

# Aufgabensammlung

## Aufgabe 1

Erläutern Sie den Unterschied zwischen Modell- und Hyperparametern.

Modellparameter sind während des Trainings anpassbare (lernbare) Parameter. Hyperparameter sind Parameter des Lernalgorithmus, die durch den Entwickler vorgegeben werden.

## Aufgabe 2

Erläutern Sie die Begriffe *Supervised* und *Unsupervised Learning*.

Supervised Learning (Überwachtes Lernen) beschreibt das Lernen durch Nachahmung. Dabei wird dem Algorithmus für den  $i$ -ten Datenpunkt ein Tupel  $(x_i, y_i)$  mit dem Datenpunkt  $x_i \in \mathbb{R}^n$  und dem zugehörigen Ergebnis (Label)  $y_i$  übergeben. Der Algorithmus lernt die Funktion  $f(x_i) = y_i$ .

Beim Unsupervised Learning (Unbewachtes Lernen) entfällt das Label, sodass dem Algorithmus ausschließlich die Datenpunkte  $x_i$  übergeben werden. Da nun keine direkte Verbindung zu einem erwarteten Ergebnis gelernt werden kann, sind Anwendungen für das unüberwachte Lernen zum Beispiel die Gruppierung ähnlicher Datenpunkte.

## Aufgabe 3

Definieren Sie den Begriff *Feature Space*.

Der Feature Space ist die Repräsentation eines Datensets, welche die Anwendung maschineller Lernverfahren erlaubt. Das bedeutet insbesondere, dass mit allen Einträgen des Feature Spaces gerechnet werden kann. Der Feature Space ist der Ausschnitt der Welt, den das Modell "sieht".

## Aufgabe 4

Nennen Sie die Formel der linearen Regression mit Polynom-Trick vom Grad 2 für die Eingabe  $x \in \mathbb{R}^2$ .

$$f(x) = \omega_0 + \omega_1 x_1 + \omega_2 x_2 + \omega_3 x_1^2 + \omega_4 x_2^2 + 2\omega_5 x_1 x_2$$

## Aufgabe 5

Beschreiben Sie den Unterschied zwischen *Hard-Margin*- und *Soft-Margin-Classifer* in Bezug auf eine *Support Vector Machine (SVM)*. Welche Konsequenzen hat die Wahl des einen oder anderen für ihre Machine-Learning-Anwendung?

Soft-Margin-Classifier erlauben Ausreißer innerhalb des Trennbereichs, der für die Abgrenzung der Daten vorgesehen ist (die "Straße"). Hard-Margin-Classifier liefern für ein nicht vollständig trennbares Datenset kein Ergebnis.

