$$\frac{\partial \mathcal{L}}{\partial W_{ij}^{[2]}} = \frac{\partial \mathcal{L}}{\partial a^{[3]}} \frac{\partial a^{[3]}}{\partial W_{ij}^{[2]}}$$

$$= \underbrace{\frac{\partial \mathcal{L}}{\partial a^{[3]}} \frac{\partial a^{[3]}}{\partial z^{[3]}}}_{} \frac{\partial z^{[3]}}{\partial a^{[2]}} \frac{\partial a^{[2]}}{\partial z^{[2]}} \frac{\partial z^{[2]}}{\partial W_{ij}^{[2]}}$$

$$= (a^{[3]} - y) \cdot W^{[3]} \operatorname{diag}[g'(z^{[2]})]$$

$$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ a_j \\ \vdots \\ \vdots \end{bmatrix} \leftarrow i\text{th position}$$

element wise product

$$= \left[\underbrace{(a^{[3]} - y)\, w^{[3]} \odot g'(z^{[2]})}_{X}\right] \begin{array}{c} \begin{bmatrix} 0 \\ 0 \\ a_j^{[1]} \\ 0 \\ 0 \end{bmatrix} \end{array} \begin{array}{l} \\ \\ \text{— } i\text{th} \\ \\ \\ \end{array}$$

$$= X_i \, a_j^{[1]}$$

[ith element of vector X

$$\frac{\partial \mathcal{L}}{\partial W^{[2]}} = X a^{[1]T}$$

Network View

$$f : \mathbb{R}^n \to \mathbb{R}^m$$

$$\frac{\partial f}{\partial x} \in \mathbb{R}^{m \times n}$$

Row vector $\delta^{[2]}$

$\text{diag}\, g' \, w^{[L-1]} \dots$

$$\underbrace{\frac{\partial \mathcal{L}}{\partial W_{ij}^{[2]}}}_{1\times 1} = \underbrace{\underbrace{\frac{\partial \mathcal{L}}{\partial a^{[L]}}}_{1\times 1} \underbrace{\frac{\partial a^{[L]}}{\partial z^{[L]}}}_{1\times 1} \underbrace{\frac{\partial z^{[L]}}{\partial a^{[L-1]}}}_{1\times m}}_{\text{Logistic Regression}} \underbrace{\cdots\cdots}_{m\times m} \underbrace{\frac{\partial a^{[2]}}{\partial z^{[2]}}}_{m\times m} \underbrace{\frac{\partial z^{[2]}}{\partial W_{ij}^{[2]}}}_{m\times 1}$$

column vector

Logistic Regression

$[y^{(i)} - h_\theta(x^{(i)})] x^{(i)} \leftarrow \frac{\partial \mathcal{L}}{\partial x}$

$(y^{(i)} - a^{[L]}) w^{[L]} \leftarrow \frac{\partial \mathcal{L}}{\partial a}$

$$\frac{\partial \mathcal{L}}{\partial w_{ij}^{[1]}} = \frac{\partial \mathcal{L}}{\partial z^{[2]}} \frac{\partial z^{[2]}}{\partial a^{[1]}} \frac{\partial a^{[1]}}{\partial z^{[1]}} \frac{\partial z^{[1]}}{\partial w_{ij}^{[1]}}$$

$1 \times 1 \quad\quad 1 \times m \quad\quad m \times m \quad\quad m \times m \quad\quad m \times 1$

$$\frac{\partial a^{[L-1]}}{\partial z^{[L-1]}} = \begin{array}{c} \begin{bmatrix} g' & & & 0 \\ & g' & & \\ & & g' & \\ 0 & & & g' \end{bmatrix} \; a \\ z \end{array}$$

Diagonal

$a_i = g(z_i)$

$$\frac{\partial z^{[2]}}{\partial w_{ij}^{[2]}} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ a_j \\ \\ \\ \end{bmatrix} \leftarrow i^{th} \text{ position}$$

