COSC 189.31 HW1 Report
Ivory Yang

The results you obtain on each model (i.e., loss, accuracy, margin, charts, measures, and bounds) and your understanding of the bounds of the two NNs you trained.

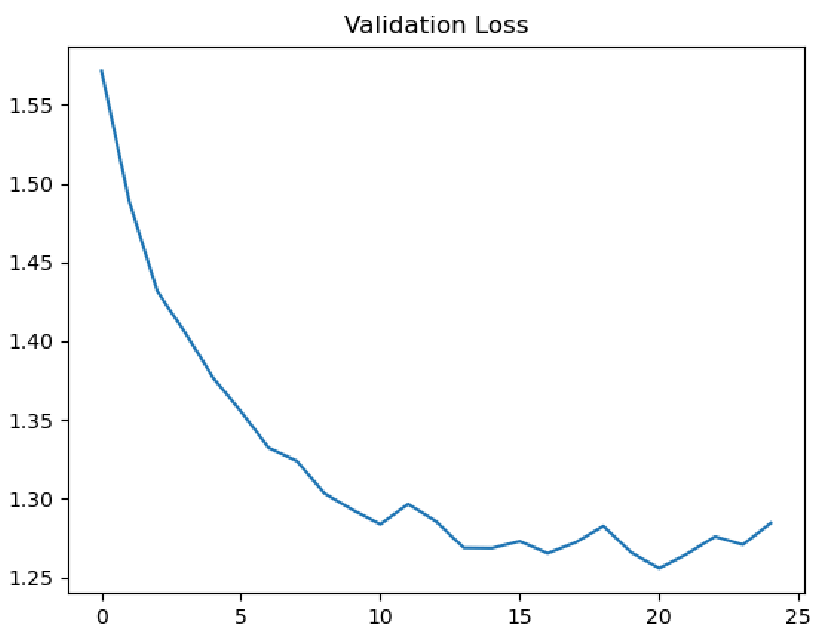
Model 1

Results:

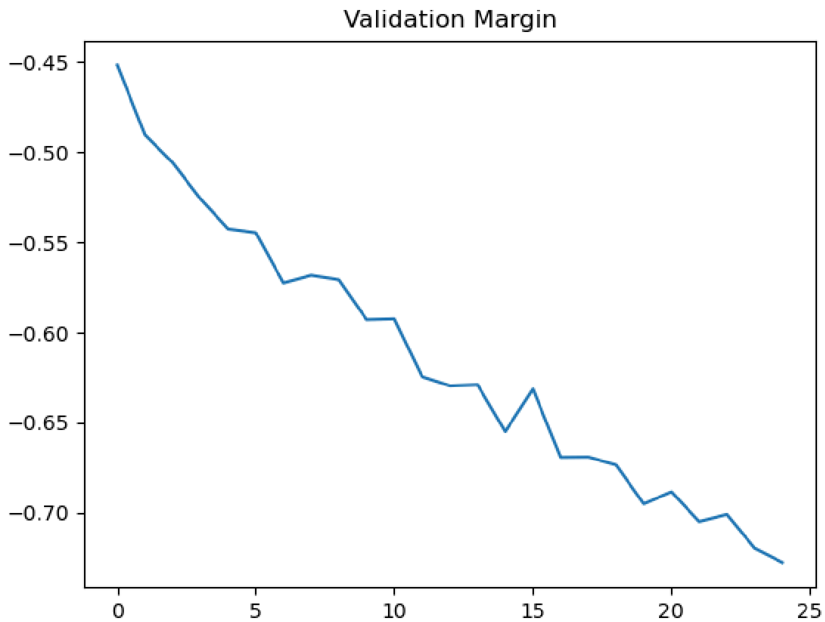
```
================== Summary ==================
 Training loss: 0.821   Training margin -0.509    Training accuracy: 0.726   Validation accuracy: 0.568

Frobenius1:      20.7
Frobenius2:      6.72
Distance1:       9.0
Distance2:       5.44
Spectral1:       1.93
Spectral2:       3.12
Fro_Fro:         1.39e+02
L1max_L1max:     2.27e+03
Spec_Dist:       33.3
Dist_Spec:       9e+02
Spec_Dist_sum:   9.33e+02
Spec_L1max:      33.1
L1max_Spec:      8.78e+02
Spec_L1max_sum:  9.11e+02
Dist_Fro:        60.5
#parameter:      3.16e+06
VC bound:        9.19e+09
L1max bound:     4.46e+11
Your bound:      1.84e+11
(base) ivoryyang@Ivorys-MacBook-Air Ivory Yang HW1 %
```