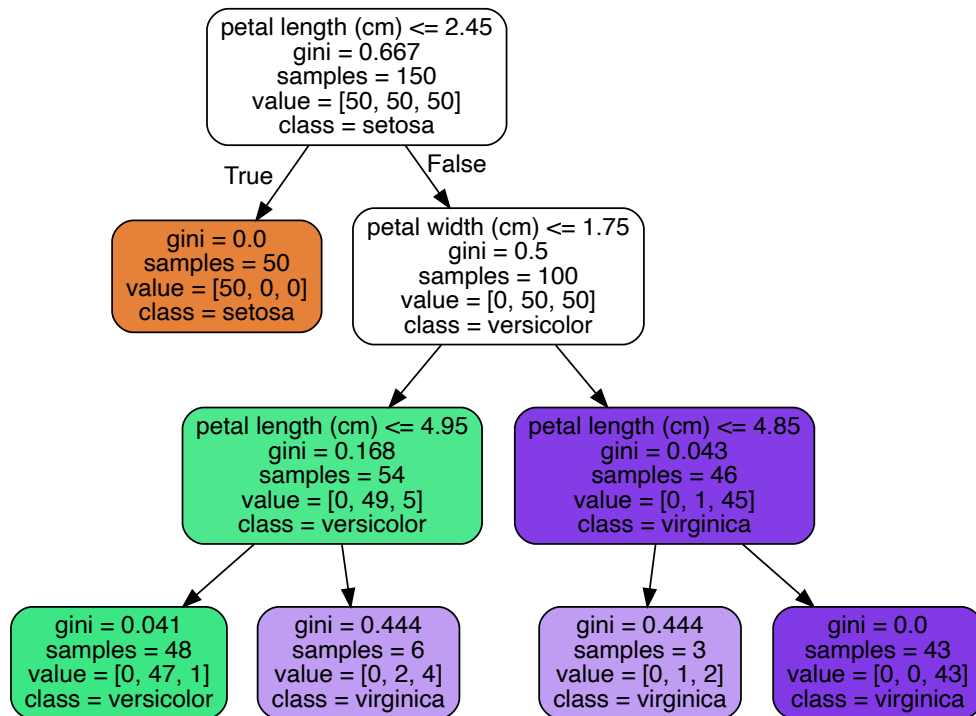
**Aufgabe 6**

Nennen Sie die Formel zur Vorhersage eines Datenpunktes  $x \in \mathbb{R}^2$  mittels einer *Support Vector Machine (SVM)*.

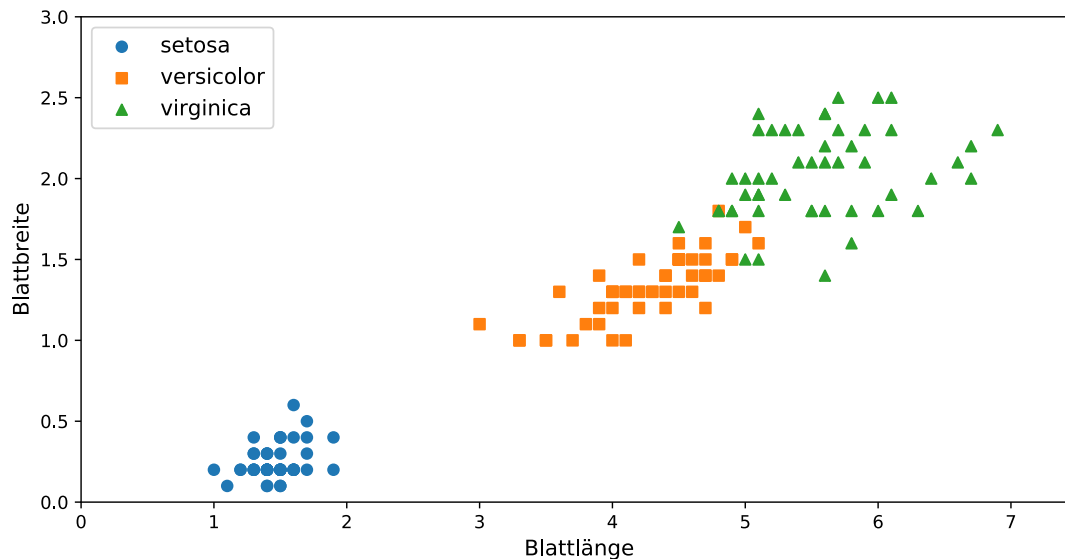
$$f(x) = \text{sign}(\omega_0 + \omega_1 x_1 + \omega_2 x_2)$$

