Nama : Handoko Supeno

NIM: 33220026

Tugas Eksplorasi Hyperparameter

PERSOALAN KLASIFIKASI

Pada eksperimen ini peneliti menggunakan CNN untuk melakukan training menggunakan dataset CIFAR10. Terdapat dua arsitektur yang diterapkan, yang dituliskan pada file cifar10_versi3.ipynb dan cifar10_versi3best.ipynb.

Berikut adalah kompararasi antara arsitektur best dengan accuracy 84% dengan arsitektur yang alternatif dengan akurasi 78%.

Arsitektur Best

```
cnn = models.Sequential([
        #cnn layers
        layers.Conv2D(filters=32, kernel_size=(3,3), activation='relu',input_shape=(32,32,3)),
        layers.MaxPooling2D((2,2)),
        layers.Conv2D(filters=64, kernel_size=(3,3), activation='relu'),
        layers.MaxPooling2D((2,2)),

        #dense layers
        layers.Flatten(),
        layers.Dense(128,activation='relu'),
        layers.Dense(64,activation='relu'),
        layers.Dense(10, activation='softmax')
      ])
```

```
cnn.compile(
    optimizer='adam',
    loss = 'sparse_categorical_crossentropy',
    metrics = ['accuracy']
)
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 30, 30, 32)        896

 max_pooling2d (MaxPooling2D  (None, 15, 15, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 13, 13, 64)        18496

 max_pooling2d_1 (MaxPooling  (None, 6, 6, 64)         0
 2D)

 flatten (Flatten)           (None, 2304)              0

 dense (Dense)               (None, 128)               295040

 dense_1 (Dense)             (None, 64)                8256

 dense_2 (Dense)             (None, 10)                650

=================================================================
Total params: 323,338
Trainable params: 323,338
Non-trainable params: 0
```

Arsitektur Alternatif

```
cnn = models.Sequential([
        #cnn layers
        layers.Conv2D(filters=32, kernel_size=(5,5), activation='relu',input_shape=(32,32,3)),
        layers.MaxPooling2D((2,2)),

        #dense layers
        layers.Flatten(),
        layers.Dense(64,activation='relu'),
        layers.Dense(10, activation='softmax')
      ])
```

```
cnn.compile(
    optimizer='adam',
    loss = 'sparse_categorical_crossentropy',
    metrics = ['accuracy']
)
```

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_4 (Conv2D)           (None, 28, 28, 32)        2432

 max_pooling2d_4 (MaxPooling  (None, 14, 14, 32)       0
 2D)

 flatten_3 (Flatten)         (None, 6272)              0

 dense_9 (Dense)             (None, 64)                401472

 dense_10 (Dense)            (None, 10)                650

=================================================================
Total params: 404,554
Trainable params: 404,554
Non-trainable params: 0
_________________________________________________________________
```

Disini dapat kita simpulkan bahwa semakin dalam sebuah deep learning maka semakin tinggi akurasinya walaupun pada arsitektur alternatif memiliki jumlah parameter yang lebih besar. Pada arsitektur best ukuran filter yang digunakan adalah 3x3 sedangkan pada arsitektur alternatif ukuran filter yang digunakan adalah 5x5. Dari percobaan ini dapat disimpulkan bahwa ukuran filter yang lebih besar tidak menjamin peningkatan performa, justru ukuran filter yang lebih kecil membuat pemrosesan lebih cepat. Selanjutnya hidden unit pada bagian fully connected di arsitektur best lebih banyak satu layer dibandingkan dengan arsitektur alternatif sehingga dapat disimpulkan bahwa dengan penambahan layer di bagian fully connected akan menambahkan akurasi.

PERSOALAN REGRESI

Telah dibangun arsitektur Fully Connected Neural Network untuk dataset Boston Housing Price pada persolan regressi. Ada dua arsitektur yaitu boston_housing_regression_dnn_best dan boston_housing_regression_dnn sebagai arsitektur alternatif. Arsitektur best berhasil mendapatkan loss 17.81 sedangkan asitektur alternatif mendapatkan loss lebih besar diangka 17.83.

Arsitektur best memiliki arsitektur sebagai berikut:

```
def HousePricePredictionModel():
  model = Sequential()
  model.add(Dense(128,activation='relu',input_shape=(train_x[0].shape)))
  model.add(Dense(64,activation='relu'))
  model.add(Dense(32,activation='relu'))
  model.add(Dense(1))
  model.compile(optimizer='rmsprop',loss='mse',metrics=['mae'])
  return model
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 128)               1792

 dense_1 (Dense)             (None, 64)                8256

 dense_2 (Dense)             (None, 32)                2080

 dense_3 (Dense)             (None, 1)                 33

=================================================================
Total params: 12,161
Trainable params: 12,161
Non-trainable params: 0
_________________________________________________________________
```

Sedangkan arsitektur alternatif memiliki arsitektur sebagai berikut:

```
def HousePricePredictionModel():
  model = Sequential()
  model.add(Dense(128,activation='sigmoid'))
  model.add(Dense(64,activation='sigmoid'))
  model.add(Dense(1))
  model.compile(optimizer='adam',loss='mape',metrics=['mae'])
  return model
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (1, 128)                  1792

 dense_1 (Dense)             (1, 64)                   8256

 dense_2 (Dense)             (1, 1)                    65

=================================================================
Total params: 10,113
Trainable params: 10,113
Non-trainable params: 0
_________________________________________________________________
```

Arsitektur best memiliki hidden layer yang lebih banyak yaitu 4 hidden layer yang berbeda dengan arsitektur alternatif dengan 3 hidden layer. Ini menyimpulkan bahwa hidden layer yang lebih banyak dapat meningkatkan akurasi. Otomatis total disemua layer juga berkurang pada arsitektur alternatif dan ini juga membuat loss yang lebih tinggi pada akhir pelatihan pada arsitektur alternatif. Arsitektur best menggunakan activation function relu sedangkan yang alternatif menggunakan activation function sigmoid. Untuk meningkatkan performa pada arsitektur alternatif menggunakan optimizer adam sedangkan yang best menggunakan rmsprop. Adam terbukti cukup kompetitif terhadap rmsprop. Loss function pada kedua arsitektur juga dibedakan dimana pada arsitektur alternatif menggunakan mape(Mean Absolute Percentage Error) sedangkan pada arsitektur best menggunakan mse(minimum squared error)

Pada bagian regresi utuk dataset boston housing price didapatkan dihasilkan regression plot sebagai berikut:


PRICE_REG
40
30
20
10
0
10
20
30
40
50
PRICE