Model with epoch 50

In [1]:
```
import pickle
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from keras.models import Sequential
from keras.layers import Conv2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten, BatchNormalization
from keras.layers import Dense, Dropout
from keras import regularizers
from keras.optimizers import SGD
from keras.preprocessing.image import ImageDataGenerator
from keras.utils import np_utils
import keras


def load_train_data(n):
    with open('data_batch_'+ str(n), 'rb') as file:
        batch = pickle.load(file, encoding='latin1')

    features = batch['data']
    Target = batch['labels']
    return features, Target


batch_1, Target_1 = load_train_data(1)
batch_2, Target_2 = load_train_data(2)
batch_3, Target_3 = load_train_data(3)
batch_4, Target_4 = load_train_data(4)
batch_5, Target_5 = load_train_data(5)


with open('test_batch', 'rb') as file:
    batch = pickle.load(file, encoding='latin1')
X_test = batch['data']
y_test = batch['labels']


X_train = np.append(batch_1, batch_2,axis=0)
X_train = np.append(X_train, batch_3,axis=0)
X_train = np.append(X_train, batch_4,axis=0)
X_train = np.append(X_train, batch_5,axis=0)
y_train = np.append(Target_1, Target_2,axis=0)
y_train = np.append(y_train, Target_3,axis=0)
y_train = np.append(y_train, Target_4,axis=0)
y_train = np.append(y_train, Target_5,axis=0)
X_train = X_train.reshape((len(X_train), 3, 32, 32)).transpose(0,2,3,1)
y_train = np_utils.to_categorical(y_train, 10)
X_test = X_test.reshape((len(X_test), 3, 32, 32)).transpose(0,2,3,1)
y_test = np_utils.to_categorical(y_test, 10)
X_train = X_train.astype('float32')
X_test= X_test.astype('float32')
X_train= X_train / 255.0
X_test= X_test/ 255.0
```

```
Using TensorFlow backend.
```

In [4]:
```
model19 = Sequential()
model19.add(Conv2D(64, (3, 3), activation='relu',kernel_initializer='he_normal',padding = 'same', input_shape=(32, 32, 3)))
model19.add(Conv2D(64, (3, 3), activation='relu',kernel_initializer='he_normal',padding = 'same'))
model19.add(MaxPooling2D((2, 2)))
model19.add(Conv2D(64, (3, 3), activation='relu',kernel_initializer='he_normal',padding = 'same'))
model19.add(Conv2D(64, (3, 3), activation='relu',kernel_initializer='he_normal',padding = 'same'))
model19.add(MaxPooling2D((2, 2)))
model19.add(Conv2D(64, (3, 3), activation='relu',kernel_initializer='he_normal',padding = 'same'))
model19.add(MaxPooling2D((2, 2)))
model19.add(Conv2D(64, (3, 3), activation='relu',kernel_initializer='he_normal',padding = 'same'))
model19.add(MaxPooling2D((2, 2)))
model19.add(Flatten())
model19.add(Dense(128, activation='relu'))
model19.add(Dropout(rate = 0.7))
model19.add(Dense(10, activation='softmax'))
model19.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_15 (Conv2D)           (None, 32, 32, 64)        1792
_________________________________________________________________
conv2d_16 (Conv2D)           (None, 32, 32, 64)        36928
_________________________________________________________________
max_pooling2d_11 (MaxPooling (None, 16, 16, 64)        0
_________________________________________________________________
conv2d_17 (Conv2D)           (None, 16, 16, 64)        36928
_________________________________________________________________
conv2d_18 (Conv2D)           (None, 16, 16, 64)        36928
_________________________________________________________________
max_pooling2d_12 (MaxPooling (None, 8, 8, 64)          0
_________________________________________________________________
conv2d_19 (Conv2D)           (None, 8, 8, 64)          36928
_________________________________________________________________
max_pooling2d_13 (MaxPooling (None, 4, 4, 64)          0
_________________________________________________________________
conv2d_20 (Conv2D)           (None, 4, 4, 64)          36928
_________________________________________________________________
max_pooling2d_14 (MaxPooling (None, 2, 2, 64)          0
_________________________________________________________________
conv2d_21 (Conv2D)           (None, 2, 2, 64)          36928
_________________________________________________________________
max_pooling2d_15 (MaxPooling (None, 1, 1, 64)          0
_________________________________________________________________
flatten_3 (Flatten)          (None, 64)                0
_________________________________________________________________
dense_5 (Dense)              (None, 128)               8320
_________________________________________________________________
dropout_3 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_6 (Dense)              (None, 10)                1290
=================================================================
Total params: 232,970
Trainable params: 232,970
Non-trainable params: 0
_________________________________________________________________
```

