Habib Mubarak, Eraycan Gökcan, Basel Ali

Aufgabenblatt 3

Hinweis: Für die Ausführung vom Code, gebe
./HPCA_PC_MLP /home/…(absoluter Dateipfad)…/mnist_datasets
In das Terminal ein.

Aufgabe 2.1
- Depth of a network:
Die Tiefe eines Netzwerks ist die Anzahl der Schichten im Netzwerk.

- Width of a layer:
Wie Schicht bzw. Layer definiert ist, variiert sehr oft. In dieser Vorlesung definieren wir Schicht als eine Menge von Neuronen. Sei es die Eingabe-Neuronen, die versteckten Neuronen, oder die Ausgabe-Neuronen. Die Breite einer Schicht ist die Anzahl der Neuronen in der Schicht.

- Training- vs. Testing:
Training eines Modells ist der Prozess, wo mithilfe von Trainingsdaten die Parameter des Modells aktualisiert werden und das Modell sodurch lernt, generell Muster zu erkennen.
Das Testen eines Modells ist die abschließende Bewertung des Modells, indem die Leistung (Mustererkennung) des Modells auf einen vom Trainingsdatensatz disjunkten Testdatensatz betrachtet wird.
Datensätze werden somit meistens in Trainings- und Testdatensätze unterteilt.

- Batch size:
Das ist die Anzahl an Samples, die man durchgeht, bevor die Parameter des Modells aktualisiert werden.
Wie groß man sie wählt, hat Einfluss darauf, wie schnell die Fortschritte sind, wie viel Speicher genutzt wird und wie hoch die Varianz im Gradienten ist.

- Epoch:
Ein Epoch beschreibt einen Durchlauf durch den ganzen Trainingsdatensatz.
Da oft nicht genug Daten zur Verfügung stehen, wird das Modell mit mehreren Epochen trainiert, wobei jedesmal die Gewichte aktualisiert werden und das Modell immer mehr lernt, Muster in den Daten zu erkennen.
Manchmal wird in einem Epoch auch eine Testings- oder Validierungsphase durchgegangen, aber das passiert mit bisher ungesehenen Daten.

- Feed forward:
Das ist die Weitergabe der Eingabedaten durch die Schichten des MLPs (oder andere ähnliche Modelle) bis zur Ausgabe. Die berechneten Ausgaben jeder Schicht werden an die nächste Schicht weitergegeben.

- Backpropagation:
Backpropagation ist der Prozess, in dem der Gradient bezüglich den Gewichten berechnet wird, sodass diese später aktualisiert werden können. Es wird herausgefunden, wie sehr die Gewichte den Loss bzw. die Fehler beeinflussen.

- Loss:
Der Loss beschreibt, wie gut oder schlecht die Ausgabe bzw. die Vorhersage des Modells ist. Die Loss-Funktion hilft uns, die Gewichte im Netzwerk zu aktualisieren, in dem man die partielle Ableitung bzw. den Gradienten berechnet.

- Learning rate:
Die Lernrate ist ein Wert, der dafür sorgt, wie stark die Gewichte aktualisiert werden. Das wird gemacht, indem man sie mit dem Wert des Gradienten multipliziert, so dass man Einfluss auf die Aktualisierung durch die Loss-Funktion hat.

Aufgabe 2.4
Wir vergleichen zwei Vorgänge:
Transpose() → MatVecMul() vs. MatTransposeVecMul()

(siehe unten für Benchmark-Resultate)

Was sind die Unterschiede in der Implementierung?
In MatTransposeVecMul() vermeiden wir, die matrix-Eingabe explizit zu transponieren. Stattdessen iterieren wir bei der Matrix-Vektor-Multiplikation in der matrix-Eingabe "transponiert", also Spalte für Spalte.
Wenn man sich die Implementierung von CalculateHiddenDeltas() anschaut...:

```
void CalculateHiddenDeltas(const std::vector<float>& nextLayerDeltas,
                           const std::vector<std::vector<float>>& weights)
{
  std::vector<std::vector<float>> weights_transpose;
  Utils::Transpose(weights, weights_transpose);
  Utils::MatVecMul(weights_transpose, nextLayerDeltas, deltas_);
.,,}
```

...sieht man, dass weights_transpose temporär erstellt wurde und das nur für die Berechnung der Matrix-Vektor-Multiplikation.

