Bio - Intelligent Algorithms - Exercise 2

Table of Contents

[Names 2](#_Toc78546653)

[Validation Score 3](#_Toc78546654)

[Note 3](#_Toc78546655)

[Scores 3](#_Toc78546656)

[GPU Accelerated (CuPy) 3](#_Toc78546657)

[Research and experiments 5](#_Toc78546658)

[Model architecture 5](#_Toc78546659)

[Hidden Layers 5](#_Toc78546660)

[Number of filters in the convolutional layers 5](#_Toc78546661)

[Number of neurons in the fully connected layers 5](#_Toc78546662)

[Output layer 5](#_Toc78546663)

[Activation function 5](#_Toc78546664)

[Optimizer 5](#_Toc78546665)

[Regularization 6](#_Toc78546666)

[L2 regularization 6](#_Toc78546667)

[Dropout 6](#_Toc78546668)

[Gaussian distribution initialization 6](#_Toc78546669)

[Augmentation 6](#_Toc78546670)

[Early stop 7](#_Toc78546671)

[Transfer Learning 7](#_Toc78546672)

[Experiments 7](#_Toc78546673)

[Experiment 1 7](#_Toc78546674)

[Experiment 2 8](#_Toc78546675)

[Experiment 3 9](#_Toc78546676)

[Experiment 4 10](#_Toc78546677)

[Experiment 5 11](#_Toc78546678)

[Experiment 6 12](#_Toc78546679)

[Experiment 7 14](#_Toc78546680)

[Experiment 8 15](#_Toc78546681)

[Experiment 9 16](#_Toc78546682)

[Experiment 10 17](#_Toc78546683)

[Experiment 11 18](#_Toc78546684)

[Experiment 12 19](#_Toc78546685)

[Experiment 13 20](#_Toc78546686)

[Experiment 14 21](#_Toc78546687)

[Experiment 15 – 22 – Optimizers 22](#_Toc78546688)

[Experiment 23 – Our model **without** transfer learning 23](#_Toc78546689)

[Experiment 24 – Our model **with** transfer learning – **Final submitted model** 25](#_Toc78546690)

# Names

* Simon Raviv, 312847478.
* Eliyahu Zilberbrand, 317937100.

# Validation Score

## Note

The predictions file supplied in this exercise, was predicted with the pre-trained **NumPy model**. Predicting with the CuPy model, will give slightly different prediction due to the difference in random state between the libraries.

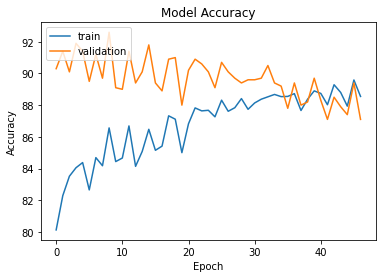
## Scores

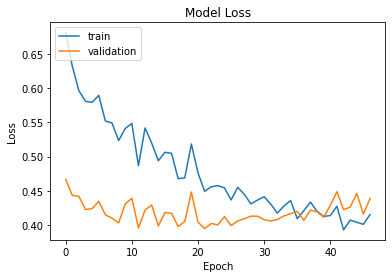
We will supply the results for the validation set only the CuPy module.

See notes regarding both accelerated implementations in *instructions\_highlits\_and\_more.docx* document. We present both with and without transfer learning result, but the submitted model is with transfer learning, see explanation later in the report.

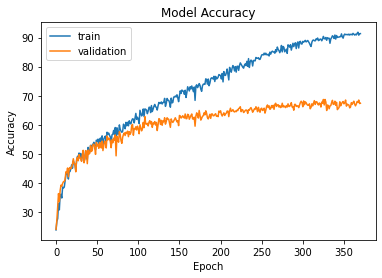
### GPU Accelerated (CuPy)

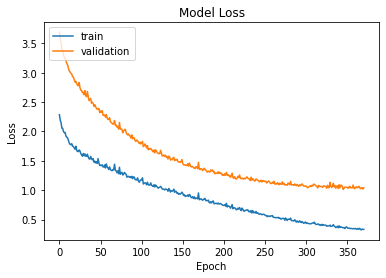
* Total validation result **with** transfer learning – **our final submitted model**:
  + **90.9 [%]**



****

* Total validation result **without** transfer learning – Just for reference:
  + **67.9 [%]**





# Research and experiments

The details below describe the different model decisions. After the details section, there are different experiments we did.

## Model architecture

### Hidden Layers

We got a small dataset; thus, we have chosen to use the following architecture in the middle of the network: 2D convolutional layer 🡪 2D max pooling layer 🡪 2D convolutional layer 🡪 2D max pooling layer 🡪 Fully connected layer 🡪 Fully connected layer, and not more. We saw that when adding more hidden layers, the accuracy didn’t improve. This behavior seems expected to us, since when adding more layers, we add more learning parameters to optimize. When having small amount of data, we assume it makes harder to *learn and not to memorize*, which leads to overfitting on the training data.

### Number of filters in the convolutional layers

We experimented with different number of filters in both layers: 16/32, 32/64 and 32/96.

We have chosen to use 32 filters in the first convolutional layer and 64 filters in the second convolutional layer, based on our experiments.

### Number of neurons in the fully connected layers

We choose to reduce the number of neurons in this model compared to the previous model, since now we have the convolutional section that give us good features of the dataset, and it is not needed compared to the previous model that had to learn both features and classification.

We have chosen to use 512 neurons in the first fully connected layer and 128 neurons in the second fully connected layer, based on our experiments.

