

Klassifikation von Erkrankungen der Retina anhand von OCT Bildern

18. Juli 2018

TU Dortmund
Physik

Aufgabenstellung

Fragestellung

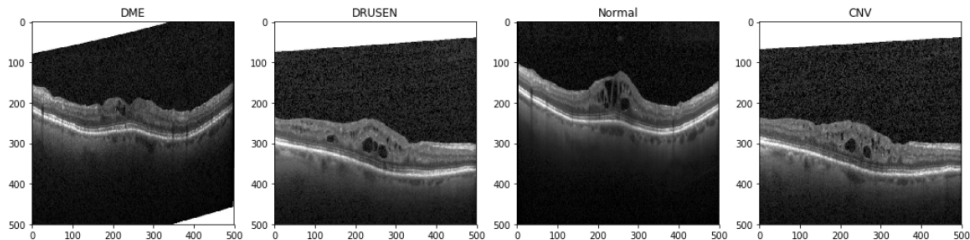
Lsst sich der Zustand einer menschlichen Retina anhand von OCT-Bildern in die 4 Klassen

[NORMAL, CNV, DRUSEN, DME]

einteilen und somit eine Diagnose mit Hilfe von ML stellen?
choroidal neovascularization, macular edema, drusen

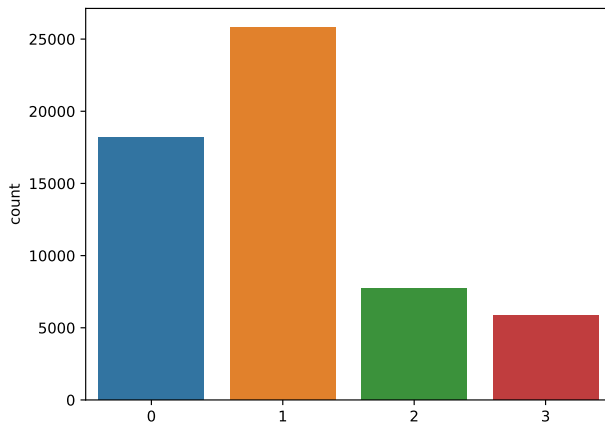
- CNV: *Bildung neuer Blutgefäße im Auge*
 - DRUSEN: *Ablagerung von proteinhaltigem Material, das verkalkt*
 - DME: *Ansammlung extrazellulärer Flüssigkeit im Bereich des menschlichen Auges*
-
- *optical coherence tomography*: hoch auflösende Bildgebung von Retina Querschnitten lebender Patienten
 - Analyse durch erfahrenen Mediziner notwendig
 - Idee: Methoden maschinellen Lernens um Krankheiten zu erkennen

Datensatz



- Inhalt: 84,495 Rntgen Bilder (500×500 px JPEG) aufgeteilt in 4 Klassen (Erkrankungen + gesund)

Verteilung auf die Klassen



Architektur des Netzes

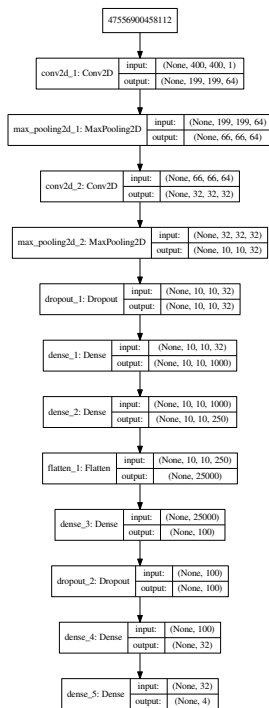
- 2 (Convolutional 2D layer + MaxPooling2D layer) *Pakete* (64/32 Filter)
- Kernel sizes: (4, 4) / (3, 3)

bergang zu Fully connected layern

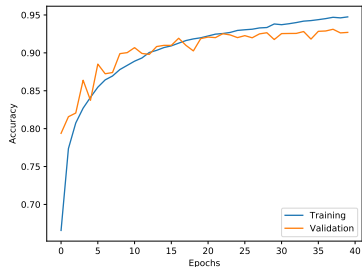
- Dropout mit rate=0.25
- Dense layer: Aktivierungsfunktion **elu**
- Filter: 1000, 250, 100, 32, 4
- Unterbrochen von einem flatten layer und weiterem Dropout (rate=0.5)
- Output layer: 4 Filter mit Aktivierungsfunktion *softmax*

