Evaluate Your ML Model by Interpreting Loss Curves

Jerry Peng
jp4906@nyu.edu

1 Interpreting Loss Curves

Loss curves can be used to evaluate the performance of a ML model. This section mainly talks about what is loss curve, some different loss curve types, and how should we interept them.

1.1 Normal Loss Curves

Normally, if you have a loss curve looks like the following image, then it means your model has a good performance. As you can see from the Fig 1, in the normal condition, the loss curve should go down quickly with the training epochs increase, and it should get to merged at some point.

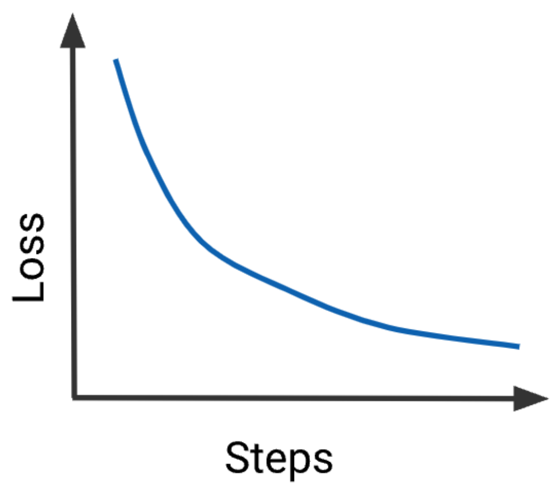

Figure 1: Ideal Loss Curve

1.2 Overfitting Loss Curves

Overfitting Overfit is a very common problem for a ML model. It occurs when a ML has very accurate prediction results for the training dataset but has bad performance on new data, such as validation or test datasets. Overfitting occurs when the model cannot generalize and fits too closely to the training dataset instead. Overfitting may caused by several possible reasons, such as:

1. The training dataset is too small.
2. The training dataset does not contains enough sampls that can represent most possible input data (low variance)
3. The training dataset contains too much noisy data (irrelevant)
4. The model is over trained on the same dataset
5. The model is too complicate

Here are some exampls of overfitting loss curves:

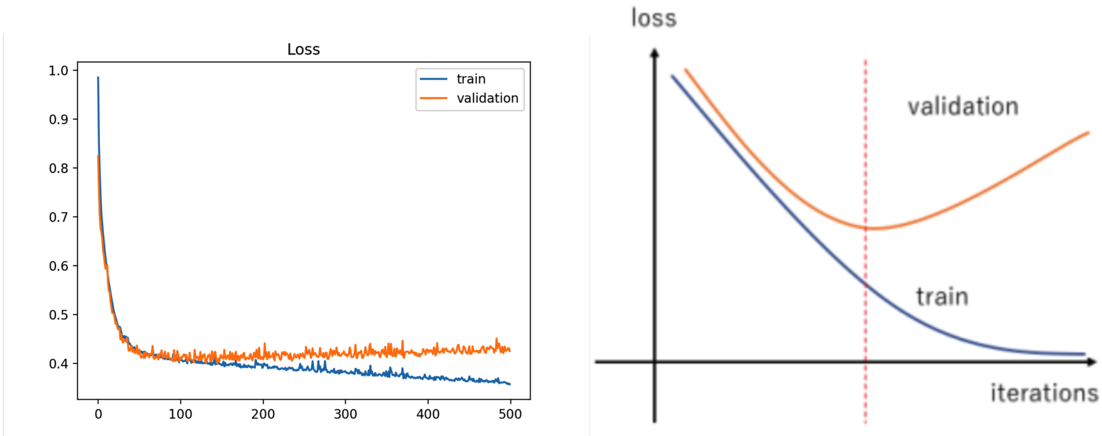

Figure 2: Overfitting Loss Curves

Solutions How can we solve this problem? We can diversifying or scaling the training dataset. Here are some strategies:

1. **Stop the Training Earlier**: From Fig2, we can see that both model start to have overfitting problems after a specific epochs. Hence, if we can stop the training earlier, we can prevent such problem.

2. **Feature Selection (Pruning)**: You can try to identify what features or parameters that impact the ML model a lots and eliminte the other irrelevant features. For example, when you try to predict the house price, the important features should be something like house locations and house size, rather than the owner's name or gender.

