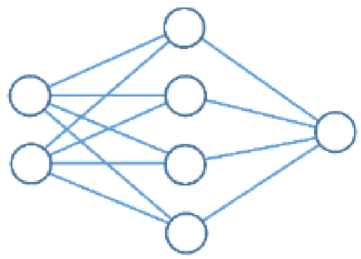
Deep Learning

Sunday, October 28, 2018 2:04 PM

2017:

1. **DEEP LEARNING [BACKPROPAGATION]**

Consider the three-layer neural network shown below, where $x = (x_1, x_2) \in \mathbb{R}^2$ are input neurons, $y = (y_1, y_2, y_3, y_4) \in \mathbb{R}^4$ are ReLU neurons, and $z \in \mathbb{R}$ is a linear neuron. Let $\tilde{z}$ be the desired output, as given by the training data.



The forward propagation of the neural network may be written as:

$$u = W^{(1)}x + b^{(1)}, \qquad y = \text{ReLU}(u), \qquad z = W^{(2)}y + b^{(2)}$$

where $\text{ReLU}(u) = \max\{0, u\}$. The point loss $\mathcal{L}_1$ is the squared loss $\frac{1}{2}(\tilde{z} - z)^2$. Using backpropagation, the gradients of the point loss are given by

$$\nabla_{W^{(1)}}\mathcal{L}_1 = \delta^{(2)}x^\top, \qquad \nabla_{W^{(2)}}\mathcal{L}_1 = \delta^{(3)}y^\top$$

$$\nabla_{b^{(1)}}\mathcal{L}_1 = \delta^{(2)}, \qquad \nabla_{b^{(2)}}\mathcal{L}_1 = \delta^{(3)}$$

where $\delta^{(3)}, \delta^{(2)}$ are the backpropagating error signals. Let $*$ represent the element-wise multiplication of two vectors, and let H be the function where $H(u) = 1$ if $u > 0$, and $H(u) = 0$ if $u \le 0$.

a. Given that $\delta^{(3)} = z - \tilde{z}$, what is the formula that computes the error signal $\delta^{(2)}$?

   Ans: $\delta^{(2)} = (W^{(2)\top}\delta^{(3)}) * H(u)$

b. One can prove that ReLU networks give rise to continuous piecewise-linear functions. In other words, the space of inputs can be cut up into regions where the output of the network is a linear function within each region, and where neighboring regions have different linear functions. For our three-layer ReLU network, what is the maximum number of regions that can be obtained?

   Ans: 11

2016:

2. **DEEP LEARNING**

a. Write down the name of the training algorithm that is based on the chain rule:

   Ans: Back propagation

b. In deep learning, because the learning objective function is highly non-convex, one major issue with gradient descent is getting stuck in a bad local minimum. This issue can be alleviated by carefully initializing the parameters. Describe one strategy for doing this initialization.

   Ans: We initialize the layers of the neural network using a layer-wise greedy algorithm. For each layer L, we first feedforward the data to L, assuming that the parameters for the earlier layers have been initialized. We then train a 3-layer sparse autoencoder, using the data in layer L as both inputs and outputs to the autoencoder. The trained weights of the first layer of the autoencoder is then used to initialize the parameters of layer L. We continue to the next layer until all the parameters have been initialized in this fashion.