Ausgewählte Kapitel sozialer Webtechnologien
Test-Klausur WiSe 2020/21

Name: _____

Matrikelnummer: _____

- Schreiben Sie auf jedes Blatt, das Sie abgeben, Name und Matrikelnummer
- Schreiben Sie sauber, nicht leserliche Abgaben können zu Punktabzug führen

Viel Erfolg!

Aufgabe:	1	2	3	4	5	6	Summe:
Punkte:	12	24	12	12	24	16	100
Ergebnis:							

1. (12 Punkte) Multiple-Choice-Fragen

   Für jede der folgenden Fragen kreisen Sie bitte die Antwort(en) ein. Es können mehrere Antworten korrekt sein. Falsche Angaben bewirken Punkteabzug (keine Negativpunkte möglich).

   (a) (2 Punkte) Um Backpropagation durch einen Max-Pooling-Layer zu machen, müssen Sie die Indizes der maximal Werte aus dem Forward-Path wissen?

   A. Wahr

   B. Falsch

   (b) (2 Punkte) Sie implementieren einen Klassifikator zum Erkennen von Krebs. Das Model unterscheidet zwischen Tumor- (label 1) und Normal-Gewebe (label 0), es soll von Onkologen_innen am Krankenhaus verwendet werden. Welche der beiden Metriken würden Sie in diesem Kontext verwenden?

   A. Precision $= \frac{TruePositives}{TruePositives+FalsePositives}$

   B. Recall $= \frac{TruePositives}{TruePositives+FalseNegatives}$

   (c) (2 Punkte) Um ein gute Generalisierung zu erreichen, sollte ein neuronales Netzwerk mindestens doppelt so viele lernbare Parameter besitzen, wie Sie Traningsdaten zur Verfügung haben.

   A. Wahr

   B. Falsch

   (d) (2 Punkte) Der Backpropagation-Algorithmus aktualisiert alle lernbaren Parameter in einem Netzwerk.

   A. Wahr

   B. Falsch

   (e) (2 Punkte) Sie haben ein simples neuronales Netzwerk mit einem Convolutional-Layer. Welche der folgenden Eigenschaften treffen auf das Netzwerk zu?

   A. Es ist Rotationsinvariant

   B. Es ist Translationsinvariant

   C. Es ist Skaleninvariant

   D. Alle Eigenschaften treffen zu

   (f) (2 Punkte) Welche Aussagen über Convolutional-Neural-Networks (CNNs) im Kontext der Bildanalyse sind korrekt?

   A. Filter in den ersten Faltungsschichten entsprechen Kantendetektoren

   B. CNNs haben bei gleicher Anzahl von Neuronen und Schichten mehr lernbare Parameter als vollständig-vernetzte Netzwerke (fully-connected networks)

   C. Pooling-Schichten reduzieren die Auflösung eines Bildes (down sampling)

   D. CNNs eignen sich nicht für Regressionsprobleme

2. (24 Punkte) Verständnisfragen (kurze Antworten)

   Geben Sie eine prägnante Antwort auf jede Frage in zwei bis drei Sätzen.

   (a) (3 Punkte) Sie haben als Eingabe ein Bild mit den Dimensionen $100x100x3$. Sie transformieren es in einen Vektor (image flattening) und nutzen es in einer vollständig vernetzten Schicht (fully-connected layer) mit 50 Neuronen. Welche Dimensionen haben die Gewichtsmatrix und der Bias-Vektor?

   Lösung: weight matrix: (100x100x3)x50 = 30000x50 bias vector: 50x1

   (b) (3 Punkte) Sie senden ein Bild X durch ein neuronales Netzwerk. Die Ausgabe ist eine Wahrscheinlichkeit $\hat{y}$. Erläutern Sie was $\frac{\partial \hat{y}}{\partial x}$ bedeutet?

   Lösung: The derivative represents how much the output changes when the input is changed. In other words, how much the input has influenced the output.

   (c) (3 Punkte) Wenn Ihre Eingabe aus nur zwei Dimensionen besteht, können Sie die Entscheidungsgrenze im euklidischen Raum darstellen und erkennen, ob Ihr Model über angepasst (overfitting) ist. Wie stellen Sie dies bei höher dimensionalen Eingaben, beispielsweise einem Bild, fest?

