COSE474 Deep Learning ASSIGNMENT3 REPORT

MuYeong Jeong
Korea University
2019320112
jmy3033@naver.com

Abstract

In this assignment, I address a text classification task using recurrent neural network architectures. Specifically, I implement two models: a vanilla RNN and a Gated Recurrent Unit (GRU). The dataset, obtained from Kaggle, comprises 2,225 text samples labeled across five distinct categories. The objective is to develop models capable of accurately classifying each text instance into its corresponding category based on its content.

1. Implementation

I will explain the implementation process step by step.

1.1. Vanilla RNN

$$h_t = \tanh(W_x x_t + W_h h_{t-1} + b_h) \quad (1)$$
$$y = Fully\ Connected\ Layer(h_t) \quad (2)$$

The vanilla RNN is the most basic type of RNN architecture. It computes the current hidden state by taking the input at the current time step along with the hidden state from the previous time step. Since the objective of the task is classification, the final hidden state is passed through a fully connected layer to perform the classification.

1.2. GRU

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \quad (1)$$
$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \quad (2)$$
$$\hat{h}_t = tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h) \quad (3)$$
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t \quad (4)$$
$$y = Fullt\ Connected\ Layer(h_t) \quad (5)$$

GRU is a type of RNN designed to handle sequential data and avoid the vanishing gradient problem by using gating mechanism.

The important part of mitigating vanishing gradient problem is $(1 - z_t) \odot h_{t-1}$. GRU mitigates vanishing gradients by allowing part of the previous hidden state to be directly passed through time steps via gating mechanisms, without being fully transformed by non-linear activations.

z_t is called update gate. The update gate determines how much of the previous hidden state h_{t-1} should be retained versus how much of the candidate hidden state $\hat{h}_t$ should be used to update the current hidden state h_t.

r_t is called reset gate. The reset gate controls how much of the past hidden state should be considered when computing the candidate hidden state. A lower value of r_t effectively resets the memory, allowing the model to discard irrelevant historical information.

$\hat{h}_t$is called candidate hidden state. This represents the potential new hidden state, calculated based on the current input and a gated version of the previous hidden state.

The final hidden state h_t is a convex combination of the previous hidden state and the candidate hidden state, where the update gate determines the contribution of each.

2. Result

For the vanilla RNN model, training was conducted over 10 epochs. The model achieved a final training loss of 0.2799. However, its performance on the validation set was limited, with a validation loss of 2.1882 and an accuracy of 38.02%. On the test set, the model reported a loss of 1.8908 and an accuracy of 43.11%, indicating suboptimal generalization performance.

In contrast, the GRU model was trained for a total of 20 epochs. At epoch 10, it achieved a training loss of 0.0561, a validation loss of 0.9176, and a validation accuracy of 72.46%. At epoch 20, the training loss further decreased to 0.0019, with a validation loss of 1.1078 and a validation accuracy of 76.05%. Notably, even at epoch 10, the GRU model already outperformed the vanilla RNN by nearly a factor of two in terms of validation accuracy. The final performance on the test set was a loss of 1.1396 with an accuracy of 76.05%.

This performance difference can be attributed to the fact that GRU, unlike vanilla RNN, employ gating mechanisms that mitigate the vanishing gradient problem, enabling more effective learning of long-term dependencies.