

Issue:

Validation loss increases and is greater than initial validation loss, but the validation accuracy is significantly greater than the initial validation accuracy.

Steven Cao

03/25/2022

The Problem

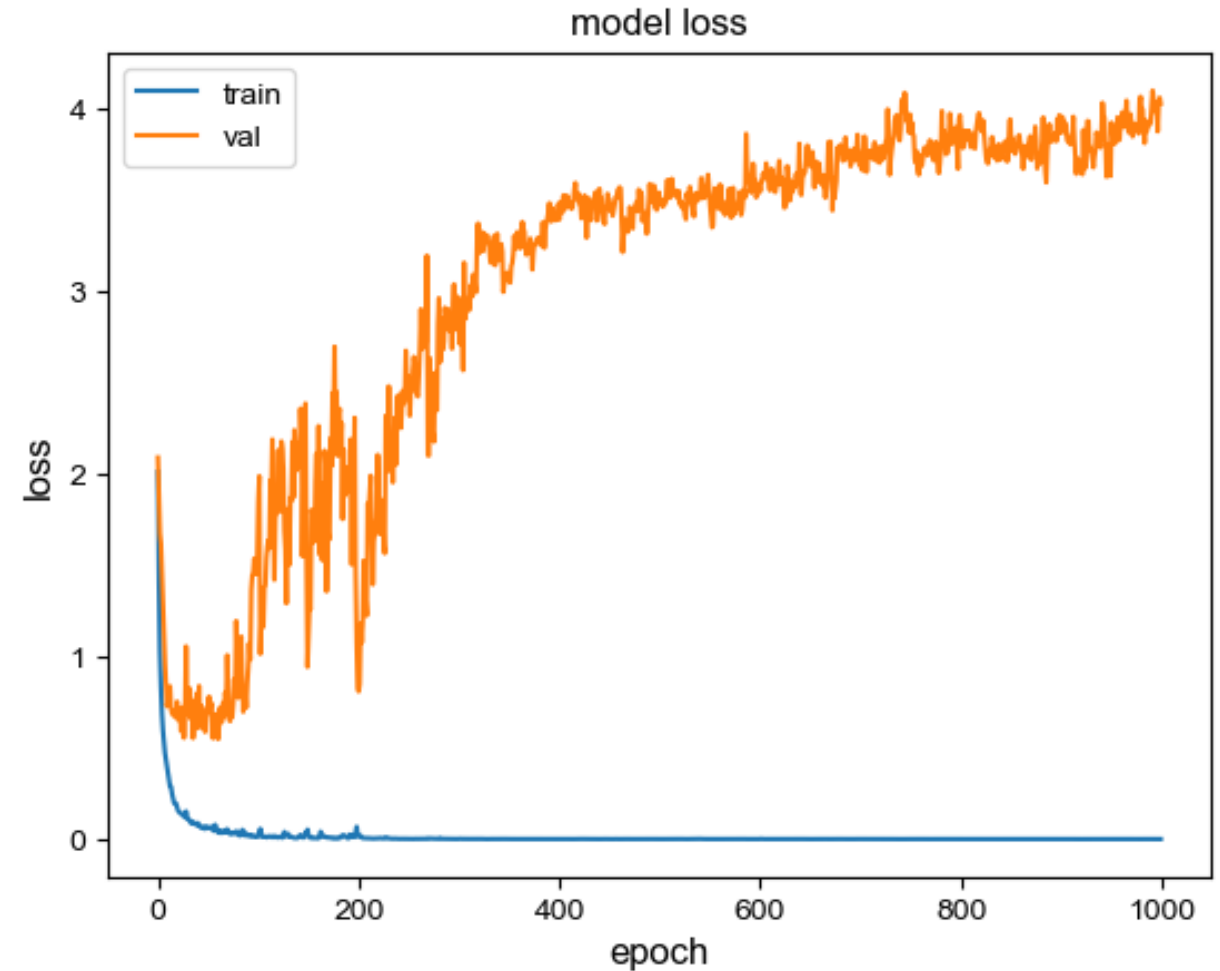
Loss increases during training and ends up higher than the initial loss.

Conclusion:

There is a problem. By the 1000th epoch, validation loss is higher than the initial loss, which is not what we want because it suggests the model is not learning as expected.

Next step:

I manually computed accuracy, precision, recall, and F1 score for each subject on the initial model, best model, and last model to confirm these observations.



