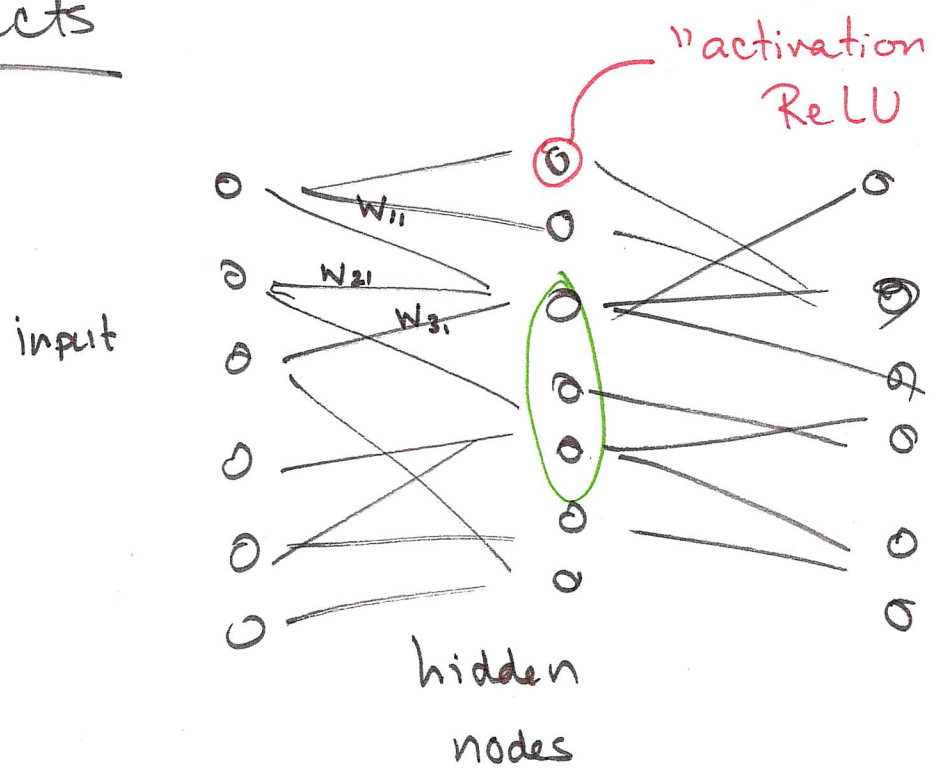
Projects



- sparse weights
  or
  small # of
  hidden nodes ?
- data augmentation
- linear activation ?
  (autoencoder NN
  w/ linear activation
  = PCA)

other ideas

- nonlinear extensions of PCA
  - kernel PCA
  - locally linear embeddings
  - nonnegative matrix factorization
  - autoencoder NN's
- LASSO
  - group LASSO
  - elastic net (LASSO + Ridge)
  - multi-task learning.
- Low-rank tensors

Lec 22 - SGD + Intro to Neural Networks

$$\underset{w}{\text{minimize}} \quad \underbrace{\ell(w)}_{\text{data fit}} + \underbrace{\lambda r(w)}_{\text{regularizer}}$$

$$f(w)$$

e.g. $f(w) = \|y - Xw\|_2^2$

$$= \sum_i (y_i - x_i^T w)^2$$

$$\Rightarrow f_i(w) = (y_i - x_i^T w)^2$$

$$\hat{w} = \underset{w}{\text{argmin}} \; f(w)$$

assume $f(w) = \sum_{i=1}^{n} f_i(w)$

SGD:

@ iteration k, choose $i_k \in \{1, \ldots, n\}$

$$\hat{w}^{(k+1)} = \hat{w}^{(k)} - \frac{\tau}{2} \nabla f_{i_k}(\hat{w}^{(k)})$$

most commonly: $i_k \sim$ uniformly at random for $1, 2, \ldots, n$

A. cyclic $i_k = k \bmod n$

if $n = 3$, i_k's : 1, 2, 3, 1, 2, 3, ...

