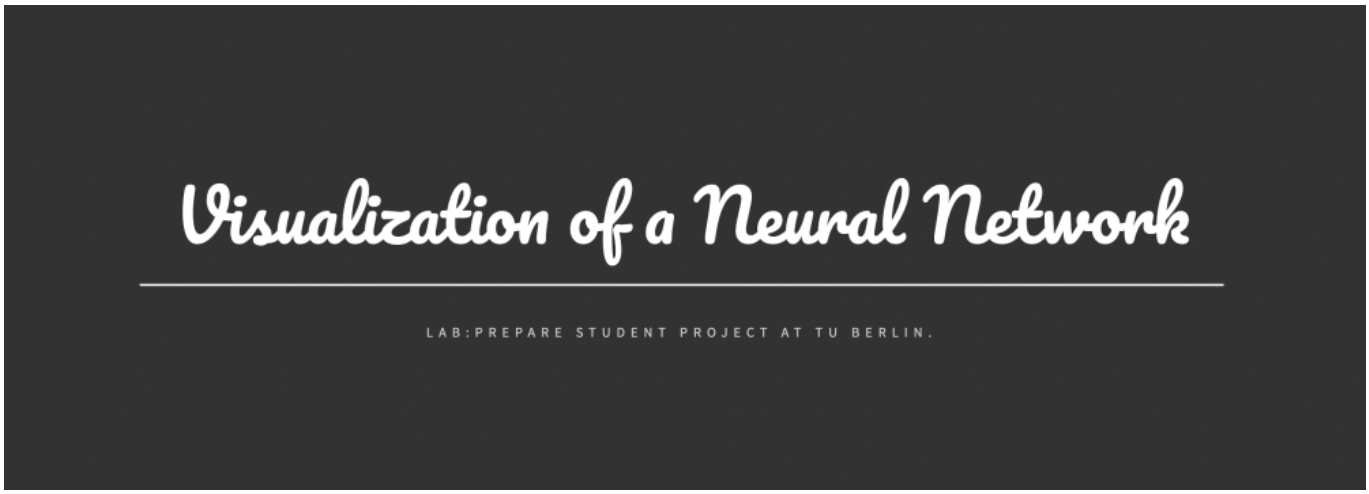
Visualisierung eines neuronalen Netzes



SoSe 20

Gruppe: Philipp Pirlet, Pablo Osinaga Arze, Annabella Kadavanich

---

Grundidee

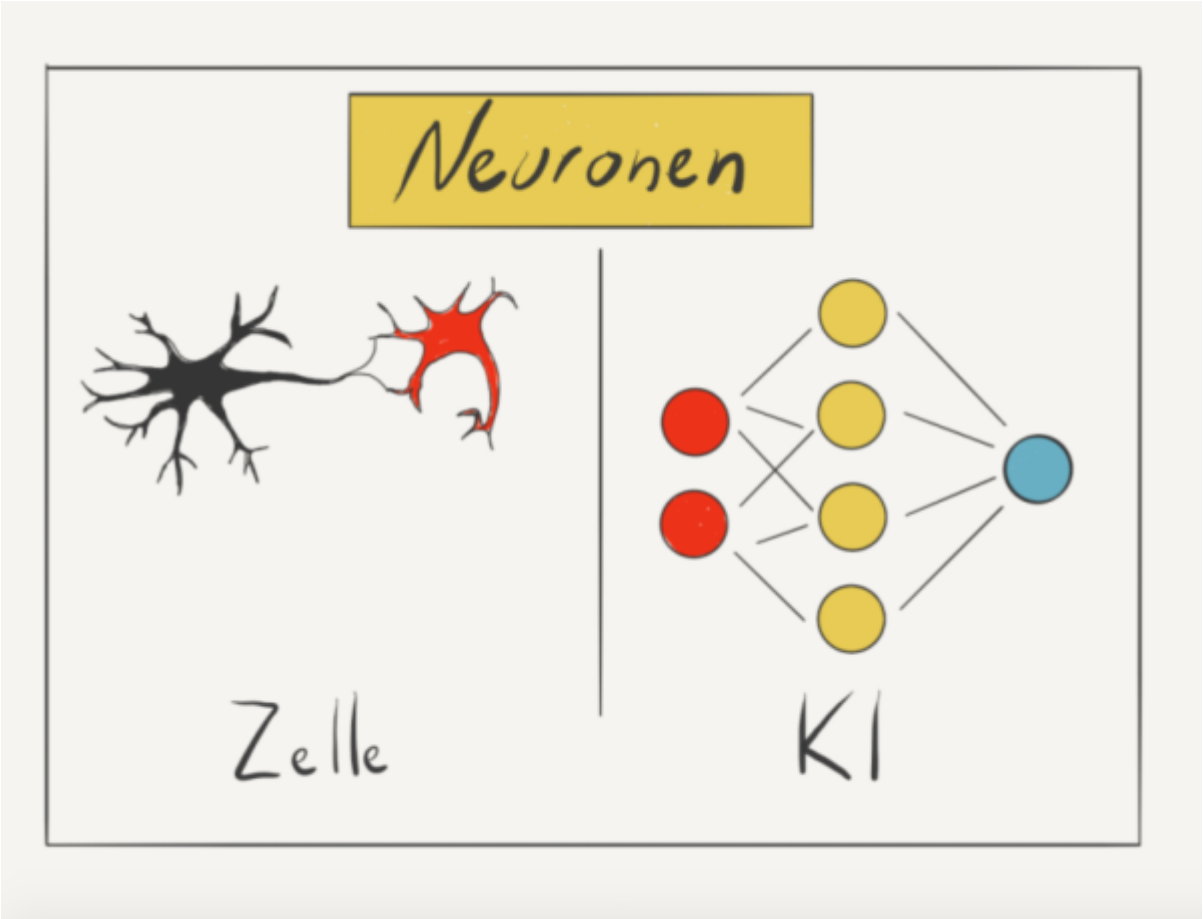
Neuronale Netze (NN) sind ein abstraktes Konzept, dass oft für Menschen ohne IT-Hintergrund schwer greifbar und verständlich ist. Das Thema soll mit Hilfe einer Visualisierung mit einer Web UI (User Interface) verständlicher gemacht werden.

NN können in vielen verschiedenen Szenarien verwendet werden. In diesem Projekt klassifiziert unser NN handgeschriebene Zahlen, basierend auf dem MNIST dataset. Unsere Nutzer können ihre Zahl bequem über unser Webinterface eingeben und dann sehen, was in jedem Layer des NN passiert.

