

# Dog Cat CNN Classifier

```
#imports
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
from tensorflow.keras.backend import clear_session
```

```
!pip install optuna
```

```
Collecting optuna
  Downloading optuna-2.9.1-py3-none-any.whl (302 kB)
    |████████████████████| 302 kB 23.0 MB/s
Collecting alembic
  Downloading alembic-1.6.5-py2.py3-none-any.whl (164 kB)
    |████████████████████| 164 kB 74.5 MB/s
Requirement already satisfied: numpy in /shared-libs/python3.7/py/lib/python3.7/site-packages (from
Requirement already satisfied: scipy!=1.4.0 in /shared-libs/python3.7/py/lib/python3.7/site-packag
Requirement already satisfied: sqlalchemy>=1.1.0 in /shared-libs/python3.7/py/lib/python3.7/site-p
Collecting cliff
  Downloading cliff-3.8.0-py3-none-any.whl (80 kB)
    |████████████████████| 80 kB 19.8 MB/s
Collecting colorlog
  Downloading colorlog-6.4.1-py2.py3-none-any.whl (11 kB)
Requirement already satisfied: packaging>=20.0 in /shared-libs/python3.7/py-core/lib/python3.7/sit
Collecting cmaes>=0.8.2
  Downloading cmaes-0.8.2-py3-none-any.whl (15 kB)
Requirement already satisfied: tqdm in /shared-libs/python3.7/py/lib/python3.7/site-packages (from
Requirement already satisfied: PyYAML in /shared-libs/python3.7/py/lib/python3.7/site-packages (fr
Requirement already satisfied: pyparsing>=2.0.2 in /shared-libs/python3.7/py-core/lib/python3.7/si
Requirement already satisfied: greenlet!=0.4.17 in /shared-libs/python3.7/py/lib/python3.7/site-pa
Requirement already satisfied: importlib-metadata in /shared-libs/python3.7/py-core/lib/python3.7/
Collecting python-editor>=0.3
  Downloading python_editor-1.0.4-py3-none-any.whl (4.9 kB)
Collecting Mako
  Downloading Mako-1.1.5-py2.py3-none-any.whl (75 kB)
    |████████████████████| 75 kB 9.4 MB/s
Requirement already satisfied: python-dateutil in /shared-libs/python3.7/py-core/lib/python3.7/sit
Collecting pbr!=2.1.0,>=2.0.0
  Downloading pbr-5.6.0-py2.py3-none-any.whl (111 kB)
    |████████████████████| 111 kB 66.0 MB/s
Collecting stevedore>=2.0.1
```

```
import optuna
```

```
#Add the model

dropout_rate = [0] * 2

def create_model(trial):

    num_layers = trial.suggest_int("num_layers", 1, 5)
    activation = trial.suggest_categorical("activation", ["relu", "linear", "selu", "elu", "exponential"])
    dropout_rate[0] = trial.suggest_uniform('dropout_rate'+str(0), 0.0, 0.5)
    dropout_rate[1] = trial.suggest_uniform('dropout_rate'+str(1), 0.0, 0.5)
    mid_units = int(trial.suggest_discrete_uniform("mid_units", 100, 300, 100))
    filters=trial.suggest_categorical("filters", [32, 64])
    kernel_size=trial.suggest_categorical("kernel_size", [3, 3])
    strides=trial.suggest_categorical("strides", [1, 2])

    classifier = Sequential()

    #step 1 - Convolution Layers

    classifier.add(
```

```

        Conv2D(
            filters=filters,
            kernel_size=kernel_size,
            strides=strides,
            activation = activation,
            input_shape=(64, 64, 3),
        )
    )

    classifier.add(MaxPooling2D(pool_size=(2, 2)))
    for i in range(1, num_layers):
        classifier.add(
            Conv2D(
                filters=filters,
                kernel_size=kernel_size,
                strides=strides,
                activation = activation,
            )
        )
    classifier.add(MaxPooling2D(pool_size=(2, 2)))
    classifier.add(Dropout(dropout_rate[0]))
    classifier.add(Flatten())
    classifier.add(Dense(units = mid_units, activation = activation))
    classifier.add(Dropout(dropout_rate[1]))
    classifier.add(Dense(units = 1, activation = 'sigmoid'))

    return classifier

