Verlustfunktion

Die Zielvorgabe fürs Training

INNOVET KI B^3
Künstliche Intelligenz in die berufliche Bildung bringen

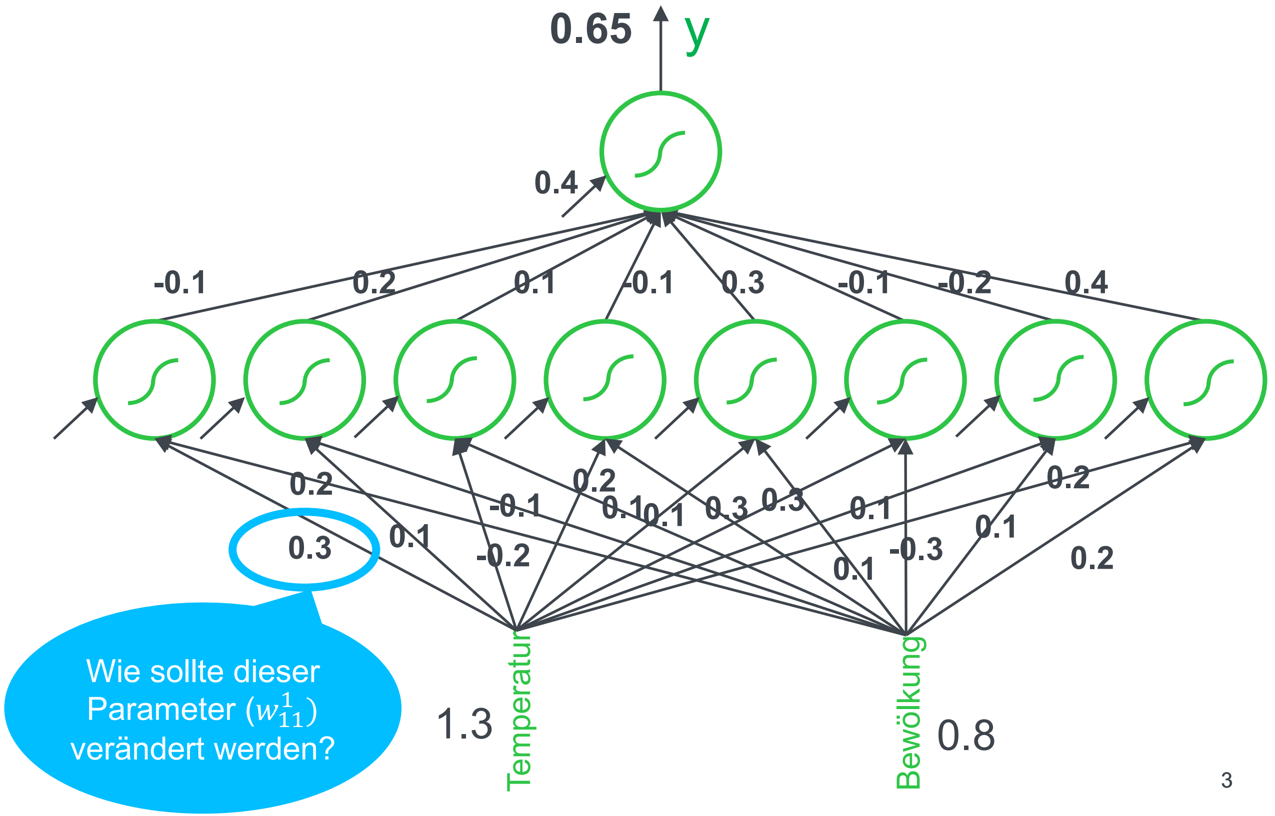
GEFÖRDERT VOM Bundesministerium für Bildung und Forschung
Bundesinstitut für Berufsbildung

Gefördert als InnoVET-Projekt aus Mitteln des Bundesministeriums für Bildung und Forschung.

