Extended

商家煜

郑铠奇

Deep Learning Foundation

Outline

1/ **Demo of the project**

2/ **Classification**

3/ **Application**

Outline

Back propagation neural network

- 把每一层权重更新的初始步长设置为 0
  - 输入到隐藏层的权重更新是 $\Delta w_{ij} = 0$
  - 隐藏层到输出层的权重更新是 $\Delta W_j = 0$
- 对训练数据当中的每一个点
  - 让它正向通过网络，计算输出 $\hat{y}$
  - 计算输出节点的误差梯度 $\delta^o = (y - \hat{y}) f'(z)$ 这里 $z = \sum_j W_j a_j$ 是输出节点的输入。
  - 误差传播到隐藏层 $\delta_j^h = \delta^o W_j f'(h_j)$
  - 更新权重步长：
    - $\Delta W_j = \Delta W_j + \delta^o a_j$
    - $\Delta w_{ij} = \Delta w_{ij} + \delta_j^h a_i$
- 更新权重，其中 η 是学习率，m 是数据点的数量：
  - $W_j = W_j + \eta \Delta W_j / m$
  - $w_{ij} = w_{ij} + \eta \Delta w_{ij} / m$
- 重复这个过程 e 代。

Outline

How to do classification using neural network

二元分类：

多元分类：

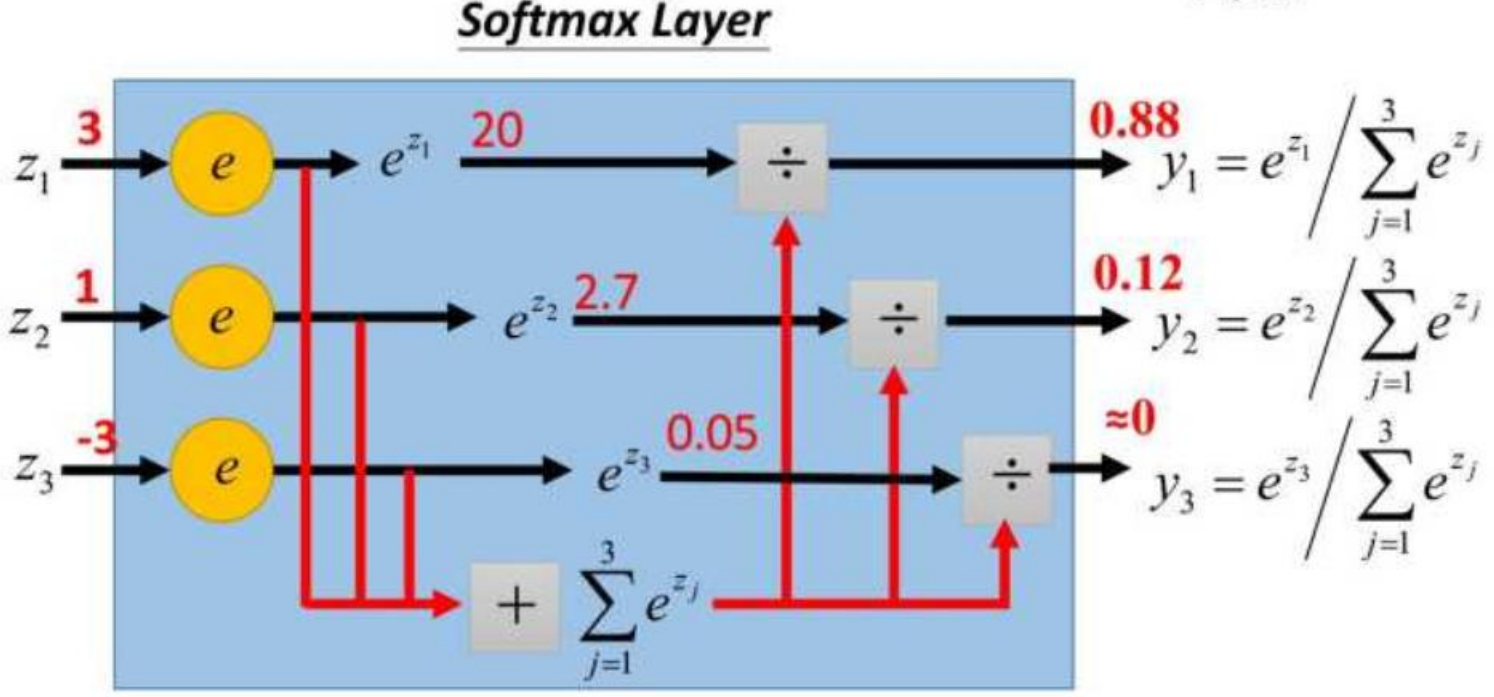
"**softmax** function is a generalization of the logistic function that maps a length-p vector of real values to a length-K vector of values"

