L7

Universal approximation theorem

$f:\ R^d \rightarrow R$

- Can be represented as a network with only one hidden layer
- There will exist an activation function
  - The problem is that this theorem states that there exists the function, and doesn't give the neural net itself
- also in any machine learning problem you don't even know the function

How many number of hidden units must be there to figure out the function?

- It is exponential (takes too long to understand)
- The weight matrix becomes too large to compute

l layers

- $O(\binom{n}{d}^{d(l-1)} n^d)$ decision regions, for d features, and n neurons per hidden later
- if l >=5 then it's called as a deep neural network

Linear Models

- Parameters are linear
  - Eg. Support vector machines
  - or use a linear activation also
- Classical ML can do nonlinear shit too btw.

if t is target
ϕ is transform
{ ($\phi(x1)$,t1), ($\phi(x2)$,t2), ($\phi(x3)$,t3).... ($\phi(xd)$,td) }
Phi can be exponential
we can use a linear model on top of this

How to make the raw features to matured features?

1. the specific basis functions can be applied by the domain expert, to get the matured featues, and then apply the linear model
   - Example, BMI
2. Kernels in SVMs, going with some kind of functions to transform. Eg. Gaussian Kernel