18AIC301J: DEEP LEARNING TECHNIQUES

B. Tech in ARTIFICIAL INTELLIGENCE, 5th semester

Faculty: **Dr. Athira Nambiar**

Section: A, slot:D

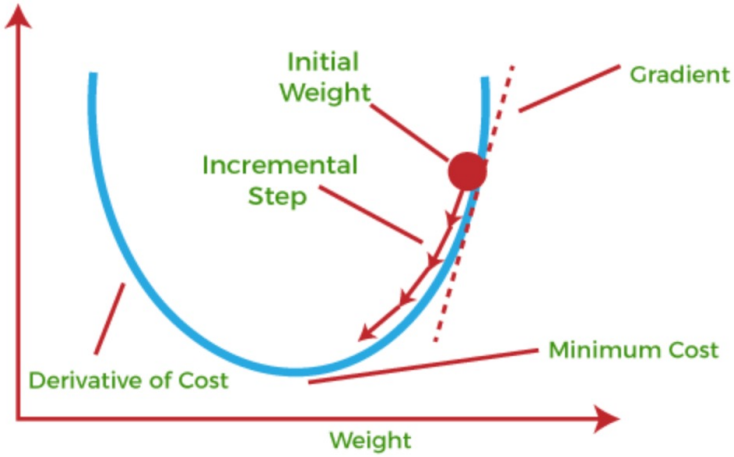
Venue: TP 804

Academic Year: 2022-22

UNIT-2

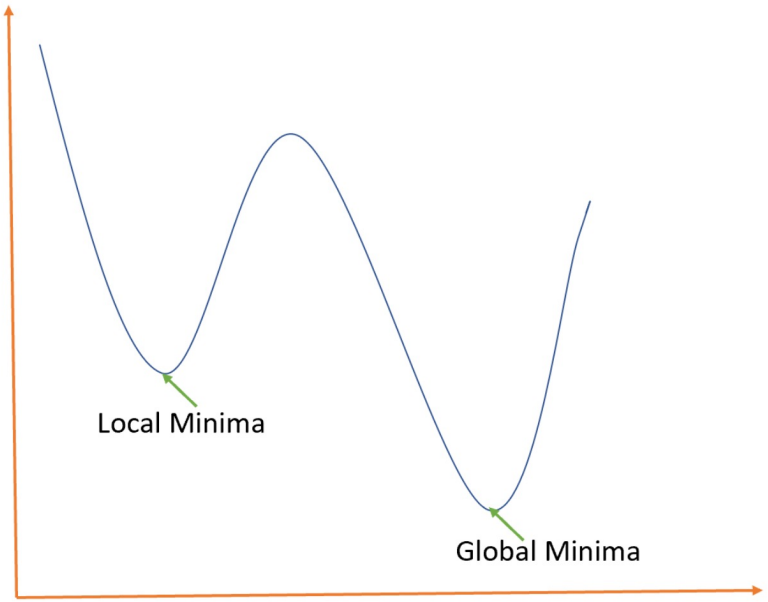
Limitations of gradient descent learning algorithm, Contour maps,
Momentum based gradient descent, Nesterov accelerated gradient Descent, AdaGrad, RMSProp, Adam learning Algorithm, Stochastic gradient descent
Implement linear regression with stochastic gradient descent
Mini-batch gradient descent, Bias Variance tradeoff, Overfitting in deep neural networks,
Hyperparameter tuning, Regularization: L2 regularization, Dataset Augmentation and Early stopping
Implement linear regression with stochastic mini-batch gradient descent and compare the results with previous exercise.
Dimensionality reduction, Principal Component Analysis, Singular value decomposition
Autoencoders, Relation between PCA and Autoencoders, Regularization in Autoencoders
Optimizing neural networks using L2 regularization, Dropout, data augmentation and early stopping.

Limitations of gradient descent learning algorithm



- Cost function(C) or Loss function measures the difference between the actual output and predicted output from the model.
- Cost function are a convex function.
- The higher the gradient, the steeper the slope and the faster a model can learn.
- But if the slope is zero, the model stops learning. In mathematical terms, a gradient is a partial derivative with respect to its inputs.

Limitations of gradient descent learning algorithm



Global minima is minimum point for the entire domain and local minima is a sub optimal point where we get a relatively minimum point but is not the global minimum point.

How can we avoid local minima and always try and get the optimized weights based on global minima?

