

Grundlagen zur Erstellung eines neuronalen Netzwerks mittels Perceptilabs

Inhalt

1. Grundlagen von Perceptilabs	2
2. Erstellen eines eigenen neuronalen Netzwerks	6
3. Modifikation der Parameter	11
3.1. Variation der Epochen:	11
3.2. Variation der Aktivierungsfunktionen	11
4. Export des neuronalen Netzes	13

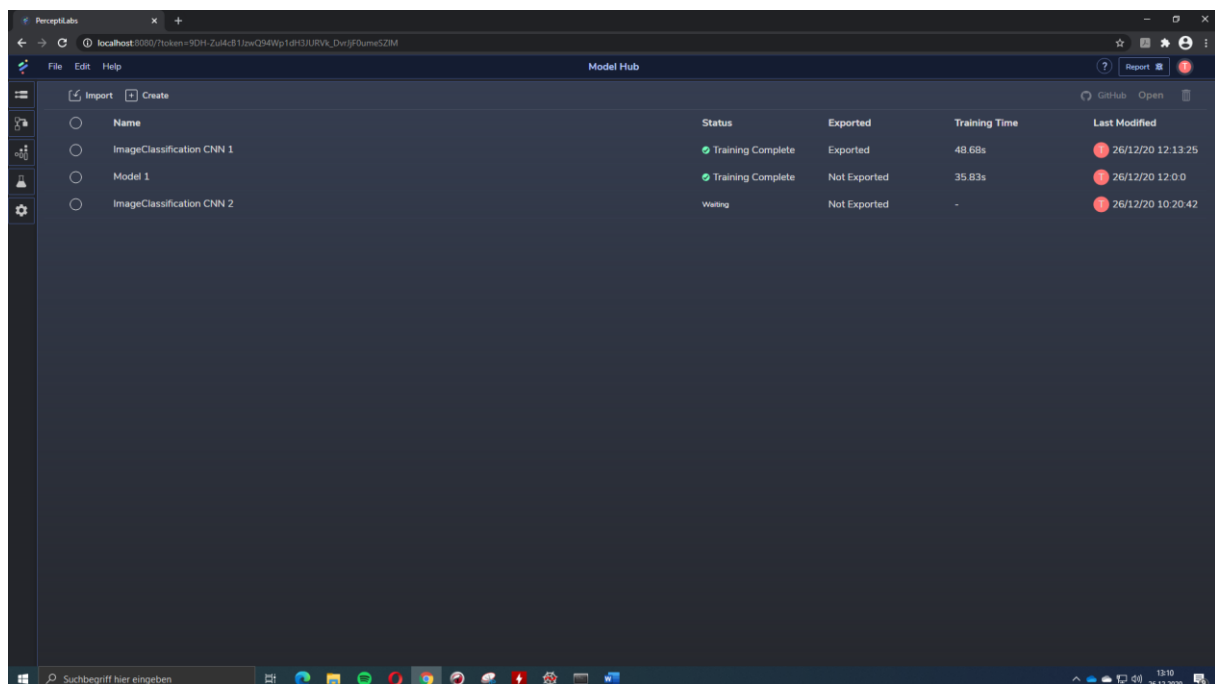
Achtung: Dieses Tool ist sehr gut geeignet um als Laie ohne Programmierkenntnisse in die Thematik der Neuronalen Netze einzusteigen. Jedoch ist es aufgrund der Testoberfläche nur bedingt für die vollständige Erstellung eines neuronalen Netzes auf Basis großer Datensätze geeignet, da die Testproben manuell durchgeklickt werden müssen.

