5043 Advanced Machine Learning - HW 6

Author: Enzo Durel

April 22, 2025

Contents

List of Figures

1 Figures

1.1 Figure 0a

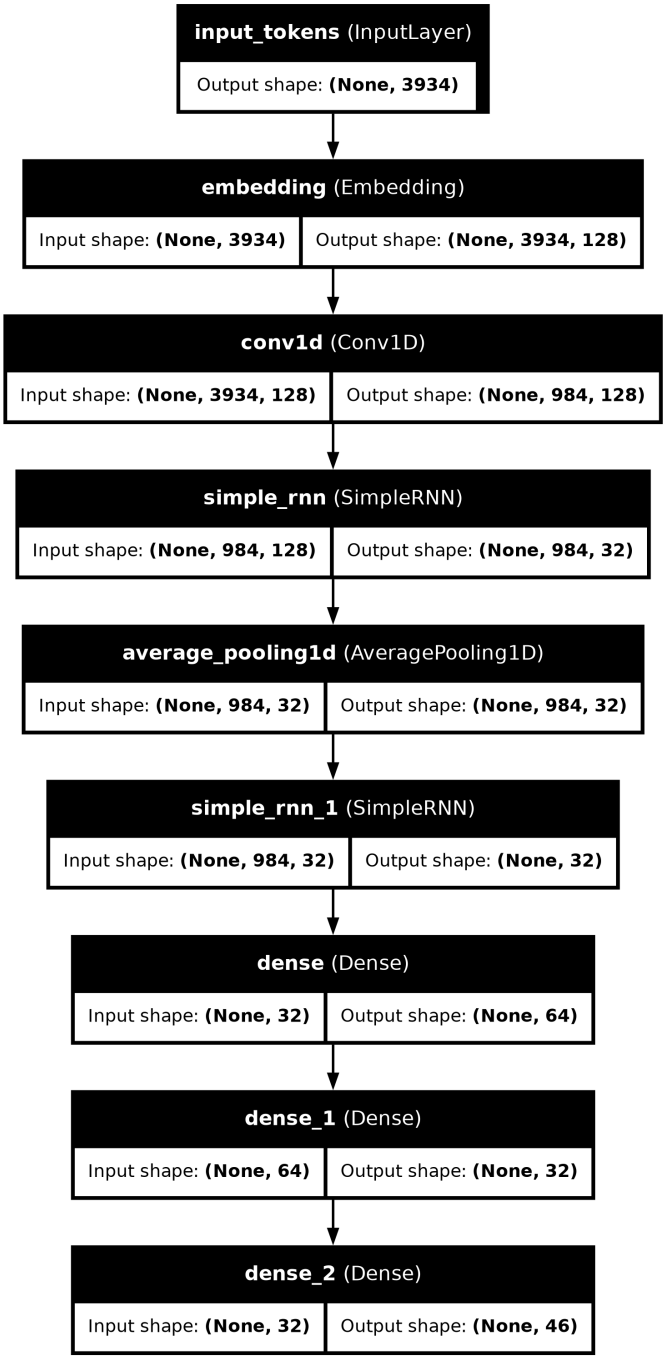

Fig. 1: RNN Model

1.2 Figure 0b

Fig. 2: GRU Model

1.3 Figure 0c

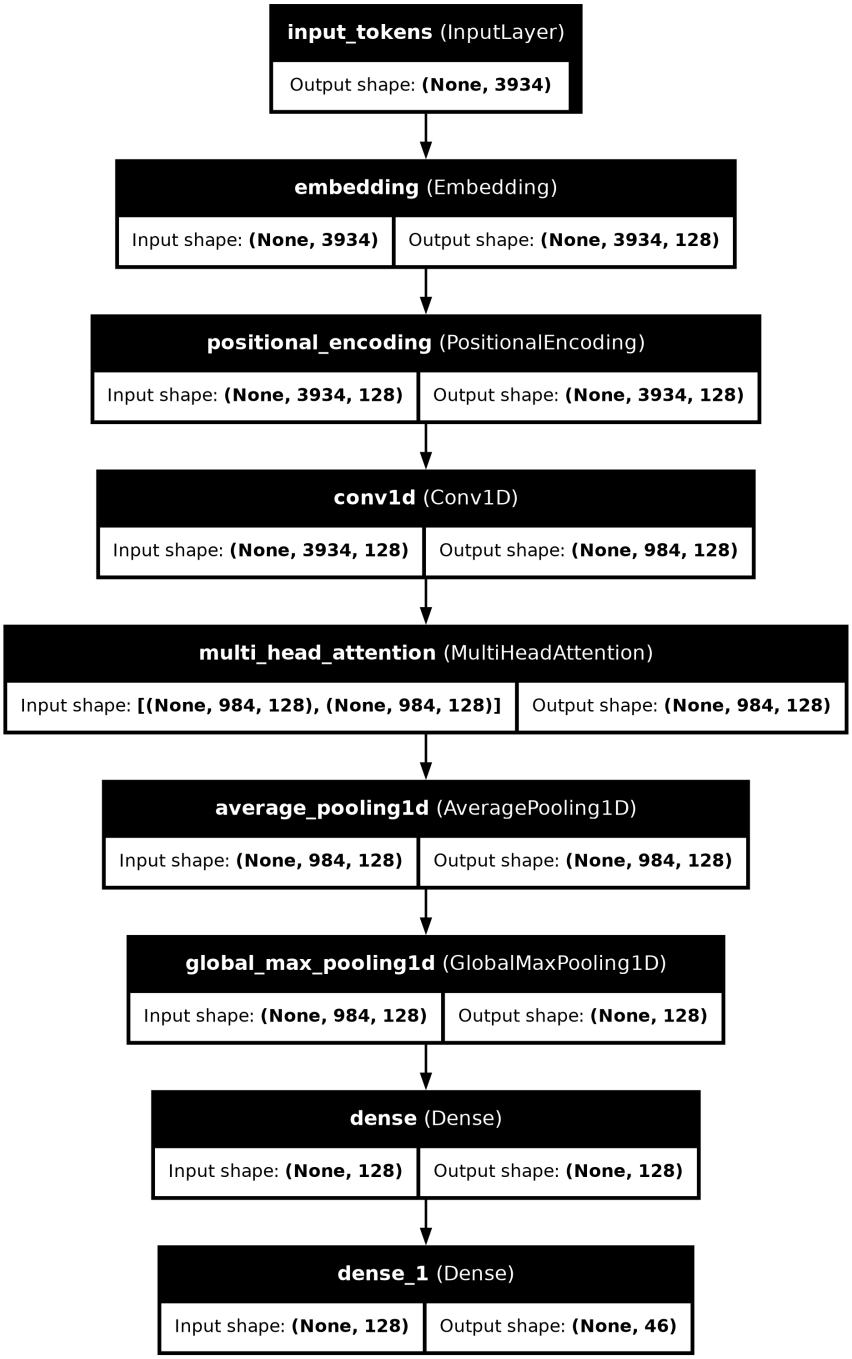

Fig. 3: Attention Model

1.4 Figure 1a

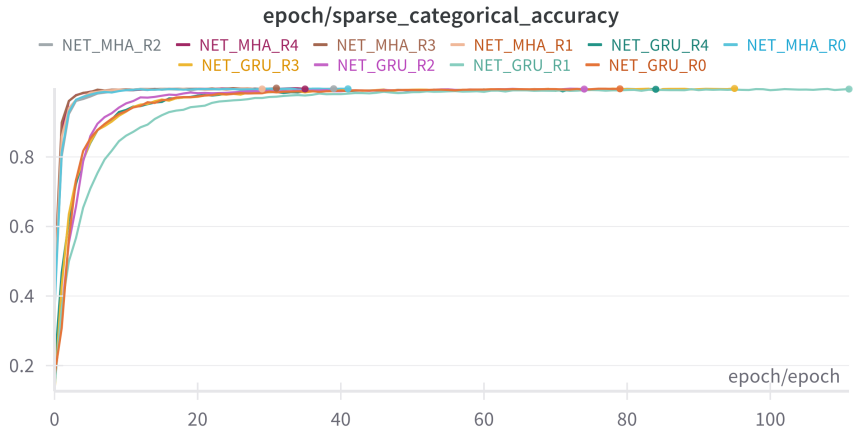

Fig. 4: Training Loss

1.5 Figure 1b

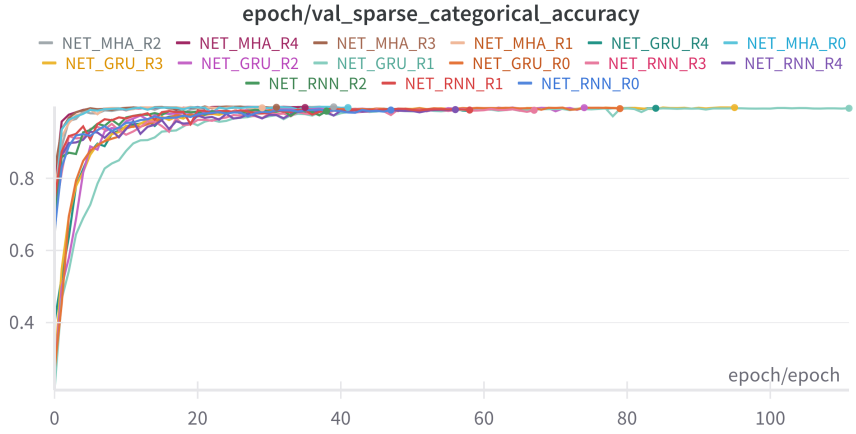

Fig. 5: Validation Loss

1.6 Figure 2

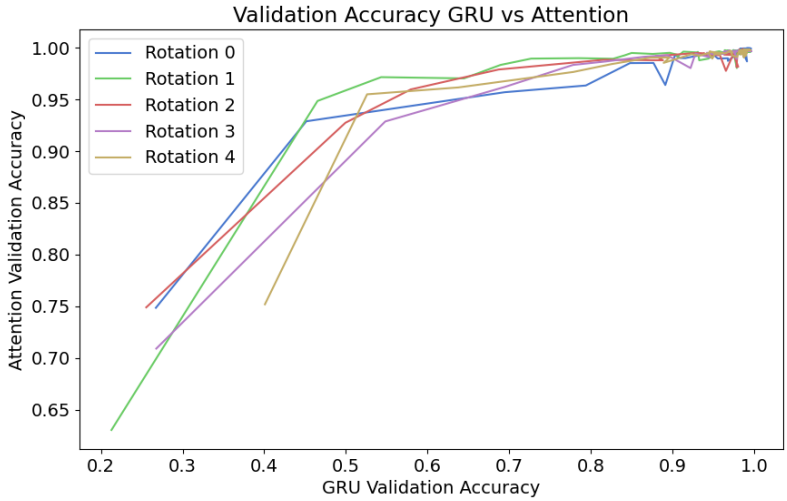

Fig. 6: Validation accuracy GRU vs Attention

1.7 Figure 3

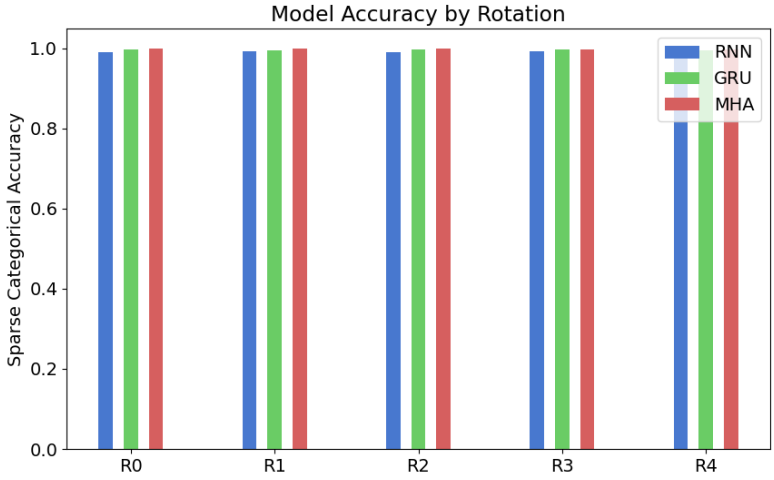

Fig. 7: Models Accuracy Across Rotations

1.8 Figure 4

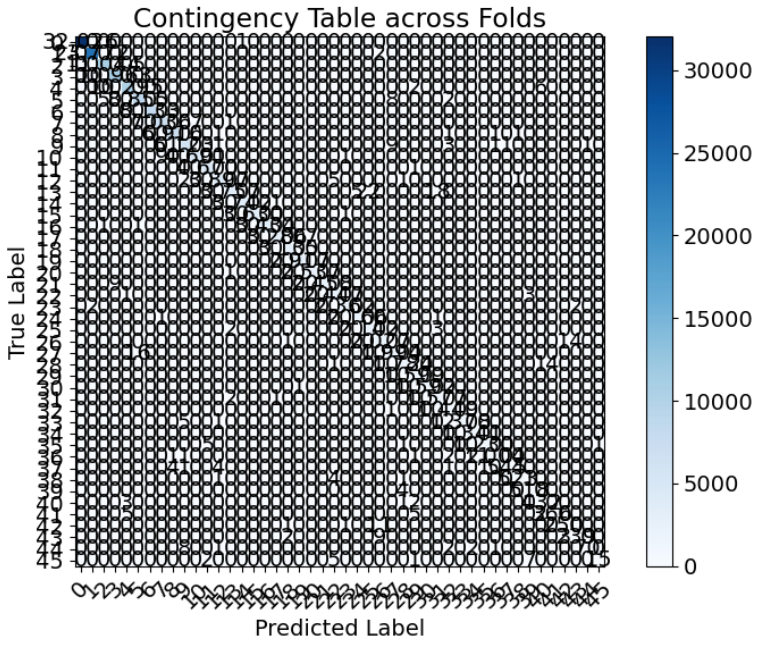

Fig. 8: MHA Test Contingency Table across the five rotations

2 Analysis & Discussion

