

1 Idee

Das GAN wurde mit dem Ziel trainiert Bilder von Hunden zu generieren. Als Datensatz wurde das Stanford Dog Dataset gewählt. Link: <http://vision.stanford.edu/aditya86/ImageNetDogs/>. Für die Implementierung wurde ein Beispiel aus der PyTorch Dokumentation verwendet. Dies beinhaltete ein Beispiel für die Implementierung eines Deep Convolutional GAN zur Erstellung von menschlichen Gesichtern. Für den oben genannten Usecase wurde es leicht angepasst und anschließend trainiert.

2 Das GAN

Das implementierte DCGAN besteht grundsätzlich aus zwei Komponenten. Einem Generator und einem Diskriminator. Hierbei besteht die Aufgabe des Generators darin, auf Basis einer Noisemap Bilder zu erzeugen, die den Diskriminator möglichst gut täuschen können. Da der Diskriminator lernt, generierte Bilder von tatsächlich "echten" Bildern zu unterscheiden, werden die beiden Netze so in einem Nullsummenspiel gegeneinander trainiert, bis keines der Netze sich mehr verbessert und die Performance des Modells somit konvergiert. Bis hierhin unterscheiden sich reguläre GANs und Deep Convolutional GANs nicht voneinander. Der Unterschied besteht jedoch darin, dass DCGANs Convolutional Layers verwenden, während GANs lediglich fully connected layer verwenden. Dies macht sie für die Verarbeitung von Bilddaten besonders attraktiv. Aus diesem Grund wurde für den hier vorliegenden Usecase ein DCGAN gewählt.

3 Preprocessing

Die Bilder aus dem oben genannten Dataset werden vorverarbeitet, indem sie auf eine vorher festgelegte Bildgröße skaliert und zugeschnitten werden. Anschließend werden die so verarbeiteten Bilder in einen Tensor umgewandelt, der vor der Eingabe in das DCGAN normalisiert wird. Die Normalisierung des gesamten Datasets läuft unabhängig vom Trainingsprozess ab.

4 Trainingsprozess

Der Erste Trainingsdurchlauf wurde mit 50 Epochen durchgeführt. Da dies jedoch kein ausreichendes Ergebnis produzierte, wurden anschließend, bei gleichbleibenden Hyperparametern 100 Epochen trainiert (Abbildung 1). Hier konnte zwar eine Verbesserung der Ergebnisse festgestellt werden, allerdings wurde noch keine zufriedenstellende Qualität erreicht. Um die Anzahl der Trainingsepochen als Ursache für die schlechte Performance ausschließen zu können, wurden anschließend 600 Epochen des Modells bei gleichbleibenden Hyperparametern trainiert (Abbildung 2). Dies resultierte in einer schwachen Verbesserung der Ergebnisse, konnte jedoch ebenfalls keine erkennbaren Bilder generieren. Zudem ist im Loss Plot (Abbildung 3) zu erkennen, dass der Loss des Diskriminators und



Abbildung 1: 100 Epochen Training DCGAN



Abbildung 2: 600 Epochen Training DCGAN

des Generators respektive sich jeweils ab ca. Epoche 120 nicht mehr verbessern. Somit wurde für das Hyper Parameter Tuning eine Epochenzahl von 120 gewählt und es konnten bei gleichbleibender Learning Rate mehrere Kombinationen anderer Hyperparameter getestet werden. Innerhalb dieses Prozesses stellte sich eine Learning Rate von 0.001 als bei weitem zu groß heraus. Deshalb wurde mit $lr = 0.0005$ experimentiert, was zu den Ergebnissen in Abbildung 4 führte. Diese stellten sich bei weiteren Versuchen mit etwas kleineren oder größeren Lernraten als bestmögliche Ergebnisse heraus und werden deshalb hier als konkludierendes Resultat der Abgabe genannt.

5 Bewertung der Ergebnisse

Die erhaltenen Ergebnisse sind nicht als zufriedenstellend zu bezeichnen. Zwar schafft es das Modell teilweise plausible Umrisse, Hintergründe, Texturen oder Körperpartien zu generieren, jedoch beschränkt sich die Qualität der Ergebnisse meist auf einen dieser Faktoren. Durch die Abbildung von verschiedenen Stadien des Trainings in einem Gif (siehe Git) lässt sich auch erkennen, dass zu keinem Zeitpunkt während des Trainingsprozesses signifikant bessere Ergebnisse erreicht werden konnten.

Die ausbaufähige Performance des Modells lässt sich höchstwahrscheinlich mit mangelndem hyperpa-

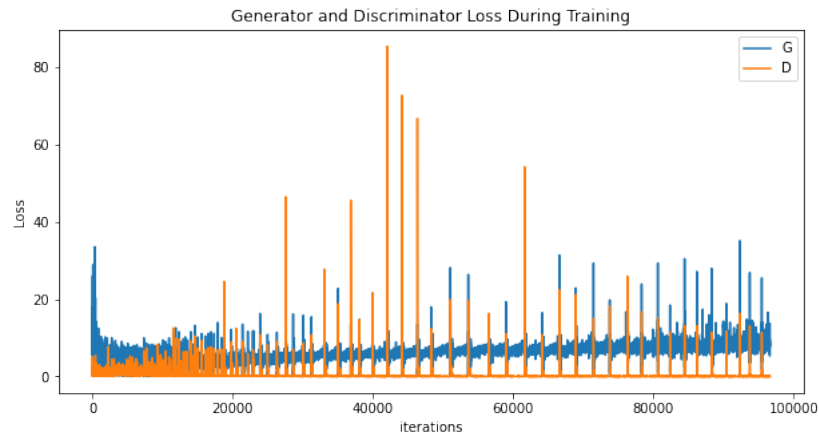


Abbildung 3: Loss über 600 Trainingsepochen

parameter Tuning erklären. Eine mangelnde Komplexität der verwendeten Netzwerke ist auszuschließen, da es im Bezug auf die Aufgabe menschliche Gesichter zu generieren akzeptabel performt. Ein aufwändigeres Hyper Parameter Tuning hätte allerdings den Rahmen dieses Projektes gesprengt, da ein Training von 120 Epochen mit den verfügbaren Ressourcen eine vollständige Nacht in Anspruch nimmt. Da parallel noch einige andere Projekte während dieses Zeitfensters zugriff auf diese Rechenleistung benötigten (Doom, Pong abgabe), konnten leider keine besseren Ergebnisse erreicht werden. Zudem ist infrage zu stellen, ob die Anwendung eines GANs, das ursprünglich für die Erstellung von Bildern menschlicher Gesichter konzipiert wurde auf Bildern von Hunde überhaupt ohne weiteres annehmbare Ergebnisse liefern kann. Das grundsätzliche Ziel, die Implementierung eines funktionierenden GANs mit annähernd plausiblen Ergebnissen, wurde jedoch erreicht.



Abbildung 4: Ergebnis: 120 Epochen mit $lr = 0.0005$