In [5]:
```
epochs = 50
model19.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model19.fit(X_train,y_train,epochs=epochs,batch_size = 32)
```

```
WARNING:tensorflow:From C:\Users\Dhanajayan\Anaconda3\lib\site-packages\tensorflow\python\ops\math_ops.py:3066: to_int32 (from tensorflow.python.ops.math_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.cast instead.
Epoch 1/50
50000/50000 [==============================] - 487s 10ms/step - loss: 1.7711 - acc: 0.3381
Epoch 2/50
50000/50000 [==============================] - 528s 11ms/step - loss: 1.2558 - acc: 0.5572
Epoch 3/50
50000/50000 [==============================] - 533s 11ms/step - loss: 1.0379 - acc: 0.6443
Epoch 4/50
50000/50000 [==============================] - 525s 10ms/step - loss: 0.9026 - acc: 0.6924
Epoch 5/50
50000/50000 [==============================] - 510s 10ms/step - loss: 0.8114 - acc: 0.7286
Epoch 6/50
50000/50000 [==============================] - 514s 10ms/step - loss: 0.7426 - acc: 0.7535
Epoch 7/50
50000/50000 [==============================] - 504s 10ms/step - loss: 0.6898 - acc: 0.7702
Epoch 8/50
50000/50000 [==============================] - 451s 9ms/step - loss: 0.6388 - acc: 0.7888
Epoch 9/50
50000/50000 [==============================] - 446s 9ms/step - loss: 0.5992 - acc: 0.8021
Epoch 10/50
50000/50000 [==============================] - 448s 9ms/step - loss: 0.5674 - acc: 0.8124
Epoch 11/50
50000/50000 [==============================] - 442s 9ms/step - loss: 0.5353 - acc: 0.8244
Epoch 12/50
50000/50000 [==============================] - 440s 9ms/step - loss: 0.5083 - acc: 0.8347
Epoch 13/50
50000/50000 [==============================] - 366s 7ms/step - loss: 0.4904 - acc: 0.8400
Epoch 14/50
50000/50000 [==============================] - 371s 7ms/step - loss: 0.4711 - acc: 0.8477
Epoch 15/50
50000/50000 [==============================] - 401s 8ms/step - loss: 0.4487 - acc: 0.8546
Epoch 16/50
50000/50000 [==============================] - 431s 9ms/step - loss: 0.4299 - acc: 0.8597
Epoch 17/50
50000/50000 [==============================] - 398s 8ms/step - loss: 0.4163 - acc: 0.8654
Epoch 18/50
50000/50000 [==============================] - 408s 8ms/step - loss: 0.3986 - acc: 0.8698
Epoch 19/50
50000/50000 [==============================] - 375s 8ms/step - loss: 0.3987 - acc: 0.8715
Epoch 20/50
50000/50000 [==============================] - 387s 8ms/step - loss: 0.3851 - acc: 0.8757
Epoch 21/50
50000/50000 [==============================] - 392s 8ms/step - loss: 0.3668 - acc: 0.8824
Epoch 22/50
50000/50000 [==============================] - 379s 8ms/step - loss: 0.3556 - acc: 0.8859
Epoch 23/50
50000/50000 [==============================] - 360s 7ms/step - loss: 0.3514 - acc: 0.8867
Epoch 24/50
50000/50000 [==============================] - 370s 7ms/step - loss: 0.3452 - acc: 0.8889
Epoch 25/50
50000/50000 [==============================] - 369s 7ms/step - loss: 0.3501 - acc: 0.8872
Epoch 26/50
50000/50000 [==============================] - 346s 7ms/step - loss: 0.3260 - acc: 0.8944
Epoch 27/50
50000/50000 [==============================] - 338s 7ms/step - loss: 0.3301 - acc: 0.8953
Epoch 28/50
50000/50000 [==============================] - 372s 7ms/step - loss: 0.3200 - acc: 0.8990
Epoch 29/50
50000/50000 [==============================] - 388s 8ms/step - loss: 0.3328 - acc: 0.8961
Epoch 30/50
50000/50000 [==============================] - 395s 8ms/step - loss: 0.3244 - acc: 0.8999
Epoch 31/50
50000/50000 [==============================] - 339s 7ms/step - loss: 0.3117 - acc: 0.9039 ls - l
oss: 0.3111 - ac
Epoch 32/50
50000/50000 [==============================] - 381s 8ms/step - loss: 0.3102 - acc: 0.9036
Epoch 33/50
50000/50000 [==============================] - 356s 7ms/step - loss: 0.3171 - acc: 0.9021
Epoch 34/50
50000/50000 [==============================] - 397s 8ms/step - loss: 0.2923 - acc: 0.9090
Epoch 35/50
50000/50000 [==============================] - 416s 8ms/step - loss: 0.2947 - acc: 0.9093
Epoch 36/50
50000/50000 [==============================] - 438s 9ms/step - loss: 0.3033 - acc: 0.9073
Epoch 37/50
50000/50000 [==============================] - 445s 9ms/step - loss: 0.3046 - acc: 0.9075
Epoch 38/50
50000/50000 [==============================] - 458s 9ms/step - loss: 0.2721 - acc: 0.9152
Epoch 39/50
50000/50000 [==============================] - 458s 9ms/step - loss: 0.2853 - acc: 0.9142
Epoch 40/50
50000/50000 [==============================] - 445s 9ms/step - loss: 0.2748 - acc: 0.9156
Epoch 41/50
50000/50000 [==============================] - 467s 9ms/step - loss: 0.2896 - acc: 0.9121
Epoch 42/50
50000/50000 [==============================] - 467s 9ms/step - loss: 0.2819 - acc: 0.9156
Epoch 43/50
50000/50000 [==============================] - 467s 9ms/step - loss: 0.2972 - acc: 0.9114
Epoch 44/50
50000/50000 [==============================] - 468s 9ms/step - loss: 0.2739 - acc: 0.9171
Epoch 45/50
50000/50000 [==============================] - 468s 9ms/step - loss: 0.2697 - acc: 0.9208
Epoch 46/50
50000/50000 [==============================] - 468s 9ms/step - loss: 0.2841 - acc: 0.9160
Epoch 47/50
50000/50000 [==============================] - 468s 9ms/step - loss: 0.2674 - acc: 0.9201
Epoch 48/50
50000/50000 [==============================] - 464s 9ms/step - loss: 0.2898 - acc: 0.9144
Epoch 49/50
50000/50000 [==============================] - 462s 9ms/step - loss: 0.2810 - acc: 0.9186
Epoch 50/50
50000/50000 [==============================] - 453s 9ms/step - loss: 0.2733 - acc: 0.9192
```

