

Mini-Challenge DEL: Hyperparameter Tuning und Model Evaluation

Ziel

Implementiere, trainiere und validiere mehrere Neuronale Netzwerk Modelle für einen Klassifikationstask mit geeignetem Dataset. Der zu modellierende Task und die dazu passenden Daten sollen am Anfang der Mini-Challenge individuell vereinbart werden. Dabei kannst Du auch Deine Wünsche einbringen und einen Task und Datensatz vorschlagen. Dabei ist aber darauf zu achten, dass der Fokus auf das Training und die Evaluation gelegt werden kann, und nicht zu viel Zeit mit Datenbeschaffung, -Analyse und -Vorverarbeitung verbraucht wird. Der Datensatz sollte auch nicht zu gross sein, um viele verschiedene Modell- und Parametervarianten evaluieren zu können. Als Basis-Modell sollte eine Architektur mit ein paar (wenigen) Convolutional Layern für die Feature-Extraktion und wenige Dense Layer für die Klassifikation verwendet werden. Davon ausgehend, sollen dann verschiedene Varianten untersucht (siehe unten). Bei einer Variante soll auch ein grösseres vortrainiertes Modell verwendet werden, welches dann fined-tuned werden soll.

Arbeitsschritte

Schritt 1: Auswahl Task / Datensatz

1. Mache Dir Gedanken, welcher Task gelernt werden soll und mit welchen Daten Du arbeiten möchtest.
2. Diskutiere die Idee mit dem Fachcoach.

Absprache/Beschluss mit Fachcoach über Daten, Task, Basismodell.

Schritt 2: Daten Kennenlernen

1. Mache Dich mit dem Datensatz vertraut, indem Du eine (kurze) explorative Analyse der Features durchführst: z.B. Vergleich der Klassen pro Feature, Balanciertheit der Klassen.
2. Führe ein geeignetes Preprocessing durch, z.B. Normalisierung der Daten.

Wie erwähnt, sollte dieser Schritt nicht viel Zeit beanspruchen, da der Datensatz einfach sein soll!

Schritt 3: Aufbau Modellierung, Basismodell

1. Implementiere ein Basismodell.
2. Lege fest, wie (mit welchen Metriken) Du Modelle evaluieren möchtest.
3. Implementiere Basisfunktionalität, um Modelle zu trainieren und gegeneinander zu evaluieren. Mache Dich vertraut mit einer MLOps Plattform (z.B. [W&B](#)), welche für die Evaluation verwendet werden soll.
4. Teste das Basismodell und auch Trainingsfunktionalität, indem Du nur mit einem einzigen Sample oder einem Batch trainierst. Damit bekommst Du zwar Overfitting, aber auch einen guten Test, der zeigt, dass Information aus den Daten aufgenommen werden kann.
5. Trainiere nun das Basismodell mit SGD, ohne Regularisierung, ohne Batchnorm unter Verwendung von allen Trainingsdaten. Führe ein Tuning der Lernrate und der Batchsize durch. Zeige die Unterschiede anhand der Lernkurven mit der gewählten

Kostenfunktion und den Metriken auf. Berücksichtige auch den Fehler in der Schätzung dieser Metriken: Wie kannst Du diesen schätzen? Welche Unterschiede sind signifikant?

Beachte: Es sollen in dieser MC **keine Verfahren zur automatischen Hyperparameter-Suche** (z.B. keine Bayes'sche Optimierung) verwendet werden!

Absprache/Beschluss mit Fachcoach: Zeige das erreichte dem Fachcoach. Besprich mit ihm die zu untersuchenden Varianten (Schritte 4).

Schritt 4: Weitergehende Evaluation

Bei der weitergehenden Evaluation verschiedener Varianten ist darauf zu achten, dass das Vorgehen stets möglichst reflektiert erfolgt und versucht wird, die Ergebnisse zu interpretieren. Am Schluss soll auch ein Fazit darüber gezogen werden, welche Variante am besten funktioniert. Deswegen empfiehlt es sich, verschiedene Parameter Settings zu betrachten - dabei aber immer nur einen Parameter auf einmal zu verändern.

1. Training mit SGD, ohne REG, ohne BN:

Verwende in den folgenden Schritten weiterhin Stochastic Gradient Descent, ohne Regularisierung (REG) und ohne Batchnorm (BN).