### Output layer

We have chosen to use *Softmax* activation function for the output layer, since we are using categorial multi class labeled data. Using this method, we are getting the probability for each class.

## Activation function

We have experimented with various activation functions: *Sigmoid, ReLU* and *Leaky ReLU.*

We have chosen to use Leaky ReLU, since it had the highest accuracy. We assume that using Leaky ReLU is more beneficial in this model, since now we have multiple hidden layers and dipper network compared to the previous model. Thus, *Leaky ReLU* might prevent vanishing negative gradients and solve *dying ReLU problem.*

## Optimizer

Aside from vanilla *SGD* optimizer, we have implemented several familiar optimizers: *momentum*, *adagrad, rmsprop* and *adam* - with mini batch implementation. The advantage of added optimizers is proportional update weights, between consecutive gradient updates. Using it gives and adaptive learning rate per each weight, unlike in *SGD*.

Each optimizer experimented with different amount of batch sizes.

We saw in our experiments that *adagrad, rmsprop* and *adam* optimizers speed up training in early epochs more than *SGD* and *momentum*, but in last epochs *SGD* and *momentum* optimizers help the model to converge better.

Therefore, we have chosen to use *momentum* optimizer with mini batch size of 50, since it had the highest final performance with better convergence duration than *SGD*.

See experiments 15-22.

## Regularization

We have experimented with two types of regularization methods: L2 regularization and dropout.

We did it to make the model not to overfit on the training data, but to generalize.

### L2 regularization

We see that when increasing the L2 regularization factor, the overfitting decreases as the factor increases. Up to some boundaries.

We choose to use regularization factor of 0.15, based on our experiments.

### Dropout

We see that when decreasing the number of active neurons, the overfitting decreases. Up to some lower limit.

We choose to use dropout factor of 0.2, based on our experiments.

## Gaussian distribution initialization

We have experimented with some numbers and found that since our data is between 0 to 1, good approach is to use expectation value of 0 and standard deviation value of 0.1, this is compared to 0.01 in the previous model.

## Augmentation

In the previous exercise, we didn’t know which type of data we are working with. In this exercise, we know the dataset represents natural images. Therefore, to get robust model and more genialized towards the real world, one that independent to direction and other changes, we augmented our samples.

We have implemented 6 different augmentations techniques, which augment part of the shuffled training data at each epoch. At the beginning of the epoch, we have chosen some percentage of the data to augment, we choose not to add to the dataset the augmented samples, but to do it online on the ones we have. This is done in order not to affect the data distribution.

Each sample chosen for augmentation used one of the augmentation techniques, according to random probability with uniform distribution.

The augmentations: Anti - Clockwise Rotation, Clockwise Rotation, Horizontal Flip, Vertical Flip, Random Noise and Blur.

We have chosen to do the augmentation on 30 % (0.3) of the data as described above.

See experiments 15-22.

## Early stop

We have implemented an additional part for our early stop algorithm, early stop criteria. We assume that it is better to stop on the prime validation loss, but not accuracy. This is since even we see higher validation loss; it doesn’t mean we will see the same on other test data. Using the loss gives us higher chance to be predict with higher accuracy on test data.

## Transfer Learning

In this exercise we know that the input pictures are natural real-world pictures. At the end of our research and experiments and due to the limited dataset, we have, we have decided to implement transfer learning technique. We have experiment with several techniques: Taking the first two convolutional layers, taking all layers, and taking all layers besides the last one.

We have trained our model on CIFAR10 dataset since it is also natural real-world pictures with 10 classes. It is done for simplicity; we could also use other datasets with more classes and adopt.

After training a model on CIFAR10 using our code, we got a trained model to do transfer learning from. We have experimented with several optimizers and chose to use the model with *rmsprop* optimizer. See experiments 23-24 for the results. We think we could get better results for CIFAR10 if we would have tuned our model for that dataset and its size, but didn’t do it due to time limitations, as is the results are good, even without modifications and tunings.

We have chosen to use the last technique, where taking all the layers, besides the last one, since we want to take all the features including the first classification layer, but let the model relearn the correct labels on our dataset on the last layer.

# Experiments

The experiments below done based on the details above and investigations we wanted to do. In each experiment, all the relevant model parameters are written. The parameters marked with bold font, are the parameters changed between consecutive experiments (or new parameters). In addition, the total validation accuracy and last epoch statistics written for additional visibility.

## Experiment 1

        \* data\_dim: 3072

        \* activation\_func: relu

        \* l1\_fc\_hidden\_size: 1024

        \* l2\_fc\_hidden\_size: 500

        \* num\_of\_classes: 10

        \* initial\_lr: 0.0

        \* lr: 0.01

        \* epochs: 300

        \* batch\_size: 65

        \* reg: 0.0

        \* input\_noise\_p: 0

        \* dropout\_p: 0.0

        \* early\_stop\_max\_epochs: 18

        \* init\_weights\_mu: 0.0

        \* init\_weights\_sigma: 0.1

        \* early\_stop\_drop: 0.5

        \* early\_stop\_epochs\_drop: 10

        \* input\_z\_score\_normalization: False

        \* input\_hight: 32

        \* input\_width: 32

        \* input\_dim: 32

        \* input\_num\_chanels: 3

        \* chanel\_len: 1024

        \* conv2d\_1\_filter\_dim: 3

        \* conv2d\_2\_filter\_dim: 3

        \* conv2d\_1\_stride: 1

        \* conv2d\_2\_stride: 1

        \* num\_filter\_l1: 16

        \* num\_filter\_l2: 32

        \* maxpool\_1\_kernel\_dim: 2

        \* maxpool\_1\_stride: 2

        \* maxpool\_2\_kernel\_dim: 2

        \* maxpool\_2\_stride: 2

        \* maxpool\_1\_hight: 16

        \* maxpool\_1\_width: 16

        \* maxpool\_1\_depth: 16

        \* maxpool\_2\_hight: 8

        \* maxpool\_2\_width: 8

        \* maxpool\_2\_depth: 32

Last epoch result:

