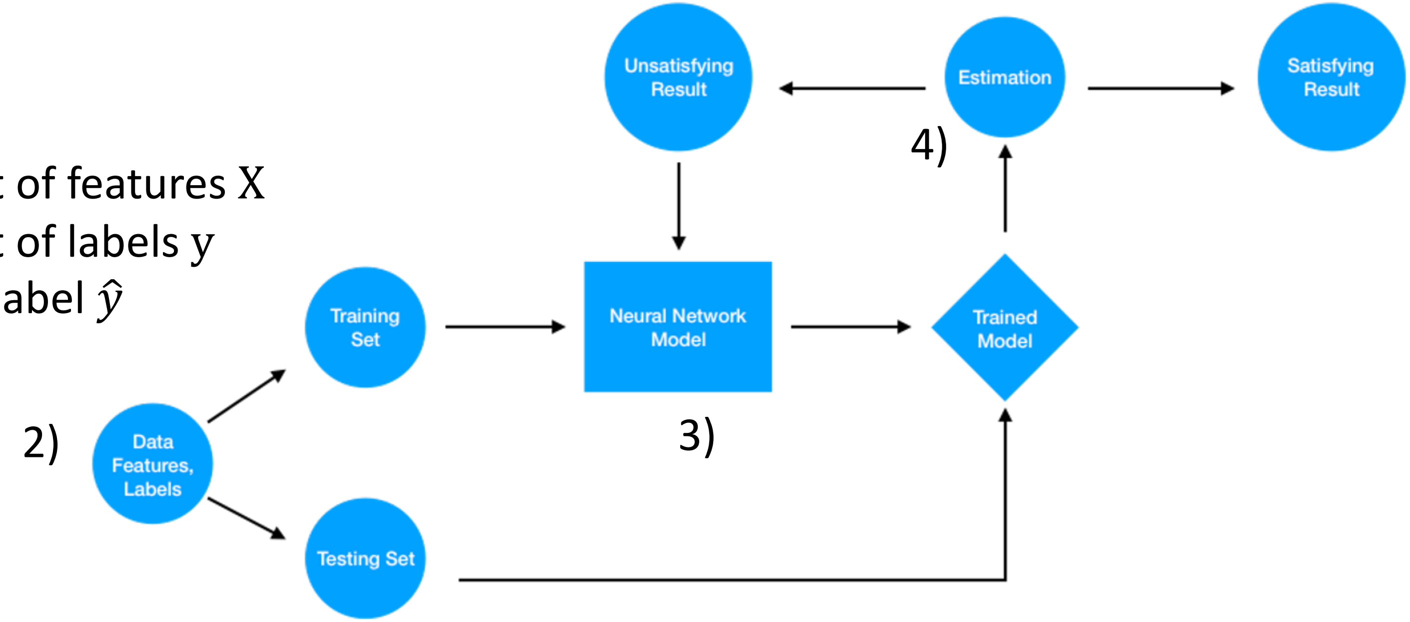
Logistic/Softmax Regression

Outline

1) Difference between Linear, Logistic, and Softmax regression?
2) How to represent the labels? (One-hot-encoding)
3) How to estimate the outputs of our model as probability? (Network Architecture, Softmax Operation)
4) How to measure the quality of our predicted probabilities? (loss function, maximum likelihood estimation, cross-entropy loss)

