

TUGAS

ANALISIS PENGARUH KOMPONEN PADA DEEP LEARNING

MATAKULIAH KAPITA SELEKTA SISTEM CERDAS II

MEI PRABOWO

NIM. 23/530818/SPA/01016

1. CNN Model 1

Pembuatan model CNN pada percobaan pertama pada setiap layer mempunyai aturan atau nilai yang sudah ditentukan seperti pada uraian Berikut ini :

```
model = models.Sequential()
# Layer 1
model.add(layers.Conv2D(64, (3, 3), activation='relu', input_shape=(32, 32, 3)))
model.add(layers.MaxPooling2D((2, 2)))
# Layer 2
model.add(layers.Conv2D(128, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
# Layer 3
model.add(layers.Conv2D(256, (3, 3), activation='relu'))
# Flatten layer
model.add(layers.Flatten())
# Dense layers
model.add(layers.Dense(128, activation='relu'))
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(10, activation='softmax'))
```

Hasil Akurasi

```
opt = keras.optimizers.Adam(learning_rate=0.001)
model.compile(optimizer=opt, loss="categorical_crossentropy", metrics=['accuracy'])
model.fit(train_images, train_labelsH, epochs=20, batch_size=200, validation_data=(test_images, test_labelsH))
```

```
Epoch 1/20
250/250 [=====] - 7s 15ms/step - loss: 1.6815 - accuracy: 0.3766 - val_loss: 1.3979 - val_accuracy: 0.4948
Epoch 2/20
250/250 [=====] - 3s 12ms/step - loss: 1.2774 - accuracy: 0.5429 - val_loss: 1.2137 - val_accuracy: 0.5733
Epoch 3/20
250/250 [=====] - 3s 12ms/step - loss: 1.1010 - accuracy: 0.6112 - val_loss: 1.0564 - val_accuracy: 0.6272
Epoch 4/20
250/250 [=====] - 3s 14ms/step - loss: 0.9655 - accuracy: 0.6602 - val_loss: 0.9788 - val_accuracy: 0.6564
Epoch 5/20
250/250 [=====] - 3s 12ms/step - loss: 0.8772 - accuracy: 0.6928 - val_loss: 0.8838 - val_accuracy: 0.6938
Epoch 6/20
250/250 [=====] - 3s 12ms/step - loss: 0.8015 - accuracy: 0.7194 - val_loss: 0.8558 - val_accuracy: 0.7071
Epoch 7/20
250/250 [=====] - 3s 12ms/step - loss: 0.7424 - accuracy: 0.7405 - val_loss: 0.8839 - val_accuracy: 0.7005
Epoch 8/20
250/250 [=====] - 3s 14ms/step - loss: 0.6807 - accuracy: 0.7635 - val_loss: 0.8081 - val_accuracy: 0.7230
Epoch 9/20
250/250 [=====] - 3s 13ms/step - loss: 0.6279 - accuracy: 0.7807 - val_loss: 0.8053 - val_accuracy: 0.7228
Epoch 10/20
250/250 [=====] - 4s 14ms/step - loss: 0.5731 - accuracy: 0.8013 - val_loss: 0.7758 - val_accuracy: 0.7439
Epoch 11/20
250/250 [=====] - 3s 12ms/step - loss: 0.5376 - accuracy: 0.8112 - val_loss: 0.7690 - val_accuracy: 0.7441
Epoch 12/20
250/250 [=====] - 3s 14ms/step - loss: 0.4891 - accuracy: 0.8279 - val_loss: 0.7603 - val_accuracy: 0.7502
Epoch 13/20
250/250 [=====] - 3s 12ms/step - loss: 0.4495 - accuracy: 0.8426 - val_loss: 0.7993 - val_accuracy: 0.7409
Epoch 14/20
250/250 [=====] - 3s 12ms/step - loss: 0.4019 - accuracy: 0.8596 - val_loss: 0.8100 - val_accuracy: 0.7501
Epoch 15/20
250/250 [=====] - 3s 13ms/step - loss: 0.3604 - accuracy: 0.8740 - val_loss: 0.8315 - val_accuracy: 0.7431
Epoch 16/20
250/250 [=====] - 3s 13ms/step - loss: 0.3227 - accuracy: 0.8859 - val_loss: 0.8817 - val_accuracy: 0.7459
Epoch 17/20
250/250 [=====] - 3s 12ms/step - loss: 0.2824 - accuracy: 0.9015 - val_loss: 0.9457 - val_accuracy: 0.7426
Epoch 18/20
250/250 [=====] - 3s 13ms/step - loss: 0.2476 - accuracy: 0.9133 - val_loss: 0.9457 - val_accuracy: 0.7487
Epoch 19/20
250/250 [=====] - 3s 13ms/step - loss: 0.2196 - accuracy: 0.9236 - val_loss: 1.0439 - val_accuracy: 0.7372
Epoch 20/20
250/250 [=====] - 3s 13ms/step - loss: 0.1931 - accuracy: 0.9320 - val_loss: 1.1041 - val_accuracy: 0.7383
<keras.src.callbacks.History at 0x7afc748787c0>
```

