



# **Advanced Topics in Machine Learning**

**Winter Semester 2024/2025**

---

Prof. Dr.-Ing. Christian Bergler | OTH Amberg-Weiden

## Overview

---

### Topics From Last Time: Introduction & Deep Learning Recap

- Signals and Signal Types
- Audio Signals in Deep Learning
- Analog/Digital (A/D) Conversion (Sampling, Quantization)
- Discrete Fourier Transform (DFT)
- Short-Time Fourier Transform (STFT)
- Spectrogram
- Multimodal Learning

### Topics of Today: Advanced Deep Learning Strategies – Part I

- Representation Learning
- Transfer Learning
- Distillation Learning
- Deep Metric Learning (Focus: Contrastive Learning)

### Traditional Pattern Recognition System (PRS) or Machine Learning Pipeline (MLP)

#### Data Acquisition

Recording, collection, ...

#### Data Preprocessing

Transformation, Cleaning,  
reduction, normalization, ...

#### Feature Extraction

Extracting representative  
Features

#### Classification

Categorization across a given set  
of classes  $\Omega = \{\omega_1, \dots, \omega_m\}$

$$x \in \mathbb{R}^n$$

$$\hat{x} \in \mathbb{R}^m \quad (m << n)$$

$$c \in \mathbb{R}^z \quad (z << m)$$

$$\omega_i$$

#### Testing/Classification Phase

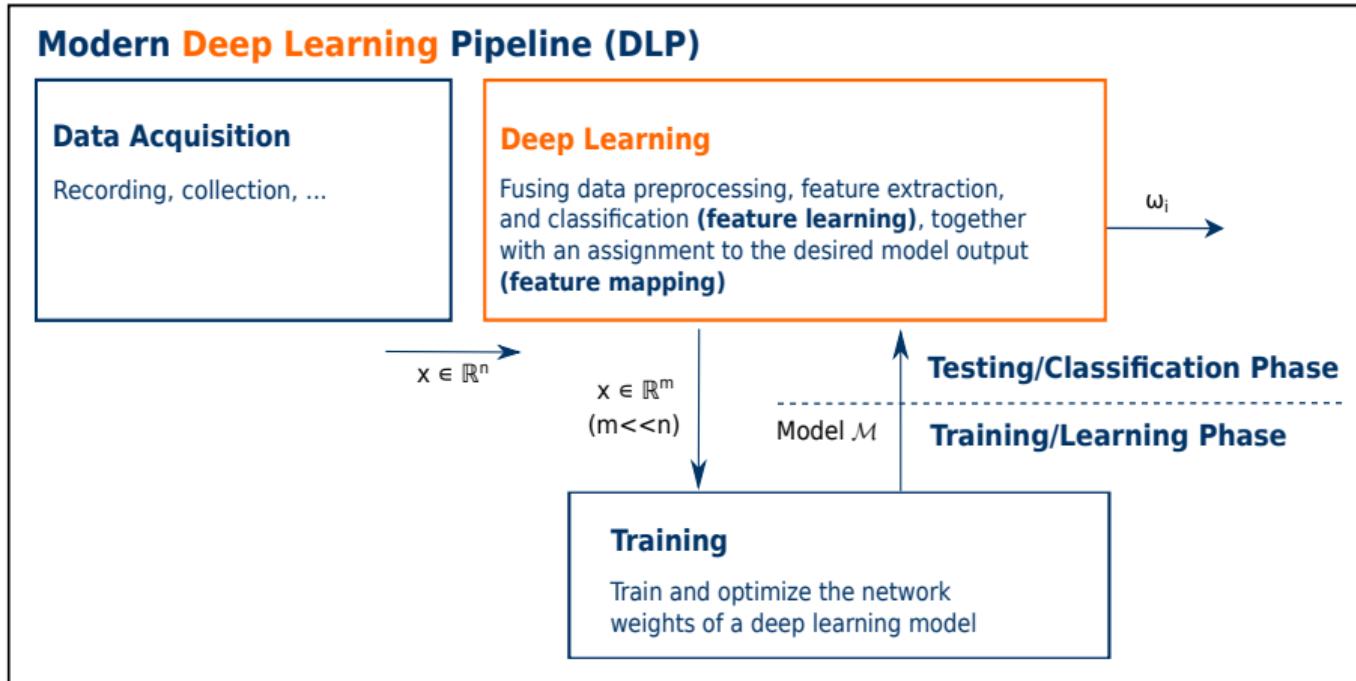
Model  $M$

#### Training/Learning Phase

#### Training

Train and optimize a given  
classification algorithm

Source: Image from Christian Bergler, Dissertation "Deep Learning Applied To Animal Linguistics", 2023



Source: Image from Christian Bergler, Dissertation "Deep Learning Applied To Animal Linguistics", 2023

## Feature Extraction & Feature Selection VS. Feature/Representation Learning

---

### Feature Extraction/Engineering

- Building a new feature subspace  $c \in \mathbb{R}^n$ , derived from information of the original features  $f \in \mathbb{R}^m$  (usually  $n \ll m$ ) to create enhanced representations for model interpretation and improvement
  - Feature transformation and (often) dimensionality reduction

## Feature Extraction & Feature Selection VS. Feature/Representation Learning

### Feature Extraction/Engineering

- Building a new feature subspace  $c \in \mathbb{R}^n$ , derived from information of the original features  $f \in \mathbb{R}^m$  (usually  $n \ll m$ ) to create enhanced representations for model interpretation and improvement
  - Feature transformation and (often) dimensionality reduction

### Feature Selection

- Picking a feature subset  $f \in \mathbb{R}^n$  from the original features  $f \in \mathbb{R}^m$  ( $n < m$ ) to reduce model complexity and efficiency
  - Dimensionality reduction while original features are maintained

# Terminology

## Feature Extraction & Feature Selection VS. Feature/Representation Learning

### Feature Extraction/Engineering

- Building a new feature subspace  $c \in \mathbb{R}^n$ , derived from information of the original features  $f \in \mathbb{R}^m$  (usually  $n \ll m$ ) to create enhanced representations for model interpretation and improvement
  - Feature transformation and (often) dimensionality reduction

### Feature Selection

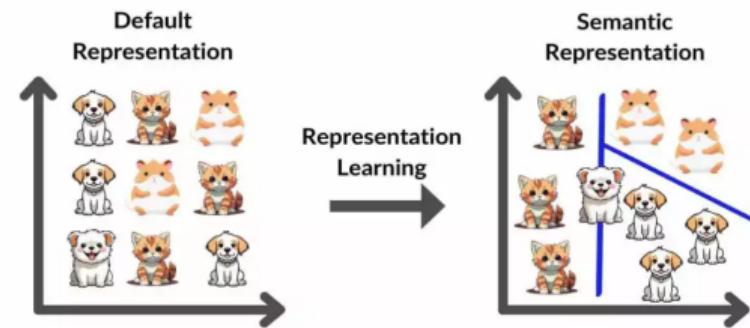
- Picking a feature subset  $f \in \mathbb{R}^n$  from the original features  $f \in \mathbb{R}^m$  ( $n < m$ ) to reduce model complexity and efficiency
  - Dimensionality reduction while original features are maintained

### Feature/Representation Learning

- Fully data-driven and automatic procedure to learn and discover the most important (unknown) underlying data characteristics (features) by transforming and reducing input

### Why we want to Learn Feature Representations?

- Discover the underlying patterns and features in raw data
- Learn informative, (often) compressed, and hierarchical representations that capture the essential characteristics

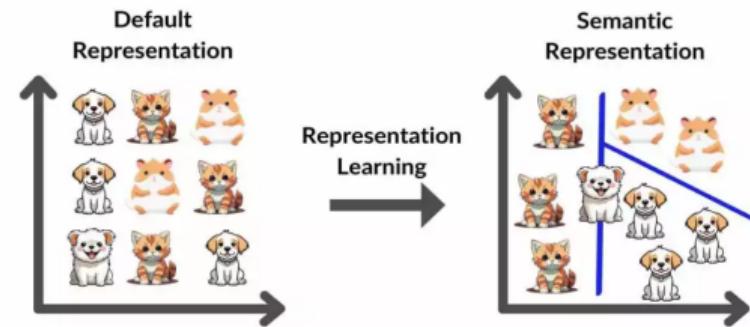


ŷ

Source: Image from <https://spotintelligence.com/2023/12/11/representation-learning/>

### Why we want to Learn Feature Representations?

- Discover the underlying patterns and features in raw data
- Learn informative, (often) compressed, and hierarchical representations that capture the essential characteristics
- Better performance on downstream tasks like classification, regression, and clustering

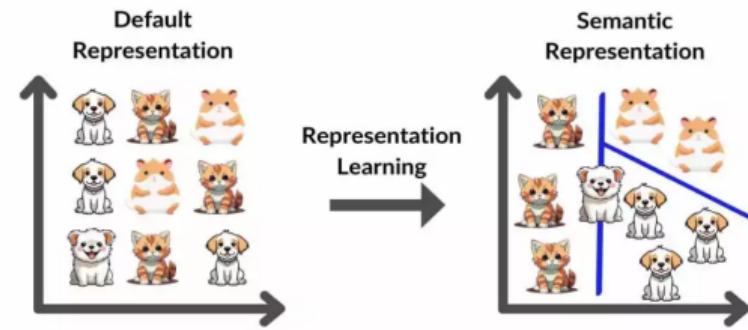


ÿ

Source: Image from <https://spotintelligence.com/2023/12/11/representation-learning/>

### Why we want to Learn Feature Representations?

- Discover the underlying patterns and features in raw data
- Learn informative, (often) compressed, and hierarchical representations that capture the essential characteristics
- Better performance on downstream tasks like classification, regression, and clustering
- Representation learning combines feature extraction/selection, dimensionality reduction and results in a better overall generalization

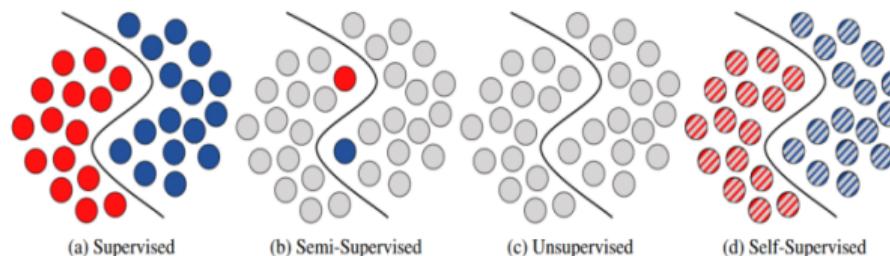


→ How about Training data? ↴

Source: Image from <https://spotintelligence.com/2023/12/11/representation-learning/>

### Different Ways of Representation Learning

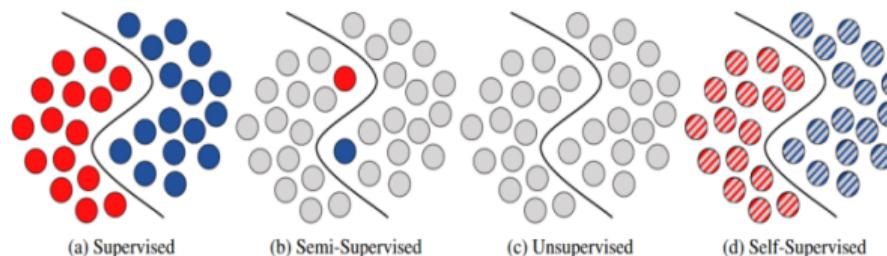
- **Supervised:** representations optimized for a specific task using labeled data (e.g. CNNs)



Source: Image from <https://arxiv.org/pdf/2002.08721>

### Different Ways of Representation Learning

- **Supervised:** representations optimized for a specific task using labeled data (e.g. CNNs)
- **Unsupervised:** representations without labeled data, relying on the inherent structure in the data to capture meaningful patterns (e.g. AE)

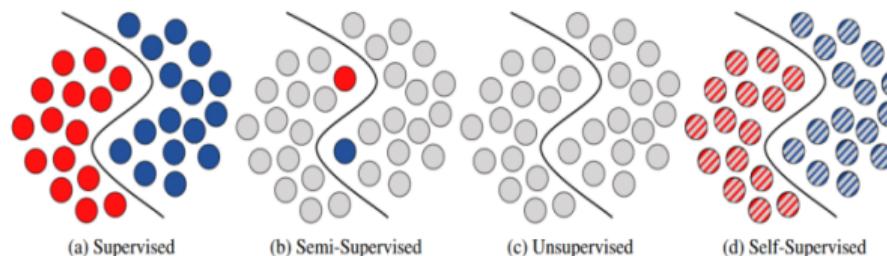


Source: Image from <https://arxiv.org/pdf/2002.08721>

## Representation Learning

### Different Ways of Representation Learning

- **Supervised:** representations optimized for a specific task using labeled data (e.g. CNNs)
- **Unsupervised:** representations without labeled data, relying on the inherent structure in the data to capture meaningful patterns (e.g. AE)
- **Self-Supervised:** type of unsupervised learning using its own (“pseudo”) labels from the unlabeled data corpus to predict missing parts & build different views of a single object

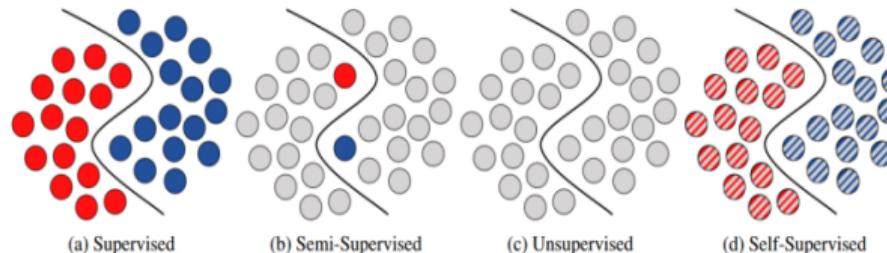


Source: Image from <https://arxiv.org/pdf/2002.08721>

## Representation Learning

### Different Ways of Representation Learning

- **Supervised:** representations optimized for a specific task using labeled data (e.g. CNNs)
- **Unsupervised:** representations without labeled data, relying on the inherent structure in the data to capture meaningful patterns (e.g. AE)
- **Self-Supervised:** type of unsupervised learning using its own (“pseudo”) labels from the unlabeled data corpus to predict missing parts & build different views of a single object
- **Semi-Supervised:** small amount of labeled data with a larger pool of unlabeled data to learn representations (self-supervised/unsupervised pre-training with supervised fine-tuning)

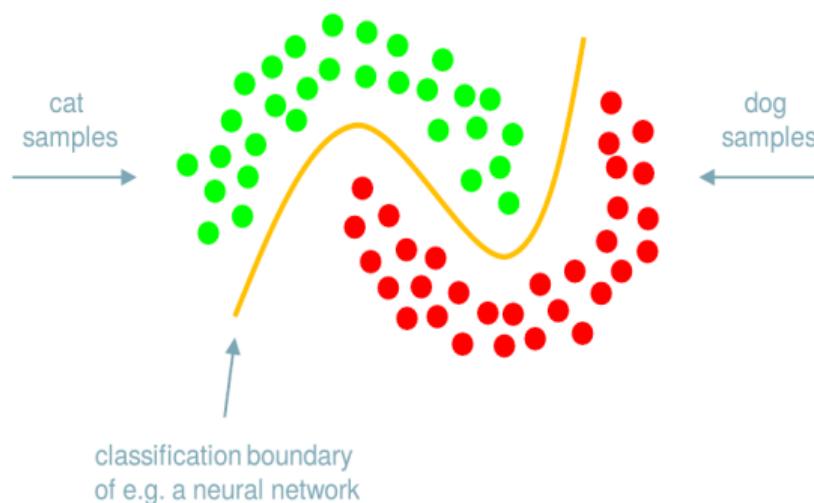


Source: Image from <https://arxiv.org/pdf/2002.08721>

# Deep Learning Strategies – Part I

## Representation Learning

Purely Supervised!



