Submission for Deep Learning Exercise

Team: shallow_learning_group
Students: Batuhan Karaca

November 11, 2023

Pen and Paper tasks

Please note that

$$u(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases}$$

is the unit step function, which is the derivative of the $ReLU$, and

$$sign(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{if } x = 0 \\ -1, & \text{if } x < 0 \end{cases}$$

is the sign function, giving the arithmetic sign of a real number.

1)

Forward propagation

$$\begin{aligned} z_0 &= \omega_0 x \\ h_0 &= g_0(z_0) = ReLU(z_0) \\ z_1 &= \omega_1 h_0 \\ h_1 &= g_1(z_1) = ReLU(z_1) \\ z_2 &= \omega_s h_0 + \omega_2 h_1 \\ \hat{y} &= g_2(z_2) = z_2 \\ \mathcal{L}(\hat{y}, y) &= |y - \hat{y}| \end{aligned}$$

Backward propagation

$$
\begin{aligned}
\frac{\delta|f(x)|}{\delta x} &= sign(f(x))\frac{\delta f(x)}{\delta x} \\
\frac{\delta L}{\delta \hat{y}} &= sign(y-\hat{y})\cdot -1 = -sign(y-\hat{y}) \\
\frac{\delta L}{\delta z_2} &= \frac{\delta L}{\delta \hat{y}}\frac{\delta \hat{y}}{\delta z_2} = -sign(y-\hat{y})\cdot 1 = -sign(y-\hat{y}) \\
\frac{\delta L}{\delta \omega_s} &= \frac{\delta L}{\delta z_2}\frac{\delta z_2}{\delta \omega_s} = -sign(y-\hat{y})h_0 \\
\frac{\delta L}{\delta \omega_2} &= \frac{\delta L}{\delta z_2}\frac{\delta z_2}{\delta \omega_2} = -sign(y-\hat{y})h_1 \\
\frac{\delta L}{\delta h_1} &= \frac{\delta L}{\delta z_2}\frac{\delta z_2}{\delta h_1} = -sign(y-\hat{y})\omega_2 \\
\frac{\delta L}{\delta z_1} &= \frac{\delta L}{\delta h_1}\frac{\delta h_1}{\delta z_1} = -sign(y-\hat{y})\omega_2 u(z_1) \\
\frac{\delta L}{\delta \omega_1} &= \frac{\delta L}{\delta z_1}\frac{\delta z_1}{\delta \omega_1} = -sign(y-\hat{y})\omega_2 u(z_1)h_0 \\
\frac{\delta L}{\delta h_0} &= \frac{\delta L}{\delta z_1}\frac{\delta z_1}{\delta h_0} + \frac{\delta L}{\delta z_2}\frac{\delta z_2}{\delta h_0} \\
&= -sign(y-\hat{y})\omega_2 u(z_1)\omega_1 - sign(y-\hat{y})\omega_s \\
&= -sign(y-\hat{y})(\omega_2 u(z_1)\omega_1 + \omega_s) \\
\frac{\delta L}{\delta z_0} &= \frac{\delta L}{\delta h_0}\frac{\delta h_0}{\delta z_0} \\
&= -sign(y-\hat{y})(\omega_2 u(z_1)\omega_1 + \omega_s)u(z_0) \\
\frac{\delta L}{\delta \omega_0} &= \frac{\delta L}{\delta z_0}\frac{\delta z_0}{\delta \omega_0} \\
&= -sign(y-\hat{y})(\omega_2 u(z_1)\omega_1 + \omega_s)u(z_0)x
\end{aligned}
$$

2)

For deeper networks, there arises the problem of vanishing gradients. The information of the layers closer to the input will become less significant as the forward propagation occurs. Furthermore, The gradients of the layers closer to the input will be close to zero (even zero due to finite precision) as the backward propagation occurs, that the layers are not capable of updating anymore. The skip connections may enable transferring the information further the layers, coping with this problem by adding a shallower network in a sense.