$z_i = \sum_j w_{ij} a_j + b_i$

$$\frac{\partial \mathcal{L}}{\partial w_{ij}^{[2]}} = \delta^{[2]} \cdot \begin{bmatrix} 0 \\ \vdots \\ a_j \\ \vdots \end{bmatrix}$$

$$= \frac{\partial \mathcal{L}}{\partial z^{[2]}} \begin{bmatrix} 0 \\ \vdots \\ a_j \\ \vdots \end{bmatrix} = \left[\frac{\partial \mathcal{L}}{\partial z^{[2]}}\right]_i a_j$$

$$\frac{\partial \mathcal{L}}{\partial W^{[2]}} = \left(\frac{\partial \mathcal{L}}{\partial z^{[2]}}^T\right)\left(a^{[1]}\right)^T \rightarrow \text{think as outerproduct}$$

$$W^{[l]} := W^{[l]} - \alpha \frac{\partial \mathcal{L}}{\partial W^{[l]}}$$

$$J(w,b) = \sum_{i=1}^{B} \mathcal{L}(y^{(i)}, \hat{y}^{(i)})$$

B is size of batch

$$\underbrace{z^{[1]}}_{m} = \underbrace{w^{[1]}}_{m \times d} \underbrace{x^{(i)}}_{d} + \underbrace{b^{[1]}}_{m}$$

$$z^{[1]} = \underbrace{w^{[1]}}_{m \times d} \underbrace{\left[x^{(1)}\ x^{(2)} \cdots x^{(B)}\right]}_{d \times B} + \underbrace{b^{[1]}}_{m \times 1}$$

$m \times B$ $\qquad$ $m \times 1$

Broadcasting

Stochastic Gradient Descent

(i)

α ... $\alpha \to 1$

$$\left.\begin{aligned}\sum_t \alpha^{(t)} &= \infty \\ \sum_t \alpha^{(t)^2} &< \infty\end{aligned}\right\}$$ if cond^n satisfied local optima SGD

NN. : learnable feature map + $\underbrace{\text{Linear model}}_{\text{GLM}}$

You learning representations [like $x \to \phi(x)$] & then you apply GLM

Associated with each n.n. is a kernel,
$K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$, ϕ takes you have input to represent. learnt in $l-1^{th}$ layer

So you can learn kernel & combine it with G.P.

Also, 2 layer NN as no. of neuron in hidden layer $\to \infty$ N.N. becomes G.P.

Universal Approximation Theorem

$y = f(x)$

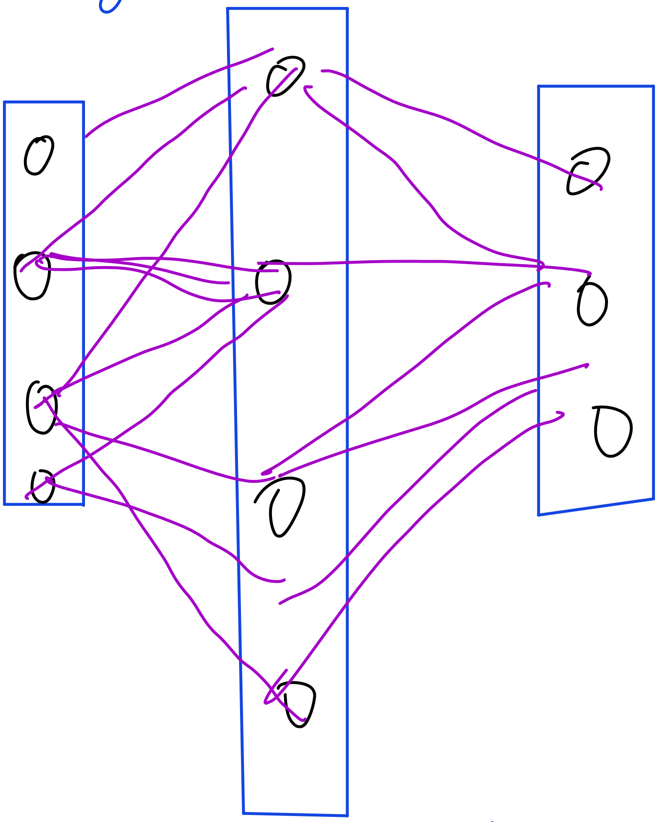
$x \in \mathbb{R}^d$

$y = \in \mathbb{R}^k$

$\varepsilon = 10^{-6}$

bounded region of x

Exists a NN of 1 hidden layer.



that mimics function to an arbitrary precision assuming function is continuous