2. CNN Model 2

Pembuatan model CNN pada percobaan kedua pada setiap layer mempunyai aturan atau nilai yang sudah ditentukan seperti pada uraian Berikut ini :

```
model = models.Sequential()  
model.add(layers.Conv2D(64,(3,3), activation='relu', input_shape=(32,32,3)))  
model.add(layers.MaxPool2D(2,2))  
model.add(layers.Conv2D(128,(3,3), activation='relu'))  
model.add(layers.MaxPool2D(2,2))  
model.add(layers.Conv2D(128,(3,3), activation='relu'))  
model.add(layers.Flatten())  
model.add(layers.Dense(128, activation='relu'))  
model.add(layers.Dense(10, activation='softmax'))
```

Hasil Akurasi

```
opt = keras.optimizers.Adam(learning_rate=0.001)  
model.compile(optimizer=opt, loss="categorical_crossentropy", metrics=['accuracy'])  
model.fit(train_images, train_labelsH, epochs=30, batch_size=200, validation_data=(test_images, test_labelsH))  
  
250/250 [=====] - 3s 11ms/step - loss: 1.2616 - accuracy: 0.5496 - val_loss: 1.1836 - val_accuracy: 0.5781  
Epoch 3/30  
250/250 [=====] - 3s 12ms/step - loss: 1.1010 - accuracy: 0.6125 - val_loss: 1.0204 - val_accuracy: 0.6373  
Epoch 4/30  
250/250 [=====] - 3s 12ms/step - loss: 0.9973 - accuracy: 0.6513 - val_loss: 1.0096 - val_accuracy: 0.6476  
Epoch 5/30  
250/250 [=====] - 3s 13ms/step - loss: 0.9034 - accuracy: 0.6840 - val_loss: 0.9417 - val_accuracy: 0.6702  
Epoch 6/30  
250/250 [=====] - 3s 11ms/step - loss: 0.8363 - accuracy: 0.7073 - val_loss: 0.8814 - val_accuracy: 0.7008  
Epoch 7/30  
250/250 [=====] - 3s 12ms/step - loss: 0.7797 - accuracy: 0.7274 - val_loss: 0.8638 - val_accuracy: 0.7010  
Epoch 8/30  
250/250 [=====] - 3s 12ms/step - loss: 0.7217 - accuracy: 0.7485 - val_loss: 0.8383 - val_accuracy: 0.7111  
Epoch 9/30  
250/250 [=====] - 3s 13ms/step - loss: 0.6802 - accuracy: 0.7623 - val_loss: 0.8757 - val_accuracy: 0.7068  
Epoch 10/30  
250/250 [=====] - 3s 12ms/step - loss: 0.6336 - accuracy: 0.7808 - val_loss: 0.8329 - val_accuracy: 0.7218  
Epoch 11/30  
250/250 [=====] - 3s 11ms/step - loss: 0.6021 - accuracy: 0.7891 - val_loss: 0.8015 - val_accuracy: 0.7255  
Epoch 12/30  
250/250 [=====] - 3s 11ms/step - loss: 0.5491 - accuracy: 0.8100 - val_loss: 0.7882 - val_accuracy: 0.7379  
Epoch 13/30  
250/250 [=====] - 3s 12ms/step - loss: 0.5246 - accuracy: 0.8151 - val_loss: 0.8071 - val_accuracy: 0.7289  
Epoch 14/30  
250/250 [=====] - 3s 11ms/step - loss: 0.4811 - accuracy: 0.8315 - val_loss: 0.8186 - val_accuracy: 0.7365  
Epoch 15/30  
250/250 [=====] - 3s 11ms/step - loss: 0.4515 - accuracy: 0.8445 - val_loss: 0.8422 - val_accuracy: 0.7309  
Epoch 16/30  
250/250 [=====] - 3s 12ms/step - loss: 0.4178 - accuracy: 0.8543 - val_loss: 0.8457 - val_accuracy: 0.7373  