Sonst braucht man sie nicht mehr. Wenn man stattdessen nur MatTransposeVecMul() aufruft, braucht man nicht so eine Matrix zu erstellen:

```
void CalculateHiddenDeltas(const std::vector<float>& nextLayerDeltas,
                           const std::vector<std::vector<float>>& weights)
{
  Utils::MatTransposeVecMul(weights, nextLayerDeltas, deltas_);
```

Warum ist die Laufzeit von MatTransposeVecMul() besser?
Nachteile (eigentlich nicht) von MatTransposeVecMul():

- Da wir matrix spaltenweise iterieren, gibt es bei der Iteration theoretisch mehr Chancen für Cache-Misses.
  Aber: Die transponierte Iteration passiert auch in Transpose(), also hebt sich dieser Nachteil auf.

Vorteile von MatTransposeVecMul():

- Man braucht nicht weights_transpose (temporär) zu erstellen und speichern => Zeit- und Speichereffizienter.
- Das Transponieren der Struktur der Matrix in Transpose() braucht Zeit und eventuell temporär mehr Speicher, was somit weiterhin noch mehr Zeit brauchen kann, wenn Speicherplatz allokiert werden muss.
- Das transponierte Einfügen der Daten in die Matrix benötigt Zeit.

Aufgabe 2.4 & 2.5 – Benchmark-Resultate
Für alle Resultate war der Code im Release-Modus und mit Optimierungslevel O3 kompiliert.

Transpose() → MatVecMul() *vs.* *MatTransposeVecMul():*

Für diesen Vergleich haben wir folgende Konfigurationen benutzt:

```
// with using a hidden layer
  std::vector<size_t> topology = {784, 800, 10};
  std::vector<std::string> activations = {"None", "LeakyReLU", "Softmax"};

---

  float learningRate_ = 0.001f;
```

Transpose() → MatVecMul()	*MatTransposeVecMul()*
- Keine signifikanten Unterschiede in Accuracy und Loss (das macht Sinn, denn beide Methoden haben dieselbe Ausgabe	- Deutlich kürzere Laufzeit pro Epoch in der Trainingsphase => MatTransposeVecMul() ist Zeiteffizienter

```
Ergebnisse für

    // mit der Transpose()- und MatVecMul()-Funktion
    std::vector<std::vector<float>> weights_transpose;
    Utils::Transpose(weights, weights_transpose);
    Utils::MatVecMul(weights_transpose, nextLayerDeltas, deltas_);

-------------------------------------------------
[INFO] Epoch 1
[INFO] Training finished in 46 seconds.
[INFO] Accuracy Training: 87.91%
[INFO] Loss Training: 0.49
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 91.77%
[INFO] Loss Testing: 0.28

-------------------------------------------------
[INFO] Epoch 2
[INFO] Training finished in 46 seconds.
[INFO] Accuracy Training: 92.78%
[INFO] Loss Training: 0.26
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 94.01%
[INFO] Loss Testing: 0.21
```

```
------------------------------------------------
[INFO] Epoch 3
[INFO] Training finished in 46 seconds.
[INFO] Accuracy Training: 94.45%
[INFO] Loss Training: 0.20
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 94.86%
[INFO] Loss Testing: 0.18

------------------------------------------------
[INFO] Epoch 4
[INFO] Training finished in 47 seconds.
[INFO] Accuracy Training: 95.44%
[INFO] Loss Training: 0.16
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 95.72%
[INFO] Loss Testing: 0.15

------------------------------------------------
[INFO] Epoch 5
[INFO] Training finished in 46 seconds.
[INFO] Accuracy Training: 96.17%
[INFO] Loss Training: 0.14
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 96.31%
[INFO] Loss Testing: 0.13

------------------------------------------------
[INFO] Epoch 6
[INFO] Training finished in 46 seconds.
[INFO] Accuracy Training: 96.74%
[INFO] Loss Training: 0.12
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 96.67%
[INFO] Loss Testing: 0.11

------------------------------------------------
[INFO] Epoch 7
[INFO] Training finished in 47 seconds.
[INFO] Accuracy Training: 97.17%
[INFO] Loss Training: 0.10
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 96.90%
[INFO] Loss Testing: 0.11

------------------------------------------------
[INFO] Epoch 8
[INFO] Training finished in 46 seconds.
[INFO] Accuracy Training: 97.49%
[INFO] Loss Training: 0.09
```

```
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 97.25%
[INFO] Loss Testing: 0.10

------------------------------------------------
[INFO] Epoch 9
[INFO] Training finished in 46 seconds.
[INFO] Accuracy Training: 97.76%
[INFO] Loss Training: 0.08
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 97.32%
[INFO] Loss Testing: 0.09

------------------------------------------------
[INFO] Epoch 10
[INFO] Training finished in 46 seconds.
[INFO] Accuracy Training: 97.98%
[INFO] Loss Training: 0.07
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 97.49%
[INFO] Loss Testing: 0.09
```

