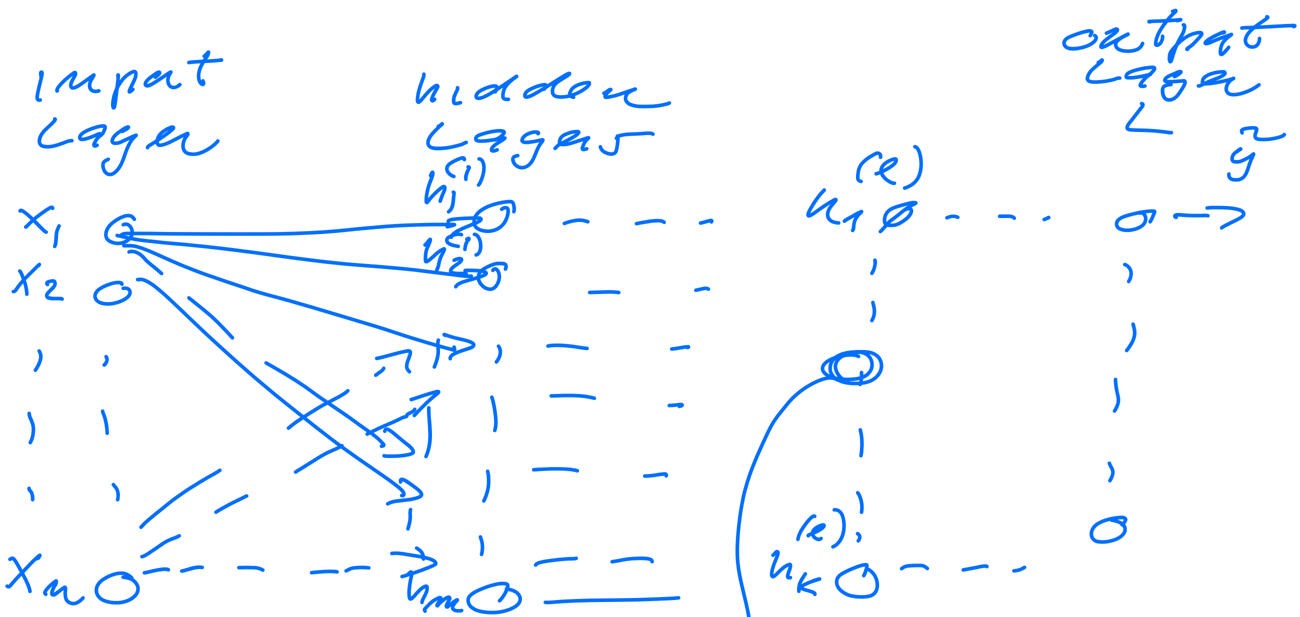
Lecture December 10

Backpropagation algorithm
+ gradient descent



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

$$z_i^l = \sum_j a_j^{l-1} w_{ji}^l + b_i^l$$

at output layer

a_i^L to be compared with our targets y_i (t_i)

Regression : MSE

$$C(w, b) = \frac{1}{2}\sum_i (a_i^L - y_i)^2$$

$$\frac{\partial C}{\partial w} = 0 \qquad \frac{\partial C}{\partial b} = 0$$

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

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

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

Backpropagation

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

$l = L-1, L-2, \ldots \underline{1}$ (first hidden layer)

$$\boxed{\begin{aligned} w_{jk}^{l} &\leftarrow w_{jk}^{l} - \eta \cdot \delta_j^{l} a_k^{l-1} \\ b_j^{l} &\leftarrow b_j^{l} - \eta \, \delta_j^{l} \end{aligned}}$$

input to gradient descent part.

Full algorithm:

$\underline{1}$) Feed Forward

- initialize weights and biases
- Define architecture:
  - # hidden layers
  - # neurons
  - activation functions
  - gradient method
- send in input and feed it through the network to the output

2. produce a_i and compute gradients of cost function

3. Backpropagate to first hidden layer

w_{ik}^{l} and b_i^{l} ← optimize

4. repeat 1-3, till cost function is properly optimized.