Source: Image from OTH-AW, Electrical Engineering, Media and Computer Science, Thomas Nierhoff – Vorlesung Advanced Topics in ML

# Deep Learning Strategies – Part I

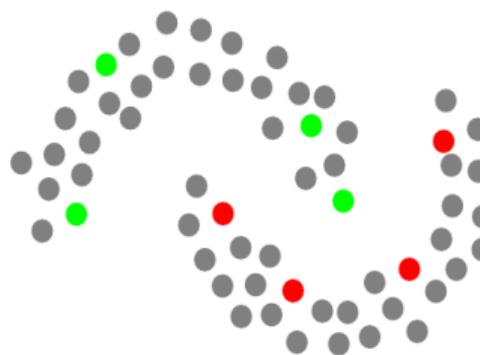
## Representation Learning

Reality!

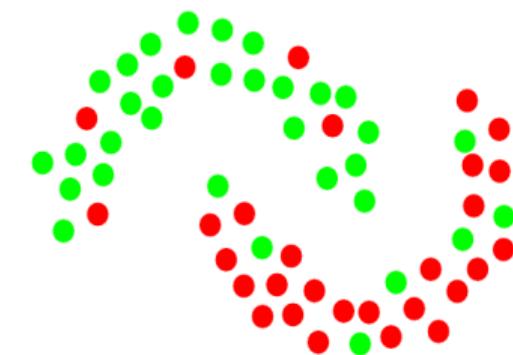
unlabeled data



partially labeled data



partially wrongly labeled data

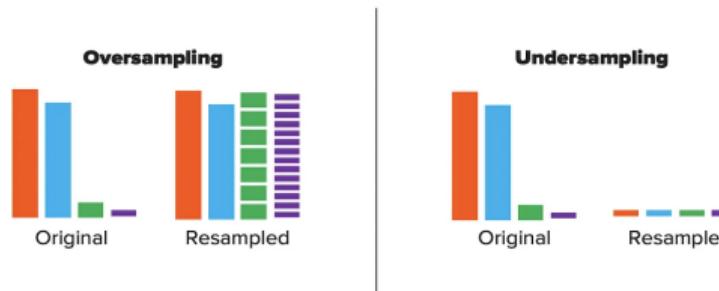


- Unlabeled data (unsupervised, self-supervised), partial labeled (semi-supervised), labeled, but partially wrong (weakly supervised)

Source: Image from OTH-AW, Electrical Engineering, Media and Computer Science, Thomas Nierhoff – Vorlesung Advanced Topics in ML

### Careful! Data Imbalance?

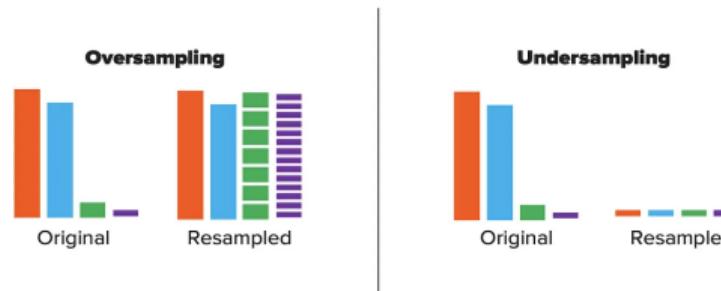
- **Data Imbalance:** classes/categories in a dataset have significantly fewer examples (minority class) than others (majority class) resulting in a biased model



Source: Image from <https://towardsdatascience.com/4-ways-to-improve-class-imbalance-for-image-data-9adec8f390f1>

### Careful! Data Imbalance?

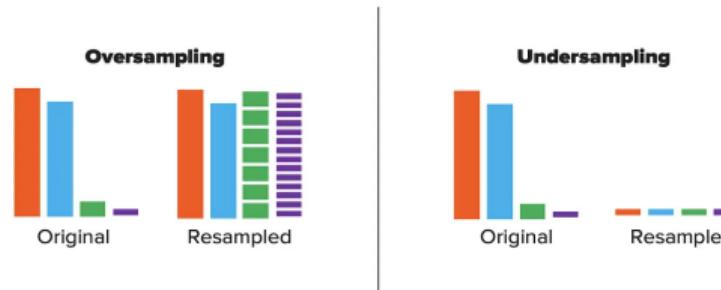
- **Data Imbalance:** classes/categories in a dataset have significantly fewer examples (minority class) than others (majority class) resulting in a biased model
- **Countermeasures:** collect/label more data of the minority class, reduce number of samples in majority class (undersampling!), increase number of samples in minority class (oversampling!)



Source: Image from <https://towardsdatascience.com/4-ways-to-improve-class-imbalance-for-image-data-9adec8f390f1>

### Careful! Data Imbalance?

- **Data Imbalance:** classes/categories in a dataset have significantly fewer examples (minority class) than others (majority class) resulting in a biased model
- **Countermeasures:** collect/label more data of the minority class, reduce number of samples in majority class (undersampling!), increase number of samples in minority class (oversampling!)
- **Techniques:** Undersampling – random deletion, condensed nearest-neighbor (NN), edited NN, tomek links; Oversampling – random copy, SMOTE (Synthetic Minority Oversampling Technique)

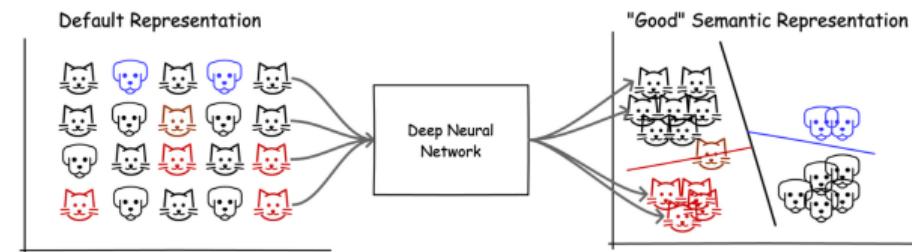
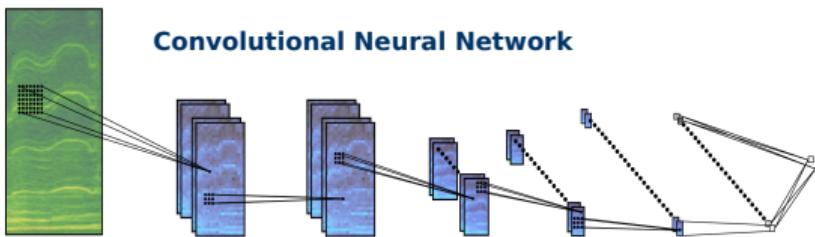


Source: Image from <https://towardsdatascience.com/4-ways-to-improve-class-imbalance-for-image-data-9adec8f390f1>

### Convolutional Neural Network

Supervised representation learning using a CNN, which consists of the following fundamental types of layers (others are of course possible):

- Convolution, normalization, activation layers for feature extraction
- Pooling layers for downsampling
- Fully Connected Layers for classification (e.g. softmax activation at the output layer)



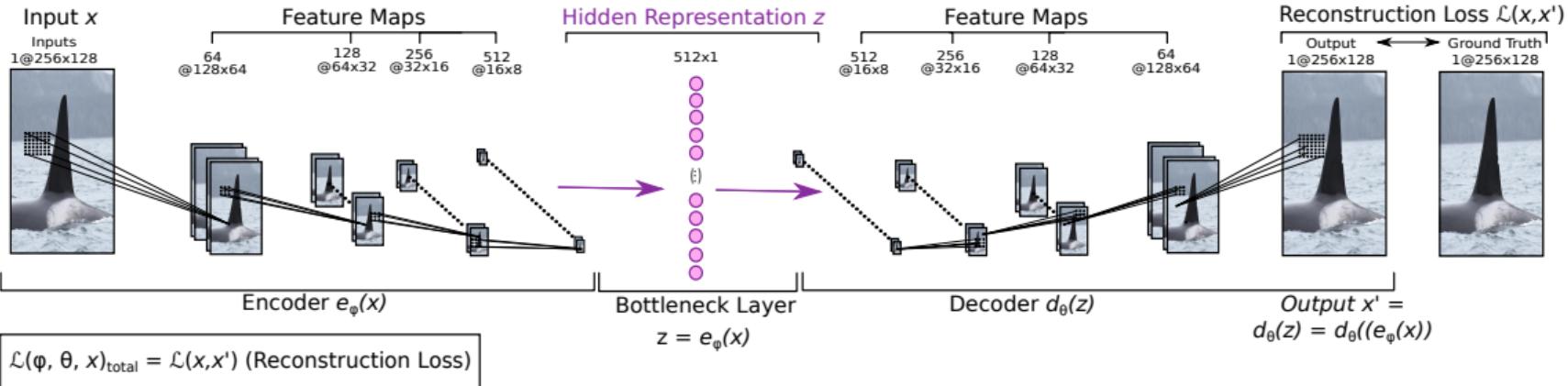
(Purely Data-Driven) Feature Learning and Classification (End-to-End)

Source: OTH-AW, Electrical Engineering, Media and Computer Science, Fabian Brunner – Vorlesung Convolutional Neural Networks

# Deep Learning Strategies – Part I

## Unsupervised Representation Learning

### Autoencoder – General Architecture and Concept

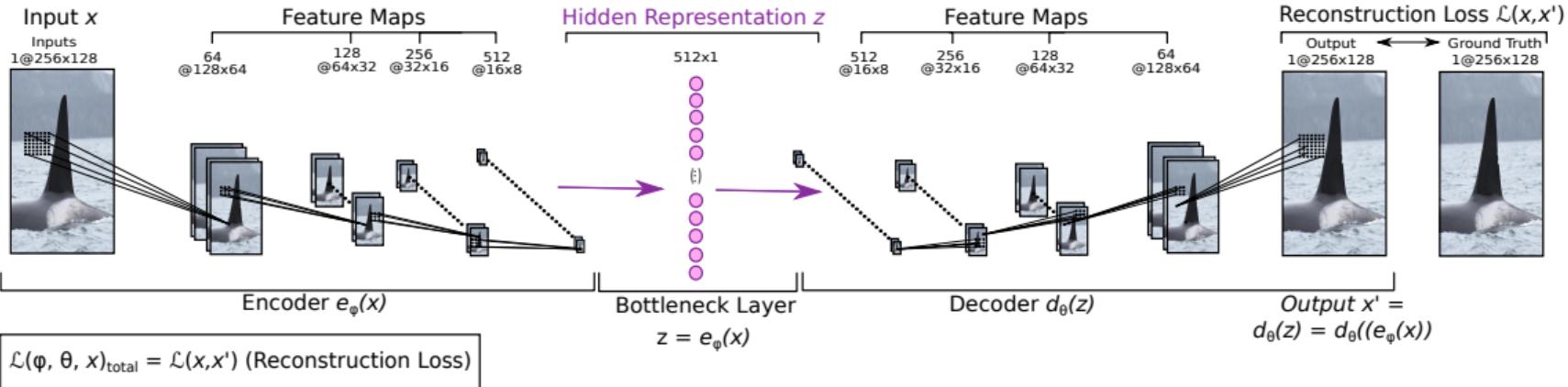


- Mapping a given input  $x$  to an output/reconstruction  $x'$  via a hidden representation  $z$  (deterministic!)

# Deep Learning Strategies – Part I

## Unsupervised Representation Learning

### Autoencoder – General Architecture and Concept

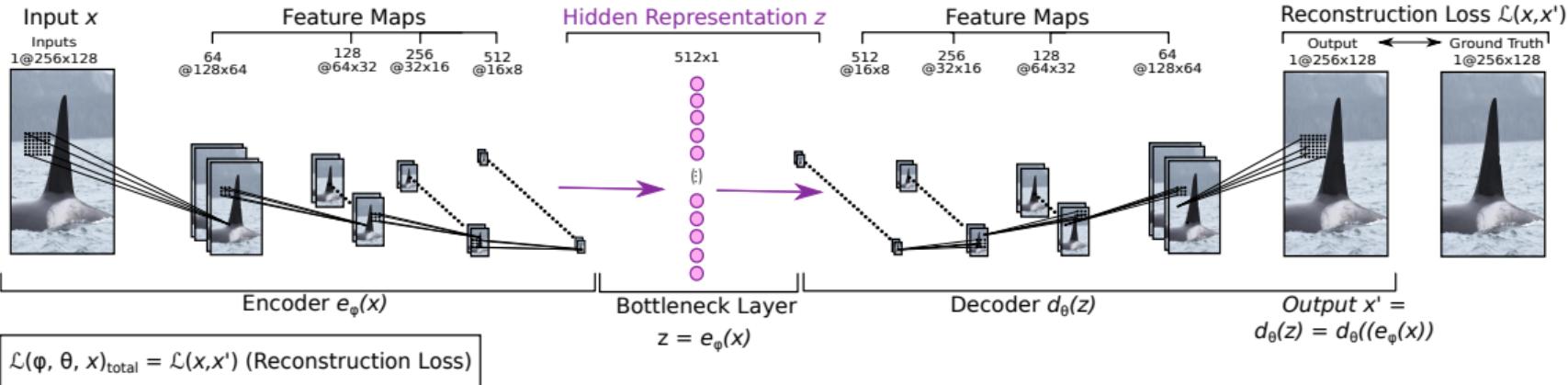


- Mapping a given input  $x$  to an output/reconstruction  $x'$  via a hidden representation  $z$  (deterministic!)
- Encoder  $e_\varphi$ , acting as a function to map (encode) the input  $x$  to the hidden representation  $z$  via  $e_\varphi(x)$

# Deep Learning Strategies – Part I

## Unsupervised Representation Learning

### Autoencoder – General Architecture and Concept

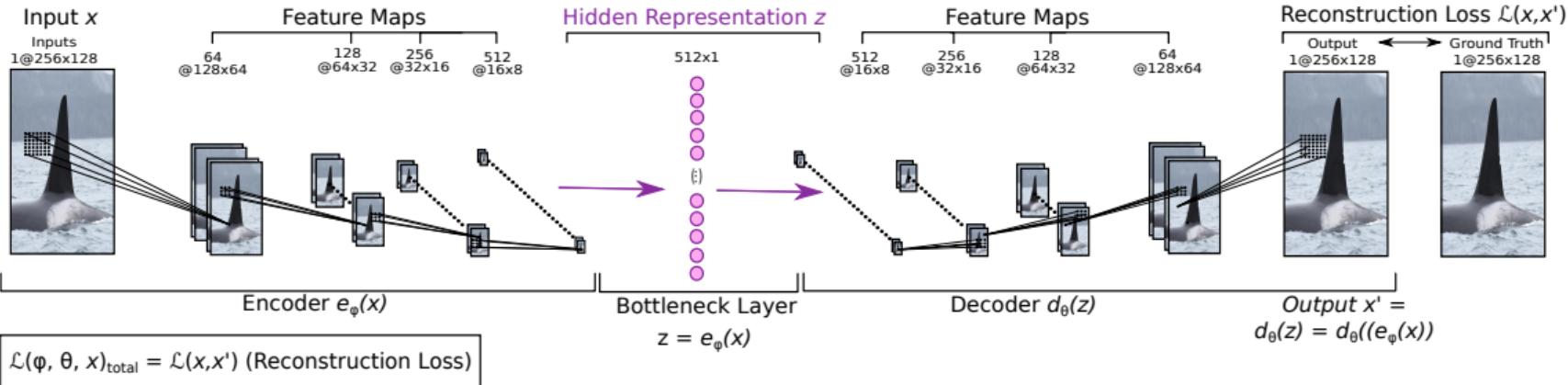


- Mapping a given input  $x$  to an output/reconstruction  $x'$  via a hidden representation  $z$  (deterministic!)
- Encoder  $e_\varphi$ , acting as a function to map (encode) the input  $x$  to the hidden representation  $z$  via  $e_\varphi(x)$
- Decoder  $d_\theta$ , acting as a function to map (decode) the hidden latent code  $z$  to the