   Lösung: Compute cost function in the dev and training set. If there is a significant difference, then you have a variance problem.

   (d) (3 Punkte) Angenommen, es folgen in einem Convolutional-Neural-Network auf eine Faltungsschicht $CONV1$ direkt drei weitere Faltungsschichten $CONV2$, $CONV3$ und $CONV4$. Alle Schichten haben eine Filtergrösse von (3,3) und eine Stride $S = 1$. Welches effektive Receptive Field wird durch diese Stapelung auf dem Input von $CONV1$ realisiert?

   Lösung: 9x9

   (e) (3 Punkte) Erläutern Sie weshalb Dropout eine Regularisierungstechnik ist?

Lösung: One of those answers is enough for full points.

(a) Dropout is a form of model averaging. In particular, for a layer of H nodes, we sampling from 2H architectures, where we choose an arbitrary subset of the nodes to remain active. The weights learned are shared across all these models means that the various models are regularizing the other models.

(b) Dropout helps prevent feature co-adaptation, which has a regularizing effect.

(c) Dropout adds noise to the learning process, and training with noise in general has a regularizing effect.

(d) Dropout leads to more sparsity in the hidden units, which has a regularizing effect. (Note that in one of the lecture videos, this was phrased as dropout shrinking the weights or spreading out the weights. We will also accept this phrasing.)

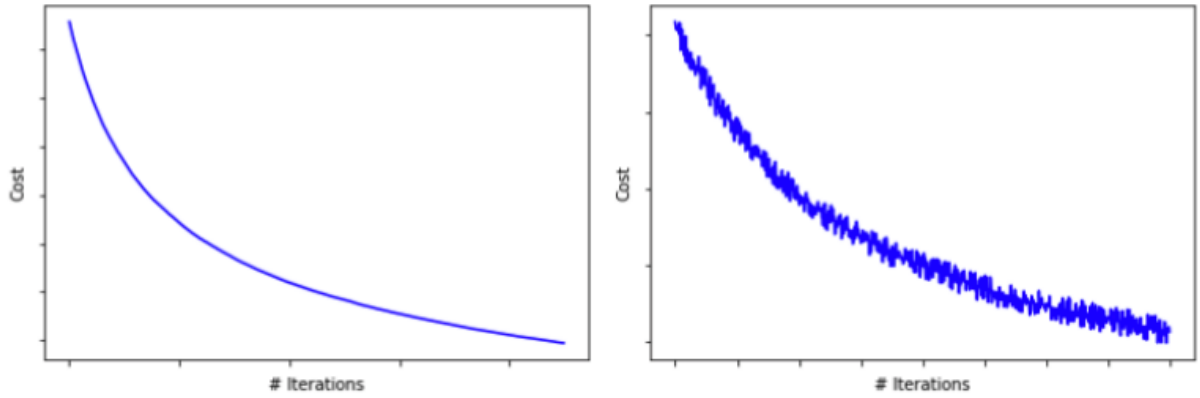
(f) (3 Punkte) Welche Probleme erwarten Sie, wenn eine zu hohe Lernrate verwendet wird? Wie können Sie dieses Problem feststellen?

Lösung: Cost function does not converge to an optimal solution and can even diverge. To detect, look at the costs after each iteration (plot the cost function v.s. the number of iterations). If the cost oscillates wildly, the learning rate is too high. For batch gradient descent, if the cost increases, the learning rate is too high.

(g) (3 Punkte) Was ist ein Sattelpunkt? Welche Vor-/Nachteile hat das Mini-Batch Gradientenabstiegsverfahren (mini batch gradient descent) bei der Problematik von Sattelpunkten?

Lösung: Saddle point - The gradient is zero, but it is neither a local minima nor a local maxima. Also accepted - the gradient is zero and the function has a local maximum in one direction, but a local minimum in another direction. SGD has noisier updates and can help escape from a saddle point

(h) (3 Punkte) Die folgenden Grafiken zeigen, wie sich die Kosten für zwei unterschiedliche Optimierungsalgorithmen über mehrere Iterationen verringern. Erläutern und begründen Sie anhand der Kurvenverläufe welcher Algorithmus das Batch-Gradientenabstiegsverfahren (batch gradient descent) und welcher das Mini-Batch-Gradientenabstiegsverfahren (mini batch gradient descent) nutzt.



