**Gradient Descent Algorithm: Illustration on a Neural Network**

**Members:**

1. Aadit Mani 22BCE1001
2. Vidyut Kini 22BCE1010
3. Viraj Pradhan 22BCE1473
4. Kota Harshith 22BCE5241

**Abstract**:

* Gradient descent is one of the most basic optimization techniques in use today, especially for neural networks and is used to adjust the model parameters so as to minimize a cost function.
* This report gives a detailed description on the working of gradient descent in a neural network. It also describes other types of gradient descent (i.e. batch, stochastic, and mini-batch), issues of learning rates, and optimizers i.e. momentum and Adam.
* These concepts are explained and evaluated through visualizations and experiments using neural network models dealing with real-world data.

**Introduction:**

* Most machine learning algorithms, especially in neural networks where gradient descent is notably employed to perform a minimization of the cost function, rely on this algorithm.
* The iterative process adjusts the weights of the network in the opposite direction to the gradient of the objective function, therefore decreasing the prediction error over time.
* The report looks into the relevance of the gradient descent optimization method to artificial neural networks, its different forms, learning rate effects and new generation optimizers one of which is Adam.

**Problem**:

* The problem to be solved is the effective implementation of optimizing neural network parameters using gradient ascent.
* One of the variants of gradient descent (for example, classical, stochastic, mini-batch) with their learning rates and optimize strategies selection directly influence on the training time and quality of the neural networks.
* These parameters must be the subject of careful analysis and study in order not to impair the performance of convergence and accuracy.

**Scope:**

This research examines the use of gradient descent especially as it applied to neural networks. Specifically, it deals with:

* Batch Gradient Descent
* Stochastic Gradient Descent (SGD)
* Mini-Batch Gradient Descent
* Impact of various learning rates
* Optimization techniques like Momentum and Adam
* Visualization of Convergence Behavior and Early Stopping

**Related Work:**

* Numerous research efforts have reported the usage of gradient descent for solving machine learning problems.
* First-degree gradient descent variants such as batch and stochastic methods have been widely studied and improvements as momentum, RMSProp and Adam have become inherent parts of neural networks training.
* In addition, some of the earlier researches concentrated on the performance of these algorithms especially on the speed and accuracy of different optimization techniques.
* This report seeks to expand on these aspects and gives a real example of a simple neural network.

**Design:**

1. **Overall Architecture:**

The network architecture used in illustrative purpose of these techniques of gradient descent different approaches to convergence consists of:

* **Input Layer:**

Takes the input data which is the scaled feature data.

* **Hidden Layer:**

A fully connected layer with a ReLU activation function for the non-linearity.

* **Output Layer:**

A softmax activated layer for multi-class classification problems.

The network is trained for different gradient descent techniques in order to show the differences in convergence and performance among the techniques.

1. **Interfaces Defined:**

The neural network connects to a dataset and gradient descent optimizer by interfacing like shown below:

* **Input Interface:**

The input layer of the neural network accepts preprocessed and scaled data like iris dataset or california housing dataset.

* **Optimizer Interface:**

The optimizer that is Adam, SGD or Mini-batch updates the model weights based on the gradients obtained from the loss function.

1. **Class Diagrams:**

The architecture can be illustrated with the following classes:

* **NeuralNetwork:**

The primary class used in building the model and the architecture.

