

Residual Networks



Alexander Kowsik

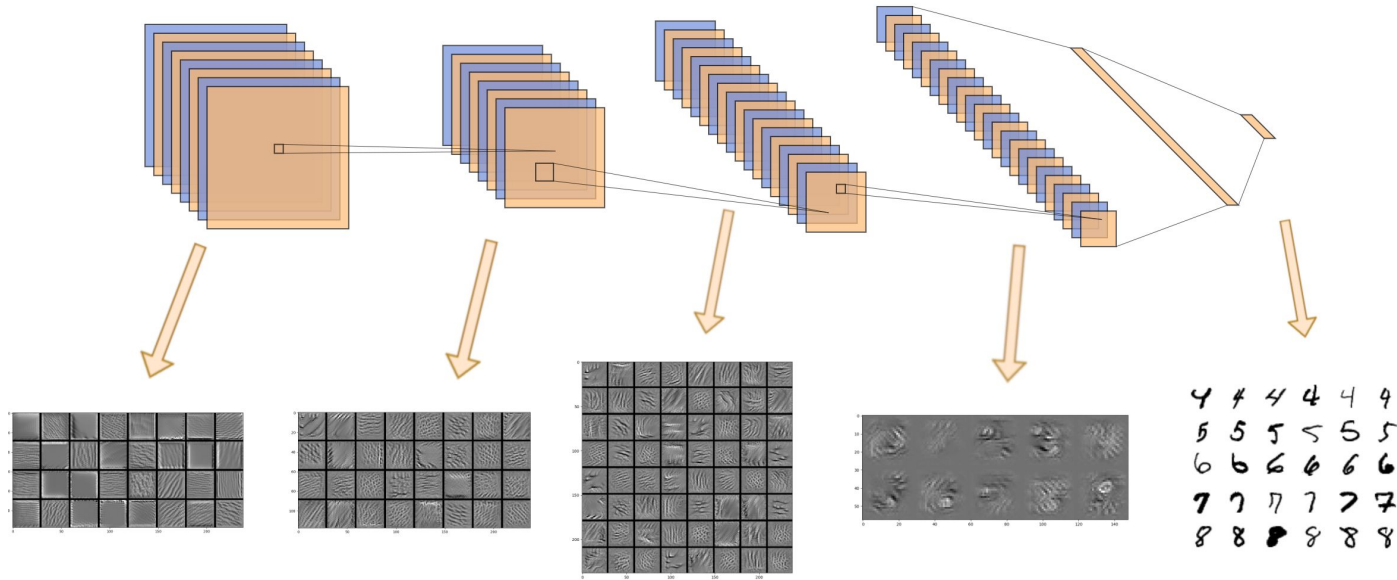
20.01.2021

Outline

1. Trainieren von sehr tiefen Netzen
2. Residual Neural Networks
3. Varianten von ResNets

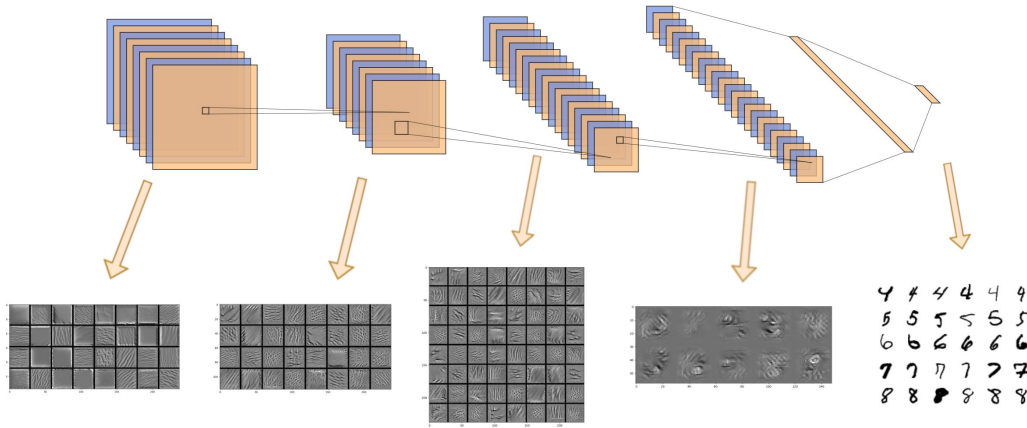
Trainieren von sehr tiefen Netzen

Warum möchte man *tiefe* Netze?



