Efficient BackProp

Key Takeaways:

1. Back Propagation is a very popular neural network learning algorithm as it is conceptually simple and computationally efficient.
2. Arbitrary choices are made during designing and training a neural network in terms of number and types of nodes, layers, learning rates, training and test sets and so forth but these choices are critical to determine the effectiveness of the output.
3. A great overview of Gradient based learning machine which focused on strategies for improving the process of minimizing the cost function and the tricks associated with increasing the speed and quality of the minimization.
4. Two main types for learning:
    a. Batch – Learn after training entire dataset, slow and can get stuck.
    b. Stochastic – Learn after every example, faster and noise find better minima.
5. Stochastic learning wins the race for the following reasons:
    a. Speed: Much faster, especially on large, redundant datasets
    b. Better Solutions: Noise can help escape local minima to find better ones.
    c. Adaptability: Can be used to track changes in the data over time.
6. Good practice to shuffle the training set so successive training samples rarely belong to the same class. This speeds up the learning process.
7. If all the inputs are positive, weight updates have the same sign – Inefficient/slow
    a. Shift the Mean: Make the average of each input variable close to zero.
    b. Scale Co-variances: Scale inputs so that the covariances are about the same.
    c. Decorrelate: Input variables should be uncorrelated if possible.
8. Smart Starts & Steps:
    a. Standard Logistic Sigmoid function: Do Not use this as the outputs are always positive which can slow the learning in subsequent layers.
    b. Hyperbolic Tangent: Symmetric keeps the average near zero – speeds up learning.
9. Weight Initialization:
    a. Goal: Activate the Sigmoid in its linear region.
    b. Why: Avoids saturated units or tiny gradients
    c. Formula: Draw weights with mean 0 and standard deviation $\sigma_w = m^{(-½)}$
10. Training a network is like navigating a bumpy, high dimensional landscape called cost surface. The learning problem is to find the value of weights (W) that minimized the cost function (E_{train})
11. The Hessian Matrix: A matrix of second derivatives that measures the curvature of error surface. Tricks reshape the error surface from the taco shell into more smooth spherical bowl making the path to the minimum faster.
12. This paper discusses about Newton Algorithm (impractical for large neural networks) and Conjugate Gradient and Quasi-newton (BFGS) (batch learning)
13. The paper proposes methods for approximating and using second-order information in a more scalable way.
    a. **Stochastic Diagonal Levenberg Marquardt - large** classification problems,
    b. **Computing the Principal Eigenvalue/Vector:**

Tools: NotebookLM helped me review and understand the details of this paper