A Deep Dive Into Learning Curves in Machine Learning

Understand machine learning better with our guide on accuracy and loss curves. We explain their differences, how to read them, and why they're important

Mostafa Ibrahim

Last Updated: Mar 13, 2024

Introduction

In this post, we will look at the differences between accuracy and loss curves, shed light on their interpretations, and discuss the characteristics of reasonable learning curves. We will also address the importance of tracking loss and accuracy during the training process and how to effectively analyze and address issues that arise. Last, we will examine examples of different learning curve shapes and provide strategies to improve model performance based on these curves.

Here's what we'll be covering:

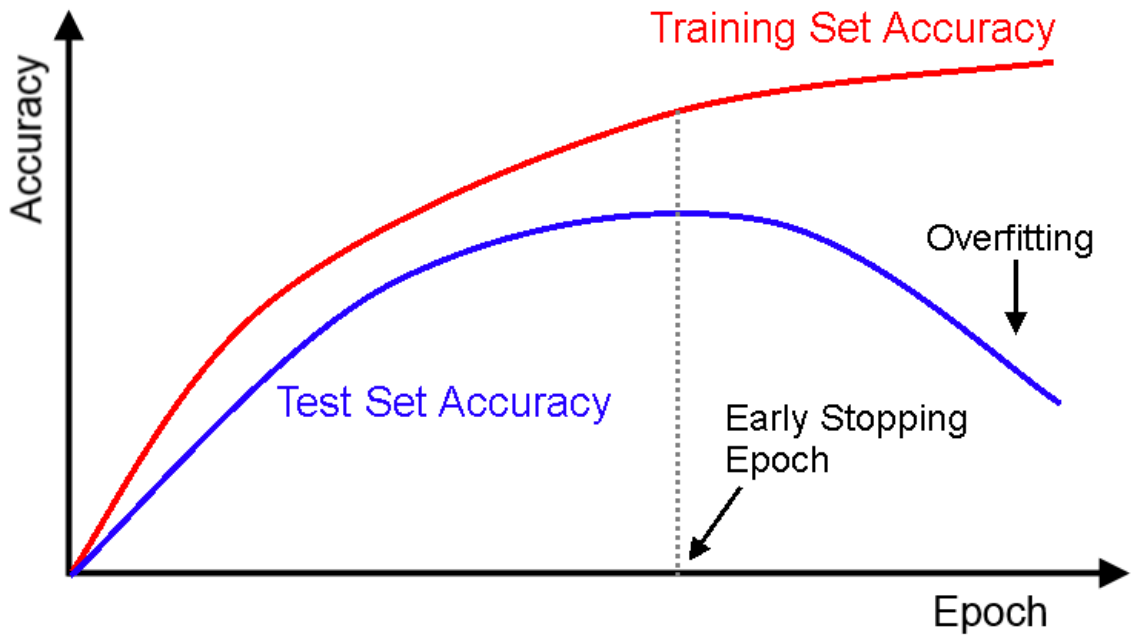
Table of Contents

What are Accuracy and Loss Curves?

Accuracy and loss curves are two common tools we use to understand how well a machine learning model is learning and getting better over time. In the simplest terms, they help us evaluate the model's performance during training.

On the one hand, the accuracy curve records how accurate the model's predictions are on the given data, while the loss curve records the actual difference between the model's prediction and the actual true output.

