Preferred Networks Intern Screening 2019 Coding Task for Machine Learning / Mathematics Fields

Ryo Kamoi
ryokamoi.jp@gmail.com

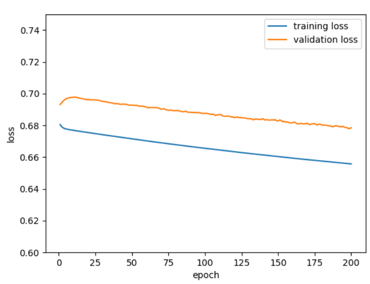
1 Optimization Methods

I compared three optimization methods (SGD, Momentum SGD, Adam) with GNN (Table 1, Figure 1). First, we can see that the training with SGD is much slower than the other two methods, and it did not converge in 200 epochs. Momentum SGD and Adam achieved almost the same loss and accuracy, but Momentum SGD converged faster than Adam in my setting. However, it is observed that the curves of loss and accuracy by training with Adam are more stable than that by Momentum SGD.

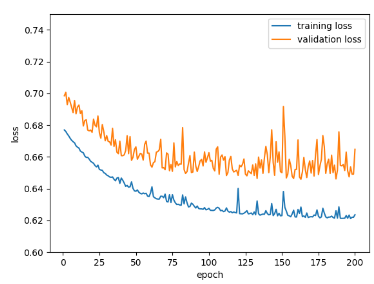
These results are performed with learning rate 0.0001, but I have also observed similar behaviors for different learning rates. Other parameters are based on "Hint of Hyperparameters".

		SGD	Momentum	Adam		SGD	Momentum	Adam
training	loss	0.656	0.628	0.629	accuracy	0.622	0.643	0.666
valudation	loss	0.678	0.646	0.655	accuracy	0.584	0.594	0.610

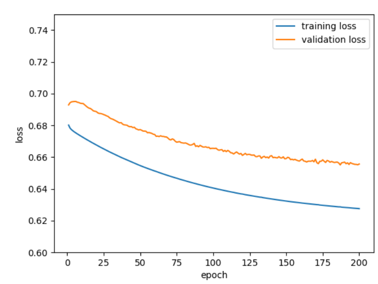
Table 1: Loss and accuracy by different optimization methods



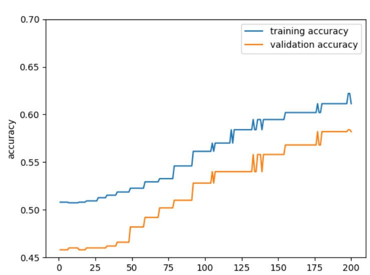
(a) SGD (Loss)



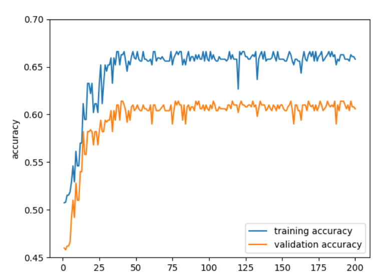
(b) Momentum SGD (Loss)



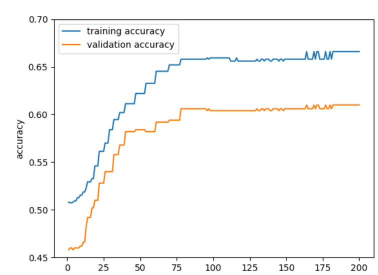
(c) Adam (Loss)



(d) SGD (Accuracy)



(e) Momentum SGD (Accuracy)



(f) Adam (Accuracy)

Figure 1: Comparison of different optimization methods (lr=0.0001)

2 Activation Function

Duvenaud et al. [1] reported that ReLU is slightly better than tanh as an activation function for aggregation. However, tanh yielded a much better result than ReLU in my setting (Table 2). The sigmoid function is also tried, and result is worse than that by tanh, but better than that by ReLU (the validation accuracy is 0.694).

3 Multi-Layer Perception

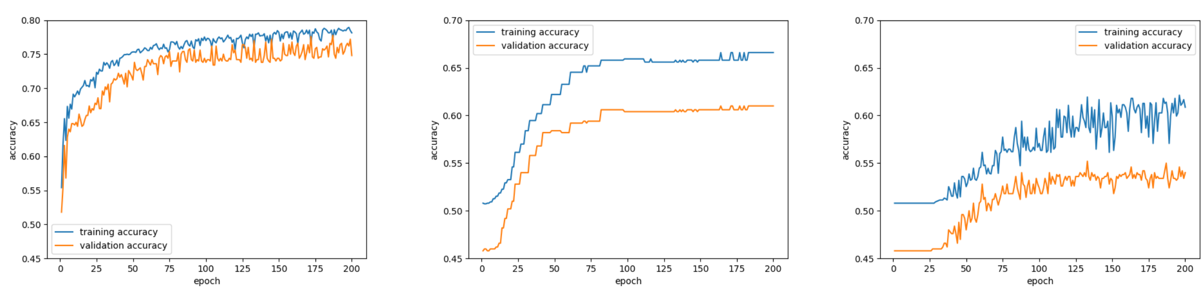
Increasing the number of layers for aggregation did not improve the result in my setting (Table 2). This result may imply that the representation power of aggregation step is not an important problem in this task.

4 Super-Node

It was observed that the super-node harms the result in this task (Table 2). One possible reason is that the super-node is not compatible with this classification task. We should check what kinds of properties of the graphs are corresponding to the labels in the dataset.

		tanh	MLP	Super-Node		tanh	MLP	Super-Node
training	loss	0.460	0.629	0.678	accuracy	0.785	0.666	0.614
valudation	loss	0.481	0.655	0.683	accuracy	0.772	0.610	0.550

Table 2: Results for modifications



(a) tanh (different axis range)

(b) MLP

(c) Super Node

Figure 2: Comparison of Modifications for GNN (Accuracy)

References

[1] David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional Networks on Graphs for Learning Molecular Fingerprints. In *Conference on Neural Information Processing System*, 2015.