Limitations of gradient descent learning algorithm

Learning Rate

Learning rate controls how much we should adjust the weights with respect to the loss gradient. Learning rates are randomly initialized.

$$w := w - \alpha \frac{\partial c}{\partial \omega}$$

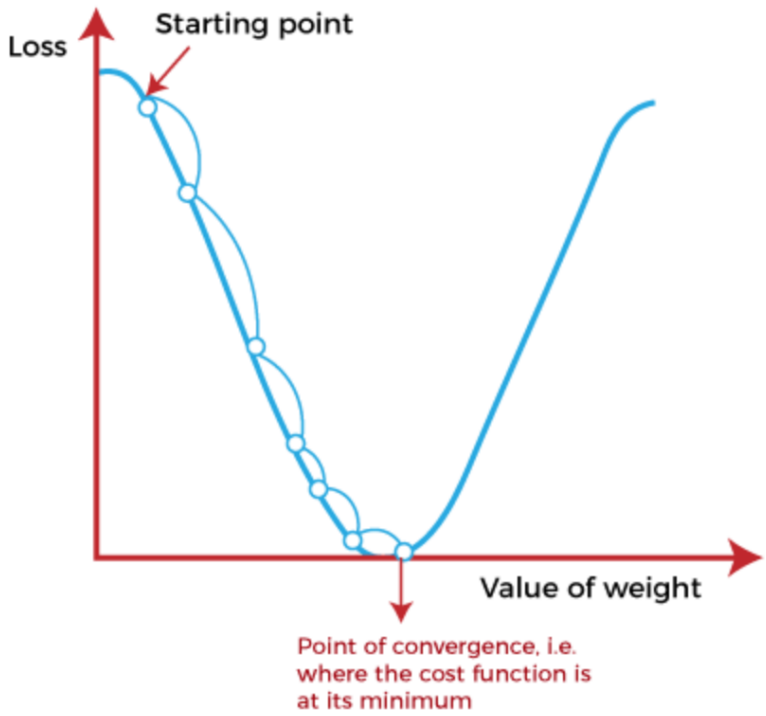
Lower the value of the learning rate, slower will be the convergence to global minima.

A higher value for learning rate will not allow the gradient descent to converge

Limitations of gradient descent learning algorithm

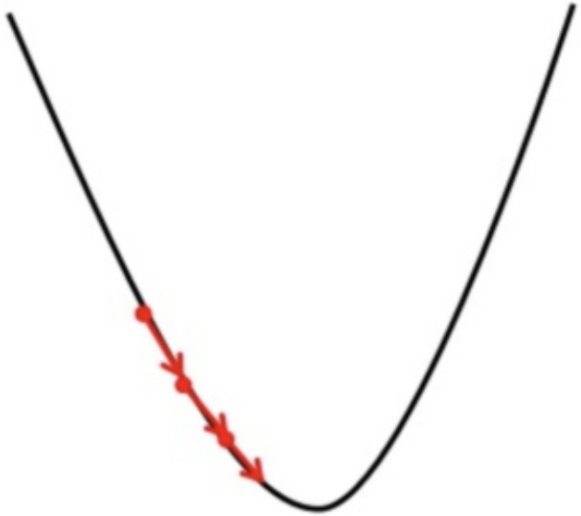
Learning Rate:

- It is defined as the step size taken to reach the minimum or lowest point.
- This is typically a small value that is evaluated and updated based on the behavior of the cost function.
- If the learning rate is high, it results in larger steps but also leads to risks of overshooting the minimum.
- At the same time, a low learning rate shows the small step sizes, which compromises overall efficiency but gives the advantage of more precision.



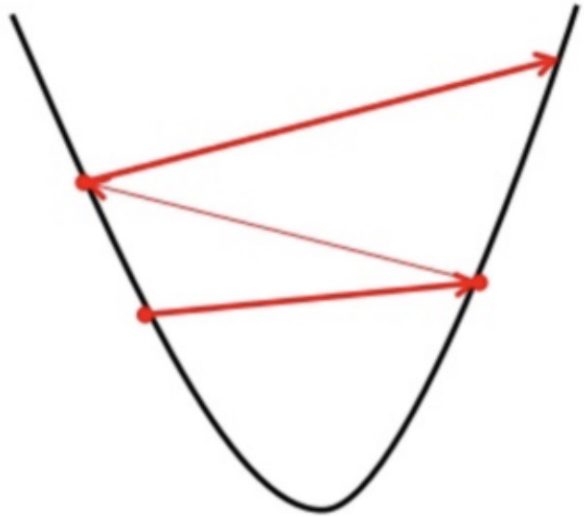
Limitations of gradient descent learning algorithm

Small learning rate



If we set the learning rate to a very small value, gradient descent will eventually reach the local minimum but that may take a while.

