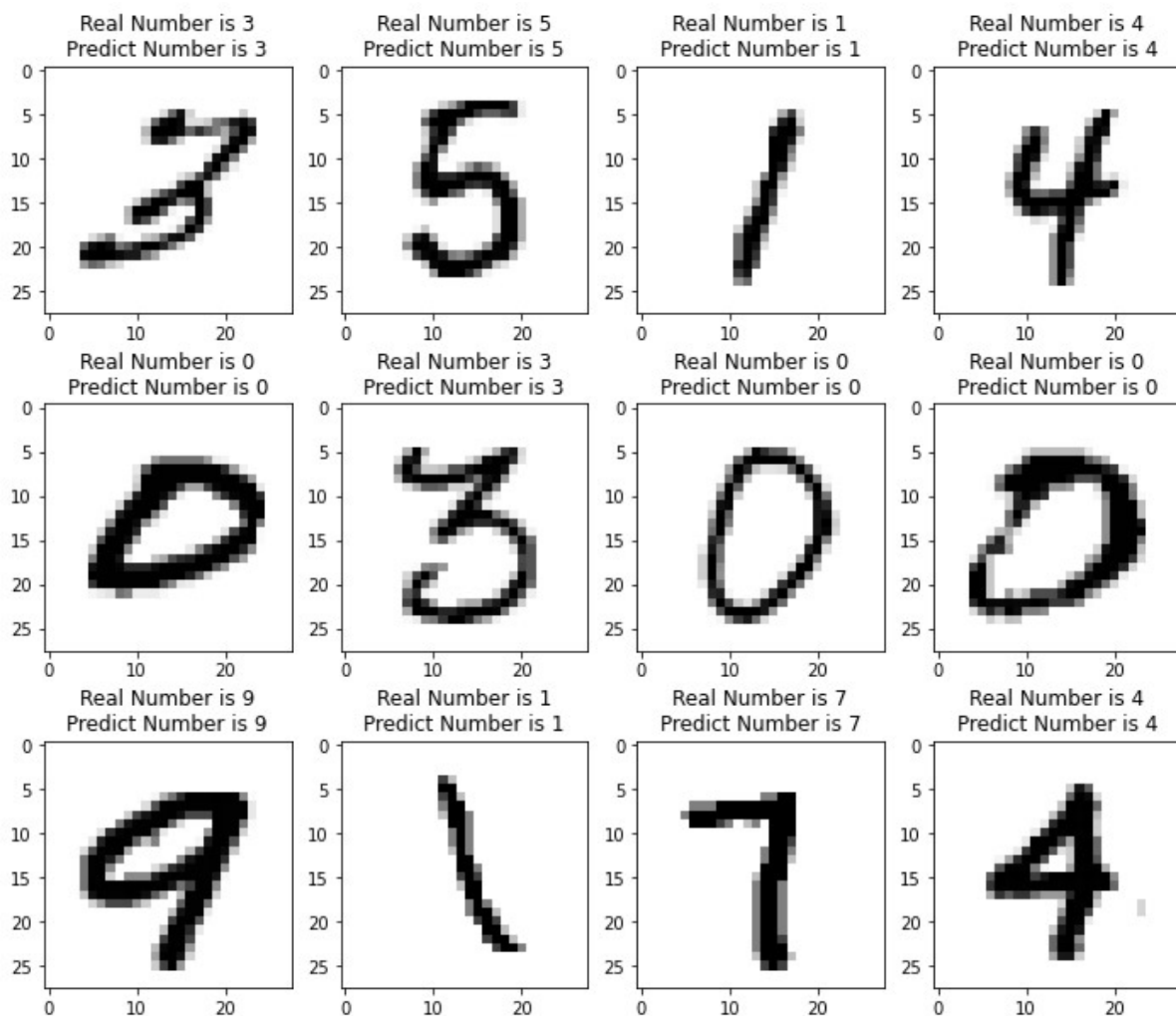
Model: "sequential_4"

Layer (type)	Output Shape	Param #
flatten_4 (Flatten)	(60, 784)	0
dense_12 (Dense)	(60, 128)	100480
dense_13 (Dense)	(60, 128)	16512
dense_14 (Dense)	(60, 10)	1290

