4. linear NN for classific

hot one encoding is the best way to classify class which do not come with a natural order

$$y \in \{(1,0,0), (0,1,0), (0,0,1)\}$$

cat dog Bat

linear model 4.1.1.1

to est condition probs of each class, we need 3 Model per class

(-1) as technicaly is 3 output we only need 2 nodes as 3rd = 1 - 1st - 2nd

4 feates + 3 output = 12 weights

4.1.1.2 Softmax

for probs output to be valid they need to be ≥ 0 & sum to 1

Can't treat Claff this regression as no guarentee these rules hold

We need to "Squish" model output

best way = expo funct $P(y=1) \propto e^{O_i}$

Result:
- Cond prob increase w/ O_i
- Monotonic
- All probs are non-neg
- ~~Sum~~ Probs sum to 1

This process is called Normalization

Putting all this together we get the SoftMax function

$\hat{y} = Softm(o)$ where $\hat{y}_i = \frac{\exp(o_i)}{\sum_j \exp(o_j)}$

O = Class output model

Softmax Sigmoid func for >2 Classes

$\text{argmax}\ \hat{y}_j = \text{Argmax}_j\ O_j$

4.1.2 loss function

Now that we are able to map features x to probabilities $\hat{y}$

Need a way to opt accuracies of map

Rely on max likelihood Functions

4.1.2.1 log-likelihood

$\hat{y}$ = output from O_i & Softmax

can be interpretted as conditional probs of each class given any input x

$\hat{y} = P(y = cat \mid X)$

Assume that X & Y are Hot one encoded

Check high-level accuracy estimate by

$$P(Y|X) = \prod_{i=1}^{n} P(y^{(i)} \mid x^{(i)})$$

Aka Joint Probability

Maximizing the prod of terms is awkward
So we take the neg of log of term

$$-\log(P(Y|X)) = \sum -\log P(y^{(i)} \mid x^{(i)})$$

this gives way to the loss function

$$\sum_{i=1}^{n} \mathcal{L}(y^{(i)}, \hat{y}^{(i)})$$

where $\mathcal{L}(y, \hat{y}) = -\sum_{j=1}^{q} y_j \cdot \log \hat{y}_j$

this is called cross-entropy loss

Recall y = Hot one encoded vector
= the 1 triggers the calc like a switch, the 0s turn off