Theorie

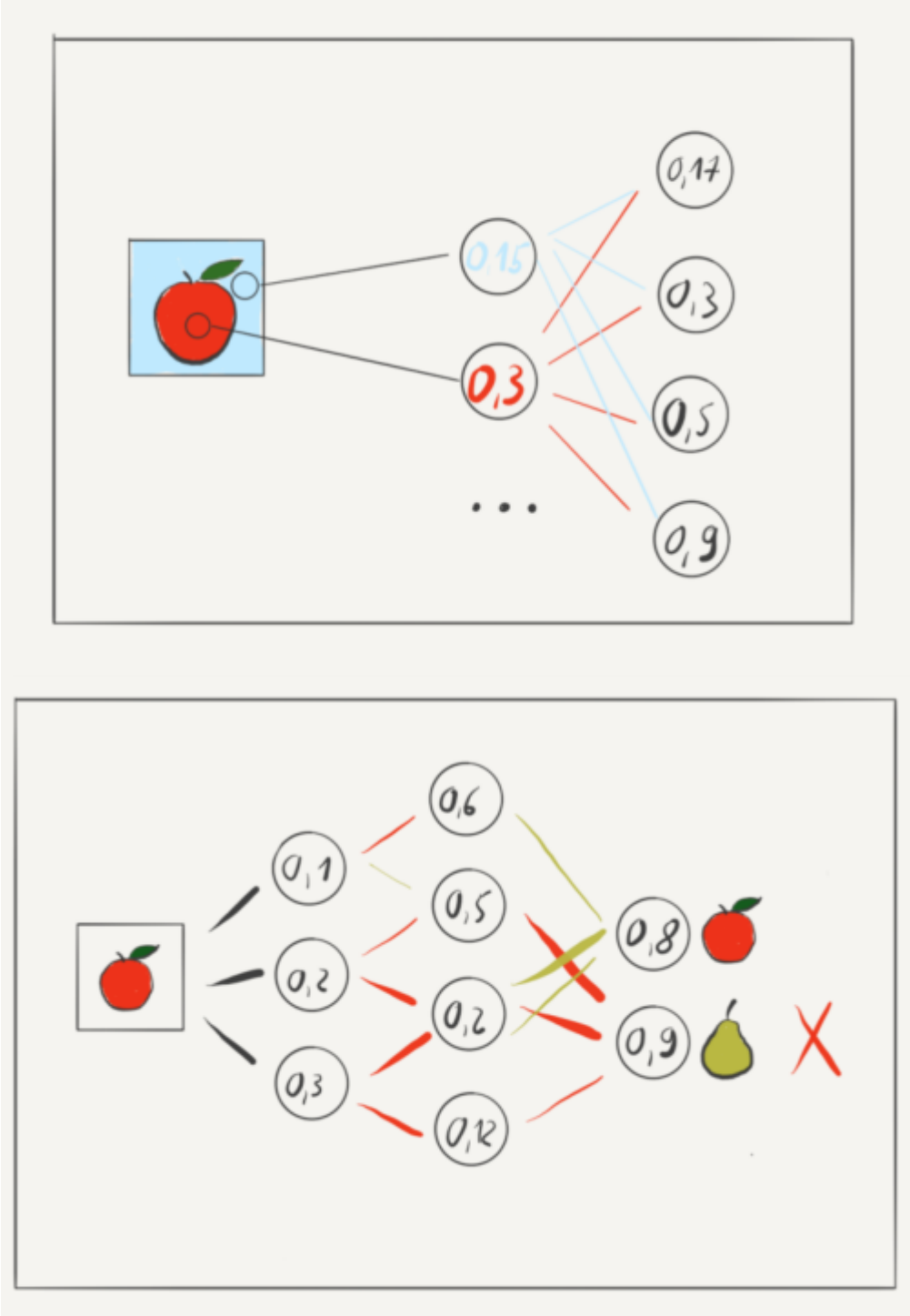
Neuronale Netze sind ein Teil des sog. supervised learning und bilden damit ein Teilgebiet des maschinellen Lernens in der Informatik. Sie sind in der Lage, große Mengen an unstrukturierten Daten besonders gut auszuwerten und Muster in ihnen zu finden. Zu unstrukturierten Daten gehören beispielsweise Bilder, Videos und Töne.

NN leihen sich ihre Begriffe aus der Biologie aus. Im menschlichen Gehirn ist ein Neuron eine Nervenzelle, die mit anderen Nervenzellen verbunden ist und elektrische Signale weitergibt. Aus Milliarden von Neuronen setzt sich das Gehirn zusammen und aus ihrer Aktivität unser Geist.



Ein künstliches neuronales Netz funktioniert ähnlich: Hier ist ein Neuron eine mathematische Formel, die einen Input verarbeitet und daraus einen Output generiert. Die Werte der Formel werden dabei durch die Ausgangsdaten definiert. Viele künstliche Neuronen arbeiten zusammen und ergeben so ein künstliches neuronales Netz (KNN).

NN bestehen aus verschiedenen Schichten (engl: Layer). Es gibt immer eine Input-, sowie eine Output-Schicht. Zudem gibt es Zwischenschichten, welche Berechnungen vornehmen. Jede Schicht besteht aus sog. Neuronen. Neurone können Berechnungen durchführen und Zwischenergebnisse speichern. Ein NN lernt, indem es immer wieder die Berechnungen und Gewichte der Neuronen anpasst, bis es die Eingabebilder richtig klassifiziert.



Damit ein NN funktioniert, muss man es zuerst mit passenden Daten trainieren. In unserem Projekt erkennt unser NN handgeschriebene Zahlen. Daher haben wir unser Netz mit verschiedenen Bildern von handgeschriebenen Zahlen trainiert. Während dem Training nimmt das NN jedes Bild, löst die einzelnen Pixel in Daten auf (z.B. Farbwerte) und berechnet dann mit diesem Daten in einer komplexen Formel ein Ergebnis, das sie danach mit der Beschriftung (eng.: Label) vergleicht. Stimmen Ergebnis der Formel und das Label überein, hat das NN ein Bild richtig erkannt. Ansonsten muss das NN noch weitertrainiert werden.

