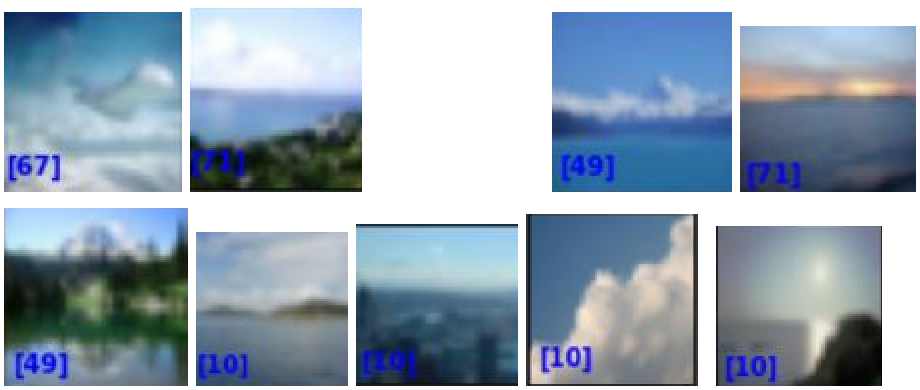
Assignment 1 – Practical Deep Learning Workshop
Amit Levizky 203911813, Alex Abramov 314129438

1. We chose Cifar-100
   a. Data size: Training: 50,000. Testing: 10,000.
   b. Samples' data:
      i. Dimensions: 32X32.
      ii. Channel: rgb.
      iii. Classes: 100.
      iv. Preprocess: By using the keras.cifar100 library, we get the samples ready to use. All we had to do is one hot encoding using "to_categorical".
      v. Augmentation: flip horizontal, width shifting, height shifting, rotation by 10 degree.
   c. The data is balanced. 500 training images and 100 testing images per class.
   d. Yes. Link: https://benchmarks.ai/cifar-100
   e. Harder to separate:

      Classes within the large natural outdoor scenes:

         cloud, forest, mountain, plain, sea



   Easier to separate:

      Samples from the vehicles, inspects and household devices classes.

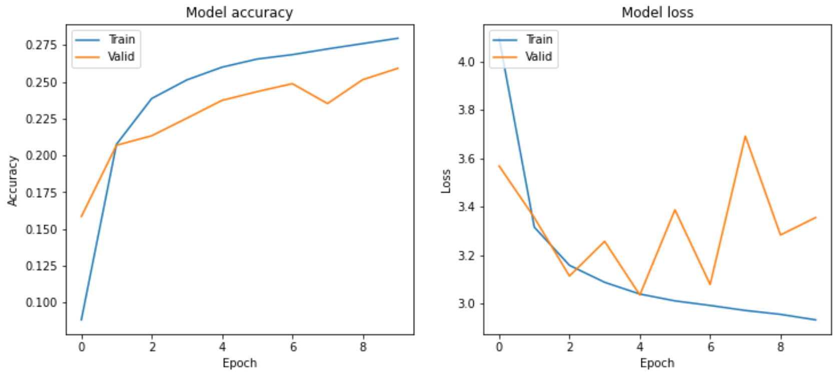
Assignment 1 – Practical Deep Learning Workshop

Amit Levizky 203911813, Alex Abramov 314129438

2.

a. We chose the validation split method with 20% since we have 50,000 (before augmentation) training samples.

b. After 10 epochs, we got a 27.88% test prediction.

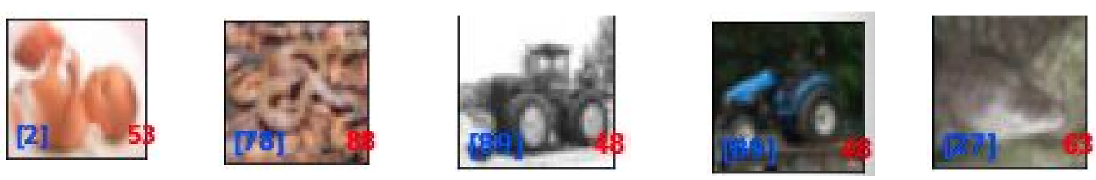


As we can see from the training, validation and test accuracy, they all have similar predictions percentages, about 25-28%.