1. Grundlagen von Perceptilabs

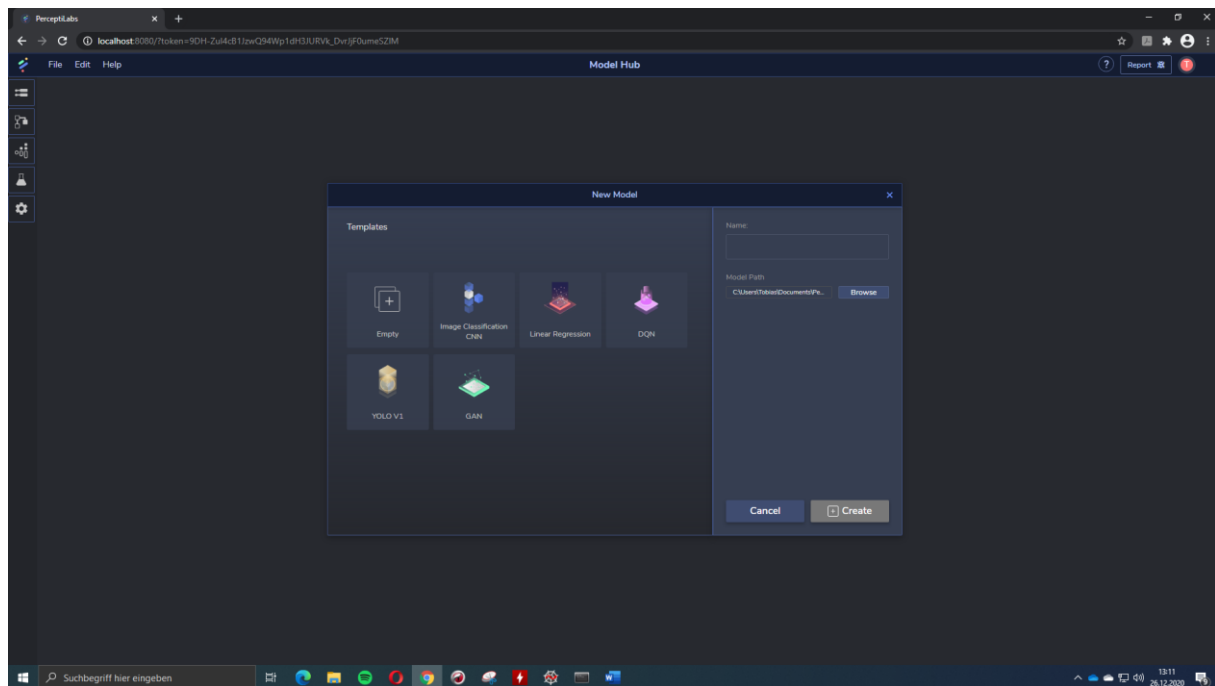
Starten von Perceptilabs (Schritt für Schritt siehe bei Installationsanleitungen)

```
Anaconda Powershell Prompt (anaconda3)

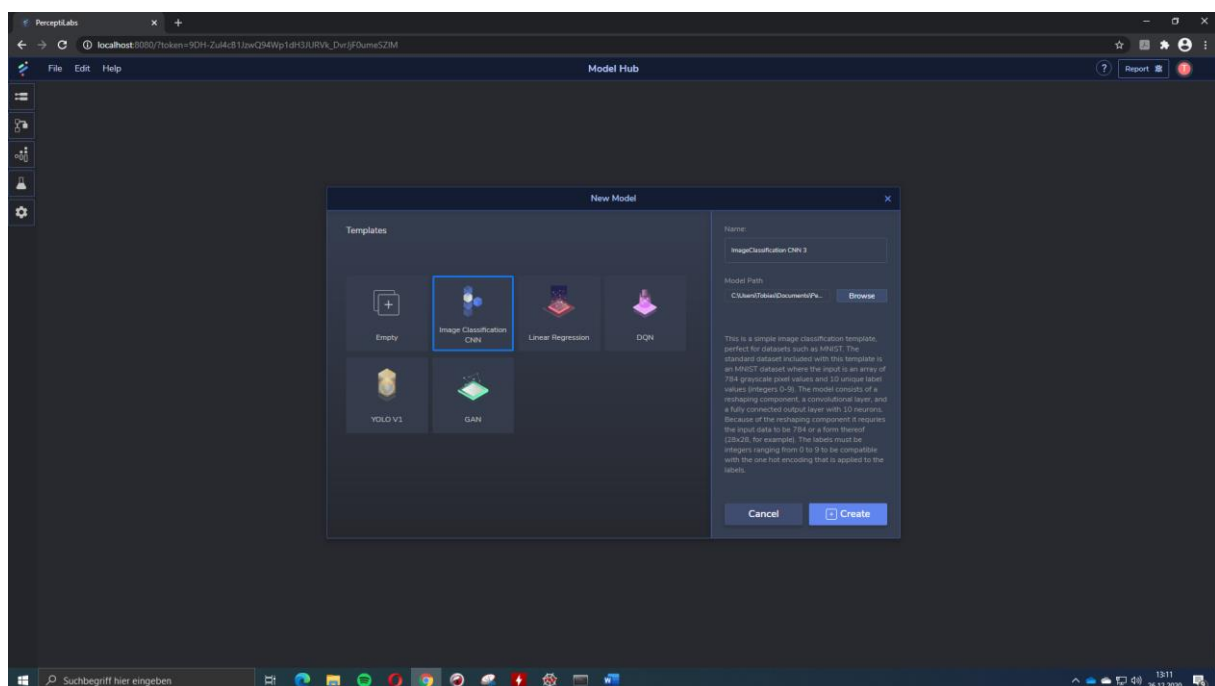
(base) PS C:\Users\Tobias> conda activate pi
(pi) PS C:\Users\Tobias> perceptilabs
Perceptilabs: Your environment does not have atari_py installed, so some functionality may not be available
Perceptilabs: To install it, please follow the directions at https://github.com/Kojoley/atari-py and then install gym th
rough 'pip install gym[atari]'
INFORMATION: Es konnten keine Dateien mit dem angegebenen
Muster gefunden werden.
Perceptilabs: Your environment does not have git installed, so interactions with GitHub will not be available
Perceptilabs: Starting
Perceptilabs: Waiting for services to listen on these ports:
Perceptilabs:   kernel on port 5000
Perceptilabs: Waiting for services to listen on these ports:
Perceptilabs:   kernel on port 5000
Perceptilabs: Waiting for services to listen on these ports:
Perceptilabs:   kernel on port 5000
Perceptilabs: Waiting for services to listen on these ports:
Perceptilabs:   kernel on port 5000
Perceptilabs: Perceptilabs Started
Perceptilabs: Perceptilabs is running at http://localhost:8080/?token=9DH-ZuI4cB1JzwQ94Wp1dH3JURVd_DvrJjF0umeSZlM
Perceptilabs: Use Control-C to stop this server and shut down all Perceptilabs processes.
```



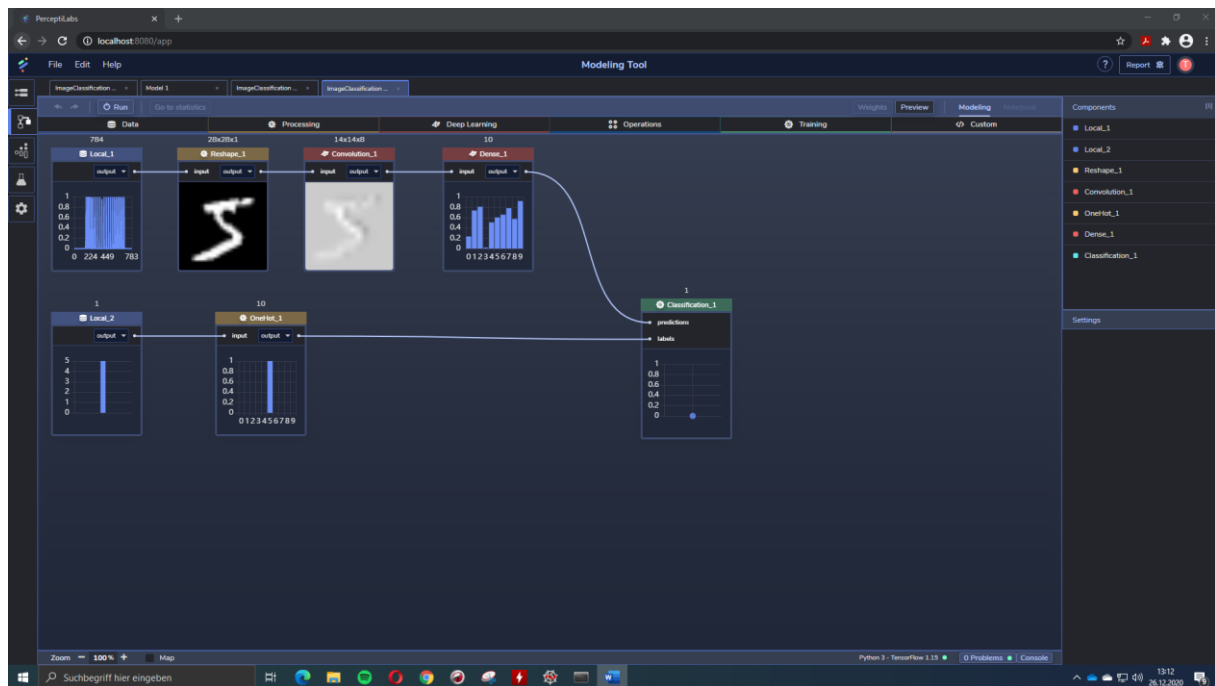
Druck klicken auf Create öffnet sich folgendes Fenster



