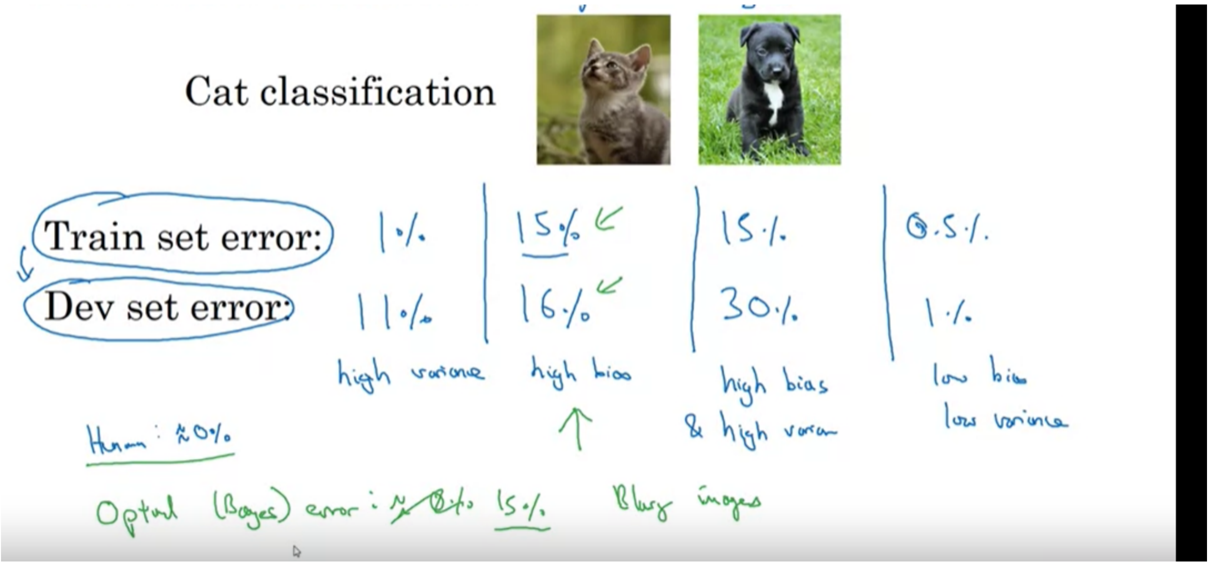
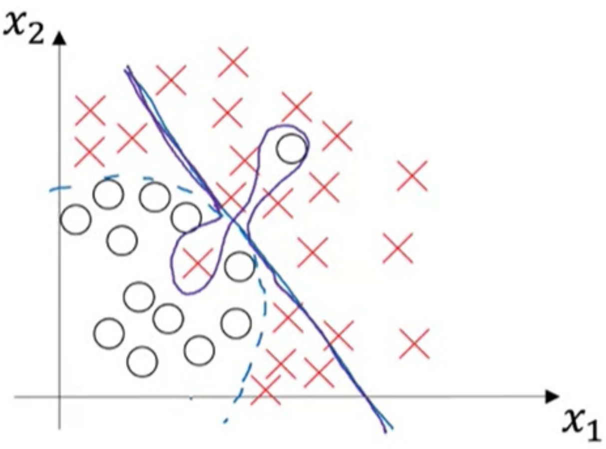
dL.ai_Improvements in Nueral Network

a)Bias-Variance



- High bias means we assume a lot of things about the classifier, we dont consider eenough dimensions.
- High variance means the data is too sensitive to each individual instance in the training data
- They are not antogonistic to each other
- A classifier having high bias and variance will look like the purple line



b) Regularization

i)L1 and L2

- L1 and L2 tries to prevent overfitting by penalizing the model for having high weights..
- We do this by adding weights into the loss function.
- example for logistic loss

$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}(\hat{y}^{(i)}, y^{(i)}) + \frac{\lambda}{2m}\|w\|_2^2$$

L2 regularization $\quad \underline{\|w\|_2^2} = \sum_{j=1}^{n_x} w_j^2 = w^T w$

L1 regularization $\quad \frac{\lambda}{2m}\sum_{i=1}^{n_x}|w| = \frac{\lambda}{2m}\|w\|_1$

