GRADIENT DESCENT PROVABLY OPTIMIZERS OVER-PARAMETERIZED NEURAL NETWORKS

SIMON S.DU, XIYU ZHAI, BARNABAS POCZOS, AARTI SINGH

University at Buffalo The State University of New York

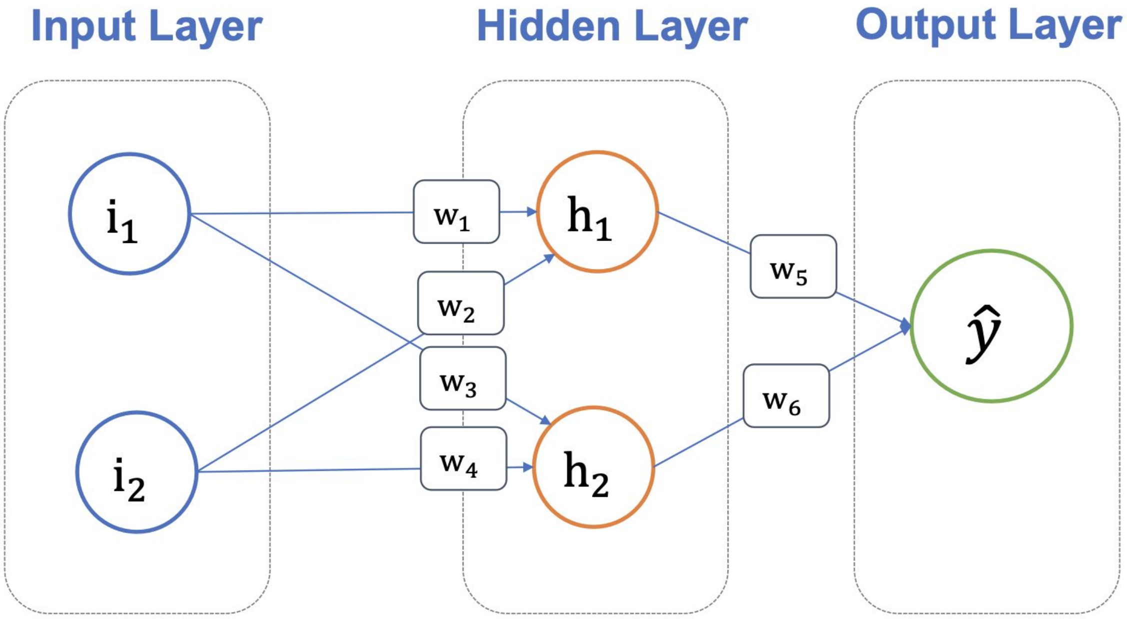
INTRODUCTION

- What is the motive
- What optimization algorithm is being used for the neural network
- What consideration or assumptions are made for proving non-convex and non smooth can achieve global minima

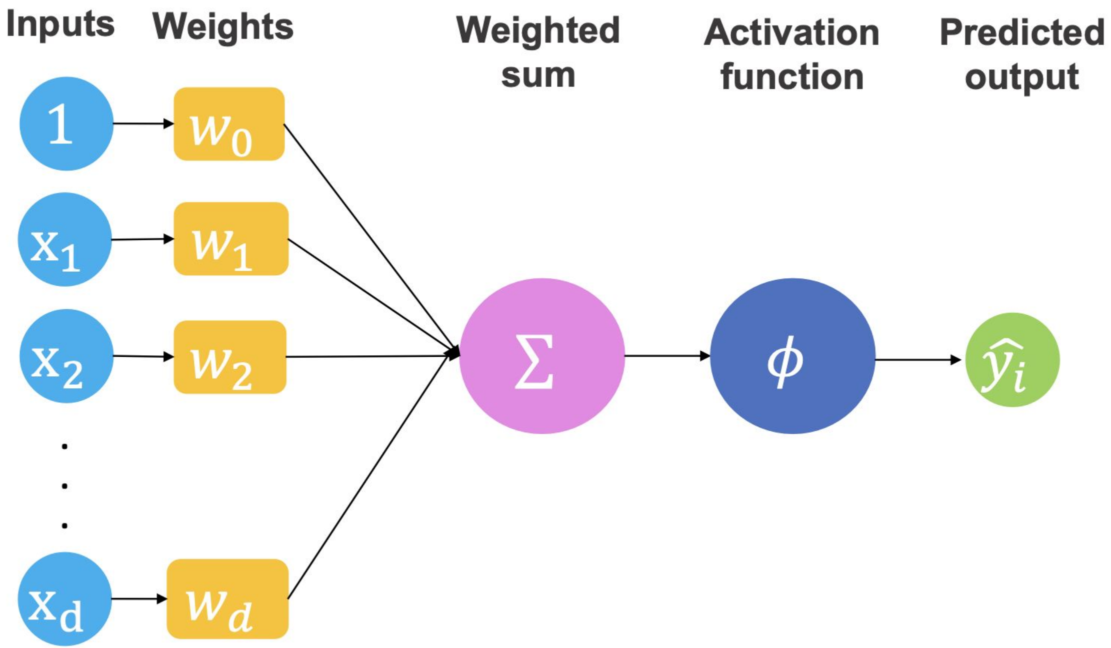
BACKGROUND

- Neural Network Basics (Forward and Backward Propagation)
- Activation Function
- What is convex & non-convex function
- Overfitting and how is it related to this paper ?
- What is objective function or loss function
- How does gradient descent optimizer achieve global minima by adjusting weights
- Over parameterized neural network

NEURAL NETWORK



NEURAL NETWORK



PREVIOUS RESULTS

Landscape Analysis

- Design of optimization algorithms
- Identify initialization methods that and hyperparameters that lead to faster convergence and better performance.