Convolutional Neural Networks (CNNs)

Ein CNN (deutsch: gefaltetes neuronales Netz) ist eine besondere Form von NN'en. Bei unserem NN handelt es sich auch um ein CNN. Es besitzt mehrere Faltungssschichten und ist für Anwendungen im Bereich Bild- und Spracherkennung sehr gut geeignet. Die einzelnen Schichten unseres CNN sind:

- Convolutional Layer
- (Max) Pooling Layer
- Flatten Layer
- Dense Layer

Convolutional-Schichten sind die eigentliche Faltungsebene. Sie ist in der Lage, in den Eingabedaten einzelne Merkmale zu erkennen und zu extrahieren. Bei der Bildverarbeitung können dies Merkmale wie Linien, Kanten oder bestimmte Formen sein. Die Verarbeitung der Eingabedaten erfolgt in Form einer Matrix. Es kommen Matrizen definierter Größe (Breite x Höhe x Kanäle) zum Einsatz.

Die **Pooling-Schicht**, auch Subsampling-Schicht genannt, verdichtet und reduziert die Auflösung der erkannten Merkmale. Hierfür verwendet die Schicht Methoden wie das Maximal-Pooling oder Mittelwert-Pooling. Das Pooling verwirft überflüssige Informationen und reduziert die Datenmenge. Durch das reduzierte Datenaufkommen erhöht sich die Berechnungsgeschwindigkeit.

Der Klassifizierer ist der letzte Schritt in einem CNN und wird als **Dense Layer** bezeichnet. In dieser Schicht ist jeder Knoten mit jedem Knoten in der vorhergehenden Ebene verbunden. Wie jeder Klassifizierer, braucht dieser individuelle Features. Hierfür wird der mehrdimensionale Output aus den Convolutions in einen eindimensionalen Vector überführt. Diesen Vorgang passiert in vorherigen Layer, dem sog. **Flatten Layer**, und wird daher auch “Flattening” genannt.

Den Abschluss des Convolutional Neural Networks bildet die **vollständig verknüpfte Schicht**. Alle Merkmale und Elemente der vorgelagerten Schichten sind mit jedem Ausgabemerkmal verknüpft. Die Anzahl der Neuronen ist abhängig von den Klassen oder Objekten, die das neuronale Netz unterscheiden soll.

Aufbau unseres NN

Unser NN besteht aus 9 Layern und ist damit im Vergleich zu anderen NN sehr klein und simpel. Die Schichten sind wie folgt angeordnet:

1. Convolutional Layer
2. Batch Normalization Layer
3. Max Pooling Layer
4. Convolutional Layer
5. Batch Normalization Layer
6. Max Pooling Layer
7. Flatten Layer
8. Dense Layer
9. Dense Layer

Projektaufbau

Unser Projekt besteht aus vier Komponenten, die im Folgenden kurz erklärt werden:

1. Eingabe: Über unsere UI kann ein Nutzer mit seinem Laptop oder Tablet eine Zahl zeichnen.
2. Neuronales Netz: Die Eingabe wird von einem NN weiter verarbeitet. Es handelt sich hierbei um ein kleines, simples NN.
3. User Interface / Visualisierung via Monitor: Wir haben eine UI implementiert, die beim Starten des Programms über den Webbrowser einsehbar ist und das NN abbildet. Ziel ist es, die Input- und Outputgewichte, sowie die Zwischenergebnisse der Berechnungen des NN darzustellen.