# Deep Learning Strategies – Part I

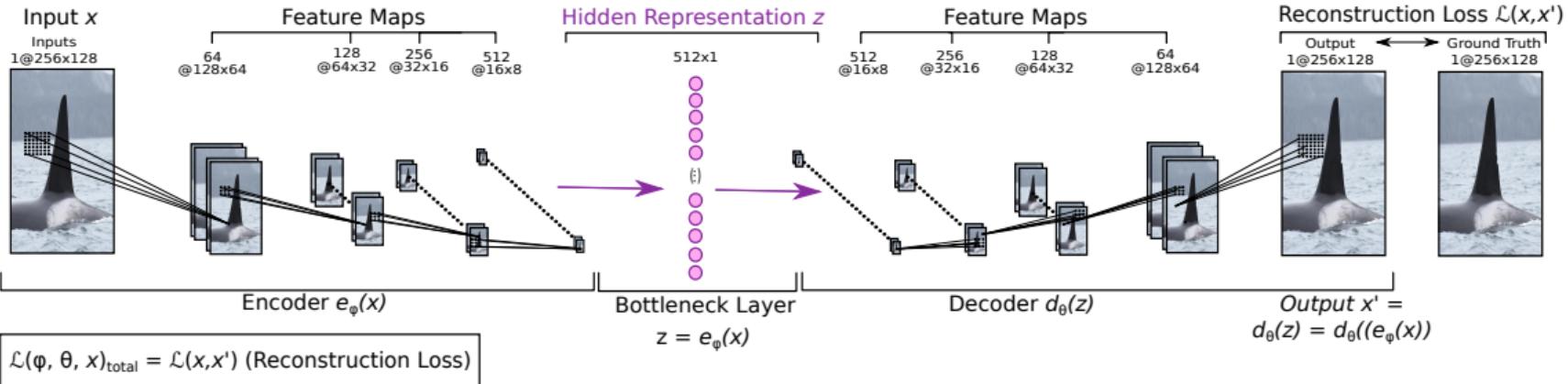
## Unsupervised Representation Learning

### Autoencoder – General Architecture and Concept



- **General Idea:** Learn an embedding  $z$  (latent code) including the most prominent features by minimizing the reconstruction loss  $L(x, d_\theta(e_\varphi(x)))$ , penalizing the dissimilarity between  $x$  and  $d_\theta(e_\varphi(x))$  (e.g. squared L<sub>2</sub>-norm)

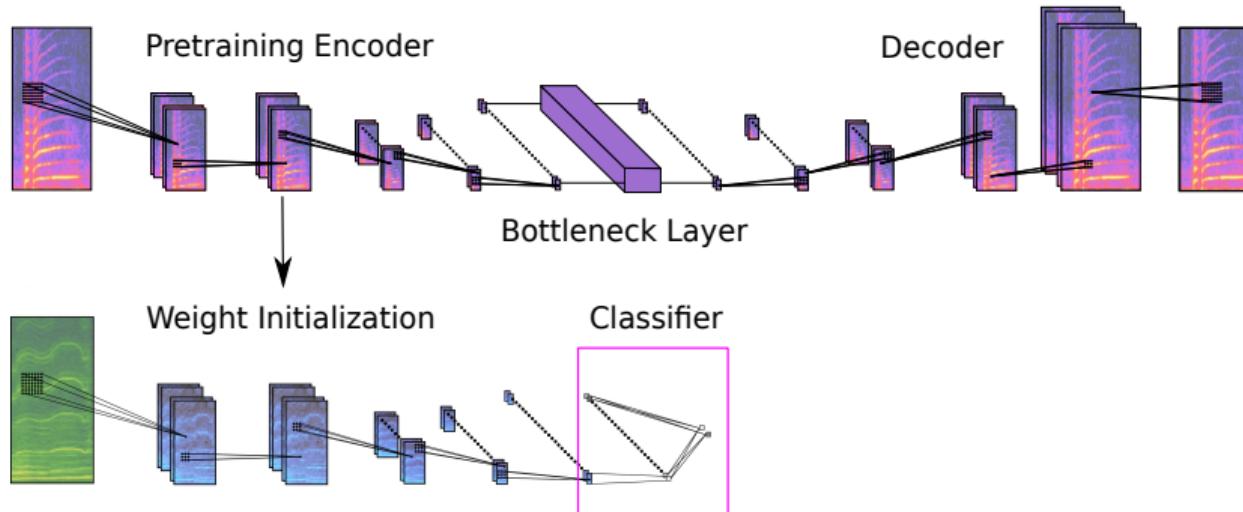
### Autoencoder – General Architecture and Concept



- **General Idea:** Learn an embedding  $z$  (latent code) including the most prominent features by minimizing the reconstruction loss  $L(x, d_\theta(e_\varphi(x)))$ , penalizing the dissimilarity between  $x$  and  $d_\theta(e_\varphi(x))$  (e.g. squared L<sub>2</sub>-norm)
- Data compression as well as data transformation (feature extraction/engineering) is done automatically

## Semi-Supervised Representation Learning – Handling Unlabeled Data?

### Combination of Supervised & Unsupervised Representation Learning (CNN + AE)



- Unlabelled data is used for pretraining (representation learning, transfer learning), whereas the learned weights are used as initialization and are further optimized using the (smaller) labelled data in a downstream supervised approach (fine-tuning)

Source: Images from Christian Bergler, Dissertation "Deep Learning Applied To Animal Linguistics", 2023

## Transfer Learning

---

- **Focus:** transferring knowledge from a given source domain  $D_s$  and task  $T_s$ , to a different target domain  $D_t$  and task  $T_t$
- Inspired by the ability of humans to transfer expertise across domains

Source: <https://www.linkedin.com/pulse/traditional-training-vs-transfer-learning-based-enjoyalgorithms>

## Transfer Learning

- **Focus:** transferring knowledge from a given source domain  $D_s$  and task  $T_s$ , to a different target domain  $D_t$  and task  $T_t$
- Inspired by the ability of humans to transfer expertise across domains
- Source domain:  $D_s = \{X, P(X)\}$  with  $X = \{x_1, \dots, x_n\}$  as the sample-specific feature vectors in the entire dataset and  $P(X)$  as the marginal distribution

Source: <https://www.linkedin.com/pulse/traditional-training-vs-transfer-learning-based-enjoyalgorithms>

## Transfer Learning

- **Focus:** transferring knowledge from a given source domain  $D_s$  and task  $T_s$ , to a different target domain  $D_t$  and task  $T_t$
- Inspired by the ability of humans to transfer expertise across domains
- Source domain:  $D_s = \{X, P(X)\}$  with  $X = \{x_1, \dots, x_n\}$  as the sample-specific feature vectors in the entire dataset and  $P(X)$  as the marginal distribution
- Source task:  $T_s = \{Y, f(\cdot)\} = \{Y, P(Y|X)\}$  with label space  $Y = \{y_1, \dots, y_k\}$  together with a learnable decision function (Deep Learning Model)  $f(\cdot)$

Source: <https://www.linkedin.com/pulse/traditional-training-vs-transfer-learning-based-enjoyalgorithms>

## Transfer Learning

- **Focus:** transferring knowledge from a given source domain  $D_s$  and task  $T_s$ , to a different target domain  $D_t$  and task  $T_t$
- Inspired by the ability of humans to transfer expertise across domains
- Source domain:  $D_s = \{X, P(X)\}$  with  $X = \{x_1, \dots, x_n\}$  as the sample-specific feature vectors in the entire dataset and  $P(X)$  as the marginal distribution
- Source task:  $T_s = \{Y, f(\cdot)\} = \{Y, P(Y|X)\}$  with label space  $Y = \{y_1, \dots, y_k\}$  together with a learnable decision function (Deep Learning Model)  $f(\cdot)$
- $f(\cdot)$  trained on pairs of feature vectors  $x_j \in X$  & associated label  $y_i \in Y$ , with  $f(x_j) = y_i$

Source: <https://www.linkedin.com/pulse/traditional-training-vs-transfer-learning-based-enjoyalgorithms>

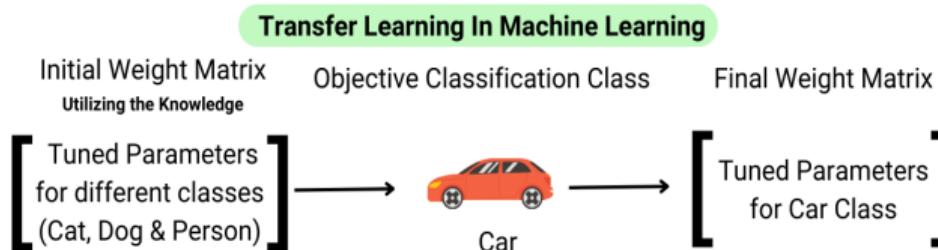
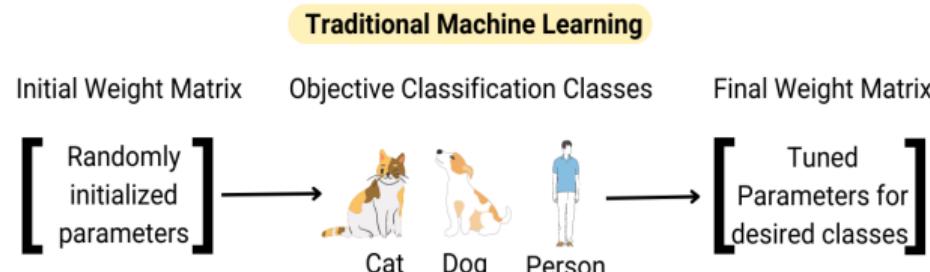
## Transfer Learning

- **Focus:** transferring knowledge from a given source domain  $D_s$  and task  $T_s$ , to a different target domain  $D_t$  and task  $T_t$
- Inspired by the ability of humans to transfer expertise across domains
- Source domain:  $D_s = \{X, P(X)\}$  with  $X = \{x_1, \dots, x_n\}$  as the sample-specific feature vectors in the entire dataset and  $P(X)$  as the marginal distribution
- Source task:  $T_s = \{Y, f(\cdot)\} = \{Y, P(Y|X)\}$  with label space  $Y = \{y_1, \dots, y_k\}$  together with a learnable decision function (Deep Learning Model)  $f(\cdot)$
- $f(\cdot)$  trained on pairs of feature vectors  $x_j \in X$  & associated label  $y_i \in Y$ , with  $f(x_j) = y_i$
- Transfer learning addresses a DL-strategy, aiming to enhance the model performance of  $f(\cdot)$  within  $D_t$  w.r.t. task  $T_t$ , by leveraging knowledge from  $f(\cdot)$  in  $D_s$  w.r.t. task  $T_s$ , while  $D_s \neq D_t$  or  $T_s \neq T_t$

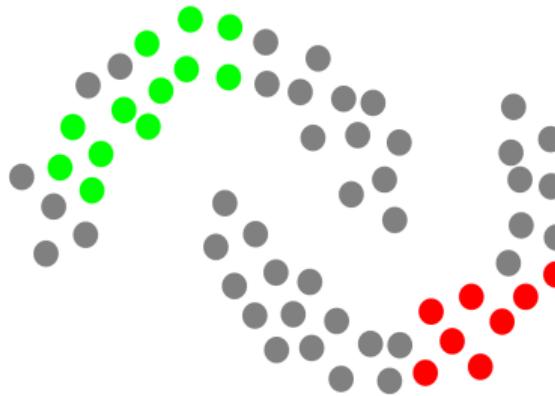
Source: <https://www.linkedin.com/pulse/traditional-training-vs-transfer-learning-based-enjoyalgorithms>

## Transfer Learning

- **Goal:** Transferring knowledge of large-scale (labeled/unlabeled) data in domain  $D_s$  and task  $T_s$ , to handle a similar and related task  $T_t$ , traditionally with a significantly lower amount of (labeled or unlabeled) data in domain  $D_t$



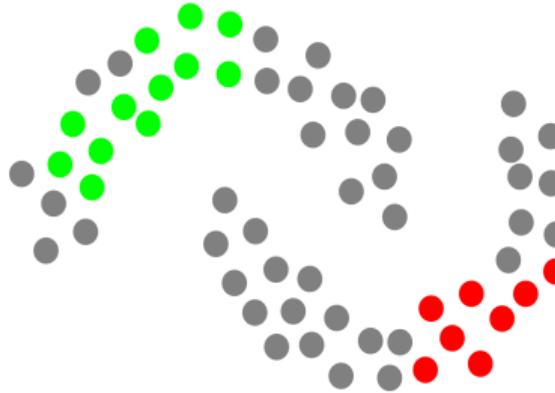
Source: Christian Bergler, Dissertation "Deep Learning Applied To Animal Linguistics", 2023



- **Smoothness Assumption:** If two samples are similar or close to each other, their labels should also be similar or close

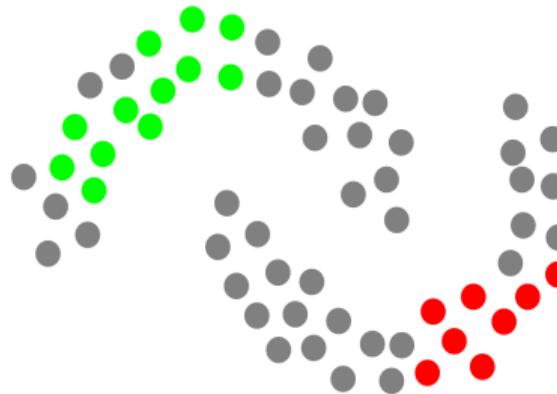
Source: Image from OTH-AW, Electrical Engineering, Media and Computer Science, Thomas Nierhoff – Vorlesung Advanced Topics in ML

## Semi-Supervised Representation Learning – Handling Unlabeled Data?



- **Smoothness Assumption:** If two samples are similar or close to each other, their labels should also be similar or close
- Use partially labeled data to train a classifier on both labeled and unlabeled data →  
**Idea:** propagating labels from labeled samples to unlabeled samples

Source: Image from OTH-AW, Electrical Engineering, Media and Computer Science, Thomas Nierhoff – Vorlesung Advanced Topics in ML