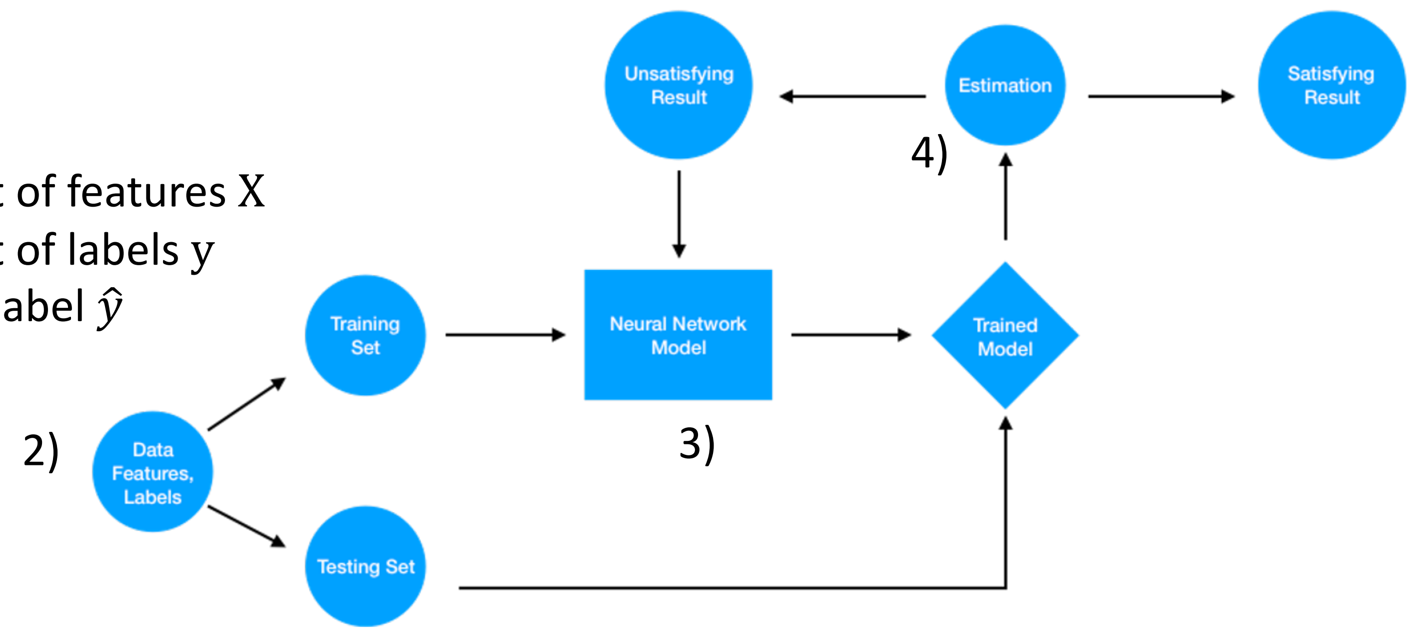
- training set of features X
- training set of labels y
- estimated label $\hat{y}$



Outline

1) **Difference between Linear, Logistic, and Softmax regression?**
2) How to represent the labels? (One-hot-encoding)
3) How to estimate the outputs of our model as probability? (Network Architecture, Softmax Operation)
4) How to measure the quality of our predicted probabilities? (loss function, maximum likelihood estimation, cross-entropy loss)

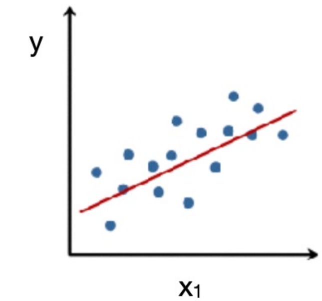
- training set of features X
- training set of labels y
- estimated label $\hat{y}$



Regression

- Target label is an interval value. “How much”
- E.g., fundamental plane of galaxies