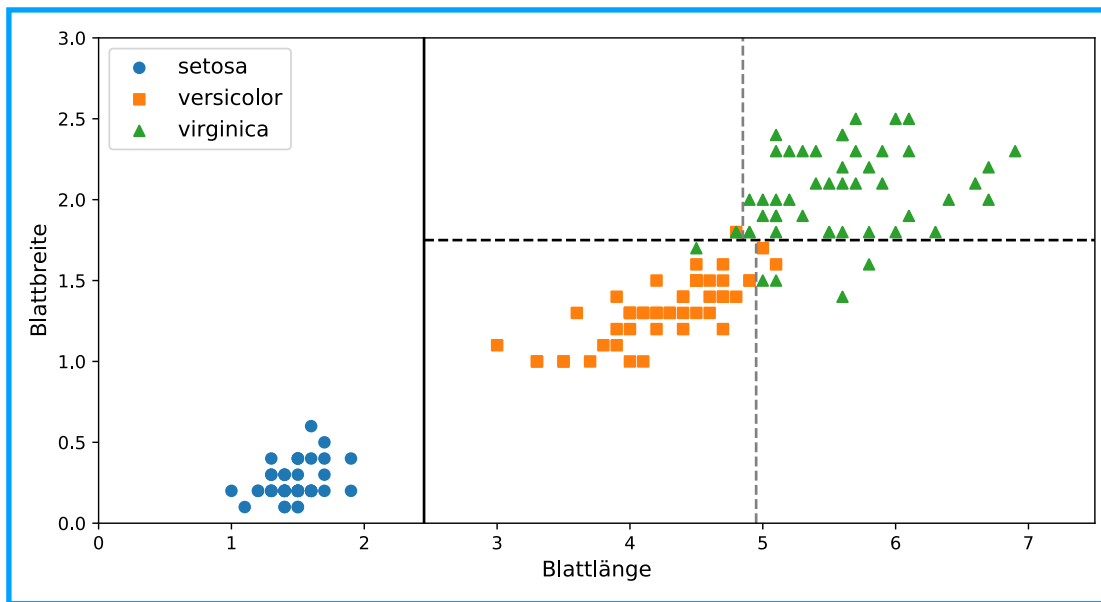
## Aufgabe 7

Zeichnen Sie im *Feature Space* die *Entscheidungsgrenze* (Decision Boundary) des abgebildeten Entscheidungsbaums ein. Welcher Klasse wird der Datenpunkt mit den Merkmalen *petal length*=4.9 und *petal width*=1.73 zugewiesen?



Zeichnen Sie *hier* Ihre Lösung ein:





Dem Datenpunkt wird die Klasse *versicolor* zugewiesen.

## Aufgabe 8

Erläutern Sie die Schritte, die der Algorithmus *CART (Classification and Regression Trees)* durchläuft, um den nächsten Split eines *Decision Trees* zu finden.

Wähle ein Feature  $f$  und einen Schwellwert (eine Wertausprägung des Features)  $s$ .

Initialisiere die leeren Mengen  $D_l$  und  $D_r$ .

Prüfe für jeden Datenpunkt  $x_i$  der Eingabemenge  $D$ , ob der Wert von  $x_i$  an der Stelle  $f$  kleiner gleich  $s$  ist.

Ist dies der Fall, füge  $x_i$  der Menge  $D_l$  hinzu. Andernfalls wird  $x_i$  der Menge  $D_r$  hinzugefügt.

Bestimme die Güte des Splits, indem die Reinheit von  $D_l$  und  $D_r$  bestimmt werden (minimale Gini-Unreinheit oder MSE).

Wiederhole den Vorgang für jede Feature-Schwellwert-Kombination.

Der Split mit größten Reinheit (minimale Unreinheit) wird übernommen.

## Aufgabe 9

Erläutern Sie wieso für einen *Random Forest* unkorrelierte Entscheidungsbäume benötigt werden.

Beim *Random Forest* wird eine Vorhersage über die kombinierte Vorhersage (z.B. der Durchschnitt) der einzelnen Entscheidungsbäume getroffen. Wären die einzelnen Entscheidungsbäume perfekt korreliert (treffen dieselben Vorhersagen), so würde die Performanz des *Random Forest* der eines einzelnen Entscheidungsbaums gleichen.

## Aufgabe 10

Nennen und beschreiben Sie zwei Maßnahmen, mit denen sichergestellt werden soll, dass für einen *Random Forest* unkorrelierte Bäume trainiert werden.

*Bagging (Bootstrap Aggregating)* führt zu unkorrelierten Entscheidungsbäumen. Dabei wird für jeden Entscheidungsbaum eine separate Stichprobe aus dem Datenset gezogen. Der Entscheidungsbaum wird ausschließlich auf seiner eigenen Stichprobe trainiert. Somit unterscheiden sich die Training-Sets der einzelnen Bäume.

Durch die Einschränkung der Parameter der Entscheidungsfunktion jedes Entscheidungsbaums können ebenfalls unkorrelierte Bäume trainiert werden. Dabei werden jedem Entscheidungsbaum beispielsweise nur Splits entlang bestimmter *Merkmale (Features)* erlaubt. Je stärker die Wahl der Parameter eingeschränkt wird, desto schwächer ist die Korrelation der Bäume.

## Aufgabe 11

Gegeben sei das folgende Datenset. Transformieren Sie die Daten in einen für das Training maschineller Lernverfahren geeigneten *Feature Space*. Beschreiben Sie ihr Vorgehen unter Benennung der durchgeführten Transformationen.