Out[5]:
```
<keras.callbacks.History at 0x2a226c61240>
```

In [8]:
```
test_loss,test_acc = model19.evaluate(X_test,y_test)
test_acc
```

```
10000/10000 [==============================] - 32s 3ms/step
```

Out[8]:
```
0.7638
```

Observation

The model overfits the data with 50 epochs

Conclusion

1. In the image classification model the following layers are used.

- Conv2D layer

  ```
  Keras Conv2D is a 2D Convolution Layer, this layer creates a convolution kernel
  that is wind with layers input which helps produce a tensor of outputs.


  Kernel: In image processing kernel is a convolution matrix or masks which can be
  used for blurring, sharpening, embossing, edge detection and more by doing a
  convolution between a kernel and an image.
  ```

- Maxpooling2D

  ```
  Calculate the maximum value for each patch of the feature map.
  ```

- Flatten

  ```
  Flatten layer which prepares a vector for the fully connected layers or dense
  layer.
  ```

- Dropout layer

  ```
  Helps to reduce the overfitting.
  ```

- Dense layer

  ```
  A dense layer represents a matrix vector multiplication.The values in the matrix
  are the trainable parameters which get updated during backpropagation.
  ```

In this models cross validation, data agumentation,batch normalization and hyperparameter tuning using sklearn are not used.

Manually checked how parameters are affecting the accuracy.

In deeplearning models there are many parameters to tune .so monitoring every parameter is mandatory.

To fix the following issues

1. Computation speed
2. overfitting and underfitting

Need to understand the behaviour of the addition of the hidden layers and its parameters.