Linear regression

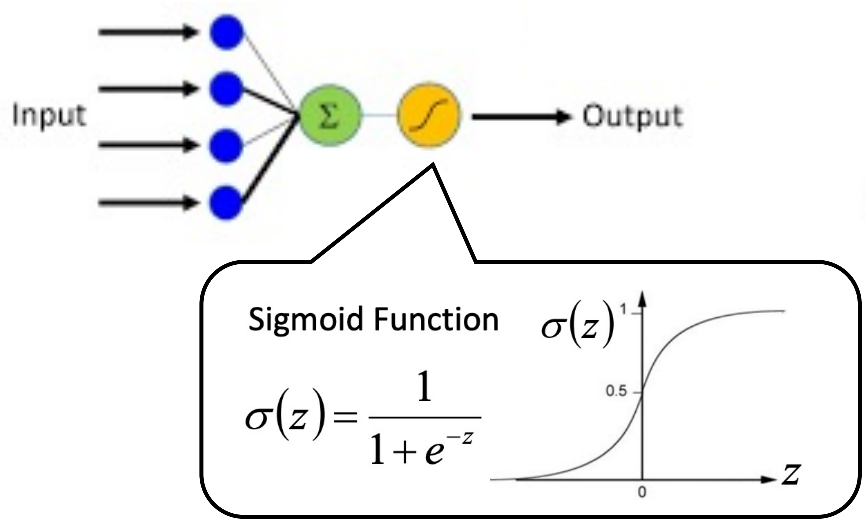


Classification

- Target label is a discrete value. “Which one”
- E.g., Galaxy Morphological Classification

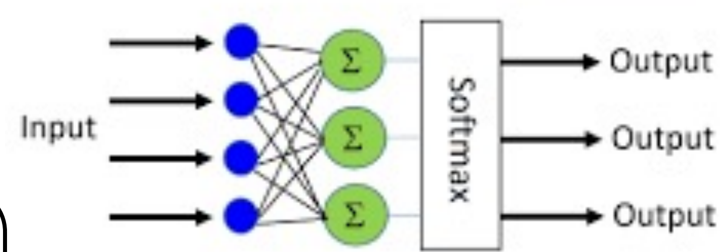
Logistic regression

- Binary classification



Softmax regression (multinomial logistic regression)

