PyTorch

Logistic Regression

主讲人：龙良曲

Recap

- for continuous: $y = xw + b$
- for probability output: $y = \sigma(xw + b)$
  - $\sigma: sigmoid\ or\ logistic$

Binary Classification

- interpret network as $f : x \rightarrow p(y|x;\ \theta)$
- output $\in [0, 1]$
- which is exactly what *logistic function* comes in!

Goal v.s. Approach

- For regression:
  - Goal: $pred = y$
  - Approach: minimize $dist(pred, y)$
- For classification:
  - Goal: maximize benchmark, e.g. *accuracy*
  - Approach1: minimize $dist(p_\theta(y|x), p_r(y|x))$
  - Approach2: minimize $divergence(p_\theta(y|x), p_r(y|x))$

Q1. why not maximize accuracy?

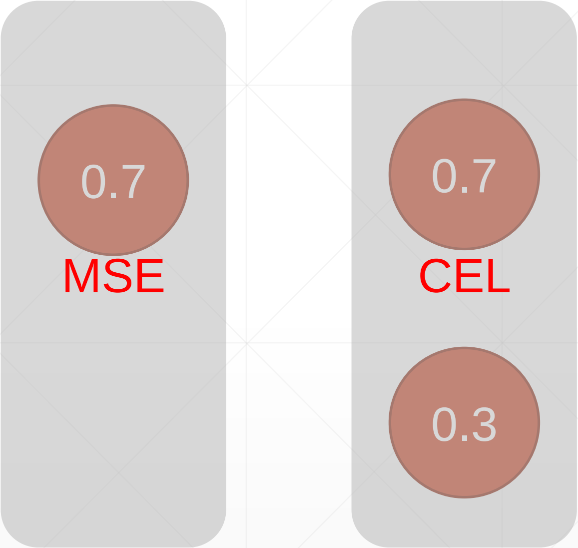
- $acc. = \frac{\sum I(pred_i == y_i)}{len(Y)}$

- issues 1. gradient $= 0$ if accuracy unchanged but weights changed

- issues 2. *gradient not continuous* since the number of correct is not continuous

Q2. why call logistic regression

- use sigmoid
- Controversial!
  - MSE => regression
  - Cross Entropy => classification



Binary Classification

- $f: x \rightarrow p(y = 1|x)$
  - if $p(y = 1|x) > 0.5$, predict as **1**
  - else predict as **0**

- minimize MSE

- confused?
  - http://www.fharrell.com/post/classification/

Multi-class classification

- $f: x \to p(y|x)$
  - $[p(y=0|x), p(y=1|x), \ldots, p(y=9|x)]$

- $p(y|x) \in [0,1]$

- $\sum_{i=0}^{9} p(y=i|x) = 1$

Softmax

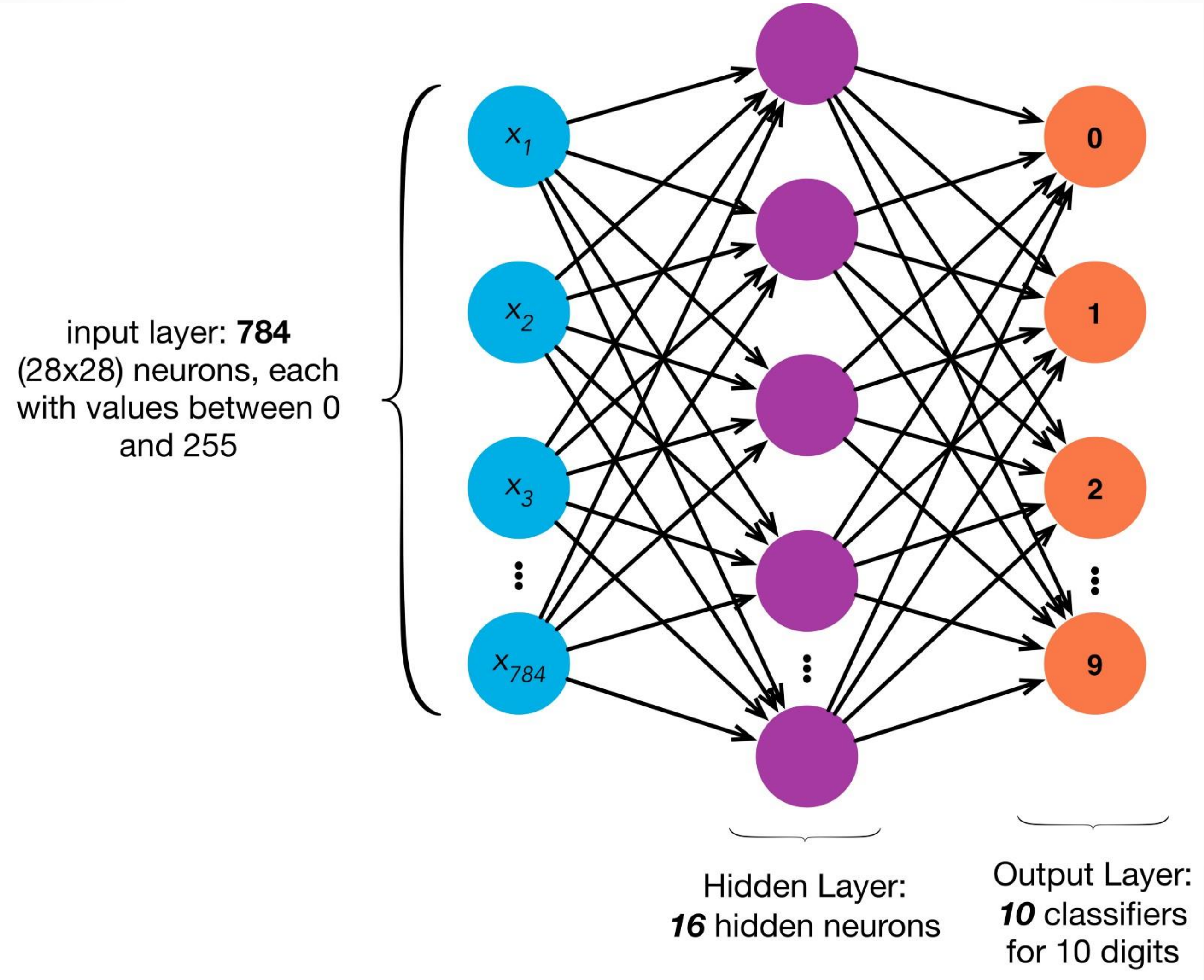
- $\sum_{i=0}^{9} p(y = i|x) = 1$

$$p_i = \frac{e^{a_i}}{\sum_{k=1}^{N} e^{a_k}}$$

Softmax

- enlarger the larger

$$p_i = \frac{e^{a_i}}{\sum_{k=1}^{N} e^{a_k}}$$



input layer: **784** (28x28) neurons, each with values between 0 and 255
x_1
x_2
x_3
x_{784}
0
1
2
9
Hidden Layer:
16 hidden neurons
Output Layer:
10 classifiers for 10 digits

下一课时

交叉熵

Thank You.