```

```

#image augumentation
from keras.preprocessing.image import ImageDataGenerator

#Data Preparation

train_datagen = ImageDataGenerator(rescale = 1./255,
                                   shear_range = 0.2,
                                   zoom_range = 0.2,
                                   horizontal_flip = True)

test_datagen = ImageDataGenerator(rescale = 1./255)

training_set = train_datagen.flow_from_directory('cats_and_dogs_filtered/train',
                                                target_size = (64, 64),
                                                batch_size = 32,
                                                class_mode = 'binary')

test_set = test_datagen.flow_from_directory('cats_and_dogs_filtered/validation',
                                           target_size = (64, 64),
                                           batch_size = 32,
                                           class_mode = 'binary')

```

Found 1645 images belonging to 2 classes.  
Found 1000 images belonging to 2 classes.

training\_set

<tensorflow.python.keras.preprocessing.image.DirectoryItera

```

#create objective
def objective(trial):
    #clear session
    clear_session()

    optimizer = trial.suggest_categorical("optimizer", ["sgd", "adam", "rmsprop", "adadelta", "adagrad", "adamax"])

    classifier = create_model(trial)

    classifier.compile(optimizer = optimizer, loss = 'binary_crossentropy', metrics = ['accuracy'])

    history = classifier.fit_generator(training_set,
                                      steps_per_epoch = 30, # num_samples // batch_size
                                      epochs = 1, # entire iteration over dataset
                                      validation_data = test_set,
                                      validation_steps = 20) #https://keras.io/api/models/model_training-apis/

```

```
return 1 - history.history["accuracy"][-1]
```

```
#perform study
study = optuna.create_study()
study.optimize(objective, n_trials=5)
```

```
[I 2021-08-24 23:01:16,844] A new study created in memory w
/shared-libs/python3.7/py/lib/python3.7/site-packages/tensc
warnings.warn(`Model.fit_generator` is deprecated and `
30/30 [=====] - 7s 210ms/step - lc
[I 2021-08-24 23:01:23,714] Trial 0 finished with value: 0.
/shared-libs/python3.7/py/lib/python3.7/site-packages/tensc
warnings.warn(`Model.fit_generator` is deprecated and `
30/30 [=====] - 6s 202ms/step - lc
[I 2021-08-24 23:01:30,339] Trial 1 finished with value: 0.
/shared-libs/python3.7/py/lib/python3.7/site-packages/tensc
warnings.warn(`Model.fit_generator` is deprecated and `
30/30 [=====] - 27s 896ms/step - l
[I 2021-08-24 23:01:57,850] Trial 2 finished with value: 0.
/shared-libs/python3.7/py/lib/python3.7/site-packages/tensc
warnings.warn(`Model.fit_generator` is deprecated and `
30/30 [=====] - 28s 877ms/step - l
[I 2021-08-24 23:02:25,666] Trial 3 finished with value: 0.
/shared-libs/python3.7/py/lib/python3.7/site-packages/tensc
warnings.warn(`Model.fit_generator` is deprecated and `
30/30 [=====] - 7s 213ms/step - lc
[I 2021-08-24 23:02:32,590] Trial 4 finished with value: 0.
```

```
study.best_params
```

```
{'optimizer': 'adagrad',
 'num_layers': 2,
 'activation': 'linear',
 'dropout_rate0': 0.4469333482678931,
 'dropout_rate1': 0.06347025441884302,
 'mid_units': 100.0,
 'filters': 64,
 'kernel_size': 3,
 'strides': 1}
```

```
study.best_value
```

```
0.4115501642227173
```

```
print("Number of finished trials: {}".format(len(study.trials)))

print("Best trial:")
trial = study.best_trial

print("  Value: {}".format(trial.value))

print("  Params: ")
for key, value in trial.params.items():
    print("    {}: {}".format(key, value))
```