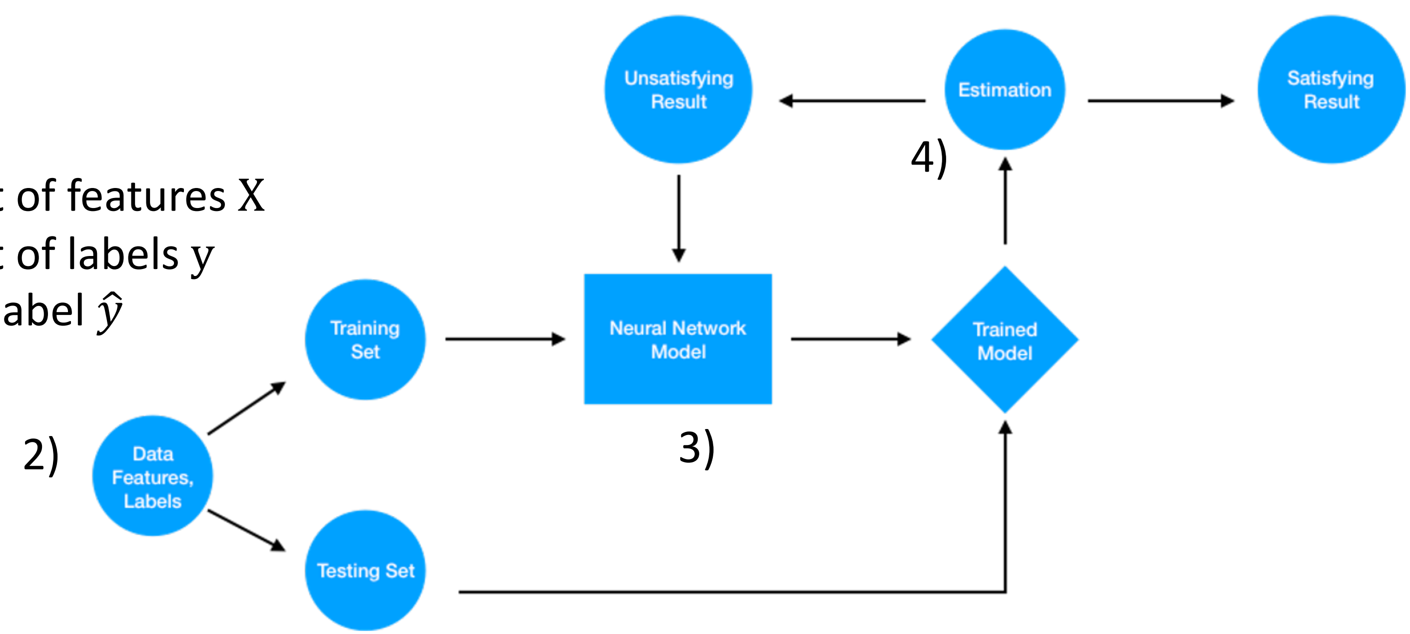
- Multi-class classification



Outline

1) Difference between Linear, Logistic, and Softmax regression?
2) **How to represent the labels? (One-hot-encoding)**
3) How to estimate the outputs of our model as probability? (Network Architecture, Softmax Operation)
4) How to measure the quality of our predicted probabilities? (loss function, maximum likelihood estimation, cross-entropy loss)

- training set of features X
- training set of labels y
- estimated label $\hat{y}$



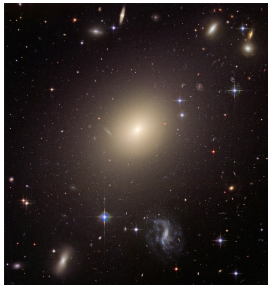
Q2: How to represent the labels?

one-hot encoding

- A one-hot encoding is a vector with as many components as we have categories. The component corresponding to particular instance's category is set to 1 and all other components are set to 0.
- $y \in \{1,2,3\}$
- $y \in \{(1,0,0), (0,1,0), (0,0,1)\}$

