CS577 Assignment 2: Final report

Theo Guidroz
Department of Computer Science
Illinois Institute of Technology

February 12, 2021

Abstract

This is a report for Assignment 3 of CS 577. The assignment served to gain a deeper understanding of optimization, loss, and regularization.

Problem Statement

Build a model and evaluate the different loss functions, optimizers, and regularization techniques.

Implementation details

This implementation question was divided into 3 main parts:

1. Evaluating different optimization techniques
2. Evaluating different loss techniques
3. Evaluating different regularization techniques

To approach this problem, the 2 first tasked were merged. Models were trained using a combination of loss parameters and optimization parameters. The model that had the best evaluation on the testing set was recorded. Then, the combination which was more prone to overfitting was used to evaluate regularization techniques. Since the main goal of regularization is to help a model generalize, it made sense to help the most overfitting combination to generalize.

Loading the datasets

Both datasets used in this assignment are available on the Sklearn library. Iris was used for multi-class classification and the Boston house prices dataset was used for single output regression. Once the data was loaded, it was split into a training set and a testing set at a ratio of 7:3. The training set was split into a training set and a validation set at the same ratio. The data was then normalized and ready to be trained!

Loss function

The loss function is the function that computes the distance between the current output of the algorithm and the expected output. For this assignment, 6 different loss functions were covered, 3 designated to perform classification and 3 for regression.

Classification

1. Categorical crossentropy: This loss function calculates the loss of an example by computing the following sum:

$$\text{Loss} = -\sum_{i=1}^{\substack{\text{output}\\ \text{size}}} y_i \cdot \log \hat{y}_i$$

   where y^_i is the i-th scalar value in the model output, y_i is the corresponding target value, and the output size is the number of scalar values in the model output.

2. Categorical hinge: For a prediction y, take all y values unequal to t, and compute the individual losses. Eventually, sum them together to find the multiclass hinge loss.

$$\ell(y) = \sum_{y \neq t} \max(0, 1 + \mathbf{w}_y \mathbf{x} - \mathbf{w}_t \mathbf{x})$$

3. Kullback Leibler divergence: It is the expectation of the logarithmic difference between the probabilities P and Q, where the expectation is taken using the probabilities P.

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log\left(\frac{P(x)}{Q(x)}\right).$$

Regression

1. Mean squared error: Measures the average squared difference between the estimated values and the true values.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2.$$

   Where Y is the actual value Y^ is the predicted value, n is the number of data point.
2. Mean absolute error: Measures the absolute difference between the estimated values and the true values.

$$\text{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

3. Logcosh:

$$L(y, y^p) = \sum_{i=1}^{n} \log(\cosh(y_i^p - y_i))$$

Where yp is the true value and y is the predicted value.

Optimization techniques

Optimization is the problem of finding a set of inputs to an objective function that results in a maximum or minimum function evaluation. For this assignment, 3 different optimization parameters were covered.

1. Stochastic Gradient Descent: The weights are updated after every example
2. AdaGrad: Use element wise scaling of gradients based on history of gradients.
3. RMSprop: Similar to AdaGrad but adds a decay factor when adding new gradients to the gradient sum
4. Adam: Combines RMSprop with momentum

A combination of each loss function and each optimization technique was used to train the models and a graph combining the different combinations can be seen in the results section and discussion section.

Regularization techniques

There are several methods that can be used to generalize a model. Four of them were covered in this assignment.

1. Batch normalization: Normalize the features similar to normalizing the input dataset. This makes sure activations are not saturated and give equal importance to features with different scales
2. Dropout: At each training stage, drop-out units in fully connected layers with a probability of (1-p) where p is a hyperparameter.
3. Weight decay: Multiply each coefficient by pG[0,1]. As iterations progress, weights that are not reinforced decay to 9.
4. Ensemble: Train multiple independent models and use majority vote during testing.

