

Probeklausur Neuronale Netze

Wintersemester 20/21

Fachbereich M

Prof. Dr. Dorra Baccar

Teil I: Multiple Choice

1) Welche der folgenden Modelle können beim unüberwachten Lernen verwendet werden?

- ☐ Autoencoders
- ☐ MLP
- ☐ CNN
- ☐ LSTM

2) Sie möchten ein neuronales Netz trainieren, um die Wohnungspreise in Frankfurt vorherzusagen. Welche der folgenden Ausgabefunktionen ist für dieses Regressionsproblem am besten geeignet?

- ☐ ReLu
- ☐ Sigmoid
- ☐ Identity
- ☐ Tanh

3) Welche Aussage ist richtig?

- ☐ Bei der Verwendung von Dropout kann das Netz als ein *Ensemble* aus mehreren unabhängigen kleineren Netzwerken gesehen werden.
- ☐ Dropout wirkt als Regularisierung.
- ☐ Dropout kann auch zur Testzeit angewendet werden.

4) Welche Methoden können helfen, Overfitting zu reduzieren?

- ☐ Erhöhung der Größe des Netzwerks.
- ☐ Verwendung von Data Augmentation.
- ☐ Frühes Beenden des Training.

5) Wie viele Parameter hat ein 2-schichtiges neuronales Netz mit 5 Neuronen in jeder Schicht?

- ☐ 50 Parameter
- ☒ 60 Parameter
- ☐ 101 Parameter

6) Welche der folgenden Aussagen zu CNNs treffen zu?

- ☒ Filter in früheren Schichten enthalten in der Regel Kantendetektoren.
- ☒ Pooling-Layers reduzieren die räumliche Auflösung des Bildes.
- ☐ CNNs haben mehr Parameter als MLPs mit der gleichen Anzahl von Schichten und der gleichen Anzahl von Neuronen in jeder Schicht
- ☐ Ein CNN kann für unüberwachte Lernaufgaben trainiert werden, während ein gewöhnliches neuronales Netz dies nicht kann.

7) Neuronale Netze

- ☒ optimieren eine konvexe Kostenfunktion.
- ☒ können sowohl zur Regression als auch zur Klassifizierung verwendet werden.
- ☐ geben immer Werte zwischen 0 und 1 aus.

8) Die Vertiefung eines neuronalen Netzes durch Hinzufügen weiterer Schichten

- ☒ verlangsamt das Training und die Inferenzgeschwindigkeit.
- ☐ reduziert immer den Trainings-Loss.
- ☐ verbessert immer die Performance bei ungesehenen Daten.

9) Betrachtet man ein trainiertes Netz mit dem Gewichtungsvektor W und eine, auf einem gegebenen Datensatz, Testgenauigkeit A . Unter der Annahme, dass es keinen Bias gibt, ändert sich die Testgenauigkeit nicht, wenn man W durch 2 teilt:

- ☐ Richtig
- ☒ Falsch

Teil II: Kurzantwortfragen

1) Gegeben ein CNN Layer mit 8 Filter, einer Filtergröße von 6, einem Stride von 2 und einem Padding von 1. Wie groß ist die Dimension der Ausgabe bei einem Eingabebild von $32 \times 32 \times 3$?

$$O = \frac{W - F + 2P}{s} + 1$$

(15,15,8)

2) Erklären Sie, wie und warum LSTM-Netzwerke oft besser sind als traditionelle RNNs?

Für traditionelle RNNs ist es aufgrund verschwindender Gradienten schwierig, langfristige Abhängigkeiten zu lernen.

Der Zellzustand in LSTMs verbessert den Gradientenfluss und ermöglicht es dem Netz dadurch, längere Abhängigkeiten zu lernen.

3) Erklären Sie, warum wir Aktivierungsfunktionen benötigen?

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

Aktivierungsfunktionen begrenzen den Output. Wenn der Neuronenwert von $-\infty$ bis $+\infty$ reichen kann, können wir nicht entscheiden, ob wir das Neuron abfeuern müssen oder nicht. Hier hilft uns die Aktivierung, das Problem zu lösen.

4) Welche Rolle spielen die Pooling-Schichten in einem CNN?

Reduzierung der räumlichen Auflösung

5) Das Wort "adversarial" im Akronym für GANs deutet auf ein Spiel mit zwei Spielern hin. Was sind die beiden Spieler und was sind ihre jeweiligen Ziele?

Generator, Diskriminator

G versucht D zu täuschen in dem er sehr gute falsche Bilder generiert.

D wird trainiert, um reale Daten von Falschen Daten zu unterscheiden.

