Deep Learning in Computer Vision

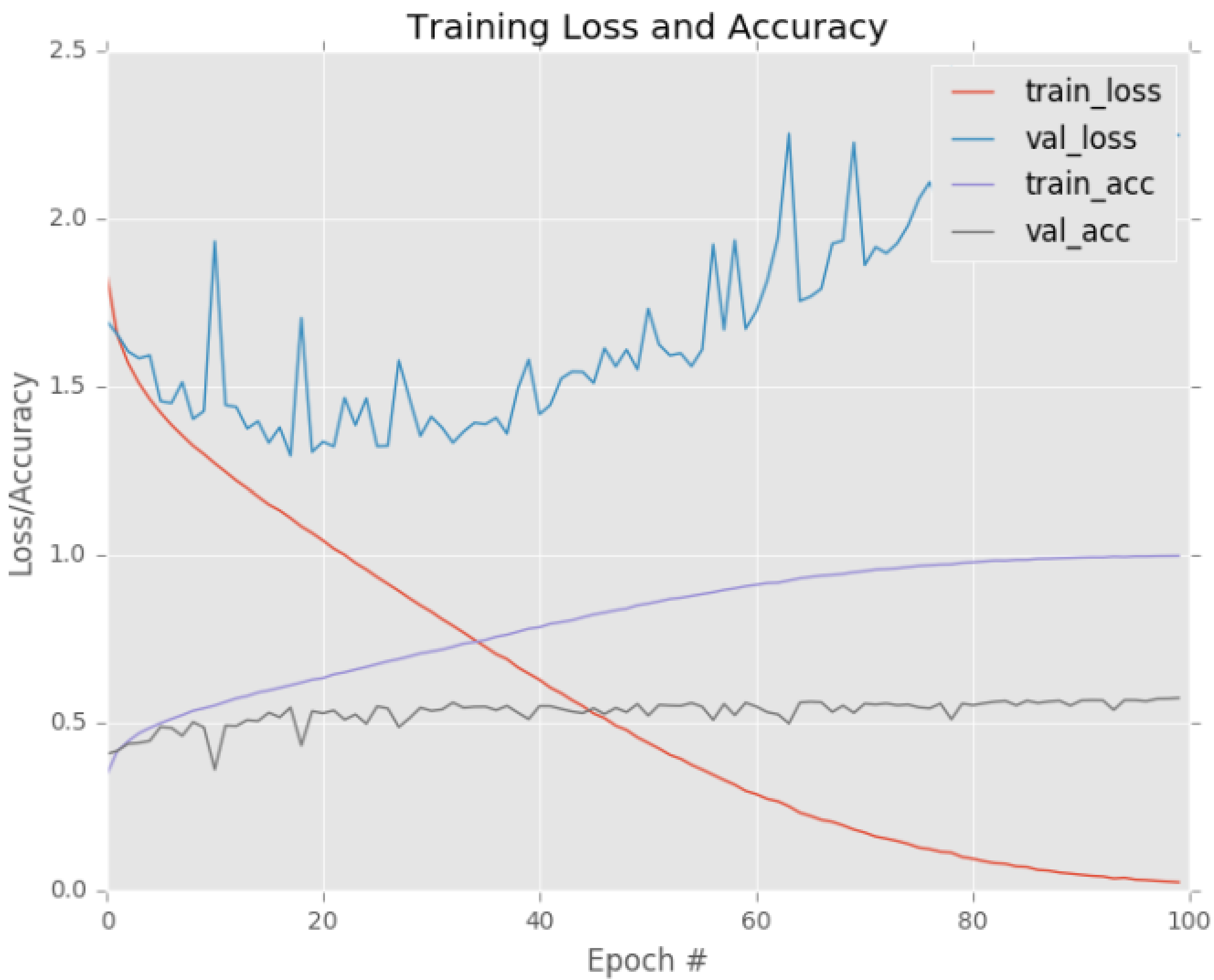
Image Classification Using Feed-Forward ANN

