SIMULTANEOUS PERTURBATION STOCHASTIC APPROXIMATION

SPSA

Chidaksh Ravuru

IIT Dharwad

November 21, 2022

Outline

Overview

- BACKPROPAGATION BASED NEURAL NETWORK
- SPSA vs GRADIENT DESCENT
- SPSA BASED NEURAL NETWORK

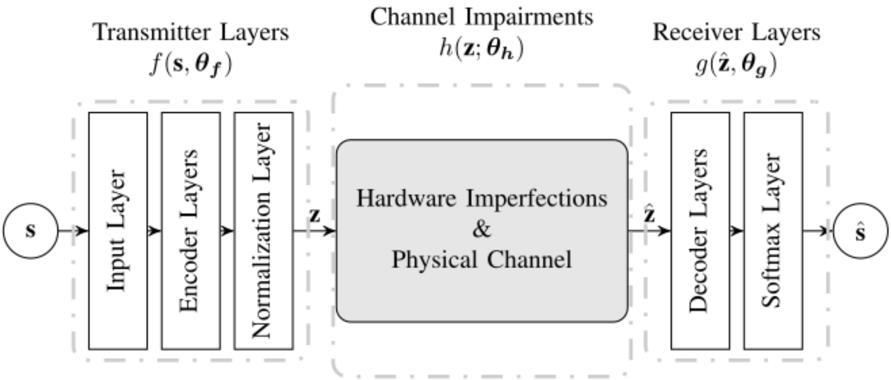

Figure: General Neural Net Framework for an end-to-end communication [2]

Neural Network Configurations

Number of encoder layers	2
Hidden encoder layer dimensions	{16,7}
Number of Decoder layers	2
Hidden decoder layer dimensions	{7, 16}
Batch size	16
Output Layer	LogSoftmax
Activation Function	RELU
Loss Function	NLLLoss
Optimizer	Adam Optimizer

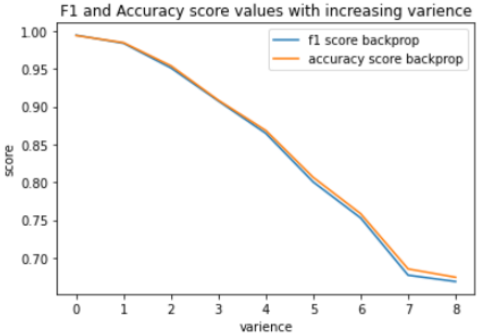

Figure: F1 and Accuracy Score plots for NN with backpropagation

SPSA Algorithm

Algorithm 1 SPSA Algorithm

Parameters: $a > 0, A \geq 0, c > 0, \alpha \in (0,1], \gamma \in (1/6, 1/2]$ and a distribution $\mathcal{D}$.
for $k = 1, 2, 3, \ldots$ **do**
    Sample a vector $\boldsymbol{\Delta} \sim \mathcal{D}$
    $a_k = \frac{a}{(k+A)^\alpha}$
    $c_k = \frac{c}{k^\gamma}$
    $\hat{g} = \frac{J(\boldsymbol{\theta}+c_k\boldsymbol{\Delta}) - J(\boldsymbol{\theta}-c_k\boldsymbol{\Delta})}{2\ c_k\boldsymbol{\Delta}}$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - a_k \cdot \hat{g}$
end for

Figure: SPSA Algorithm [2]

SPSA and Gradient Descent Convergence

Gradient Descent is also proved to converge for Differentiable and convex functions [3], we can use subgradient descent variatio for non-differentiable functions. SPSA method is proved to converge for continuous (need not be differentiable) convex functions [1]

