**Problem:** The networks (Alexnet, Googlenet) fail to converge with a poor initial learning rate. And when the networks converge with a better initial learning rate, it shows an increasing problem of overfitting as it converges.

**Need:** The networks converge while keeping a good fit throughout.

**Factors to consider:**

***Initial learning rate (0.005, 0.0001, 0.00001):*** Explained in the previous document. Lowering the initial learning rate can prevent gradient explosion, however having it too low may fail to reach the most optimal loss and RMSE [1].
***Solver for train network (sgdm, rmsprop, adam):*** The three most optimal solvers for deep learning are SGDM (Stochastic Gradient Descent with Momentum), RMProp (root mean square propagation), and ADAM (adaptive moment estimation). While all three adaptive methods take contrasting approaches, they can all converge to an optimal local minimum [2].
***L2Regularization (1.0, 0.1, 0.01):*** L2Regularization, also known as weight decay, is one of the main techniques of regularization used to solve the problem of overfitting. While another optimal solution is dropout, we do not have to worry about that as networks such as Alexnet already include dropouts to solve overfitting. A large L2Regularization factor causes the network to prefer smaller weights, while a small factor causes the network to prefer minimizing the original cost function [3].

**Testing:** After training the network on multiple different combinations of options, I decided to continue with the following options.

***Initial learning rate = 0.0001*** (as any higher would expose gradient explosion, and lower would make the network fail to converge.)
***L2Regularization = 0.1*** (as any higher would make the network fail to converge, and lower would expose overfitting)

**Results:** Training on the different solvers resulted in little difference with each solver having slight advantages in different aspects. SGDM slowly converges while staying fit at validation RMSE = 2.0236. RMSProp shows similar results with Validation RMSE = 2.0585. Adam converges faster than the previous two with validation RMSE = 1.7324 however shows slight overfitting.

**Reference:**


[1] https://towardsdatascience.com/understanding-learning-rates-and-how-it-improves-performance-in-deep-learning-d0d4059c1c10

[2] https://blog.paperspace.com/intro-to-optimization-momentum-rmsprop-adam/

[3] http://neuralnetworksanddeeplearning.com/chap3.html#overfitting_and_regularization