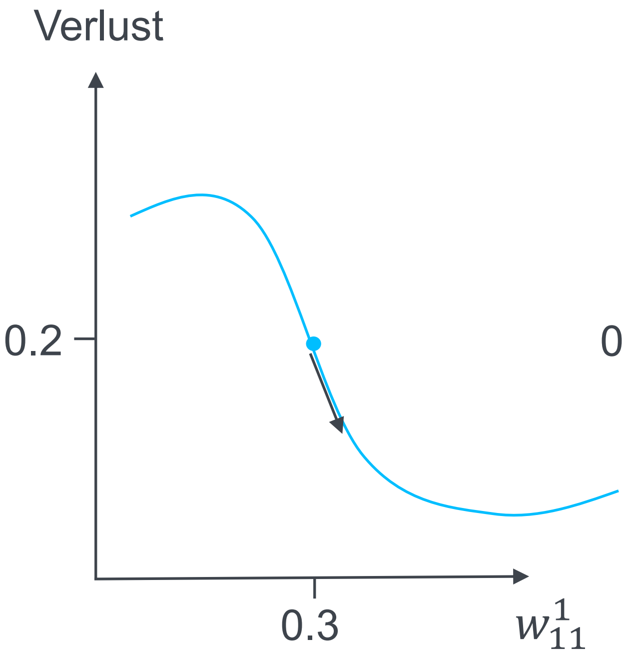
Rückblick: Training neuronaler Netze

Folie aus: Training von neuronalen Netzen

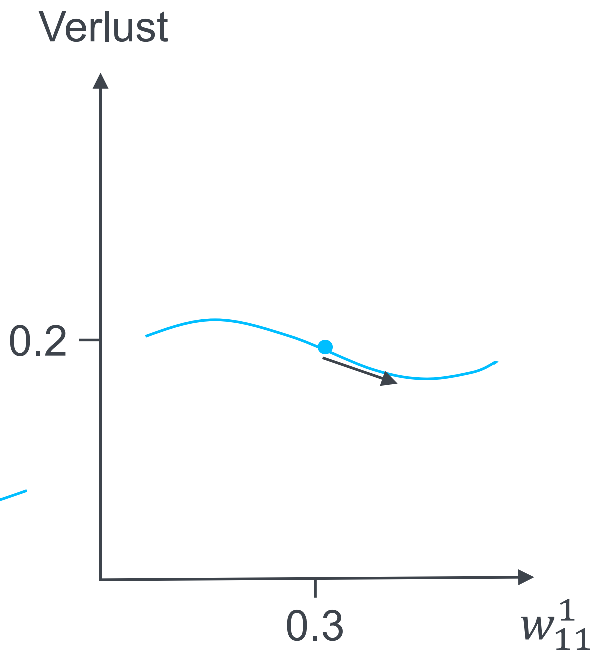
- Mit diesen Parametern:
  - Bei Input 1.3, 0.8
  - Ergebnis 0.65
- Abgleich mit korrektem Ergebnis:
  - Korrektes Ergebnis 0.45
  - „Verlust“ (Fehler): 0.2
- Gesucht:
  - Bessere Parameter
  - D.h. mit kleinerem Fehler



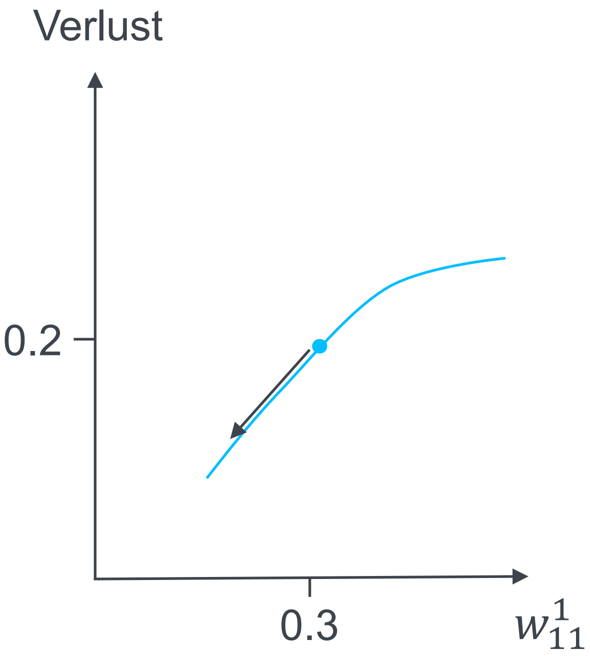
Folie aus: Training von neuronalen Netzen



