4.6 Loss and cost function

4.6.1 Regression

$$\underline{y} = \underline{f}(\underline{x}; \underline{\theta}) + \underline{z} \qquad \underline{z} \text{ noise}$$

Case A: $\underline{z} \sim N(\underline{0}, \sigma^2 \underline{\underline{I}})$; White Gaussian Noise

$$\underline{y} \sim N(\underline{f}(\underline{x}; \underline{\theta}), \sigma^2 \underline{\underline{I}})$$

$$q(\underline{y} \mid \underline{x}; \underline{\theta}) = \frac{1}{(2\pi)^{d/2} \sigma^d} \exp\left(-\frac{1}{2\sigma^2} \|\underline{y} - f(\underline{x}; \underline{\theta})\|^2\right)$$

$$l(\underline{x}, \underline{y}; \underline{\theta}) = -\ln q = \cancel{\text{const}} + \cancel{\frac{1}{2\sigma^2}} \|\underline{y} - f(\underline{x}; \underline{\theta})\|^2$$

$$L(\underline{\theta}) = \frac{1}{N} \sum_{n=1}^{N} l(\underline{x}(n), y(n); \underline{\theta})$$

$$= \frac{1}{N} \sum_{n=1}^{N} \|y(n) - f(x(n); \underline{\theta})\|^2$$

mean squared error (MSE) loss, L2 loss,

method of least squares

SASP

- If $f()$ is linear in $\underline{\theta}$ (eg in perceptron)
  $\Rightarrow$ closed form solution for $\underline{\theta}$
- $f()$ for DNN $\rightarrow$ non linear in $\underline{\theta}$
  $\Rightarrow$ need iterative method ch 4.7, Ch 5.

Case B $\underline{z} \sim N(\underline{0}, \underline{\underline{C}})$, coloured Gaussian noise

$$L(\underline{\theta}) = \frac{1}{N}\sum_{n=1}^{N}\left(\underline{y}(n) - f(\underline{x}(n);\underline{\theta})\right)^T C^{-1}\left(\underline{y}(n) - f(\underline{x}(n);\underline{\theta})\right)$$

Weighted mean squared error loss.

Rarely used in practice
- How to know $\underline{\underline{C}}$?
- $\underline{\underline{C}}^{-1}$ is computationally very expensive.

Slide 4-14

4-14.1
4.6.2 Classification

$\underline{x} \in \mathbb{R}^d$: input

$\underline{y} \in \{\underline{e}_1, \ldots \underline{e}_c\}$: class label for $\underline{x}$ in one-hot coding

$P_i = P(\underline{y} = \underline{e}_i \mid \underline{x})$: true posterior

Ch 3.2 : $P(\underline{y}|\underline{x})$
$= \prod_{i=1}^{c} P_i^{y_i}$: true PMF

But P_i is unknown

DNN
- output $f(\underline{x};\underline{\theta}) = [f_i(\underline{x};\underline{\theta})] \in \mathbb{R}^c$ as estimates for $[P_i]$
- i.e. $P(\underline{y}|\underline{x})$ approximated by $Q(\underline{y}|\underline{x};\underline{\theta})$

$$= \prod_{i=1}^{c} f_i(x_i;\underline{\theta})^{y_i}$$

In order to ensure

- $0 < f_i(\underline{x}; \underline{\theta}) < 1 \quad \forall i$
- $\sum_{i=1}^{c} f_i(\underline{x}; \underline{\theta}) = 1$

softmax is used in the output layer:

$$\underline{x}_L = f(\underline{x}; \underline{\theta}) = \text{softmax}(\underline{a}_L) \quad \text{see ch 4.4}$$

$$\Rightarrow \text{loss } l(\underline{x}, \underline{y}; \underline{\theta}) = -\ln Q$$
$$= -\sum_{i=1}^{c} y_i \ln f_i(\underline{x}; \underline{\theta})$$
$$= -\underline{y}^T \ln f(\underline{x}; \underline{\theta}) > 0$$

$$\text{Cost function } L(\underline{\theta}) = \frac{1}{N} \sum_{n=1}^{N} \left[-\underline{y}^T(n) \cdot \ln f(\underline{x}(n); \theta) \right]$$

Categorical cross entropy loss

Special case: Binary classification, $c = 2$

ch 4.4: softmax for $c = 2$

$\Updownarrow$

One o/p neuron with sigmoid activation f^n

$$f(\underline{x}; \underline{\theta}) = \sigma(a_L)$$

Let $y_1 = y \quad y_2 = 1 - y \; ; \; f_1 = f, \; f_2 = 1 - f$

$$\Rightarrow l(\underline{x}, \underline{y}; \underline{\theta}) = -\left[y \ln f(\underline{x}; \underline{\theta}) + (1-y) \ln(1 - f(\underline{x}; \underline{\theta})) \right]$$

