

Aufgabe 4

Aufgabe 1: Grundelemente (18 Punkte)

1. Beschreiben Sie die Charakteristiken des “Convolutional Neural Network”.

Das Convolutional Neural Network ist ein künstliches neuronales Netzwerk, der für die Verarbeitung von Bildern und anderen mehrdimensionalen Datenstrukturen entwickelt wurde.

Gelernte Muster sind translationsinvariant: bedeutet, dass es die Muster in Bildern, welches es gelernt hat, überall erkennen kann. Somit braucht es wenig Trainingsdaten, um eine Generalisierbarkeit zu schaffen.

Lernen räumliche Hierarchien von Mustern: die erste Schicht lernt kleine lokale Muster, die zweite Schicht lernt aus der ersten Schicht und setzt diese in grössere Muster zusammen.

2. Beschreiben Sie alle Komponenten die in einem “CNN” benutzt werden.

Convolutional-Schicht: Ein Bild beinhaltet mehrere Merkmale, z.B. einzelne Linien, Formen oder Kanten. Die Convolutional-Schicht ist dafür zuständig, diese Merkmale in den Eingangsbilder zu erkennen und zu verarbeiten. Diese haben eine bestimmte Pixelgröße und scannen die Grafik nach und nach auf ihre Eigenschaften ab. Dabei entstehen grosse Datenmengen.

Nichtlinearität: Nach der C-Schicht wird die nichtlineare Aktivierungsfunktion angewendet.

Pooling-Schicht: Die Pooling-Schicht verarbeitet als nächstes die grossen Datenmengen und entscheidet, welche Daten das wichtig sind und welche nicht. Es filtert die Besten herauf und verwirft die Schwachen. Dies verbessert die Verarbeitungsgeschwindigkeit enorm. Hier gibt es das Max-Pooling oder das Average-Pooling als Herangehensweise.

Vollständig verbundene Schicht: diese Schicht verbindet die Ergebnisse der beiden Schichten miteinander. Dabei verknüpft es sowohl die eingegangenen als auch die ausgegangenen Werte aus beiden Schichten. Diese vollständig verknüpfte Schicht beinhaltet alle verbundenen Neuronen, die ausgewertet werden können und geben die Modellvorhersage aus.

Output Layer: Ausgabeschicht der Neuronen. Anzahl der Neuronen kann variieren.

3. Worin unterscheidet sich das Vorgehen des “CNN” von einem “Feed Forward Network”

Die CNN ist speziell für Bilderverarbeitung oder mehrdimensionalen Datenstrukturen konzipiert, mit den Convolutional- und Pooling-Schichten. Im Gegensatz dazu besteht ein Feed Forward Network aus vollständig verbundenen Schichten, bei denen jedes Neuron mit allen Neuronen in den vorherigen und nachfolgenden Schichten verbunden ist. Die Verbindungen dazwischen sind einfach gewichtet. Des Weiteren sind im CNN die Verbindungen zwischen den Neuronen gewichtet (werden durch die Faltungsschichten bestimmt). Zudem sind die Muster im CNN im Gegensatz zu FFN translationsinvariant. Des Weiteren ist die Anzahl der Parameter im CNN geringer, da die schlechten Daten herausgeworfen werden. Im FFN bleiben diese aber drin. Deshalb werden die FFN in der Breite oft angewendet, ausser bei mehrdimensionalen Datenstrukturen oder Bildverarbeitung.

4. Beschreiben Sie den Prozess der Information Extraktion in einem “Convolutional Neural Network”

Die Convolution-Schicht werden die Merkmale aus dem Input erfasst (Convolution), an die Pooling-Schicht weitergeben, welche dann die „Filterung“ der Daten vornimmt und die Dimension reduziert (Pooling). Zum Schluss werden die Pooling-Schichten in ein-dimensionalen Vektor gewandelt (Flattening). Diese werden in den vollständig verbundenen Layer verarbeitet mit allen Neuronen und es entsteht der Output Layer für die Vorhersage.

