

# Digital Design for Machine Learning

Mikail Yayla, Michael Krekker

**Department of Computer Science, Chair 12  
TU Dortmund University, Germany**

October 14, 2021

## Machine Learning

- Generierung von Wissen aus Erfahrung
- ML-Algorithmen konstruieren mithilfe von Trainingsdaten mathematische Modelle
- Tools: Trainings- und Testdaten, SW

## Digital Design

- Hardwaresysteme und Komponenten entwickeln und testen
- Tools: HDL, SW für Analyse und Synthese von HDL, FPGAs
- Arbeit auf unterschiedlichen HW-Abstraktionsebenen: Gatter, Register, RAM, etc.

## Generierung eines ANNs

- Anwendungsbereiche: Bilderkennung, Spracherkennung, Texterkennung, etc.
- Wähle geeignetes Modell (ANN Architektur) und Hyperparameter
- Optimierte Parameter von ANN mithilfe Trainingsinputs, Zielwerten, und Zielfunktion mit DL-SW und spezieller HW
- Teste die Accuracy des ANNs mithilfe von Testinputs

## Trainiertes ANN: Inferenz

- Ausführen des Modells basierend auf bestimmten Inputs
- Output des Modells sind gewünschte Ausgaben

**Ziel 1:**  
**NN erstellen für bestimmtes Problem**

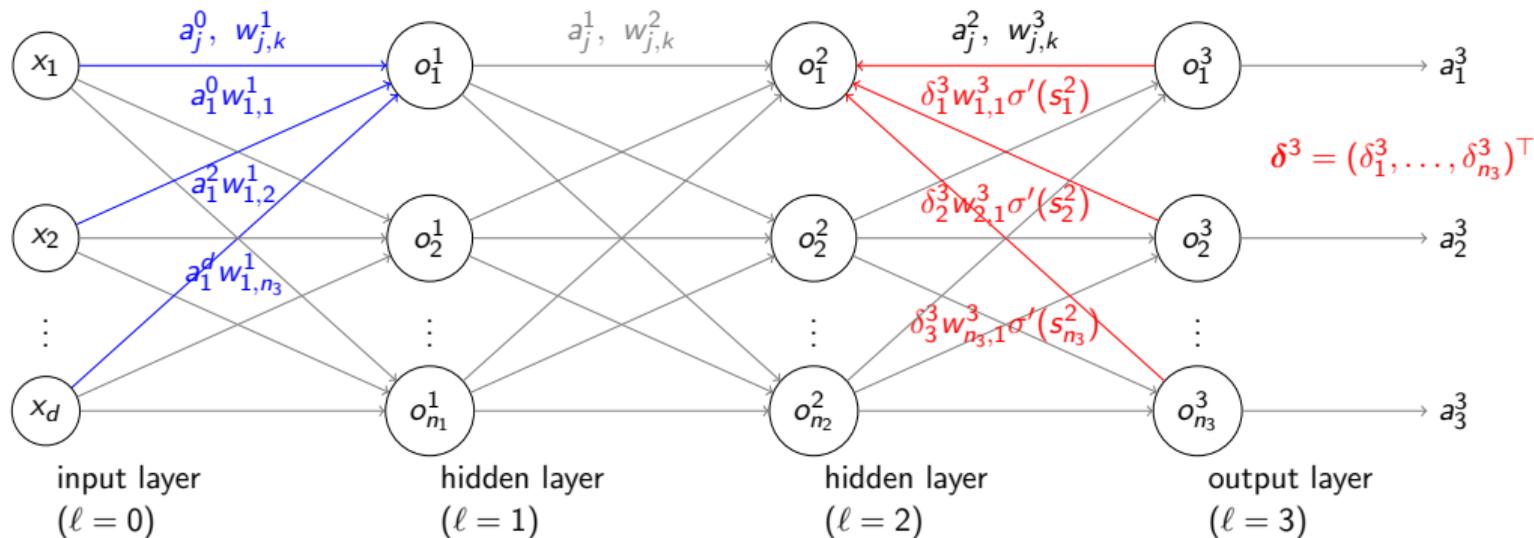
# Berechnungen in NNs: Forward und Backward

$$a_j^l = \sigma_j^l \left( \sum_{k=1}^{n_{l-1}} w_{j,k}^l a_k^{l-1} + b_j^l \right)$$

$$\delta_j^2 = \sum_k \delta_k^3 w_{k,j}^3 \cdot \sigma'(s_j^2)$$

$$w_{1,1}^3 = w_{1,1}^3 - \lambda a_1^2 \delta_1^2$$

$$\delta_j^3 = \frac{\partial \mathcal{L}}{\partial s_j^3} = \frac{\partial \mathcal{L}}{\partial a_j^3} \sigma'(s_j^3)$$