$$\sigma(\mathbf{z})_j = \frac{e^{\mathbf{z}_j}}{\sum_{k=1}^{K} e^{\mathbf{z}_k}}$$

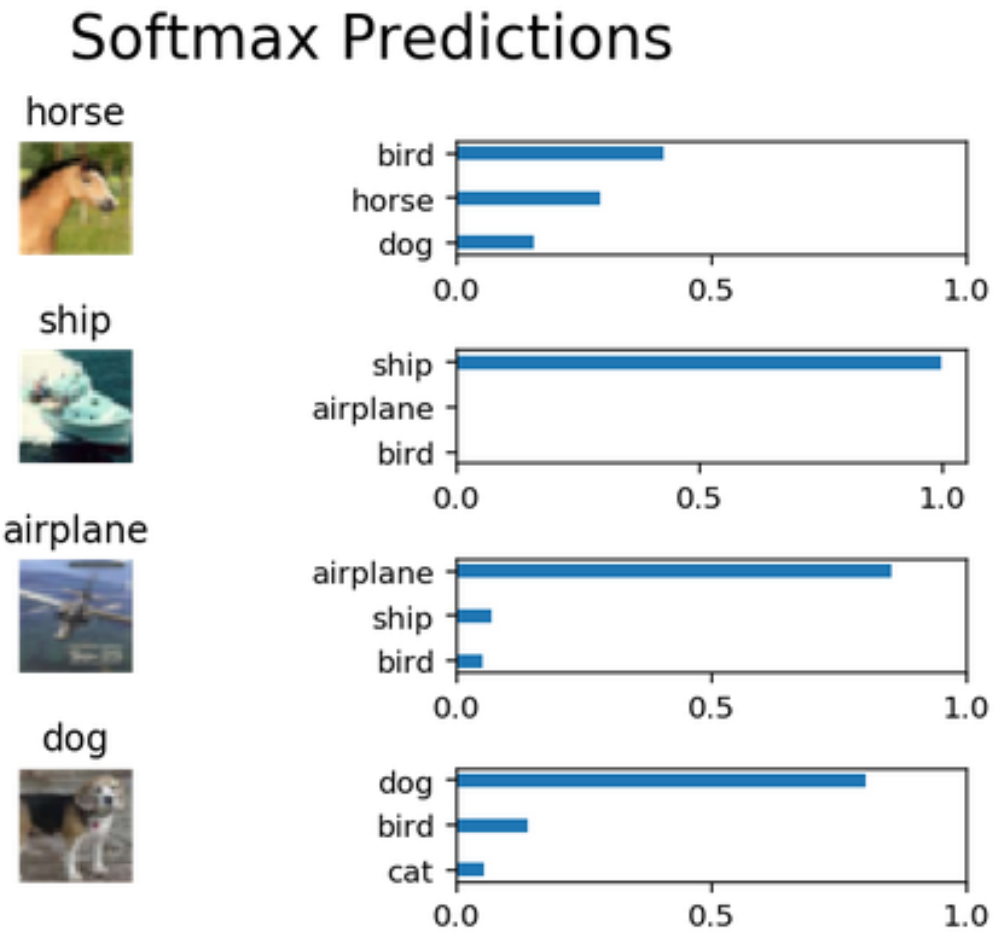
How to do classification using neural network

