

Hướng dẫn cách huấn luyện mô hình mạng nơ ron CNN với Keras

1. Đầu tiên chuẩn bị tập dữ liệu

Tập dữ liệu chữ cái được tải ở trang:

<http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/>. Với tên tệp là [EnglishImg.tgz](#)

2. Xử lý dữ liệu

- Gán nhãn cho từng thư mục hình theo thứ tự bảng chữ cái
- Thay đổi kích thước hình ảnh 32x32
- Chuyển về dạng mảng (dtype=np.uint8)

3. Chia tập dữ liệu để huấn luyện và kiểm tra

```
#phan chia tap du lieu
X_train, X_test, Y_train, Y_test = train_test_split(dig, lab, test_size = 0.2, random_state = 100, shuffle = True)
```

Chia tập train và test theo tỉ lệ 8:2

4. Chuyển về dạng mảng np.array

```
X_train = np.array(X_train)
Y_train = np.array(Y_train)
#tap test
X_test = np.array(X_test)
Y_test = np.array(Y_test)
```

5. Thay đổi cấu trúc reshape

```
img_rows, img_cols = 32,32 # kích thước ảnh
X_train = X_train.reshape(X_train.shape[0], img_rows, img_cols,1)
X_train = X_train.astype('float32')
X_train /= 255

X_test = X_test.reshape(X_test.shape[0], img_rows, img_cols,1)
X_test = X_test.astype('float32')
X_test /= 255
```

$X_train /= 255$ là chuyển về dạng nhị phân

6. Chuyển nhãn về dạng 0 1 (one hot encoding)

```
#one hot encoding
Y_train = keras.utils.to_categorical(Y_train, num_classes)
Y_test = keras.utils.to_categorical(Y_test, num_classes)
```

Với num_class là số nhãn (ở đây có 62 nhãn)

Đây là các bước tiền xử lý dữ liệu để huấn luyện. Tiếp theo sẽ là xây dựng mô hình mạng với keras.

1. Khai báo thư viện cần thiết

```
from keras.layers import Conv2D, Dense, Dropout, Flatten, MaxPooling2D
from keras.models import Sequential
```

Model sử dụng là Sequential

Các tầng mạng sử dụng là Conv2D, Dense, Dropout, Flatten, MaxPooling2D

2. Khởi tạo model

```
#khởi tạo model
model = Sequential()
```

3. Thêm các tầng mạng

```
model.add(Conv2D(32,(3,3), input_shape = (32,32,1), activation='relu'))
model.add(Conv2D(32,(3,3),activation='relu')),
model.add(MaxPooling2D(pool_size=(2,2))),
model.add(Dropout(0.25)),
model.add(Conv2D(64,(3,3),activation='relu')),
model.add(Conv2D(64,(3,3),activation='relu')),
model.add(MaxPooling2D(pool_size=(2,2))),
model.add(Dropout(0.25)),
model.add(Flatten(input_shape=(32,32))), #chuyển về vector
model.add(Dense(128, activation='relu')),
model.add(Dropout(0.5)),
model.add(Dense(62))
```

Thêm các tầng bằng *model.add*

- Tầng đầu là Conv2D có 32 filter, mỗi filter có size (3x3). Với input_shape là kích thước ảnh đầu vào (32x32x1). Hàm kích hoạt là hàm “relu”
- Tầng 2 là tầng chập giống của tầng đầu
- Tầng 3 là tầng tổng hợp MaxPooling2D với size là (2x2)
- Tầng 4 là tầng Conv2D có 64 filter, mỗi filter có size (3x3)
- Tầng 5 là tầng chập tương tự tầng 5
- Tầng 6 là tầng tổng hợp MaxPooling với size là (2x2)
- Tầng 7 là tầng làm phẳng Flatten
- Tầng 8 là tầng ẩn Dense
- Tầng cuối Dense là lớp đầu ra

Ở đây có tầng Dropout là dùng để tránh trường hợp Overfitting

Sau khi có được các tầng mạng thì bước tiếp theo sẽ huấn luyện

4. Training model

```
#train model
model.compile(optimizer='adam',
              loss=tf.keras.losses.CategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])#train the model
```

Hàm tối ưu hóa optimizer là Adam, hàm mất mát là CategoricalCrossentropy sử dụng cho phân nhiều lớp, và ma trận là độ chính xác

```
model.fit(X_train, Y_train, epochs=50)# evaluate the model
```

Huấn luyện với epoch 50

5. Sau khi đạt được độ chính xác cao thì lưu lại mô hình

```
filename = 'modelkeras_conv.h5'
keras.models.save_model(model, filename)
```

Lưu với đuôi là h5.