CS 316: Introduction to Deep Learning

Softmax Regression
Week 5

Dr Abdul Samad

Lecture Outline

- Regression vs Classification
- Multiclass Classification
- Network Architecture
- One-hot Encoding
- Softmax
- Cross entropy loss

Regression vs Classification

- Regression estimates a continuous value
- Classification predicts a discrete category

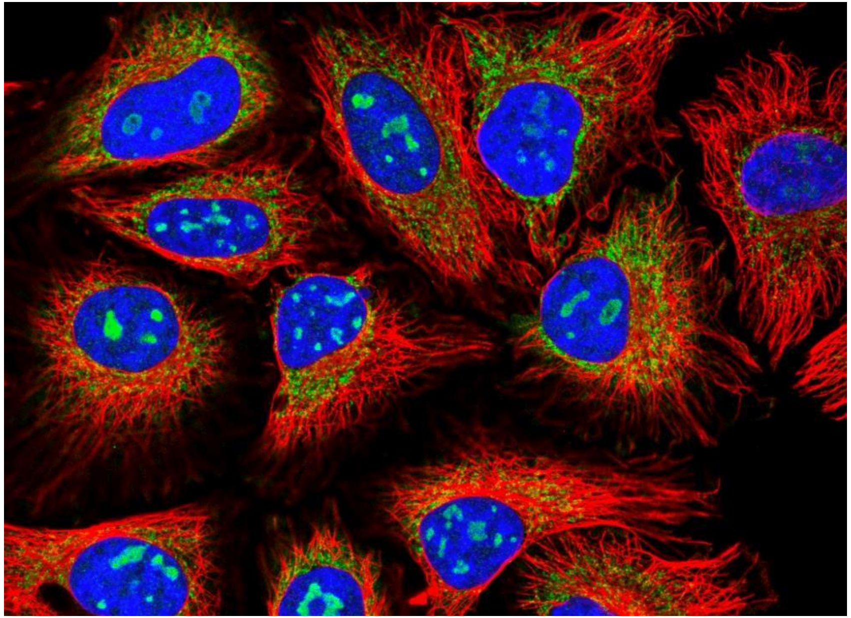
MNIST: classify hand-written digits (10 classes)

ImageNet: classify nature objects (1000 classes)

Figures taken from [MNIST Digits Dataset](), [ImageNet Dataset]()

Classification Tasks at Kaggle

Classify human protein microscope images into 28 categories



0. Nucleoplasm
1. Nuclear membrane
2. Nucleoli
3. Nucleoli fibrillar
4. Nuclear speckles
5. Nuclear bodies
6. Endoplasmic reticu
7. Golgi apparatus
8. Peroxisomes
9. Endosomes
10. Lysosomes
11. Intermediate fila
12. Actin filaments
13. Focal adhesion si
14. Microtubules
15. Microtubule ends
16. Cytokinetic bridg

[Human Protein Atlas Image Classification]

Classification Tasks at Kaggle

Classify malware into 9 categories

[Malware Classification]

Classification Tasks at Kaggle

Classify toxic Wikipedia comments into 7 categories

comment_text	toxic	severe_toxic	obs(
Explanation\nWhy the edits made under my usern...	0	0	0
D'aww! He matches this background colour I'm s...	0	0	0
Hey man, I'm really not trying to edit war. It...	0	0	0
"\nMore\nI can't make any real suggestions on ...	0	0	0
You, sir, are my hero. Any chance you remember...	0	0	0

[Jigsaw Toxic Comment Classification Challenge]