2.1 Describe the specific choices you made in the model architectures.

2.1.1 RNN & GRU

The RNN and GRU follow the same architecture. Only the RNN part is replaced with GRU layers. First, we have the embeddings, then we have the convolution as we see in class.

About the RNN layers, I only use 1 inner RNN layer with the `return_sequence` parameter as false to keep dimensions. I designed my build model function as we can use as much RNN layers as we want. I firstly testing with one inner layer and it works great so I decided to keep this architecture.

We always have a "final" RNN layer to make the link with the hidden layers. Then we only have 2 dense layers. Same approach, I begin small and it works well so there was no need to scale up.

2.1.2 MHA

Concerning the MHA, I follow the same approach. First the embedding layer followed by the positional encoder. Then, the convolutional layer use to "skip" connections.

There is only one multi head attention layer which was working perfectly so I keep it small too. We've got the average pooling followed by the global pooling to "filter" the parameters.

Finally, we've got the same hidden layers as RNN & GRU. Only 2 small layers of size 64 and 32.

2.2 Discuss in detail how consistent your model performance is across the different rotations.

We can see in figure 1a and 1b that all accuracy curves follow the same tendance independantly of the rotation.

However, we can see that the RNN layer validation accuracy curves are less stable than the GRU and MHA ones. We can also see that in the rotation 1, the GRU model is "lower" than the other GRU models.

MHA models are really stable and almost identical.

2.3 Discuss the relative performance of the three model types.

In figure 2, we can that the MHA accuracy is getting quicly better but converge with the GRU accuracies. The GRU accuracy is getting converge slower but get almost the same accuracy at the end.

In figure 3, we can see that all the models have almost the same final accuracy independantly of the rotation. However, we can see that MHA is slightly better in every rotation.

2.4 Compare and contrast the different models with respect to the required number of training epochs and the amount of time required to complete each epoch. Why does the third model type require so much compute time for each epoch?

For each models, I used steps per epoch equal to 100.

For RNN layer, I've got in average, 37s per epochs For GRU layer, I've got in average, 14s per epochs For MHA layer, I've got in average, 10s per epochs.