Steigung stark negativ



Steigung schwach negativ



Steigung positiv

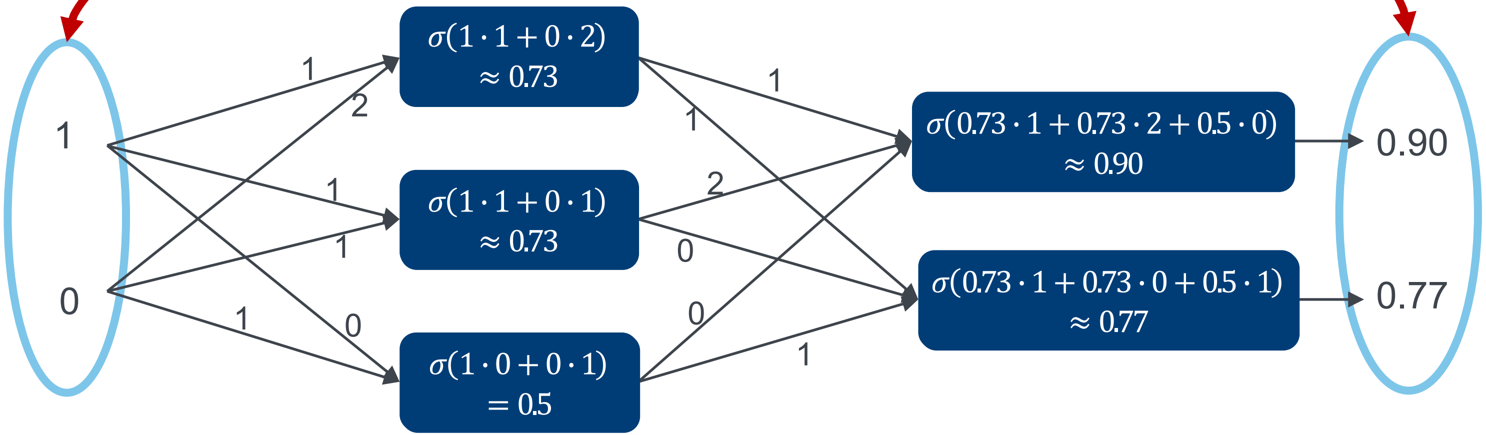
Verlustfunktion bei Regressionsproblemen

Beispiel – Netzwerk* für Regression

- Trainingsdatenpunkt:

Input	Ergebnis
1	5
0	-3

Ziel: Fehler möglichst klein!



Einzelne Fehler

$5 - 0.90 = 4.10$

$-3 - 0.77 = -3.77$

*der Einfachheit halber ohne Bias-Parameter

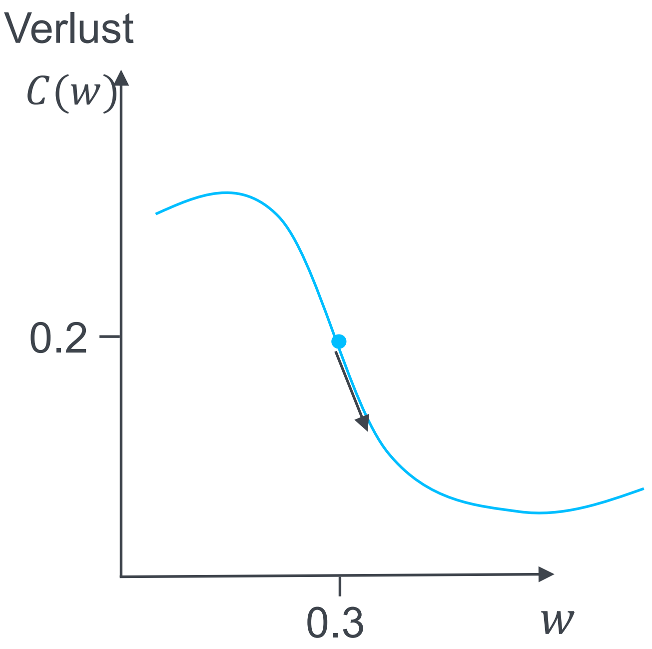
Verlust bei Regression

- Idee: wie bei Evaluationsmaßen für Regression
- Quadriere Einzelfehler, bilde Mittelwert
- Mean Squared Error, MSE
- Vorteile:
  - Positive und negative Fehler heben sich nicht gegenseitig auf
  - Größere Fehler werden stärker bestraft