* **GradientDescent:**

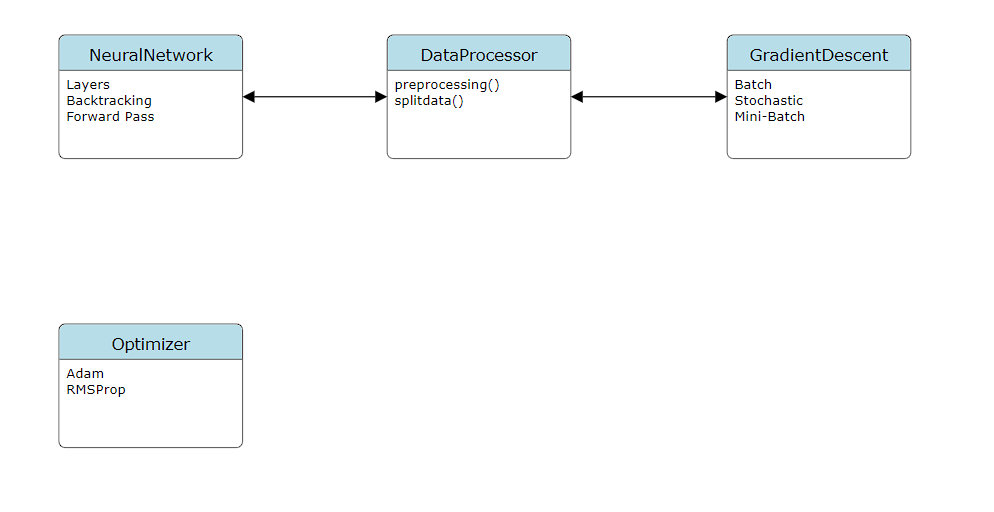
This class contains the implementation of basic batch stochastic and mini-batch gradient descent algorithms.

* **Optimizer:**

The base class for other optimizers including Adam and RMSProp.

* **DataProcessor:** Handles data programming and splitting.

**Class Diagram:**



**Observations:**

1. **Batch Gradient Descent:**

* This approach is safe and steady, but we also have to face that batch gradient descent is very slow since the entire dataset will be needed before each and every update.
* The neural network has eventually reached convergence, it is just that the overhead in computations is quite evident.

1. **Stochastic Gradient Descent (SGD):**

* With this approach of course parameters are changing every after each sample and therefore the path to convergence is a lot bumpier than with the previous modes.
* It takes less time to converge than batch gradient descent most of the times because of more data, though the results may be mixed.
* This may lead to the results being jittery because of the inconsistency of the jumps made.

1. **Mini-Batch Gradient Descent:**

* Mini-batch gradient descent is neither fully batch nor fully stochastic. It uses a small portion of the full data for the update, making it more efficient than the batch one and less prone to stochastic variations than the former.
* This method shows smooth and faster convergence for the neural network.

1. **Learning Rate Impact:**

* **Proper Learning Rate:** If the optimal learning rate is picked, the network trained is likely to reach a solution in a short time and with less oscillation.
* **Large Learning Rate:**

This tends to oscillate or diverge from the minimum because of the zoom in – zoom out steps that are very large compared to the scale of the surface.

* **Small Learning Rate:**

Most of the times results in very sluggish convergence i.e. a lot of iterations are done before the minimum is reached

1. **Optimization Techniques:**

* **Momentum:**

This technique facilitates the rate of convergence by enhancing the gradient and at the same time including the velocity term in the computation which imitates a ball going down a slope.

* **Adam:**

It is a combination of the benefits that come with using RMSProp and the momentum, therefore there is an allowance for learning rates that are adaptive, and this makes it possible to run the programs even faster in the presence of networks that are equipped with sparse gradients.

1. **Convergence Behavior:**

* As the training goes on, the cost incurred by the neural network reduces at every stage of training until a certain stage when the cost becomes constant since the network has attained ideal or optimal weights.
* Early stopping is another provision in that objectives are met without overfitting the model by stopping training when the validation loss starts to increase.

**Acknowledgements:**

* Aadit Mani 22BCE1001
* Vidyut Kini 22BCE1010
* Viraj Pradhan 22BCE1473
* Kota Harshith 22BCE5241

**References:**

* He, K., Zhang, X., Ren, S., & Sun, J. (2016). "Deep Residual Learning for Image Recognition." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
* Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). "ImageNet Classification with Deep Convolutional Neural Networks." Advances in Neural Information Processing Systems (NeurIPS).
* Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). "Learning to Forget: Continual Prediction with LSTM." Neural Computation.
* Mnih, V., Kavukcuoglu, K., Silver, D., et al. (2015). "Human-Level Control Through Deep Reinforcement Learning." Nature.
* Bottou, L. (2010). "Large-Scale Machine Learning with Stochastic Gradient Descent." Proceedings of COMPSTAT'2010.