(a) Graph A (b) Graph B

Lösung: Batch gradient descent - Graph A, Minibatch - Graph B. Batch gradient descent - the cost goes down at every single iteration (smooth curve). Mini-batch - does not decrease at every iteration since we are just training on a mini-batch (noisier)

3. (12 Punkte) Lineare Algebra

Ihnen sind folgende Informationen zu einem neuronalen Netzwerk gegeben. Gewichtsmatrizen der Schichten, wobei das erste Element jeder Zeile einem Bias entspricht:

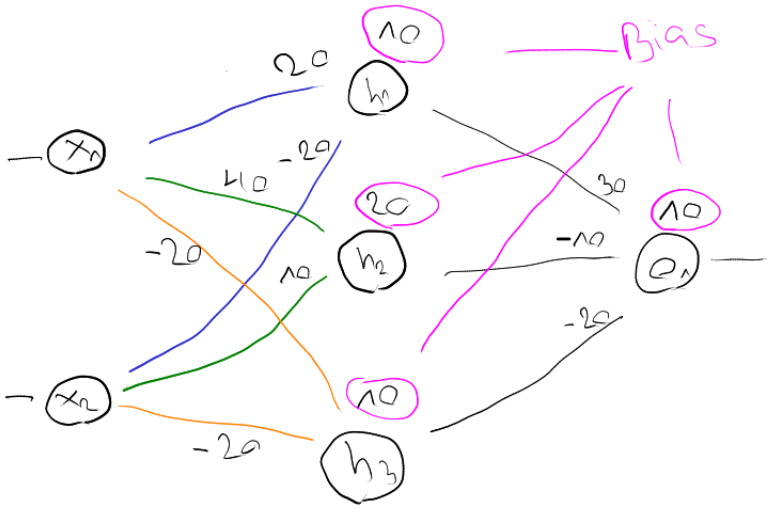
$$W_{Hidden} = \begin{pmatrix} 10 & 20 & -20 \\ 20 & 40 & 10 \\ 10 & -20 & -20 \end{pmatrix}$$

$$W_{Output} = \begin{pmatrix} 10 & 30 & -10 & -20 \end{pmatrix}$$

Aktivierungsfunktion g(z), die bei allen Neuronen gleich ist:

$$g(z) = \begin{cases} 1 & z \leq -10 \\ 0 & z \geq 10 \\ 0.5 & sonst \end{cases}$$

(a) (4 Punkte) Zeichnen Sie das beschriebene neuronale Netzwerk mit allen Neuronen und deren Verbindungen auf. Notieren Sie Gewichte und Bias an die jeweiligen Verbindungen. Zeichnen Sie sauber, nicht leserliche Skizzen werden nicht gewertet.



Lösung:

(b) (8 Punkte) Erstellen Sie aus den gegebenen Vektoren x_1, x_2, x_3 eine Mini-Batch-Matrix. Berechnen Sie basierend auf diesem Input den Ouput des Netzwerks. Notieren Sie auf dem Weg ebenfalls die Präaktivitäten der verdeckten Neuronen, deren Aktivierungen und die Präaktivitäten des Ausgangs. Um die vollständige Punktzahl zu erhalten, müssen alle Schritte mittels Matrix-Operationen berechnet werden.

$$\vec{x}_{(1)} = [0,0], \quad \vec{x}_{(2)} = [0,1], \quad \vec{x}_{(3)} = [1,0], \quad \vec{x}_{(4)} = [1,1]$$

Lösung:

$$PRE_{hidden} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} * \begin{pmatrix} 10 & 20 & 10 \\ 20 & 40 & -20 \\ -20 & 10 & -20 \end{pmatrix} = \begin{pmatrix} 10 & 20 & 10 \\ -10 & 30 & -10 \\ 30 & 60 & -10 \\ 10 & 70 & -30 \end{pmatrix}$$