→ immer abstraktere Features →

Warum möchte man *tiefe* Netze?

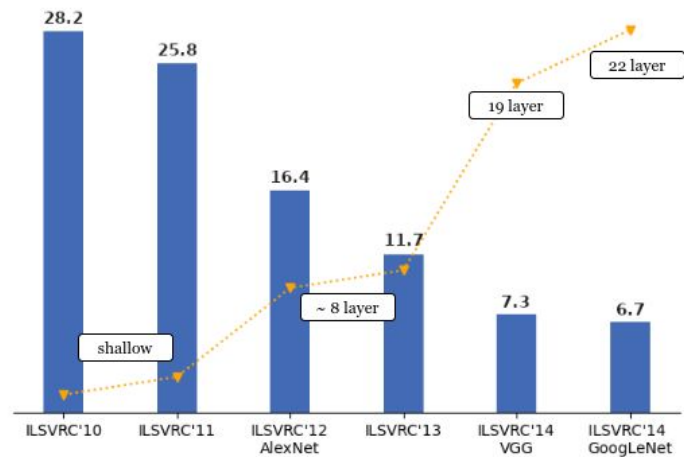


Neuronale Netze sind
**universelle
Funktions-
approximatoren**

Mehr Schichten bedeutet

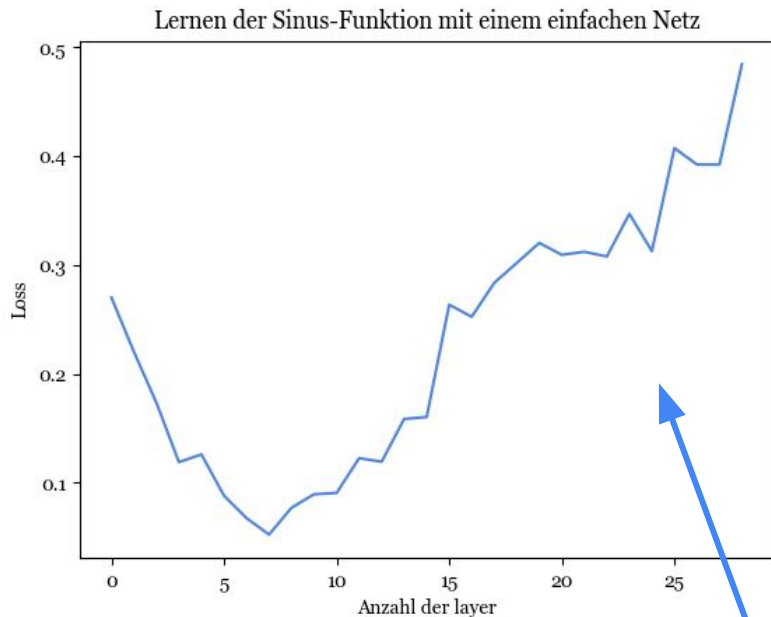
- mehr **abstraktere Features**
- **einfacheres Lernen** von komplexen Funktionen

ImageNet Competition 2010-2015



mehr Schichten = immer bessere Performance?

Degeneration Problem



hohe Trainings- & Test-Error

Overfitting?

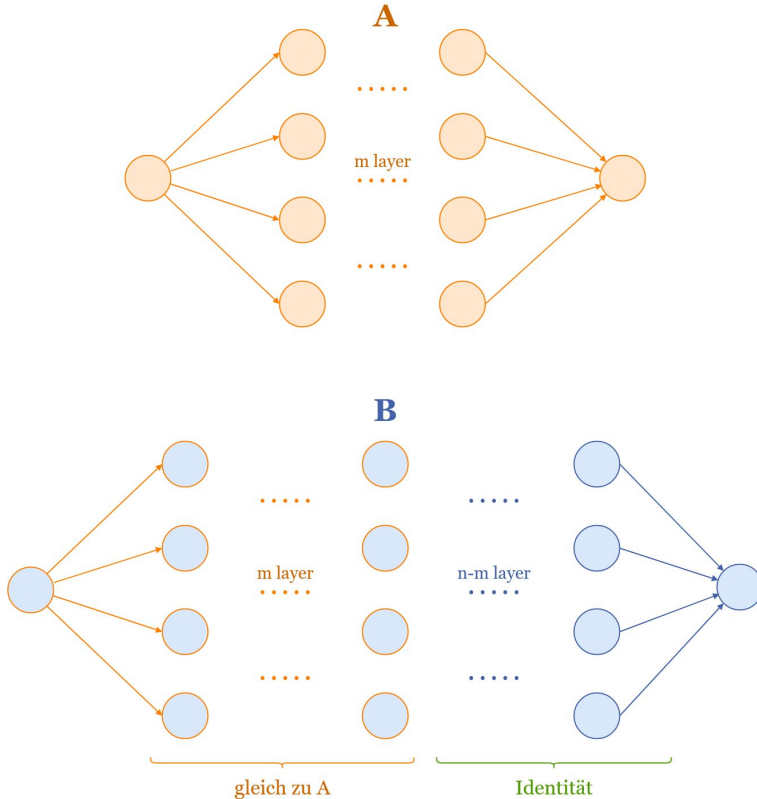
→ **Nein. Regularisierung hilft nicht!**