- a. *Initialisierung Gewichte:* Starte mit dem Basismodell und überlege Dir, wie Du die Gewichte initialisiert hast. Ändere die Initialisierung und zeige, wie sich das aufs Training auswirkt.
- b. *Modell-Komplexität:* Untersuche nun 4-5 Modell-Varianten unterschiedlicher Komplexität (unterschiedliche Anzahl Layer, Units oder Filter pro Layer). Für jedes Modell führe ein sorgfältiges Tuning der Lernrate durch (wähle die fürs Basismodell optimale Batchgrösse). Achte mit Blick auf die Lernkurven darauf, dass das Training stabil verläuft. Merke Dir bei jedem Trainingsexperiment, den Loss, die Performance Metrik(en) inkl. Schätzfehler, die verwendete Anzahl Epochen, Lernrate und Batch-Grösse.
- c. *Settings für die Convolutions:* Für eine fest gewählte Architektur aus dem vorhergehenden Schritt, untersuche bei den Convolutional Layern die Wahl der Kernel-Grösse, Stride, Padding, um eine möglichst gute Performance zu erreichen.
- d. Fasse die Ergebnisse zusammen in einem geeigneten Plot, bilde eine Synthese und folgere, welche Modell-Komplexität und Wahl der Convolutional Settings Dir am sinnvollsten erscheint.

2. Nutzen der Regularisierung

Ziehe nun verschiedene Regularisierungsmethoden bei den Dense Layern in Betracht:

- a. L1/L2 Weight Penalty
- b. Dropout

Evaluere den Nutzen der Regularisierung unter Berücksichtigung verschiedener Regularisierungs-Stärken. Starte hierzu von einer Modell-Variante, bei welcher Overfitting zu erkennen ist.

Beschreibe kurz, was das Ziel von Regularisierungs-Methoden ist (Regularisierung im

Allgemeinen, sowie auch Idee der einzelnen Methoden). Inwiefern wird dieses Ziel im gegebenen Fall erreicht?

Experimentiere mit Data-Augmentation. Überlege Dir welche Augmentierungsmethoden beim gegebenen Problem Sinn machen und untersuche, ob Data-Augmentation die Performance verbessert.

3. **Nutzen von Batchnorm BN** (ohne REG, mit SGD)
Evaluieren, ob Batchnorm etwas bringt. Beschreibe, was die Idee von BN ist, wozu es helfen soll.
4. **Nutzen von Adam** (ohne BN, ohne / mit REG)
Evaluieren, ob Du mit Adam bessere Resultate erzielen kannst.

Schritt 5: Transfer Learning

Untersuche schliesslich Transfer Learning: Wähle ein grösseres Modell, welches Dir für die gegebene Problemstellung sinnvoll erscheint und für welches vortrainierte Gewichte verfügbar sind.

Führe dann eine sorgfältiges Fine-Tuning durch.

Vergleiche die Performance mit dem bisher Erreichten.

Schritt 6: Präsentation, Bericht

1. **Präsentation** (~10m): Kurze Präsentation (Slides, nicht Notebook) mit Diskussion der wichtigsten Ergebnisse.
Q&A (~10min): Verständnisfragen zu Stochastic Gradient Descent, Parameter Tuning, Regularisierung, Batchnorm, Optimizers, Transfer Learning.
2. **Bericht** in Form **eines (!)** gut dokumentierten, übersichtlichen Jupyter Notebooks. Dieses soll dem Fachexperten erlauben, die Schritte nachzuvollziehen. Die Outputs (z.B. von W&B) sollten im Notebook integriert sein, so dass die Trainings nicht alle nochmals durch den Fachcoach durchgeführt werden müssen.

Zeitlicher Rahmen:

Wird beim Schritt 1 verbindlich festgelegt.

Beurteilung

Beurteilt wird auf Basis des abgegebenen Notebooks und der Präsentation inkl. Q&A:

- Vollständige und korrekte Umsetzung der vereinbarten Aufgabestellung.
- Klare, gut-strukturierte Umsetzung.
- Schlüssige Beschreibung und Interpretation der Ergebnisse. Gut gewählte und gut kommentierten Plots und Tabellen.
- Vernünftiger Umgang mit (Computing-)Ressourcen.
- Verständliche Präsentation der Ergebnisse, korrekte Antworten auf die Fragen.