If $x \in$ class 1

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



If $x \in$ class 2

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

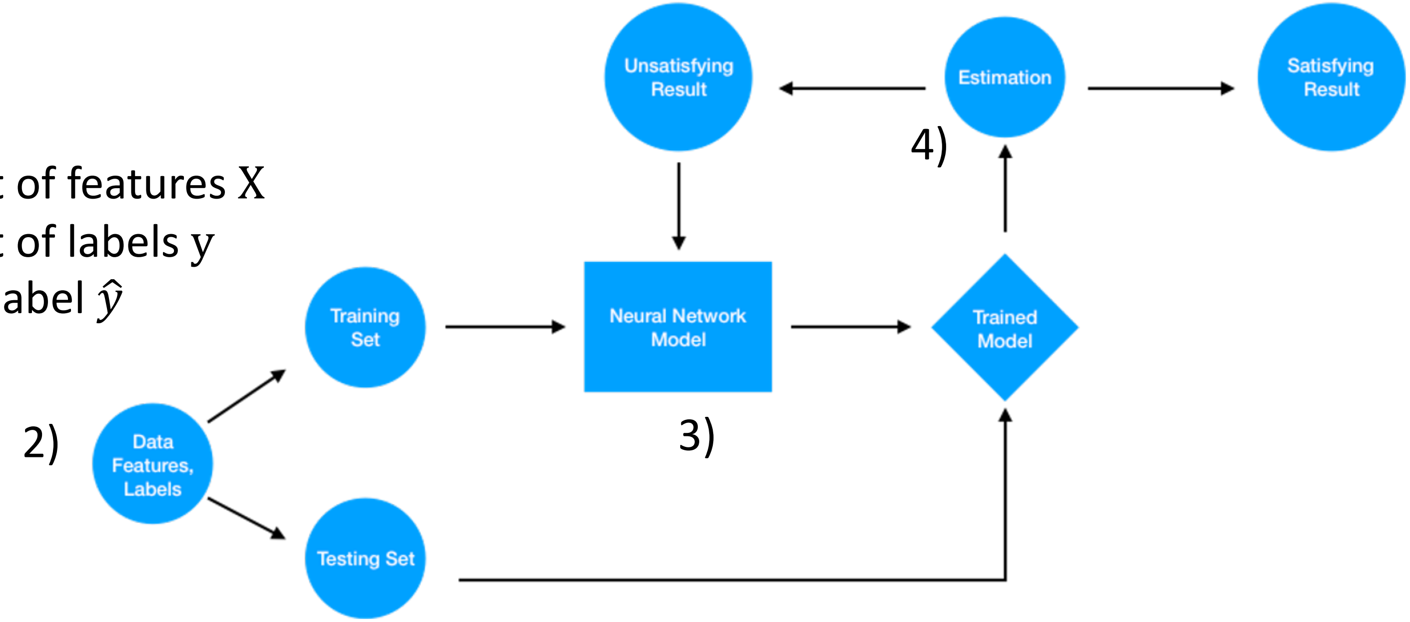
If $x \in$ class 3

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

Outline

1) Difference between Linear, Logistic, and Softmax regression?
2) How to represent the labels? (one-hot-encoding)
3) **How to estimate the outputs of our model as probability? (Network Architecture, Softmax Operation)**
4) How to measure the quality of our predicted probabilities? (loss function, maximum likelihood estimation, cross-entropy loss)

- training set of features X
- training set of labels y
- estimated label $\hat{y}$



Q3: How to estimate the outputs of our model as probability?

Network Architecture

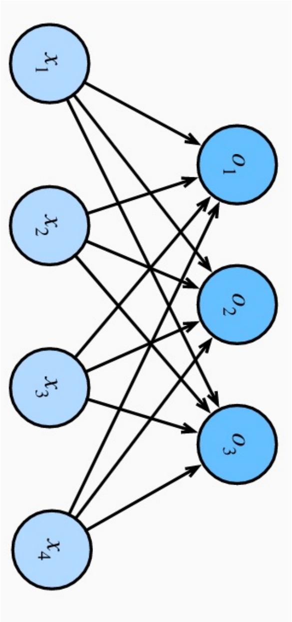
- In order to estimate the conditional probabilities associated with all the possible classes, we need a model with multiple outputs, one per class.

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

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

$\boldsymbol{o}$: *logits*
$\boldsymbol{x}$: independent variables or feature vector
$\boldsymbol{w}$: the weight vector (of the linear model)
$\boldsymbol{b}$: bias or y-intercept.