Accuracy Curve

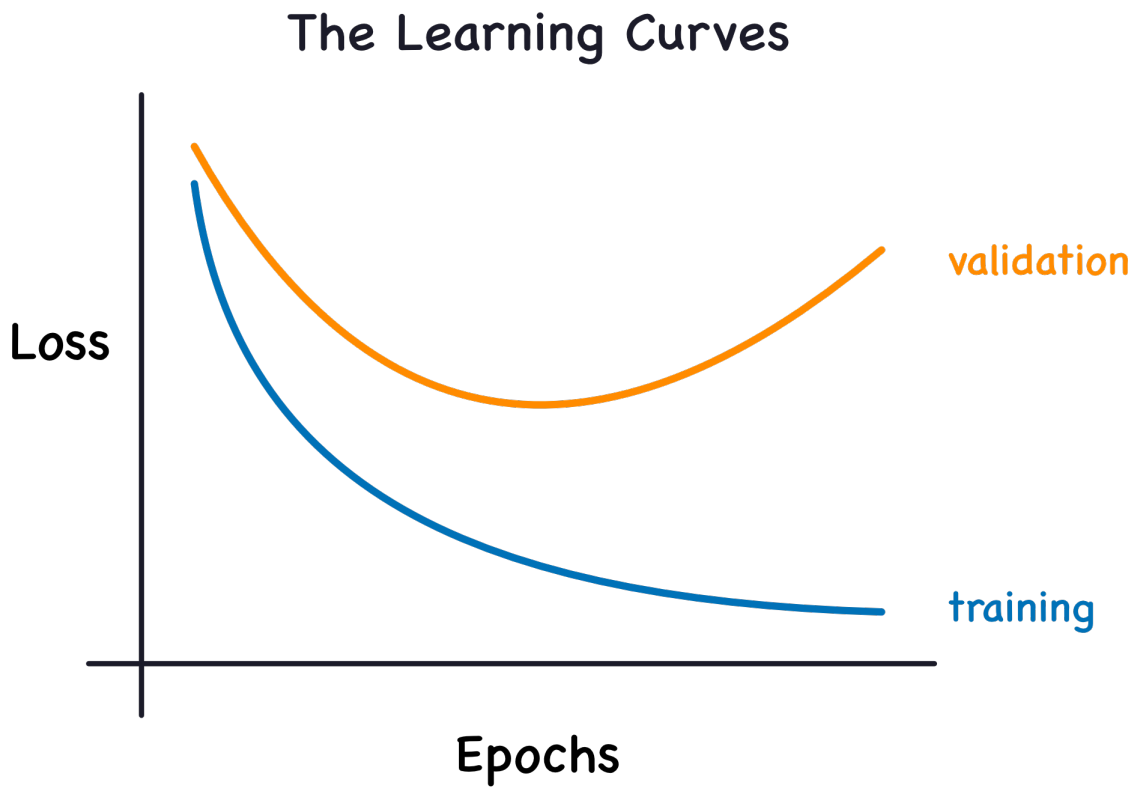


Source

The accuracy curve (also known as the training accuracy curve) shows us how good the model is at making correct predictions on the training data as it goes through the training process. Accuracy is measured in percentages and tells us the proportion of instances the model correctly classified out of the total number of instances. So, the accuracy curve gives us a sense of how well the model fits the training data and improves its ability to make accurate predictions.

Loss Curve

The loss curve, or training loss curve, gives us insights into how the model's performance improves over time by measuring the error (or dissimilarity) between its predicted output and the true output. The loss represents how far off the model's predictions are from the actual values.



Source

By minimizing the loss, the model aims to make its predictions as close as possible to the true values.

Put simply: the loss curve shows us how the model's error decreases as it learns, which indicates an improvement in its performance.

What Are the Attributes of a Learning Curve?

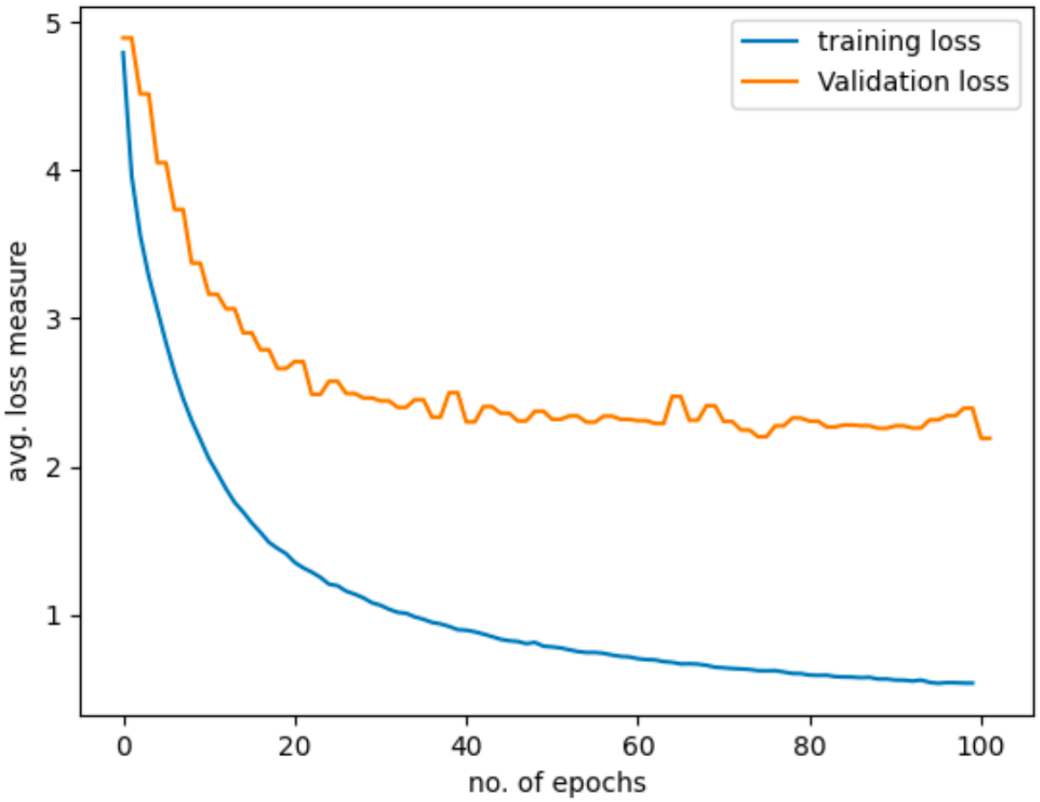
When we talk about a reasonable learning curve, we're looking at certain features that tell us how well a model is learning. These features include the smoothness of the curve, its convergence, and how generlizable the learning rate is.

Having said that, let's dive in deeper and explore what each of those terms implies.

Smoothness

A smooth learning curve means that the model's performance changes gradually and

consistently as it goes through training. We like to see a smooth curve because it indicates that the model is steadily improving over time. Of course, there might be small ups and downs along the way, but if the curve has big and sudden jumps, it could be a sign that something is not quite right. A smooth learning curve suggests that the model is learning in a stable and consistent manner.



