Lecture -19

95 (4) Use CV to evaluate λ

(5) Pick λ^*

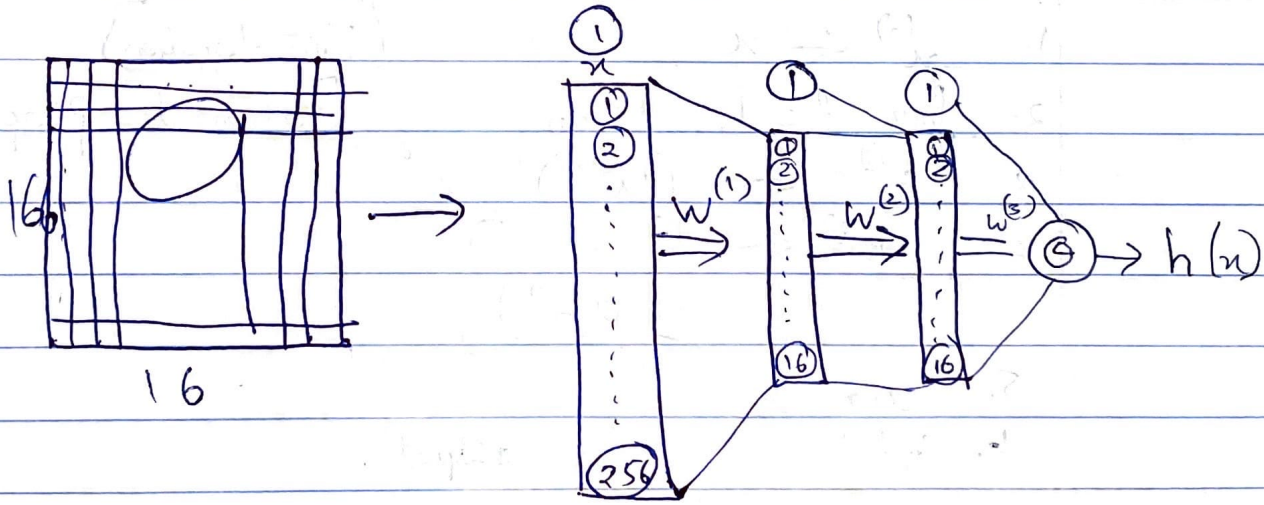
(6) $E_{out}(g^{\lambda^*}) \approx E_{test}(g^{\lambda^*})$.

Use 8998 test examples.

show 99 % error bar

$$E_{test} - \varepsilon \leq E_{out} \leq E_{test} + \varepsilon$$

$$E_{out} \leq E_{test} + \sqrt{\frac{1}{2N} \log 2M'}$$



how many nodes?

$$257 + 17 + 17 + 1 \Rightarrow 291$$

How many weights? $\Rightarrow 257 \times 16 + 17 \times 16 + 17 \times 16 \Rightarrow$

is $w^{(1)} = 257 \times 16 +$

is $w^{(2)} = 17 \times 16 + 17 \times 16 \rightarrow w^{(3)}$

The linear signal $s^{(l)}$

$s^{(l)}$ is a linear combination (using $W^{(l)}$) of the outputs of the previous layer $x^{(l-1)}$

$$s^{(l)} = \left(W^{(l)}\right)^T x^{(l-1)}$$

Forward Propagation : Computing $h(x)$

$$x = x^{(0)} \xrightarrow{W^{(1)}} s^{(1)} \xrightarrow{\theta} x^{(1)} \xrightarrow{W^{(2)}} s^{(2)} \xrightarrow{\theta} x^{(2)} \rightarrow \cdots \rightarrow s^{(L)} \rightarrow x^{(L)} = h(x)$$

1. $x^{(0)} \leftarrow x$ [Initialization]
2. for $l = 1 \ldots L$ do [Forward propagation]
3. $s^{(l)} \leftarrow \left(W^{(l)}\right)^T x^{(l-1)}$
4. $x^{(l)} \leftarrow \begin{bmatrix} 1 \\ \theta(s^{(l)}) \end{bmatrix}$
5. end for.
6. $h(x)$ output.

Dimensions of:

$$W^{(l)} = \left(d^{(l-1)} + 1\right) \times d^{(l)}$$

$$s^{(l)} = d^{(l)} \times 1$$

$$x^{(l)} = \left(d^{(l)} + 1\right) \times 1$$

For classification, regression:

$$E_{in}(h) = E_{in}(w) = \frac{1}{N} \sum_{n=1}^{N} (h(x_n) - y_n)^2$$

For logistic regression,

$$E_{in}(h) = \frac{1}{N} \sum_{n=1}^{N} \ln\left(1 + e^{y_n h(x_n)}\right)$$

Fitting the Data: Minimizing E_{in}

$$E_{in}(h) = E_{in}(w) = \frac{1}{N} \sum_{n=1}^{N} (h(x_n) - y_n)^2$$

where $W = \{w^{(1)}, w^{(2)}, \ldots w^{(L)}\}$

take gradient descent set it to 0 and solve for variab

$$w(t+1) = w(t) - \eta \nabla E_{in}(w(t))$$

$$w^{(l)} \leftarrow w^{(l)} - \eta \frac{\partial E_{in}}{\partial w^{(l)}}$$

Need to compute the gradient of E_{in} at $w(t)$

Gradient of E_{in}

Given weights W,

$$E_{in}(W) = \frac{1}{N}\sum_{n=1}^{N} e_n = \frac{1}{N}\sum_{n=1}^{N} e\{h(x_n; W), y_n\}$$

the gradient of $E_{in}(W)$ is:

$$\frac{\partial E_{in}(W)}{\partial W^{(l)}} = \frac{1}{N}\sum_{n=1}^{N} \frac{\partial e(h(x_n), y_n)}{\partial W^{(l)}}$$

Try not to use for loops, instead use matrix.

Computing the Gradient:
Numerical, Finite Difference approach.

$$\frac{\partial e(x)}{\partial W_{ij}^{(l)}} \approx \frac{e(x|W_{ij}^{(l)} + \Delta) - e(x|W_{ij}^{(l)} - \Delta)}{2\Delta}$$

Back

$$O((N|w| + N|V|)|w|)$$

$|V|$ → tan h

just to compute (e)

For every weight we have to count. So multiply $|w|$

Backpropagation

$$O((N|w| + N|V|)|w|)$$

A dynamic Programming Algorithm.

chain Rule.

Node i only effects the output h(x) through its input $s_i^{(\ell)}$

$$\frac{\partial e}{\partial w_i^{(\ell)}} = \frac{\partial s_i^{(\ell)}}{\partial w_i^{(\ell)}} \cdot \frac{\partial e}{\partial s_i^{(\ell)}}$$

how error changes w.r.t. $w_i^{(\ell)}$

how input signal changes w.r.t. $w_i^{(\ell)}$

how error changes w.r.t $s_i^{(\ell)}$

$$s_i^{(\ell)} = (w_i^{(\ell)})^T x^{(\ell-1)} \Rightarrow \frac{\partial s_i^{(\ell)}}{\partial w_i^{(\ell)}} = x^{(\ell-1)}$$

$$\frac{\partial e}{\partial w_i^{(\ell)}} = \frac{\partial s_i^{(\ell)}}{\partial w_i^{(\ell)}} \delta_i$$