Vanishing/exploding gradients?

→ **Teilweise**

⇒ **Optimierung +
Parameter-Initialisierung**

Tief vs. weniger tief



Initiale Parameter sind
normalverteilt mit Mittelwert 0

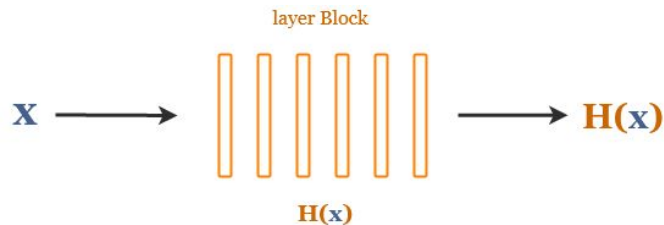
→ Nullfunktion einfach,
Identität schwer

0	0	0
0	1	0
0	0	0

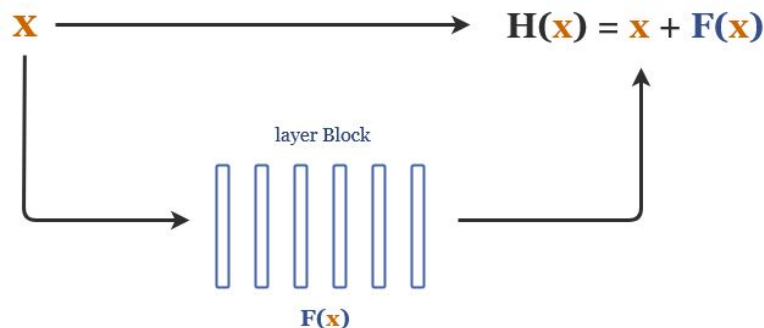
Beispiel

Residual Networks

Lernen von Residuen



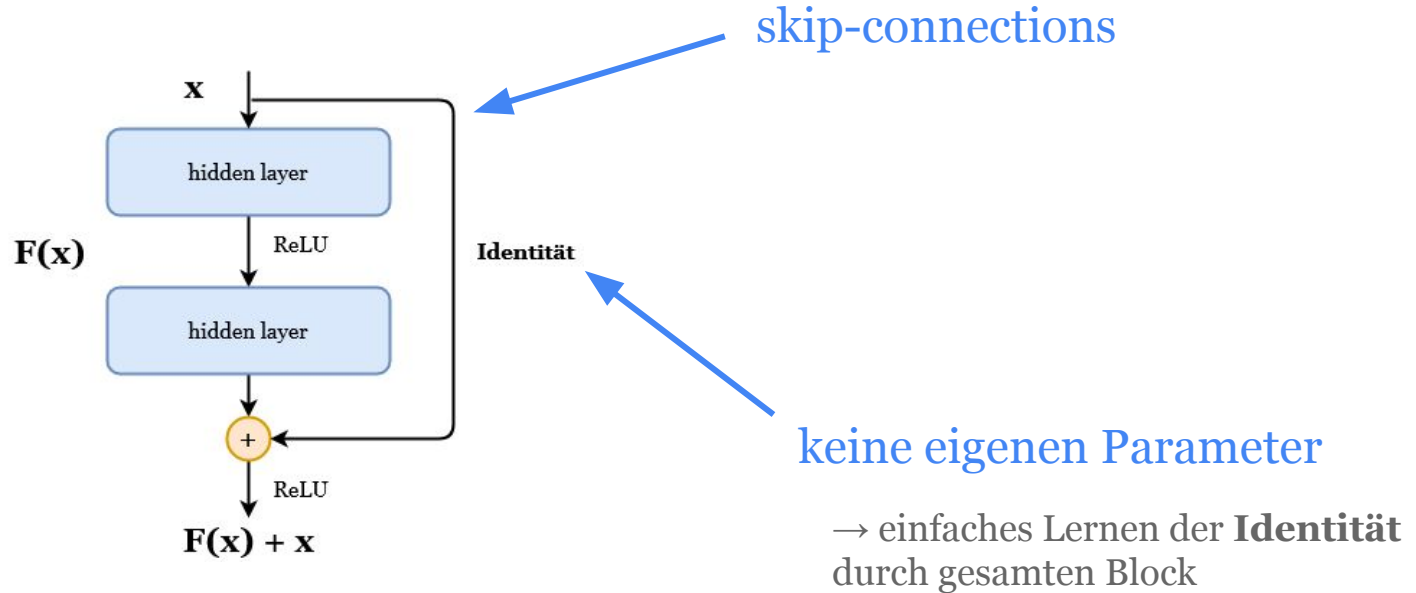