- Softmax layer as the output layer

Probability:
- $1 > y_i > 0$
- $\sum_i y_i = 1$

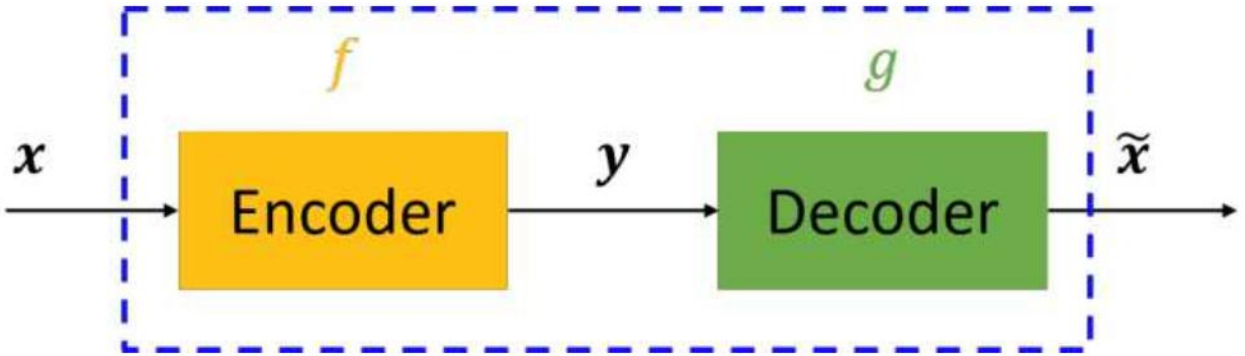


How to do classification using neural network



Outline

Autoencoder

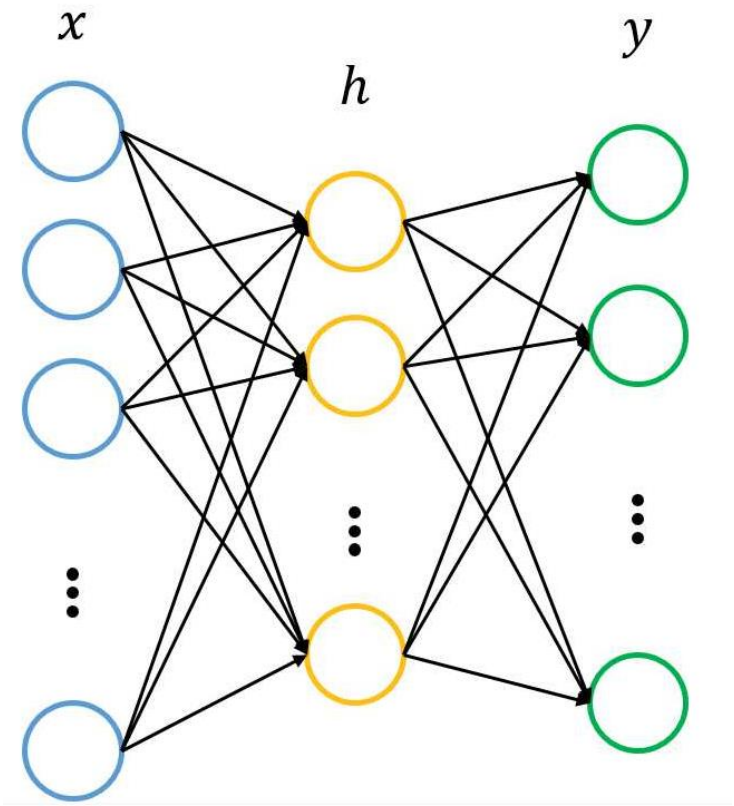


图中，虚线蓝色框内就是一个自编码器模型，它由编码器（Encoder）和解码器（Decoder）两部分组成，本质上都是对输入信号做某种变换。编码器将输入信号x变换成编码信号y，而解码器将编码y转换成输出信号 $\tilde{x}$ 。即