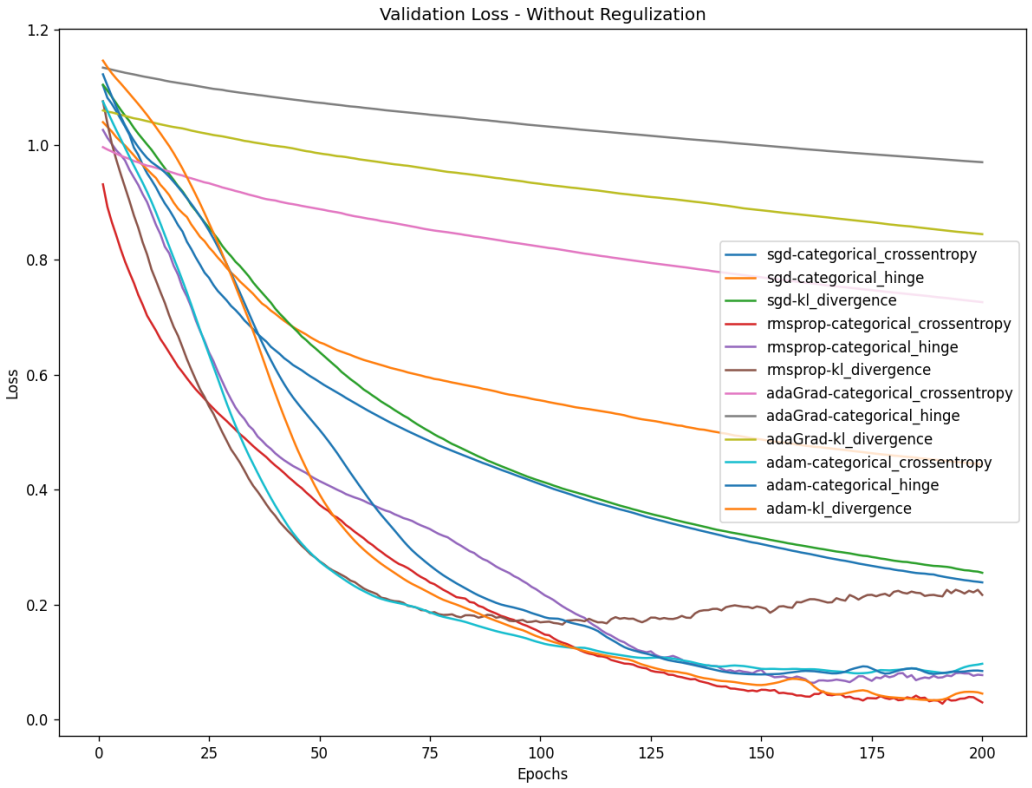
The general flow of actions to take to prevent overfitting is a follow:

1. Add batch normalization
2. Add dropout
3. Add weight decay
4. Create ensemble

The model which had overfitted the most during training underwent all 4 steps and a graph after each step can be seen in the next section.

