# Multi-Target Prediction with Deep Neural Networks: A hands-on Tutorial

Willem Waegeman[1] and Dimitrios Iliadis[1]

[1]Department of Data Analysis and Mathematical Modelling, Ghent University, Belgium

ECML/PKDD 2022, Grenoble, September 19th

# Belgian Evenepoel wins 'historic' Grand Tour at Vuelta a Espana

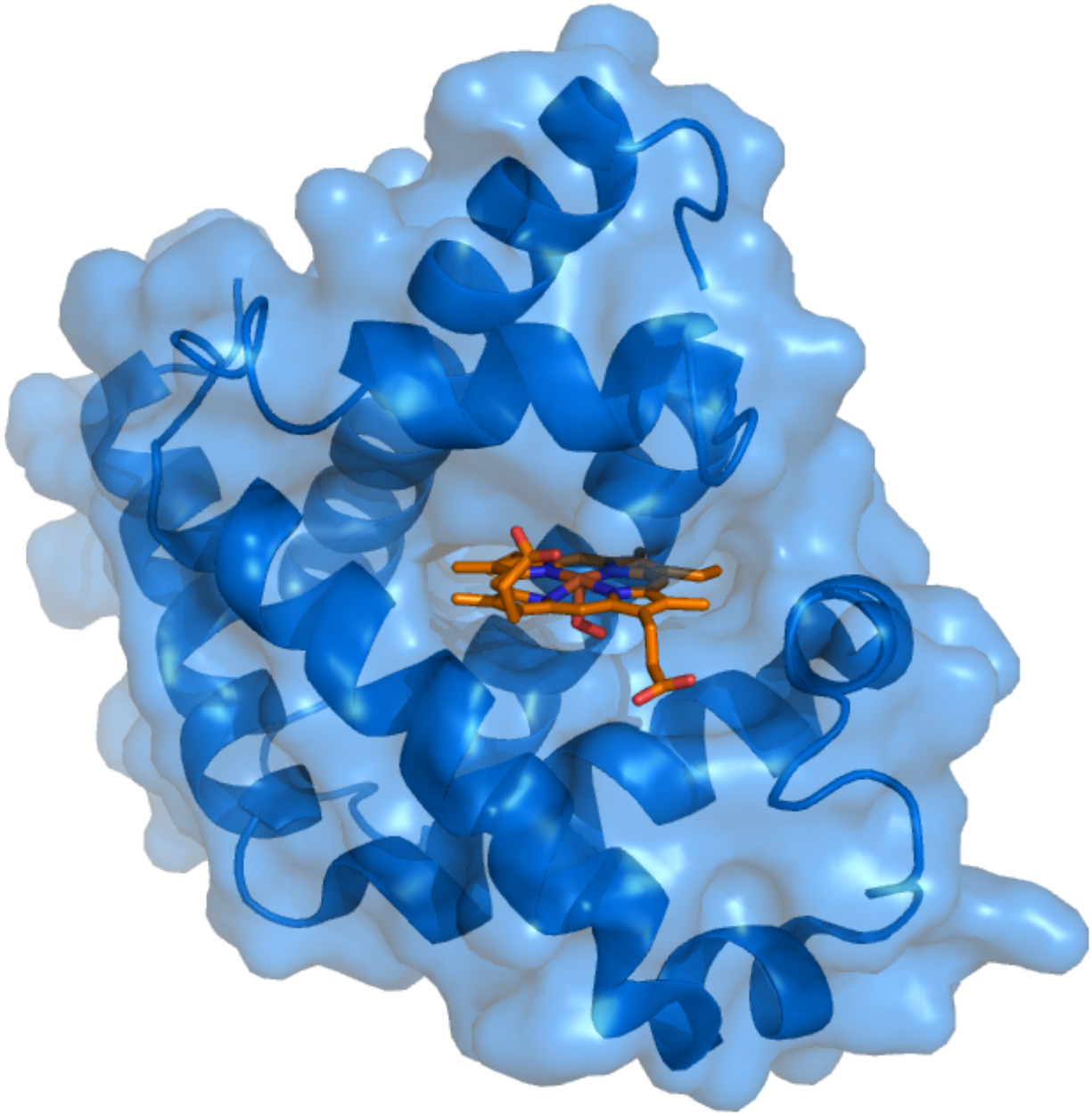
Issued on: 11/09/2022 - 21:58

# Multi-label classification: the example of document categorization

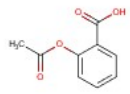
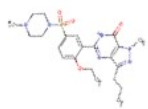
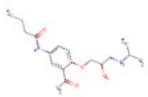
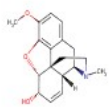
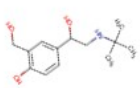
| | | Tennis | Football | Biking | Movies | TV | Belgium |
|---|---|---|---|---|---|---|---|
| 01101 | Text_1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 00111 | Text_2 | 1 | 0 | 0 | 0 | 0 | 1 |
| 01110 | Text_3 | 0 | 0 | 0 | 1 | 1 | 0 |
| 10001 | Text_4 | 0 | 0 | 1 | 0 | 1 | 0 |
| 01011 | Text_5 | 1 | 0 | 0 | 1 | 0 | 0 |
| 11110 | Text_6 | ? | ? | ? | ? | ? | ? |

# Another multi-label classification example: gene function prediction


| | func class_1 | func class_2 | func class_3 | func class_4 | func class_5 |
|---|---|---|---|---|---|
| gene_1 | 1 | 0 | 0 | 0 | 0 |
| gene_2 | 0 | 0 | 1 | 0 | 1 |
| gene_3 | 0 | 1 | 1 | 0 | 0 |
| gene_4 | 1 | 1 | 0 | 1 | 0 |
| gene_5 | 1 | 1 | 0 | 0 | 0 |
| gene_6 | ? | ? | ? | ? | ? |



# Multivariate regression: the example of protein-ligand interaction prediction

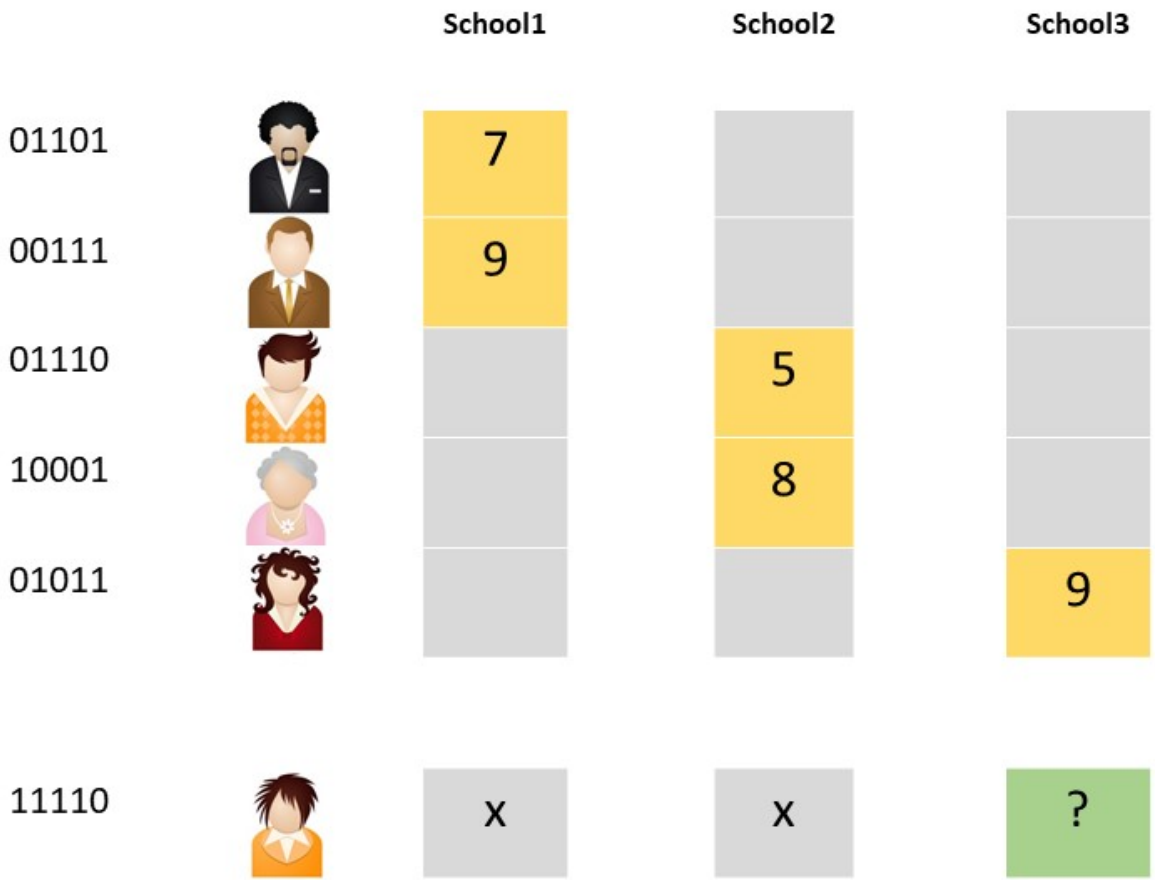
| | Mol1 | Mol2 | Mol3 | Mol4 | Mol5 | Mol6 |
|---|---|---|---|---|---|---|
|  | 1.3 | 0.2 | 1.4 | 1.7 | 3.5 | 1.3 |
|  | 2 | 1.7 | 1.5 | 7.5 | 8.2 | 7.6 |
|  | 0.2 | 0 | 0.3 | 0.4 | 1.2 | 2.2 |
|  | 3.1 | 1.1 | 1.3 | 1.1 | 1.7 | 5.2 |
|  | ? | ? | ? | ? | ? | ? |

# Another multivariate regression example: predicting environmental pollution

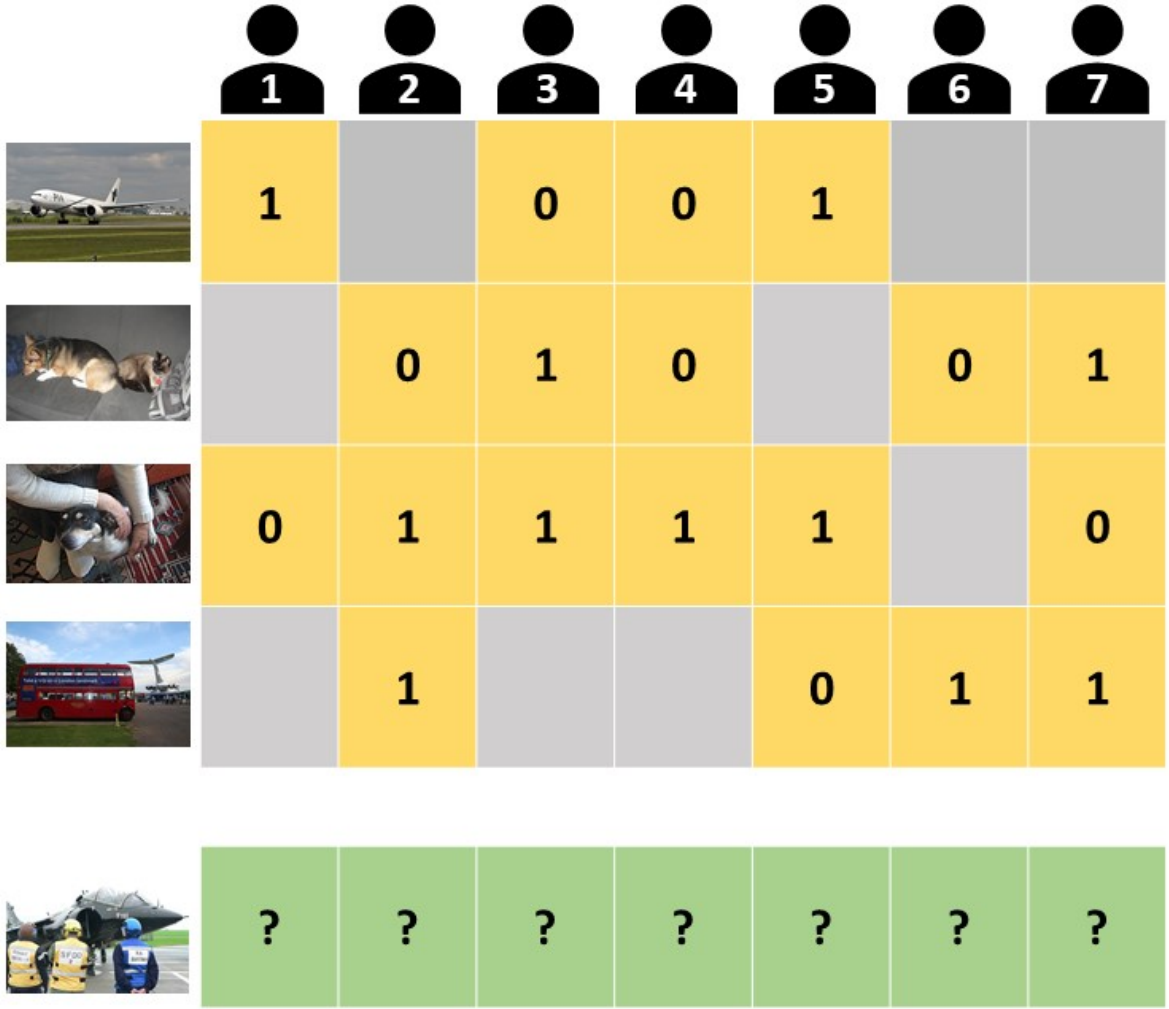
| | Cd | Co | Cu |
|---|---|---|---|
| location_1 | 1.74 | 9.32 | 25.7 |
| location_2 | 1.33 | 10 | 24.7 |
| location_3 | 10.6 | 10.6 | 8.88 |
| location_4 | 2.15 | 11.9 | 22.7 |
| location_5 | 1.56 | 16.3 | 34.3 |
| location_6 | ? | ? | ? |


SUNNYBANK
MIAMI
SEMPER FLOREAT COSTA
KEDRON
TRINITY BAY
ENDEAVOUR
YERONGA
SCIENTIA AC HUMANITATE
CORINDA
ROCHEDALE
SHS
RECTE CRESCAM
The Cathedral School
TOWNSVILLE
BRISBANE STATE HIGH SCHOOL
SCIENTIA est POTESTAS
FERNY GROVE STATE HIGH SCHOOL
ALWAYS AIM HIGH
ELANORA STATE HIGH SCHOOL
URANGAN S.H.S.
WORKING TOGETHER
CENTENARY HEIGHTS
CLEVELAND
HIGH
SEMPER DIGNE
SUNSHINE BEACH HIGH
QUALITY IN ALL WE DO
MERRIMAC STATE HIGH SCHOOL
PRIDE IN EXCELLENCE
MAROOCHYDORE
HIGH SCHOOL
NITI ET SERVIRE
STATE HIGH SCHOOL
CAIRNS
COMPASSION
FAITH
?

# Multi-task learning:
# the example of predicting student marks



# A second multi-task learning example: predicting whether users labeled an image correctly



# Motivation for this tutorial

- Over the last two decades, a lot of tailor-made methods that solve specific multi-target prediction methods have been proposed
- There is a need to understand which methods are useful under which conditions
- In addition, there is also a need for generic software tools that can be employed in a semi-automated way
- We believe that deep learning methods have generic building blocks that can be used for tackling various MTP problems

Key references:

W. Waegeman, K. Dembczynski, E. Hüllermeier. Multi-target prediction: A unifying view on problems and methods. Data Mining and Knowledge Discovery, 33(2), 2019.

D. Iliadis, B. De Baets and W. Waegeman. Multi-target prediction for dummies with two-branch neural networks, Machine Learning 2022.

# Overview of this talk

1. Introduction (10 min)
2. A unifying view on MTP problems (20 min)
3. A unifying view on MTP methods (50 min)
4. Coffee break (30 min)
5. Hands-on part (80 min)

# General framework

## Definition (Multi-target prediction)

A multi-target prediction setting is characterized by instances $\boldsymbol{x} \in \mathcal{X}$ and targets $\boldsymbol{t} \in \mathcal{T}$ with the following properties:

P1. A training dataset $\mathcal{D}$ consists of triplets $(\boldsymbol{x}_i, \boldsymbol{t}_j, y_{ij})$, where $y_{ij} \in \mathcal{Y}$ denotes a score that characterizes the relationship between the instance $\boldsymbol{x}_i$ and the target $\boldsymbol{t}_j$.

P2. In total, $n$ different instances and $m$ different targets are observed during training, with $n$ and $m$ finite numbers. Thus, the scores $y_{ij}$ of the training data can be arranged in an $n \times m$ matrix $Y$, which is in general incomplete, i.e., $Y$ may have missing values.

P3. The score set $\mathcal{Y}$ consists of nominal, ordinal and/or real values.

P4. The goal consists of predicting scores for any instance-target couple $(\boldsymbol{x}, \boldsymbol{t}) \in \mathcal{X} \times \mathcal{T}$.

# Conventional MTP settings

- Side information for targets is normally not available.
- Multi-label classification: (e.g., assigning appropriate category tags to documents).
- Multivariate regression: (e.g., predicting whether a protein will bind to a set of experimentally developed small molecules).
- Multi-task learning: (e.g., predicting student marks in the final exam for a typical high-school course).
- Other settings: label ranking, multi-dimensional classification, etc.

# Conventional MTP settings

## Definition (Multi-label classification)

A multi-label classification problem is a specific instantiation of the general framework, which exhibits the following additional properties:

P5. The cardinality of $\mathcal{T}$ is $m$; this implies that all targets are observed during training.

P6. No side information is available for targets. Again, without loss of generality, we can hence identify targets with natural numbers, such that the target space is $\mathcal{T} = \{1, ..., m\}$.

P7. The score matrix $Y$ has no missing values.

P8a. The score set is $\mathcal{Y} = \{0, 1\}$.

# Conventional MTP settings

## Definition (Multivariate regression)

A multivariate regression problem is a specific instantiation of the general framework, which exhibits the following additional properties:

P5. The cardinality of $\mathcal{T}$ is $m$. This implies that all targets are observed during training.

P6. No side information is available for targets. Without loss of generality, we can hence assign the numbers $1$ to $m$ as identifiers to targets, such that the target space is $\mathcal{T} = \{1, ..., m\}$.

P7. The score matrix $Y$ has no missing values.

P8b. The score set is $\mathcal{Y} = \mathbb{R}$.

# Conventional MTP settings

## Definition (Multi-task learning)

A multi-task learning problem is a specific instantiation of the general framework, which exhibits the following additional properties:

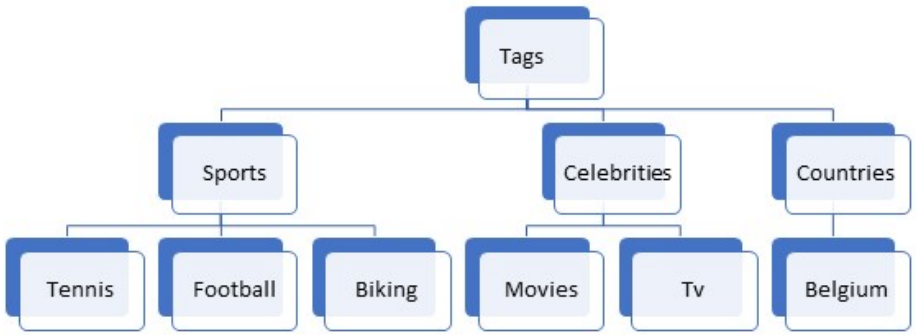
P5. The cardinality of $\mathcal{T}$ is $m$; this implies that all targets are observed during training.

P6. No side information is available for targets. Again, the target space can hence be taken as $\mathcal{T} = \{1, ..., m\}$.

P7. ~~The score matrix $Y$ has no missing values~~.

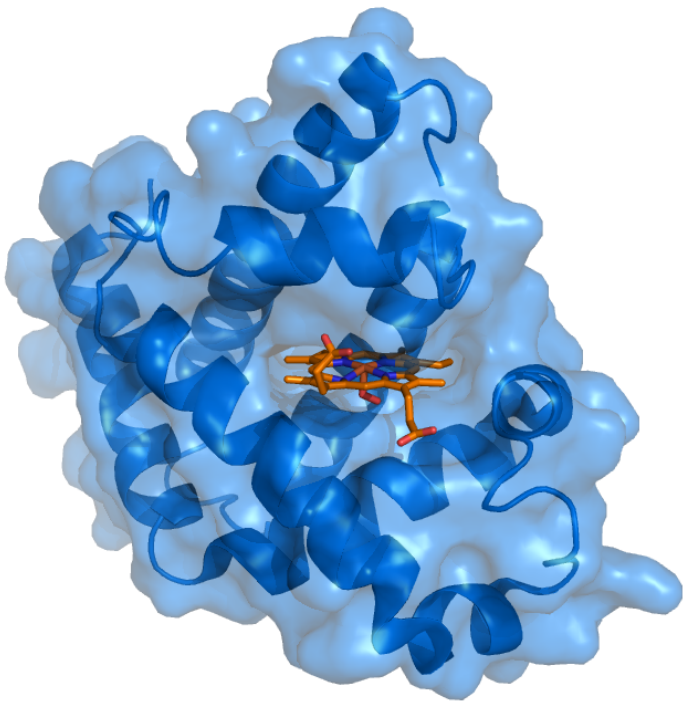
P8c. The score set consists of binary or real values: $\mathcal{Y} = \{0, 1\}$ or $\mathcal{Y} = \mathbb{R}$.

