Backpropagation on the exact gradient as solution to 1st order PDE.

March 3, 2019

1 Normal Backpropagation:

Through out this document, we follow the convention that indices that appear on one side of the equation and not on the other are summed over. To make it convinent later on, we split the usual neural network function into two steps (the linear transformation step and the non-linear activation step):

$$\begin{aligned} O^K_{by} &= W^K_{yx} I^K_{bx} + B^K_y \\ \text{For : } K &\geq 1 \\ \text{We have : } I^K_{bx} &\rightarrow O^{K-1}_{bx} \\ O^{K+1}_{by} &= \sigma\left[O^K_{by}\right] \\ \sigma &\Rightarrow \text{Activation Function.} \end{aligned}$$

To make this more friendly for matrix multiplication, the linear part can be written as:

$$O^K_{by} = I^K_{bx} \left(W^K\right)^T_{xy} + B^K_y$$

To learn features by backpropagation (denoting the last layer by the index L):

$$\begin{aligned} \epsilon &\equiv \left(O^L_{by} - \mathcal{Y}_{by}\right)^2 \\ \delta\epsilon &= 2\left(O^L_{by} - \mathcal{Y}_{by}\right)\left(\delta O^L_{by}\right) \end{aligned}$$

Let us denote:

$$\Delta^L_{by} \equiv 2\left(O^L_{by} - \mathcal{Y}_{by}\right)$$

Hence, we get:

$$\delta\epsilon = \Delta^L_{by}\left(\delta O^L_{by}\right)$$

Now, consider the first (linear) cases:

$$\begin{aligned} \Delta^K_{by}\delta O^K_{by} &= \left(\delta W^K_{yx}\right)\Delta^K_{by} I^K_{bx} + \Delta^K_{by} W^K_{yx}\left(\delta I^K_{bx}\right) + \Delta^K_{by}\left(\delta B^K_y\right) \\ \Delta^K_{by}\delta O^K_{by} &= \left(\delta W^K_{yx}\right)\Delta^K_{by} O^{K-1}_{bx} + \Delta^K_{by} W^K_{yx}\left(\delta O^{K-1}_{bx}\right) + \Delta^K_{by}\left(\delta B^K_y\right) \end{aligned}$$

This gives the backpropagation recursion relation:

$$\begin{aligned} \Delta^{K-1}_{bx} &= \Delta^{K}_{by} W^{K}_{yx} \\ \text{Hence : } \Delta^{K}_{by}\left(\delta O^{K}_{by}\right) &\rightarrow \Delta^{K-1}_{bx}\left(\delta O^{K-1}_{bx}\right) \end{aligned}$$

And also the gradient of W^K_{yx} and B^K_y which will be used for the gradient descent:

$$\begin{aligned} \delta W^{K}_{yx} &= \Delta^{K}_{by} O^{K-1}_{bx} = \left(\Delta^{K}\right)^{T}_{yb} O^{K-1}_{bx} \\ \delta B^{K}_{y} &= \Delta^{K}_{by} \end{aligned}$$

Using the sum convention mentioned at the beginning, we can interpret the second equation as:

$$\delta B^{K}_{y} = \sum_{b} \Delta^{K}_{by}$$

Similar relations can be derived for the activation layer:

$$\begin{aligned} \Delta^{K}_{by}\left(\delta O^{K}_{by}\right) &= \left(\sigma'\left[O^{K-1}_{by}\right]\Delta^{K}_{by}\right)\left(\delta O^{K-1}_{by}\right) \\ \text{Hence : } \Delta^{K-1}_{by} &= \Delta^{K}_{by}\sigma'\left[O^{K-1}_{by}\right] \end{aligned}$$

We now have all the recursion relations for derivatives (backpropagation) required to perform gradient descent.

2 Backpropagation on the gradient:

We now show that the above procedure can also be used to learn on the gradient of the neural network instead of the plain output of the network.

$$
\begin{aligned}
O_{by}^{K} &= W_{yx}^{K} I_{bx}^{K} + B_{y}^{K} \\
\left(\frac{dO_{by}^{K}}{dI_{bm}^{0}}\right) &= W_{yx}^{K} \left(\frac{dI_{bx}^{K}}{dI_{bm}^{0}}\right) \\
\left(\frac{dO_{by}^{K}}{dI_{bm}^{0}}\right) &= W_{yx}^{K} \left(\frac{dO_{bx}^{K-1}}{dI_{bm}^{0}}\right) \\
\mathfrak{D}_{bym}^{K} &\equiv \left(\frac{dO_{by}^{K}}{dI_{bm}^{0}}\right) \\
\mathfrak{D}_{bym}^{K} &= W_{yx}^{K} \mathfrak{D}_{bxm}^{K-1}
\end{aligned}
$$

