REPORT

CECS 551 – Assignment 7

Design convolutional neural networks to classify cifar-10 images using keras library.

Name: Aishwarya Bhavsar

CSULB ID: 029371509

Date – 23trd March, 2022

I certify that this submission is my original work – AVB

- **GOAL:** **To** Evaluate multiple combinations of activation function, optimizer, hyper-parameter, generalization (including data augmentation), and architecture
- **STEPS/PROCEDURE:**

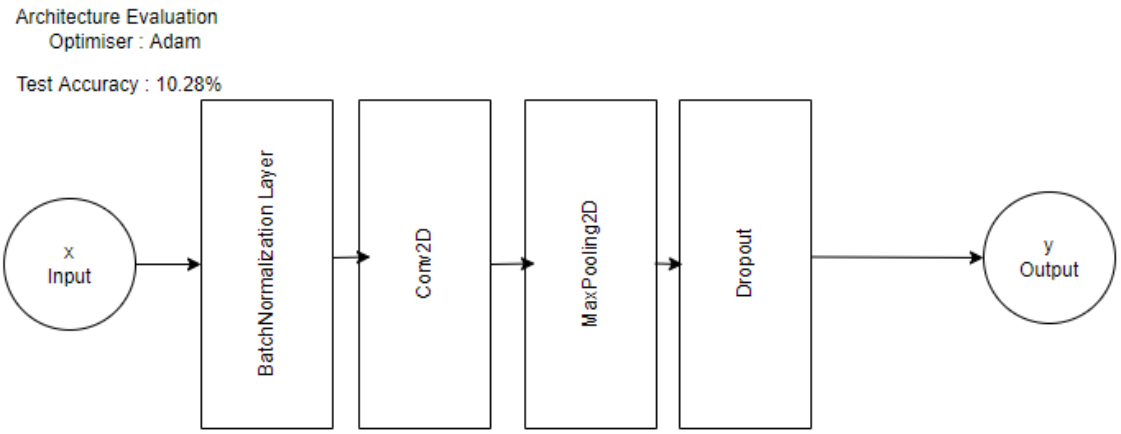
    1. Train the Model
    2. Compile The Model
    3. Fit The Model
    4. Evaluate The Model
    5. Determine The Accuracy.

- **RESULTS:**
- **Architecture Evaluation With Lowest Accuracy:**

The model consists of :

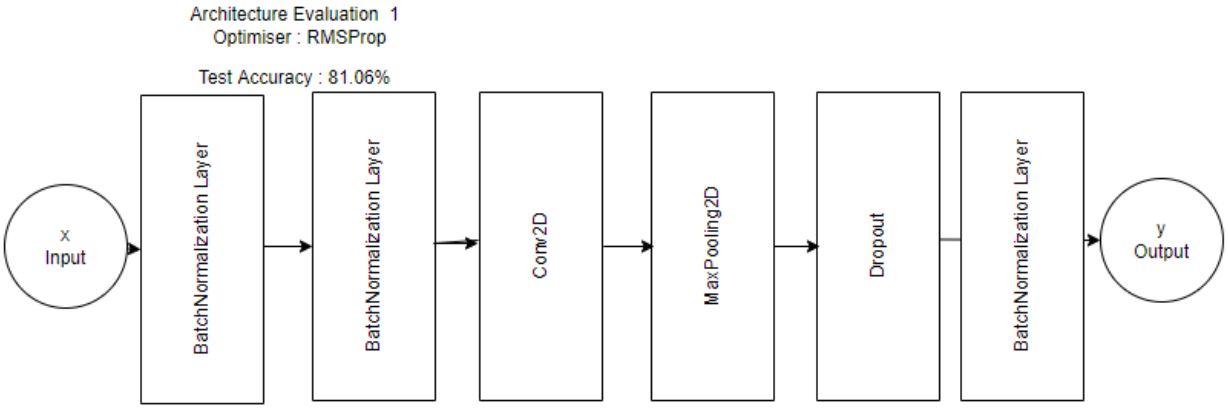
Batch Normalization Layer, Conv2D Layer, MaxPooling2D Layer and Dropout Layer.

