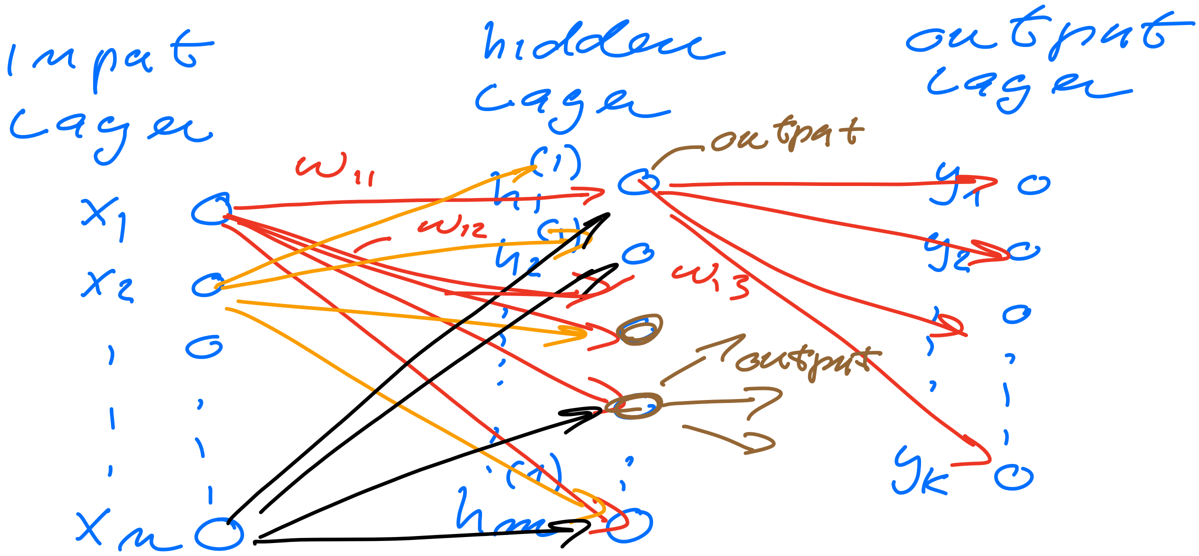
Lecture CompSci Jan 17

Basic elements of a Neural Network (NN)

- input layer (Design/feature matrix X)
- hidden layer(s) with a given number of nodes (neurons/units)
- output layer: compare with our targets.

one hidden layer



to be compared with targets t_i

Regression problem, cost function

$$MSE = \frac{1}{2} \sum_{i=0}^{n-1} (y_i - t_i)^2$$

Fully connected network (no connections between nodes in a specific layer).

Feed Forward NN

l = layer l

$h_i^{(l)} = h_i \leftarrow$ node $-i-$

$\rightarrow$ output $= a_i^l$

$$z_i^l = \sum_{\substack{j = \text{all nodes} \\ \text{in layer } l-1}} w_{ji}^l a_j^{l-1} + b_i^l$$

Bias

output from node j in layer $l-1$

Define activation function f

$$a_i^l = f(z_i^l)$$

$$= f\left(\sum_j w_{ji}^l a_j^{l-1} + b_i^l\right)$$

parameters to define

$$\Theta = \left\{ W, b \right\}$$

weights connecting all layers

biases

- Architecture of FFNN :
  - input layer (fixed)
  - output layer (fixed)
  - # of hidden layers
    - # nodes
    - activation function

$$f(z_i^l) = \frac{1}{\;}$$

$1+e^{-t_i}$

- $w_{ij}^{\ell} \to W$
- b_i^{ℓ} biases

- Training algo: Back propagation algorithm

  ↳ gradient optimization

- regularization (L_1, L_2 norm)

Basic math of a FFNN

$$W \in \mathbb{R}^{m \times n}$$

$$W = \begin{bmatrix} | & & | \\ w_1 & \cdots & w_n \\ | & & | \end{bmatrix}$$

input to all nodes

$$z(x) = Wx + b$$

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

$$x \longmapsto f(z) = A(x)$$

activation

$$A(x) = [a_1(x), a_2(x) \ldots a_n(x)]$$

with a total of $-L-$

(L = output layer)

a set of outputs

$$A_\ell \qquad 1 \le \ell \le L$$

$$A_1 : \mathbb{R}^m \to \mathbb{R}^n$$

$$A_\ell : \mathbb{R}^{n_{\ell-1}} \to \mathbb{R}^{n_\ell}$$

for $2 \le \ell \le L$

$$F(x; \in) =$$

$$f_L(A_L(\ldots\ldots f_1(A_1(x))))$$

Back propagation

L = output layer.