Einzelne Fehler	MSE
$5 - 0.90$ $= 4.10$ $-3 - 0.77$ $= -3.77$	$\frac{1}{2} \cdot 4.10^2$ $+\frac{1}{2} \cdot (-3.77)^2$
Summe 0.33	Summe 15.5

Vom Fehler zur Verlustfunktion

- Beim Training interessant:
  - Wie verändert sich der Verlust, wenn die Gewichte verändert werden?
  - Verlust in Abhängigkeit von den Parametern des Netzwerks
  - Verlust als Funktion $C(w_{11}^1, w_{12}^1, \ldots, b_1^1, b_2^1, \ldots)$
- Für jeden Parameter $w_{11}^1, w_{12}^1, \ldots, b_1^1, b_2^1, \ldots$:
  - Berechne den Gradienten nur für diesen Parameter w

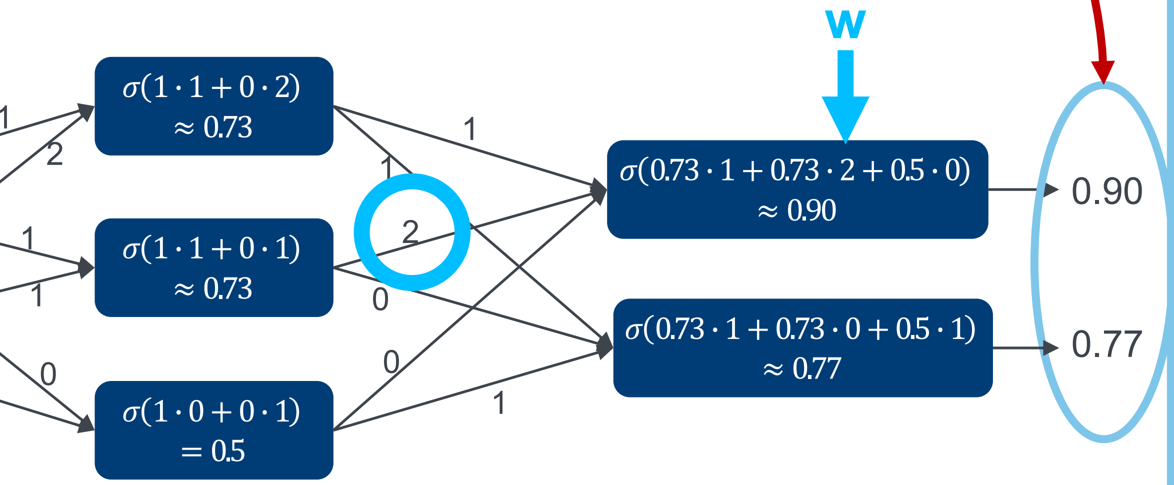


INNOVET KI B3

Beispiel-Exkurs: Verlustfunktion für ein Gewicht

- Trainingsdatenpunkt:

Input	Ergebnis
1	5
0	-3



w

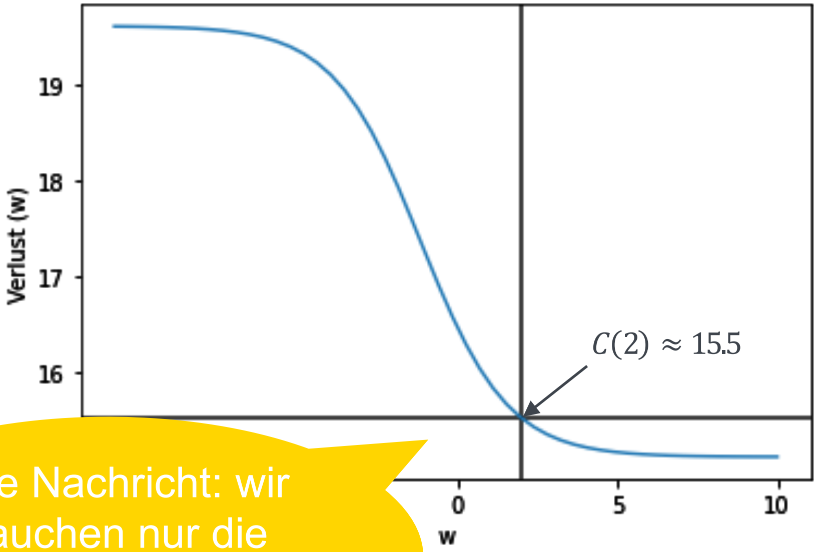
$$\frac{1}{2} \cdot \left(5 - \sigma(0.73 \cdot 1 + 0.73 \cdot 2 + 0.5 \cdot 0)\right)^2$$

$$+$$

$$\frac{1}{2} \cdot \left(-3 - \sigma(0.73 \cdot 1 + 0.73 \cdot 0 + 0.5 \cdot 1)\right)^2$$