$\uparrow$

binary Cross Entropy loss

slide 4.15, 4.16*

4.6.3 Semantic Image Segmentation

$\hat{=}$ pixelwise classification

Slide 4-18

Categorical cross entropy loss

$$l(\underline{\underline{X}}, \underline{\underline{Y}}; \underline{\theta}) = \underbrace{\sum_{h=1}^{H} \sum_{w=1}^{W} \underbrace{-\underline{y}_{hw}^{T} \ln \hat{\underline{y}}_{hw}(\underline{\underline{X}}; \underline{\theta})}_{\text{loss for 1 pixel}}}_{\text{loss for 1 image}}$$

Problem : imbalanced classes

eg. 90% backgroundpixels, 10% object pixels

$\Rightarrow$ loss cares more about the major class

$\Rightarrow$ loss cares less about the minor class

$\Rightarrow$ reduced segmentation accuracy for minor class

Solutions:

1. Weighted categorical CE loss
2. Region-based loss

Slide 4-19 $\rightarrow$ Let A be true posn of color pixels and B be the predicted true posn of color pixels

J and D : are not suitable for training

- ratio of integers - not differentiable
- $\underline{A}, \underline{B}, \underline{y}_{hw}$: hard labels $\in \{0, 1\}$

  $\hat{\underline{y}}_{hw}$: soft output / probabilities $\in \mathbb{R}$ $0 \leq \cdot \leq 1$

Hence use Soft J and D loss

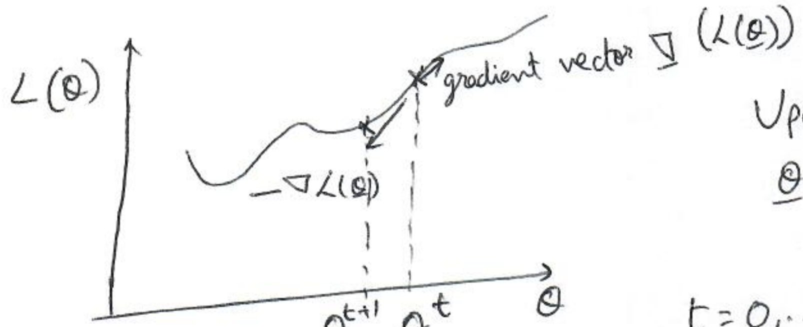
Slide 4-21

Q for Ilias → How is soft J and D differentiable ? They still are ratio b/w 2 numbers.

4.7 Training

- training set $D_{train} = \{\underline{x}(n), \underline{y}(n), 1 \leq n \leq N\}$
- cost function $L(\underline{\theta}) = \frac{1}{N} \sum_{n=1}^{N} l(\underline{x}(n), \underline{y}(n); \underline{\theta})$
- task : $\min_{\underline{\theta}} L(\underline{\theta})$
- optimizer: optimization algorithm to min $L(\underline{\theta})$

DL : gradient descent (and variants)
need only 1st order derivative of $L(\underline{\theta})$



Update rule:

$$\underline{\theta}^{t+1} = \underline{\theta}^{t} - \underbrace{\gamma^{t}}_{\text{step size}} \nabla L(\underline{\theta} | \underline{\theta} = \underline{\theta}^{t})$$

$t = 0, \dots$ iteration index

$\gamma^{t} > 0$: step size, learning rate

Calculation of $\underline{\nabla} L(\underline{\theta})$: non-trivial

slide 4-22

Chain rule of derivative:

$$\frac{d}{d\theta} f(g(\theta)) = \frac{df}{dg} \cdot \frac{dg}{d\theta}$$

Layer L:

$$\frac{\partial L(\underline{\theta})}{\partial w_{L,ij}} = \frac{\partial L(\underline{\theta})}{\partial \underline{x}_L} \cdot \frac{d\underline{x}_L}{d\underline{a}_L} \cdot \frac{d\underline{a}_L}{dw_{L,ij}}$$

$$= \underline{\underline{J}}_L(\underline{x}_L) \cdot \underline{\underline{J}}_{x_L}(\underline{a}_L)\, \underline{\underline{J}}_{\underline{a}_L}(w_{L,ij})$$

$$1 \times 1 \quad = \underbrace{1 \times M_L \quad M_L \times M_L}_{\underline{\underline{J}}_L(a_L) = 1 \times M_L} \quad M_L \times 1$$

Notation: $\underline{\underline{J}}_y(\underline{x}) = \frac{d\underline{y}}{d\underline{x}}$