Loss Functions & Output Layer

Yao Zhang

The guy is a populace

Mostly based on Thomas Hofmann's lecture in ETH

https://zhims.github.io/

Dec 1, 2019

Reminder: Notation

- Neural networks implements map $F : \mathbb{R}^n \to \mathbb{R}^m$
- Compositional structure layers:

$$F = F^L \circ F^{L-1} \circ \cdots F^1 \tag{1}$$

- Linear + activation function

$$F^l = \sigma^l \circ \overrightarrow{F}^l, \quad \overrightarrow{F}^l(x) = W^l x + b^l, \quad l = 1, ..., L \tag{2}$$

- F minus output layer non-linearity

$$\overline{F} = \overline{F}^L \circ F^{L-1} \circ \cdots \circ F^1 \tag{3}$$

Loss Function

For learning, we need to assess the goodness-of-fit of network.

Definition 1 (Loss function)

A loss (or cost) function is non-negative function

$$\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}, \quad (y, \nu) \mapsto \ell(y, \nu) \tag{4}$$

such that $\ell(y, y) = 0$ $(\forall y \in \mathcal{Y})$ and $\ell(y, \nu) > 0$ $(\forall \nu \neq y)$.

1. here: $\mathcal{Y}$: output space
2. general convention: y is the truth and ν predicted

Loss Function: Examples

Example 1 (Squared-error)

$$\mathcal{Y} = \mathbb{R}^m,\ \ell(y, \nu) = \frac{1}{2} \|y - \nu\|_2^2 = \frac{1}{2} \sum_{i=1}^{m} (y_i - \nu_i)^2 \tag{5}$$

Example 2 (Classification error)

$$\mathcal{Y} = [1 : m],\ \ell(y, \nu) = 1 - \delta_{y\nu} \tag{6}$$

with Kronecker delta:

$$\delta_{ab} = \begin{cases} 1, & \text{if } a = b \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

Expected Risk

Definition 2 (Expected Risk)

Assume inputs and outputs are governed by a distribution $p(x, y)$ over $\mathcal{X} \times \mathcal{Y}$, $\mathcal{X} \subseteq \mathbb{R}^n$, $\mathcal{Y} \subseteq \mathbb{R}^m$. The expected risk of F is given by

$$\mathcal{R}^\star(F) = E_{x,y}[\ell(y, F(x))] \tag{8}$$

1. as p is generally unknown, we cannot evaluate $\mathcal{R}^\star$ directly, but it serves as a point of reference in learning theory
2. $\mathcal{R}^\star$ is a functional (mapping functions to scalars)
3. parameterized functions $\{F_\theta : \theta \in \Theta\} \Rightarrow \mathcal{R}^\star(\theta) \triangleq \mathcal{R}^\star(F_\theta)$

Empirical Risk

Definition 3 (Empirical Risk)

Assume we have a random sample of N input-output pairs,

$$\mathcal{S}_N \triangleq \left\{ (x_i, y_i) \overset{i.i.d.}{\sim} p \ : \ 1, ..., N \right\}. \tag{9}$$

The empirical risk of F is defined as

$$\mathbb{R}(F, \mathcal{S}_N) = \frac{1}{N} \sum_{i=1}^{N} \ell\left(y_i, F\left(x_i\right)\right) \tag{10}$$

1. a.k.a. training risk = expected risk under the empirical distribution induced by the sample $\mathcal{S}_N$.

Empirical Risk Minimization

For a family $\mathcal{F} = \{F_\theta : \theta \in \Theta\}$ (e.f. neural network) and training data $\mathcal{S}_N$: find function with lowest empirical risk.

Definition 4 (Empirical risk minimization)

The empirical risk minimizer is defined as

$$\widehat{F}(\mathcal{S}_N) \in \arg\min_{F \in \mathcal{F}} \mathcal{R}(F, \mathcal{S}_N) \tag{11}$$

with the corresponding parameters $\widehat{\theta}(\mathcal{S}_N)$.

1. one may also add a regularizer $\Omega(F)$ or $\Omega(\theta)$ to the risk (more on that later)
2. finding $\widehat{F} \in \mathcal{F}$ amounts to solving on optimization problem

Probability Distributions as Outputs

It is often constructive to think of functions F as mappings from inputs to distribution $\mathcal{P}(\mathcal{Y})$ over outputs $y \in \mathcal{Y}$.

$$F : \mathbb{R}^n \to \mathbb{R}^m, \quad x \mapsto \nu, \quad \nu \overset{\textit{fixed}}{\mapsto} p(y, \nu) \in \mathcal{P}(\mathcal{Y}), \quad y \sim p(\cdot, \nu) \tag{12}$$

Each F effectively defines a conditional probability distribution (or conditional probability density function) via

$$p\left(y|x,\ F\right) = p\left(y,\ \nu = F\left(x\right)\right) \tag{13}$$

Example: Multivariate Normal Distribution

Example 3 (mean of a normal distribution)

$$p\left(y|x,\ F\right) = \left[\frac{1}{\sqrt{2\pi\gamma}}\right]^m e^{\left[-\frac{1}{2\gamma^2}\|y-F(x)\|^2\right]} \tag{14}$$

so that

$$-\log\ p\left(y|x,\ F\right) = mC\left(\gamma\right)\ +\ \frac{1}{2\gamma^2}\|y - F\left(x\right)\|^2 \tag{15}$$

which is equivalent to the squared error loss.