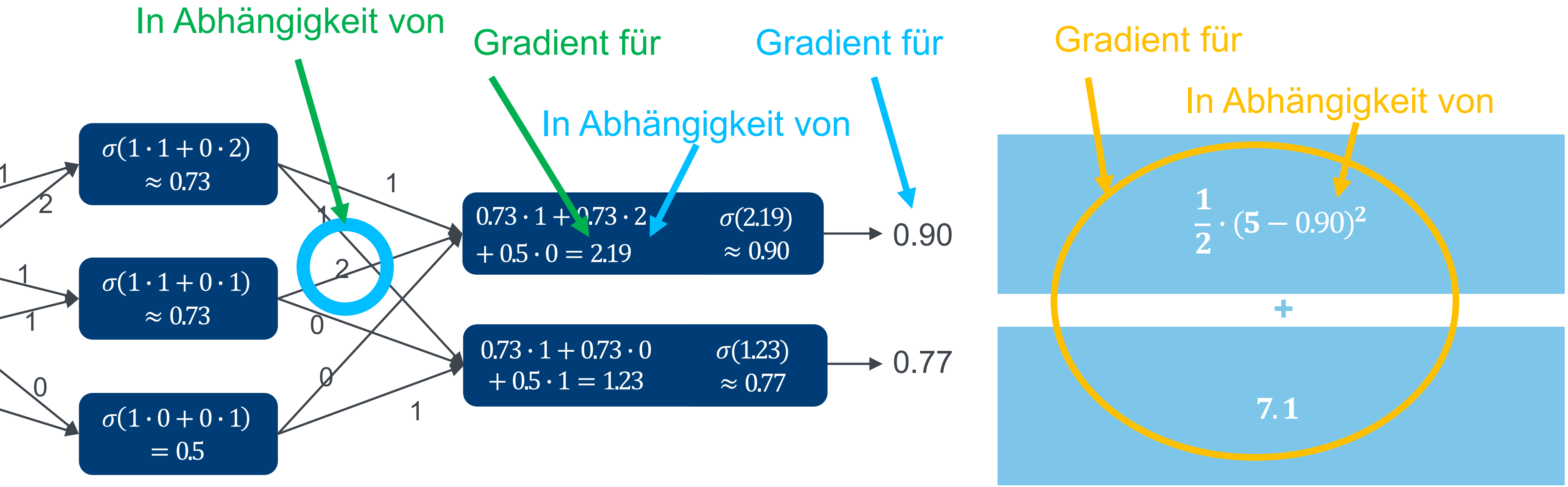
Beispiel-Exkurs: Verlustfunktion für ein Gewicht

- $C(w)$
- Verlust bei $w = 2$:
  - $C(2) \approx 15.5$ ✓
- Verlust wird geringer, wenn w größer wird (Gradient negativ)



Gute Nachricht: wir brauchen nur die Steigung, d.h. den Gradienten!

Beispiel: Gradienten für ein Gewicht schrittweise berechnen



$$(-5 + 0.90) \cdot \sigma(2.19) \cdot (1 - \sigma(2.19)) \cdot 0.73 \approx -0.27$$

Wann berechnet man die Verlustfunktion?

- Beim Training, aber nur indirekt
- Zunächst Berechnung des Verlusts bei den aktuellen Parametern
- Dann Berechnung der **Gradienten** der Verlustfunktion für jeden Parameter
  - Berechnung oft einfacher als die Berechnung der Funktion selbst
  - Wird bei Back Propagation Schritt für Schritt berechnet

Bei Regressionsproblemen in neuronalen Netzen wird meist das mittlere Fehlerquadrat (Mean squared error, MSE) als Verlustfunktion verwendet.

