OVERFITTING & REGULARIZATION FOR NEURAL NETWORKS

LECTURE 7
SECTION 3
JUNE 12TH

INSTITUTE FOR APPLIED COMPUTATIONAL SCIENCE
AT HARVARD UNIVERSITY

UNIVERSITY of RWANDA

VARIANCE REDUCTION FOR NEURAL NETWORKS

OVERFITTING AND REGULARIZATION:

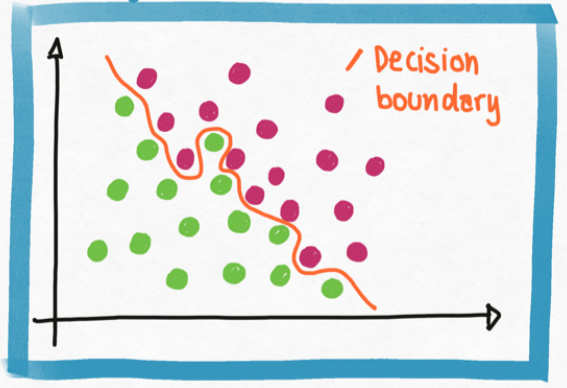
Neural networks are complex functions and they can easily overfit

Just like in polynomial regression, we can penalize the parameters w of the polynomial to keep the values of w small.

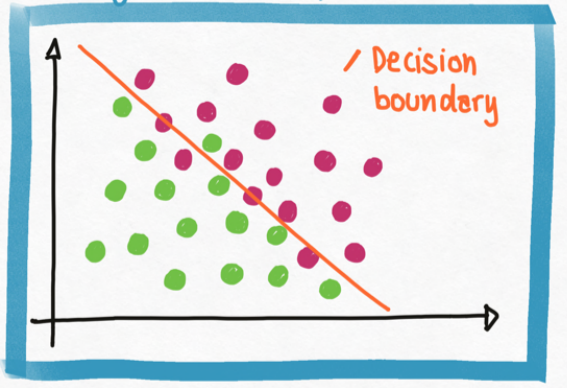
This changes our training objective to be:

$$W^* = \arg\min_W \underbrace{-\ell(W)}_{\text{negative log likelihood}} + \underbrace{\|W\|_2}_{\sum_i W_i}$$

Unregularized NN Classifier



Regularized NN Classifier



REGULARIZATION IN NEURAL NETWORKS:

In keras you can penalize the weights or the outputs of the hidden nodes in two ways:

Ridge or ℓ_2: $\|W\|_2 = \sum_i W_i^2$

LASSO or ℓ_1: $\|W\|_1 = \sum_i |W_i|$

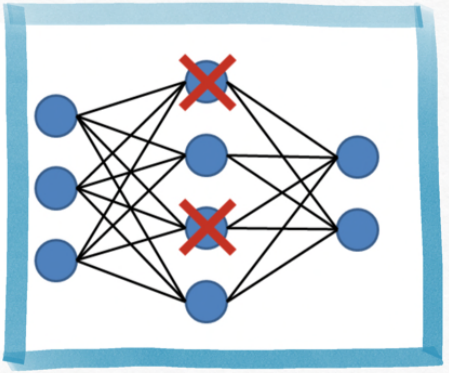
```
from keras import regularizers
model.add(Dense(64, input_dim=64,
                kernel_regularizer=regularizers.l2(0.01),
                activity_regularizer=regularizers.l1(0.01)))
```

this puts ℓ_2 penalty on the weights with $\lambda = 0.01$

this puts ℓ_1 penalty on the outputs of the nodes

TRAINING WITH DROPOUT:

Another way to prevent ovefitting is to not always use very hidden node in the network:



1. in each evaluation of the network during training, we randomly set each weight to zero with a fixed probability called the drop-out probability.
   This prevents the network from being too complex
2. for each X, we evaluate the network S times, each time randomly dropping out weight. We then average the S predictions.
   This is just training an ensemble of S models.

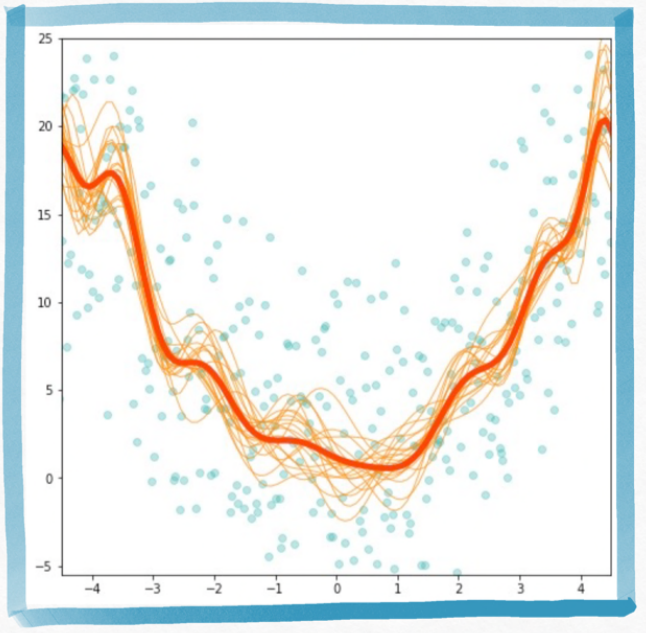
Dropout in keras

```
from keras.layers import Dropout,
model.add(Dense(64, activation='relu', input_dim=20))
model.add(Dropout(0.5)) ← add dropout to this layer with probability 0.5
```

ENSEMBLING NEURAL NETWORKS:

When we fit complex models, they will capture the noise in the data: sometimes the model will predict values that are higher than it should, sometimes the model will predict values that are lower than it should.

But when we average their predictions, these errors (too high and too low) will cancel out.



/ Bootstrap Neural Networks

/ Average prediction of the ensemble