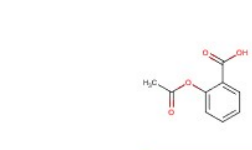
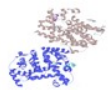
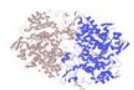
# Let's assume a document hierarchy:
# How would you call this machine learning problem?



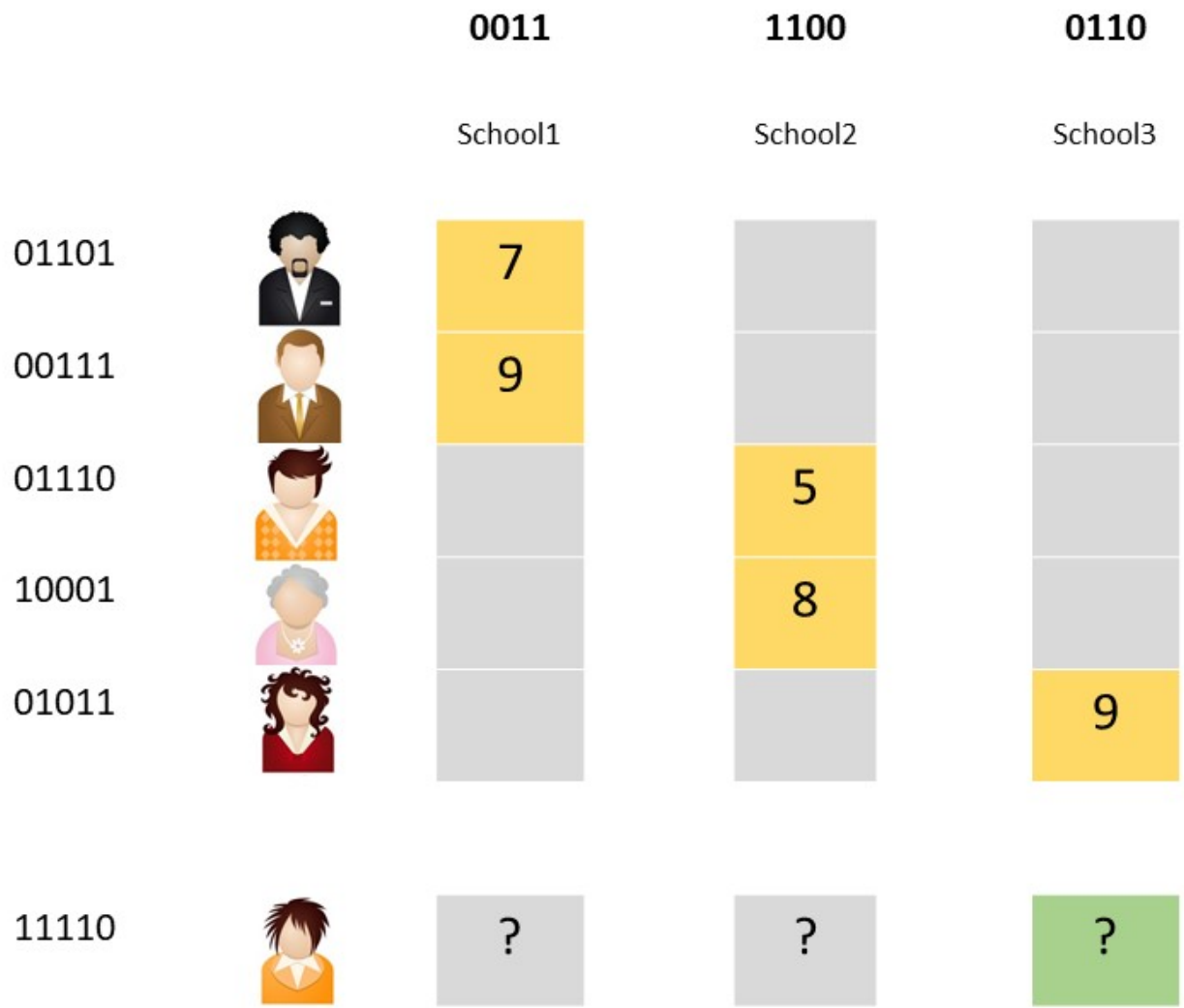

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01101 | **Text1** | 0 | 0 | 0 | 0 | 0 | 1 |
| 00111 | **Text2** | 0 | 0 | 1 | 0 | 1 | 1 |
| 01110 | **Text3** | 0 | 0 | 0 | 1 | 1 | 0 |
| 10001 | **Text4** | 0 | 0 | 1 | 0 | 1 | 0 |
| 01011 | **Text5** | 1 | 0 | 0 | 1 | 0 | 0 |
| 11110 | **Text6** | ? | ? | ? | ? | ? | ? |

# Let's assume a structured representation:
# How would you call this machine learning problem?



| | | | | | |
|---|---|---|---|---|---|
|  | | | | | |
|  | 1.3 | 0.2 | 0.9 | 1.9 | 3.1 |
|  | 1.7 | 5.1 | 0.3 | 0.1 | 1.1 |
|  | 0.1 | 2.2 | 0.1 | 0.5 | 4.2 |
|  | 0.9 | 1.4 | 1.6 | 0.2 | 0.8 |
|  | ? | ? | ? | ? | ? |

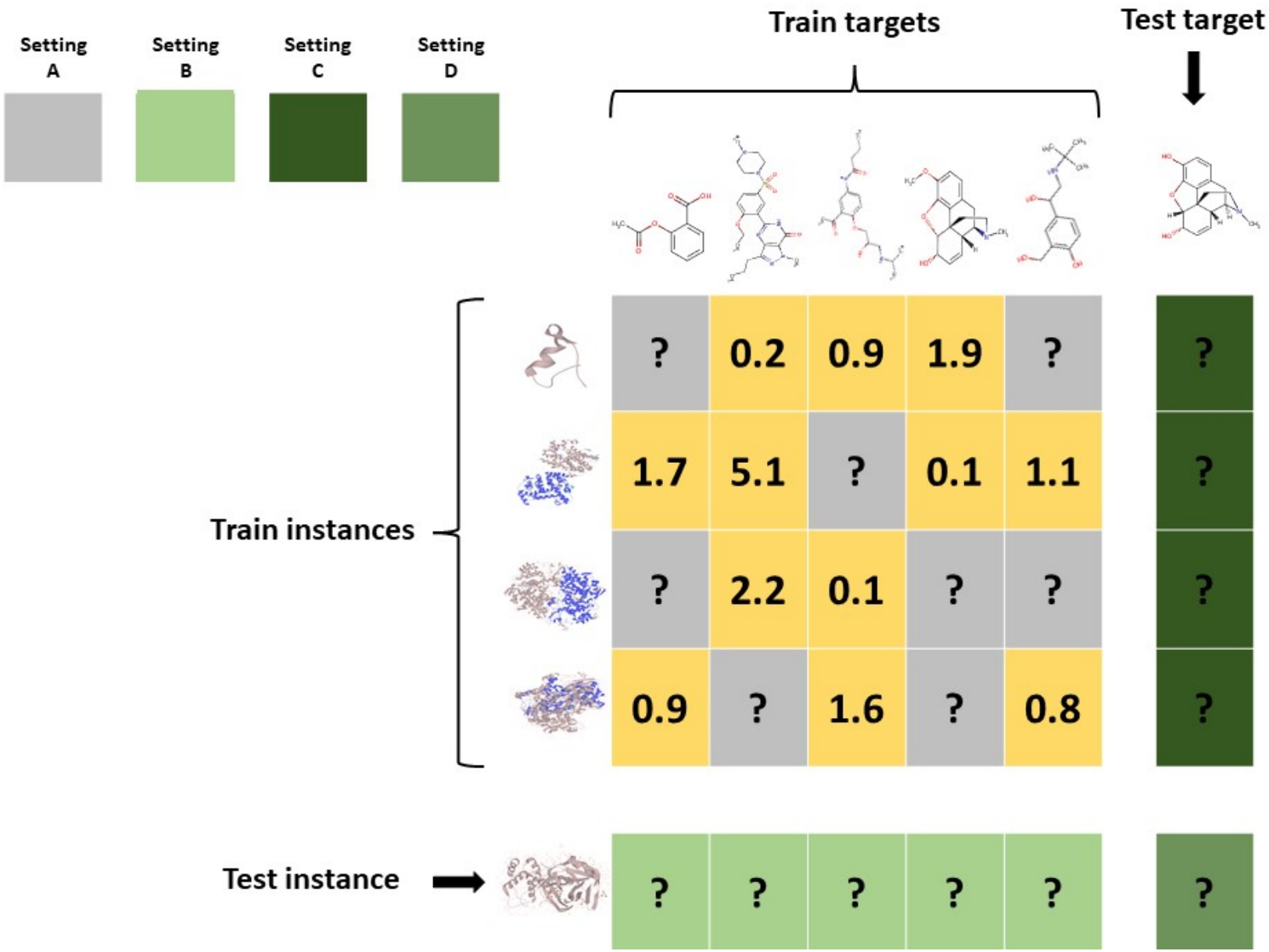
# Let's assume a vector representation:
# How would you call this machine learning problem?



# Learning with side information on targets

- Examples:
  - Taxonomy on document categories (knowledge of relations between targets).
  - Representation for the target molecules in drug design application (structured representation).
  - Information about schools and courses (geographical location, qualifications of the teachers, reputation of the school, etc.) in student mark forecasting application (vector or feature representation).
- Such problems are often referred to as dyadic prediction, pairwise learning, link prediction, or network inference settings.
- However, MTP terminology is rarely used in this literature.
- Side information is of crucial importance for generalizing to novel targets that are unobserved during the training phase.

# Inductive versus transductive MTP problems



# Inductive versus transductive learning problems

## Definition (Zero-shot learning)

A zero-shot learning problem is a specific instantiation of the general framework with the following additional property:

P5*. $m < m^* = |\mathcal{T}|$. Some targets are hence not observed during training, but may nevertheless appear at prediction time.

- By substituting P5 with P5*, one now tackles problems that are inductive instead of transductive w.r.t. targets.
- The same subdivision can be made for instances.
- In total, the four different settings referred to as A, B, C, D can be distinguished (in the presence of side information).
- Theoretically, settings B and C are identical/symmetric, though there are practical differences/asymmetries.

# A typical application of Setting A: recommender systems

# Inductive versus transductive learning problems

## Definition (Matrix completion)

A matrix completion problem is a specific instantiation of the general framework with the following additional properties:

P5. The cardinality of $\mathcal{T}$ is $m$. This implies that all targets are observed during training.

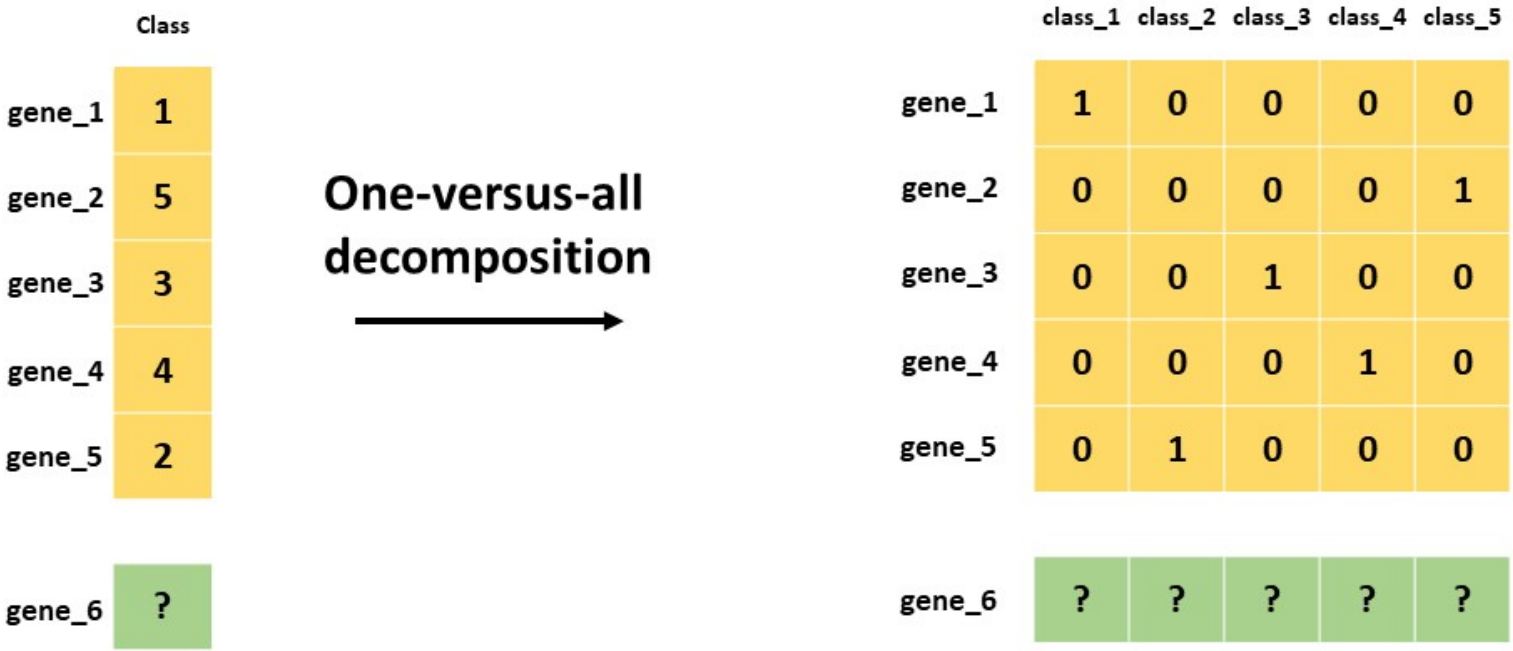
P6. No side information is available for targets. Without loss of generality, we can hence assign identifiers to targets from the set $\{1, ..., m\}$ such that the target space is $\mathcal{T} = \{1, ..., m\}$.

P9. The cardinality of $\mathcal{X}$ is $n$. This implies that all instances are observed during training.

P10. No side information is available for instances. Without loss of generality, we can hence assign identifiers to instances from the set $\{1, ..., n\}$, such that the instance space is $\mathcal{X} = \{1, ..., n\}$.

# What we usually don't cover under MTP

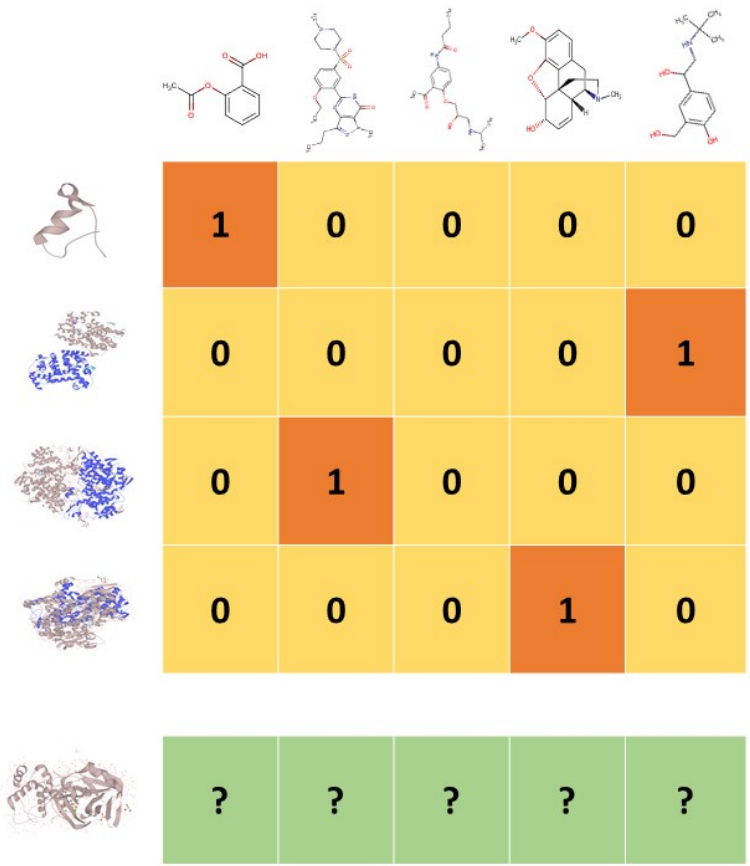
## Multi-class classification




- The one-versus-all decomposition of multi-class classification could be seen as a multi-target prediction problem
- Other decompositions (one-versus-one, ECOC, etc.) cannot be represented using the MTP framework

# What we usually don't cover under MTP

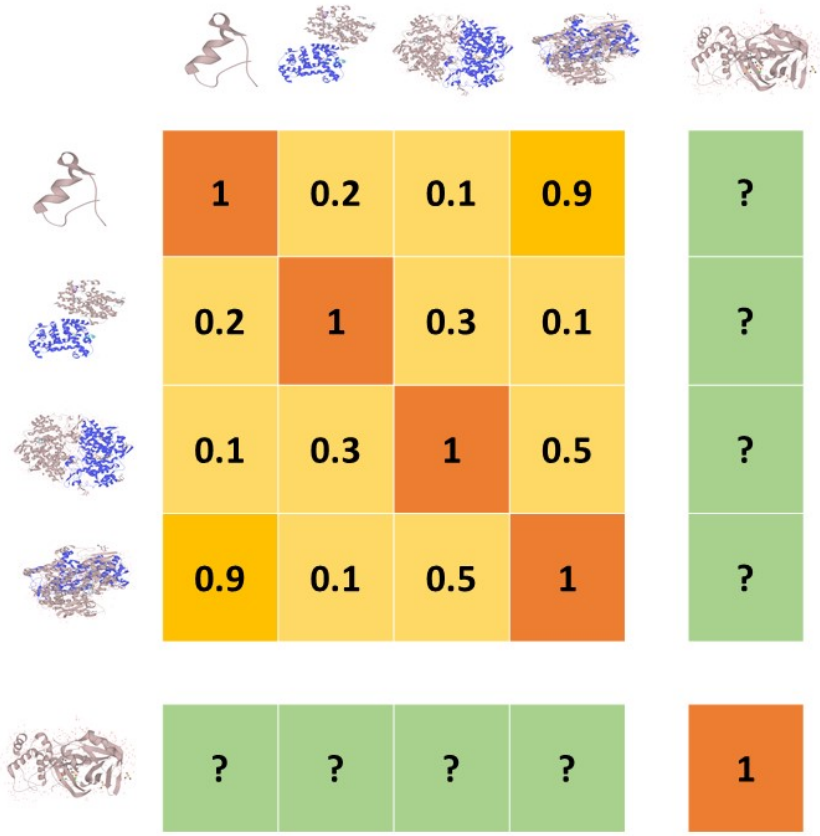
## Structured output prediction




- Structured output prediction considers a mapping of the form $\mathcal{X} \rightarrow \mathcal{T}$
- Could be covered by considering a target representation $\boldsymbol{t}_j \in \mathcal{T}$ as an output
- Complications arise because the cardinality of $\mathcal{T}$ would be very large, or even infinite

# What we usually don't cover under MTP

## Learning monadic relations




- Settings where in $Y$ the rows and columns are identical, thus $\mathcal{X} = \mathcal{T}$
- Appears in various areas such as metric learning, similarity learning, link prediction, pairwise preference learning, etc.
- Complications arise since the matrix $Y$ would exhibit additional properties, such as symmetry, antisymmetry, reciprocity, etc.