Reformulierung des
Lernprozesses



Lernen von **Residuen** ist einfacher!

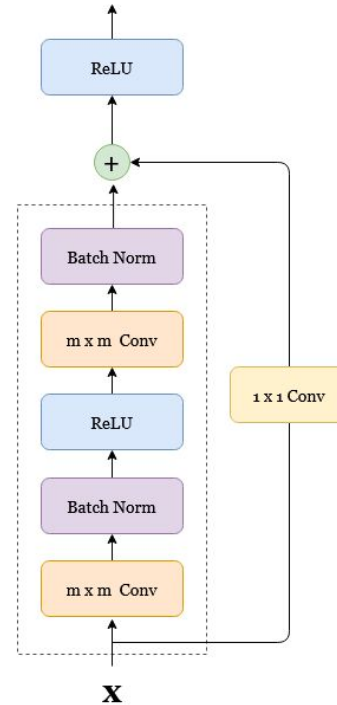
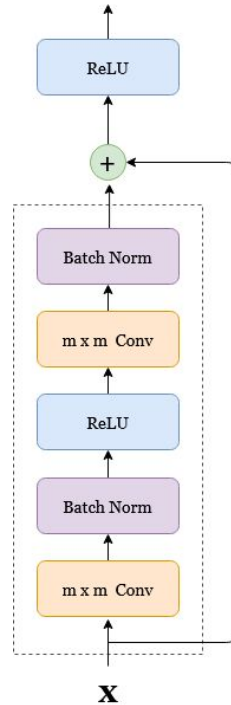
Lernen der **Identität** ist einfacher!

Aufbau eines ResNet Blocks



Realisiert **Lernen von Residuen**

Anpassung der Dimension

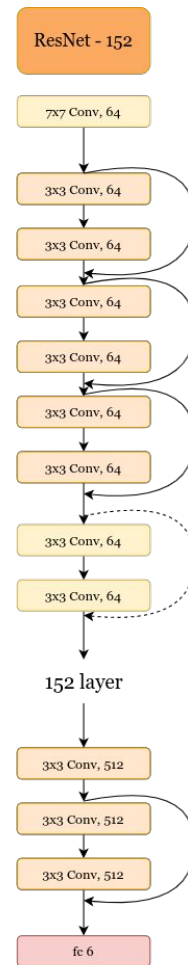
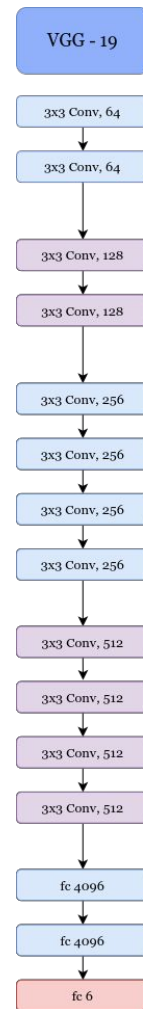