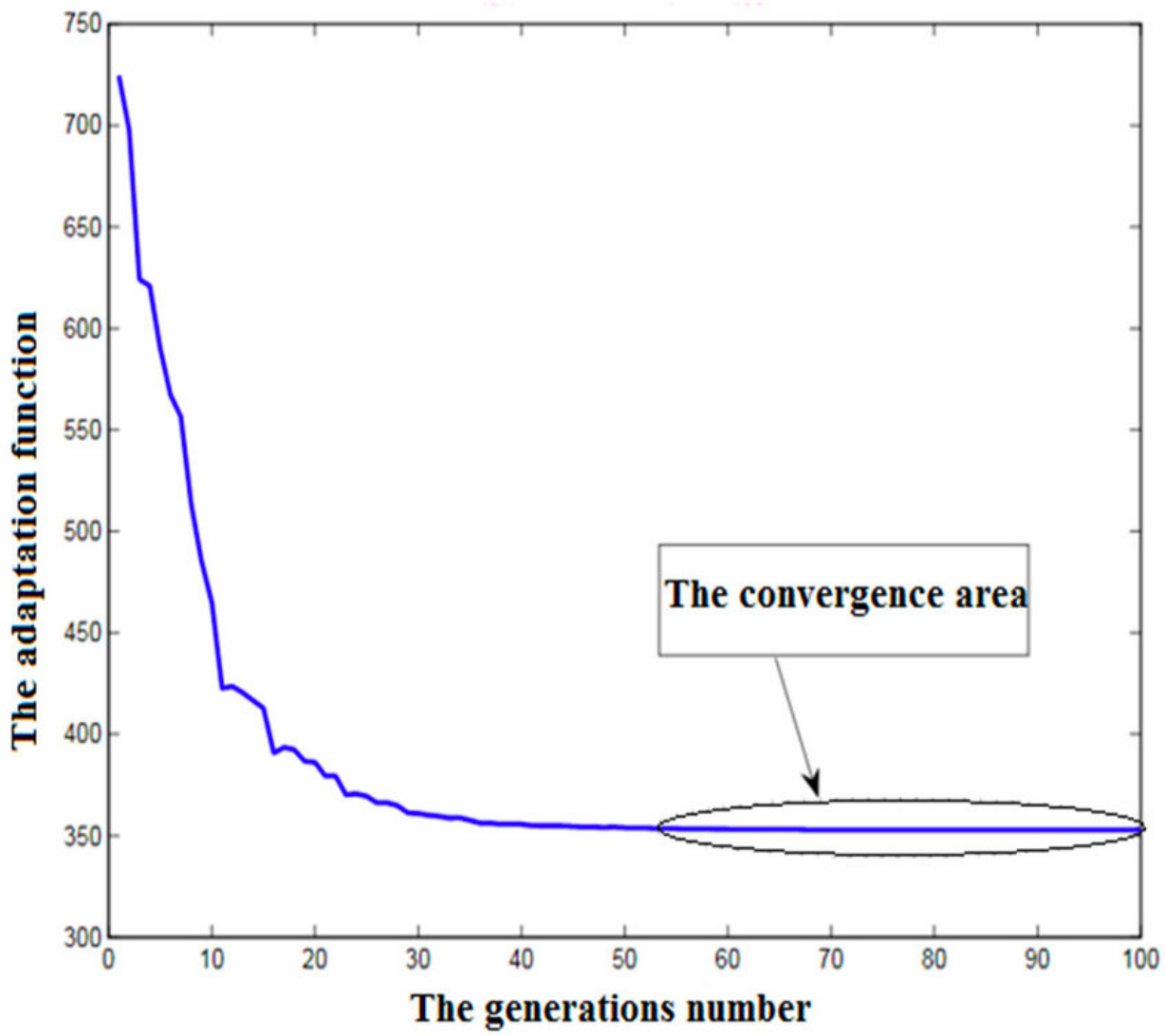
Source

The above figure shows 2 different learning curves, the smooth training curve in blue indicated by blue and a rougher validation curve in orange.

Convergence

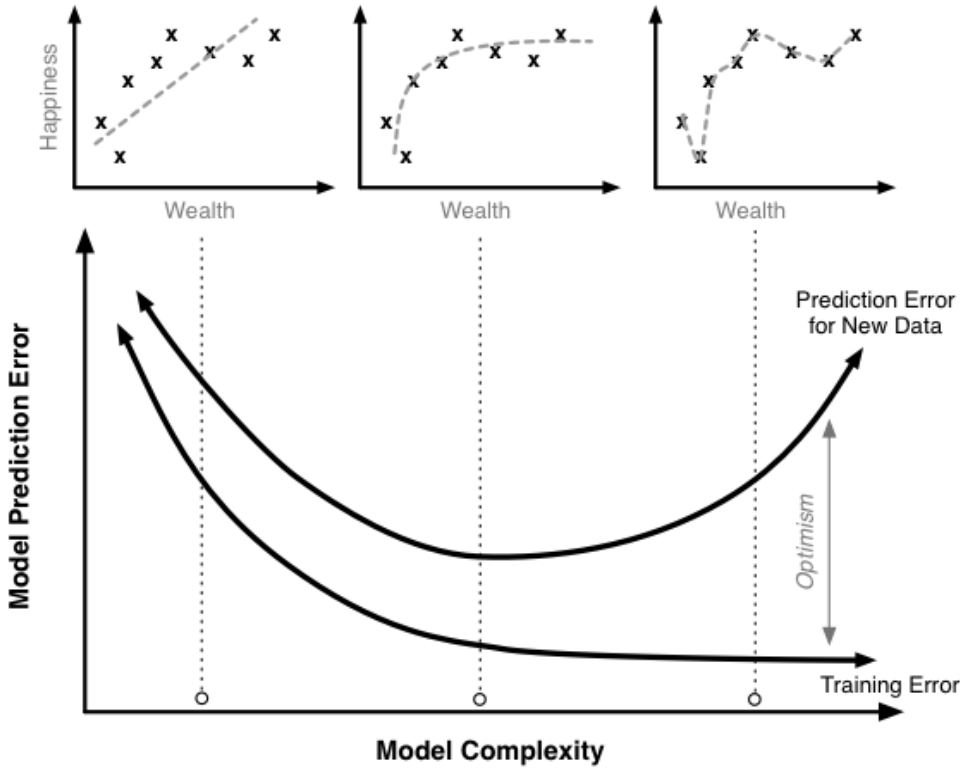
Convergence is a term of art for when the learning curve reaches a stable or optimal state. Ideally, we want the learning curve to converge to a point where further training doesn't lead to significant improvements. This means the model has learned as much as it can from the training data and has reached its best performance.