Some good and bad predictions:

Good classifications:

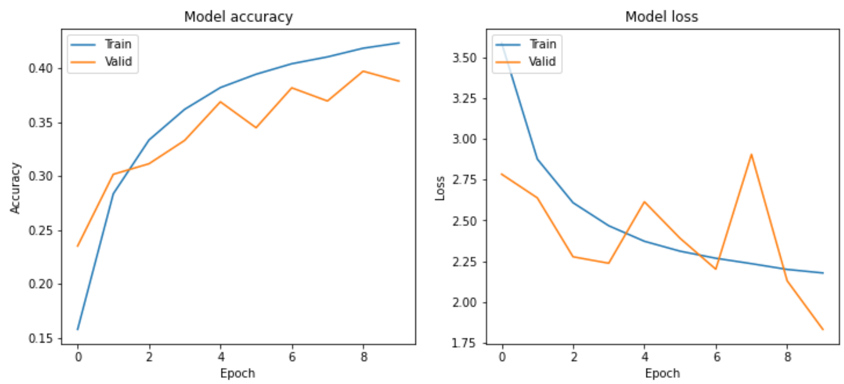
Bad classifications:



When looking at some predictions results, we can see that predicting unclear pictures of animals and tractors is a harder job for our model compared to predicting clear images of animals which is a category that has a little bit better predictions.

c. As we can see from the results above, we are in an under-fitting position.
Our suggestions to improve the model results are (sorted by priority):
- Increasing the number of neurons of layers in the model, and by that adding complexity to the model.
- Increasing the number of layers in the model.
- Changing the type of layers we are using.

d. We have added complexity to our model by increasing the number of neurons and adding more layers. As a result, we managed to increase the test prediction by 11%, and got 38.57%. Loss got lower and accuracy has been improved. When comparing the training, validation and testing, we see that prediction rates are pretty similar.



Some good and bad predictions:

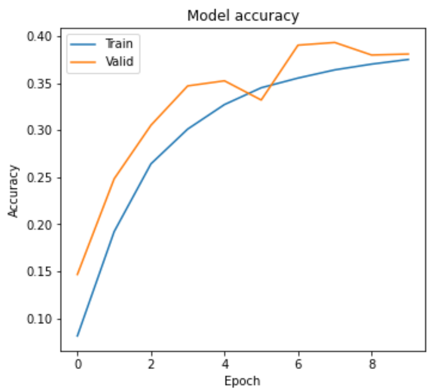
Good classifications:

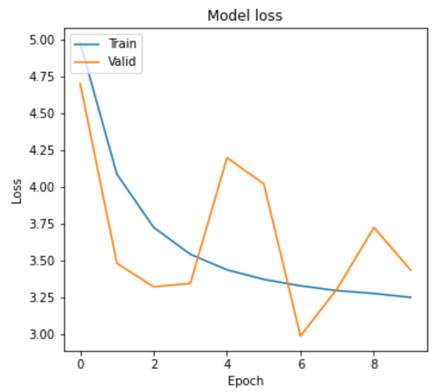
Bad classifications:

After we have improved our model, we can see that it is able to predict unclear pictures of animals better than it could before, however, the tractors remain hard to predict for our model.

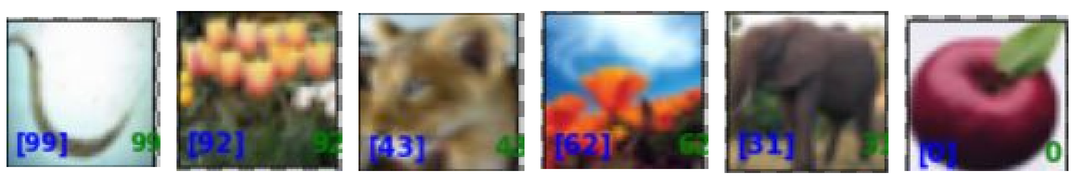
Assignment 1 – Practical Deep Learning Workshop
Amit Levizky 203911813, Alex Abramov 314129438

3.
  a. Done in code
  b. Done in code
  c. After running 10 epochs using VGG16, we were able to see improvements from the original model. The test accuracy was 40.91%, which is similar to our predictions accuracy in the training phase.