Instructor: Dr. Mehrandezh
Student: Marzieh Zamani

Important Parts in a Neural Network Classifier

- Dataset
  - Training dataset
  - Validation dataset
- Loss Function
- Model Architecture
  - Number of layers and nodes
- Optimization Method
  - Minimising training loss to maximise classification accuracy (validation accuracy)

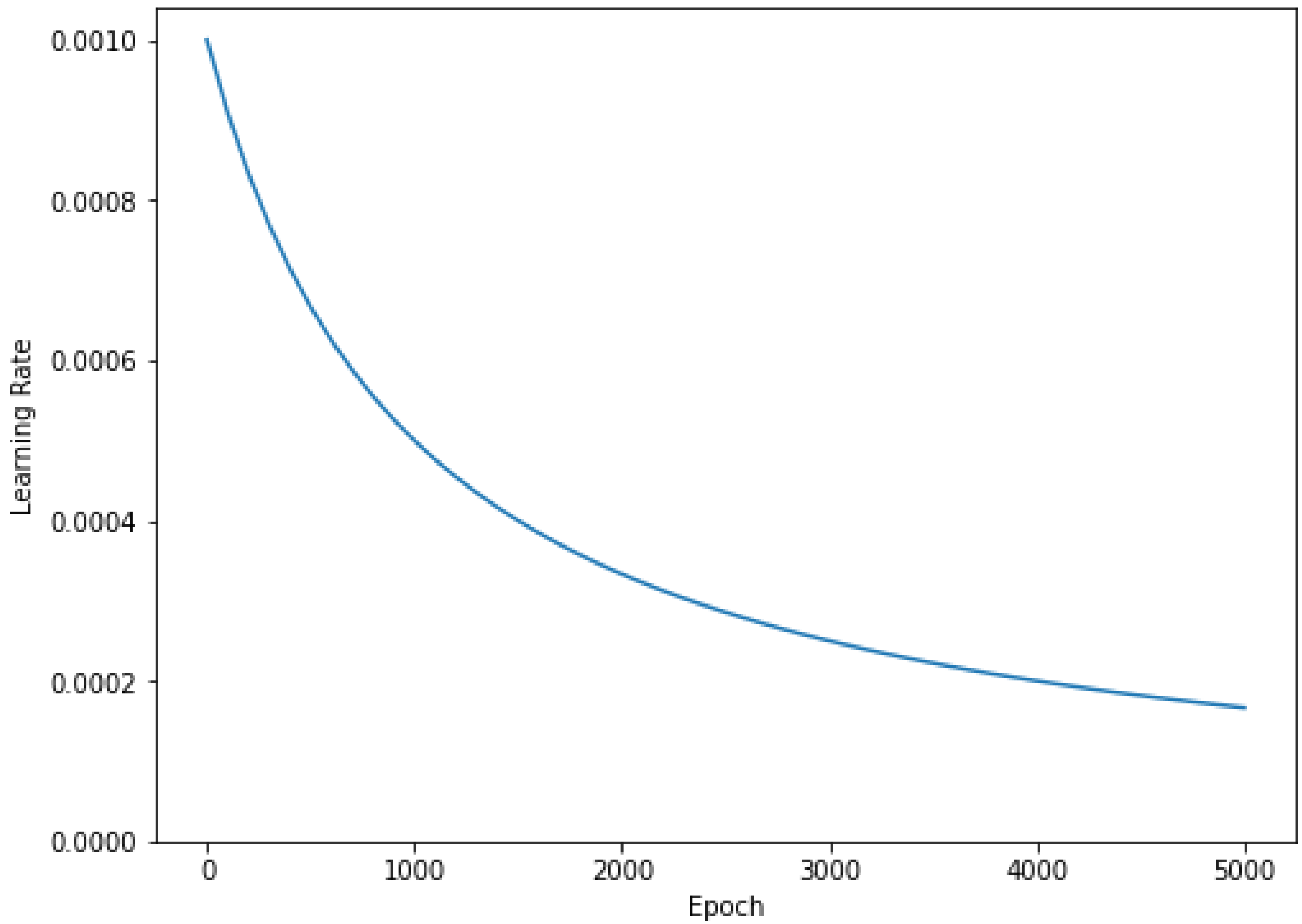
Training Challenge: Overfitting



Strategies to Prevent Overfitting [2]