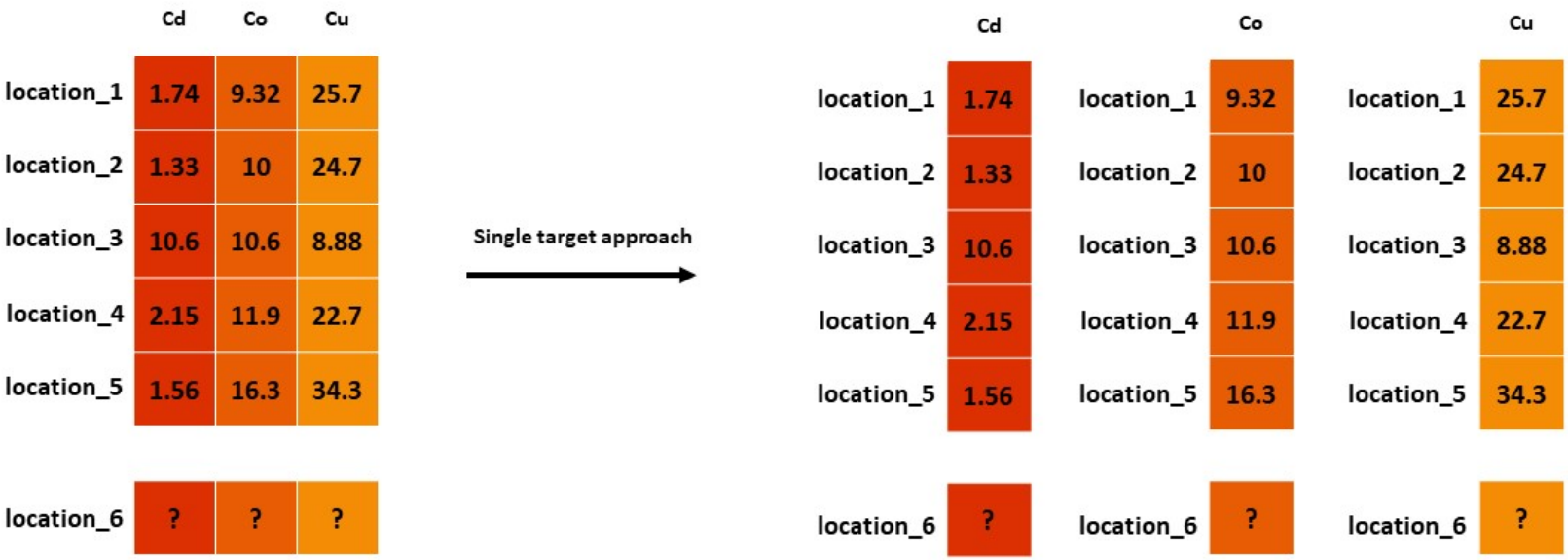
# Overview of this talk

1. Introduction (10 min)
2. A unifying view on MTP problems (20 min)
3. A unifying view on MTP methods (50 min)
4. Coffee break (30 min)
5. Hands-on part (80 min)

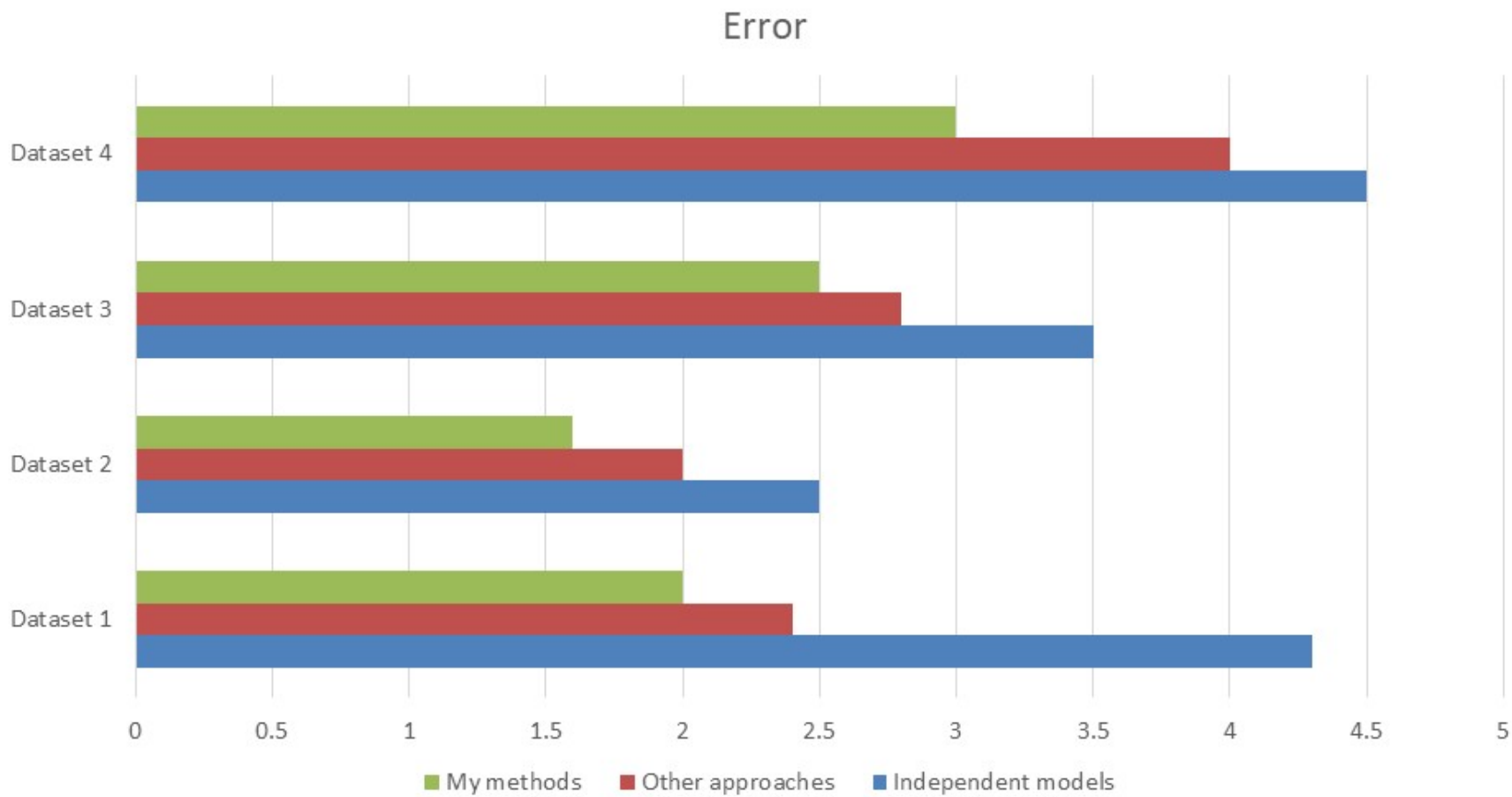
# A unifying view on MTP methods

| Group of methods | Applicable setting |
|---|---|
| **Independent models** | B |
| Similarity-enforcing methods | B |
| Relation-exploiting methods | B and D |
| Relation-constructing methods | B |
| Representation-exploiting methods | B and D |
| Representation-constructing methods | A and B |

# A baseline method:
# learning a model for each target independently



# The results section of a typical MTP paper...

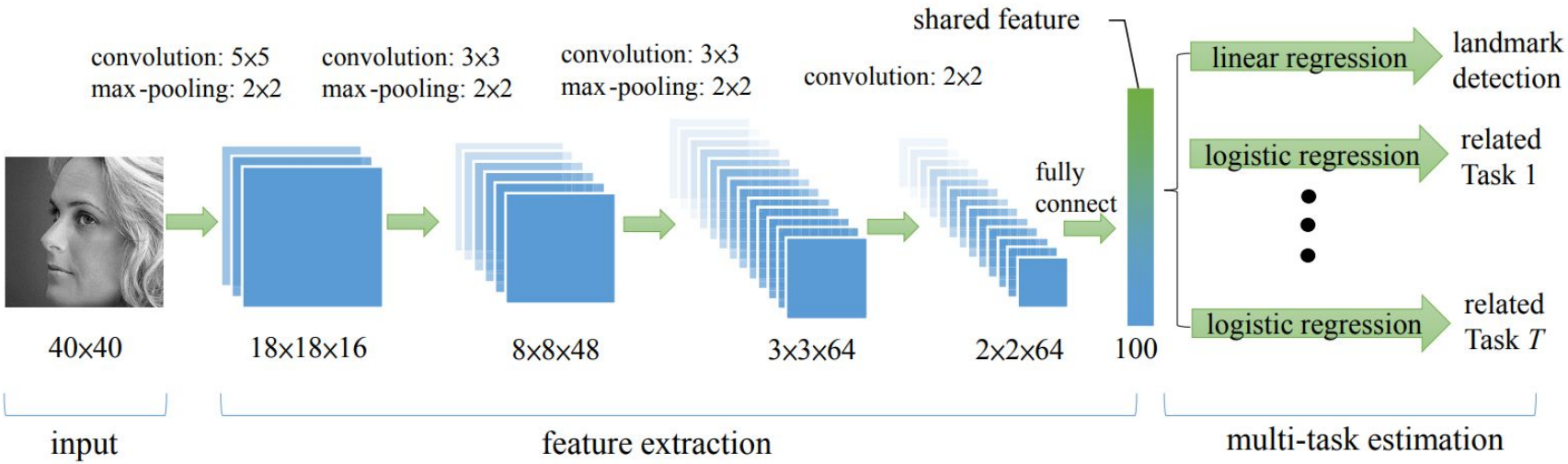



Independent models a.k.a. binary relevance, models that do not exploit target dependencies, one-versus-all, etc.

# A unifying view on MTP methods

| Group of methods | Applicable setting |
|---|---|
| Independent models | B |
| **Similarity-enforcing methods** | B |
| Relation-exploiting methods | B and D |
| Relation-constructing methods | B |
| Representation-exploiting methods | B and D |
| Representation-constructing methods | A and B |

# The basic deep architecture for multi-target prediction




Joint feature learning $\phi(\boldsymbol{x})$ among targets[1][2]:

$$f_j(\boldsymbol{x}_i) = \boldsymbol{a}_j^\intercal \phi_{\boldsymbol{\theta}}(\boldsymbol{x}_i) \qquad p_j(\boldsymbol{x}_i) = P(y_{ij} = 1 \mid \boldsymbol{x}_i) = \frac{\exp(-f_j(\boldsymbol{x}_i))}{1 + \exp(-f_j(\boldsymbol{x})_i)}$$

Typical loss function for regression and classification:

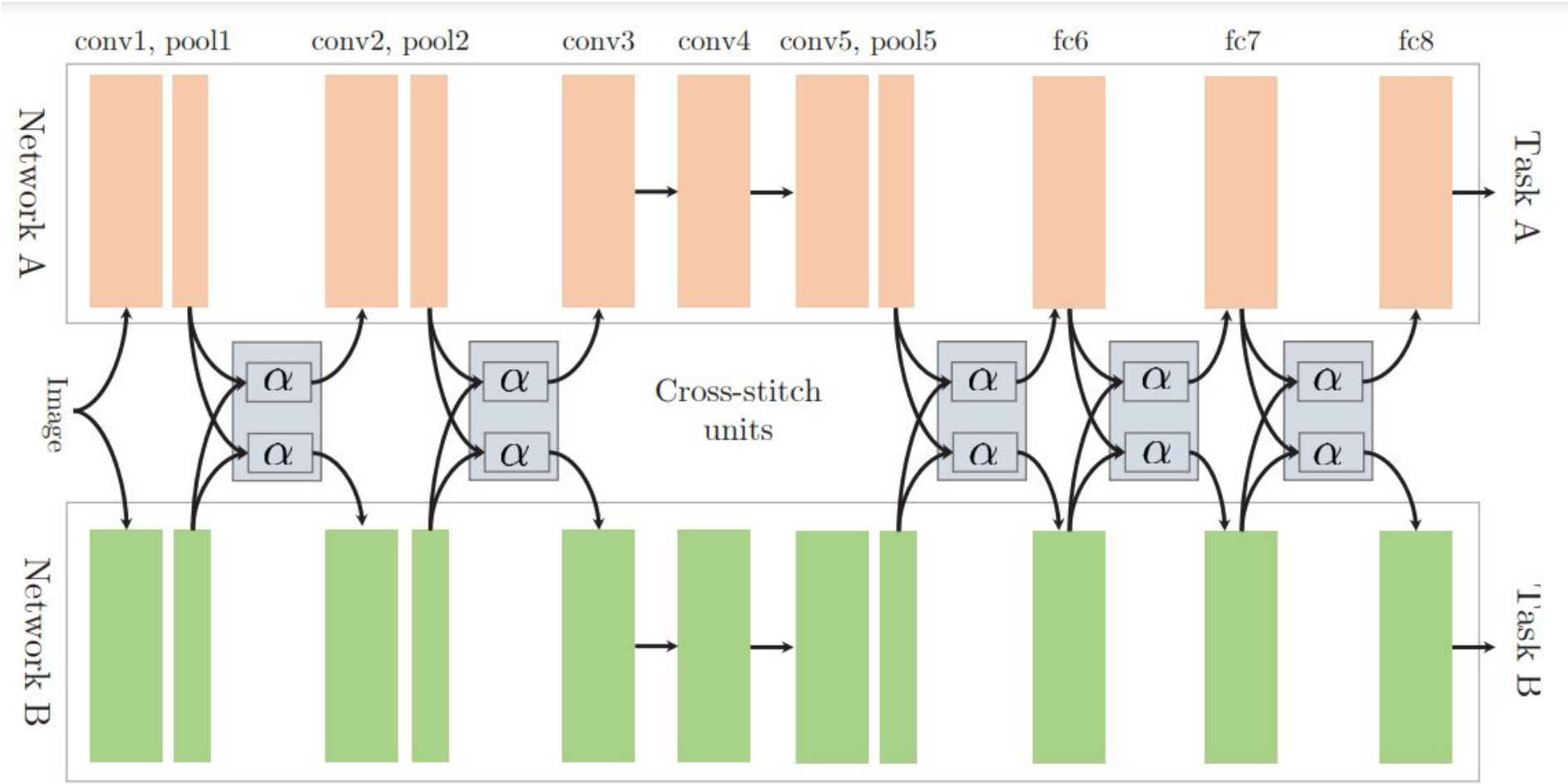
$$\min_{\boldsymbol{a}_1,\ldots,\boldsymbol{a}_m,\boldsymbol{\theta}} \sum_{i=1}^{n}\sum_{j=1}^{m}(y_{ij} - f_j(\boldsymbol{x}_i))^2 \qquad \min_{\boldsymbol{a}_1,\ldots,\boldsymbol{a}_m,\boldsymbol{\theta}} \sum_{i=1}^{n}\sum_{j=1}^{m} CE(y_{ij}, p_j(\boldsymbol{x}_i))$$

[1] First presented in: Caruana, Multitask learning: A knowledge-based source of inductive bias. Machine Learning 1997

[2] Figure from: Zhang et al. Facial landmark detection by deep multi-task learning, ECCV 2014

# Tailor-made architectures, e.g. cross-stitch netowrks

Learn a combination of target-specific and shared representations[3]:




[3] Figure from Misra et al. Cross-stitch networks for multi-task learning, CVPR 2016

# Neural architecture search for multi-task learning [4]

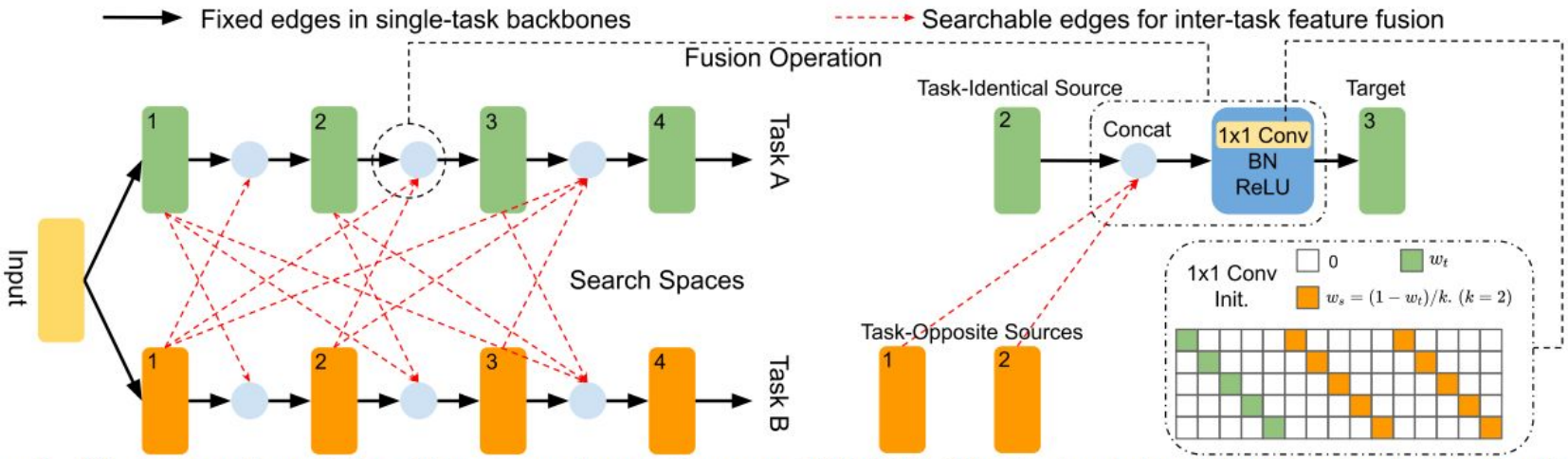
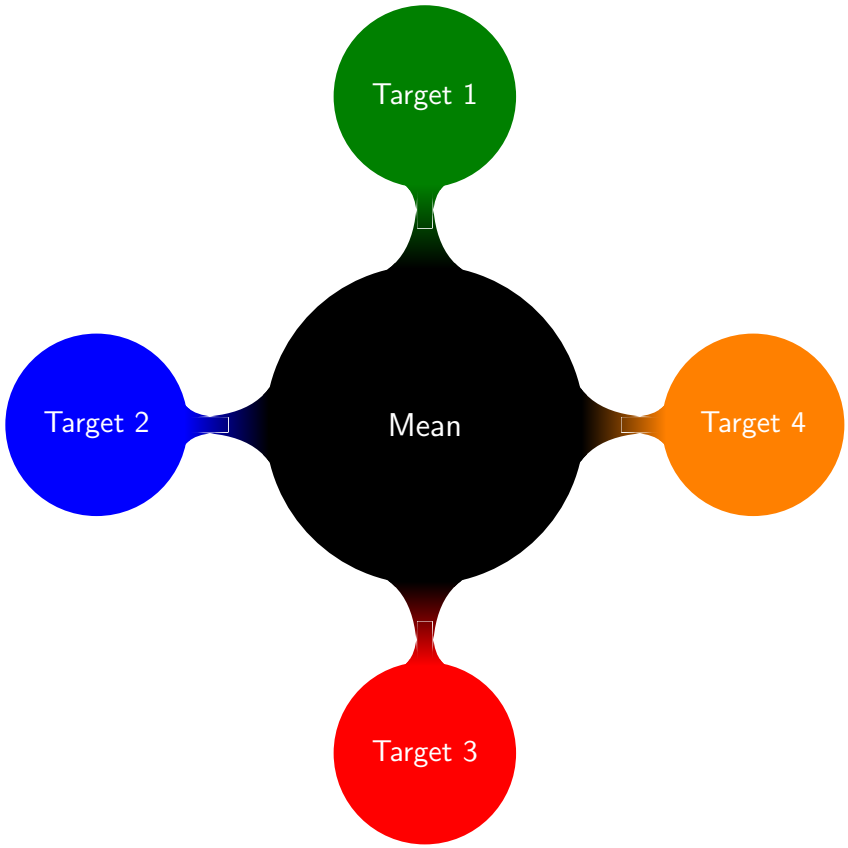



Figure 1. The problem formulation of the proposed general-purpose MTL-NAS. We disentangle the GP-MTL networks into fixed task-specific single-task backbones and general feature fusion schemes between them. This allows us to define a general task-agnostic search space compatible with any task combinations, as shown in the leftmost subfigure. The right-top subfigure illustrates the inter-task fusion operation, which is motivated by and extends from the NDDR-CNN [13]. We show the initialization of the fusion operation in the right-bottom subfigure. As we are inserting new edges between the fixed and well-trained single-task network backbones, we wish to make a minimal impact on the original output at each layer at initialization (*i.e.*, initializing with a large $w_t$) (best viewed in color).

[4] Gao et al. Task-agnostic neural architecture search towards general-purpose multi-task learning, CVPR 2020

# Regularization terms for multi-target prediction[5]

- Simple assumption: models for different targets are related to each other.
- Simple solution: the parameters of these models should have similar values.
- Approach: bias the parameter vectors towards their mean vector.




$$\min_{\boldsymbol{a}_1,\ldots,\boldsymbol{a}_m,\boldsymbol{\theta}} \sum_{i=1}^{n}\sum_{j=1}^{m}(y_{ij} - f_j(\boldsymbol{x}_i))^2 + \lambda \sum_{j=1}^{m} ||\boldsymbol{a}_j - \frac{1}{m}\sum_{l=1}^{m}\boldsymbol{a}_l||^2 ,$$

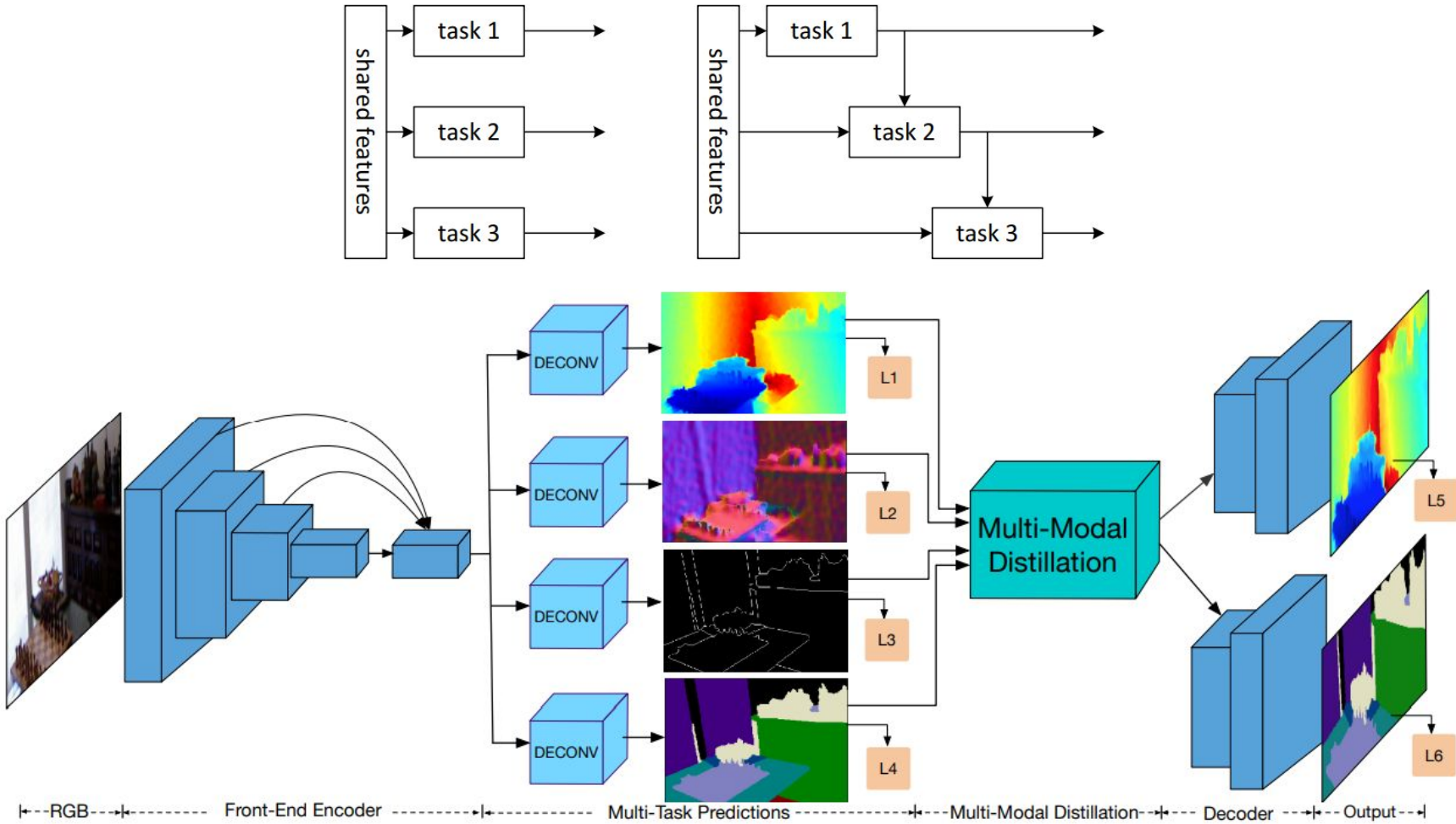
[5] Evgeniou and Pontil, Regularized multi–task learning, KDD 2004.