SPSA vs Gradient Descent

We compared Gradient Descent vs SPSA for many functions,

Function	Algorithm	Convergence Value	Average Time Taken	Final Point of Convergence
$y = x^2 + x + 2$	Back Propagation	-0.5	**149 ms ± 2.19 ms**	-0.5000000000000027
	SPSA		309 ms ± 4.2 ms	-0.500000000000001
$y = xsinx$	Back Propagation	-5	**152 ms ± 2.07 ms**	4.913180439434884
	SPSA		315 ms ± 6.03 ms	4.91317337
$y = \|x\|$	Back Propagation	0	**255 ms ± 115 ms**	0.0085930954226148
	SPSA		366 ms ± 106 ms	$-8.93737133e-20$

In the third case, we used SubGradient Descent instead of Gradient Descent as we can't use Gradient Descent for a non-differentiable function. This clearly shows SPSA can dominate over subgradient Descent for non-differentiable function at the cost of time.

For a convex function $f : \mathbb{R}^n \to \mathbb{R}$,

$$f(y) \geq f(x) + \nabla f(x)^T (y - x)$$

for all x,y. Hence, linear approximation always underestimates f.
A **subgradient** of convex $f : \mathbb{R}^n \to \mathbb{R}$ at any x is $g \in \mathbb{R}^n$ such that,

$$f(y) \geq f(x) + \nabla g^T (y - x)$$

for all y,

- Always exist
- If f differentiable at x, then $g = \nabla f(x)$

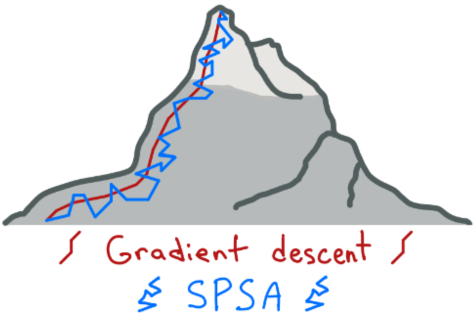

Figure: Image taken from the Internet showing convergence paths taken by SPSA and Gradient Descent [link]

Neural Network with SPSA

Using the algorithm mentioned in the above figure 3 we implemented a Neural Network with the following values of the SPSA hyperparameters and the general hyperparameters as mentioned in the above table 5.

a	0.05
A	25
c	0.1
alpha	0.9
beta	0.3

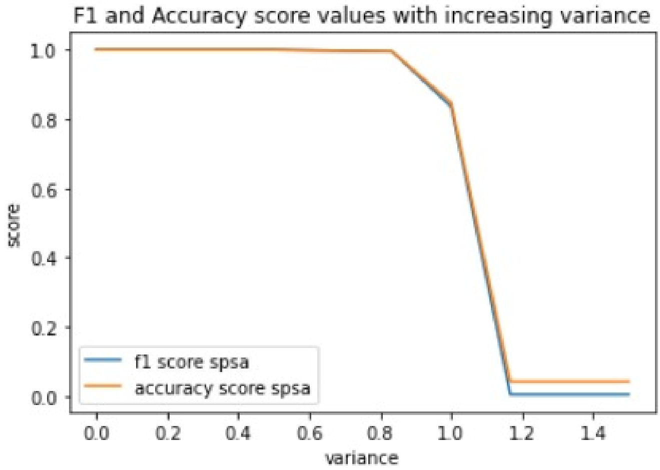

Figure: Neural Net with SPSA

References

[1] Ying He, M.C. Fu, and S.I. Marcus. "Convergence of simultaneous perturbation stochastic approximation for nondifferentiable optimization". In: *IEEE Transactions on Automatic Control* 48.8 (2003), pp. 1459–1463. DOI: 10.1109/TAC.2003.815008.

[2] Vishnu Raj and Sheetal Kalyani. "Backpropagating Through the Air: Deep Learning at Physical Layer Without Channel Models". In: *IEEE Communications Letters* 22.11 (2018), pp. 2278–2281. DOI: 10.1109/LCOMM.2018.2868103.

[3] Wardi Shapiro A. "Y. Convergence analysis of gradient descent stochastic algorithms.". In: *Journal of Optimization Theory and Applications* 91.2 (1996), pp. 439–454. DOI: 10.1007/BF02190104.