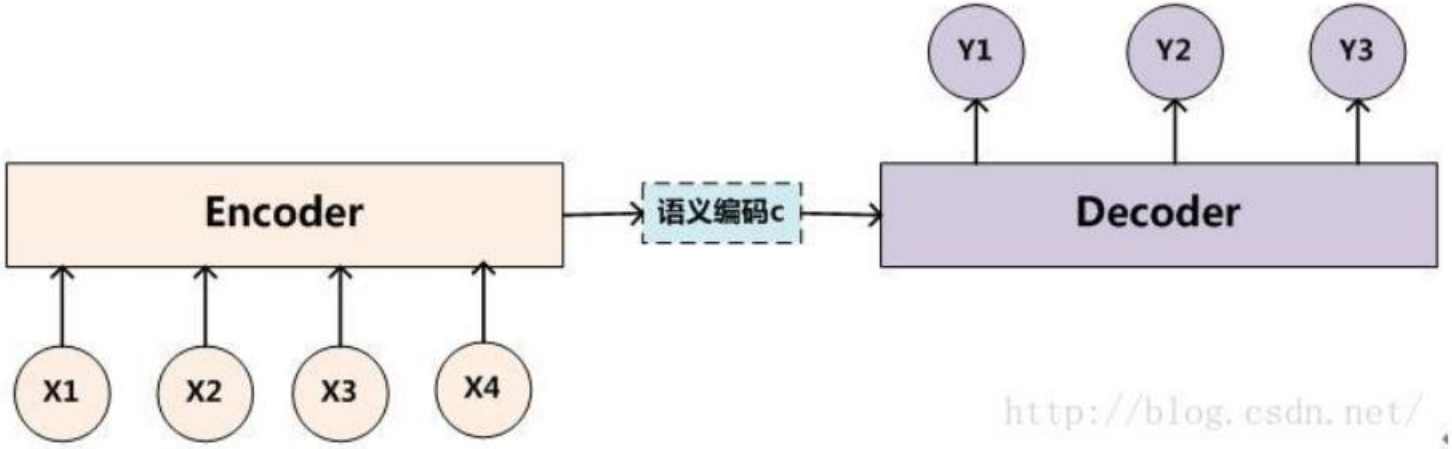
y=f(x)

$\tilde{x}$ =g(y)=g(f(x))