Alter (Jahre)	Einkommen (€)	Bildungsabschluss	Kredit (Label)
47	60k	M.Sc.	1
19	5000,00 €	Abitur	0
-	43000,00 €	B.Sc.	0
25	39000,00 €	M.Sc.	1
31	80000,00 €	M.Sc.	-


Alter	Einkommen	Abitur	B.Sc.	M.Sc.	Kredit (Label)
47	60	0	0	1	1
19	5	1	0	0	0
30.5	43	0	1	0	0
25	39	0	0	1	1

Ersetze fehlende Werte der Spalte Alter durch den Durchschnittswert.

Entferne alle Zeilen mit fehlendem Label.

One-Hot-Encoding der ordinal skalierten Variablen.

Einheitliche Skala für das Feature Einkommen.

## Aufgabe 12

Gegeben seien die Ergebnisse einer Evaluierung verschiedener Machine Learning Modelle für die Prognose über eine Corona-Infektion. Für welches Modell würden Sie sich entscheiden? Begründen Sie Ihre Wahl.

### Modell 1

Accuracy: 0,90

Patient (Zeile)/ Prediction (Spalte)	0	1
0	81	0
1	9	0

### Modell 2

Accuracy: 0,80

Patient (Zeile)/ Prediction (Spalte)	0	1
0	67	14
1	4	5

### Modell 3

Accuracy: 0,74

Patient (Zeile)/ Prediction (Spalte)	0	1
0	58	23
1	0	9

**Modell 2:** Modell 2 besitzt nach Modell 1 die höchste Genauigkeit. Modell 1 hat keine Unterscheidung der Klassen gelernt, sondern sagt immer die 0 vorher (eine Spalte in der *Confusion Matrix* ist komplett 0).

**Modell 3:** Modell 3 besitzt zwar die niedrigste Genauigkeit, führt jedoch nicht zu *False Negatives*, die eine erkrankte Person als gesund einstufen würden. Dies kann eine wichtige Qualitätseigenschaft sein.

## Aufgabe 13

Beschreiben Sie die Schritte, die zur Durchführung der *Principal Component Analysis (PCA)* notwendig sind, um die  $n$  Hauptkomponenten eines Datensets zu bestimmen.

Berechne die Kovarianzmatrix des Datensets.

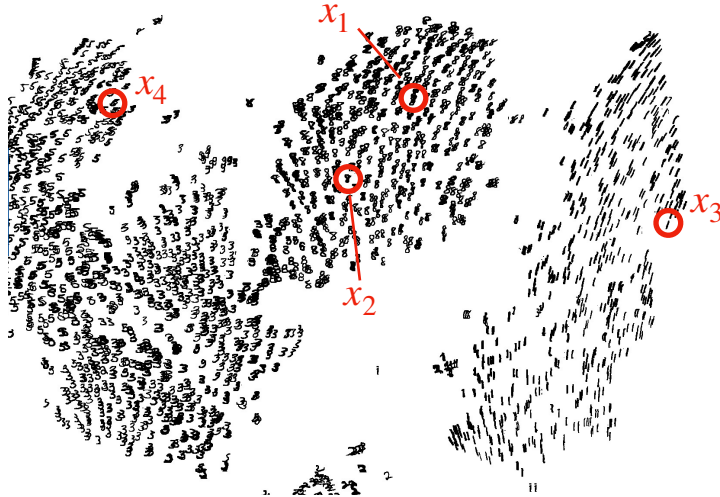
Bestimme die Eigenwerte und -vektoren der Kovarianzmatrix.

Sortiere die Eigenwerte absteigend.

Die Eigenvektoren zu den ersten  $n$  Eigenwerten bilden die  $n$  Hauptkomponenten.

## Aufgabe 14

Die Abbildung zeigt einen Ausschnitt aus dem MNIST-Datenset, welches mithilfe des Algorithmus *t-Stochastic Neighbourhood Embedding* auf zwei Dimensionen reduziert wurde. Welche Aussage können Sie über die Datenpunkte  $x_1$  und  $x_2$  treffen? Welche Bedeutung hat der Abstand zwischen  $x_3$  und  $x_4$ ? Begründen Sie Ihre Aussage anhand der Funktionsweise des *t-SNE*-Algorithmus.