Input layer Output layer



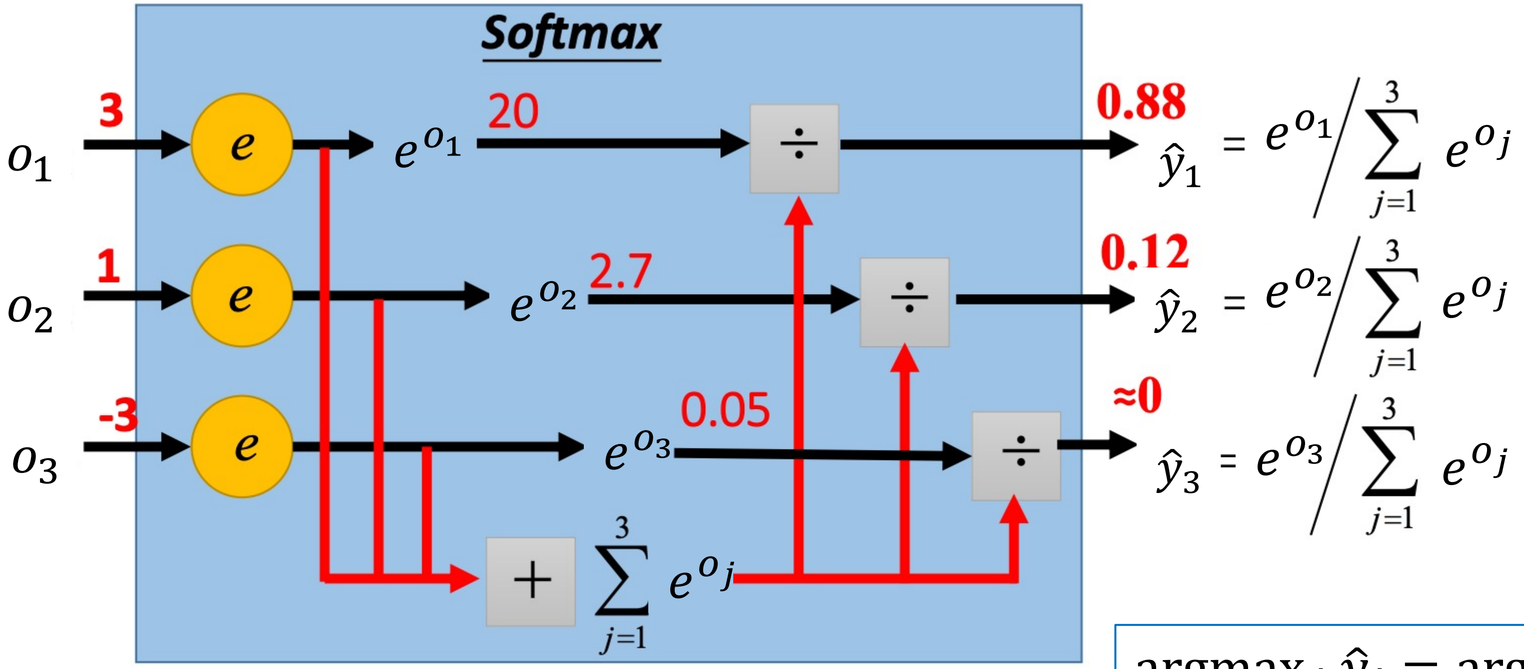
W matrix : 4x3 parameters
b bias : 3 parameters (for 3 neurons in the output layer)

Q3: How to estimate the outputs of our model as probability?

- To generate predictions, we will set a threshold.
  - E.g., choosing the label with the maximum predicted probabilities.
- Can we interpret the logits o directly as our outputs of interest?
  - No, we need the **Softmax Operation**

$$\hat{y} = \mathrm{softmax}(\mathbf{o}), \text{ where } \hat{y}_j = \frac{\exp(o_j)}{\sum_k \exp(o_k)}$$

Probability
- $0 \leq \hat{y}_j < 1$
- $\sum_j \hat{y} = 1$



$$\mathrm{argmax}_j\, \hat{y}_j = \mathrm{argmax}_j\, o_j$$

Vectorization for Minibatche

$$\mathbf{O} = \mathbf{X}\mathbf{W} + \mathbf{b}$$
$$\hat{\mathbf{Y}} = \mathrm{softmax}(\mathbf{O})$$

- To improve computational efficiency and take advantage of GPUs, we typically carry out vector calculations for minibatches of data.
- Assume that we are given a minibatch $\mathbf{X}$ of examples with feature dimensionality (number of inputs) d and batch size n. Also, we have q categories in the output.

$\mathbf{X}$ (d feature dimensionality; n examples) $\quad \mathbf{W}$ (q outputs; d dim.) $\quad + \quad \mathbf{b}$

$$\begin{pmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_d^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \cdots & x_d^{(2)} \\ \vdots & \vdots & & \vdots \\ x_1^{(n)} & x_2^{(n)} & \cdots & x_d^{(n)} \end{pmatrix} \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1q} \\ w_{21} & w_{22} & \cdots & w_{2q} \\ \vdots & \vdots & & \vdots \\ w_{d1} & w_{d2} & \cdots & w_{dq} \end{pmatrix} + \begin{pmatrix} b_1 & b_2 & \cdots & b_q \\ (b_1 & b_2 & \cdots & b_q)_{1\times q} \\ \vdots & \vdots & & \vdots \\ b_1 & b_2 & \cdots & b_q \end{pmatrix}$$