- where lambda is the regularization parameter.
- In nueral network ||w|| by the frobenius norm gives

$$\|W^{[l]}\|_F^2 = \sum_{i=1}^{n^{[l-1]}}\sum_{j=1}^{n^{[l]}} \left(W_{ij}^{[l]}\right)^2$$

- during backpropagation when we find dw an additional term $\frac{\lambda}{m}w$ is also considered this leads to the magnitude of weight decreasing.Henc4e l2 is also called weight decay.

$$dW^{[l]} = \boxed{(\text{from backprop}) + \frac{\lambda}{m} W^{[l]}} \qquad \frac{\partial J}{\partial W^{[l]}} = dW^{[l]}$$

$$W^{[l]} := W^{[l]} - \alpha\, dW^{[l]}$$

decay

$$W^{[l]} := W^{[l]} - \alpha\left[(\text{from backprop}) + \frac{\lambda}{m} W^{[l]}\right]$$

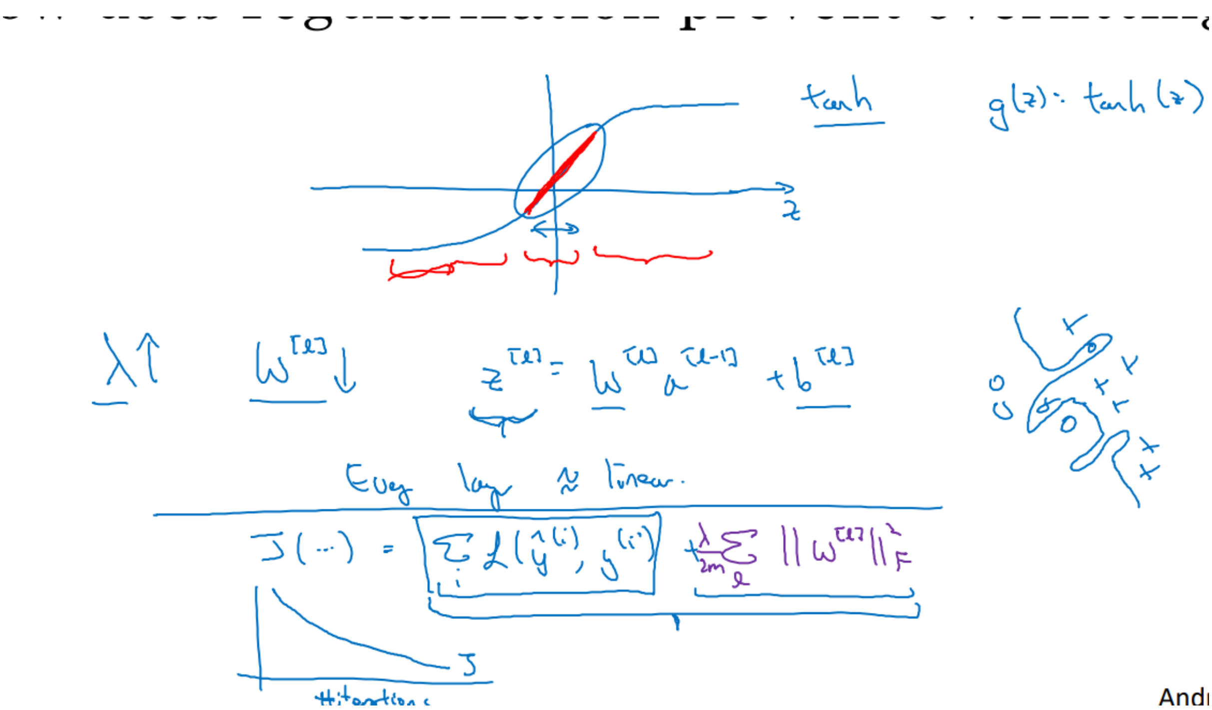
$$(1-\frac{\alpha\lambda}{m}) \qquad = W^{[l]} - \frac{\alpha\lambda}{m} W^{[l]} - \alpha(\text{from backprop})$$

Ar

- L1 regularization leads to sparsity in the model.It completely makes some of the weights zero which leads to some nodes being always inactivated leading to a smaller model.I hope this because when we