B. permutations

i_k's : 3, 1, 2, 1, 3, 2 2, 1, 3 ...

epoch 1 epoch 2

C. i_k's unif at random

i_k's : 1, 1, 1, 1, 3, 1, 2, 2, 3, 1

if i_k's are unif @ random,

$$E\left[f_{i_k}\right] = f/n$$

Ex:
$$f(w) = \|y - Xw\|_2^2 + \lambda \|w\|_2^2$$
$$= \sum_{i=1}^{n} \underbrace{\left[(y_i - x_i^T w)^2 + \frac{\lambda}{n} \|w\|_2^2 \right]}_{f_i(w)}$$

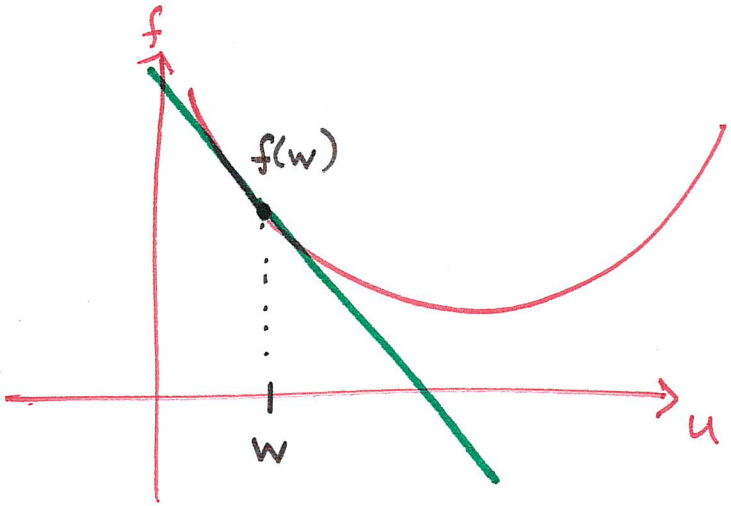
$$\nabla f_i(w) = -2(y_i - x_i^T w) x_i + \frac{2\lambda}{n} w$$

SGD:
$$\hat{w}^{(k+1)} = \hat{w}^{(k)} - \frac{2}{2}\tau \left[-(y_{i_k} - x_{i_k}^T \hat{w}^{(k)}) x_{i_k} + \frac{2\lambda}{n} \hat{w}^{(k)} \right]$$

can replace gradients with subgradients:

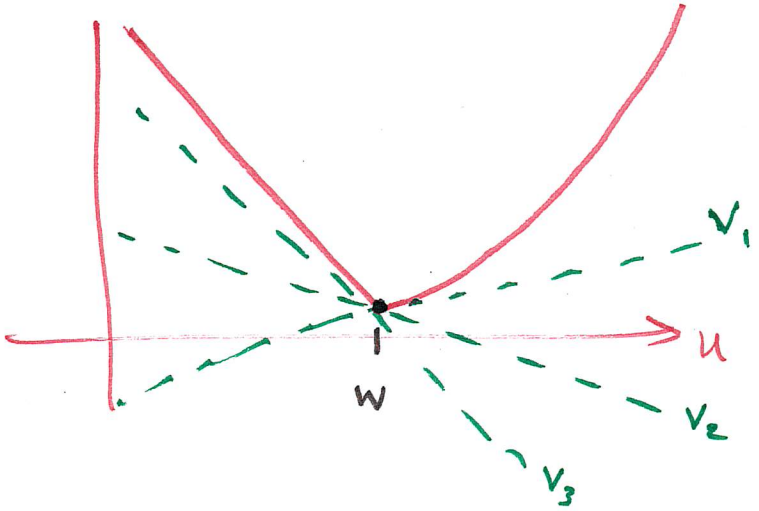
Recall: if f is convex and differentiable:

$$f(u) \geq f(w) + (u-w)^T \nabla f(w)$$



if f is convex but not differentiable, then v is a subgradient of f at w if

$$f(u) \geq f(w) + (u-w)^T v$$



set of subgradients @ w is called "differential set" denoted $\partial f(w)$

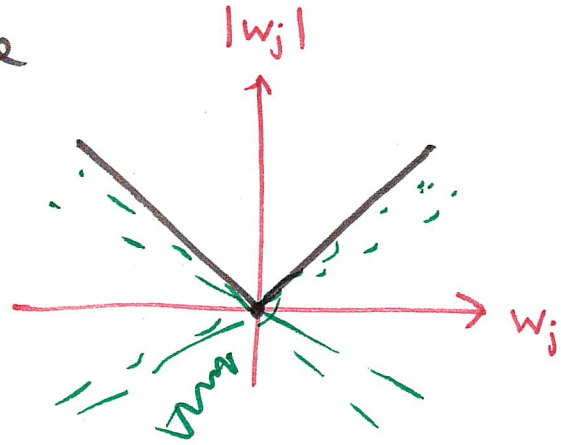
write $v \in \partial f(w)$

e.g. $r(w) = \|w\|_1 = \sum_{j=1}^{p} |w_j|$

for $w_j \neq 0$, $|w_j|$ is differentiable

deriv is $= \text{sign}(w_j)$

for $w_j = 0$, then $v_j \in [-1, +1]$



for $\underline{v} \in \partial r(w)$

then $v_j = \begin{cases} = \text{sign}(w_j) & \text{if } w_j \neq 0 \\ \in [-1, +1] & \text{if } w_j = 0 \end{cases}$