Convergence tells us that the model has understood the underlying patterns and relationships in the data and is making accurate predictions. We usually see convergence when the learning curve levels off or plateaus.



Source

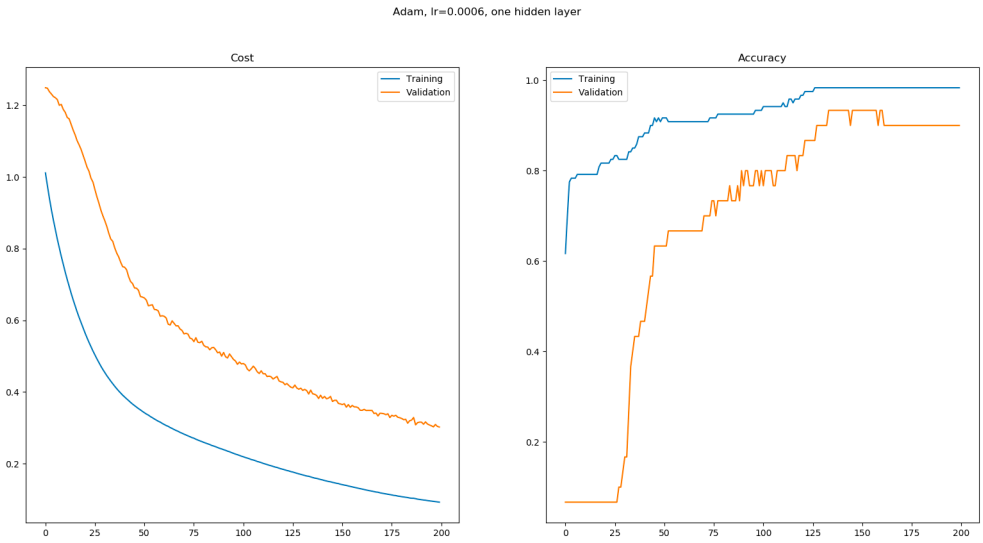
Generalization



Source

A good learning curve should also show that the model can generalize well. This means it doesn't just perform well on the training data but also on new, unseen data. If the learning curve shows a big difference between the model's performance on training data and test data, it's a sign of a problem. It could mean that the model is overfitting, which happens when it memorizes the training data too well and struggles to handle new examples. A reasonable learning curve demonstrates that the model can generalize its learnings to unseen data.

What is the Correlation Between Loss and Accuracy?



Source

Let's look at the picture above. Both the loss curve (on the left side) and the corresponding accuracy curve (on the right side) are displayed. These curves exhibit distinct characteristics from each other. Let's examine each in detail.

The **loss curve** shows the values of the model's loss over time. Initially, the loss is high and gradually decreases, indicating that the model is improving its performance. A decrease in the loss value suggests that the model is making better predictions, as the loss represents the error or dissimilarity between the predicted output and the true output. Therefore, a lower loss indicates enhanced performance.

Now let's shift our focus to the **accuracy curve**. It represents the model's accuracy over time. The accuracy curve begins with a value of zero and increases as the training progresses. The accuracy measures the proportion of correctly classified instances out of the total number of instances. So, as the accuracy curve rises, it signifies that the model is making more correct predictions, thereby enhancing its overall performance.

One notable difference between the curves is the smoothness and the presence of "plaque lines" on the accuracy curve. While the presence of plaque lines is not a requirement for either curve, in this particular case, the plaque lines indicate discrete jumps in accuracy instead of a continuous increase.

This behavior is attributed to how accuracy is measured, particularly in binary classification tasks where the model's final output is compared to the true output to determine accuracy. As the model's predictions approach the true values, the accuracy may remain the same until it reaches a threshold where the predictions align precisely with the true values. These thresholds are represented by the plaque lines, causing the

accuracy to jump to a higher value.

To provide an example, let's consider a binary classifier that distinguishes between cats and dogs, assigning a value of 0 for cats and 1 for dogs. Suppose, during the first training step, the model's predicted value is 0.3, but the true value is actually 1, indicating a dog. In binary classification, the predicted value is rounded to either 0 or 1. In this case, it is rounded down to 0, resulting in an incorrect classification. While this training step contributes to the loss curve, it has no actual contribution to the accuracy curve since the prediction was incorrect.