Epoch: 61 | Train - Accuracy: 91.85    % ; Loss: 0.227725 | Validation - Accuracy: 48.8     % ; Loss: 2.755188

Total validation result:

Total validation accuracy: 49.5 [%]

## Experiment 2

        \* data\_dim: 3072

        \* activation\_func: relu

**\* l1\_fc\_hidden\_size: 256**

**\* l2\_fc\_hidden\_size: 32**

        \* num\_of\_classes: 10

        \* initial\_lr: 0.0

        \* lr: 0.01

**\* epochs: 500**

        \* batch\_size: 65

**\* reg: 0.1**

        \* input\_noise\_p: 0

**\* dropout\_p: 0.2**

**\* early\_stop\_max\_epochs: 50**

        \* init\_weights\_mu: 0.0

        \* init\_weights\_sigma: 0.1

        \* early\_stop\_drop: 0.5

        \* early\_stop\_epochs\_drop: 10

        \* input\_z\_score\_normalization: False

        \* input\_hight: 32

        \* input\_width: 32

        \* input\_dim: 32

        \* input\_num\_chanels: 3

        \* chanel\_len: 1024

        \* conv2d\_1\_filter\_dim: 3

        \* conv2d\_2\_filter\_dim: 3

        \* conv2d\_1\_stride: 1

        \* conv2d\_2\_stride: 1

        \* num\_filter\_l1: 16

        \* num\_filter\_l2: 32

        \* maxpool\_1\_kernel\_dim: 2

        \* maxpool\_1\_stride: 2

        \* maxpool\_2\_kernel\_dim: 2

        \* maxpool\_2\_stride: 2

        \* maxpool\_1\_hight: 16

        \* maxpool\_1\_width: 16

        \* maxpool\_1\_depth: 16

        \* maxpool\_2\_hight: 8

        \* maxpool\_2\_width: 8

        \* maxpool\_2\_depth: 32

Last epoch result:

Epoch: 351 | Train - Accuracy: 87.0875  % ; Loss: 0.444209 | Validation - Accuracy: 61.2     % ; Loss: 1.200302

Total validation results:

Total validation accuracy: 62.6 [%]

## Experiment 3

        \* data\_dim: 3072

        \* activation\_func: relu

        \* l1\_fc\_hidden\_size: 256

        \* l2\_fc\_hidden\_size: 32

        \* num\_of\_classes: 10

        \* initial\_lr: 0.0

        \* lr: 0.01

        \* epochs: 500

**\* batch\_size: 250**

        \* reg: 0.1

        \* input\_noise\_p: 0

        \* dropout\_p: 0.2

**\* early\_stop\_criteria: validation\_loss**

**\* early\_stop\_max\_epochs: 25**

        \* init\_weights\_mu: 0.0

        \* init\_weights\_sigma: 0.1

        \* early\_stop\_drop: 0.5

        \* early\_stop\_epochs\_drop: 10

        \* input\_z\_score\_normalization: False

        \* input\_hight: 32

        \* input\_width: 32

        \* input\_dim: 32

        \* input\_num\_chanels: 3

        \* chanel\_len: 1024

        \* conv2d\_1\_filter\_dim: 3

        \* conv2d\_2\_filter\_dim: 3

        \* conv2d\_1\_stride: 1

        \* conv2d\_2\_stride: 1

        \* num\_filter\_l1: 16

        \* num\_filter\_l2: 32

        \* maxpool\_1\_kernel\_dim: 2

        \* maxpool\_1\_stride: 2

        \* maxpool\_2\_kernel\_dim: 2

        \* maxpool\_2\_stride: 2

        \* maxpool\_1\_hight: 16

        \* maxpool\_1\_width: 16

        \* maxpool\_1\_depth: 16

        \* maxpool\_2\_hight: 8

        \* maxpool\_2\_width: 8

        \* maxpool\_2\_depth: 32

Last epoch result:

Epoch: 500 | Train - Accuracy: 56.975   % ; Loss: 1.22548  | Validation - Accuracy: 52.8     % ; Loss: 1.547655

Total validation results:

Total validation accuracy: 52.7 [%]