- 1. Getting more training data
- 2. Descending Learning Rate
- 3. Reducing the capacity of the network.
- 4. Adding weight regularization.
- 5. Adding dropout
- 6. Data-augmentation
- 7. Batch normalization

Descending Learning Rate



Adding weight regularization

- L1 regularization

$$R(W) = \sum_i \sum_j |W_{i,j}|$$

- L2 regularization

$$R(W) = \sum_i \sum_j W_{i,j}^2$$

Adding dropout

- One of the most effective and most commonly used regularization techniques for neural networks,
  - developed by the University of Toronto.

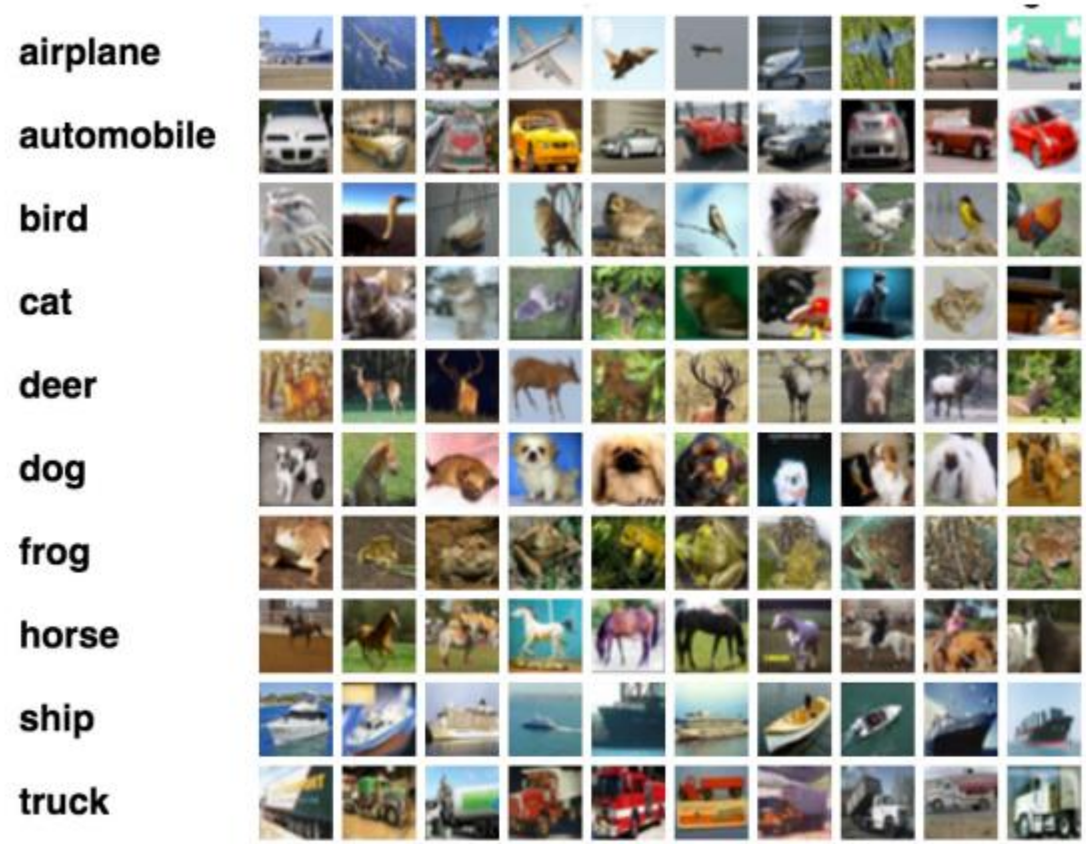
- Randomly "dropping out" (i.e. set to zero) a number of output features of the layer during training.

