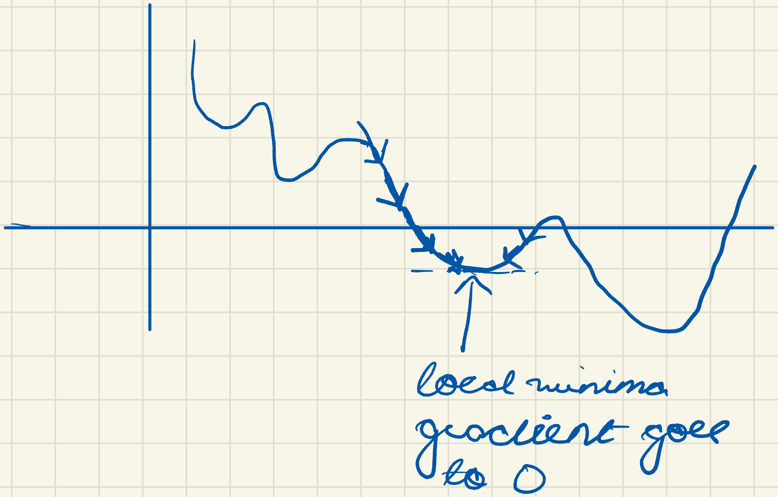
Problem 1

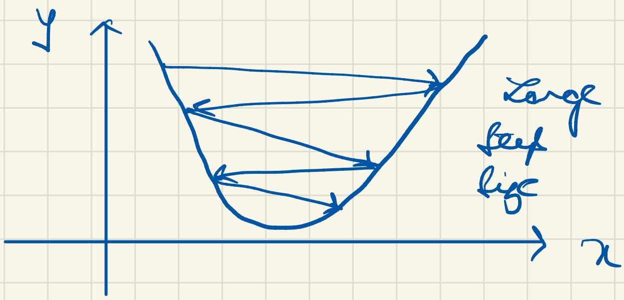
Q1 Let gradient of the cost C w/ respect to x be $C'(x)$

If $C'(x) < 0$, the gradient descent takes a step $x + \epsilon$ & if $C'(x) > 0$ the algorithm takes a step $x - \epsilon$ where $\epsilon > 0$

This would mean that the gradient descent algorithm will converge to a local minima



If ϵ is large we overshoot while for small ϵ it takes a long time for gradient descent to converge



Q2

An advantage would be that learning occurs faster

While with full a batch size of one the gradient descent will be very noisy with the cost going up in certain iterations

Problem 2

4 fundamental equations of Backpropagation:

1. $\delta^L = \nabla_a C \odot \sigma'(z^L)$

2. $\delta^l = (w^{l+1\,T})(\delta^{l+1}) \odot \sigma'(z^l)$

3. $\frac{\partial C}{\partial b_j^l} = \delta_j^l$

4. $\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l$

1.

Let $z = \sum w_j x_j + u$

Then we need $\sigma'(z)$ which would be the derivative of f w.r.t z

other than this there are no modifications required to the network

2. $\delta_j^L = \frac{\partial C}{\partial z_j^L}$

$$C = \frac{1}{n}\Sigma\left(y \log(a) + (1-y)\log(1-a)\right)$$

$$\frac{\partial C}{\partial a} = -\frac{1}{n}\Sigma\left(\frac{y}{a} - \frac{(1-y)}{(1-a)}\right)$$

$$= -\frac{1}{n}\Sigma\left(\frac{y - ay - (1-y)a}{a(1-a)}\right)$$

$$\frac{\partial a}{\partial z} = \frac{\partial}{\partial z}\left(\frac{e^z}{1+e^z}\right)$$

$$= \frac{e^z}{1+e^z} - \frac{e^z}{(1+e^z)^2} e^z$$

$$= \frac{e^z}{1+e^z}\left(1 - \frac{e^z}{1+e^z}\right)$$

$$= a(1-a).$$

$$\frac{\partial C}{\partial z} = \frac{\partial C}{\partial a} \cdot \frac{\partial a}{\partial z}$$

$$\frac{\partial C}{\partial z} = -\left(\frac{y - ay - a + ay}{a(1-a)}\right) \cdot (a(1-a))$$

$$= a - y$$

Thus proved $\delta_j^L = a_j^L - y_j$

3. If $\sigma(z) = z$

$$\Rightarrow \sigma'(z) = 1$$

Eq^n 1

$$\delta^L = \nabla_a C \odot \sigma'(z_j^L)$$

$$\Rightarrow \delta^L = \nabla_a C$$

Eq^n 2

$$\delta^l = w^{L+1\,T} \delta^{L+1} \odot \sigma'(z^L)$$

$$\Rightarrow \delta^l = w^{L+1\,T} \delta^{L+1}$$

Eq^n 3

$$\frac{\partial C}{\partial b_j^l} = \delta_j^l$$

Eq^n 4

$$\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1}\,\delta_j^l$$

$$\Rightarrow \frac{\partial C}{\partial w_{jk}^l} = z_k^{l-1}\,\delta_j^l$$

Problem 3

Q1 The correct cross-entropy function is $-(y \ln a + (1-y)\ln(1-a))$

The incorrect expression is $-(a \ln y + (1-a)\ln(1-y))$ isn't defined for $y=0$ or $y=1$, because $\ln(x)$ isn't defined for $x=0$