## Experiment 4

**\* activation\_func: leaky\_relu**

        \* l1\_fc\_hidden\_size: 256

        \* l2\_fc\_hidden\_size: 32

        \* num\_of\_classes: 10

        \* initial\_lr: 0.0

        \* lr: 0.01

        \* epochs: 500

**\* batch\_size: 100**

        \* reg: 0.1

        \* input\_noise\_p: 0

        \* dropout\_p: 0.2

        \* early\_stop\_criteria: validation\_loss

        \* early\_stop\_max\_epochs: 25

        \* init\_weights\_mu: 0.0

        \* init\_weights\_sigma: 0.1

        \* early\_stop\_drop: 0.5

        \* early\_stop\_epochs\_drop: 10

        \* input\_z\_score\_normalization: False

        \* input\_hight: 32

        \* input\_width: 32

        \* input\_dim: 32

        \* input\_num\_chanels: 3

        \* chanel\_len: 1024

**\* f1\_num\_filter: 32**

**\* f2\_num\_filter: 64**

        \* conv2d\_1\_filter\_dim: 3

        \* conv2d\_2\_filter\_dim: 3

        \* conv2d\_1\_stride: 1

        \* conv2d\_2\_stride: 1

        \* maxpool\_1\_kernel\_dim: 2

        \* maxpool\_1\_stride: 2

        \* maxpool\_2\_kernel\_dim: 2

        \* maxpool\_2\_stride: 2

        \* maxpool\_1\_hight: 16

        \* maxpool\_1\_width: 16

**\* maxpool\_1\_depth: 32**

        \* maxpool\_2\_hight: 8

        \* maxpool\_2\_width: 8

**\* maxpool\_2\_depth: 64**

Last epoch result:

Epoch: 421 | Train - Accuracy: 83.15    % ; Loss: 0.532371 | Validation - Accuracy: 63.4     % ; Loss: 1.379112

Total validation results:

Total validation accuracy: 63.5 [%]

## Experiment 5

        \* data\_dim: 3072

        \* activation\_func: leaky\_relu

        \* l1\_fc\_hidden\_size: 256

        \* l2\_fc\_hidden\_size: 32

        \* num\_of\_classes: 10

        \* initial\_lr: 0.0

        \* lr: 0.01

        \* epochs: 500

        \* batch\_size: 100

        \* reg: 0.1

        \* input\_noise\_p: 0

        \* dropout\_p: 0.2

        \* early\_stop\_criteria: validation\_loss

        \* early\_stop\_max\_epochs: 25

        \* init\_weights\_mu: 0.0

        \* init\_weights\_sigma: 0.1

        \* early\_stop\_drop: 0.5

        \* early\_stop\_epochs\_drop: 10

        \* input\_z\_score\_normalization: False

        \* input\_hight: 32

        \* input\_width: 32

        \* input\_dim: 32

        \* input\_num\_chanels: 3

        \* chanel\_len: 1024

        \* f1\_num\_filter: 32

        \* f2\_num\_filter: 64

**\* conv2d\_1\_filter\_dim: 5**

        \* conv2d\_2\_filter\_dim: 3

        \* conv2d\_1\_stride: 1

        \* conv2d\_2\_stride: 1

        \* maxpool\_1\_kernel\_dim: 2

        \* maxpool\_1\_stride: 2

        \* maxpool\_2\_kernel\_dim: 2

        \* maxpool\_2\_stride: 2

        \* maxpool\_1\_hight: 16

        \* maxpool\_1\_width: 16

        \* maxpool\_1\_depth: 32

        \* maxpool\_2\_hight: 8

        \* maxpool\_2\_width: 8

        \* maxpool\_2\_depth: 64

Last epoch result:

Epoch: 439 | Train - Accuracy: 83.9625  % ; Loss: 0.504408 | Validation - Accuracy: 63.5     % ; Loss: 1.454099

Total validation results:

Total validation accuracy: 66.0 [%]

## Experiment 6

        \* data\_dim: 3072

        \* activation\_func: leaky\_relu

        \* l1\_fc\_hidden\_size: 256

        \* l2\_fc\_hidden\_size: 32

        \* num\_of\_classes: 10

        \* initial\_lr: 0.0

        \* lr: 0.01

        \* epochs: 500

        \* batch\_size: 100

        \* reg: 0.1

        \* input\_noise\_p: 0

        \* dropout\_p: 0.2

        \* early\_stop\_criteria: validation\_loss

        \* early\_stop\_max\_epochs: 25

        \* init\_weights\_mu: 0.0

        \* init\_weights\_sigma: 0.1

        \* early\_stop\_drop: 0.5

        \* early\_stop\_epochs\_drop: 10

        \* input\_z\_score\_normalization: False

        \* input\_hight: 32

        \* input\_width: 32

        \* input\_dim: 32

        \* input\_num\_chanels: 3

        \* chanel\_len: 1024

**\* f1\_num\_filter: 64**

**\* f2\_num\_filter: 96**

        \* conv2d\_1\_filter\_dim: 5

        \* conv2d\_2\_filter\_dim: 3

        \* conv2d\_1\_stride: 1

        \* conv2d\_2\_stride: 1

        \* maxpool\_1\_kernel\_dim: 2

        \* maxpool\_1\_stride: 2

        \* maxpool\_2\_kernel\_dim: 2

        \* maxpool\_2\_stride: 2

        \* maxpool\_1\_hight: 16

        \* maxpool\_1\_width: 16

**\* maxpool\_1\_depth: 64**

        \* maxpool\_2\_hight: 8

        \* maxpool\_2\_width: 8

**\* maxpool\_2\_depth: 96**

Last epoch result:

Epoch: 500 | Train - Accuracy: 94.3875  % ; Loss: 0.409317 | Validation - Accuracy: 64.6     % ; Loss: 2.767867

Total validation results:

Total validation accuracy: 64.6 [%]

