1) $How\ to\ choose\ weights:\ \ Ans\ \ Gradient\ Descent$
2) $How\ to\ choose\ activation\ functions$
3) $Hidden\ layer\ problem$
4) $How\ to\ speed\ up\ the\ process$

$$we\ know\ that\ the\ more\ hidden\ layers,\ model\ complexity\ will\ increase$$

$$when\ model\ complexity\ increases\ model\ become\ overfit$$

$$Overfit:\ \ Train\ error\ is\ less\ and\ Test\ error\ is\ more$$

$$Overfit:\ \ \ Low\ bias\ -\ High\ variance$$

$$Model\ complexity\ ====\ analogy\ ====\ Increase\ the\ more\ features$$

$$Model\ complexity\ ====\ analogy\ ====\ Increase\ the\ depth\ of\ the\ tree$$

$$Model\ complexity\ ====\ analogy\ ====\ Increase\ the\ more\ hidden\ layers$$

$$It\ is\ very\ important\ to\ avoid\ model\ overfit:\ \ Regularization$$

$Model\ overfit\ happnens\ because\ two\ cases$

- $adding\ more\ layers$
- $less\ train\ data$

1) $Ridge\ and\ Lasso\ Regression$
2) $Drop\ out$
3) $Data\ augmentaion$
4) $Early\ stopping$

$Ridge\ Regression$

$Generally\ In\ Linear\ regression,\ our\ main\ aim\ is\ to\ Reduce\ the\ Cost\ function\ (J)$

$By\ providing\ the\ suitable\ coefficients$

$For\ \ a\ train\ data,\ we\ developed\ a\ LR\ model\ \ \ we\ calculated\ train\ error\ \ i.e.\ cost\ function$

$$J\ =\ \Sigma\left(y_a - y_p\right)^2$$

$$J\ =\ cost\ function$$

$$J\ =\ 0\ \ \ (feel\ very\ happy)\ you\ are\ ready\ to\ test\ the\ data$$

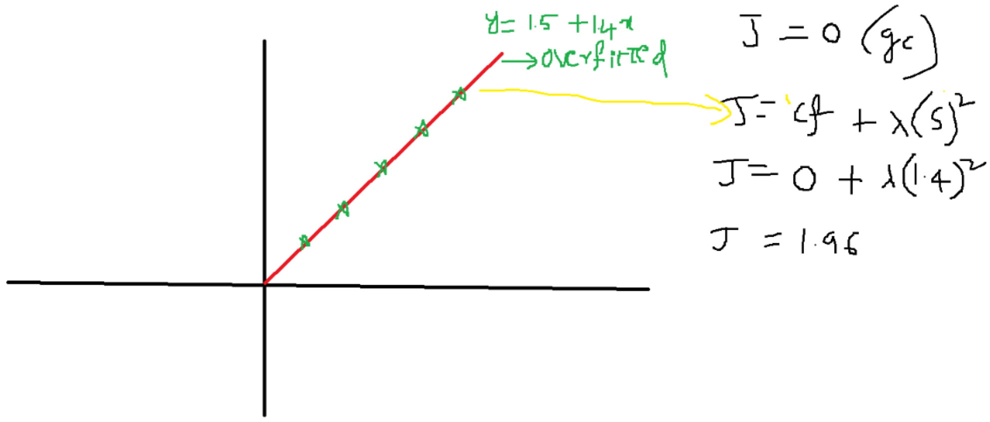
$when\ you\ tested\ \ the\ Error\ becomes\ huge\ (\ BAAAM!):\ \ my\ model\ become\ overfitted$

$Q1)\ when\ will\ know\ model\ become\ overfitted:$

$when\ we\ test\ with\ Test\ data$

$Q2)\ \ who\ is\ giving\ the\ wrong\ direction\ to\ us:\ \ \ J$

$Imagine\ \ J\ != \ 0\ \ \ (\ will\ feel\ the\ equation\ is\ not\ good\ ,\ wil\ try\ with\ another\ equation)$



$J\ =\ cost\ function\ +\ \lambda\ *\ (Slope)^2\ \ \ (\ Ridge\ Regression\ line)$

$J\ =\ cost\ function\ \ (\ General\ Regression\ line)$

$y\ =\ b\ +\ w\ *\ x$

$J\ =\ \Sigma\left(y_a - y_p\right)^2\ \ (\ General\ Regression\ line)$

$J\ =\ \Sigma\left(y_a - y_p\right)^2\ +\ \lambda\ *\ (w)^2\ \ (Ridge\ cost\ function)$