5. Wieso verwenden wir den "Pooling Layer"?

- Reduktion der Datenmenge/Rechenkomplexität durch Dimensionsreduktion
- Reduktion von Overfitting, da unwichtige Details verworfen werden
- Das Modell wird auf die relevanten Merkmale reduziert → bessere Generalisierung
- Es kann trotz Verschiebungen und leichten Variationen die Muster stets erkennen

6. Worin unterscheidet sich der "Max Pooling Layer" vom "Average Pooling Layer"?

Beim Max-Pooling wird im Pooling-Filter die grösste Aktivierung in einem Bereich ausgewählt und weiterverwendet. Dies ignoriert unwichtige Merkmale und hebt die Besten hervor. So wird die räumliche Dimension reduziert und die Translationsinvarianz erhöht.

Im Average-Pooling hingegen wird der Durchschnitt der Aktivierungen in einem Bereich bestimmt. Hier werden alle Merkmale gleichmässig gewichtet und man erhält glattere Resultate.

Aufgabe 2: Cats and Dogs (15 Punkte)

Auf Moodle haben Sie einen Code zur Verfügung gestellt bekommen, welche Ihnen den Datensatz Hund und Katzen von Kaggle aufbereitet. In diesem Fall verwenden wir nur einen kleinen Teil des Datensatzes. Sie sollen nun folgende Aufgaben lösen:

1. Erstellen Sie ein Netzwerk mit drei "Convolutional Layers" und den entsprechenden "Max Pooling" Layers. Der erste "Convolutional Layer" soll 32 Filters enthalten, der zweite 64 und der dritte 128.

Siehe Python-Code

2. Interpretieren Sie den Output von "model.summary".

Es gibt insgesamt 19'034'177 Parameter im Modell, wobei die meisten (18'940'416) aus dem Dense stammen. Im CNN gibt es anfangs nicht viele wichtige Parameter. Im Pooling gibt es keine Parameter. Der Output-Shape wurde dort zudem jeweils halbiert. Die Anzahl der Layers wird nach dem Flatten auf von 4 auf 2 reduziert. Der Output am Schluss ist nur noch im Shape (None, 1).

3. Führen Sie das Modell durch. Beschreiben Sie die Schritte die notwendig waren.

Die Daten müssen korrekt vorbereitet werden (Test, Training- und Validierungssets). Dies wurde im ersten Teil des Codes bereits zur Verfügung gestellt. Anschliessend wird das Modell definiert.

Im Modell werden zuerst die Convolutional Layers mit der Aktivierungsfunktion bestimmt, inkl. Input-Shape. Danach wird das Max-Pooling durchgeführt. Im Anschluss wird die Pooling-Schicht erstellt. Danach kommt das Flattening, (vollständig verbundene Schicht), welches die Neuronen erstellt. Zum Schluss werden diese im Dense durchlaufen, wobei der Output Layer entsteht. Anschliessend wird das Modell durch die Verlustfunktion und die Metriken ausgewertet. Das Modell kann jetzt Trainiert werden. Zuletzt wird das Modell evaluiert.

4. Zeichnen Sie die Verlustfunktion und interpretieren Sie das Resultat.

Die Verlustfunktion bei Validation schwankt stark und liegt zwischen 1 und 2.5. Dies ist viel zu hoch. Eine niedrige Verlustfunktion wäre das Ziel. Somit weist unser Modell eine zu hohe Fehlerquote auf. (Zeichnung siehe Python-Code)

5. Zeichnen Sie die Metrik und interpretieren Sie das Resultat.

Die Validation-Accuracy liegt zwischen 69%- 74%. Die Train-Accuracy liegt weit darüber bei fast 100%. Die Validierung liegt somit deutlich unter dem Train. Dies deutet auf Overfitting hin. Das heisst, dass das Modell zu stark auf die Trainingsdaten angepasst ist und es nicht schafft, die neuen Muster zu generalisieren. Hier müsste man jetzt das Modell anpassen und Overfitting reduzieren.

(Zeichnung siehe Python-Code)