3)

Forward propagation

$$
\begin{aligned}
z_0 &= \omega_0 x_1 = 0.5 \cdot 1 = 0.5 \\
h_0 &= g_0(z_0) = ReLU(z_0) = 0.5 \\
z_1 &= \omega_1 h_0 = 0.5 \cdot 0.5 = 0.25 \\
h_1 &= g_1(z_1) = ReLU(z_1) = 0.25 \\
z_2 &= \omega_s h_0 + \omega_2 h_1 \\
&= 0.5 \cdot 0.5 + 0.5 \cdot 0.25 \\
&= 0.375 \\
\hat{y} &= g_2(z_2) = z_2 = 0.375 \\
\mathcal{L}(\hat{y}, y_1) &= |y_1 - \hat{y}| = |-3 - 0.375| = |-3.375| = 3.375
\end{aligned}
$$

Backward propagation

$$
\begin{aligned}
-sign(y_1 - \hat{y}) &= -sign(-3.375) = 1 \\
\frac{\delta L}{\delta \hat{y}} &= -sign(y_1 - \hat{y}) = 1 \\
\frac{\delta L}{\delta z_2} &= -sign(y_1 - \hat{y}) = 1 \\
\frac{\delta L}{\delta \omega_s} &= -sign(y_1 - \hat{y}) h_0 = 0.5 \\
\frac{\delta L}{\delta \omega_2} &= -sign(y_1 - \hat{y}) h_1 = 0.25 \\
\frac{\delta L}{\delta h_1} &= -sign(y_1 - \hat{y}) \omega_2 = 0.5 \\
\frac{\delta L}{\delta z_1} &= -sign(y_1 - \hat{y}) \omega_2 u(z_1) = 0.5 \\
\frac{\delta L}{\delta \omega_1} &= -sign(y_1 - \hat{y}) \omega_2 u(z_1) h_0 = 0.25 \\
\frac{\delta L}{\delta h_0} &= -sign(y_1 - \hat{y})(\omega_2 u(z_1) \omega_1 + \omega_s) \\
&= 1 \cdot (0.5 \cdot 1 \cdot 0.5 + 0.5) \\
&= 0.75 \\
\frac{\delta L}{\delta z_0} &= -sign(y_1 - \hat{y})(\omega_2 u(z_1) \omega_1 + \omega_s) u(z_0) = 0.75 \\
\frac{\delta L}{\delta \omega_0} &= -sign(y_1 - \hat{y})(\omega_2 u(z_1) \omega_1 + \omega_s) u(z_0) x_1 = 0.75
\end{aligned}
$$

Update parameters (Please note that learning rate is denoted as α)

$$\begin{aligned}
\omega_{0_{new}} &= \omega_0 - \alpha \frac{\delta \mathcal{L}}{\delta \omega_0} \\
&= 0.5 - 1 \cdot 0.75 \\
&= -0.25 \\
\omega_{1_{new}} &= \omega_1 - \alpha \frac{\delta \mathcal{L}}{\delta \omega_1} \\
&= 0.5 - 1 \cdot 0.25 \\
&= 0.25 \\
\omega_{2_{new}} &= \omega_2 - \alpha \frac{\delta \mathcal{L}}{\delta \omega_2} \\
&= 0.5 - 1 \cdot 0.25 \\
&= 0.25 \\
\omega_{s_{new}} &= \omega_s - \alpha \frac{\delta \mathcal{L}}{\delta \omega_s} \\
&= 0.5 - 1 \cdot 0.5 \\
&= 0
\end{aligned}$$

Questions on experiments

1)

After 1-step update the loss is (almost) the same because parameters have very small (almost zero) gradients that they do little update.

2)

The losses show mostly a decreasing trend. However, due to large learning rates (with especially large gradients), sometimes there could be huge leaps that could make the coordinate of the weights farther away from the minima. Therefore, some increase may occur.

3)

Depending on the initialization of the weights, the network may end up in sub-optimal minima or no minima at-all (this is due to both small learning rate and finite number of optimizing steps, or epochs).

4)

When I see loss increasing (i.e. large descents due to gradient) I switch to smaller learning rates. Otherwise, if there is a smaller convergence rate, I try larger rates.