Zhang 3

Linear NN's for Regression

Shallow neural Networks

- basis NN training
- Parametrizing the output layer
- Handling data
- Specifying loss func
- Training model

This chapter = focus narrow on linear regress — Next expands repertoire to classification

3.1 linear Regression

Assume relationship between features x & output one linear

⇒ the conditional mean $E[Y|X=x]$ can be expressed as the weighted sum of the features

$$\text{Price} = w_A \cdot \text{Area} + w_{ag} \cdot \text{Age} + b$$

ML data tends to be high Dims
so more convein to use complach linA

$$\hat{y} = w^T x + b$$

goal is to learn appropriate vals
for w, b that model the output
distribution

Before we search for best paras
we need two more things
- A measure of Quality
- A procedure for updating the model

loss func = Diff between actual & Pred

Common regression loss func = Square error

$$l^{(i)}(w,b) = \frac{1}{2}\left(\hat{y}^{(i)} - y^{(i)}\right)^2$$

$$L(w,b) = \frac{1}{n}\sum^{n} \frac{1}{2}\left(w^T x^{(i)} + b - y^{(i)}\right)^2$$

$$w^*, b^* = \operatorname{argmin} L(w,b)$$

3.2 Object-Oriented Design for implementation

This chapter guides & gives tips on how to set up effective code structure & Design ~~to~~ for training ~~it~~, opt & valid models

1st Setup helper classes of things that will be reused alot

Add to class, set hyperparams, ~~prog~~ visual tools

Set up a main class & make it a sub class of above tools

- ~~Module~~ Model Class
- Data handlere ~~[illegible]~~ Class
- Trainer Class

3.3 Synthetic Regression Data

Creating & training on syn data is useful because ~~it~~ if we know the correct paras & distribs

then we can check if our model can recover them (Paras & Dists)

3.4 lin Reg Implem from Scratch

1. Model
2. loss func
3. ~~training func~~ Mini Batch optimizer
4. Combined train func

DL frameworks have all of this built in but coding from scratch allow full understanding