# A unifying view on MTP methods

| Group of methods | Applicable setting |
|---|---|
| Independent models | B |
| Similarity-enforcing methods | B |
| **Relation-exploiting methods** | B and D |
| Relation-constructing methods | B |
| Representation-exploiting methods | B and D |
| Representation-constructing methods | A and B |

# Encoding target relationships in deep architectures[6][7]

## Usually tailormade architectures for specific applications!

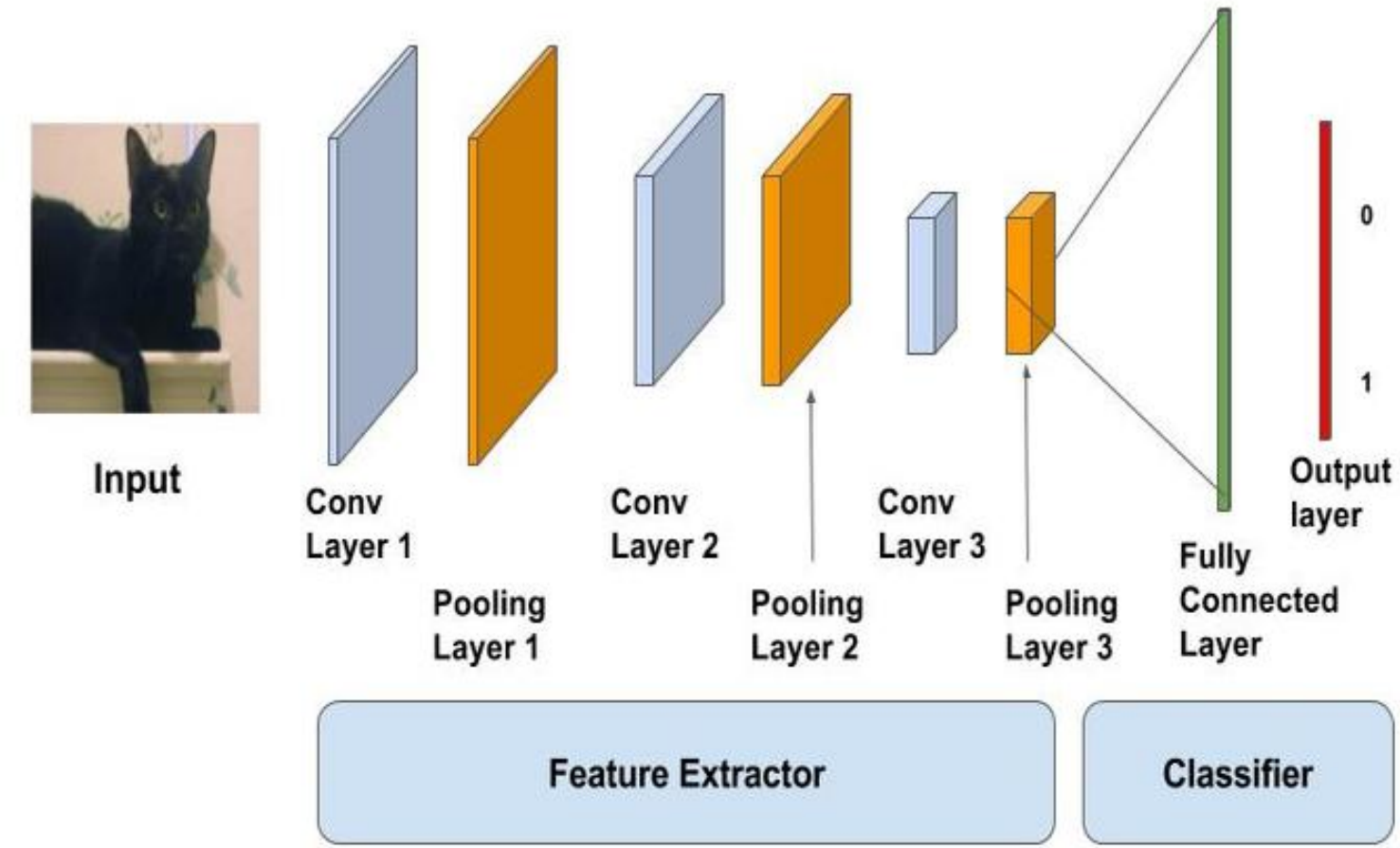



[6] Dai et al., Instance-aware Semantic Segmentation via Multi-task Network Cascades, CVPR 2016

[7] Xu et al., PAD-Net: Multi-Tasks Guided Prediction-and-Distillation Network for Simultaneous Depth Estimation and Scene Parsing, CVPR 2018

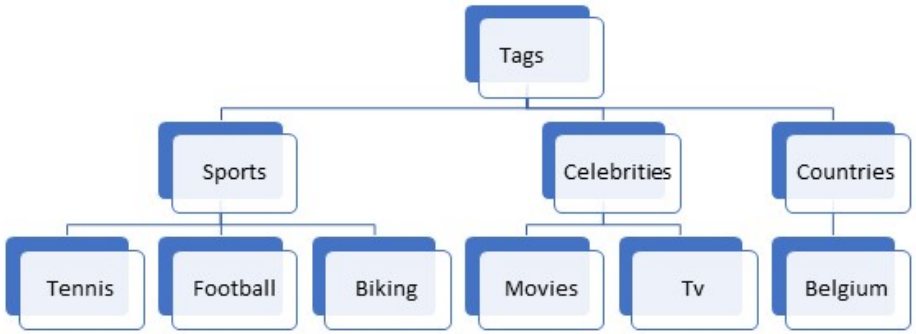
# Re-using Pretrained Models in (Deep) Neural Networks

Commonly-used training method: first train on targets that have a lot of observations, only train some parameters for targets that have few observations [8]




[8] Keras Tutorial: Transfer Learning using pre-trained models

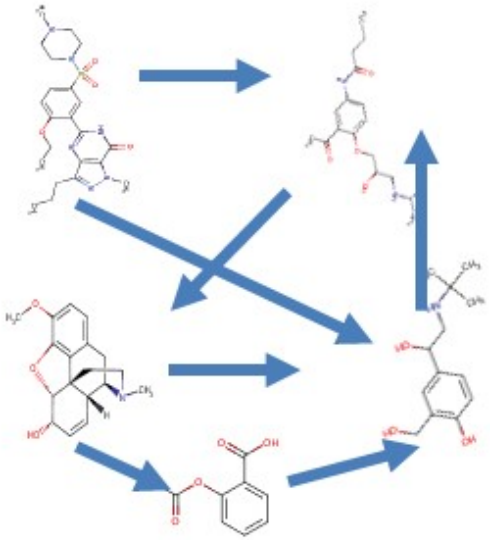
# An example from the introduction revisited



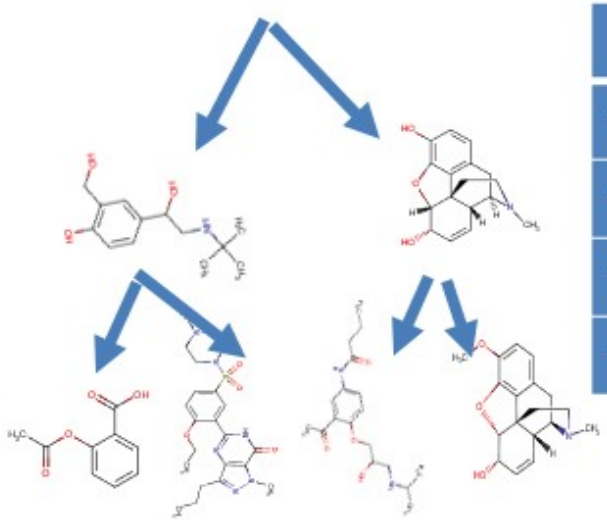

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01101 | **Text1** | 0 | 0 | 0 | 0 | 0 | 1 |
| 00111 | **Text2** | 0 | 0 | 1 | 0 | 1 | 1 |
| 01110 | **Text3** | 0 | 0 | 0 | 1 | 1 | 0 |
| 10001 | **Text4** | 0 | 0 | 1 | 0 | 1 | 0 |
| 01011 | **Text5** | 1 | 0 | 0 | 1 | 0 | 0 |
| 11110 | **Text6** | ? | ? | ? | ? | ? | ? |

# Exploiting relations in regularization terms

Graph



Tree



Similarity

| | T 1 | T 2 | T 3 | T 4 |
|---|---|---|---|---|
| T 1 | 1 | 0.1 | 0.4 | 0.0 |
| T 2 | 0.1 | 1 | 0.6 | 0.7 |
| T 3 | 0.4 | 0.6 | 1 | 0.4 |
| T 4 | 0.0 | 0.7 | 0.4 | 1 |

Graph-based regularization is an approach that can be applied to various types of relations[9]:

$$\min_{\boldsymbol{a}_1,\ldots,\boldsymbol{a}_m,\boldsymbol{\theta}} \sum_{i=1}^{n}\sum_{j=1}^{m}(y_{ij} - f_j(\boldsymbol{x}_i))^2 + \lambda \sum_{j=1}^{m}\sum_{l\in\mathcal{N}(j)} ||\boldsymbol{a}_j - \boldsymbol{a}_l||^2$$

[9] Gopal and Yang, Recursive regularization for large-scale classification with hierarchical and graphical dependencies, KDD 2013

# What does it mean for targets to be "related"?

We distinguish between conditional and unconditional (in)dependence of targets[10].

- Unconditional/marginal dependence:

$$P(\boldsymbol{y}) \neq \prod_{j=1}^{m} P(y_i)$$

  Often due to model similarities, i.e., $y_{ij} = f_j(\boldsymbol{x}_i) + \epsilon_{ij}$ for $j = 1, \ldots, m$, with similarities in the structural parts $f_j(\cdot)$, which implies correlation between targets.

- Conditional dependence:

$$P(\boldsymbol{y} \,|\, \boldsymbol{x}) \neq \prod_{j=1}^{m} P(y_j \,|\, \boldsymbol{x})$$

[10] Dembczynski et al., On Label Dependence and Loss Minimization in Multi-Label Classification. Machine Learning, 88, 2012.

# Marginal (in)dependence $\not\Leftrightarrow$ conditional (in)dependence

- Example:

| $x_1$ | $y_1$ | $y_2$ | $P$ | $x_1$ | $y_1$ | $y_2$ | $P$ |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0.25 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0.25 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0.25 |
| 0 | 1 | 1 | 0.25 | 1 | 1 | 1 | 0 |

- Strong conditional dependence, for example $P(y_1 = 0|x_1 = 1)P(y_2 = 0|x_1 = 1) = 0.5 \times 0.5 = 0.25 \neq 0.$
- Yet, labels are marginally independent: Joint probability is the product of the marginals $P(y_1 = 0) = P(y_2 = 0) = 0.5.$
- Domain knowledge w.r.t. marginal dependence is often available
- Conversely, domain knowledge w.r.t. conditional dependence is almost never available, and needs to be learned from data

# A unifying view on MTP methods

| Group of methods | Applicable setting |
|---|---|
| Independent models | B |
| Similarity-enforcing methods | B |
| Relation-exploiting methods | B and D |
| **Relation-constructing methods** | B |
| Representation-exploiting methods | B and D |
| Representation-constructing methods | A and B |

# Conditional dependence in multi-label classification

- For a feature vector $\boldsymbol{x}$, predict a vector of responses $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)$ using a function/hypothesis $\boldsymbol{h}$:

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_p) \quad \xrightarrow{\boldsymbol{h}(\boldsymbol{x})} \quad \hat{\boldsymbol{y}} = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_m)$$

- In multi-label classification, a broad spectrum of multi-label loss functions

$$\ell : \{0,1\}^m \times \{0,1\}^m \to \mathbb{R}$$

  is conceivable.

- Problem: Given a target loss $\ell$, find a (Bayes) predictor $\boldsymbol{h}$ that minimizes expected loss with regard to $\ell$.

$$\boldsymbol{h}^*(\boldsymbol{x}) = \arg \min_{\hat{\boldsymbol{y}} \in \{0,1\}^m} \sum_{\boldsymbol{y} \in \{0,1\}^m} L(\boldsymbol{y}, \hat{\boldsymbol{y}}) P(\boldsymbol{y} \mid \boldsymbol{x})$$

# Two simple yet extreme multi-label losses

$$\boldsymbol{h}^*(\boldsymbol{x}) = \arg \min_{\hat{\boldsymbol{y}} \in \{0,1\}^m} \sum_{\boldsymbol{y} \in \{0,1\}^m} L(\boldsymbol{y}, \hat{\boldsymbol{y}}) P(\boldsymbol{y} \mid \boldsymbol{x})$$

- Key question: Can we achieve this goal through simple reduction, i.e., by training one model for each target independently? Or can we do better with more sophisticated methods?
- The Hamming loss averages over mistakes on individual labels:

$$\ell_H(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{m} \sum_{i=1}^{m} [\![ y_i \neq \hat{y}_i ]\!]$$

- The subset 0/1 loss simply checks for entire correctness:

$$\ell_{0/1}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = [\![ \boldsymbol{y} \neq \hat{\boldsymbol{y}} ]\!] = \max_i [\![ y_i \neq \hat{y}_i ]\!]$$

# Hamming vs. subset 0/1 loss

- The risk minimizer for the Hamming loss is the marginal mode:

$$h_i^*(\boldsymbol{x}) = \arg \max_{y_i \in \{0,1\}} P(y_i \,|\, \boldsymbol{x}) \,, \quad i = 1, \ldots, m,$$

  while for the subset 0/1 loss it is the joint mode[11]:

$$\mathbf{h}^*(\boldsymbol{x}) = \arg \max_{\boldsymbol{y} \in \mathcal{Y}} P(\boldsymbol{y} \,|\, \boldsymbol{x}) \,.$$

- Marginal mode vs. joint mode.

| $\boldsymbol{y}$ | $P(\boldsymbol{y} \,\vert\, \boldsymbol{x})$ |
|---|---|
| 0 0 0 0 | 0.30 |
| 0 1 1 1 | 0.17 |
| 1 0 1 1 | 0.18 |
| 1 1 0 1 | 0.17 |
| 1 1 1 0 | 0.18 |

Marginal mode: 1 1 1 1
Joint mode: 0 0 0 0

[11] Dembczynski et al. On label dependence and loss minimization in multi-label classification, Machine Learning 2012

# Probabilistic classifier chains[12]

- Estimate the joint conditional distribution $P(\boldsymbol{y} \,|\, \boldsymbol{x})$.
- For optimizing the subset 0/1 loss:

$$\ell_{0/1}(\boldsymbol{y}, \hat{y}) = [\![\boldsymbol{y} \neq \hat{y}]\!]$$

- Repeatedly apply the product rule of probability:

$$P(\boldsymbol{y} \,|\, \boldsymbol{x}) = \prod_{i=1}^{m} P(y_i \,|\, \boldsymbol{x}, y_1, \ldots, y_{i-1}) \,.$$

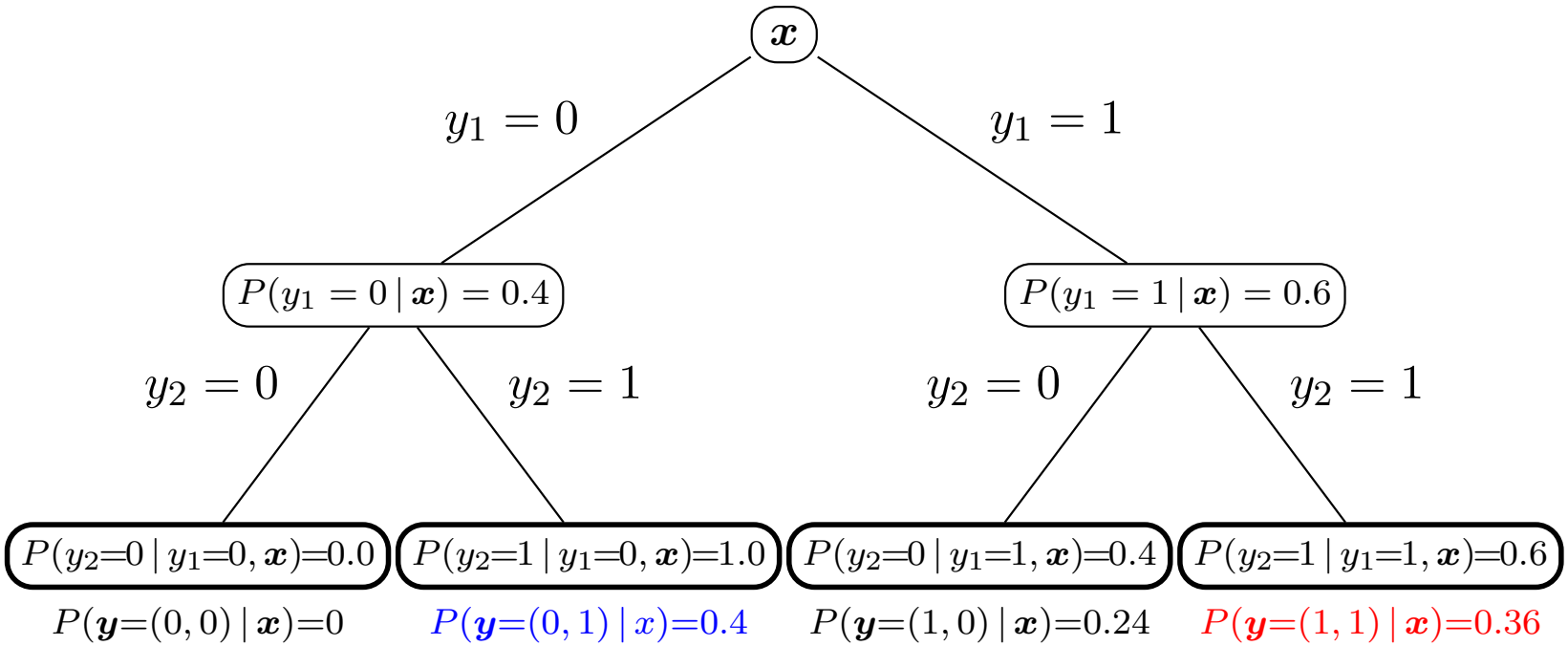
- Learning relies on constructing probabilistic classifiers for estimating

$$P(y_i \,|\, \boldsymbol{x}, y_1, \ldots, y_{i-1}) \,,$$

  independently for each $i = 1, \ldots, m$.

[12] Dembczysnki et al. Bayes-optimal multi-label classification with probabilistic classifier chains, ICML 2010.

- Inference relies on exploiting a probability tree[13] [14]:




- $(0, 1)$ is the joint mode and the minimizer of the subset zero-one loss
- Can be found with specific inference algorithms
- $(1, 1)$ is the minimizer of the Hamming loss
- Compute the minimizer on a sample from $P(\boldsymbol{y} \mid \boldsymbol{x})$

---

[13] Dembczynski et al., An analysis of chaining in multi-label classification, ECAI 2012

[14] Mena et al. A family of admissible heuristics for A* to perform inference in probabilistic classifier chains, Machine Learning 2017.