## Experiment 7

        \* data\_dim: 3072

        \* activation\_func: leaky\_relu

**\* data\_augmentation\_ratio: 0.3**

        \* l1\_fc\_hidden\_size: 256

        \* l2\_fc\_hidden\_size: 32

        \* num\_of\_classes: 10

        \* initial\_lr: 0.0

        \* lr: 0.01

**\* epochs: 1000**

        \* batch\_size: 100

        \* reg: 0.1

        \* input\_noise\_p: 0

        \* dropout\_p: 0.2

        \* early\_stop\_criteria: validation\_loss

        \* early\_stop\_max\_epochs: 25

        \* init\_weights\_mu: 0.0

        \* init\_weights\_sigma: 0.1

        \* early\_stop\_drop: 0.5

        \* early\_stop\_epochs\_drop: 10

        \* input\_z\_score\_normalization: False

        \* input\_hight: 32

        \* input\_width: 32

        \* input\_dim: 32

        \* input\_num\_chanels: 3

        \* chanel\_len: 1024

**\* f1\_num\_filter: 32**

**\* f2\_num\_filter: 64**

        \* conv2d\_1\_filter\_dim: 5

        \* conv2d\_2\_filter\_dim: 3

        \* conv2d\_1\_stride: 1

        \* conv2d\_2\_stride: 1

        \* maxpool\_1\_kernel\_dim: 2

        \* maxpool\_1\_stride: 2

        \* maxpool\_2\_kernel\_dim: 2

        \* maxpool\_2\_stride: 2

        \* maxpool\_1\_hight: 16

        \* maxpool\_1\_width: 16

**\* maxpool\_1\_depth: 32**

        \* maxpool\_2\_hight: 8

        \* maxpool\_2\_width: 8

**\* maxpool\_2\_depth: 64**

Last epoch result:

Epoch: 941 | Train - Accuracy: 88.0     % ; Loss: 0.480944 | Validation - Accuracy: 66.0     % ; Loss: 1.581208

Total validation results:

Total validation accuracy: 66.2 [%]

## Experiment 8

        \* activation\_func: leaky\_relu

        \* num\_of\_classes: 10

        \* data\_dim: 3072

        \* chanel\_len: 1024

        \* num\_of\_input\_chanels: 3

        \* input\_dim: 32

        \* f1\_num\_filter: 32

        \* f2\_num\_filter: 64

**\* l1\_fc\_hidden\_size: 512**

**\* l2\_fc\_hidden\_size: 128**

        \* conv2d\_1\_filter\_dim: 5

        \* conv2d\_2\_filter\_dim: 3

        \* conv2d\_1\_stride: 1

        \* conv2d\_2\_stride: 1

        \* maxpool\_1\_kernel\_dim: 2

        \* maxpool\_1\_stride: 2

        \* maxpool\_2\_kernel\_dim: 2

        \* maxpool\_2\_stride: 2

        \* maxpool\_1\_hight: 16

        \* maxpool\_1\_width: 16

        \* maxpool\_1\_depth: 32

        \* maxpool\_2\_hight: 8

        \* maxpool\_2\_width: 8

        \* maxpool\_2\_depth: 64

        \* epochs: 1000

**\* batch\_size: 50**

        \* data\_augmentation\_ratio: 0.3

        \* lr: 0.04

**\* initial\_lr: 0.04**

**\* lr\_decay\_drop\_ratio: 0.2**

**\* lr\_decay\_epochs\_drop: 15**

        \* reg: 0.1

        \* input\_noise\_p: 0

        \* dropout\_p: 0.2

        \* early\_stop\_criteria: validation\_loss

        \* early\_stop\_max\_epochs: 25

        \* init\_weights\_mu: 0.0

        \* init\_weights\_sigma: 0.1

        \* input\_z\_score\_normalization: False

Last epoch result:

Epoch: 683 | Train - Accuracy: 91.475   % ; Loss: 0.340195 | Validation - Accuracy: 67.5     % ; Loss: 1.346056

Total validation results:

Total validation accuracy: 67.2 [%]

## Experiment 9

        \* activation\_func: leaky\_relu

        \* num\_of\_classes: 10

        \* data\_dim: 3072

        \* chanel\_len: 1024

        \* num\_of\_input\_chanels: 3

        \* input\_dim: 32

        \* f1\_num\_filter: 32

**\* f2\_num\_filter: 96**

        \* l1\_fc\_hidden\_size: 512

        \* l2\_fc\_hidden\_size: 128

        \* conv2d\_1\_filter\_dim: 5

        \* conv2d\_2\_filter\_dim: 3

        \* conv2d\_1\_stride: 1

        \* conv2d\_2\_stride: 1

        \* maxpool\_1\_kernel\_dim: 2

        \* maxpool\_1\_stride: 2

        \* maxpool\_2\_kernel\_dim: 2

        \* maxpool\_2\_stride: 2

        \* maxpool\_1\_hight: 16

        \* maxpool\_1\_width: 16

        \* maxpool\_1\_depth: 32

        \* maxpool\_2\_hight: 8

        \* maxpool\_2\_width: 8

**\* maxpool\_2\_depth: 96**

        \* epochs: 1000

        \* batch\_size: 50

        \* data\_augmentation\_ratio: 0.3

        \* lr: 0.03

**\* initial\_lr: 0.03**

**\* lr\_decay\_drop\_ratio: 0.15**

        \* lr\_decay\_epochs\_drop: 15

**\* reg: 0.15**

        \* input\_noise\_p: 0

        \* dropout\_p: 0.2

        \* early\_stop\_criteria: validation\_loss

        \* early\_stop\_max\_epochs: 25

        \* init\_weights\_mu: 0.0

        \* init\_weights\_sigma: 0.1

        \* input\_z\_score\_normalization: False

Last epoch result:

Epoch: 579 | Train - Accuracy: 92.35    % ; Loss: 0.323469 | Validation - Accuracy: 68.1     % ; Loss: 1.331067