Ergebnisse für:

```
    // mit der MatTransposeVecMul()-Funktion
    Utils::MatTransposeVecMul(weights, nextLayerDeltas, deltas_);

[INFO] Epoch 1
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 87.49%
[INFO] Loss Training: 0.49
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 91.90%
[INFO] Loss Testing: 0.28

------------------------------------------------
[INFO] Epoch 2
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 92.70%
[INFO] Loss Training: 0.26
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 93.96%
[INFO] Loss Testing: 0.21

------------------------------------------------
[INFO] Epoch 3
[INFO] Training finished in 39 seconds.
```

```
[INFO] Accuracy Training: 94.34%
[INFO] Loss Training: 0.20
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 95.00%
[INFO] Loss Testing: 0.17

---------------------------------------------------
[INFO] Epoch 4
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 95.36%
[INFO] Loss Training: 0.16
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 95.74%
[INFO] Loss Testing: 0.15

---------------------------------------------------
[INFO] Epoch 5
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 96.07%
[INFO] Loss Training: 0.14
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 96.18%
[INFO] Loss Testing: 0.13

---------------------------------------------------
[INFO] Epoch 6
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 96.68%
[INFO] Loss Training: 0.12
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 96.70%
[INFO] Loss Testing: 0.11

---------------------------------------------------
[INFO] Epoch 7
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 97.03%
[INFO] Loss Training: 0.10
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 96.84%
[INFO] Loss Testing: 0.10

---------------------------------------------------
[INFO] Epoch 8
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 97.41%
[INFO] Loss Training: 0.09
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 97.01%
[INFO] Loss Testing: 0.10
```

```
------------------------------------------------
[INFO] Epoch 9
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 97.67%
[INFO] Loss Training: 0.08
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 97.21%
[INFO] Loss Testing: 0.09

------------------------------------------------
[INFO] Epoch 10
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 97.90%
[INFO] Loss Training: 0.07
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 97.47%
[INFO] Loss Testing: 0.09
```

Mit (LeakyReLU) Hidden Layer vs. Ohne (LeakyReLU) Hidden Layer:

Für diesen Vergleich und alle anderen kommenden Vergleiche haben wir die MatTransposeVecMul()-Funktion angewendet.

