Optimisation Techniques

Problems with vanilla - SGD

- **High Condition Number** - Very slow progress when the loss function has a high condition number , i.e. ratio of largest to smallest singular value of Hessian Matrix is large. (Hessian Matrix - a square matrix of 2nd order partial derivatives). Intuition - When the gradient of a parameter is high in one direction, and another parameter has a small gradient in another direction (which is the optimum path), SGD takes a lot of time to take significant updates towards the second parameter gradient.

- **Local Minima** - It is very likely for SGD to get stuck at local minima, as the Loss function is not completely convex. The gradient at local minima is 0, and thus parameters are not updated.

- **Saddle Point** - Gradient at this point is 0, and are of opposite signs on the two sides of this point. Saddle point is a higher problem than local minima, as it is very easy for two parameters to have different signs on gradient in a very high dimensional space. Even the regions around saddle point have small gradients, thus, slow progress.

- **Mini-Batch** - Gradients that come from a mini-batch are noisy, hence takes time to converge.

The main problem with SGD is that it is a First Order optimisation method. It does not take into account the curvature of loss function at points.

Second-Order optimisation methods -

Newton - Method - It makes a quadratic approximation of the loss function, and then jumps right to the minimum.
It does not have any learnable parameter(No learning rate)
Disadvantage - It involves the calculation of Hessian matrix and also inverting it. Which is computationally very expensive (as the Hessian Matrix is a NxN matrix).

Since we cannot use second-order methods, we try to use it's idea to take into account how gradient changes itself.

SGD + Momentum

This basically is the exponential weighted average of the past gradients, and updates parameters using the weighted average.
It resolves the problem of saddle point, local minima , and also High condition number.

RMS prop

It uses the exponential weighted average of the squares of the gradient. Update is done by dividing the gradient with the square root of the weighted average.
This method too reduces the High Condition number of loss function.

If a parameter had a large gradient then, dividing the gradient with its weighted average would rescale it to desirable order. (same for a parameter with small gradient).

Adam

It incorporates the ideas of both Momentum and RMS prop. It also has an extra bias correction, to give some weight to initial gradients.