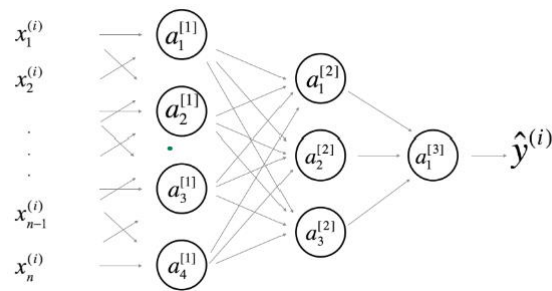
6) Ein Telekommunikationsunternehmen hat einen Klassifikator entwickelt, der erkennt, ob eine Mobilfunk-Basisstation defekt ist oder nicht. Wir evaluieren den Klassifikator auf einer Testmenge. Hier ist die Konfusionsmatrix. (In der Tabelle bedeutet F fehlerhaft und NF nicht fehlerhaft.)

	F	NF
F	10	5
NF	20	985

Berechnen Sie die Genauigkeit (accuracy) des Klassifikators

$$10+985/(10+5+20+985)=0,97$$

7) Sie entwerfen das folgende 2-schichtige FC neuronale Netz



Alle Aktivierungen sind sigmoid und der Optimierer ist stochastischer Gradientenabstieg (SGD). Sie initialisieren alle Gewichte und Bias auf null. Wie groß ist \hat{y} bei der ersten Forward Propagation?

0,5

8) Sie entwerfen ein Deep-Learning-System zur Erkennung von Fahrmüdigkeit in Autos. Es ist entscheidend, dass Ihr Modell Müdigkeit erkennt, um Unfälle zu vermeiden. Welche der folgenden Metriken ist die am besten geeignete: Accuracy, Precision, Recall oder Loss Value?

$$Precision = \frac{TP}{TP+FP}, Recall = \frac{TP}{TP+FN}$$

Recall (Trefferquote)

Recall misst, wie genau das Modell alle positive Vorhersagen von allen Daten trifft, das heisst wie viel Prozent positive Vorhersage von allen Daten richtig ist.

Teil III:

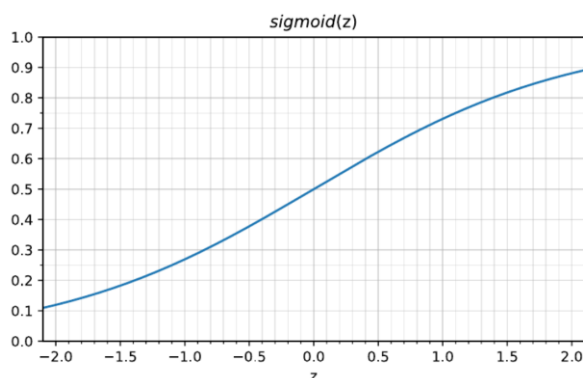
1) Wir möchten einen binären Classifier auf Basis eines logistischen Regressionsmodells erstellen, der Hunde erkennt, d. h. eine Wahrscheinlichkeit p dafür liefert, dass eine gegebene Probe ein Hund ist, während die Wahrscheinlichkeit, dass es sich nicht um einen Hund handelt, $1 - p$ ist.

a) Zur Vorhersage der Wahrscheinlichkeiten p verwenden wir die Sigmoidfunktion $\sigma(z)$. Bitte geben Sie die Formel für $\sigma(z)$ an.

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

b) Zeigen Sie, dass die Ableitung der Sigmoidfunktion $\sigma(z)$ nach dem Eingang z gleich $\sigma(z)(1 - \sigma(z))$ ist.

c) Angenommen, Sie trainieren den Klassifier auf einem einzelnen Bild eines Hundes ($y=1$), mit den Merkmalen $X = [0.8, 0.5, -1]$. Die aktuellen Modellparameter sind $W = [0.5, 0.4, -0.3]^T$ und $b = 0.2$.



Berechnen Sie \hat{y} .