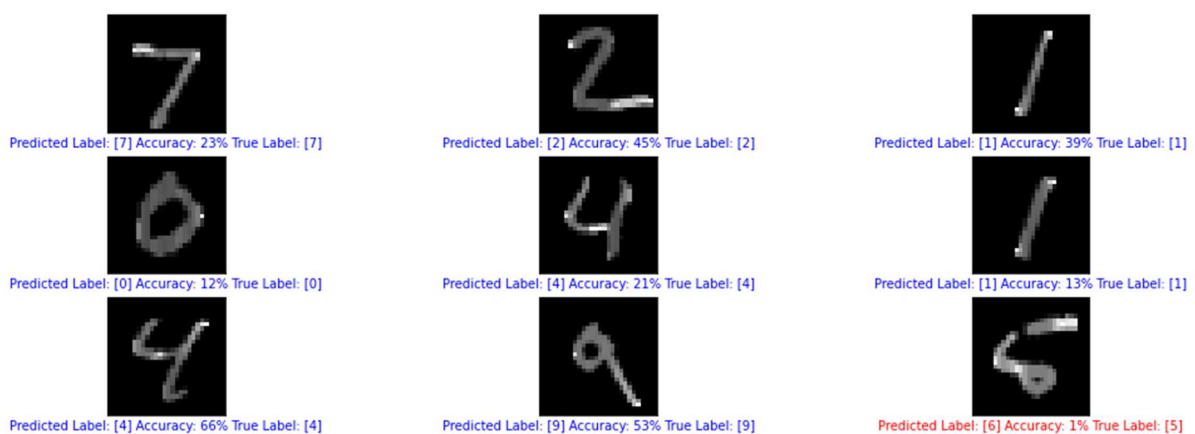
=====
Total params: 118,282
Trainable params: 118,282
Non-trainable params: 0
=====



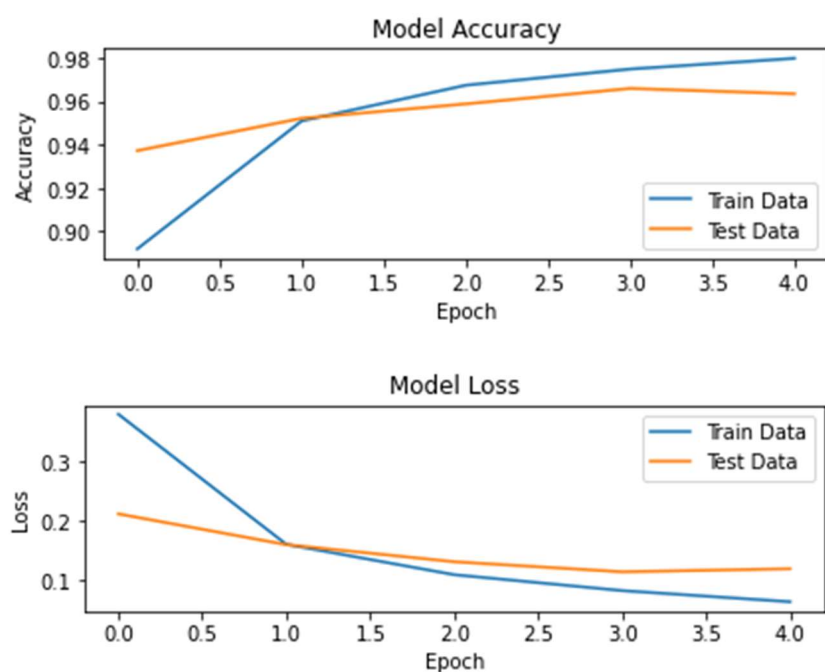
contoh beberapa tampilan input



Tampilan prediksi yang salah seperti gambar di bawah ini, hanya saja dengan model CNN yang berbeda