# Deep learning methods that estimate $P(\boldsymbol{y} \mid \boldsymbol{x})$

- Use PCC with a neural network as base learner
- Use a recurrent neural network to reduce the length of the chain in PCC by only predicting positive labels[15]
- Alternative methods to estimate $P(\boldsymbol{y} \mid \boldsymbol{x})$, e.g. conditional random fields and their deep extensions[16]
- Instead of estimating $P(\boldsymbol{y} \mid \boldsymbol{x})$, consider energy-based models[17]:

$$\boldsymbol{h}(\boldsymbol{x}) = \arg \min_{\boldsymbol{y} \in \{0,1\}^m} E(\boldsymbol{y}, \boldsymbol{x})$$

- For certain loss functions, such as the instance-wise F-measure, it suffices to estimate specific properties of $P(\boldsymbol{y} \mid \boldsymbol{x})$ [18] [19]

---

[15] Nam et al. Maximizing Subset Accuracy with Recurrent Neural Networks in Multi-label Classification, Neurips 2017

[16] Zheng et al. Conditional Random Fields as Recurrent Neural Networks, ICCV2015

[17] Bellanger and MacCallum, Structured Prediction Energy Networks, ICML 2016

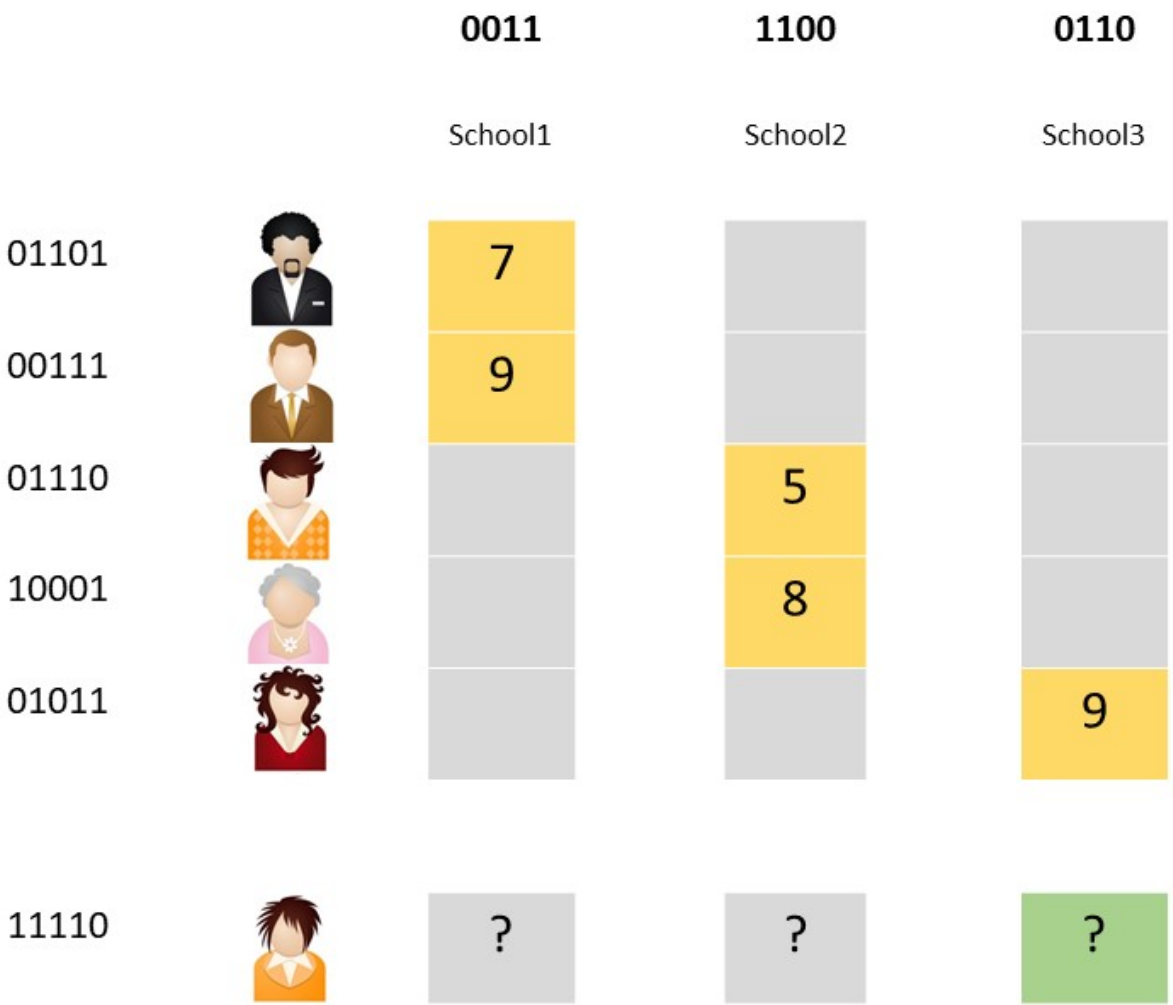
[18] Waegeman et al. On the Bayes-optimality of F-measure maximizers, JMLR 2014

[19] Decubber et al., Deep F-measure maximization in multi-label classification:a comparative study, ECML/PKDD 2018.

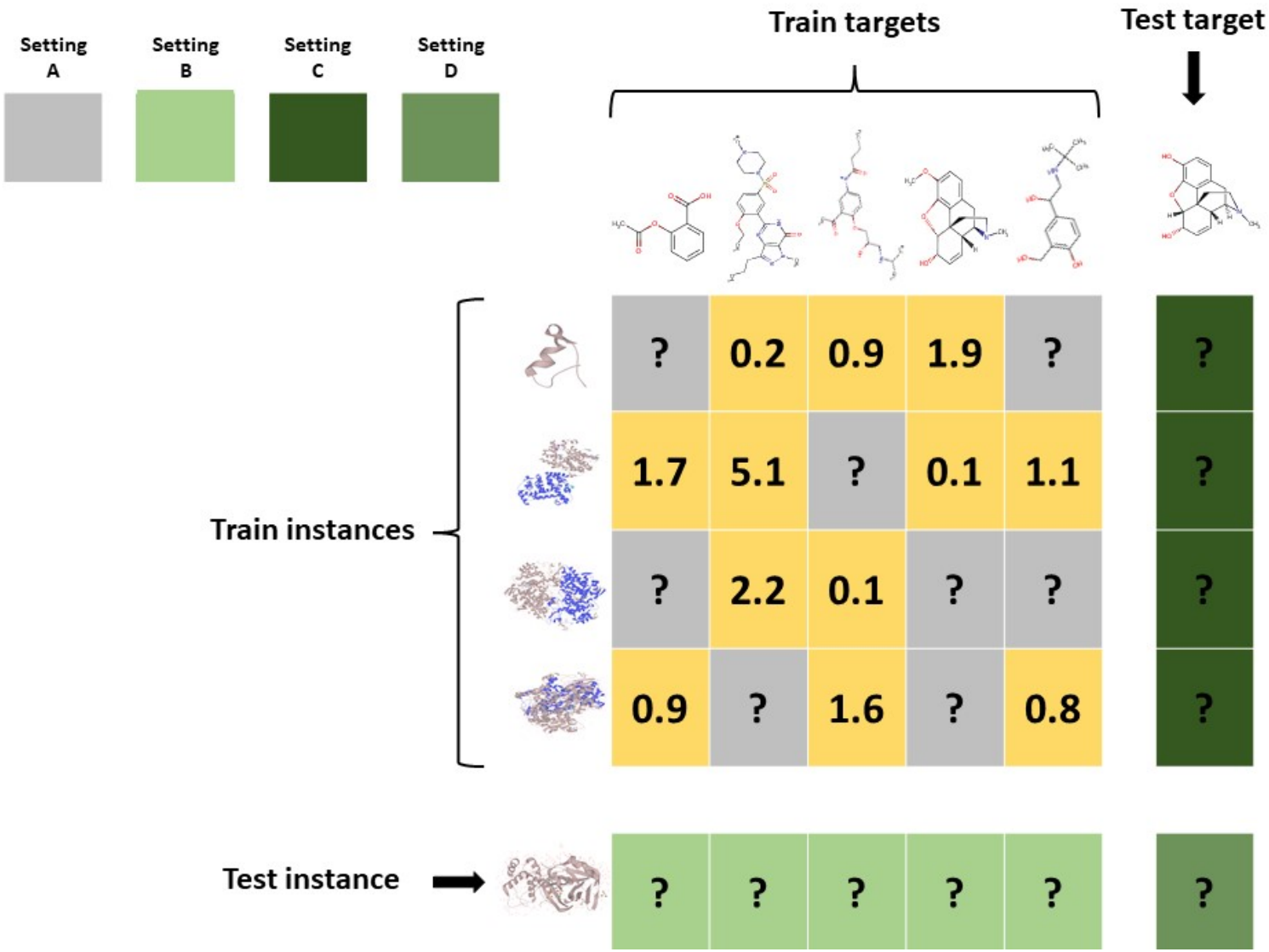
# A unifying view on MTP methods

| Group of methods | Applicable setting |
|---|---|
| Independent models | B |
| Similarity-enforcing methods | B |
| Relation-exploiting methods | B and D |
| Relation-constructing methods | B |
| **Representation-exploiting methods** | B and D |
| Representation-constructing methods | A and B |

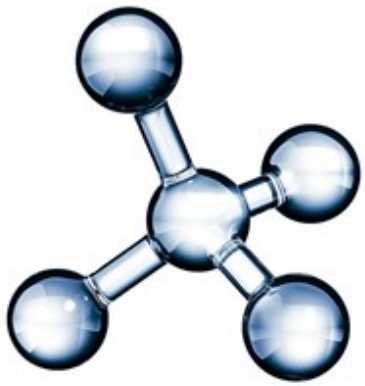
# An example revisited

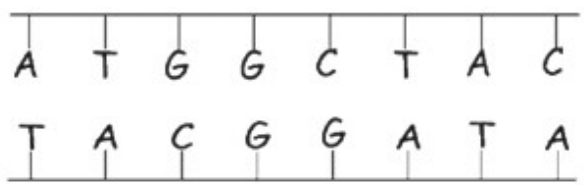


# Another example revisited



# Target representations can take many forms

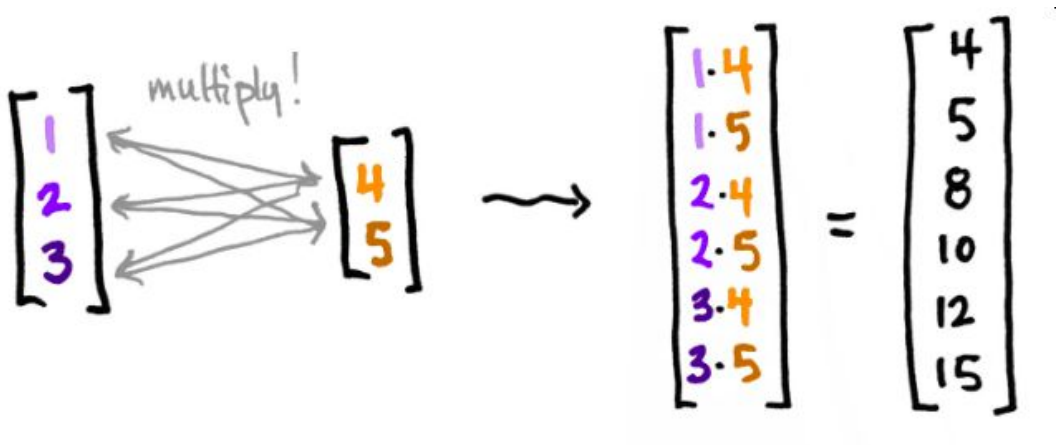






# Traditional approach: Kronecker kernel ridge regression

Tensor product model representation in the primal formulation:

$$f(\boldsymbol{x}, \boldsymbol{t}) = \boldsymbol{w}^T \left(\phi(\boldsymbol{x}) \otimes \psi(\boldsymbol{t})\right)$$

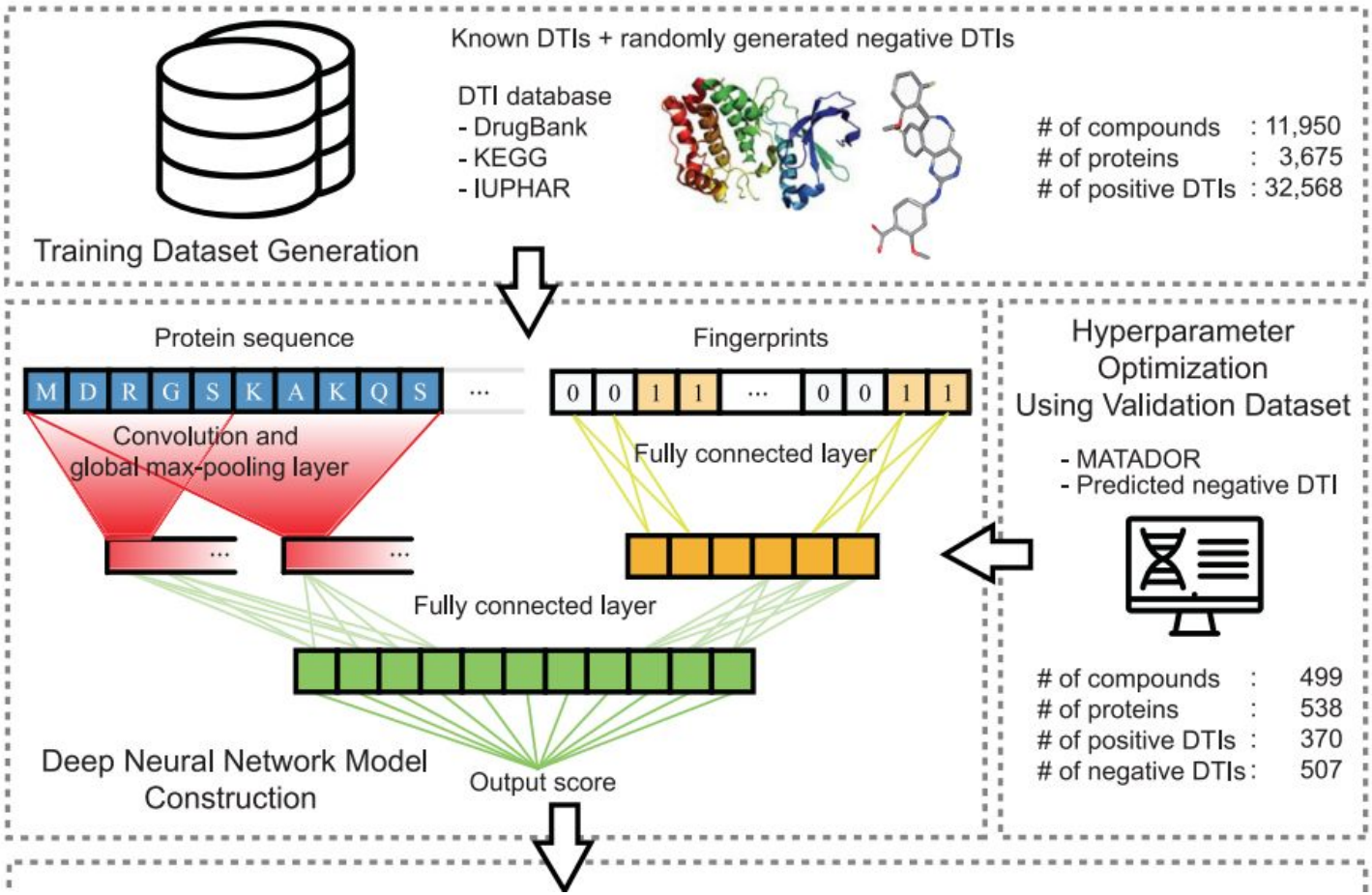



Kronecker product pairwise kernel in the dual formulation[20]:

$$f(\boldsymbol{x}, \boldsymbol{t}) = \sum_{(\bar{\boldsymbol{x}}, \bar{\boldsymbol{t}}) \in \mathcal{D}} \alpha_{(\bar{\boldsymbol{x}}, \bar{\boldsymbol{t}})} k(\boldsymbol{x}, \bar{\boldsymbol{x}}) \cdot g(\boldsymbol{t}, \bar{\boldsymbol{t}}) = \sum_{(\bar{\boldsymbol{x}}, \bar{\boldsymbol{t}}) \in \mathcal{D}} \alpha_{(\bar{\boldsymbol{x}}, \bar{\boldsymbol{t}})} \Gamma((\boldsymbol{x}, \boldsymbol{t}), (\bar{\boldsymbol{x}}, \bar{\boldsymbol{t}}))$$

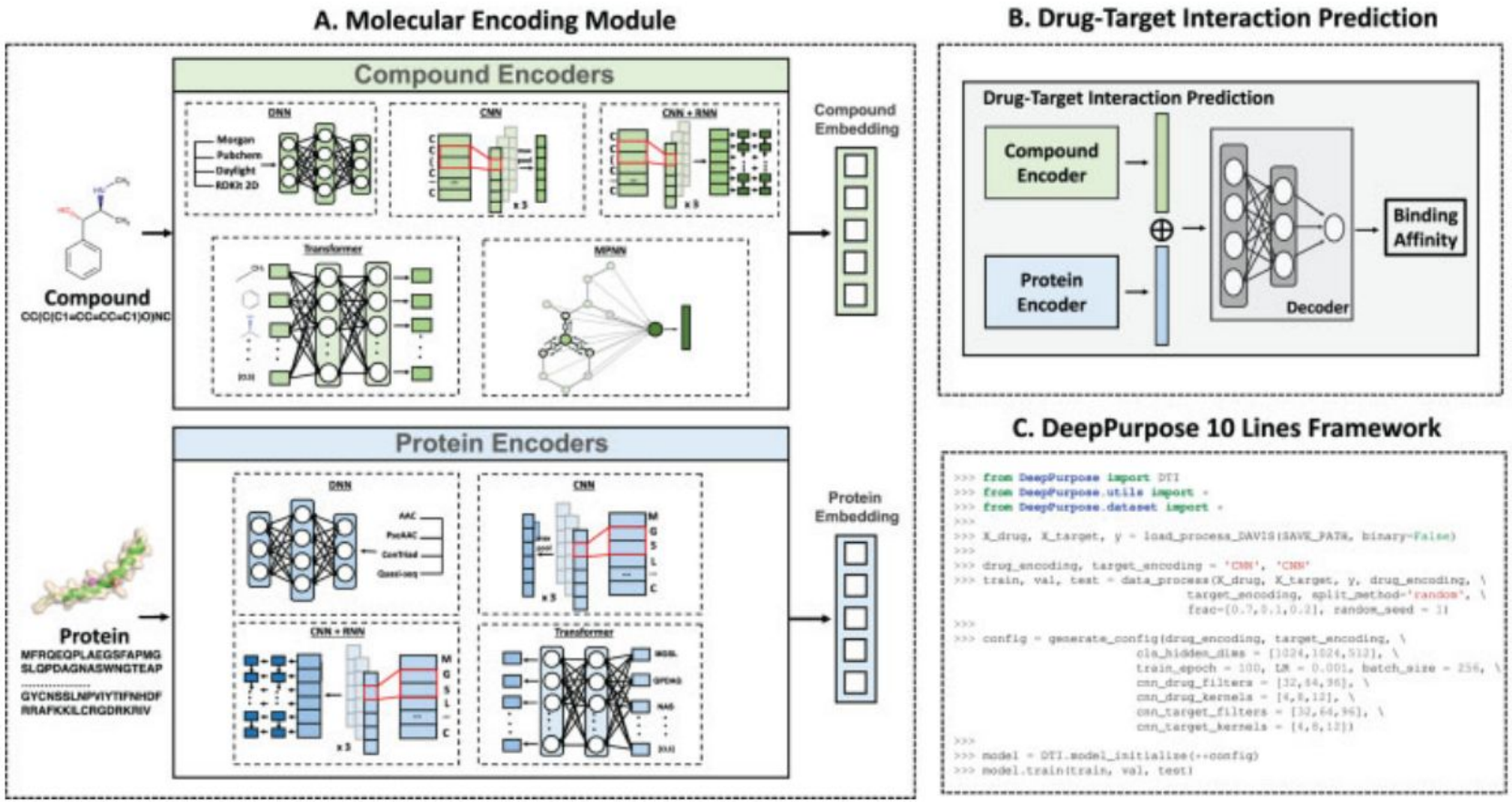
[20] Stock et al., A comparative study of pairwise learning methods based on kernel ridge regression, Neural Computation 2018
Figure taken from https://www.math3ma.com/blog/the-tensor-product-demystified

# Pairwise model representations in neural networks[21]



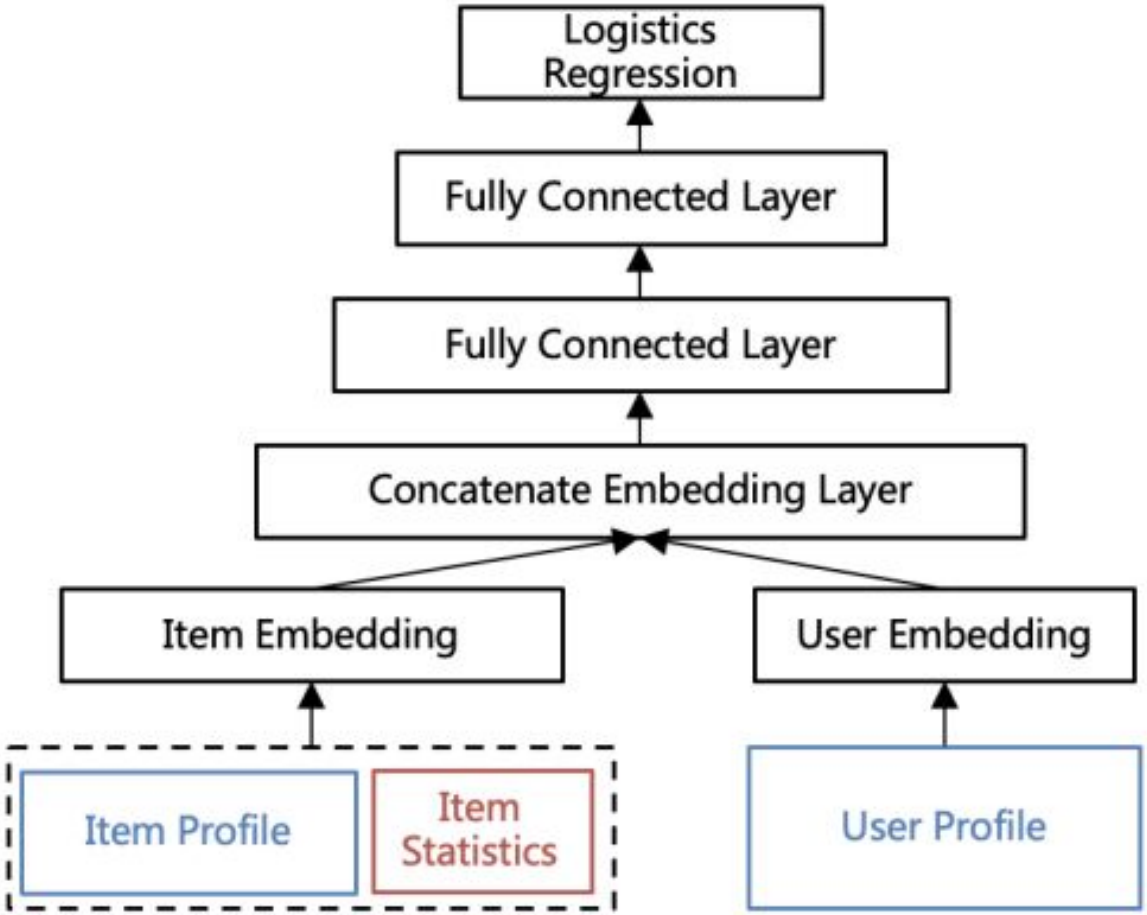

[21] Lee et al. DeepConv-DTI: Prediction of drug-target interactions via deep learning with convolution on protein sequences, PLOS Computational Biology, 2018.