Total validation results:

Total validation accuracy: 69.6 [%]

## Experiment 10

      \* activation\_func: leaky\_relu

**\* optimizer: rmsprop**

        \* num\_of\_classes: 10

        \* data\_dim: 3072

        \* chanel\_len: 1024

        \* num\_of\_input\_chanels: 3

        \* input\_dim: 32

        \* f1\_num\_filter: 32

        \* f2\_num\_filter: 96

        \* l1\_fc\_hidden\_size: 512

        \* l2\_fc\_hidden\_size: 128

        \* conv2d\_1\_filter\_dim: 5

        \* conv2d\_2\_filter\_dim: 3

        \* conv2d\_1\_stride: 1

        \* conv2d\_2\_stride: 1

        \* maxpool\_1\_kernel\_dim: 2

        \* maxpool\_1\_stride: 2

        \* maxpool\_2\_kernel\_dim: 2

        \* maxpool\_2\_stride: 2

        \* maxpool\_1\_hight: 16

        \* maxpool\_1\_width: 16

        \* maxpool\_1\_depth: 32

        \* maxpool\_2\_hight: 8

        \* maxpool\_2\_width: 8

        \* maxpool\_2\_depth: 96

        \* epochs: 1000

        \* batch\_size: 50

**\* data\_augmentation\_ratio: 0.4**

        \* lr: 0.001

**\* initial\_lr: 0.001**

**\* lr\_decay\_drop\_ratio: 0.2**

**\* lr\_decay\_epochs\_drop: 25**

        \* reg: 0.1

        \* input\_noise\_p: 0

        \* dropout\_p: 0.2

        \* early\_stop\_criteria: validation\_loss

**\* early\_stop\_max\_epochs: 20**

        \* init\_weights\_mu: 0.0

        \* init\_weights\_sigma: 0.1

        \* input\_z\_score\_normalization: False

Last epoch result:

Epoch: 73 | Train - Accuracy: 82.8875  % ; Loss: 0.565406 | Validation - Accuracy: 60.9     % ; Loss: 1.155291

Total validation results:

Total validation accuracy: 63.4 [%]

## Experiment 11

        \* activation\_func: leaky\_relu

**\* optimizer: momentum**

        \* num\_of\_classes: 10

        \* data\_dim: 3072

        \* chanel\_len: 1024

        \* num\_of\_input\_chanels: 3

        \* input\_dim: 32

        \* f1\_num\_filter: 32

        \* f2\_num\_filter: 96

        \* l1\_fc\_hidden\_size: 512

        \* l2\_fc\_hidden\_size: 128

        \* conv2d\_1\_filter\_dim: 5

        \* conv2d\_2\_filter\_dim: 3

        \* conv2d\_1\_stride: 1

        \* conv2d\_2\_stride: 1

        \* maxpool\_1\_kernel\_dim: 2

        \* maxpool\_1\_stride: 2

        \* maxpool\_2\_kernel\_dim: 2

        \* maxpool\_2\_stride: 2

        \* maxpool\_1\_hight: 16

        \* maxpool\_1\_width: 16

        \* maxpool\_1\_depth: 32

        \* maxpool\_2\_hight: 8

        \* maxpool\_2\_width: 8

        \* maxpool\_2\_depth: 96

        \* epochs: 1000

        \* batch\_size: 50

        \* data\_augmentation\_ratio: 0.4

**\* lr: 0.01**

**\* initial\_lr: 0.01**

        \* lr\_decay\_drop\_ratio: 0.2

        \* lr\_decay\_epochs\_drop: 25

        \* reg: 0.1

        \* input\_noise\_p: 0

        \* dropout\_p: 0.2

        \* early\_stop\_criteria: validation\_loss

        \* early\_stop\_max\_epochs: 20

        \* init\_weights\_mu: 0.0

        \* init\_weights\_sigma: 0.1

        \* input\_z\_score\_normalization: False

Last epoch result:

Epoch: 96 | Train - Accuracy: 90.5625  % ; Loss: 0.333    | Validation - Accuracy: 63.8     % ; Loss: 1.165878

Total validation results:

Total validation accuracy: 65.2 [%]

## Experiment 12

        \* activation\_func: leaky\_relu

        \* optimizer: momentum

        \* num\_of\_classes: 10

        \* data\_dim: 3072

        \* chanel\_len: 1024

        \* num\_of\_input\_chanels: 3

        \* input\_dim: 32

        \* f1\_num\_filter: 32

**\* f2\_num\_filter: 64**

        \* l1\_fc\_hidden\_size: 512

        \* l2\_fc\_hidden\_size: 128

        \* conv2d\_1\_filter\_dim: 5

        \* conv2d\_2\_filter\_dim: 3

        \* conv2d\_1\_stride: 1

        \* conv2d\_2\_stride: 1

        \* maxpool\_1\_kernel\_dim: 2

        \* maxpool\_1\_stride: 2

        \* maxpool\_2\_kernel\_dim: 2

        \* maxpool\_2\_stride: 2

        \* maxpool\_1\_hight: 16

        \* maxpool\_1\_width: 16

        \* maxpool\_1\_depth: 32

        \* maxpool\_2\_hight: 8

        \* maxpool\_2\_width: 8

**\* maxpool\_2\_depth: 64**

        \* epochs: 1000

        \* batch\_size: 50

        \* data\_augmentation\_ratio: 0.4

        \* lr: 0.01

        \* initial\_lr: 0.01

        \* lr\_decay\_drop\_ratio: 0.2

        \* lr\_decay\_epochs\_drop: 25

        \* reg: 0.1

        \* input\_noise\_p: 0

        \* dropout\_p: 0.2

        \* early\_stop\_criteria: validation\_loss

        \* early\_stop\_max\_epochs: 20

        \* init\_weights\_mu: 0.0

        \* init\_weights\_sigma: 0.1

        \* input\_z\_score\_normalization: False

Last epoch result:

Epoch: 122 | Train - Accuracy: 91.85    % ; Loss: 0.310122 | Validation - Accuracy: 65.6     % ; Loss: 1.117736

Total validation results:

Total validation accuracy: 65.6 [%]

## Experiment 13

        \* activation\_func: leaky\_relu

**\* optimizer: sgd**

        \* num\_of\_classes: 10

        \* data\_dim: 3072

        \* chanel\_len: 1024

        \* num\_of\_input\_chanels: 3

        \* input\_dim: 32

        \* f1\_num\_filter: 32

        \* f2\_num\_filter: 64

        \* l1\_fc\_hidden\_size: 512

        \* l2\_fc\_hidden\_size: 128

        \* conv2d\_1\_filter\_dim: 5

        \* conv2d\_2\_filter\_dim: 3

        \* conv2d\_1\_stride: 1

        \* conv2d\_2\_stride: 1

        \* maxpool\_1\_kernel\_dim: 2

        \* maxpool\_1\_stride: 2

        \* maxpool\_2\_kernel\_dim: 2

        \* maxpool\_2\_stride: 2

        \* maxpool\_1\_hight: 16

        \* maxpool\_1\_width: 16

        \* maxpool\_1\_depth: 32

        \* maxpool\_2\_hight: 8

        \* maxpool\_2\_width: 8

        \* maxpool\_2\_depth: 64

        \* epochs: 1000

        \* batch\_size: 50

        \* data\_augmentation\_ratio: 0.4

        \* lr: 0.04

**\* initial\_lr: 0.04**

        \* lr\_decay\_drop\_ratio: 0.2

        \* lr\_decay\_epochs\_drop: 25

        \* reg: 0.1

        \* input\_noise\_p: 0

        \* dropout\_p: 0.2

        \* early\_stop\_criteria: validation\_loss

        \* early\_stop\_max\_epochs: 20

        \* init\_weights\_mu: 0.0

        \* init\_weights\_sigma: 0.1

        \* input\_z\_score\_normalization: False

Last epoch result:

Epoch: 237 | Train - Accuracy: 88.5875  % ; Loss: 0.406546 | Validation - Accuracy: 66.3     % ; Loss: 1.127008

Total validation results:

Total validation accuracy: 66.5 [%]

## Experiment 14

        \* activation\_func: leaky\_relu

        \* optimizer: sgd

        \* num\_of\_classes: 10

        \* data\_dim: 3072

        \* chanel\_len: 1024

        \* num\_of\_input\_chanels: 3

        \* input\_dim: 32

        \* f1\_num\_filter: 32

        \* f2\_num\_filter: 64

        \* l1\_fc\_hidden\_size: 512

        \* l2\_fc\_hidden\_size: 128

        \* conv2d\_1\_filter\_dim: 5

        \* conv2d\_2\_filter\_dim: 3

        \* conv2d\_1\_stride: 1

        \* conv2d\_2\_stride: 1

        \* maxpool\_1\_kernel\_dim: 2

        \* maxpool\_1\_stride: 2

        \* maxpool\_2\_kernel\_dim: 2

        \* maxpool\_2\_stride: 2

        \* maxpool\_1\_hight: 16

        \* maxpool\_1\_width: 16

        \* maxpool\_1\_depth: 32

        \* maxpool\_2\_hight: 8

        \* maxpool\_2\_width: 8

        \* maxpool\_2\_depth: 64

        \* epochs: 1000

        \* batch\_size: 50

**\* data\_augmentation\_ratio: 0.3**

        \* lr: 0.03

**\* initial\_lr: 0.03**

**\* lr\_decay\_drop\_ratio: 0.15**

**\* lr\_decay\_epochs\_drop: 15**

**\* reg: 0.15**

        \* input\_noise\_p: 0

        \* dropout\_p: 0.2

        \* early\_stop\_criteria: validation\_loss

        \* early\_stop\_max\_epochs: 20

        \* init\_weights\_mu: 0.0

        \* init\_weights\_sigma: 0.1

        \* input\_z\_score\_normalization: False

Last epoch result:

Epoch: 295 | Train - Accuracy: 87.525   % ; Loss: 0.476594 | Validation - Accuracy: 64.9     % ; Loss: 1.144074

Total validation results:

Total validation accuracy: 64.2 [%]

## Experiment 15 – 22 – Optimizers

**Validation Accuracy**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Adam | RMSprop | adagrad | Momentum | SGD (LR decay) | Augmentation Ratio | Learning Rate |
| 63.9 | 63.2 | 59.09 | 69.2 | 68.1 | **0.3** | **0.001** |
| 65.3 | 64.1 | - | - | - | **0.5** |
|  |  | - | - | 67.0 | **0.3** | **0.003** |
| - | - | - | 66.9 | - | **0.4** |

**Epochs to Converge**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Adam | RMSprop | adagrad | Momentum | SGD (LR decay) | Augmentation Ratio | Learning Rate |
| 156 | 65 | - | 501 | 633 | **0.3** | **0.001** |
| 107 | 80 | - | - | - | **0.5** |
|  |  | - | - | 365 | **0.3** | **0.003** |
| - | - | - | 260 | - | **0.4** |