alternativ Padding

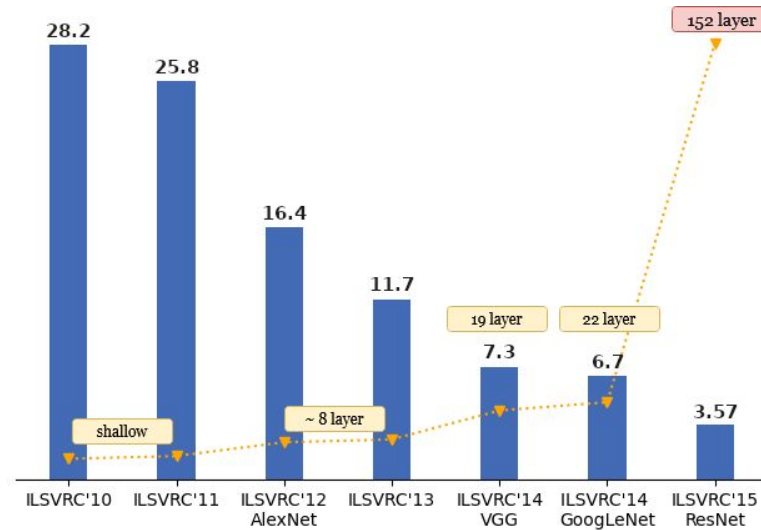
Beispiel: ResNet-152

8x mehr Layer als das VGG-19

Trotzdem viel weniger Parameter!

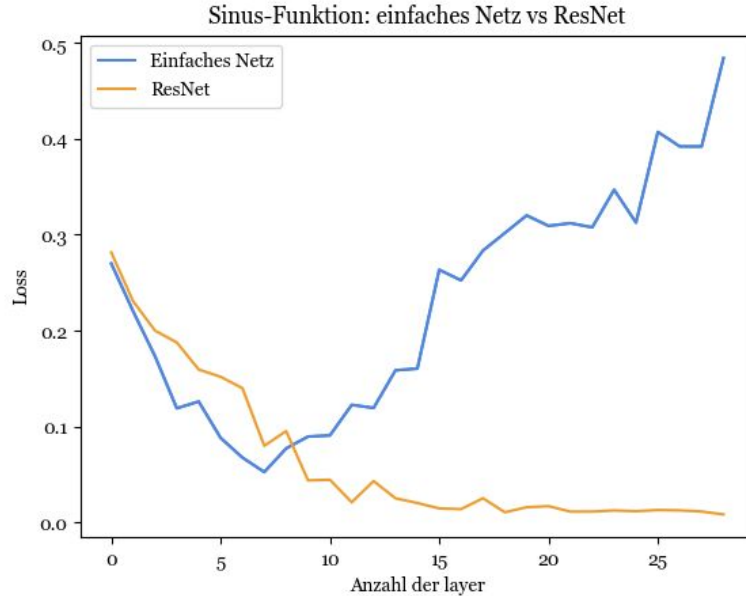


Performanz von ResNets



ImageNet Competition 2015

Performanz von ResNets



mehr Schichten
=
bessere
Performance

Grenzen: Overfitting, lange Trainingszeit

Warum genau funktioniert das nochmal?

- Lernen von Residuen scheint einfacher zu sein
- Einfacheres Lernen der Identitätsfunktion
- \Rightarrow Überspringen der Layer, die der Performance schaden
 - \rightarrow keine Hyperparameter für Anzahl der Layer benötigt
 - \rightarrow ähnelt Training eines Ensembles
- Stets nur ein fine-tuning der Eingabe eines Blocks

Training von ResNets

skip-connections (ohne Parameter) einzige Änderung

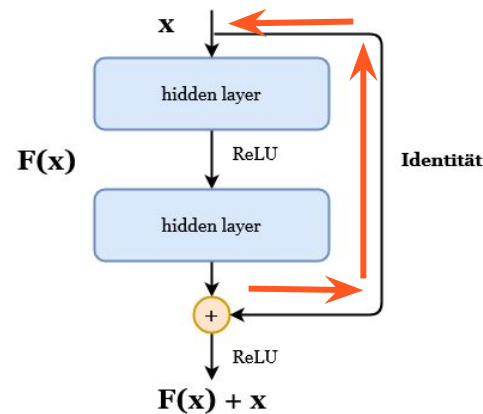
→ unverändertes Training mit herkömmlichen Methoden

Gradienten können viel besser zu vorigen Schichten “fließen”

