CSCI 567 - Project Report
Exploring Complex Regularization Techniques on Image Classification and Sequence Labeling

Abhinav Gupta agupta67@usc.edu
Akshita Kapur kapuraks@usc.edu
Hrishikesh Thakur hthakur@usc.edu
Shreyas Malewar malewar@usc.edu

1 Project Proposal

1.1 What research question are you trying to answer?

Amongst the topics listed, our team was very fascinated by the sub-topic summary for (2a). This sub-topic talks about the effects of variation in models, network structure, datasets, and most importantly, regularization on the output of a research experiment. We want to take the oldest and most reliable path of trial and error to understand the concept of regularization through this project.

1.2 Why is this question interesting to you?

In CSCI567 homework assignments, we had the opportunity to experiment with various *regularization methods* and noticed that even a tiny constant added to complex models during training can significantly improve performance and prevent overfitting. While it's often believed that copious amounts of data are generated in the modern age, a more analytical perspective reveals that not all available data is useful. For instance, in a tumor segmentation problem where data acquisition is costly, using a complex, high-performing model might lead to overfitting, ultimately reducing its effectiveness. This prompts us to explore different regularization techniques to prevent overfitting and effectively utilize all available data.

1.3 What kind of data are you collecting or what datasets will you use?

We have taken 2 vastly diverse datasets - Image Classification and Natural Language Processing. In addition to experimenting with these 2 different datasets, we also plan to scale down each dataset and re-run all experiments to observe and document the results obtained of using regularization on smaller size datasets. A brief description of the datasets:

1. **tiny-imagenet**
   (a) **Dataset Summary** - Tiny ImageNet contains 100,000+ images of 200 classes (500 for each class) downsized to 64×64 colored images. Each class has 500 training images, 50 validation images, and 50 test images.
   (b) **Data Feature Dimensions** -
      i. Image: A PIL.Image.Image object containing the image.
      ii. Label: an int classification label. -1 for the test set as the labels are missing. Check classes.py for the map of numbers and labels.
2. **nltk-brown + nltk-treebank + nltk-conll2000**
   (a) **Dataset Summary** - The combination of these 3 datasets gives us a large corpus of textual data that can be used for training a model that performs sequence labeling with a total size of 72,000+ tagged sentences. The nltk library takes the base dataset and performs tokenization to prepare it for the task of sequence labeling.

(b) **Data Feature Dimensions** -
   i. Input Sequence - A sentence in english.
   ii. Output Sequence - POS tags of each word of the sentence.

1.4 What algorithms will you try?

From the tons of different regularization methods, we have selected 5 that will be applied to our 2 datasets in some combination. We will use a Classification model for the Image Classification task and a Sequence Labeling model for the NLP task. It is understood that in order to fit the model to these different datasets and regularization methods, cosmetic changes will be made to the models features and hyperparameters. A brief description of the different regularization methods:

1. **L2 Regularization** - modifies the loss function. Applied to both datasets.
2. **Data Augmentation** - modifies the data. For Dataset1 we plan to use **RandomErasing** - *RandomErasing* is concerned about removing and randomly adding information on the blank space, such as noise. For Dataset2 we plan to use **Random Synonym Replacement** - *Random Synonym Replacement* is concerned about removing and replacing with a synonym.
3. **MaxDropout** - modifies training approach. Applied to both datasets.
4. **Ensemble Regularization 1** - applying *RandomErasing* and *MaxDropout* together. Applied to Dataset1.
5. **Ensemble Regularization 2** - applying *RandomSynonymReplacement* and *MaxDropout* together. Applied to Dataset2.

1.5 What experiments and analysis will you run?

In our project, we aim to compare and analyze the use of different regularization methods on different types of datasets. We will apply the selected regularization techniques to the two datasets we've chosen. Additionally, we intend to experiment with combining different regularization methods to create ensemble regularization algorithms and check their impact on model performance. It is well known that regularization aids in mitigating overfitting, which often arises from insufficient data. Therefore, we intend to apply our regularization techniques to a subset of the two datasets and data sizes, to check if a certain regularization is good or the model is just performing well due to sufficient data.

1.6 What do you plan to finish by the midterm report

For the midterm report we plan to process our datasets and prepare the models that will be used for each of the tasks - *Image Classification* and *Sequence Labeling*. We also aim to start building the regularization methods for at least one of the two tasks by the midterm report.

2 Midterm Project Report

2.1 Which algorithms and baselines have you implemented?

Our experiment is split into two parts - testing the effect of different regularizations on two different types of real-world data, image and text. We first aimed to work on image classification. Tasks completed till now:

1. Loading Tiny-ImageNet dataset.
2. Processing the dataset to form tensors.
3. Implementing a pre-trained CNN model - EfficientNet B0 to perform the image classification task.
4. Building a Convolutional Neural Network (CNN) from scratch will be used for image classification. The CNN has the following structure:

(a) Convolutional Layers: Feature extraction using convolutions with filter sizes of 11x11, 5x5, and 3x3, comprising 64, 192, 384, 384, and 256 filters respectively, followed by ReLU activation.

(b) Normalization Layers: Incorporating local response normalization after specific convolutional layers, adjusting activations within local neighborhoods, with parameters depth radius of 5, bias of 2.0, alpha of 1e-4, and beta of 0.75.

(c) Pooling Layers: Employing max-pooling with pool sizes of 3x3 and strides of 2 for downsampling, reducing spatial dimensions while maintaining crucial features.

(d) Flattening: Transforming convolutional layer outputs into one-dimensional vectors for seamless integration into fully connected layers.

(e) Fully Connected Layers: Dense layers with 1024 neurons each, activated by ReLU, facilitating complex mappings between flattened features and output classes, culminating in a softmax activation layer for classification.

5. Defined and implemented 4 types of regularization methods to test on our CNN, namely:

   (a) L2

   (b) Data Augmentation - Horizontal Axis Flipping/Rotation

   (c) MaxDropout

   (d) Ensemble - L2 + MaxDropout

6. Additionally, we have started working on our sequence classification task's dataset - NLTK- and designing the model architecture.

2.2 What challenges did you encounter? How did you solve it or how are you planning on solving it?

1. **Deciding the Baseline Model**

   - This is a very important step as the baseline model provides the reference point against which the performance of more complex regularization methods is compared. This is essential to evaluate the effectiveness of our implemented regularization techniques in improving model performance.
   - **Solution:** Propose a model from scratch and train the data using that model, which will allow us to control each individual layer. We have complete control that helps us accurately apply regularization techniques.

2. **Model Tuning**

   - We have implemented a pre-trained CNN model - EfficientNet B0 that is trained for the image classification task. Additionally, we have also built a CNN model from scratch. We can only continue with one model.
   - **Solution:** We are in the process of performing hyper-parameter tuning and optimization of both of these models. We plan to pick the better one.

2.3 What would you like to complete by final report? What are additional experiments you would like to include if there is time?

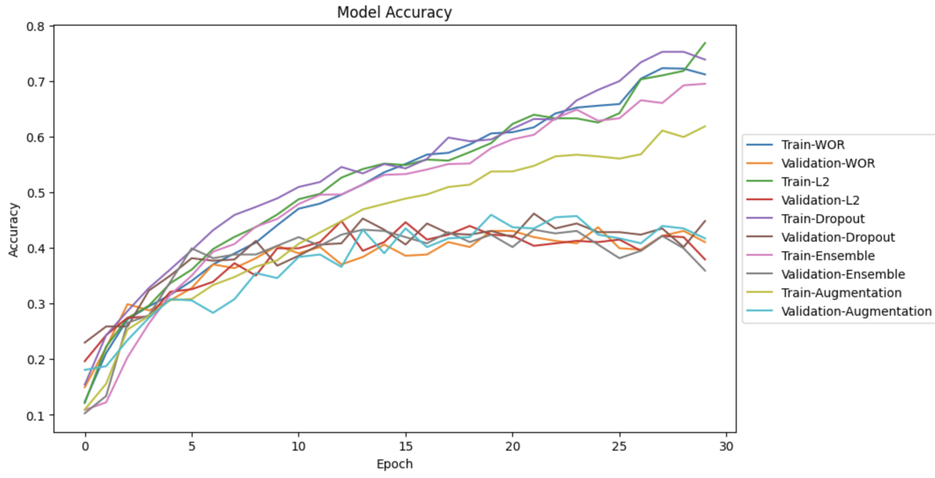
1. We will complete the image classification task and report our results and analysis about the effect of regularization on the model's performance.

2. Our next primary objective is to conduct seq2seq labeling task while exploring the impact of different regularization techniques on text data. Specifically, we aim to assess how the regularized models handles the complexities inherent in textual data.

3. Furthermore, we will conduct comparative analysis across the different datasets and models.

2.4 Discuss preliminary results and analysis if available or analysis proposed.

1. **Comparative Analysis of regularization on Baseline CNN Model:**



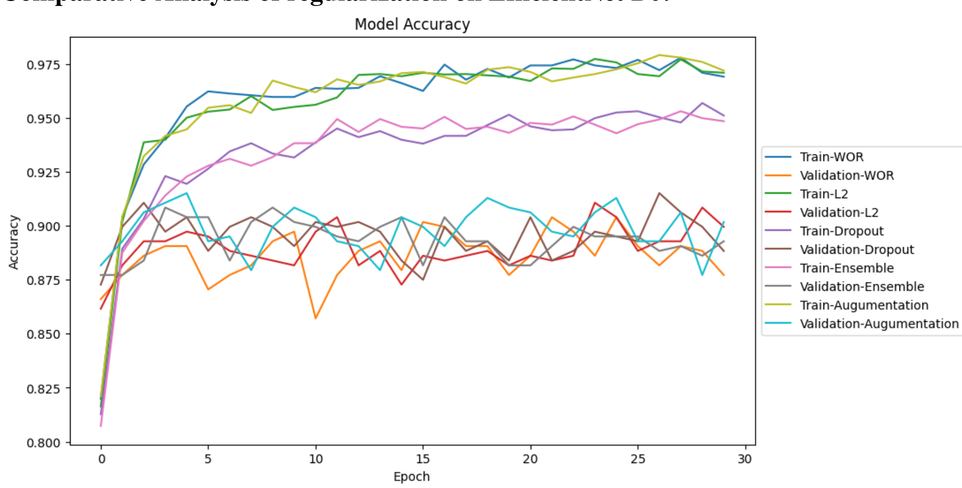
   (a) **L2 Regularization** - Here, both train and Validation accuracies perform similarly to the model without regularization.

   (b) **Dropout** - The model with dropout shows improved train and Validation accuracies, though the generalization gap remains similar, indicating improved performance.

   (c) **Augmentation** - Data augmentation contributes to improved generalization. Validation accuracy decreases while test accuracy increases compared to the model without regularization.

   (d) **Ensemble** - The ensemble model showcases improved generalization. Validation accuracy decreases, resulting in a smaller generalization gap compared to the individual models.

2. **Comparative Analysis of regularization on EfficientNet B0:**



   (a) **L2 Regularization** - Both train and validation accuracies perform similarly to the model without regularization.

   (b) **Dropout** - Adding dropout improves generalization. Train accuracy decreases, resulting in a smaller generalization gap compared to the individual model.

   (c) **Augmentation** - There is a slight improvement in validation accuracy, leading to better generalization.

   (d) **Ensemble** - Using ensemble methods helps the model generalize better, as evidenced by the reduction in test accuracy.

2.5 Any changes in scope or project direction?

Our project is right on track with our proposed timeline. However, there are a couple of small changes we have made:

1. To mitigate the computational burden inherent in the image classification task, we opted to downscale the dataset from 200 classes to 10 classes, with each class comprising 500 samples. This reduction in dataset size was necessary to allow us to perform easier debugging and model training. Once the model is finalized, we can re-run on a scaled-up dataset version.
2. We revised the ensemble regularization approach for both the ImageNet and NLTK datasets. Changed L2 + DataAugmentation to L2 + MaxDropout. Upon further literature review, we realized that the former is counter-productive. One introduces variability to the training data, making the training landscape more intricate, while the other simplifies model weights for easier training. Whereas the latter is synergetic. Dropout randomly deactivates nodes during training, enhancing feature exploration and diversification, and L2 stabilizes training by controlling the sizes of weights.