

# Einführung in die Neuroinformatik

Tim Luchterhand, Paul Nykiel (Gruppe P)

13. Juni 2018

## 1 Cross Entropy

1. Man kann und darf keine Aussage über die „Schwere“ des Fehlers machen. Nur weil für die Eingabe  $x_1$  um „zwei Klassen“ vom Traininssignal abweicht, ist die Klassifikation nicht schlechter als für  $x_2$ , welches nur um eine Klasse abweicht.
2. Wurde erfolgreich gelesen ✓
3. (a)

$$\begin{aligned}H_A(B) &= \sum_x B(x) \cdot \log_2 \left( \frac{1}{A(x)} \right) \\&= \frac{1}{2} \log_2(8) + \frac{1}{4} \log_2(2) + \frac{1}{8} \log_2(4) + \frac{1}{8} \log_2(8) \\&= \frac{19}{8}\end{aligned}$$

(b)

$$\begin{aligned}H_B(A) &= \sum_x A(x) \cdot \log_2 \left( \frac{1}{B(x)} \right) \\&= \frac{1}{8} \log_2(2) + \frac{1}{2} \log_2(4) + \frac{1}{4} \log_2(8) + \frac{1}{8} \log_2(8) \\&= \frac{9}{4}\end{aligned}$$

(c)

$$\begin{aligned}H_B(B) &= \sum_x B(x) \cdot \log_2 \left( \frac{1}{B(x)} \right) \\&= \frac{1}{2} \log_2(2) + \frac{1}{4} \log_2(4) + 2 \cdot \frac{1}{8} \log_2(8) \\&= \frac{7}{4}\end{aligned}$$

(d)

$$\begin{aligned}D_A(B) &= H_A(B) - H(B) = \frac{19}{8} - \frac{7}{4} = \frac{5}{4} \\D_B(B) &= H_B(B) - H(B) = H(b) - H(B) = 0 \\D_B(A) &= H_B(A) - H(A) = \frac{9}{4} - \frac{7}{4} = \frac{1}{2}\end{aligned}$$

4. Es gilt:

$$D_Q(P) = H_Q(P) - H(P) \geq H(P) - H(P) = 0 \quad \forall P, Q \in X$$

denn  $H_Q(P) \geq H(P)$  und  $H(P) < 0 \quad \forall P$

Außerdem gilt:

$$D_Q(Q) = H_Q(Q) - H(Q) = H(Q) - H(Q) = 0 \quad \forall Q \in X$$

5. (a) Das Minimum muss auftreten, wenn  $C = B$

$$\Rightarrow t \cdot \frac{3}{4} \stackrel{!}{=} \frac{1}{2} \Rightarrow t = \frac{2}{3}$$

(b)

```
1 bob = [1/2 1/4 1/8 1/8];
2 deltaT = 0.01;
3 t = 0:deltaT:1;
4
5 KLdivergences = zeros(size(t));
6
7 for c = 1:length(t)
8     tmp = C(t(c));
9     for x = 1:length(bob)
10         KLdivergences(c) = KLdivergences(c) + bob(x) *
            log2(bob(x)/tmp(x));
11     end
12 end
13
14 plot(t, KLdivergences);
15 ylabel("D_C(B)");
16 xlabel("t");
17 title("Changing the code of Charles to fit to Bob's
        distribution");
18 hold on;
19 plot([2/3 2/3], [0 3]);
```

```

20 legend("KL-Divergence","Minimum");
21
22 print("plot.eps","-depsc");

1 function [v] = C(t)
2     v(1) = t*3/4;
3     v(2) = (1-t)*3/4;
4     v(3) = 1/8;
5     v(4) = 1/8;
6 end

```

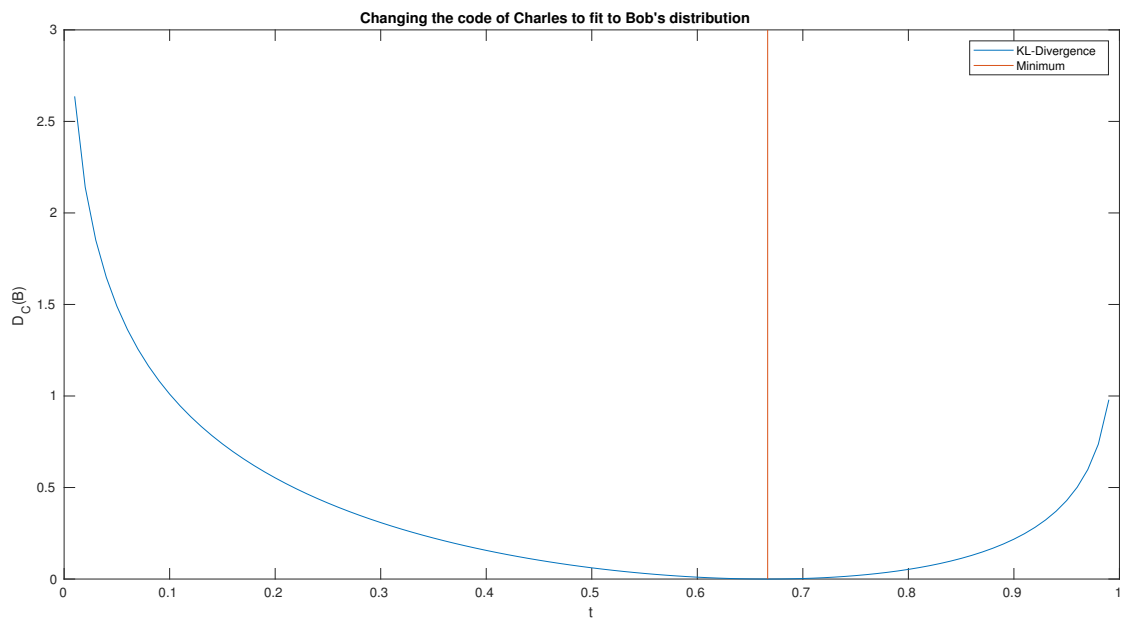


Abbildung 1: Ergebnis des Matlab Skripts

## 2 Cross Entropy als Kostenfunktion

1. (a)