$y\ =\ 1.5\ +\ 1.4\ *\ x\ \ \ (\ assume\ it\ is\ overfitted\ line):\ \ Error\ or\ Cost\ function\ =\ 0$

$J\ =\ \Sigma\left(y_a - y_p\right)^2\ \ (\ General\ Regression\ line) =\ 0$

$J\ =\ \Sigma\left(y_a - y_p\right)^2\ +\ \lambda\ *\ (w)^2\ \ (Ridge\ cost\ function) =\ 0\ +\ \lambda\ *\ (1.4)^2\ =\ 1.96$

$Ridge\ regression\ told\ that\ ,\ I\ have\ error\ so\ go\ for\ another\ equation$

$when\ we\ go\ for\ another\ equation\ means\ we\ are\ avoiding\ the\ overfitted\ line$

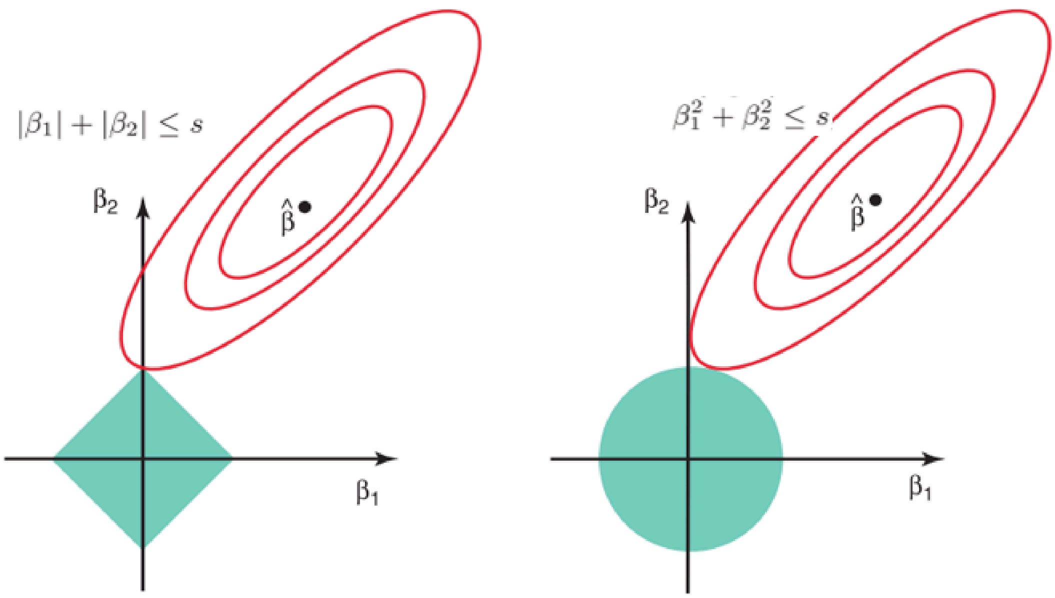
$For\ the\ ridge\ regression\ \ J\ never\ becomes\ zero,\ Error\ increasing\ means\ \ slope\ is\ decreasing$

$vice\ versa\ ,\ to\ main\ tain\ the\ trade\ off\ we\ choose\ \lambda$

$\lambda\ =\ 0\ to\ inf$

$if\ \lambda = 0\ \ Normal\ line\ \ becomes\ \ Ridge\ regression\ line$

1) $Ridge\ regression\ avoids\ overfit$
2) $Suppose\ you\ want\ avoid\ overfit,\ with\ out\ droping\ the\ features : Ridge\ regression$



Lasso Regression Ridge Regression

$Ridge\ regression$	$Laso\ regression$
$Power\ =\ 2\quad L2\ -\ Norm$	$Power\ =\ 1\quad L1\ -\ Norm$
$(slope)^2 = (w)^2\ or\ \left(w_1\right)^2 + (w_2)$^2	$\lvert slope\rvert = \lvert w_1\rvert + \lvert w_2\rvert$
$J\ =\ \Sigma\left(y_a - y_p\right)^2 +\ \lambda * (w)^2$	$J\ =\ \Sigma\left(y_a - y_p\right)^2 +\ \lambda * \lvert w\rvert$
$J\ = CF\ +\ \lambda * (slope)^2$	$J\ = cf\ +\ \lambda * \lvert slope\rvert$

avoid overfit	avoid overfit
if you want to use all the features	if you want to use less fetaure
	Feature selection method

Ridge regression and Lasso regression : Elastic Net regression

Elastic Net regression alpha and lambda (instead of saying two lambda)