*Führen Sie eine Gradientenabstiegsiteration mit der Lernrate $\alpha = 0,1$ durch und aktualisieren Sie das Gewicht W (die Indizierung beginnt mit 1) in Bezug auf den

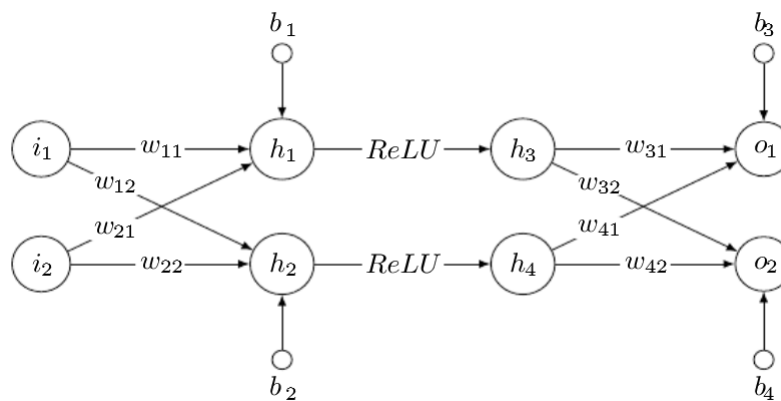
binären Kreuzentropieverlust. Sie können das obige Diagramm verwenden, um Werte der Sigmoidfunktion zu schätzen.

$$\text{Score: } \hat{y} = x \cdot w + b = 0.9 + 0.2 = 1.1$$

$$\text{Probability (from plot) } p = \sigma(y) = \sigma(1.1) \approx 0.75$$

Teil IV:

Gegeben sei das folgende neuronale Netzwerk mit zwei Eingabeeinheiten (i_1, i_2), vier versteckte Einheiten (h_1, h_2) und (h_3, h_4) und ReLU-Aktivierungen.



Die Ausgabeeinheiten werden als (o_1, o_2) und ihre Targets als (y_1, y_2) angegeben. Die Gewichte und die Bias der FC Layer werden als w und b bezeichnet.

Die Werte der Variablen sind in der folgenden Tabelle angegeben:

Variable	i_1	i_2	w_{11}	w_{12}	w_{21}	w_{22}	w_{31}	w_{32}	w_{41}	w_{42}	b_1	b_2	b_3	b_4	t_1	t_2
Value	2.0	-1.0	1.0	-0.5	0.5	-1.0	0.5	-1.0	-0.5	1.0	0.5	-0.5	-1.0	0.5	1.0	0.5

- a) Schreiben Sie Ihre Berechnungen über den Vorwärtsthroughlauf auf, d. h. ermitteln Sie den mittleren quadratischen Loss (MSE)

Forward pass:

$$h_1 = i_1 \times w_{11} + i_2 \times w_{21} + b_1 = 2.0 \times 1.0 - 1.0 \times 0.5 + 0.5 = 2.0$$

$$h_2 = i_1 \times w_{12} + i_2 \times w_{22} + b_2 = 2.0 \times -0.5 + -1.0 \times -1.0 - 0.5 = -0.5$$

$$h_3 = \max(0, h_1) = h_1 = 2$$

$$h_4 = \max(0, h_2) = 0$$

$$o_1 = h_3 \times w_{31} + h_4 \times w_{41} + b_3 = 2 \times 0.5 + 0 \times -0.5 - 1.0 = 0$$

$$o_2 = h_3 \times w_{32} + h_4 \times w_{42} + b_4 = 2 \times -1.0 + 0 \times 1.0 + 0.5 = -1.5$$

$$MSE = \frac{1}{2} \times (t_1 - o_1)^2 + \frac{1}{2} \times (t_2 - o_2)^2 = 0.5 \times 1.0 + 0.5 \times 4.0 = 2.5$$

- b) Wenn wir ein neuronales Netz implementieren, implementieren wir normalerweise eine "forward-" und eine "backward-" Funktion für jede Schicht. Erklären Sie, was diese Funktionen tun, welche Argumente sie annehmen und was sie zurückgeben.

Forward-Funktion:

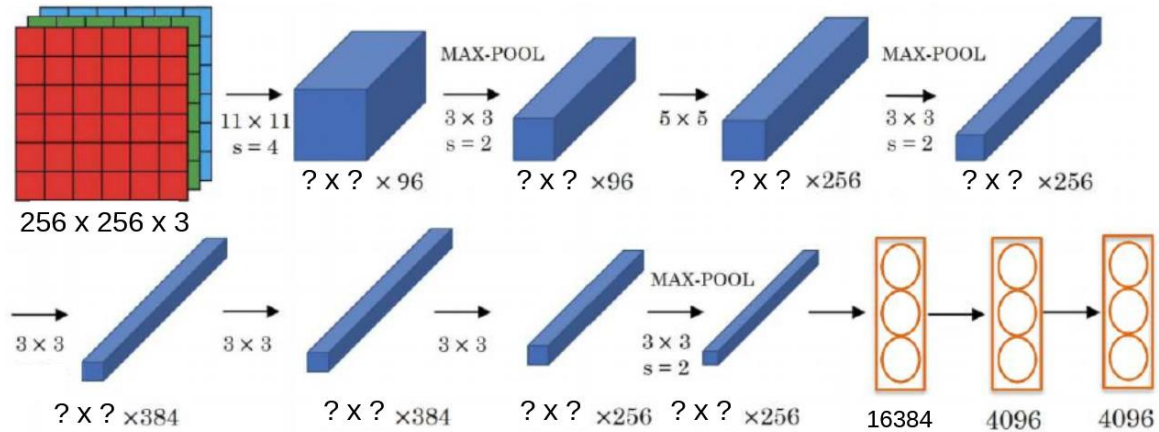
- nimmt die Ausgabe der vorherigen Schicht, führt die Operation aus, gibt das Ergebnis zurück.
- zwischenspeichert Werte, die für die Gradientenberechnung beim Backpropagation benötigt werden.