Results from a few metrics

1:

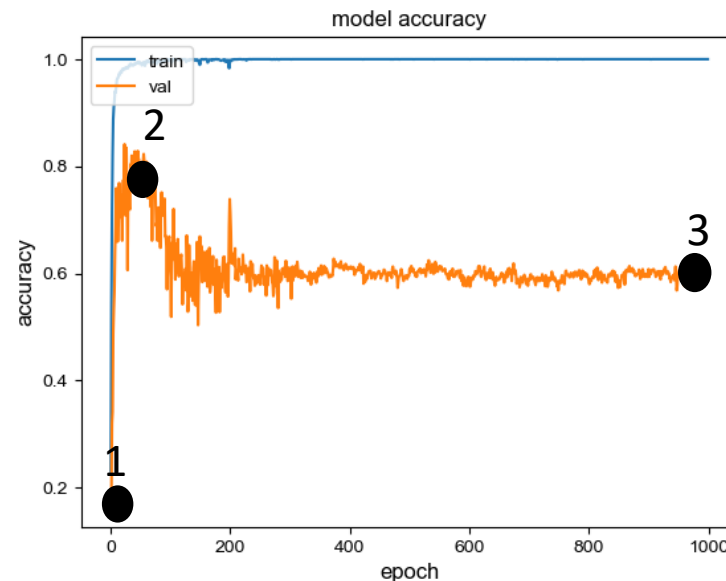
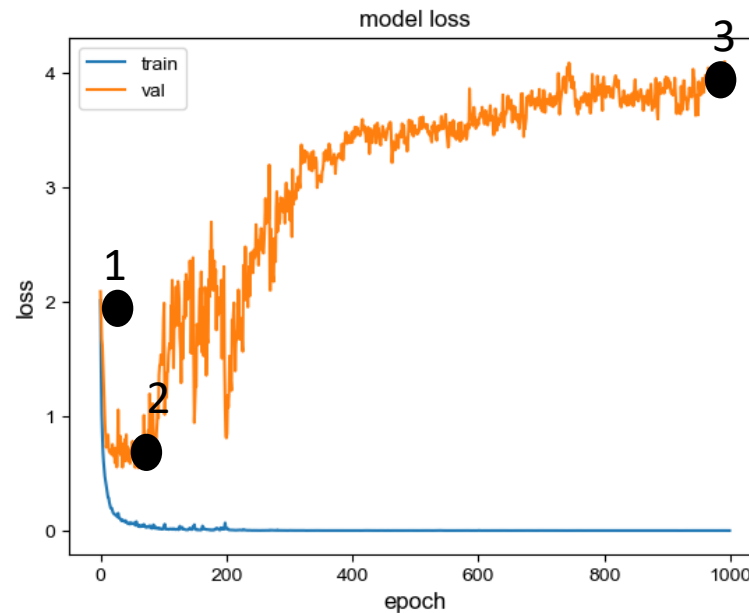
| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.14 | 0.11 | 0.13 | 115 |
| 1 | 0.07 | 0.07 | 0.07 | 115 |
| 2 | 0.17 | 0.10 | 0.12 | 115 |
| 3 | 0.13 | 0.18 | 0.15 | 115 |
| 4 | 0.11 | 0.10 | 0.11 | 113 |
| 5 | 0.09 | 0.05 | 0.07 | 115 |
| 6 | 0.11 | 0.17 | 0.13 | 115 |
| 7 | 0.12 | 0.13 | 0.12 | 115 |
| 8 | 0.15 | 0.16 | 0.15 | 115 |
| accuracy | | | 0.12 | 1033 |
| macro avg | 0.12 | 0.12 | 0.12 | 1033 |
| weighted avg | 0.12 | 0.12 | 0.12 | 1033 |

2:

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.90 | 0.92 | 0.91 | 115 |
| 1 | 0.74 | 1.00 | 0.85 | 115 |
| 2 | 0.96 | 0.83 | 0.89 | 115 |
| 3 | 0.76 | 0.92 | 0.83 | 115 |
| 4 | 0.69 | 0.93 | 0.79 | 113 |
| 5 | 0.97 | 0.54 | 0.69 | 115 |
| 6 | 1.00 | 0.90 | 0.95 | 115 |
| 7 | 1.00 | 0.59 | 0.74 | 115 |
| 8 | 0.86 | 0.98 | 0.91 | 115 |
| accuracy | | | 0.85 | 1033 |
| macro avg | 0.87 | 0.85 | 0.84 | 1033 |
| weighted avg | 0.87 | 0.85 | 0.84 | 1033 |

3:

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.33 | 1.00 | 0.50 | 115 |
| 1 | 0.93 | 0.91 | 0.92 | 115 |
| 2 | 0.00 | 0.00 | 0.00 | 115 |
| 3 | 0.72 | 1.00 | 0.84 | 115 |
| 4 | 0.49 | 0.95 | 0.64 | 113 |
| 5 | 1.00 | 0.09 | 0.16 | 115 |
| 6 | 1.00 | 0.99 | 1.00 | 115 |
| 7 | 1.00 | 0.53 | 0.69 | 115 |
| 8 | 0.00 | 0.00 | 0.00 | 115 |
| accuracy | | | 0.61 | 1033 |
| macro avg | 0.61 | 0.61 | 0.53 | 1033 |
| weighted avg | 0.61 | 0.61 | 0.53 | 1033 |



Conclusion:

The metric results show that validation accuracy is higher than the initial validation accuracy.

This does not support the earlier observation that the model is not learning.

Next step:

Let's check whether there is an issue with how categorical cross-entropy loss is being calculated.

Calculating categorical cross-entropy

$$\text{Loss} = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^K y_{ic} \log(p_{ic})$$

```
losses = []
for label, pred in zip(y_true_hot, y_pred):
    pred /= pred.sum(axis=-1, keepdims=True)
    losses.append(np.sum(label * -np.log(pred), axis=-1, keepdims=False))
print(losses)
print(np.mean(losses))
```

Conclusion:

There is no issue with how the model computes loss.

Next step:

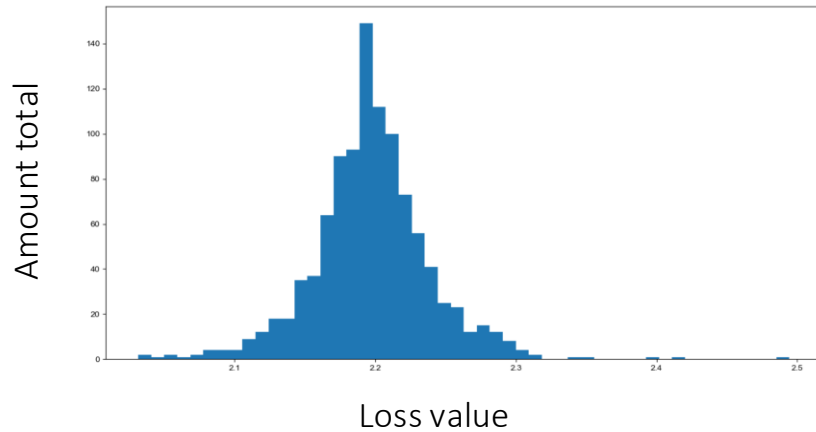
Understand the distribution of epoch losses that make up the overall loss.

Cross-entropy loss check

1: model : 2.08882737159729 ; my calculation: 2.197060735389547
2: model : 0.54115468263626 ; my calculation: 0.5411547039803284
3: model : 4.026108644442618 ; my calculation: 4.026108644442618

