NeuralAssignment-7.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share M

+ Code + Text

RAM Disk

```
# Simple CNN model for CIFAR-10
import numpy
from keras.datasets import cifar10
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout
from keras.layers import Flatten
#from keras.constraints import maxnorm
from keras.optimizers import SGD
from keras.layers import Conv2D
from keras.layers import MaxPooling2D
from keras.utils import to_categorical
#from keras import backend as K
#K.set_image_dim_ordering('th')

# fix random seed for reproducibility
seed = 7
numpy.random.seed(seed)
# load data
(X_train, y_train), (X_test, y_test) = cifar10.load_data()
# normalize inputs from 0-255 to 0.0-1.0
X_train = X_train.astype('float32')
X_test = X_test.astype('float32')
X_train = X_train / 255.0
X_test = X_test / 255.0
# one hot encode outputs
y_train =to_categorical(y_train)
y_test =to_categorical(y_test)
num_classes = y_test.shape[1]
# Create the model
model = Sequential()
model.add(Conv2D(32, (3, 3), input_shape=(32, 32, 3), padding='same', activation='relu'))
model.add(Dropout(0.2))
model.add(Conv2D(128, (3, 3), activation='relu', padding='same'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(flatten())
model.add(Dense(1024, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(num_classes, activation='softmax'))
# Compile model
epochs = 5
lrate = 0.01
decay = lrate/epochs
```

✓ 1s completed at 7:38 PM

NeuralAssignment-7.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share M

+ Code + Text

RAM Disk

```
decay = lrate/epochs
sgd = SGD(lr=lrate)
model.compile(loss='categorical_crossentropy', optimizer=sgd, metrics=['accuracy'])
print(model.summary())
# Fit the model
history = model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=epochs, batch_size=32)
# Final evaluation of the model
scores = model.evaluate(X_test, y_test, verbose=0)
print("Accuracy: %.2f%%" % (scores[1]*100))
```

```
Downloading data from https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz
170498071/170498071 [==============================] - 2s 0us/step
WARNING:absl:`lr` is deprecated in Keras optimizer, please use `learning_rate` or use the legacy optimizer, e.g.,tf.keras.optimizers.legacy.SGD.
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 32, 32, 32)        896

 dropout (Dropout)           (None, 32, 32, 32)        0

 conv2d_1 (Conv2D)           (None, 32, 32, 128)       36992

 max_pooling2d (MaxPooling2  (None, 16, 16, 128)       0
 D)

 flatten (Flatten)           (None, 32768)             0

 dense (Dense)               (None, 1024)              33555456

 dropout_1 (Dropout)         (None, 1024)              0

 dense_1 (Dense)             (None, 10)                10250

=================================================================
Total params: 33603594 (128.19 MB)
Trainable params: 33603594 (128.19 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
None
Epoch 1/5
1563/1563 [==============================] - 615s 389ms/step - loss: 1.8292 - accuracy: 0.3402 - val_loss: 1.6758 - val_accuracy: 0.4167
Epoch 2/5
1563/1563 [==============================] - 611s 391ms/step - loss: 1.5449 - accuracy: 0.4543 - val_loss: 1.4517 - val_accuracy: 0.4877
Epoch 3/5
1563/1563 [==============================] - 614s 393ms/step - loss: 1.3696 - accuracy: 0.5135 - val_loss: 1.2700 - val_accuracy: 0.5488
```

✓ 1s completed at 7:38 PM

NeuralAssignment-7.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share M

+ Code + Text

```
1563/1563 [==============================] - 618s 395ms/step - loss: 1.2337 - accuracy: 0.5643 - val_loss: 1.1732 - val_accuracy: 0.5884
Epoch 5/5
1563/1563 [==============================] - 614s 393ms/step - loss: 1.1342 - accuracy: 0.6000 - val_loss: 1.1265 - val_accuracy: 0.6034
Accuracy: 60.34%
```

```
[2] import numpy as np
    predictions = model.predict(X_test[:4])
    predicted_labels = np.argmax(predictions, axis=1)
    actual_labels = np.argmax(y_test[:4], axis=1)

    print("Predicted labels:", predicted_labels)
    print("Actual labels:   ", actual_labels)
```

```
1/1 [==============================] - 0s 141ms/step
Predicted labels: [3 1 8 0]
Actual labels:    [3 8 8 0]
```

```
[3] import matplotlib.pyplot as plt

    plt.plot(history.history['loss'])
    plt.plot(history.history['val_loss'])
    plt.title('Model Loss')
    plt.ylabel('Loss')
    plt.xlabel('Epoch')
    plt.legend(['train', 'val'], loc='upper right')
    plt.show()
    plt.plot(history.history['accuracy'])
    plt.plot(history.history['val_accuracy'])
    plt.title('Model Accuracy')
    plt.ylabel('Accuracy')
    plt.xlabel('Epoch')
    plt.legend(['train', 'val'], loc='lower right')
    plt.show()
```

1s completed at 7:38 PM

NeuralAssignment-7.ipynb

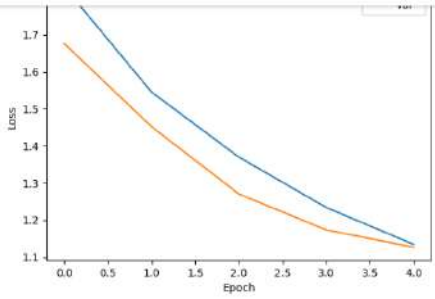
File Edit View Insert Runtime Tools Help All changes saved

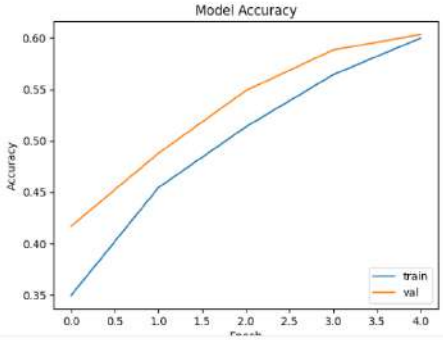
Comment Share M

+ Code + Text

RAM Disk

[3]





1s completed at 7:38 PM