Mit (LeakyReLU) Hidden Layer	*Ohne (LeakyReLU) Hidden Layer*
- Deutlich bessere Accuracy und weniger Loss => die Hidden Layer lohnen sich für die Vorhersagen	- Extrem schnelle Laufzeit, da wir sehr viele Neuronen nicht mehr verwenden

```
Ergebnisse für:

  // with using a hidden layer
  std::vector<size_t> topology = {784, 800, 10};
  std::vector<std::string> activations = {"None", "LeakyReLU", "Softmax"};

[INFO] Epoch 1
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 87.49%
[INFO] Loss Training: 0.49
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 91.90%
[INFO] Loss Testing: 0.28

------------------------------------------------
[INFO] Epoch 2
```

```
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 92.70%
[INFO] Loss Training: 0.26
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 93.96%
[INFO] Loss Testing: 0.21

-------------------------------------------------
[INFO] Epoch 3
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 94.34%
[INFO] Loss Training: 0.20
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 95.00%
[INFO] Loss Testing: 0.17

-------------------------------------------------
[INFO] Epoch 4
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 95.36%
[INFO] Loss Training: 0.16
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 95.74%
[INFO] Loss Testing: 0.15

-------------------------------------------------
[INFO] Epoch 5
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 96.07%
[INFO] Loss Training: 0.14
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 96.18%
[INFO] Loss Testing: 0.13

-------------------------------------------------
[INFO] Epoch 6
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 96.68%
[INFO] Loss Training: 0.12
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 96.70%
[INFO] Loss Testing: 0.11

-------------------------------------------------
[INFO] Epoch 7
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 97.03%
[INFO] Loss Training: 0.10
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 96.84%
```

```
[INFO] Loss Testing: 0.10

-------------------------------------------------
[INFO] Epoch 8
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 97.41%
[INFO] Loss Training: 0.09
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 97.01%
[INFO] Loss Testing: 0.10

-------------------------------------------------
[INFO] Epoch 9
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 97.67%
[INFO] Loss Training: 0.08
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 97.21%
[INFO] Loss Testing: 0.09

-------------------------------------------------
[INFO] Epoch 10
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 97.90%
[INFO] Loss Training: 0.07
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 97.47%
[INFO] Loss Testing: 0.09
```

```
Ergebnisse für:

  // without using a hidden layer
  std::vector<size_t> topology = {784, 10};
  std::vector<std::string> activations = {"None", "Softmax"};

[INFO] Epoch 1
[INFO] Training finished in 0 seconds.
[INFO] Accuracy Training: 86.88%
[INFO] Loss Training: 0.53
[INFO] Testing finished in 0 seconds.
[INFO] Accuracy Testing: 90.44%
[INFO] Loss Testing: 0.36

-------------------------------------------------
[INFO] Epoch 2
[INFO] Training finished in 0 seconds.
[INFO] Accuracy Training: 90.12%
[INFO] Loss Training: 0.36
```

```
[INFO] Testing finished in 0 seconds.
[INFO] Accuracy Testing: 91.22%
[INFO] Loss Testing: 0.33

-------------------------------------------------
[INFO] Epoch 3
[INFO] Training finished in 0 seconds.
[INFO] Accuracy Training: 90.72%
[INFO] Loss Training: 0.33
[INFO] Testing finished in 0 seconds.
[INFO] Accuracy Testing: 91.62%
[INFO] Loss Testing: 0.31

-------------------------------------------------
[INFO] Epoch 4
[INFO] Training finished in 0 seconds.
[INFO] Accuracy Training: 91.14%
[INFO] Loss Training: 0.32
[INFO] Testing finished in 0 seconds.
[INFO] Accuracy Testing: 91.60%
[INFO] Loss Testing: 0.30

-------------------------------------------------
[INFO] Epoch 5
[INFO] Training finished in 0 seconds.
[INFO] Accuracy Training: 91.44%
[INFO] Loss Training: 0.31
[INFO] Testing finished in 0 seconds.
[INFO] Accuracy Testing: 91.98%
[INFO] Loss Testing: 0.29

-------------------------------------------------
[INFO] Epoch 6
[INFO] Training finished in 0 seconds.
[INFO] Accuracy Training: 91.62%
[INFO] Loss Training: 0.30
[INFO] Testing finished in 0 seconds.
[INFO] Accuracy Testing: 91.92%
[INFO] Loss Testing: 0.29

-------------------------------------------------
[INFO] Epoch 7
[INFO] Training finished in 0 seconds.
[INFO] Accuracy Training: 91.74%
[INFO] Loss Training: 0.29
[INFO] Testing finished in 0 seconds.
[INFO] Accuracy Testing: 92.15%
[INFO] Loss Testing: 0.29

-------------------------------------------------
```

```
[INFO] Epoch 8
[INFO] Training finished in 0 seconds.
[INFO] Accuracy Training: 91.88%
[INFO] Loss Training: 0.29
[INFO] Testing finished in 0 seconds.
[INFO] Accuracy Testing: 92.02%
[INFO] Loss Testing: 0.28

-------------------------------------------------
[INFO] Epoch 9
[INFO] Training finished in 0 seconds.
[INFO] Accuracy Training: 92.01%
[INFO] Loss Training: 0.29
[INFO] Testing finished in 0 seconds.
[INFO] Accuracy Testing: 92.05%
[INFO] Loss Testing: 0.28

-------------------------------------------------
[INFO] Epoch 10
[INFO] Training finished in 0 seconds.
[INFO] Accuracy Training: 92.07%
[INFO] Loss Training: 0.28
[INFO] Testing finished in 0 seconds.
[INFO] Accuracy Testing: 92.28%
[INFO] Loss Testing: 0.28
```