As the model is trained on more data, the next prediction might be 0.4. Again, this is an incorrect prediction, as the actual output should be that of a dog. This step also contributes to the Loss curve while creating a plaque in the accuracy curve. A change in the accuracy curve occurs only when the model makes a correct prediction in the subsequent step.

By understanding these dynamics, we can observe how the accuracy curve shows discrete jumps rather than a smooth progression as the model gradually improves and reaches correct predictions.

What is the Importance of Tracking Loss and Accuracy During the Training Process?

Checking Performance: As you may have guessed, loss and accuracy metrics help us understand how well our model is actually doing. They give us a measure of whether the model is learning from the training data and making accurate predictions. By keeping track of these numbers, we can see if the model is improving over time or if there are any issues that need attention.

Choosing the Best Model: Loss and accuracy act like a scorecard that allows us to compare different models. When we're trying out different models or settings, these metrics help us decide which one is the most effective. By looking at their performance based on loss and accuracy, we can make informed choices about which model to use in real-world applications.

Catching Problems Early on: Monitoring loss and accuracy lets us catch any issues before they become major problems. If we see sudden changes or wild swings in these metrics during training, it could be a sign that something is off. It might mean that the model is memorizing the training data too much or not capturing the patterns well, or

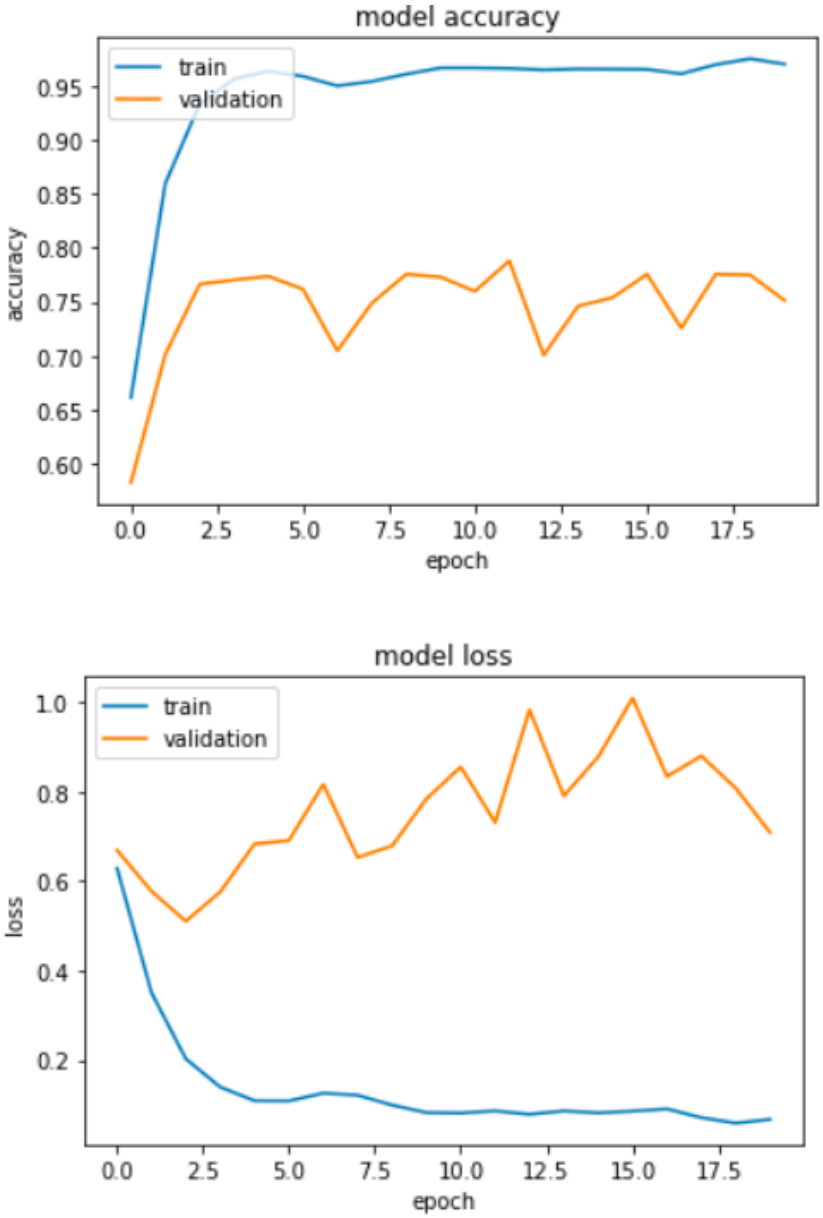
there could be problems with the data itself. By paying close attention to loss and accuracy, we can spot these issues early on and take steps to address them.

Fine-tuning for the Best Performance: Loss and accuracy metrics help us fine-tune the model's settings. Things like the learning rate, batch size, or regularization strength can greatly impact how well the model performs. By observing how loss and accuracy change as we adjust these settings, we can find the optimal values that maximize the model's performance.

Visualizing and Communicating: Loss and accuracy curves provide visual representations of the model's training progress. They offer a clear picture of how the model is learning and improving over time. We can use these visuals to have meaningful discussions, identify areas that need improvement, and help others understand how the model is behaving.

How Can the Shape of the Learning Indicate Issues With the Model or Data?