Validation Loss Plot:



Validation Margin Plot:



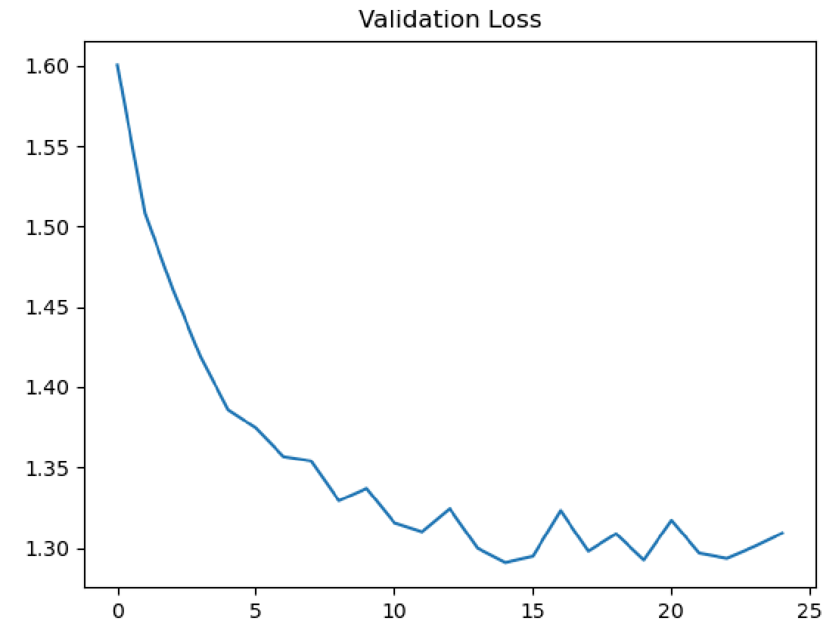
Model 2

Results:

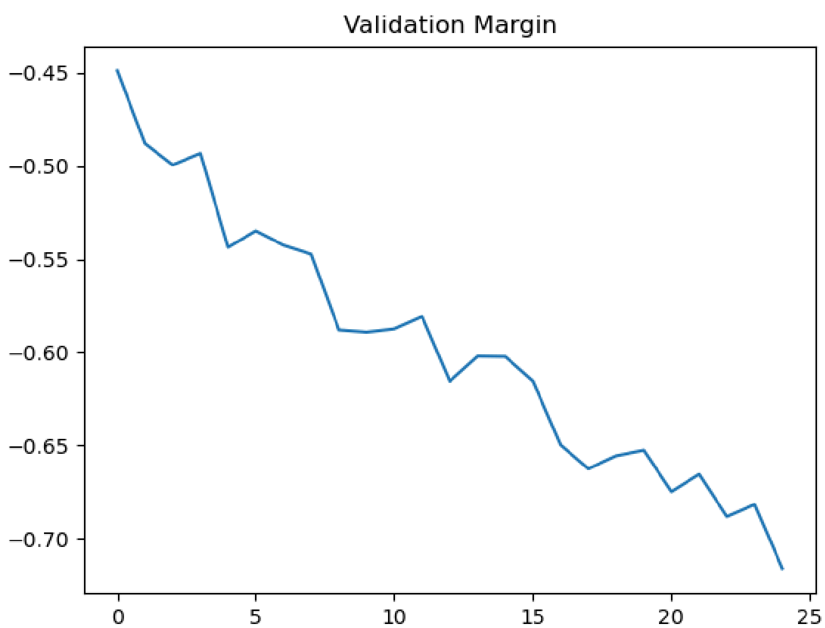
```
================== Summary ==================
 Training loss: 0.966   Training margin -0.571    Training accuracy: 0.668   Validation accuracy: 0.555

Frobenius1:      12.3
Frobenius2:      5.5
Distance1:       8.16
Distance2:       4.21
Spectral1:       1.91
Spectral2:       2.75
Fro_Fro:         67.6
L1max_L1max:     1.23e+03
Spec_Dist:       25.4
Dist_Spec:       3.6e+02
Spec_Dist_sum:   3.85e+02
Spec_L1max:      25.1
L1max_Spec:      3.54e+02
Spec_L1max_sum:  3.79e+02
Dist_Fro:        44.9
#parameter:      7.89e+05
VC bound:        2.03e+09
L1max bound:     1.03e+11
Your bound:      3.43e+10
(base) ivoryang@Ivorys-MacBook-Air Ivory Yang HW1 %
```