Big learning rate

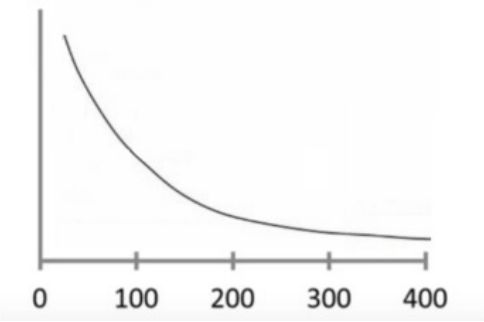


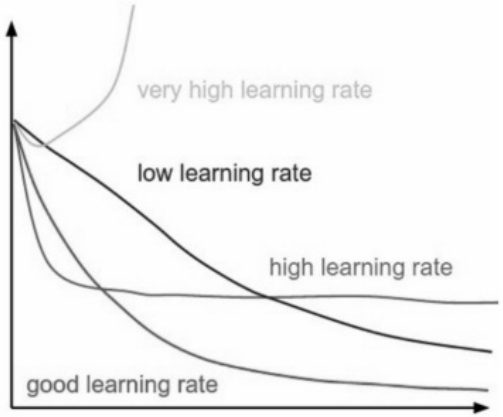
If the steps are too big, it may not reach the local minimum because it bounces back and forth between the convex function of gradient descent

Limitations of gradient descent learning algorithm

In order for Gradient Descent to work we must set the learning rate to an appropriate value.

A good way to make sure gradient descent runs properly is by plotting the cost function as the optimization runs. Put the number of iterations on the x-axis and the value of the cost-function on the y-axis.





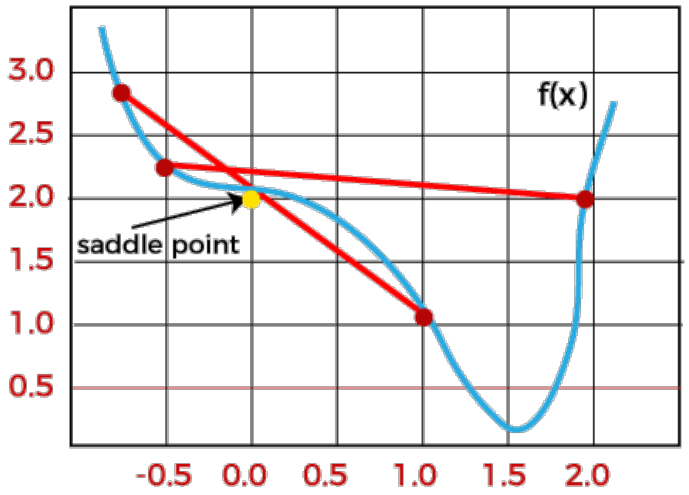
This helps you see the value of your cost function after each iteration of gradient descent, and provides a way to easily spot how appropriate your learning rate is.

Limitations of gradient descent learning algorithm

Local Minima and Saddle Point:

For convex problems, gradient descent can find the global minimum easily, while for non-convex problems, it is sometimes difficult to find the global minimum

Whenever the slope of the cost function is at zero or just close to zero, this model stops learning further.