By definition of σ, $0 < \sigma(z) < 1$ & thus this problem doesn't arise in the correct definition

Q2 $C(a) = -(y\ln a + (1-y)\ln(1-a))$

$$C'(a) = \frac{-y}{a} + \frac{1-y}{1-a}$$

If $C'(a) = 0$

$$-\frac{y}{a} + \frac{1-y}{1-a} = 0$$

$$\Rightarrow \frac{1-y}{1-a} = \frac{y}{a}$$

$$y - ay = a - ay$$

$$\Rightarrow y = a$$

There's an extremum point at $y = a$

$$C''(a) = \frac{y}{a^2} + \frac{1-y}{(1-a)^2}$$

Now we take $y \in (0,1)$

By definition $a \in (0,1)$

Thus $C''(a) \geq 0$ meaning our func^n is convex

Thus func^n is minimized at $a = y$

Q3

$y_1 = 0.01$, $y_2 = 0.99$

$$C = -[y_1 \ln a_1 + (1-y_1)\ln(1-a_1)]$$
$$-[y_2 \ln a_2 + (1-y_2)\ln(1-a_2)]$$

$$z_{h1} = w_1 \cdot i_1 + w_3 \cdot i_2 + b_1$$
$$= .15(.05) + 0.25(0.1) + 0.35$$
$$= 0.3825$$

$$a_{h1} = \sigma(.3825) = \frac{1}{1+e^{-.3825}} = .5945$$

$zh_2 = w_2 \cdot i_1 + w_4 \cdot i_2 + b_1$

$= .2 \times 0.05 + .7 \times .1 + .35 = .39$

$ah_2 = \sigma(.39) = \dfrac{1}{1+e^{-.39}} = .5963$

$zo_1 = w_5 ah_1 + w_7 ah_2 + b_2$

$= 0.4 \times 0.5945 + 0.5 \times 0.5963$
$+ .6$

$= 1.136$

$ao_1 = \sigma(1.136) = \dfrac{1}{1+e^{-1.136}} = 0.7569$

$zo_2 = w_6 \cdot ah_1 + w_8 \cdot ah_2 + b_2$

$= 0.45 \times 0.5945 + 0.55(.5963)$
$+ .6 = 1.195$

$ao_2 = \sigma(1.195) = \dfrac{1}{1+e^{-1.195}} = .7676$

$C = C_{o1} + C_{o2}$

$$co1 = -(y1 \ln a01 + (1-y1)\ln(1-a01))$$

$$= -(0.01 \ln(0.7569)$$

$$+(1-0.01)\ln(1-.7569))$$

$$= 1.403$$

$$co2 = -y2 \ln a2 + (1-y2)\ln(1-a2)$$

$$= -(0.99 \ln(.7676) + (1-.99)\ln(1-.7676)]$$

$$= .2764$$

$$C = 1.403 + .2764 = 1.6794$$

Bonus Question

Q1. The largest element in the input vector remains the largest element after the softmax funcⁿ is applied to the vector, hence the max part. The soft signifies that the funcⁿ takes into account all the non maximal elements, as opposed to a standard maximum function

Optional Questions

$\delta^L = \sigma'(z^L) \odot \nabla_a C$ can be written as $\delta^L = \Sigma'(z^L) \nabla_a C$

If $x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$, $y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$

then $x \odot y = y \odot x$

$= \sum_{k=1}^{n} x_k y_k$

$$= \begin{bmatrix} x_1 & 0 & 0 & \cdots \\ 0 & x_2 & \cdots & \\ & & \ddots & \\ & & & x_n \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

This reasoning proves the eq^{n}s for Q1 & Q2

Q3 To prove this eq^n

we repeatedly replace δ^{L+1} in δ^L

$$= \Sigma'(z^l)(w^{L+1})^T \delta^{L+1}$$

with

$$\Sigma'(z^{l+1})(w^{L+2})^T \delta^{L+2}$$

until reaching $l+1 = L$

then,

using $\delta^L = \sigma'(z^L) \odot \nabla_a C$

to get

$$\delta^L = \Sigma'(z^l)(w^{L+1})^T \cdots \Sigma'(z^{L-1})(w^L)^T \Sigma'(z^L)\nabla_a C$$