Hier kann man sich z.B. ein funktionierendes CNN erstellen, welches auf den MNIST-Trainingsdatensatz ausgelegt ist. Dazu klickt man auf Image Classification CNN und anschließend auf create.



Es öffnet sich folgende Ansicht mit der interaktiven Schaltfläche:

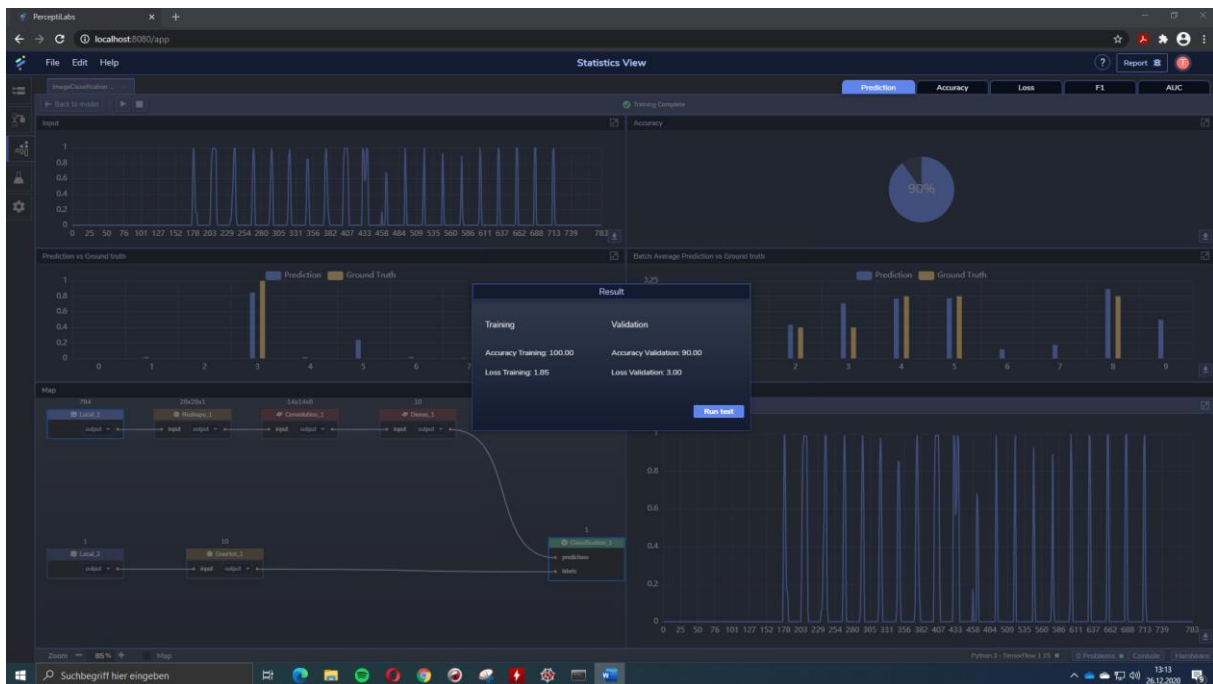


Mit einem Klick auf run startet das neuronale Netz mit dem Training.



Man kann das Training pausieren oder ggf. abbrechen. Dazu sind nach Laden und Beginn des Trainings 2 Schaltflächen zu sehen. Zudem ist mittig in der Trainingsansicht ein Fortschrittbalken sichtbar.

Nach Abgeschlossenem Training werden die Resultate des Trainings angezeigt:



Mit Klick auf Run Test beginnt der Test des neuronalen Netzwerks. Mit Klick auf Next Sample wird mit dem Training der nächsten Probe begonnen.