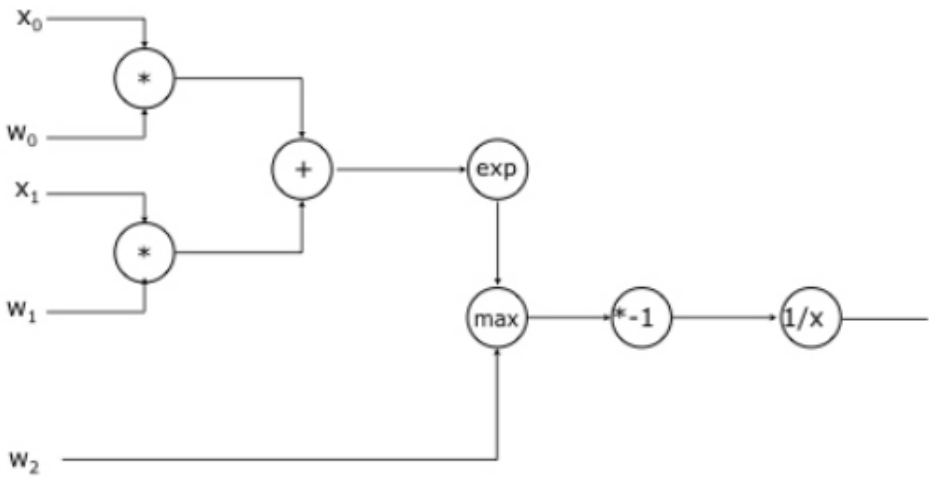
$$OUT_{hidden} = g(PRE_{hidden}) = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

$$PRE_{out} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} 10 \\ 30 \\ -10 \\ -20 \end{pmatrix} = \begin{pmatrix} 10 \\ 20 \\ -10 \\ -10 \end{pmatrix}$$

$$OUT = g(PRE_{out}) = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

4. (12 Punkte) Backpropagation

Berechnen Sie die partiellen Ableitungen $\frac{\partial out}{\partial w_0}$, $\frac{\partial out}{\partial w_1}$ und $\frac{\partial out}{\partial w_2}$ mittels Backpropagation in folgendem gegebenen Computational Graph:

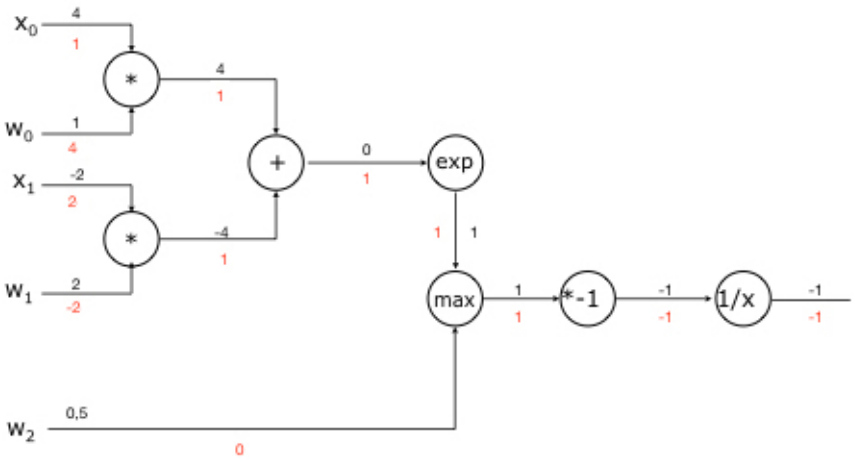


Dabei haben die Variablen folgende Werte:

- $x_0 = 4 \quad w_0 = 1$
- $x_1 = -2 \quad w_1 = 2$
- $w_2 = 0.5$

Hinweis: Exponentialfunktion: $exp = e^x$ und die Ableitung $\frac{d}{dx}(e^x) = e^x$

Lösung:



5. (24 Punkte) Convolution

Gegeben sind exemplarisch die Werte von drei Kanälen in der linken unteren Ecke einer Aktivierungsschicht A^l. Die vollständigen Dimensionen des Volumens A^l beträgt $16x16x32$. Im Netzwerk wird ReLU als Aktivierungsfunktion verwendet und nach jeder Faltung durchgeführt.

$$A_1^l = \begin{pmatrix} \dots & \dots & \dots \\ 9 & 9 & \dots \\ 2 & 8 & \dots \\ 1 & 3 & \dots \end{pmatrix}, \quad A_2^l = \begin{pmatrix} \dots & \dots & \dots \\ 9 & 9 & \dots \\ 3 & 8 & \dots \\ 2 & 5 & \dots \end{pmatrix}, \quad A_3^l = \begin{pmatrix} \dots & \dots & \dots \\ 6 & 6 & \dots \\ 3 & 8 & \dots \\ 1 & 5 & \dots \end{pmatrix}$$

Folgende Werte haben die Gewichte der Kernel K_{ij} (Filter) und die Bias b_i des folgenden Faltungsschichtneurons $CONV^{l+1}$ (*convolutionallayer*).