Learning Rate = 0.001 vs. Learning Rate = 0.01 vs. Learning Rate = 0.0001:

Learning Rate = 0.001	*Learning Rate = 0.01*	*Learning Rate = 0.0001*
- Zwischen 0.01 und 0.0001 von Accuracy und Loss her	- Deutlich beste Accuracy und wenig Loss	- Deutlich schlechteste Accuracy und großer Loss

```
Ergebnisse für:

  float learningRate_ = 0.001f;

[INFO] Epoch 1
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 87.49%
[INFO] Loss Training: 0.49
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 91.90%
[INFO] Loss Testing: 0.28
```

```
------------------------------------------------
[INFO] Epoch 2
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 92.70%
[INFO] Loss Training: 0.26
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 93.96%
[INFO] Loss Testing: 0.21

------------------------------------------------
[INFO] Epoch 3
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 94.34%
[INFO] Loss Training: 0.20
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 95.00%
[INFO] Loss Testing: 0.17

------------------------------------------------
[INFO] Epoch 4
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 95.36%
[INFO] Loss Training: 0.16
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 95.74%
[INFO] Loss Testing: 0.15

------------------------------------------------
[INFO] Epoch 5
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 96.07%
[INFO] Loss Training: 0.14
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 96.18%
[INFO] Loss Testing: 0.13

------------------------------------------------
[INFO] Epoch 6
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 96.68%
[INFO] Loss Training: 0.12
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 96.70%
[INFO] Loss Testing: 0.11

------------------------------------------------
[INFO] Epoch 7
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 97.03%
[INFO] Loss Training: 0.10
```

```
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 96.84%
[INFO] Loss Testing: 0.10

-------------------------------------------------
[INFO] Epoch 8
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 97.41%
[INFO] Loss Training: 0.09
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 97.01%
[INFO] Loss Testing: 0.10

-------------------------------------------------
[INFO] Epoch 9
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 97.67%
[INFO] Loss Training: 0.08
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 97.21%
[INFO] Loss Testing: 0.09

-------------------------------------------------
[INFO] Epoch 10
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 97.90%
[INFO] Loss Training: 0.07
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 97.47%
[INFO] Loss Testing: 0.09
```

```
Ergebnisse für:

  float learningRate_ = 0.01f;

[INFO] Epoch 1
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 93.51%
[INFO] Loss Training: 0.22
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 97.12%
[INFO] Loss Testing: 0.09

-------------------------------------------------
[INFO] Epoch 2
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 97.37%
[INFO] Loss Training: 0.09
[INFO] Testing finished in 5 seconds.
```

```
[INFO] Accuracy Testing: 97.34%
[INFO] Loss Testing: 0.09

------------------------------------------------
[INFO] Epoch 3
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 98.18%
[INFO] Loss Training: 0.06
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 98.03%
[INFO] Loss Testing: 0.06

------------------------------------------------
[INFO] Epoch 4
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 98.76%
[INFO] Loss Training: 0.04
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 98.03%
[INFO] Loss Testing: 0.07

------------------------------------------------
[INFO] Epoch 5
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 99.21%
[INFO] Loss Training: 0.03