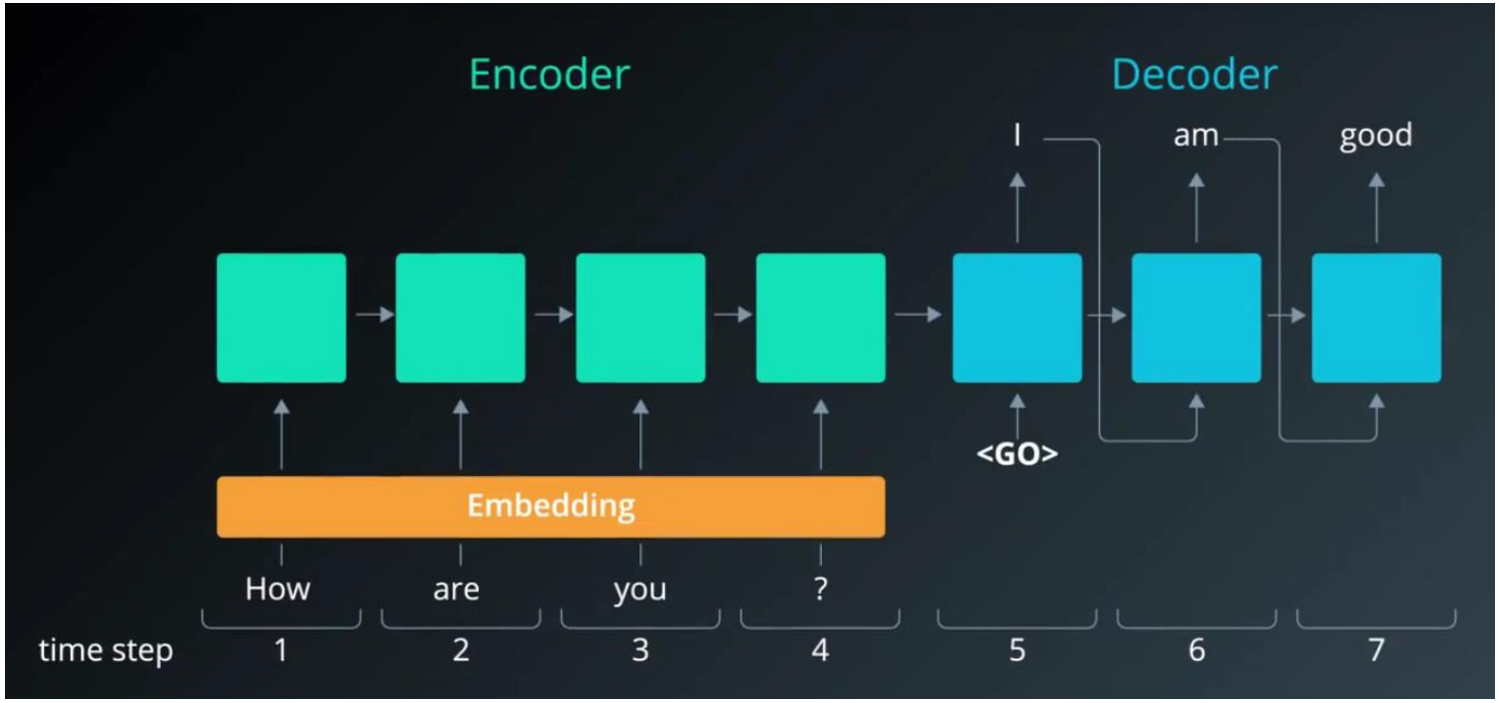
Autoencoder



Sequence-to-Sequence model



Sequence-to-Sequence model

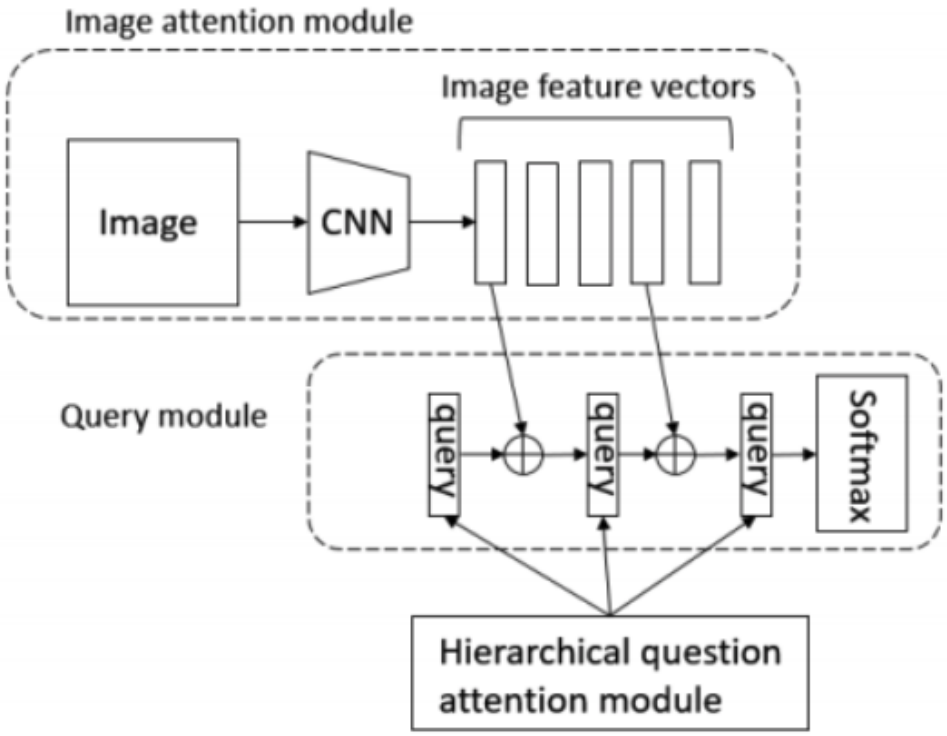


Visual Question Answering

- Aims to solve the holistic scene understanding problem by teaching computers to capture information

Fig.1 Visual Question Answer

3. Application



3. Application

注意事项

- 实验报告截止日期：
- **2017.12.06 晚 23:59:59 前**提交至 FTP 文件夹
- 提交文件：
  - 测试集结果：15******_wangxiaoming.txt 每一行对应的是测试样例的标签。
  - 实验报告：15******_wangxiaoming.pdf
  - 代码：15******_wangxiaoming.zip 如果代码分成多个文件，最好写份readme

THANKS