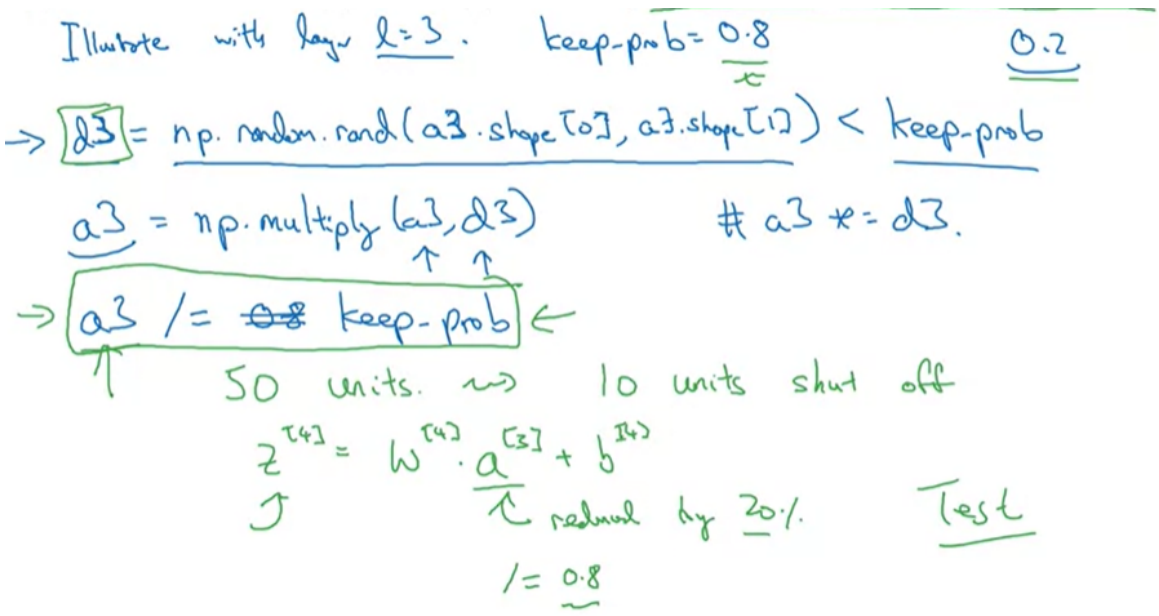
$$J = CE + \frac{\lambda}{m}|w|$$
$$dw = (bpterm) + \frac{\lambda}{m}$$
$$w^{[l]} - dw^{[l]} = w^{[l]} - a(bpterm) - a\frac{\lambda}{m}$$

- let $w^{[l]}$ be [1,2,3,4,5] then each weight in w will be decreased by the same constant value till they reach zero or no of epochs complete.
- However in L2 since $\frac{\lambda}{m}w$ is proprtional to weight the extreme values are penalised more and the lower values are penalised lesser.It tends to make all the weights smaller but not exactly zero.
- Even though variance and bias are not perfectly complementary we can say that if a model is closer to a linear regressor it would decrease the variance
- so if value of w is less the value of z would be less, if the value of z is less then sigmoid function will more or less converge to a linear y-x graph. Hence the whole model comes closer to a linear regressor.



ii) Inverted Dropout

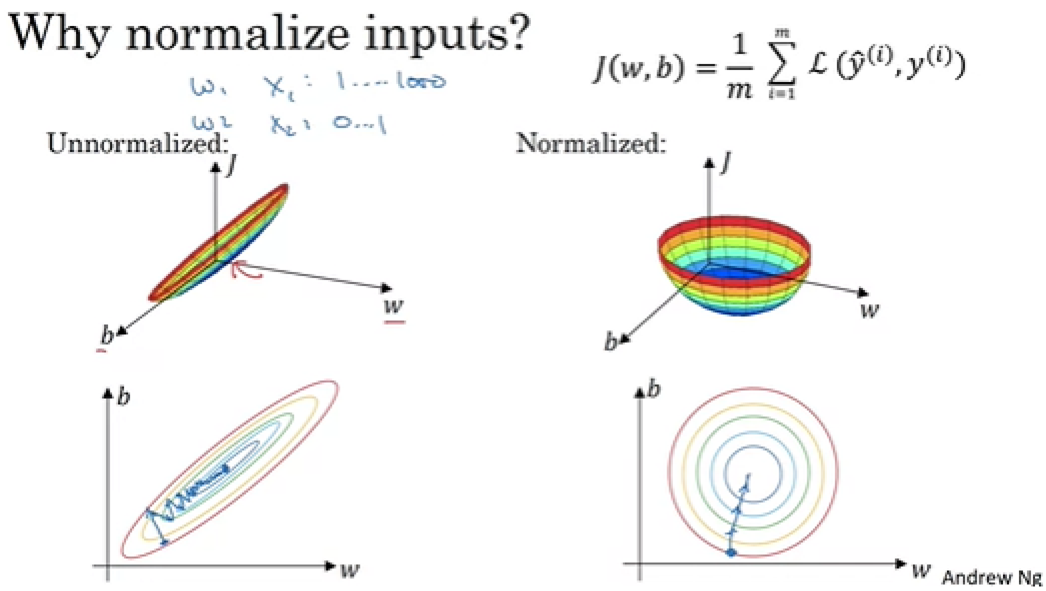
- During training in a given layer all the nodes are not taken a fixed ratio of nodes are always dropped out.