- E.g. [0.2, 0.5, 1.3, 0.8, 1.1] => [0, 0.5, 1.3, 0, 1.1].

- The "dropout rate" is the fraction of the features that are being zeroed-out; (0.2-0.5)

Implementation
Step #1 – Load Data

- Dataset for Training and Validation
  - 12,000 images from Cifar-10 dataset (60,000 images)



Implementation
Step #2 – Define Keras Model

Model Architecture		Weight regularization & Dropout
4-layer model 3072 x 1024 x 512 x 10		None
	X	weight regularization (L2)
3-layer model 3072 x 1024 x 10		weight regularization (L2) + dropout

Implementation
Step #3 – Compile Keras Model

Loss Function		Optimizer		Learning Rate		Momentum
				Constant (0.01)		0
categorical_ crossentropy	X	SGD	X		X	
				Descending		0.5

Implementation
Step #4 – Fit Keras Model

- Using training and testing data

Implementation
Step #5 – Evaluate Keras Model

1. Cifar-10 truck images
2. 167 Unknown truck images (original)
3. 167 Unknown truck images (scaled)
4. 167 Unknown truck images (blurred)
5. 167 Unknown truck images (flipped horizontally)
6. 167 Unknown truck images (flipped vertically)
7. 167 Unknown truck images (rotated 180 degree)
8. 167 Unknown truck images (rotated 90 degree)