Architektur des Netzes

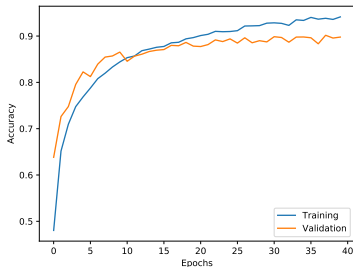
Layer (type)	Output Shape	Anzahl Parameter
Conv 2D	(None, 199, 199, 64)	1088
Max Pooling 2D	(None, 66, 66, 64)	0
Conv 2D	(None, 32, 32, 32)	32800
Max Pooling 2D	(None, 10, 10, 32)	0
Dropout	(None, 10, 10, 32)	0
Dense	(None, 10, 10, 1000)	33000
Dense	(None, 10, 10, 250)	250250
Flatten	(None, 25000)	0
Dense	(None, 100)	2500100
Dropout	(None, 100)	0
Dense	(None, 32)	3232
Dense	(None, 4)	132
Total		2820602



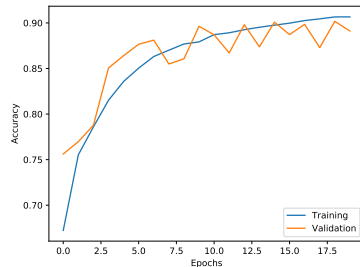
Ergebnisse Convolutional network



(a) elu whole

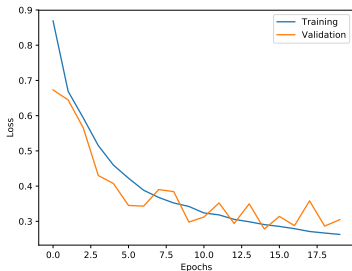


(b) elu equal

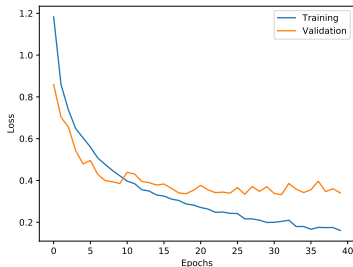


(c) relu

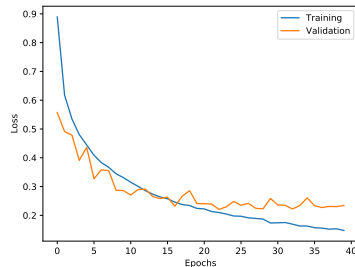
Ergebnisse Convolutional network



(a) relu

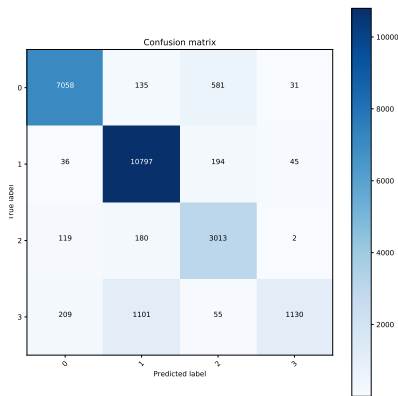


(b) elu equal

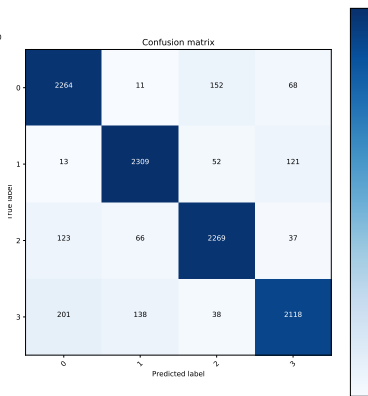


(c) elu

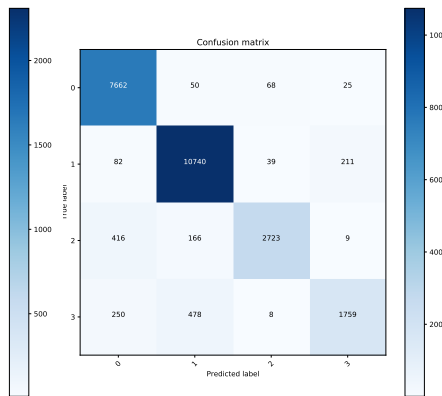
Ergebnisse Convolutional network



(a) relu



(b) elu equal



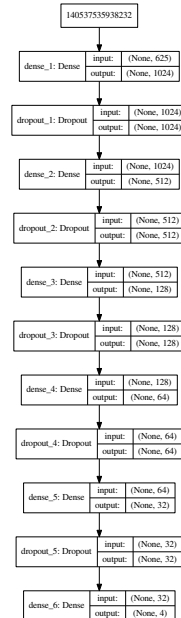
(c) elu

Alternative Methode

- Datenvorbereitung:
 - Gelabelte Bilder (50,100) in hdf5-Format abgespeichert
 - Feinere Krnung bringt keine Verbesserung!
 - Gewichte fr jedes Bild \Rightarrow Ausgleich der Unterschiede der Klassenmenge
- Drei Anstze:
 - Fttere Pixel nacheinander ins neuronale Netz (\Rightarrow (zu) viele Inputfeature Bilder grer als (50,100))
 - Berechne sowohl in x - und y -Richtung die Mittelwerte aller Pixel einer Linie ((100,100) Bild \rightarrow 200 Werte)
 - Definiere Fenster und berechne Mittelwerte der im Fenster liegenden Pixel ((50,100) abgerastert mit (2,4) Fenster \Rightarrow 625 Inputfeature)