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 98.17%
[INFO] Loss Testing: 0.06

------------------------------------------------
[INFO] Epoch 6
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 99.36%
[INFO] Loss Training: 0.02
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 98.03%
[INFO] Loss Testing: 0.06

------------------------------------------------
[INFO] Epoch 7
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 99.67%
[INFO] Loss Training: 0.01
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 98.00%
[INFO] Loss Testing: 0.07

------------------------------------------------
[INFO] Epoch 8
```

```
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 99.82%
[INFO] Loss Training: 0.01
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 98.25%
[INFO] Loss Testing: 0.06

-------------------------------------------------
[INFO] Epoch 9
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 99.92%
[INFO] Loss Training: 0.01
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 98.39%
[INFO] Loss Testing: 0.06

-------------------------------------------------
[INFO] Epoch 10
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 99.96%
[INFO] Loss Training: 0.00
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 98.34%
[INFO] Loss Testing: 0.06
```

```
Ergebnisse für:

  float learningRate_ = 0.0001f;

[INFO] Epoch 1
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 72.57%
[INFO] Loss Training: 1.42
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 85.22%
[INFO] Loss Testing: 0.74

-------------------------------------------------
[INFO] Epoch 2
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 86.12%
[INFO] Loss Training: 0.60
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 88.31%
[INFO] Loss Testing: 0.47
```

```
------------------------------------------------
[INFO] Epoch 3
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 88.27%
[INFO] Loss Training: 0.45
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 89.52%
[INFO] Loss Testing: 0.40

------------------------------------------------
[INFO] Epoch 4
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 89.32%
[INFO] Loss Training: 0.39
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 90.29%
[INFO] Loss Testing: 0.36

------------------------------------------------
[INFO] Epoch 5
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 89.98%
[INFO] Loss Training: 0.36
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 90.66%
[INFO] Loss Testing: 0.33

------------------------------------------------
[INFO] Epoch 6
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 90.53%
[INFO] Loss Training: 0.34
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 91.16%
[INFO] Loss Testing: 0.32

------------------------------------------------
[INFO] Epoch 7
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 90.87%
[INFO] Loss Training: 0.33
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 91.46%
[INFO] Loss Testing: 0.30

------------------------------------------------
[INFO] Epoch 8
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 91.21%
[INFO] Loss Training: 0.31
```

```
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 91.79%
[INFO] Loss Testing: 0.29

------------------------------------------------
[INFO] Epoch 9
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 91.55%
[INFO] Loss Training: 0.30
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 92.02%
[INFO] Loss Testing: 0.28

------------------------------------------------
[INFO] Epoch 10
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 91.84%
[INFO] Loss Training: 0.29
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 92.35%
[INFO] Loss Testing: 0.27
```

LeakyReLU-Aktivierungsfunktion vs. TanH-Aktivierungsfunktion:

Die TanH-Aktivierungsfunktion wird meistens in den hidden Layers verwendet.