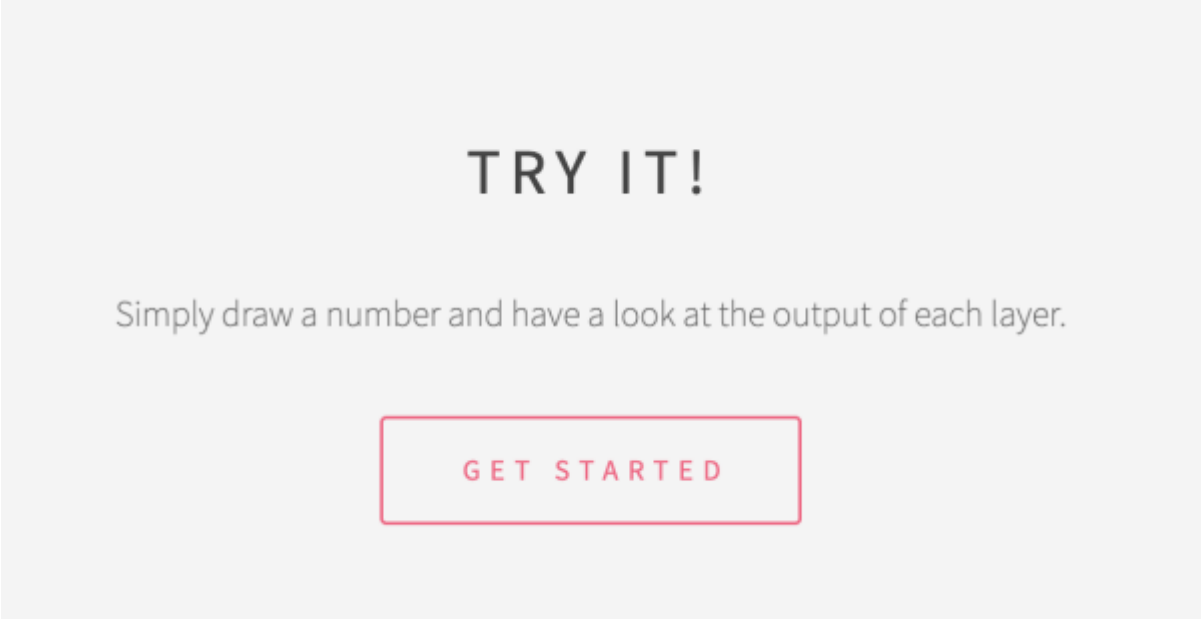
Alle 4 Komponenten werden mit Hilfe von APIs verbunden.

Starten des Projekts

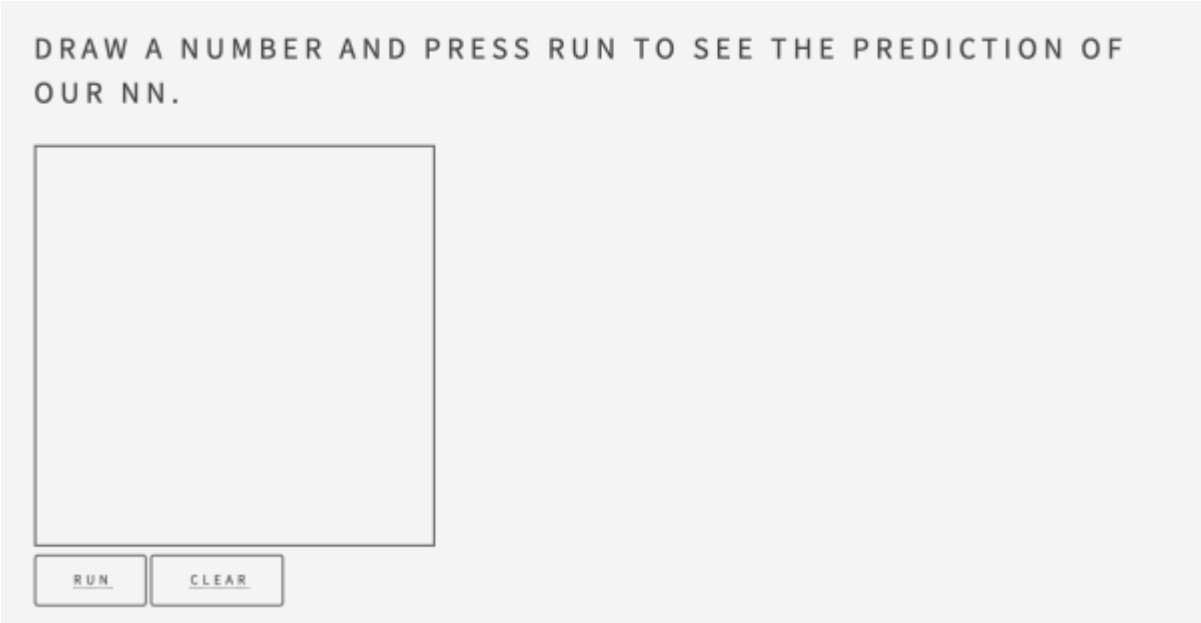
1. Clone unser Projekt von GitHub

2. Folge der Anleitung in unserem README. Wenn du auf http://127.0.0.1:8000 gehst, solltest du jetzt unsere Website sehen.

