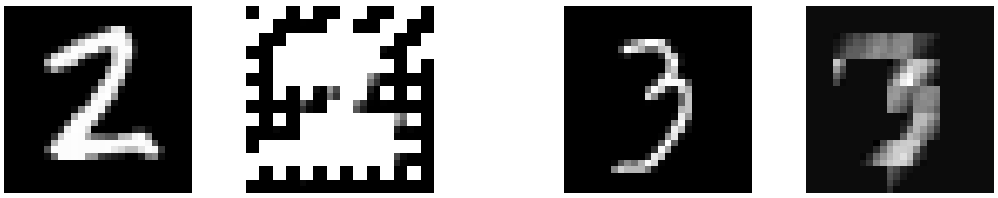
Modellierung 1, Praktikum, 17.07.2017 bis 28.07.2017
Gruppe Deep Learning
Christian Ali, Janka Bauer, Niklas Bockius

Montag, 17.07.17:

- Vorbesprechung und Einführung in Deep Learning
- Initialisierung eines Git-Repository
- Einarbeiten

Dienstag, 18.07.17:

- AutoEncoder für MNIST aufbauend auf dem Tutorial: erste Hälfte (bis Bottleneck) aus Tutorial übernommen, zweite Hälfte selbst erstellt (alles reverse)
  am Ende des Tages unschöne Ergebnisse:
  - erst regelmäßiges Rauschen
  - dann Zahl sehr verpixelt, aber erkennbar
- Verbesserungen durch:
  - loss-Funktion mit least squares statt mit cross-entropy berechnet
  - Anzahl der Channels (auf dem Rückweg) erhöht



(a) pic1, Original

(b) pic1, generiert mit cross-entropy und einem Channel

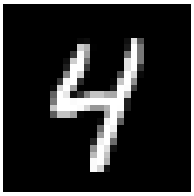
(c) pic2, Original

(d) pic2, generiert mit least squares aber nur einem Channel

Abbildung 1: Die unterschiedlichen Stadien des AutoEncoders für MNIST

Mittwoch, 19.07.17:

- AutoEncoder für MNIST fertiggestellt, schöne Ergebnisse:
  - Zahlen klar erkennbar, minimal verschwommen



(a) pic3, Original

(b) pic3, generiert mit least squares und 32 Channels

Abbildung 2: AutoEncoder für MNIST mit mehr Channels

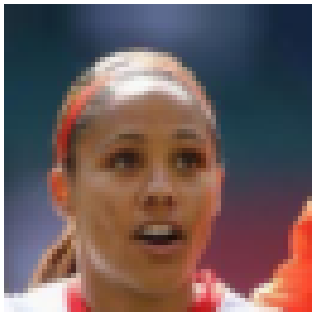
Donnerstag, 20.07.17:

- Implementierung eines GAN angefangen, Diskriminator- und Generator-Netzwerk geschrieben, jedoch noch ohne loss-Funktionen!
- AutoEncoder für CelebA-images umzuschreiben:
  - trotz Übergang von schwarz-weiß Bildern zu RGB-Bildern (zusätzliche Dimension) Verwendung von conv2D. (RGB-Dim als "Start-Features", analog zu den durch die Filter erstellten Features, behandelt)
  - Anpassen der Anzahl Channels vor und nach Bottleneck, Bottleneck "größer" gewählt, um nicht zu viele Informationen zu verlieren
  - Bildergröße anpassen (statt Eingabe 218x178 Ränder abgeschnitten auf 176x176 oder skaliert auf 64x64, wobei 64x64 vorausschauend für GAN gewählt wurde)
- Test, ob Zahlen mit zweiter Hälfte von AutoEncoder mit zufälliger Eingabe erzeugt werden können:
  - Ergebnis nur Rauschen! (Im Nachhinein haben wir herausgefunden, dass dies an der Wahl bzw. der Behandlung der Variablen lag. Bei CelebA wurde dieser Fehler behoben!)

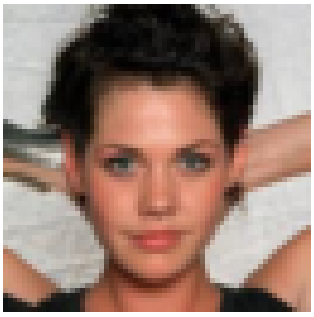
Freitag, 21.07.17:

- AutoEncoder für CelebA fertiggestellt:
  - Bilder verschwommen, aber erkennbar
  - individuelle Details werden entfernt/ gehen verloren
- Weiterentwicklung des GAN
  - Test des Diskriminator-Netzwerks
  - Erste Klassifizierungen mithilfe des Diskriminator-Netzwerks auf einem Label aus den vorgegebenen Annotationen von CelebA