LeakyReLU-Aktivierungsfunktion	*TanH-Aktivierungsfunktion*
- Lernt schneller, also Accuracy wird besser mit wenigeren Epochen (Lernrate 0.0001) - Accuracy besser und Loss weniger	- Lernrate 0.001 ist auch eine bessere Wahl für TanH - Accuracy schlechter und Loss mehr

```
Ergebnisse für:

  // with using a hidden layer
  std::vector<size_t> topology = {784, 800, 10};
  std::vector<std::string> activations = {"None", "LeakyReLU", "Softmax"};

  float learningRate_ = 0.001f;
```

```
[INFO] Epoch 1
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 87.49%
[INFO] Loss Training: 0.49
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 91.90%
[INFO] Loss Testing: 0.28

------------------------------------------------
[INFO] Epoch 2
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 92.70%
[INFO] Loss Training: 0.26
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 93.96%
[INFO] Loss Testing: 0.21

------------------------------------------------
[INFO] Epoch 3
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 94.34%
[INFO] Loss Training: 0.20
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 95.00%
[INFO] Loss Testing: 0.17

------------------------------------------------
[INFO] Epoch 4
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 95.36%
[INFO] Loss Training: 0.16
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 95.74%
[INFO] Loss Testing: 0.15

------------------------------------------------
[INFO] Epoch 5
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 96.07%
[INFO] Loss Training: 0.14
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 96.18%
[INFO] Loss Testing: 0.13

------------------------------------------------
[INFO] Epoch 6
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 96.68%
[INFO] Loss Training: 0.12
[INFO] Testing finished in 5 seconds.
```

```
[INFO] Accuracy Testing: 96.70%
[INFO] Loss Testing: 0.11

-------------------------------------------------
[INFO] Epoch 7
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 97.03%
[INFO] Loss Training: 0.10
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 96.84%
[INFO] Loss Testing: 0.10

-------------------------------------------------
[INFO] Epoch 8
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 97.41%
[INFO] Loss Training: 0.09
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 97.01%
[INFO] Loss Testing: 0.10

-------------------------------------------------
[INFO] Epoch 9
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 97.67%
[INFO] Loss Training: 0.08
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 97.21%
[INFO] Loss Testing: 0.09

-------------------------------------------------
[INFO] Epoch 10
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 97.90%
[INFO] Loss Training: 0.07
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 97.47%
[INFO] Loss Testing: 0.09
```

```
Ergebnisse für:

  // with using a hidden layer and the TanH activation function instead of
LeakyReLU
  std::vector<size_t> topology= {784, 800, 10};
  std::vector<std::string> activations = {"None", "TanH", "Softmax"};

  float learningRate_ = 0.001f;

[INFO] Epoch 1
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 87.82%
[INFO] Loss Training: 0.47
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 91.47%
[INFO] Loss Testing: 0.31

-------------------------------------------------
[INFO] Epoch 2
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 91.27%
[INFO] Loss Training: 0.31
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 92.11%
[INFO] Loss Testing: 0.28

-------------------------------------------------
[INFO] Epoch 3
[INFO] Training finished in 40 seconds.
[INFO] Accuracy Training: 91.89%
[INFO] Loss Training: 0.28
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 92.30%
[INFO] Loss Testing: 0.27

-------------------------------------------------
[INFO] Epoch 4
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 92.52%
[INFO] Loss Training: 0.26
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 92.89%
[INFO] Loss Testing: 0.25

-------------------------------------------------
[INFO] Epoch 5
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 93.18%
[INFO] Loss Training: 0.24
```

```
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 93.44%
[INFO] Loss Testing: 0.23

-------------------------------------------------
[INFO] Epoch 6
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 93.76%
[INFO] Loss Training: 0.22
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 93.91%
[INFO] Loss Testing: 0.21

-------------------------------------------------
[INFO] Epoch 7
[INFO] Training finished in 40 seconds.
[INFO] Accuracy Training: 94.39%
[INFO] Loss Training: 0.20
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 94.63%
[INFO] Loss Testing: 0.19

-------------------------------------------------
[INFO] Epoch 8
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 94.85%
[INFO] Loss Training: 0.18
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 95.02%
[INFO] Loss Testing: 0.17

-------------------------------------------------
[INFO] Epoch 9
[INFO] Training finished in 40 seconds.
[INFO] Accuracy Training: 95.32%
[INFO] Loss Training: 0.17
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 95.23%
[INFO] Loss Testing: 0.17

-------------------------------------------------
[INFO] Epoch 10
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 95.77%
[INFO] Loss Training: 0.15
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 95.62%
[INFO] Loss Testing: 0.16
```

