

# Assignment\_zh4zn

February 18, 2020

## 0.1 The source of data:

I will be using the SUN20 dataset.

## 0.2 The decision problem:

In my project, for example, I want to train a computer to understand an image. To be more specific, by providing data and images as input, we can use computers to recognize objects, people, animals, places, etc. Some application of my model can be used by people who want to make friends, for example, users can upload an image or a photo to our model, then the model will classify and create a label for it. Moreover, it will match people who with the same label. In my classification project, computers or software are supposed to classify scenery from a branch of photos. Thus, the decision-maker is a program. The agent may confront some decisions. Some of them is, for example, which sort of scenery it is.

## 0.3 The decision make's action set $\mathbb{A}$ :

A vector with 20 entries. Each entry represents a score with respect to a given image, and  $\sum_{i=1}^{20} a_i = 1$ .

## 0.4 Sample space:

Training set is a dataset consisting of 60k training images belonging to 10 categories of garments. The images are rescaled to a size of 28x28 pixels

## 0.5 Data generating process:

The whole dataset was split into two parts: Training set and validation set. The validation set has 10k images. The classifier will take input vectors  $x$  of size  $1 \times 28 \times 28 = 784$ , which is the size of the images. The function representing the linear function is:

$$\hat{a} = \text{softmax}(Wx + b)$$

$$\hat{a}_i = \frac{\exp(Wx + b)}{\sum_{k=1}^{10} \exp(Wx + b)}$$

To be more specific,

$$\hat{a}_i = W_{i1}x_{i1} + W_{i2}x_{i2} + \dots + W_{ij}x_{ij} + b_i$$

,  $W$  and  $b$  can be initialized with random values.  $W_{ij} \in \mathbb{N}(\theta_1, \sigma_1), b \in \mathbb{N}(\theta_2, \sigma_2)$

## 0.6 The parameter space:

$\theta_1, \theta_2, \sigma_1, \sigma_2 \in \mathbb{R}X(0, \infty)$ , and  $P$  in  $[0,1]$

## 0.7 The decision-maker's prior beliefs:

$W$  and  $b$  are normally distributed. Based on previous experience, we can select  $W$  and  $b$  from  $\mathbb{N}(0, 0.01)$

The result of classification depends on the number of images. Training the model or program is time-consuming in that the image input should be large enough to produce classification results. If the image sets are small, the model cannot classify images as we expected. It may, for example, classify a mountain as a canyon. However, it is difficult to collect a large amount of images, which needs time and cutting-edge equipment. That is to say, the cost of making wrong decisions is extremely high. After making a wrong decision, researchers need to adjust the module, resample the dataset and change parameters. This is costs realized from poorer decisions. According to this example, stakes of decision is the cost when our model makes a wrong decision. We may lose money and time because of a poorer decision. But from the decision, we can collect ideas that can optimize our model, making our decision better next time.

## 0.8 The predictive tool:

From the predictive tools, we can learn about what will happen in the future based on historical data. In the case of images, our prediction model can potentially tell you the type, the label of a certain image. To generate a useful prediction, we need to know the process of classification. By comparing the result produced by our model, we can get feedback data. This is data you collect when the prediction machine is operating in real situations. Feedback data is often generated from a richer set of environments than training data. To some extent feedback data is more realistic than the data used for training. So, you can improve the accuracy of predictions further with continual training using feedback data.

## 0.9 The process for getting better information.

The decision-maker's beliefs about uncertain parameter values can be sharpen with Gradient Descent where we computer derivatives of loss function with respect to  $W$  and  $b$ . Then we update the parameters according to the following rules:

$$w_{ij} = w_{ij} - \lambda \frac{\partial \ell}{\partial w_{ij}}$$

$$b_i = b_i - \lambda \frac{\partial \ell}{\partial b_i}$$

## 0.10 Relevant payoffs:

The model will match 2 people in terms of meeting successful or unsuccessful

### 0.10.1 Loss function:

The loss function between the prediction and true label is:

$$\ell(y, \hat{y}) = - \sum_i y_i \log(\hat{y}_i)$$

Since the true label vector  $y$  only has one non-zero entry, we know this sum really has only one term that is non-zero so we will usually use this loss function simply stated as:

$$\ell(\text{label}, \hat{y}) = -\log(\hat{y}_{\text{label}})$$

.

### 0.11 The rule the decision-maker will use to choose a preferred action:

I will use  $\theta \in (0,1)$  to represent the potential matches of people. If  $\theta > 0.7$  we can make a prediction that the 2 people can be matched.