Results: Training Loos and Accuracy for Model

Results:
Training Loos and Accuracy for Model

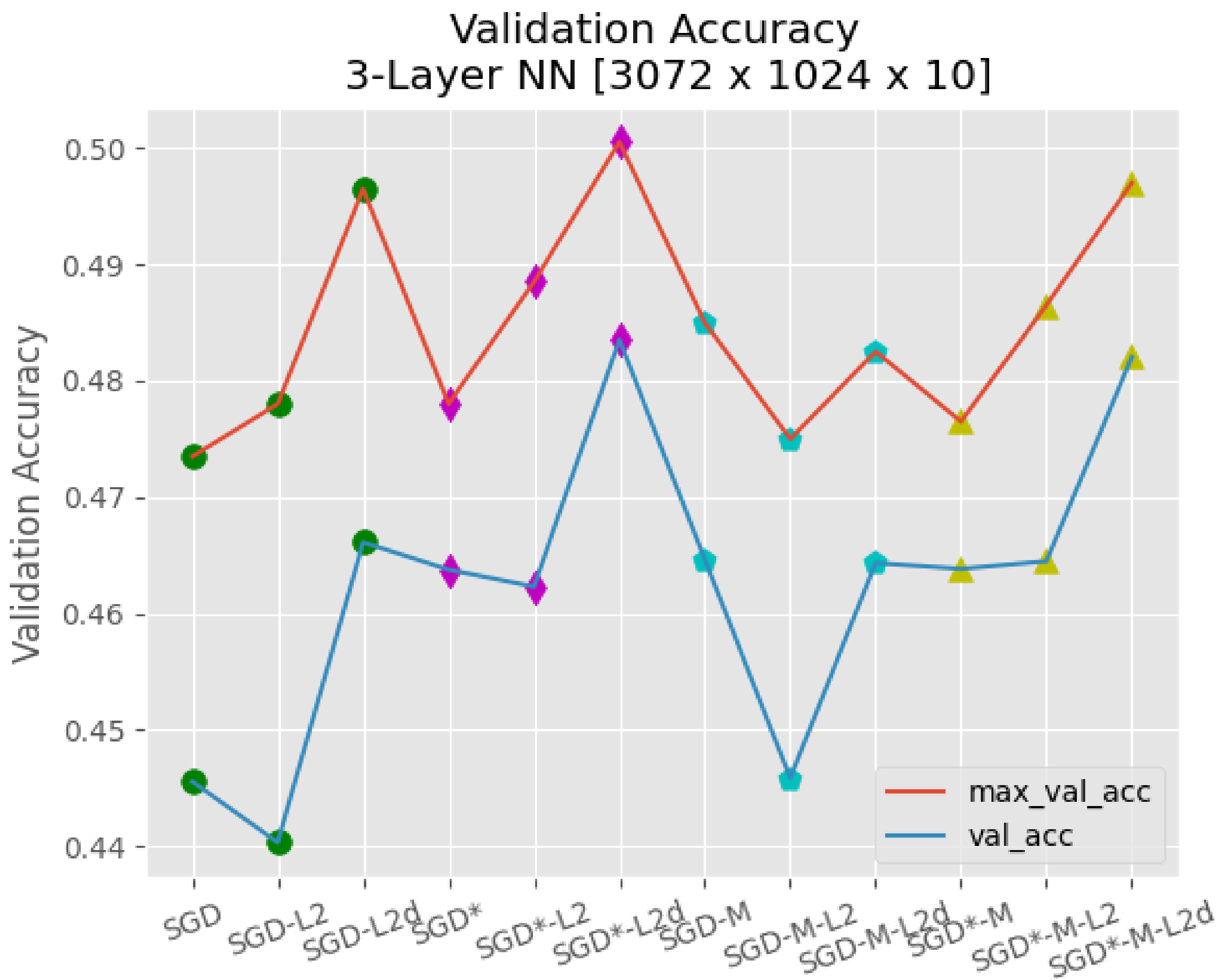
- **Loss:**

Training loss is constantly decreasing but validation loss is first decreasing and then starts increasing; (Overfitting)

