I:

1. Gradient descent

Gradient descent is an optimization algorithm which is commonly-used to train machine learning models and neural networks. Training data helps these models learn over time, and the cost function within gradient descent specifically acts as a barometer, gauging its accuracy with each iteration of parameter updates. Until the function is close to or equal to zero, the model will continue to adjust its parameters to yield the smallest possible error.

* Gradient Descent for a function of 1 variable:

xt+1=xt−ηf′(xt)

η: is a positive number called learning rate.

* Gradient Descent for multi-variable functions:

θt+1=θt−η∇θf(θt)

Advantages: Easy to understand, easy to implement.

Disadvantage: Can converge slowly on jaw surfaces that are not convex or have many saddle points.

1. Momentum

Goal: Improve convergence speed and reduce fluctuations during optimization.

Momentum is an optimization technique used during machine learning model training to improve convergence speed and reduce fluctuations in the weight space.

*vt*​=*β*⋅*vt*−1​+(1−*β*)⋅∇*J*(*θt*​)

* Advantages: In some cases, especially in deep learning, momentum can help support delayed learning by retaining momentum from previous steps.

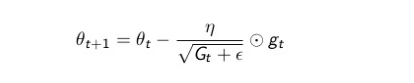
When approaching the optimal point, momentum helps the model overcome optimal thresholds and continue to learn better, avoiding the phenomenon of early stopping.

* Disadvantages: : The performance of momentum depends largely on the correct choice of momentum coefficient β. An inappropriate value can lead to unwanted fluctuations or the risk of "overshooting" the optimal point.

The performance of momentum can vary depending on the characteristics of the loss function, and is sometimes not always beneficial.

1. Adagrad

Adagrad is an adaptive optimization algorithm designed to automatically adjust the learning rate of each parameter over time.



there :

n : constant

gt : gradient at time t

ϵ: error avoidance factor (divided by sample equals 0)

G: is a diagonal matrix where each element on the diagonal (i,i) is the square of the parameter vector derivative at time t.

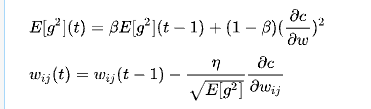
Adaptive Learning Rate: Based on gradient accumulation, it adjusts the learning rate for each parameter so that parameters with larger gradients are updated with a lower learning rate and vice versa.

* Advantages:
* Adagrad automatically adjusts the learning rate for each parameter based on the square of the gradient, reducing the learning rate for parameters with large gradients and increasing the learning rate for parameters with small gradients.
* Good for problems using sparse data because it automatically adjusts the learning rate for each parameter based on their gradient history.
* Disadvantages:
* Due to the accumulation of squared gradients, Adagrad can accumulate too fast and lead to a significant decrease in learning rate after a large number of training steps. This can cause the learning rate to become too small and cause learning lag.
* Not every problem is suitable for Adagrad, especially when there is data imbalance or when there are parameters with large gradients.

1. RMSporp

RMSprop is an algorithm optimization designed to reduce the learning rate in a manner consistent with the history of the gradient, driving improvements in the algorithm's performance during machine learning training.

RMSprop retains part of the gradient history information by accumulating the square of the previous gradient. The learning rate is adjusted based on the square root of the weighted average of the gradient squares.



* Advantages:
* RMSprop automatically adjusts the learning rate based on historical gradients, helping to reduce the risk of the learning rate being too large or too small.
* Effective in problems using sparse data because it can handle different gradient magnitudes for each parameter.
* Disadvantages: RMSprop can cause overaccumulation and lead to a significant decrease in learning rate after a large number of training steps, especially when used for deep learning models.

II)

1. Continuous learning:

* Continuous learning is a learning method in which a modeling machine is continuously trained on new data without having to retrain from scratch on the entire data. The goal is to retain and update previously learned knowledge when adding in new knowledge.
* Characteristic: The main risk is catastrophic forgetting, where the model can forget knowledge from previous data sets when learning from new data. The task is to minimize this phenomenon through methods such as reusing old data, regularization techniques, or knowledge transfer.
* Application: Continuous learning is important in systems that require continuous learning such as in reinforcement learning robotics, where the robot needs to learn from the new environment and tasks it encounters.

1. Test production:

* Test Production is a method in building machine learning solutions where the testing process is deeply integrated into the production and deployment process of the machine learning model. The goal is to ensure that the model works properly and effectively in a real-world environment.
* Characteristic: Testing steps are not only part of the model development process, but are also closely related to the process of deploying and maintaining the model in a production environment.
* Application: In mission-critical systems such as healthcare or automated driving, test production helps ensure model safety and performance under real-world conditions and in the face of new data diversity.

Integration in problem solving:

Continuous Learning: When solving a problem that involves ever-evolving data or changing requirements, Continuous Learning ensures that the model remains adaptable and can take advantage of new information without influence past knowledge.

Production Testing: As part of the development lifecycle, Production Testing is essential to verify that the model performs well across different scenarios and identify potential issues before deployment.