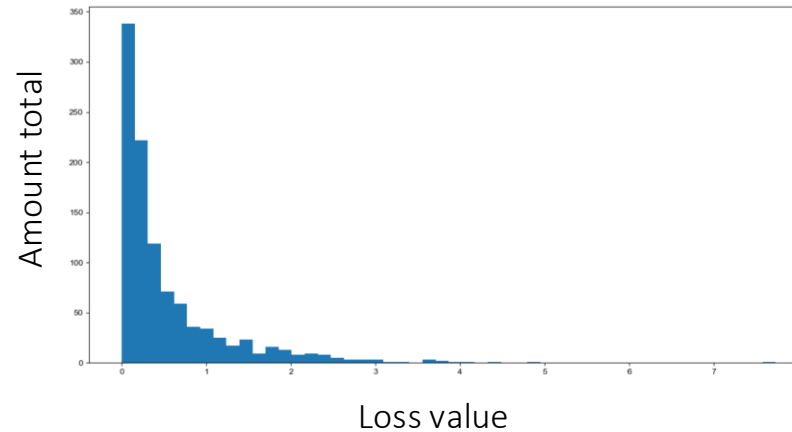
Distribution of losses

Model 1



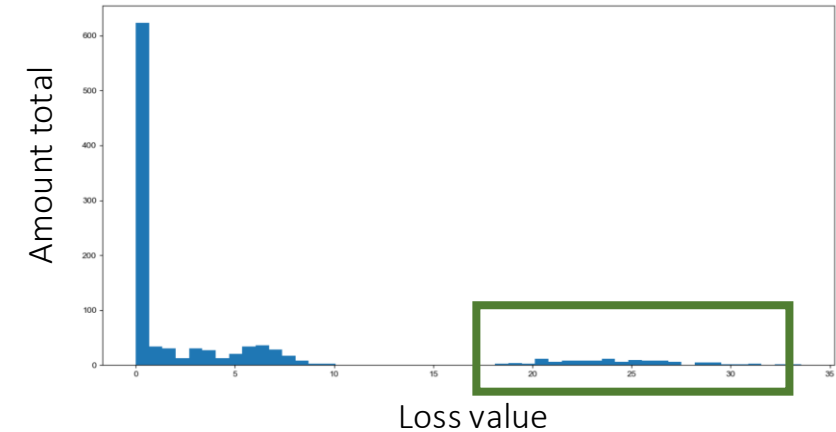
Average Loss: 2.08882737159729

Model 2



Average Loss: 0.54115468263626

Model 3



Average Loss: 4.026108644442618

Conclusion:

A small number of inputs are blowing up the mean validation loss.

Next step:

Find why those samples produce such large losses.

Why we have those outliers

1) Some inputs have very large loss because the model assigns an extremely low probability to the true class.

Example

`Y_pred[-4] = [9.9981767e-01, 6.8540763e-15, 5.5824216e-07, 1.1135705e-11, 5.9789218e-13, 2.4447195e-09, 4.2163074e-06, 1.7761470e-04, 2.9714092e-14]`

`Y_true_hot[-4] = [0., 0., 0., 0., 0., 0., 0., 0., 1.]`

Input loss: $1 * -\log(2.9714092e-14) = \underline{\underline{31.24}}$

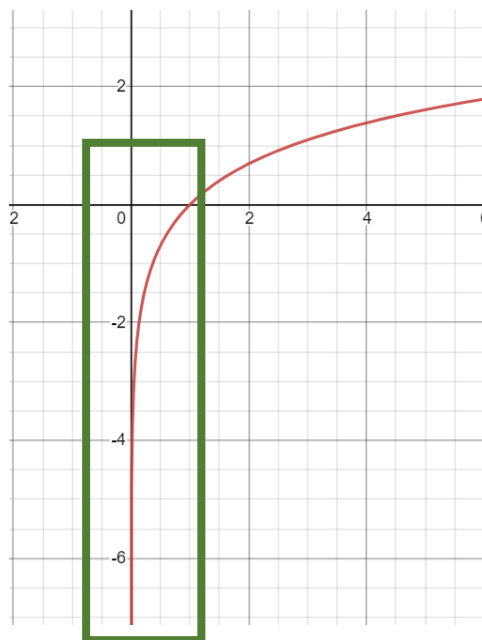
Conclusion:

Current categorical cross-entropy is not a good standalone reflection of model performance in this case.

Next step:

We need a way to reduce the impact of extreme outlier losses on the mean.

2) With cross-entropy classification loss, very bad predictions are penalized much more than good predictions are rewarded.



$Y = \ln(x)$ function

3) So even if many inputs are predicted correctly (low loss), a few misclassified inputs can have very high loss and blow up the mean loss.

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.33 | 1.00 | 0.50 | 115 |
| 1 | 0.93 | 0.91 | 0.92 | 115 |
| 2 | 0.00 | 0.00 | 0.00 | 115 |
| 3 | 0.72 | 1.00 | 0.84 | 115 |
| 4 | 0.49 | 0.95 | 0.64 | 113 |
| 5 | 1.00 | 0.09 | 0.16 | 115 |
| 6 | 1.00 | 0.99 | 1.00 | 115 |
| 7 | 1.00 | 0.53 | 0.69 | 115 |
| 8 | 0.00 | 0.00 | 0.00 | 115 |
| accuracy | | | 0.61 | 1033 |
| macro avg | 0.61 | 0.61 | 0.53 | 1033 |
| weighted avg | 0.61 | 0.61 | 0.53 | 1033 |

Model 3

How can we prevent those huge losses?

One possible solution

- Idea: Categorical cross-entropy is unbounded.
- Consequence: A few very wrong predictions can make the mean loss blow up.
- Proposed fix: Use a bounded version of cross-entropy by capping per-sample loss at 3 standard deviations above the initial model's mean loss (treated as the random/worst-case baseline).

Other pathways

- Investigate the data and understand why those specific samples cause the model to be grossly wrong.
- Inspect the model architecture to see whether any layer/design choice could contribute to this behavior.

Works cited

- <https://stats.stackexchange.com/questions/282160/how-is-it-possible-that-validation-loss-is-increasing-while-validation-accuracy>
- <https://stats.stackexchange.com/questions/258166/good-accuracy-despite-high-loss-value/448033#448033>