Epoch 17/30  
250/250 [=====] - 3s 13ms/step - loss: 0.3833 - accuracy: 0.8655 - val_loss: 0.8795 - val_accuracy: 0.7313  
Epoch 18/30  
250/250 [=====] - 3s 11ms/step - loss: 0.3571 - accuracy: 0.8762 - val_loss: 0.8999 - val_accuracy: 0.7327  
Epoch 19/30  
250/250 [=====] - 3s 12ms/step - loss: 0.3335 - accuracy: 0.8837 - val_loss: 0.9301 - val_accuracy: 0.7255  
Epoch 20/30  
250/250 [=====] - 3s 12ms/step - loss: 0.2937 - accuracy: 0.8991 - val_loss: 0.9958 - val_accuracy: 0.7232  
Epoch 21/30  
250/250 [=====] - 3s 13ms/step - loss: 0.2747 - accuracy: 0.9046 - val_loss: 0.9822 - val_accuracy: 0.7344  
Epoch 22/30  
250/250 [=====] - 3s 12ms/step - loss: 0.2404 - accuracy: 0.9161 - val_loss: 1.0161 - val_accuracy: 0.7293  
Epoch 23/30  
250/250 [=====] - 3s 12ms/step - loss: 0.2161 - accuracy: 0.9249 - val_loss: 1.1465 - val_accuracy: 0.7280  
Epoch 24/30  
250/250 [=====] - 3s 12ms/step - loss: 0.1933 - accuracy: 0.9333 - val_loss: 1.1737 - val_accuracy: 0.7267  
Epoch 25/30  
250/250 [=====] - 3s 13ms/step - loss: 0.1715 - accuracy: 0.9416 - val_loss: 1.2475 - val_accuracy: 0.7140  
Epoch 26/30  
250/250 [=====] - 3s 12ms/step - loss: 0.1622 - accuracy: 0.9436 - val_loss: 1.3290 - val_accuracy: 0.7118  
Epoch 27/30  
250/250 [=====] - 3s 11ms/step - loss: 0.1349 - accuracy: 0.9538 - val_loss: 1.3924 - val_accuracy: 0.7148  
Epoch 28/30  
250/250 [=====] - 3s 12ms/step - loss: 0.1251 - accuracy: 0.9576 - val_loss: 1.4172 - val_accuracy: 0.7241  
Epoch 29/30  
250/250 [=====] - 3s 13ms/step - loss: 0.1178 - accuracy: 0.9597 - val_loss: 1.5301 - val_accuracy: 0.7098  
Epoch 30/30  
250/250 [=====] - 3s 12ms/step - loss: 0.1036 - accuracy: 0.9643 - val_loss: 1.5580 - val_accuracy: 0.7135  
<keras.src.callbacks.History at 0x7c3d1452d450>
```

KESIMPULAN

Pada percobaan pertama dengan komposisi pada layer convolusi pertama 64 filter, konvolusi kedua 128 filter dan konvolusi ketiga 256 filter. Optimasi yang digunakan yaitu Adam dengan epoch 20 hasil akurasi terbaik yaitu 93%.

Pada percobaan kedua dengan komposisi pada layer convolusi pertama 64 filter, konvolusi kedua 128 filter dan konvolusi ketiga 128 filter. Optimasi yang digunakan yaitu Adam dengan epoch 30 hasil akurasi terbaik yaitu 96%.

Menambahkan layer pada jaringan CNN dapat meningkatkan kemampuannya untuk mempelajari hal ini dapat meningkatkan akurasi model. Namun, hal berimbas pada meningkatkan waktu dan beban komputasi. Epoch lebih banyak memiliki akurasi lebih tinggi akan tetapi membutuhkan waktu lebih lama sedangkan Epoch lebih sedikit pelatihan lebih cepat akan tetapi akurasi lebih rendah.