# Hyperparameters Tuning and Model Optimization of Neural Network Architecture in Myanmar Stemmer

Yadanar Oo
Natural Language Processing Lab
University of Computer Studies, Yangon
Yangon, Myanmar
yadanaroo@ucsy.edu.mm

Khin Mar Soe
Natural Language Processing Lab
University of Computer Studies, Yangon
Yangon, Myanmar
khinmarsoe@ucsy.edu.mm

***Abstract*—Neural network architectures have many hyperparameters that can be tuned to achieve optimal performance for a given data set. Moreover, the selection of optimal parameters makes a significant difference to conventional designs and is state of the art in many research areas. In this paper, we evaluate the importance of various hyperparameters in the Myanmar neural sequence labeling task. Performance evaluation was performed based on various hyperparameters such as: different optimizers, different learning rates and dropout rates for neural network architectures against 30,000 data from Myanmar News. It also provides configuration recommendations suitable for our particular dataset.**

***Keywords—Myanmar stemmer, neural network architecture, hyperparameters.***

## I. Introduction

In Myanmar Language, texts usually contain various forms of roots. In general, morphological variants are the most common problems with spelling errors, translation errors, and irrelevant searches. Since Myanmar written language does not use spaces to indicate word boundaries, segmenting Myanmar texts becomes an essential task for Myanmar language processing. Besides word segmentation, it is necessary to identify the stem word in the sentence. Stemming refers to the process of marking each word in the word segmentation result with a correct word type, for example, root word, single word, prefix, suffix, etc.

Stemming is usually considered a separate process from segmentation. This new approach integrates segmentation and stemming as a lexical analysis system. This integration of segmentation and stemming benefits both processes. There are many stemmers for major languages, but no one for our language. The main reason is to create Myanmar stemmer, which also solves the problem of word segmentation. This is the first work on joint segmentation and stemming in Myanmar.

Moreover, deep learning approaches to NLP tasks are becoming more and more popular today. This system deals with a CNN-BiLSTM-CRF network architecture that learns two processes together. Although it is generally accepted that hyperparameters selection plays an important role in the widespread use of neural network architectures, little research has been published on hyperparameters evaluation to date.

This paper evaluates different types of hyperparameters, including optimizers, learning rates and dropout rates. The main contribution of this paper is a detailed analysis of hyperparameters for optimizing Myanmar-Stemmer models. To do this, we evaluated 30,000 data from Myanmar News on CNN-BiLSTM-CRF architecture for the sequence labeling task. We used the NCRF++ toolkit [4] to create an architecture for labeling neural sequences of joint processes in Myanmar. Experimental settings are trained and discussed on Nvidia Tesla K80 GPU servers. Training takes about 18 hours, whereas for CoNLL 2003 marking the test set takes about 60 seconds.

## II. Neural Sequence Labeling Model

Most of the NLP processes such as word segmentation, part-of-speech tagging and named entity detection have been improved significantly from the earliest dictionary based approach to CRF approaches with handcrafted features and task-specific resources. With advances in deep learning, neural models have given state-of-the-art results on many sequence labeling tasks. In general, many existing neural sequence labeling models utilize word-level structure to represent global sequence information inference layer to capture dependencies between neighboring labels.

Neural sequence labeling framework contains three layers; a character sequence representation layer, a word sequence representation layer and an inference layer, as shown in figure 1.

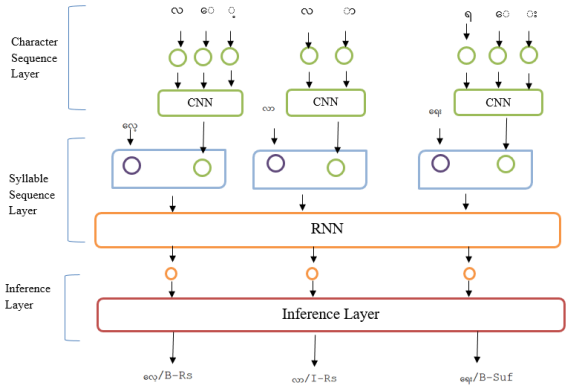


Fig. 1. Neural sequence labeling model for word "ဇလုပ်ခရေး". Green, purple, blue and orange represent character embeddings, syllable embeddings, character sequence representations and syllable sequence representations, respectively.

## III Experimental Results

In Myanmar language, there is no standard corpus so use a training set selected from manually segmented 30K and divide the training corpus into two sets, the first 80% of the data to training and 10% each to development and test set. In this part, joint model is evaluated on different setups like the

importance learning rate, dropout rate and different kinds of optimizers that have a large impact on the performance.