1. Low Training Loss, High Validation Loss

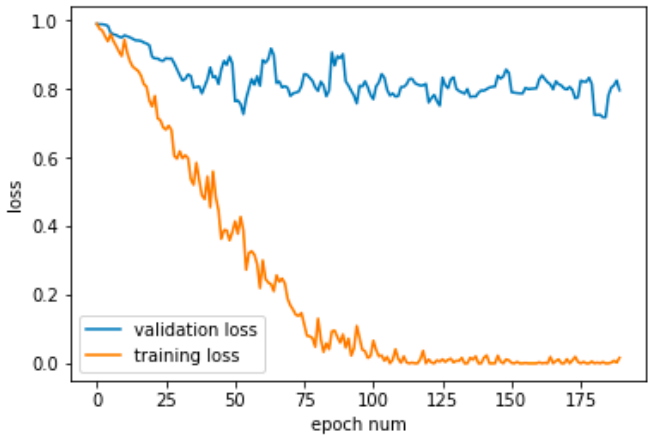


Source

When a machine learning model demonstrates a situation where the training loss is low but the validation loss is high, it indicates that the model is overfitting. Overfitting occurs when the model becomes too focused on capturing the patterns of the training data, resulting in poor performance when presented with new, unseen data.

The low training loss suggests that the model has effectively learned to fit the training data, but the high validation loss indicates its inability to generalize well. This often arises due to factors such as limited data, the complexity of the model, overemphasis on specific features, or excessive training. To address this issue, one can explore techniques such as augmenting the training data, simplifying the model architecture, incorporating regularization methods, or employing early stopping strategies.

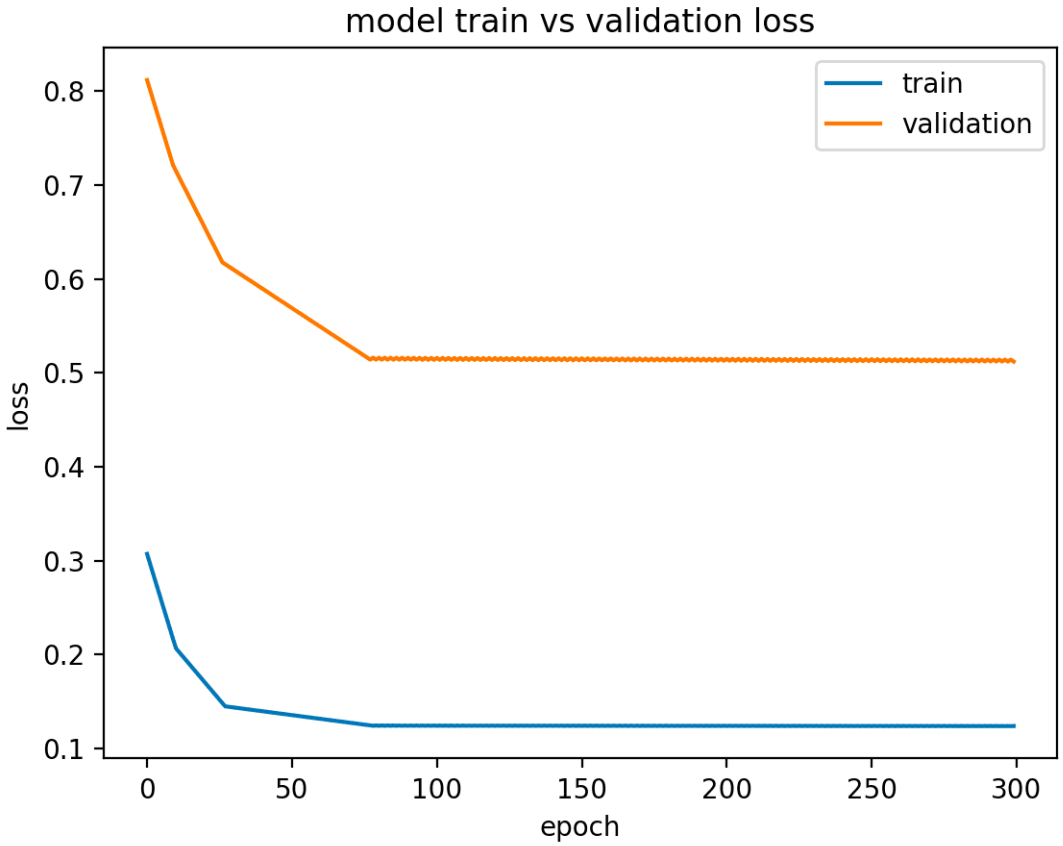
2. Training Loss Decreases, Validation Loss Plateaus



Source

If the training loss decreases while the validation loss plateaus or decreases at a slower rate, it suggests the model may be overfitting. The model is becoming too specialized in fitting the training data, but it fails to generalize well to new data. This discrepancy indicates that the model may have learned the specific patterns in the training set but is struggling to apply them to unseen examples.