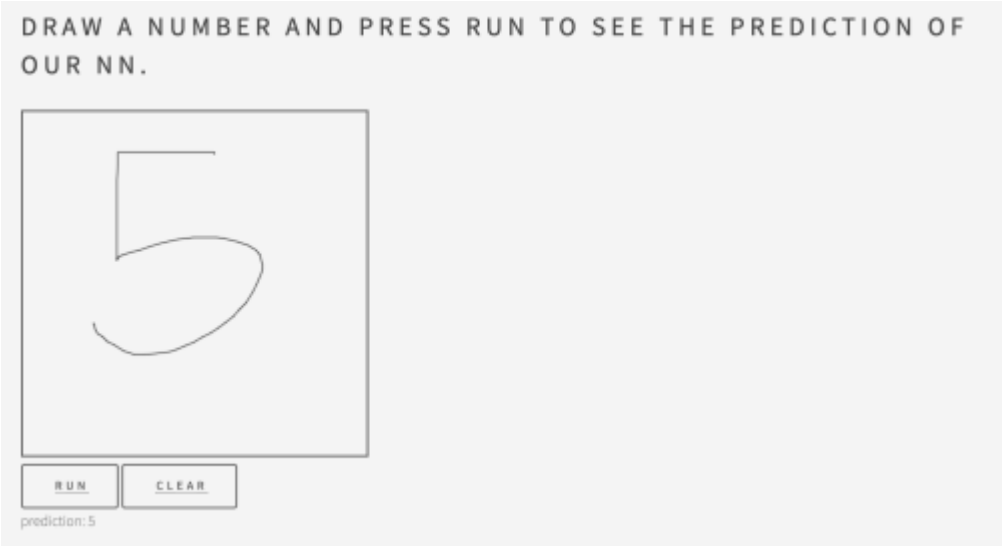
3. Scroll ganz runter und drück den Button *Try it*.



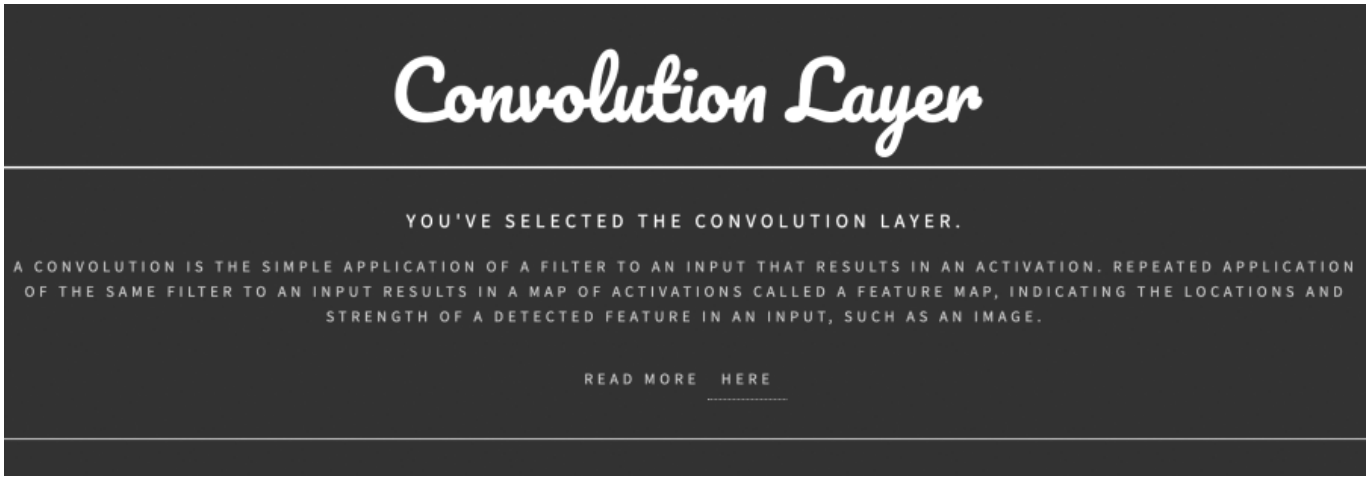
4. Du solltest nun die folgende Eingabemaske sehen:



5. Du kannst hier jetzt eine Zahl malen und auf *Run* drücken um zu sehen, wie das NN deine Zahl klassifiziert (Die Prediction steht dann unter dem Run Knop).