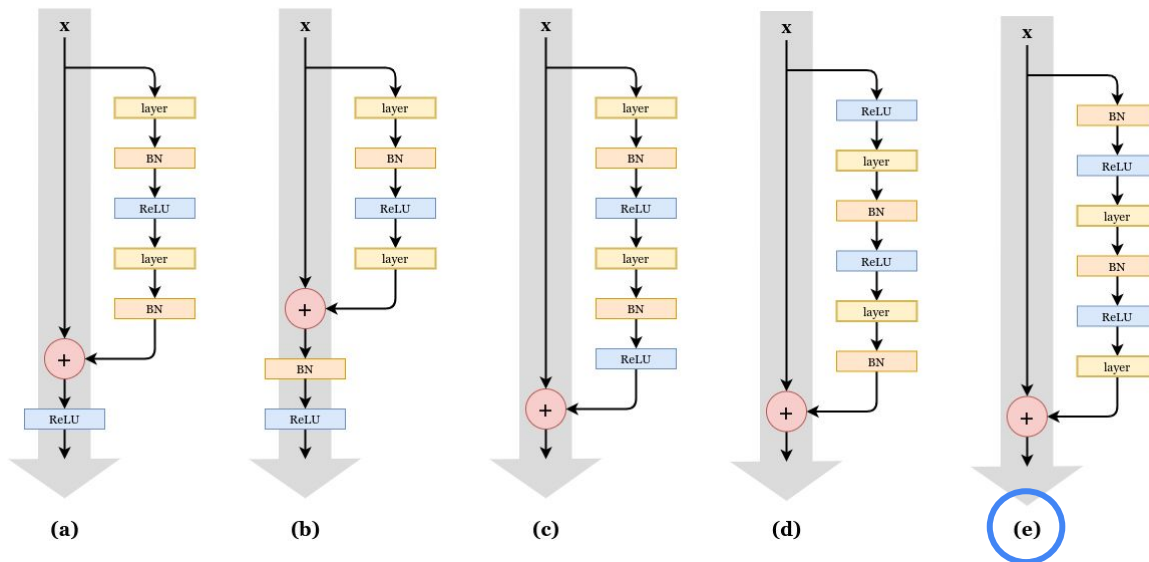
→ effizienteres Training, weniger vanishing/exploding gradients

zwar **mehr Schichten**, i.d.R. jedoch **viel weniger Parameter**

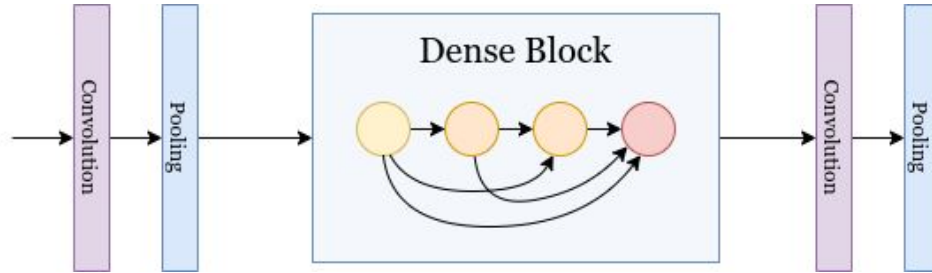
→ kürzere Trainingszeiten



Varianten von ResNets



Aufbau, Varianten und Architekturen



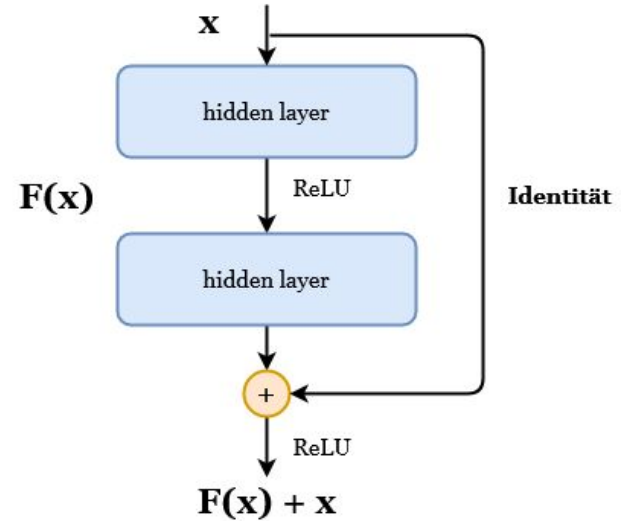
DenseNets: Verbindung zwischen allen Layern in einem Block

HighwayNets: gate-Parameter für skip-connections

ResNeXt: parallele Schaltung von ResNet Blöcken für ein 'weiteres' Netz

Zusammenfassung

- Reformulierung des Lernprozesses
→ **Lernen von Residuen**
- **Skip-connections**
- Mehr layer = bessere Performance



ResNets ermöglichen sehr tiefe Netze