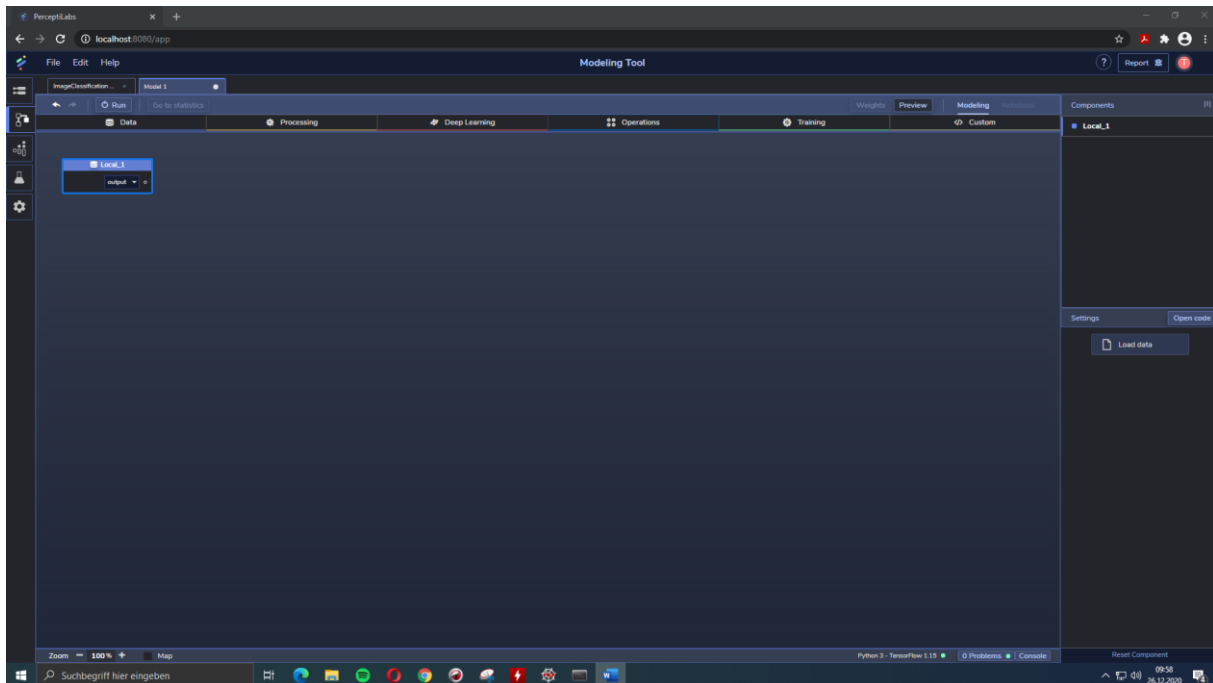
Die Zuordnung ist korrekt, wenn der blaue und der gelbe Balken (Prediction und Ground Truth) auf der selben Zahl zu sehen sind.



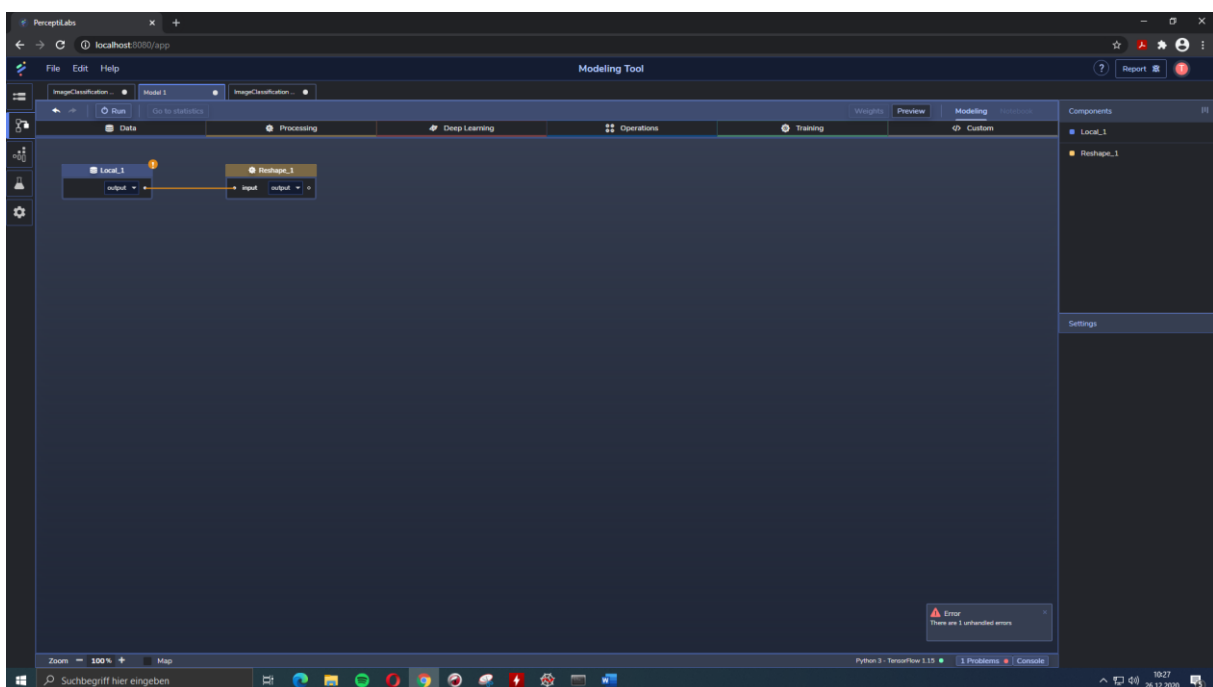
2. Erstellen eines eigenen neuronalen Netzwerks

In diesem Beispiel wird ein neuronales Netz zur Klassifizierung der mnist Daten erstellt.

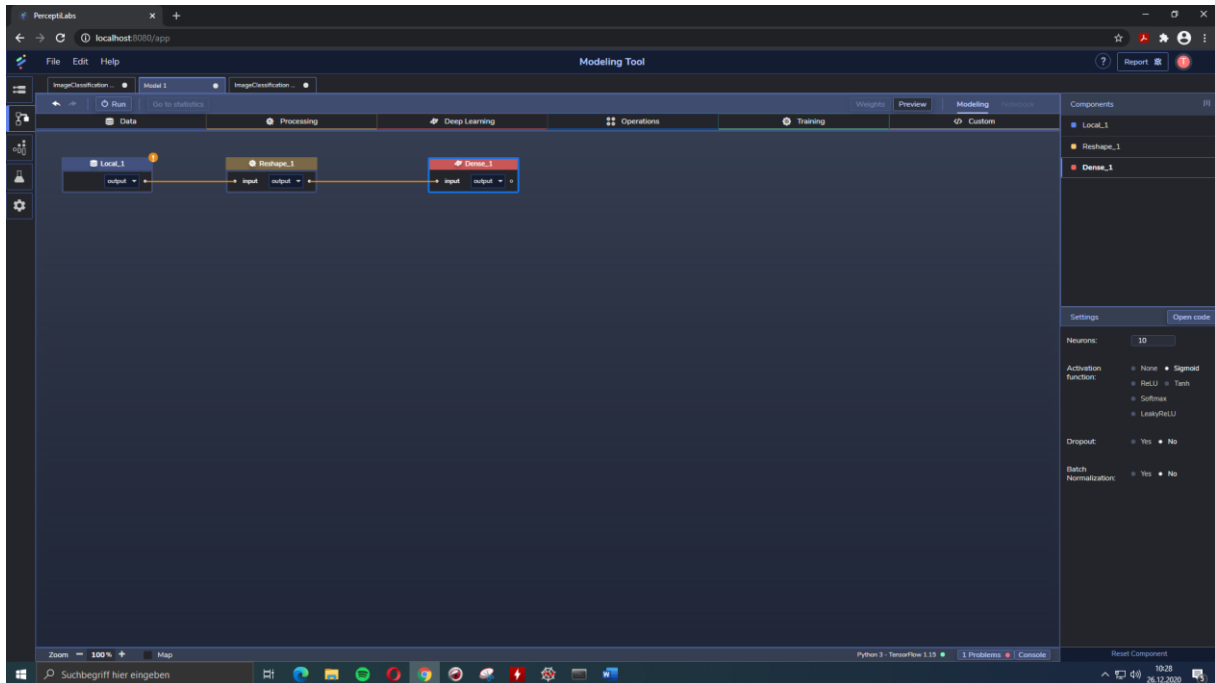
Schritt 1: Mit Klick auf Data Local Data auswählen und an eine beliebige Position ziehen.