1. $F\left(x\right) = \nu$ and y live in same space $\left(\mathbb{R}^m\right)$

Generalized Linear Models

Definition 5 (Generalized linear model (simplified))

A generalized linear model over $y \in \mathcal{Y} \subseteq \mathbb{R}$ takes the form

$$E\left[y|x\right] = \sigma\left(w^T x\right). \tag{16}$$

where σ is invertible and σ^{-1} is called the link function.

1. can be extended to also predict variances or dispersions
2. can be extended to multidimensional outputs

Example: Logistic Regression

Example 4 (Logistic regression)

$\mathcal{Y} = \{0, 1\}, \mathcal{P} = [0, 1], \sigma = \frac{1}{1+e^{-x}}$, *then:*

$$E[y|x] = p(1|x) = \sigma\left(w^T x\right) = \frac{1}{1 + e^{-w^T x}} \tag{17}$$

Link function: logit

$$\sigma^{-1}(t) = \log\left(\frac{t}{1-t}\right), \quad t \in (0, 1) \tag{18}$$

Example: Multinomial Logistic Regression

Example 5

$\mathcal{Y} = [1:m], \mathcal{P}(\mathcal{Y})$*can be represented via soft-max*

$$p(y|x) = \frac{e^{z_y}}{\sum_{i=1}^{m} e^{z_i}}, \quad z \triangleq w_i^T x, \quad i = 1, ..., m \tag{19}$$

1. over-parametrized model: set $w_1 = 0$, s.t. $z_1 = 0$ (w.l.o.g)
2. generalizes (binary) logistic regression

Generalized Linear Units

In neural networks:

- non-linear functions replace linear functions
- output layer units implement inverse link function

Example 6 (Normal model)

Linear output layer

$$E[y|x] = \overline{F}(x) = W^L \left(F^{L-1} \circ \cdots \circ F^1 \right)(x) + b^L \tag{20}$$

Example 7 (Logistic model)

Sigmoid output layer

$$E[y|x] = \sigma\left(\overline{F}(x)\right) \tag{21}$$

Log-Likelihood

Use conditional probability distribution to define generalized loss between target value $y \in \mathcal{Y}$ and a distribution over $\mathcal{Y}$.

Definition 6 (Negative log-loss)

Canonical way of defining a generalized loss functions: negative of a log-likelihood function

$$\ell(y, \theta, x) = -\log\ p(y|x,\ \theta) \tag{22}$$

1. non-linearity of output layer is "absorbed" in loss function
2. i.e. ℓ depends on $\overline{F}$
3. provides a "template" for generalized loss/risk functions

Cross-Entropy Loss

Let us look at the (implied) risk function for the logistic function

Definition 7 (Cross-entropy Loss)

Use shorthand $z \triangleq \overline{F}(x) \in \mathbb{R}$ then the cross entropy loss over a binary response variable $y \in \{0, 1\}$ is defined as

$$\begin{aligned} -\log \; p(y|z) &= -\log \sigma\left((2y-1)z\right) \\ &= \zeta\left((1-2y)z\right) \end{aligned} \tag{23}$$

where $\zeta = log(1 + e^{(\cdot)})$ is the soft-plus function.

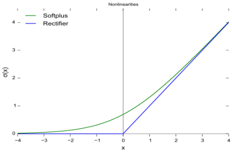

Figure 1: rectifier and softplus functions

Multinomial Log-Likelihood

Definition 8 (Multinomial cross-entropy loss)

Assume multinomial response variable $y \in [1:m]$. Use shorthand:

$$z \triangleq \overline{F}(x) \in \mathbb{R}^m \tag{24}$$

then with the soft-max activation function

$$\begin{aligned}\ell\left(y, \overline{F}(x)\right) &= -\log\ p\left(y \middle| \overline{F}(x)\right) = -\log\left[\frac{e^{z_y}}{\sum_{i=1}^{m} e^{z_i}}\right] \\ &= -z_y + \underbrace{\log \sum_{i=1}^{m} e^{z_i}}_{\log - partition} = \log\left[1 + \sum_{i \neq y} e^{(z_i - z_y)}\right]\end{aligned} \tag{25}$$

Thank you all of you! –Yao