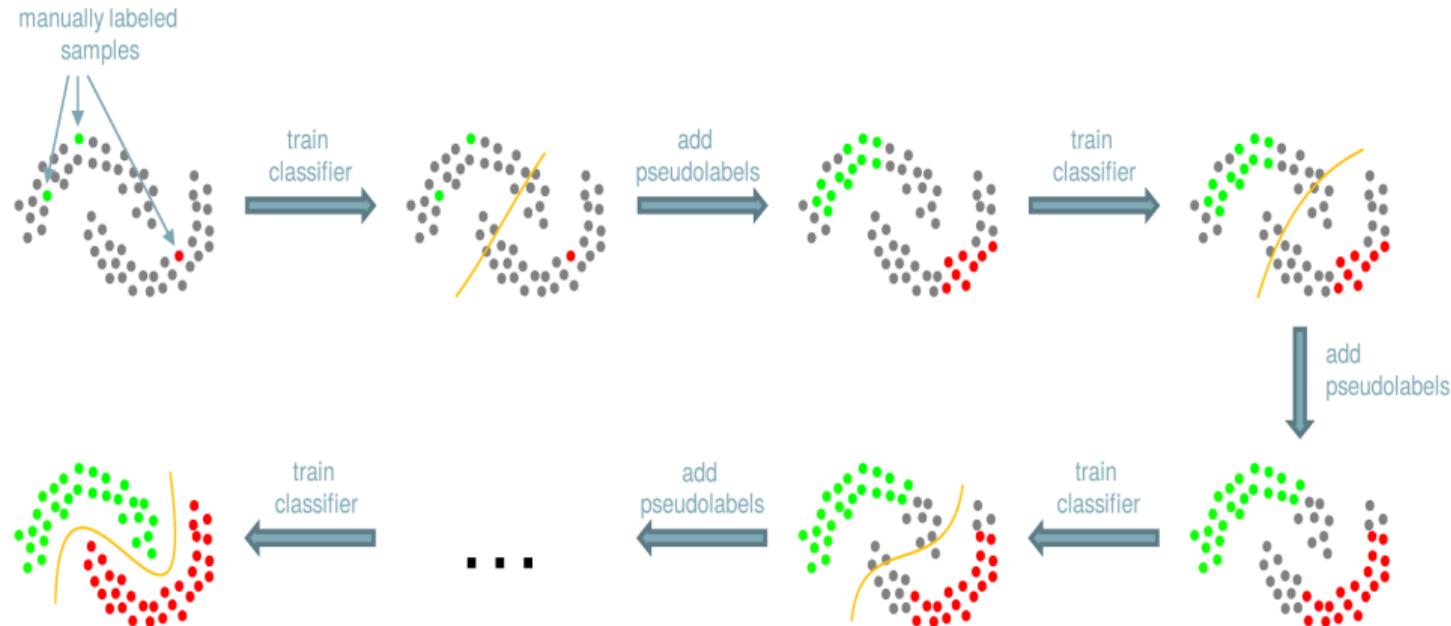
- **Smoothness Assumption:** If two samples are similar or close to each other, their labels should also be similar or close
- Use partially labeled data to train a classifier on both labeled and unlabeled data →  
**Idea:** propagating labels from labeled samples to unlabeled samples
- **Approaches:** Pseudo Labeling & Consistency Regularization

Source: Image from OTH-AW, Electrical Engineering, Media and Computer Science, Thomas Nierhoff – Vorlesung Advanced Topics in ML

# Deep Learning Strategies – Part I

## Semi-Supervised Representation Learning – Handling Unlabeled Data?

### Pseudo Labeling



- Iterative Procedure: a) train initial classifier with existing labels, b) predict unknowns, c) assign pseudo-labels to samples with high confidence, d) re-train, e) repeat b–c)

Source: Image from OTH-AW, Electrical Engineering, Media and Computer Science, Thomas Nierhoff – Vorlesung Advanced Topics in ML

## Semi-Supervised Representation Learning – Handling Unlabeled Data?

### Pseudo Labeling – Ways to Improve!

- Confidence Thresholding: only incorporate pseudo-labels for training in case the prediction is above a given threshold  $\delta$
- Soft labels (softmax, e.g. [0, 0.3, 0.7]) VS. hard labels (e.g. [0, 0, 1]) for pseudo labels



Source: Image from OTH-AW, Electrical Engineering, Media and Computer Science, Thomas Nierhoff – Vorlesung Advanced Topics in ML

## Semi-Supervised Representation Learning – Handling Unlabeled Data?

### Pseudo Labeling – Ways to Improve!

- Confidence Thresholding: only incorporate pseudo-labels for training in case the prediction is above a given threshold  $\delta$
- Soft labels (softmax, e.g. [0, 0.3, 0.7]) VS. hard labels (e.g. [0, 0, 1]) for pseudo labels



- Label Refinement: refine pseudo-labels over time by re-evaluating them after each training iteration or epoch

Source: Image from OTH-AW, Electrical Engineering, Media and Computer Science, Thomas Nierhoff – Vorlesung Advanced Topics in ML

## Semi-Supervised Representation Learning – Handling Unlabeled Data?

### Pseudo Labeling – Ways to Improve!

- Confidence Thresholding: only incorporate pseudo-labels for training in case the prediction is above a given threshold  $\delta$
- Soft labels (softmax, e.g. [0, 0.3, 0.7]) VS. hard labels (e.g. [0, 0, 1]) for pseudo labels



- Label Refinement: refine pseudo-labels over time by re-evaluating them after each training iteration or epoch
- Ensemble Learning: incorporate boosting or bagging

Source: Image from OTH-AW, Electrical Engineering, Media and Computer Science, Thomas Nierhoff – Vorlesung Advanced Topics in ML

## Semi-Supervised Representation Learning – Handling Unlabeled Data?

### Pseudo Labeling – Ways to Improve!

- Confidence Thresholding: only incorporate pseudo-labels for training in case the prediction is above a given threshold  $\delta$
- Soft labels (softmax, e.g. [0, 0.3, 0.7]) VS. hard labels (e.g. [0, 0, 1]) for pseudo labels



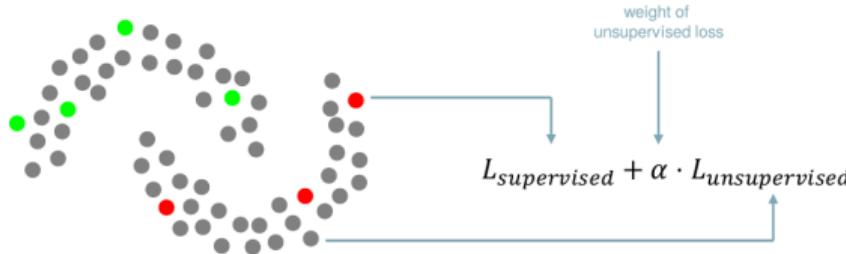
- Label Refinement: refine pseudo-labels over time by re-evaluating them after each training iteration or epoch
- Ensemble Learning: incorporate boosting or bagging
- Co-Training: optimizing the model on multiple “representations” (views) at each step (feature split, PCA, ...)



Source: Image from OTH-AW, Electrical Engineering, Media and Computer Science, Thomas Nierhoff – Vorlesung Advanced Topics in ML

## Semi-Supervised Representation Learning – Handling Unlabeled Data?

### Consistency Regularization

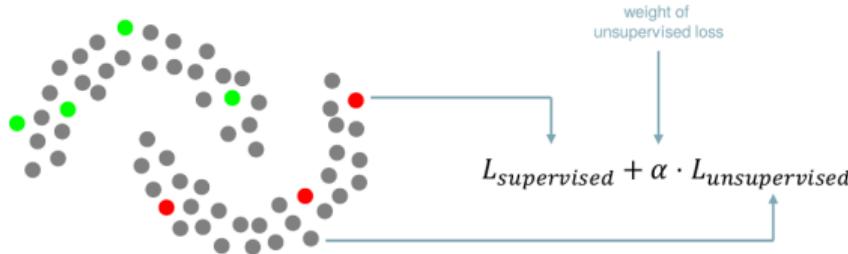


- **Idea:** use a bipartite loss as a combination between the supervised loss  $L_{sup}$  and unsupervised loss  $L_{unsup}$

Source: Image from OTH-AW, Electrical Engineering, Media and Computer Science, Thomas Nierhoff – Vorlesung Advanced Topics in ML

## Semi-Supervised Representation Learning – Handling Unlabeled Data?

### Consistency Regularization

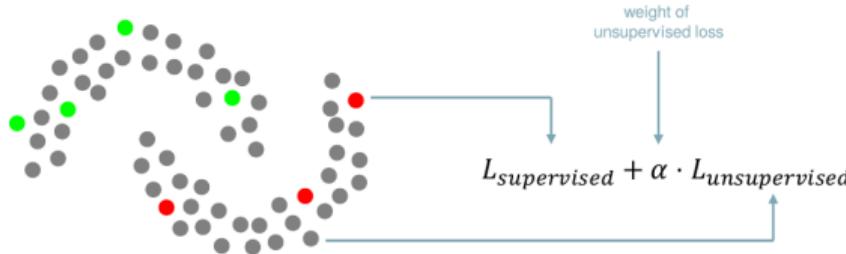


- **Idea:** use a bipartite loss as a combination between the supervised loss  $L_{\text{sup}}$  and unsupervised loss  $L_{\text{unsup}}$
- Recap “Smoothness”: unsupervised loss ensures that close samples, lead to close predictions, while the supervised loss ensures the predictions to be correct

Source: Image from OTH-AW, Electrical Engineering, Media and Computer Science, Thomas Nierhoff – Vorlesung Advanced Topics in ML

## Semi-Supervised Representation Learning – Handling Unlabeled Data?

### Consistency Regularization

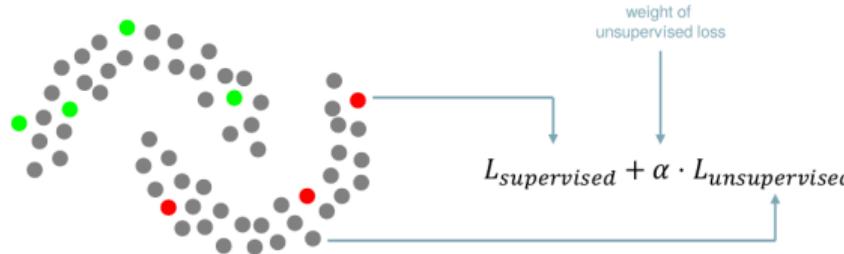


- **Idea:** use a bipartite loss as a combination between the supervised loss  $L_{sup}$  and unsupervised loss  $L_{unsup}$
- Recap “Smoothness”: unsupervised loss ensures that close samples, lead to close predictions, while the supervised loss ensures the predictions to be correct
- Consistency regularization is weighted by the  $\alpha$  parameter

Source: Image from OTH-AW, Electrical Engineering, Media and Computer Science, Thomas Nierhoff – Vorlesung Advanced Topics in ML

## Semi-Supervised Representation Learning – Handling Unlabeled Data?

### Consistency Regularization

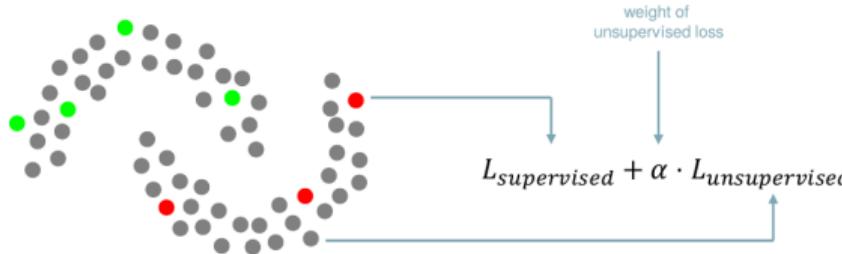


- **Idea:** use a bipartite loss as a combination between the supervised loss  $L_{sup}$  and unsupervised loss  $L_{unsup}$
- Recap “Smoothness”: unsupervised loss ensures that close samples, lead to close predictions, while the supervised loss ensures the predictions to be correct
- Consistency regularization is weighted by the  $\alpha$  parameter
- If  $\alpha$  is large, the model relies more on consistency for unlabeled data, and if small, the focus is more on the labeled data

Source: Image from OTH-AW, Electrical Engineering, Media and Computer Science, Thomas Nierhoff – Vorlesung Advanced Topics in ML

## Semi-Supervised Representation Learning – Handling Unlabeled Data?

### Consistency Regularization



- **Idea:** use a bipartite loss as a combination between the supervised loss  $L_{\text{sup}}$  and unsupervised loss  $L_{\text{unsup}}$
- Recap “Smoothness”: unsupervised loss ensures that close samples, lead to close predictions, while the supervised loss ensures the predictions to be correct
- Consistency regularization is weighted by the  $\alpha$  parameter
- If  $\alpha$  is large, the model relies more on consistency for unlabeled data, and if small, the focus is more on the labeled data
- Using the unlabeled data to reinforce the models learning

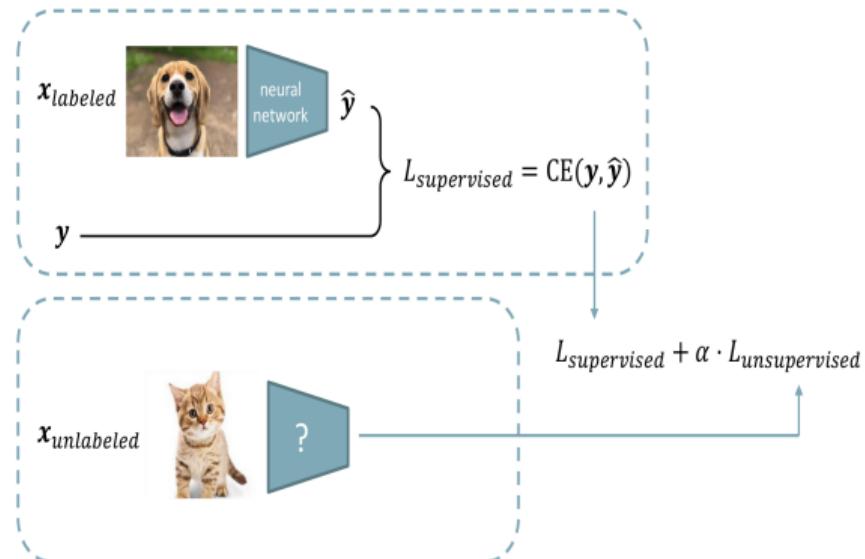
Source: Image from OTH-AW, Electrical Engineering, Media and Computer Science, Thomas Nierhoff – Vorlesung Advanced Topics in ML

## Semi-Supervised Representation Learning – Handling Unlabeled Data?

### Consistency Regularization – General Approach

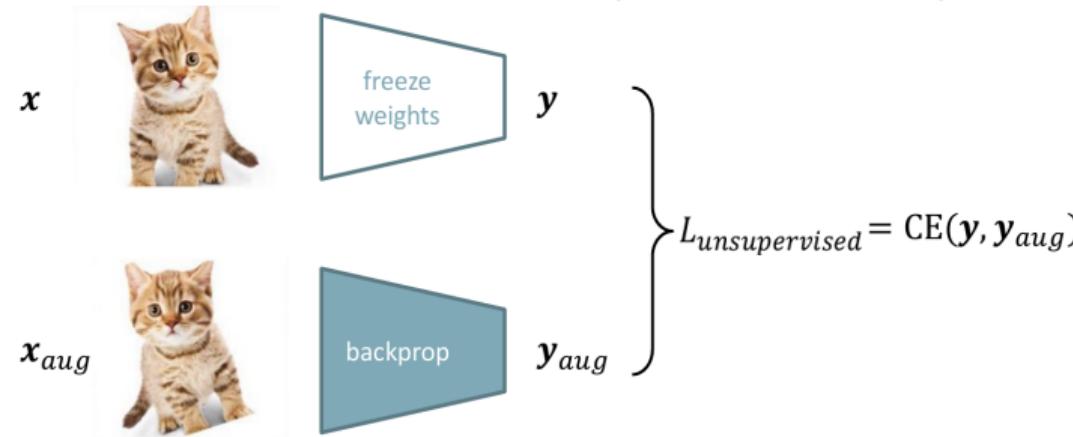
1. Input: take an unlabeled input sample  $x$
2. Augmentation/Perturbation: create a number of  $n$  versions of this sample  $\tilde{x}^1, \tilde{x}^2$  (rotation, scaling, noise)
3. Prediction:  $n$  predictions  $f(\tilde{x}^1)$  &  $f(\tilde{x}^2)$
4. Consistency Loss: difference between these two predictions (e.g. MSE-Loss), penalizing the model if its predictions for the two perturbed images are not similar

→ Minimizing consistency loss w.r.t. unlabeled data, thus being invariant to small changes?



Source: Image from OTH-AW, Electrical Engineering, Media and Computer Science, Thomas Nierhoff – Vorlesung Advanced Topics in ML

### Consistency Regularization – Data Augmentation (Rotation, Scaling)



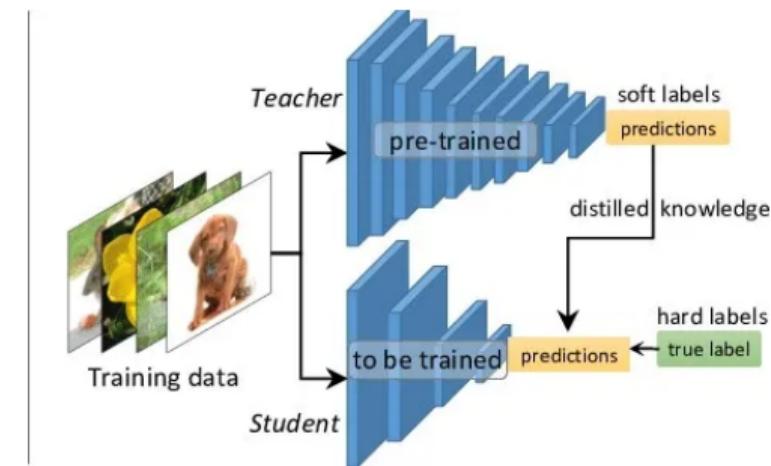
- Computing the unsupervised loss  $L_{unsup}$  by comparing the model prediction  $y = f(x)$  with the model prediction using the augmented input  $y_{aug} = f(x_{aug})$
- Similar samples  $x_{aug}$  are computed via various Augmentation techniques
- Different learning concepts following the “Student-Teacher” Learning Approach

Source: Image from OTH-AW, Electrical Engineering, Media and Computer Science, Thomas Nierhoff – Vorlesung Advanced Topics in ML

### Different Concepts (Student-Teacher)

#### 1. Frozen Teacher:

- ▶ Teacher weights are frozen during training
- ▶ Robust pre-trained teacher model required



Source: Image from <https://towardsdatascience.com/knowledge-distillation-simplified-dd4973dbc764>

# Deep Learning Strategies – Part I

## Distillation Learning

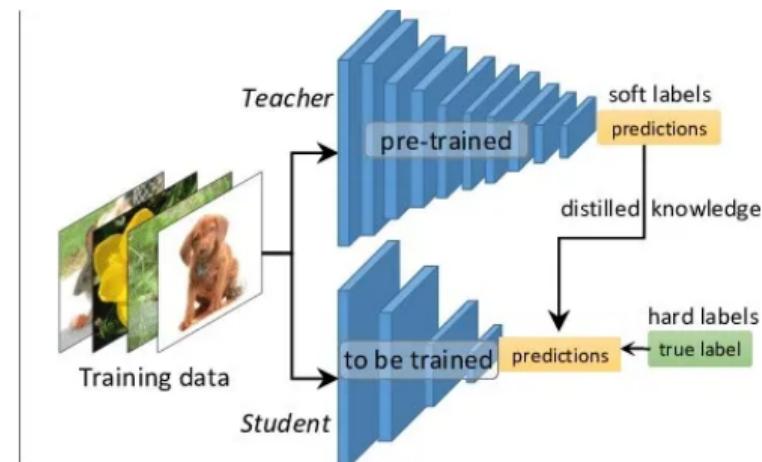
### Different Concepts (Student-Teacher)

#### 1. Frozen Teacher:

- ▶ Teacher weights are frozen during training
- ▶ Robust pre-trained teacher model required

#### 2. Exponential Moving Average (EMA) Teacher

- ▶ Teachers weights are updated as EMA of the students weights (smoothed version), allowing the teacher to gradually capture improvements



Source: Image from <https://towardsdatascience.com/knowledge-distillation-simplified-dd4973dbc764>

# Deep Learning Strategies – Part I

## Distillation Learning

### Different Concepts (Student-Teacher)

#### 1. Frozen Teacher:

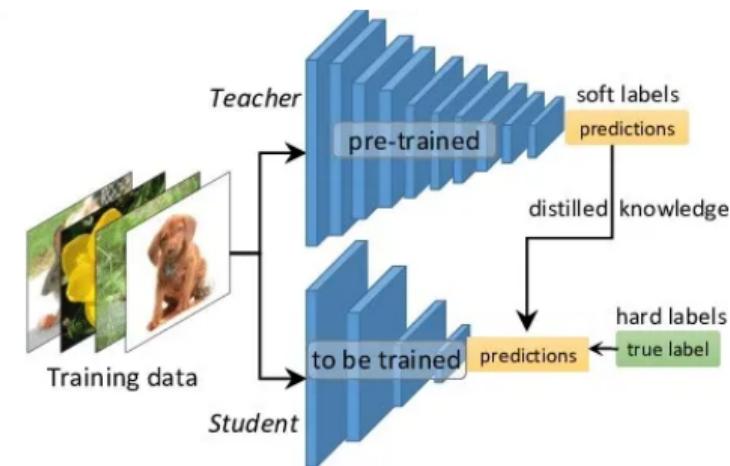
- ▶ Teacher weights are frozen during training
- ▶ Robust pre-trained teacher model required

#### 2. Exponential Moving Average (EMA) Teacher

- ▶ Teachers weights are updated as EMA of the students weights (smoothed version), allowing the teacher to gradually capture improvements

#### 3. Teacher Knowledge-Distillation

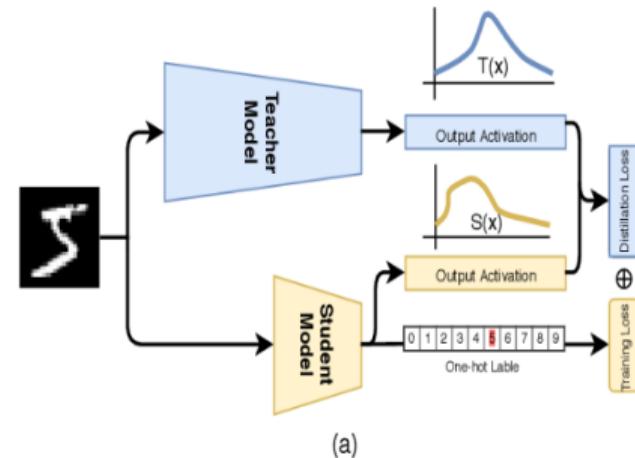
- ▶ Large pre-trained teacher model generates “soft” labels (using temperature scaling), which the (smaller) student uses as targets, combined with the hard labels (two objectives)



Source: Image from <https://towardsdatascience.com/knowledge-distillation-simplified-dd4973dbc764>

### 4. Self-Distillation

- Model serves as both teacher & student but at different stages of training (first supervised, later using own pseudo-labels) → layer distillation (knowledge from deeper layers to shallower), prediction distillation (re-use own predictions plus hard labels as targets)



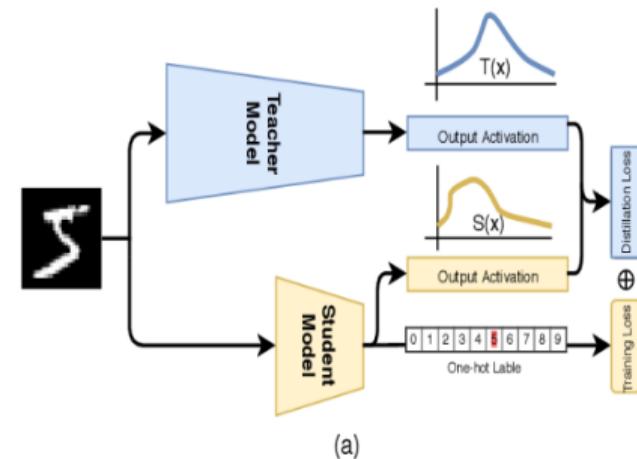
Source: Image from Zeyi Tao et al., Neuron Manifold Distillation for Edge Deep Learning

### 4. Self-Distillation

- Model serves as both teacher & student but at different stages of training (first supervised, later using own pseudo-labels) → layer distillation (knowledge from deeper layers to shallower), prediction distillation (re-use own predictions plus hard labels as targets)

### 5. Noisy Student Training

- Strong pretrained teacher model, while the student is trained with noisy inputs (strong augmentations, dropout, ...) learning semantic similarity



Source: Image from Zeyi Tao et al., Neuron Manifold Distillation for Edge Deep Learning

### 4. Self-Distillation

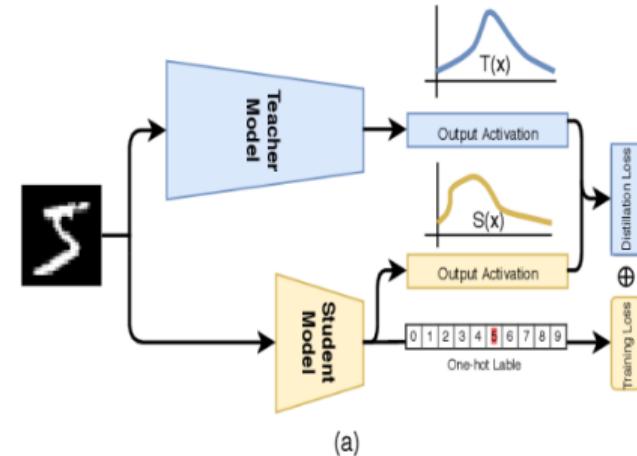
- Model serves as both teacher & student but at different stages of training (first supervised, later using own pseudo-labels) → layer distillation (knowledge from deeper layers to shallower), prediction distillation (re-use own predictions plus hard labels as targets)

### 5. Noisy Student Training

- Strong pretrained teacher model, while the student is trained with noisy inputs (strong augmentations, dropout, ...) learning semantic similarity

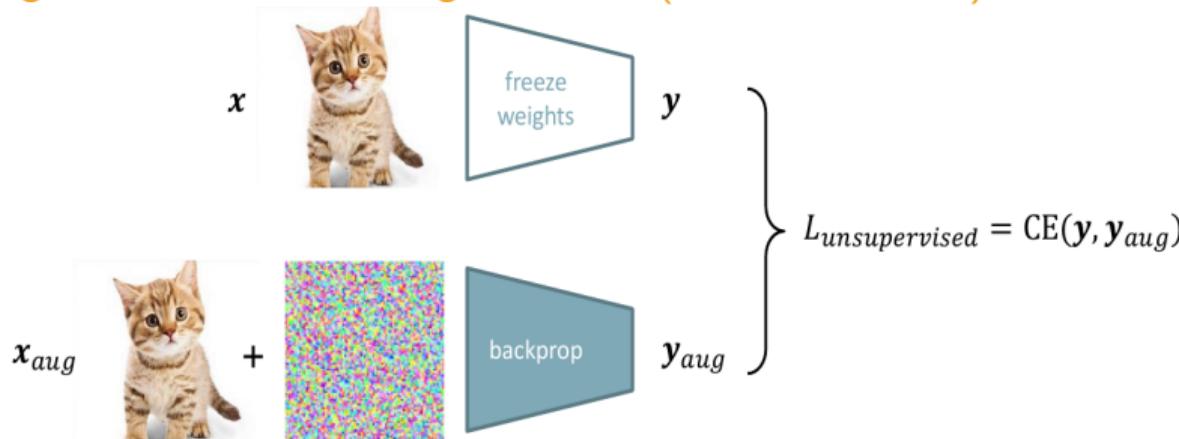
### 6. Teacher-Student Co-Teaching

- Models teach each other by exchanging predictions during training, while the models using the others predictions as targets → Various perspectives!



Source: Image from Zeyi Tao et al., Neuron Manifold Distillation for Edge Deep Learning

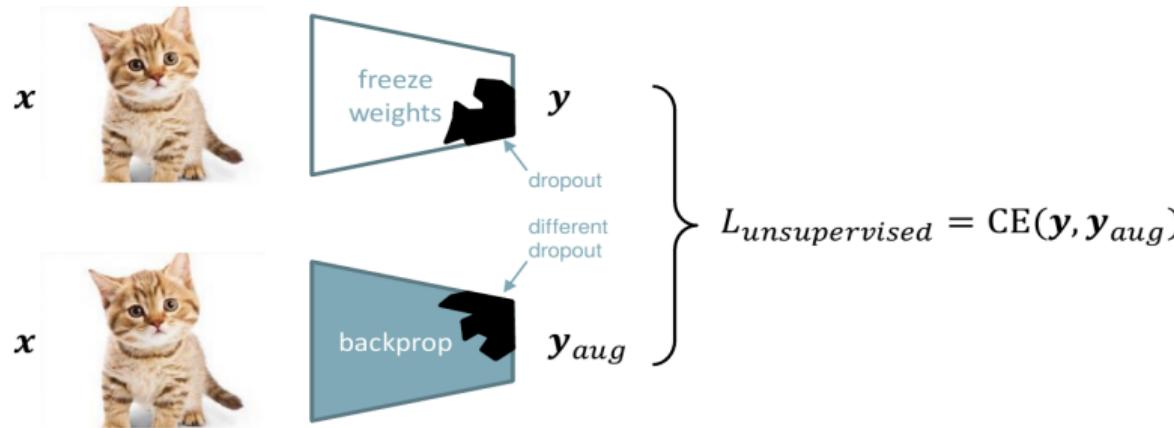
### Consistency Regularization – Data Augmentation (Noise Distortion)



- Computing the unsupervised loss  $L_{unsup}$  by comparing the model prediction  $y = f(x)$  with the model prediction using the augmented input  $y_{aug} = f(x_{aug})$
- Similar samples  $x_{aug}$  are computed via Noise-Augmentation
- Similar/Close Inputs = Similar/Close Predictions

Source: Image from OTH-AW, Electrical Engineering, Media and Computer Science, Thomas Nierhoff – Vorlesung Advanced Topics in ML

### Consistency Regularization – Network Modifications (Dropout)

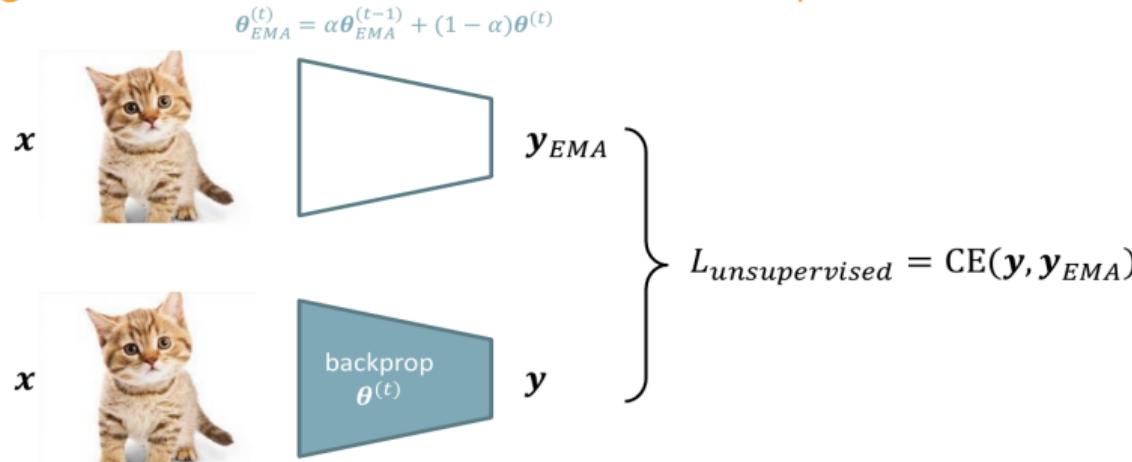


- Computing the unsupervised loss  $L_{unsup}$  by comparing the model prediction  $y = f(x)$  with the model prediction using the augmented input  $y_{aug} = f(x)$
- Identical training samples  $x$  for both models (changing data VS. changing models)
- Differing student-teacher models, e.g. by using distinct dropouts

Source: Image from OTH-AW, Electrical Engineering, Media and Computer Science, Thomas Nierhoff – Vorlesung Advanced Topics in ML

## Semi-Supervised Representation Learning – Handling Unlabeled Data?

### Consistency Regularization – EMA Student-Teacher Concept

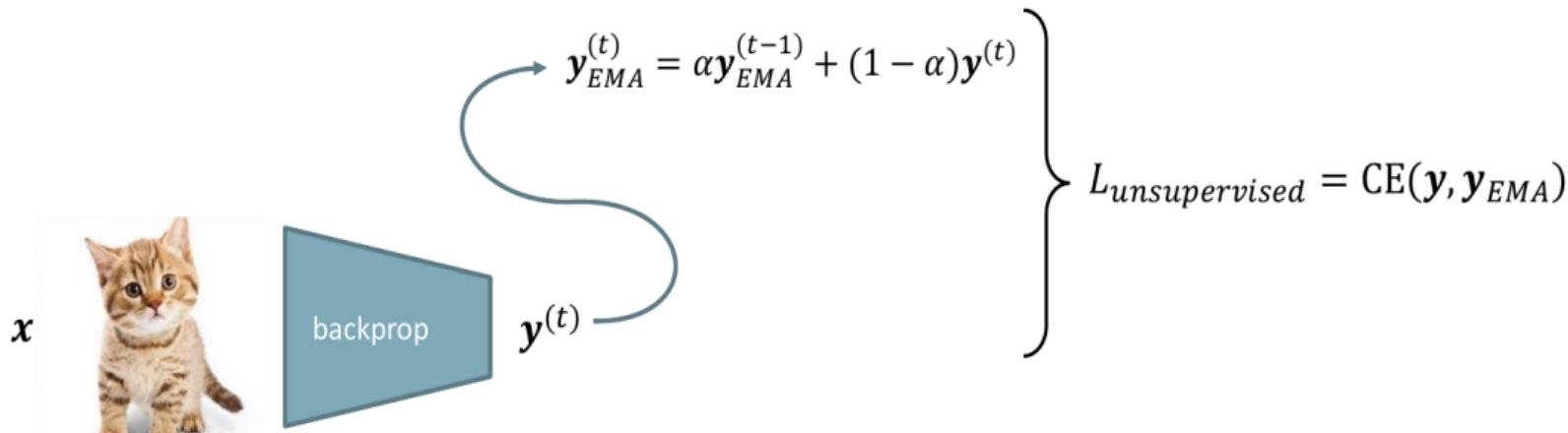


- The teacher's weights are updated as an exponential moving average (EMA) of the student's weights, making the teacher a "smoothed" version of the student over time
- The EMA approach provides a stable target for the student, while allowing the teacher to gradually capture improvements as the student learns

Source: Image from OTH-AW, Electrical Engineering, Media and Computer Science, Thomas Nierhoff – Vorlesung Advanced Topics in ML

## Semi-Supervised Representation Learning – Handling Unlabeled Data?

### Consistency Regularization – Temporal Ensembling Student-Teacher Concept

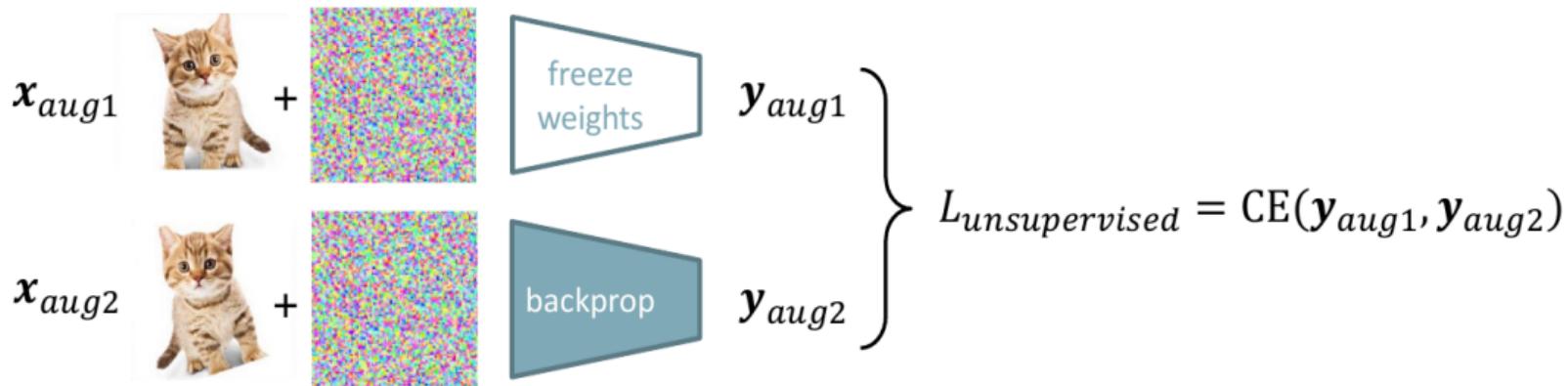


- Instead of maintaining a separate teacher, temporal ensembling aggregates the students predictions over multiple training epochs, to build an ensemble prediction of each sample
- Ensemble prediction acts as a “teacher” for future student predictions, stabilizing learning without an explicit teacher model → Smoothed “ensemble” prediction!

Source: Image from OTH-AW, Electrical Engineering, Media and Computer Science, Thomas Nierhoff – Vorlesung Advanced Topics in ML

## Semi-Supervised Representation Learning – Handling Unlabeled Data?

### Consistency Regularization – Combining Approaches



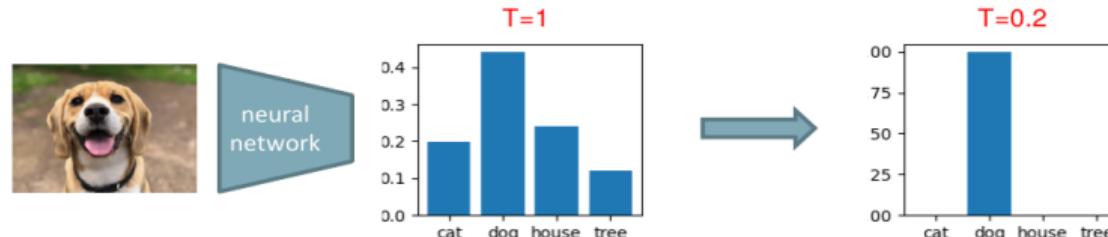
- Computing the unsupervised loss  $L_{unsup}$  between two augmented, noise-distorted versions of a given (unlabeled) input  $y_{aug_1} = f(x_{aug_1})$  and  $y_{aug_2} = f(x_{aug_2})$
- Compare and see also **Π-Model** → 2× Augmentation + Noise + Dropout
- Other loss functions than CE-Loss possible (MSE-Loss/L2-Loss,...)

Source: Image from OTH-AW, Electrical Engineering, Media and Computer Science, Thomas Nierhoff – Vorlesung Advanced Topics in ML

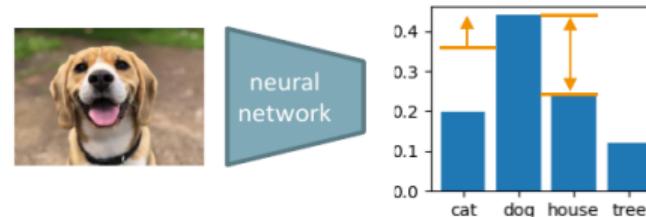
# Deep Learning Strategies – Part I

## Semi-Supervised Representation Learning – Handling Unlabeled Data?

### Consistency Regularization – Softmax Temperature Scaling & Confidence Masking



- Softmax using temperature scaling parameter  $\tau$  (learnable!) to sharpen predictions



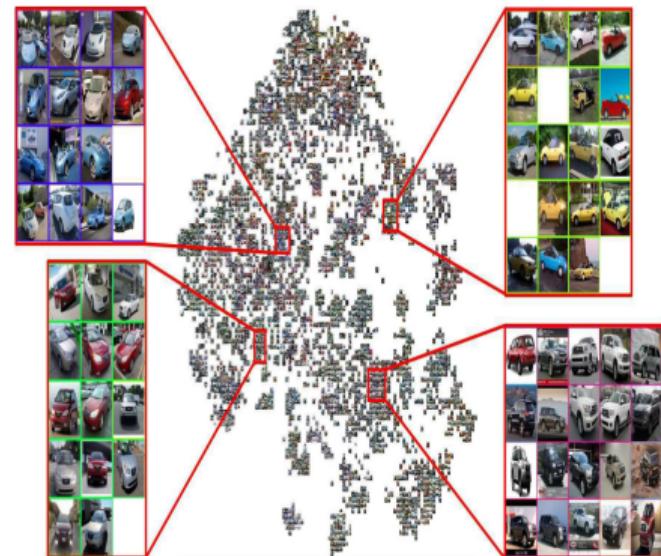
- Only consider samples for unsupervised loss where the classifier is confident (confidence-based masking) largest value  $> x$ , margin between largest/second largest value  $> x$ , entropy  $H = -\sum y_c \log(y_c) < x$

Source: Image from OTH-AW, Electrical Engineering, Media and Computer Science, Thomas Nierhoff – Vorlesung Advanced Topics in ML

# Deep Learning Strategies – Part I

## Deep Metric Learning

- **Goal:** Learn and structure the feature space where similar instances are closer together, and dissimilar instances are further apart

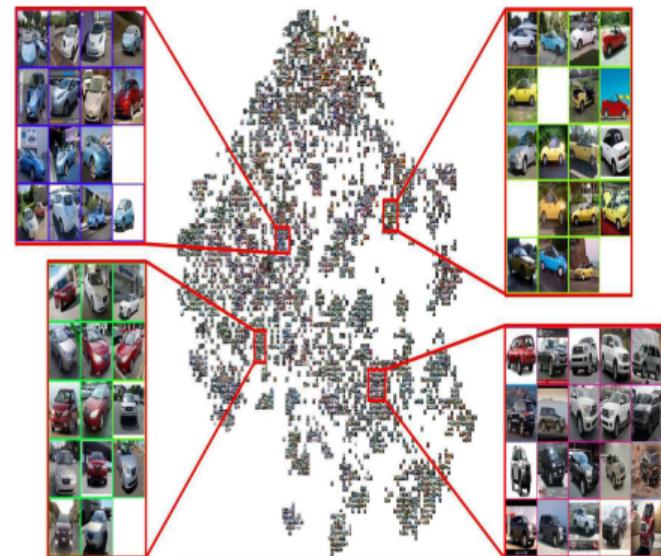


Source: Image from Yueqi Duan et al., Deep Adversarial Metric Learning

# Deep Learning Strategies – Part I

## Deep Metric Learning

- **Goal:** Learn and structure the feature space where similar instances are closer together, and dissimilar instances are further apart
- **Latent Embedding Space:** mapping samples to a continuous vector space to use distance measurements between points reflecting their similarity

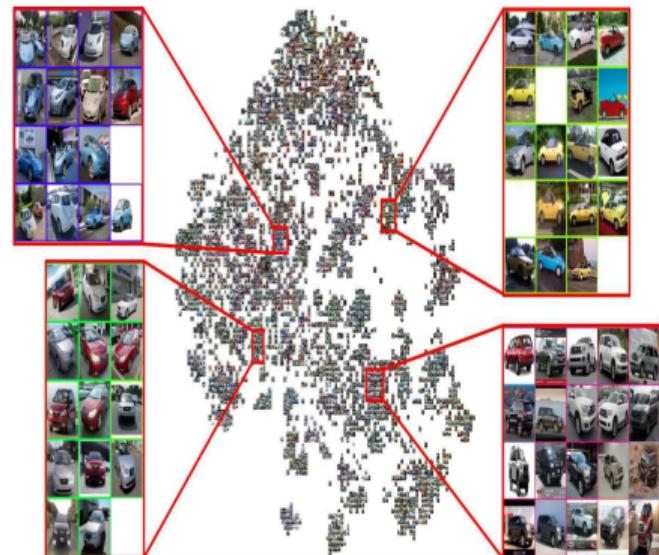


Source: Image from Yueqi Duan et al., Deep Adversarial Metric Learning

# Deep Learning Strategies – Part I

## Deep Metric Learning

- **Goal:** Learn and structure the feature space where similar instances are closer together, and dissimilar instances are further apart
- **Latent Embedding Space:** mapping samples to a continuous vector space to use distance measurements between points reflecting their similarity
- **Distance Metric:** Euclidean distance and cosine similarity to quantify similarity between data samples

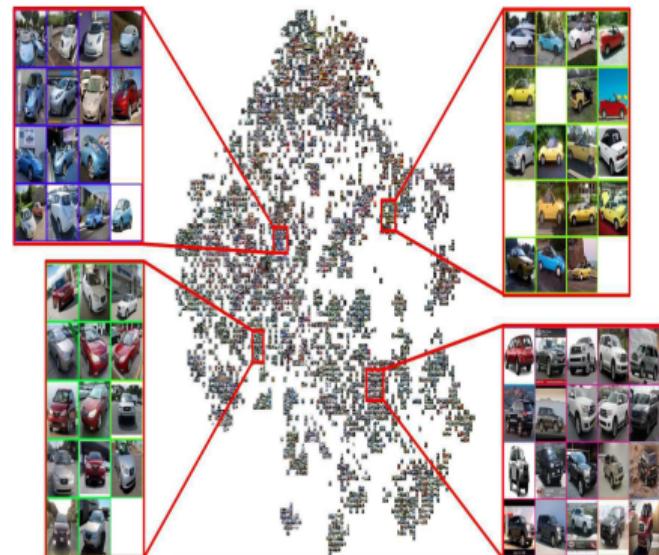


Source: Image from Yueqi Duan et al., Deep Adversarial Metric Learning

# Deep Learning Strategies – Part I

## Deep Metric Learning

- **Goal:** Learn and structure the feature space where similar instances are closer together, and dissimilar instances are further apart
- **Latent Embedding Space:** mapping samples to a continuous vector space to use distance measurements between points reflecting their similarity
- **Distance Metric:** Euclidean distance and cosine similarity to quantify similarity between data samples
- **Loss Functions:**
  - ▶ Contrastive Loss: minimizes distance for similar pairs, maximizes distance for dissimilar pairs
  - ▶ Triplet Loss: ensures the anchor point is closer to a positive example than a negative example
  - ▶ N-pair Loss: generalizes triplet loss with multiple negatives

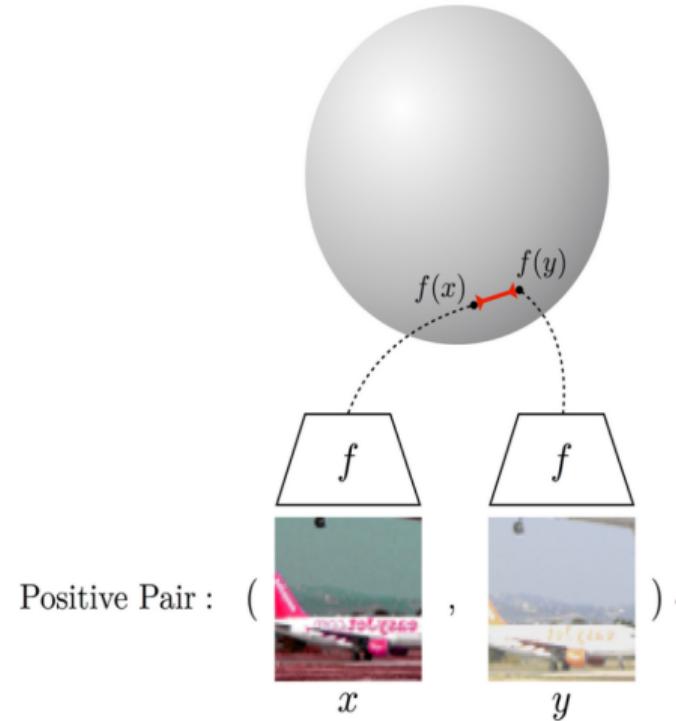


Source: Image from Yueqi Duan et al., Deep Adversarial Metric Learning

# Deep Learning Strategies – Part I

## Contrastive Learning

- **Goal:** learning hidden (latent) representations by contrasting positive and negative pairs



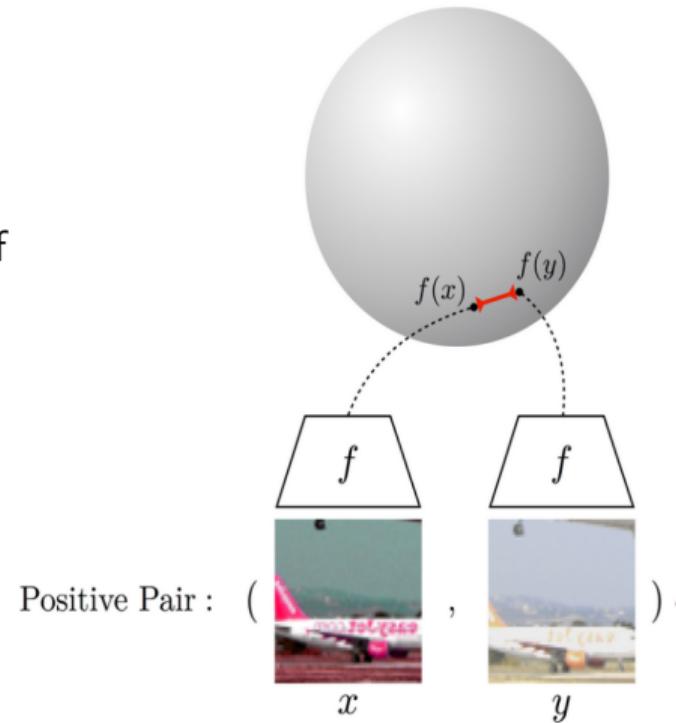
Positive Pair : ( , )

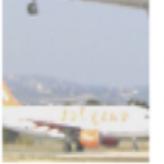
Source: Image from Tongzhou Wang et al., Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere

# Deep Learning Strategies – Part I

## Contrastive Learning

- **Goal:** learning hidden (latent) representations by contrasting positive and negative pairs
- **Self-supervised Learning:** No labels needed, uses data augmentations to create positive pairs (different views of the same instance)



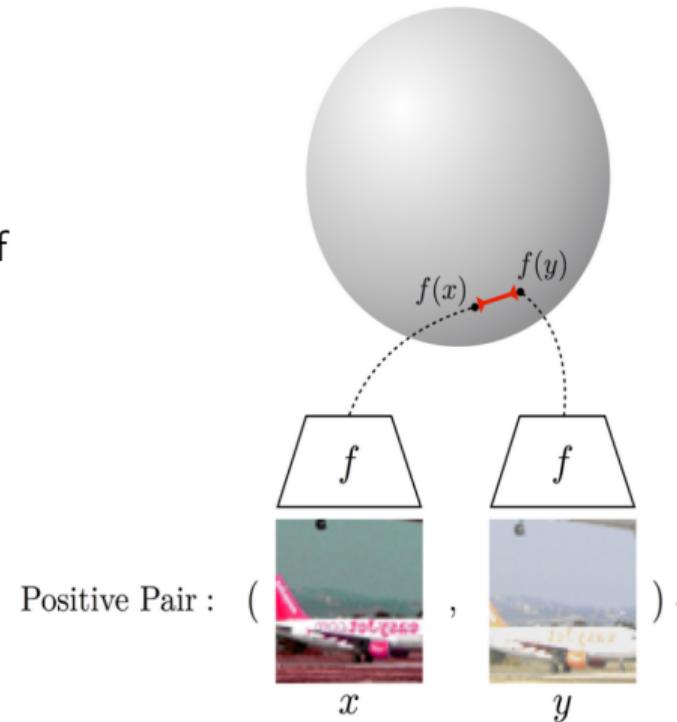
Positive Pair : (  ,  )  
 $x$        $y$

Source: Image from Tongzhou Wang et al., Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere

# Deep Learning Strategies – Part I

## Contrastive Learning

- **Goal:** learning hidden (latent) representations by contrasting positive and negative pairs
- **Self-supervised Learning:** No labels needed, uses data augmentations to create positive pairs (different views of the same instance)
- **Positive vs. Negative Pairs:**
  - ▶ Positive pairs: Augmentations of the same instance
  - ▶ Negative pairs: Samples from different instances



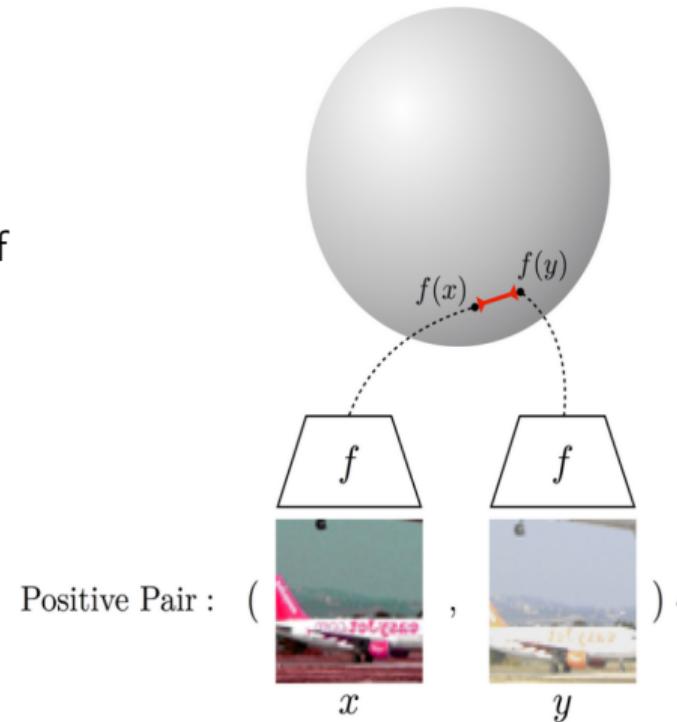
Positive Pair : ( , )

Source: Image from Tongzhou Wang et al., Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere

# Deep Learning Strategies – Part I

## Contrastive Learning

- **Goal:** learning hidden (latent) representations by contrasting positive and negative pairs
- **Self-supervised Learning:** No labels needed, uses data augmentations to create positive pairs (different views of the same instance)
- **Positive vs. Negative Pairs:**
  - ▶ Positive pairs: Augmentations of the same instance
  - ▶ Negative pairs: Samples from different instances
- **Contrastive Loss:** Encourages positive pairs to have similar embeddings while maximizing the distance for negative pairs.



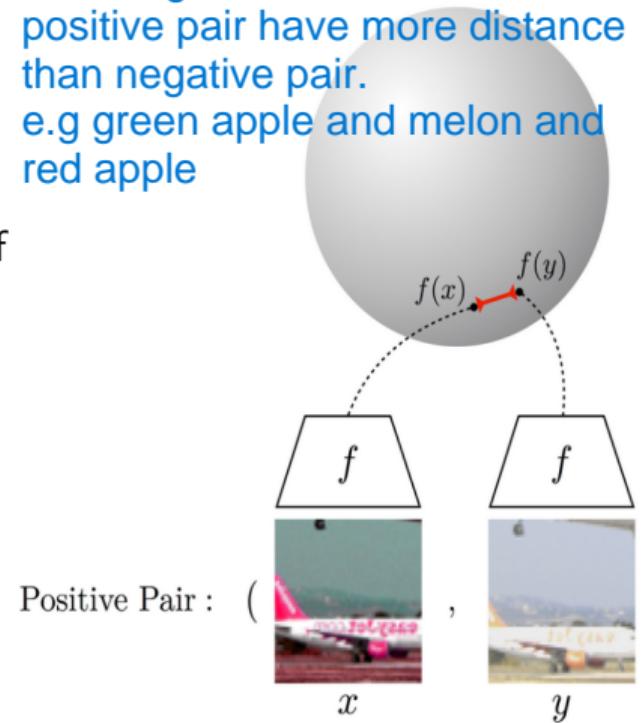
Source: Image from Tongzhou Wang et al., Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere

# Deep Learning Strategies – Part I

## Contrastive Learning

- **Goal:** learning hidden (latent) representations by contrasting positive and negative pairs
- **Self-supervised Learning:** No labels needed, uses data augmentations to create positive pairs (different views of the same instance)
- **Positive vs. Negative Pairs:**
  - ▶ Positive pairs: Augmentations of the same instance
  - ▶ Negative pairs: Samples from different instances
- **Contrastive Loss:** Encourages positive pairs to have similar embeddings while maximizing the distance for negative pairs.
- **Instance Differentiation:** Each instance is considered its own class, creating highly discriminative representations

new negative pairs is important.  
hard negatives sometimes  
positive pair have more distance  
than negative pair.  
e.g green apple and melon and  
red apple

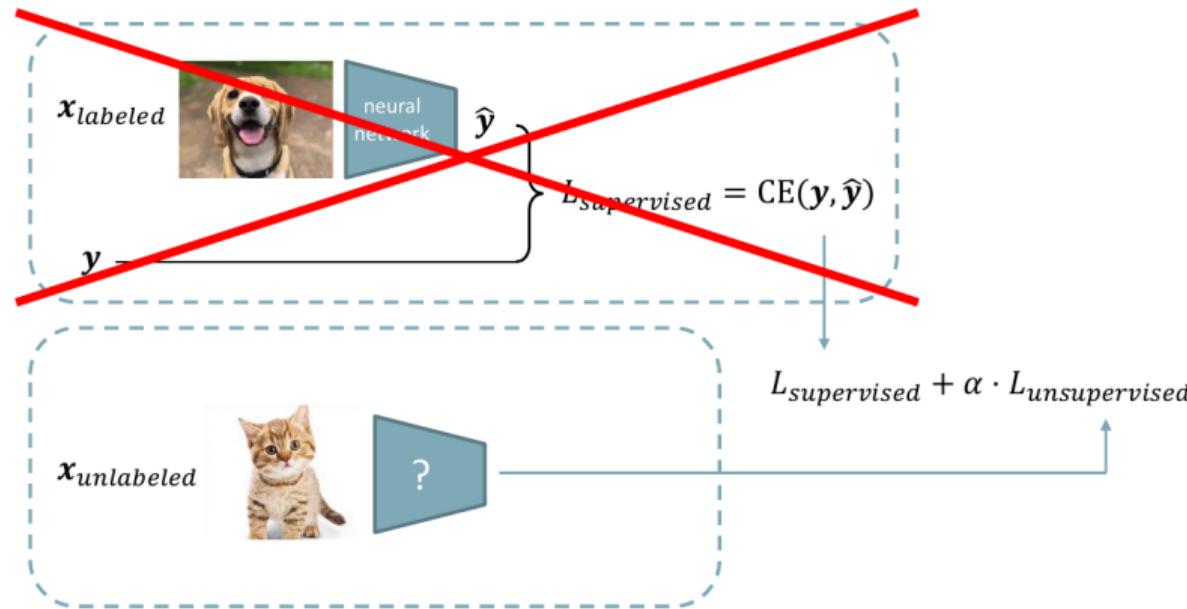


Positive Pair : ( , )

Source: Image from Tongzhou Wang et al., Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere

# Deep Learning Strategies – Part I

## Contrastive Learning

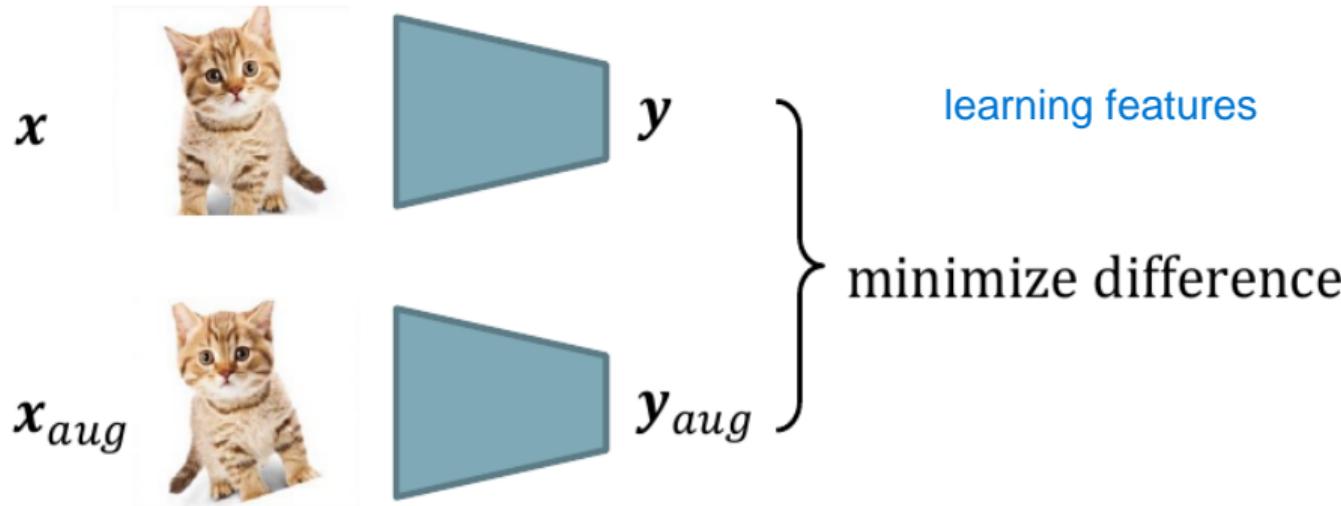


- **Goal:** (Semi-)Supervised learning always requires labeled data → How about computing similarities in a completely unsupervised fashion to build a classification model?