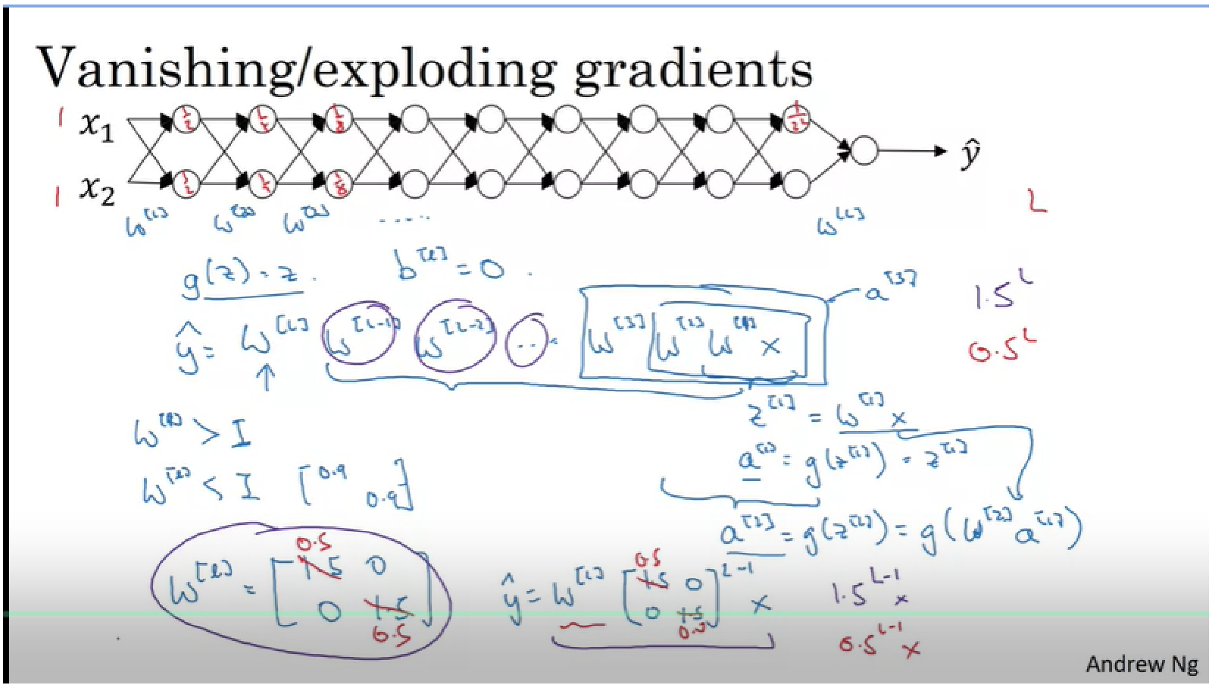
- Understand how the value of z doesnt change even though though several a3 i have become zero
- Dropout reduces the dependence of a model on a single node
- Dropout make Loss function more vague

- Drop out ensures that the network cant depend entirely on any one feature this ensures that the weights are spread out more. This has a similar effect to L2 regularization
- During each iteration of training the input data only passes through a smaller nn.

c)Normalizing



d)Vanishing/Exploding Gradient


Vanishing/exploding gradients
x_1
x_2
$\hat{y}$
Andrew Ng