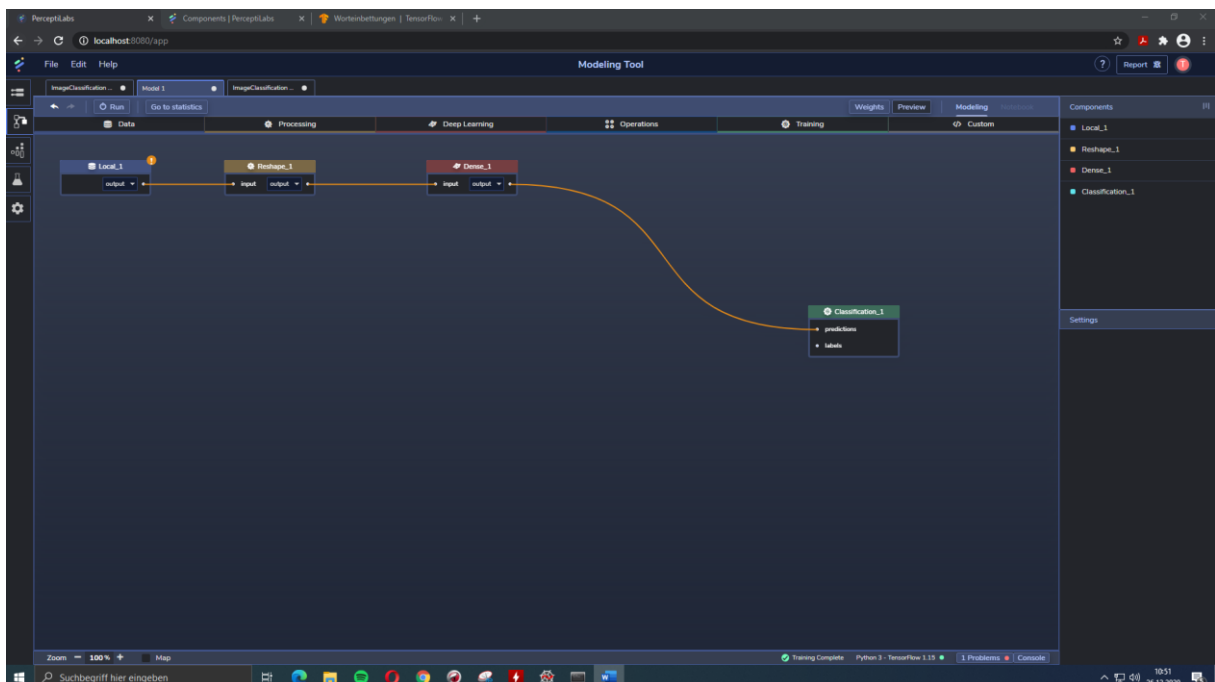
Schritt 2: Abhängig von den Daten muss eine Datenvorverarbeitung stattfinden. Eine Skalierung der Daten kann z.B. durch Klick auf Processing und dann auf Reshape in das Netz eingefügt werden. Wichtig: Die beiden Module müssen miteinander verbunden werden. Dies kann durch die Kreise auf der rechten Seite der Module durchgeführt werden.



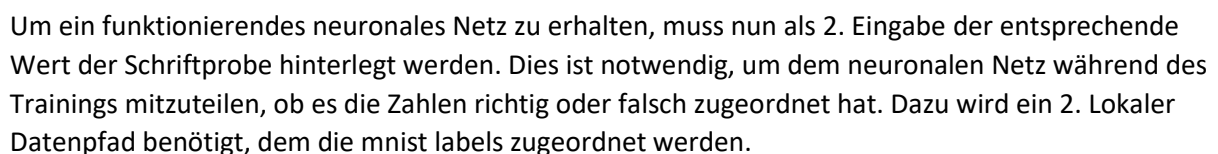
Anschließend muss der Deep Learning Bereich des neuronalen Netzes designt werden. Hier kann z.B. durch Klicken auf Deep Learning und Auswahl von Dense eine Hidden Layer hinzugefügt werden. Auch dieses Modul muss wieder mit dem vorherigen Modul verbunden werden.