# Matrixmultiplikationen in NNs

$W_\ell$ -matrix for weights of layer,  $A_{\ell-1}$ -matrix for Activations

$$W_\ell = \begin{pmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,m_\ell} \\ w_{2,1} & w_{2,2} & \dots & w_{2,m_\ell} \\ \vdots & \ddots & & \\ w_{n_\ell,1} & w_{n_\ell,2} & \dots & w_{n_\ell,m_\ell} \end{pmatrix}, \quad A_{\ell-1} = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,q} \\ a_{2,1} & a_{2,2} & \dots & a_{2,q} \\ \vdots & \ddots & & \\ a_{m_{\ell-1},1} & a_{m_{\ell-1},2} & \dots & a_{m_{\ell-1},q} \end{pmatrix}$$

$$S = W_\ell \times A_{\ell-1}$$
$$a_{j,k}^\ell = \sigma_j^\ell(s_{j,k}^\ell) = \sigma_j^\ell\left(\sum_{k=1}^{n_{\ell-1}} w_{j,k}^\ell a_{k,j}^{\ell-1}\right)$$

# Wie kann man Hardware Designen?

---

## Digital Design von komplexen Systemen im Fachprojekt

- In RS/HaPra gab es Designs mit mäßiger Komplexität
- Komplexe Designs benötigen mehr Arbeit
- Mehrere Arbeitsgruppen, Mitglieder haben bestimmte Aufgaben
- Lernen, wie man komplexe Aufgaben in einer größeren Gruppe löst

## Zerlegung der Probleme

- Design-Plan aufzeichnen, Komponenten zerlegen bis einfache Probleme übrig
- Einfache Komponenten implementieren und testen
- Mehrere einfache Komponenten miteinander verbinden und testen
- Komplexität verstecken: komplexe Komponente wird einfach

# VHDL (VHSIC Hardware Description Language)

---

## Hardwarebeschreibungssprache

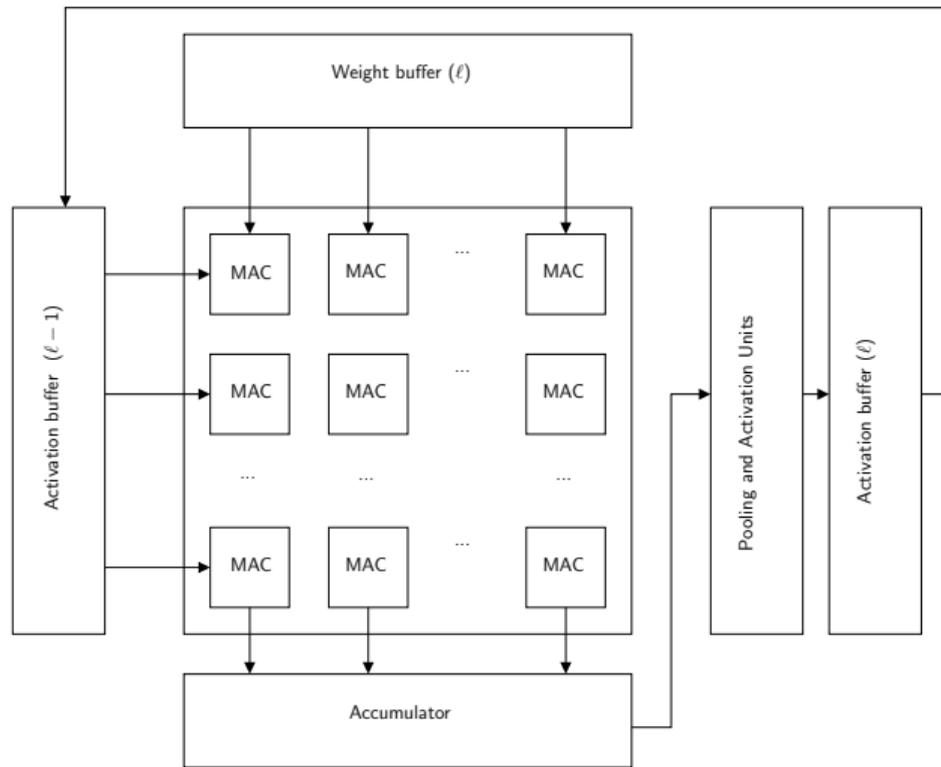
- Beschreiben von digitalen Schaltungen
- Simulation von Schaltungen für Tests
- Kann synthetisiert werden für echte HW