3. **Regularization**: Regularization methods try to eliminate those factors that do not impact the prediction outcomes by grading features based on importance. Some common regularizers are *L1* and *L2*. *L1* regularization is the sum of the absolute weights, while *L2* regularization is the sum of the squeared weights

4. **Data Augmentation**: You can try to perform data augmentation to prevent overfitting problems. This strategy can change the training dataset slightly and improve the performance of the model. For example, you can apply transformations to the data, such as translation, rotation, image enhancement, blur, flipping, etc.

1.3 Underfitting Loss Curves

Underfitting Underfitting happens when a ML model does not describe the relationship between the input data and the target accurately. Usually the model has high error/loss on both training and new datasets. High bias and low variance is a good indicator of underfitting. Underfitting can be caused by the following reasons:

1. The model is not complicated enough.

2. The training time is too short.

Here are some exmaples of underfitting: Fig3

Solutions We can avoid underfitting problem by maintaining adequate model complexity. Here are some strategies that can be used to avoid or reduce underfitting:

1. **Training Longer**: Train the ML model for longer time can reduce underfitting, but becareful, you don't want to train it too long or your model could have overfitting problem.

Figure 3: Underfitting Loss Curves

2. **Make the Model More Complicate**: The left figure in Fig 3 is a good example of the loss curve from a too simple model. If you find your loss curve is too flat or going down too slow, it's because your model is too simple. For example, you can add few more layers for your CNN model.

3. **Deregularization**: If the data features become too uniform, the model is unable to identify the dominant trend, leading to underfitting. By decreasing the amount of regularization, more complexity and variation is introduced into the model, allowing for successful training of the model.

1.4 Exploded Loss Curves

Gradient Explosion An error gradient is the direction and magnitude calculated during the training of a neural network that is used to update the network weights in the right direction and by the right amount. In deep networks or recurrent neural networks, error gradients can accumulate during an update and result in very large gradients. The explosion occurs through exponential growth by repeatedly multiplying gradients through the network layers that have values larger than 1.0. Here are some reasons that can cause such problem:

1. NaNs in the input data
2. Anomalous data
3. Division by zero
4. Logarithm of zero or negative numbers
5. Learning rate is too high

Below is a sample graph of gradient explosion:

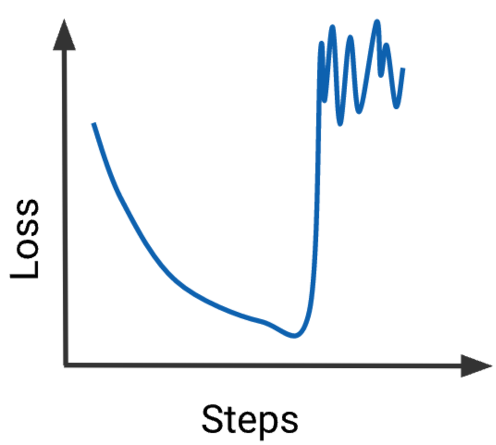

Figure 4: Gradient Explosion Loss Curve

Solutions Now we can talk about how to improve or avoid gradient explosion

1. **Re-design the Network**: It is efficient to improve the gradient explosion if you can make the model has fewer layers. Also, smaller batch size can be a good choice too.

2. **Clipping**: Gradient clipping is a method where the error derivative is changed or clipped to a threshold during backward propagation through the network, and using the clipped gradients to update the weights. By rescaling the error derivative, the updates to the weights will also be rescaled, dramatically decreasing the likelihood of an overflow or underflow.

   There are two ways to perform gradient clipping: 1) **clipped-by-value**, which needs a min and max values, and if the gradient value is out of the range, then we clip the value to the max or min threshold value; 2) **clipped-by-norm**, which clips the gradient by multiplying the unit vector of the gradients with the threshold.