$$
\begin{aligned}
\text{if: } \mathfrak{D}_{bxm}^{K-1} &= \delta_{xm} \quad \text{(first layer)} \\
\mathfrak{D}_{bym}^{K} &= W_{yx}^{K} \delta_{xm} \\
\mathfrak{D}_{bym}^{K} &= W_{ym}^{K}
\end{aligned}
$$

$$
\begin{aligned}
O_{bi}^{K} &= \sigma\left(I_{bi}^{K}\right) \\
\frac{dO_{bi}^{K}}{dI_{bm}^{0}} &= \sigma'\left(I_{bi}^{K}\right) \frac{dI_{bi}^{K}}{dI_{bm}^{0}} \\
\left(\frac{dO_{bi}^{K}}{dI_{bm}^{0}}\right) &= \sigma'\left(I_{bi}^{K}\right) \left(\frac{dO_{bi}^{K-1}}{dI_{bm}^{0}}\right) \\
\mathfrak{D}_{bim}^{K} &= \sigma'\left(O_{bi}^{K-1}\right) \mathfrak{D}_{bim}^{K-1} \\
\left(\mathfrak{D}_{bim}^{K} - \mathcal{A}_{bim}\right)^{2} &= \epsilon \\
2\left(\mathfrak{D}_{bim}^{K} - \mathcal{A}_{bim}\right) \partial \mathfrak{D}_{bim}^{K} &= \partial \epsilon
\end{aligned}
$$

So:

$$
\left(\mathfrak{D}_{bim}^{K} - \mathcal{A}_{bim}\right) \equiv \Delta_{bim}^{K}
$$

$$
\begin{aligned}
\mathfrak{D}_{bym}^{K} &= W_{yx}^{K} \mathfrak{D}_{bxm}^{K-1} \\
\Delta_{bym}^{K} \partial \mathfrak{D}_{bym}^{K} &= W_{yx}^{K} \Delta_{bym}^{K} \partial \mathfrak{D}_{bxm}^{K-1} \\
\Rightarrow \Delta_{bxm}^{K-1} &= W_{yx}^{K} \Delta_{bym}^{K} \\
\Rightarrow \left(\Delta_{b}^{K-1}\right)_{xm} &= \left(W^{K}\right)_{xy}^{T} \left(\Delta_{b}^{K}\right)_{ym} \\
\left(\Delta_{b}^{K-1}\right)_{xm} &= \left[\left(W^{K}\right)^{T} \Delta_{b}^{K}\right]_{xm}
\end{aligned}
$$

For evaluating W:

$$\begin{aligned}
\Delta^K_{bym}\partial\mathfrak{D}^K_{bym} &= \left(\Delta^K_{bym}\partial\right)\left[W^K_{yx}\mathfrak{D}^{K-1}_{bxm}\right] \\
&= \Delta^K_{bym}\mathfrak{D}^{K-1}_{bxm}\left(\partial W^K_{yx}\right) \\
\delta W^K_{yx} &= \Delta^K_{bym}\mathfrak{D}^{K-1}_{bxm} \\
\delta W^K_{yx} &= \left(\Delta^K_b\right)_{ym}\left(\mathfrak{D}^{K-1}_b\right)_{xm} \\
\delta W^K_{yx} &= \left(\Delta^K_b\right)_{ym}\left(\mathfrak{D}^{K-1}_b\right)^T_{mx} \\
\delta W^K_{yx} &= \left[\left(\Delta^K_b\right)\left(\mathfrak{D}^{K-1}_b\right)^T\right]_{yx} \\
\text{if}: \left(\mathfrak{D}^{K-1}_b\right)_{xm} &= \delta_{xm} \\
\delta W^K_{yx} &= \left(\Delta^K_b\right)_{ym}\left(\delta_{xm}\right) \\
\delta W^K_{yx} &= \sum_b \left(\Delta^K_b\right)_{yx} \\
\delta W^K_{yx} &= \left(\Delta^K_b\right)_{ym}\left(\mathfrak{D}^{K-1}_b\right)^T_{mx}
\end{aligned}$$

For the activation layers:

