CENG 499 Homework-1 Report

Choosing Best Hyperparameters

In order to decide the best hyperparameters, I changed learning rate, epoch number, activation function, optimizer and layer number to see how they effect on test accuracy. I listed my observations for every variable below:

Learning rate: Small learning rates showed best performances on every layer numbers.

Epoch number: After 10 epochs, test accuracy becomes steady and it decreases after 20 epochs.

Activation function: Activation functions didn't show significant effect on accuracy but GELU performed better among others.

Optimizer: Adam performs better than SGD as an optimizer in this dataset.

Layer number: Increasing layer numbers caused more accurate test results.

Accuracy

The accuracy metric is suitable for CIFAR-10 dataset. The problem of the accuracy metric arises from unbalanced class distribution. If half of the dataset belongs to one class, then a model can achieve %50 accuracy just by classifying every instance as this class. However, CIFAR-10 has 10 classes with 6000 instances in each class, so the accuracy metric is suitable for this specific dataset.

Overfitting

Overfitting means that a model learns the training data very well even with noise and its performance starts to decrease on new data. The aim of a good model is to generalize from training data to show high performance on new data. There are some ways to prevent the overfitting.

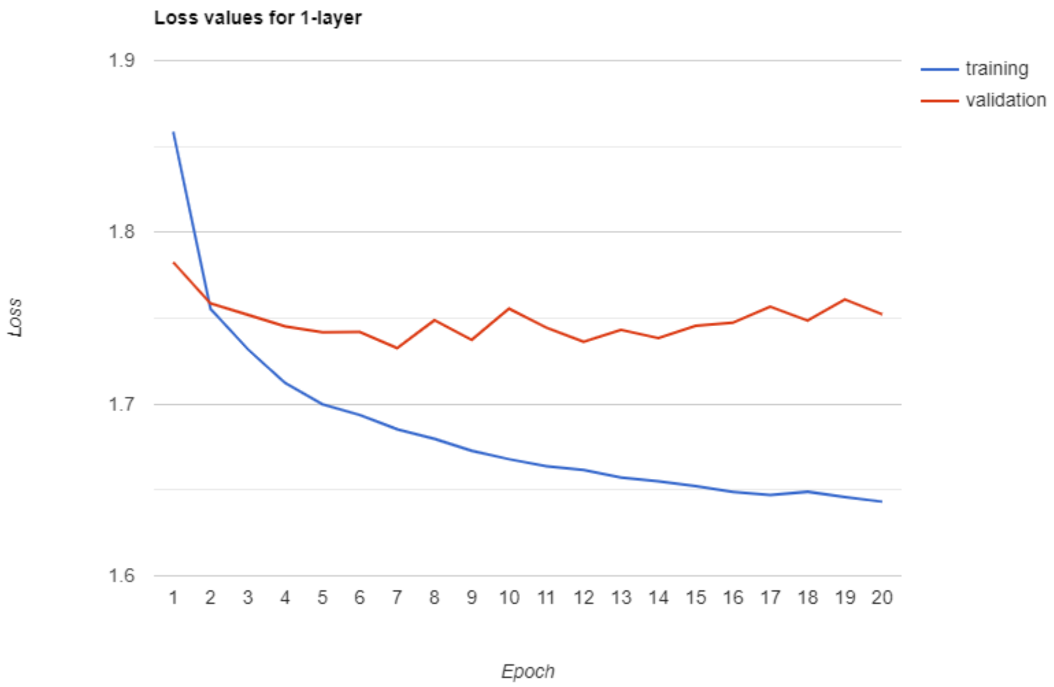
One of them is to use part of the training data as validation data. Validation data is not used in the training process. After each epoch iteration, performance of the model on validation set is measured. The performance starts to decrease after some iterations. Then, we can know that the model starts to overfit the training data and we can stop the process.

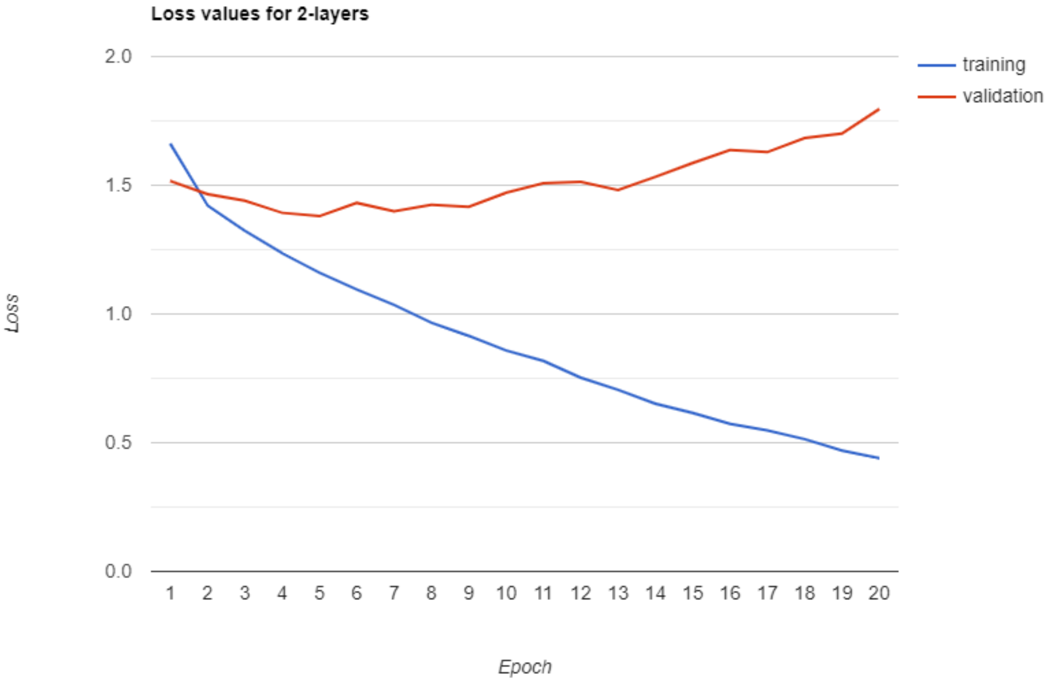
Also, cross validation can be used to prevent overfitting. In cross validation, training data is divided into equally sized parts. Each part is iteratively hold out and the model trains with the remained parts and the hold out part is used as test data. It provides tuning hyperparameters with only training set.