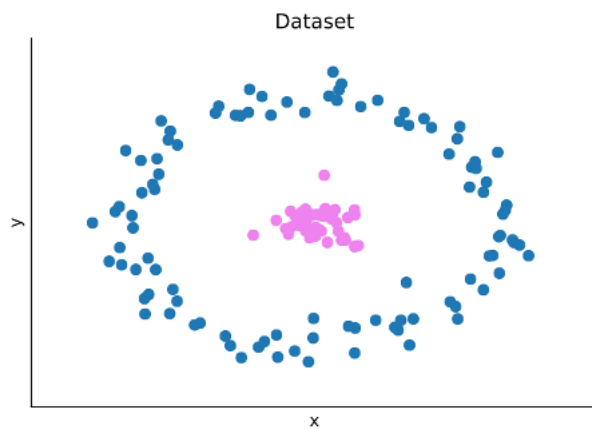
Die Datenpunkte  $x_1$  und  $x_2$  liegen in derselben Punktwolke. Da t-SNE ähnliche Datenpunkte in lokaler Nähe zueinander abbildet, spricht dies dafür, dass  $x_1$  und  $x_2$  ähnlich zueinander sind (derselben Klasse angehören).

Die Datenpunkte  $x_3$  und  $x_4$  liegen in unterschiedlichen Punktwolken. Da t-SNE die globale Struktur eines Datensets nicht erhält, können wir keine Aussage darüber treffen, ob  $x_3$  ähnlicher zu  $x_1$  oder zu  $x_4$  ist.



## Aufgabe 15

Das abgebildete Datenset soll mithilfe des *k-Means*-Algorithmus gruppiert werden. Halten Sie diese Wahl für sinnvoll? Begründen Sie Ihre Antwort unter Einbeziehung der Funktionsweise von *k-Means* und verschiedener Werte für den Hyperparameter *k*.

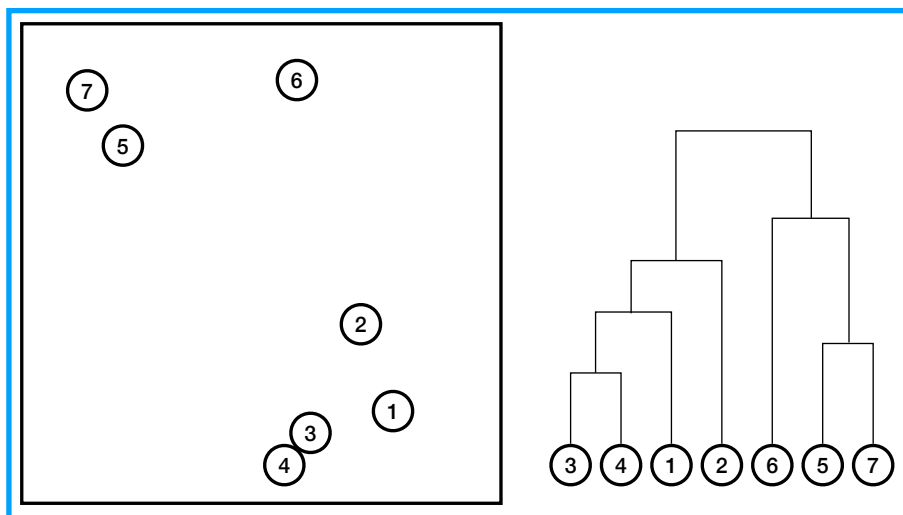
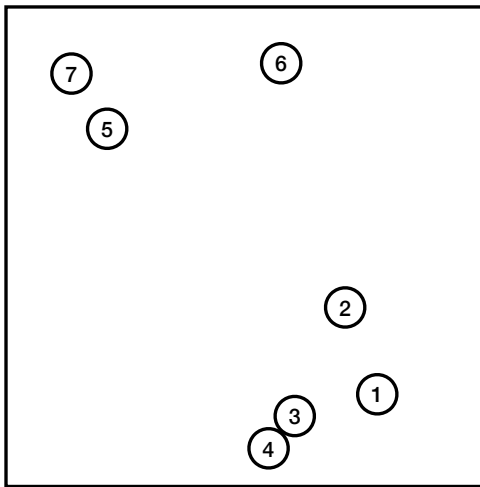