- i: Index für den Ausgabekanal (*outputfeaturemap*)
- j: Index für den Eingabekanal (hier A_1^l, A_2^l, A_3^l)

$$K_{11} = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, \quad K_{12} = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}, \quad K_{13} = \begin{pmatrix} -1 & 0 & 2 \\ -2 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}, \quad b_1 = 1$$

$$K_{21} = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}, \quad K_{22} = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}, \quad K_{23} = \begin{pmatrix} -2 & -2 & -2 \\ 0 & 0 & 0 \\ 2 & 2 & 2 \end{pmatrix}, \quad b_2 = -1$$

Die Kernel werden mit einer Stride $S = 1$ verschoben. Ferner wird Zero-Padding zur Erhaltung der räumlichen Struktur verwendet. Die Dimension in Breite und Höhe von A^l entspricht also der von A^{l+1}.

(a) (2 Punkte) Welchen Wert hat das Zero-Padding P um die räumlichen Struktur zu erhalten?

Lösung: P=1

(b) (4 Punkte) In der Faltungsschicht $CONV^{l+1}$ befinden sich ingesamt 8 Kernel ($i \in 1, \dots, 8$). Welche Dimension hat das Ausgabevolumen A^{l+1} und wieviele lernbare Parameter enthält die Schicht $CONV^{l+1}$?

Lösung: $Output_{DIM} = 16x16x8$
$CONV1_{PARA} = 3*3*3*8+8 = 224$

(c) (18 Punkte) Berechnen Sie konventionell (nicht vektorisiert) alle Werte von A^{l+1} des Neurons $CONV^{l+1}$, die mit den gegebenen Informationen berechenbar sind. Achten Sie auf eine klare Darstellung der Vorgehensweise und des Rechenweges. Nicht nachvollziehbare Antworten können nicht gewertet werden.

Lösung: Kernel 1: Position 1:

$$R*K_{11}=\begin{pmatrix}0&9&9\\0&2&8\\0&1&3\end{pmatrix}*\begin{pmatrix}-1&0&1\\-2&0&2\\-1&0&1\end{pmatrix}=28, G*K_{12}=\begin{pmatrix}0&9&9\\0&3&8\\0&2&5\end{pmatrix}*\begin{pmatrix}-1&0&1\\-1&0&1\\-1&0&1\end{pmatrix}=22$$

$$B*K_{13}=\begin{pmatrix}0&6&6\\0&3&8\\0&1&5\end{pmatrix}*\begin{pmatrix}-1&0&2\\-2&0&1\\-1&0&1\end{pmatrix}=25 \quad Akt\ K1_{Pos1}=28+22+25+1=76$$

Position 2:

$$R*K_{11}=\begin{pmatrix}0&2&8\\0&1&3\\0&0&0\end{pmatrix}*\begin{pmatrix}-1&0&1\\-2&0&2\\-1&0&1\end{pmatrix}=14, G*K_{12}=\begin{pmatrix}0&3&8\\0&2&5\\0&0&0\end{pmatrix}*\begin{pmatrix}-1&0&1\\-1&0&1\\-1&0&1\end{pmatrix}=13$$

$$B*K_{13}=\begin{pmatrix}0&3&8\\0&1&5\\0&0&0\end{pmatrix}*\begin{pmatrix}-1&0&2\\-2&0&1\\-1&0&1\end{pmatrix}=21 \quad Akt.\ K1_{Pos2}=14+13+21+1=49$$