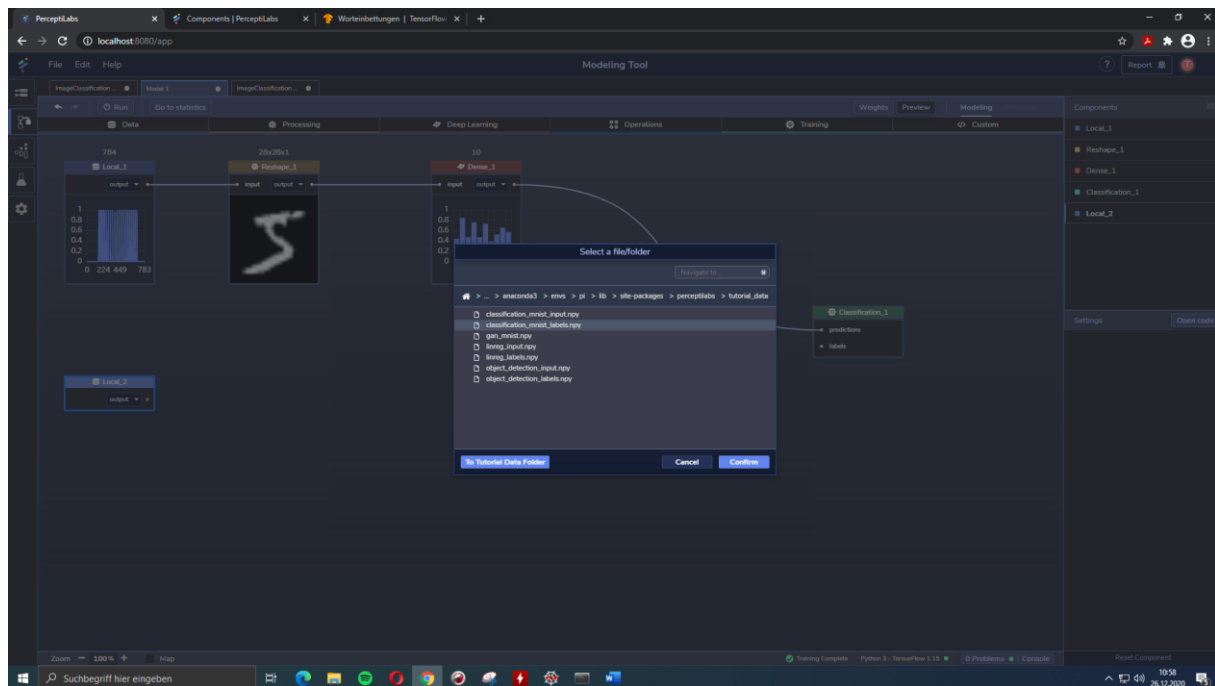


Mit Klick auf Training kann ausgewählt werden welchen Einsatzzweck das neuronale Netz z.B. haben soll. Hier kann man z.B. im Bereich Training eine Klassifizierung auswählen und mit dem neuronalen Netz verbinden.

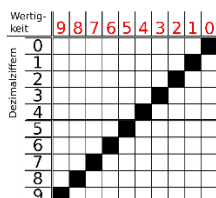


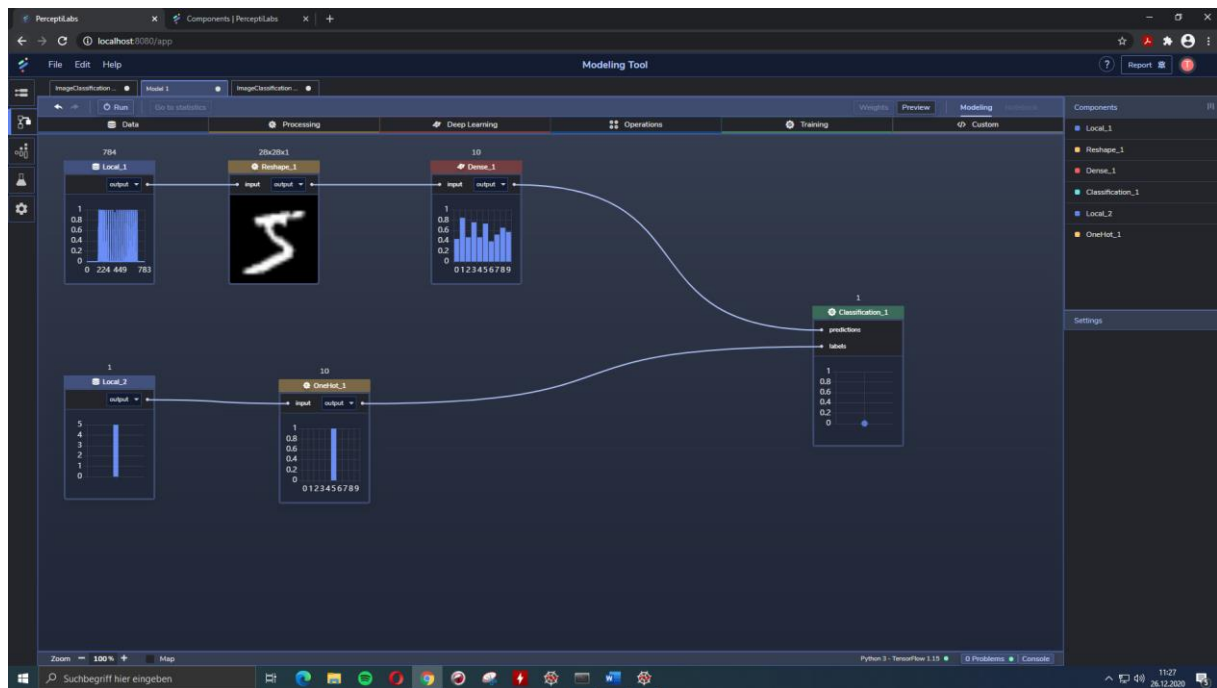
Soll z.B. eine Klassifizierung aus Zeichen der MNIST Datenbank durchgeführt werden, so kann der in Perceptilab hinterlegte Datensatz verwendet werden. Dazu auf Local und anschließend auf Load Data klicken. Es öffnet sich folgendes Fenster