- **Accuracy:**

While training accuracy is reaching nearly 100%, validation accuracy is less than 50%.

Results:
Validation Accuracy for 3-Layer Models



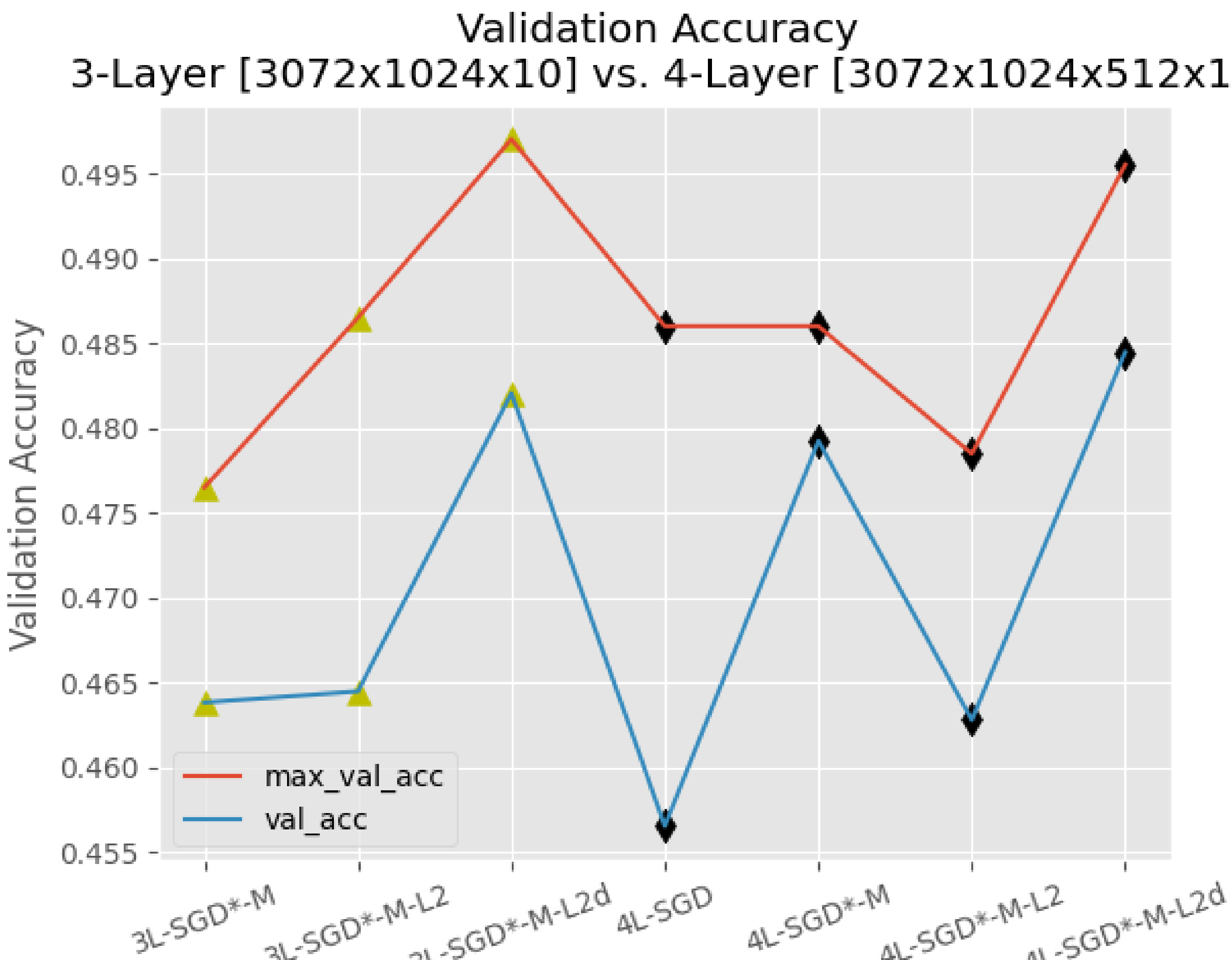
Results: Validation Accuracy for 3-Layer Models