- Letzte Methode am vielversprechendsten!
- Werte eines Bildes werden auf den maximalen Wert eines Bildes normiert

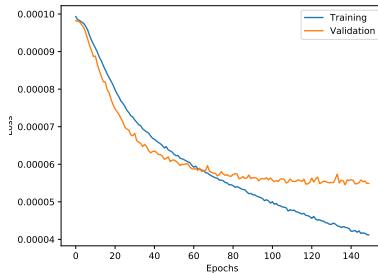
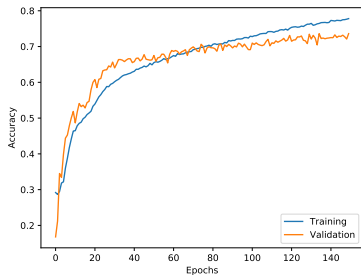
Referenzstruktur des Neuronalen Netzes

- Vollständig vernetztes NN bestehend aus Dense Layer
- Festgelegte Referenzstruktur \Rightarrow
- Dropoutrates:
 - Nach 1. Layer: 0.5
 - Nach 2. Layer: 0.4
 - Nach 3. Layer: 0.3
 - Nach 4. Layer: 0.2
 - Nach 5. Layer: 0.2
- Aktivierungsfunktionen:
 - Hidden Layer: relu
 - Outputlayer: softmax
- Loss-Funktion: Kategorische Entropie
- Adam mit angepasster Lernrate von 0.0001 als Optimierer



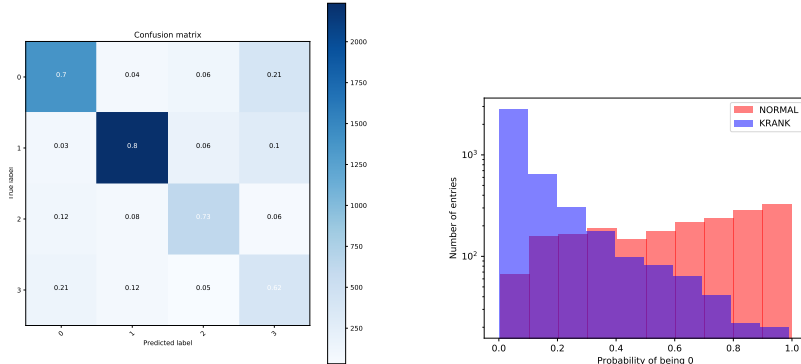
Performance des Referenznetzes

- Trainingsdatensatz: 67.5 % des Datensatzes
- Validierungsdatensatz: 25 % des Datensatzes
- Testdatensatz: 7.5 % des Datensatzes



- Sttigung ab ca. 80 Epochen auf dem Validierungsdatensatz
- Erreicht ca. 71 % Genauigkeit beim Validierungsdatensatz

Performance des Referenznetzes



- ⇒ Gute Unterscheidung zwischen kranken und gesunden Augen möglich
- ⇒ hnliche Struktur der Verwirrungsmatrix wie bei nomineller Methode

Laufende Grid Search

- Optimierungsparameter:
 - Batchgre: 50, 64, 100, 128, 256, 512
 - Aktivierungsfunktion: elu oder relu
 - Outputaktivierungsfunktion: softmax oder sigmoid
 - Layerstruktur (Dropout & Dense)
- ⇒ 120 verschiedene Netzwerkkonfigurationen werden getestet
- Modelle in .json und trainierte Gewichte in .hdf5 Files abgespeichert
- Erste Prognose: ca. 78 % ist drin