## Experiment 23 – Our model **without** transfer learning

\* model\_architecture: {'F1': (5, 5, 3, 32), 'B1': (32,), 'F2': (3, 3, 32, 64), 'B2': (64,), 'W3': (512, 4096), 'B3': (512, 1), 'W4': (128, 512), 'B4': (128, 1), 'W5': (10, 128), 'B5': (10, 1)}

\* activation\_func: leaky\_relu

\* optimizer: momentum

\* num\_of\_classes: 10

\* data\_dim: 3072

\* chanel\_len: 1024

\* num\_of\_input\_chanels: 3

\* input\_dim: 32

\* f1\_num\_filter: 32

\* f2\_num\_filter: 64

\* l1\_fc\_hidden\_size: 512

\* l2\_fc\_hidden\_size: 128

\* conv2d\_1\_filter\_dim: 5

\* conv2d\_2\_filter\_dim: 3

\* conv2d\_1\_stride: 1

\* conv2d\_2\_stride: 1

\* maxpool\_1\_kernel\_dim: 2

\* maxpool\_1\_stride: 2

\* maxpool\_2\_kernel\_dim: 2

\* maxpool\_2\_stride: 2

\* maxpool\_1\_hight: 16

\* maxpool\_1\_width: 16

\* maxpool\_1\_depth: 32

\* maxpool\_2\_hight: 8

\* maxpool\_2\_width: 8

\* maxpool\_2\_depth: 64

\* epochs: 1000

\* batch\_size: 50

\* data\_augmentation\_ratio: 0.3

\* lr: 0.001

\* initial\_lr: 0.0

\* lr\_decay\_drop\_ratio: 0.15

\* lr\_decay\_epochs\_drop: 15

\* reg: 0.15

\* input\_noise\_p: 0

\* dropout\_p: 0.2

\* early\_stop\_criteria: validation\_loss

\* early\_stop\_max\_epochs: 25

\* init\_weights\_mu: 0.0

\* init\_weights\_sigma: 0.1

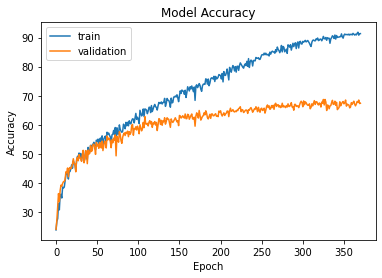
\* input\_z\_score\_normalization: False

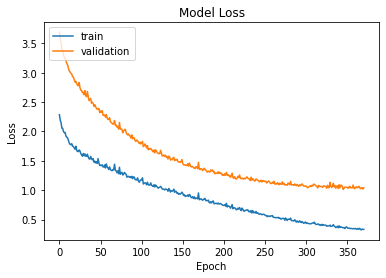
Last epoch result:

Epoch: **371** | Train - Accuracy: **91.3625 %** ; Loss: 0.334235 | Validation - Accuracy: **67.4 %** ; Loss: 1.042357

Total validation result:

Total validation accuracy: **67.9 [%]**





## Experiment 24 – Our model **with** transfer learning – **Final submitted model**

\* model\_architecture: {'F1': (5, 5, 3, 32), 'B1': (32,), 'F2': (3, 3, 32, 64), 'B2': (64,), 'W3': (512, 4096), 'B3': (512, 1), 'W4': (128, 512), 'B4': (128, 1), 'W5': (10, 128), 'B5': (10, 1)}

\* activation\_func: leaky\_relu

\* optimizer: momentum

\* num\_of\_classes: 10

\* data\_dim: 3072

\* chanel\_len: 1024

\* num\_of\_input\_chanels: 3

\* input\_dim: 32

\* f1\_num\_filter: 32

\* f2\_num\_filter: 64

\* l1\_fc\_hidden\_size: 512

\* l2\_fc\_hidden\_size: 128

\* conv2d\_1\_filter\_dim: 5

\* conv2d\_2\_filter\_dim: 3

\* conv2d\_1\_stride: 1

\* conv2d\_2\_stride: 1

\* maxpool\_1\_kernel\_dim: 2

\* maxpool\_1\_stride: 2

\* maxpool\_2\_kernel\_dim: 2

\* maxpool\_2\_stride: 2

\* maxpool\_1\_hight: 16

\* maxpool\_1\_width: 16

\* maxpool\_1\_depth: 32

\* maxpool\_2\_hight: 8

\* maxpool\_2\_width: 8

\* maxpool\_2\_depth: 64

\* epochs: 1000

\* batch\_size: 50

\* data\_augmentation\_ratio: 0.3

\* lr: 0.001

\* initial\_lr: 0.0

\* lr\_decay\_drop\_ratio: 0.15

\* lr\_decay\_epochs\_drop: 15

\* reg: 0.15

\* input\_noise\_p: 0

\* dropout\_p: 0.2

\* early\_stop\_criteria: validation\_loss

\* early\_stop\_max\_epochs: 25

\* init\_weights\_mu: 0.0

\* init\_weights\_sigma: 0.1

\* input\_z\_score\_normalization: False

Last epoch result:

Epoch: **47** | Train - Accuracy: **88.55 %** ; Loss: 0.415342 | Validation - Accuracy: **87.1 %** ; Loss: 0.438707 | Duration: 00:00:48

Total validation result:

Total validation accuracy: **90.9 [%]**