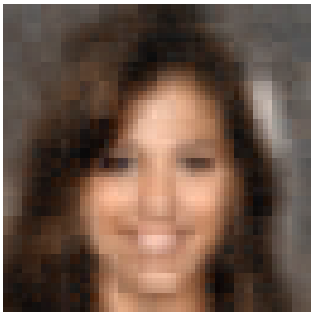
(a) pic1, Original



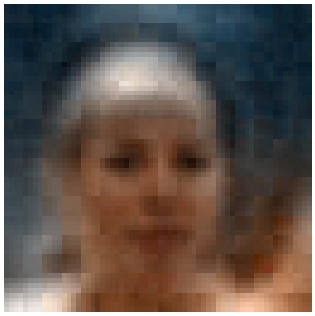
(b) pic2, Original



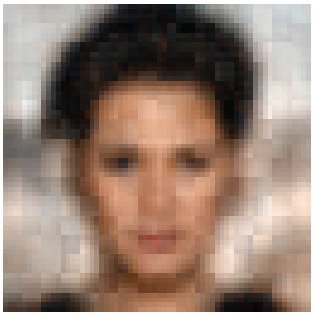
(c) pic3, Original



(d) pic1, generiert



(e) pic2, generiert



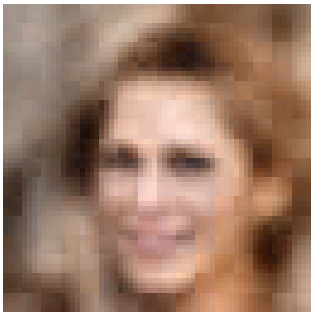
(f) pic3, generiert

Abbildung 3: AutoEncoder für CelebA mit least squares und 48 Channels

Montag, 24.07.17:

- Erzeugung gemittelter Bilder, d.h. im Bottleneck der CelebA AutoEncoders zwei (oder mehr) Bilder gemittelt und als ein Bild zurückgeneriert
  - nach wie vor verschwommene Bilder, aber je nach gewählter Kombination der Eingabebilder sind aus jedem Bild einzelne Details gut erkennbar

(a) pic1, Original



(b) gemitteltes Bild

(c) pic2, Original

Abbildung 4: Erzeugung eines gemittelten Bildes aus zwei Originalbildern von CelebA.

- Erzeugung zufälliger Gesichter

  - anfangs nur Rauschen erkennbar

- Verbesserung durch Einbindung einer PCA/ Kovarianzmatrix, sodass Werte (im Bottleneck) mit der entsprechenden Verteilung gesamplet werden können!
  - Danach folgende Bilder:

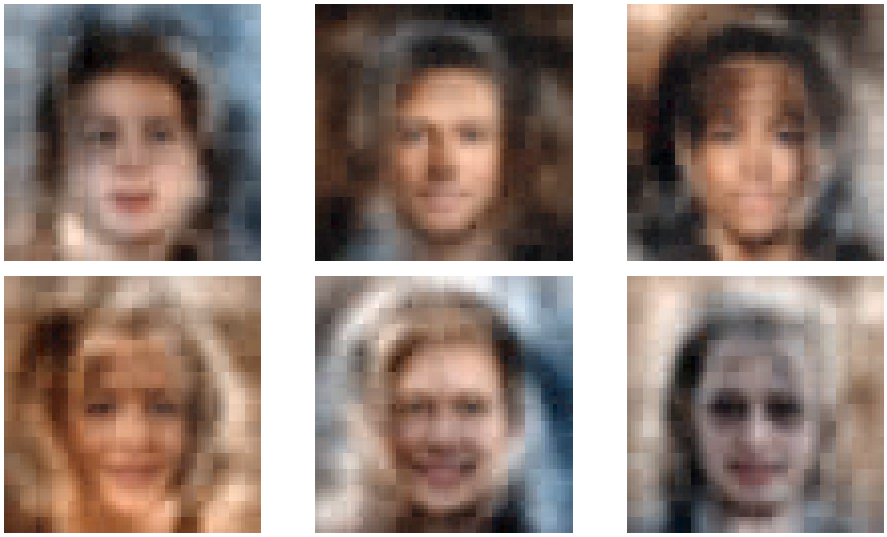


Abbildung 5: zufällig erzeugte CelebA-Gesichter

- Einbindung von Schiebereglern, der entlang der Hauptachsen der PCA die Gewichtung der Werte verändert, sodass die zufällig erzeugten Bilder in Echtzeit verändert werden können
  - Veränderungen (bei entsprechend hoher Skalierung der Gewichte) gut erkennbar, jedoch ändern sich bei den meisten Schiebereglern ähnliche "Eigenschaften" wie Hintergrund, Haarlänge, Haarfarbe... hier muss noch etwas gefeilt werden.
- erstmalige Kombinierung von Generator und Diskriminator, sehr viel Debugging

Dienstag, 25.07.17:

- Weitere Bearbeitung der Schieberegler, sodass Bild direkt neben den Schiebereglern angezeigt und aktualisiert wird
- Debugging des GAN

Mittwoch, 26.07.17:

- Debugging des GAN
  - viele Änderungen der loss-Funktionen
  - läuft nun durch, Ergebnisse jedoch noch nicht ausgewertet

Donnerstag, 27.07.17:

- Verschiedene Versuche, unser GAN zu optimieren und sinnvolle Ergebnisse zu erzielen
  - Abwägung und Balancierung des Trainings des Diskriminators und des Generators. Dabei soll verhindert werden, dass eine Partei gewinnt oder sich beide in einem lokalen Minimum verlieren.
  - Es ist nicht klar, ob sich in unserem Programm noch ein Bug befindet, oder ob es an der Abstimmung der Netzwerke aufeinander hängt. Als Hinweis haben wir bekommen, das nächste Mal die einzelnen Netzwerke nicht selbst zu schreiben, sondern auf funktionierende Netzwerke zurückzugreifen und diese zu kombinieren, um damit die Fehlerquellen einzuschränken.
  - Alternativ noch auszuprobieren: Zusätzlich den AutoEncoder in unser GAN einbauen.
  - Es konnten leider keine guten Ergebnisse erzielt werden!