Source: Image from OTH-AW, Electrical Engineering, Media and Computer Science, Thomas Nierhoff – Vorlesung Advanced Topics in ML

# Deep Learning Strategies – Part I

## Contrastive Learning



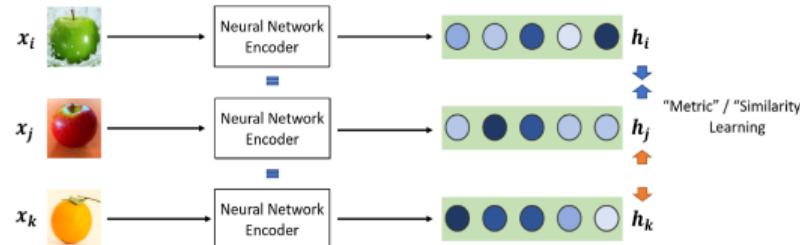
- In semi-supervised learning, a set of augmented data samples  $x_{aug_n}$  was created in order to build similar input pairs  $x/x_{aug_n}$  (no labeling required!), which should result in similar predictions  $y/y_{aug_n}$  according to the smoothness assumption → Minimizing Difference!

Source: Image from OTH-AW, Electrical Engineering, Media and Computer Science, Thomas Nierhoff – Vorlesung Advanced Topics in ML

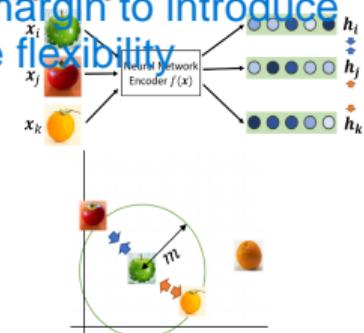
## Contrastive Learning

### Contrastive Loss

- Contrastive loss works with pairs, while each can either be a positive pair (two similar examples) or a negative pair (two dissimilar examples)
- $L(x_i, x_q) = y \cdot \|f(x_i) - f(x_q = x_j)\|_2^2 + (1-y) \cdot \max(0, \epsilon - \|f(x_i) - f(x_q = x_k)\|_2^2)$  with  $y = 1$  (positive pair) and  $y = 0$  (negative pair)
- Minimize distance between embeddings of positive pairs and maximize the distance for negative pairs, up to a certain margin  $\epsilon$



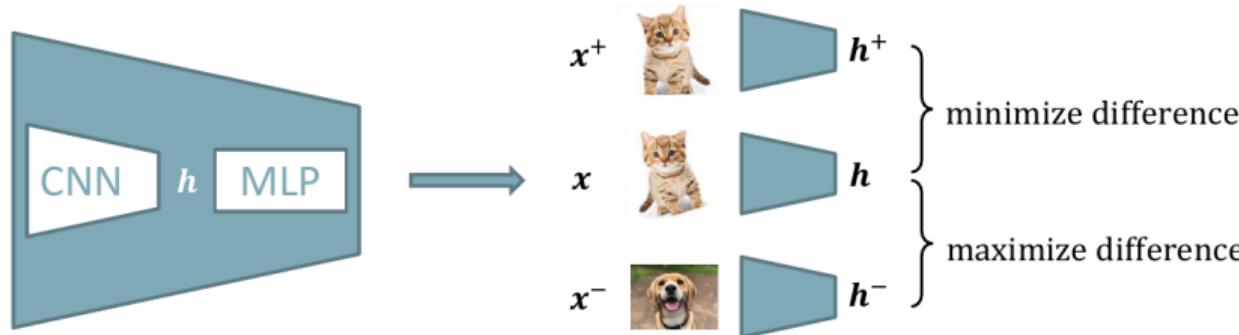
$\epsilon$  is margin to introduce some flexibility



Source: Image from FAU Erlangen-Nuremberg, Pattern Recognition Lab, K.Breininger, V. Christlein, Advanced Deep Learning – Representation Learning

## Contrastive Learning

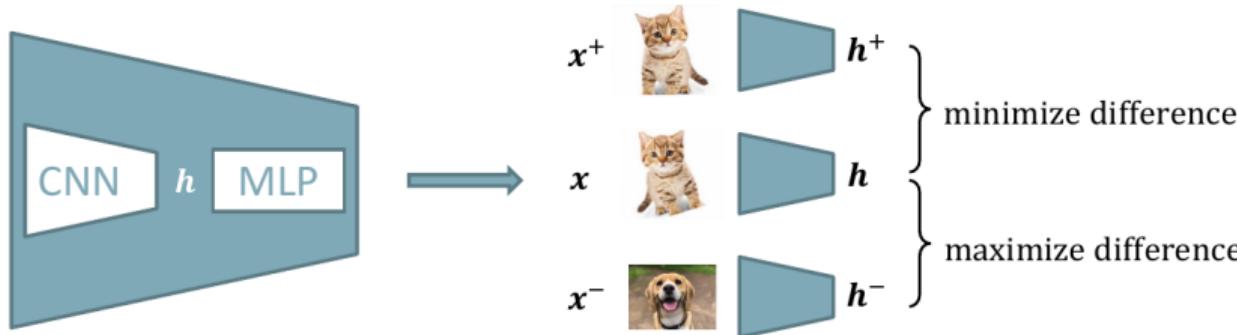
### Triplet Loss



- Extend the pair consisting of an unlabeled data sample  $x$  and a (similar) augmented version  $x_{aug_n} = x^+$  to a **triplet** by introducing a dissimilar sample  $x^-$

Source: Image from OTH-AW, Electrical Engineering, Media and Computer Science, Thomas Nierhoff – Vorlesung Advanced Topics in ML

### Triplet Loss

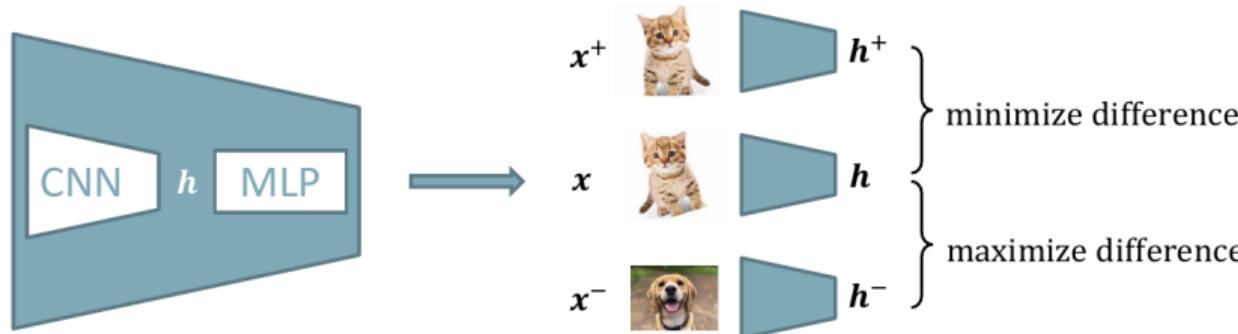


- Extend the pair consisting of an unlabeled data sample  $x$  and a (similar) augmented version  $x_{aug_n} = x^+$  to a **triplet** by introducing a dissimilar sample  $x^-$
- Goal:** minimize difference between prediction  $y$  and positive sample  $y^+$ , while maximizing difference between sample  $y$  and negative sample  $y^-$

Source: Image from OTH-AW, Electrical Engineering, Media and Computer Science, Thomas Nierhoff – Vorlesung Advanced Topics in ML

## Contrastive Learning

### Triplet Loss

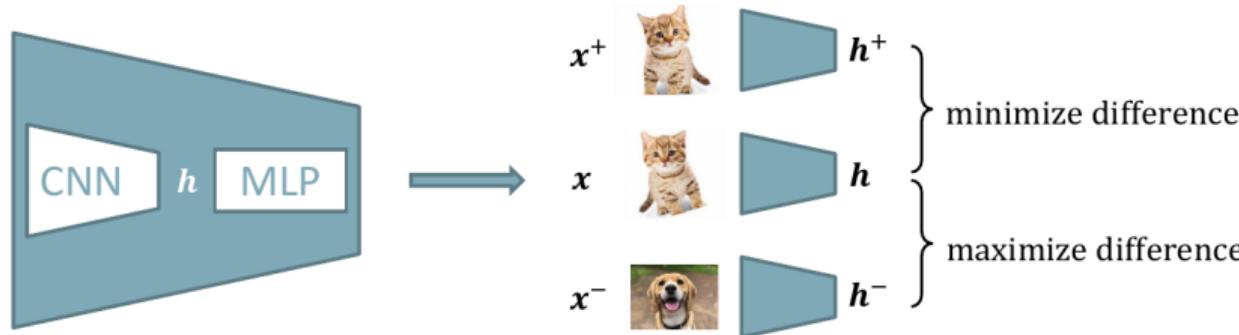


- Extend the pair consisting of an unlabeled data sample  $x$  and a (similar) augmented version  $x_{aug_n} = x^+$  to a **triplet** by introducing a dissimilar sample  $x^-$
- Goal:** minimize difference between prediction  $y$  and positive sample  $y^+$ , while maximizing difference between sample  $y$  and negative sample  $y^-$
- Note:** Comparing  $y$  and  $y^-$  is not practical (no labeled data!), thus the (latent/hidden) features  $h$  are used for comparison instead of the classifier outputs  $y$

Source: Image from OTH-AW, Electrical Engineering, Media and Computer Science, Thomas Nierhoff – Vorlesung Advanced Topics in ML

## Contrastive Learning

### Triplet Loss



- Extend the pair consisting of an unlabeled data sample  $x$  and a (similar) augmented version  $x_{aug_n} = x^+$  to a **triplet** by introducing a dissimilar sample  $x^-$
- Goal:** minimize difference between prediction  $y$  and positive sample  $y^+$ , while maximizing difference between sample  $y$  and negative sample  $y^-$
- Note:** Comparing  $y$  and  $y^-$  is not practical (no labeled data!), thus the (latent/hidden) features  $h$  are used for comparison instead of the classifier outputs  $y$
- Contrastive learning is about to organize the latent feature space!**

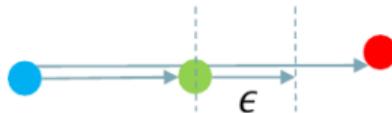
Source: Image from OTH-AW, Electrical Engineering, Media and Computer Science, Thomas Nierhoff – Vorlesung Advanced Topics in ML

# Deep Learning Strategies – Part I

## Contrastive Learning

### Easy negatives

- $L = 0$
- no training possible



### Semi-hard negatives

- $L > 0$
- training difficult



### Hard negatives

- $L > 0$
- training difficult



- **Contrastive Loss:** general idea of forcing to move the original sample  $h$  (anchor, blue) towards the positive sample  $h^+$  (positive, green) and away from the negative feature  $h^-$  (negative, red)

$$L = \|h - h^+\|_2^2 + \max(0, \epsilon - \|h - h^-\|_2^2) = \max(0, \|h - h^+\|_2^2 - \|h - h^-\|_2^2 + \epsilon)$$

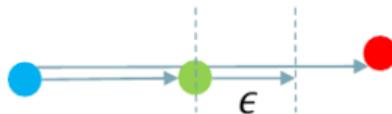
known as **Triplet-Loss**, with: minimization  $\|h - h^+\|_2^2$ , maximization  $\|h - h^-\|_2^2$  up to  $\epsilon$

# Deep Learning Strategies – Part I

## Contrastive Learning

### Easy negatives

- $L = 0$
- no training possible



### Semi-hard negatives

- $L > 0$
- training difficult



### Hard negatives

- $L > 0$
- training difficult



- **Contrastive Loss:** general idea of forcing to move the original sample  $h$  (anchor, blue) towards the positive sample  $h^+$  (positive, green) and away from the negative feature  $h^-$  (negative, red)  
first we train on easy negatives and then dataloaded shuffle after certain epochs which wull include semi and hard -ves

$$L = \|h - h^+\|_2^2 + \max(0, \epsilon - \|h - h^-\|_2^2) = \max(0, \|h - h^+\|_2^2 - \|h - h^-\|_2^2 + \epsilon)$$

known as **Triplet-Loss**, with: minimization  $\|h - h^+\|_2^2$ , maximization  $\|h - h^-\|_2^2$  up to  $\epsilon$

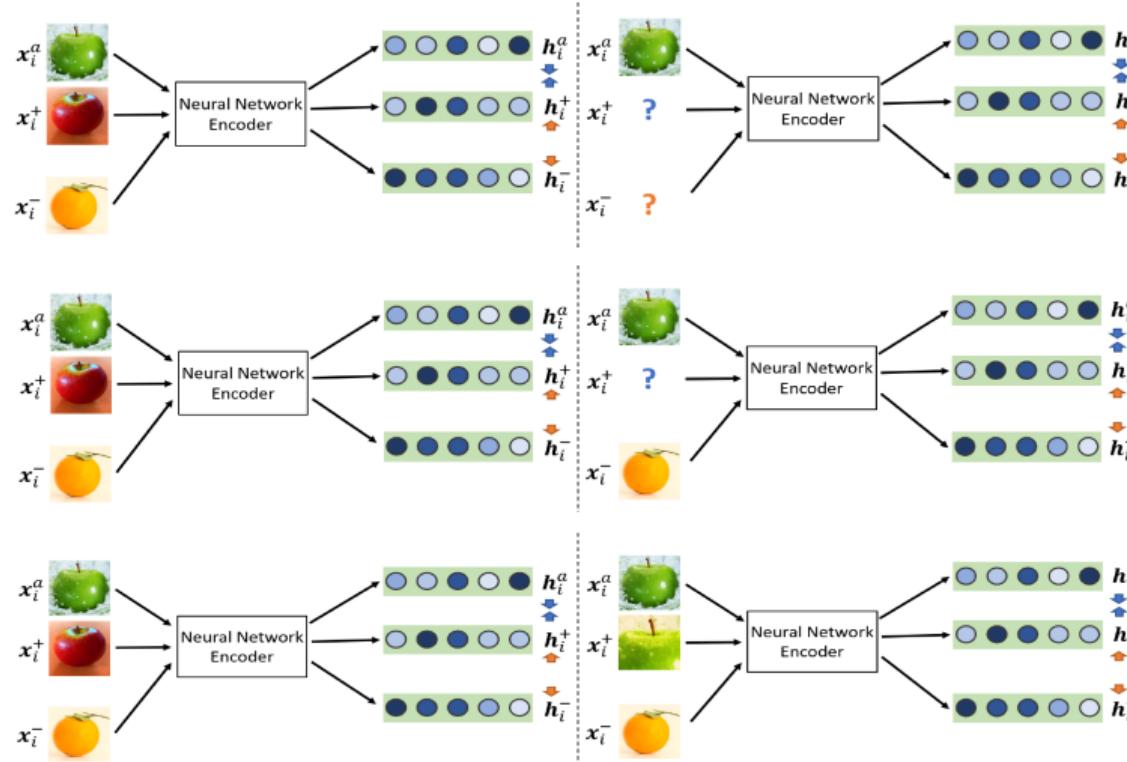
- Easy Triplets:  $\delta(a, p) + \epsilon < \delta(a, n)$
- Semi-Hard Triplet:  $\delta(a, p) < \delta(a, n) < \delta(a, p) + \epsilon$
- Hard Triplets:  $\delta(a, n) < \delta(a, p)$

check this out

Source: Image from OTH-AW, Electrical Engineering, Media and Computer Science, Thomas Nierhoff – Vorlesung Advanced Topics in ML

# Deep Learning Strategies – Part I

## Contrastive Learning – Supervised VS. Unsupervised



Source: Images from FAU Erlangen-Nuremberg, Pattern Recognition Lab, K.Breininger, V. Christlein, Advanced Deep Learning – Representation Learning

# Further Questions?



<https://www.oth-aw.de/hochschule/ueber-uns/personen/bergler-christian/>

Source: <https://emekaboris.medium.com/the-intuition-behind-100-days-of-data-science-code-c98402cdc92c>