1.5 Loss Curve Not Converge

If your loss curve looks like Fig 5, then your loss curve is not converging. Usually this problem occurs when your ML model is unable to establish correlations between input data and target labels. You can try to solve such problem by:

1. Check if your input features can predict target labels
2. Try to detect if there are bad data examples in your dataset
3. Reduce your learning rate
4. Simplify your model architecture
5. Simplify your dataset

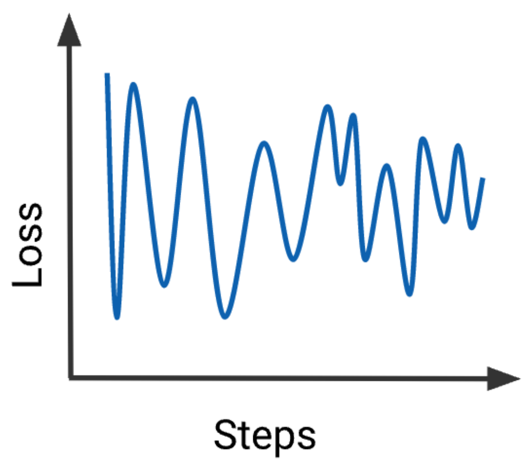

Figure 5: Loss curves is not converging

1.6 Other Types of Problematic Loss Curves

Repetitive The loss curve likes Fig 6 is showing repetitive, step-like behavior. It's probable that the input data seen by the model is itself exhibiting repetitive behavior. Ensure that shuffling is removing repetitive behavior from input data.

Unrepresentative Training Dataset If your loss curves looks like Fig 7, it means the training dataset does not provide sufficient information to learn the problem, relative to the validation dataset used to evaluate it. This may occur if the training dataset has too few examples as compared to the validation dataset. This situation can be identified by a learning curve for training loss that shows improvement and similarly a learning curve for validation loss that shows improvement, but a large gap remains between both curves.

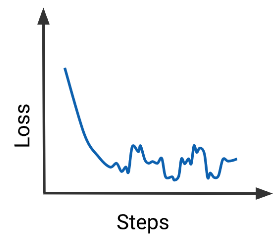

Figure 6: Loss curves is repetitive

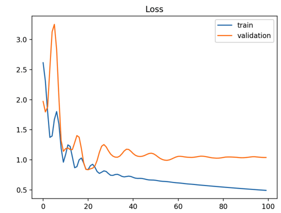

Figure 7: Unrepresentative Training Dataset

Unrepresentative Validation Dataset An unrepresentative validation dataset means that the validation dataset does not provide sufficient information to evaluate the ability of the model to generalize. This may occur if the validation dataset has too few examples as compared to the training dataset.

There are two types of figure can show if your validation set is unrepresentative. If your loss curve looks like Fig 8, which means learning curve for training loss that looks like a good fit (or other fits) and a learning curve for validation loss that shows noisy movements around the training loss, then your validation set is unrepresentative. The second one is like Fig 9, which means the validation dataset may be easier for the model to predict than the training dataset.

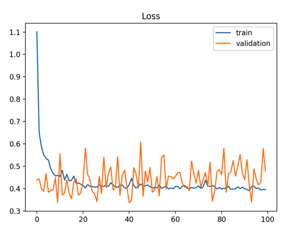

Figure 8: Validation set is unrepresentative

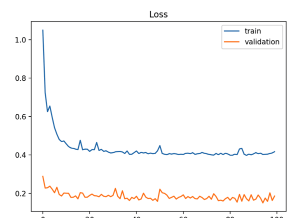

Figure 9: Validation set is easier than training set

References

[ama] What is overfitting? https://aws.amazon.com/what-is/overfitting/?nc1=h_ls. Accessed: 2023-04-03.

[Baj] Aayush Bajaj. Understanding gradient clipping (and how it can fix exploding gradients problem). https://neptune.ai/blog/understanding-gradient-clipping-and-how-it-can-fix-exploding-gradients-problem#:~:text=What%20is%20gradient%20clipping%3F,gradients%20to%20update%20the%20weights. Accessed: 2023-04-03.

[Bro] Jason Brownlee. A gentle introduction to exploding gradients in neural networks. https://machinelearningmastery.com/exploding-gradients-in-neural-networks/. Accessed: 2023-04-03.

[Goo] Interpreting loss curves. https://developers.google.com/machine-learning/testing-debugging/metrics/interpretic. Accessed: 2023-04-03.

[Gre93] George D. Greenwade. The Comprehensive Tex Archive Network (CTAN). *TUGBoat*, 14(3):342–351, 1993.

[IBM] What is underfitting? https://www.ibm.com/topics/underfitting#:~:text=the%20next%20step-,What%20is%20underfitting%3F,training%20set%20and%20unseen%20data. Accessed: 2023-04-03.