Man berechnet dann also zur Anpassung der Gewichte beim Training für jedes Gewicht den Gradienten der MSE-Funktion.

INNOVET KI B³

Multiple Choice: Verlustfunktionen bei neuronalen Netzen

Drag the Words: MSE als Verlustfunktion

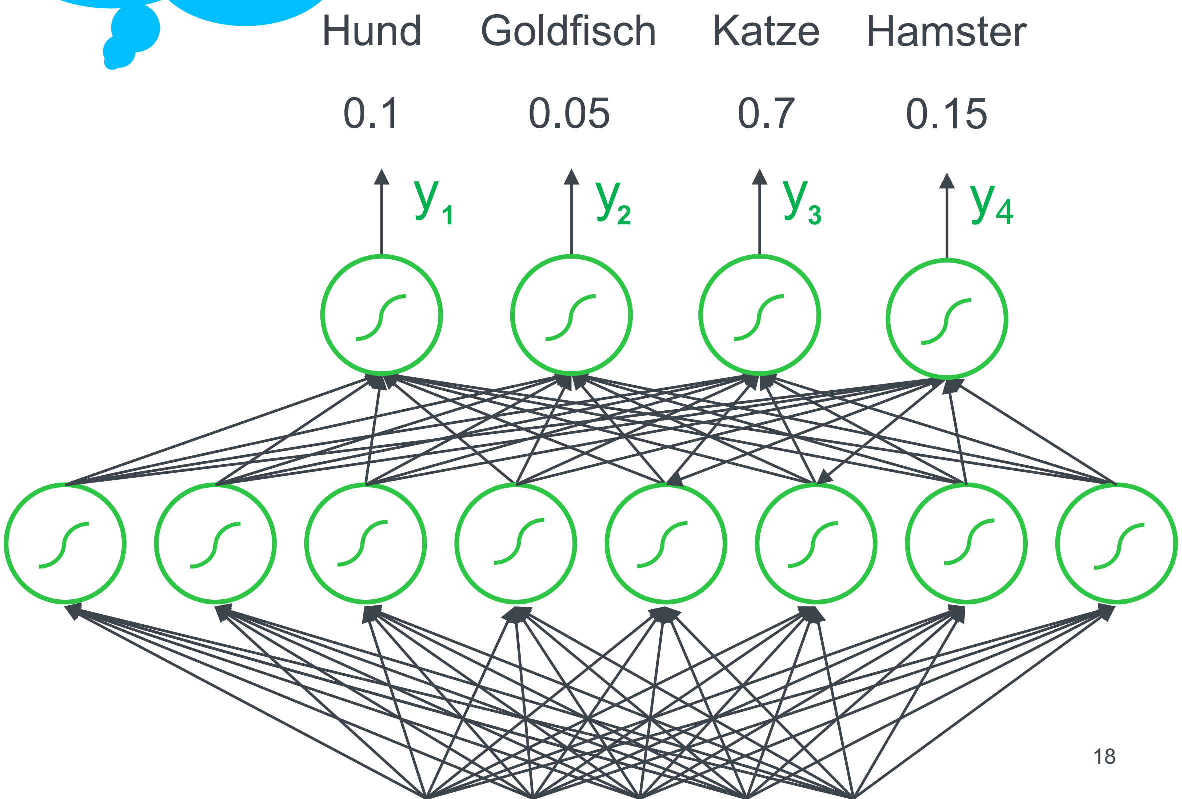
Multiple Choice: Eigenschaften des Mean Squared Error

Verlustfunktion bei Klassifikation

Netzwerk für Klassifikation

Wie vergleicht man Wahrscheinlichkeits-verteilungen?

- Mehrere Ausgänge
- Ausgangswerte entsprechen Wahrscheinlichkeiten
- Insgesamt Wahrscheinlichkeit 1 (100%)
- „Wahrscheinlichkeits-verteilung“
- Ziel: möglichst nah an der „wahren“ Wahrscheinlich-keitsverteilung



Rückblick: Entropie beim Bau von Klassifikationsbäumen

- Maß für Unordnung
  - Hund 30% -log(0.3) * 0.3
  - Katze 40% -log(0.4) * 0.4
  - Hamster 30% -log(0.3) * 0.3

Summe: Entropie

Überraschungs-wert

Klassen-wahrscheinl.