Each time we change the parameters, the accuracy changes.



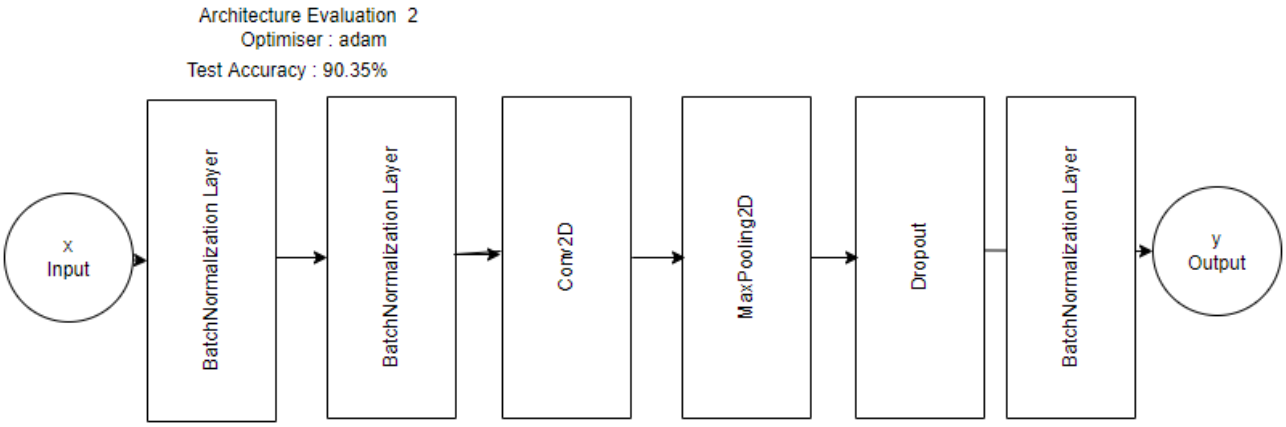
- The proposed model is made of four MLP layers.
- In Keras, an MLP layer is referred to as Dense, which stands for the densely connected layer.
- The hidden layer contains 256 units, followed by Exponential Linear Unit (elu) activation and dropout.
- The Conv2D layer consists of 64, 128, 256 units in every iteration.
- The dropout is kept constant at 0.25, it means 25% of the neurons will be deactivated.
- The main data structure in Keras is the Sequential class, which allows the creation of a basic neural network.
- The Sequential class of the Keras library is a wrapper for the sequential neural network model that Keras offers and can be created in the following way:
- from keras.models import Sequential
- model = Sequential()
- The model in Keras is considered as a sequence of layers and each of them gradually “distills” the input data to obtain the desired output.
- In Keras, we can add the required types of layers through the add() method.
- The last network we have added consists of Flatten, dense layers followed by elu and softmax activation and dropout=0.5.
- optimizer='adam',
- loss='sparse_categorical_crossentropy',
- metrics='accuracy'
- The accuracy is significantly low when we tune the parameters in the above-mentioned way. However, it can be increased by using the correct optimisers and activation functions as shown below.