popular choice: $v_j = \begin{cases} \text{sign}(w_j) & \text{if } w_j \neq 0 \\ 0 & \text{if } w_j = 0 \end{cases}$ = "$\text{sign}(w)$"

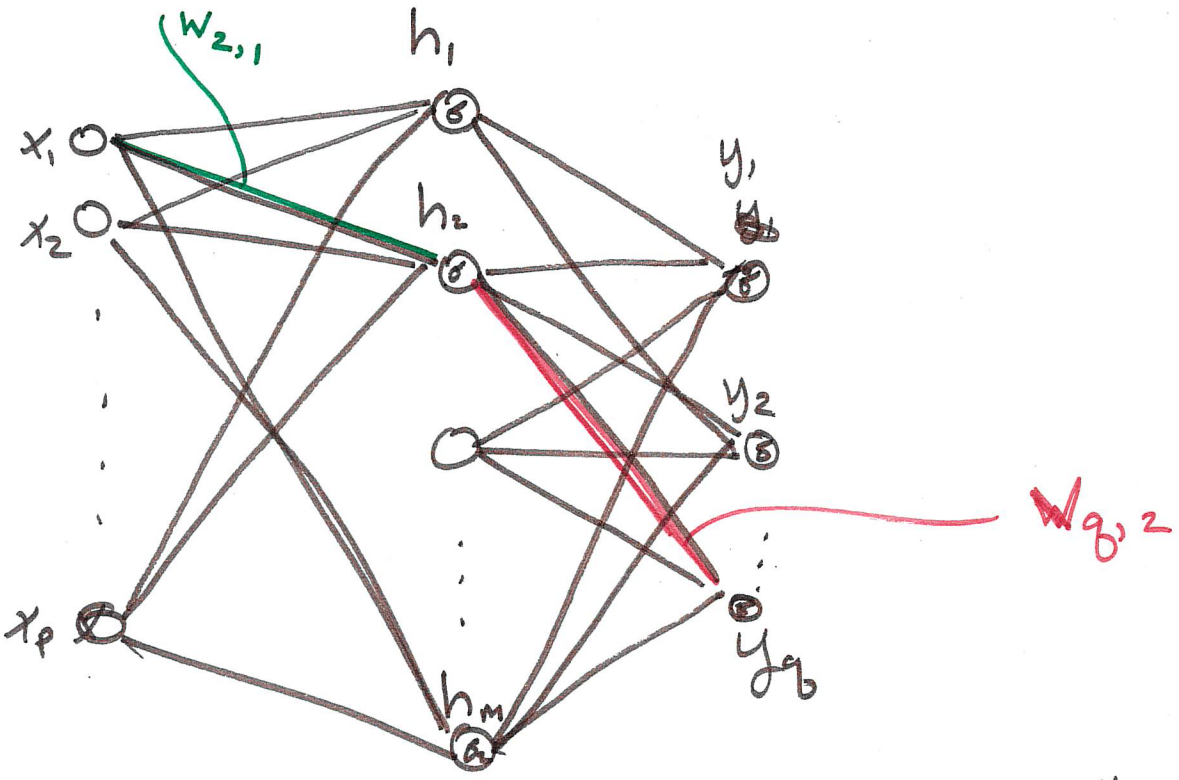
Ex 2 SGD : $f(w) = \|y - Xw\|_2^2 + \lambda \|w\|_1$ (LASSO)

$$= \sum_{i=1}^{n} \underbrace{\left[(y_i - x_i^T w)^2 + \frac{\lambda}{n} \|w\|_1 \right]}_{f_i(w)}$$