## Vorteile von VHDL

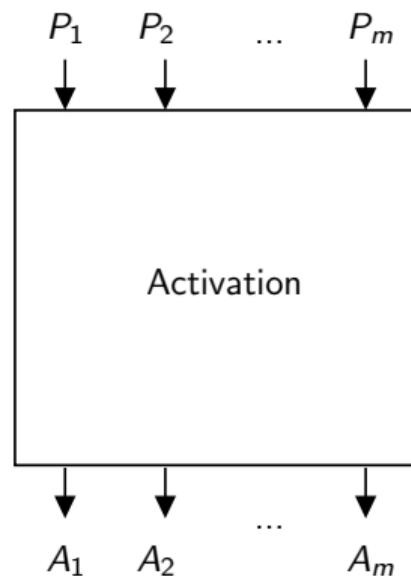
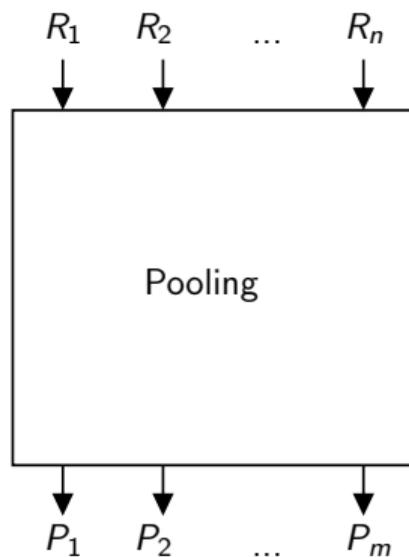
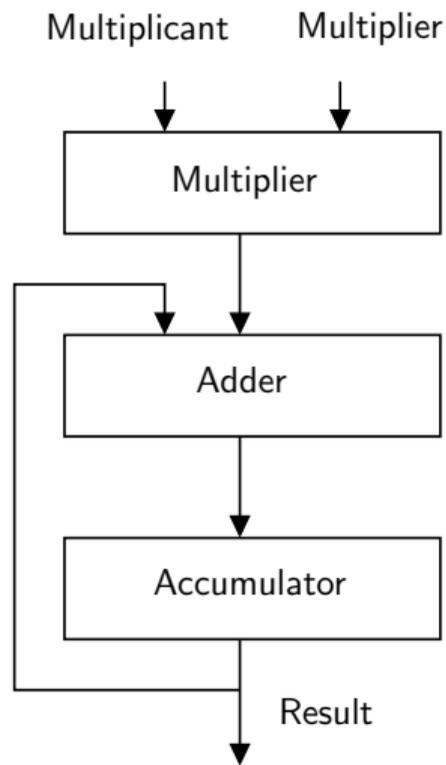
- Verbos, aber relativ wenig Debugging nötig
- In der Industrie weit verbreitet
- Funktioniert mit GHDL und GTKWave

