NNW-Übung 8

Dokumentieren Sie in Ihrer Gruppe die Ergebnisse und insbesondere die Antworten zu den Fragen (bitte Fragennummer angeben) entweder als Markup-Zellen im Jupyter-Notebooks oder als Kommentare.

1 ConvNets

Das fully-connected Netz vom vorigen Aufgabenblatt soll jetzt durch ein Convolutional (Neural) Network (ConvNet, CNN) ersetzt werden. Kopieren Sie am besten das Notebook (bzw. speichern Sie es unter einem neuen Namen) und ändern es ab.

a) Implementieren Sie ein ConvNet, das wie folgt aufgebaut ist:

   1. Convolutional Layer `nn.Conv2d` mit 8 3x3 großen Kernels, *keine* Änderung der räumlichen Ausdehnung, ReLU Aktivierung.
   2. Max-Pooling Layer `nn.MaxPool2d` mit Größe 2x2 und Stride 2, kein Padding (default).
   3. Convolutional Layer mit 16 3x3 großen Kernels, *keine* Änderung der räumlichen Ausdehnung, ReLU Aktivierung.
   4. Max-Pooling Layer mit Größe 2x2 und Stride 2, kein Padding (default).
   5. Fully-connected Layer mit 20 Neuronen, ReLU Aktivierung.
   6. Ausgabe-Layer

   Für eine „normale“ `nn.Linear`-Layer muss die Anzahl der Inputs in die Layer angegeben werden. Diese Anzahl herauszufinden bzw.zu berechnen ist manchmal gar nicht so einfach. Eine Möglichkeit ist ein Testbild durch die Schichten vor die `nn.Linear`-Layer zu schicken und die Größe der Ausgabe zu bestimmen – was etwas umständlich ist.

   Die einfachere Möglichkeit besteht (seit PyTorch 1.8) darin eine `nn.LazyLinear`-Layer zu verwenden – für manche andere Layer gibt es ebenfalls eine „`Lazy`“-Variante. Allerdings kann die Anzahl der Parameter dann nicht sofort abgefragt werden – da sie noch nicht bekannt ist. Sie wird festgelegt, wenn zum ersten Mal ein Bild von dem Netzwerk verarbeitet wurde.

   Dazu können Sie auf ein Bild mit Zufallswerten verwenden. Alternative: Die `summary` Funktion, die Sie auf dem letzten Aufgabenblatt kennengelernt haben, hat einen optionalen Parameter `input_size`. Beachten Sie jeweils, dass `nn.Conv2d`-Layers Inputs der Shape (N, C_{in}, H, W) erwarten.

b) Lassen Sie sich Anzahl der Gewichte je Schicht ausgeben (siehe auch das vorige Aufgabenblatt).

c) Rechnen Sie die ausgegebene Anzahl der Gewichte je Schicht nach und übernehmen Sie die Anzahl und den Rechenweg in die Auswertung.

d) Wenden Sie das (noch untrainierte) Netz auf ein Bild mit Zufallswerten an und kontrollieren Sie, ob die Dimension zurückgelieferten Ergebnisses so ist, wie sie sein sollte.

e) Trainieren Sie das Netz für den **FashionMNIST**-Datensatz mit denselben Einstellungen wie zuletzt auf dem vorigen Aufgabenblatt. Bestimmen Sie dabei, wie lange das Training dauert. Wenn Sie ein Jupyter-Notebook unter VSCode verwenden und das Training in einer eigenen Zelle realisieren, zeigt Ihnen VSCode die Dauer direkt an. Wenn Sie es genauer und in einer Variable haben wollen, können Sie das Package `time` importieren, die Zeit vor und nach dem Training mit `time.time()` abfragen und die Differenz dieser beiden Zeiten bilden.

f) Wie groß ist die Accuracy?

g) Stellen Sie die Ergebnisse für die verschiedenen Varianten (Fully-Connected-Varianten des letzten Aufgabenblatts[1] aus der vorigen Aufgabe und das ConvNet), in einer Tabelle in Bezug auf Accuracy auf den Trainings- und auf den Testdaten, ungefähre Trainingszeiten je Epoche und Anzahl der Gewichte zusammen.

h) Optional: Google Colab ist ein kostenloser Cloud-Dienst von Google, der es ermöglicht, Jupyter-Notebooks im Browser auszuführen und dabei Rechenressourcen wie GPUs und TPUs zu nutzen. Trainieren Sie das Netz erneut, diesmal aber auf der GPU, die Sie bei Google Colab unter Edit/Notebook Settings aktivieren können. Wie lange hat das Training einer Epoche jetzt in etwa gedauert? Übernehmen Sie auch hier die Werte in die Ausarbeitung.

2 DataSets und DataLoaders

Lesen Sie sich das Tutorial DataSets und DataLoaders durch.

a) Was ist der Unterschied zwischen einem `DataSet` und einem `DataLoader`?

b) Was bewirkt die Methode `squeeze()` in dem Beispiel? Lassen Sie sich die Dimensionen des Tensors vor und nach Aufruf von `squeeze` ausgeben.

c) Gibt es auch eine Methode, die das Gegenteil macht? Wie könnte sie heißen? Hat sie dieselben (Pflicht-)Parameter?

d) Das Tutorial enthält Code zur Anzeige von Bildern aus dem Datensatz mit Beschriftungen. Übernehmen Sie diesen Code und passen Sie ihn so an, dass die ersten *falsch* klassifizierten Bilder angezeigt werden und beschriften Sie diese mit der Klasse, zu der sie falsch zugeordnet wurden, und dahinter (in Klammern) der richtigen Klasse.

[1] Bei der Fully-Connected aus der vorigen Aufgabe darauf achten, dass Sie die Ergebnisse für `FashionMNIST` eintragen.