$$y_i > 0 \forall i, \text{ da } e^{c \cdot u_i} > 0 \forall c, u_i \in \mathbb{R}$$

und

$$\begin{aligned}
 \sum_{i=1}^n y_i &= \sum_{i=1}^n \left( e^{c \cdot u_i} \cdot \frac{1}{\sum_{j=1}^n e^{c \cdot u_j}} \right) \\
 &= \frac{\sum_{i=1}^n e^{c \cdot u_i}}{\sum_{j=1}^n e^{c \cdot u_j}} = 1
 \end{aligned}$$

(b)

$$\begin{aligned}
y_1 &= \frac{e^{c \cdot u_1}}{e^{c \cdot u_1} + e^{c \cdot u_2} + e^{c \cdot u_3}} \\
&= \frac{e^{-c \cdot u_1}}{e^{-c \cdot u_1}} \cdot \frac{e^{c \cdot u_1}}{e^{c \cdot u_1} + e^{c \cdot u_2} + e^{c \cdot u_3}} \\
&= \frac{1}{1 + e^{c \cdot (u_2 - u_1)} + e^{c \cdot (u_3 - u_1)}}
\end{aligned}$$

(c) Fall  $u_1 > u_2 > u_3$ :

$$\lim_{c \rightarrow \infty} y_1 = 1$$

Fall  $u_2 > u_1 > u_3$ :

$$\lim_{c \rightarrow \infty} y_1 = 0$$

Fall  $u_2 > u_3 > u_1$ :

$$\lim_{c \rightarrow \infty} y_1 = 0$$

(d) Für  $c \rightarrow \infty$  konvergiert die Netzwerkausgabe gegen einen der Trainingsvektoren. D.h.  $y_i = 0 \ \forall i \neq k \wedge y_k = 1$

(e) Fall  $c > 0$ :

Alle  $y_i$  werden gegenüber dem  $y_k$ , für welches  $u_k > u_i \ \forall i \neq k$  gilt, gedämpft.

Fall  $c < 0$ :

Alle  $y_i$  werden gegenüber dem  $y_k$ , für welches  $u_k < u_i \ \forall i \neq k$  gilt, gedämpft.

Fall  $c = 0$ :

Alle  $y_i$  haben den gleich Wert  $y_i = \frac{1}{n}$

2.

$$E = -t_1 \log(y_1(u_1(w_1), u_2(w_2))) - t_2 \log(y_2(u_1(w_1), u_2(w_2)))$$

(a)

$$\begin{aligned}
\frac{\partial E}{\partial y_1} &= -t_1 \cdot \frac{1}{y_1} \\
\frac{\partial E}{\partial y_2} &= -t_2 \cdot \frac{1}{y_2}
\end{aligned}$$

(b)

$$\begin{aligned}\frac{\partial y_1}{\partial u_2} &= \frac{\partial}{\partial u_2} \left( \frac{e^{u_1}}{e^{u_1} + e^{u_2}} \right) = e^{u_1} \left( -\frac{e^{u_2}}{(e^{u_1} + e^{u_2})^2} \right) \\ &= -\frac{e^{u_1}}{e^{u_1} + e^{u_2}} \cdot \frac{e^{u_2}}{e^{u_1} + e^{u_2}} \\ &= -y_1 \cdot y_2 \\ \frac{\partial y_2}{\partial u_2} &= y_2 + e^{u_2} \cdot \left( -\frac{e^{u_2}}{(e^{u_1} + e^{u_2})^2} \right) \\ &= y_2 + \frac{e^{u_2}}{e^{u_1} + e^{u_2}} \cdot \left( -\frac{e^{u_2}}{e^{u_1} + e^{u_2}} \right) \\ &= y_2 + y_2(-y_2) = y_2(1 - y_2)\end{aligned}$$

(c)

$$\begin{aligned}u_2 &= w_2 x + b \\ \Rightarrow \frac{\partial u_2}{\partial w_2} &= x\end{aligned}$$

(d)

$$\begin{aligned}\frac{\partial E}{\partial w_2} &= \frac{\partial E}{\partial y_1} \frac{\partial y_1}{\partial u_2} \frac{\partial u_2}{\partial w_2} + \frac{\partial E}{\partial y_2} \frac{\partial y_2}{\partial u_2} \frac{\partial u_2}{\partial w_2} \\ &= \frac{-t_1}{y_1} \cdot (-y_1 y_2) \cdot x + \frac{-t_2}{y_2} \cdot y_2(1 - y_2)x \\ &= x \cdot (t_1 y_2 - t_2(1 - y_2)) \\ &= x \cdot (y_2 \cdot (t_1 + t_2) - t_2) = x \cdot (y_2 - t_2)\end{aligned}$$

(e) Im Gegensatz zur Quadratischen-Fehlerfunktion ist diese Fehlerfunktion nicht quadratisch. Außerdem wird die Netzwerkeingabe berücksichtigt.