Beim *k-Means*-Algorithmus werden die *Centroide* (*Cluster-Center*) immer wieder in die Mitte der zugeordneten Datenpunkte geschoben. Dies führt dazu, dass für  $k=2$  die Datenpunkte beispielsweise in eine linke und eine rechte Hälfte aufgeteilt werden. Die tatsächliche Gruppenzugehörigkeit geht dabei verloren. Wählt man ein höheres  $k$  werden mehrere Centroide Bereiche des blauen Rings abdecken. Dieser wird also in mehrere Gruppen aufgeteilt. Die pinke Gruppe hingegen kann auch ein Centroid abgebildet werden.

*k-Means* eignet sich folglich nur für Datensets, in denen die Punktwolken ansatzweise einer Hyperkugel (2D=Kreis, 3D=Kugel) entsprechen. Dies ist hier nicht der Fall, sodass *k-Means* ungeeignet ist.

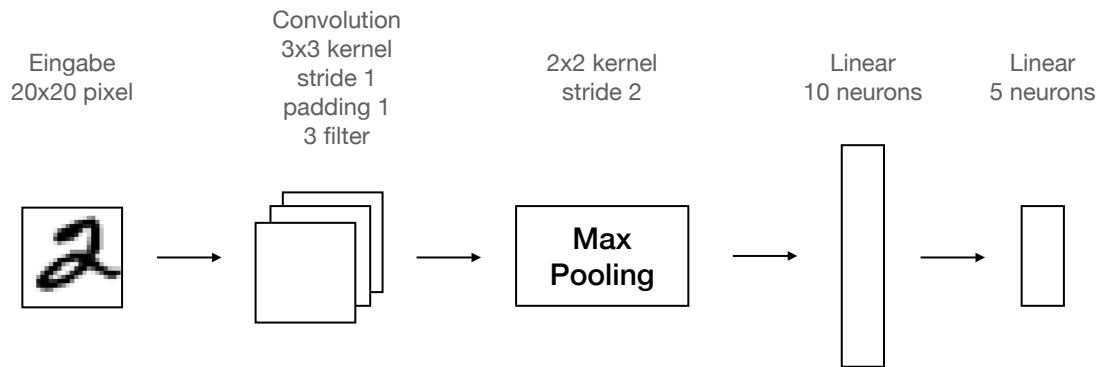
**Aufgabe 16**

Wenden Sie den Algorithmus *Agglomerative Clustering* mit dem Unähnlichkeitsmaß *Single Linkage* (mit euklidischer Distanz) auf den gegebenen Datensatz an und vervollständigen Sie das zugehörige Dendrogramm.



## Aufgabe 17

Wieviele Modell-Parameter hat das abgebildete Neuronale Netz?