**Ziel 2:**  
**HW-Design für Ausführung von NNs**

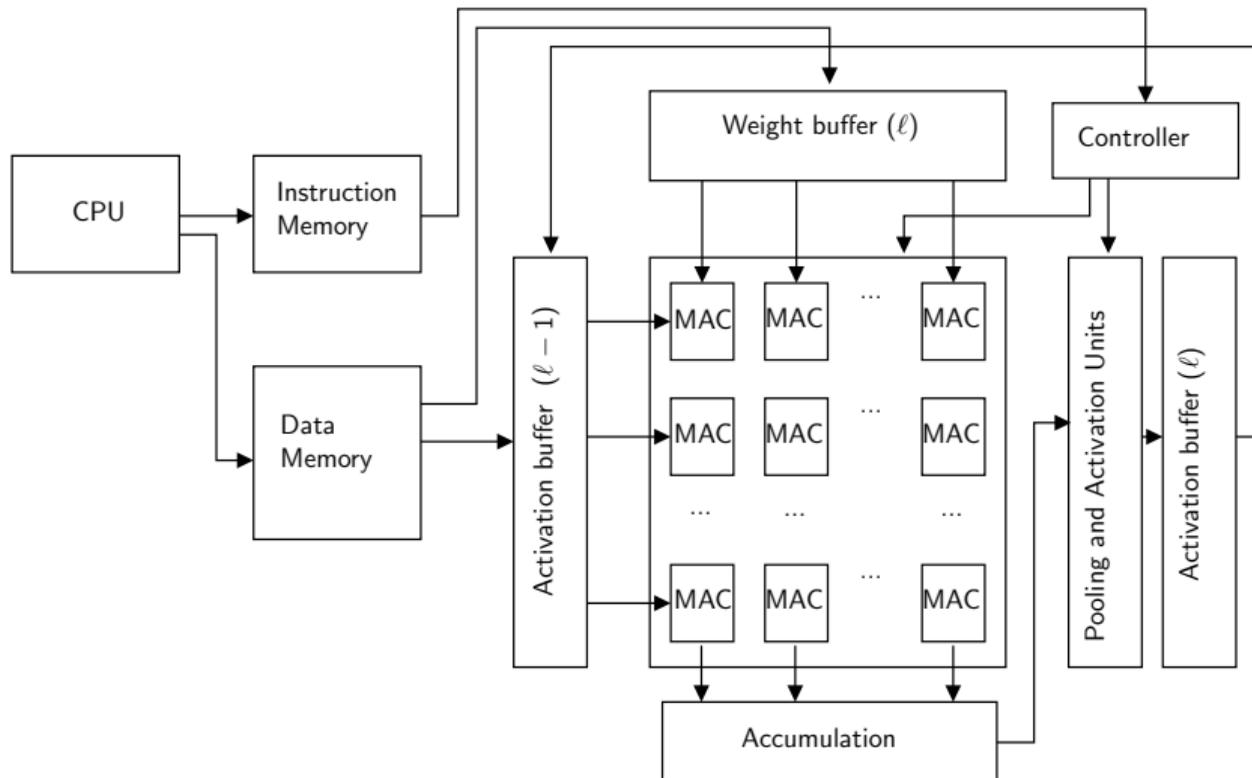
# Genereller Überblick: HW zur Ausführung von NNs



# Einige Einzelprobleme: MAC-Unit, Pooling, Activation



# HW zur Ausführung von NNs mit CPUs, Speichern, Controller



# Semesterplan für das Fachprojekt

---

Beschreibung	Termin
Einführungsveranstaltung	13.10.21, 16 Uhr
Vorträge von Michael und Somar	20.10.21, 16-18 Uhr
Seminarvorträge	27.10.21, 16-18 Uhr
Diskussion der Projektpläne	03.11.21, 16-18 Uhr
Meetings für Berichte	jede zwei Wochen (10.11.21 - Ende)

## Vier Arbeitsgruppen

- Gruppe 1: Optimierung eines NNs
- Gruppe 2: Post-training Quantisierung/Schnittstelle zwischen DL-SW und VHDL
- Gruppe 3: Processing Logic Design
- Gruppe 4: CPU und Controller Design

## Themenbereich 1: ML-Software

- Einführung in NNs
- Training-Software für NNs (PyTorch)
- Berechnungen in NNs und NN Modelle
- Beispiele für ML-Anwendungen

## Themenbereich 2: Schnittstelle zwischen SW/HW

- Post-Training Quantisierung
- Data Movement und Mapping von Workloads aus ML-SW zu HW

## Themenbereich 3: Processing Logic

- Addierer, Multiplizierer, MAC-units, Pooling, Activation
- Buffer, FIFOs, Systolic Array Organization von MAC-units

## Themenbereich 4: CPU and Control

- Pipelined RISC-V CPU
- Kontrolleinheit

# Weitere TODOs

---

## Moodle-Raum

- Materialien
- Emails

## Seminarthemen

- Zweiergruppen, 15 Minuten für einen Vortrag mit Folien (7-8 Min./Person)
- Erklärungen zu Seminarthemen
- Verteilung der Seminarthemen