- „Bewertet“ eine Wahrscheinlichkeitsverteilung
- Niedrig, wenn es sehr wahrscheinliche Ereignisse gibt:
  - $-\log(0.999) \cdot 0.999 \approx 0.001 \cdot 0.999 \approx 0.001$
  - $-\log(0.001) \cdot 0.001 \approx 6.908 \cdot 0.001 \approx 0.007$
  - $-\log(0.998) \cdot 0.998 \approx 0.002 \cdot 0.998 \approx 0.002$
  - $-\log(0.002) \cdot 0.002 \approx 6.215 \cdot 0.002 \approx 0.012$

„billiger“ „teurer“

INNOVET KI B³

Vergleich zweier Wahrscheinlichkeitsverteilungen

Überraschungs-wert

Korrekte Wahrscheinlichkeits-verteilung

- Von Modell vorhergesagte Wahrscheinlichkeit
  - Hund 10%
  - Goldfisch 5%
  - Katze 70%
  - Hamster 15%

$$-\log(0.1) \approx 2.30 \quad \cdot 0.2$$
$$-\log(0.05) \approx 3.00 \quad \cdot 0.1$$
$$-\log(0.7) \approx 0.36 \quad \cdot 0.4$$
$$-\log(0.15) \approx 1.90 \quad \cdot 0.3$$

- Hund 20%
- Goldfisch 10%
- Katze 40%
- Hamster 30%

Summe: 1.47

Kreuzentropie

- Teuer, wenn für eine Klasse, die in Wirklichkeit sehr wahrscheinlich ist, ein niedriger Wert vorhergesagt wird

Die Kreuzentropie (engl. Cross Entropy) erhält man, indem man die Überraschungswerte der vorhergesagten Klassenwahrscheinlichkeiten mit den wahren Klassenwahrscheinlichkeiten multipliziert und aufsummiert.

Je niedriger die Kreuzentropie, desto ähnlicher sind sich die vorhergesagte und die wahre Wahrscheinlichkeitsverteilung.

Bei neuronalen Netzwerken für Klassifikationsprobleme wird meist die Kreuzentropie als Verlustfunktion eingesetzt.

Man bezeichnet diese Verlustfunktion dann manchmal auch als „Log Loss“.

Kreuzentropie in der praktischen Anwendung

- In der praktischen Anwendung ist die wahre Verteilung der Klassenwahrscheinlichkeiten nicht bekannt
- Statt dessen Abgleich mit Trainingsdatenpunkt
- Von Modell vorhergesagte Wahrscheinlichkeit
  - Hund 10%
  - Goldfisch 5%
  - Katze 70%
  - Hamster 15%
- Trainingsdatenpunkt:
  - Katze
    - Hund 0%
    - Goldfisch 0%
    - Katze 100%
    - Hamster 0%

Kreuzentropie:
$-\log(0.7) \cdot 1 \approx 0.36$

Kreuzentropie bei falsch vorhergesagter Klasse

- Von Modell vorhergesagte Wahrscheinlichkeit
  - Hund 10%
  - Goldfisch 5%
  - Katze 70%
  - Hamster 15%

- Trainingsdatenpunkt:
  - Hamster
    - Hund 0%
    - Goldfisch 0%
    - Katze 0%
    - Hamster 100%