Results and discussion

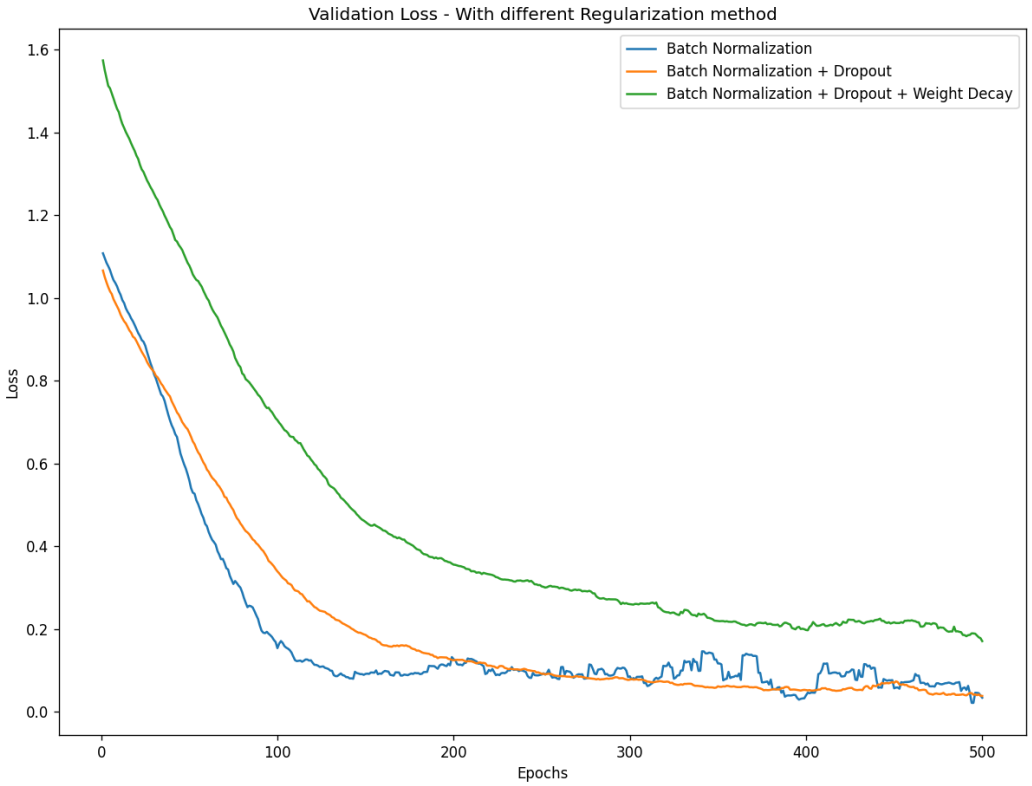
Classification



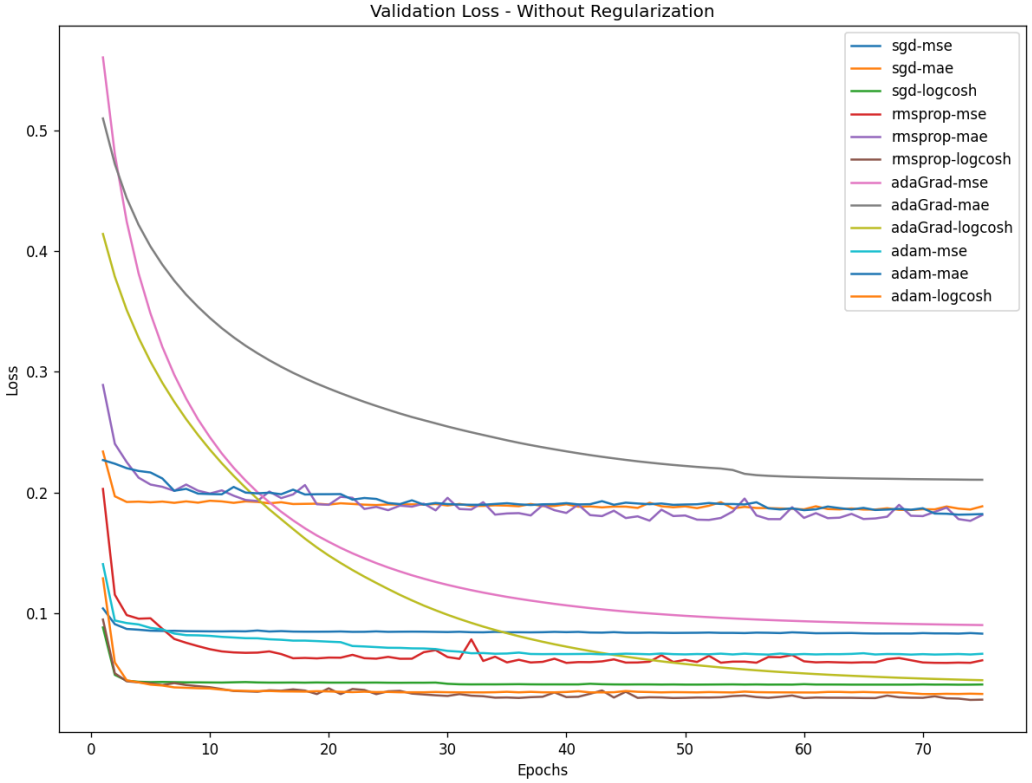
The graph above shows the training of each combination of loss and optimizers. The model that performed the best on the testing set was adam-categorical_hinge with a loss of 0.190 and an accuracy of 0.911.
As can be seen on the graph, the graph more susceptible to overfitting is RMSprop-kl_divergence. It has a loss of 0.249 and an accuracy of 0.911. The different regularization techniques were implemented to help it generalize.

Time taken

As seen in the graph, the model starts converging around 175 epochs. The combination that converges the fastest is the adam-kl diverge at a time of 5.42 seconds.

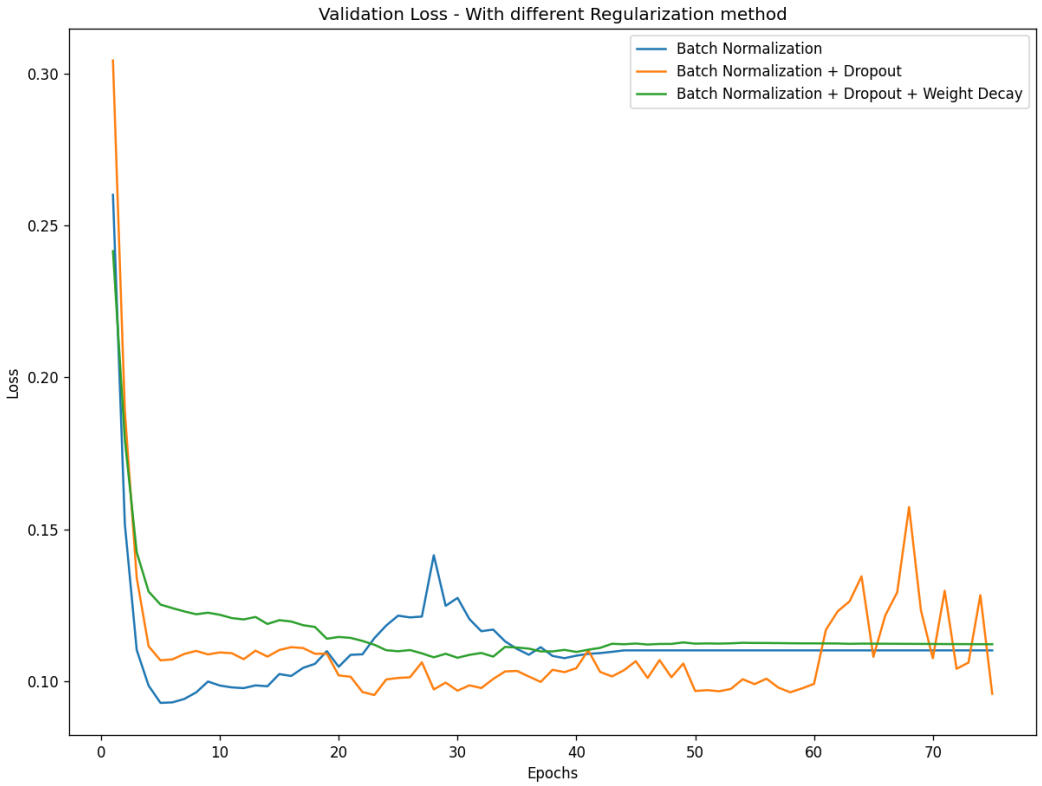


The graph above shows the different steps taken to reduce overfitting. As more steps were added, the model generalized better. The model that went through batch normalization and dropout performed the best on the test set with a loss of 0.144 and an accuracy of 0.956. This proves that batch normalization, dropout, and weight decay help to generalize a model. Ensemble was also implemented and the accuracy of this technique was 0.933. For assemble, majority technique was used.



The graph above shows the training of each combination of loss and optimizers. The model that performed the best on the testing set was RMSprop-logcosh with a loss of 0.02396 and an MAE of 0.169.

The combination RMSprop-mse was subject to overfitting. Therefore, regularization methods will be applied on it. It has a loss of 0.0512 and an MAE of 0.184.



The graph above shows the different steps taken to reduce overfitting. As more steps were added, the model generalized better.The model that only undergoes batch normalization still overfits however, the model where batch normalization, drop out and weight decay was implemented generalizes the best. The combination which worked best on the test set is batch normalization and drop out. It has a loss of 0.0847 and an MAE of 0.211. Ensemble was also produced and the MAE recorded was 0.156 which outperforms any independent model. The mean was used to perform ensemble.

Time taken

As seen in the graph, the model starts converging around 30 epochs. The combination of regularization techniques that converges the fastest is the batch normalization only at a time of 2.53 seconds.

Conclusion

Overall, this was a great way to understand how modifying the parameters such as the loss function, the optimizer, and the regularization can help improve the accuracy of the model and prevent it from overfitting.

References

Professor’s slides
https://scikit-learn.org/
https://iq.opengenus.org/types-of-loss-function/