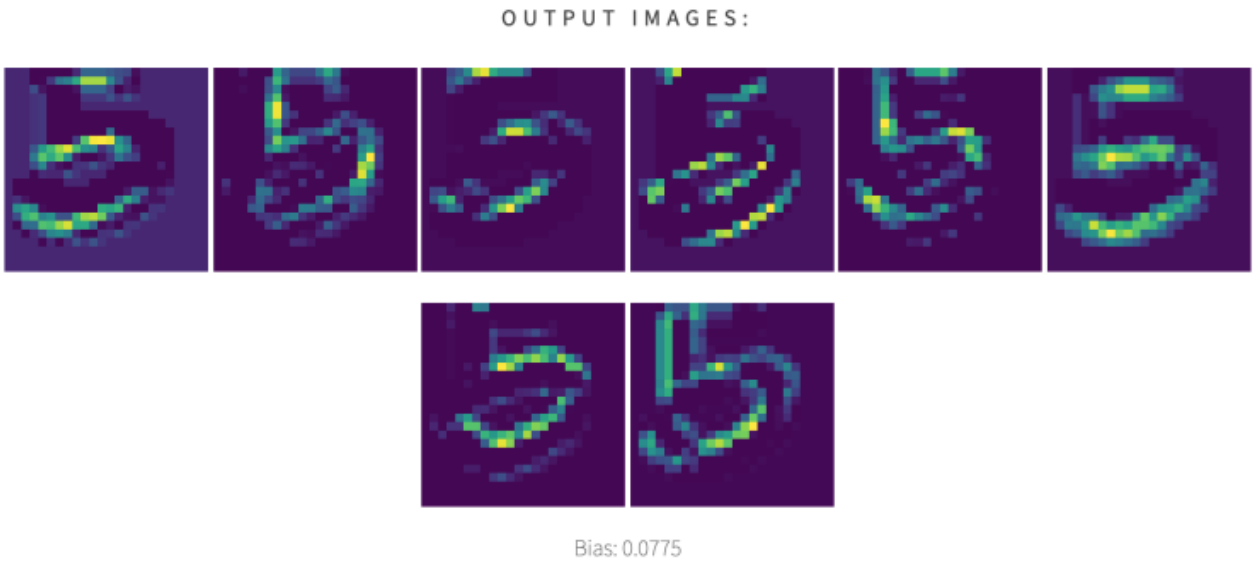
6. Wenn du weiter runter scrollst, kannst du dir anschauen, was in den einzelnen Schichten des NN passiert ist. Wir empfehlen dir, mit dem ersten Layer (Layer 0) zu starten. Klick also einfach auf diesen Layer. Du solltest dann zu folgender Seite weitergeleitet werden:



Du solltest nun den Namen deiner ausgewählten Schicht, sowie einen kleinen Begleittext sehen.

7. Wenn du weiter runter scrollst, findest du noch weitere Informationen, die je nach Layer etwas variieren. Auf jeden Fall solltest du immer den Input und Output sehen. Beim convolutional Layer jetzt siehst du beispielsweise ein Bild von deiner Eingabe:

8. … und ganz unten die erzeugten Output Bilder:



9. Wenn du wieder ganz rauf scrollst, findest du oben rechts einen kleinen grauen Knopf mit dem Text *Next*. Du kannst dich jetzt durch alle Layer klicken und dir ansehen, was in jedem Layer passiert ist. Wenn du zurück zur Startseite möchtest, findest du immer ganz unten den Knopf *Main Page*.

Viel Spaß beim Ausprobieren! :)

Projektverwaltung

Zur Verwaltung und zuordnung der Aufgaben und zur Versionsverwaltung verwenden wir ein Git-Repository welches unter folgenden Link einsehbar ist:
https://github.com/paguos/lab-prepare-deep-neural-network

Quellen

Theorie NN (1), Stand: 24.09.2020

Theorie NN (2), Stand: 24.09.2020

Flatten + Dense Layer, Stand: 24.09.2020