Ergebnisse für:

```
  // with using a hidden layer
  std::vector<size_t> topology = {784, 800, 10};
  std::vector<std::string> activations = {"None", "LeakyReLU", "Softmax"};

  float learningRate_ = 0.01f;
```

```
[INFO] Epoch 1
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 93.51%
[INFO] Loss Training: 0.22
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 97.12%
[INFO] Loss Testing: 0.09

-------------------------------------------------
[INFO] Epoch 2
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 97.37%
[INFO] Loss Training: 0.09
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 97.34%
[INFO] Loss Testing: 0.09

-------------------------------------------------
[INFO] Epoch 3
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 98.18%
[INFO] Loss Training: 0.06
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 98.03%
[INFO] Loss Testing: 0.06

-------------------------------------------------
[INFO] Epoch 4
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 98.76%
[INFO] Loss Training: 0.04
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 98.03%
[INFO] Loss Testing: 0.07

-------------------------------------------------
[INFO] Epoch 5
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 99.21%
[INFO] Loss Training: 0.03
```

```
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 98.17%
[INFO] Loss Testing: 0.06

-------------------------------------------------
[INFO] Epoch 6
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 99.36%
[INFO] Loss Training: 0.02
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 98.03%
[INFO] Loss Testing: 0.06

-------------------------------------------------
[INFO] Epoch 7
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 99.67%
[INFO] Loss Training: 0.01
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 98.00%
[INFO] Loss Testing: 0.07

-------------------------------------------------
[INFO] Epoch 8
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 99.82%
[INFO] Loss Training: 0.01
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 98.25%
[INFO] Loss Testing: 0.06

-------------------------------------------------
[INFO] Epoch 9
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 99.92%
[INFO] Loss Training: 0.01
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 98.39%
[INFO] Loss Testing: 0.06

-------------------------------------------------
[INFO] Epoch 10
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 99.96%
[INFO] Loss Training: 0.00
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 98.34%
[INFO] Loss Testing: 0.06
```

```
Ergebnisse für:

  // with using a hidden layer and the TanH activation function instead of
LeakyReLU
  std::vector<size_t> topology= {784, 800, 10};
  std::vector<std::string> activations = {"None", "TanH", "Softmax"};

  float learningRate_ = 0.01f;

[INFO] Epoch 1
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 90.08%
[INFO] Loss Training: 0.34
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 94.10%
[INFO] Loss Testing: 0.20

-------------------------------------------------
[INFO] Epoch 2
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 95.04%
[INFO] Loss Training: 0.17
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 95.87%
[INFO] Loss Testing: 0.14

-------------------------------------------------
[INFO] Epoch 3
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 96.70%
[INFO] Loss Training: 0.11
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 96.38%
[INFO] Loss Testing: 0.11

-------------------------------------------------
[INFO] Epoch 4
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 97.54%
[INFO] Loss Training: 0.08
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 97.05%
[INFO] Loss Testing: 0.09

-------------------------------------------------
[INFO] Epoch 5
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 98.03%
[INFO] Loss Training: 0.06
```

```
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 97.00%
[INFO] Loss Testing: 0.10

-------------------------------------------------
[INFO] Epoch 6
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 98.39%
[INFO] Loss Training: 0.05
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 97.60%
[INFO] Loss Testing: 0.08

-------------------------------------------------
[INFO] Epoch 7
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 98.77%
[INFO] Loss Training: 0.04
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 97.45%
[INFO] Loss Testing: 0.08

-------------------------------------------------
[INFO] Epoch 8
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 99.01%
[INFO] Loss Training: 0.03
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 97.62%
[INFO] Loss Testing: 0.08

-------------------------------------------------
[INFO] Epoch 9
[INFO] Training finished in 39 seconds.
[INFO] Accuracy Training: 99.22%
[INFO] Loss Training: 0.02
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 97.68%
[INFO] Loss Testing: 0.08

-------------------------------------------------
[INFO] Epoch 10
[INFO] Training finished in 40 seconds.
[INFO] Accuracy Training: 99.45%
[INFO] Loss Training: 0.02
[INFO] Testing finished in 5 seconds.
[INFO] Accuracy Testing: 97.90%
[INFO] Loss Testing: 0.07
```