### 1. Optimizers

During the training process, the optimizer tunes and modifies the model's parameters to minimize the loss function and make the best possible predictions. There is general agreement that optimizer choice plays an important role. The optimizer is responsible for minimizing the neural network's loss function. A frequently chosen optimizer is Stochastic Gradient Descent (SGD). Finding the best optimizer is not easy. In this section, we find the best optimizers among Adagrad, Adadelta, Adam, RMSProp, and SGD.

Table 1: Comparison of different Optimizer

| Optimizers | *Precision* | *Recall* | *F-Score* |
|---|---|---|---|
| **SGD** | **89.59** | **88.45** | **89.40** |
| Adagrad | 88.82 | 88.72 | 88.77 |
| Adadelta | 89.01 | 88.89 | 88.95 |
| Adam | 88.78 | 88.57 | 88.67 |
| RMSProp | 88.03 | 87.87 | 87.95 |

According to the experimental result, SGD optimizer gets the best result among different optimizers.

### 2. Dropout Rate

Dropout is more effective on overfitting problems where there is a limited amount of training data and the model is likely to overfit the training data. Myanmar is low resource language. Thus, for the task of joint word segmentation and stemming in Myanmar Language, different dropout rates are used to tune upon previous best optimizer SGD.

Table 2: Comparison of different dropout rate

| Dropout Rate | *Precision* | *Recall* | *F-Score* |
|---|---|---|---|
| 0.1 | 89.84 | 89.64 | 89.74 |
| **0.2** | **89.75** | **91.40** | **90.56** |
| 0.3 | 89.04 | 90.12 | 89.57 |
| 0.4 | 89.35 | 89.69 | 89.52 |
| 0.5 | 89.59 | 88.49 | 89.40 |
| 0.6 | 87.21 | 88.57 | 87.89 |
| 0.7 | 75.35 | 77.45 | 76.38 |
| 0.8 | 68.16 | 63.92 | 65.98 |
| 0.9 | 59.42 | 48.07 | 53.15 |

As stated in the table, tuning different dropout rate with SGD optimizer between 0.1 to 0.9, the best F-Measure is dropout 0.2 (90.56%). To be conclude, dropout rate 0.0 to 0.5 can give the best results. The larger the dropout rate, the lower performance we get.

### 3. Learning Rate

The learning rate controls how much the model changes in response to the estimation error each time the model weights are updated. Choosing a learning rate is a difficult task. Too small a value can lengthen the system's training process, while too large a value can cause a suboptimal set of weights to be learned too quickly or the training process to be unstable.

Table 4: Comparison of different Learning Rate

| Learning Rate (lr) | *Precision* | *Recall* | *F-Score* |
|---|---|---|---|
| lr 0.001 | 88.16 | 87.48 | 87.82 |
| lr 0.002 | 89.44 | 89.36 | 89.40 |
| **lr 0.003** | **91.36** | **91.14** | **91.25** |
| lr 0.004 | 89.59 | 88.49 | 89.40 |
| lr 0.005 | 90.55 | 89.85 | 90.20 |
| lr 0.006 | 90.40 | 89.39 | 89.89 |
| lr 0.007 | 90.98 | 90.08 | 90.53 |
| lr 0.008 | 87.55 | 87.58 | 87.56 |
| lr 0.009 | 90.59 | 90.14 | 90.37 |

Experimental results show that the choice of learning rate has impact on system performance. A learning rate of 0.003 gives the best performance compared to the others.

## IV Conclusions

This paper focus on the experimental results of different hyper parameter tuning on neural sequence labeling for Myanmar stemmer. According to the experimental result, hyperparameters are important to optimize the model and it has a significance impact on the performance result. The performance improve nearly 2% because of tuning hyperparemeters. Moreover, we run on both Nvida Tesia K80 GPU and MacBook Pro 8-core CPU and 14-core GPU. Firstly, running speed is different. When using GPU, it takes 18hours for training. With M1 Pro, it takes, more than 25 hours. But the performance result is not very different.

## REFERENCES


1. Thu, Y.K., Finch, A., Sagisaka, Y., Sumita, E.: “A Study of Myanmar Word Segmentation Schemes for Statistical Machine Translation”. In Proceedings of 12th International Conference on Computer Applications, Yangon, Myanmar, pp.167-179, 2014
2. W.P.Pa, Y.K.Thu, A.Finch and E.Sumita, “Word Boundary Identification for Myanmar Text Using Conditional Random Field”, Springer, Switzerland, 2016
3. Nils Reimers and Iryna Gurevych. “Optimal hyperparameters for deep lstm-networks for sequence labeling tasks.” 2017a,,arXiv preprint arXiv:1707.06799.
4. Jie Yang and Yue Zhang. NCRF++: An Open-source Neural Sequence Labeling Toolkit. arXiv:1806.05626v2 [cs.CL] 17 Jun 2018.
5. Jie Yang, Shuailong Liang, and Yue Zhang. “Design challenges and misconceptions in neural sequence labeling”. In COLING, 2018.