

## بخش 1:

### 1) AI winter:

در اوایل دهه شصت میلادی بسیاری از پژوهشگران بر این باور بودند با توجه به دستاوردهای AI در آن زمان ، در زمان بسیار کمی رشد زیادی خواهد کرد به همین دلیل سرمایه گذاری های بسیاری در این حوزه صورت گرفت اما بعد از گذشت چند سال و حاصل نشدن نتیجه مطلوب سرمایه گذاران ناامید شدن این اولین زمستان هوش مصنوعی بود به طور کلی زمستان هوش مصنوعی زمانی است که علاقه و سرمایه گذاری روی هوش مصنوعی به کمترین حالت خود می رسد و سایر زمستان به شرح زیر اند:

دهه 70 میلادی، دهه 90 میلادی، 1998-1999

### 2) Backpropagation:

برای تنظیم وزن های شبکه در راستای درست نیاز به مینیموم کردن مقدار Loss function می باشد به اینصورت که از مقدار Loss function لایه اخر با استفاده از chain rule تاثیر پارامترهای لایه های قبل بر روی loss function را حساب کرده در نهایت شیب تغییرات loss function روی همه وزن ها محاسبه می شود ( gradient descent ) و برای محاسبه میزان جدید پارامتر ها تنها نیاز است درخلاف جهت شیب با یک مقدار معین حرکت کنیم.

### 3) Objective function :

همان Loss function است، مقداری است که نشان میدهد مقداری که شبکه بدست آورده با مقدار واقعی چقدر متفاوت است

### 4) kernel methods:

این روش به دسته بندی مقادیر بوسیله پیدا کردن مرزهای جدایی می پردازد از معروف ترین انها می توان به svm اشاره کرد

### 5) 4D tensors vs. 4-dimensional vector :

در 4-dimensional vector در یک محور (axis) چهار درایه دارد و 4d tensors در 4 محور به صورت دلخواه درایه دارد

### 6) Element-wise product vs. Tensor product:

در element wise عملیات ضرب در درایه های نظیر به نظیر انجام

می شود اما در tensor عملیات ضرب ماتریسی انجام می شود یعنی ضرب سطر و ستون های متناظر.

## بخش 2:

(۲۸)

سپام یا اسپم؟

	$F_1$	$F_2$	$F_3$
$\frac{6}{10}$ Spam	$P(F_1=1 Spam) = \frac{1}{6}$	$P(F_2=1 Spam) = \frac{1}{6}$	$P(F_3=0 Spam) = \frac{2}{6}$
	$P(F_1=0 Spam) = \frac{5}{6}$	$P(F_2=0 Spam) = \frac{5}{6}$	$P(F_3=1 Spam) = \frac{4}{6}$
$\frac{4}{10}$ Not-Spam	$P(F_1=1 Not-Spam) = 0$	$P(F_2=1 Not-Spam) = \frac{1}{4}$	$P(F_3=1 Not-Spam) = \frac{3}{4}$
	$P(F_1=0 Not-Spam) = 1$	$P(F_2=0 Not-Spam) = \frac{3}{4}$	$P(F_3=0 Not-Spam) = \frac{1}{4}$

$$x_1 = [110] = ? \text{ Spam}$$

$$P(\text{spam}) = P(F_1=1|Spam) \times P(F_2=1|Spam) \times P(F_3=0|Spam) \\ = \frac{1}{6} \times \frac{1}{6} \times \frac{2}{6} = \frac{1}{18} \approx 0.055$$

$$P(\text{not-spam}) = P(F_1=1|Not-Spam) \times P(F_2=1|Not-Spam) \times P(F_3=0|Not-Spam) \\ = 0 \times \frac{1}{4} \times \frac{1}{4} = 0$$

$$x_2 = [111] = ? \text{ Spam}$$

$$P(\text{spam}) = \frac{1}{6} \times \frac{1}{6} \times \frac{4}{6} = \frac{1}{18} \approx 0.055$$

$$P(\text{not-spam}) = 1 \times \frac{1}{4} \times \frac{3}{4} = \frac{3}{16} \approx 0.1875$$

### بخش 3:

5000 نمونه داریم که هر کدام عکس  $32 \times 32$  پیکسل هستند در فرمت rgb داریم  $32 \times 32 \times 3$  و در حالت کلی

(5000,32,32,3) و رنگ ما برابر با 4 می باشد تایپ دیتا ها unit-8

```
#data type
print((x_train.dtype, y_train.dtype, x_test.dtype, y_test.dtype))
[4] ✓ 0.2s
... (dtype('uint8'), dtype('uint8'), dtype('uint8'), dtype('uint8'))

#shape
print((x_train.shape, y_train.shape, x_test.shape, y_test.shape))
[5] ✓ 0.2s
... ((50000, 32, 32, 3), (50000, 1), (10000, 32, 32, 3), (10000, 1))

from numpy.linalg import matrix_rank
[6] ✓ 0.6s

#rank
print((matrix_rank(x_train), matrix_rank(y_train), matrix_rank(x_test), matrix_rank(y_test)))
[7] ✓ 5.4s
... (array([[3, 3, 3, ..., 3, 3, 3],
          [3, 3, 3, ..., 3, 3, 3],
          [3, 3, 3, ..., 3, 3, 3],
          ...,
          [3, 3, 3, ..., 3, 3, 3],
          [3, 3, 3, ..., 3, 3, 3],
          [3, 3, 3, ..., 3, 3, 3]], 1, array([[3, 3, 3, ..., 3, 3, 3],
          [3, 3, 3, ..., 3, 3, 3],
          [3, 3, 3, ..., 3, 3, 3],
          ...,
          [3, 3, 3, ..., 3, 3, 3],
          [3, 3, 3, ..., 3, 3, 3],
          [3, 3, 3, ..., 3, 3, 3]], 1))
```

### بخش 4:

ابتدا از داده های داده شده اطلاعات prior را کشف می کنیم ، از آنجا که تو توزیع داده های برای هر دسته یکسان است احتمال رخداد هر دسته  $1/3$  می شود داده ها را به نسبت 80 به 20 جدا می کنیم تا دیتا های train و test را استخراج کنیم و سپس واریانس و میانگین را به وسیله groupby پانداس برای هر کلاس در 4 فیچر sepal length (cm), sepal width (cm) , petal width (cm) ، petal length (cm) بدست آوردیم در نهایت برای هر داده تست احتمال شرطی هر کدام از فیچر ها را بدست آوردیم در نهایت در prior ضرب کرده وبا argmax کلاس با بیشترین احتمال را برای هر داده تست بدست آوردیم و دقت تقریبا 97 درصد شد.