Define some quantities

$$z_j^l = \sum_i w_{ij}^l a_i^{l-1} + b_j^l$$

$$a_j^l = f(z_j^l) = \underbrace{\frac{1}{1+e^{-z_j^l}}}_{\text{sigmoid function}}$$

useful quantities

$$\frac{\partial z_j^l}{\partial w_{ij}^l} = a_i^{l-1}$$

$$\frac{\partial z_j^l}{\partial a_i^{l-1}} = w_{ij}^l$$

$$\frac{\partial a_j^l}{\partial z_j^l} = f(z_j^l)\left(1 - f(z_j^l)\right)$$

$$= a_j^l(1 - a_j^l)$$

$$C(\theta^L) = C(w^L, b^L)$$

$$= \frac{1}{2}\sum_{i=1}^{n}(y_i - t_i)^2$$

Regression problem

$$y_i = a_i^L$$

$$= \frac{1}{2}\sum_{i=1}^{n}(a_i^L - t_i)^2$$

$$\frac{\partial C}{\partial w_{jk}^L} = (a_j^L - t_j)\frac{\partial a_j^L}{\partial w_{jk}^L}$$

$$\frac{\partial a_j^L}{\partial w_{jk}^L} = \frac{\partial a_j^L}{\partial z_j^L}\frac{\partial z_j^L}{\partial w_{jk}^L}$$

$$= a_j^L(1 - a_j^L)\, a_k^{L-1}$$

$$\frac{\partial C}{\partial w_{jk}^L} = \underbrace{(a_j^L - t_j)\, a_j^L(1-a_j^L)}_{\delta_j^L}\, a_k^{L-1}$$

$$\delta_j^L = a_j^L(1-a_j^L)(a_j^L - t_j)$$

$$= f'(z_j^L)\frac{\partial C}{\partial a_j^L}$$

$$\boxed{\frac{\partial C}{\partial w_{jk}^L} = \delta_j^L a_k^{L-1}}$$

$$\delta_j^L = \frac{\partial C}{\partial z_j^L} = \frac{\partial C}{\partial a_j^L}\frac{\partial a_j^L}{\partial z_j^L}$$

$$\boxed{\delta_j^L = \frac{\partial C}{\partial b_j^L}\frac{\partial b_j^L}{\partial z_j^L} = \frac{\partial C}{\partial b_j^L}}$$

We need δ_j^l

$$\delta_j^l = \frac{\partial C}{\partial z_j^l}$$

$$\delta_j^l = \sum_k \frac{\partial C}{\partial z_k^{l+1}}\frac{\partial z_k^{l+1}}{\partial z_j^l}$$

$$= \sum \delta_k^{l+1} \frac{\partial z_k^{l+1}}{\partial z_j^l}$$

$$\frac{\partial}{\partial z_j^l} z_j^{l+1} = \sum_i w_{ij}^{l+1} a_i^l + b_j^{l+1}$$

$$\delta_j^l = \sum_k \delta_k^{l+1} w_{kj}^{l+1} \underline{f'(z_j^l)}$$

$$\delta_j^l = \sum_k \delta_k^{l+1} w_{kj}^{l+1} f'(z_j^l)$$

$$\delta_j^L = f'(z_j^L) \frac{\partial C}{\partial a_j^L}$$

$$l = L-1, L-2, \ldots 2$$

for all $l = L-1, L-2, \ldots, 2$

$$w_{jk}^l \leftarrow w_{jk}^l - \eta \delta_j^l a_k^{l-1}$$

$$b_j^l \leftarrow b_j^l - \eta \frac{\partial C}{\partial b_j^l} =$$

$$b_j - \eta \delta_j$$

Back propagation algo (equation)

Gradient descent to find b_j^{ℓ} and w_{jk}^{ℓ}