$$\begin{aligned}
\mathfrak{D}^K_{bim} &= \sigma'\left(O^{K-1}_{bi}\right)\mathfrak{D}^{K-1}_{bim} \\
\partial\mathfrak{D}^K_{bim} &= \left[\partial\sigma'\left(O^{K-1}_{bi}\right)\right]\mathfrak{D}^{K-1}_{bim} + \sigma'\left(O^{K-1}_{bi}\right)\left[\partial\mathfrak{D}^{K-1}_{bim}\right] \\
\partial\mathfrak{D}^K_{bim} &= \left[\partial O^{K-1}_{bi}\right]\mathfrak{D}^{K-1}_{bim}\sigma''\left(O^{K-1}_{bi}\right) + \sigma'\left(O^{K-1}_{bi}\right)\left[\partial\mathfrak{D}^{K-1}_{bim}\right] \\
\Delta^K_{bim}\partial\mathfrak{D}^K_{bim} &= \left\{\Delta^K_{bim}\mathfrak{D}^{K-1}_{bim}\sigma''\left(O^{K-1}_{bi}\right)\right\}\left[\partial O^{K-1}_{bi}\right] + \left\{\Delta^K_{bim}\sigma'\left(O^{K-1}_{bi}\right)\right\}\left[\partial\mathfrak{D}^{K-1}_{bim}\right] \\
\Delta^K_{bim}\partial\mathfrak{D}^K_{bim} &= \left(\delta^{K-1}_{bi}\right)\left[\partial O^{K-1}_{bi}\right] + \left(\Delta^{K-1}_{bim}\right)\left[\partial\mathfrak{D}^{K-1}_{bim}\right] \\
\left(\delta^{K-1}_{bi}\right) &\equiv \left\{\Delta^K_{bim}\mathfrak{D}^{K-1}_{bim}\sigma''\left(O^{K-1}_{bi}\right)\right\} \\
\left(\Delta^{K-1}_{bim}\right) &\equiv \left\{\Delta^K_{bim}\sigma'\left(O^{K-1}_{bi}\right)\right\}
\end{aligned}$$

Where the first term represents a normal back propagation while the second term represents a gradient backpropagation.

3 Solving a toy example

Using this, we are now in a position to solve simple partial differential equations. For benchmarking purposes, consider:

$$\frac{dy}{dx} = \frac{1}{x}$$

The solution to this equation is: $y = \log(x) + c$. Solving this using the gradient backpropagation, we get the remarkable graph Figure 1

An implementation of this method (in C++) is available at: [https://github.com/aravindhv10/CPP_Wrappers/tree/master/LatestHeaders]. The implementation so far only works on the CPU (uses CPU blas and doesnot run on GPU), this result was obtained with just a few minutes of training (ofcourse, the differential equation is also very simple).

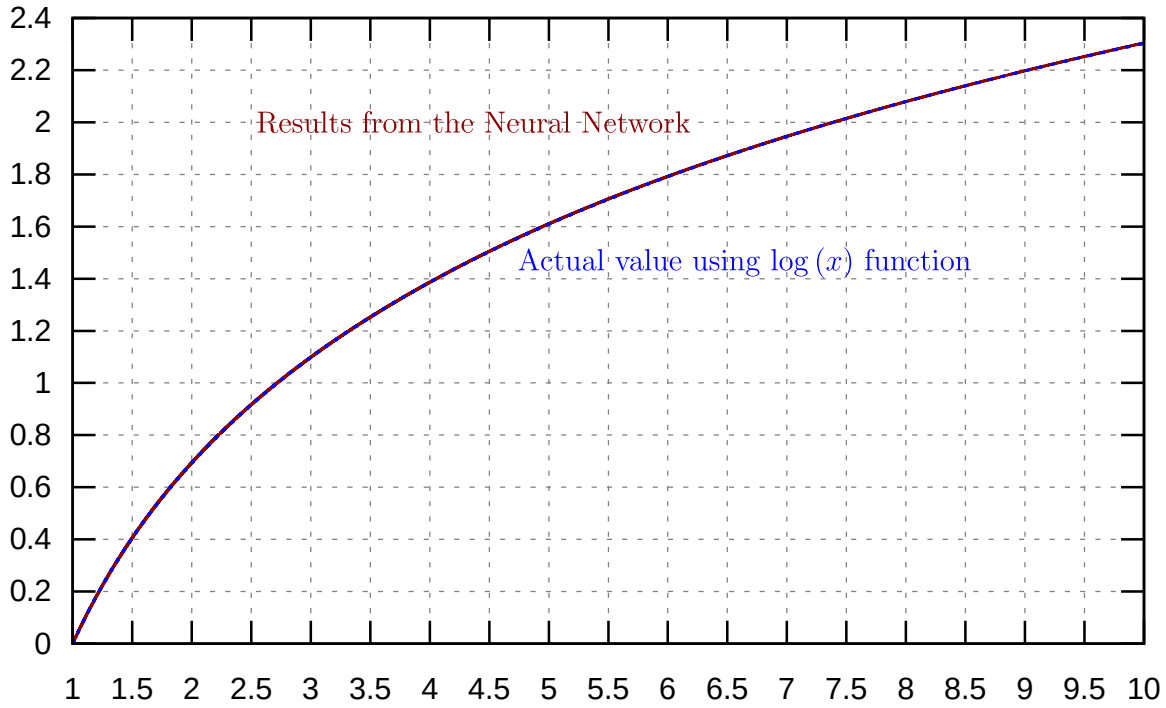

Figure 1: Comparing solution obtained from Neural Network to the exact solution for the simple differential equation mentioned above.