broadcasting
$1\times q \rightarrow n\times q$

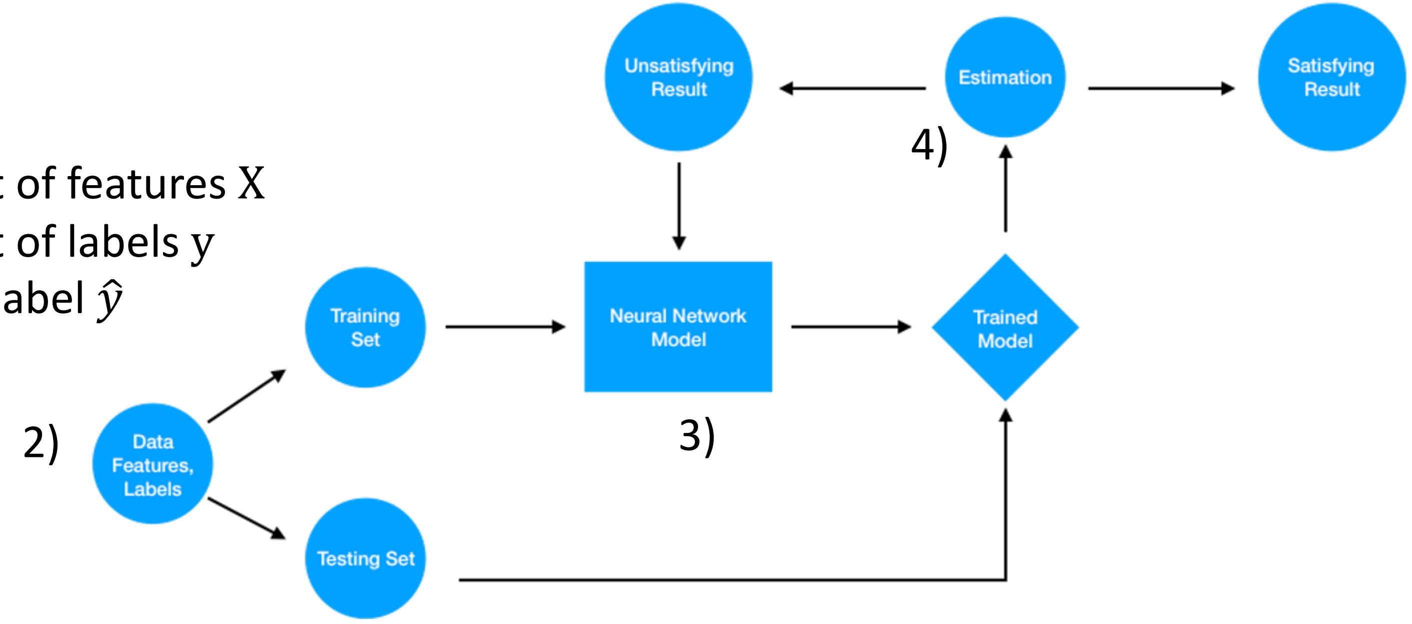
$\mathbf{O}$ (q output; n examples) $\qquad \hat{\mathbf{Y}}$ (q outputs; n examples)

$$= \begin{pmatrix} o_1^{(1)} & o_2^{(1)} & \cdots & o_q^{(1)} \\ o_1^{(2)} & o_2^{(2)} & \cdots & o_q^{(2)} \\ \vdots & \vdots & & \vdots \\ o_1^{(n)} & o_2^{(n)} & \cdots & o_q^{(n)} \end{pmatrix} \xrightarrow{\text{softmax}} \begin{pmatrix} \hat{y}_1^{(1)} & \hat{y}_2^{(1)} & \cdots & \hat{y}_q^{(1)} \\ \hat{y}_1^{(2)} & \hat{y}_2^{(2)} & \cdots & \hat{y}_q^{(2)} \\ \vdots & \vdots & & \vdots \\ \hat{y}_1^{(n)} & \hat{y}_2^{(n)} & \cdots & \hat{y}_q^{(n)} \end{pmatrix}$$