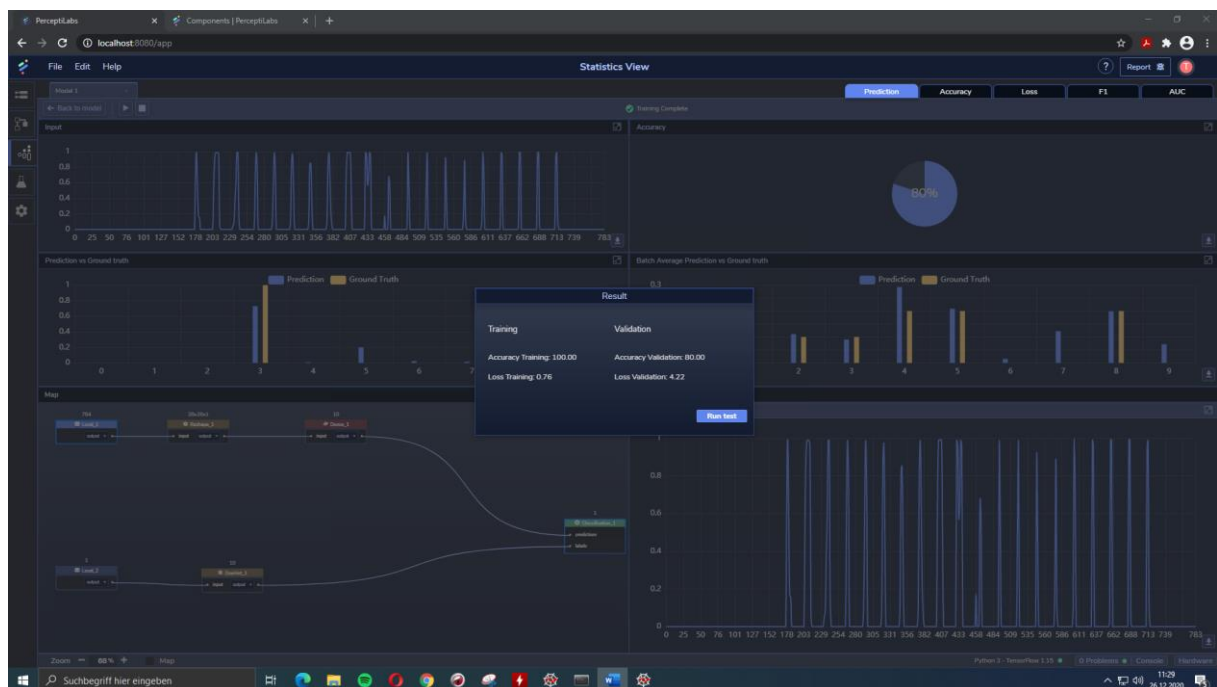
Da es sich hierbei um die Ergebnisse handelt, die als Zahlenwert dem neuronalen Netz zugeführt werden, ist hier kein Reshape notwendig. Stattdessen wird eine OneHot-Codierung durchgeführt. Dies ist notwendig, um die Zahlen binär darstellen zu können. Dabei wird allen Bits mit Ausnahme von einem eine 0 zugeordnet. Die stelle mit der 1 entspricht dabei dem Wert der Dezimalzahl. Dies ist in folgender Grafik zu sehen (schwarz=1):





Damit ist das neuronale Netz fertiggestellt und kann durch Klicken auf Run trainiert werden.

Nach dem Training ist folgendes Feld zu sehen.



Mit dem eben erstellten neuronalen Netz wird auf Basis der Validierungsdaten eine Genauigkeit von 80% erreicht.

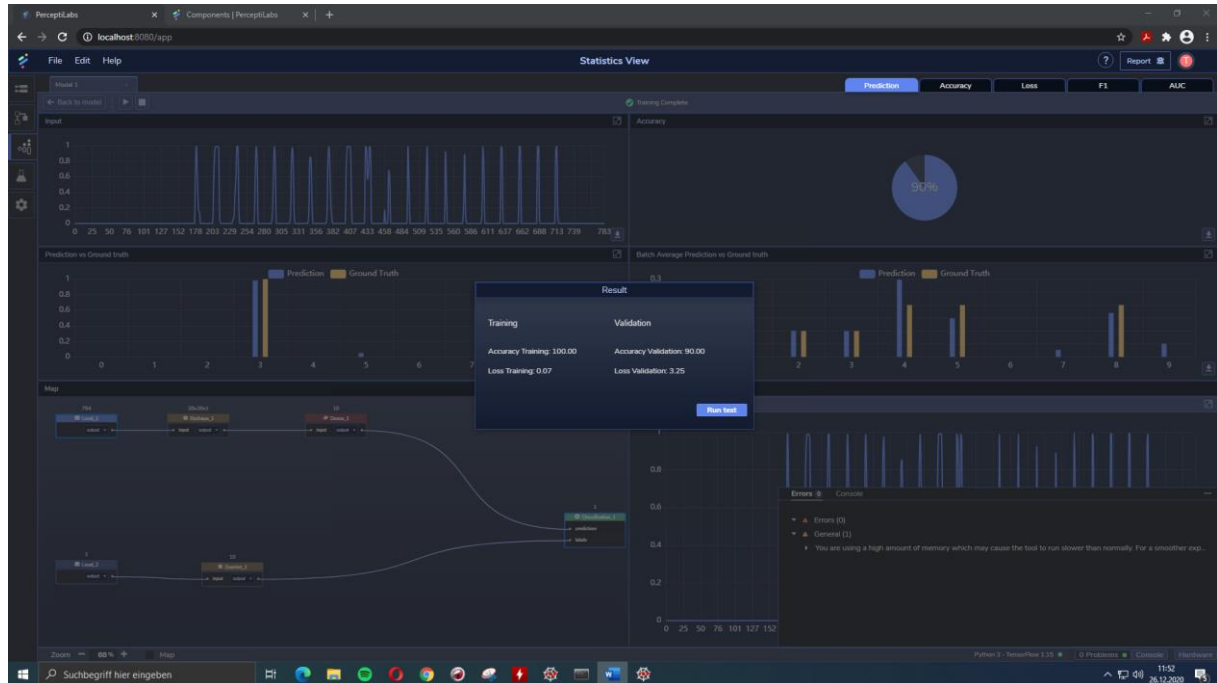
3. Modifikation der Parameter

Durch Modifikation des neuronalen Netzes kann dessen Genauigkeit beeinflusst werden.

Exemplarisch wird dies hier auf Basis der Epochen und Aktivierungsfunktionen demonstriert.