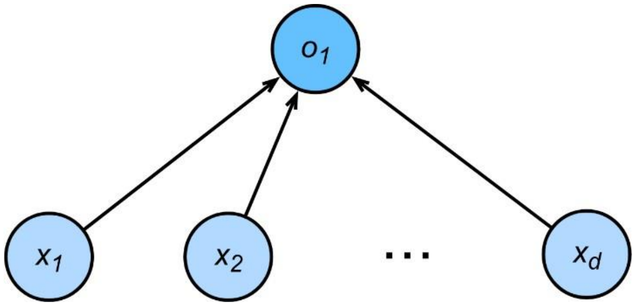
Multi-class Classification

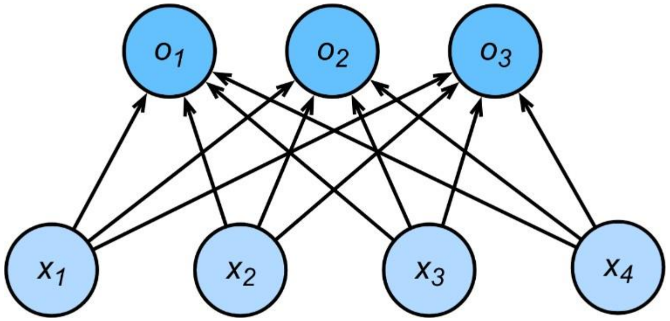
Regression

- Single output value.
- Output is a real value. $o_1 \in \mathbb{R}$
- Loss is defined as difference of $\hat{y} - y$

Classification

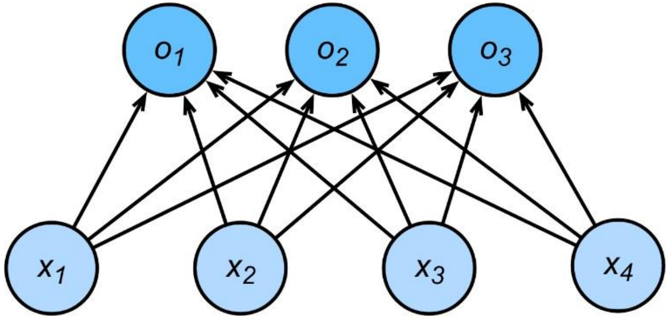
- Multiple outputs
- Output represents probability i.e. $o_1, o_2, o_3 \in (0, 1)$
- Score represents amount of confidence in predicting a class





Figures taken from [Linear Regression](), [SoftMax Regression]()

Network Architecture



$$o_1 = x_1 w_{11} + x_2 w_{12} + x_3 w_{13} + x_4 w_{14} + b_1$$
$$o_2 = x_1 w_{21} + x_2 w_{22} + x_3 w_{23} + x_4 w_{24} + b_2$$
$$o_3 = x_1 w_{31} + x_2 w_{32} + x_3 w_{33} + x_4 w_{34} + b_3$$

$$\mathbf{o} = \mathbf{W}\mathbf{x} + \mathbf{b}$$

Figure Source: [SoftMax Regression]

One-hot Encoding

$$\text{Given } \mathbf{y} = [y_1, y_2, y_3, \cdots, y_n]^\top$$

$$y_i = \begin{cases} 1 & \text{if i = class} \\ 0 & \text{otherwise} \end{cases}$$

One-hot Encoding - Example

$$\text{Given } \mathbf{y} = [\text{Cat, Dog, Cat, Frog}]^\top$$

One-hot Encoding - Example

$$\text{Given } \mathbf{y} = [\text{Cat}, \text{Dog}, \text{Cat}, \text{Frog}]^\top$$

$$\mathbf{y} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Softmax

- Model output can be interpreted as probabilities.
- Optimize the parameters to generate probabilities that maximize the probability of the observed data.

$$\hat{\mathbf{y}} = softmax(\mathbf{o})$$

$$\hat{y}_i = \frac{\exp(o_i)}{\sum_{j=1}^{k} \exp(o_j)}$$

where k is the number of classes.

$$\hat{y}_1 + \cdots + \hat{y}_k = 1$$

$$\operatorname*{argmax}_i \hat{y}_i = \operatorname*{argmax}_i o_i.$$

Vectorized Approach

$$\mathbf{X} \in \mathbb{R}^{n \times d}$$

$$\mathbf{W} \in \mathbb{R}^{d \times q}$$

$$\mathbf{b} \in \mathbb{R}^{1 \times q}$$

where n is the number of examples , d is the number of features and q is the number of classes.

$$\mathbf{O} = \mathbf{X}\mathbf{W} + \mathbf{b}$$

$$\mathbf{Y} = softmax(\mathbf{O})$$

Example - Softmax

Input Image

Model

Logit (O)

