HW-3 Neural Nets Theory

VASISHT DUDDU
2015137

1. We cannot build a neural network of arbitrary length to learn XOR using linear activation function. A network with hidden layers with linear activation function is still a generalized linear model & XOR being not linearly seperable we can't classify it.

$$y = f(\omega_2(\omega_1 x + b_1) + b_2)$$

This is same as the SVM with linear function or simple logistic regression.

$$f(x) = B(Ax + a) + b$$
$$= BAx + BA + b$$
$$= Cx + c$$

(also a linear function)

2. One possible reason is sigmoid saturation and killing gradients. When a sigmoid neuron's activation saturates at either 0 or 1, the gradient at these regions is almost 0. Hence, there is no signal flow through the neuron to its weights and recursively to its data.

Extra care must be taken while initializing the weights of neurons to prevent saturation. Usually a small value should be taken as initial values of weights.

ReLU performs better than sigmoid and it is preferred to use ReLU over sigmoid. However, it may have a dead neuron problem which could effect the training of neural network. ReLU can be fragile during training an can die. A large gradient could update weights in such a way that gradient flowing could be zero from then on. This can be irreversible and kill multiple neurons.

This occurs if the learning rate is high. Hence, to prevent this we set use low learning rate.

Few pre-processing technique like standard scalar, normalization & min-max scaling. The problems can be avoided by ensuring small values of weights and learning rates.

3) <u>Quadratic Cost Function</u>

$$C = \frac{(y - \sigma(z))^2}{2} \qquad z = wx + b$$

$$\frac{\partial C}{\partial w} = (\sigma(z) - y)\,\sigma'(z)x \qquad \frac{\partial C}{\partial b} = (\sigma(z) - y)\,\sigma'(z)$$

Putting $x = 0$ & $y = 0$

$$\frac{\partial C}{\partial w} = \sigma(z)\,\sigma'(z) \qquad \frac{\partial C}{\partial b} = \sigma(z)\,\sigma'(z)$$

Learning slowdown occurs when the partial derivates are small. Small values of $\frac{\partial C}{\partial w}$ & $\frac{\partial C}{\partial b}$, then learning becomes very slow.

<u>Cross - Entropy Cost Function</u>

$$C = -\frac{1}{n} \sum_x [y \ln a + (1-y) \ln(1-a)]$$

$$\frac{\partial C}{\partial w_j} = -\frac{1}{n} \sum_x \left(\frac{y}{\sigma(z)} - \frac{1-y}{1-\sigma(z)}\right) \sigma'(z)\, x_j$$

$$= \frac{1}{n} \sum_x (\sigma(z) - y)\, x_j$$

We observe that learning rate is dependent on $(\sigma(z) - y)$, i.e error of o/t. It avoids the learning slowdown caused by $\sigma'(z)$ in the quadratic cost function.