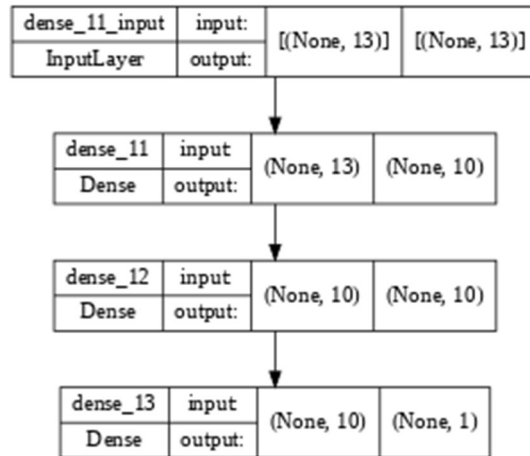
Kinerja model



Pada Persoalan Regresi

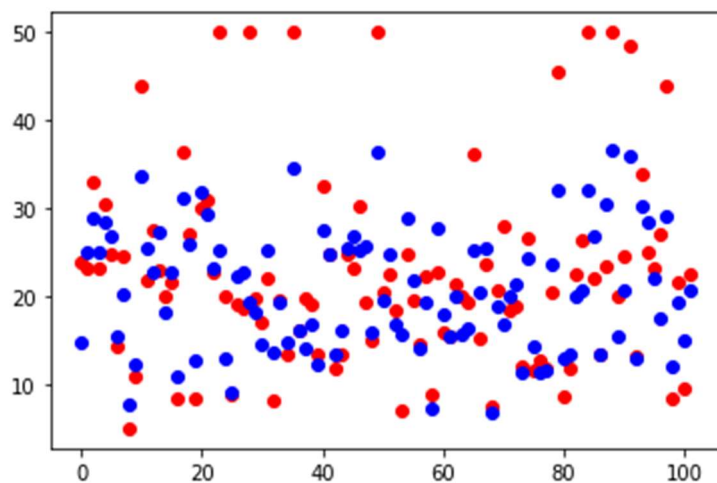
Model: "sequential_3"

Layer (type)	Output Shape	Param #
dense_11 (Dense)	(None, 10)	140
dense_12 (Dense)	(None, 10)	110
dense_13 (Dense)	(None, 1)	11
Total params: 261		
Trainable params: 261		
Non-trainable params: 0		



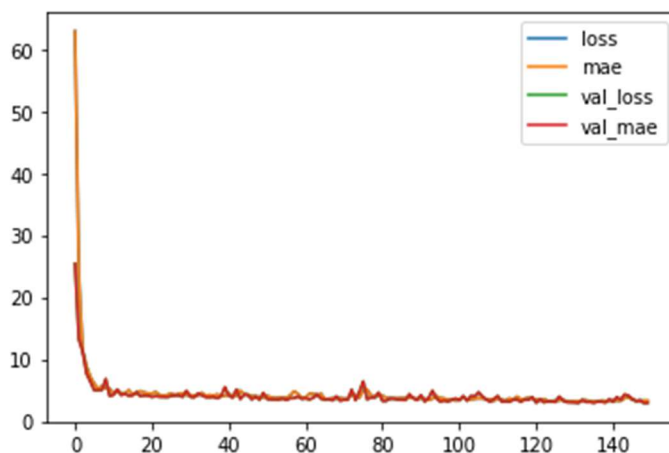
Contoh Tampilan Regresi

```
plt.figure()
plt.scatter(range(len(y_test)), y_test, color='red')
plt.scatter(range(len(y_pred)), y_pred, color='blue');
```



Kinerja model

```
pd.DataFrame(history_1.history).plot();
```



Kesimpulan akhir dari hasil eksplorasi ini

- ✓ Dari percobaan klasifikasi dapat disimpulkan dengan bertambahnya layer dalam CNN maka hasil akurasi akan bertambah, untuk activation function yang cocok digunakan adalah RELU, SOFTMAX dan untuk Optimizer menggunakan ADAM optimizer sedang untuk setiap iterasi dalam epoch semakin banyak epoch maka akan meningkatkan akurasi dan Saat learning rate kecil maka perubahan bobot semakin kecil dan kedekatan pola ke kelas lain dari inisial target juga semakin jauh.
- ✓ Untuk persoalan regresi pada persoalan ini dapat disimpulkan 1 layer sudah cukup untuk mendapatkan nilai MAE (Mean Absolute Error) lebih kecil, serta untuk activation function yang cocok digunakan adalah RELU dan LINEAR, Adam optimizer masih unggul sebagai optimizer, Loss function yang digunakan pada persoalan ini adalah MAE (Mean Absolute Error).