Conv-Layer:  $3 * 3 * 3 = 27$  Parameter

Linear-Layer-1:  $(20/2) * (20/2) * 3 * 10 + 10 = 3010$  Parameter

Linear-Layer-2:  $10 * 5 + 5 = 55$  Parameter

**Gesamt:**  $27 + 3010 + 55 = 3092$  Parameter

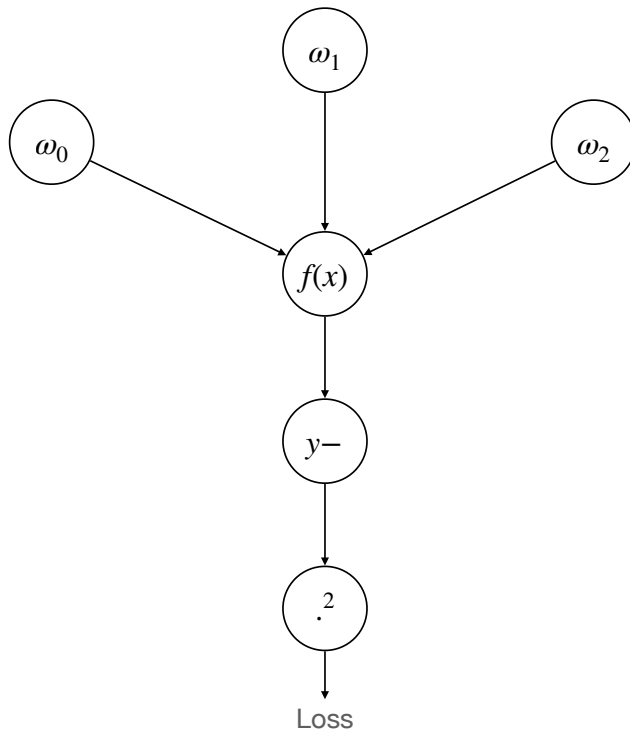
**Aufgabe 18**

Bestimmen Sie den Gradient  $\frac{\partial L}{\partial \omega}$  der Kostenfunktion  $L$  an der Stelle  $x_i$ . Gehen Sie dabei analog zur *Backpropagation* vor und notieren Sie die Zwischenschritte im gegebenen Berechnungs-Graph.

$$x_i = (1, 3), \quad y_i = 2.3, \quad \omega = (0.1, 0.1, 0.1)$$

$$f(x) = \omega_0 x_1^2 + \omega_1 x_2^2 + \omega_2 x_1 x_2$$

$$L = \frac{1}{N} \sum_{i=1}^N (y - f(x))^2$$

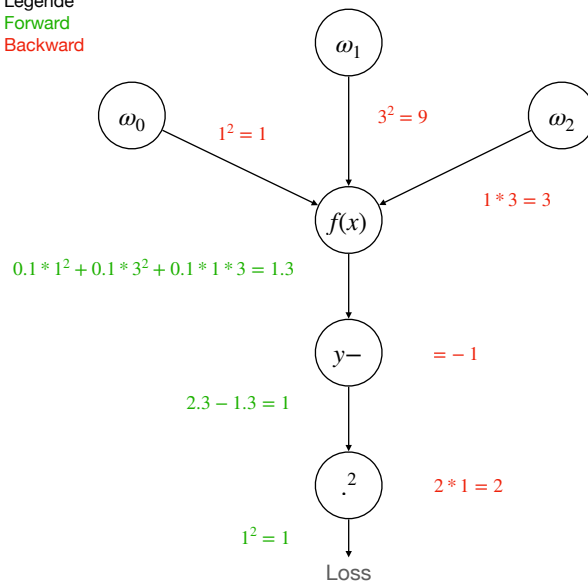


$$\frac{\partial L}{\partial w_0} =$$

$$\frac{\partial L}{\partial w_1} =$$

$$\frac{\partial L}{\partial w_2} =$$

Legende  
Forward  
Backward



$$\frac{\partial L}{\partial w_0} = 2 * (-1) * 1 = -2$$

$$\frac{\partial L}{\partial w_1} = 2 * (-1) * 9 = -18$$

$$\frac{\partial L}{\partial w_2} = 2 * (-1) * 3 = -6$$

## Aufgabe 19

Führen Sie eine *Convolution* auf dem Bild unter Anwendung des gegebenen *Kernels* durch. Gehen Sie dabei von einem *Zero-Padding* von 1 entlang jeder Dimension und einer Schrittweite (*stride*) von ebenfalls 1 in jede Richtung aus.

Kernel			Bild					Ergebnis				
0	-1	0	0	0	1	1	0					
-1	4	-1	0	1	1	1	0					
0	-1	0	1	1	1	1	0					
			0	0	1	1	0					
			0	0	1	1	0					

Kernel			Bild					Ergebnis				
0	-1	0	0	0	1	1	0	0	-2	2	2	-1
-1	4	-1	0	1	1	1	0	-2	2	0	1	-1
0	-1	0	1	1	1	1	0	3	1	0	1	-1
			0	0	1	1	0	-1	-2	1	1	-1
			0	0	1	1	0	0	-1	2	2	-1

**Aufgabe 20**

Wenden Sie ein  $2 \times 2$  *Max-Pooling* auf das gegebene Bild mit einer Schrittweite (*stride*) von 2 in jede Richtung an.

7	3	8	0
5	5	9	0
5	5	1	10
3	4	0	0


7	3	8	0
5	5	9	0
5	5	1	10
3	4	0	0

7	9
5	10