- Validation accuracies for models with same optimizer (but different weight regularizer) are displayed with the same color and marking

- Effect of weight regularization:
  - L2 weight regularization reduces average validation accuracy but increases max validation accuracy
  - However L2d (L2 + dropout) significant increases both average and max validation accuracies

- Effect of learning rate and momentum:
  - There is not a clear pattern for the effect of momentum;
  - However, SGD* and SGD*-M (descending learning rate with/without momentum) are slightly better than SGD and SGD-M (constant learning rate with/without momentum).

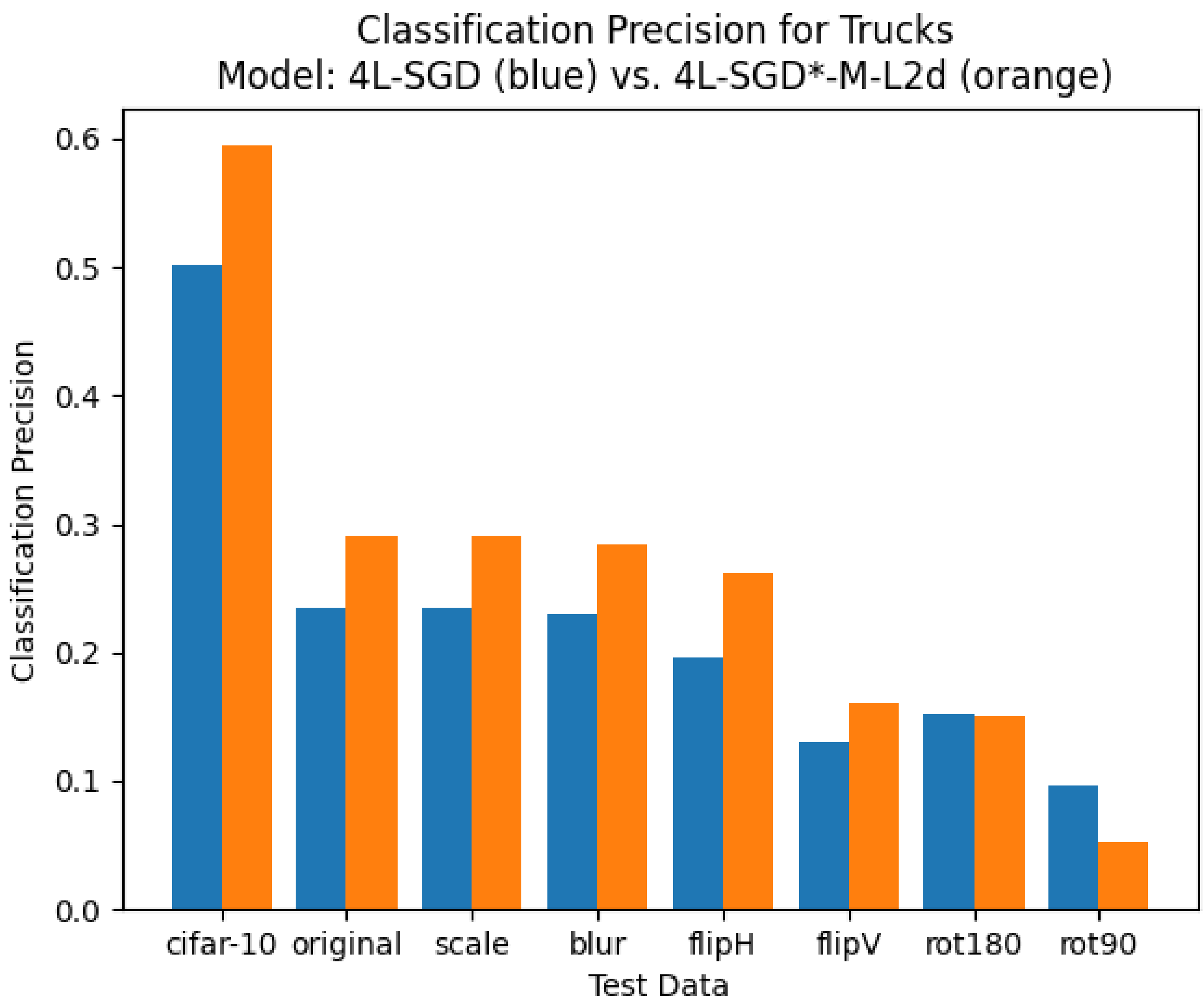
Results:
Validation Accuracy for 3-Layer vs. 4-Layer Models



Results:
Validation Accuracy for 3-Layer vs. 4-Layer Models

- Validation accuracies for models with same architecture (but different weight regularizer) are displayed with the same color and marking
- Effect of parameter modification
  - 4L-SGD points are the average and max accuracies for base model (default parameters)
  - The average accuracy of all modified models (3-layer & 4-layer) is higher than base model;
  - There is no specific pattern for max accuracies
- Effect of weight regularization for 4-Layer models:
  - L2d (L2 + dropout) significant increases both average and max validation accuracies
- Effect of model architecture
  - 4-layer models have accuracies slightly higher than 3-layer models but the difference is less than expected

Results:
Classification Precision for Trucks for Two Models



Results:
Classification Precision for Trucks for Two Models

- Dataset vs. unknown image:
  - Although test dataset is only used for validation, and not for training, their classification accuracy is about 50%: almost 2 time more than unknown images
- Base model (4L-SGD) vs. modified model (4L-SGD*-M-L2d)
  - Almost for all test batches, modified model has higher accuracies than base model
- Effect of filters
  - Scale has no effect as all images are scaled to 32x32x3 in pre-processing step (30%)
  - Blur and horizontal flip have very little effects (30%)
  - Vertical flip and 180-degree rotation are very similar with less than 20% accuracies
  - Among all filters, 90-degree rotation has most effect, reducing accuracy from 30% to less than 10%

Challenges and validity of results

- The effect of modifying parameters is not always clear
- There needs to be multiple runs for each model to ensure repeatability which is very time consuming

References

- Rosebrock, A. (2017). Deep learning for computer vision with python: starter bundle. PyImageSearch.
- https://www.tensorflow.org/tutorials/keras/overfit_and_underfit
- Cifar-10 Dataset: https://pjreddie.com/projects/cifar-10-dataset-mirror/

Thanks for your attention

Questions are welcome