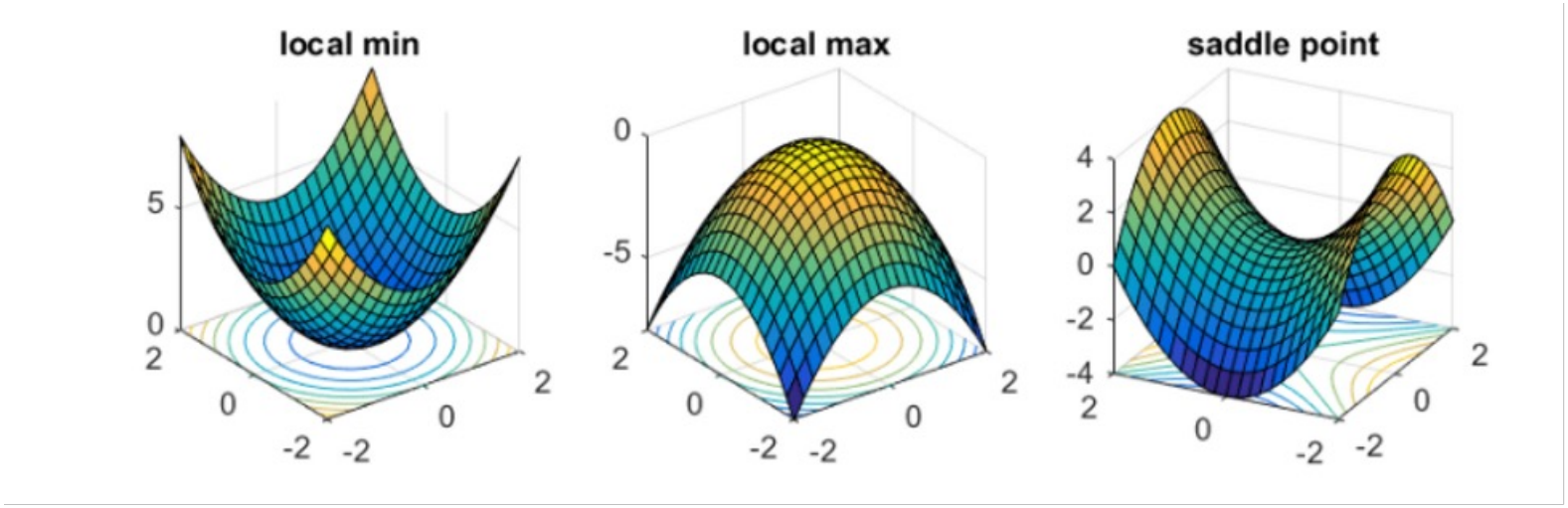


Local minima generates the shape similar to the global minimum, where the slope of the cost function increases on both sides of the current points.

In contrast, with **saddle points,** the negative gradient only occurs on one side of the point, which reaches a local maximum on one side and a local minimum on the other side.

Limitations of gradient descent learning algorithm

Saddle point on the surface of loss function is a point where, from one perspective, that critical point looks like a local minima, while from another perspective, it looks like a local maxima.



Saddle point injects confusion into the learning process. Model learning stops (or becomes extremely slow) at this point, thinking that “minimum” has been achieved since the slope becomes zero (or very close to zero).

https://www.offconvex.org/2016/03/22/saddlepoints/

Limitations of gradient descent learning algorithm

Vanishing and Exploding Gradient

In a deep neural network, if the model is trained with gradient descent and backpropagation, there can occur two more issues other than local minima and saddle point.

Vanishing Gradients:

Vanishing Gradient occurs when the gradient is smaller than expected. During backpropagation, this gradient becomes smaller that causing the decrease in the learning rate of earlier layers than the later layer of the network. Once this happens, the weight parameters update until they become insignificant.

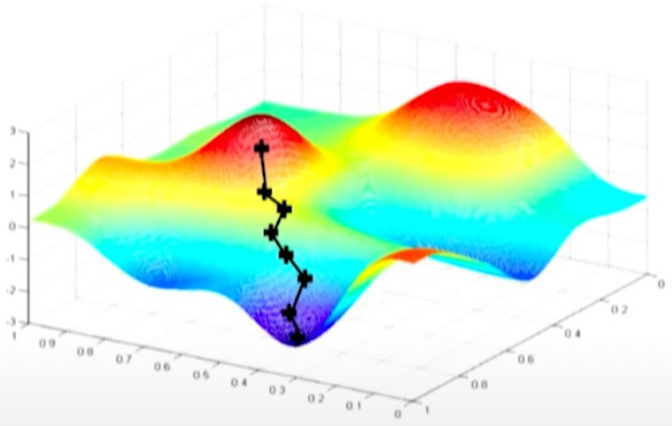
Exploding Gradient:

Exploding gradient is just opposite to the vanishing gradient as it occurs when the Gradient is too large and creates a stable model. Further, in this scenario, model weight increases, and they will be represented as NaN. This problem can be solved using the dimensionality reduction technique, which helps to minimize complexity within the model.

Gradient Descent

Algorithm

1. Initialize weights randomly $\sim \mathcal{N}(0, \sigma^2)$
2. Loop until convergence:
3. Compute gradient, $\frac{\partial J(\boldsymbol{W})}{\partial \boldsymbol{W}}$
4. Update weights, $\boldsymbol{W} \leftarrow \boldsymbol{W} - \eta \frac{\partial J(\boldsymbol{W})}{\partial \boldsymbol{W}}$
5. Return weights



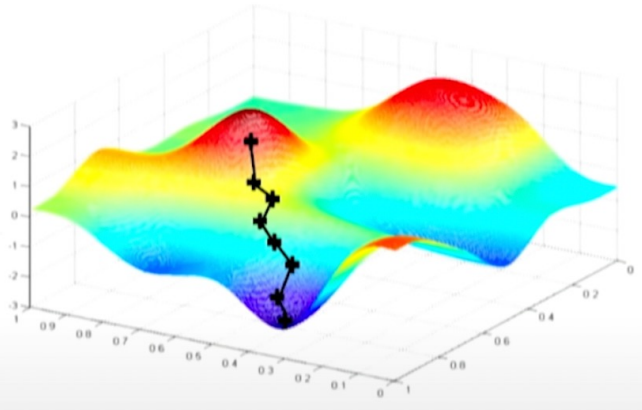
Can be very **computationally intensive** to compute!