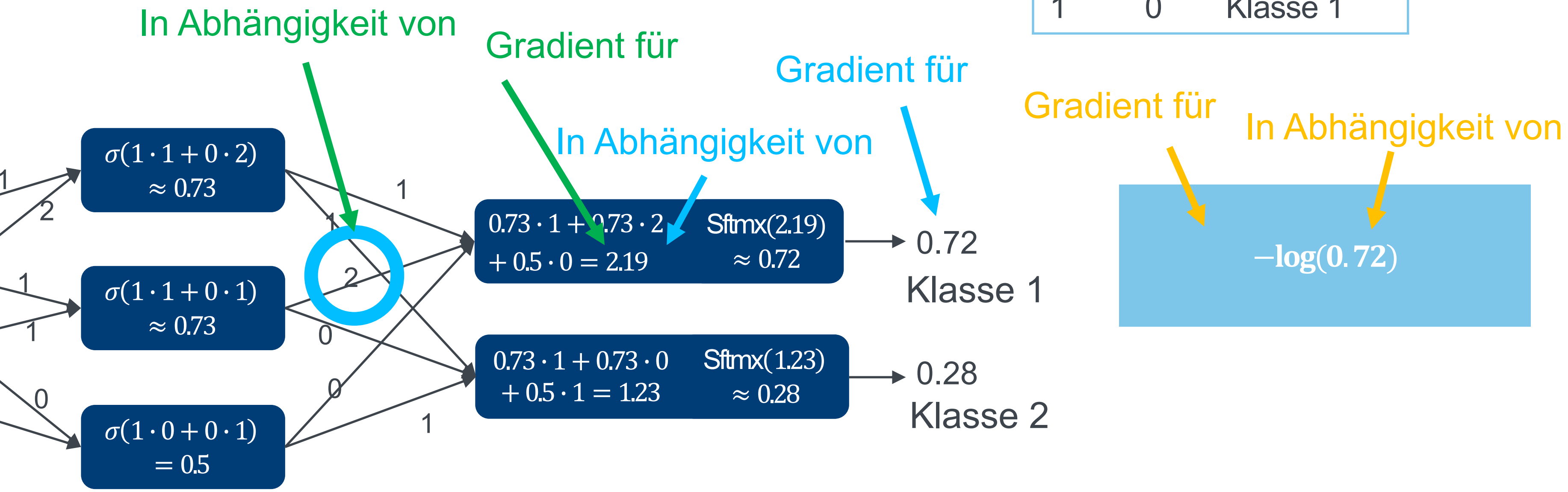
Kreuzentropie:
$-\log(0.15) \cdot 1 \approx 1.90$

Bei der Berechnung der Kreuzentropie für einzelne Trainingsdatenpunkte nimmt man als wahre Wahrscheinlichkeitsverteilung 100% für die laut Trainingsdatenpunkt korrekte Klasse an, und 0% für alle anderen Klassen.

Dadurch vereinfacht sich die Berechnung der Kreuzentropie auf den Überraschungswert der für die korrekte Klasse vorhergesagten Wahrscheinlichkeit.

Beispiel: Gradienten für ein Gewicht schrittweise berechnen

Input		Ergebnis
1	0	Klasse 1



$$-\frac{1}{0.72} \cdot 2.19 \cdot (1 - 2.19) \cdot 0.73 \approx 2.64$$

Zusammenfassung

- Die Verlustfunktion gibt beim Training das Ziel vor
- Ziel: den Verlust minimieren
- Anpassung der Parameter immer mithilfe der Gradienten der Verlustfunktion
- Verlustfunktionen:
  - Bei Regressionsproblemen: MSE
  - Bei Klassifikationsproblemen: Kreuzentropie
- In beiden Fällen wird einmal der Gesamtverlust berechnet (Forward Pass)
- Anschließend bei der Back Propagation Schritt für Schritt die Gradienten der Verlustfunktion für alle Parameter

Question Set: Entropie und Kreuzentropie

Multiple Choice: Gradienten der Verlustfunktion

INNOVET
KI B³
Künstliche Intelligenz in die berufliche Bildung bringen

Dr. Antje Schweitzer

Universität Stuttgart
Institut für Maschinelle Sprachverarbeitung

Universität Stuttgart
Institut für Maschinelle Sprachverarbeitung
Institut für Software Engineering

IHK Industrie- und Handelskammer Reutlingen
Reutlingen | Tübingen | Zollernalb

IHK Region Stuttgart

IHK Industrie- und Handelskammer Karlsruhe

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

GEFÖRDERT VOM

INNOVET

Bundesministerium für Bildung und Forschung

bibb Bundesinstitut für Berufsbildung

Gefördert als InnoVET-Projekt aus Mitteln des Bundesministeriums für Bildung und Forschung.

Lizenzbestimmungen

“Verlustfunktion" von Antje Schweitzer, KI B³ / Uni Stuttgart

Quellenverzeichnis

Titelfoto: Markus Winkler (https://unsplash.com/de/@markuswinkler), ohne Titel, auf Unsplash (https://unsplash.com/de/fotos/afW1hht0NSs), ist lizenziert unter Unsplash-Lizenz (https://unsplash.com/license). Bildausschnitt verändert.