Kernel 2: Position 1:

$$R*K_{21}=\begin{pmatrix}0&9&9\\0&2&8\\0&1&3\end{pmatrix}*\begin{pmatrix}-1&-2&-1\\0&0&0\\1&2&1\end{pmatrix}=-22, G*K_{22}=\begin{pmatrix}0&9&9\\0&3&8\\0&2&5\end{pmatrix}*\begin{pmatrix}-1&-1&-1\\0&0&0\\1&1&1\end{pmatrix}=-11$$

$$B*K_{23}=\begin{pmatrix}0&6&6\\0&3&8\\0&1&5\end{pmatrix}*\begin{pmatrix}-2&-2&-2\\0&0&0\\2&2&2\end{pmatrix}=-12 \quad Akt\ K1_{Pos1}=-22-11-12-1=-46$$

Position 2:

$$R*K_{21}=\begin{pmatrix}0&2&8\\0&1&3\\0&0&0\end{pmatrix}*\begin{pmatrix}-1&-2&-1\\0&0&0\\1&2&1\end{pmatrix}=-12, G*K_{22}=\begin{pmatrix}0&3&8\\0&2&5\\0&0&0\end{pmatrix}*\begin{pmatrix}-1&-1&-1\\0&0&0\\1&1&1\end{pmatrix}=-11$$

$$B*K_{23}=\begin{pmatrix}0&3&8\\0&1&5\\0&0&0\end{pmatrix}*\begin{pmatrix}-2&-2&-2\\0&0&0\\2&2&2\end{pmatrix}=-22 \quad Akt.\ K1_{Pos2}=-12+(-11)+(-22)-1=-46$$

Feature maps:

$$A_1^{l+1}=\begin{pmatrix}76\\49\end{pmatrix}, ReLU(A_2^{l+1})=\begin{pmatrix}-46\\-46\end{pmatrix}, A_2^{l+1}=\begin{pmatrix}0\\0\end{pmatrix}$$

6. (16 Punkte) Optimizer

   (a) (12 Punkte) Erläutern Sie das Gradientenabstiegsverfahren mit Momentum. Geben Sie sowohl eine formale Definition des Verfahrens an als auch eine anschauliche Beschreibung an. Treffen Sie eine Aussage über die Einstellung der Hyperparamter. Nennen Sie die Vorteile und Eigenschaften, welche das Verfahrens gegenüber dem Gradientenabstiegsverfahren ohne Momentum hat. Welches typische Verhalten zeigt das Verfahren während des Optimierungsprozesses?

Lösung: Bei SGD wird das Update der lernbaren Parameter über die aktuelle Position a_t und den Gradienten dieser Position $F(a_x)$ berechnet, wobei die Gradienten nur zu einem Faktor γ eingehen.

$$a_{t+1} = a_t - \gamma \nabla F(a_t)$$

SGD mit Momentum erweitert dieses Verfahren durch eine Geschwindigkeit v und eine Reibung ρ, die - anstatt der aktuellen Position selbst - mit den Gradienten der aktuellen Position modifiziert werden. Dieses entspricht dem Momentum-Term, welcher einem exponentiell gleitenden Mittelwert über die Zeit t darstellt:

$$v_{t+1} = \rho v_t - \gamma \nabla F(a_x)$$
$$a_{x+1} = a_x + v_{t+1}$$

Anschaulich kann sich das SGD+Momentum-Verfahren durch einen schweren Ball der eine Hügellandschaft hinab rollt vorgestellt werden. Der Ball wird bei monotonen Steigungsrichtung Geschwindigkeitaufnehmen und bei einem Wechsel der Richtung die Geschwindigkeit reduzieren. Durch diese Eigenschaften kann es passieren, dass SGD+Momentum über das optimale Minimum zunächst hinausschieSSt und dann seine Route korrigieren muss. Jedoch entstehen durch den Geschwindigkeitsaufbau folgende Vorteile, die diese Eigenschaft überwiegen:

- Die Chance in lokalen Minima und insbesondere Sattelpunkten hängen zu bleiben wird signifikant reduziert
- Das Verfahren konvergiert schneller, da die Geschwindigkeit in monotone Richtungen aufgebaut wird. Ferner wird durch diese Eigenschaft ein Oszillation des Updates, beispielsweise in Schluchten, reduziert.

Ein typischer Wert für den Hyperparameter ρ liegt zwischen 0.9 und 0.99.

(b) (4 Punkte) Erläutern Sie anhand einer geometrischen Darstellung oder der arithmetischen Formeln der Verfahren den Unterschied zwischen Momentum und Nesterov-Momentum.

Lösung: