TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

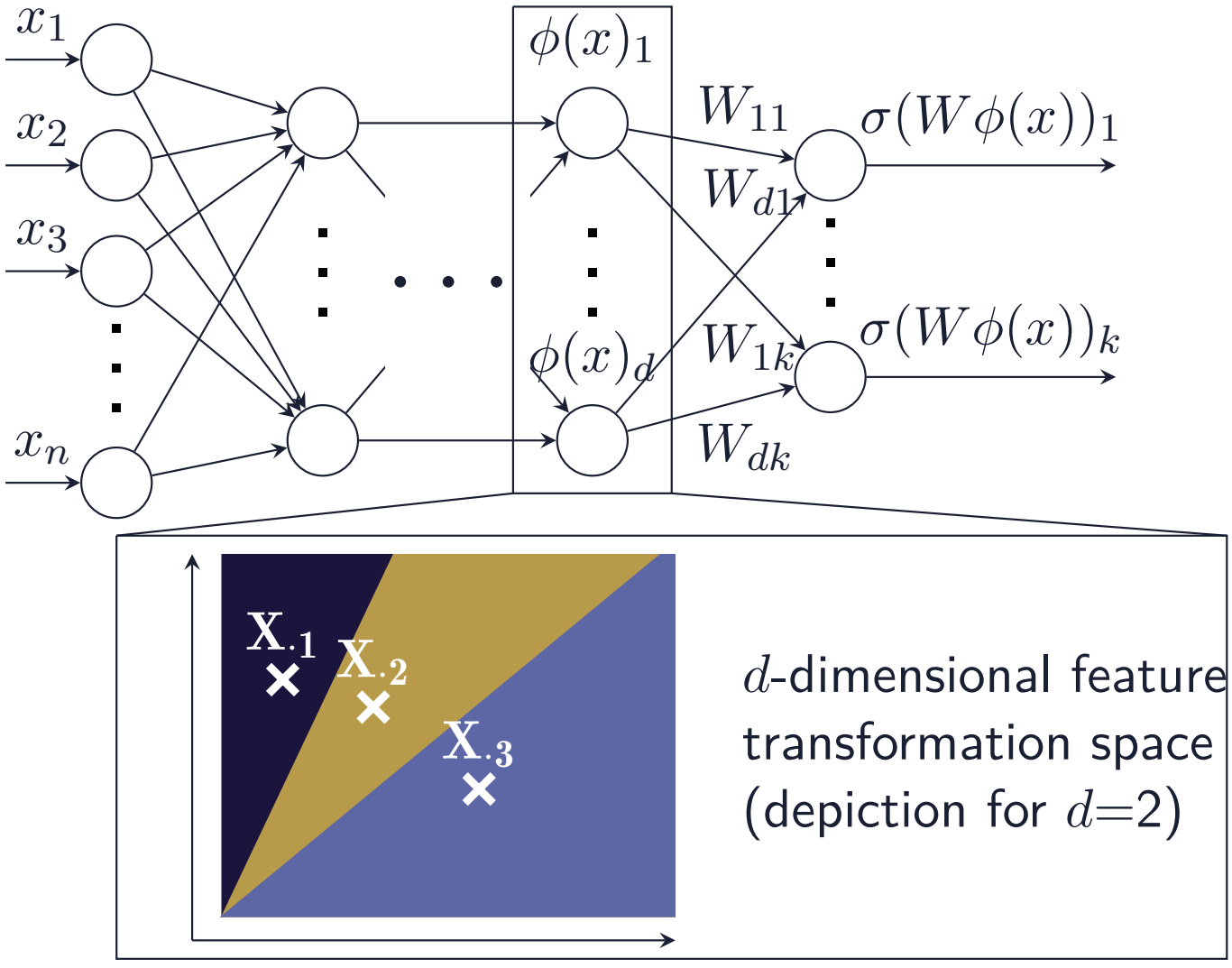
MF TUTORIAL PART 5

WHERE ELSE CAN WE GO FROM HERE?

Sibylle Hess and Michiel Hochstenbach

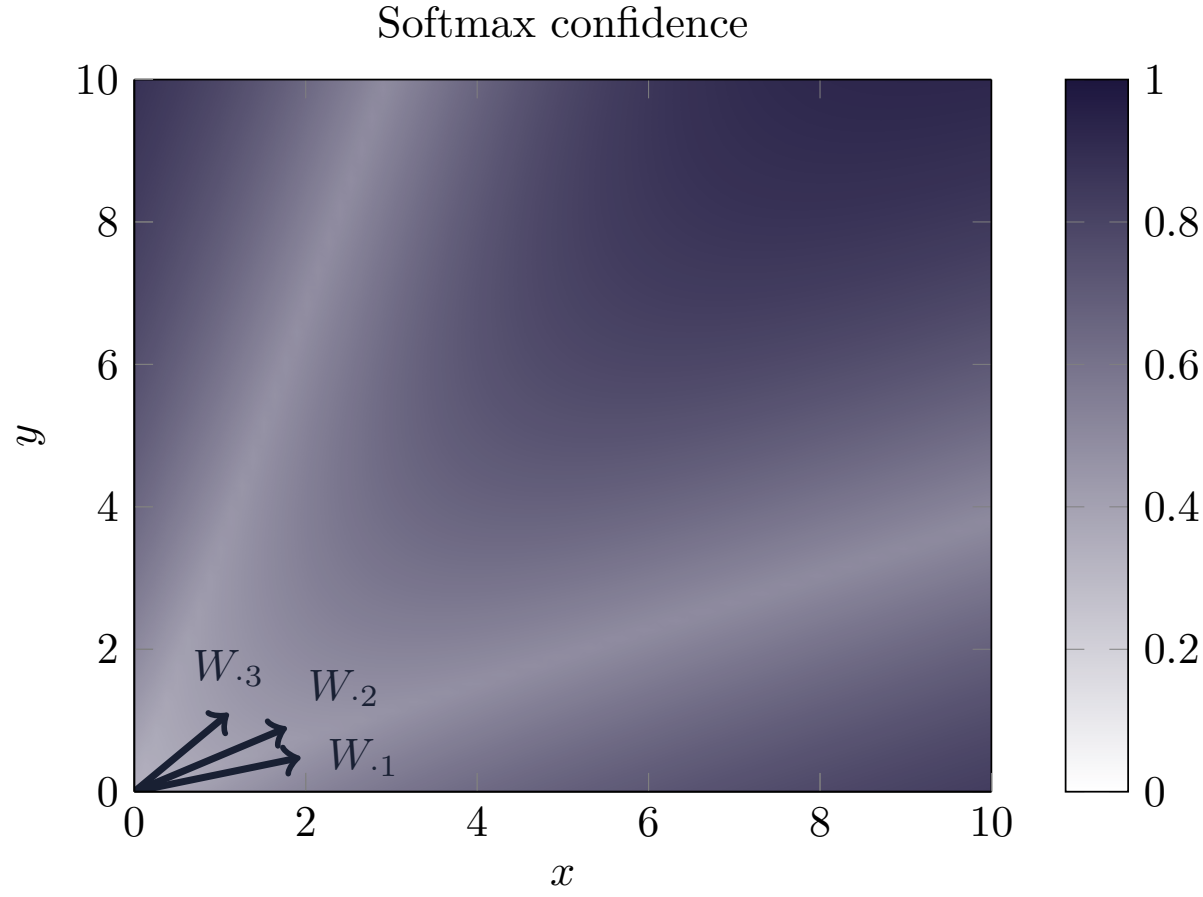
A CLUSTERING PERSPECTIVE ON DEEP LEARNING

DNNS ARE CENTROID-BASED CLASSIFIERS



Hess et al. 2020

THE PENULTIMATE LAYER SPACE IS SEPARATED INTO SOFTMAX-CONFIDENTLY ASSIGNED CLASSES

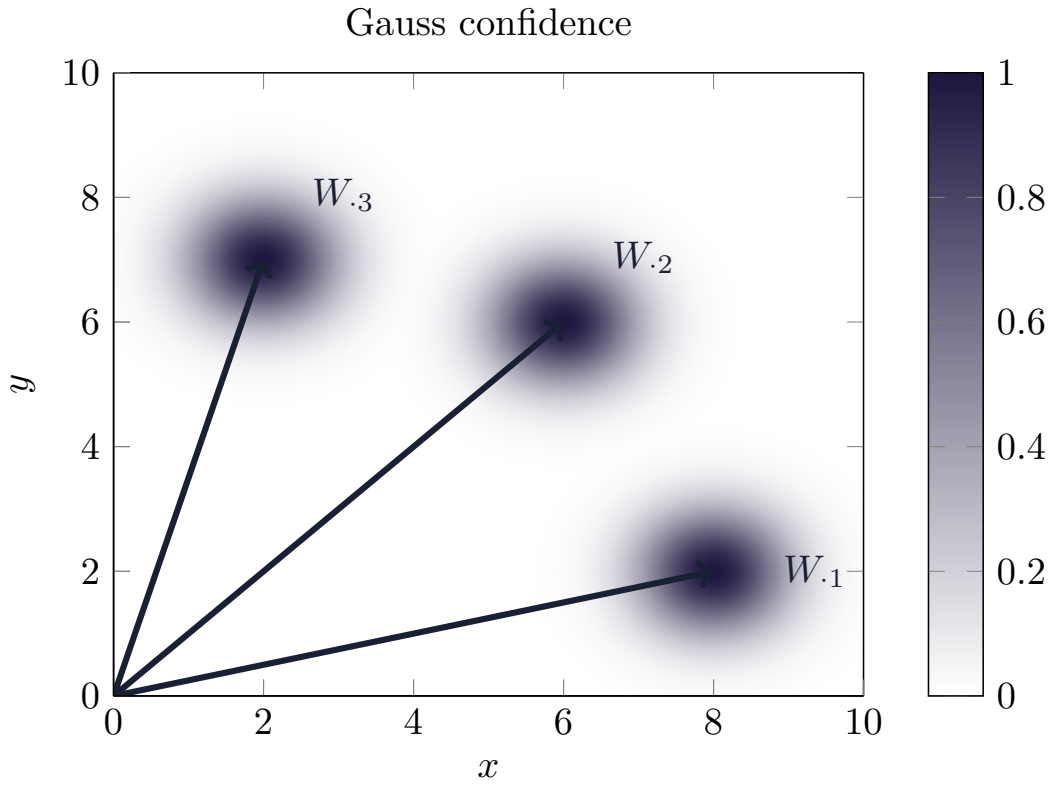


SOFTMAX CONFIDENCE IS OFTEN OVERCONFIDENT

True Label	COVID-19 (Training Data)		COVID-19 (Unseen Data)		Cat (Unrelated Data)	
						
Model	**Prediction**	**Confidence**	**Prediction**	**Confidence**	**Prediction**	**Confidence**
DNN	COVID-19	99.7%	Non-COVID	75.1%	COVID-19	100%
BNN	COVID-19	95.5%	COVID-19	67.1%	COVID-19	99.8%
Ours	COVID-19	99.9%	COVID-19	69.0%	COVID-19	50.1%

Softmax has no possibility to reflect what it **DOESN'T KNOW**, hence using another confidence would make sense.

INTRODUCING A CENTROID-BASED CONFIDENCE

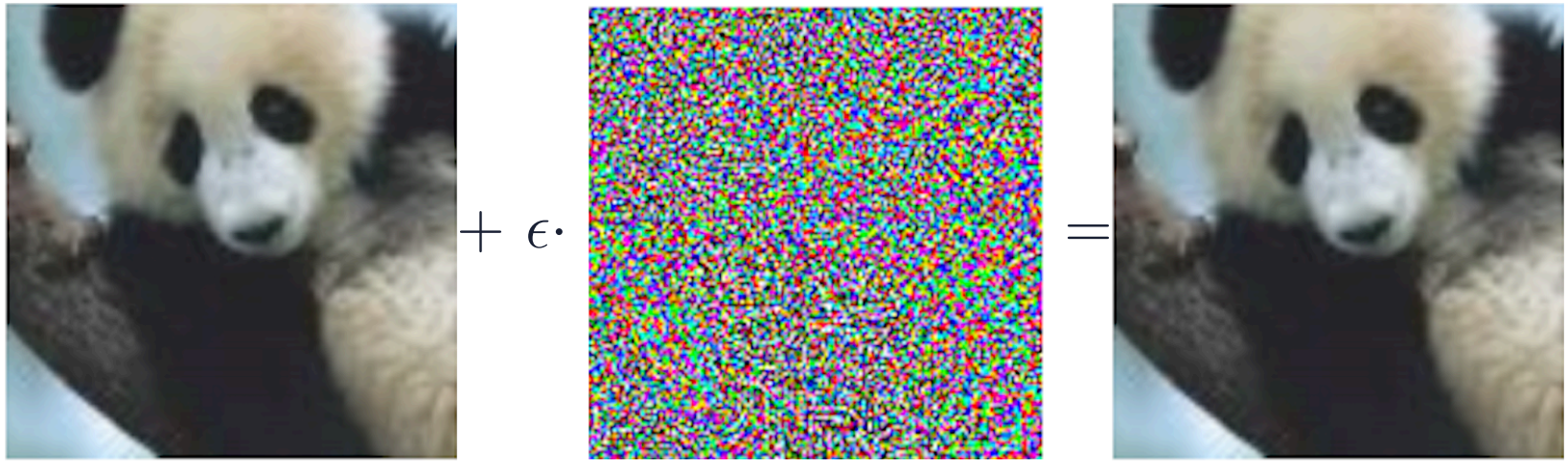


We introduce the **GAUSS-CONFIDENCE**:

$$\boldsymbol{\kappa}(x, W; \gamma)_k = \exp(-\gamma \|\phi(x) - W_{\cdot k}\|^2) \in (0, 1].$$

A **CENTROID-BASED CONFIDENCE** makes DNNs also more **ROBUST** to attacks.

DNNS ARE SENSITIVE TO ATTACKS



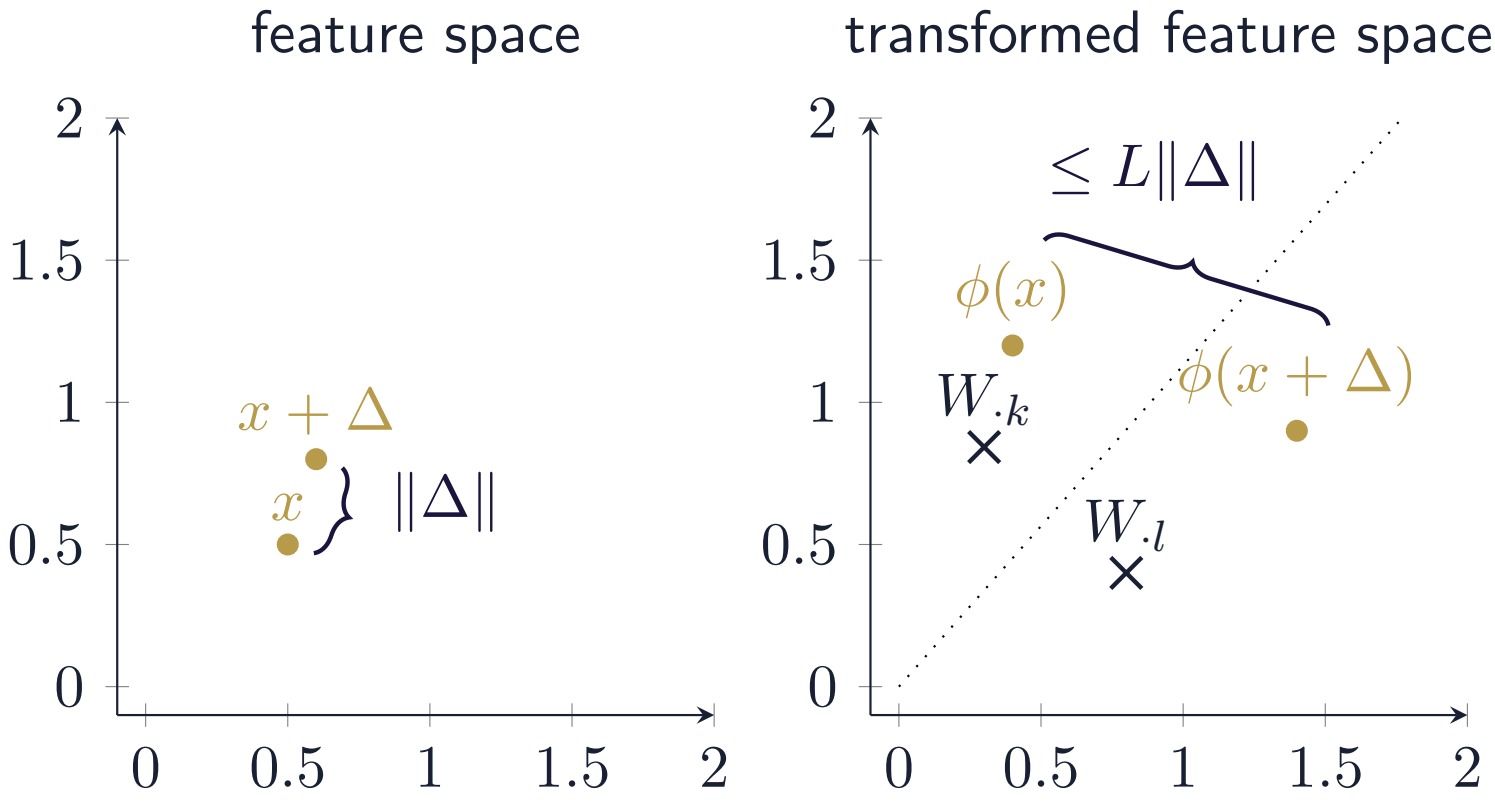
panda

57.7% confidence

gibbon

99.3% confidence

A CENTROID-BASED CONFIDENCE CAN REFLECT VULNERABILITY TO ATTACKS



DNN PREDICTIONS AS A K-MEANS CLUSTER ASSIGNMENT STEP

For DNN **CLASSIFIERS** there exist centroids $X \in \mathbb{R}^{d \times k}$ such that the predicted classes for m data points are computed as

$$\hat{Y} = \arg\min \|\phi(D) - YX^\top\|^2 \qquad \text{s.t. } Y \in \mathbb{1}^{m \times k}$$

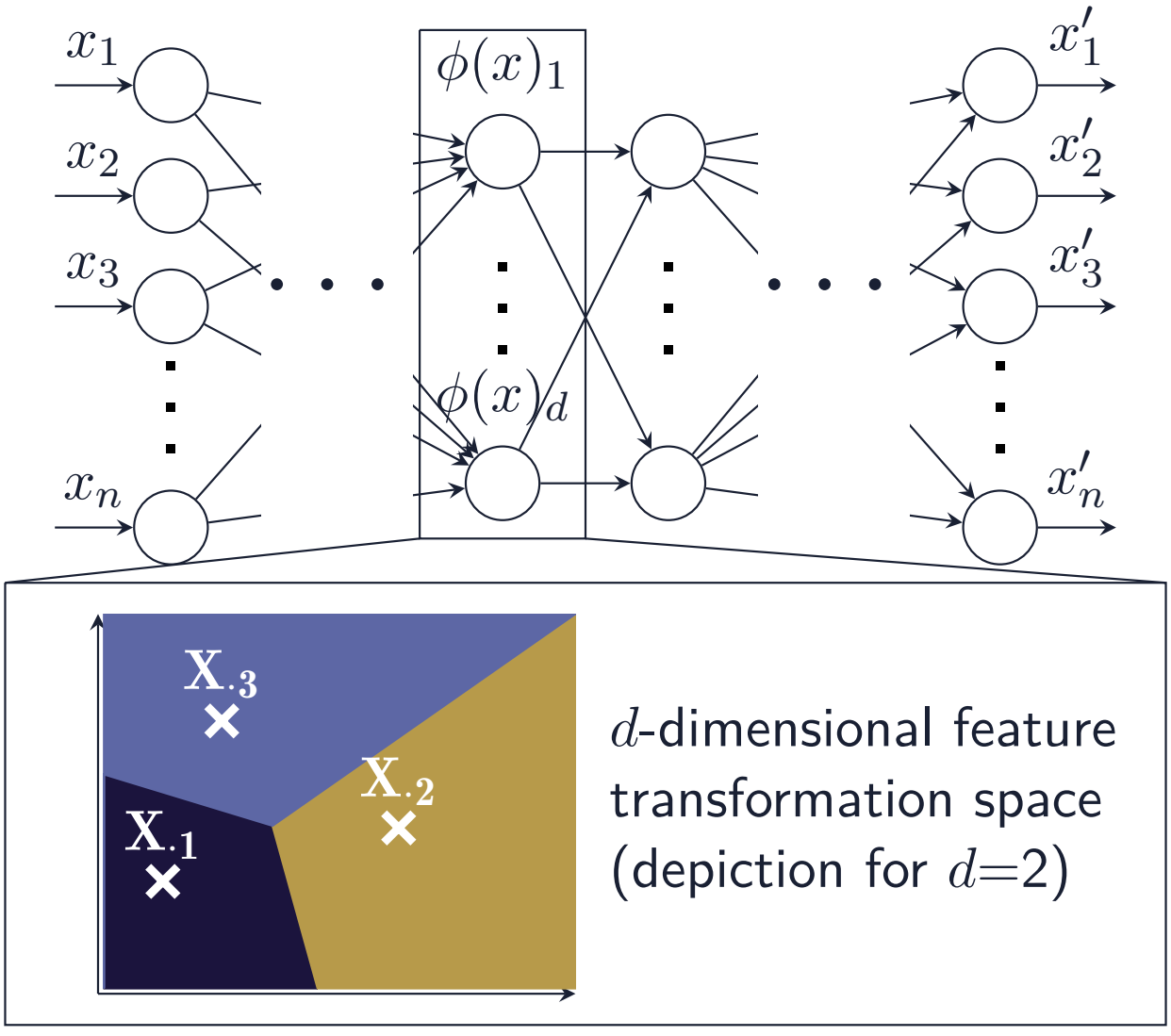
Can we use the success of deep learning to compute good kernel k-means **CLUSTERS**?

Nope, because the objective

$$\min_{\phi, X, Y} \|\phi(D) - YX^\top\|^2 \qquad \text{s.t. } Y \in \mathbb{1}^{m \times k}, X \in \mathbb{R}^{d \times k}$$

has a trivial solution: $\phi(D) = \mathbf{0}$ and $X = \mathbf{0}$

DEEP CLUSTERING



d-dimensional feature transformation space (depiction for d=2)

Trivial solutions are prevented by using an autoencoder, requiring that the embedded space contains semantic information.

Other binary learning tasks

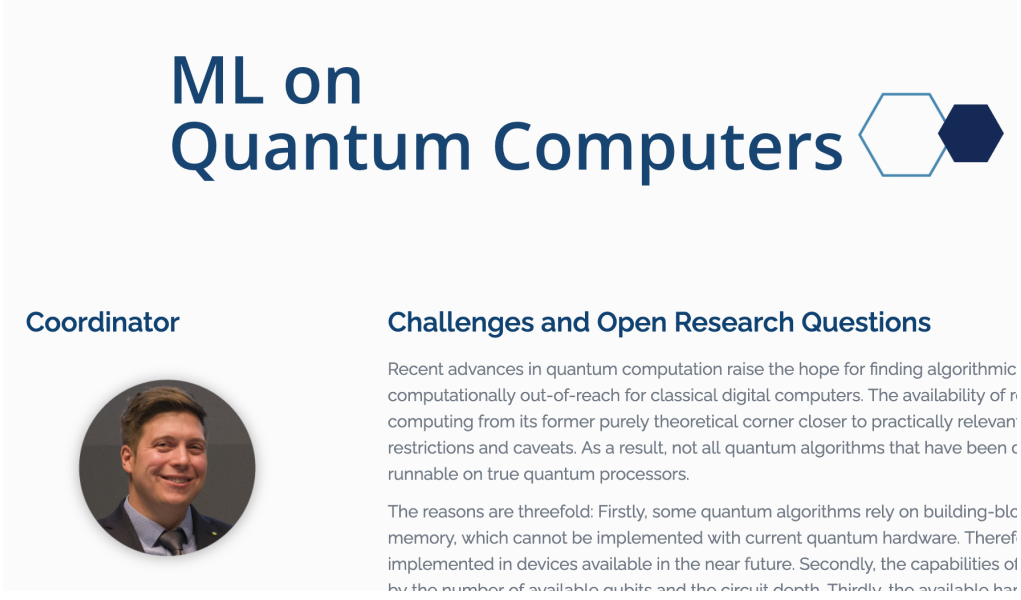
QUANTUM COMPUTING

Adabiatic quantum computing is able to solve **QUADRATIC UNCONSTRAINED BINARY OPTIMIZATION (QUBO)** objectives:

$$\min_y \; y^\top Q y + y^\top q \qquad \text{s.t. } y \in \{-1, 1\}^m$$

Quantum computing opens up entirely novel optimization possibilities. Many clustering objectives have been transformed to QUBOs.

Link to website



SEMANTIC HASHING

Hash similar points $D_{j\cdot}$ to similar codes $Y_{j\cdot}$.

↓

010110

↓

010111

↓

010110

$$\min_{Y} \operatorname{tr}(Y^\top L Y) \quad \text{s.t. } Y^\top \mathbf{1} = \mathbf{0},\ \frac{1}{m} Y^\top Y = I,\ Y \in \{-1, 1\}^{m \times k}$$

The optimization methods use known tricks from clustering.

Interesting is in hashing the problem to derive codes for **NEWLY ARRIVING DATA POINTS** $\rightarrow$ autoencoders and eigenfunctions

Weiss et al. 2008, Wang et al. 2017, Ying et al. 2020

Conclusions

DISCUSSION

- Objectives and principles of constrained matrix factorization are found in most data mining/machine learning tasks
- One of the biggest challenge today is to learn representations which match the learning task $\rightarrow$ deep learning
- Finding good optimization methods to handle binary constraints is relevant for many learning tasks outside of clustering
- Many of the MF principles transfer to tensor factorization, which is for example applied in knowledge graph completion.

Lacroix et al. 2018, Dettmers et al. 2018

SOME REFERENCES PART V

Hess et al. 2020	DNNs as centroid-based classifiers
Cavazza et al. 2018	Dropout as a Low-Rank Regularizer for Matrix Factorization
Kadu and Leeuwen 2018	Binary Tomography
Yun et al. 2020	Proximal SGD for deep learning
Buschjäger et al. 2021	Binary DNNs for resource aware applications