Backward-Funktion:

- nimmt den vorgelagerten Gradienten, gibt alle partiellen Ableitungen zurück.

Teil V:

Sie müssen Bilder der Größe 256×256 in eine von 1000 Klassen klassifizieren. Da Sie gelernt haben, dass CNNs für solche Aufgaben hervorragend geeignet sind, beschließen Sie, mit der folgenden CNN-Architektur zu beginnen:



- Die Werte direkt unter den Pfeilen geben die Filtergröße f der entsprechenden convolutional- und pooling-Operationen an.
- s steht für stride. Wenn kein Stride angegeben ist, wird ein Stride von $s = 1$ verwendet.
- Alle convolutional und pooling Layers verwenden ein Padding von $p = \frac{f-1}{2}$, bei entsprechender Filtergröße f .

a) Welche klassische Netzwerkarchitektur ist der obigen ähnlich?

AlexNet

b) In der obigen Abbildung fehlt die Ausgabeschicht. Erklären Sie, welche Art von Schicht Sie dort verwenden würden, welche Dimensionen sie hätte und welche Aktivierungs- und Lossfunktionen Sie wählen würden.

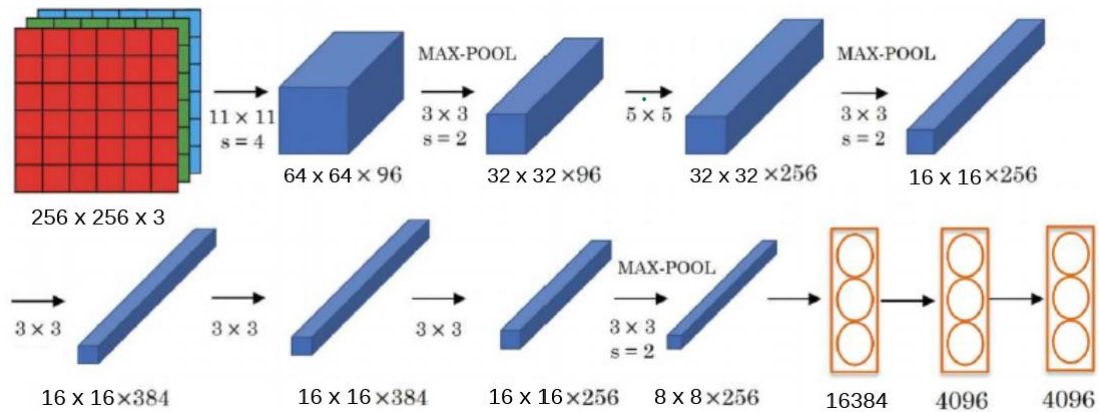
FC mit 1000 Neuronen

Aktivierungsfunktion: Softmax

Loss: Crossentropy

- c) Erklären Sie, wie man die Ausgabegrößen von convolutional Layers berechnet. Wenden Sie dann Ihre Formel an, um die Höhen und Breiten aller Schichten des Netzwerks zu ermitteln.

$$\text{out} = \text{in} - f + 2 * p / s + 1$$



- d) Berechnen Sie die Anzahl der Parameter in der ersten convolutional Layer.

$$11 * 11 * 3 * 96 + 96 = 34944$$

- e) Leider ist die Datenmenge eher klein. Online haben Sie einen viel größeren Datensatz für die Bildklassifikation gefunden, der fast ähnliche Bilder enthält, aber die Anzahl der Klassen ist anders. Erläutern Sie, wie Sie den größeren Datensatz nutzen könnten und wie jede Schicht in Ihrem Netzwerk dabei beeinflusst werden würde.

Wir können den größeren Datensatz für das Transfer-Lernen verwenden, indem wir das Modell zunächst auf dem großen Datensatz vortrainieren und es dann auf unseren Datensatz abstimmen.