3. Large Gap between Training and Validation Performance

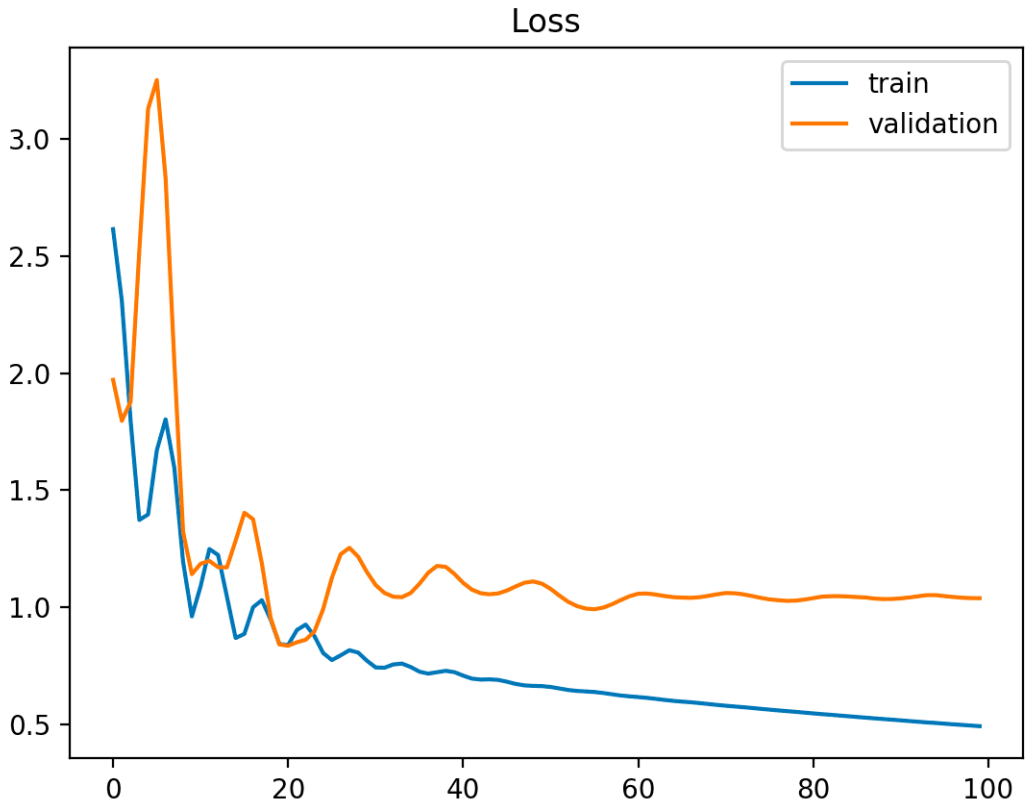


Source

If there is a significant difference between the model's performance on the training data and the validation data, it indicates potential issues with generalization. A large gap suggests that the model is not able to generalize its learnings from the training data to new examples effectively. This could be due to factors such as dataset biases, imbalanced data, or inadequate model regularization.

4. Erratic or Unstable Learning Curve

If the learning curve exhibits erratic behavior with frequent fluctuations or inconsistent changes in performance, it might indicate problems with the model or data. This could be a result of issues such as a high learning rate causing instability, inadequate preprocessing of data, or noisy data affecting the model's learning process.



Source

What Are the Common Issues With Learning Curves, and How Can We Address Them?

Model Overfitting

In the case of model overfitting, acquiring a more diverse dataset that represents a distribution closer to real-world data can help the model learn more generalized patterns, thus reducing overfitting. Another solid approach would be to introduce regularization techniques such as L1 or L2 regularization, dropout, or early stopping to prevent overfitting by adding constraints on the model's parameters or stopping the training process at an optimal point.

Model Underfitting

An underfitting model is usually a simple model. Thus, we should consider increasing

its complexity by adding more layers, increasing the number of hidden units, or using a more sophisticated architecture. Another approach to fixing underfitting is to experiment with different hyperparameter settings such as learning rate, batch size, optimizers, or regularization strength to find the optimal combination that improves the model's performance.

Poor Model Evaluation

Imagine putting in countless hours of effort into developing a sophisticated model, only to end up with disappointing performance when evaluating its overall effectiveness. Overcoming such challenges requires adopting various approaches, some of which include:

Cross-Validation: Use cross-validation techniques to assess the model's performance on multiple subsets of the data and obtain a more reliable estimate of its generalization ability.

Monitor Learning Curves: Continuously analyze the learning curves during training to identify any signs of overfitting, underfitting, or convergence issues. Adjust the model or training process accordingly.

Addressing Learning Curves Issues Using Weights and Biases

To address the above issues, tools such as Weights and Biases do come in handy. To check that the model is not overfitting we need to plot both the eval and the training loss graphs.

To do this using W&B, add the following code snippets to your code:

Step 1: Initialze a New Wieghts and Baises Project

```
wandb.init(project="Insert Project Name Here", name="Insert Your Run Name Here")
```

Step 2: Saving our training and eval loss throughout the process

```
num_epochs = 10
for epoch in range(num_epochs):
    train_loss = train_one_epoch(model, train_loader, optimizer, criterion)
    eval_loss, eval_accuracy = evaluate(model, eval_loader, criterion)
    print(f"Epoch [{epoch+1}/{num_epochs}], Train Loss: {train_loss:.4f}, Eval Loss: {e
```

Step 3: Logging the Training Loss and Eval Loss Graphs into W&B

After training your model and saving the training and evaluation loss

```
wandb.log({
    "Train Loss": train_loss,
    "Eval Loss": eval_loss
})
```