3.2

1.3

0.2

0.8

Softmax

$$\hat{y}_i = \frac{\exp(o_i)}{\sum_{j=1}^{k} \exp(o_j)}$$

Probabilities

Label

Classes
Dog
Cat
Bear
Tiger

Figure Source: [Dog Image]

Example - Softmax

Input Image	Model	Logit (O)	Softmax	Probabilities	Label

Logit (O):

3.2
1.3
0.2
0.8

Softmax:

$$\hat{y}_i = \frac{\exp(o_i)}{\sum_{j=1}^{k} \exp(o_j)}$$

Probabilities:

0.78
0.12
0.04
0.07

Classes
Dog
Cat
Bear
Tiger

Figure Source: [Dog Image]

Loss Function

- The dataset contains n instances represented by $(\mathbf{X}, \mathbf{Y})$, where $\mathbf{x}^{(i)}$ represents the i^{th} instance and $\mathbf{y}^{(i)}$represents the one-hot encoded label vector.
- Given the features, how likely are the actual classes according to our model?

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

- Using Maximum Likelihood (MLE),

$$-\log P(\mathbf{Y} \mid \mathbf{X}) = \sum_{i=1}^{n} -\log P(\mathbf{y}^{(i)} \mid \mathbf{x}^{(i)}) = \sum_{i=1}^{n} l(\mathbf{y}^{(i)}, \hat{\mathbf{y}}^{(i)})$$

Loss Function

$$P(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}) = \prod_{j=1}^{q} \left[P(\mathbf{y}_j^{(i)}|\mathbf{x}^{(i)})\right]^{y_j}$$

$$P(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}) = \prod_{j=1}^{q} \left[\hat{y}_j\right]^{y_j}$$

$$-\log P(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}) = -\sum_{j=1}^{n} y_j \log\left(\hat{y}_j\right) = l(\mathbf{y}^{(i)}, \hat{\mathbf{y}}^{(i)})$$

$$-\log P(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}) = l(\mathbf{y}^{(i)}, \hat{\mathbf{y}}^{(i)})$$

Example – Cross Entropy Loss

$$\text{Given } \hat{\mathbf{y}}^{(i)} = \begin{bmatrix} 0.78 \\ 0.12 \\ 0.04 \\ 0.07 \end{bmatrix} \text{ and } \mathbf{y}^{(i)} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \text{ compute } l(\hat{\mathbf{y}}^{(i)}, \mathbf{y}^{(i)})$$

Example – Cross Entropy Loss

$$\text{Given } \hat{\mathbf{y}}^{(i)} = \begin{bmatrix} 0.78 \\ 0.12 \\ 0.04 \\ 0.07 \end{bmatrix} \text{ and } \mathbf{y}^{(i)} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \text{ compute } l(\hat{\mathbf{y}}^{(i)}, \mathbf{y}^{(i)})$$

$$l(\hat{\mathbf{y}}^{(\mathbf{i})}, \mathbf{y}^{(\mathbf{i})}) = -\sum_{j=1}^{q} y_j \log\left(\hat{y}_j\right)$$

$$l(\hat{\mathbf{y}}^{(\mathbf{i})}, \mathbf{y}^{(\mathbf{i})}) = -\log(0.78) = 0.108$$

Derivative of Softmax

$$l(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{j=1}^{q} y_j \log \frac{\exp(o_j)}{\sum_{k=1}^{q} \exp(o_k)}$$

Using log properties to simplify the expression

$$l(\mathbf{y}, \hat{\mathbf{y}}) = \sum_{j=1}^{q} y_j \log \sum_{k=1}^{q} \exp(o_k) - \sum_{j=1}^{q} y_j o_j$$

$$l(\mathbf{y}, \hat{\mathbf{y}}) = \log \sum_{k=1}^{q} \exp(o_k) - \sum_{j=1}^{q} y_j o_j$$

$$\frac{\delta l(\mathbf{y}, \hat{\mathbf{y}})}{\delta o_j} = \frac{\exp(o_j)}{\sum_{k=1}^{q} \exp(o_k)} - y_j = \mathrm{softmax}(\mathbf{o})_j - y_j.$$