Validation Loss Plot:



Validation Margin Plot:



Do you see any difference in the bounds you computed on your two NNs? If so, what do you think is the reason behind such a difference, and can you think of a scenario in which the difference can be minimized and why?

The observed difference in bounds computed on the two NNs can be attributed to the higher number of hidden units in Model 1, compared to that in Model 2. The increased complexity allows Model 1 to capture more intricate patterns and relationships within the data, resulting in greater bound and higher accuracy.

Minimizing the difference in bounds between Model 1 and 2 can be achieved by training both models for a comparable number of epochs. Training for the same duration allows both models to undergo a similar learning process, which then provides them an equal opportunity to converge to optimal weights. Both models are sufficiently exposed to the dataset, potentially reducing the performance gap caused by differences in training duration.

We expect the network to provide a positive margin. However, is it always the case? Can you explain why sometimes the margin can be negative?

We expect the network to generate predictions with a positive margin, showing a clear distinction between the correct class and other classes. However, this is not always the case. The concept of margin refers to the difference between the predicted score for the correct class and the highest score amongst incorrect classes. The margin becomes negative when the network's prediction is incorrect, referring to a misalignment between the predicted and true class. As the margin is computed by subtracting the score of the highest incorrect class from the score of the correct class, when the network predicts incorrectly, and the predicted class score is lower than that of an incorrect class, the margin turns negative. For example, in Model 1, the validation accuracy is 0.568. This indicates that the network makes correct predictions only about 56.8% of the time. For the instances where the network mis-classifies, and there is a mismatch between the predicted class and true class, the margin becomes negative.

Prove the output volume will have size $M \times M \times V'$, where $M = (N-n+2Z)/S + 1$. [Hint] Drawing the convolution operation with Z and S on it can help you get some intuition of proving the formula.

Let:
N x N x V be the size of the input volume
n x n x V be the size of the filters
Z be the number of zeros padded at the top/bottom/left/right of the image
S the stride
V' be the number of filters.

1. The input volume is padded with Z zeros on each side, so the resulting volume before convolution is N+2Z, with depth V.

2. The size of the filters are n x n x V. We have to apply the filters from position 0 to position (N+2Z)-n, with the right edge of the filter at position N+2Z.

3. Stride S subdivides the interval into (N+2Z-n)/S intervals. However, given that both the starting and ending positions are applied to, we have an extra position for the filter, so the number of intervals becomes (N-n+2Z)/S + 1

4. As a result, each applied filter produces a volume of size M x M, where M = (N-n+2Z)/S + 1. As we have V' filters, we can prove that the resulting volume will be of size M x M x V'.