# Simultaneous Propagation

Daniel Kunin

April 2018

## 1 Numerical Linear Algebra Error Analysis

In numerical linear algebra we are very concerned with how error propagates through an algorithm. Because numbers cannot be represented to infinite precision, roundoff errors are introduced into any calculation. Our goal is to bound this error and design algorithms that minimize its propagation.

To this end we generally talk about two types of error: Forward Error and Backward Error. Let us assume our algorithm can be represented as a surjective function $f : X \to Y$ (a map from the input space $X$ to the output space $Y$). Let $\tilde{f} : X \to Y$ be the actual result of running our algorithm when accounting for roundoff error. The goal of error analysis in numerical linear algebra is to understand the relationship between $f$ and $\tilde{f}$. The two types of error can formally and pictorially be defined as follows:

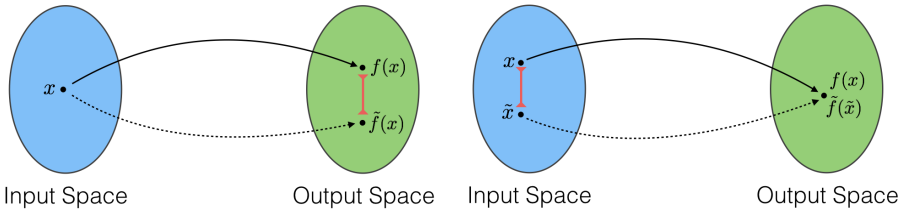


Figure 1: Forward Error (left) is the distance in output space for a given input: $||f(x) - \tilde{f}(x)||$. Backward Error (right) is the distance in input space such that the outputs are equal: $||x - \tilde{x}||$.

Interestingly, its actually the ratio of these two errors that we are generally most concerned with. This is called the sensitivity:

$$\frac{||f(x) - \tilde{f}(x)||}{||x - \tilde{x}||} = \frac{||\tilde{f}(\tilde{x}) - \tilde{f}(x)||}{||x - \tilde{x}||}$$

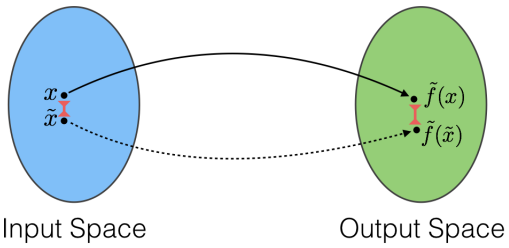


Figure 2: Sensitivity is the ratio of forward error with backward error. It can be understood as a measure of how much our output, $\tilde{f}(x)$, changes when our input, $x$, is perturbed

## 2 Applications to Deep Learning

Linear algebra is the study of linear transformations, however, the error analysis paradigm described previously is not unique to linear transformations. In fact this framework lends itself very nicely to deep learning. A deep learning task can be understood as learning a non-linear $\tilde{f}$ from a set of training samples: $(x, f(x))$. We learn $\tilde{f}$ by minimizing some loss function, generally defined as a distance or distortion measure between $f(x)$ and $\tilde{f}(x)$, evaluated on our training samples. We then evaluate our network's performance by computing the same loss on a different set of test samples. Notice that this loss function is a form of forward error. If our loss function is a form of forward error for deep learning, what is the analogous backward error and sensitivity.

Adversarial attacks are perturbed input images that are generally misclassified by neural networks, but appear indistinguishable to the original correctly classified input image. These perturbations are found by maximizing a trained network's prediction error. In other words, adversarial attacks are generated by finding small perturbations in the input space that propagate to large errors in the output space. By definition this is the sensitivity scenario for a neural network. An area of activate research is determining methods to increase the adversarial robustness of neural networks. Using the framework of error analysis from numerical linear algebra there is an obvious strategy.

In order to minimize sensitivity we must minimize forward error and maximize backward error. Minimizing the forward error is handled by our ordinary loss functions, but how can we maximize backward error? To achieve this goal we will introduce a new term to the loss function:

$$-||\tilde{X} - X||_2^2$$

While training our network we will generate input samples that are correctly classified by our network to our training labels, $\tilde{X}$. We can evaluate the backward error of our network at any iteration of training as the distance between these inputs and our training input, $X$. We will update the parameters of our network to both minimize forward error and maximize backward error. In doing so we will be conditioning our network's sensitivity and hopefully increasing its robustness to adversarial attacks and its capacity for generalization.

## 3 Introducing Backward Error to the Loss Function

Simultaneous Propagation - Finding a network that minimizes a loss in the output space and maximizes a loss in the input space.