Analysis of Algorithm Dynamics

- Convergence behavior of the algorithm
- Identify the factors that influence its performance
- Studied in terms of Trajectory of Model

DYNAMICS OF PREDICTIONS

Neural Network : $f(\mathbf{W}, \mathbf{a}, \mathbf{x}) = \frac{1}{\sqrt{m}} \sum_{r=1}^{m} a_r \sigma\left(\mathbf{w}_r^\top \mathbf{x}\right)$

Loss Function: $L(\mathbf{W}, \mathbf{a}) = \sum_{i=1}^{n} \frac{1}{2} \left(f(\mathbf{W}, \mathbf{a}, \mathbf{x}_i) - y_i\right)^2$

Gradient Descent Optimizer: $\mathbf{W}(k+1) = \mathbf{W}(k) - \eta \frac{\partial L(\mathbf{W}(k), \mathbf{a})}{\partial \mathbf{W}(k)}$

Gradient Descent Weight Vector: $\frac{\partial L(\mathbf{W}, \mathbf{a})}{\partial \mathbf{w}_r} = \frac{1}{\sqrt{m}} \sum_{i=1}^{n} (f(\mathbf{W}, \mathbf{a}, \mathbf{x}_i) - y_i) \mathbf{a}_r \mathbf{x}_i \mathbb{I}\left\{\mathbf{w}_r^\top \mathbf{x}_i \geq 0\right\}$

CONVERGENCE RATE OF GRADIENT FLOW

- Gradient flow with infinitesimal step size
- This theorem establishes that if m is large enough, the training error converges to 0 at a linear rate. m= $\Omega\left(\frac{n^6}{\lambda_0^4\delta^3}\right)$ (m→Hidden Nodes)
  - n-> number of samples, Lambda-> regularization, Delta-> amount of noisy data.
- Gram Matrix induced by activation function.
  - (Objective) To check the closeness of later iterations to that of the initialization phase. [EigenValue, EigenVector]
- Paper
- Regularization

CONVERGENCE RATE OF GRADIENT DESCENT

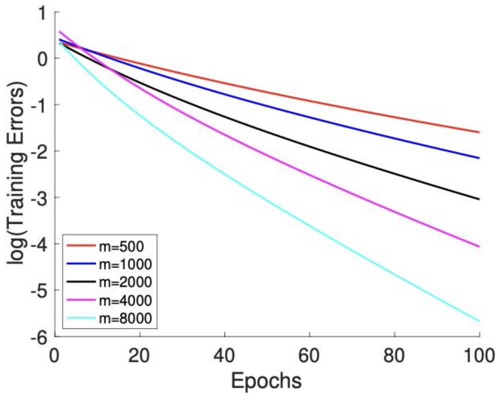
- Randomly initialized gradient descent with a constant positive step size converges to the global minimum at a linear rate?
- What is step function?
- Even though the objective function is non-smooth and non-convex, gradient descent with a constant step size still enjoys a linear convergence rate?
- Is that all?
- Lipschitz continuous Regularizer : $|f(x) - f(y)| \leq K * |x - y|$
  - K is a measure of how fast the function can change.
- Bound on the rate at which the function can change.
- Matrix perturbation analysis tool to show most of the patterns do not change

FINALLY !

- Over-parameterization, Random initialization, and the Linear convergence jointly restrict every weight vector to be close to its initialization.

EXPERIMENTS

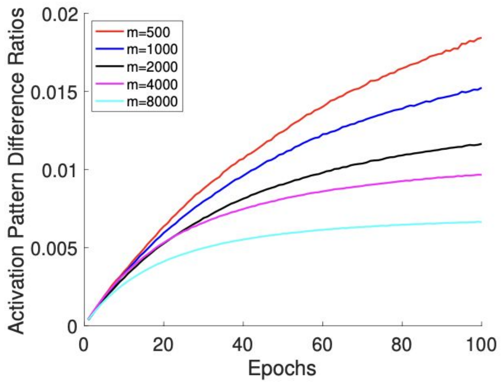
- Epoches =100 of Gradient Descent
- Fixed Step Size
- Uniform Generations of n=1000 data points



(a) Convergence rates.

EXPERIMENTS

- Epoches =100 of Gradient Descent
- Fixed Step Size
- Uniform Generations of n=1000 data points

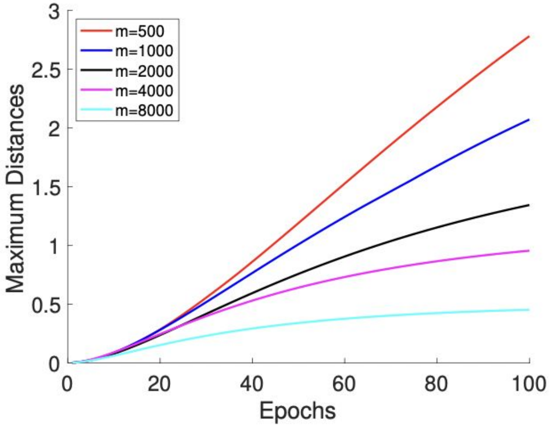


(b) Percentiles of pattern changes.

The reason is as m becomes larger, H(t) matrix becomes more stable

EXPERIMENTS

- Epoches =100 of Gradient Descent
- Fixed Step Size
- Uniform Generations of n=1000 data points



(c) Maximum distances from initialization.

Percentiles of pattern changes and the maximum distance from the initialization become smaller

CONCLUSION

In this paper we show with over-parameterization, gradient descent provable converges to the global minimum of the empirical loss at a linear convergence rate. The key proof idea is to show the over-parameterization makes Gram matrix remain positive definite for all iterations, which in turn guarantees the linear convergence.

FUTURE DISCUSSIONS

- Width Shrinking
- Check with other Loss Functions

THANK YOU :)