Stochastic Gradient Descent

Algorithm

1. Initialize weights randomly $\sim \mathcal{N}(0, \sigma^2)$
2. Loop until convergence:
3. Pick single data point i
4. Compute gradient, $\frac{\partial J_i(W)}{\partial W}$
5. Update weights, $W \leftarrow W - \eta \frac{\partial J(W)}{\partial W}$
6. Return weights

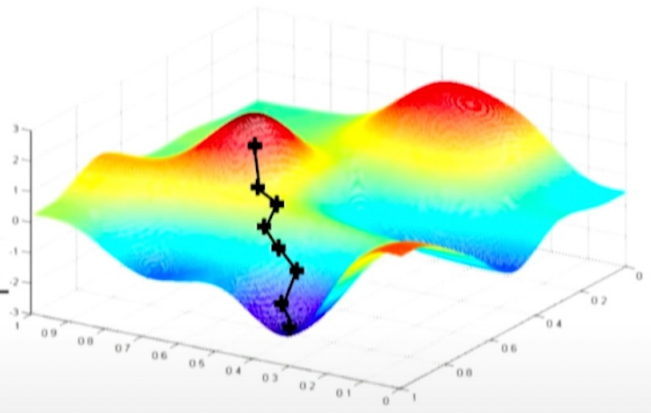
Easy to compute but **very noisy** (stochastic)!



Mini-batch gradient descent algorithm

Algorithm

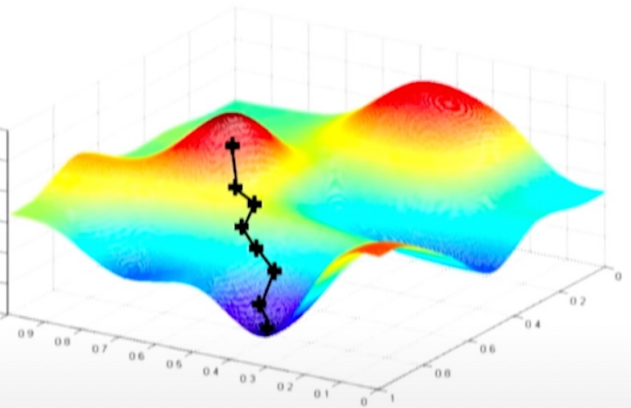
1. Initialize weights randomly $\sim \mathcal{N}(0, \sigma^2)$
2. Loop until convergence:
3. Pick batch of B data points
4. Compute gradient, $\frac{\partial J(\boldsymbol{W})}{\partial \boldsymbol{W}} = \frac{1}{B} \sum_{k=1}^{B} \frac{\partial J_k(\boldsymbol{W})}{\partial \boldsymbol{W}}$
5. Update weights, $\boldsymbol{W} \leftarrow \boldsymbol{W} - \eta \frac{\partial J(\boldsymbol{W})}{\partial \boldsymbol{W}}$
6. Return weights



Mini-batch gradient descent algorithm

Algorithm

1. Initialize weights randomly $\sim \mathcal{N}(0, \sigma^2)$
2. Loop until convergence:
3. Pick batch of B data points
4. Compute gradient, $\frac{\partial J(W)}{\partial W} = \frac{1}{B} \sum_{k=1}^{B} \frac{\partial J_k(W)}{\partial W}$
5. Update weights, $W \leftarrow W - \eta \frac{\partial J(W)}{\partial W}$
6. Return weights



Fast to compute and a much better estimate of the true gradient!

Mini-batches while training

- More accurate estimation of gradients
- Smoother convergence
- Allows for larger learning rates

- Mini-batches lead to fast training!
- Can parallelise computation+achieve significant speed increase on GPU's.

Learning Resources

- Charu C. Aggarwal, Neural Networks and Deep Learning, Springer, 2018.
- Eugene Charniak, Introduction to Deep Learning, MIT Press, 2018.
- Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press, 2016.
- Michael Nielsen, Neural Networks and Deep Learning, Determination Press, 2015.
- Deng & Yu, Deep Learning: Methods and Applications, Now Publishers, 2013.
- https://www.youtube.com/watch?v=7sB052Pz0sQ&t=2529s
- https://www.javatpoint.com/gradient-descent-in-machine-learning
- http://blog.datumbox.com/tuning-the-learning-rate-in-gradient-descent/\
- https://www.encora.com/insights/problems-with-gradient-descent

Thank you