# Pairwise model representations in neural networks[22]



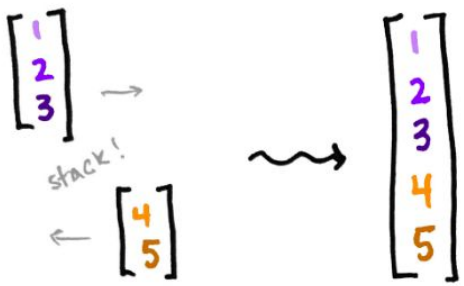

[22] Huang et al. DeepPurpose: a deep learning library for drug–target interaction prediction, Bioinformatics 2020.

# Pairwise model representations in neural networks[23]



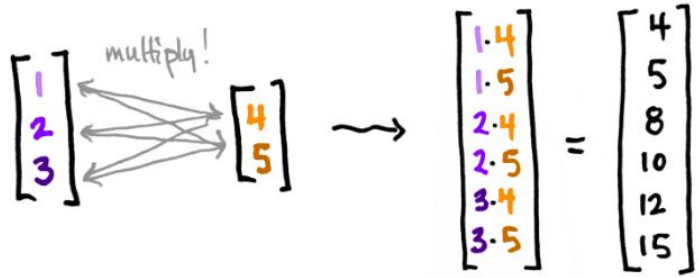

[23] Xin et al. ATNN: Adversarial Two-Tower Neural Network for New Item's Popularity Prediction in E-commerce.

# Differences between direct sum and tensor product




$(\vec{v}, \vec{w})$ in $\mathbb{R}^3 \oplus \mathbb{R}^2$

direct sum



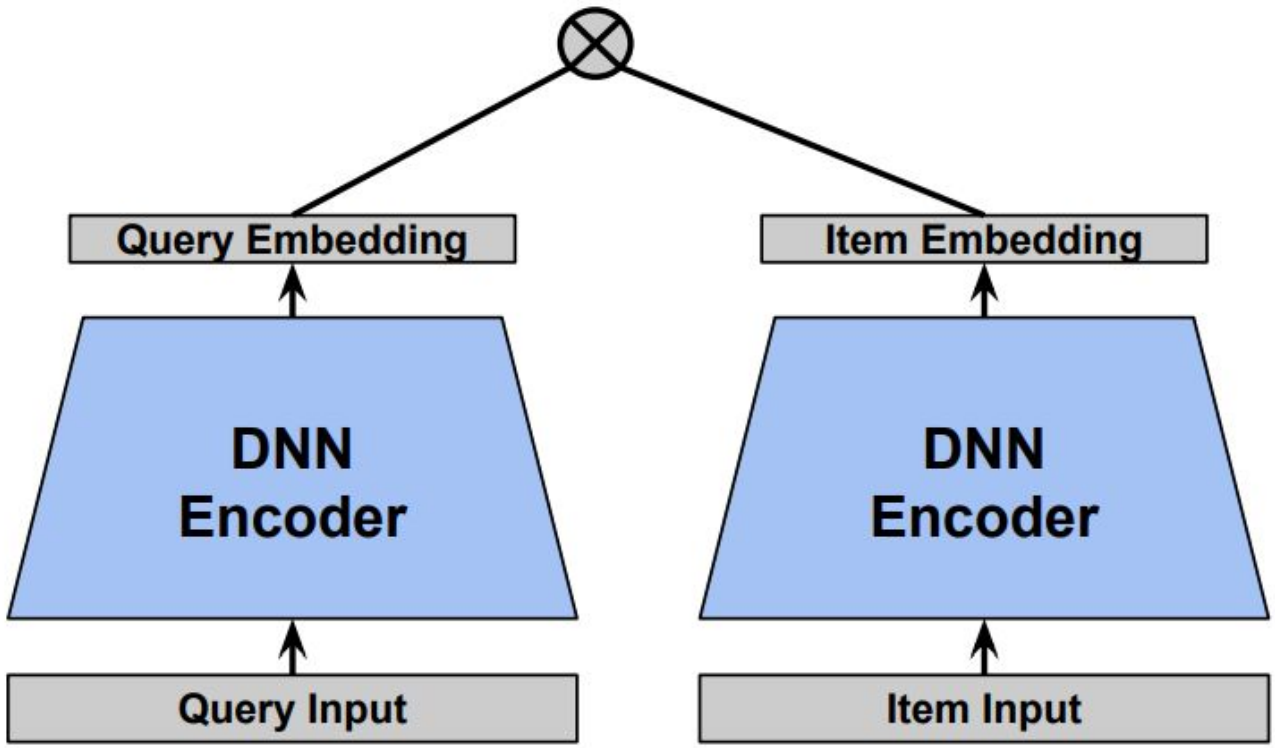

$\vec{v} \otimes \vec{w}$ in $\mathbb{R}^3 \otimes \mathbb{R}^2$

tensor product

- In kernel methods, the tensor product is needed to guarantee universality[24]
- In neural networks, the direct sum can mimic the tensor product when an MLP is used to learn interactions
- However, this comes at the cost of more model parameters, so more training data is needed

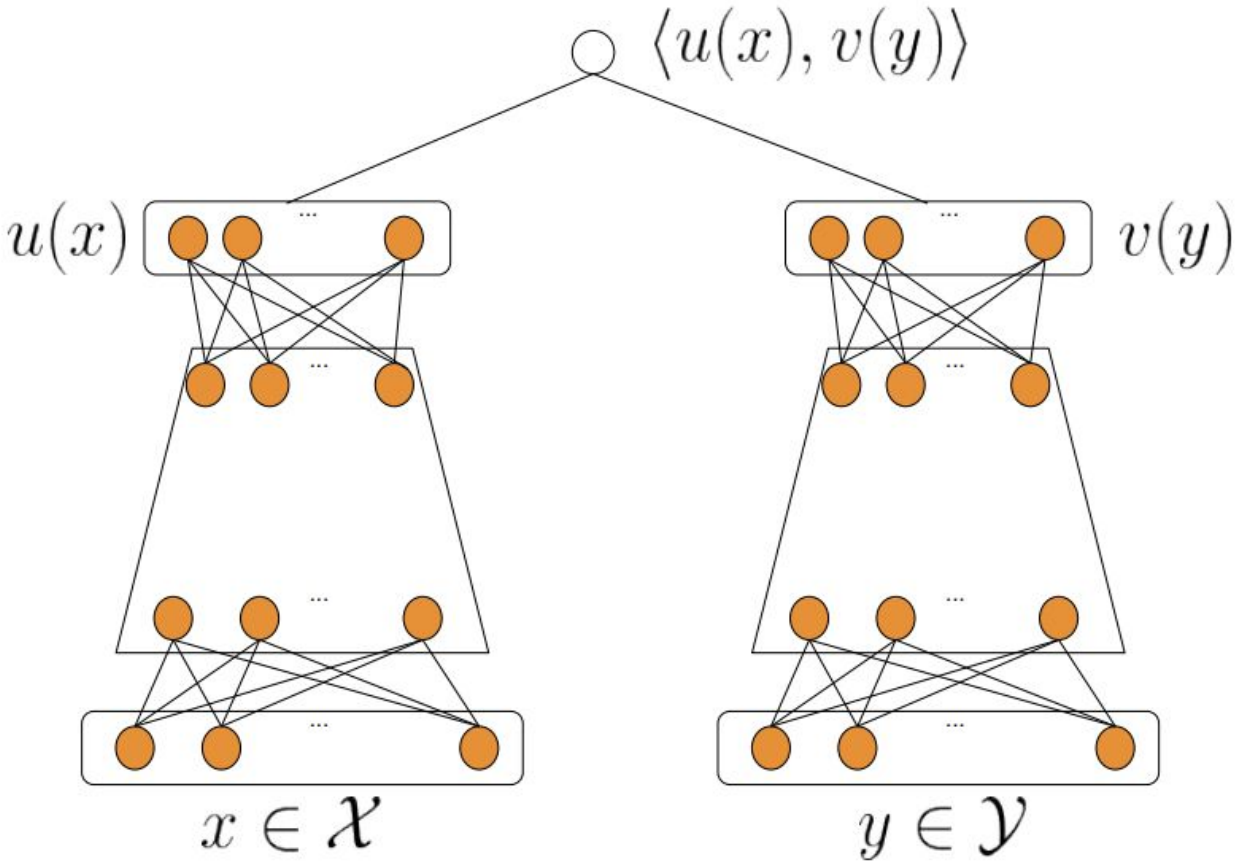
[24] Waegeman et al. A kernel-based framework to learn graded relations from data, IEEE Transactions on Fyzzy Systems, 2012. Picture taken from https://www.math3ma.com/blog/the-tensor-product-demystified

# Pairwise model representations in neural networks[25]



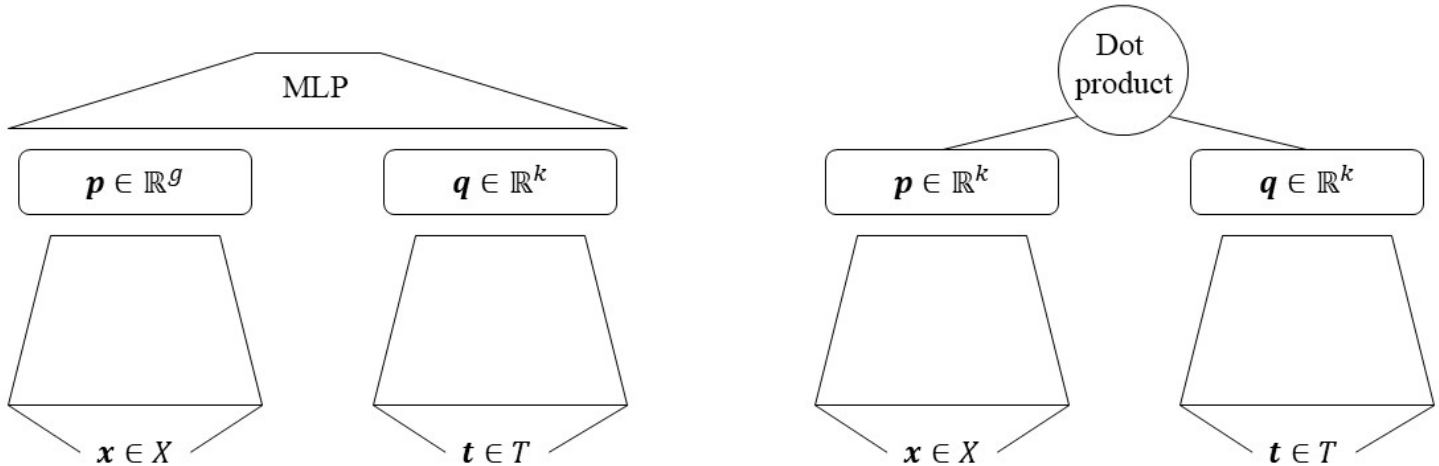

[25] Yang et al. Mixed Negative Sampling for Learning Two-tower Neural Networks in Recommendations, WWW 2020.

# Pairwise model representations in neural networks[26]




[26] Yi et al. Sampling-Bias-Corrected Neural Modeling for Large Corpus Item Recommendations, RecSys 2019.

# Differences between direct sum and dot product




- Compared to the direct sum plus MLP, the dot product looks much more restrictive because it has less parameters
- The direct sum allows for different dimensions for instance and target embeddings
- In the area of recommender systems, the two methods give comparable empirical results[27]

[27] Rendle et al., Neural Collaborative Filtering vs. Matrix Factorization Revisited, RecSys 2020.

# A unifying view on MTP methods

| Group of methods | Applicable setting |
|---|---|
| Independent models | B |
| Similarity-enforcing methods | B |
| Relation-exploiting methods | B and D |
| Relation-constructing methods | B |
| Representation-exploiting methods | B and D |
| **Representation-constructing methods** | A and B |

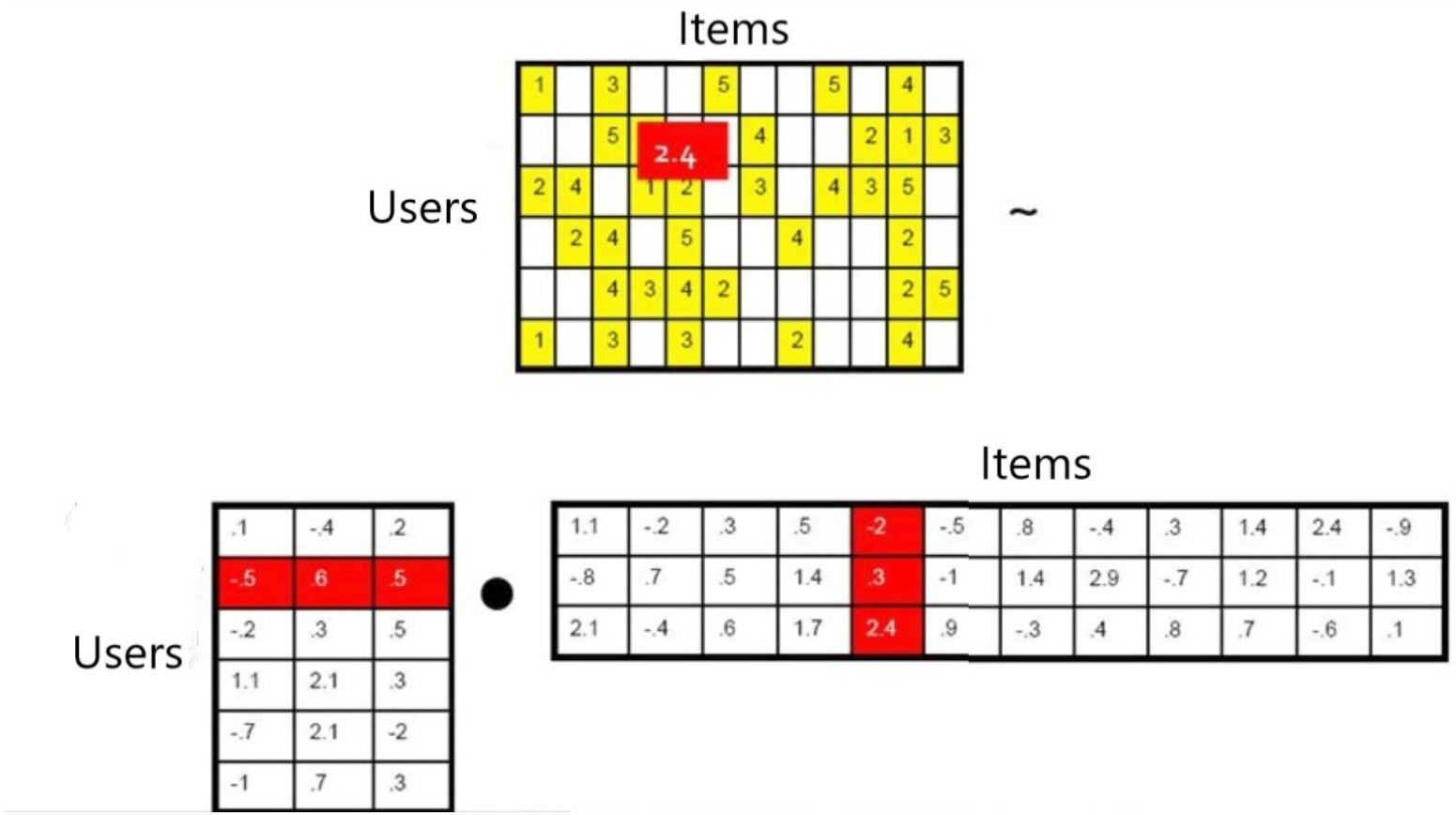
Setting A:
Matrix completion without or with side information


Setting B:
Side information for instances is required

# Setting A (Matrix completion without side information)

Traditional approach: Factorize the matrix $Y$ into two smaller matrices[28]:

$$Y \quad = \quad U \quad \times \quad V$$




[28] See e.g. Jain et al., Low-rank matrix completion using alternating minimization, ACM Symposium on Theory of Computing 2013

# Setting A (Matrix completion with side information)


- Construct implicit features $(\boldsymbol{x}^I, \boldsymbol{t}^I)$ for users and items with matrix factorization methods
- Exploit explicit features $(\boldsymbol{x}^E, \boldsymbol{t}^E)$ (a.k.a. side information)
- Concatenate:

$$\boldsymbol{x}^C = \boldsymbol{x}^I \oplus \boldsymbol{x}^E, \qquad \boldsymbol{t}^C = (\boldsymbol{t}^I \oplus \boldsymbol{t}^E)$$

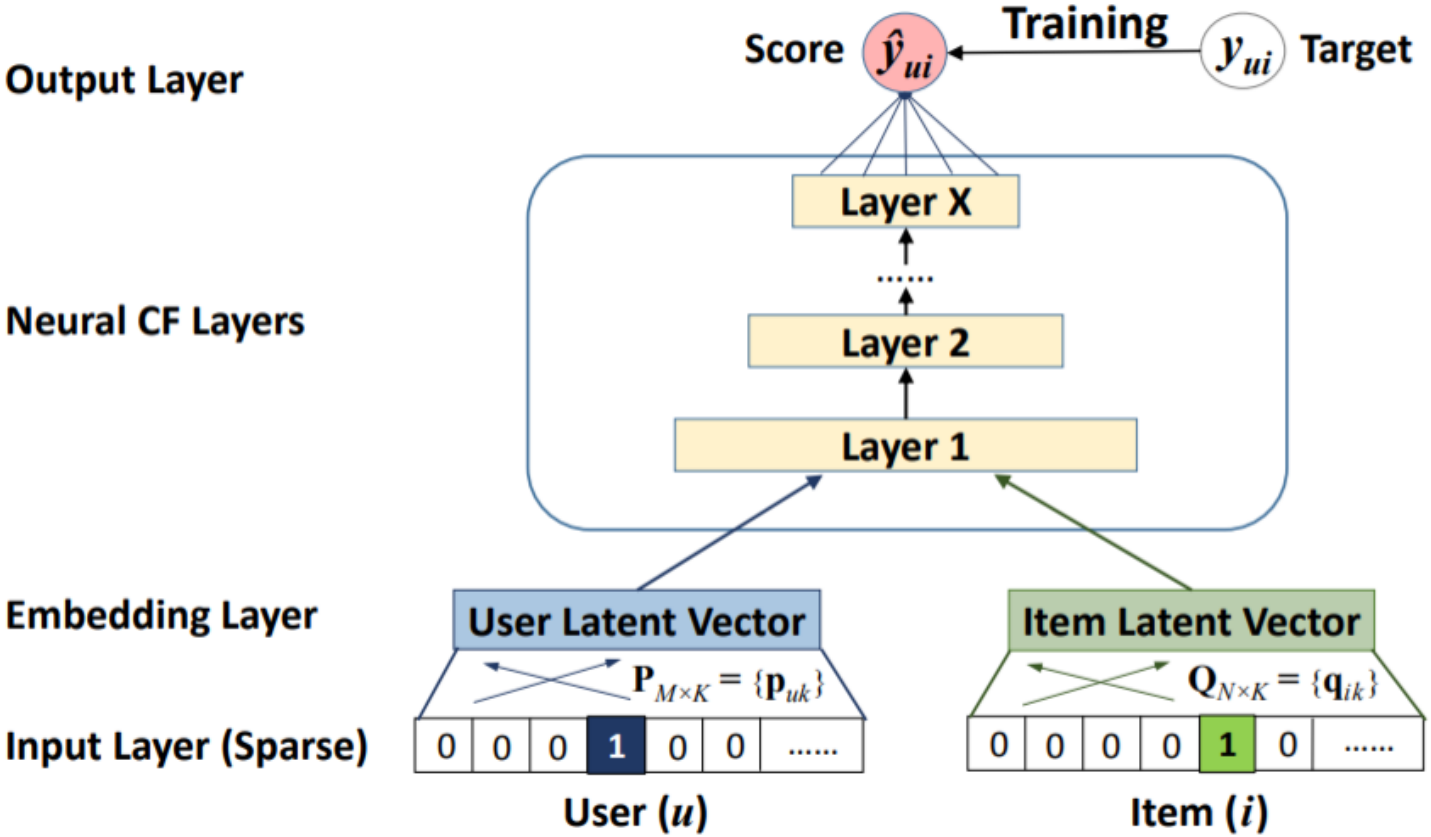
- Apply methods that we have seen before[29][30]:

$$f(\boldsymbol{x}^C, \boldsymbol{t}^C) = \boldsymbol{w}^T \big(\phi(\boldsymbol{x}^C) \otimes \psi(\boldsymbol{t}^C)\big)$$

[29] Menon and Elkan, A log-linear model with latent features for dyadic prediction, ICDM 2010

[30] Volkovs and Zemel, Collaborative filtering with 17 parameters, NIPS 2012

# Matrix completion with neural networks[31]

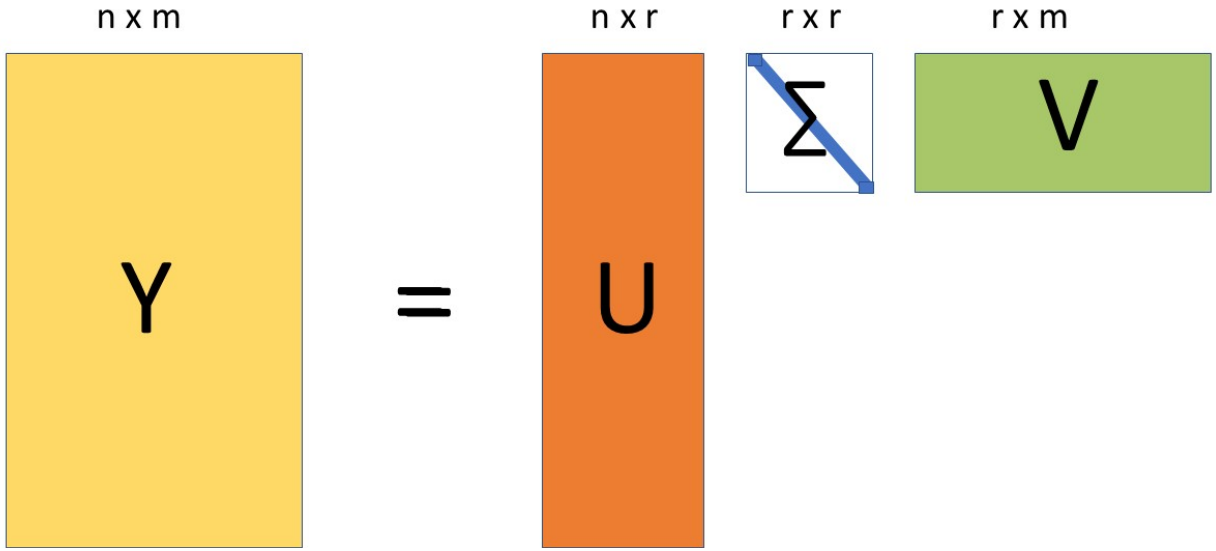



[31] He et al., Neural collaborative filtering, WWW 2017

# When is it useful to construct target representations?

- **Theorem:** Singular Value Decomposition

  Any $n \times m$ matrix $Y$ can be decomposed as follows:




- $\sigma_1, \sigma_2, ...$: singular values of $Y$
- $r =$ Rank of $Y =$ number of non-zero singular values
- High rank when a lot of singular values differ from zero
- Low rank when a lot of singular values are zero
- Singular values give insight in what can be gained

# Low-dimensional target representations in Setting B

- As before, consider a linear model for every target:

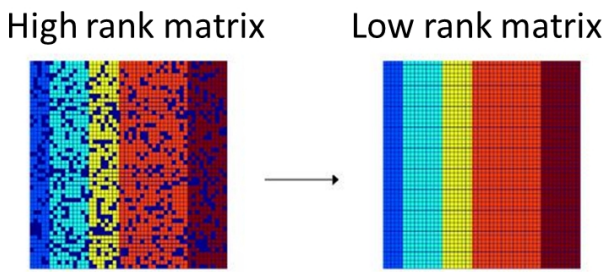
$$f_j(\boldsymbol{x}_i) = \boldsymbol{a}_j^\intercal \boldsymbol{x}_i$$

- Can be written as a linear transformation:

$$\boldsymbol{f}(\boldsymbol{x}) = A\boldsymbol{x} \quad \text{with} \quad A = \begin{bmatrix} \boldsymbol{a}_1^T \\ \vdots \\ \boldsymbol{a}_m^T \end{bmatrix}$$

- Consider a low-rank approximation of the parameter matrix[32]:

$$\min_{\boldsymbol{a}_1,\ldots,\boldsymbol{a}_m} \sum_{i=1}^{n}\sum_{j=1}^{m}(y_{ij} - f_j(\boldsymbol{x}_i))^2 + \lambda \operatorname{rank}(A)$$

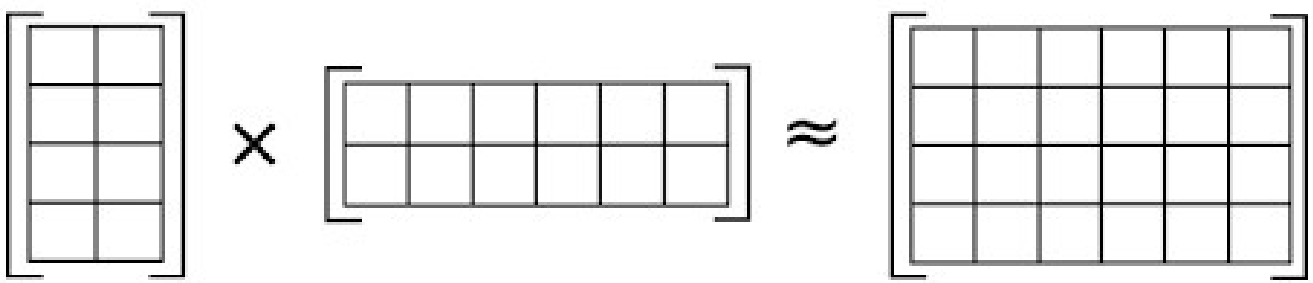



[32] Chen et al., A convex formulation for learning shared structures from multiple tasks, ICML 2009.

# Low-dimensional target representations in Setting B

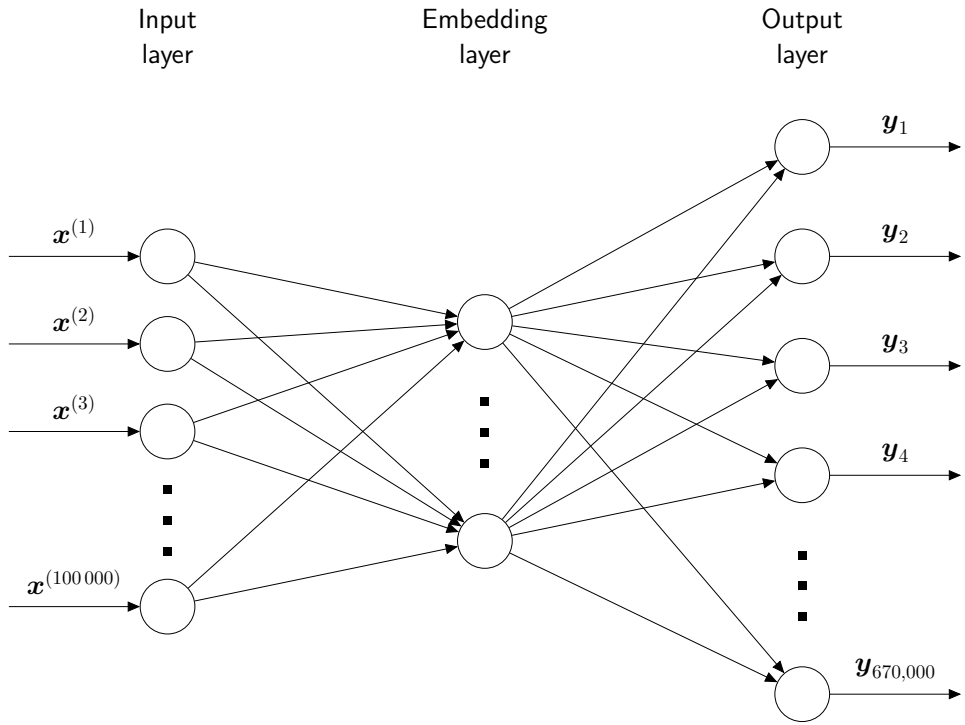
- $A$: parameter matrix of dimensionality $m \times p$
- $p$: the number of features
- $m$: the number of targets
- Assume a low-rank structure of $A$:

$$\underset{(m \times r)}{U} \times \underset{(r \times p)}{V} = \underset{(m \times p)}{A}$$



- $r$ is the rank of $A$
- We can write $A = VU$ and $A\boldsymbol{x} = VU\boldsymbol{x}$

# Applying this principle to neural networks




- Mapping input to output via bottleneck layer
- Nonlinear alternative to $A\boldsymbol{x} = VU\boldsymbol{x}$[33]

[33] Wicker et al., A nonlinear label compression and transformation method for multi-label classification using autoencoders, PAKDD 2016

# Conclusions

- Multi-target prediction is an active field of research that connects different types of machine learning problems
- In the corresponding subfields of machine learning, problems have typically been solved in isolation, without establishing connections between methods
- When analyzing MTP methods, it is important to understand several concepts, such as the influence of loss functions, and the availability and absence of side information

**Further reading:**

W. Waegeman, K. Dembczynski, E. Hüllermeier. Multi-target prediction: A unifying view on problems and methods. Data Mining and Knowledge Discovery, 33(2), 2019.
https://arxiv.org/abs/1809.02352

D. Iliadis, B. De Baets, W. Waegeman, Multi-target prediction for dummies using two-branch neural networks, Machine Learning, to appear.
https://arxiv.org/abs/2104.09967

# Overview of this talk

1. Introduction (10 min)
2. A unifying view on MTP problems (20 min)
3. A unifying view on MTP methods (50 min)
4. Coffee break (30 min)
5. Hands-on part (80 min)

# Setting selection in Multi-Target Prediction

user

+

dataset

**Multi-label classification?**

**Multivariate regression?**

**Hierarchical Multi-label classification?**

**Multi-task learning?**

**Zero-shot learning?**

**Dyadic prediction?**

**Matrix completion?**

**Hybrid Matrix completion?**

# Multi-target prediction for dummies using two-branch neural networks (DeepMTP)

DeepMTP, first step towards a framework that performs setting selection and trains a model in an end-to-end approach.

1. An MTP setting selection step that is based on a custom-made questionnaire.
2. A flexible neural network architecture that can be used for the several subfields of MTP.

# Purpose-made questionnaire

**Q1**: Is it expected to encounter novel instances during testing? **(yes/no)**

**Q2**: Is it expected to encounter novel targets during testing? **(yes/no)**

**Q3**: Is there side information available for the instances? **(yes/no)**

**Q4**: Is there side information available for the targets? **(yes/no)**

**Q5**: Is the score matrix fully observed? **(yes/no)**

**Q6**: What is the type of the target variable?
**(binary/nominal/ordinal/real-valued)**

# Multi-label classification (document categorization)

| | | Tennis | Football | Biking | Movies | TV | Belgium |
|---|---|---|---|---|---|---|---|
| 01101 | Text_1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 00111 | Text_2 | 1 | 0 | 0 | 0 | 0 | 1 |
| 01110 | Text_3 | 0 | 0 | 0 | 1 | 1 | 0 |
| 10001 | Text_4 | 0 | 0 | 1 | 0 | 1 | 0 |
| 01011 | Text_5 | 1 | 0 | 0 | 1 | 0 | 0 |
| 11110 | Text_6 | ? | ? | ? | ? | ? | ? |

**Q1**: **yes**

**Q2**: **no**

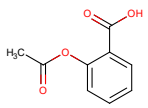
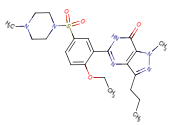
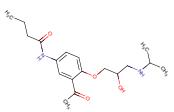
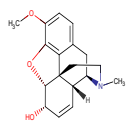
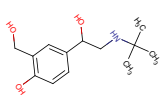
**Q3**: **yes**

**Q4**: **no**

**Q5**: **yes**

**Q6**: **binary**

# Multivariate regression (protein-ligand interaction prediction)

| | Mol1 | Mol2 | Mol3 | Mol4 | Mol5 | Mol6 |
|---|---|---|---|---|---|---|
|  | 1.3 | 0.2 | 1.4 | 1.7 | 3.5 | 1.3 |
|  | 2 | 1.7 | 1.5 | 7.5 | 8.2 | 7.6 |
|  | 0.2 | 0 | 0.3 | 0.4 | 1.2 | 2.2 |
|  | 3.1 | 1.1 | 1.3 | 1.1 | 1.7 | 5.2 |
|  | ? | ? | ? | ? | ? | ? |

**Q1**: yes

**Q2**: no

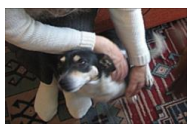
**Q3**: yes

**Q4**: no

**Q5**: yes

**Q6**: real-valued

# Multi-task Learning (crowd-sourced annotation)

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
|  | 1 | | 0 | 0 | 1 | | |
|  | | 0 | 1 | 0 | | 0 | 1 |
|  | 0 | 1 | 1 | 1 | 1 | | 0 |
|  | | 1 | | | 0 | 1 | 1 |
|  | ? | ? | ? | ? | ? | ? | ? |

**Q1**: yes

**Q2**: no

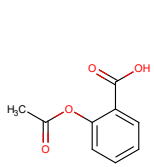
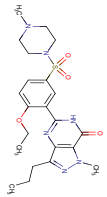
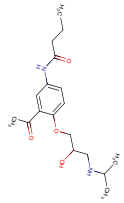
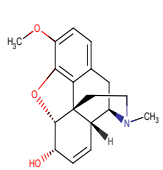
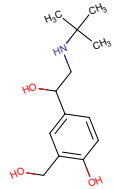
**Q3**: yes

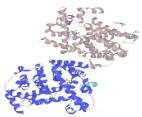
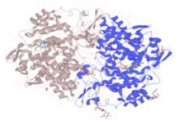
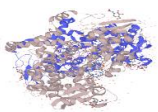
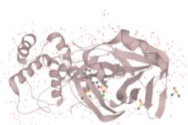
**Q4**: no

**Q5**: no

**Q6**: binary/ real-valued

# Dyadic prediction (protein-ligand interaction prediction)

| | | | | | |
|---|---|---|---|---|---|
|  | 1.3 | 0.2 | 0.9 | 1.9 | 3.1 |
|  | 1.7 | 5.1 | 0.3 | 0.1 | 1.1 |
|  | 0.1 | 2.2 | 0.1 | 0.5 | 4.2 |
|  | 0.9 | 1.4 | 1.6 | 0.2 | 0.8 |
|  | ? | ? | ? | ? | ? |

**Q1**: yes

**Q2**: no

**Q3**: yes

**Q4**: yes

**Q5**: no

**Q6**: binary/ real-valued

# Matrix Completion (Content recommendation)


| | | | | | |
|---|---|---|---|---|---|
| 1 | 8 | 5 | ? | 10 | ? |
| 2 | 4 | ? | 8 | ? | 3 |
| 3 | ? | 7 | 2 | ? | 5 |
| 4 | 1 | ? | ? | 7 | ? |
| 5 | 5 | ? | 5 | ? | ? |

**Q1**: **no**

**Q2**: **no**

**Q3**: **no**

**Q4**: **no**

**Q5**: **no**

**Q6**: **binary/ real-valued**

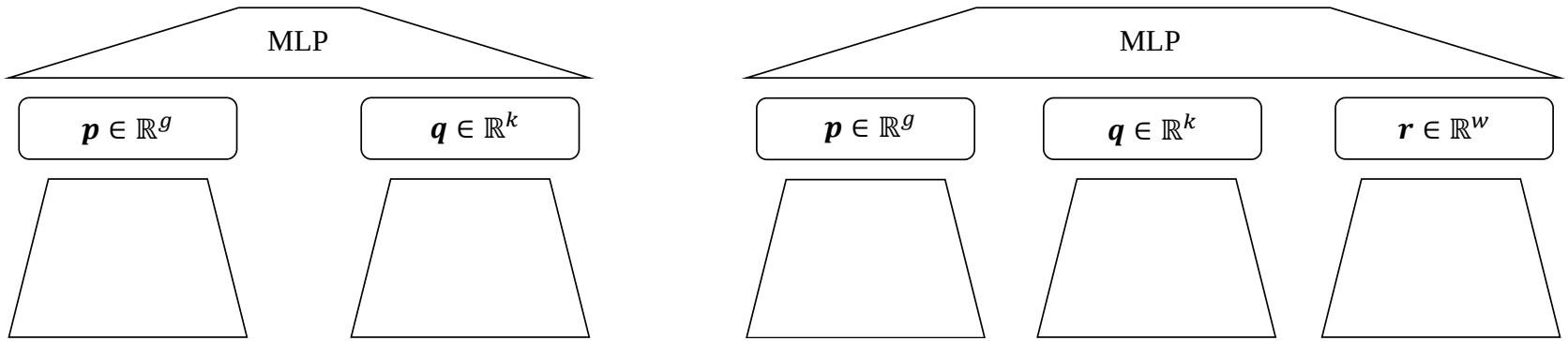
# Setting selection in Multi-Target Prediction

| Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | MTP method |
|---|---|---|---|---|---|---|
| yes | no | yes | no | yes | binary | Multi-label classification |
| yes | no | yes | no | yes | real-valued | Multivariate regression |
| yes | no | yes | no | no | - | Multi-task learning |
| yes | no | yes | yes (hierarchy) | yes | binary | Hierarchical Multi-label classification |
| yes | no | yes | yes | no | - | Dyadic prediction |
| yes | yes | yes | yes | no | - | Zero-shot learning |
| no | no | no | no | no | - | Matrix completion |
| no | no | yes | yes | no | - | Hybrid Matrix completion |
| yes | yes | yes | yes | no | - | Cold-start Collaborative filtering |
| yes | no | yes | no | yes | nominal/categorical | Multi-dimensional classification |

- These questions generate 128 different combinations
- Most of them lead to impossible tasks
- **Simple rule:** generalization to novel instances or targets necessitates the corresponding side information.

# A flexible neural network architecture

Popularized by the neural collaborative filtering (NCF)[34] method in the field of recommender systems

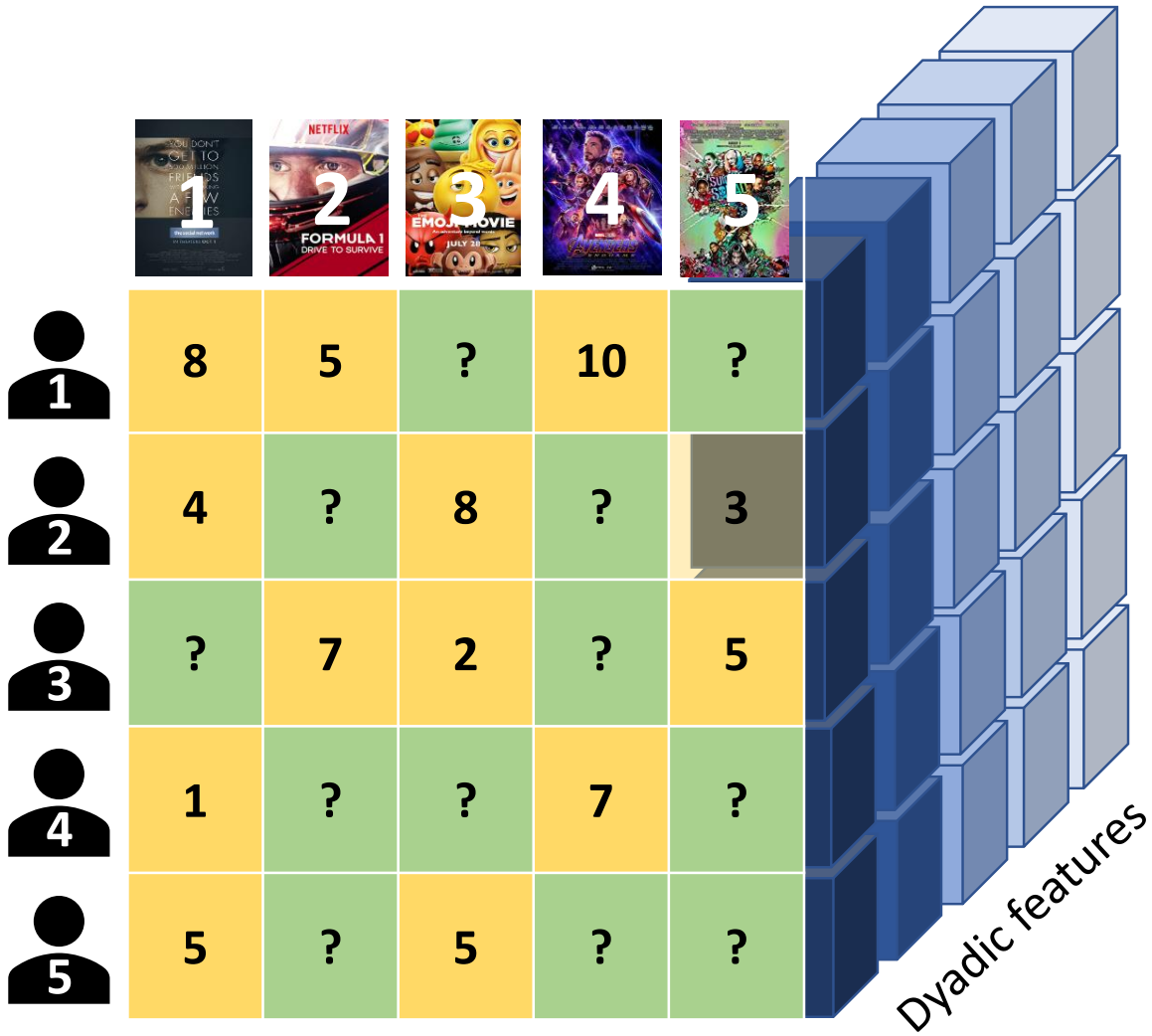



The multi-branch architecture has two versions:

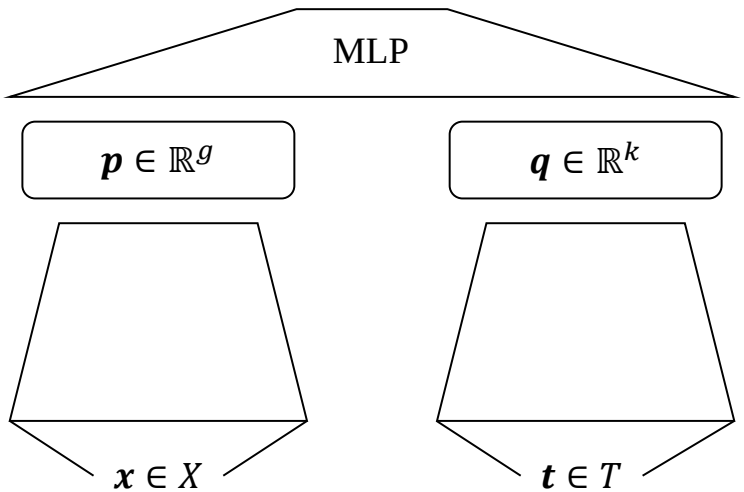
1. The dual-branch architecture has to input branches (instances and targets).
2. The tri-branch architecture adds a third input for any available dynamic side information.