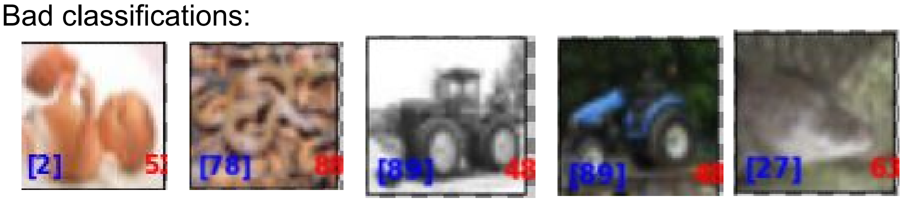




Some good and bad predictions:
Good classifications:



Bad classifications:



  d. Using our trained original model, with omitting the last layer, we used a ML algorithm to try to increase our predictions percentage. After prediting with this new model, we got a prediction rate of 30.14% on the test set, compared to 27.88% we had before this improving trial. We can say that this change did really make a little better result on predicting.

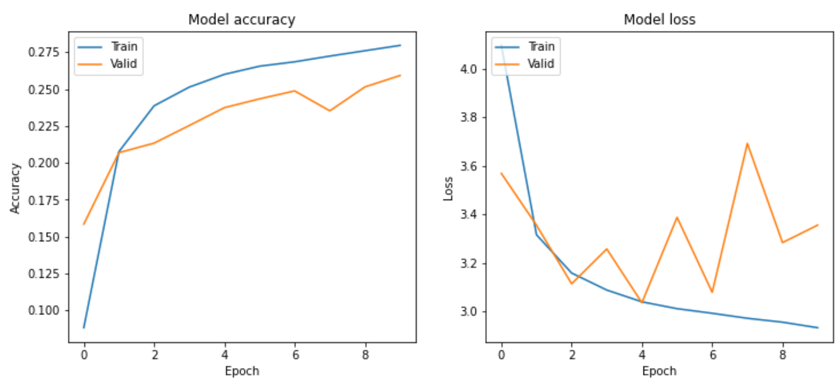
Assignment 1 – Practical Deep Learning Workshop
Amit Levizky 203911813, Alex Abramov 314129438

4. We chose to do our Deep Learning classification research on the Cifar100 dataset. This dataset has 100 different classes, each with 600 images. There are 500 training images and 100 testing images per class.

   In the first stage we examined the type of data we want to make a prediction on. With the help of the keras.cifar100 library, we have been given a ready-to-use training set and test set. In addition, in order to increase the amount of data and reduce the chance of overfitting, we augmented all the images with flip, shifting (width and height) and rotation of 10 degrees.

   In the second stage, we built a neuron network for the purpose of performing the desired classification. We chose to split the training data into an 80 percent of training set and 20 percent of validation set. With this division, we can know at the end of the model training whether we have overfitting or underfitting problems by observing the outcoming metrics.

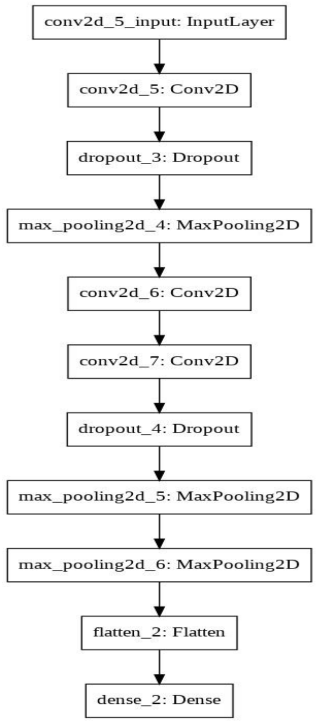
   Based on many image classification problems, we chose to focus on convolutional layers to achieve high accuracy and low loss as short as possible.After running our CNN with 10 epochs, we got both low accuracy and high loss. As expected, when we tried to predict the test dataset, We got 27.88% of correct classifications. That means we were in an underfitting situation.