Outline

1) Difference between Linear, Logistic, and Softmax regression?
2) How to represent the labels? (One-hot-encoding)
3) How to estimate the outputs of our model as probability? (Network Architecture, Softmax Operation)
4) **How to measure the quality of our predicted probabilities? (loss function, maximum likelihood estimation, cross-entropy loss)**

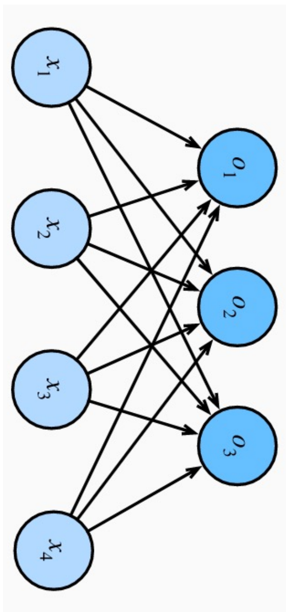
- training set of features X
- training set of labels y
- estimated label $\hat{y}$



Q4: How to measure the quality of our predicted probabilities?

- we need a loss function to measure the quality of our predicted probabilities. We will rely on maximum likelihood estimation

training set of features x



softmax $\longrightarrow$

estimated label $\hat{y}$

$\hat{y}_1$

$\hat{y}_2$

$\hat{y}_3$

$\hat{y}_4$

Cross Entropy

$$-\sum_{j=1}^{q} y_i \log \hat{y}_j$$

Training set of label y

y_1

y_2

y_3

y_4

If $x \in$ class 1

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

If $x \in$ class 2

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

If $x \in$ class 3

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

Cross Entropy Loss



	category ID	one-hot encoding label **y**		model prediction $\hat{\mathbf{y}}$	
spiral galaxy	1	y_1	0	0.02	P(spiral | image **x**)
elliptical galaxy	2	y_2	1	0.75	P(elliptical | image **x**)
merging pairs	3	y_3	0	0.01	P(merger | image **x**)
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
irregular	q	y_q	0	0.03	P(irregular | image **x**)

Likelihood of a training example $\quad P(\mathbf{y} \mid \mathbf{x}) = \prod_{j=1}^{q} \hat{y}_j^{\,y_j} = \prod_{j=1}^{q} \mathrm{pow}(\hat{y}_j, y_j)$

Likelihood of the entire dataset $\{ (\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), \ldots, (\mathbf{x}^{(n)}, \mathbf{y}^{(n)}) \}$ $\quad P(\mathbf{Y} \mid \mathbf{X}) = \prod_{i=1}^{n} P(\mathbf{y}^{(i)} \mid \mathbf{x}^{(i)})$

log-likelihood of the entire dataset $\quad \log P(\mathbf{Y} \mid \mathbf{X}) = \sum_{i=1}^{n} \log P(\mathbf{y}^{(i)} \mid \mathbf{x}^{(i)}) = \sum_{i=1}^{n} \sum_{j=1}^{q} y_j^{(i)} \log(\hat{y}_j^{(i)})$

maximize the log-likelihood ↔ minimize the cross entropy loss

cross entropy loss for a training example $\quad \ell(\mathbf{y}^{(i)}, \hat{\mathbf{y}}^{(i)}) = -\sum_{j=1}^{q} y_j^{(i)} \log(\hat{y}_j^{(i)})$

Summary

1) Logistic/Softmax regression for classification problem
2) To represent the labels: One-hot-encoding
3) To the outputs of our model as probability:
   Network Architecture, Softmax Operation, Vectorization
4) To measure the quality of our predicted probabilities:
   minimizing cross-entropy loss