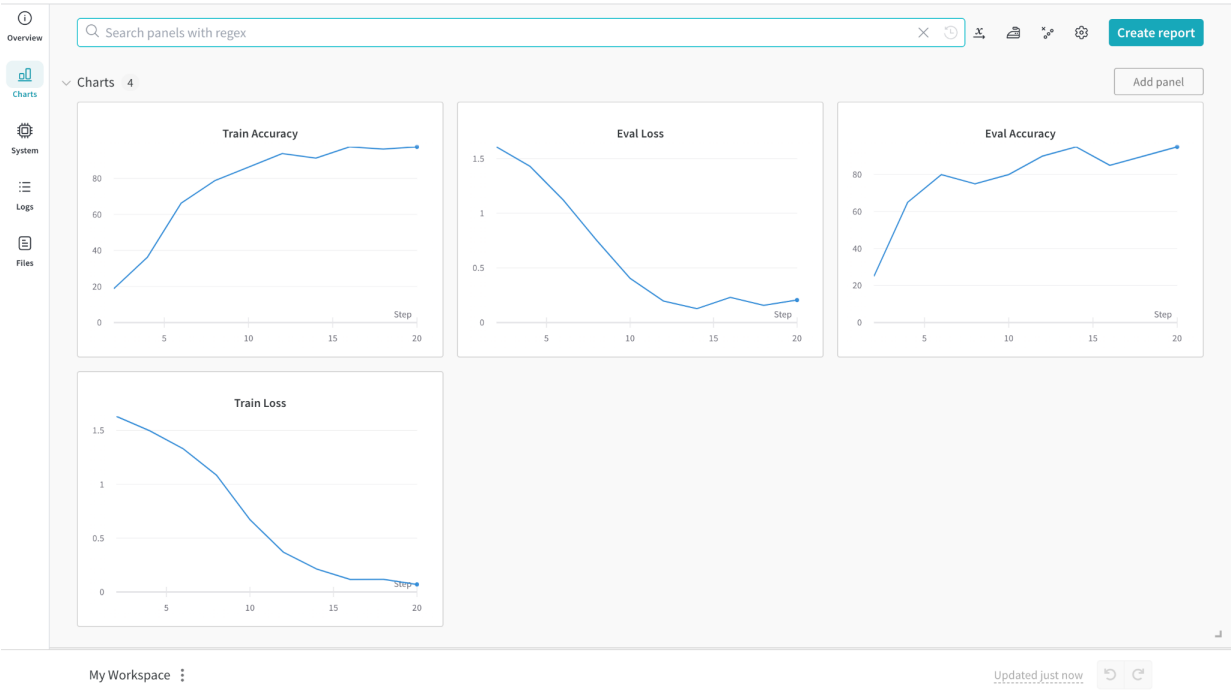
Step 4: Open Weights and Biases to Check the given graphs

Once the model training concludes, a unique link will be generated in your execution output. Clicking on this link will open a new W&B dashboard tab. This dashboard will showcase all the data and metrics that you've logged during your code's execution. In our case, both the training and eval loss graphs.

```
View run random_masking at: https://wandb.ai/mostafaibrahim-17/time_masking_experiments/runs/s6fwlucp
Synced 6 W&B file(s), 25 media file(s), 0 artifact file(s) and 0 other file(s)
Find logs at: ./wandb/run-20231015_190529-s6fwlucp/logs
```

Step 5: Evaluate Learning Issues using the Learning Curves

After clicking on the provided link, the subsequent display will emerge.



Based on our observations so far, the trends in the curves are quite promising. Both the training and evaluation loss curves display a consistent decline, indicating that the model is effectively learning as it progresses through the epochs. Importantly, the

absence of overfitting suggests that the model retains its capability to make accurate predictions on unseen data, as evidenced by the performance on the evaluation dataset.

In the case of both accuracy curves, both curves display a small zigzag-like pattern. In situations where both the evaluation and training loss curves are smooth, yet the accuracy curves display a zigzag pattern, it suggests a nuanced interplay between model accuracy and loss. The loss, being a continuous measure, can change subtly across epochs or batches, which might not necessarily result in a significant change in accuracy. Conversely, even small fluctuations in the model's confidence about its predictions can lead to changes in accuracy, especially in scenarios with fewer classes or more balanced datasets where a few misclassifications can cause noticeable percentage shifts.

Conclusion

In conclusion, learning curves in machine learning are like valuable windows into the performance of our models during training. They provide us with exciting insights and allow us to gauge how well our models are learning and improving over time. Accuracy curves cheerfully reveal how accurately our models make predictions, while loss curves show us the journey of minimizing the gap between predictions and reality. Furthermore, by addressing issues like overfitting and poor generalization, we can unlock the full potential of our models and achieve remarkable results. With that being said, understanding the meaning behind those learning curves is always beneficial.

Tags: Articles, Domain Agnostic, Beginner, Tutorial

Created with ❤️ on Weights & Biases.

https://wandb.ai/mostafaibrahim17/ml-articles/reports/A-Deep-Dive-Into-Learning-Curves-in-Machine-Learning--Vmlldzo0NjA1ODY0

Never lose track of another ML project. Try W&B today.

SIGN UP TRY W&B NOW

Weights & Biases

Get weekly updates with the latest W&B news.

Subscribe

PRODUCTS
Dashboard
Sweeps
Artifacts
Reports
Tables

QUICKSTART
Documentation

RESOURCES
Courses
Forum
Tutorials
Benchmarks

W&B
About Us
Authors
Contact

Copyright ©2024 Weights & Biases. All rights reserved. Terms of Service Privacy Policy