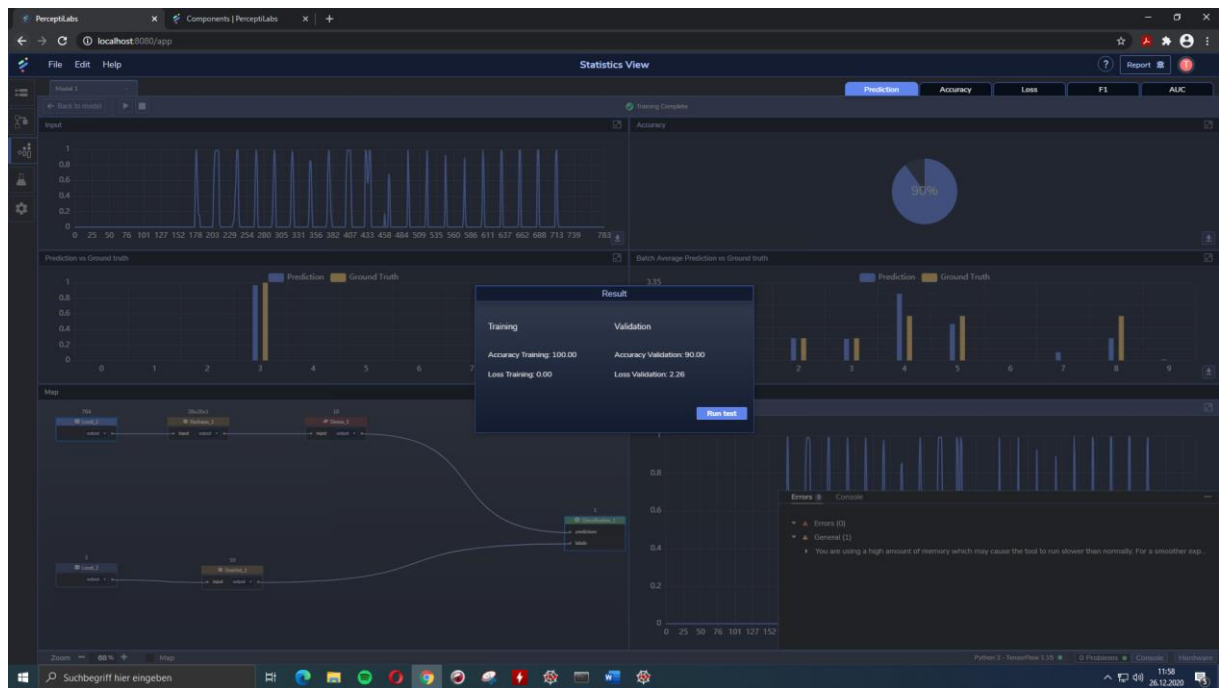
3.1. Variation der Epochen:

Die Genauigkeit lässt sich von 80% auf 90% erhöhen, wenn die Epochen von 10 auf 100 erhöht werden.

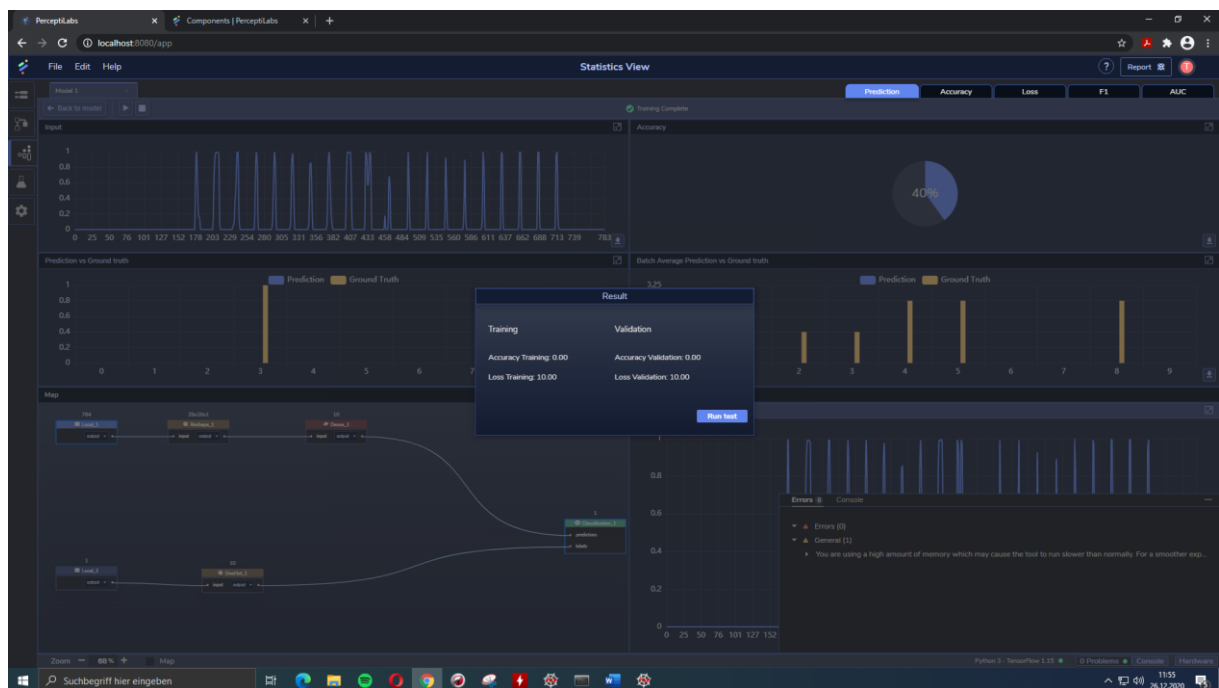


3.2. Variation der Aktivierungsfunktionen

Ebenfalls erhöht sich die Genauigkeit, wenn die Aktivierungsfunktion von Sigmoid zu Softmax geändert wird.



Wird jedoch die Aktivierungsfunktion von Sigmoid auf ReLU geändert, ist keine Zuordnung mehr möglich.



4. Export des neuronalen Netzes

Der Export des neuronalen Netzes kann auf 2 Arten erfolgen.

1. Als Tensorflow Model
2. Jupyter Notebook.

Dazu auf File und anschließend auf Export klicken, den Dateipfad und die Art des Exports (Tensorflow Model oder Jupyter Notebook) auswählen und auf Export klicken.