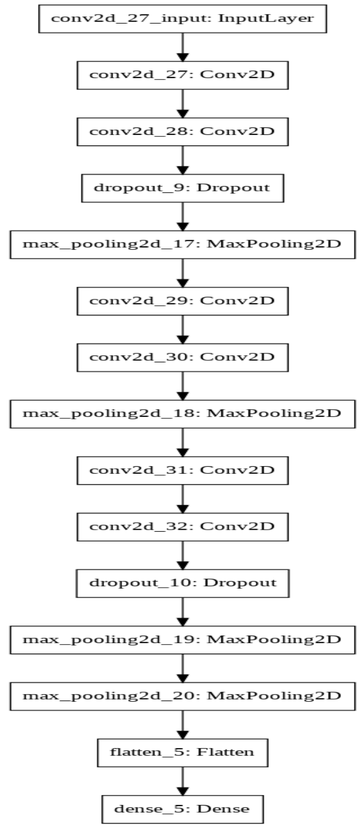
   Following the results, we decided to try to address the problem’s solution with the addition of complexity. We added new layers and added more neurons to existing ones.

Assignment 1 – Practical Deep Learning Workshop
Amit Levizky 203911813, Alex Abramov 314129438

Original model:

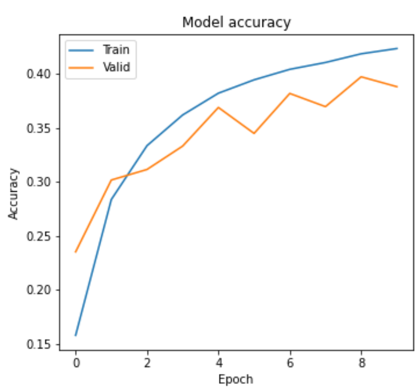


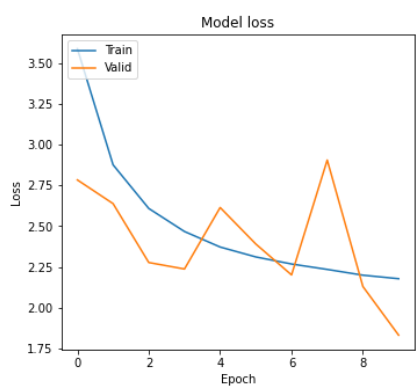
Improved model:



As we can see, there are additional 3 new layers and we increased the number of neurons for each layer as well.

After training the improved model, much like the previous model, with 10 epochs. This time, as expected, we saw that accuracy has been increased and loss has been decreased. In addition, and perhaps more importantly, we obtained prediction accuracy on test data of 38.57%.

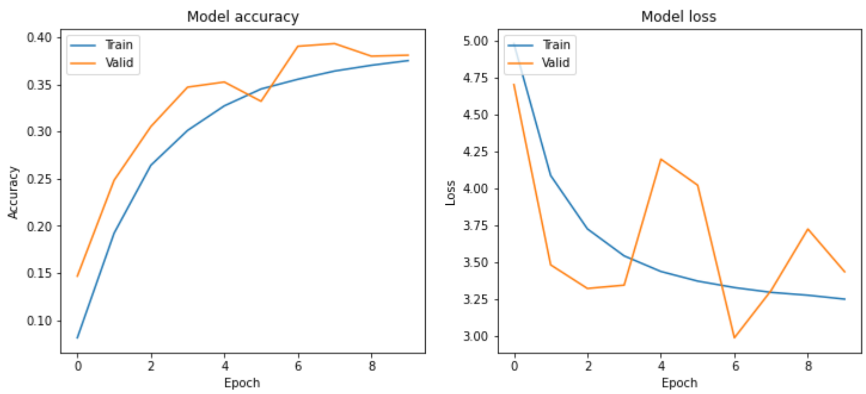




Assignment 1 – Practical Deep Learning Workshop
Amit Levizky 203911813, Alex Abramov 314129438

In the third part of the assignment, we selected a network that is already trained and existing from the Keras Library. We chose the VGG16 network. This network has high complexity but it is known to be very good and with proven results.

For comparison, we chose to train the model for 10 epochs also and this time we also saw an improvement in accuracy and loss. In addition, we saw a relative improvement on the predictions of the test set with about 40.91% success.



In the fourth and final phase, we used our original model as a feature extractor for a familiar machine learning model and examined the results and whether the results of the test on the model improved. The process we did is training the original model on the train set, then we pulled the last layer from the model. Next, we performed a prediction for the training set that we will use to train the new model, and a prediction on the test set that will be used for us as the input for the new model. We have connected to the new model (without the last layer), a machine learning algorithm based model, *Logistic Regression*. By that, after running the described model, we could see that we managed to increase the prediction rate in 3%, to a total of 30.14% succession rate.