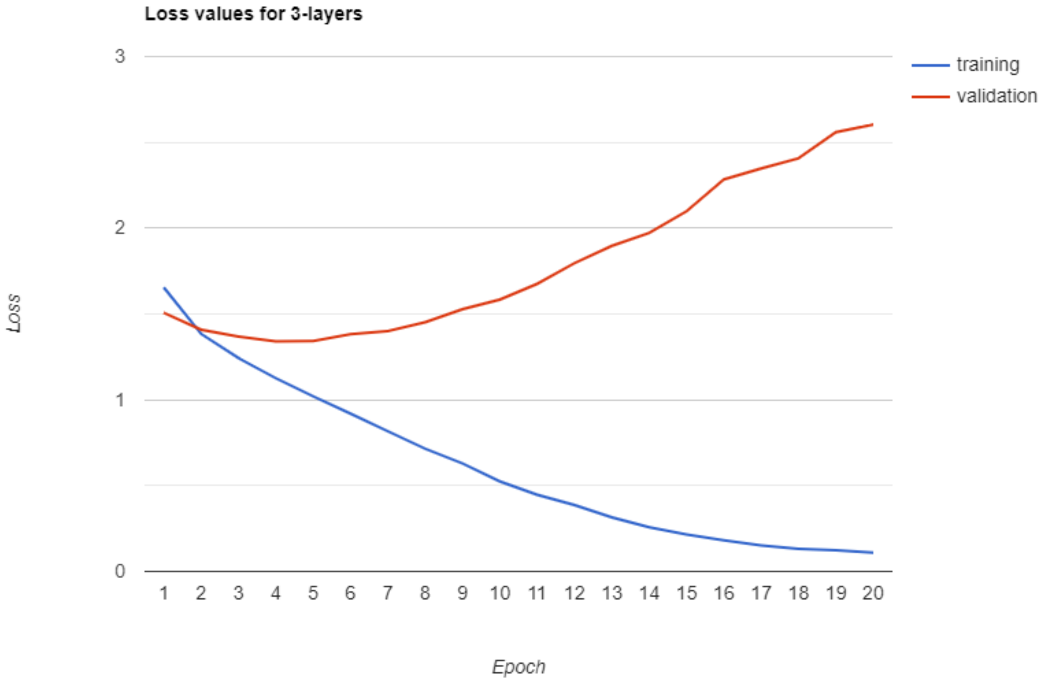
Test Results

For each k-layer network with best performing hyperparameter configuration, validation and training losses are shown in the following figure. Also, test accuracy of all different configurations is presented with tables at the end of the report.

Note: *To test the configurations, you can choose the layer numbers, activation function and learning rate in this order. The program has straightforward instructions.*




Loss values for 2-layers
2.0
1.5
1.0
0.5
0.0
Loss
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Epoch
training
validation


Loss values for 3-layers
3
2
1
0
Loss
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Epoch
training
validation

		Learning Rate				
		0.1	**0.03**	**0.01**	**0.003**	**0.001**
Activation function	**LeakyReLU**	29.4	32.11	33.75	38.41	40.36
	SELU	27.52	31.54	34.68	38.66	39.15
	GELU	27.53	30.43	33.65	38.29	39.32

1-layer Implementation

		Learning Rate				
		0.1	**0.03**	**0.01**	**0.003**	**0.001**
Activation function	**LeakyReLU**	26.81	37.06	46.44	50.4	51.49
	SELU	26.19	31.97	45.44	49.0	51.67
	GELU	24.68	35.68	48.59	51.16	52.43

2-layers Implementation

		Learning Rate				
		0.1	**0.03**	**0.01**	**0.003**	**0.001**
Activation function	**LeakyReLU**	38.65	49.4	51.27	52.41	54.14
	SELU	9.99	10.04	46.92	52.51	54.36
	GELU	10.01	9.99	48.92	53.06	54.84

3-layers Implementation