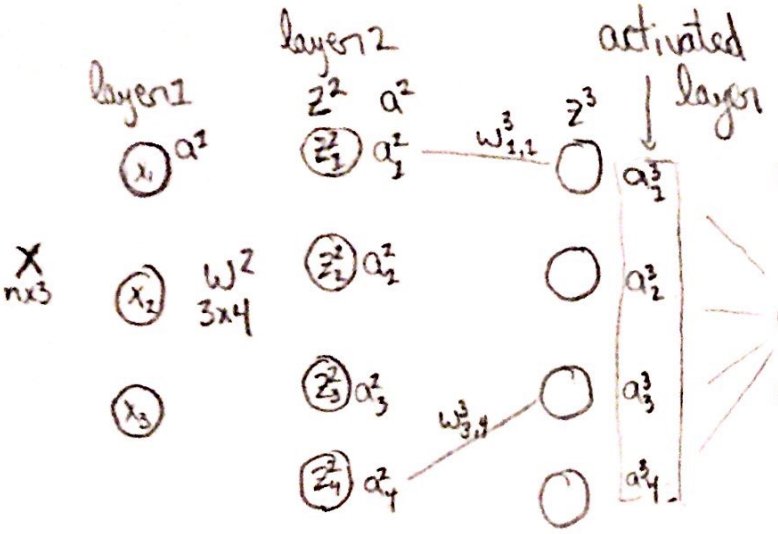

X nx3

x_n - input n

w^n - weight matrix before layer n

z^n - pre-activated layer n

a^n - activated layer n

w^n_{jk} - weight from k^{th} node in layer n-1 to j^{th} node in layer n

z^n_j - pre-activated node j in layer n

a^n_j - activated node j in layer n

$L(a^L, y)$ - loss function between last layer of activations and target, y.

$\odot$ - hadamard product, elementwise multiplication between vectors

Want: $\frac{\partial L}{\partial w^n}$, aka how much changing a weight in layer n changes the loss, L.

Things we know: $z^n = a^{n-1} \cdot w^n \Rightarrow \frac{\partial z^n}{\partial a^{n-1}} = w^{nT}$ & $\frac{\partial z^n}{\partial w^n} = a^{n-1}$

Procedure: ① Input X, set vector as first activations a^1

② Feed forward: for each of the layers, n, compute:

$$z^n = a^{n-1} \cdot w^n \quad \& \quad a^n = \sigma(z^n)$$

③ Calculate error of L wrt last pre-activated nodes, z^L:

$$\frac{\partial L}{\partial z^L} = \frac{\partial L}{\partial a^L} \frac{\partial a^L}{\partial z^L} \rightarrow \sigma'(z^L)$$

④ Backpropagate error: for each layer, n (L-2, L-2), going backward compute:

$$\frac{\partial L}{\partial z^n} = \frac{\partial L}{\partial z^{n+1}} \frac{\partial z^{n+1}}{\partial a^n} \rightarrow (w^n)^T \quad \frac{\partial a^n}{\partial z^n} \rightarrow \sigma'(z^n)$$

⑤ Calculate gradient of loss wrt. weights:

$$\frac{\partial L}{\partial w^n} = \frac{\partial L}{\partial z^n} \frac{\partial z^n}{\partial w^n} \rightarrow a^{n-1}$$

← calculated in step 4

⑥ Update weights:

$$w^n \leftarrow w^n - \lambda \frac{\partial L}{\partial w^n}$$

Isomorphism up to an abuse of derivative notation