Let $v = -2(y_i - x_i^T w) x_i + \frac{\lambda}{n} \operatorname{sign}(w)$

SGD: $\hat{w}^{(k+1)} = \hat{w}^{(k)} - \frac{\tau}{2}(-v_{i_k})$

$$= \hat{w}^{(k)} + \tau (y_{i_k} - x_{i_k}^T \hat{w}^{(k)}) - \frac{\tau \lambda}{2n} \operatorname{sign}(\hat{w}^{(k)})$$



m-th hidden node's output

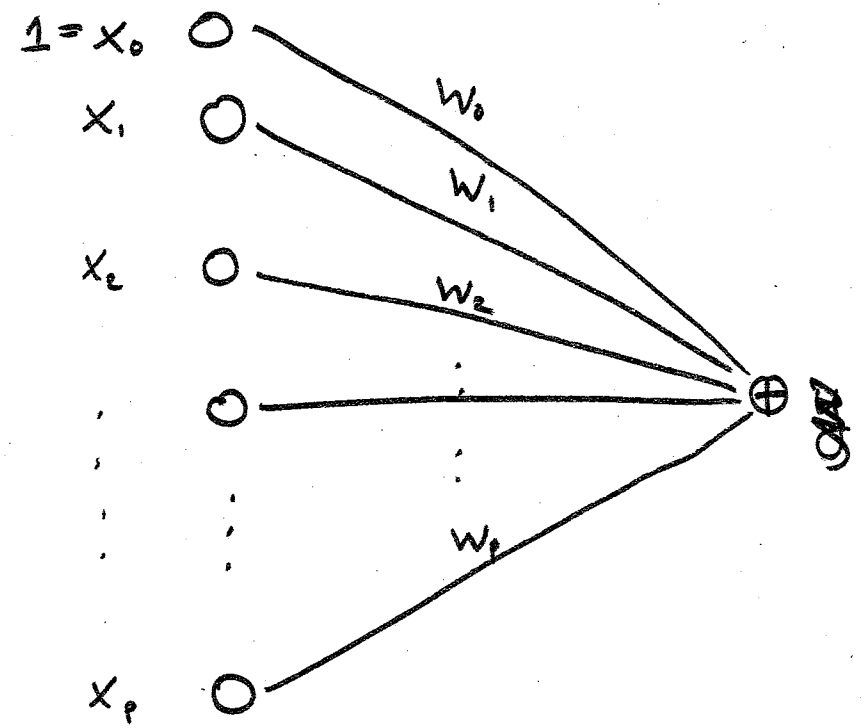
$$h_m = \sigma\left(\sum_{j=1}^{p} x_j w_{m,j}\right)$$

,

k^{th} output node

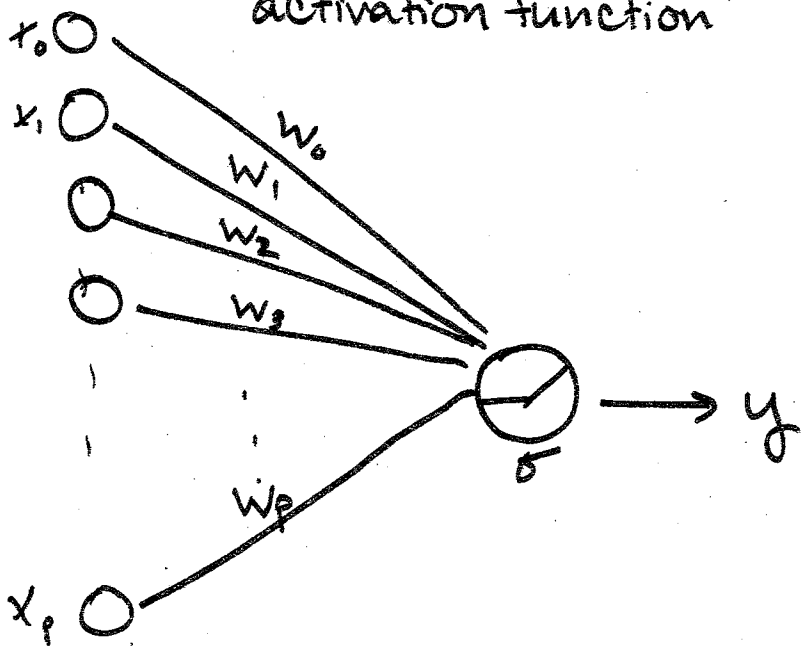
$$y_k = \sigma\left(\sum_{m=1}^{M} h_m v_{k,m}\right)$$

$y = x^T w$



$y = \sigma(x^T w)$

↑ "activation function"



- $\sigma(z) = \max(0, z) = \text{ReLU}$
- $\sigma(z) = \text{sign}(z) \in [-1, 0, +1]$
- $\sigma(z) = \frac{1}{1+e^{-z}} \in [0, 1]$