1. Architecture Evaluation 1



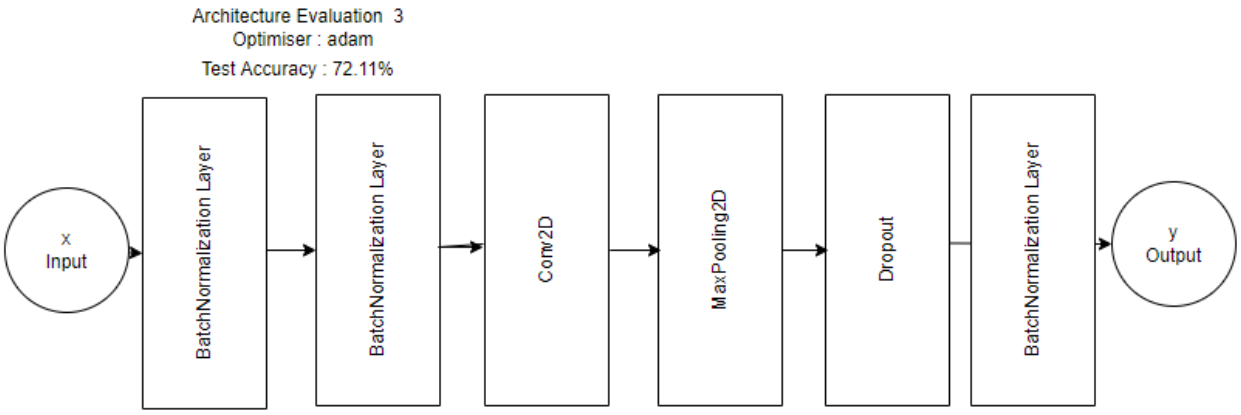
- The proposed model is made of six MLP layers.
- In Keras, an MLP layer is referred to as Dense, which stands for the densely connected layer.
- The hidden layer contains 256 units each, followed by relu and softmax activation and dropout.
- The Conv2D layer consists of 32,64, 128 units in every iteration.
- The dropout is kept constant at 0.25, it means 25% of the neurons will be deactivated.
- The main data structure in Keras is the Sequential class, which allows the creation of a basic neural network.
- The Sequential class of the Keras library is a wrapper for the sequential neural network model that Keras offers and can be created in the following way:
- from keras.models import Sequential
- model = Sequential()
- The model in Keras is considered as a sequence of layers and each of them gradually “distills” the input data to obtain the desired output.
- In Keras, we can add the required types of layers through the add() method.
- The last network we have added consists of Flatten, dense layers followed by relu and softmax activation and dropout=0.5.
- optimizer = 'rmsprop',
- loss = 'categorical_crossentropy',
- metrics = 'accuracy'

2. Architecture Evaluation 2



- The proposed model is made of six MLP layers.
- In Keras, an MLP layer is referred to as Dense, which stands for the densely connected layer.
- The hidden layer contains 256 units each, followed by relu, **sigmoid** activation and dropout.
- The main data structure in Keras is the Sequential class, which allows the creation of a basic neural network.
- The Sequential class of the Keras library is a wrapper for the sequential neural network model that Keras offers and can be created in the following way:
- from keras.models import Sequential
- model = Sequential()
- The model in Keras is considered as a sequence of layers and each of them gradually “distills” the input data to obtain the desired output.
- In Keras, we can add the required types of layers through the add() method.
- optimizer = 'adam',
- loss = 'categorical_crossentropy',
- metrics = 'accuracy'

3. Architecture Evaluation 3



- The proposed model is made of six MLP layers.
- The hidden layer contains 256 units each, followed by **sigmoid, sigmoid and tanh** activation and dropout.
- The main data structure in Keras is the Sequential class, which allows the creation of a basic neural network.
- The conv2D layers consists of 32, 64, 128 hidden units respectively.
- The Sequential class of the Keras library is a wrapper for the sequential neural network model that Keras offers and can be created in the following way:
- from keras.models import Sequential
- model = Sequential()
- The model in Keras is considered as a sequence of layers and each of them gradually “distills” the input data to obtain the desired output.
- In Keras, we can add the required types of layers through the add() method.
- optimizer = 'adam',
- loss = 'categorical_crossentropy',
- metrics = 'accuracy'

- **CONCLUSION:**
  - **Data Augmentation:** I have performed data augmentation on all the three architectures, thus I'm able to see that data augmentation reduces the risk of overfitting and enhances the accuracy.
  - Amongst the three architectures, **adam optimiser** gives **best accuracy** when combined with **relu and sigmoid** activation functions.
  - But the same optimiser **(adam)** gives significantly **less accuracy** when combined with **tanh and sigmoid** activation functions.
  - The accuracy of any model depends on the activation function and not on the number of layers.
  - Therefore, choosing correct optimisers and activation functions is important and will yield fruitful results.
  - The **average Test Accuracy** of all the three models is:

    (81.06% + 90.35% + 72.11%)/3 = **81.17%**