Blatt11

November 20, 2020

1 Aufgabe 22

a) Was beschreibt die Lossfunktion und wofür wird sie benötigt?

Die Verlustfunktion misst, "wie gut" das neuronale Netzwerk in Bezug auf diese Trainingsdaten und die erwarteten Antworten ist.

Sie wird zur Berechnung des Fehlers zwischen realen und erhaltenen Parametern verwendet.

Schlechte Klassifikation → hohe Kosten

Gute Klassifikation → niedrige Kosten

Eine gebräuchliche Kostenfunktion ist die Kreuzentropie:

$$H(p,q) = -\sum_k p(k) \log q(k) \tag{1}$$

($p(x) \hat{=}$ wahre Wahrscheinlichkeitsdichte, $q(x) \hat{=}$ geschätzte Wahrscheinlichkeitsdichte)

Liefert kleinere Werte je ähnlicher die wahre Wahrscheinlichkeitsdichte zu der geschätzten Wahrscheinlichkeitsdichte ist

b) Wie kann die Lossfunktion minimiert werden?

Die Lossfunction kann minimiert werden, indem man der Richtung des Gradienten in jedem Schritt folgt

c) Welche Funktion haben die Aktivierungsfunktionen bzw. welches Problem wird durch diese ge-löst? Nennen Sie drei gängige Aktivierungsfunktionen.

Die Aktivierungsfunktion stellt den Zusammenhang zwischen dem Netzinput und dem Aktivitätslevel eines Neurons dar.

- sigmoid-Funktion:$\sigma(x) = \frac{1}{1+\exp -x}$
- Tangens hyperbolicus :$\tanh(x)$
- Rectified Linear Unit: $max(0, x)$

Die Aktivierungsfunktion ist von großer Bedeutung, da sie dabei hilft, das wirklich komplizierte und nichtlineare komplexe funktionale Mapping zwischen den Eingangsdaten und den abhängigen Ergebnissen zu lernen und zu verstehen.

d) Was ist ein Neuron?

Ein Neuron bildet die Basis eines neuronalen Netzwerkes und ist einem Neuron in der Biologie nachempfunden. Sie können Informationen von außen oder von anderen Neuronen aufnehmen und modifiziert an andere Neuronen weiterleiten oder als Endergebnis ausgeben

e) Nennen Sie drei Anwendungsbeispiele für Neuronale Netze und beschreiben Sie kurz warum siefür diese Beispiele besonders geeignet sind.

Anwendungsbeispiele sind zum Beispiel:

1. Gesichts/Bilderkennung
2. Texterkennung
3. Spracherkennung

Diese Beispiele eignen sich besonders, da bei denen nur geringes systematisches Lösungswissen vorliegt und eine große Menge von teils unpräzisen Eingabeinformationen zu einem konkreten Ergebnis verarbeitet werden müssen.

2 Aufgabe 23

Aufgabenteile a bis c sind handschriftlich im PDF angefügt

Aufgabenteil d:

[1]:
```
import pandas as pd
import numpy as np
```

[20]:
```
#einlesen

p0 = pd.read_hdf("populationen.hdf5", key="P_0")
p1 = pd.read_hdf("populationen.hdf5", key="P_1")
p0 = p0.assign(label=0)
p1 = p1.assign(label=1)
p = pd.concat([p0, p1])
```

```
             x         y  label
0     0.926612  4.717092      0
1    -3.953953  1.274478      0
2    -7.161693 -0.984415      0
3    -0.956840  1.115828      0
4    -0.046090  2.083444      0
…         …         …    …
9995  3.604288  2.839583      1
9996  0.855397 -0.963416      1
9997  7.416902  4.290466      1
9998  9.685447  7.290335      1
9999  9.195642  6.324578      1

[20000 rows x 3 columns]
```

[15]:
```
W = np.random.random((2,2))
b = np.random.random(2)

rate = 0.5
epochen = 100
```

[16]:
```
def softmax(f):
    return np.exp(f)/np.sum(np.exp(f))
```

[26]:
```
for i in range(epochen):
    f = np.matmul(p[["x", "y"]],W)+ b.reshape(1,2)

    #Gradienten bestimmen
    # irgenwie sind wir verwirrt von den ganzen ableitungen
    dW =
    db =

    #Parameterupdate
    W = W - rate * dw
    b = b - rate * db
```

[ ]:
```

```

Blatt 11 Aufgabe 23

a)

x_i : Spaltenvektor $(M \times 1)$

c : Kostenfunktion (1)

W : Gewichtungsmatrix $(K \times M)$

b : Biasvektor $(K \times 1)$

$\nabla_W \hat{C}$: $K \times M$ $\quad \frac{\partial f_{ki}}{\partial W}$: $K \times M$ $\quad \frac{\partial f_i}{\partial b}$: $K \times 1$

$\nabla_{f_i} \hat{C}$: $K \times 1$

b) $\nabla_{f_a} C(f) = \frac{1}{m} \sum_m \nabla_{f_a} \hat{C}(f_i)$

$$\nabla_{f_a} \hat{C}(f_i) = -\nabla_{f_a} \sum_k \mathbb{1}(y_i = k) \log\left(\frac{e^{f_{ki}}}{\sum_j e^{f_{ji}}}\right) = -\nabla_{f_a} \log\left(\frac{e^{f_{y_i i}}}{\sum_j e^{f_{ji}}}\right)$$

$$= \frac{-\sum_j e^{f_{ji}}}{e^{f_{y_i i}}} \left(\frac{(\nabla_{f_a} e^{f_{y_i i}}) \cdot \sum_j e^{f_{ji}} - e^{f_{y_i i}} \left(\nabla_{f_a} \sum_j e^{f_{ji}}\right)}{\left(\sum_j e^{f_{ji}}\right)^2}\right)$$

[...] das kann man bestimmt umformen zu

$$= \frac{e^{f_{a,i}}}{\sum_j e^{f_{a,i}}} - \mathbb{1}(y_i = a)$$

$$\Rightarrow \quad \nabla_{f_a} C(f) = \frac{1}{m} \sum_m \left[\frac{e^{f_{a,i}}}{\sum_j e^{f_{a,i}}} - \mathbb{1}(y_i = a)\right]$$

c)

$$\frac{\partial f_{ki}}{\partial W} = \frac{\partial (W_k x_i + b_k)}{\partial W} = \begin{pmatrix} 0 & \cdots & 0 \\ \vdots & & \vdots \\ x_{i,1} & \cdots & x_{i,M} \\ \vdots & & \vdots \\ 0 & & 0 \end{pmatrix} \rightarrow k\text{-te Zeile}$$

$$\frac{\partial f_{ki}}{\partial b} = \frac{\partial (W_k x_i + b_k)}{\partial b} = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \rightarrow k\text{-te Zeile}$$