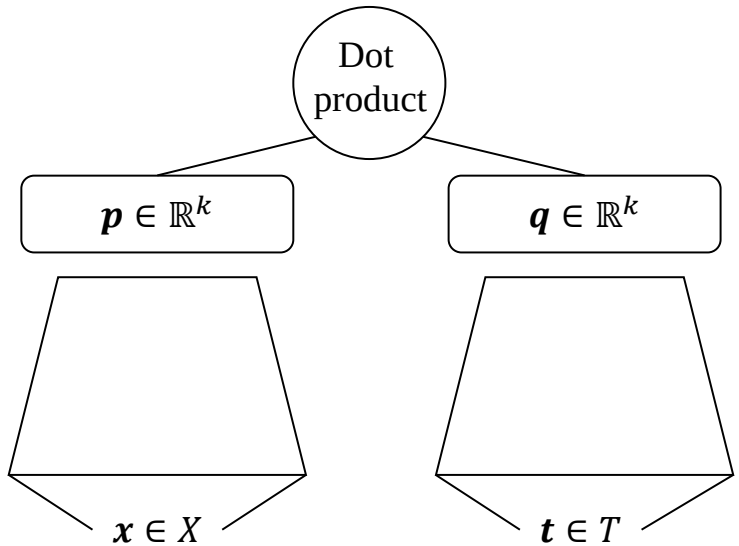
[34] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering.In: Proceedings of the 26th International Conference on World Wide Web, pp. 173–182(2017)

# Matrix completion with Dyadic features



# Combining embedding vectors

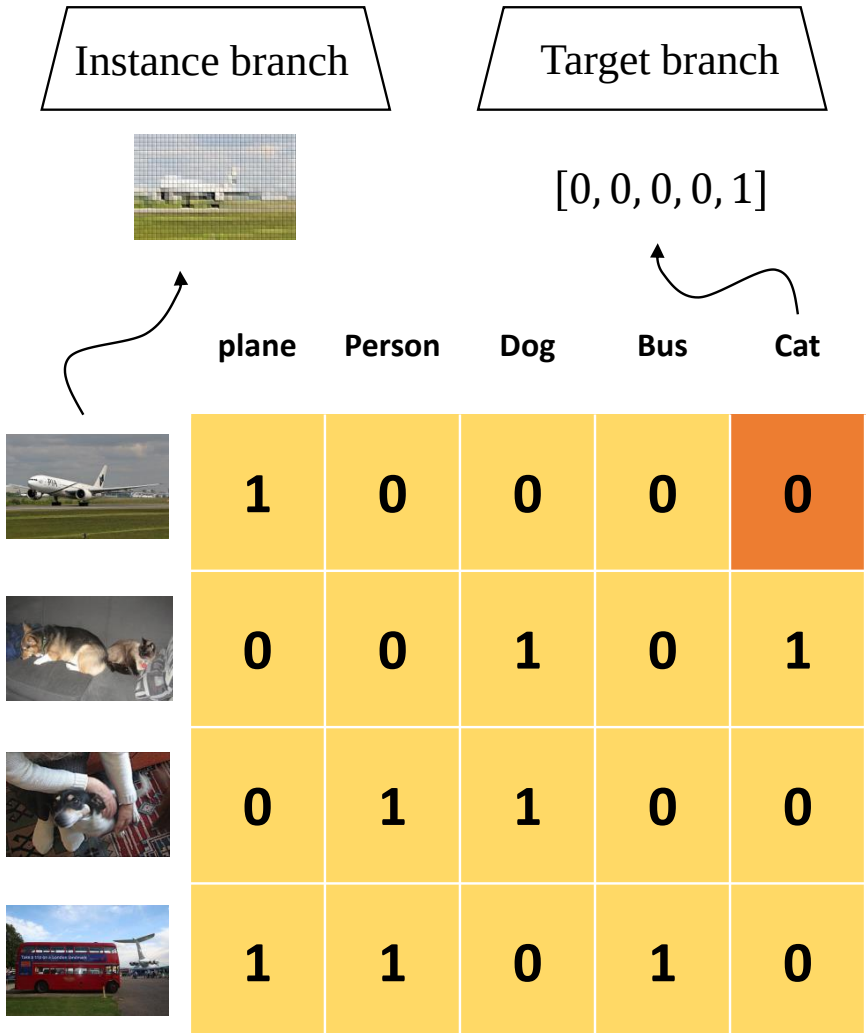


# A flexible neural network architecture 2

Depending on the availability of side information:

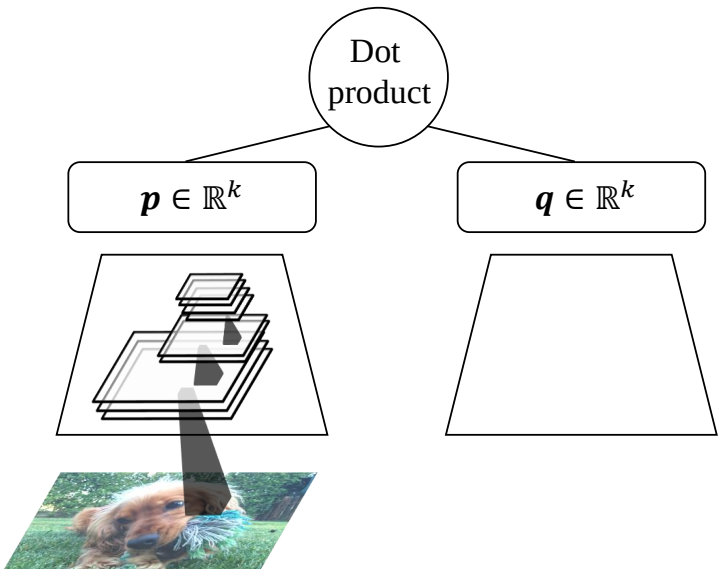
1. If side information is available: use it...
2. If side information is missing: create one-hot encoded vectors.

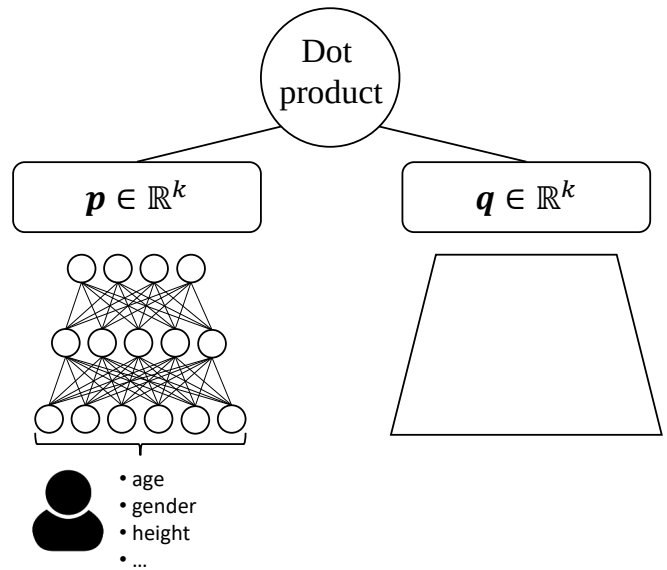


# A flexible neural network architecture 3

Depending on the type of side information

1. tabular data: fully connected layers.
2. images: convolutional neural network.
3. time-series: RNNs, LSTMs..



# A flexible neural network architecture 4

Depending on the type of target variable:

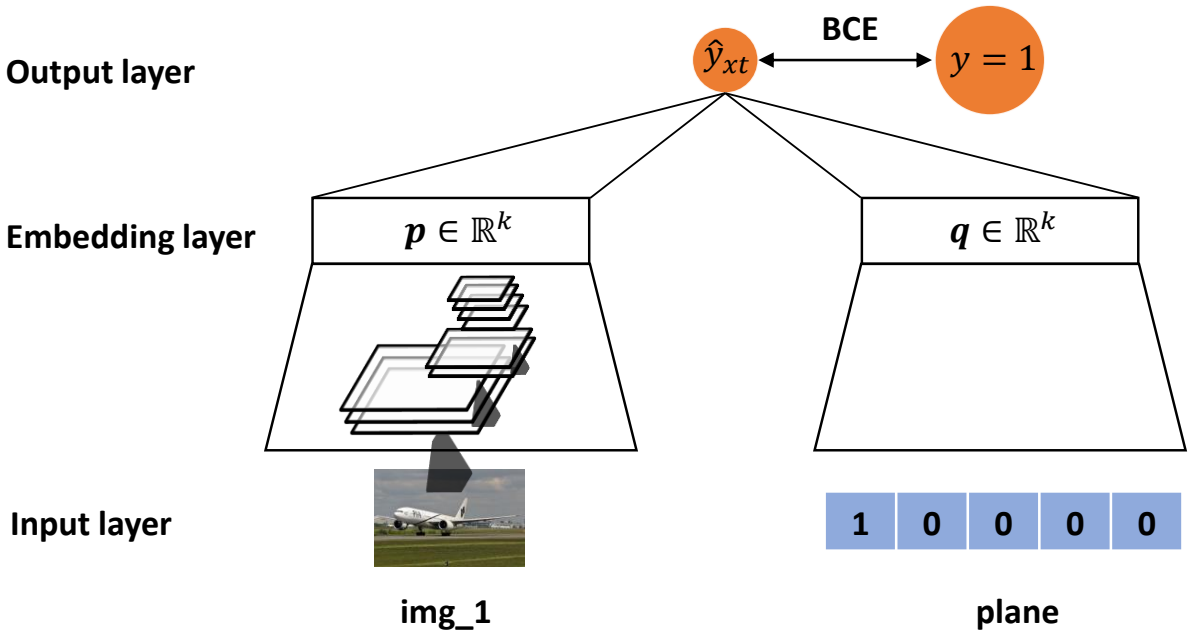
1. if the target variable is binary: use BCE

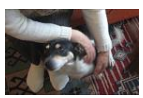
$$\min_{\boldsymbol{a}_1,\ldots,\boldsymbol{a}_m,\boldsymbol{\theta}} \sum_{i=1}^{n}\sum_{j=1}^{m} CE(y_{ij}, p_j(\boldsymbol{x}_i)) \tag{1}$$

2. if the target variable is real-valued: use MSE

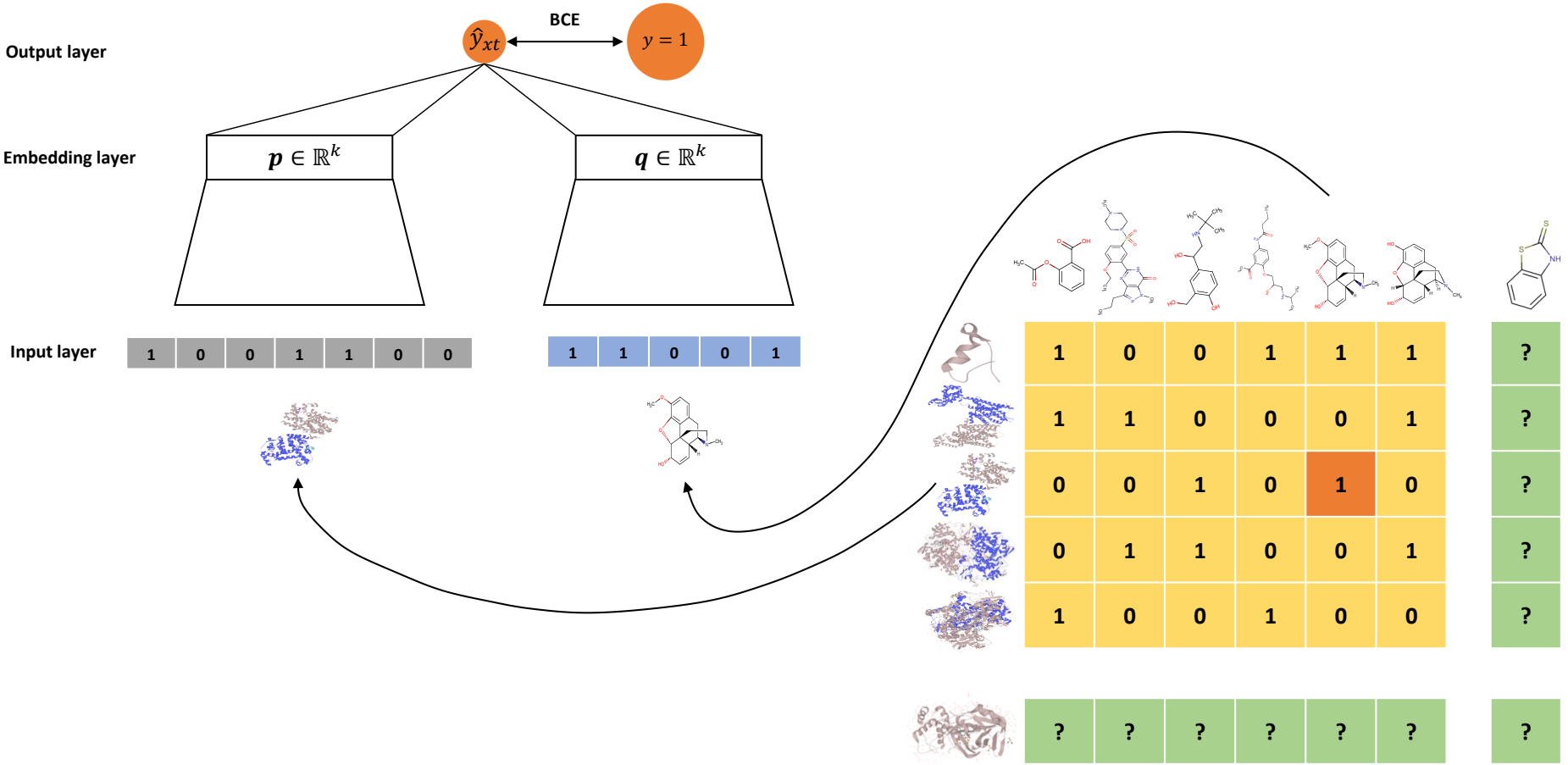
$$\min_{\boldsymbol{a}_1,\ldots,\boldsymbol{a}_m,\boldsymbol{\theta}} \sum_{i=1}^{n}\sum_{j=1}^{m} (y_{ij} - f_j(\boldsymbol{x}_i))^2 \tag{2}$$

# A closer look

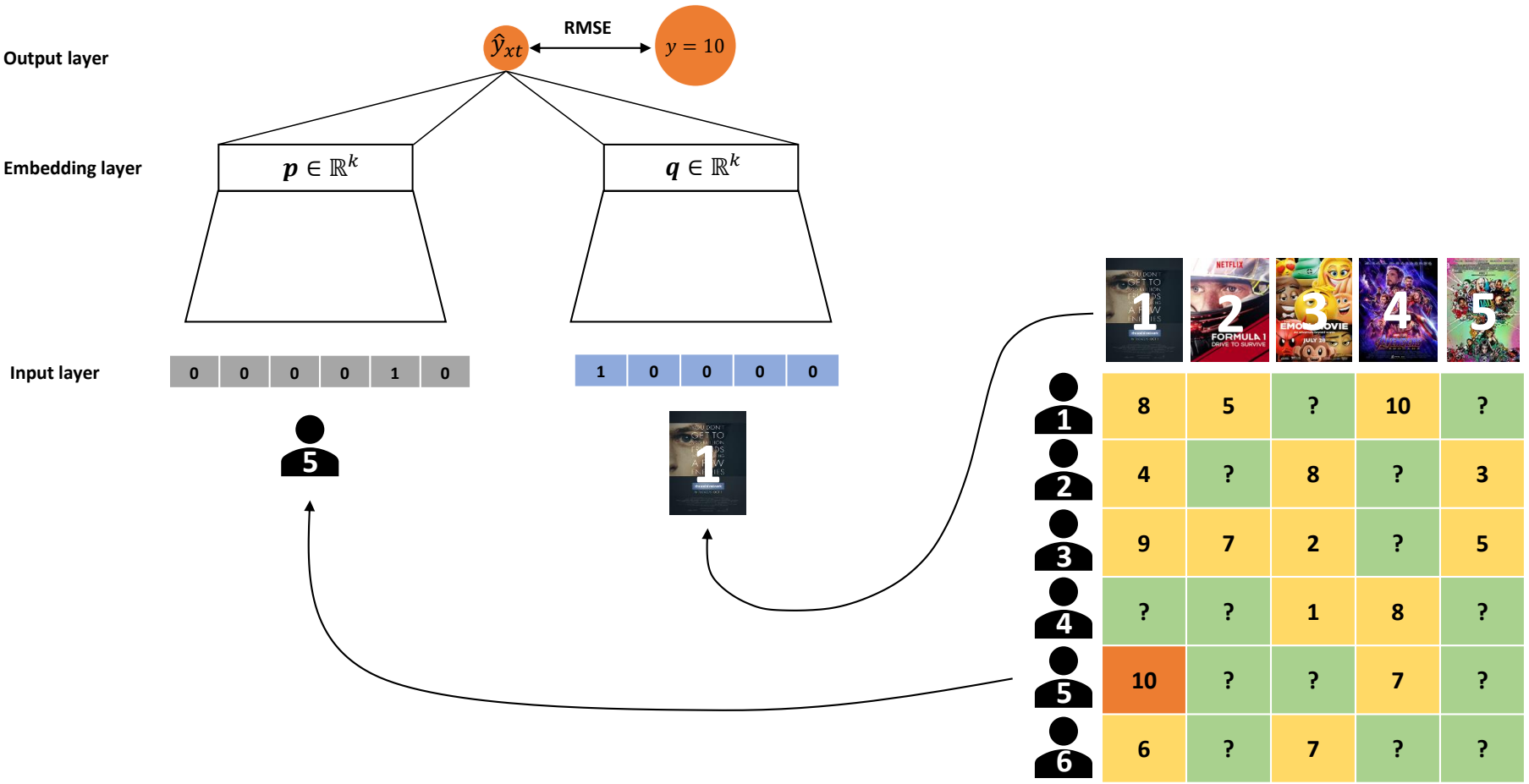



| | | plane | person | dog | bus | cat |
|---|---|---|---|---|---|---|
|  | img_1 | 1 | 0 | 0 | 0 | 0 |
|  | img_2 | 0 | 0 | 1 | 0 | 1 |
|  | img_3 | 0 | 1 | 1 | 0 | 0 |
|  | img_4 | 1 | 0 | 0 | 1 | 0 |

# A closer look (Dyadic Prediction)



# A closer look (Matrix Completion)



# Validating the framework

To investigate the effectiveness of the proposed architecture we experimented with multiple datasets from various MTP problem settings [35]

- Multi-label classification
  - Yeast
  - Scene
  - Bibtex
  - Corel5k
- Multivariate regression
  - Enb
  - Jura
  - Water quality
  - Oes97
  - Oes10
  - Puma8NH
  - Puma32H

| **Hamming loss** | **BR(SVM)** | **MLP** | **DeepMTP** |
|---|---|---|---|
| Yeast | 0.1935 | 0.2406 | 0.2309 |
| Bibtex | 0.0130 | 0.0198 | 0.0157 |

| **aRRMSE** | **SVR/target** | **MLP** | **DeepMTP** |
|---|---|---|---|
| Enb | 0.1161 | 0.0933 | 0.0954 |
| Oes97 | 0.5394 | 0.7885 | 0.4843 |
| Puma32H | 0.9634 | 1.0008 | 1.0002 |

[35] Iliadis et al. Multi-target prediction for dummies using two-branch neural networks. Machine Learning Journal, 2021

# Validating the framework 2

- Hierarchical multi-label classification
  - VOC2007
  - MS COCO
- Matrix completion
  - Movielens 100k, 1M
- Multi-task learning
  - Dogs, Birds (crowdsourced annotation)
- Dyadic prediction
  - DPI-E
  - DPI-IC
  - SRN
  - ERN

| **micro-AUC** | **eBICT** | **DeepMTP** |
|---|---|---|
| DPI-E | 0.8053 | 0.8571 |
| SRN | 0.8169 | 0.8166 |
| ERN | 0.8536 | 0.8874 |

# DeepMTP in practice

- A tutorial on DeepMTP is available in the following google colab notebook: `https://colab.research.google.com/drive/1jc8z10_0lDcsJtsdpshegaWltGon4V1i?usp=sharing`

# Questions? Remarks?